

Beckhoff logiikan- ja servomotto-ohjauksen oppimisympäristön
suunnittelu, toteutus ja käyttöönotto.

Janne Urpunen

Teollisuuden ja luonnonvarojen opinnäytetyö
Sähkötekniikan koulutusohjelma
Insinööri (AMK)

KEMI 2014

ALKUSANAT

Tahtoisin kiittää insinööri Aila Petäjäjärveä opinnäytetyön ohjauksesta, sekä laboratorioinsinöörejä Jouko Alanivaa ja Juha-Matti Kvistiä kaikesta avusta. Erityiskiitokset myös Lappian ammattiopiston Timo Jokelaiselle asennusalustojen työstämisestä.

Janne Urpunen

Kemi, 31.10.2014

TIIVISTELMÄ

LAPIN AMMATTIKORKEAKOULU, Teollisuuden ja luonnonvarojen osaamisala

Koulutusohjelma:	Sähkötekniikan koulutusohjelma
Opinnäytetyön tekijä:	Janne Urpunen
Opinnäytetyön nimi:	Beckhoff logiikan ja servomoottoriohjauksen oppimisympäristön suunnittelu, toteutus, ja käyttöönotto.
Sivuja (joista liitesivuja):	62 (22)
Päiväys:	7.11.2014
Opinnäytetyön ohjaaja:	Ins. Aila Petäjäjärvi
<p>Opinnäytetyön aiheena ja tavoitteena oli suunnitella, toteuttaa, ja ottaa käyttöön oppimisympäristö, jossa käyttäjät voivat tutustua servomoottorien toimintaan, käyttöön sekä ohjaukseen ohjelmoitavan logiikan kautta.</p> <p>Teoriaosuudessa käsitellään opinnäytetyön kokonaisuuteen liittyviä osia ja niiden toimintaa. Lisäksi osuudessa esitellään oppimisympäristön laitteisto sekä laitteiston valmistaja. Laitteiston pohjalta tulee suunnitella oppimisympäristö ja toteuttaa se.</p> <p>Aineistona opinnäytetyössä käytettiin ensisijaisesti laitteiston valmistajien omia dokumentteja, koska opinnäytetyön tärkeimpänä tuotoksena on itse oppimisympäristön toteuttaminen. Oppimisympäristöä varten tuotettiin suunnitelma siitä, kuinka laitteisto sijoitetaan ja kuinka käyttäjä tulee ohjaamaan sitä. Tarvittavat osat tilattiin ja hankittiin suunnitelmien pohjalta. Lopuksi ympäristö koottiin, johdotettiin, sekä testattiin.</p> <p>Lopputuloksena saatiin laitteistosta koottua oppimisympäristö ja testattua sen osien toimivuus. Oppimisympäristö testattiin luomalla testiohjelma logiikalle. Tämän ohjelman luonnin yhteydessä kirjoitettiin myös dokumentaatio, jonka avulla käyttäjä pääsee nopeasti alkuun laitteiston kanssa.</p>	
Asiasanat: Servotekniikka, oppimisympäristö, sähkömoottorit, ohjelmoitavat logiikat	

ABSTRACT

LAPLAND UNIVERSITY OF APPLIED SCIENCE, Industry and natural resources

Degree program:	Degree of electrical engineering
Author:	Janne Urpunen
Thesis title:	Design, Execution, and Realization of Learning Environment Using Beckhoff's PLC and Servomotor Drive.
Pages (of which appendixes)	62 (22)
Date:	7 November 2014
Thesis Instructor:	Aila Petäjäjärvi, B.Sc
<p>Subject of this thesis was to design, execute, and realize a learning environment where the users can familiarize themselves with how the servomotor functions and how to operate and control one by using a programmable logic controller.</p> <p>Theory segment of this thesis covers all the parts related to the learning environment. This segment also presents all the hardware used in the environment. Purpose of this work is to design a learning environment based on the hardware and realize it.</p> <p>The primary source for documentation in this thesis was the material provided by the hardware manufacturers, as the main end-product of this thesis is the practical work to deliver a finished learning environment. Designs were produced on how to position the hardware on to this learning environment and how the user would control it. Necessary parts were ordered and acquired according these plans. Lastly, the environment was assembled, its circuitry was wired and then it was tested afterwards.</p> <p>As an end result the hardware was assembled in to a proper learning environment and the functionality of its parts was tested. The learning environment was tested by creating a suitable testing program for the programmable logic controller. Documents for helping the end-user to quickly get started with the hardware and programming environment were made alongside this program.</p>	
Key words: Servomechanism, learning environment, electric motors, programmable logic controller	

SISÄLLYS

ALKUSANAT	2
TIIVISTELMÄ	3
ABSTRACT	4
SISÄLLYS	5
KÄYTETYT MERKIT JA LYHENTEET	7
1 JOHDANTO	8
2 OHJELMOITAVAT LOGIIKAT	9
2.1 Historia.....	9
2.2 IEC 61131-3 Standardi.....	9
2.3 Ohjelmointikielet.....	9
2.3.1 Instruction List	10
2.3.2 Ladder Diagram	10
2.3.3 Function Block Diagram.....	11
2.3.4 Sequential Function Chart.....	11
2.3.5 Structured Text.....	12
2.3.6 Continous Function Chart	12
3 SERVOTEKNIikka.....	13
3.1 Servomoottorit	13
3.2 Resolverin ja enkooderin toiminta	14
3.2.1 Resolverin rakenne.....	15
3.2.2 Enkooderin rakenne	15
3.3 Inkrementaalisen- ja absoluuttisen anturin eroavaisuudet	16
4 TAKAISINKYTKENTÄ	17
4.1 Positiivinen ja negatiivinen takaisinkytkentä.....	17
4.2 PID-säätö.....	18
5 TWINCAT	19
5.1 TwinCATin ominaisuudet.....	19
5.2 TwinCAT ohjelmointiympäristö	19
6 LAITTEISTO.....	21
6.1 Beckhoff Automation GmbH.....	21
6.1.1 Beckhoff yrityksenä.....	21
6.1.2 Beckhoff Suomessa.....	22

6.2 Sulautettu-PC, CX5010.....	22
6.3 Servovahvistin, AX5201	22
6.4 Servomootorit, AM8021-0B20	23
6.5 EtherCAT	23
6.6 Kone- ja laiteturvallisuus	24
7 SUUNNITTELU	25
7.1 Oppimisympäristön vaatimukset.....	25
7.2 Oppimisympäristön suunnittelu	26
7.3 Suunnittelun ongelmakohdat.....	27
7.4 Servomootorin käyttökohteen suunnittelu	28
7.5 Laitteiston ohjauksen suunnittelu.....	29
7.6 Johdotuksen suunnittelu	29
8 TOTEUTUS	31
8.1 Kone- ja laiteturvallisuus käytännössä.....	31
8.2 Logiikka, laitteiston esitestaus	32
8.3 Asennuskehikon kokoaminen	32
8.4 Laitteiston kokoaminen asennuskehikkoon	33
8.5 Logiikan ja muun laitteiston johdotus.....	35
8.4 Ohjelmointi ja dokumentointi	36
9 POHDINTA	36
9.1 Suunnittelun tärkeys.....	36
9.2 Servomootorien käyttö	37
9.3 Oppimisympäristön muokattavuus	37
LÄHTEET	39
LIITTEET	40

KÄYTETYT MERKIT JA LYHENTEET

Beckhoff	Bechhoff Automation GmbH
CE	Conformité Européenne, European Conformity
CFC	Continous function chart
EU	Euroopan unioni
FBD	Function block diagram
I/O	Input/output
IEC	International Electrotechnical Commission
IL	Instruction list
LD	Ladder diagram
NC	Numerical control, numeerinen ohjaus
OCT	One cable technology
PC	Personal computer, tietokone
PID	Proportional-intergral-derivative
PLC	Programmable logic controller, ohjelmoitava logiikka
SFC	Sequential function chart
ST	Structured text

1 JOHDANTO

Tässä opinnäytetyössä tavoitteena on luoda servomoottorioppimisympäristö, jossa käyttäjät voivat opiskella servomoottorin käyttöä, ohjausta, ja ohjelmointia. Oppimisympäristössä hyödynnetään Beckhoff Automation-yrityksen laitteistoa. Tämän yrityksen laitteistoa ei ole vielä muussa käytössä oppilaitoksessa. Laitteistona toimii ohjelmoitava logiikka, servovahvistin sekä kaksi servomoottoria. Oppimisympäristöä ei sijoiteta kiinteästi mihinkään tilaan, joten sen tulisi olla kuljetettavissa käsivoimin.

Tämän opinnäytetyön aihe valittiin mielenkiinnosta servomoottoreihin ja servomoottorikäytön toteuttamiseen. Oppimisympäristön luomisessa oli lähes vapaat kädet. Toiveena oli, että ympäristö olisi kannettavaa pöytämallia ja että käyttäjien olisi mahdollista oppia ympäristöllä servomoottoreiden käytöstä sen mukana toimitettavan dokumentaation ja ohjelman avulla.

Oppimisympäristön laitteistolle tulee suunnitella sopivat asennusalustat ja niitä varten asennuskehikko. Oppimisympäristön johdotus ja sähköinen toiminta tulee myös suunnitella ja testata.

Käyttäjille laaditaan ohjeet oppimisympäristön toiminnasta. Ohjeilla on myös tarkoitus opastaa käyttäjää pääsemään alkuun TwinCAT 3 ohjelmoinnin kanssa.

2 OHJELMOITAVAT LOGIIKAT

2.1 Historia

Ohjelmoitavat logiikat, (Programmable logic controller, PLC). Ensimmäisen ohjelmoitavan logiikan suunnittelu aloitettiin 60-luvun lopulla ja se johti ensimmäisen kaupallisen logiikan nimeltä MODICON 084 valmistamiseen. Tärkeimpänä perustana ohjelmoitavien logiikoiden suunnitteluun oli monimutkaisten relepohjaisten järjestelmien korvaaminen halvemmalla ja moneen käyttökohteeseen muunnettavissa olevalla vaihtoehdolla. Nykyään ohjelmoitavat logiikat ovat yksi tärkeimmistä automaatiotekniikan osista. (Hendricks, Morley 2002, Hakupäivä 3.11.2014.)

2.2 IEC 61131-3 Standardi

Ennen logiikoiden yleistymistä käytettiin valmistajasta riippuen useita eri ohjelmointikieliä ja jännite- ja virtaviestejä. Vasta logiikoiden yleistyessä alettiin laitteistojen välille kaivata yhtenevää standardia. Vuonna 1993 painettiin ensi kertaa IEC 61131-standardi ohjelmoitavien logiikoiden yhdenmukaistamiseksi.

IEC 61131-standardi koostuu kahdeksasta osasta, ja niistä tärkein ohjelmointia koskeva osa on IEC 61131-3, joka käsittelee ohjelmointikieliä. Kyseisessä standardin osassa määritellään yhtenäiset ohjelmointikielet joita voidaan käyttää ohjelmoimaan kaikkia standardin mukaisia ohjelmoitavia logiikoita. Etuna standardin mukaistamisessa on mahdollisuus hyödyntää jo kertaalleen tehtyä koodia uudestaan eri laitteistolla säästäten näin aikaa tulevissa projekteissa. Myös tarve eri ohjelmointikielten koulutuksille vähenee, eikä käyttäjien tarvitse huolehtia eri valmistajien laitteiden yhteensopivuudesta.

2.3 Ohjelmointikielet

IEC 61131-3-standardi koostuu viidestä ohjelmointikielestä. Kahdesta tekstipohjaisesta, Instruction List (IL), Structured Text (ST), sekä kolmesta graafisesta kielestä, Ladder Diagram (LD), Function Block Diagram (FBD), sekä Sequential Function Chart (SFC). Ohjelmointikieli itsessään ei vaikuta ohjelman toimintaan, ainoastaan sen esittämiseen.

Monet ohjelmointiympäristöt antavat mahdollisuuden käyttää eri kieliä ohjelman eri toimilohkoissa, tai jopa vaihtaa kielten välillä lennosta. Tämä tarkoittaa, että voidaan valita kieli joka parhaiten sopii ohjelman luomiseen ja havainnollistamiseen.

2.3.1 Instruction List

Rakenteeltaan Instruction List on erittäin lähellä Assembly-tyylisiä ohjelmointikieliä. Instruction list tunnetaan Siemensin logiikoissa nimellä Statement List. Instruction List kielessä on vain yksi komento riviä kohden, ja sen perässä muuttuja tai arvo (kuva 1).

```

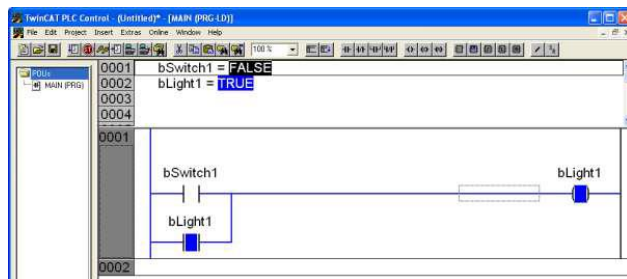
MAIN (PRG-IL)
0001 PROGRAM MAIN
0002 VAR
0003   Speed: INT;
0004 END_VAR
0005
0001 LD   Speed
0002 ADD  5
0003 ST   Speed
0004

```

Kuva 1: Esimerkki Instruction List esitystavasta (Beckhoff Automation 2013).

2.3.2 Ladder Diagram

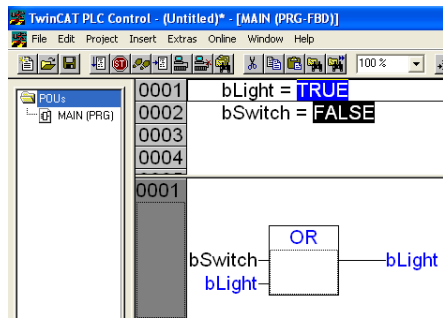
Tikapuukaavio eli Ladder Diagram on yksi ensimmäisistä käyttöön otetuista logiikoiden ohjelmointikielistä. Se on aikoinaan luotu helpottamaan logiikan ohjelmien ymmärtämistä laitosten relejärjestelmien asentajille, sillä logiikka oli luotu korvaamaan nämä vanhat järjestelmät. Nimensä mukaiset tikapuut tulevat reunoilla sijaitsevista jännitekojoja esittävistä pystypalkeista. Ulkonäöltään ohjelmointikieli on suunniteltu muistuttamaan relekaavioita (kuva 2).



Kuva 2: Esimerkki pitopiiristä tikapuukaaviona (Beckhoff Automation 2013).

2.3.3 Function Block Diagram

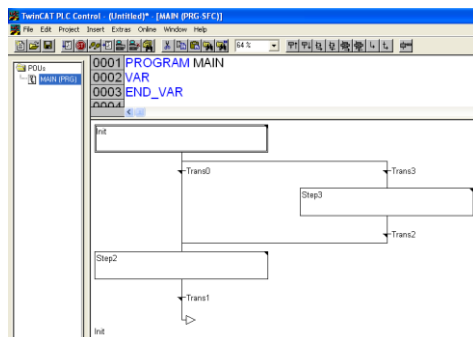
Toimilohkokaavioesitys eli Function Block Diagram on graafinen kieli, ja se on ulkoiseltaan lähempänä elektroniikan piirikaavioita, joissa komponentit kuvataan piirrosmerkein. Toimilohkoilla voidaan kuvata tavallisia loogisia portteja kuten AND ja OR (kuva 3), tai monimutkaisempia toimintoja, kuten laskureita tai ajastimia.



Kuva 3: Pitopiiri toimilohkokaaviossa (Beckhoff Automation 2013).

2.3.4 Sequential Function Chart

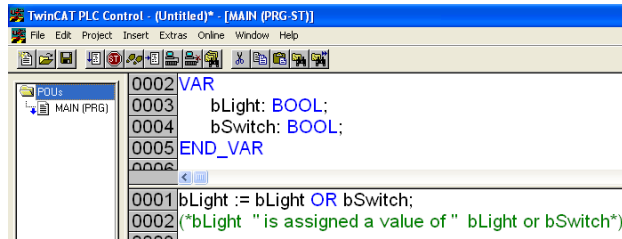
Vaikka SFC on määritelty ohjelmointikieleksi, se on paremminkin tapa järjestellä koodia ja ohjata sen toimintaa. Jokainen askel ja siirtymä SFC:ssä voidaan ohjelmoida käyttämällä muita standardin ohjelmointikieliä. Myös askeleet ja siirtymät voivat sisältää uusia SFC-tasoja. Tämä tekee ohjelmointikielestä visuaalisesti yksinkertaisen (kuva 4), näin helpottaen suuren ohjelman jäsentämistä ja lukemista.



Kuva 4: SFC näkymä. Jokainen lohko sisältää lisää ohjelmakoodia (Beckhoff Automation 2013).

2.3.5 Structured Text

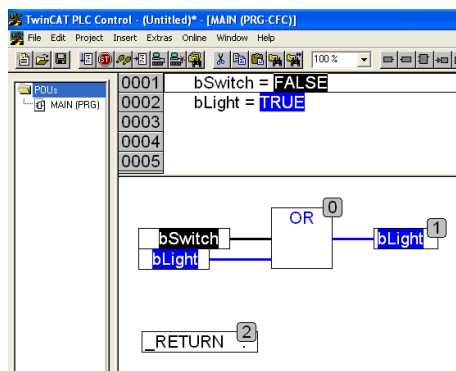
Structured Text on Pascal-ohjelmointikielen pohjalta luotu korkean tason ohjelmointikieli. Standardin kielistä ST on tehokkain ja monipuolisin. Ohjelmoijan on mahdollista esittää vähin koodirivein se, mihin vaadittaisiin useita toimilohkoja FBD-ohjelmointikielessä (kuva 5).



Kuva 5: Kuvan 3 pitopiiri on esitetty yhdellä ohjelmarivillä. Alemmalla rivillä ohjelmoijan lisäämä kommenttirivi vihreällä (Beckhoff Automation 2013).

2.3.6 Continous Function Chart

TwinCAT ohjelmointiympäristöt sisältävät tavallisten IEC 61131-3-standardin mukaisen ohjelmointikielten lisäksi Beckhoffin oman ”Continous Function Chart” kielensä. CFC on graafinen kieli joka on ulkoasultaan hyvin lähellä FBD-ohjelmointikieltä. CFC-Ohjelmointikielen omana erikoisuutena ovat lohkojen yhteydessä näkyvät numerot, joilla ilmaistaan suoritettavien komentojen järjestys (kuva 6). Ohjelmoijan on mahdollista vaikuttaa tähän järjestykseen tarvittaessa.



Kuva 6: CFC-ohjelmointikieli, jossa numerot ilmaisevat suoritusten prioriteetin (Beckhoff Automation 2013).

3 SERVOTEKNIikka

Servotekniikassa säädetään dynaamisesti jonkun suureen tilaa tarkasti ja nopeasti. Tällaisia suureita ovat esimerkiksi nopeus, asento, vääntömomentti, voima tai kiihtyvyyys. Servojärjestelmä koostuu usein moottorista, servovahvistimesta, servo-ohjaimesta ja takaisinkytkennästä. Yleisimmin käytettyjä servoja ovat asema- ja nopeusservot. Järjestelmässä on yleensä jonkinlainen käyttöliittymä, jonka avulla järjestelmään voidaan syöttää haluttuja parametreja. (Pesu 2010; Fonselius 1998, 7-11.)

Digitaalisissa järjestelmissä servo-ohjain suorittaa säätöarvojen laskennan signaaliprosessorilla. Takaisinkytkennät tuodaan servo-ohjaimelle. Signaaliprosessori hoitaa takaisinkytkentäarvojen ja ohjearvojen vertailun ja laskee tarvittavat arvot moottorin säätämiseksi. Nämä arvot lähetetään servovahvistimelle, joka syöttää moottorille jännitettä ja virtaa säätöarvojen mukaisesti. Säättöjärjestelmän tehtävänä on pitää erosiignaali nollassa, tällöin ohjesignaali on sama kuin takaisinkytkentäsignaali. Järjestelmä toimii tällöin ohjeen mukaisesti. (Pesu 2010; Fonselius 1998, 7-11.)

3.1 Servomoottorit

Servomoottoreiksi kutsutaan sähkömoottoreita, joissa on sisäänrakennettu servo-ohjain. Servomoottorit eivät silti ole oma moottoreiden alatyypinsä, vaan kyseessä on useimmiten vain moottori jota ohjataan servotekniikalla. Tämän ohjaimen avulla moottorin on mahdollista saada takaisinkytkennän kautta tietoa moottorin akselin asemasta, ja ohjata toimintaansa sen mukaan. Servoa käytetään yleisesti puhekielessä sekä itse servoelementistä että servomoottoreista.

Servomoottorit ovat erikoisrakenteisia tasa- tai vaihtovirtamoottoreita. Sähkömoottoreita joiden ominaisuudet soveltuvat takaisinkytkettyyn säätöön, kutsutaan servomoottoreiksi. Servomoottoreiden suurin eroavaisuus muihin moottoreihin on niiden pieni hitausmomentti eli kyky vastustaa pyörimistä. Servomoottorit on suunniteltu toimimaan ylikuumentamatta laajalla nopeusalueella sekä ylläpitämään nollanopeudella riittävän suuri momentti kuorman paikallaan pitämiseksi. (Pesu 2010.)

Servomootorit ovat erittäin hyödyllisiä muun muassa robotiikassa, jossa niiden etuna ovat niiden fyysiset mitat ja kokoonsa nähden tehokas vääntömomentti. Servomoottorien vaatima teho on myös suhteellinen moottorin vaikuttavaan mekaaniseen kuormaan nähden. (Ross 2004, Hakupäivä 3.9.2014.)

Moottorin akseli voidaan ohjata haluttuun asemaan lähettämällä servolle koodisignaalia. Niin kauan kuin signaalia lähetetään servomoottorille, se yrittää pitää asemansa. Kun koodisignaali vaihtuu, myös akselin asema vaihtuu.

Servomootoreissa käytetään anturina moottorin aseman ja nopeuden mittauksessa joko resolveria, jossa tieto saadaan moottorin staattorissa olevien muuntajien ja roottorin välistä magneetikentän vahvuutta mittaamalla, tai resolverin digitaalista vastinetta, enkooderia, jonka viestit saadaan tulkitsemalla valopulsseja. Anturin tieto lähetetään takaisinkytkennän kautta takaisin järjestelmän alkupäähän ja ohjausta muutetaan sen mukaan.

3.2 Resolverin ja enkooderin toiminta

Resolverin ja enkooderin rakenne eroaa toisistaan huomattavasti, mutta molemmilla on sama käyttötarkoitus servomoottorikäytössä, akselin aseman ja nopeuden mittaaminen. Molemmissa yhteneväistä on että tämä tieto saadaan nykyään ilman fyysistä kosketusta staattorin ja roottorin välillä. Näin vältetään sekä moottorin kulumiselta, että anturin vahingoittumiselta. (Barnett 2014, hakupäivä 17.11.2014.)

Resolveri koostuu kahdesta käämistä, joihin roottorin pyörimisestä syntyvä magneettivoivo indusoi jännitettä. Molempien käämeihin indusoituneen jännitteen käyristä pystytään päättelemään roottorin asema ja vauhti.

Enkooderit toimivat roottorissa olevan koodikiekkon avulla. Koodikiekkossa olevia viivoja lukemalla saadaan selville roottorin asema ja vauhti digitaalisena signaalina. Koodikiekkojen muodostamalle signaalille ei ole asetettu omaa standardia, jokainen valmistaja voi halutessaan luoda omansa. Binääri- ja Gray-koodaukset ovat tavanomaisimmat esitystavat koodikiekoissa.

3.2.1 Resolverin rakenne

Resolvereissa käytetään käämien välistä magneettivuota samaan tapaan kuin rautasydämissä muuntajissa. Tyypillinen resolveri koostuu kahdesta staattorin kehälle sijoitetuista käämistä, jotka ovat kehällä 90 asteen kulman päässä toisistaan, jolloin käämien välille on saatu vaihesiirtoa. Vaihtosähkön vaikuttaessa roottoriin syntyy magneettivuota, joka indusoi jännitettä staattoreissa oleviin käämeihin. Käämien jännitekäyriä tulkitsemalla saadaan selville roottorin vauhti ja asema. Roottorin liike muodostaa käämeissä sini- ja cosinimuotoisia jännitekäyriä. Jännitekäyrien erotuksen avulla saadaan selville roottorin asema, ja jännitekäyrien jaksonpituudesta voidaan selvittää roottorin vauhti. Resolverien muodostama signaali on muodoltaan analogista. (Barnett 2014, hakupäivä 17.11.2014.)

3.2.2 Enkooderin rakenne

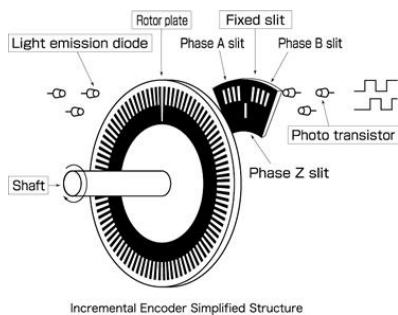
Enkooderit koostuvat kahdesta tai useammasta valoa tunnistavasta anturista, jotka lukevat koodikiekossa olevien valoa läpäisevien tai heijastavien, viivamuotoisten halkioiden avulla muodostuvia pulssimuotoisia signaaleja. Yksinkertainen koodikiekko koostuu tasaisin välein sijoitetuista halkioista, joiden avulla roottorin inkrementaalinen asema voidaan päätellä kiekon avulla muodostuvia pulsseja laskemalla. (Barnett 2014, hakupäivä 17.11.2014.)

Absoluuttinen sijainti on mahdollista saada monimuotoisemmalla koodikiekolla, jossa halkiot ovat kiekon kehällä binäärikoodin kaltaisissa riveissä. Roottorin pyörimissuunta voidaan selvittää sijoittamalla anturit hiukan toisiaan jälkeen, jolloin suunta saadaan selville lukemalla kumpi antureista tuottaa tietyn pulssin ensin.

Optisen enkooderin koodikiekon haittana resolveriin verrattuna on mahdollisten luettavissa olevien asemien lukumäärä, joka on riippuvainen kiekolle mahtuvista merkinnöistä. Myös anturin lukutarkkuus vaikuttaa mahdollisimman monen eri aseman tilan erottamiseen. Enkooderit ovat myös resolvareita herkempiä ympäristötekijöille, eivätkä ne kestä pölyä tai tärinää työympäristössä yhtä hyvin kuin resolverit. (Barnett 2014, hakupäivä 17.11.2014)

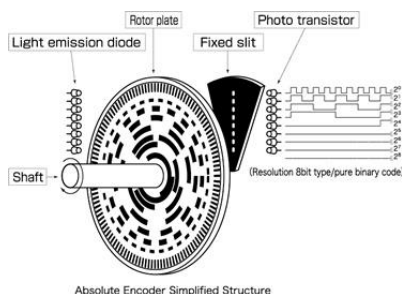
3.3 Inkrementaalisen- ja absoluuttisen anturin eroavaisuudet

Inkrementaalissa enkooderissa koodikiekkko koostuu joko yhdestä tai kahdesta rivistä tasaisin välein sijoitetuista halkioista (kuva 7). Näistä muodostuvan pulssimuotoisen signaalin taajuus on yhteen verrannollinen moottorin pyörimisnopeuden kanssa. Inkrementaalisia enkoodereita käytetään usein nopeutta tai paikkaa ohjattaessa, esimerkiksi renkaissa tai kuljetushihnoissa. Roottorin asemaa ei pystytä selvittämään inkrementaalilla anturilla, joten sitä käytetään sovelluksissa joissa asematieto ei ole tärkeässä roolissa. (Tamagawa seiki Co, Ltd 2014, Hakupäivä 3.9.2014.)



Kuva 7: Inkrementaalisen enkooderin koodikiekkon rakenne (Tamagawa seiki Co. Ltd 2014, Hakupäivä 3.9.2014).

Absoluuttisissa enkooderissa koodikiekkko koostuu binäärikoodin tapaan jäsenellyistä halkioista (kuva 8). Antureille muodostuu siten binäärimuotoista pulssisignaalia. Kiekkon jokaiselle asemalle on siten asetettu oma koodiarvonsa, jonka perusteella roottorin akselin asema saadaan selville joka tilanteessa. (Tamagawa seiki Co, Ltd 2014, Hakupäivä 3.9.2014.)

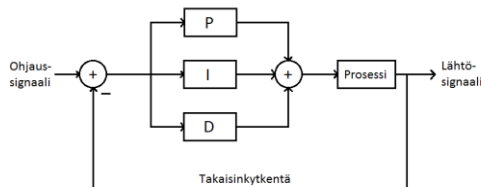


Kuva 8: Absoluuttisen enkooderin koodikiekkon rakenne (Tamagawa seiki Co, Ltd 2014, Hakupäivä 3.9.2014).

Absoluuttisia enkoodereita käytetään ohjauksessa kun roottorin sen hetkinen asema on tärkeä, kuten esimerkiksi teollisuusrobotti- tai työöstökonekäytössä.

4 TAKAISINKYTKENTÄ

Takaisinkytkentä on säätötekniikan prosessi, jossa lähtösignaalin perusteella muokataan järjestelmän toimintaa niin, että se saadaan lähemmäksi haluttua asetusarvoa. Takaisinkytkennässä lähtösignaali lähetetään järjestelmän alkupäähän, jossa se summauksella tai erotuksella lisätään ohjaussignaaliin. Prosessi on saanut nimensä lähtösignaalin takaisinpäin suuntautuvasta kytkennästä (kuva 9).



Kuva 9: PID-säädin ja takaisinkytkentä.

Takaisinkytkentä on tärkeä osa servotekniikkaa, sillä servomoottoreissa olevilta antureilta saatava tieto lähetetään takaisinkytkennän avulla servovahvistimelle. Servomoottoreille on tyypillistä yrittää päästä haluttuun pisteeseen sitä nopeammin, mitä kauempana se piste sijaitsee. Tämä on D-säätimen ennakoivaan säätötapaan verrattavissa olevaa toimintaa.

4.1 Positiivinen ja negatiivinen takaisinkytkentä

Takaisinkytkentä voi olla joko positiivista tai negatiivista. Positiivisen takaisinkytkennän käytön etuna on sen kyky haluttaessa saavuttaa erittäin suuri signaalin vahvistus. Ongelmana on, että signaalista tulee erittäin altis oskillaatiosta johtuvaan signaalin värähtelyyn ja positiiviselle takaisinkytkennälle ominaiseen prosessin epävakaaisuuteen. Epävakaa signaali ei koskaan asetu asetusarvoonsa, vaan kasvaa äärettömästi. (Storr 2014, Hakupäivä 17.11.2014.)

Negatiivinen takaisinkytkentä on yleisessä käytössä säätötekniikassa. Yleensä prosessilla ei ole mahdollista saavuttaa haluttua asetusarvoa luonnollisen häiriön tai muiden tekijöiden vuoksi, mutta negatiivisen takaisinkytkennän avulla järjestelmä näkee, että lähtösignaali ei ole vielä saavuttanut haluttua asetusarvoa ja pyrkii korjaamaan toimintaansa sen mukaan, kunnes lähtösignaali saadaan vastaamaan asetusarvoa. Asetusarvon saa-

vutettuaan järjestelmä on vakaa, eikä säädön tarkkuudesta riippuvaa signaalin huojuntaa lukuun ottamatta muutu arvostaan ilman syytä. (Storr 2014, Hakupäivä 17.11.2014.)

4.2 PID-säätö

Prosesseihin voidaan vaikuttaa muun muassa signaalia vahvistamalla, integroimalla ja derivoimalla, jolloin aikaan saadaan niin kutsuttua PID-säätöä. P-säätö, eli proportsionaalinen, verrannollinen säätö on matemaattisesti yksinkertaisin osa PID säätöä. Kun P:n arvo on kaksi, on lähtösignaalin arvo kaksinkertainen tulosignaaliin verrattuna. P-säätöä voidaan ajatella kertoimena, ja käyttää sitä signaalin vahvistamiseen tai heikentämiseen, prosessista riippuen. On yleistä, ettei P-säätö itsessään aina riitä saavuttamaan asetusrvoa, ellei P:n kerrointa ole tarkasti säädetty soveltuvaan prosessiin. (Kosonen 2012, Hakupäivä 17.11.2014.)

I-säädin eli Integral action perustuu nimensä mukaisesti integraaliin, ja sillä määritellään ohjausarvon ja siihen pyrkivän ohjaussignaalin erotuksen pinta-alaa. Tämän tiedon avulla nähdään, onko signaali nousemassa, eli onko signaalin ja ohjauksen välisen erotuksen arvo positiivinen, vai laskemassa kun erotuksen arvo on negatiivinen. I-säätö pyrkii integrointiajasta riippuen kohti asetusrvoa, yrittäen jatkuvasti pienentää lähtö- ja ohjaussignaalin välistä erotusta. (Kosonen 2012, Hakupäivä 17.11.2014.)

D-säädin saa nimensä derivaatasta. Verrattuna muihin säätöihin on D-säätö erittäin nopea ja aggressiivinen. Tällä säädöllä yritetään saavuttaa asetusrvo mahdollisimman nopeasti. Tämä johtaa usein asetusrvon ylittämiseen jopa kaksinkertaisin arvoin. Tämä näkyy käyrissä korkeina piikkeinä asetusrvon muutoskohdissa, jotka tasoittuvat takaisin nolnaan säädetyn derivointiajan kuluessa. (Kosonen 2012, Hakupäivä 17.11.2014.)

Prosessien ohjauksessa eniten käytetty säädin on kaikki edellä mainitut säätimet yhdistävä PID-säädin. PID-säätimessä yhdistyvät kaikkien säätimien hyvät puolet, jolloin saadaan nopeasti asetusrvoa kohti nouseva käyrä, joka saadaan ilman tavanomaista suurempaa asetusrvon ylitystä laskeutumaan asetusrvoon. Säätimen P, I, ja D arvot voidaan erikseen asettaa prosessin suhteen. Tätä toimenpidettä kutsutaan säätimen virittämiseksi. Virittämällä on tarkoitus saavuttaa vakaa järjestelmä jossa asetusrvot saavutetaan nopeasti ja mahdollisimman pienellä poikkeamalla. (Kosonen 2012, Hakupäivä 17.11.2014.)

5 TWINCAT

TwinCAT on akronyymi sanoista “The Windows Control and Automation Technology”, ja se on Beckhoffin luoma ohjaus- ja ohjelmointiympäristö. TwinCAT ohjelmistolla on mahdollista muuttaa lähes mikä tahansa yhteensopiva PC-järjestelmä reaaliaikaiseksi ohjausjärjestelmäksi tai ohjelmointiympäristöksi ohjelmoitaville logiikoille (PLC), ja numeeriselle ohjaukselle, (NC). Vaikka TwinCAT on alun perin luotu Beckhoffin omille laitteille, on kaikkien valmistajien halutessaan mahdollista käyttää tätä ohjelmistoa omien laitteidensa kanssa ilman lisenssimaksuja. (Beckhoff Automation, 2014.)

5.1 TwinCATin ominaisuudet

TwinCAT muodostuu ajonaikaisista järjestelmistä joilla ohjaukset suoritetaan reaaliajassa, sekä kehittämissympäristöistä ohjelmointiin, diagnostiikkaan ja kokoonpanoihin. Mikä tahansa Windows-ohjelma joka hyödyntää Microsoftin rajapintoja, esimerkiksi visualisointi tai Office-ohjelmat voivat päästä käsiksi TwinCATin sisältämään dataan ja suorittaa sen komentoja. (Beckhoff Automation 2014, Hakupäivä: 20.3.2014.)

Eräänä tärkeimmistä TwinCATin ominaisuuksista ovat sen täsmälliset jaksovälit. Ohjelmat suoritetaan itsenäisesti muista prosessorin tehtävistä, jotta ohjelmat pystyttäisiin suorittamaan mahdollisimman deterministisillä ominaisuuksilla. Saavuttaakseen tämän käyttäytymismallin mukaisen toiminnan asettaa TwinCAT itse suorituksen aikaisen kuorman PC:lle. TwinCAT ilmoittaa myös suorituksen aikana käynnissä olevien ohjelmien kuormituksen. Kuorman enimmäistaso on asetettavissa niin, että voidaan taata tarvittava laskentateho sekä suoritettaville ohjelmille että käyttöjärjestelmille. (Beckhoff Automation 2014, Hakupäivä: 20.3.2014.)

5.2 TwinCAT ohjelmointiympäristö

Ohjelmointi TwinCAT-ohjelmointiympäristössä tapahtuu käyttämällä IEC 61131-3 standardin mukaisia ohjelmointikieliä. Lisäksi on mahdollista käyttää Beckhoffin omaa Continuous Function Chart-ohjelmointikieltä (CFC), jossa on mahdollista asettaa koodin toimilohkoille omat suoritusprioriteetit. Myös tuki yleisimmille kenttäväyläprotokollille

sisältyy ohjelmaan. Ohjelmalla on myös mahdollista ajaa numeerisen ohjauksen sovelluksia.

Beckhoffin periaate TwinCATin ohjelmointiympäristöä suunnitellessa on ollut vähentää menoeriä nykyisten monimutkaisten laitteiden ja tekniikkaan hallinnan kanssa siirtymällä kohti modulaarisia ohjausratkaisuja.

Kaikki itsenäiset toiminnot, koneistot ja järjestelmäkokonaisuudet voidaan mieltää moduuleiksi. Näiden moduuleiden tulisi olla mahdollisimman itsenäisiä, ja rakenteeltaan hierarkiaa noudattava. Rakenteiden pitäisi noudattaa kaavaa jossa hierarkian alhaisimmat moduulit ovat yksinkertaisia ja uudelleenkäytettäviä peruselementtejä, jotka voidaan standardin mukaisten liitäntöjen avulla asettaa korkeampi-arvoisten moduuleiden käyttöön, muodostaen siten monimuotoisempia laitekokonaisuuksia. Ideaalisesti yksittäisiä moduuleita voitaisiin ottaa käyttöön, laajentaa, tai poistaa käytöstä ilman että se vaikuttaisi kokonaisuuden muihin moduuleihin. (Beckhoff Automation 2014, Hakupäivä: 20.3.2014.)

6 LAITTEISTO

Suurin osa tämän opinnäytetyön laitteistosta on saksalaisen Beckhoff Automation GmbH (lyhyesti Beckhoff) yrityksen valmistamia. Beckhoffin laitteita ei ole ollut oppilaitoksen käytössä aikaisemmin, joten oppilaitoksella on ollut tarkoituksena hankkia Beckhoffin laitteistoa jota voitaisiin soveltaa opetuskäyttöön.

Tarjouspyynnöt mahdollisista väyläohjaus-, sekä servomoottori kokonaisuuksista oli lähetetty Beckhoffin Suomen tytäryhtiöön, jotka tekivät tarjouksen mahdollisista laitteistoista.

6.1 Beckhoff Automation GmbH

Beckhoff Automation on maailmanlaajuinen yritys, joka on perustettu Saksassa, Verlin kaupungissa 80-luvulla. Tavoitteena Beckhoff Automationilla on johtoaseman saavuttaminen innovaation kautta. Yksi Beckhoffin uran kohokohtiin kuuluvista innovaatioista on vuonna 2003 julkistettu EtherCAT, reaaliaikainen Ethernet-pohjainen kenttäväylä-protokolla, joka vuonna 2007 hyväksyttiin kenttäväylä-standardeiden joukkoon. (Beckhoff Automation 2014.)

6.1.1 Beckhoff yrityksenä

Beckhoffin neljä tärkeintä tuoteryhmää ovat Teollisuus-PC:t, I/O-järjestelmät, liikkeenohjausjärjestelmät, kuten servomoottorit ja -vahvistimet, sekä ohjelmistot omien tuotteidensa sekä muiden standardinmukaisten automaatiokokonaisuuksien järjestelmiin.

Beckhoffin tuotteita ja järjestelmäratkaisuja käytetään maailmanlaajuisesti monenlaisissa sovelluksissa nopeista työstökeskuksista aina älykkääseen rakennusautomaatioon. (Beckhoff Automation 2014, Hakupäivä 27.3.2014.)

6.1.2 Beckhoff Suomessa

Beckhoff on toiminut Suomessa vuodesta 1986, ensin edustajan välityksellä ja vuodesta 2000 eteenpäin yhtiönä. Beckhoff Suomen pääkonttori on Hyvinkäällä. lisäksi Hyvinkäällä sijaitsee Beckhoff komponenttivarasto.

Haarakonttorit sijaitsevat Tampereella, Seinäjoella ja jatkossa jopa Tallinnassa. Kaikissa Beckhoff Suomen konttoreissa on myynti, tekninen tuki, koulutus, tuotekehitys, sovellukset ja huolto edustettuna. (Beckhoff Automation 2014, Hakupäivä 27.3.2014.)

6.2 Sulautettu-PC, CX5010

Laitteiston logiikkana toimii Beckhoffin CX5010 sulautettu-PC DIN-kiskolle. Modulaarisen rakenteensa vuoksi laitteessa on mahdollista käyttää vain niitä I/O-yksiköitä, joita järjestelmässä tarvitsee näin säästäten tarvittavasta asennustilasta.

Tätä kokoonpanoa varten oli tilattu moduulit EL1008 digital input, EL2008 digital output, sekä EL3002 Analog input. I/O-yksiköiden numeron ensimmäinen luku ilmaisee perustyyppiä, viimeinen luku taas tulojen/lähtöjen lukumäärää.

6.3 Servovahvistin, AX5201

Laitteistossa on AX5200-sarjan 2-kanavainen digitaali-servovahvistin. Tähän servovahvistimeen on mahdollista kytkeä 2x1,5 A servomoottoria. Molemmille kanaville löytyy vahvistimesta omat takaisinkytkentäpaikat, jos käytetyt moottorit eivät tue Beckhoffin omaa OCT-kaapelitekniikkaa. Kanavien lähtövirta on myös jaettavissa epätasaisesti, moottoreiden käyttämä maksimivirta voi olla esimerkiksi 2,5 A ja 0,5 A, kunhan yhteenlaskettu virta ei ylitä servovahvistimen enimmäisvirtaa. Servovahvistin sisältää myös jarrutusyksikön, jossa on sisäänrakennettu 250 W:n jarrutusvastus, sekä mahdollisuus kytkeä ulkoinen, maksimissaan 6 kW:n jarruvastus. (Beckhoff Automation 2014, Hakupäivä 27.3.2014.)

Teknisiltä ominaisuuksiltaan AX5201 on erittäin nopea suorittamaan säätöjä, virransäädin kykenee toimimaan nopeimmillaan 62.5 μ s:ssa, nopeudensäätö 125 μ s:ssa ja paikansäätö 250 μ s:ssa. (Beckhoff Automation 2014 Hakupäivä 27.3.2014.)

6.4 Servomootorit, AM8021-0B20

Laitteiston servomootoreina toimivat AM8000-sarjan synkroniservomootorit. Tämän sarjan moottorit ovat suunniteltu yksinomaan nopeus- ja/tai momenttiohjattuun käyttöön Beckhoffin omien digitaali-servovahvistimien välityksellä.

Moottoreissa käytetään Beckhoffin omaa One Cable Technology-kaapelia (OCT), jossa moottorin syöttö- ja takaisinkytkentäkaapeli on yhdistetty yhdeksi kaapeliksi. AM8000-sarjan moottoreissa on matala hitausmomentti ja erittäin suuri ylikuormituskestoisuus. AM8021-servomootorin nollanopeusmomentti on 0,5 Nm. Sen momentti on pienin sarjan moottoreista. Tuotekoodin 0B20 tarkoittaa, että kyseinen servo on ”multi-turn” tekniikan moottori, eli moottorin absoluuttinen asema ei ole rajattuna yhteen kierrokseen. Lisäksi servomootorissa ei sisällä erillistä sähkömagneettista jarrua. (Beckhoff Automation 2014 Hakupäivä 27.3.2014.)

6.5 EtherCAT

EtherCAT on Beckhoff Automationin lanseeraama, Ethernet-pohjainen kenttäväyläprotokolla. EtherCAT luotiin koska oli tarvetta kenttäväylälle, jossa tieto voitaisiin välittää mahdollisimman reaaliaikaisesti, synkronoida ja jossa ei olisi tarvetta kalliille erikoiskenttäväyläkaapeleille tai niiden lähettimille ja vastaanottimille. Ethernet-pohjaisuus takaa halvan hinnan ja hyvän yhteensopivuuden PC-laitteiden välillä, sillä yhteyskaapelina on mahdollista käyttää halpaa ja erittäin yleisessä käytössä olevaa verkkokaapelia. Hinnan, nopeuden ja avoimuutensa takia EtherCAT vakiintui osaksi IEC-standardeja. (Real Time Automation 2009, Hakupäivä 4.9.2014.)

Protokollan toimintaperiaatteena on, että viestiä ei kohdisteta tiettyä solmua kohden, joten sen ei tarvitse pysähtyä väylän jokaisessa solmukohtassa, vaan viesti lähtee master-yksiköltä, kulkien kehää jokaisen slave-yksikön kautta. Slave-yksiköt käsittelevät

saapuvan viestin, lisäävät viestiin oman datansa, ja viesti jatkuu eteenpäin seuraavalle yksikölle. Tätä liikennöintiä voidaan verrata tavarajunaan, joka pystyisi liikkeessä tyhjentämään ja täyttämään vaununsa aina aseman ohittaessaan. (Real Time Automation 2009, Hakupäivä 4.9.2014.)

6.6 Kone- ja laiteturvallisuus

Kone- ja laiteturvallisuuden säädökset on syytä ottaa huomioon kaikenlaisia laitteistoja kootessa. Koneturvallisuusdirektiivin uusin versio on säädetty vuonna 2006, ja sitä voidaan soveltaa moniin koneisiin ja laitteisiin. Direktiivi sisältää koneen suunnittelua ja rakentamista koskevat olennaiset terveys- ja turvallisuusvaatimukset. Vaatimuksia noudattaville tuotteille on mahdollista myöntää CE-merkintä, joka sallisi niiden myynnin EU-jäsenmaissa.

Oppimisympäristössä on tavoiteltavaa, että noudatettaisiin tärkeimpiä direktiivin vaatimuksia yleisen käyttäjäturvallisuuden tähden. Näihin sisältyy esimerkiksi raja-arvojen määrittely, mahdollisten ennakoitavissa olevien väärinkäyttöjen estäminen, potentiaalisten vaarojen ja vaaratilanteiden tunnistaminen, riskien arviointi sekä niiden tavoitteen mukainen pienentäminen. (Eur-lex, 2006/42/EY.)

7 SUUNNITTELU

Uuden oppimisympäristön luominen koostuu useasta vaiheesta ja jokaisen vaiheen kanssa tulisi olla toimintasuunnitelma. Suunnitelmissa laadittiin ensin, kuinka oppimisympäristön laitteet sijoitetaan sopivaksi laitteistokokonaisuudeksi ja kuinka kokonaisuus saadaan koottua esittelykelpoiseksi oppimisympäristöksi. Myös oppimisympäristön jännitteensyöttö ja johdotus täytyi ottaa huomioon jo suunnitteluvaiheessa. Oppimisympäristön käyttötapa täytyi olla selvillä jo alkuvaiheessa. Täydellisessä suunnitelmien pohjalta kootussa ympäristössä kaikki tekijät olisivat huomioitu etukäteen.

Suunnittelun ideaalisena lopputuloksena olisi, että projekti valmistuu aikataulun mukaisesti ilman ongelmia, sillä kaikki mahdolliset ongelmat olivat jo huomioitu ja ratkaistu. Suunnittelu on tärkeä osa projektia, mutta sen suhteen ei tule jättää huomioimatta muita projektin osa-alueita, sillä ajankäyttö on tärkeässä roolissa isoissa projekteissa.

7.1 Oppimisympäristön vaatimukset

Oppimisympäristön suunnittelussa oli otettava huomioon tiettyjä vaatimuksia ja toiveita, joiden avulla ympäristöstä saataisiin paremmin oppilaitokselle ja opetuskäyttöön soveltuva. Perusvaatimuksina ympäristölle olivat esimerkiksi:

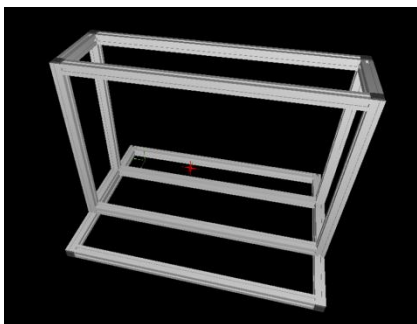
- Oppimisympäristön tulisi sopia mahdollisimman pieneen tilaan, jotta se voidaan säilyttää hyllyillä.
- Laitteistoa ei tulisi kätkeä koteloon tai muuten suljettuun tilaan, servovahvistimen ja logiikan julkisivut tulisivat olla näkyvillä.
- Laitteiston tulevaisuutta ajatellen olisi hyvä olla mahdollisuus laajentaa ympäristön ominaisuuksia lisäämällä logiikalle uusia moduuleita.
- Servomoottorit sekä jännitteelliset osat on suojattava kosketukselta, ympäristön laite- ja koneturvallisuuden parantamiseksi.
- Servovahvistimen ja logiikan ilmankierto täytyy ottaa huomioon, joten molemmat tulee sijoittaa ohjekirjan mukaisesti pystyasentoon.

7.2 Oppimisympäristön suunnittelu

Oppimisympäristön laitteiston sijoittamisessa oli perusideana, että kuuden kilogramman painoinen servovahvistin täytyy sijoittaa mahdollisimman keskelle ympäristöä, jotta painopiste saataisiin pysymään mahdollisimman keskellä. Tämä helpottaa ympäristön liikuttamista, mutta ohjauksen ja logiikan sijoittaminen tuottaa lieviä ongelmia. Logiikka ja ohjaus voidaan sijoittaa joko vastakkaisille reunoille tai päällekkäin. Servomootorit saadaan sijoitettua ympäristön laidalle.

Beckhoffin laitteiston sijoittamisen lisäksi tarvitsevat logiikka ja servovahvistin omat tasajännitesyöttönsä. Jännitelähteiden sijoittamiseen täytyi ympäristössä varata tarvittava tila. Lisäksi täytyi varata tilaa ympäristön ohjaukselle. Ohjaus on suunniteltu tapahtuvan logiikan digitaalitulojen ja painonappien avulla. Ohjaukseen sisältyy myös merkivalot, jotka ovat yhteydessä logiikan digitaali-lähtöihin. Näin ohjelmoija pystyy hyödyntämään ympäristössä logiikan I/O-digitaaliyksiköitä.

Kun laitteiston käyttämä tila saatiin arvioitua, mitoitettiin sen pohjalta sopivat asennusalustat. Asennusalustoille suunniteltiin myös sopiva asennuskehikko. Asennuskehikon suunnitelmien pohjana toimivat oppilaitoksen muut pöytämalliset asennuskehikot. Kehikkoon tarvittavat osat tilattiin alustojen mittojen mukaan. Kehikko rakennetaan neliskanttisista alumiinitangoista. Näissä tangoissa on urat johon asennusalustat voidaan asettaa (kuva 10; liite 1).



Kuva 10: Asennuskehikon 3D-malli

Servomootoreilla ohjattavan toiminnan alkuperäisenä ideana oli rakentaa kuljetinhihna. Kuljettimessa moottoreita olisi käytetty siirtämään esineitä nopeasti ja tarkasti asetettuihin pisteisiin. Tätä ratkaisua ei olisi ollut mahdollista toteuttaa ilman, että ympäristölle suunniteltuja asennusalustoja ja -kehikkoa olisi laajennettu huomattavasti.

Mahdollisimman tilaa säästävänä ideana oli että servomoottorit sijoitettaisiin päällekkäin, ja että niillä ohjattava toiminto olisi pystysuuntaista. Tämä ajatus toimii suunnitelmien pohjana alusta lähtien (liite 2).

7.3 Suunnittelun ongelmakohdat

Servovahvistimen sijoittaminen aiheutti ongelmia. Vahvistin on kookas syvyys suunnassa, joten kaikkea laitteistoa ei voida sijoittaa samalle levyille ilman, että se ulottuisi muuta laitteistoa edemmäksi. Ratkaisuna tähän servovahvistimen syvyyttä kompensoidaan lisäämällä kehikkoon toinen levy (liite 2), johon muut tärkeät laitteiston osat sijoitettaisiin. Näin logiikan ja servovahvistimen julkisivut saadaan samalle tasolle.

Toinen levy tuottaisi omat ongelmansa laitteiston kokoamisessa, sillä uloimpaan levyyn täytyisi leikata reikä servovahvistimen rungon läpiviemiseksi. Tämä tarkoittaisi sitä, että laitteiston levyt täytyisi asettaa asennuskehikkoon yhtä aikaa, jolloin mahdolliset muutokset ja johdottaminen täytyisi tehdä levyjen ollessa paikoillaan.

Servovahvistin olisi mahdollista sijoittaa oppimisympäristön vasempaan reunaan. Näin ympäristö voitaisiin suunnitella niin, ettei päällimmäiseen levyyn tarvitse erikseen leikata läpivientireikää servovahvistimelle (liite 3). Oppimisympäristön huono painopiste ja epäsymmetrinen muoto johtivat tämän idean hylkäämiseen.

Suunniteltiin myös vaihtoehtoista ratkaisua oppimisympäristön muun laitteiston ja moottoreiden jakamisesta kahteen eri kehikkoon (liite 4). Tässä ratkaisussa moottorit olisivat olleet mahdollista halutessa kääntää kyljelleen, näin mahdollistaen niiden käytön kuljettimena. Erilliset osiot olisivat myös säästäneet hyllytilassa, koska moottorit ja muu laitteisto olisi mahdollista säilyttää eri tiloissa. Helposti liikuteltavissa olevat moottorit nostavat liikaa vaaratilanteiden todennäköisyyttä tällaisessa ympäristössä. Kyseisiä moottoreita ei ole suunniteltu käytön aikana liikuteltaviksi.

Lopullinen ratkaisu servomoottoriympäristön kehikolle oli pitäytyä yhtenäisessä ympäristössä, jossa päällimmäiseen levyyn leikataan reikä servovahvistinta varten. Servomoottoreiden käyttökohteelle varattiin tilaa asennusalustasta noin kolmannes.

7.4 Servomootorin käyttökohteen suunnittelu

Sopiva käyttökohde servomootoreille oli pitkään auki. Servomootoreille on mahdollista hyödyntää niin monissa sovelluksissa. Tarkoituksena olisi saada oppimisympäristökäyttöön mahdollisimman havainnollistava käyttökohde. Kuljetinkäyttöä ei tilan puutteen takia pystyttäisi toteuttamaan vaakatasossa. Pystysuuntaisissa kuljettimissa ongelmaksi muodostuu, kuinka kappaletta tultaisiin kuljettamaan. Sopivin pystysuuntainen ratkaisu servomootoreiden käyttökohteeksi oli hissi.

Mootorit nostaisivat hissinä toimivaa kappaletta vastakkaisista kulmista. Lisäksi mootoreiden hammasrattaisiin pitäisi voida vaikuttaa lisäämällä niihin painoja. Painot aiheuttaisivat servomootorien pyörimisliikkeeseen muutosta. Saavuttaakseen tasaisen pyörimisliikkeen täytyy servo-ohjaimen kompensoida massan aiheuttamaa muutosta liikkeeseen lisäämällä tai vähentämällä virransyöttöä mootoreille. Nämä muutokset tulisivat näkyviin servomootoreiden virrankäyttökäyrissä.

Normaalitilanteessa mootorit nostaisivat hissin haluttuun korkeuteen tasaisesti, mutta säätöön ja moottorin liikkeeseen vaikuttamalla saataisiin tilanteita, joissa moottorien toiminnan ero aiheuttaisi silmämääräisesti huomattavissa olevaa eroa hissin tasossa.

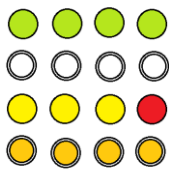
Hissiratkaisussa on kuitenkin merkittäviä ongelmia. Jos käyttäjä tarkoituksella ohjaa mootoreita yhtä aikaa eri suuntiin, kallistuisi hissi niin paljon, että hissivaunu ottaisi kosketusta moottorin hihnoihin, tai saattaisi irrota. Tämä voi johtaa ajan myötä laitteiston pysyvään vaurioitumiseen tai siihen, että laitteistoa täytyy purkaa vian korjaamiseksi.

Vaihtoehtoisesti hissi korvataan kahdella pystysuuntaisella hihnalla, joiden asetettu nol-lapiste merkittäisiin. Vertaamalla näiden merkkien korkeutta toisiinsa, voidaan emuloida hissivaunun tasoa ilman hissiä. Ratkaisun etuna on sen helppous ja huoltovapaus. Tämä myös mahdollistaa laitteistolle käyttökohteen vaihtamisen tulevaisuudessa.

7.5 Laitteiston ohjauksen suunnittelu

Ohjaus on tärkeässä roolissa oppimisympäristössä, käyttäjän on jotenkin pystyttävä vaikuttamaan prosessiin. Ohjauksena toimii logiikan digitaalituloa vastaavat painonapit. Ohjaukselle varattuun tilaan tulee painonappien lisäksi myös logiikan digitaalilähtöjä vastaavat merkkilamput. Logiikan tulojen ja lähtöjen käyttötarkoitusta on mahdollista muokata vapaasti ohjelmointia muokkaamalla, joten jokainen merkkilamppu ja painonappi voidaan asettaa suorittamaan mitä tahansa haluttua toimintaa. Tätä voidaan kuvastaa käyttämällä visuaalisesti toisistaan erottuvia merkkilamppuja ja painonappeja.

Samana sävyisten painikkeiden visuaaliselta samanarvoisuudelta voidaan näin välttyä. Monipuoliset värit helpottavat ulkopuolisia ohjelman ymmärtämisen kanssa. Ympäristön sisältäessä erivärisiä merkkivaloja, voidaan niitä vastaavien painikkeiden ymmärtää sisältävän tärkeämpiä toimintoja (kuva 10). Ohjelmoija on silti vapaa käyttää kytkimiä ja merkkilamppuja haluamallaan tavalla.



Kuva 10: Oppimisympäristön painonapit ja merkkilamput.

Käytettävissä olevat värit painonapeille olivat valkoinen ja keltainen, ja merkkilampuille keltainen, vihreä ja punainen. Tarkoituksena on, että käyttäjät voisivat oppimisympäristöä ohjelmoidessa loogisesti hyödyntää näitä merkkivaloja. Esimerkiksi punainen merkkivalo voidaan ohjelmoida virhetilanteen lampuksi. Punaisen merkkivalon alapuolella olevaa painonappia voitaisiin taas käyttää esimerkiksi järjestelmän pysäyttämiseksi tai uudelleenkäynnistysnappina.

7.6 Johdotuksen suunnittelu

Kun laitteiston paikat asennusalustoilla olivat päätetty, täytyi näiden välille suunnitella mahdollisimman yksinkertainen ja kaunis johdotus. Asennuskehikon ja -alustojen väliin jääviin tyhjiin tiloihin koottava johdotus jää ympäristön sivuilta näkyväksi, joten sen ulkonäkö täytyy ottaa huomioon. Laitteiston johdotuksen lopullisessa suunnitelmassa

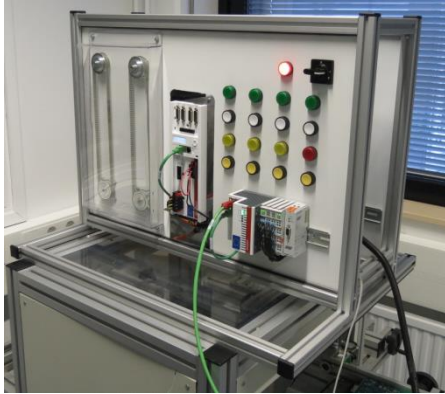
tärkeänä päätekijänä on ollut, että jos laitteiston etummainen asennusala täytyy poistaa paikaltaan, se voidaan tehdä mahdollisimman vähällä johdinten irrottamisella.

Alkuperäisessä johdotussuunnitelmassa oli tarkoituksena sovittaa kaikki riviliittimet yhdelle kiskolle jättäen taaemman alustan tyhjän tilan muuhun käyttöön tulevaisuudessa. Ongelmaksi tässä muodostui tilan puute, ja vaadittavien hyppyjohdinten määrä. Johdotus koottiin lopulta kahdelle eri riviliitinkiskolle.

Logiikkaan liittyvät johtimet ja riviliittimet kootaan logiikan kääntöpuolella olevalle riviliitinkiskolle, jonka alapuolelta kulkee läpivienti logiikan moduulien liittimille. Kääntöpuolen riviliitinkiskolle on sijoitettu myös tasajännitelähteiden syöttö, sekä logiikalle ja ohjaukseen tarkoitettu 24 voltin tasajännite. Tästä jännitteestä logiikka ja ohjaus saavat tarvittavan jännitteensä.

8 TOTEUTUS

Toteutusvaiheessa on tarkoitus koota oppimisympäristö valmiiksi suunnitelmien pohjalta. Tähän vaiheeseen sisältyy laitteiston kokoaminen, johdotus, testaus, sekä muu työ. Lopputuloksena toteutusvaiheessa saadaan valmis oppimisympäristö (kuva 11).



Kuva 11: Valmis oppimisympäristö testivaiheessa

8.1 Kone- ja laiteturvallisuus käytännössä

Tärkeimmässä osassa Laite- ja koneturvallisuuden toteutumisessa on, että turvallisuustekijät ovat otettu huomioon jo suunnitteluvaiheessa. Esimerkiksi jo laitteiden sijoittelulla voidaan parantaa turvallisuutta.

Oppimisympäristön liikkuvat osat, eli servomoottorit ja niihin liittyvät hammaspyörät ja -hihnat kosketussuojataan polykarbonaattikuorella (kuva 12). Näin vähennetään vaaratekijöiden riskiä siitä, että esimerkiksi kenenkään vaatteet tarttuisivat kyseisiin osiin.



Kuva 12: Servomoottoreita kosketukselta suojaava polykarbonaattikuori.

Hätäseis, eli mahdollisuus laitteiston äkilliseen pysäytykseen, oli pitkään osa oppimisympäristön suunnitelmia, mutta sen sijoitus osoittautui tilan puolesta ongelmalliseksi. Ratkaisuksi tähän sen virransyöttö täytyy ottaa luokan vikavirtasuojakytkimillä varustettujen työpöytien kautta. Näin oppimisympäristön jännitteet saadaan katkaistua luokkatilojen omia hätäseis-kytkimiä käyttämällä.

Laiteturvallisuutta voidaan parantaa entisestään lisäämällä ylimääräiset suojamuovit oppimisympäristön reunoihin. Tämä voidaan halutessa tehdä sovittamalla sopivan kokoiset muovilevyt ympäristön alumiinitankojen uriin.

8.2 Logiikka, laitteiston esitestaus

Ennen kuin tarvittavat osat asennuskehikkoon saatiin toimitettua, käytettiin odotusaika hyödyksi tutustumalla ohjelmoitavaan logiikkaan ja sen ohjelmointiympäristö- ja hallintasovellukseen, TwinCATiin. Ohjelman kanssa oli aluksi ongelmia muodostaa yhteys logiikan ja tietokoneen välille, mutta Beckhoffin Suomen jaoston avustuksella, sekä heidän järjestämän lyhyen opetuskurssin avulla ongelmatilanteet saatiin korjattua.

Opetuskurssilla käsiteltiin ohjelman käyttöä, ja ratkaistiin logiikan ja tietokoneen välinen yhteydenmuodostamisongelma. Ongelmana oli vanhan TwinCAT-version ja logiikan sisällä olevan uudemman TwinCATin yhteensopimattomuus. Uudempaa versiota käyttämällä saatiin logiikan digitaalitulo- ja digitaalilähtömoduulien toiminta testattua ennen varsinaista oppimisympäristön kokoamista. Myös servovahvistimen ja moottoreiden toiminta testattiin tämän opetuskurssin aikana.

8.3 Asennuskehikon kokoaminen

Oppimisympäristöön käytiin leikkauttamassa oikean mittaiset asennusalustat läheisen ammattiopiston konepuolen tiloissa. Alustojen oli alun perin tarkoitus olla alumiinia, mutta alumiinilevyjä ei ollut sillä hetkellä varastossa. Oppimisympäristön rakentamisessa täytyi siis tyytyä teräslevyihin.

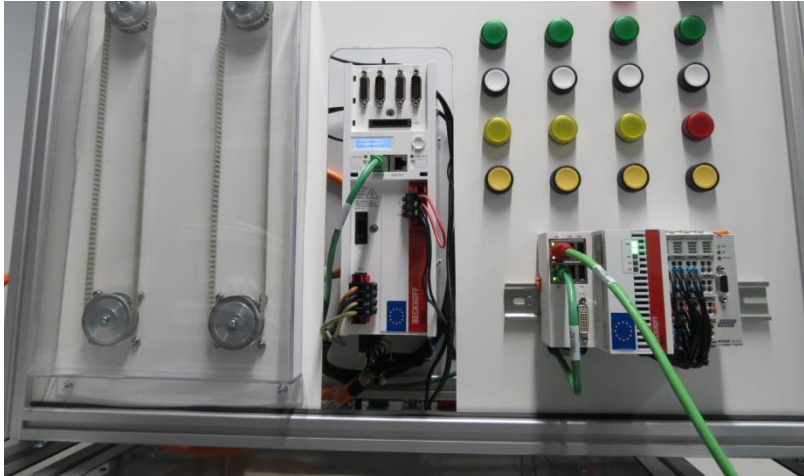
Alumiinilevyihin verrattuna ovat teräslevyjen suurimpina ongelmina niiden paino ja työstövaikeus. Alkuperäisenä suunnitelmana oli, että tarvittavat reiät ja läpiviennit levyyn työstettäisiin oppilaitoksen omilla työkaluilla, mutta näitä työkaluja ei ole suunniteltu teräslevyjen käsittelyyn. Levyjen suurimmat reiät täytyi käydä työstämässä samassa ammattiopiston konepuolen tilassa. Pienimmät ruuvinreiät pystyttiin poraamaan käsivoimin oppilaitoksessa. Valmiit levyt maalattiin mattavalkoiseksi.

Asennuskehikkoa varten Maytec-yritykseltä tilatut alumiinitangot saapuivat. Toimitetut osat tarkastettiin ja todettiin tilausta vastaaviksi. Alumiinitankoihin on tilauksen mukaisesti työstetty sopivan mittaiset reiät liitospaloille. Liitospaloja käyttämällä alumiinitangot voidaan kiristää erittäin tiukasti kiinni toisiinsa. Näin oppimisympäristöstä saadaan tarpeeksi jyrkevä kannettavaksi.

Kokoamisvaiheessa ilmeni, että asennuskehikkoon liittyviä sivutuotteita olisi kannattanut tilata enemmän. Esimerkiksi alumiinitankojen kiertämistä estäviä momenttikiiloja olisi ollut hyvä olla enemmän. Momenttikiilojen ruuvitkin olisi täytynyt tilata erikseen. Myös tilattujen käsikahvojen kiinnitysmalli ei vastannutkaan suunniteltua. Toinen käsikahvoista saatiin käyttöön sijoittamalla se pohjalevyn takaosaan. Tämä helpottaa servovahvistimen sisältävän alustan pois nostamista keihokosta. Asennuskehikko koottiin tässä vaiheessa alustojen lisäämistä vaille valmiiksi.

8.4 Laitteiston kokoaminen asennuskehikkoon

Asennuskehikon taaimmaiselle alustalle kootaan vain servovahvistin, kaksi tasajännitelähdettä logiikan ja vahvistimen syötöksi, sekä riviliitinkisko. Laitteiden lisäämisen jälkeen se asetetaan omalle paikalleen asennuskehikkoon. Päällimmäiseen asennusalustaan on työstövaiheessa leikattu lovi, jonka ansiosta sekin voidaan laskea asennuskehikkoon (kuva 13). Alustalle kootaan tämän jälkeen servomoottorit, logiikka, sekä ohjauksen painonapit ja merkkilamput. Servomoottoreiden akseleihin ja niiden vastakappaleisiin kiinnitetään tämän jälkeen hammashihnapyörät ja hammashihnat. Lopuksi lisättiin servomoottoreiden liikkuvat osat kosketussuojaava polykarbonaatista valmistettu kuori.



Kuva 13: Servovahvistinta varten työstetty lovi helpotti oppimisympäristön kokoamista.

Servomoottoreiden akselin ulkomuoto tuotti ongelmia hammashihnapyörien valinnan kanssa. Moottoreiden akseleiden ei oletettu olevan sileät, vaan kiilamuotoiset. Tämän vuoksi akseleihin asetettavien hammashihnapyörien täytyisi olla sellaisia, joissa on mahdollista käyttää lukitusruuveja. Onneksi servomoottoreiden akseleihin kärkeen on työstetty sopivat kierteet ruuveille, joten hammaspyörät voidaan kiinnittää käyttämällä ruuveja ja tarpeeksi isoja aluslaattoja.

Hammashihnojen vastakappaleet olisi kannattanut jälkikäteen ajateltuna sijoittaa levyn alapäähän ja servomoottorit levyn yläpäähän, jolloin hihnojen kiristyksen suorittaminen helpottuisi. Servomoottoreiden sijoitus levyn alapäähän on silti kannattavampi ratkaisu pyörimisliikkeestä aiheutuvan värinän vähentämisessä (kuva 14).



Kuva 14: Servomoottorit ovat sijoitettu oppimisympäristön alalaitaan.

8.5 Logiikan ja muun laitteiston johdotus

Oppimisympäristön laitteiston kokoamisen jälkeen aloitettiin johdotus. Logiikan lohkojen ja syötön vaatimat johtimet kuljetettiin läpiviennin kautta riviliittimille. Nämä johtimet on erittäin tärkeä johdottaa mahdollisimman siististi, sillä kyseessä on lähes ainoat julkisivulle näkyvissä olevat johtimet. Logiikalle menevien johdinten lisäksi, jää ympäristön julkisivulle näkyviin servovahvistimen kolmivaihe- ja tasajännitesyötön johtimet.

Oppimisympäristöön kytketään jännite nokkakytkimen käynnistyksellä. Vaihtojännite kulkeutuu tasajännitelähteiden syötön riviliittimille, Jännitelähteiden tuottama tasajännite viedään releille, jotka saavat muodostettua pitopiirin tasajännitelähteiden sekä servovahvistimen syötöille.

Digitaalilähtöjen ja ohjauksen välistä johdotusta lukuun ottamatta kaikki onnistui suunnitelmien mukaisesti. Ohjauksessa tarvittavien merkkilamppujen kannat olivat loppuneet varastosta, jonka vuoksi niiden johdotus täytyi siirtää myöhemmäksi (kuva 15). Oppimisympäristöön saatiin kuitenkin asennettua kolme merkkilamppua kantoineen, joten logiikan ja ohjauksen välistä toimintaa päästiin silti testaamaan ohjelmointiympäristön kautta. Tilatut osien saavuttua varastoon saatiin puuttuvat lampunkannat asennettua, ja oppimisympäristön johdotus suoritettua loppuun normaalisti. Valmiista johdotuksista tehtiin myös sähköpiirustukset (liite 6). Sähköpiirustuksissa on kuvattuna oppimisympäristön riviliitinkiskojen johdotus.



Kuva 15: Vaiheessa oleva oppimisympäristön ohjauksen johdotus.

8.4 Ohjelmointi ja dokumentointi

Oppimisympäristön fyysisesti valmistuttua, voitiin keskittyä sen ohjelmalliseen puoleen. TwinCAT 3-sovelluksella luodaan uusi projekti ja muodostetaan yhteys logiikan ja tietokoneen välille. Yhteyden muodostamisen jälkeen projektiin lisätään kaikki laitteiston osat joko manuaalisesti, tai automaattista hakua käyttämällä. Yhteyden saatua on mahdollista muokata laitteiston eri osien parametreja mieleiseksi. Näitä parametreja ovat esimerkiksi servomoottorin toimintatilan, kiihtyvyyden, nopeuden, ja muut servomoottori ohjaukseen liittyvät arvot. Muokatut arvot voidaan sen jälkeen tallentaa logiikan sisäiseen muistiin.

Logiikan ja servomoottoreiden toimintaa voidaan TwinCAT-ohjelmassa testata ohjaamalla näiden tilaa niin sanotussa ”online” tilassa. Tässä tilassa on mahdollista pakottaa logiikan bittejä eri tiloihin, sekä sallia servomoottorin ohjaaminen TwinCAT-sovelluksen näppäimillä. Näin esimerkiksi servomoottorin asetukset ja käyttötapa voidaan testata ennen ohjelmoinnin aloittamista.

Jotta käyttäjä oppisi käyttämään tätä ohjelmointiympäristöä sekä luomaan logiikalle oman ohjelmansa, tulee käyttäjää perehdyttää ohjelmointiympäristön käyttöön asianmukaisella dokumentoinnilla. Tätä varten laadittiin oppimisympäristölle käyttöohjeet (liite 7). Näitä käyttöohjeita noudattamalla ja ohjelmaan tutustumalla, olisi käyttäjän tarkoitus päästä alkuun ympäristön ohjelmoinnissa.

9 POHDINTA

Aiheena opinnäytetyöhön minulle tarjottiin laitteiston hyödyntämistä joko Beckhoffin kenttäväyläkäytön- tai servomoottorikäytön oppimisympäristössä. Vaikka kokemukseni servomoottorisovelluksista opetusikässä olivatkin olemattomat, valitsin servomoottorikäytön puhtaasta mielenkiinnosta servotekniikkaa kohden. Minulla oli jo servomoottorikäytön valitessani huoli siitä, kuinka laaja ja monimutkaisempi tästä projektista tulisi vaihtoehtoon verrattuna. Valinta oli silti mielestäni erittäin hyvä.

9.1 Suunnittelun tärkeys

Epäilykset projektin monimutkaisuudesta osoittautuivat todeksi erinäisten ongelmien ilmennyttyä muun muassa TwinCAT ohjelman ja logiikan välillä, sekä erinäisten pienten, ei niin tärkeiden osien kanssa, kuten ruuvit, mutterit, sekä merkkilampuille tarvittavat kannat, joita ei ollut oletetusti tarpeeksi varastossa. Erinäiset ongelmatilanteet johtavat helposti aikataulusta poikkeamiseen. Tämä pisti minut miettimään, kuinka tärkeää suunnitelmissa on myös pienien asioiden ja riskien huomioon ottaminen. Kallisarvoista aikaa kuluu jokaisen ongelman ratkaisemiseen, joten mahdollisimman moneen asiaan tulee osata varautua, varsinkin työn koostuessa useista vaiheista.

9.2 Servomoottorien käyttö

Servomoottorien käyttötarkoitus oli erittäin pitkään viimeistelemättä. Laitteistolle on niin monta mahdollista käyttökohdetta. Servomoottoreiden tyypillinen käyttötarkoitus on tarkka ja nopea ajo haluttuun pisteeseen, esimerkiksi esineen siirtäminen kuljettimella kymmenesosamillin tarkkuudella siihen mihin haluttiin. Ongelmina kuljettimissa on se, että ne vievät suunnattomasti tilaa vaakatasossa, varsinkin jos halutaan havainnollistaa sekä nopeutta, että tarkkuutta, lyhyessä kuljettimessa ei kumpakaan juuri ehdi todistamaan, joten jo alkusuunnitelmissa ideana oli jonkin pystysuuntainen kuljetinratkaisu servomoottoreille, mutta ei välttämättä tavanomainen kuljetinhihna.

Epätavallisena, mutta omasta mielestäni hyvin säätöasetuksia havainnollistavana käyttökohteena olisi ollut hissi, jota nostettiin molemmilta puolilta, mutta tämä idea toimi paremmin paperilla. Hissi-idean kautta päädyttiin takaisin pystysuuntaisten kuljetinhihnojen käyttöön. Tämä mahdollisti sen, että alkuperäisiä suunnitelmia, joissa moottoreille varattu tila oli pystysuuntaista, voitaisiin noudattaa asennusalustan ja kosketussuojauksen kanssa.

9.3 Oppimisympäristön muokattavuus

Oppimisympäristöä, ja varsinkin oppimisympäristön ohjausta suunnitellessa, oli tarkoituksena tähdätä siihen, että ympäristöstä saataisiin muokattavissa oleva. Tämä pohdinta pohjautuu hiukan omaan ajatukseeni siitä, mitä oppimisympäristön täytyisi olla. Ohjelmoitavien logiikoiden hyötynä on juurikin näiden ohjelmoitavuus, joten kaikki logiikan

digitaalitulot ja -lähdöt tulisi olla käytettävissä laitteistossa, vaikka kaikille ei ohjelmallisesti mitään käyttökohdetta olisikaan. Tavoitteena on, että tulevaisuudessa oppimisympäristöön voitaisiin ohjelmoida aina uusia ominaisuuksia tarpeen mukaan parantaakseen sen toimintaa oppimisympäristönä. Näin kyseessä ei olisi vain kiinteä laitteisto kiinteällä ohjelmalla, vaan oikea oppimisympäristö, johon käyttäjät voisivat vaikuttaa.

Oppimisympäristön puolesta toivon, että sille löytyisi käyttö myös tulevaisuudessa ja että sitä saataisiin paranneltua ajan myötä lisäämällä ympäristöön uusia moduuleita tai antureita. Toivon myös, että TwinCAT-ohjelma tulisi tutuksi oppimisympäristön käyttäjille ja että käyttäjät voisivat hyödyntää tästä saatua oppimista myös muissa tulevaisuuden projekteissaan.

LÄHTEET

Barnett Donald E. 2014, Resolver vs Encoder, Hakupäivä 17.11.2014
<<http://www.optoresolver.com/>>

Beckhoff Automation, 2014, Beckhoff, Hakupäivä 3.11.2014
<<http://www.beckhoff.com/>>

Beckhoff Automation, 2014, Beckhoff Suomessa, hakupäivä 3.11.2014
<<http://www.beckhoff.fi/fi/beckhoff/default.htm>>

Beckhoff Automation, 2013, TwinCAT 2 manual v.3.0.1, PDF-tiedosto.

Beckhoff Automation, 2014, TwinCAT 3, Hakupäivä: 3.11.2014
<<http://infosys.beckhoff.com/content/1033/tcinfosys3/html/startpage.htm?id=12962>>

Fonselius Jaakko, 1998, Koneautomaatio. Servotekniikka. Helsinki: Opetushallitus.

Hendricks Howard, Morley Dick, Hakupäivä 3.11.2014
<<http://www.barn.org/FILES/historyofplc.html>>

Konedirektiivi, Euroopan parlamentin ja neuvoston direktiivi 2006/42/EY

Kosonen Harri, 2012, Säätimet, Hakupäivä 17.11.2014
<http://cna.mamk.fi/Public/KosonenH/sähkö_ ja_ automaatio/PID_säätimet/>

Pesu Jouko, 2010, Putkenkatkaisukone, ohjauksen muutos logiikka- ja servokäyttöksi, Seinäjoen ammattikorkeakoulu, Opinnäytetyö

Pesu Jouko, Valtion Teknillinen Tutkimuskeskus, 2010, AC-servomoottorin rakenne, vikaantuminen, havaitsemismenetelmät. PDF-tiedosto.

Real Time Automation, 2014 EtherCAT Protocol Overview. Hakupäivä 31.10.2014
<<http://www.rtaautomation.com/ethercat>>

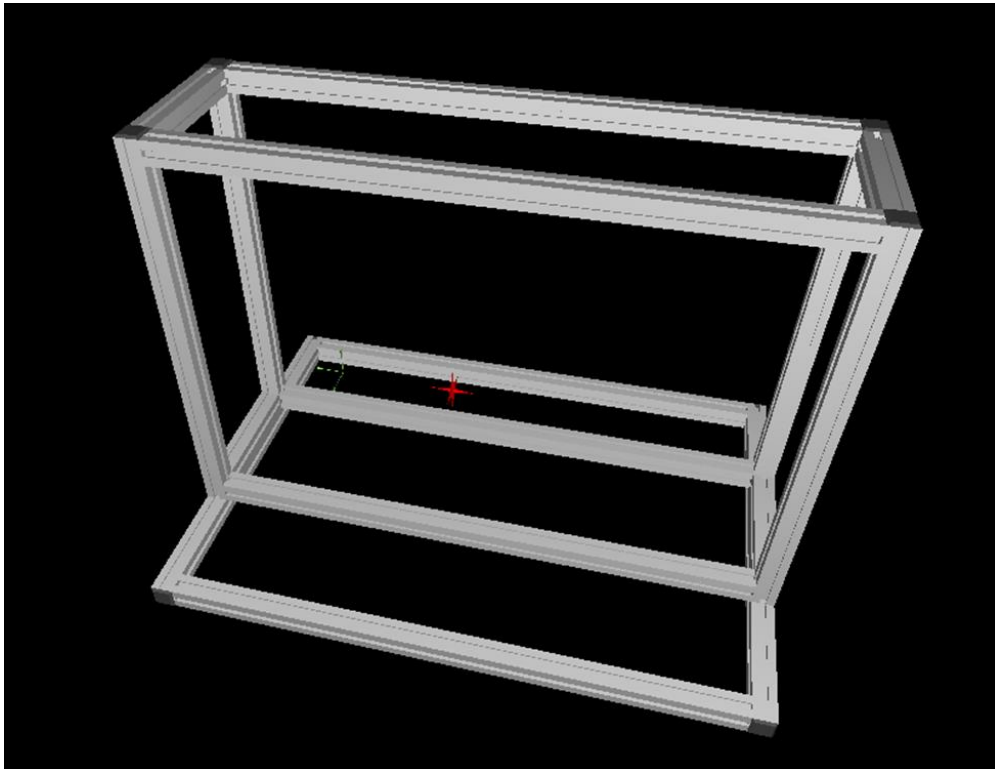
Ross Kevin, 2004 What's a servo? Hakupäivä 3.9.2014
<<http://www.seattlerobotics.org/guide/servos.html>>

Storr Wayne, 2014, Feedback systems, Hakupäivä 17.11.2014 <<http://www.electronicstutorials.ws/systems/feedback-systems.html>>

Tamagawa seiki co, Rotary encoders, Hakupäivä: 3.11.2014
<<http://www.tamagawa-seiki.co.jp/english/encoder/>>

LIITTEET

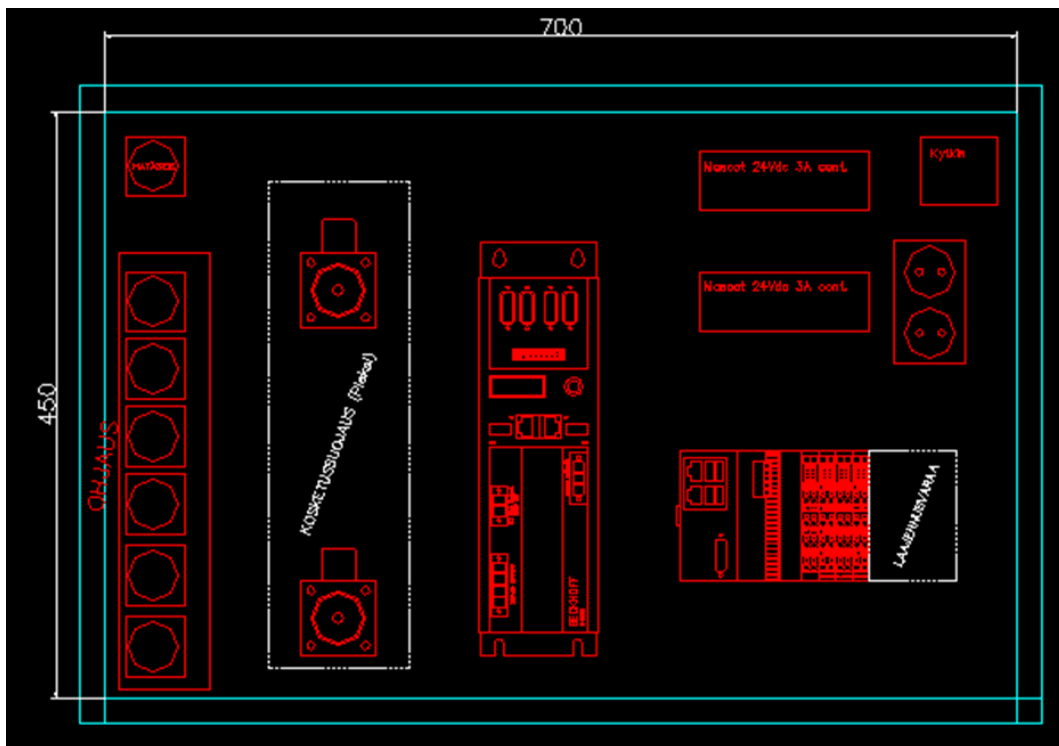
- Liite 1. 3D-malli, Asennuskehikko
- Liite 2. CADS-luonnos, Ensimmäinen vedos
- Liite 3. CADS-luonnos, Vaihtoehtoinen servovahvistimen sijoitustapa
- Liite 4. CADS-luonnos, Kaksiosainen ympäristö, logiikkaosuus
- Liite 5. CADS-luonnos, Servomoottorihissi
- Liite 6. Sähköpiirustus, Beckhoff oppimisympäristö
- Liite 7. Beckhoff oppimisympäristön käyttöohjeet



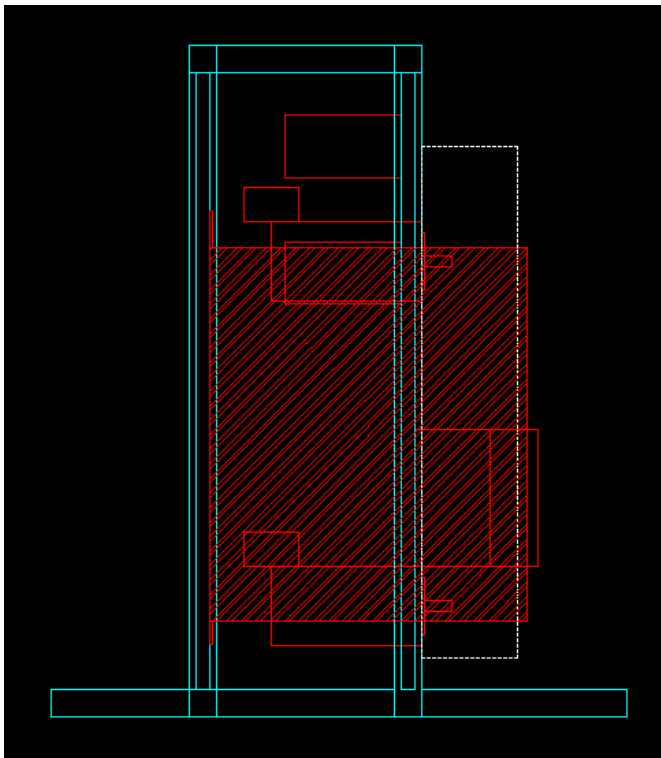
Kuva: Asennuskehikon 3D-malli.



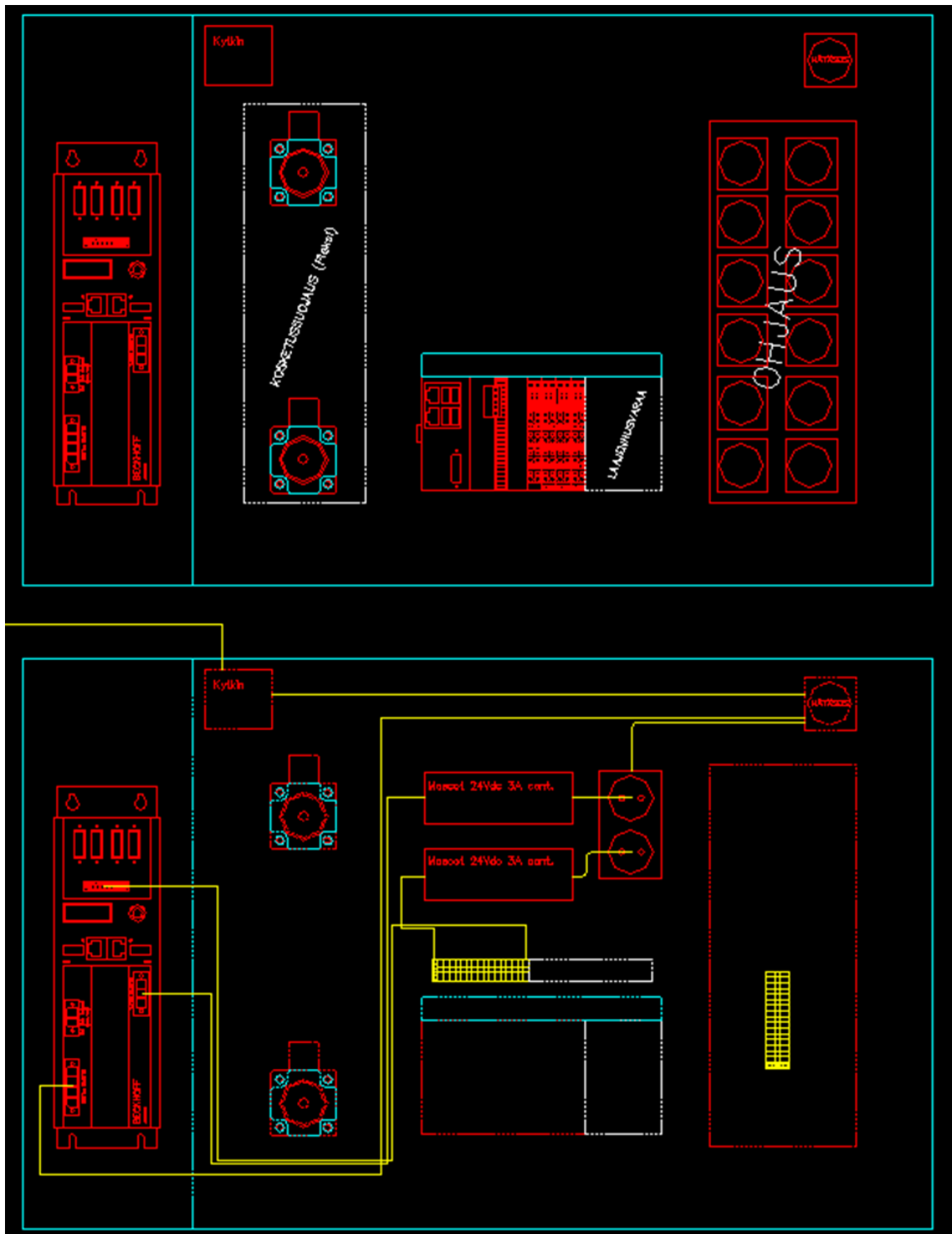
Kuva: 3D-mallinnus siitä miten mahdolliset levyt sijoitettaisiin. Levyt poikkeavat lopullisesta suunnitelmista.



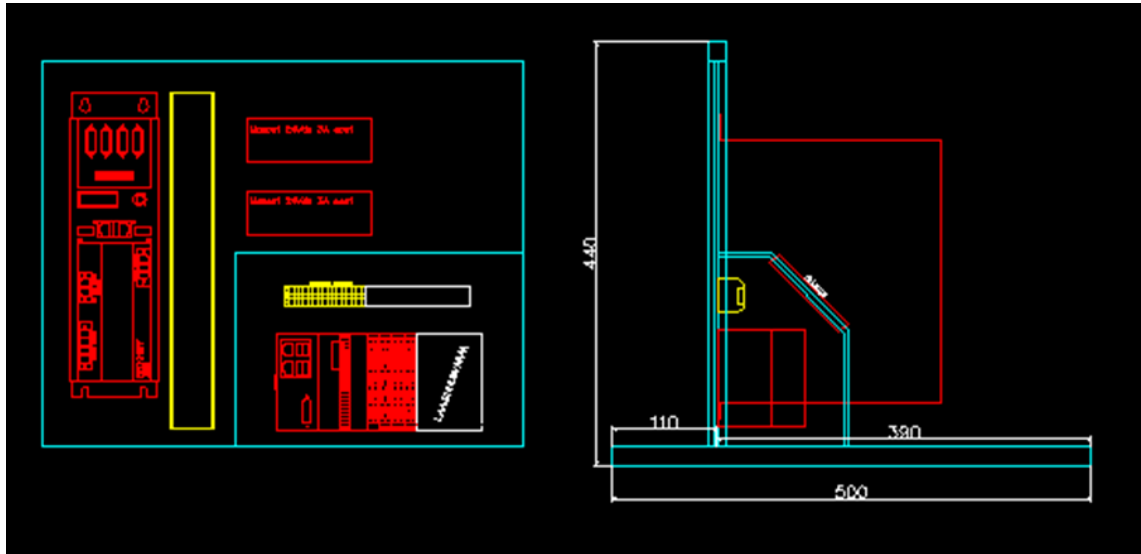
Kuva: Ensimmäinen vedos mahdollisesta laitteiston sijoittamisesta.



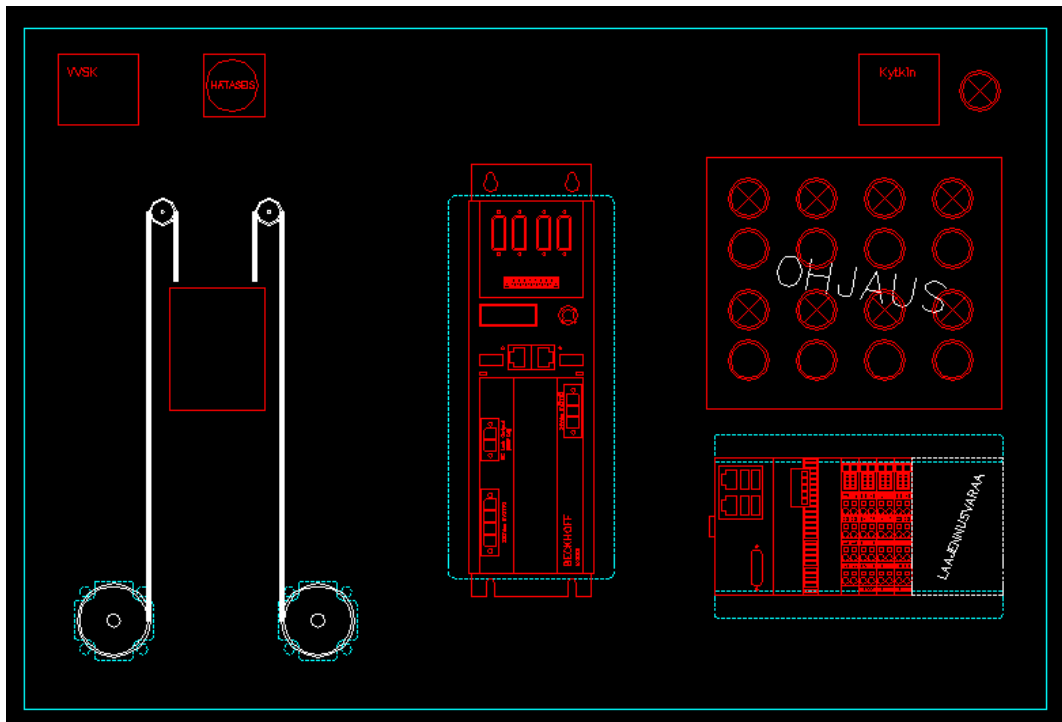
Kuva: Toisen levyn lisääminen suunnitelmiin servovahvistimen koon takia. (Kuvassa viivoitettuna).



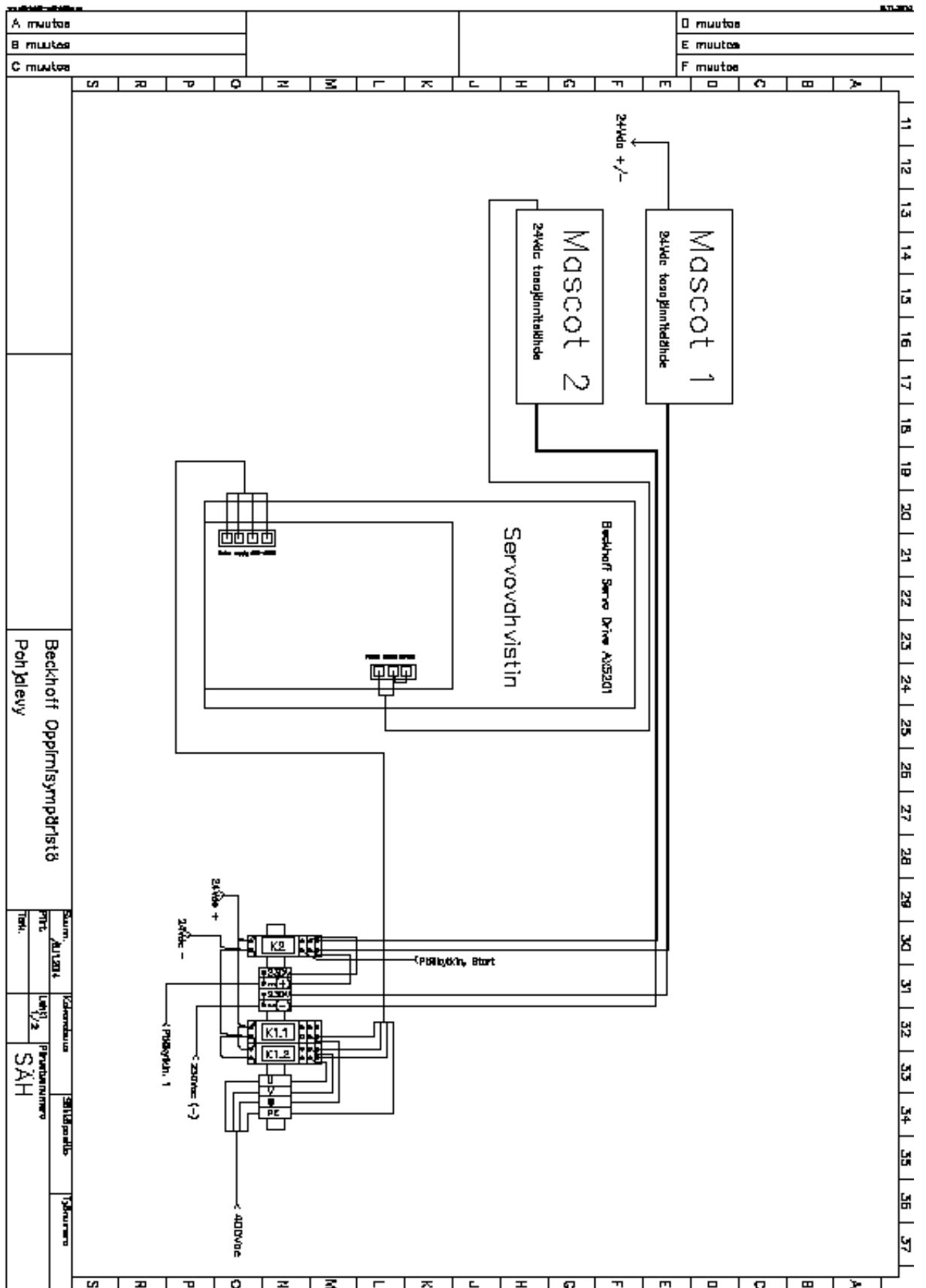
Kuva: Vaihtoehtoinen suunnitelma servovahvistimen sijoituksesta. Ylemmässä luonnoksessa on ympäristön julkisivu, alemmassa luonnoksessa pohjalevy.



Kuva: Luonnos mahdollisesta kaksiosaisesta oppimisympäristöstä. Servomootorit olisi sijoitettu omalle pienemmälle asennuskehikolle.



Kuva: Servomootorihissi-idea, joka toimii pohjana lopullisessa työssä. Hissin väkipyörät korvataan moottoreiden yläpuolelle sijoitettavilla hammaspyörillä. Häätäseis ja VVSJK poistettiin ympäristölle tarpeettomina.



Beckhoff Oppimisympäristö
Pohjoisin

Suunn. ALI/2014
PRT. 1/2
Teh. SÄH

Kaivomäärä: 3811/2015
Puhelinnumero: 708000000

Beckhoff logiikan ja servomoottorioppimisympäristön käyttöohjeet

Oppimisympäristön käynnistäminen ja siihen liittyvät mahdolliset ongelmatilanteet:

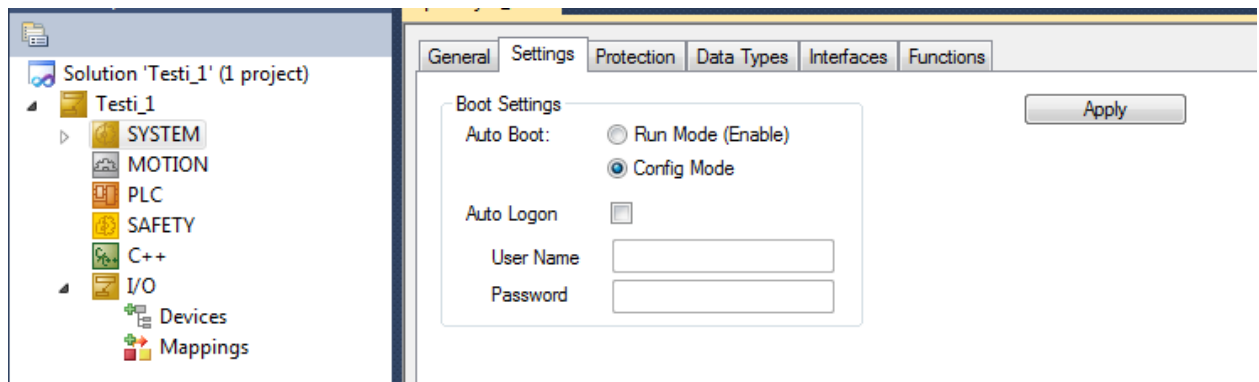
Kun kytket ympäristöön jännitteen, pidä kytkintä START asennossa 2-3 sekuntia, näin tasajännitelähteet ehtivät käynnistyä ja pitopiirit pääsevät vetämään. Logiikkaan, servovahvistimeen, sekä kytkimen vieressä oleva merkkivalo syttyvät tässä tapauksessa.

Jos logiikkaan ei ole ohjelmoitu ohjelmaa, tai jos ohjelmaa ei ole asetettu käynnistymään logiikan käynnistymisen yhteydessä, on mahdollista että servovahvistin välkyttää SYNC LOST virhekoodia. Tällä ei yleensä ole väliä, sillä ohjelman saatua uudestaan yhteyden moottoreihin, virhe kuittaautuu itsestään.

Virheet voidaan kuitata manuaalisesti menemällä kohtaan M>Error Reset servovahvistimen nuolinäppäimillä, sekä kuitata virhe pitämällä painikkeen keskiosaa hetken pohjassa. Virhe täytyy kuitata molemmille kanaville (ylös/alas näppäimet, channel A/channel B).

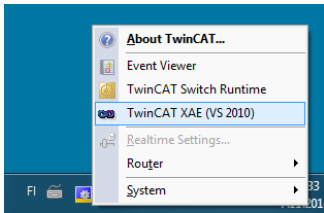
Logiikalle täytyy antaa aikaa käynnistyä, käynnistysvaiheessa logiikan HDD ledi palaa punaisena, tämä tarkoittaa että logiikka käynnistää sisäistä Windows käyttöjärjestelmäänsä. Logiikka on täysin ladannut kun TC valo syttyy. Logiikkaa ei tulisi sammuttaa kun käyttöjärjestelmä vielä latautuu.

Jos TC valo syttyy sinisenä, logiikka on config tilassa, tämä tarkoittaa että logiikka on ohjelmointivalmiudessa. Logiikka voidaan ohjelmoinnin yhteydessä asettaa käynnistymään run tilaan, jolloin logiikalle ladattu ohjelma käynnistyy logiikan myötä. Tällöin jännitteen kytkettyä ja käyttöjärjestelmän ladattua, TC valo palaa vihreänä.



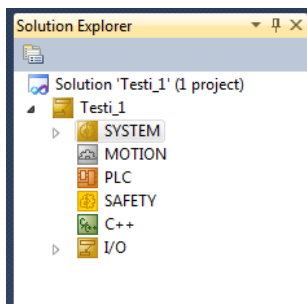
Yhteys logiikkaan ja ohjelmoinnin aloittaminen:

Ohjelmoinnin aloittamiseksi logiikan X00 ethernet paikan ja tietokoneen välille täytyy kytkeä >CAT5 kaapeli. Tietokoneessa täytyy myös olla TwinCAT 3.0 ohjelma, (tai parempi). Ohjelma käynnistetään ilmoitusalueen kuvakkeestaan, kohdasta TWINCAT XAE (VS 2010).



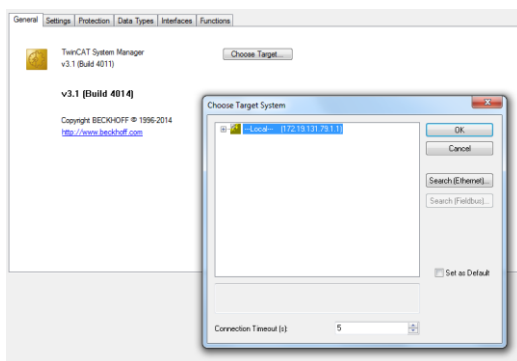
Jos ohjelmaa ei ole käytetty koneella ennen, MS Visual Studio joutuu suorittamaan alkutoimintonsa, ja tässä voi kestää hetki.

Ohjelma sen jälkeen käynnistyy ja lataa Start Page aloitussivun. Sivun lataaminen epäonnistuu joillakin koneilla, jolloin ruudussa näkyy "Content Load Error" virheviesti. Tämä viesti ei vaikuta ohjelman toimintaan mitenkään. Käynnistä uusi projekti File > New > Project kohdasta.



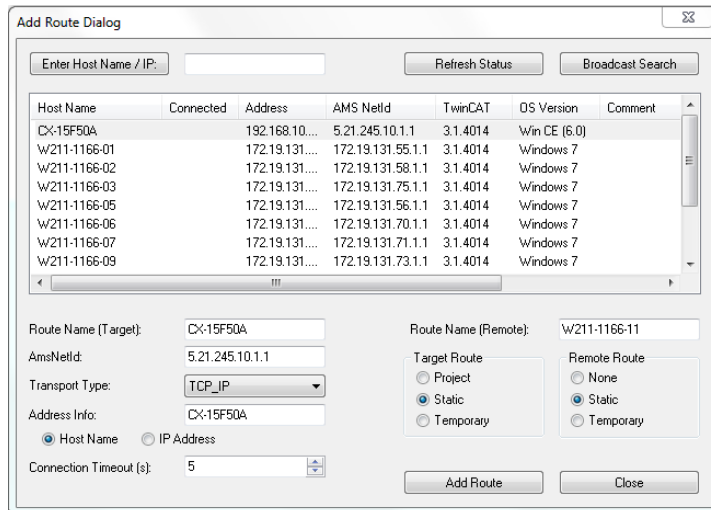
Projektiin sisältyvät kohdat näkyvät Solution Explorer kohdassa, ja tupla klikkaamalla SYSTEM kohtaa päästään projektin yleisasetuksiin.

Jotta logiikan ohjelmaan voitaisiin vaikuttaa, täytyy tietokoneen ja logiikan välille saada yhteys. Tämä voidaan tehdä "SYSTEM" kohdan "general" välilehden kautta. Choose Target... > Choose Target System > Search (Ethernet)...



Beckhoffin logiikat ovat nimetty tyyliin CX-000000, ja tämän ympäristön logiikan nimenä on CX-15F50A. Valitse se listasta.

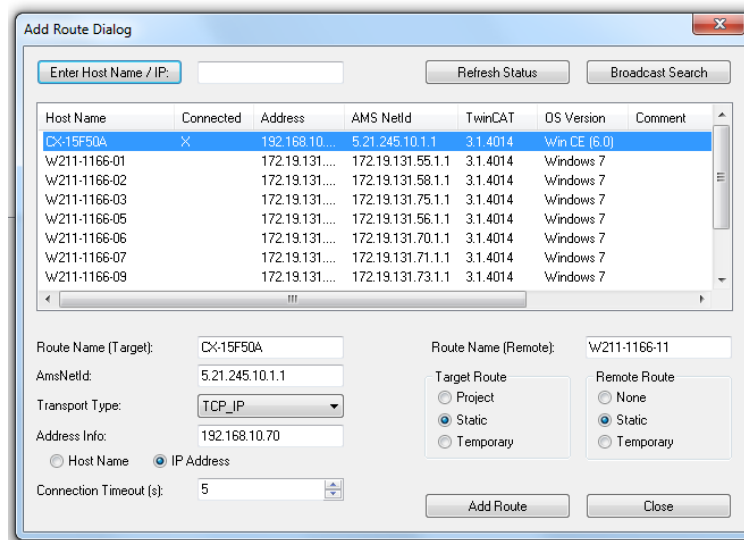
Jos logiikka ei näy listassa, ja vika ei ole verkkokortissa, voi logiikka olla hakenut itselleen väärän IP-osoitteen DHCP:n kautta. Tässä tapauksessa tulisi kääntyä IT-tuen puoleen.



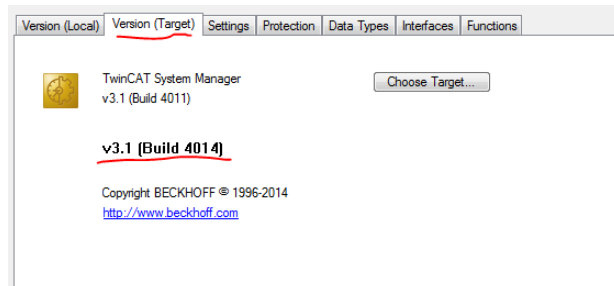
Tämän jälkeen kysytään käyttäjätunnusta ja salasanaa. Molemmat voidaan jättää tyhjäksi.

Jos yhteyden muodostaminen epäonnistuu asetuksia muuttamatta, toimii usein "Address Info" kohdan vaihtaminen Host nimestä > IP addressiksi.

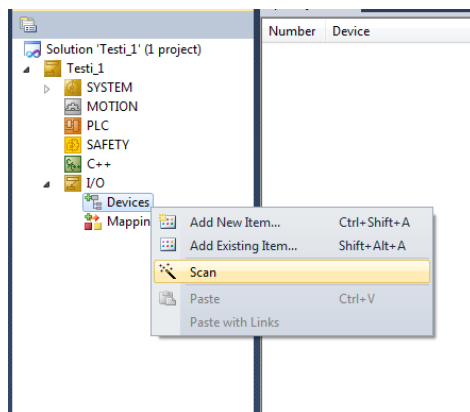
Yhteyden saaminen on ehdottomasti ongelmallisin vaihe, sillä Route Dialogin suljettua ohjelma usein käyttäytyy kuin yhteys olisi, vaikka jostain syystä yhteyttä ei saatukaan. Tarkista silloin onko yhteys varmasti muodostettu. Yhteys on varmasti saavutettu kun Connected kohtaan ilmestyy "X".



Toinen tapa todentaa että yhteys on saavutettu, on lisäämisen jälkeen ilmestyvä ”Version (Target)” kohdan versionumero. Tämä kohta näyttää jäävän aina tyhjäksi jos yhteys ei ole onnistunut.



Kun yhteys on saavutettu ja logiikka asetettu CONFIG tilaan, voidaan ympäristön laitteet skannata projektiin. Scan komentoa ei ole mahdollista valita jos yhteyttä ei ole onnistuttu muodostamaan tai logiikka on RUN tilassa.



Skannauksen avulla kaikkien laitteiden pitäisi löytyä kuin itsestään, Vastaa kyllä kaikkiin kohtiin, ja ohjelman pitäisi löytää kaikki logiikan lohkot, sekä moottorit, ja automaattisesti lisätä ne projektiin.

Äärimmäisessä ongelmatapauksessa yhteydenmuodostamisen kanssa, kannattaa kääntyä IT-tuen puoleen. Logiikan Windows CE järjestelmään voidaan päästä käsiksi CERHOST.EXE ohjelmalla, jolla voidaan ottaa etäyhteys CE järjestelmiin jotka ovat samassa verkossa. Yhteyden muodostamiseen tarvitaan joko laitteen nimi tai IP osoite. Ohjelman löytää alla olevasta linkistä, tai AMK_Tekniikka aseman Twincat 3 kansioista.

http://infosys.beckhoff.com/english.php?content=../content/1033/sw_os/html/cx1000_os_ce_remotedisplay.htm&id=

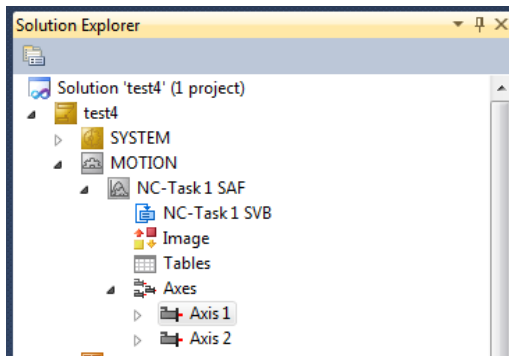
Logiikalle voidaan antaa tämän ohjelman kautta mm. oma IP-osoite.

Servomootoreiden testaus/pyörittäminen

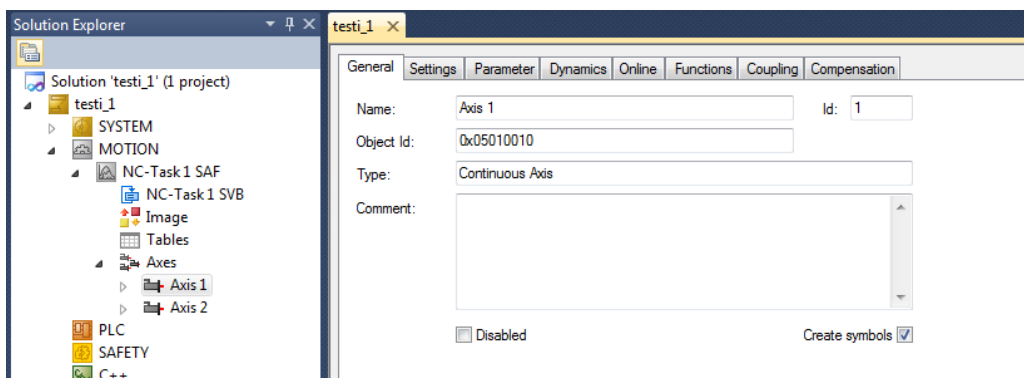
TwinCAT ohjelmalla on mahdollista sekä ohjata servomootoreita, sekä logiikan digitaalilähtöjä ilman että ohjelmaa ladataan logiikalle. Tämä toimii ohjelman ollessa niin sanotussa Free Run State-tilassa. Oikeassa ylälaudassa olevassa työkalurivistä on mahdollista laittaa tämä tila päälle. Tilan kuvakkeessa on ympyrän kehää kiertävä punainen nuoli. Joissakin tapauksissa on mahdollista että tätä kuvaketta ei voida valita, mutta Free Run State tuntuu silti olevan toiminnassa.



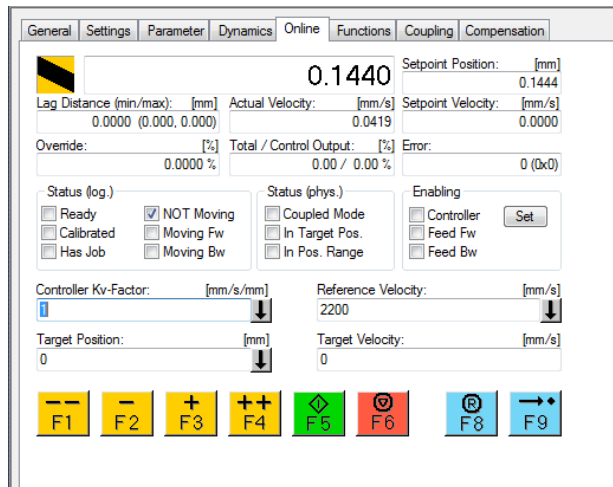
Servomootoreita voidaan hallita MOTION kohdan alta, Axes -> Axis 1 & Axis 2 kohdasta.



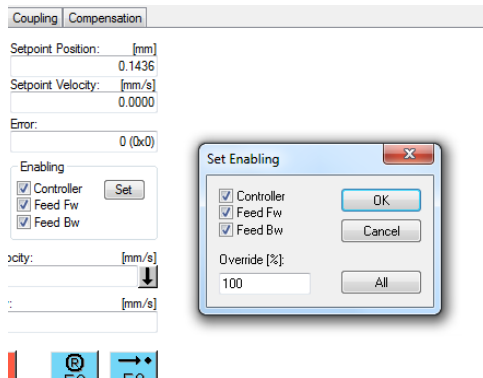
Valitun moottorin akselin erinäiset välilehdet tulisi tulla näkyviin. Jotkin välilehdistä on piilotettu, ja näkyvät vain jos "Show Online Data" kuvaketta on painettu. (kuvakkeena silmälasit, free run staten vieressä).



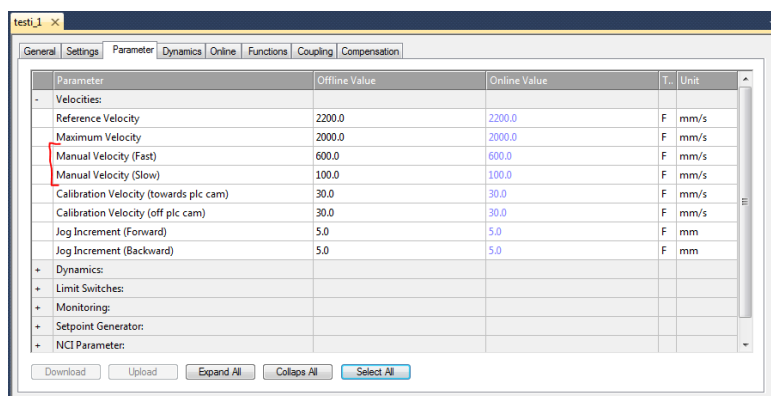
Tärkeimmät näistä välilehdistä ei ilmesty näkyviin ilman "show online dataa". Moottoria voidaan ohjata "Online" välilehdestä. Jos jostain syystä online tilan napit eivät ole väreissä, paina ensin "activate configuration" kuvaketta, joka löytyy samasta työkalurivistä kuin "show online datakin". Näin moottorin parametrit saadaan päivitettyä.



Tämän ikkunan kautta moottoria voidaan pyörittää molempiin suuntiin, se voidaan ajaa nollassa pisteeseen, ja muuten testata moottorin toimintaa. Moottoreita voidaan esimerkiksi ohjata eri suuntiin F1-F4 painikkeita pohjassa pitämällä. Mutta jotta nämä näppäimet toimivat, on moottorille asetettava lupa ajaa. Tämä tapahtuu Enabling kohdan "Set" napista.

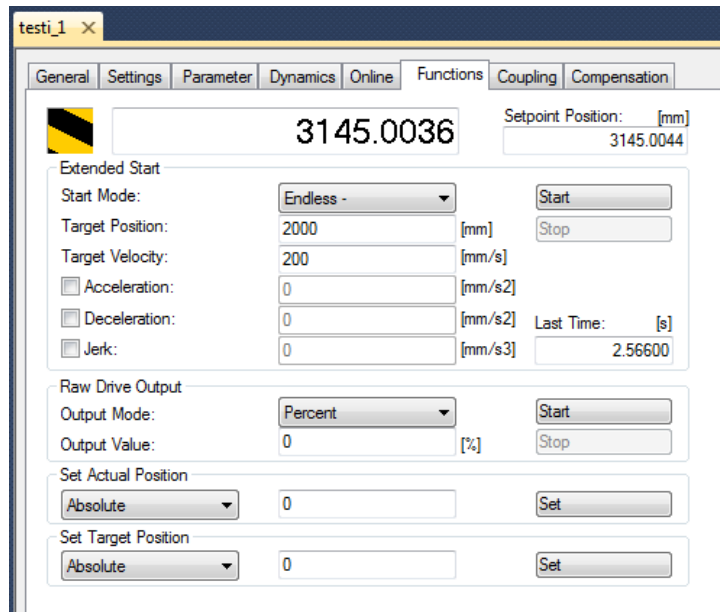


Nappia painamalla ilmestyy uusi ikkuna, jossa kysytään mitä toimintoja halutaan sallia. Valitse kaikki, ja varmista että Override prosentti asetetaan 100:aan. Nyt välilehden näppäimet toimivat. Painamalla näppäimiä F2 ja F3, voidaan moottoria pyörittää hitaasti, ja näppäimillä F1 ja F4 nopeasti. Pyöriksen nopeudet riippuvat servomoottorille asetetuista Manual Velocity parametreista. Näitä, ja muita moottorin pyörittämiseen liittyviä parametrejä voidaan muokata "parameters" välilehdestä.



Muokkaukset täytyy aina päivittää moottoreille käyttämällä ”Activate Configuration” kuvaketta.

Toinen tärkeä välilehti on functions, josta voidaan asettaa esimerkiksi endless+/- tilat, jossa moottori pyörii myötä-/vastapäivään halutulla nopeudella koskaan saavuttamatta päätepistettä.

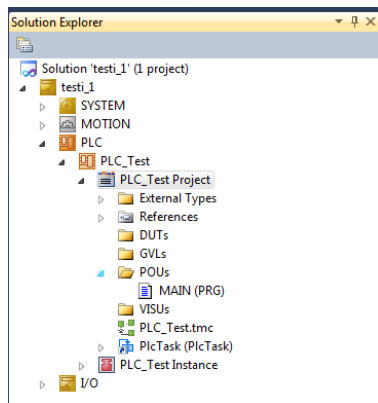


Start Mode kohdassa on tarjolla useita eri pyöristystapoja. Moottoria voidaan ajaa tiettyyn pisteeseen relative & absolute positionin mukaan, tai endless moodissa pyörittää haluttuun suuntaan loputtomasti. Reversing sequencessa moottorit pyörivät kahden niille asetetun kohde- etäisyyksien välillä. Start/Stop sequencessa pysähdytään aina kun haluttu matka on saatu kuljettuaan.

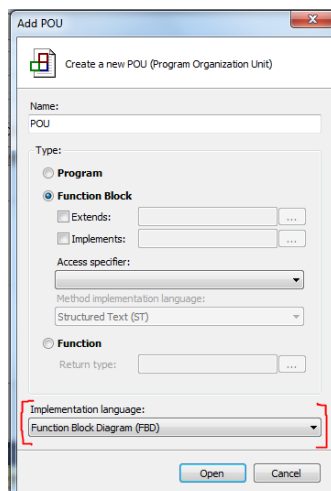
Joskus on tarvetta että moottoria ajetaan aina tiettyyn kulmaan pyörimisetäisyyksien sijaan. Settings välilehdestä on mahdollista vaihtaa millimetrit johonkin muuhun muotoon, kuten asteisiin. Nämä muutokset vaikuttavat mm. functions välilehden paikka- ja nopeusarvoihin.

Ohjelmoinnin aloittaminen

Ohjelmointi voidaan aloittaa lisäämällä PLC kohdasta uuden PLC projektin. Vaihtoehtoina on joko täysin tyhjä PLC projekti, tai standardi projekti, jossa on valmiina tyhjä MAIN(PRG), sekä valmiit kansiot joihin kaikki projektiin kuuluvat osat voidaan järjestää.

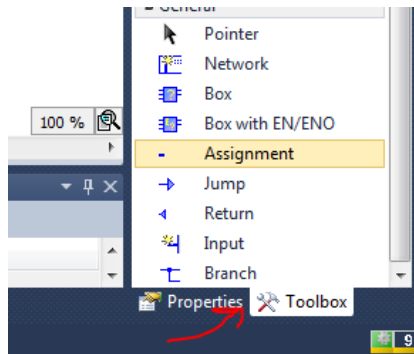


Projektiin voidaan lisätä lisää Programmeja tai Function Blockkeja painamalla oikeaa hiiren nappia Project kohdan päällä, ja menemällä Add > POU... kohtaan. Tässä kohdassa on mahdollista valita uuden osion ohjelmointikieli.



Itse ohjelmoinnista löytyy kattava PDF-manuaali Beckhoffin internet-sivuilta. Internetin puolelta on myös löydettävissä quick start opas ohjelmoinnin aloittamiseen. Tämän dokumentin kirjoittamisvaiheessa on myös mahdollista löytää nämä tiedostot jaetulta AMK_Tekniikka asemalta, Beckhoff kansioista.

Jos ohjelmointikielenä toimii Function block diagram, kirjasto tavanomaisista toimilohkoista löytyy oikeasta reunasta vaihtamalla ”properties” välilehti ”toolbox”iin. Ensimmäisellä kerralla kestää hetki ladata kaikki toimilohkot muistiin.



Esimerkki ohjelman luonnista

Ohjelma voidaan tehdä suoraan MAIN(PRG) lohkon sisään. POUt voidaan avata yksinkertaisesti kaksoisklikkaamalla niiden nimeä. Lohkon avaaminen tuo esiin kaksiosaisen näkymän ruudun keskelle. Ylempään voidaan osoittaa ohjelman muuttujat, sekä alempaan tehdä itse ohjelma. Standardi projektin oletus POU käyttää ohjelmointikielensä Structured Textiä. Tämän lohkon voi poistaa ja luoda uudestaan, jos tahdotaan käyttää toista ohjelmointikieltä.

Logiikassa on 8 digitaalituloa, ja 8 digitaalilähtöä, ja nämä ovat yhteydessä kahdeksaan eri painikkeeseen, sekä kahdeksaan eri merkkivaloon. Näille tarvitsee luoda muuttujat.

Muuttujat voidaan luoda VAR, ja END_VAR kohdan väliin. Oletetaan että tahdotaan luoda muuttuja kolmannelle napille. Nimeämiskäytäntönä on hyvä osoittaa ensin muuttujan tyyppi, input I_, sekä nimi josta sen tunnistaa, S3, kolmas painike. Muuttujan osoittaminen tehdään vielä kuvan osoittamalla tavalla. I_S3 on yksi inputeista, sekä tyybiltään bitti. Digitaalituloja ja -lähtöjä käytettäessä tulee käyttää BYTEä.

```

TEST* x MAIN testi_1
1 PROGRAM TEST
2 VAR
3 I_S3 AT %I*:BYTE;
4 END_VAR
5

```


Lähtömuuttujat osoitetaan juuri samaan tapaan. Q_H6, eli lähtö, kuudes merkkilamppu, on output & bitti.

```

TEST* x MAIN testi_1
1 PROGRAM TEST
2 VAR
3     I_S3 AT %I*:BYTE;
4     Q_H6 AT %Q*:BYTE;
5 END_VAR
6

```

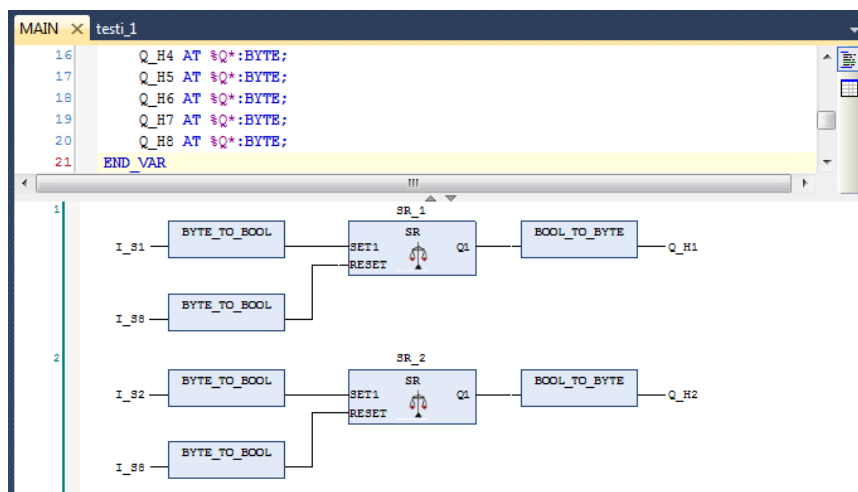
VAR kohdan sijaan kannattaa luoda sekä VAR_INPUT, ja VAR_OUTPUT kohdat, joihin nämä inputit saadaan kätevästi sijoitettua. Kuvassa näkyvistä [-] merkeistä voidaan inputit ja outputit piilottaa. Tästä on etua kun käytössä on useampi eri muuttuja. VAR kohdan voi jättää ohjelman sisäisille muuttujille.

```

MAIN x testi_1
1 PROGRAM MAIN
2 VAR_INPUT
3     I_S1 AT %I*:BYTE;
4     I_S2 AT %I*:BYTE;
5 END_VAR
6 VAR_OUTPUT
7     Q_H5 AT %Q*:BYTE;
8 END_VAR
9 VAR
10 END_VAR
11
12 Q_H5 := I_S1 AND I_S2;
13

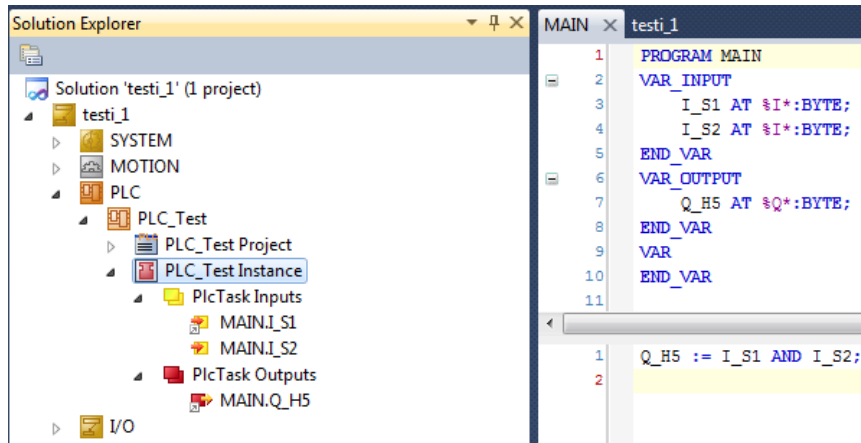
```

Kuvassa on luotu yksinkertainen AND-porttia vastaava kytkentä ST-kielillä. Painikkeita S1 ja S2 pohjassa pitämällä merkkivalo H5n tulisi syttyä. Painikkeita on pidettävä pohjassa jotta valo palaisi, tämän voi korjata yksinkertaisesti lisäämällä Set/Reset toiminnot.

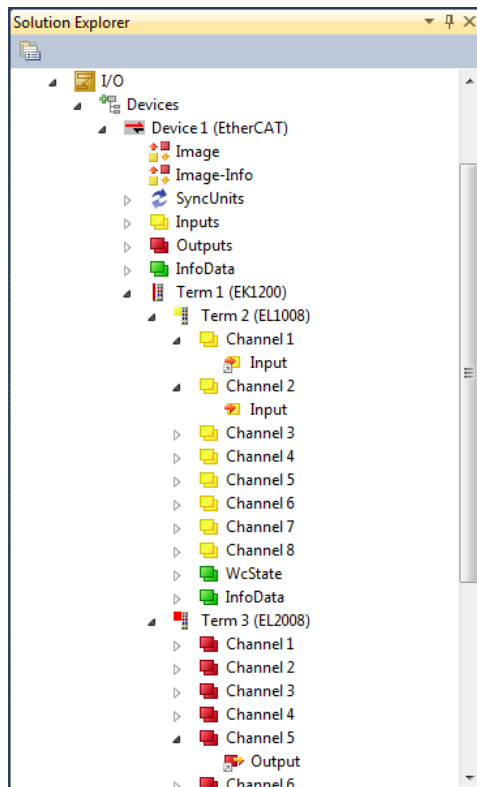


Set-Reset lohkojen käyttö vaatii boolean muotoisen muuttujan käyttöä. Kuvassa esimerkki kuinka muuntolohkot toimivat.

Huomaa kuinka ohjelmassa ei missään viitata logiikan tuloihin ja lähtöihin. Tämä tehdään linkittämällä inputit ja outputit keskenään. Tämä tapahtuu "Instance" kohdassa. Luodut, määrittelemättömät muuttujat ilmestyvät tänne. Muuttujat voidaan linkittää jonkin muuttujan kanssa valitsemalla "Change Link..." oikean hiirenpainikkeen valikosta.

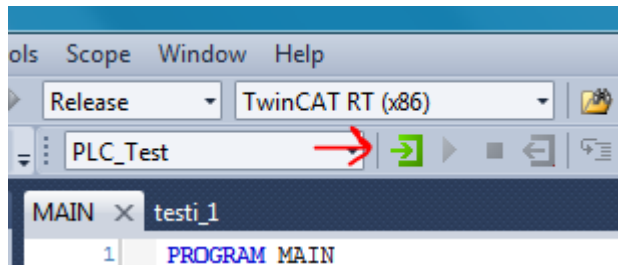


Kun linkitys on tehty, ilmestyy pieni pikakuvake symboli muuttujaan. Huomaa I_S1 ja I_S2n ero. Nämä linkit näkyvät sekä Instance kohdan listassa, että Device kohdan logiikan moduulien listoissa. Logiikan kautta on hyvä tarkistaa että muuttuja on varmasti linkitetty oikeaan osoitteeseen.

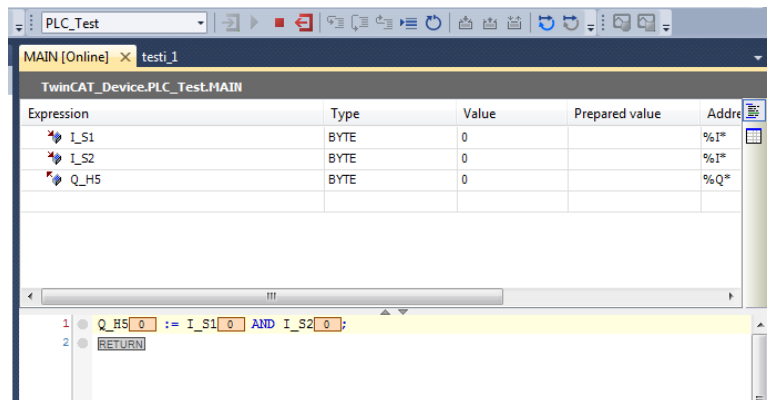


Tämän jälkeen voidaan PLC projekti rakentaa uudelleen (Build/Rebuild) Ja valita "Activate Configuration", nyt ohjelman pitäisi latautua logiikalle ja logiikan siirtyä RUN tilaan.

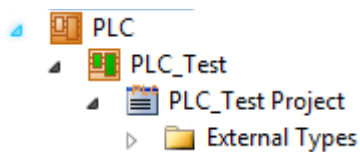
Ruudun keskialueella olevaan työkaluriviin pitäisi syttyä uusi kuvake Login.



Painamalla nappia ja sen vieressä syttyvää Start komentoa saadaan ohjelma ajettua. Nyt logiikalle viety ohjelma on käytettävissä. Ohjelmointi-ikkuna muuttuu myös näyttämään muuttujien sen hetkisen arvon.



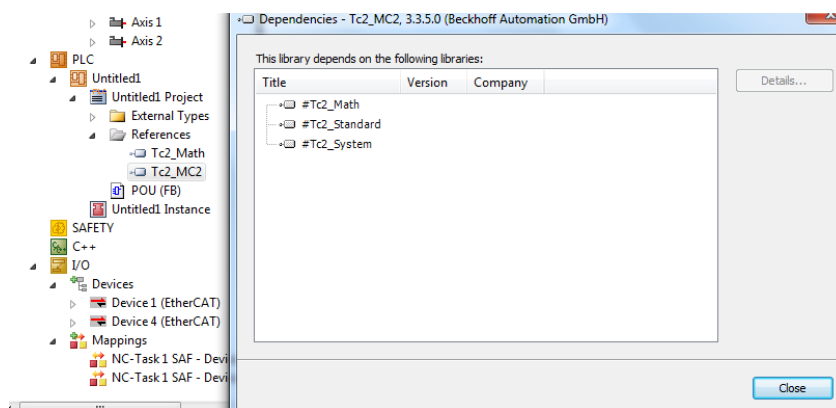
PLC symboli muuttuu myös ohjelman tilan mukaan.



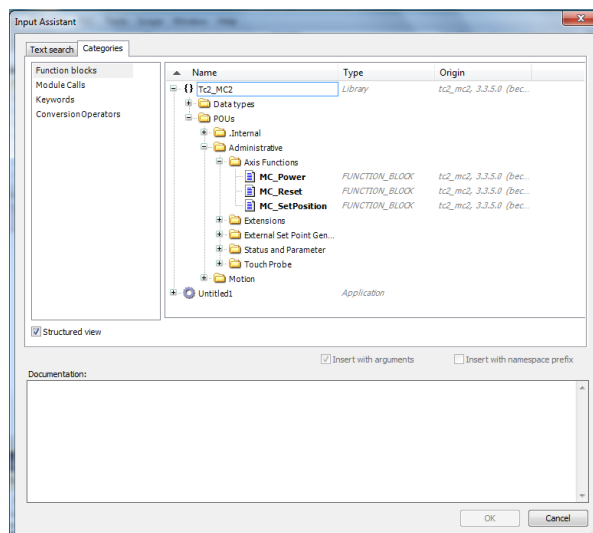
Vihreä valo oikeassa neliössä tarkoittaa että ohjelma on toiminnassa logiikalla, punainen taas että ohjelma on pysäytetty.

Servomoottorin hallinnan ohjelmointi

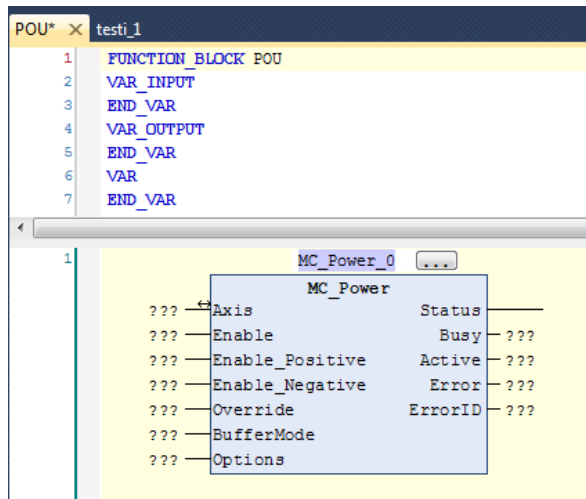
Tärkeää! References kohdasta voidaan lisätä kirjastoja projektiin, näin voidaan tarvittaessa käyttää epätavallisempia toimilohkoja. Servomoottoareiden ohjaamisessa tarvittava toimilohko löytyy Motion > PTP > Tc2_MC2. Tämä kirjasto vaatii toimiakseen muita kirjastoja, mm. Tc2_Math. Lista tarvittavista kirjastoista löytyy lisäämällä tämä kirjasto, sen jälkeen klikkaamalla tätä oikealla hiiren napilla, valikon ”Dependencies” kohdasta.



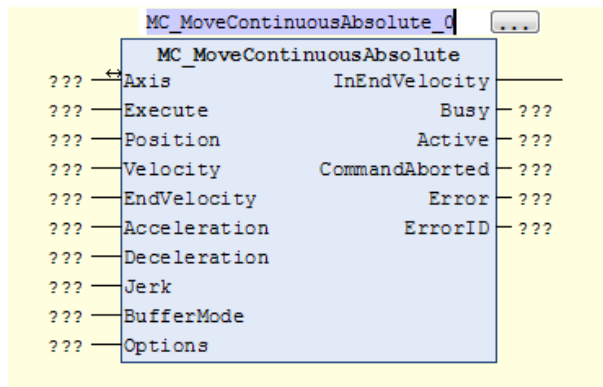
Haluttuun networkkiin lisätään moottorin tarvitsemat lohkot painamalla oikeaa hiirennäppäintä lohkossa, ja valitsemalla Insert Box. Kirjasto Tc2_MC2 on nyt ilmestynyt valikkoon.



Kirjaston sisältä löytyy toimilohkoja joilla moottoria voidaan ohjata. Esimerkiksi MC_Power toimilohkon avulla akselille voidaan antaa ohjausluvut.



Enable, Enable_Positive, Enable_Negative, sekä Override ovat samat komennot kuin akselien Online välilehden takaa löytyvät Set Enablet.



Lisäksi löytyy muitakin toimilohkoja jota kautta ohjausta voidaan suorittaa. Kuvassa Motion > Point to Point Motion kohdan alta löytyvä MC_MovecontinuousAbsolute joka vastaisi Functions välilehden Endless+/- tilaa.

Näillä ohjeilla sekä Beckhoffin omia manuaaleja tulkitsemalla, pitäisi päästä alkuun servomoottoriin liittyvän ohjelmoinnin kanssa. Beckhoffin ohjelmointikielet noudattavat IEC-standardia, joten myös TwinCAT3 ohjelman ulkopuolisista ohjeista voi olla apua.

Kuljetinhihnojen kiristäminen:

Joissakin tapauksissa, kuten esimerkiksi pitkän käyttötaun, oppimisympäristön liikuttamisen yhteydessä tapahtuvan kolhiintumisen, tai liian rajun moottoreiden ajamisen takia on mahdollista että kuljetinhihna löystyy. Jos hihna on päässyt löystymään niin paljon että on tarvetta kiristää hihnoja, tee laitteisto jännitteettömäksi sekä anna tarvittava aika mahdolliselle sähkövarauksen purkaantumiselle, joka voi servovahvistimen kanssa olla 5 minuuttia.

Irrota suojamuovi moottoreiden päältä, ja tukemalla hammaspyörän takana olevaa mutteria, löysää sisäpuolen mutteria, ja nosta pulttia samalla. Kiristä sisäpuolen mutteri levyn vastapuolella olevaa mutteria vasten, sen sijaan että kiristetään koko pulttia. Näin hammaspyörille jää liikkumavaraa. Tarkista että pultti on suorassa kiristämisen jälkeen, muuten hihnan kireys hiukan vaihtelee liikkeessä.