



samk

Satakunnan ammattikorkeakoulu
Satakunta University of Applied Sciences

HENRI NYSTRÖM

Hyllyvarastoa palvelevan robotin ohjelman modernisointi

SÄHKÖ- JA AUTOMAATIOTEKNIIKAN
TUTKINTO-OHJELMA
2021

Tekijä(t) Nyström, Henri	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä Syyskuu 2021
	Sivumäärä 25	Julkaisun kieli Suomi
Julkaisun nimi Hyllyvarastoa palvelevan robotin ohjelman modernisointi		
Tutkinto-ohjelma Sähkö- ja automaatiotekniikan koulutusohjelma		
<p>Tämän opinnäytetyön tavoitteena oli toteuttaa Satakunnan ammattikorkeakoulun automaatiolaboratoriossa sijaitsevaan Beckhoffin komponenteista koostuvan varasto laitteiston ohjaus soveltamalla TwinCAT 3 ohjelmaa. Ohjelman täytyi siirrellä tuotteita varastosta kuljettimelle ja toisin päin. Vaatimuksena oli, että käyttäjän tarvitsee antaa vain varastopaikan koordinaatit ja tieto siitä haetaanko tuote varastosta kuljettimelle vai toisin päin. Varastossa olevien tuotteiden tiedot tulitaisiin tallentamaan ylätason ohjelmaan. Työn pohjana oli jo olemassa oleva ohjelma, johon tuli ensin tutustua.</p> <p>Työssä perehdytään projektissa käytettyihin Beckhoffin eri komponentteihin, TwinCat ohjelmointiin ja PLCOpen Motion Control ohjelmointilohkoihin. Ohjelman tekoon käytettiin Beckhoffin TwinCat 3 ohjelmaa. Ohjelmoinnissa uudelleenkäytettiin ohjelman osia. Käytettiin käyttäjän tekemiä tietotyyppejä (DUT) vähentämään erillisiä muuttujia ja yleiskäyttöistä toimilohkoa, jota voitiin käyttää kaikkien akselien ohjauksien toteuttamiseen.</p> <p>Työn tavoitteet saavutettiin. Varastoon tehtiin ohjelma, joka siirtää tuotteita halutulla tavalla.</p>		
<u>Asiasanat</u> Liikkeenohjaus, Beckhoff, PLC		

Author(s) Nyström, Henri	Type of Publication Bachelor's thesis	Date September 2021
	Number of pages 25	Language of publication: Finnish
<p>Modernization of a robot program serving a shelf storage</p>		
<p>Electrical- and automation engineering</p>		
<p>The purpose of this thesis was to implement a program for a storage equipment consisting of Beckhoff components using the TwinCAT 3 software and located in the automation laboratory Satakunta University of Applied Sciences. The program had to transfer products from the storage to the conveyor and vice versa. The requirement was that the user only needs to provide the coordinates of the storage location and whether the product is retrieved from the storage to the conveyor or vice versa. Data for products in stock would be stored in a top-level program. The equipment already had a simple existing program which had to be studied first.</p> <p>The work introduces the various Beckhoff components used in the project, TwinCAT programming and PLCOpen Motion Control programming blocks. Beckhoff's TwinCAT 3 program was used to create the program. Parts of the program were reused. Data User Types (DUT) were used to reduce individual variables and a general-purpose function block that could be used to implement control of all the axes.</p> <p>The goals of the work were achieved. A program was created for the warehouse that transfers products as desired.</p>		
<p><u>Key words</u> Motion Control, Beckhoff, PLC</p>		

SISÄLLYS

1 JOHDANTO	5
2 KUVAUS ICIM LAITTEISTON TOIMINNASTA.....	5
2.1 Varasto	6
2.1.1 Akselit ja anturit	7
3 PROJEKTISSA KÄYTETTÄVÄT KOMPONENTIT	8
3.1 Beckhoffin TwinCAT	8
3.1.1 Beckhoff	8
3.1.2 TwinCAT	8
3.2 PLC	9
3.3 Hajautettu I/O ja EK1100.....	9
3.4 EL7342	11
3.5 EL1008.....	11
3.6 EL6900 TwinSAFE Logic	11
4 OLEMASSA OLEVA OHJELMA	12
4.1 Muuttujat	13
4.2 MAIN-lohko.....	13
4.3 Akselien ohjaus	14
4.3.1 PLCOpen Motion Control	14
4.3.2 MC-lohkot.....	14
4.4 Sekvenssiohjaus	15
4.5 Kotiinajosekvenssi	15
4.6 Automaattiajo ja tila	15
4.7 Käyttöliittymän sivut.....	16
5 OHJELMAN KIRJOITTAMINEN.....	18
5.1 Muuttujien uudelleen nimeäminen ja ryhmittely	18
5.2 Yleiskäyttöinen FB.....	19
5.3 Turvatoimet törmätessä.....	20
5.4 Automaattisekvenssi	20
5.5 Käyttöliittymä	22
5.6 Laite ongelmat.....	22
6 LOPPUTULOS JA JATKOTOIMET	23
LÄHTEET	
LIITTEET	

1 JOHDANTO

Satakunnan ammattikorkeakoulun (SAMK) automaatiolaboratorioon on hankittu linjasto, jossa kuljetin kierrättää tuotelaattoja eri toimintapisteille. Yhdessä näistä pisteistä on 40-paikkainen varasto, jossa on kolmella akselilla liikkuva kelkka, jonka on tarkoitus siirtää tuotteita kuljettimelta varastoon, ja toisin päin. Tälle laitteistolle on jo ennestään tehty ohjelma, jossa on perustoiminnot kaikille kolmelle kelkan liikeakselille.

Tämän opinnäytetyön tarkoituksena oli tutustua tähän laitteistoon ja olemassa olevaan ohjelmaan ja sen pohjalta jatkojalostaa ohjelmaa Beckhoffin TwinCAT -kehitysympäristössä siten, että edellä mainittua kelkkaa voidaan ohjata hakemaan tai viemään tuote automaattisesti tiettyihin käyttäjän antamissa koordinaattipisteissä sijaitsevaan varastopaikkaan. Tähän kuului esimerkiksi olemassa olevien akseli-ohjausten yhtenäistäminen ja käyttöliittymän sivujen muokkaaminen vastaamaan uusia toimintoja. Myös akselien parametointi piti säätää tarkempaa liikkumista varten ja ohjelmoida varotoimia erilaisia vahinkoja, kuten törmäysten varalle.

2 KUVAUS ICIM LAITTEISTON TOIMINNASTA

Laitteisto on aikanaan hankittu Festolta SAMK:n Rauman kampukselle ja siirretty Porin kampukselle n. 10 vuotta sitten. Ohjausjärjestelmä modernisoitiin 3 vuotta sitten. Laitteisto koostuu kuljettimesta, työstökonepisteestä, kokoonpanopisteestä, tarkastus- asemasta ja varastosta. Kokoonpano- ja työstökonepisteelle on myös sijoitettu robotit. Laitteiston on tarkoitus tehdä pohjakappale, johon sitten järjestelmästä luetun tilauksen mukaan sijoitetaan halutut komponentit. Kuljettimella kulkee alustoja, joiden päällä tuotteet kulkevat linjastolla. Jokaisessa alustassa on yksilöllinen ”kooditagi”,

josta jokaisella linjaston pisteellä oleva ”tagin”- lukija tunnistaa, mikä alusta pisteellä on ja välittää tiedon PLC:lle binäärilukuna. Raaka-aineiden, pohjakappaleen ja valmiin tuotteen kuljetusta ja varastointia varten ne sijoitetaan neliön muotoisten pallettien päälle, joissa on reunoissa reikiä. Sekä kuljettimen alustoissa, että varastopaikoissa on tapit, joihin nämä reiät sopivat näin pitäen palleitit paikoillaan. Raaka-aineet, komponentit ja valmiit tuotteet varastoidaan linjaston varrella olevaan varastoon

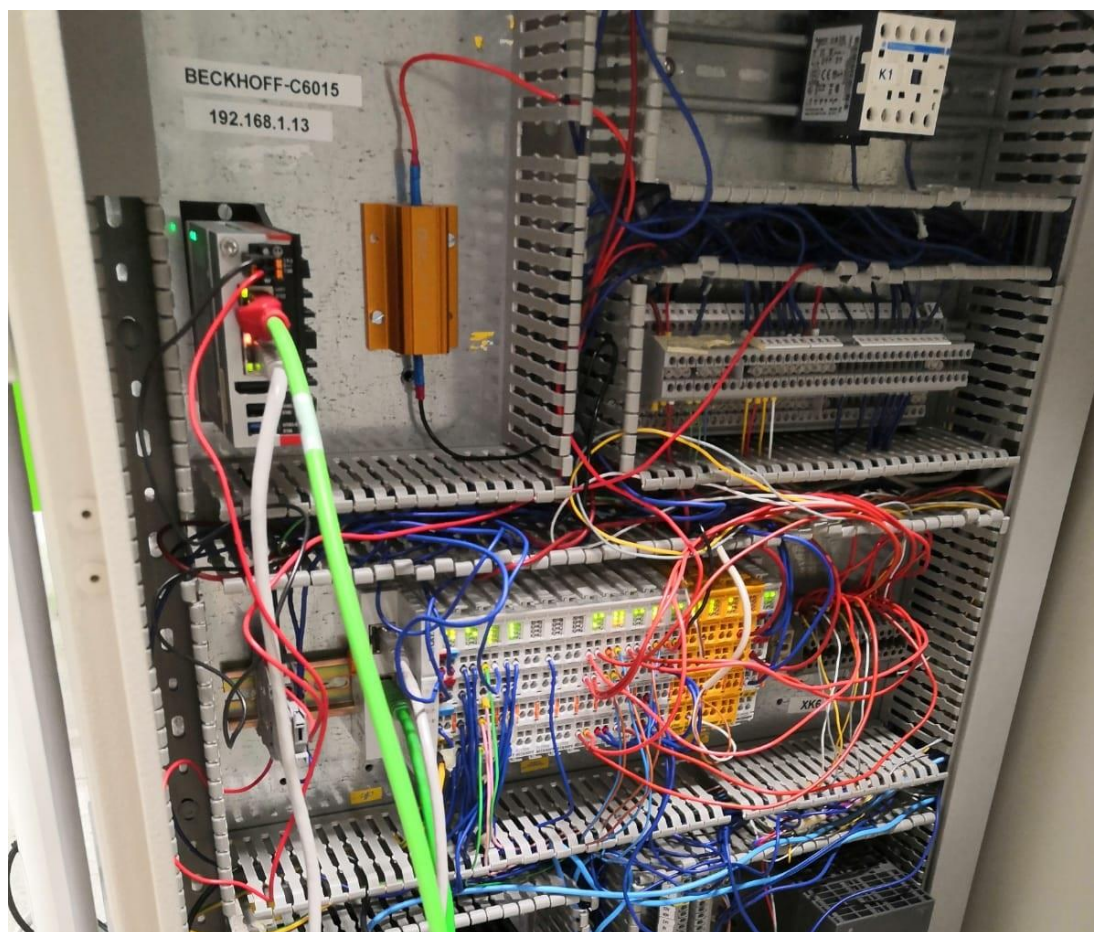
2.1 Varasto

Varasto muodostuu neljästäkymmenestä identtisestä varastopaikasta. Varastopaikkoja on 5 vaakasuoraa riviä ja 8 pystysuoraa riviä (Kuva 1).



Kuva 1. Kuva varastosta.

Kaikki ohjausjärjestelmän komponentit olivat valmiiksi asennettuna varaston vieressä sijaitsevaan kytkentäkaappiin (Kuva 2). Kaapissa on komponentteja ainoastaan varastoon. Muiden työstöpisteiden komponentit on hajautettu vastaavien pisteiden läheisyyteen. Tässä projektissa tutustuttiin ja käytettiin vain niitä komponentteja, jotka oli kytketty varasto-osuuteen.



Kuva 2. Kuva kytkentäkaapista.

2.1.1 Akselit ja anturit

Varaston robottikelkka pystyy liikkumaan kolmella akselilla, joiden suunnat määräytyvät oikean käden säännön mukaan. Akselit ovat Vaakasuora X-akselilla, pystysuora Z-akselilla ja eteen ja taakse liikkuva Y-akseli. Kaikkien näiden akselien molemmissa päissä on raja-anturit, jotka aktivoituvat akselin ajaessa niiden päälle ja tietokoneella tehty ohjelma pysäyttää ne, ettei kelkka törmää päätyreunoihin ja vahingoitu. Jokaisella akselilla on myös oma kalibrointianturinsa, jota käytetään akselien kotinajossa. Kaikki anturit on yhdistetty kytkentäkaapissa oleviin tulokortteihin, josta niiden tiedot siirtyvät PLC:n ohjelmaan.

Jokaiseen akseliin kuuluu oma moottorinsa, jonka avulla kelkkaa ohjataan positiiviseen ja negatiiviseen suuntaan akselilla muuttamalla moottorin pyörimissuuntaa. Jokaiseen moottoriin on liitetty myös pulssianturi, jonka tietojen mukaan lasketaan, missä kohtaa akselia kelkka sillä hetkellä on.

3 PROJEKTISSA KÄYTETTÄVÄT KOMPONENTIT

3.1 Beckhoffin TwinCAT

3.1.1 Beckhoff

Beckhoff on Saksasta lähtöisin oleva laitetoimittaja. Se tarjoaa avoimia automaatiojärjestelmiä, jotka perustuvat PC-pohjaiseen ohjausteknologiaan. Beckhoffin päätuotteita ovat liikkeenohjaustuotteet, automaatio-sovellus ohjelmistot, teollisuus-PC:t ja kenttäväyläkomponentit. Laitteet voivat toimia yksittäisenä komponenttina tai useammasta laitteesta voidaan yhteen sovittaa yhtenäinen ohjausjärjestelmä. Beckhoff on maailmanlaajuinen toimija ja sen sovelluksia käytetään älykkäästä rakennusautomaatiosta aina nopeisiin työstökeskuksiin asti. (*Company | Beckhoff Suomi, 2021.*)

3.1.2 TwinCAT

TwinCAT (The Windows Control and Automation Technology) on Beckhoffin kehittämä automaatio-ohjelmisto, joka muodostaa ohjelmistopuolella koko ohjausjärjestelmän ytimen. TwinCAT muuttaa lähes minkä tahansa PC järjestelmän reaaliaikaseksi ohjaimeksi ja voi korvata tavalliset PLC- ja NC-/CNC-ohjaimet ja käyttölaitteet. Yhdellä ohjelmalla voidaan hoitaa automaatiolaitteiden ohjelmointi, parametointi, diagnosointi ja määrittely. (*TwinCAT | Automation Software | Beckhoff Suomi, 2021.*)

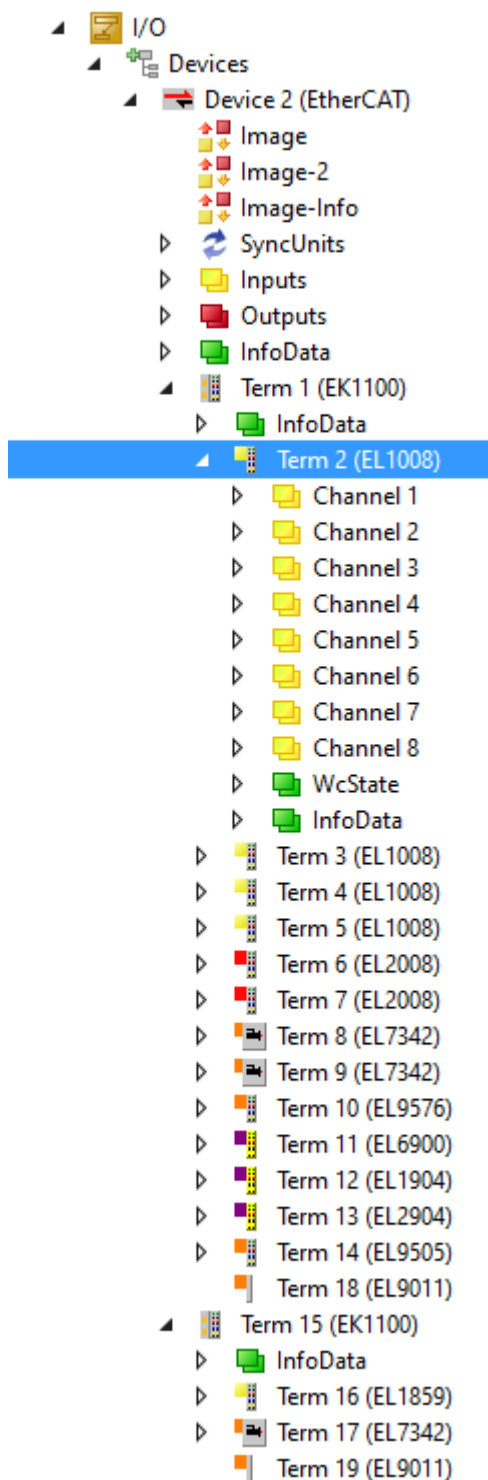
3.2 PLC

PLC eli ohjelmoitava logiikka on pieni tietokone ja se laite, johon tehty ohjelma ladataan ja minkä kautta sitä ohjataan. Logiikka suorittaa ohjelmaa ja ohjaa prosessin laitteita tarpeiden mukaan tuloilla ja lähdöillä. Tuloliitännöistä PLC saa tiedot, mitä prosessissa tapahtuu ja lähtöliitännöillä se ohjaa laitteita. Tulot ja lähdöt voivat olla joko modulaarisia eli PLC:hen liitetään erillisiä tulo/lähtömoduuli-laitteita tai integroituja eli tulot ja lähdöt ovat PLC:ssä itsessään. (*Ohjelmoitava Logiikka – Wikipedia, 2021.*)

Projektissa PLC:nä oli käytössä Beckhoffin C6015 teollisuus-PC. Se on pienin Beckhoffin tarjoama teollisuus-PC. Erittäin pienen koon, monipuolisten kiinnitysvaihtoehtojen ja tehokkaan suoritustehon ansiosta sitä voidaan käyttää monissa eri käyttökohteissa. Monenlaisten automaatiotehtävien lisäksi C6015 sopii pääasiassa rajapintalaitteeksi. Tämän perustana toimii TwinCAT:n ja EtherCAT:n täysi integrointi ja yhteensopivuus. (*C6015 / Fanless Ultra-Compact Industrial PC | Beckhoff Suomi, 2021.*)

3.3 Hajautettu I/O ja EK1100

Linjastolla I/O-laitteet on hajautettu. Tämä tarkoittaa sitä, että kaikki I/O-laitteet eivät ole samassa paikassa PC:n vieressä vaan ne ovat sijoitettu ohjaamansa työstöpisteen läheisyyteen ja yhdistetty toisiinsa kenttäväylällä. Nämä eri työstöpisteiden etä I/O yksiköt on ketjussa yhdistetty PC:seen, josta koko prosessia ja kaikkia etä I/O yksiköitä sitten ohjataan yhdessä. Linjastolla on ensin PC yhdistetty varaston etä I/O yksikköön, joka taas on yhdistetty kuljettimen vastaavaan jne. (Kuva 3).



Kuva 3. Hajautettu I/O

EK1100 on etä I/O yksikkö, joka yhdistää kenttäväylätason EtherCAT:n ja tulo/lähtökortit toisiinsa. Laite koostuu itse kytkimestä (EK1100) ja kuinka vaan monesta päätelaitteesta (EL7342 ja EL1008 yms.). Päätelaitteet saavat tarvitsemansa virran tiedon siirtoon suoraan päätelaitteelta. (*EK1100 / EtherCAT Coupler / Beckhoff Suomi, 2021.*)

3.4 EL7342

Beckhoffin EL7342 on EtherCAT-päätelaite, joka mahdollistaa 2 tasavirtamoottorin suoran ohjauksen. Automaatiolaite määrittelee nopeuden tai sijainnin 16-bittisen arvon avulla. Kun laitteistoon kytketään vielä inkrementtianturi, on yksinkertaisen servoakselin toteuttaminen mahdollista. Laitteessa on kaksi kanavaa, joiden tila ilmenee laitteen LED-valoista ja mahdollistavat nopean paikallisen diagnoosin. (*EL7342 / 2-Channel DC Motor Terminal, 48 V DC, 3.5 A / Beckhoff Suomi, 2021.*)

Projektissa käytetään kahta tällaista korttia, joihin DC-moottorit on yhdistetty. Näiden korttien kautta PLC ohjaa moottoreita. Toinen kortista hoitaa X- (pystysuora liike) ja Y-akselien (Kelkan edestakainen) ohjauksen. Toinen kortti ohjaa ainoastaan vaakasuoraa Z-akselia.

3.5 EL1008

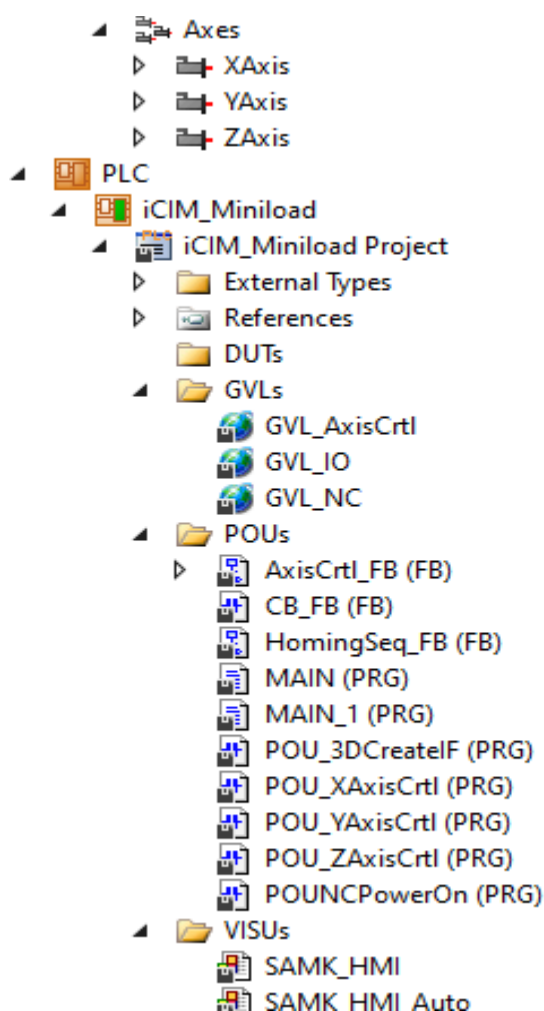
EL1008 on digitaalitulo-kortti, johon tulee 24V:n ohjaussignaalit prosessitasolta ja se siirtää ne ylemmän tason automaatioyksikköön. Jokaisessa kortissa on 8 kanavaa, joiden tilan näkee LED:eistä (*EL1008 / 8-Channel Digital Input Terminal 24 V DC, 3 Ms / Beckhoff Suomi, 2021.*). Näihin kortteihin tulee järjestelmässä olevien raja-anturien ja kalibrintianturien tilatiedot, joita käytetään ohjelmassa.

3.6 EL6900 TwinSAFE Logic

EL6900 TwinSAFE on nimeen omaan turvallisuuteen keskittyvien sovellusten ohjaukseen Boolean-tyyppisillä arvoilla. Tähän tarkoitukseen on tarjolla sertifioituja toimintalohkoja, joilla voidaan käsitellä turvainputteja ja turvaoutputtien tuottamisen. Lohkojen toiminnot voivat olla yksinkertaisista AND ja OR toiminnoista monimutkaisempiin toimintolohkoihin turvafunktion mykistämiseksi. Varmuuskopiointi ja palautus toiminto mahdollistaa myös yksinkertaisen tavan vaihtaa turvaohjainta huollon yhteydessä. EL6900 pystyy kommunikoimaan jopa 128 turvalaitteen kanssa. (*EL6900 / TwinSAFE Logic / Beckhoff Suomi, 2021.*)

4 OLEMASSA OLEVA OHJELMA

Laitteistoon on ennestään tehty ohjelma, johon on luotu kolme virtuaalista akselia, jotka on yhdistetty laitteiston fyysisiin akseleihin. Kaikki tarvittavien anturien ja muiden toimilaitteiden inputit ja outputit on myös yhdistettynä ohjelmaan ja niille on määritelty muuttujat. Muuttujat on jaoteltu kolmeen eriliseen muuttujalistaan. Akselien perustoiminnot ovat toteutettu useammalla ohjelmalohkolla ja niiden ohjausta varten on luotu kaksi käyttöliittymän sivua. Kaikki akselit, muuttujalistat ja ohjelmalohkot näkyvät projektipuusta. (Kuva 4.)



Kuva 4. Kuva ohjelmapuusta

4.1 Muuttujat

Ohjelmassa on valmiina kolme GVL:ää (Global Variable List) eli muuttujalistaa, joissa on globaaleja muuttujia. Globaaleja muuttujia voi käyttää missä vain osaa ohjelmaa. Tämän vastakohta on paikalliset muuttujat, joita voidaan tehdä ohjelmalohkon sisälle ja käyttää vain kyseessä olevassa lohossa.

GVL_AxisCtrl:ssä on kaikki akselien ohjaukseen tarvittavat muuttujat. Näihin kuuluu esimerkiksi käyttöliittymien tekstinsyöttökenttiä, merkkivaloja ja painonappeja esimerkiksi käsiajon positiivinen ja negatiivinen suunta. Koska jokaiselle akselilla on annettu omat erilliset muuttujat, muuttujalista on melko pitkä ja muuttujien nimissä ei muutu kuin akselin määrittävä X, Y tai Z kirjain nimen alussa. Tätä rakennetta tullaan muokkaamaan johdonmukaisemmaksi ja yksinkertaisemmaksi. GVL_IO-listaan on sijoitettu päätyraja- ja kalibrointianturien tilatiedot eli muuttujat menevät päälle, kun akseli liikkuu kyseessä olevan anturin päälle. GVL_NC sisältää jokaisen akselin AXIS_REF muuttujan. AXIS_REF datatyypin sisältää akselitiedot. Niitä käytetään liikkeenohjausohjelmointilohkoissa erittelemään, mitä akselia halutaan ohjata.

4.2 MAIN-lohko

MAIN-lohko on se lohko, jota aletaan suorittaa ohjelman käydessä. Sen takia kaikkia muita lohkoja täytyykin kutsua MAIN-lohkoissa, jotta ne suoritettaisiin. Lohkoa kutsuttaessa täytyy sen sisäiset input ja output muuttujat yhdistää globaaleihin muuttujiin. Esimerkiksi lohkon sisällä oleva muuttuja, joka aloittaa akselin liikkeen täytyy yhdistää muuttujaan, joka on käyttöliittymän sivun painonapissa. Jos muuttujaa tarvitaan vain lohkon sisällä, se voidaan jättää lohkon sisäiseksi. Jokaiselle akselille on tehty oma ohjauslohkonsa, jota kutsutaan ja jokaisen akselin tilatiedot luetaan erikseen MC_ReadStatus-funktiolla. Tämän takia MAIN-lohkoissa on paljon turhia rivejä muuttujille, joita ei edes käytetä missään.

4.3 Akselien ohjaus

Jokaista akselia ohjataan sen oman POU_AxisCtrl-lohkon kautta. Se on Program (PRG) tyyppinen lohko. Program on lohko, joka suoritettaessa palauttaa yhden tai useamman arvon. Suorittamisen jälkeen arvot säilytetään, kunnes ohjelma suoritetaan uudelleen. (*Object Program / Beckhoff Suomi, 2021.*) Program-lohkoista ei voi luoda instansseja eli sitä voidaan kutsua vain kerran ja sen sisäisillämuuttujilla voi olla vain yhdet arvot.

4.3.1 PLCOpen Motion Control

POU_AxisCtrl-lohkon sisällä käytetään PLCopen:in standardoiman Tc2_MC2-ohjelmointikirjaston lohkoja. PLCopen on riippumaton organisaatio, joka tarjoaa tehokkuutta teollisuusautomaatioon käyttäjien tarpeiden mukaan. Se keskittyy teknisiin määräyksiin IEC 61131-3 standardin ympärille. Tästä esimerkkinä on standardoidut kirjastot eri sovellusaloille. (*PLCopen - Wikipedia, 2021.*)

4.3.2 MC-lohkot

MC_Power lohkolla aktivoidaan akseli käyttöä varten. Akselin molemmat ajosuunnat voidaan aktivoida tai vain toinen suunta kerrallaan. Ilman tätä aktivointia akseli ei liiku mihinkään. Tätä voidaan käyttää ohjelmoitaessa turvatoimia esimerkiksi ohjelmoidulla aktivointi katkeamaan akselin ajaessa liikuttavan akselin päätyrajalle asti. MC_Power lohkot on ohjelmoitu omaan POUNCPowerOn lohkoonsa.

MC_Jog lohkolla akselia voidaan liikuttaa käsiajolla positiiviseen ja negatiiviseen suuntaan yleensä lohkon JogForward ja JogBackwards muuttujiin liitetyillä käyttöliittymän sivun painonapeilla. Lohkoon voidaan myös määrittää haluttu nopeus akselin liikkeelle.

MC_Stop lohkolla voidaan pysäyttää ja lukita akseli niin ettei se enää ota vastaan uusia liikekäskyjä. Akseli liikkuu vasta, kun stop-lohkon suorittaminen on lopetettu ja uusi

liikekäsky on annettu. Tämä on hyvä erikoistilanteissa tapahtuvissa pysäytyksissä, joissa halutaan estää akselin lisäliikkeet.

MC_Halt lohko toimii vähän kuin Stop-lohko, mutta toisin kuin Stop-lohko, joka pysäyttää koko akselin ja liikekäslyn, halt-lohkokalla voidaan vain ikään kuin tauottaa akselin liike. Jos akseli saa uuden liikekäslyn tai halt-käslyn poistuessa vanha käsky on vielä voimassa, lähtee akseli jatkamaan liikettä siitä mihin se jäi.

4.4 Sekvenssiohjaus

Sekvenssiohjauksessa ohjelmaa suoritetaan askeleittain, järjestyksessä ja seuraavaan askeleeseen siirrytään vasta kun tietty ehto on täyttynyt. Esimerkiksi ohjelma alkaa liikuttaa akselia johonkin asemaan eikä siirry seuraavaan liikkeeseen ennen kuin saa tiedon, että edellinen liike on saatu päätökseen.

4.5 Kotiinajosekvenssi

HomingSeq_FB lohkon on tehty ohjelma, joka suorittaa akselien kotiinajosekvenssin. Käyttöliittymän sivulta nappia painamalla ohjelma aktivoi jokaiselle akselille yksi kerrallaan POU_AxisCtrl-lohkon sisällä olevan MC_Home lohkon. Aktivoituessaan MC_Home alkaa liikuttaa akselia, kunnes se saapuu päätyrajan lähellä olevalle kalibrointianturille. Anturin saavutettuaan akseli kulkee toiseen suuntaan ennalta määrätyn etäisyyden ennalta tiedettyyn pisteeseen. Nyt ohjelma tietää, missä akseli on ja voi alkaa laskea akselin tulevia koordinaatteja. Seuraavan akselin kotinajo ei ala ennen, kuin edellinen on valmis ja akseli palauttaa Homed-tiedon. Kotinajo täytyy aina suorittaa ohjelman käynnistyksen yhteydessä, ennen kuin akseleita voidaan alkaa liikuttaa.

4.6 Automaattiajo ja tila

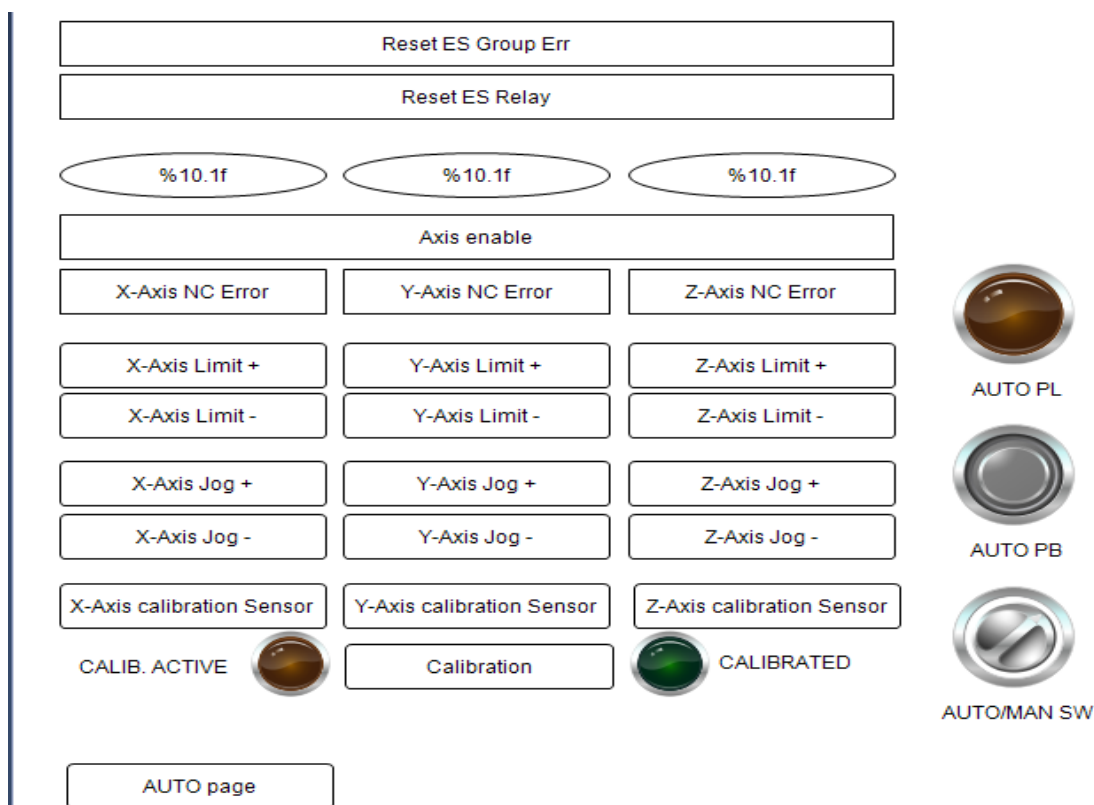
Akseleita voidaan liikuttaa joko käsiajolla tai automaattiajolla. Käsiajolla akseleita liikutetaan käyttöliittymän sivulta löytyvillä painonapeilla juuri niin kauan, kun nappeja painetaan. Käsiajosta huolehtii edellä mainittu Jog-lohko.

Automaattiajtoa varten ohjelma täytyy asettaa automaattitilaan. Automaattitilan hallintaa varten ohjelmassa on CB_FB lohko. Lohkoon on määritelty, että akselien tulee olla kalibroitu, käyttöliittymän sivulta säädin (Auto_SW) täytyy olla automaattiasennossa ja painonappia (Auto_PB) täytyy painaa, jotta automaattitila menee päälle. Automaattitilassa ohjelmalle voidaan antaa käyttöliittymän sivulta koordinaatti, ja ohjelma liikuttaa akselin automaattisesti annettuun paikkaan. Painonappi käynnistää AxisCtrl_FB-lohkokossa olevan sekvenssin. Sekvenssi käynnistää MC_AbsoluteMove-lohkon. AbsoluteMove-lohko saa käyttöliittymältä paikan ja suorittaa akselin liikkeen. Sekvenssi antaa käyttöliittymän sivulle tiedon, kun liike tulee valmiiksi tai se peruutetaan. Liike keskeytyy, jos annetaan halt- tai stop-käskey tai automaattitila menee pois päältä.

4.7 Käyttöliittymän sivut

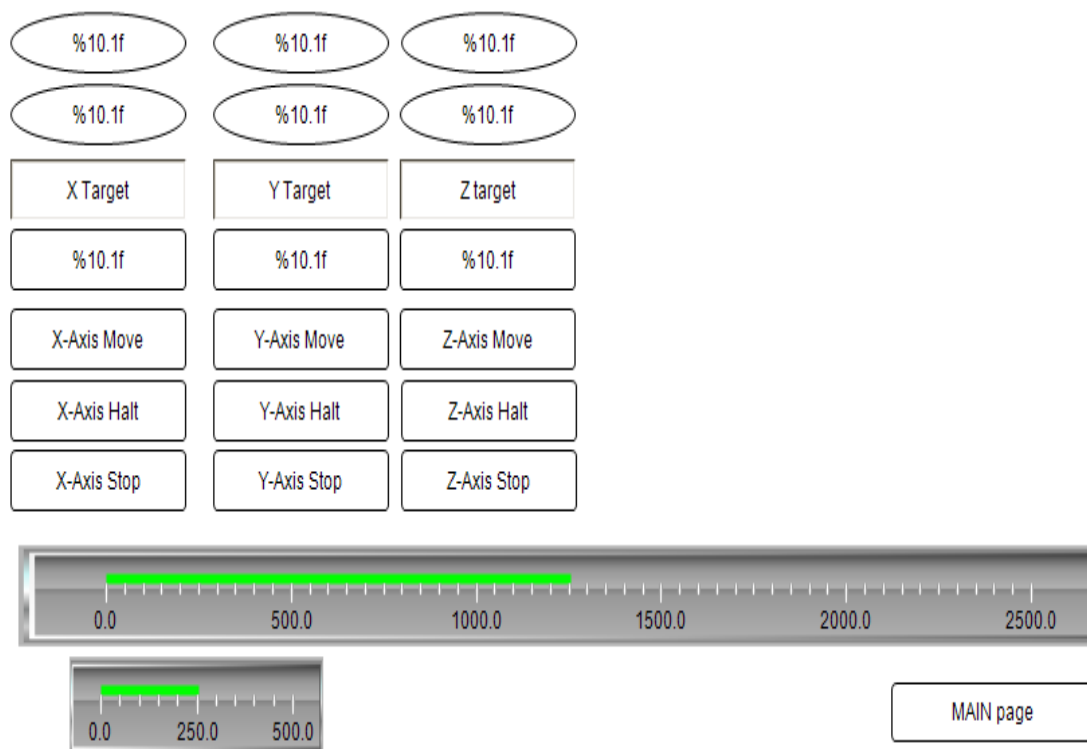
Käyttöliittymä on ohjelmiston tai laitteen osa, jonka avulla käyttäjä käyttää sitä. Esimerkiksi tietokoneohjelmaa käytettäessä käyttöliittymä on se, mitä käyttäjä näkee ruudullaan ja tapa, jolla hän käyttää kyseessä olevaa ohjelmaa. (*Käyttöliittymä – Wikipedia, 2021.*)

Tässä ohjelmassa on kaksi erillistä käyttöliittymän sivua. SAMK_HMI-käyttöliittymä (Kuva 5.) sisältää ohjelman perustoiminnot ja tilatiedot. Tästä käyttöliittymän sivusta nähdään esimerkiksi ovatko akselien päätyrajakytkimet tai kalibrointianturi aktivoituneena. Siihen tulee myös ilmoitukset akselien virhetiloista. Se ei osaa kertoa, millainen virhe on tapahtunut, ainoastaan että virhe on tullut. Tulevia virheitä ei pystytä myöskään kuittaamaan suoraan käyttöliittymän sivulta vaan se täytyy tehdä liikeohjaimesta. Käyttöliittymän sivun kautta voidaan myös aktivoida jokainen akseli yhdellä painonapilla, ohjata jokaista erikseen käsiajolla ja nähdä niiden sen hetkinen paikkatieto. Tätä kautta aloitetaan myös kalibrointisekvenssi, jonka aktiivisuutta indikoi merkkilamppu. Sekvenssin valmistumiselle on myös oma merkkilamppunsa. Automaatti tila saadaan päälle ehtojen täytyttyessä painonapilla (Auto PB).



Kuva 5. SAMK_HMI

Automaatiotilan ohjaukseen on oma käyttöliittymän sivu SAMK_HMI_Auto (Kuva 6.). Tästä käyttöliittymän sivusta voidaan ohjata automaatiosekvenssiä. Jokaiselle akselille on oma rivinsä, jolla sitä voidaan ohjata. Kentät näyttävät akselin sen hetkisen paikan, nopeuden ja Target-pisteen. Target-piste on se paikka, johon akseli pyrkii siirtymään, kun automaatiosekvenssi aloitetaan move-painonapilla. Target voidaan antaa käyttöliittymän sivulla target-kenttään ja alempi kenttä näyttää sen sitten takaisin käyttäjälle. Target-kenttään on asetettu rajat, jotta käyttäjä ei voi antaa sellaista arvoa, jota akseli ei voi saavuttaa. Halt- ja stop-painonapit vastaavasti suorittavat akseliin liitetyt halt- ja stop-lohkot. Ohjelman kahden käyttöliittymän sivun välillä voidaan liikkua helposti sivujen alareunoissa olevien nappien avulla.



Kuva 6. SAMK_HMI_Auto

5 OHJELMAN KIRJOITTAMINEN

Valmiissa ohjelmassa täytyy olla tietyt toiminallisuudet. Käyttäjän täytyy pystyä antamaan käyttöliittymän sivulla varastopaikan koordinaatit ja päättämään suoritetaanko tuotteen vienti linjastolta siihen varastopaikkaan vai haetaanko varastopaikasta tuote linjastolle.

5.1 Muuttujien uudelleen nimeäminen ja ryhmittely

Työ aloitettiin muokkaamalla jo olemassa olevien muuttujien nimiä ja ryhmittelemällä niitä tietotyyppeihin. Ohjelmaan voidaan tehdä muuttujille DUT (Data User Type) eli käyttäjän oma tietotyyppi. Aikaisemmin ohjelmassa jokainen muuttuja oli luotu loh-

koihin erikseen ja erikseen yhdistetty main-lohkossa. Tietotyyppiin voidaan lisätä tietyssä lohkoissa tarvittavat muuttujat ja yhdistää tämä tietotyyppi lohkoon, jolloin kaikki tarvittavat muuttujat saadaan käyttöön.

Tehdään jokaiselle MC-lohkolle oma tietotyyppi. Esimerkiksi käsiajoa ohjaavaa MC_Jog lohkoa varten tietotyyppiin tulee akselitieto (refAxis), eteenpäin-muuttuja (jogFwd) ja taaksepäin-muuttuja (jogBwk). Nämä tietotyypit taas vuorostaan lisätään typeMcCmd-tietotyyppiin, johon tulee kaikkien MC-lohkojen tietotyypit ja se lisätään GVL_Axes-muuttujalistaan. Koska kaikki MC-lohkot palauttavat samoja tilatietoja, voidaan näitä statusmuuttujia varten tehdä myös oma tietotyyppi typeMCStatus, joka voidaan lisätä MC-lohkojen tietotyyppiin. Näin saadaan poistettua GVL_AxisCtrl-muuttujalista ja vähennettyä muuttujia kolmannekseen, kun jokaisella akselilla ei tarvitse olla omaa muuttujaa. Tietotyyppien nimien alkuun lisätään sana type eli tietotyyppien nimet ovat muotoa type”MC-lohkon nimi”.

5.2 Yleiskäyttöinen FB

Aikaisemmin kaikille akseleille oli oma erillinen program tyyppinen ohjauslohkonsa. Nyt ohjelmaan tehtiin yksi yleiskäyttöinen function block (FB). Function block eroaa edellä mainitusta program lohkoista siinä, että se ei ole kertakäyttöinen vaan sitä voidaan kutsua ohjelmassa useammin kuin kerran. Kutsuttaessa function blockia ohjelma luo siitä instanssin. Instanssi on kopio function blockista, johon tallentuu lohkon palauttavat tiedot. Eli kun ohjelmassa lohkoa kutsutaan kolme kertaa, Main-lohkoon tulee kolme instanssia lohkoista, yksi jokaiselle akselille.

Tätä varten luotiin fbMCAxis-lohko. Tähän lohkoon lisättiin kaikki aiemmassa AxisCtrl programmeissa olleet MC-funktiot eli MC_Jog, MC_Home, MC_MoveAbsolute, MC_Stop ja MC_Halt. Lohkoon lisättiin myös aiemmin omassa lohkoissaan ollut MC_Power ja Main lohkoissa jokaiselle akselille kutsuttu MC_ReadStatus. Virheiden hallintaa varten lisättiin MC_Reset, joka resatoi liikeakselin ja poistaen sen hetkiset virhetilat. Kaikista MC-lohkoista luodaan instanssi fbMCAxis-lohkoon. Lohkoja varten täytyy luoda sisäisiämuuttujia. typeMcCmd-tietotyypin avulla saamme

suurimman osan muuttujista kerralla. Sen lisäksi tarvitaan akselitieto `AXIS_REF` ja antureihin ja käyttöliittymän sivuille yhdistettävät muuttujat.

5.3 Turvatoimet törmätessä

Ohjelmaan tarvittiin jonkinlainen varotoimi pysäyttämään akselit, jos ne törmäisivät johonkin ajon aikana. Yksi vaihtoehto olisi etsiä ohjaimesta tilatieto, joka kertoisi moottorin käyttämän virran arvon. Kun akseli törmäisi johonkin, se ottaisi enemmän virtaa yrittäessään jatkaa liikettä. Tähän voitaisiin ohjelmoida raja-arvo, jonka ylittäessään moottorit sammuisivat.

Tiedustelin Beckhoffilta saisiko tällaista tietoa mistään ulos. Beckhoffilta kerrottiin, että tieto löytyy, mutta kehoitettiin sen sijaan käyttämään jättämän valvontaa. Jättämän valvonnassa ohjelma vertaa liikkeenohjaimen laskemaa paikkaa akselin todelliseen paikkaan ja mikäli asetettu raja-arvo ylitetään, akseli pysäytetään. Jättämän valvontaa varten ohjelmalle täytyy antaa tarkat parametrit, siitä kuinka paljon akseli liikkuu, kun moottoriin kytketty pulssianturi pyörii yhden pulssin verran.

Jokaista akselia tarvitsi ajaa päästä päähän ja katsoa mikä oli suurin jättämän arvo. Tämän jälkeen akselien jättämän valvontaan asetettiin tätä hieman suurempi raja-arvo. Näin akseli ei ehdi ajaa törmättyä kohdetta päin kuin pienen hetken, välttäen suuremmat vahingot.

5.4 Automaattisekvenssi

Nyt voitiin siirtyä itse varastosekvenssin ohjelmointiin. Aluksi muokataan `Axisc-rtl_FB`-lohkoa. Muutetaan sen nimeksi `fbMoveAxis` ja muokataan se seuraamaan paremmin, onko akselin liike valmis vai onko liike pausella (`Halt`-lohko) tai `aborted` (`Stop`-lohko). Lisätään myös kuittausmuuttujat, joita käytetään varmistamaan, että ohjelman eri sekvenssit pysyvät samassa tahdissa eivätkä siirry seuraavaan steppiin ennen kuin kaikki ovat valmiita.

Luodaan fbAutoCtrl-lohko, johon tulee ohjelman pääsekvenssi. Tämä pääsekvenssi ohjaa pienempää fbMoveAxis-lohkossa olevaa sekvenssiä. Sekvenssi käynnistyy, kun automaattitila on päällä ja mikään akseleista ei ole aborted-tilassa ja jää odottamaan tietoa siitä suoritetaanko tuotteen hakeminen varastosta vai tuotteen vieminen varastoon kuljettimelta. Käyttäjä antaa tämän tiedon käyttöliittymän sivujen painonapeilla. Tiedon saatuaan ohjelma alkaa suorittaa valittua puolta sekvenssistä askel kerrallaan.

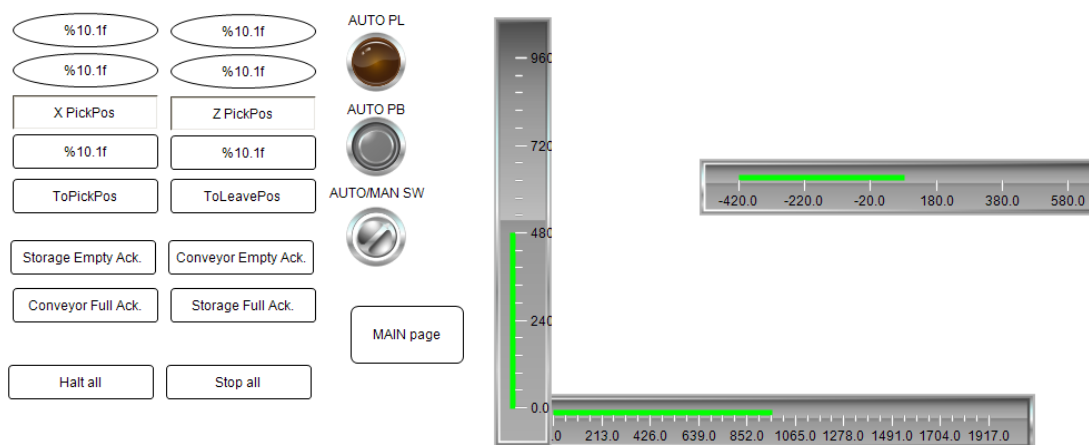
Varastosta hakemisessa ohjelma lukee käyttäjän käyttöliittymän sivulle antamat varastopaikan X- ja Z-koordinaatit ja siirtää akselit yksi kerrallaan niihin. Pääsekvenssi (fbAutoCtrl) antaa käskyn pienemmälle sekvenssille (fbMoveAxis) käynnistää vuorossa olevan akselin MC_MoveAbsolute-lohkon, joka liikuttaa akselin haluttuun paikkaan. Varastopaikalle päästyään Y-akseli kurottuu varastopaikalla olevan laatan alle ja kelkkaa nostetaan sen verran, että laattaa nousee kiinnitystapeilta, jotka pitävät sitä paikallaan. Y-akselin palattua takaisin keskiasentoon kelkka kuljetetaan kuljettimen eteen. Tämän laatan jättöpaikan koordinaatit ovat aina samat ja ne ovat valmiina ohjelmassa. Kelkka ajetaan kuitenkin hieman korkeammalle kuin jättöpaikka on, että laatta saadaan laskettua tapeille. Lopuksi Y-akseli ajetaan taas keskiasentoon ja ohjelma jää odottamaan seuraavia koordinaatteja ja haku- tai vientikäskyä. Varastoon vieminen on sama prosessi, mutta käänteinen. Kelkka kuljetetaan kuljettimen vakio koordinaatteihin ja nostaa laatan kuljettimelta, ohjelma lukee halutun varastopaikan ja liikkuu sen eteen hieman paikkaa ylemmäs, laatta lasketaan varastopaikkaan ja kelkka palautuu keskiasentoon. Sekvenssi on melko yksinkertaisesti toteutettu ja liikuttaa vain yhtä akselia kerrallaan. Tämän takia ohjelma on melko pitkä ja vaikea selkoinen myöhempää tarkkailua ajatellen.

Ohjelmaan lisättiin vielä toiminto, jossa ohjelma tarkistaa jättöpaikan tilan ennen kuin yrittää täyttää sitä eli jos varastopaikassa tai kuljettimella on jo laatta, kun kelkka aikoo jättää siihen laattaa, ohjelma pysähtyy. Näin vältytään vahingoilta, joita voisi sattua, jos varastopaikkaan tungettaisiin kahta laattaa. Vastaavasti jos noutopaikka on tyhjä, ohjelma ei turhaan jatku. Tarkastuksessa hyödynnetään kelkassa kiinni olevaa anturia, joka huomaa laatoissa olevat kiiltävät tarrat. Ohjelman saa taas takaisin päälle poistamalla ylimääräisen laatan ja kuittaamalla tilanteen käyttöliittymästä. Pelkkä kuittaus ei riitä, koska ohjelma suorittaa tarkastuksen uudelleen kuittauksen jälkeen ennen liikumista.

5.5 Käyttöliittymä

Kelkkaa ohjaavan sekvenssin valmistuttua täytyi käyttöliittymän sivuja muokata vastaamaan ohjelman uusia toimintoja. SAMK_HMI pysyi päätoiminnoiltaan samanlaisena. Ainoa lisäys oli akselien virheilmoituksia varten olevat kentät, jotka muuttuvat punaisiksi virheen sattuessa. Kenttiin lisättiin toiminto, jossa painettaessa kenttä aktivoi MC_Reset-lohkon resetoiden lohkon ja samalla poistaen virheen. Käyttöliittymän sivuihin tehtiin myös pieniä muutoksia painikkeiden teksteihin ja aseteluihin.

Automaattitilan ohjaukseen käytettyyn SAMK_HMI_Auto (Kuva 7.) ruutuun tehtiin enemmän muutoksia. Y-akselin koordinaattien syöttökenttä poistettiin, koska se liikkuu aina ennalta määrättyihin koordinaatteihin, joten käyttäjän ei tarvitse muuttaa niitä. Akselien erilliset move-painikkeet korvattiin sekvenssin aloittavilla ToPickPos ja ToLeavePos-painikkeilla. Stop ja halt komennoille tehtiin myös molemmille painikke, joka pysäyttää kaikki akselit kerralla. Uutena lisäyksenä tulivat kuittaukset tilanteisiin, joissa halutussa paikassa oli jo laatta tai hakupaikassa ei olekaan laattaa.



Kuva 7. SAMK_Auto-käyttöliittymä sivu muokattuna

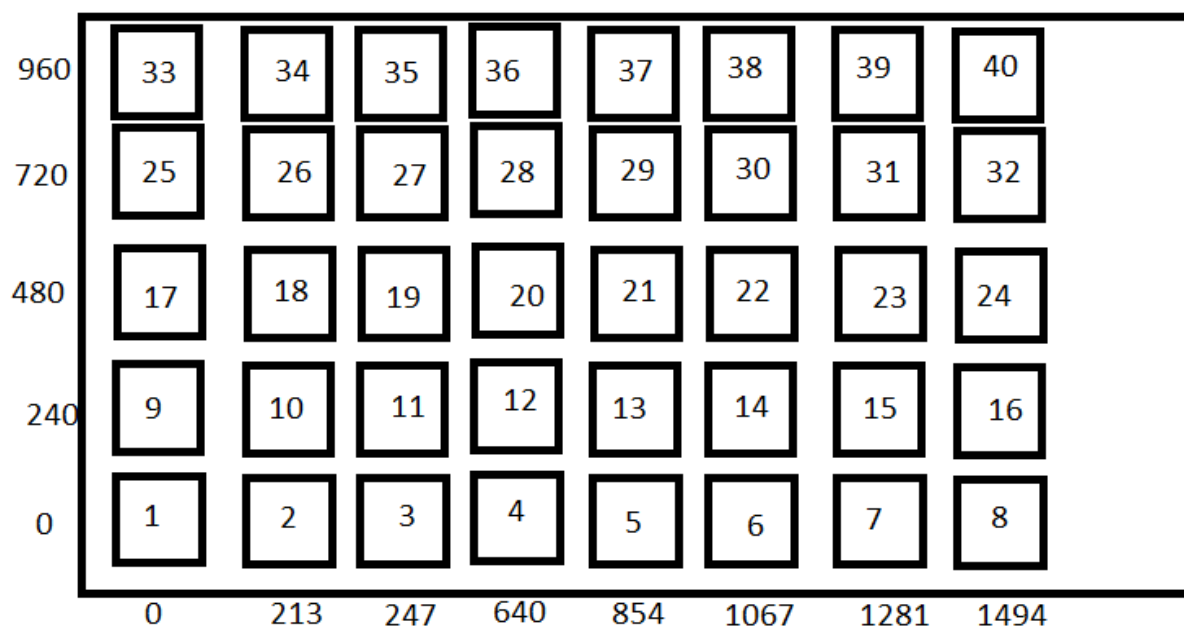
5.6 Laite ongelmat

Ohjelman testien loppupuolella ilmeni jonkinlainen vika laitteistossa. Pystysuuntainen Z-akseli liikkui alaspäin normaalisti, mutta ei enää jaksanut kulkea ylöspäin kuin pienen matkan kerrallaan. Joskus akseli jopa lähti putoamaan alaspäin. Ilmeni, että moot-

tori antoi jostain syystä periksi heti kun akselille asetettiin enemmän painoa. Vika yritettiin korjata asentamalla moottoriin ja liikeohjaimen yhtymäkohtaan kiila, joka antaisi sille enemmän pitoa. Ongelma oli kuitenkin vastakappaleessa, johon ei saatu tehtyä sopivaa muokkausta. Vastakappale puhdistettiin ja akseli alkoi taas toimia paremmin. Lisäpainoa se ei vielääkään kestänyt, mutta käyttötarkoitukseensa akselin pito oli riittävä.

6 LOPPUTULOS JA JATKOTOIMET

Ohjelma saatiin valmiiksi ja se täyttää annetut vaatimukset. Se siirtää laattoja halutulla tavalla varaston ja kuljettimen välillä. Käyttäjän ei tarvitse, kuin antaa koordinaatit ja vienti suunta. Tällä hetkellä käyttäjän täytyy antaa ohjelmalle varastopaikan tarkat koordinaatit (Kuva 8). Jos käyttäjä antaa vääränlaiset koordinaatit voidaan kelkka ajaa varastoa päin. Tähän on suunniteltuna jatkotyötä. Ohjelmaa jalostettaisiin niin, että tarkkojen koordinaattien sijaan käyttäjä antaa vain varastopaikan numeron ja ohjelma etsii tietokannasta varastopaikan koordinaatit.



Kuva 8. Varastopaikojen koordinaatit

Kokonaisuudessaan koin työn mielisiksi projektiksi. Aihe oli jo hieman tuttu aikaisemmista opinnoistani ja työ virkistä ja syvensi jo ennestään oppimiani asioita entisestään. Haasteita kuitenkin tuotti laiteongelmat ja tauko siitä, kun oli viimeksi ohjelmoinut TwinCAT ympäristössä. Vaikeuksia oli myös itse kirjoittamisessa, joka oli ajoittain takkuilevaa. Opin kuitenkin paljon työn aikana esimerkiksi ohjelmoinnista ja liikkeenohjauksesta yleensä ja toivon, että pääsen tekemään sitä jatkossakin.

LÄHTEET

- C6015 | Fanless ultra-compact Industrial PC | Beckhoff Suomi.* (2021). Retrieved April 12, 2021, from <https://www.beckhoff.com/fi-fi/products/ipc/pcs/c60xx-ultra-compact-industrial-pcs/c6015.html>
- Company | Beckhoff Suomi.* (2021). Retrieved April 12, 2021, from <https://www.beckhoff.com/fi-fi/company/>
- EK1100 | EtherCAT Coupler | Beckhoff Suomi.* (2021). Retrieved April 21, 2021, from <https://www.beckhoff.com/fi-fi/products/i-o/ethercat-terminals/ek1xxx-bk1xx0-ethercat-coupler/ek1100.html>
- EL1008 | 8-channel digital input terminal 24 V DC, 3 ms | Beckhoff Suomi.* (2021). Retrieved April 21, 2021, from <https://www.beckhoff.com/fi-fi/products/i-o/ethercat-terminals/el1xxx-digital-input/el1008.html>
- EL6900 | TwinSAFE Logic | Beckhoff Suomi.* (2021). Retrieved April 23, 2021, from <https://www.beckhoff.com/fi-fi/products/automation/twinsafe/twinsafe-hardware/el6900.html>
- EL7342 | 2-channel DC motor terminal, 48 V DC, 3.5 A | Beckhoff Suomi.* (2021). Retrieved April 12, 2021, from <https://www.beckhoff.com/fi-fi/products/i-o/ethercat-terminals/el7xxx-compact-drive-technology/el7342.html>
- Käyttöliittymä – Wikipedia.* (2021). Retrieved June 18, 2021, from <https://fi.wikipedia.org/wiki/Käyttöliittymä>
- Object Program | Beckhoff Suomi.* (2021). https://infosys.beckhoff.com/english.php?content=../content/1033/tc3_plc_intro/2530274187.html&id=
- Ohjelmoitava logiikka – Wikipedia.* (2021). Retrieved April 21, 2021, from https://fi.wikipedia.org/wiki/Ohjelmoitava_logiikka
- PLCopen - Wikipedia.* (2021). Retrieved June 18, 2021, from <https://en.wikipedia.org/wiki/PLCopen>
- TwinCAT | Automation software | Beckhoff Suomi.* (2021). Retrieved April 12, 2021, from <https://www.beckhoff.com/fi-fi/products/automation/twincat/>