

Sahan tuorelajittelun kuljetin ohjaus

LAB-ammattikorkeakoulu

Insinööri (AMK), Konetekniikka

2021

Jerri Sipilä

Tiivistelmä

Tekijä(t) Sipilä, Jerri	Julkaisun laji Opinnäytetyö, AMK	Valmistumisaika 2021
	Sivumäärä 35	
Työn nimi Sahan tuorelajittelun kuljetin ohjaus		
Tutkinto Insinööri (AMK)		
Toimeksiantajan nimi, titteli ja organisaatio Raimo Seppänen, Suunnitteluinsinööri, Pronor Control Oy.		
Tiivistelmä <p>Työn tarkoituksena oli suunnitella ja toteuttaa sahalaitoksen kuljetinohjauksille malliohjelmarunko. Malliohjelmarunkoa sovellettiin sahalaitoksen modernisoinnissa. Malliohjelmarungon tarkoituksena on nopeuttaa ohjelmoijan tekemää työtä sekä vakioida kuljetinohjauksien rakennetta.</p> <p>Työ suunniteltiin ja toteutettiin Siemensin TIA-portal ohjelmointiympäristöön. Tietoa aiheesta kerättiin alan kirjallisuudesta sekä aikaisemmista edellisen sukupolven ohjelmointiympäristöön toteutetuista projekteista.</p> <p>Lopputuloksena oli helposti muokattavissa oleva vakiomalliohjelma, jota saatiin sovellettua onnistuneesti sahalaitoksen modernisoinnin kuljetinohjauksissa. Kuljetinohjauksien rakenne saatiin pysymään yhtenäisenä vakiorungon ansiosta.</p>		
Asiasanat PLC, Logiikka, Ohjelmointi, Kuljetinohjaus, Siemens, TIA-portal		

Abstract

Author(s) Sipilä, Jerri	Type of Publication Thesis, UAS	Published 2021
	Number of Pages 35	
Title of Publication Sawmill green sorting plc-controls		
Name of Degree Engineer (UAS)		
Name, title and organization of the client Raimo Seppänen, Design Engineer, Pronor Control Oy.		
Abstract <p>The purpose of the work was to design and implement a model program framework for sawmill conveyor controls. The model program framework was applied in the modernization of the sawmill. The purpose of the model program framework was to speed up the work done by the programmer as well as to standardize the structure of the conveyor controls.</p> <p>The work was designed and implemented for Siemens TIA-Portal programming environment. Subject related information was collected from the literature as well as from projects implemented with the previous generation programming environment.</p> <p>The result was an easily customizable standard model program that could be successfully applied to conveyor controls for sawmill modernization. The structure of the conveyor controls was kept uniform thanks to the standard frame.</p>		
Keywords plc, logic, programming, conveyor controls, siemens, tia portal		

Sisällys

1	Johdanto.....	1
1.1	Työn tausta.....	1
1.2	Pronor control.....	1
1.3	Tavoitteet ja rajaukset.....	1
1.4	Käytetyt menetelmät.....	2
2	Sahausprosessi.....	3
2.1	Tukkilajittelu.....	3
2.2	Kuorinta ja sahaus.....	3
2.3	Tuorelajittelu.....	5
2.4	Kuivaus.....	6
2.5	Kuivalajittelu.....	6
3	Automaatiosuunnittelu.....	7
3.1	PLC-ohjelmointi.....	7
3.1.1	IEC 61131 Standardi.....	7
3.2	Logiikat yleisesti.....	8
3.3	Siemens logiikat.....	8
3.3.1	Simatic S7.....	9
3.3.2	TIA portal.....	9
3.3.3	Turvalogiikka.....	9
3.3.4	Väyläteknikka.....	10
4	Malliohjelman rakenne.....	12
4.1	Malliohjelmanrunko.....	12
4.2	Toiminto FC.....	12
4.3	Multi-instanssi.....	14
4.4	Sovelluskohteet.....	15
5	Malliohjelman soveltaminen käytännössä.....	17
5.1	Asiakasyritys.....	17
5.2	Lähtötiedot.....	18
5.3	Hardware.....	19
5.4	Lohkojako.....	20
5.5	Ohjelman rakenne.....	21
6	Käyttöliittymä.....	27
6.1	HMI.....	27
6.2	WinCC.....	28

7	Testaus ja käyttöönotto.....	29
7.1	Ohjelman testaus.....	29
7.2	Käyttöliittymän testaus.....	29
7.3	Virtuaalinen käyttöönotto	29
7.4	Käyttöönotto	30
8	Yhteenveto	31
	Lähteet	32

Liitteet

Liite 1. Käytönnoton tarkistuslista

Liite 2. Käyttöliittymän ja logiikan välisen rajapinnan tarkistuslista

1 Johdanto

1.1 Työn tausta

Ohjelmoitavat logiikat ovat kehittyneet aikojen kuluessa. Edelleen teollisuudessa käytetään laajasti vanhempiin ohjelmointiympäristöihin tehtyjä sovelluksiakin. Sovelluksia ylläpidetään, kunnes tulee tarve uudistaa niitä ohjausominaisuuksien puutteen vuoksi tai tehtaan modernisoinnin yhteydessä. Myös tuotantoprosessin kapasiteetin nosto tai laadun parantaminen voi johtaa ohjausjärjestelmän muutokseen. Uudet projektit tehdään pääsääntöisesti nykyaikaisilla ohjelmointityökaluilla.

Opinnäytetyössä on tarkoitus suunnitella sahalaitoksen tuorelajittelun kuljetinohjaukseen vakio-ohjelmarakenne. Vanhemmissa ohjelmointiympäristöissä, kuten Siemensin Step7, on laajasti erilaisia pohjaprojekteja käytettävissä uuden ohjelmarakenteen malliksi. Ohjelmia pystyy myös siirtämään eli konvertoimaan vanhemmasta toteutusympäristöstä uudempaan. Konvertointi vanhasta ympäristöstä ei kuitenkaan ole useinkaan optimaalisin ratkaisu. Uudemmassa ohjelmointiympäristössä toteutetussa ohjelmassa voidaan hyödyntää ohjelmiston tarjoamia uusia ominaisuuksia. Toimeksiantajayrityksessä on tehty projekteja myös uudemmalla ohjelmointiympäristöllä. Kyseisiä projekteja pystyy hyödyntämään mallina uuden projektin toteutuksessa, mutta varsinaista malliohjelmarunkoa soveltamiseen ei ole. Opinnäytetyössä suunniteltavan malliohjelmarungon tarkoituksena onkin nopeuttaa ohjelmointityötä projekteissa, joissa käytetään uudempaa ohjelmointiympäristöä.

1.2 Pronor control

Opinnäytetyön toimeksiantaja Pronor Control on perustettu 2005 ja toimipiste sijaitsee Heinolassa. Pronor Control keskittyy teollisuusautomaatioon. Suurin yritystä työllistävä teollisuuden ala on sahateollisuus. Yritys tekee sahateollisuuteen erilaisia ohjausjärjestelmiä, esim. kuljetin- ja laiteohjauksia. Ohjaukset ovat logiikkapohjaisia eli logiikkaan määriteltyjen ehtojen perusteella ohjataan kuljettimia ja muita tuotantolinjan laitteita. Ohjauksissa hyödynnetään asiakkaan tuotannonohjausjärjestelmiä, joiden pohjalta logiikka määrittelee tuotantolinjan ohjaukset. Valvomoratkaisut ja käyttöliittymät ovat myös olennainen osa ohjauksia. Projektit ovat asiakaslähtöisiä ja tehdään asiakkaan tarpeiden mukaisesti. Projektit kohdistuvat tyypillisesti joko vanhan modernisointiin tai uusiin investointeihin.

1.3 Tavoitteet ja rajaukset

Opinnäytetyön tavoitteena on toteuttaa malliohjelmarunko, josta voidaan pienillä muokkauksilla toteuttaa projektikohtainen sovellusohjelma moninaisten asiakkaiden tarpeisiin.

Malliohjelmarunko tehdään Siemens TIA portal-ohjelmistolla. Työssä suunnitellaan malliohjelman rakenne ja listataan sen tuomia hyötyjä projektikohtaiseen suunnittelutyöhön. Ohjelmarunkoa sovelletaan myös asiakasprojektin kuljetinohjauksiin. Opinnäytetyön eri vaiheista selviää automaatio suunnittelijan laaja työnkuva. Työ sisältää myös käyttöönottoa helpottavan tarkistuslistan tekemisen. Se jää toimeksiantajalle myös dokumentiksi laatujärjestelmää varten. Ohjelmarunko ja asiakasprojektiin tehtävä ohjelma testataan ennen käyttöönottoa, jotta mahdolliset muokkaukset käyttöönotossa voidaan minimoida. Opinnäytetyön tuloksena on malliohjelmarunko Pronor Controlille, jota voidaan soveltaa ja hyödyntää projekti-toimituksissa. Työn tarkoitus on myös hyödyntää opintojen aikana omaksutut ohjaustekniikan perustiedot ja kehittää niitä työelämän tarpeisiin.

Tämän työn tarkoituksena ei ole käydä läpi TIA portalin käyttöä, vaan kuvata muokattavan malliohjelmarungon toteutus ja siitä saatavat hyödyt. Työstä on rajattu käyttöliittymän suunnittelu pois. Työssä käsitellään kuitenkin logiikan ja käyttöliittymän rajapintaa. Opinnäytetyön laajuuden ja aikataulullisista syistä johtuen ei tähän opinnäytetyöhön ole mahdollista sisällyttää kyseisen projektin käyttöönottoa. Tuotantolinjan käyttöönotto tapahtuu loppuasiakkaan luona sahalaitoksessa Ranskassa. Käyttöönotossa pääsee fyysisesti havainnollistamaan joko mekaaniset tai ohjelmalliset haasteet ja sitä kautta todentamaan ohjelman toimivuus käytännön olosuhteissa. Käyttöönotosta kerrotaan kuitenkin yleisesti muutamia pelisääntöjä, miten siellä tulisi toimia.

1.4 Käytetyt menetelmät

Opinnäytetyössä on perehdytty vanhalla ohjelmointiympäristöllä toteutettuihin projekteihin ja niiden rakenteeseen. Työssä on tutustuttu myös alan toimijoihin sekä Siemensin ohjelmointiympäristöihin. Työn tukena on käytetty logiikkaohjelmointiin liittyviä julkaisuja ja alaan liittyvää kirjallisuutta.

2 Sahausprosessi

2.1 Tukkilajittelu

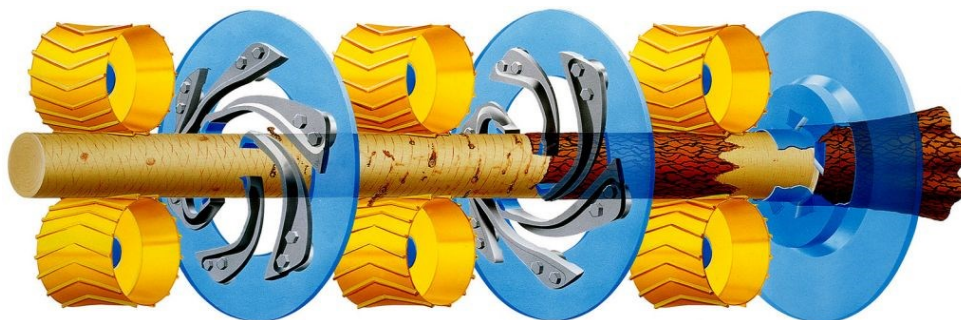
Tukit lajitellaan sahausluokittain ja lajittelun laajuus riippuu sahalaitoksen kapasiteetista ja lopputuotteiden laatuluokkien määrästä. Tukkilajittelussa lokeroita tarvitaan sitä enemmän, mitä enemmän on sahattavia lopputuotteita. Tukit etenevät pituussuunnassa linjaa pitkin, jossa ne kuvataan 3D-kameralla tai röntgen-mittalaitteella. Kuvauksen perusteella tukit puotetaan omiin lokeroihin linjalta. Tukkien lajittelulinjasto ja lajittelulokerot on esitetty kuvassa 1. (Varis 2017, 59.)



Kuva 1. Tukkilajittelu (Vänskä 2018)

2.2 Kuorinta ja sahaus

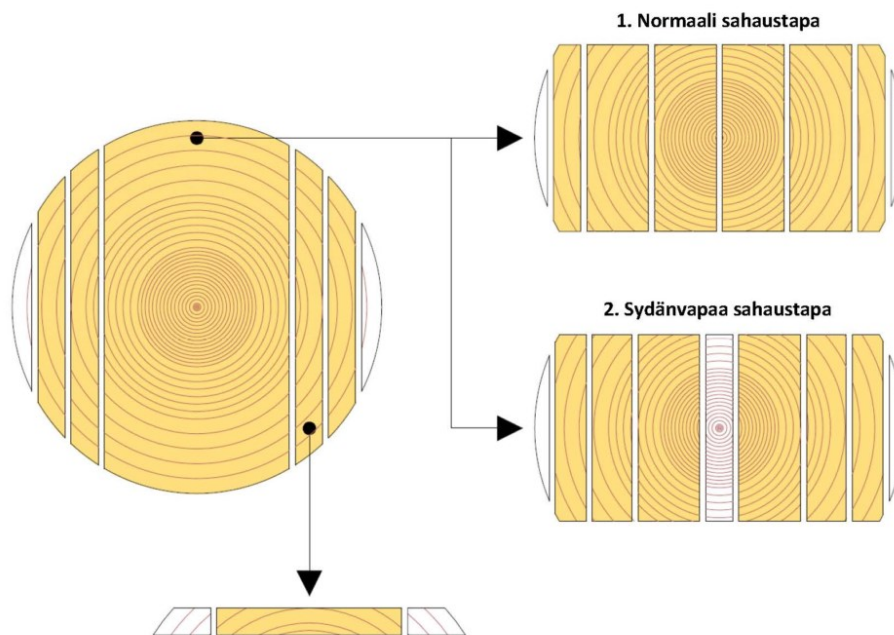
Ennen sahausta tukit kuoritaan. Yleisin kuorintalaite on roottorikuorinta. Tukki syötetään pituussuunnassa kuorintalaitteen läpi. Roottorin terät pyörivät tukin ympäri sen kulkiessa eteenpäin. Samassa yhteydessä voidaan tehdä tyven sievennys, jolloin tukista saadaan tasapaksu. Kuorintakoneella irrotettu materiaali voidaan hyödyntää lämpöenergiana voimalaitoksella. Kuorintalaitteelta tukit menevät tukin syöttöön, josta ne otetaan sahaukseen. Tyven sieventävä kuorintalaite on esitetty kuvassa 2.



Kuva 2. Roottorikuorinta (Valon kone 2021)

Sahaustapoja on erilaisia ja ne riippuvat tuotantolinjan ominaisuuksista ja jalostettavista lopputuotteista. Yleisimpiä sahaustapoja ovat vannesahaus ja pyörösahaus. Pyörösahauksessa käytetään usein lisäksi myös pelkkahakkuria, jolla tukin kahdelta sivulta poistetaan materiaalia. Tukkiäihiota kutsutaan pelkaksi hakkurin jälkeen, jolloin se on kahdelta sivulta tasainen. Tämän jälkeen tukki sahataan joko yksi- tai kaksivaiheisesti. Sivulaudat sahataan irti ja ne kuljetetaan särmäykseen. Siinä vajaakanttiset reunat ajetaan puhtaaksi tai jätetään hiukan vajaasärmäisiksi dimensiosta riippuen. Jäljellä jää tämän jälkeen puun neliskanttinen äihio. Siitä sahataan tukin sydäntavara. Sydäntavara on päätuote tukin sahausessa ja muut sahausessa syntyvät tuotteet ovat sivutuotteita, esim. sivulaudat.

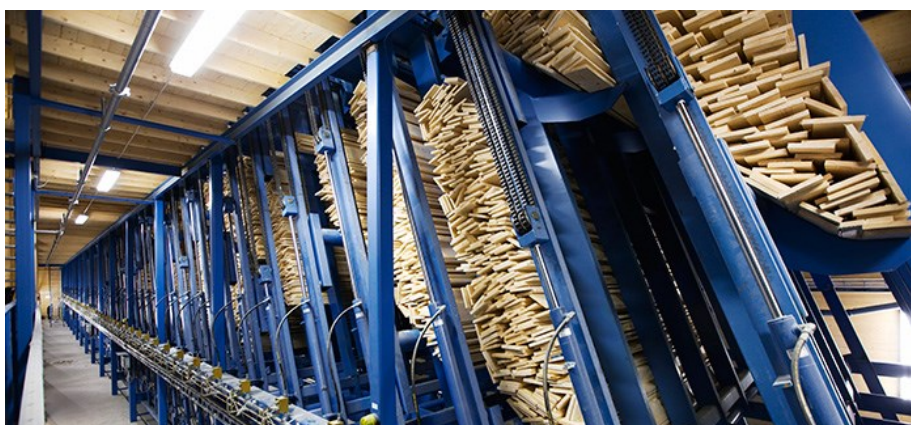
Joihinkin sahoihin on yhdistetty pelkkahakkuri ja sivulautojen sahaus sekä särmäys. Yhdistetyissä sahoissa tukkiäihio tarvitsee vähemmän kuljettimia, koska tukin kääntövaiheita ja kuljetusta vaiheiden välillä tarvitaan vähemmän. Tällöin sahattu tukki eli sydäntavara ja sivulaudat tulee sahattua yhtenä pakettina. Ne erotellaan tuorelajittelussa. Useampivaiheisessa sahausvaihe on pidempi ja sivulaudat kuljetetaan jo siinä vaiheessa erilleen. Vannesahauksen periaate on melko samanlainen, työvaiheita on kuitenkin enemmän. Sahaus voi tapahtua joko sahaamalla tukki läpi asti, jolloin kaikki sahatut laudat ajetaan kanttien särmäyksen kautta tai kolmivaiheisesti. Kolmivaiheisessa sahausessa sivulaudat sahataan irti ja tukista tehdään pelkka. Sahalaitoksesta riippuen sivulaudat ajetaan myös toiselta sivultaan, jolloin pelkka on nelikulmion mallinen. Vaihtoehtoisesti sivulautojen sahaus suoritetaan sydäntavaran sahausyhteydessä. Sivulaudat otetaan erilleen sydäntavaran sahausessa ja ne ajetaan särmäyksen kautta. Viimeisessä vaiheessa sahataan sydäntavara. Kuvassa 3 on esitetty erilaisia sahauskuvioita. Vasemmalla tukkiäihioista irrotetaan ensin sivulaudat, jolloin jäljelle jää pelkka, josta sahataan sydäntavara. Irrotettavat sivulaudat joko särmätään tai jätetään vajaasärmäisiksi.



Kuva 3. Sahauskuvioita (Puuinfo 2020)

2.3 Tuorelajittelu

Tuorelajittelua kutsutaan myös dimensiolajitteluksi. Tuorelajittelu alkaa heti sahauksen jälkeen. Tuorelajittelussa ajetaan yleensä sahauksessa käytetyn asetteen mukaan useampaa dimensiota samanaikaisesti. Sahattu puutavara lajitellaan dimensioiden eli leveyden ja paksuuden mukaisesti. Tuorelajittelussa lajiteltu puutavara ajetaan dimensioiden mukaisesti omiin lokeroihin välivarastoon. Kun lokeroissa on rimoituspinon verran lautoja, aletaan lokeroita purkamaan. Lokeroista puut kuljetetaan rimoitukseen. Rimoituksessa muodostetaan lautakerros kerrallaan puutavara pinoon ja jokaisen kerroksen väliin laitetaan rimat. Rimojen tarkoitus on parantaa ilman kiertoa, jolla saavutetaan parempi ja tasaisempi kuivaustulos. Tuorelajittelun lokerostoja esitetty kuvassa 4.



Kuva 4. Tuorelajittelun lokerosto (Renholmen 2021)

2.4 Kuivaus

Kuivauksessa puut pyritään kuivaamaan tyypillisesti 8-25% kosteuteen. Kuivauksessa puu saa tasapainokosteuden. Tasapainokosteus on tila, jossa puun kosteusprosentti pysyy vakiona. Puita kuivataan jopa n. 2-3 viikkoa puulajista ja puutavaran paksuudesta riippuen, jotta päästään haluttuun kosteuteen. Kuivaamista ei voi nopeuttaa, jotta voidaan minimoida laatuun vaikuttavia vikoja, kuten esim. halkeaminen, vääristyminen ja väriviat.

Kuivaustapoja on erilaisia, esim. korkean lämpötilan kuivaus, tyhjiökuivaus ja lauhdekuivaus. Yleisimpiä kuivaamotyyppisiä on kamarikuivaamo ja kanavakuivaamo. Kamarikuivaamossa rimoitettu sahatavaranippu pidetään paikallaan ja lämpötila stabiilina koko kuivausjakson ajan. Kanavakuivaamossa rimoitettu sahatavaranippu liikkuu hitaasti eteenpäin ja lämpötilat vaihtelevat kuivausjakson aikana. (Varis 2017, 129-134.) Rimoitettu sahatavaranippu kanavakuivaamossa on esitetty kuvassa 5.



Kuva 5. Kamarikuivaamo (Valutec 2021)

2.5 Kuivalajittelu

Kuivalajittelussa kuivatut sahatavaraniput puretaan. Sahatavara laajitellaan ja katkaistaan haluttuun mittaan. Kuivalajittelu on kuljetintarpeiltaan hyvin samankaltainen tuorelajittelun kanssa. Merkittävin ero on lajittelun eri tarkoitus. Joissain sahalaitoksissa onkin kuiva- ja tuorelajittelu toteutettu kombilajittelulaitoksella. Kombilaitoksella voidaan ajaa kumpaa tahansa lajittelua tuotannon tarpeiden mukaisesti. Laajittelun ja katkaisun jälkeen sahatavarat ajetaan paketointiin. Paketoitu tuote on valmis tuote jatkojalostukseen tai asiakkaalle.

3 Automaatiosuunnittelu

3.1 PLC-ohjelmointi

PLC-ohjelmointi (PLC = Programmable Logic Controller) eli logiikkaohjelmointi on automaattisten prosessien määrittelyä. Sitä käytetään yleisesti teollisuuden automaatiosovelluksissa. Ohjelmointi PLC:lle tehdään antureiden ja toimilaitteiden tuottamien tietojen perusteella. Anturin havaitessa kappaleen sen tulopisteen tila muuttuu tilasta 0 tilaan 1, jolloin tämä tieto kulkee PLC:lle. PLC:lle tulevasta tiedosta voidaan tehdä loogisia päätelmiä, joiden perusteella ohjataan lähtöä, joka esim. käynnistää moottorin.

Logiikkaohjelmointi ja siihen liittyvät loogiset päätelmät ennen mikroprosessorien yleistymistä 70-luvulla on pääsääntöisesti tehty releillä. Releohjauksen ohjauspiirissä on käytetty piirilevyjä, releitä, kytkimiä ja laskureita. Releiden johdottaminen vie paljon aikaa ja siihen on käytetty relekaavioita. Relekaaviot ovat toimineet pohjana nykyisin käytettävälle ladder-ohjelmointikielelle. Muutoksien tekeminen releohjattuihin järjestelmiin on vienyt valtavasti aikaa, koska yhdestä releestä voi lähteä johtimia moneen kohteeseen ja releen toiminta voi vaikuttaa useampaan laitteeseen. Pahimmassa tapauksessa koko johdotus pitää uusia muutoksien tai vikatoimintojen yhteydessä. Laitteiden testaus on myös rajallista ilman viimeisteltyä johdotusta. (Hanssen 2015, 24.) Puutteellisen testauksen johdosta tapaturmariski kasvaa virheellisen kytkennän tai suunnitelman vuoksi.

3.1.1 IEC 61131 Standardi

Standardin IEC 61131 ensimmäinen luonnos on julkaistu 1982. Standardin tarkoituksena on yhtenäistää ohjelmoitavien logiikoiden käyttöä. Ajatuksena on, että toisella ohjelmistolla luotua ohjelmaa voisi käyttää myös toisen valmistajan PLC:llä. Standardin perustajaryhmä on luotu IEC:n (International Electrotechnical Commission) toimesta. Perustajaryhmään on kuulunut PLC-laitevalmistajien työntekijöistä koostuvia asiantuntijoita. Standardi on niin laaja-alainen, että työryhmä on jaettu viidelle eri osa-alueelle. (Hanssen 2015, 199.)

Työryhmien käsittelemät osa-alueet sisältävät:

- yleistä tietoa ja ohjeita
- laitteistoa ja testaamista koskevat vaatimukset
- ohjelmointikielet
- käyttöliittymät
- laitteiden välistä kommunikointia.

Ohjelmointikielien yhtenäistämiseksi luodun standardin IEC 61131-3 ensimmäinen osa on julkaistu 1993. Standardiin on tullut lisäyksiä 2002. Kolmas ja toistaiseksi viimeisin osa on julkaistu 2013. (Hanssen 2015, 200.)

Standardin mukaisesti käytettävät ohjelmointikieliset ovat:

- jäsennelty teksti (ST)
- toimintalohkokaavio (FBD)
- tikapuukaavio (LD)
- käskylista (IL)
- juokseva funktiokaavio (SFC).

Käytetyimpiä ohjelmointikieliä on LD, FBD ja ST. Valmistajien käyttämät nimitykset kielistä ja niiden lyhenteistä vaihtelevat, mutta rakenteeltaan ne ovat yhteneväisiä.

3.2 Logiikat yleisesti

Tunnetuimpia logiikkavalmistajia ovat Siemens, Beckhoff, Allen Bradley, Mitsubishi ja Omron. Logiikkatoimittajat valmistavat logiikan keskusyksikön lisäksi tarvittavia komponentteja, esim. tulo- ja lähtökortit, virtalähteet, hajautusyksiköt, Ethernet reitittimet ja ohjelmointiympäristöt. Ohjelmointiympäristöllä luodaan looginen ohjelmakaavio sekä määritellään toimilaitteet ja niiden välinen kommunikointi.

Hajautusyksiköitä käytetään, jotta voidaan minimoida kaapelointia sekä kaapelireittien suunnitteluun käytettävää aikaa. Hajautuksessa PLC:n keskusyksiköltä toteutetaan esim. Profinet-yhteys hajautusyksiköille, joiden kautta toimilaitteiden välinen kommunikointi suoritetaan. Ilman hajautusyksiköitä jokainen I/O-laite pitää yhdistää PLC:hen erillisillä johtimilla. Hajautusyksikkö sijoitetaan niiden toimilaitteiden läheisyyteen, jossa sitä käytetään. Hajautusyksiköt ovat yleistyneet 90-luvulta lähtien. Ethernet-verkon yleistyessä hajautuksessa käytettävä tiedonsiirtoprotokolla on laajentunut sisältämään myös Ethernet-protokollan, joka mahdollistaa entistä laajempien tietomäärien siirtämisen kenttäväyläteknologioilla.

3.3 Siemens logiikat

Siemensin ohjelmoitavia logiikoita kutsutaan nimellä Simatic. Siemensin ohjelmoitavien logiikoiden tuoteperheet ovat julkaisuvuoden mukaisessa aikajärjestyksessä: (Siemens 2021a.)

- simatic G, 1959
- simatic N, 1964
- simatic 3, 1973

- simatic 5, 1979
- simatic 7, 1994

G ja N sarjat ovat perustuneet komponenteilla ohjattaviin piireihin. Ne ovat toimineet pohjana ensimmäiselle ohjelmoitavalle logiikalle S3. Se on prosessoriohjattu PLC, joka on mahdollistanut ohjelman muokkaamisen johdotuksen muuttamista. Simatic 5-sarjaan on lisätty huomattavasti lisää toimintoja ja laskentatehoa, jonka prosessoreiden kehittyminen on mahdollistanut. (Siemens 2021a.) Teollisuuden prosessien monipuolistuminen ja kasvavat tarpeet ovat myös sitä vaatineet. Vielä nykyisinkin teollisuudessa on jonkin verran käytössä S5-logiikalla ohjattuja sovelluksia.

3.3.1 Simatic S7

Viimeisin tuotesarja Siemensin ohjelmoitavissa logiikoissa on sarja S7. Se on julkaistu v.1994. Sarjaan ovat kuuluneet silloin tuotteet S7-200, S7-300, S7-400. Uuden sarjan laskentatehoa on kasvatettu ja lisätty toimintoja. Merkittävin uudistus on ollut laitteiden liitettävyyttä tietoverkkoon, joka on mahdollistanut toimilaitteiden välisen kommunikoinnin. Vuonna 2009 S7-sarjaan julkaistiin S7-1200 PLC. 1200-sarja mahdollisti PLC:n ja siihen liittyvien laitteiden modulaarisen yhdistämisen. Aikaisemmin PLC ja siihen liitettävät tulo- ja lähtökortit ovat olleet erillisiä komponentteja. 1500-sarja on julkaistu 2013. Sen ominaisuuksiin kuuluu TIA-portalin tuomat laajemmat ohjelmistomahdollisuudet ja turvatekniikka. (Siemens2021a.)

3.3.2 TIA portal

TIA-portal (TIA = Totally Integrated Automation) on ollut Siemensin visiona vuodesta 1996. Sen tarkoituksena on yhdistää Siemensin eri ohjelmistot yhteen sovellukseen. Ensimmäinen versio TIA-portaalista julkaistiin 2011. TIA-portal-sovelluksella voidaan määrittää kaikkien projektiin liittyvien laitteiden kommunikointi hajautusyksiköiden ja toimilaitteiden välillä. TIA-portal toimii ohjelmointiympäristönä logiikalle ja käyttöliittymälle. Käyttöliittymän suunnittelu ja sen toimintojen linkitys logiikkaohjelmointiin tapahtuu TIA-portaalin kautta. Samassa ohjelmointiympäristössä voidaan myös parametroida moottorikäyttöjen ohjaimia. (Siemens 2021a.)

3.3.3 Turvalogiikka

Turvalogiikan tarkoituksena on välttää henkilöön, ympäristöön tai koneeseen liittyvä vaara silloin, kun sovellusohjelma ei toimi halutulla tavalla. Suunniteltava turvaohjelma varmistaa myös, ettei vika- ja häiriötilanteissa pääse tapahtumaan vakavaa tapaturmaa. Turvalogiikan

suunnittelua ohjaa SFS-EN 954-1-standardi, jonka tarkoituksena on hierarkisilla ratkaisulla ohittaa sovellusohjelman toiminnot vikatilanteissa. (Kivipuro & Malm 2004, 9.) Koneen turvallisen toiminnan kannalta tarkentavia ja suuntaa-antavia standardeja on useita. Tärkeintä on aikakriittinen ja pääohjelmasta riippumaton toiminta vika- ja häiriötilanteissa. Tyypillisiä turvatoimintoja esitetty taulukossa 1.

Päälle-/pois-tyyppiset toiminnot	<ul style="list-style-type: none"> - Käynnistäminen, pysäyttäminen, hätäpysäyttäminen, energian katkaisu, jarruttaminen, venttiilin sulku tai avaus - Toimintatavan valinta
Yksinkertaisen suureen valvonta	<ul style="list-style-type: none"> - Lämpötilan, paineen, nopeuden, massan tms. suureen valvonta - Liikealueen rajan tunnistaminen
Laskentaa edellyttävän suureen valvonta	<ul style="list-style-type: none"> - Kuormituksen, vakavuuden, tms. valvonta - Turvallisen toiminnan loogisuuden valinta - Työkalupisteen nopeus
Turvallisuuteen liittyvät säädöt ja ohjaukset	<ul style="list-style-type: none"> - Nopeuden, kiihtyvyyden säätö - Liikkeen tai liikeradan ohjaus ja valvonta - Virtauksen, paineen tms. säätö - Työkalun kiinnittäminen ja valvonta

Taulukko 1. Tyypillisiä turvatoimintoja (Kivipuro & Malm 2004, 10)

Turvakomponentit erottaa muista komponenteista värityksen perusteella. Turvakomponentit ovat keltaisia. Siemensin PLC:n mallimerkintä sisältää kirjaimen F, jos siinä on turvatoiminnot mukana.

3.3.4 Väylätekniiikka

Teollisuusympäristö asettaa haasteita Ethernet-verkolle. Teollisuuden tiedonsiirto on aikakriittistä ja toimilaitteet vaativat lyhyitä vasteaikoja erityisesti turvasovelluksissa. Teollisuuden tuotantolaitoksissa käytetään paljon raskaita koneita, joissa on isot moottorit.

Moottorikäytöt saattavat aiheuttaa elektromagneettisia häiriöitä verkkoon ja raskaat laitteet aiheuttavat tärinää, joka häiritsee tietoliikennettä. (Nurro 2016, s12.)

Teollisuuden tarpeisiin on standardoitu Ethernet-verkkoa hyödyntävä Profinet-teollisuusväylä. Profinetin tietoliikenne on reaaliaikaista ja aikakriittistä. Tämä mahdollistaa samanaikaisesti TCP/IP-toiminnot. Profinet-standardi on kehitetty yhteistyössä Siemensin sekä Profibus ja Profinet Internationalin kanssa. Profinetissä välitettävän datan siirrot on määritelty niin, että kaikkea dataa ei tarvitse kuitata vastaanotetuksi ja osa datasta ”häviää”. Profinetin data vastaanotetaan ja lähetetään tärkeysjärjestyksessä niin nopeasti kuin mahdollista toimilaitteen tarvitseman tiedon mukaan. (Nieminen 2019, s13.) Profinet-standardissa määritellään, että siirrettävä tieto saadaan kohteelle millisekunneissa, joka on tärkeää varsinkin paikoitussovelluksissa. Ethernet-verkko ei pysty sitä takaamaan. Profinetin tarkoitus on taata tarvittava määrä tietoa tarvittavassa ajassa teollisuussovelluksissa ja se on osittain kehitetty paikkaamaan Profibus-kenttäväylän puutteita tiedonsiirron määrän suhteen.

Profibus-yhteyttä on käytetty ennen Profinetin standardisointia. Profibus-väylästä on olemassa 3 eri versiota käyttötarkpeiden mukaan. Profibus FMS (FMS = Fieldbus Messaging Specification) on kehitetty siirtämään tieto suurilla nopeuksilla logiikan ja PC:n välillä. FMS on kuitenkin sisältänyt puutteita kokonaisuuksien kasvaessa. Profibus DP (DP = Decentralized Periphery) on kehitetty paikkaamaan edellisen version puutteita yksinkertaisemmissa tiedonsiirroissa. Profibus DP:n tiedonsiirtonopeutta on kasvatettu ja se on tarkoitettu kommunikoimaan hajautetuissa ohjausjärjestelmissä. Kolmas versio Profibus PA (PA = Process Automation) on kehitetty prosessiteollisuutta varten ja se on kehitetty DP:n pohjalta. (Perälä 2013, 13-16.)

4 Malliohjelman rakenne

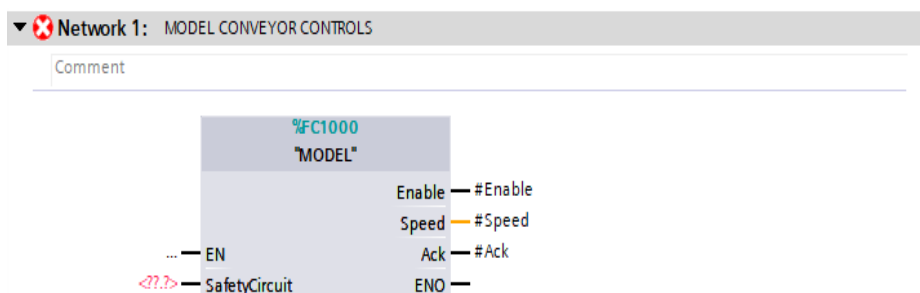
4.1 Malliohjelman runko

Kuljetinohjauksen malliohjelman runko koostuu kahdesta aliohjelmasta, jotka yksilöidään kyseessä olevan kuljettimen moottorinumeron tai sähkölaitetunnisteen mukaan. Ohjelman monitorointi sekä mahdollisen vian paikantaminen on helpompaa yksilöllisen ja tunnistettavan nimeämisen johdosta. Aliohjelmia kutsutaan pääohjelmassa ja logiikka suorittaa aliohjelman sisällön ohjelmakierron mukaisesti loogisessa järjestyksessä.

4.2 Toiminto FC

Kuljetinohjauksissa toistuvia komentoja ovat eteen- ja taakse-ohjaukset. Kuljettimen toimiessa automaattiohjauksella ei yleensä tapahdu taaksepäin ajoa poikkeustilanteita lukuun ottamatta. Pääsääntöisesti sahalaitoksien kuljetinohjauksien suunnan määrittelyt tehdään komennoille automaattiohjaus eteenpäin sekä käsiohjaus taakse- ja eteenpäin. Kuljetinohjauksessa määritellään kyseiselle kuljettimelle kriittiset turvalaitteet ja -piirit sekä kuittaus häiriöille. Lohkossa kutsutaan erillisellä aliohjelmalla suoritettavaa suunnanvaihtoviivettä ja ohjausta ajonopeuteen nolla. Erillinen aliohjelma on osa aikaisemmin toteutettuja ohjelmia ja sitä käytetään myös malliohjelman kanssa. Erillinen aliohjelma on olennainen osa kuljetinohjauksen runkoa.

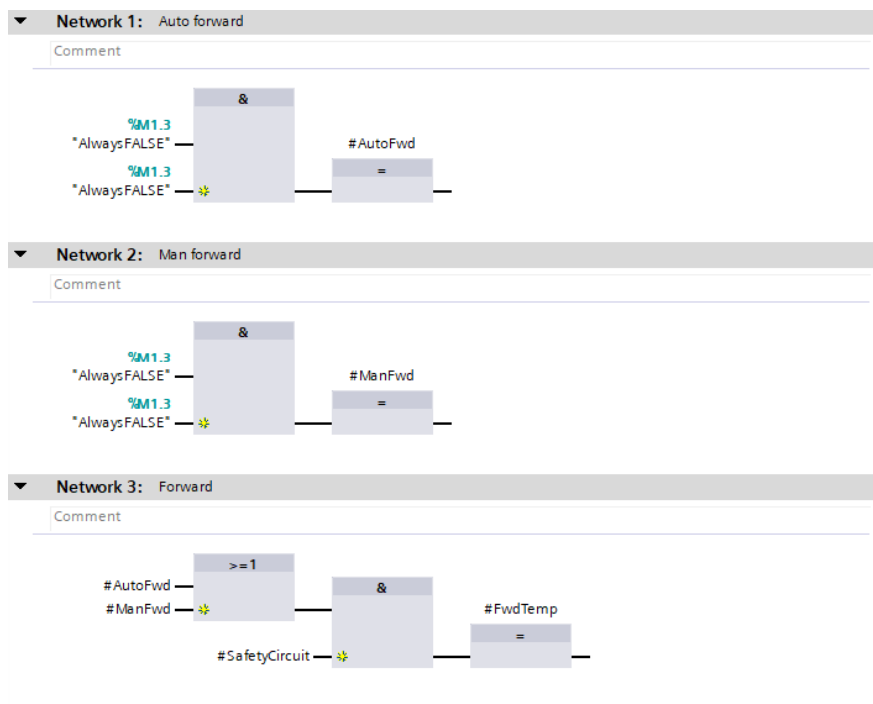
Kuljettimen ohjauksen aliohjelmakutsu on esitetty kuvassa 6. Lohkon sisääntulotietona käytetään turvapiirin tilatietoa. Turvapiirin ollessa kuitattuna lohkon sisältö käsitellään loogisessa järjestyksessä. Lohkon ulostulotietona saadaan lupa taajuusmuuttajalle olla toiminnassa ja nopeusohje kyseiselle taajuusmuuttajalle. Jos taajuusmuuttajan toiminnassa tapahtuu häiriö, saadaan se kuitattua lohkon kuittaustiedolla.



Kuva 6. Malliohjelman kutsu pääohjelmasta

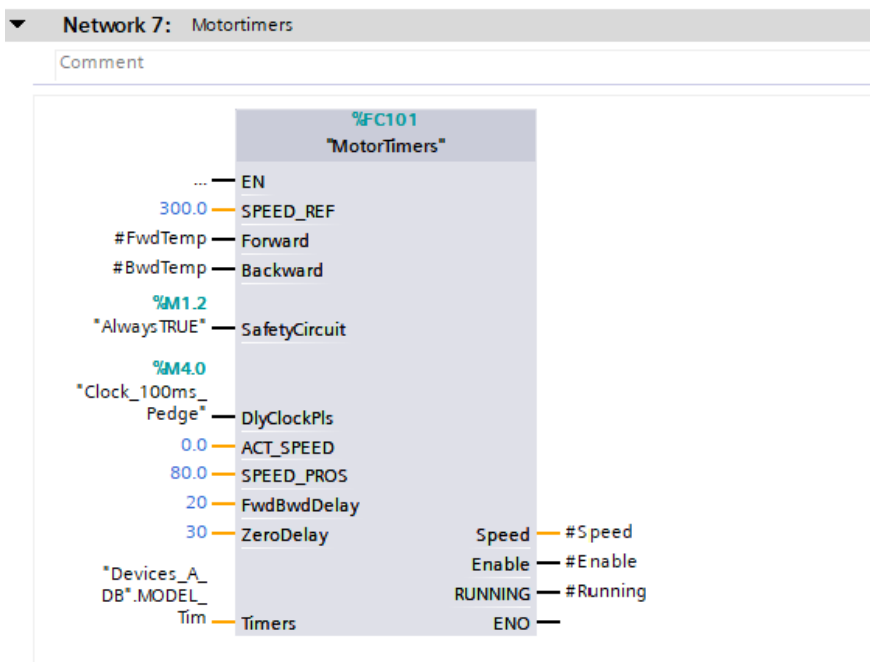
Eteenpäinajon toimintolohkokaavio on esitetty kuvassa 7. Eteenpäinajot määritellään automaatti- ja käsiajolle, joiden määrittelyjen perusteella FwdTemp-muuttuja antaa luvan

taajuusmuuttajalle ajaa eteenpäin. Malliohjelmassa käytetään AlwaysFALSE-muuttujia ennen kuin siihen liitetään projektikohtaisia tulotietoja. Jos samanaikaisesti luodaan useampi pohja kuljetinohjauksiin, niin ohjelma saadaan käännöksestä läpi ilman virheilmoituksia. AlwaysFALSE-muuttuja ei nimensä mukaisesti ole milloinkaan aktiivinen, jolloin ei ole riskiä siitä, että jokin kuljetin lähtisi käyntiin ohjelman latauksen jälkeen.



Kuva 7. Toimintolohkokaavio moottorin eteenpäinajoa varten

Kuljettimia ajettaessa käsiajolla on tarpeellista suorittaa viive suunnanvaihdokselle, ettei taajuusmuuttaja riko. Kuljettimilla olevien kappaleiden massa voi liikuttaa joko kuljetinta tai kappaleita eteenpäin äkillisellä suunnanvaihdoksella. Suunnanvaihdos voi aiheuttaa laiterikon, koska hetkellinen kuormitus kasvaa. Lohkossa suoritetaan nopeuteen nolla ajo. Jos kuljetin pysäytetään, PLC ohjaa kuljettimen nopeuteen nolla eikä vain katkaise ohjausta äkillisesti. Tällöin pysäyttäminen on hallitumpaa ja uuden nopeusohjeen suorittaminen nopeampaa, koska moottorien käämit pysyvät magnetoituneena. Suunnanvaihtoviive ja 0-nopeusohjeen ohjelmalohko on esitetty kuvassa 8.



Kuva 8. Moottorin suunnanvaihtoviive ja 0-nopeus ajo-ohje

4.3 Multi-instanssi

Toimintolohkoa FB (FB = Function Block) käytettäessä ohjelmalohkolle luodaan ohjelma-kutsu. Kun toimintolohkoa kutsutaan, TIA portal luo toimintolohkolle datalohkon eli DB:n (DB = Data Block). Datalohkoa käytetään tietojen tallentamiseen ja se on silloin toimintolohkon instanssi. Jos ohjelmassa käytetään useita eri toimintolohkoja voi ohjelma tulla epäselväksi, koska jokaiselle toimintalohkolle luodaan oma tietolohko. Ohjelma sisältää rajallisen määrän tietolohkoja. Ohjelman selkeyttämiseksi voidaan toimintolohkolle luoda multi-instanssi DB. Datalohkosta varataan silloin oma alue kyseiselle toimintolohkolle. Multi-instanssia käyttämällä datalohkoja tarvitaan vain yksi. Toimintolohkon on kuitenkin oltava rakenteeltaan samanlainen ja toimintolohkolle on ohjelmassa oltava kutsuja enemmän kuin yksi. (Siemens 2007b.) Taajuusmuuttajan toimintolohko on esitetty kuvassa 9. Toimintolohko on Siemensin valmis lohko, jolle tehdään multi-instanssi DB. Kuljetinmäärän kasvassa instanssit nimetään yksilöidyillä nimillä static-muuttujiin ja niitä kirjoitetaan ja luetaan multi-instanssi-datalohkossa. Kaikki taajuusmuuttajat käyttävät samaa toimintolohkoa. Jokaiselle kuljettimelle on varattu oma muistialue, jolloin sen käyttö on mahdollista ilman, että toinen ohjelmalohko kirjoittaisi dataa toisen päälle. Kuljettimet voivat siten käyttää samaa toimintolohkoa samanaikaisesti.

Ohjelman toiminnan kannalta ei ole merkittävää luodaanko jokaiselle lohkolle oma instanssi ja DB vai tehdäänkö globaali DB jonne multi-instanssit luodaan. Multi-instanssin etuna on, että se vähentää komponenttien määrää projektipuussa, jolloin se pysyy helpommin

tulkittavana. Multi-instanssin luominen myös vähentää käytettävien tietokanta-DB:n tarvitsemia muistia ja laskentatehoa. Sen vaikutus on kuitenkin nykyaikaisilla tehokkailla CPU-yksiköillä vähäinen. (Kovalainen 2021.)

Network 2: MODEL Drive

Comment

```

#MODEL_Inst
%FB10
"PZD_G120_TeL_352"
... EN
0 address
#Enable ON_OFF1 ready_for_
TRUE OFF2 switching_on false
TRUE OFF3 ready_for_
enable_ operation false
TRUE operation enabled false
TRUE release_RFG fault_active false
TRUE enable_RFG OFF2_inactive false
enable_ OFF3_inactive false
TRUE setpoint closing_
#Ack acknowledge lockout_active false
FALSE CTW_bit_08 alarm_active false
FALSE CTW_bit_09 speed_
PLC_is_master deviation_in_
direction_ PLC_control_
reversal requested false
FALSE CTW_bit_12 comp_speed_
MOP_up limit_reached false
FALSE MOP_down holding_
FALSE CTW_bit_15 brake_open
activate_ alarm_motor_
manual_ rotating_right false
mode no_thermal_
#Speed speed_setpoint overload false
reference_ actual_speed 0.0
3000.0 speed_P2000 actual_current 0.0
reference_ actual_torque 0.0
0.0 current_P2002 actual_alarm 16#0
reference_ actual_fault 16#0
0.0 torque_P2003 ENO

```

Kuva 9. Malliohjelman taajuusmuuttajan instanssin kutsu pääohjelmassa

4.4 Sovelluskohteet

Malliohjelmatarun on suunniteltu erityisesti taajuusmuuttajalla ohjattuihin kuljettimiin. Malliohjelmatarun soveltuu myös suoran moottorikäytön ohjauksiin. Ohjelmatarungosta poistetaan tällöin viiveohjaukset suunnanvaihdolle ja nolla-nopeusohje. Suoran käytön ohjaukselle ei luoda multi-instanssia. Suora moottorikäyttö ei tällöin kutsu erillisiä aliohjelmia ja sen kutsussa pääohjelmassa ei tarvitse lähtötietoja. Turvaohjelma liitetään kutsun sisääntulotietoihin. Samalla periaatteella malliohjelmatarungolla voidaan toteuttaa hydraulisia

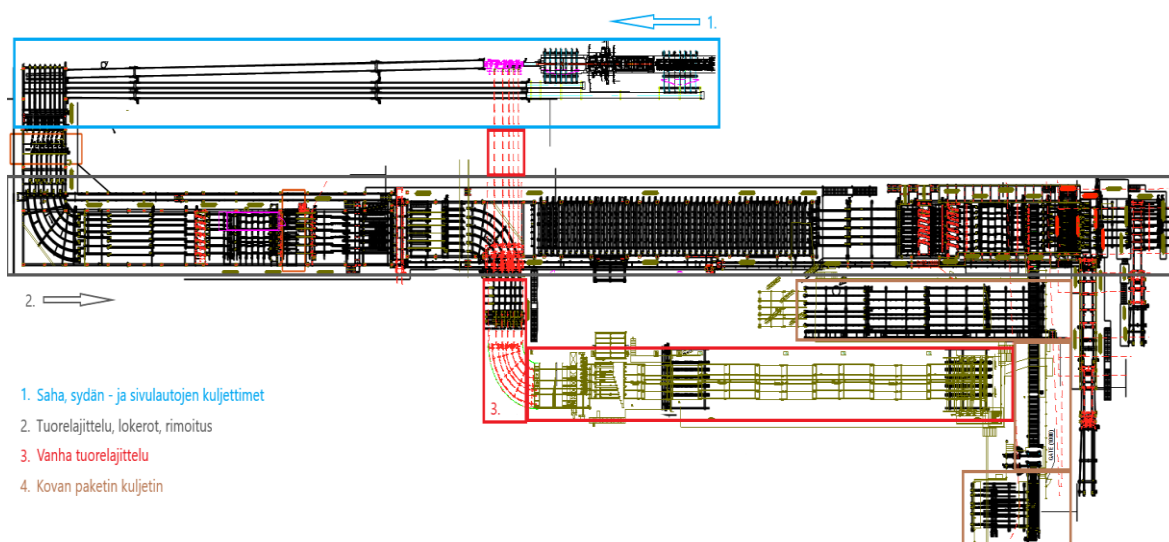
ohjauksia. Tällöin lohkon sisällä voidaan ohjata suoraan lähtötilan tietoa esim. magneettiventtiilille, joka aukeaa tai menee kiinni. Hydraulista ohjausta käytetään esim. rimoituksen hississä. Magneettiventtiin lähtö sijoitetaan FwdTemp-muuttujan perään. Hissin ollessa kyseessä muuttujan nimi muokataan Down- tai UpTemp-muuttujaksi.

Uuden kuljettimen luominen ohjelmaan tapahtuu kopioimalla kuljetinohjauslohko-FC ja nimeämällä se kuljettimen mukaisesti. Tämän jälkeen sitä kutsutaan FB-lohkossa ja sille luodaan nimensä mukainen instanssi Static-muuttujiin. Niitä kirjoitetaan ja luetaan multi-instanssi DB-datalohkossa.

5 Malliohjelman soveltaminen käytännössä

5.1 Asiakasyritys

Malliohjelman perusteella toteutettua logiikkaohjausta sovelletaan käytännössä Ranskassa sijaitsevaan sahalaitokseen, joka kuuluu Piveteau-ryhmään. Sahalaitos sijaitsee Fargesbois-teollisuusalueella Egletonsin kaupungissa Ranskassa. Fargesin saha on Ranskan neljänneksi suurin sahatavaran valmistaja. Sahalaitos tuottaa sivutuotteista pellettiä ja on pelletin valmistusmäärältään Ranskan suurin tuottaja. Laitos modernisoi sahan toimintaa tuorelajittelun ja rimoituksen osalta. Modernisoinnin tarkoituksena on nostaa vuosittaista tuotantokapasiteettia sekä tehdä tuotteelle parempi ja tarkempi laadutus. Parempi laadutus ja lisääntynyt tuotantokapasiteetti nostavat tuotteen jalostusarvoa. Piveteau-ryhmä käsittelee vuosittain n. 700 000 m³ tukkipuuta sahalinjallaan. Modernisoinnin tavoitteeksi on tuorelajittelun tahtiajan suhteen määritetty 180 kolaväliä minuutissa. Tehtaan layout on esitetty kuvassa 10.



Kuva 10. Modernisoinnin layout

Tuorelajittelun ja rimoituksen modernisointi tapahtuu kolmessa vaiheessa ja tarkoitus on pystyä ajamaan tuotantoa modernisoinnin ajan vanhalla linjastolla. Kun modernisoinnin kukin vaihe saadaan mekaanisesti ja ohjelmallisesti valmiiksi, sahaustoiminta pysäytetään ja uudet laitteet liitetään vanhoihin. Ensimmäisessä vaiheessa vanha pakettikuljetinlinjasto liitetään uuteen sidonta- ja poissyöttölinjastoon. Vanhan pakettilinjaston kommunikointi uuteen logiikkaan tehdään erillisellä kommunikointi-toimintolohkolla Profinet-kenttäväylän välityksellä. Toisessa vaiheessa valmistellaan tuorelajittelun kuljettimet, rimoitus sekä lajittelu käyttökuntoisiksi. Toisen vaiheen aikana vanha tuorelajittelu on käynnissä. Kolmannessa vaiheessa vanha ja uusi tuorelajittelu liitetään toisiinsa. Opinnäytetyössä suunniteltavaa

ohjelmarunkoa sovelletaan modernisoitavan sahalaitoksen jokaisessa vaiheessa. Kohdeprojektissa on yli 100 taajuusmuuttajalla ohjattua kuljetinta, joiden pohjana käytetään malliohjelmarunkoa. Lisäksi malliohjelmarunkoa voidaan soveltaa suoriin moottorikäyttöihin tai hydraulisiin käyttöihin, kuten hisseihin. Ilman taajuusmuuttajaa toteutettavat käytöt eivät tarvitse taajuusmuuttajan toimintolohkoa. Tällöin ohjelman kutsuihin lisätään suoran käytön toimintokutsu.

5.2 Lähtötiedot

Malliohjelman soveltamisen lähtötietoina toimii toimitettavan modernisoinnin layout, käyttölaitelista, I/O-lista ja toimintakuvaus. Layout on kaikkien suunniteltujen laitteiden yleisnäkymä, josta näkee jokaisen laitteen sijainnin. Layoutin pohjalta on helpompi hahmottaa materiaalivirran kulkusuunta. Tämä auttaa logiikkaohjelmoinnin suunnittelussa. Mekaniikkasuunnittelussa jokainen laite on oma positionsa. Suunnittelun lopuksi positiot sijoitellaan yhdeksi kokonaisuudeksi eli layoutiksi. Logiikkaohjelmointi suunnitellaan myös positioittain. Jokaisen position logiikkaohjelma on itsenäinen kokonaisuus kunnes ne yhdistetään. Yhdistäessä positioden välillä on rajapinta, millä määritellään materiaalin siirto seuraavaan positioon. Rajapinta voi olla esim. anturin tilatieto (onko kuljetin vapaa) tai tietokantaan tallennettu tieto siitä mitä seuraavaksi tuotannossa tarvitaan.

Käyttölaitelista määrittelee käyttötyypin kullekin positiolle. Käyttö tarkoittaa miten positiota liikutetaan. Jos kyseessä on esim. ketjukuljetin, sen käyttövoimana toimii taajuusmuuttajalla ohjattu sähkömoottori. Listassa on määritelty taajuusmuuttajien malli. Tätä tietoa tarvitaan TIA portalin laitekonfiguraation määrittelyssä.

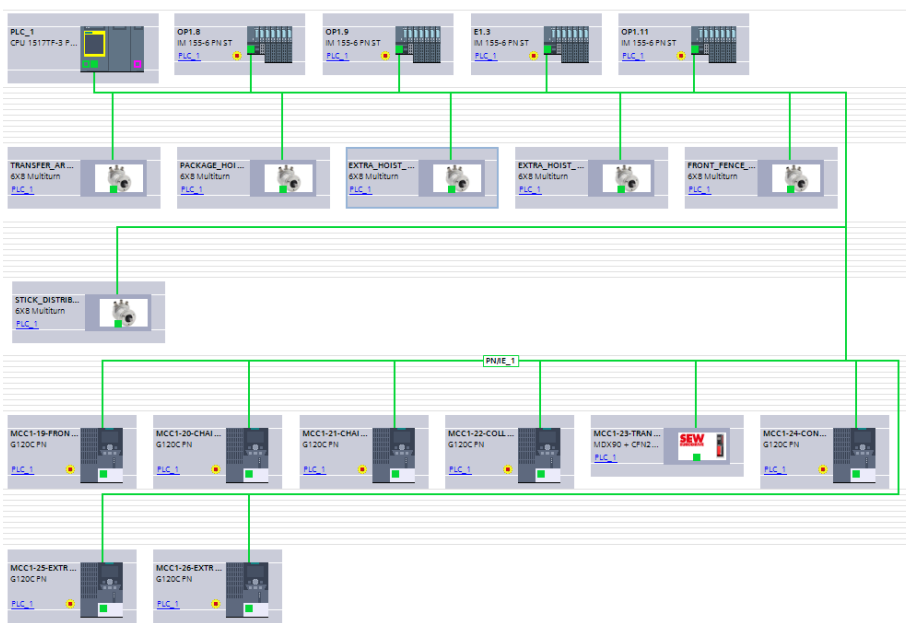
Sähkösuunnittelija suunnittelee I/O-listan, jossa määritellään kaikkien toimilaitteiden tulo- ja lähtötietojen osoitteet. I/O-lista on ohjeena sähköasentajille ja logiikkaohjelman suunnittelijalle. Listassa olevien osoitteiden perusteella logiikkaohjelmoija yhdistää esim. oikean kytkimen oikeaan ohjauspaneeliin ja sähköasentaja johtimen hajautusyksikön tulokorttiin. Toimilaitteiden tulotietona ovat esim. anturit ja kytkimet. Turvalogiikassa tulotietoja ovat esim. hätäseisäkytkimet, turvaportit ja valoverhot. Logiikkaohjelmoinnissa tulotietoa käytetään apuna määriteltäessä, milloin lähtö ohjataan päälle. Logiikka suorittaa toimilaitteen ohjauksen eli lähdön päälle, esim. moottorin pyörintä, merkkivalon syttyminen tai sylinterin liike.

Toimintakuvaus on mekaniikkasuunnittelijan tuottama dokumentti kyseisen laitteen toiminnasta. Kuvauksessa käsitellään yksityiskohtaisesti, esim. millainen anturi laitteessa on, missä kohdassa laitetta ja miten sen tiedon perusteella tehdään laiteohjauksia. Logiikkaohjelman on toimittava toimintakuvausten määrittelemällä tavalla.

5.3 Hardware

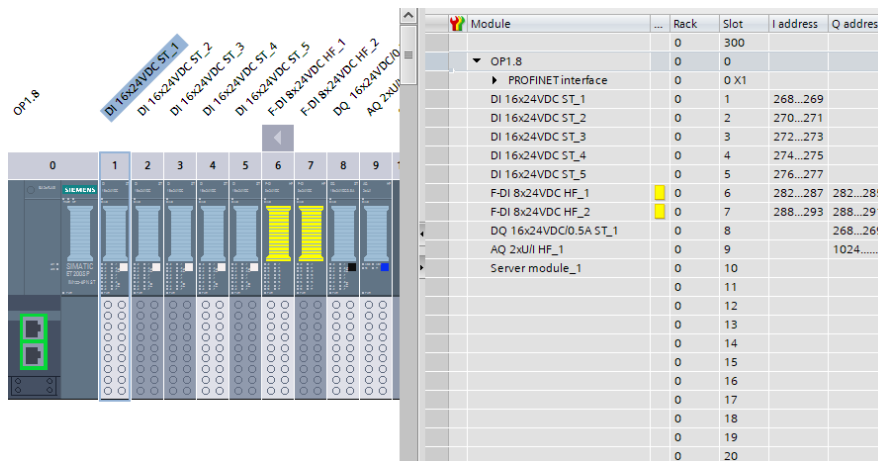
Tia portalin tarkoituksena on yhdistää toimilaitteiden ja hajautuksien kommunikointi. Hardwaren määrittelyssä lisätään ohjausjärjestelmän komponentit, esim. logiikka, absoluuttianturit, taajuusmuuntajat ja hajautusyksiköt. Komponenttien malli pitää määrittelyssä olla sama kuin fyysisen laitteen. Toimilaitteet nimetään positioiden tai moottoreiden mukaisesti. Laitteille määritellään telegrammi ja ip-osoite. Telegrammeja on erilaisia eri käyttötarkoituksiin. Telegrammin valinta perustuu siihen kuinka paljon kyseinen toimilaitte lähettää tai vastaanottaa tietoa. Telegrammi varaa logiikalta I/O-alueen toimilaitteen antaman tai tarvitseman tiedon käsittelyyn. Määrittelyllä varmistetaan laitteiden välinen välttämätön tiedonsiirto.

Kuvassa 10. on esitetty laitekonfiguraatio Profinet-väylään. Kuvassa olevat laitteet liittyvät tuorelajittelun rimoitukseen. Kuvan symbolit ovat vasemmalta oikealle ja ylhäältä alas lueteltuina: logiikka, hajautusyksikkö, absoluuttianturi, taajuusmuuttaja ja servovahvistin.



Kuva 10. Hardware kytkettynä Profinet-kenttäväylään

Kuvassa 11 on hajautusyksikön määrittely. Jokaisen tulo- tai lähtökortin malli valitaan sähkösuunnitelmien mukaisesti sekä sille määriteltä potentiaaliryhmä. Kuvassa näkyy I/O-osoitteet, jolle kunkin kortin osoitteet on määritetty. Turvakorteista määritellään erikseen liittimet, jotka ovat käytössä ja saako kyseinen toimilaitte virran tulokortilta vai ulkopuolisesta lähteestä.



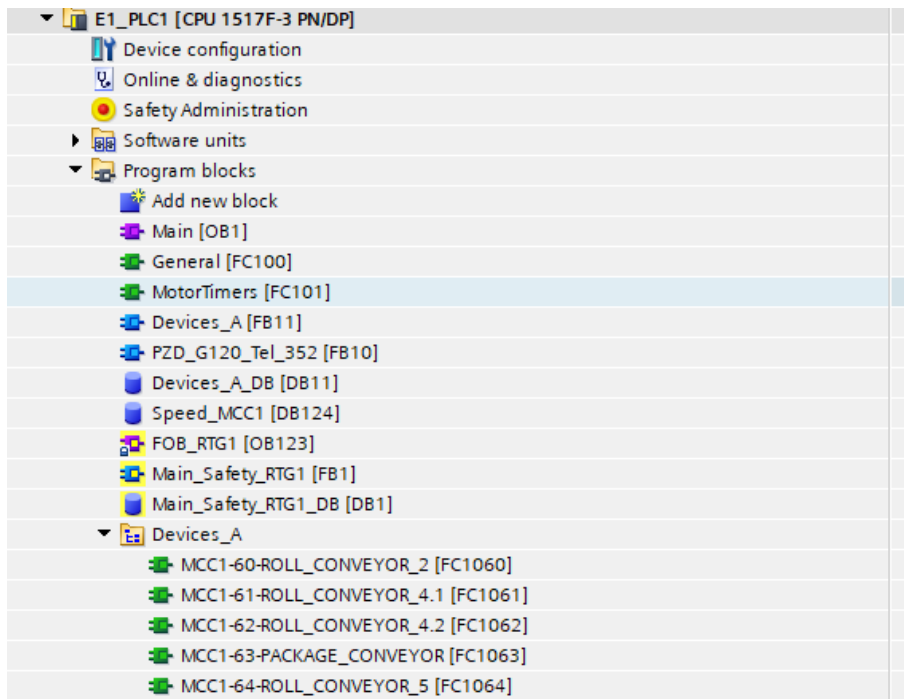
Kuva 11. Hajautusyksikkö

5.4 Lohkojako

Sahalaitoksen ohjelman suunnittelussa ohjattavat kuljettimet ja laitteet on jaoteltu omiin lohkoihin. Lohkot on määritelty prosessin suuntaisesti tai sijainnin mukaan. Tukin sahauksen jälkeen on kolme pitkää kuljetinta, joista yksi on sydäntavaralle ja kaksi muuta sivulautoille. Sydäntavaraa ajetaan tuotantolinjalla päätuotteena. Sydäntavaraan liittyvät kuljettimet ovat oma lohkonsa. Sydäntavaran ja sivulautojen linjastot yhdistyvät toisiinsa prosessin myöhemmässä vaiheessa. Sahalaitoksen tuotanto suunnitellaan niin, että jos sivulautoja tulee vähän, saadaan niitä ajettua välivarastoihin. Tällöin tuotanto voidaan keskittää sydäntavaraan. Loppupään tuotanto on sijaintipohjainen eli samassa lohkoissa on saman alueen laitteet. Lohkojen koko määritellään niin, että ohjelma pysyisi mahdollisimman selkeälukuisena. Lohkot ovat nimetty aakkosjärjestyksen mukaisesti.

5.5 Ohjelman rakenne

Ohjelmaan on määritelty yleisiä toimintoja ja tietokantalohkoja sekä toistuvia toimintoja, joita ei ole sijoitettu kuljetinohjauslohkoon. Yleisiä toimintoja voidaan käyttää missä tahansa ohjelmassa. Ohjelman rakennepuu on esitetty kuvassa 12.



Kuva 12. Ohjelman rakennepuu

Ohjelman rakenne alkaa aina pääohjelmasta, jossa kutsutaan seuraavaksi suoritettavia ohjelmia. Pääohjelmassa kutsuttu ohjelma tai toiminto suoritetaan pääohjelman kutsujen mukaisesti loogisessa järjestyksessä. Pääohjelma kutsuu kuljetinohjauslohkoa ja siihen liittyvää tietokantalohkoa ja suorittaa niiden sisältämät ohjelmat. Pääohjelman kutsut on esitetty kuvassa 13.



Kuva 13. Pääohjelman kutsut

Toimintalohkokaavioon sijoitetaan kutsuttavat toiminnot. Toimintolohko FC on kuljetinkohdainen kutsu. Logiikka määrittää kuljetinkohdaisen toimintolohkon päätelmänä annetaanko taajuusmuuntajalle lupa ajaa sekä nopeusasettelun. Päätelyn perusteella logiikka suorittaa seuraavaksi kuljettimen taajuusmuuttajan toiminnot. Kuljetinlohkon sisältö on esitelty

kuvassa 14. Kuvasta voidaan havaita, miten jokainen kuljetin kutsuu samaa taajuusmuuttajan toimintalohkoa. Jokainen kuljetin on luonut oman instanssin toimintalohkoon, jolloin toimintalohkoon liitetty tietokanta toimii multi-instanssina.

Farges_Phase1 ▶ E1_PLC1 [CPU 1517F-3 PN/DP] ▶ Program blocks ▶ Devices_A [FB11]

Devices_A

Name	Data type	Default value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervis...	Comment
1 ▶ Input									
2 ▶ Output									
3 ▶ InOut									
4 ▶ Static									
5 ▶ MCC1-70_Package_C...	"PZD_G120_Tel_352"								
6 ▶ MCC1-70_Package_C...	"DelayTm"		Non-retain						
7 ▶ MCC1-71_Package_C...	"PZD_G120_Tel_352"								
8 ▶ MCC1-71_Package_C...	"DelayTm"		Non-retain						
9 ▶ MCC1-72_Package_C...	"PZD_G120_Tel_352"								
10 ▶ MCC1-72_Package_C...	"DelayTm"		Non-retain						
11 ▶ MCC1-73_Packane_C...	"PZD_G120_Tel_352"								

Block title: ...
Comment

- Network 1: MCC1-70_Package_Conveyor802.1 Controls
- Network 2: MCC1-70_Package_Conveyor802.1 Drive
- Network 3: MCC1-71_Package_Conveyor802.2 Controls
- Network 4: MCC1-71_Package_Conveyor802.2 Drive

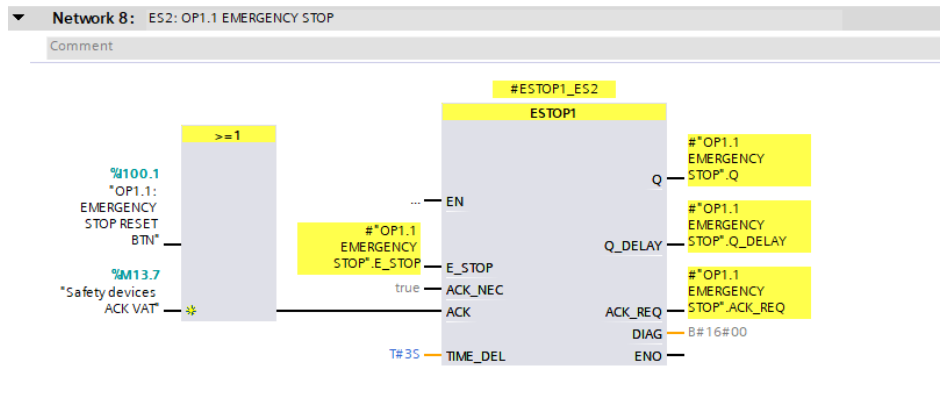
Kuva 14. Devices A:n sisältö

Malliohjelman runnon soveltaminen käytännössä kuljetinohjauksen toimintalohkoon tapahtuu lisäämällä tulot ja lähdöt malliohjelman. IO-listaa käytetään apuna määriteltäessä kuljettimen toimintoja. IO-listasta selviää antureiden ja toimilaitteiden osoitteet. Kuvassa 15 on esitetty automaattivalinnalla eteenpäin-toiminto, johon on lisätty tulotietoja. I/O-osoitteiden lisäksi ohjelmassa käytetään M-alueen apumuistia. M-alueen muisti voi olla esimerkiksi useammasta tulotiedosta määritelty toiminto. M-alueen muistiin määriteltyä toimintoa voidaan käyttää useammassa eri kohdassa. M-alueen muisti eroaa tietokantalohkoista siten, että sen muisti nollautuu, jos virrat eivät ole päällä.



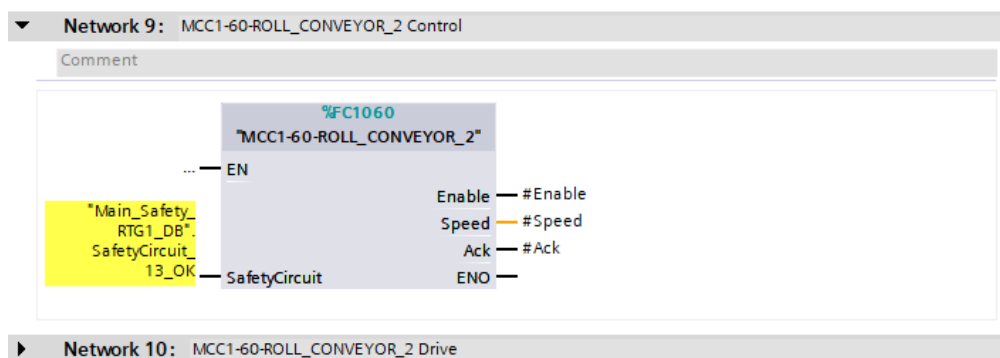
Kuva 15. Automaattinen eteenpäinajo ja lisätyt määritelmät.

Turvaohjelmaa ei kutsuta ohjelmassa erikseen vaan se käydään pääohjelman kierron aikana läpi ilman erillistä kutsua 100 ms välein riippumatta ohjelmakierron vaiheesta. Turvaohjelman hätäseistoimintolohko on esitetty kuvassa 16. Erilaisia turvatoimilaitteita voi olla esim. hätäseis-painike, valoverho tai turvaportti. Näiden turvatoimilaitteiden tilatietoja tutkitaan hätäseis-lohkolla. Jos hätäseis-lohkon lähtötieto menee päälle, katkaisee se kyseisen kuljetinlohkon kontaktorin pois päältä.



Kuva 16. Hätäseis-lohko

Turvaohjelmat suunnitellaan alueittain, eli ohjelma voi sisältää alueellisesti useita hätäseis-lohkoja ja niiden lähtötilatiedoilla voidaan tallentaa apumuistiin muuttuja, jota käytetään kuljetinkohtaisen lohkon sisääntulotietona. Eli jos yksikin hätäseis-lohkon ulostulotieto muuttaa tilaansa saattaa se pysäyttää useamman kuljettimen kyseiseltä turva-alueelta. Turvaohjelman liittäminen kuljetinkohtaiseen ohjauslohkoon on esitetty kuvassa 17.



Kuva 17. Turvaohjelman liittäminen kuljettimen ohjauslohkoon

Tuorelajittelun kuljetinohjauksissa tarvittava tiedonsiirto kyseisessä tuotantolaitoksessa hoidetaan digitaalitietona. Lajitteluohjelmisto on toisen toimittajan tekemä. Siitä logiikasta saadaan lokeroissa olevan puutavaran tietoja. Lokeroita purkaessa pitää rimoitukseen saada tieto purettavista dimensioista ja määrästä. Dimensiosta riippuen kuljetinnopeudet säädetään tuotteen mukaisesti. Rimoituksessa tärkein tieto on määrä ja kappaleen leveys, jotta tiedetään oikea kerrosmäärä rimanipulle ja voidaan säätää ajonopeudet hissille ja

siirtovarsille. Rimanipun leveydeksi kyseisessä tuotantolaitoksessa on määritelty kaksi metriä. Siirtovarret siirtävät valmiin lautan keräyskuljettimelta pinontahissille. Lautan takareunan paikka on kiinteä, jolloin laudat tiputtavan etuvasteen on tiedettävä lautan leveys ja liikuttava takaisin siirtovarsien mukana määriteltyyn kahden metrin leveyteen. Pituus on myös ratkaiseva tieto rimakasettien käytön kannalta. Pituus on oltava tiedossa, ettei käytetä ylimääräisiä rimoja, jotka eivät osu rimanipun päälle. Pronorin toimittamassa ohjelmassa ei käsitellä muita pakettitietoja rimoituksen suhteen, jolloin datan siirto eri toimittajien logiikoiden kanssa on vähäistä. Kovien pakettien linjalla vanhasta tuorelajittelusta tulevien pakettien tiedot kuljetetaan pakettien mukana kuljettimelta toiselle. Pakettitiedot tulevat vanhalta logiikalta ja ne siirtyvät pakettien mukana. Pakettitiedoista tärkein on pituus, joka vaikuttaa paketin sidontaan. Lisäksi tarvitaan tieto, tuleeko paketti alle aluspuut vai ei.

Ohjelmarungon lisäksi työssä tarvitaan joillekin laitteille tarkkaa paikoitustietoa, esimerkiksi rimoituslaitteelle. Rimoituslaitteeseen kuuluu ladontavarret, jotka siirtävät valmiin lautan apuhissille. Apuhissin tarkoituksena on säästää aikaa sillä välin, kun päähissi laskee valmiin rimanipun alas. Tällöin ladontavarret voivat siirtää lauttoja keskeytymättömästi. Apuhissilla on omat kiinteät kuormavarret, joita liikutetaan kuormatilaan ja pois sieltä. Apuhissi on toiminnassa ainoastaan pinonnan alkuvaiheessa tiettyyn kerrosmäärään asti ja jää sen jälkeen odottamaan luovutuspisteeseen, kunnes päähissi vapautuu. Päähissi on joko odotuspisteessä kunnes apuhissille määritetty kerrosmäärä tulee täyteen tai suorittaa edellistä kuorman laskua loppuun. Tämän jälkeen päähissi nousee vähän apuhissin luovutuspistettä ylemmäs, jolloin kuorma siirtyy sen päälle. Tämän jälkeen apuhissi suorittaa nopean liikkeen alas kotiasemaansa samalla siirtäen kuormavarret pois pakettitilasta. Kun kuormavarret ovat siirtyneet pois pakettitilasta apuhissi palaa odotuspisteeseen ja päähissi jatkaa toimintaansa normaalisti, kunnes määritelty kerrosmäärä on saavutettu.

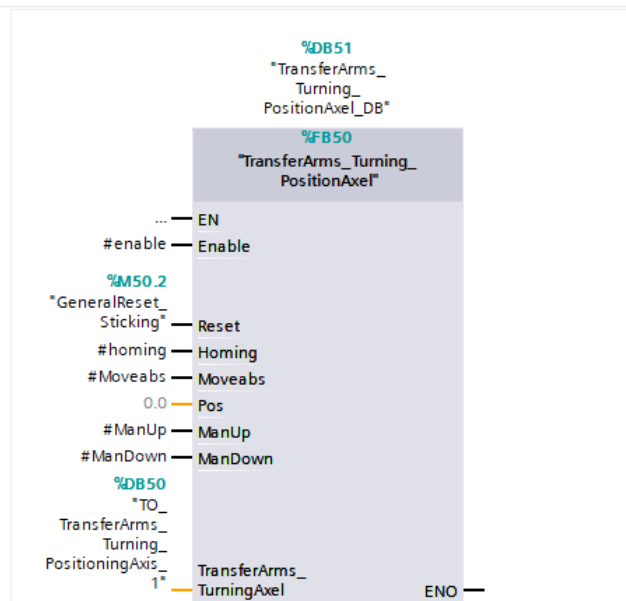
Pinontahissin päällä kulkee rimanjakelulaite, joka tarvitsee myös paikoitustietoa. Laitteeseen on yhdistetty rimoitusnipun paksummat rimat ja niiden jakelukasetit sekä kovan paketin ohuemmat välirimat. Laite pitää paikoittaa ajettavan pakettityypin mukaan. Paikoituksessa ja liikkeenohjauksessa vaaditaan tarkkaa paikkatietoa, joka toteutetaan absoluuttiantureilla. Absoluuttianturi kiinnitetään yleensä moottorin akseliin tai akseliin, johon moottorin liike välitetään. Absoluuttianturilla voidaan mitata sijaintia, kulmaa tai kierrosmäärää sen tuottamien pulssien avulla.

Projektissa käytettävät absoluuttianturit ovat monikierrosantureita, joilla voidaan mitata paikkaa myös kierroksen sisällä. Kyseiset anturit antavat 8192 pulssia yhden kierroksen aikana. Telegram 81 on Siemensin määrittelemä sanoma, jota käytetään anturin tiedonsiirtoon, tilatietojen lukuun ja ohjaustietoihin. Sanomia on erilaisia, ne eroavat tiedonmäärän

laajuudeltaan ja kohteen tarvitseman laajuuden mukaan. Telegram 81 sisältää 32-bitin tilatiedot. Paikoituksessa tärkein on sanoman neljäs tavu, josta saadaan anturin pulssilukema. Jos anturin antama pulssimäärä on olennainen operaattorin kannalta, voidaan pulssiluku muuntaa metriseen järjestelmään, jolloin sitä voidaan seurata käyttöliittymästä. Logiikkaohjelmassa pulssilukua käytetään paikoituskäytöissä. Määritellyt paikkapisteet selvitetään käyttöönötossa tai virtuaalisessa käyttöönötossa. Rimoituslaitteen kokonaisuuden muodostamat laitteet toimivat jatkuvasti ja nopealla syklillä. Liikkuvia osia on monta ja törmäysriski laitteiden välillä suuri ilman niiden tarkkaa paikoitusta ja paikkatietoa.

Absoluuttianturia voidaan hyödyntää liikkeenohjauksissa, joita ohjataan säätöpiirillä. Siirtovarsien ylöskäännön työliike tehdään hydraulisesti. Hydraulisyliinteri työntää akselia, joka nostaa varret ylös. Absoluuttianturi on sijoitettu akselin päähän. Hydraulisyliinterissä virtaavaa öljyn suuntaa ja määrää ohjataan proportionaali-venttiilillä. Venttiiliä ohjataan virta- tai jänniteviestillä yleensä $\pm 4\text{-}20$ mA tai $\pm 0\text{-}10$ V. Ohjausviesti on analogiamuodossa oleva word-muuttuja, joka on kokonaisluku. Jos venttiili halutaan täysin auki, sen ohjearvo määrätään kokonaisluvulla $\pm 27\ 000$, jonka analogiakortti muuttaa virta- tai jänniteviestiksi venttiilille. Säätöpiirin ohjaus voidaan toteuttaa ohjelmallisesti laskemalla haluttu kokonaisluku jokaiselle absoluuttianturin antamalle paikalle. Avautumis- ja sulkeutumiskäskeyn voidaan laskea paikan perusteella hidastusramppi, jolloin liikkeelle lähtö ja pysähtyminen on hillittyä. Varsinainen työliike suoritetaan tasaisella ohjauksella, esim. venttiili ohjataan 100% auki kokonaisluvulla +27000. Venttiili antaa logikalle takaisin tietoa onko se saavuttanut määrätyn oloarvon.

Siemens on kehittänyt säätöpiirin ohjaukseen laskentaa helpottavan toiminnon. Tätä toimintoa kutsutaan teknologiaobjektiksi. Teknologiaobjektiin linkitetään ohjattava analogialähtö ja absoluuttianturi. Teknologiaohko laskee lohkon sisällä absoluuttianturin pulssien ja venttiilin antaman analogiatiedon perusteella venttiilin suunnan ja nopeuden. Objektiin ei tarvitse antaa kuin haluttu paikka ja muutamia parametritietoja, kuten hidastusramppien aika. Objektiin määritellään miten työliike toteutetaan ja paikkatieto luetaan absoluuttianturilta. Teknologiaobjektia käyttämällä säätöpiiriohjauksessa ei tarvitse suorittaa erillistä laskentaa. Teknologiaobjekti suorittaa laskennan ja ohjaukset silloin, kun se on parametroitu oikein. Parametroinnin jälkeen objekti tarvitsee vain kohdeposition. Teknologiaobjektin kutsu on esitetty kuvassa 18.

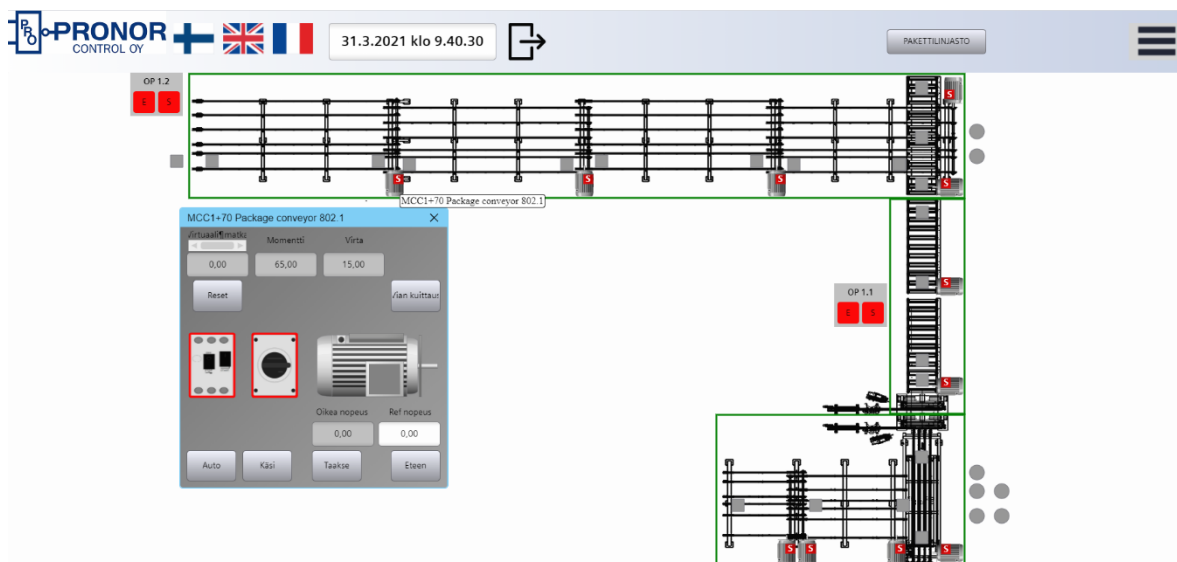


Kuva 18. Teknologiaobjektin toimintolohko

6 Käyttöliittymä

6.1 HMI

Käyttöliittymän eli HMI (HMI = Human-Machine Interface) on tarkoitus mahdollistaa vuorovaikutus automaattisten toimintojen ja tuotantolinjaa käyttävän operaattorin välillä. Tuotantolinjoja ohjaa PLC, mutta operaattori ei pääse siihen käsiksi. Käyttöliittymä voidaan esittää tietokoneen näytöltä tai erillisellä tarkoitukseen suunnitellulla näytöltä, joka voidaan integroida ohjauspulpettiin tai sijoittaa käytettävän laitteen lähelle. Käyttöliittymässä tehtävät valinnat kirjoittavat tiedon logiikkaan. Käyttöliittymässä tehtävät valinnat vaikuttavat PLC ohjelman, sellaisiin kohtiin, jotka ei ole laitteen toiminnan kannalta kriittisiä. HMI:stä vaihdettavia parametrejä ovat esim. kuljettimien nopeus. HMI:ssä voidaan myös tehdä valintoja, joka määrittelee useamman kuljettimen tai koko linjaston toimintaa, esim. linjanopeudet ja säädöt ajettavan dimension mukaan. Käyttöliittymässä voidaan havainnollistaa visuaalisesti tuotantolinjan toimintaa, sekä esittää tuotantoraportteja tai tilatietoja. Tuotantolinjan diagnostiikka ja hälytykset voidaan esittää HMI:n kautta. Alla olevassa kuvassa 19 esitetään paketoitilinjän näkymä käyttöliittymästä. Kuvassa on rajattu vihreällä turva-alueet. Moottorikohtaista lisäinformaatiota aukeaa erilliseen ikkunaan moottori-symbolia näytöltä painettaessa.



Kuva 19. Paketointilinja käyttöliittymässä

Linjaston operaattoreille on käytössä ohjelmaan valmiiksi tehdyt parametriarvot linjan asetusarvoille, kuten linjaston nopeusparametrit. Parametrit on valmiiksi määritelty tietokantalohkoon. Parametrien teko valmiiksi parametrilohkoon helpottaa käyttäjää. Käytettävät parametrit tulevat esiin operaattorille alavetovalikoista. Valikkoja käytettäessä ei tarvitse erikseen ladata ohjelmaa uudestaan logiikalle, kun jotain arvoa muutetaan. Käyttöliittymään on

tehty parametrien arvoille muokkausmahdollisuus. Muokkaamalla parametrejä ne voidaan tallentaa samaan tietokantalohkoon mistä alavetovalikot saavat arvonsa, jolloin käyttäjät voivat itse muokata arvoja. Esimerkiksi jotkut dimensiot vaativat hienosäätöä, jonka tarve voidaan todentaa vasta pidemmällä käyttöjaksolla. Hienosäätö ei tällöin vaadi ohjelman uudelleen lataamista logiikkaan.

6.2 WinCC

Siemensin TIA-portaaliin suunniteltu käyttöliittymä on nimeltään WinCC Unified. Käyttöliittymässä käytetään samoja osoitteita kuin pääohjelmassakin. Anturin havaitessa kappale anturin tilatieto vaihtuu. Tilatiedon vaihtuminen voidaan esittää käyttöliittymässä. Samassa ohjelmointiympäristössä käytettävä käyttöliittymä suoriutuu aikakriittisistä tehtävistä varmemmin verrattuna erillisessä laiteympäristössä toimivaan laitteeseen. Unified toimii serverikoneen kautta verkko-osoitteella.

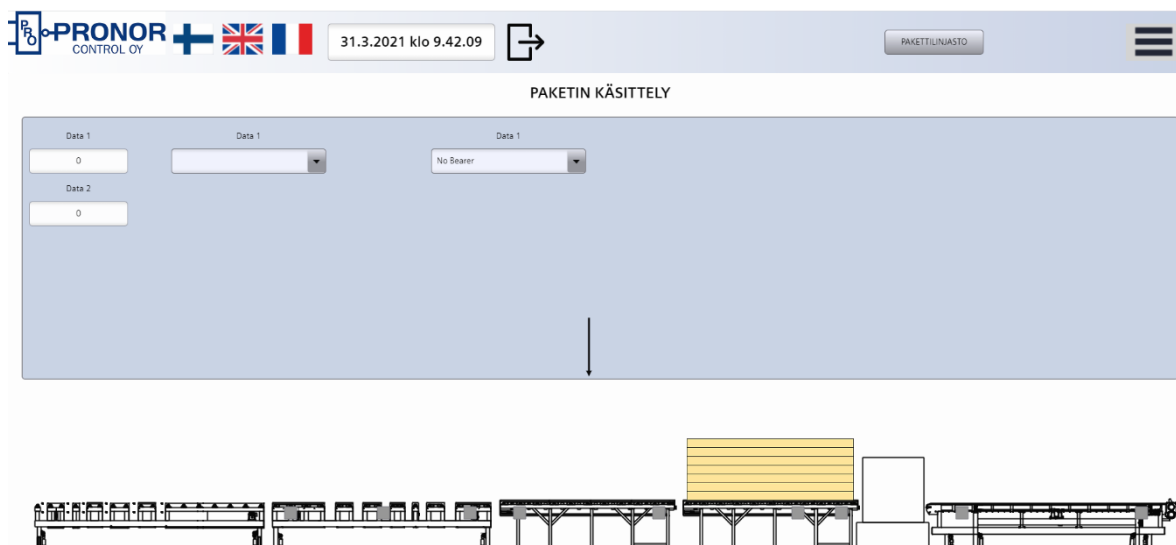
7 Testaus ja käyttöönotto

7.1 Ohjelman testaus

Työn aikana ei ollut käytettävissä moottoria ja taajuusmuuttajaa, joiden avulla voitaisiin todentaa multi-instanssin ja ohjelman toiminta fyysisillä laitteilla. Ohjelmassa käytettävä logiikka oli käytettävissä. Ohjelmaa voidaan testata lataamalla ohjelma logiikkaan ja monitoroimalla, sekä pakottamalla muuttujia päälle. Testauksessa testataan ohjelman tärkeimmät toiminnot eli käsiajot eteen ja taakse sekä automaattiajon toiminta sekä saako taajuusmuuttaja luvan toimia ja antaako se liikkumiskäskyn moottorille.

7.2 Käyttöliittymän testaus

Käyttöliittymän ja logiikan välisen rajapinnan testaus tapahtuu muuttujia pakottamalla. Kuvassa 20 on esitetty paketoitilinjän sidontakoneen osio, joissa on informaatio-ikkunoita, sekä alasvetovalikoita käyttäjälle. Kuvassa näkyvät dataosiot näyttävät paketin pituuksia sekä valinnan laitetaanko pakettiin välipuut vai tuleeko pakettiin pelkkä sidonta ilman välipuita. Kaikki vastaavat toiminnot testataan ennen käyttöönottoa eli toimivatko alasvetovalikot, moottoreiden parametrit ja näyttävätkö anturit tilatietoa.



Kuva 20. Paketoitilinjän parametrien valinta.

7.3 Virtuaalinen käyttöönotto

Projektissa testataan yhteistyössä Siemensin edustajien kanssa virtuaalista käyttöönottoa. Simuloitava laite on rimoituskone tuorelajittelun päässä. Rimoituskoneen mekaanisissa ratkaisuisissa käytetään uutta teknologiaa, joita ei ole testattu käytännössä. Rimoituskoneessa

on paljon liikkuvia osia ja niiden virtuaalinen testaus on äärimmäisen tärkeää sujuvan toiminnan ja törmäyksien välttämiseksi. Rimoituskoneella siirretään lautta kerrallaan pinoon, jonka jälkeen sen päälle lasketaan rimat ja hissi laskee kerroksen verran alaspäin. Yhden lauttakerroksen siirto ja rimoituskoneen muihin toimintoihin yhdellä kierrolla kuluva aika voi dimensiosta riippuen olla alle kolme sekuntia.

Virtuaalisen käyttöönoton tarkoituksena on välttää käyttöönoton viivästyminen ratkaisemalla ongelma virtuaalisesti. Tästä käytetään nimitystä virtuaalinen kaksonen. Testattava laite mallinnetaan ja määritetään sen liikeradat NX Mechatronics Concept Design ympäristössä. Simit-työkalulla simuloidaan kommunikointi toimilaitteiden välillä, eli luodaan virtuaalisesti tiedonsiirto verrattuna Profinet-väylään. Logiikalta tiedot kulkevat Simit-ohjelman kautta NX-ohjelmaan jossa liiketoiminnot tapahtuvat. (Siemens 2021c.)

7.4 Käyttöönotto

Käyttöönotossa testataan modernisoitavat laitteet mekaanisesti ja ohjelmallisesti. Jotta automaattiset laitteet saadaan toimimaan, on ensimmäiseksi tarkistettava PLC:n toiminta. Kuljettimien taajuusmuuttajat tarkistetaan sen jälkeen. Tarkistettavia asioita on kaapelointi, saako laite virtaa ja onko laitteiden välinen kommunikointi kunnossa. Taajuusmuuttajat tarvitsevat usein myös parametrissa säätöä käyttökohteesta riippuen. Tia portal-ohjelmointiympäristössä voidaan taajuusmuuttajiin ladata oletusparametriarvot, jos käytettävässä kohteessa on Siemensin taajuusmuuttajat. Muiden valmistajien taajuusmuuttajat pitää käydä yksitellen parametroimassa. Taajuusmuuttajien toiminta testataan suoraan taajuusmuuttajaa käsiajolla käyttämällä.

Käyttöönotossa tehdään myös I/O-testaus toimilaitteille. Kytkimistä tarkastetaan, että ne antavat oikeaa tilatietoa. Myös antureiden toiminta tarkistetaan. Tuotannon testaus suoritetaan alueittain yksittäisen kappaleen ajolla ja samalla tarkistetaan, toimiiko automaattiset sekvenssit ja käsiajo-valinnat. Kun kaikki alueet on käyty läpi, testataan koko linjan toiminta. Kappaletestauksella voidaan havaita joko mekaanisia tai ohjelmallisia vikoja, jotka voidaan korjata ennen tuotannon ylösajoa. Tuotannon ylösajossa ajonopeutta pyritään nostamaan tasolle, joka on modernisoinnin tavoitteissa määritetty.

8 Yhteenveto

Tarkoituksena oli suunnitella malliohjelmanrunko Pronor Controlille TIA portal-ohjelmointiympäristöön. Malliohjelmalla tuli voida soveltaa kuljetinohjauksissa tulevissa projekteissa. Aikaisemmat projektit ovat keskittyneet enemmän aiemman ohjelmaversioon ohjelmointiympäristöön ympäristöön. Malliohjelman tarkoituksena on vähentää suunnittelu-aikaa ja yhtenäistää ohjelman rakennetta.

Työssä perehdyttiin tuorelajitteluun ja sen vaatimiin kuljetinohjauksiin. Työssä käytettiin apuna aiemman ohjelmointiympäristön projekteja sekä alaan liittyvää kirjallisuutta. Malliohjelman soveltaminen käytännössä onnistui hyvin, vain muutamat kuljettimet vaativat lisäyksiä. Taaksepäin ajo automaattilla on sahalaitoksella harvinaista. Laskeva kuljetin, kuten hissin alla oleva kuljetin, voi liikkua molempiin suuntiin.

Toimintolohko, jossa suoritetaan suunnanvaihtoviiveet ja 0-nopeusohje on hyödynnetty aiemmista projekteista. Lohko on pääosin tehty käskylistalla, joka voi olla vaikea monitoroida. Sitä ei myöskään opetusohjelmassa opeteta, joten sen seuraaminen ja ymmärtäminen voi olla haasteellista. Tulevaisuuden kannalta lohkon muuttaminen toimintolohkokaavilla tehdyksi selkeyttää sen ymmärtämistä. Lisäksi tulevaisuudessa pitää jatkuvasti arvioida uusia projekteja tehdessä tarvitseeko malliohjelmanrunkoa muokata vastaamaan uusia tarpeita. Muutenkin tulee huomioida toistuvien kommentojen teko ja voiko niistä tehdä toimintolohkon ja tallentaa kirjastoon.

Lähteet

- Hansen, D. 2015. Programmable Logic Controllers : A Practical Approach to IEC 61131-3 Using CoDeSys. Chichester: John Wiley & Sons, Ltd. Viitattu 16.2.2021. Saatavissa <https://ebookcentral-proquest-com.ezproxy.saimia.fi/lib/lab-ebooks/detail.action?docID=4040112>
- Kivipuro, M. Malm, T. 2004. Turvallisuuteen liittyvät ohjausjärjestelmät konesovelluksissa. Tiedote. VTT. Viitattu 1.3.2021 Saatavissa <https://www.vttresearch.com/sites/default/files/pdf/tiedotteet/2004/T2264.pdf>
- Kovalainen, V 2021. VS: Tia portal opinnäytetyö. Sähköpostiviesti. Siemens. Vastaanottaja Sipilä, J. Lähetetty 16.4.2021.
- Nieminen, I 2019. Profinet. Profinet prosessiteollisuuden automaatio suunnittelussa. Opinnäytetyö. XAMK. Viitattu 1.3.2021. Saatavissa https://www.theseus.fi/bitstream/handle/10024/159772/Nieminen_Ismo.pdf?sequence=1&isAllowed=y
- Nurro, J. 2016. Teollinen Ethernet -ympäristö. Turvalogiikan käyttöönotto. Opinnäytetyö. Metropolia. Viitattu 1.3.2021. Saatavissa https://www.theseus.fi/bitstream/handle/10024/105915/Janne_N.pdf?sequence=1
- Perälä, Matti. 2013. Profibus. Profibus DP -harjoittelulaitteiston kehittäminen. Opinnäytetyö. Seinäjoen ammattikorkeakoulu. Viitattu 4.4.2021. Saatavissa <https://www.theseus.fi/handle/10024/57821>
- Puuinfo. 2020. Puutieto sahatavara ja sen jalosteet: Tukin sahaus. Viitattu 14.12.2020. Saatavissa <https://puuinfo.fi/puutieto/sahatavara-ja-sen-jalosteet/tukin-sahaus>
- Renholmen. 2021. Pystylokerikko. Viitattu 28.3.2021. Saatavissa <https://renholmen.se/tuorelajittelu/lokerikko/pystylokerikko/?lang=fi>
- Siemens. 2021a. Simatic throughout history. Viitattu 28.2.2021. Saatavissa <https://new.siemens.com/global/en/company/about/history/specials/60-years-of-simatic.html>
- Siemens. 2007b. How do you create and update multiple instances. FAQ. Viitattu 5.3.2021. Saatavissa <https://support.industry.siemens.com/cs/document/18723826/how-do-you-create-and-update-multiple-instances-?dti=0&lc=en-WW>
- Siemens. 2021c. Tehosta suunnittelua ja käyttöönottoa. Viitattu 4.4.2021. Saatavissa <https://new.siemens.com/fi/fi/tuotteet/teollisuus/virtuaalinen-kayttoonotto.html>

Valon Kone. 2021. Tyvensievennys kuorinnan yhteydessä. Viitattu 28.3.2021. Saatavissa <https://valonkone.com/fi/kuorinnan-osaaja/vk-teknologia/tyvensievennys>

Valutec. 2021. Kanavakuivaamot. Viitattu 28.3.2021. Saatavissa <https://www.valutec.fi/tuotteet/kanavakuivaamot/>

Varis, Ritva. 2017. Sahateollisuus. 2. painos. Palokka: Kirjakaari Oy.

Vänskä, Pentti. 2018. Maailman pisin, nopein ja tehokkain tukkilinja vauhdissa Keiteleellä. Maaseudun tulevaisuus. Viitattu 28.3.2021. Saatavissa <https://www.maaseuduntulevaisuus.fi/talous/artikkeli-1.220780>

Liite 1. Käyttöönotontarkistuslista

Pvm			
Tekijä			
Alue/Positio			
Projekti/Asiakas			
Projekti Nro			
	OK	EI OK	HUOM
Tarkista tuleeko logiikalle virta			
Tarkista tuleeko taajuusmuuttajalle virta			
Jos edellä mainitut ei ok, tarkista kytkennät			
Hajautusyksiköiden tarkistus, jos kaapelointi valmis			
Lataa ohjelma logiikkaan			
Taajuusmuuttajien parametointi			
Kuljettimien käsiajotesti taajuusmuuttajalta			
Ohjauspaneelien I/O testi			
Antureiden I/O testi			
Ajotestit ohjelmaa monitoroimalla			
Yksittäiskappaletesti			
Tuotannon ylösajo ja optimointi			
Muita huomioita:			
Tarkastanut			

Liite 2. Käyttöliittymän ja logiikan välisen rajapinnan tarkistuslista

Pvm			
Tekijä			
Alue/Positio			
Projekti/Asiakas			
Projekti Nro			
	OK	EI OK	HUOM
Tilatiedot anturit,moottorit jne.			
Kielenvaihto			
Hälytykset			
Visualisointi			
Logiikan laskennat			
Toimiiko ikkunointi			
Syötettävät arvot toimii			
Muita huomioita:			
Tarkastanut			