

LOGIIKKAOHJELMAN TESTAUKSEN KEHITTÄMINEN

Raute Oyj

LAHDEN AMMATTIKORKEAKOULU
Tekniikan ala
Kone- ja tuotantotekniikan koulutusohjelma
Opinnäytetyö
Kevät 2018
Ossi Taavila

Tiivistelmä

Tekijä(t) Taavila, Ossi	Julkaisun laji Opinnäytetyö, AMK Sivumäärä 26	Valmistumisaika Kevät 2018
Työn nimi Logiikkaohjelman testauksen kehittäminen Raute Oyj		
Tutkinto Kone- ja tuotantotekniikka		
Tiivistelmä <p>Opinnäytetyön tarkoituksena oli tehostaa Raute Oyj:n automaatio suunnittelua etsimällä eri keinoja logiikkaohjelman valmiusasteen toteamiseen. Tavoitteena on löytää logiikkaohjelmista virheet jo suunnitteluvaiheessa ja näin samalla lyhentää käyttöönottoon kuluva aikaa.</p> <p>Työssä perehdyttiin Siemensin TIA Portal -ohjelmointiympäristön ja simulointityökalujen ominaisuuksiin. Lisäksi kartoitettiin logiikkaohjelman testaukseen soveltuvia kaupallisia ohjelmistoja ja tutustuttiin Rauten viulun jatkoslinjan toimintaan.</p> <p>Lopputuloksena saatiin toteutettua I/O-osoitteiden vertailupohja Excelissä sekä kehitettyä useita ideoita TIA Portal Openness-rajapinnan ja Siemensin simulointityökalujen hyödyntämiseksi.</p>		
Asiasanat Logiikkaohjelma, Automaatio suunnittelu, Ohjelmointi		

Abstract

Author(s) Taavila, Ossi	Type of publication Bachelor's thesis	Published Spring 2018
	Number of pages 26	
Title of publication Improving testing of PLC-program Raute Oyj		
Name of Degree Mechanical and production engineering		
Abstract <p>The objective of the Bachelor's thesis was to enhance automation engineering in Raute Oyj by finding different methods for PLC-program testing. The goal is to find programming mistakes early in the planning stage of the project, which would lead to reduced workload in the commissioning stage.</p> <p>In this thesis we got acquainted with the features of Siemens TIA Portal-software and simulation tools. Also, supply of commercial PLC-program testing software and operational principle of Raute's veneer scarf-jointing line were on focus.</p> <p>As a result, various ways of detecting completeness of the PLC-program were found and implemented.</p>		
Keywords PLC-program, Automation engineering, Programming		

SISÄLLYS

LYHENTEET JA MÄÄRITELMÄT	1
1 JOHDANTO	2
1.1 Tausta ja tavoitteet	2
1.2 Työn rajaus ja toteutus	2
1.3 Raute Oyj	2
1.3.1 Liiketoiminta	3
1.3.2 Talous.....	3
2 VIILUN JATKOSLINJA.....	5
2.1 Viilun jatkaminen	5
2.2 Toimintaperiaate.....	6
2.3 Automaatio	7
2.3.1 Hajautettu I/O	7
2.3.2 Turvajärjestelmä.....	8
2.3.3 Käyttöliittymät	8
3 OHJELMAN VALMIUSTASON TOTEAMINEN TIA PORTALISSA.....	9
3.1 Havaitut virheet TIA Portal -projektissa	9
3.1.1 Excelin automatisointi VBA-ohjelmoinnilla	9
3.1.2 I/O-listojen vertailuohjelma.....	9
3.2 Hälytysbittien testaus käyttöliittymästä.....	12
3.3 TIA Portal Openness	13
3.3.1 Soveltuvuus ohjelman testaamiseen.....	14
4 SIMULOINTI	16
4.1 S7-PLCSIM V15	16
4.2 PLCSIM Advanced	17
4.2.1 Ominaisuudet	18
4.2.2 API	19
4.2.3 Ulkoinen ohjelma	20
5 PLC CHECKER	22
5.1 Yleistä.....	22
5.2 Ohjelmointisäännöstö	23
5.3 Soveltuvuus Rautelle	23
6 YHTEENVETO	24
LÄHTEET	25

LYHENTEET JA MÄÄRITELMÄT

CPU

Central processing unit eli suoritin tai prosessori. Käytetään usein synonyyminä ohjelmoitavalle logiikalle.

DATABLOKKI

Datalohko (DB), jota käytetään logiikkaohjelmoinnissa datan säilömiseen.

FUNKTIOBLOKKI

Logiikkaohjelmoinnissa käytettävä funktiolohko (FB), joka suorittaa siihen ohjelmoidun funktion ja säilyttää arvonsa datablokin muistissa.

HMI

Lyhenne sanoista Human-machine interface. Se tarkoittaa käyttöliittymää koneen käyttäjän ja koneen välillä.

I/O

Lyhenne sanoista input/output. Logiikkaohjelmoinnista puhuttaessa I/O tarkoittaa logiikan tuloja ja lähtöjä, jotka yhdistävät anturit ja toimilaitteet ohjaukseen.

LVL

Laminated lumber veneer eli viilupuu. Rakenteellinen puutuote, joka valmistetaan liimaamalla viiluja kerroksittain puun syyt yhdensuuntaisesti.

PLC

Programmable Logic Controller eli ohjelmoitava logiikka. Mikroprosessoripohjainen ohjain teollisuuden automaatioprosesseja varten.

TAGI

Logiikkaohjelman tuloille, lähtöille ja muistialuille ja muuttujille määriteltävä yksilöllinen nimi, joka toimii symbolisena osoitteena. Sen tarkoitus on selkeyttää ohjelmaa ja vähentää tarvetta käyttää absoluuttisia osoitteita ohjelmoinnissa.

TIA PORTAL

Siemensin ohjelmointialusta ohjelmoitavien logiikoiden, HMI-paneelien ja moottorikäyttöjen ohjelmointiin.

1 JOHDANTO

1.1 Tausta ja tavoitteet

Kiristyneet aikataulut ja entistä korkeammat laatuvaatimukset korostavat automaatio suunnittelun tehokkuuden merkitystä. Projektitoimitusten viivästyminen aiheuttaa merkittäviä lisäkustannuksia lisääntyneen suunnittelutyön lisäksi mahdollisten sopimussakkojen kautta. Automaatio suunnittelussa suunnittelutyön virheet ja keskeneräisyydet käyvät ilmi viimeistään projektitoimituksen käyttöönottovaiheessa, jolloin virheiden korjaukseen kuluva aika aiheuttaa merkittäviä lisäkustannuksia lisääntyneen työmäärän ja aikataulun venymisen johdosta.

Tämän opinnäytetyön tavoitteena oli tehostaa Raute Oyj:n automaatio suunnittelua etsimällä keinoja logiikkaohjelman valmiusasteen toteamiseen. Tutkivan työn lisäksi mahdollisimman moni idea pyrittiin toteuttamaan työn aikataulun puitteissa.

1.2 Työn rajaus ja toteutus

Logiikkaohjelman testaus ja käyttöönotto virtuaalisesti 3D-simulaation avulla olisi ideaalinen menetelmä suunnitteluvirheiden havaitsemiseksi. Koska Rauten toimittamat tuotantolinjat ontehty asiakkaiden tarpeiden mukaan, olisi hankalaa ja työlästä rakentaa jatkuvasti uusia linjan 3D-simulaatiomalleja. PLC-ohjelman testauksen näkökulmasta simulaatiomallin olisi myös oltava hyvin yksityiskohtainen ja tarkka, jotta siitä hyödyttäisiin riittävästi. Näiden syiden vuoksi 3D-simulaatiomallien käyttö rajattiin työn ulkopuolelle.

Työ toteutettiin tutkimalla Siemensin TIA Portal -ohjelmointiympäristön ominaisuuksia ja Siemensin simulointityökalujen soveltuvuutta ohjelman testaukseen. Lisäksi etsittiin mahdollisia testauskeinoja TIA Portal -ympäristön ulkopuolelta. Opinnäytetyössä otettiin tarkasteluun Rauten viulun jatkoslinjan logiikkaohjelma ja samalla perehdyttiin viulun valmistusprosessiin, linjan toimintaperiaatteeseen ja sen automaatiokomponentteihin.

1.3 Raute Oyj

Rauten historia alkoi jo vuonna 1908, kun kaksi vuotta aiemmin perustetusta konepajasta muodostettiin Lahden Rauta- ja Metalliteollisuus Oy. Alkuvuosinaan yritys valmisti rautasänkyjä, höyrykoneita ja jopa höyrylaivoja. Puuteollisuuden koneiden valmistus aloitettiin 1930-luvulla ja ensimmäiset tuotteet olivat sahoja, vanerikoneita ja tasohöyliä. (Lahden kaupungin museo 2018.) Sodan jälkeen sotakorvaustuotteiden valmistus loi hyvät suhteet Neuvostoliittoon ja samalla auttoi Rautea muuttumaan vientiyritykseksi. Ensimmäinen kokonainen puutuotetehdas toimitettiin vuonna 1968 Pelloksen

vaneritehtaalle Ristiinaan ja vuonna 1970 Raute toimitti koneet vaneritehtaalle Neuvostoliittoon, Bratskiin. (Raute Oyj 2018c.)

1980-luku oli yritykselle voimakasta kasvun aikaa erityisesti viennin osalta ja 1983 silloisen Lahden Rautateollisuus Oy:n viralliseksi nimeksi muutettiin Raute Oy, joka oli jo tullut tutuksi vanerikoneiden tuotemerkkinä. Vuonna 1994 Raute Oy:n A-osake noteerattiin Helsingin arvopaperipörssissä. (Raute Oyj 2018c.)

2000-luvulla Raute alkoi keskittymään puutuoteteknologiaan luopumalla punnitus- ja annostusteknologian alalla toimivasta Raute Precision-liiketoiminnasta. Samalla vahvistettiin tarjontaa ostamalla konenäkösovellutuksia valmistava Mecano Group Oy. Vuonna 2017 Raute osti sahatavaran ja viilun lujuuslajitteluteknologian markkinajohtajan Metriguard Inc:n liiketoiminnan. (Raute Oyj 2018d.)

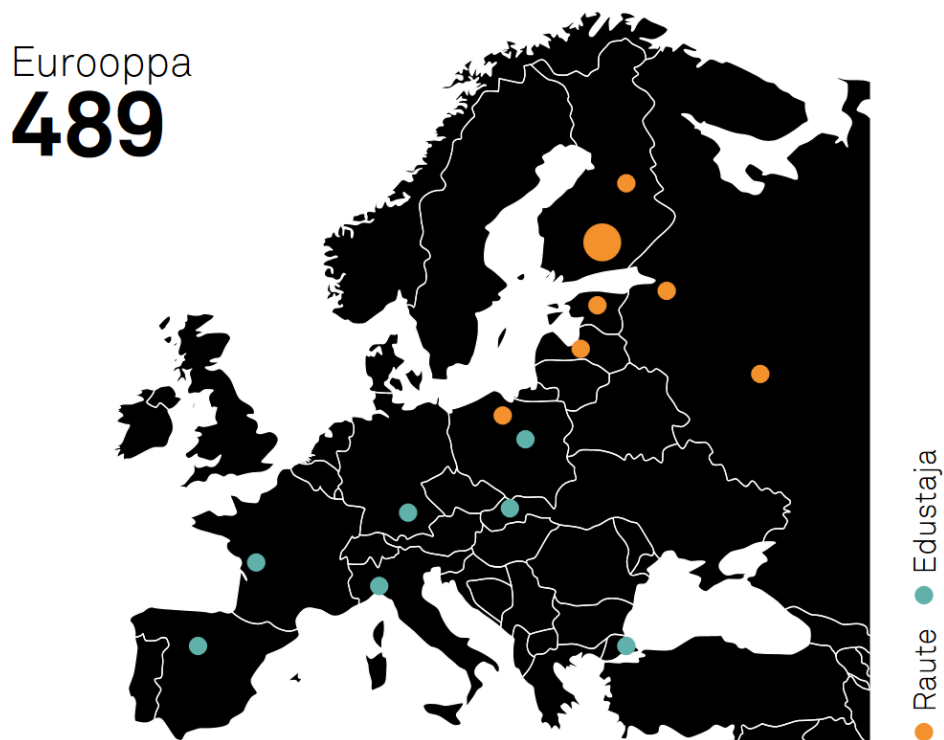
1.3.1 Liiketoiminta

Raute on puutuoteteollisuutta maailmanlaajuisesti palveleva yritys, joka tarjoaa asiakkailleen teknologiaa ja palveluita. Yritys toimittaa ympäri maailmaa toimiville asiakkailleen viilun, vanerin ja LVL:n valmistukseen tarvittavia tuotantoprosesseja ja palveluja. Raute on maailmanlaajuinen markkinajohtaja vaneriteollisuudessa 15 - 20 % markkinaosuudellaan ja LVL-teollisuudessa sillä on vielä suurempi asema, sillä Rauten koneilla tuotetaan yli puolet maailman LVL:stä. (Raute Oyj 2018a.)

Rauten toiminta koostuu projektitoimituksista ja teknologiapalveluista. Projektitoimitukset sisältävät kokonaiset tehtaat, tuotantolinjat sekä yksittäiset koneet. Teknologiatarjonta on monipuolinen perustuotantoteknologiasta aina pitkälle automatisoituun tuotantoon sisältäen konenäkösovellutuksia ja edistynyttä mittausteknologiaa. Tuotetarjonta kattaa asiakkaan koko tuotantoprosessin raaka-aineen käsittelystä valmiin tuotteen pakkaamiseen. Teknologiapalvelut sisältävät kunnossapidon, varaosapalvelut, modernisoinnit, koulutukset sekä erilaiset konsultointipalvelut asiakkaan liiketoiminnan kehittämiseksi. (Raute Oyj 2018a.)

1.3.2 Talous

Rauten liikevaihto vuonna 2017 oli 148,6 M€, josta yli 80 % tuli viennistä. Liiketulos samana vuonna oli 11,6 M€ ja henkilöstömäärä yhteensä 704, josta 67 % työskentelee Suomessa. (Raute Oyj 2018b, 3.) Kuvassa 1 on esitetty toimipaikkojen ja edustajien sijainnit sekä henkilöstömäärä Euroopassa.



Kuva 1. Rauten toimipaikat ja henkilöstömäärä Euroopassa (Raute Oyj 2018b, 3)

Rautella on tuotantoyksiköt Nastolan lisäksi Kajaanissa, Vancouverin alueella Kanadassa, Shanghaiin alueella Kiinassa ja Pullmanissa, Washingtonin osavaltiossa Yhdysvalloissa. Lisäksi useissa eri maissa on huolto- ja varaosapalveluita sekä myyntiedustajia (kuva 2). (Raute Oyj 2018b, 3.)



Kuva 2. Rauten toimipaikat ja henkilöstömäärä maailmanlaajuisesti (Raute Oyj 2018b, 3)

2 VIILUN JATKOSLINJA

2.1 Viilun jatkaminen

Viilu on vanerin perusosa, jota valmistetaan sorvaamalla puupölistä ohut levy. Sorvattu viilu leikataan, kuivataan ja lajitellaan jalostusta varten. Jalostus muodostuu monesta eri työvaiheesta, joista yleisimpiä ovat paikkaus, jatkaminen, saumaus ja päistään viillisen viilujen sahaus. (Koponen, 1995, 63.)

Viilun jatkamisen tarkoitus on liittää viilut toisiinsa pituussuunnassa, jolloin liitossauma on poikittain puun syhyyn nähden. Näin saadaan valmistettua yksittäistä sorvipölliä pidempiä viiluja, joita voidaan käyttää suurikokoisten vanerien valmistukseen. (Koponen, 1995, 66 – 67.)

Erikoiskokoisten vanerien kysyntä on lisääntynyt maailmanlaajuisesti ja siksi yrityksillä on entistä suurempi tarve investoida joustavaan vanerintuotantoon. Jatkettu viilu voidaan leikata käytännössä mihin pituuteen tahansa. (Raute Oyj 2018b.)

Jatkamisen työvaiheet ovat seuraavat:

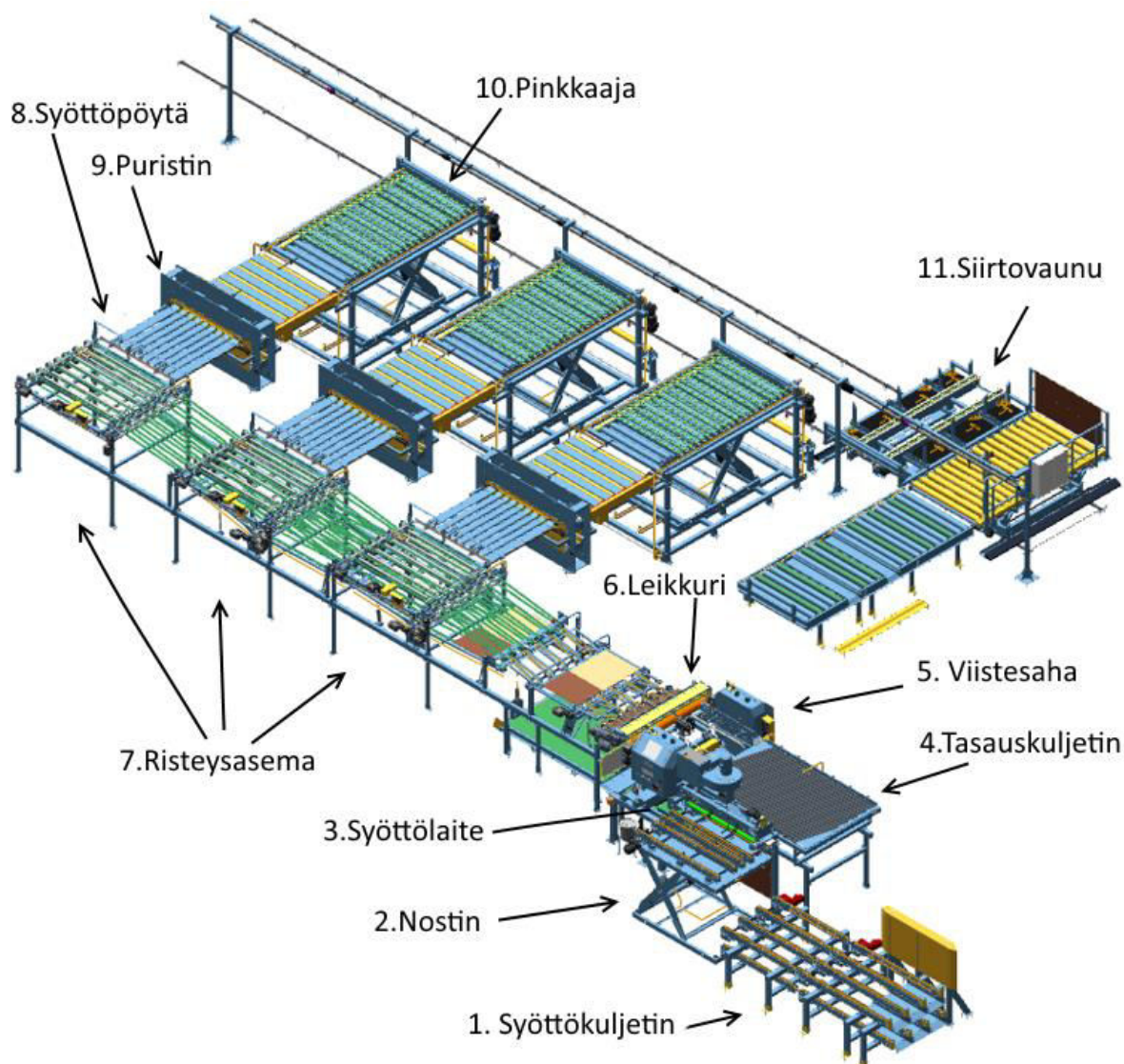
- Viilun päät sahataan suoraksi.
- Päät sahataan viistoon.
- Liima levitetään työstettyyn pintaan.
- Liitos puristetaan korkeassa lämpötilassa.
- Jatkettu viilu leikataan pituusmittaan. (Koponen 1995, 66-67.)

Jatketulle viilulle voidaan asettaa seuraavia vaatimuksia:

- Jatkoksen on oltava riittävän kestävä seuraaviin työvaiheisiin ja vanerin käyttöolosuhteisiin.
- Jatkoksen paksuusero ei saa olla liian suuri.
- Jatkoksen on oltava sileä eikä se saa olla palanut. (Koponen 1995, 66-67.)

2.2 Toimintaperiaate

Rauten toimittamien viilun jatkoslinjojen perustoimintaperiaate on aina samankaltainen. Viilupinkan syöttö- ja poistokuljettimien rakenne sekä puristimien määrä vaihtelee asiakkaan vaatimusten mukaan. Kuvassa 3 on 3D-malli eräästä Rauten toimittamasta viilun jatkoslinjasta.



Kuva 3. Viilun jatkoslinja (mukailtu Raute Oyj 2017)

Syöttökuljettimelle tuodaan trukilla viilupinkka, jonka kuljetin siirtää nostimelle. Viilupinkka nousee nostimella sopivalle korkeudelle, jolloin syöttölaite tarttuu päällimmäiseen viiluun alipaineella ja siirtää niitä yksi kerrallaan tasauskuljettimelle. Nostin nousee jatkuvasti sitä mukaan, kun pinkka madaltuu. Tasauskuljetin työntää viilun vasteseen kiinni ja kohti

viistesahaa. Viistesaha sahaa viilun päät ensin suoraksi ja sahaa sitten visteet. Ylöspäin osoittavaan viisteeseen levitetään liima, ja viilu siirtyy leikkurille, joka leikkaa etureunan suoraksi.

Hihnakuljetin siirtää viilun risteysasemaan, jossa se pysähtyy vasteeseen. Syöttöpöytä siirtää viilun viistetty reuna edellä kohti aikaisemmin syötetyn viilun viistettyä takareunaa, johon on levitetty liima. Päällekkäin olevat viistettyt reunat siirtyvät puristimeen, jossa liitos puristetaan kiinni korkeassa lämpötilassa. Puristimen ulostulon puolella on leikkuri, joka leikkaa standardipituisia viiluja. Pinkkaaja siirtää valmiin viilun pinkan päällimmäiseksi ja nostin laskee pinkkaa alas samalla, kun pinkan korkeus kasvaa. Täysi viilupinkka siirretään siirtovaunuun, joka vie pinkan kuljettimelle. Kuljetin siirtää valmiin viilupinkan odottamaan pois vientiä. (Raute Oyj 2017.)

2.3 Automaatio

Viilun jatkoslinjan ohjauksessa käytetään Siemensin S7-1500-sarjan ohjelmoitavia logiikoita. Ne ovat niin sanottuja turvalogiikoita, joissa perus- ja turvatoiminnot on integroitu samaan ohjaimeen mahdollistaen yhteisen ohjelmointiympäristön TIA Portalin Step 7 -suunnitteluohjelmistossa (Siemens AG 2018c). Siemensin turvalogiikat tunnistaa tyyppimerkinnän F-kirjaimesta.

Tiedonsiirtoon käytetään ethernetiin perustuvaa PROFINET-väylää. Turva-automaation tiedonsiirtoon PROFINET käyttää PROFIsafe-hybridiväyläteknikkaa. Se on ensimmäinen turvastandardin IEC 61508 mukainen tiedonsiirtostandardi, joka mahdollistaa perus- ja turvatiedonsiirron samassa väyläkaapelissa (Siemens AG 2008, 24).

2.3.1 Hajautettu I/O

Viilun jatkoslinjan automaatiojärjestelmän tulot ja lähdöt on nykyaikaisen koneautomaation tapaan hajautettu kentälle kauemmas PLC:stä. Tätä kutsutaan hajautetuksi I/O:ksi. Se tarkoittaa, että PLC kommunikoi PROFINET-väylän avulla kauempana kentällä sijaitsevien Siemensin ET 200 -sarjan hajautusasemien kanssa, joiden I/O-moduuleihin on kytketty tulo- ja lähtöpiirit. Hajautettu I/O säästää kaapelointikustannuksissa, mahdollistaa joustavan muunneltavuuden ja parantaa tiedonsiirron häiriönsietoa (Siemens AG 2018a). Turvapiireille käytetään erillisiä turvahyväksytyjä F-IO-moduuleita.

Osaa linjan toiminnoista ohjataan Siemensin ET 200SP CPU:lla, joka on hajautusyksikköön integroitu S7-1500-sarjan ohjelmoitava logiikka. Hajautetulla logiikalla saavutetaan parempi suorituskyky aikakriittisissä sovelluksissa, kuten liikkeenohjauksessa (Siemens AG 2016a, 24).

2.3.2 Turvajärjestelmä

Viilun jatkoslinjan turvajärjestelmä on jaettu useisiin turva-alueisiin. Ihmisen kulku turva-alueelle tunnistetaan porttien turvarajakytkimillä tai valoverhoilla, jolloin turva-alueen koneet pysähtyvät. Vaarallisimpien turva-alueiden ovissa käytetään sähköisiä turvalukkoja, jotka avautuvat koneiden ollessa turvallisessa tilassa. Kaikista ohjauspulpeteissa ja keskuksista löytyy hätäpysäytystä varten hätä-seis-painike. Toimitettavasta linjasta riippuen voidaan käyttää paineeseen perustuvia turvareunoja, joilla tunnistetaan mahdolliset törmäykset tai väliin jäävät varpaat.

Turvalogiikassa on perusohjelmasta erillään oleva turvaohjelma, joka sisältää keltaisesta väristä tunnistettavia turvablokkeja. Turvaohjelmoinnissa käytetään vain LAD- ja FBD-ohjelmointikieliä ja käytettävissä oleva käskykanta sekä datatyypit ovat tavallista suppeampia. Turvatoimintojen ohjelmointiin löytyy valmiita turvafunktiota, kuten hätä-seispiirin valvonta, samanaikaisuusvalvonta ja kontaktorivalvonta.

2.3.3 Käyttöliittymät

Linjan käyttäjälle pääasiallinen käyttöliittymä on kosketusnäytöllinen SIMATIC HMI-paneeli. Näytön etusivulta koneen käyttäjä näkee tietoja linjan tilasta sekä tietoja päivittäisestä tuotantomäärästä. Hälytyssivulle on listattu aktiiviset ja muistiin jääneet hälytykset.

Käyttöliittymästä linjan käyttäjä määrittelee viilun jatkamiseen liittyviä parametrejä, kuten syötettävän viilun mitat ja jatkettun viilun pituus. Tuotantoparametrit tallennetaan muistiin, jolloin voidaan nopeasti vaihtaa tuotettavan viiluarkin kokoa. Diagnostiikkasivut sisältävät tietoa linjan turvalaitteiden, moottoreiden ja anturien tilasta. Turva-alueen pysäyttänyt turvalaite tai vikatilaan mennyt moottorikäyttö paikallistetaan nopeasti linjan yleiskuvasta, josta näkyy laitteiden sijainnit.

Asetussivuilta määritellään parametrejä liittyen viilun syöttöön, sahaukseen ja jatkoksen puristukseen. Osa parametreistä on tarkoitettu vain koneen huoltoa varten ja niitä on mahdollista muokata kirjautumalla käyttöliittymään huoltotunnuksilla. Puristimille löytyy omat HMI-paneelit, joista löytyvät puristimien diagnostiikkatiedot sekä käsikäyttö huoltoa ja testausta varten. HMI:n ja PLC:n välisen kommunikoinnin rajapintana toimii kaksi datablokkia sekä yksi hälytyksiä varten tarkoitettu datablokki.

3 OHJELMAN VALMIUSTASON TOTEAMINEN TIA PORTALISSA

3.1 Havaitut virheet TIA Portal -projektissa

Logiikkaohjelmaan jää usein puutteita ja virheitä, jotka paljastuvat vasta käyttöönottilanteessa. Osa linjaan kuuluvista I/O:ista saattaa löytyä TIA Portalin PLC Tags -listasta, mutta niitä ei ole muistettu käyttää itse ohjelmassa. TIA Portalin cross-reference-työkalulla voidaan tagilistasta suodattaa tagit, joita ei käytetä tai käytetään ohjelmassa. Tarve oli kuitenkin päästä vertailemaan ohjelmassa käytettyjä I/O-tageja sähkösuunnittelijalta saatuun I/O-listaan.

Vertailu päätettiin toteuttaa tekemällä Exceliin automatisoitu pohja, joka hakee tiedot I/O-listasta sekä TIA Portalista saadusta listasta. Ongelmia tuotti TIA Portal V15:n ominaisuudet, sillä cross-reference-työkalun tekemää listaa ei vietyä ulos TIA Portalista eikä sitä voi myöskään kopioida. Tiedot saatiin kuitenkin ulos TIA Portalista tekemällä cross-reference-listasta tuloste PDF-tiedostoon, joka taas muunnettiin Excelille sopivaan tiedostomuotoon.

3.1.1 Excelin automatisointi VBA-ohjelmoinnilla

Microsoft Excelin toimintoja voidaan laajentaa ja automatisoida Visual Basic for Applications-ohjelmoinnin eli VBA:n avulla. Ohjelmointia voi tehdä yksinkertaisimmillaan nauhoittamalla VBA-makron, jossa Excelissä tehtävä toimenpiteiden sarja tallennetaan VBA-ohjelmakoodiksi. Kuitenkin täysi hyöty ohjelmoinnista saavutetaan kirjoittamalla ohjelmakoodia itse. (Taanila 2013, 1.)

VBA:n käytöstä Excelissä on suuri apu, kun suoritetaan paljon toistuvia toimenpiteitä jotka ovat automatisoitavissa. Sen avulla voidaan suorittaa hyvinkin monipuolisia tehtäviä, jotka perustuvat itse määriteltyihin ehtoihin ja samalla se laajentaa huomattavasti Excelin perustoimintoja. VBA:n avulla voidaan parantaa Excelin käytettävyyttä esimerkiksi tekemällä tietojen syöttämistä varten lomake tai antamalla käyttäjälle ohjeita erilaisilla ilmoituksilla. Myös kommunikointi muiden Microsoft Officen ohjelmien välillä on mahdollista. (Microsoft 2016).

3.1.2 I/O-listojen vertailuohjelma

VBA-ohjelmoinnin avulla luotiin Exceliin ohjelma, joka hakee automaattisesti I/O-osoitteet ja niiden nimikkeet käyttäjän valitsemasta kahdesta tiedostosta. Nimikket kerätään rinnakkaisiin listoihin ja osoitteita verrataan keskenään. Ohjelma merkkaa osoitteet, joille

löytyy toisesta listasta vastaava osoite. Näin logiikkaohjelmoija näkee, mitä I/O-listan osoitteita käytetään logiikkaohjelmassa ja mitä ei ole muistettu käyttää.

Vertailuohjelma haluttiin olevan käytettävyydeltään mahdollisimman yksinkertainen ja selkeä. Kuvassa 4 näkyy tekemäni pohja, johon vertailtavat I/O:t saadaan selkeästi rinnakkain siten, että vasemmalla on TIA Portalista saatu cross-reference –lista ja oikealla sähkösuunnittelijan tekemä IO-lista. Vasempaan reunaan lisättiin painikkeet vertailuohjelman toimintoja varten.

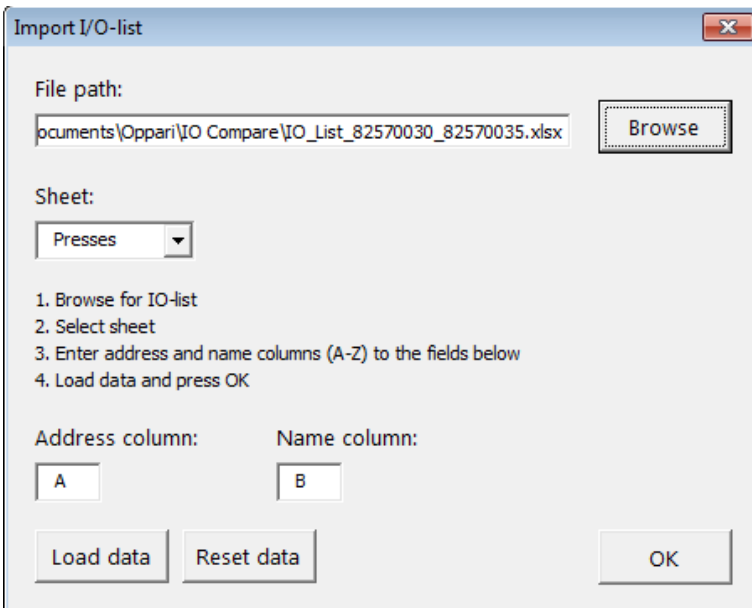
	Tag	Address	Address	Name
Tag table from TIA Portal	Anturi1	I0.0	I0.0	Dust remova
Cross-references2.xlsx	Anturi2	I0.1	I0.1	Vaste convey
Import file	Anturi3	I0.2	I0.2	Reserve
	Anturi4	I0.3	I0.3	Reserve
	Anturi5	I0.4	I0.4	Reserve
	Anturi6	I0.5	I0.5	Reserve
	Anturi7	I0.6	I0.6	Reserve
	Anturi8	I0.7	I0.7	Reserve
I/O-list	Anturi9	I1.0	Q0.0	Alarm horn
IO_List_82570030_82570035.xlsx	Anturi10	I1.1	Q0.1	Reserve
Import file	Anturi11	I1.2	Q0.2	Reserve
	Anturi12	I1.3	Q0.3	Reserve
	Anturi13	I1.4	Q0.4	Reserve
	Anturi14	I1.5	Q0.5	Reserve
	Anturi15	I1.6	Q0.6	Reserve
	Anturi16	I1.7	Q0.7	Reserve
	Anturi17	I2.0	I48.0	Basic positio
	Anturi18	I2.1	I48.1	Reserve
	Anturi19	I2.2	I48.2	Reserve

Kuva 4. Vertailuohjelma Excelissä

Import-painike suorittaa VBA-makron, johon on ohjelmoitu lomakkeen avaus. Lomakkeen browse-painikkeella pääsee valitsemaan tiedoston, josta haluaa tuoda IO-nimikkeet(kuva 5). Tiedostopolku näkyy tekstikentässä ja tiedoston nimi näkyy myös omassa solussaan Excel-pohjassa. Pudotusvalikkoon ilmestyy tiedoston sisältämät lehdet, ja samalla valitaan lehti, josta tuodaan I/O-nimikkeet. Usein I/O-listoissa on eritelty jokaisen CPU:n IO:t omille lehdilleen, mutta käytäntö vaihtelee sähkösuunnittelijan mukaan. Käytettävyyden parantamiseksi lomakkeeseen on tarkoitus lisätä vielä mahdollisuus tuoda tietoja useasta eri lehdestä samanaikaisesti.

Koska I/O-listojen rakenne vaihtelee paljon, täytyy käyttäjän määritellä itse, missä sarakkeissa osoitteet ja nimet sijaitsevat. Osoitteiden etsiminen automaattisesti olisi voinut onnistua mahdollisesti toistorakenteen ja merkkijonoja käsittelevien VBA-komentojen avulla. Kunkin I/O-osoitteen nimen etsiminen automaattisesti olisi kuitenkin mahdotonta, joten järkevintä oli tehdä osoite- ja nimisaraketta varten omat tekstikentät.

Load data-painikkeella ohjelma tuo tiedostosta I/O-osoitteet ja nimet Excel-pohjalle. Käytännössä VBA-koodi avaa määritellyn Excel-tiedoston taustalle, valitsee sarakkeet, kopioi tiedot, liittää ne vertailupohjaan ja sulkee tiedoston. Ohjelma poistaa samalla osoitesarakkeesta tyhjt solut ja osoitteista ylimääräiset merkit sekä välilyönnit. Käyttäjälle tietojen haku ei näy mitenkään, koska VBA-koodissa on ohjelmoitu hetkellisesti näytön virkistys pois päältä "Application.Screenupdating"-komennolla. Näytön vilkkumisen estämisen lisäksi komento parantaa koodin suoritusnopeutta merkittävästi.



Import I/O-list

File path:
Documents\Oppari\IO Compare\IO_List_82570030_82570035.xlsx

Sheet:
Presses

1. Browse for IO-list
2. Select sheet
3. Enter address and name columns (A-Z) to the fields below
4. Load data and press OK

Address column: Name column:

Kuva 5. Lomake I/O-listan tuontia varten

Lomakkeen ohjelmoinnissa oli otettava huomioon mahdolliset virhetilanteet, joita saattoi tulla esimerkiksi jos käyttäjä peruu tiedoston valinnan. Osoitepolun tekstikenttään estettiin kokonaan käsin kirjoittaminen, koska polku olisi todennäköisesti virheellinen. Sarakkeiden tekstikenttiin on mahdollista kirjoittaa vain kirjaimia väliltä A - Z ja load data-painike vapautuu vasta, kun kaikki lomakkeen kentät on täytetty. VBA-ohjelmassa käytetyt muuttujat säilyttävät arvonsa niin kauan, kun koodia suoritetaan. Jotta lomakkeen kenttien tiedot säilyisivät muistissa, ohjelma tallentaa tiedot Excel-pohjaan piilotetun lehden soluihin.

TIA-Portalin cross-reference -tiedoston tuontiin käytetään lähes samanlaista lomaketta. Erona on lähinnä, ettei siitä löydy mahdollisuutta lehden valitsemiseen. Se hakee PLC-tagit ja osoitteet jokaisesta lehdestä käyttäjän määrittelemistä sarakkeista. Koska TIA Portalin cross-reference-lista oli ensin tulostettu PDF-tiedostoksi ja sitten muunnettu Excelille sopivaan muotoon, siinä ei ollut selkeää muotoilua. Osoitteet ja tagit olivat kyllä

omissa riveissä ja sarakkeissaan, mutta seassa oli paljon ylimääräisiä soluja. Osoitteet saatiin suodatettua käyttämällä find-toimintoa, jolla voi etsiä tiettyjä merkkejä sisältäviä soluja. Siemensin STEP 7:ssä käytetään absoluuttisen osoitteen etuliitteenä %-merkkiä, jonka avulla osoitteet saatiin etsittyä helposti.

Ohjelma muotoilee solun taustan oranssiksi jos toisesta listasta löytyy vastaava osoite. Tämä onnistui käyttämällä conditional formatting-toiminnon ehtona COUNTIF-funktiota. Excel-pohjaan tehtiin myös mahdollisuus järjestää nimikkeet osumien perusteella tai arvon mukaan nousevaan järjestykseen. Pienen hankaluuden aiheutti osoitteiden formaatti, jossa ensin on muistialuetta merkitsevä kirjain ja perässä osoitteen numero. Excelin sort-toiminto ei osaa huomioida järjestämisessä kirjaimen ja numeron yhdistelmää, joten järjestämisen ajaksi ne täytyy erottaa vierekkäisiksi soluiksi. Osoitteiden järjestely tapahtuu välimuistina toimivassa piilotetussa Excel-lehdessä.

Muutamien ohjelmavirheiden korjaamisen jälkeen Excel-pohja osottautui käteväksi työkaluksi I/O-listan vertaamiseen TIA Portalin cross-reference-tietojen kanssa. Sillä voi myös verrata vaikka kaikkia logiikkaohjelman PLC-tageja I/O-listaan ja samalla silmämääräisesti tarkistaa, että tagit ovat nimetty oikein. Ohjelmaan voisi jatkokehittää mahdollisuuden korjata PLC-tagien nimiä, jolloin ne siirtyisivät TIA Portalista tuotuun Excel-tiedostoon. Tarkoituksena on vielä myöhemmin tehdä vielä toiminto, jolla kaikki HMI:n hälytystekstit saadaan päivitettyä kerralla.

3.2 Hälytysbittien testaus käyttöliittymästä

Käyttöliittymälle haluttiin lisätä mahdollisuus testata hälytyksiä asettamalla hälytysbittejä päälle yksi kerrallaan. Hälytysbittiä vastaava hälytysteksti voitaisiin tarkistaa HMI:ltä helposti.

Toiminto toteutettiin tekemällä logiikkaohjelmaan funktioblokki (kuva 6), joka kirjoittaa hälytysdatablokkiin hälytyksiä bitti kerrallaan. Ohjelmointikieleksi valittiin SCL, koska ylemmän tason ohjelmointikielenä se mahdollistaa toistorakenteiden käytön ja sisältää hyviä tiedonkäsittelyyn sopivia komentoja, kuten POKE. Sillä voidaan kirjoittaa suoraan halutulle muistialueelle määrittämällä vain osoitteen tavu ja bitti.


```

Network 8: GenerateAlarms
Comment
1      CALL "GenerateAlarms", "GenerateAlarms_DB"                %FB5, %DB12
2      DBW_Index := "97_HMIToPLC".Common.TrimAlarmDBWindex      %DB97.DBW4
3      Next_Bit  := "97_HMIToPLC".Common.TrigAlarmNext          %DB97.DBX0.5
4      Previous_Bit := "97_HMIToPLC".Common.TrigAlarmPrev       %DB97.DBX0.6
5      Free_roll  := "97_HMIToPLC".Common.TrigAlarmFreeRoll     %DB97.DBX0.7
6      Active_DBX := "99_PLCToHMI".Common.TrigAlarmActiveDBX    %DB99.DBX54
7
8
9
10

```

Kuva 6. Funktioblokin kutsu

Käyttöliittymän diagnostiikkavalikoon lisättiin oma sivu, jossa on lista aktiivisista hälytyksistä. Listan alla on syötekenttä, johon annetaan hälytysdatablokin haluttua wordia vastaava luku. Painonapeilla voidaan kirjoittaa bittejä yksi kerrallaan kumpaan suuntaan tahansa. Lisäksi tehtiin toiminto, jolla bitti siirtyy automaattisesti eteenpäin lyhyin aikaväleihin ja samalla aktiivinen bitti näkyy omassa kentässään (kuva 7).

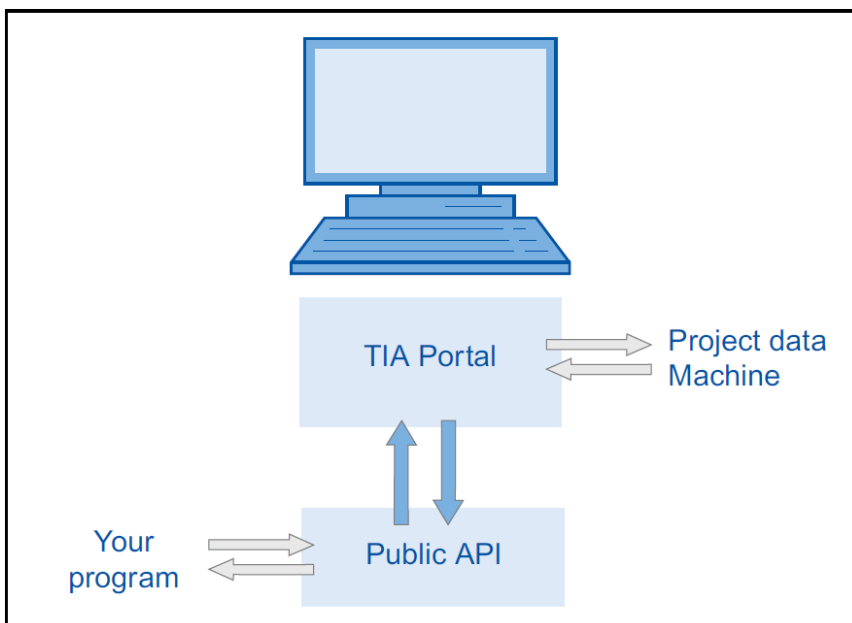


Kuva 7. Rajattu kuva käyttöliittymästä

3.3 TIA Portal Openness

TIA Portalista löytyy TIA Portal Openness ohjelmointirajapinta (API), joka mahdollistaa ulkoisten C#- tai Visual Basic .NET-kielellä ohjelmoitujen ohjelmien kommunikoinnin TIA Portalin kanssa. Rajapinnan avulla TIA Portal-projektin objekteja, kuten tekstejä ja ohjelmablokkeja, voidaan viedä ulkoiseen ohjelmaan XML-muodossa. Objektien siirto onnistuu myös päinvastoin ulkoisesta ohjelmasta TIA Portal-projektiin. Rajapinta mahdollistaa esimerkiksi automatisoidun projektien, hardwaren, HMI-tekstien ja ohjelmablokkien luonnin. (Siemens AG 2017b, 28-29.)

TIA Portal Openness on löytynyt TIA Portalin asennuspaketista vasta versiosta V14 SP1 eteenpäin joten siitä ei löydy juuri käyttökokemuksia ja se kehittyy jatkuvasti. Siemensillä on olemassa muutamia esimerkkiohjelmia, joilla esitellään API:lle mahdollisia käyttötapoja. Kuvassa 8 on esitetty API:n yksinkertaistettu toimintaperiaate.



Kuva 8. Openness API rajapintana TIA Portalin ja oman ohjelman välillä. (Siemens AG 2017b, 40.)

3.3.1 Soveltuvuus ohjelman testaamiseen

Openness-ohjelmointirajapinnan mahdollisia käyttötapoja Rauten ohjelmien testaamiseen pohdittiin selvittämällä sen ominaisuuksia. Rajapinta mahdollistaa ohjelmablokkien ja tagien tuonnin TIA Portalista XML-muodossa. Näitä tietoja voisi hyödyntää logiikkaohjelman laadun tarkistamista varten. Ohjelman laadun tarkistamiseen saatiin seuraavanlaisia ideoita:

1. Nimeämissäännöt

Tätä varten voisi kehittää ohjelman, joka tarkistaa, onko muuttujien nimeämiset, kommentoinnit ja ohjelmablokkien rakenteet Rauten ohjelmointistandardin mukaisia.

2. Kirjoittamattomat muuttujat

Funktioblokkiin voi joskus jäädä ohjelmointivirhe, että staattinen muuttuja luetaan jossain kohtaa ohjelmaa, mutta sitä ei kirjoiteta missään. Omalla testausohjelmalla voisi etsiä logiikkaohjelman funktioblokeista käyttämättömiä tai vain osittain käytettyjä muuttujia. Tämä onnistuisi helpoiten hyödyntämällä TIA Portalin ristiviittaustoimintoa Openness-rajapinnan avulla. TIA Portal Openness V15:ssä ei ollut vielä mahdollisuutta tuoda ristiviittaustietoja TIA Portalista. Toiminnon vielä puuttuessa saman voisi toteuttaa analysoimalla XML-muotoon käännettyä ohjelmablokkia.

3. HMI-tagit

PLC:sta HMI:lle menevä data kirjoitetaan muistiin datablokkiin, josta HMI lukee tiedot. HMI:llä jokaiselle luettavalle muistialueelle annetaan oma symbolinen osoite, eli HMI-tagit. Kuvissa 9 ja 10 näkyy, että datablokin ja HMI:n tagien nimeäminen on selkeyden vuoksi tehty samaan tyyliin.

16		Status_631M1	Word	844.0	W#16#1001
17		Status_632M1	Word	846.0	W#16#1001
18		Status_633M1	Word	848.0	W#16#1001

Kuva 9. Moottorin tilatietojen tagit logiikkaohjelman datablokissa

	Name ▲	Dat...	Connection	PLC tag	Address
	PLCToHMI\MotorStatus\Status_631M1	Word	CPU 1517F-3 DP	<Und...	%DB98.DBW844
	PLCToHMI\MotorStatus\Status_632M1	Word	CPU 1517F-3 DP	<Und...	%DB98.DBW846
	PLCToHMI\MotorStatus\Status_633M1	Word	CPU 1517F-3 DP	<Und...	%DB98.DBW848

Kuva 10. Moottorien tilatiedot HMI-tageissa

HMI-tagien määrittelyssä on mahdollista, että esimerkiksi moottorin 631M1 tilatieto haetaankin moottorin 632M1 osoitteesta, josta seuraa virheellinen moottorin tilatieto HMI:n diagnostiikkasivulla.

Oma testiohjelma voisi verrata HMI-tagin ja siihen kiinnitetyn datablokin tagin nimeämistä ja näin etsiä mahdollisesti väärästä osoitteesta haettuja HMI-tageja. Tagien nimet ovat lähestulkoon aina toisiaan vastaavat, joten nimivertailun toteuttaminen onnistuisi melko hyvin.

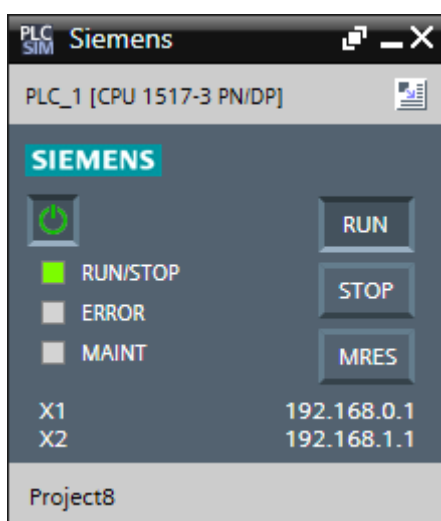
4. Hälytystekstit

HMI lukee hälytykset niille varatusta datablokista PLC:ltä. Jokaiselle hälytysbitille on häiriön tilaa vastaava hälytysteksti, jonka nimi on kirjoitettu muotoon ”(osoite)hälytysteksti”. Oma testiohjelma voisi etsiä poikkeavuuksia nimessä mainitun osoitteen ja todellisen hälytysosoitteen väliltä.

4 SIMULOINTI








4.1 S7-PLCSIM V15

SIMATIC STEP 7 Professional-ohjelmistoon sisältyy S7-PLCSIM, joka on tarkoitettu logiikkaohjelman testaukseen ja validointiin ilman fyysistä CPU:ta. Ohjelmaa käytetään TIA Portalin rinnalla ja sillä simuloidaan fyysistä logiikkaohjainta. Ohjelman online-seuranta ja lataus logiikkaan tapahtuu samalla tavalla kuin oikeassa logiikassa. Kuvassa 11 on ohjelma kompaktinäkyssä, jossa näkyy simulaatiologiikan tilatiedot ja painikkeet.



Kuva 11. PLCSIM kompaktinäkyssä

PLCSIM sisältää kompaktin näkymän lisäksi oman projektinäkymän, josta löytyy kaksi työkalua: Sim table ja Sequence editor. Sim tablen on taulukkonäkyssä voidaan monitoroida ja muokata tagien arvoja kuten TIA Portalin watch tablella. Sequence editorilla (kuva 12) voidaan tehdä aikaportaisiin perustuvia testisekvenssejä, joilla simuloidaan fyysisten tulosten vaikutusta ohjelman toimintaan. Testausta saadaan siis hieman automatisoitua, jolloin ohjelmoija voi samalla seurata ohjelman toimintaa TIA Portalissa tai Sim tablella. Testisekvenssin aloitukseen voi käyttää myös tagin arvoon perustuvaa liipaisuehtoa.

Sequence_1							
 Default interval <input type="text" value="50 ms"/>  <input type="text" value="hh:mm:ss:ms"/>  							
	Time	Name	Address	Display format	Action	Action parameter	Comment
	00:00:00.00				Start immediately		
	00:00:00.00	*Tag_1*:P	%I0.0:P	Bool	Set to value	TRUE	
	00:00:02.00	*Tag_2*:P	%I0.1:P	Bool	Set to value	FALSE	
	00:00:03.00	*Tag_3*:P	%I0.2:P	Bool	Set to value	TRUE	
	00:00:04.00	*Tag_4*	%M0.0	Bool	Set to value	TRUE	
	00:00:05.00			DEC	Set to value	0	
	00:00:05.05				Stop sequence		

Kuva 12. Projektinäkömän sequence editor

S7-PLCSIM:in soveltuvuutta viulun jatkamislinjan ohjelman testaamiseen kokeiltiin testaamalla ohjelmaa osissa. Testauksessa hankaluutta aiheutti logiikkaohjelman monimutkainen rakenne, koska usein yksittäisessä funktiolohkossa luetaan arvoja monesta eri instanssi-datablokista. Yksittäinen funktioblokki on siis helposti riippuvainen kymmenestä muusta funktioblokista.

Toinen ongelmia aiheuttanut asia oli viulun jatkamislinjan suuri logiikkaohjelma. PLCSIM:in projektinäkömä meni todella hitaaksi heti, kun simulaatiologiikkaan oli ladattuna koko ohjelma. Hitauden lisäksi ohjelma saattoi kaatua käytettäessä tiettyjä toimintoja tai vaihdettaessa ikkunaa TIA Portalista PLCSIM:iin.

Hitauden syitä selviteltiin paljon. Käytetyn tietokoneen suorituskyky ja muistin määrä oli reilusti yli Siemensin vaatimusten. Lopulta syyksi paljastui liian vanha versio virtuaalikoneesta, johon TIA Portal on asennettu.

PLCSIM V15 on kätevä työkalu ohjelman osien yksinkertaiseen testaukseen ja virheenetsintään, joskin sen ominaisuudet ovat turhan rajalliset.

4.2 PLCSIM Advanced

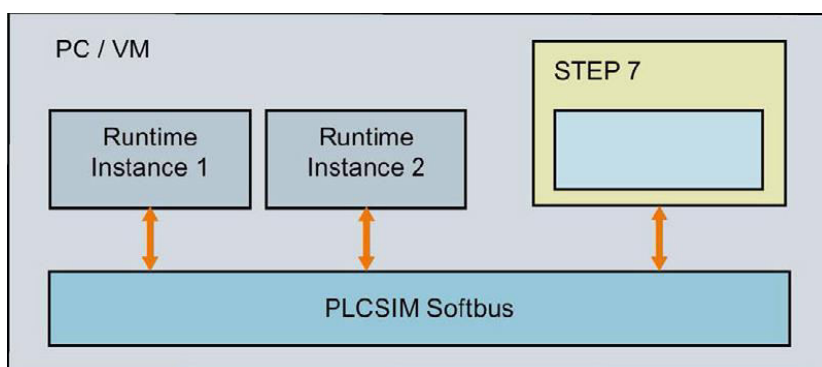
S7-PLCSIM:iä kehittyneempi simulointityökalu on PLCSIM Advanced, joka on TIA Portalin lisäosa. Sillä voidaan toteuttaa virtuaalista käyttöönottoa, jossa koneen toimintaa testataan jo suunnitteluvaiheessa. PLCSIM Advanced voidaan laittaa esimerkiksi ohjaamaan 3D-mallinnettua konetta, jolloin testaamiseen ei tarvita lainkaan fyysistä laitteistoa (Siemens AG 2017a, 30).

4.2.1 Ominaisuudet

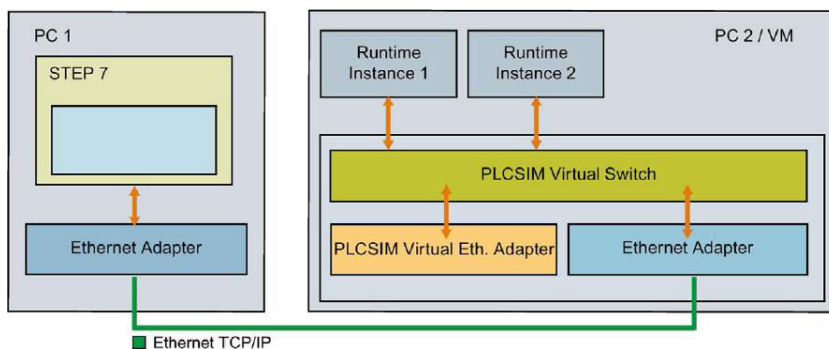
PLCSIM Advancedissa on tavallisen softbus-kommunikoinnin lisäksi virtuaalinen ethernet-sovitin. Se mahdollistaa hajautetun simuloinnin, jossa TIA Portal kommunikoi ethernetin välityksellä esimerkiksi toisessa tietokoneessa pyörivän simulaatiomallin kanssa.

PLCSIM-instansseja eli simuloitavia CPU:ita voi olla jopa samanaikaisesti jopa 16 ja ethernetillä voidaan käyttää myös fyysisiä CPU:ita ja HMI:itä osana simulointia. (Siemens AG 2017a, 51.)

Kuvassa 13 on esitetty paikallinen simulointi yhden PC:n sisällä. Kuvan 14 periaatekuvassa on hajautettu simulointi kahden PC:n välillä käyttäen ethernetiä.



Kuva 13. Tavallinen softbus-kommunikointi STEP7:n ja instanssien välillä (Siemens AG 2017a, 53.)



Kuva 14. Hajautettu ethernet-kommunikointi kahden PC:n välillä (Siemens AG 2017a, 54.)

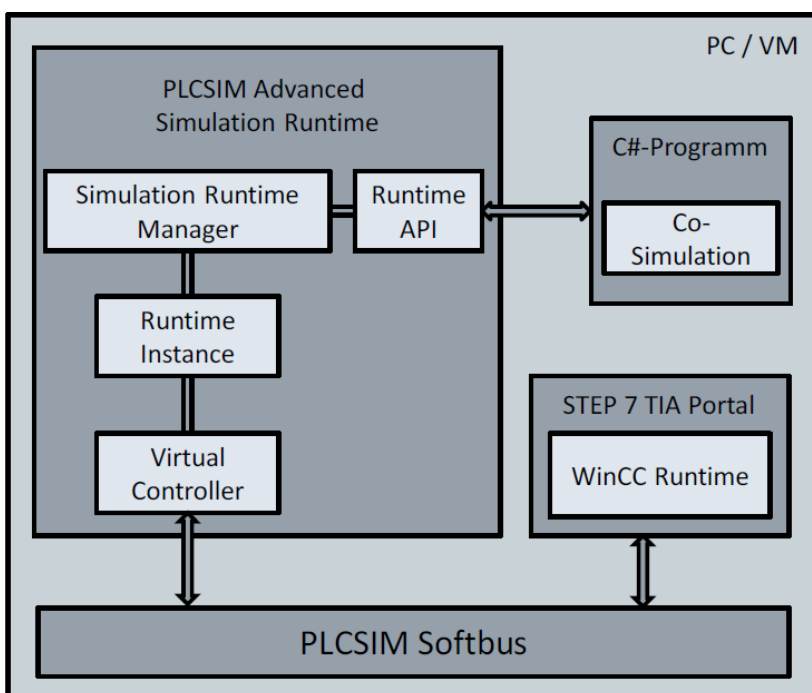
4.2.2 API

Tiedonsiirto simulaatiologiikan ja ulkoisten ohjelmien välillä on mahdollista PLCSIM Advancedin ohjelmointirajapinnan (API) avulla. Ulkoinen ohjelma voi olla esimerkiksi 3D-simulointiohjelma tai oma C# - tai C++ -kielinen ohjelma.

Ohjelmointirajapinnan tehtävät:

- PLCSIM-instanssin luominen
- virtuaali-CPU:n tilojen muuttaminen
- I/O-tietojen siirto virtuaalisen CPU:n ja ulkoisen ohjelman välillä (Siemens AG 2016b, 5)

Kuvan 15 periaatekaaviossa on esitetty ulkoisen C# -ohjelman kommunikointi PLCSIM Advancedin ja TIA Portalin kanssa.



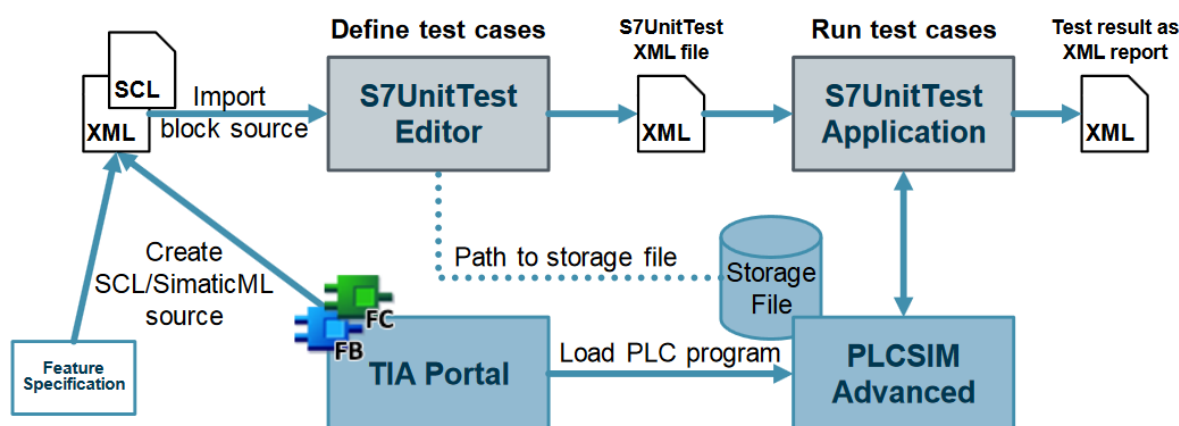
Kuva 15. PLCSIM ja ulkoinen ohjelma. (Siemens AG 2016b, 5.)

4.2.3 Ulkoinen ohjelma

Viilun jatkamislinjan ohjelman testausta kokeiltiin Siemensin S7UnitTest-työkalulla, jolla voi tehdä automatisoituja ohjelmatestejä logiikkaohjelman osille. Ohjelma on tarkoitettu esimerkiohjelmaksi, jota voi jatkokehittää omiin tarpeisiin.

S7UnitTestin muokkaustyökalulla tehdään tiedosto, jossa voidaan rakentaa erilaisia testejä, jotka sisältävät testiaskelia. Testiaskeleeseen määritellään PLC tagit, jotka halutaan kirjoittaa tai lukea yhden ohjelmasyklin aikana. Lisäksi tageille määritellään odotusarvot datatyypille ja lukuarvolle.

Testitiedosto ladataan ohjelmaan sisään, jolloin ohjelma käynnistää simulaatiologiikan ja suorittaa ohjelmoidun testin. Ohjelma tekee testistä raportin XML-tiedostoon. Kuvassa 16 on esitetty ohjelman toimintaperiaate.



Kuva 16. S7UnitTest-työkalun toimintaperiaate. (Siemens AG 2018b, 4.)

S7UnitTestiä kokeiltiin viilun jatkamislinjan logiikkaohjelman testaukseen lataamalla koko ohjelma simulaatiologiikkaan. Testaukseen valittiin yksinkertainen funktioblokki, johon rakennettiin ohjelmalla testiaskeleet.

Testissä havaittiin useita ongelmia S7UnitTestin ja simulaatiologiikan välisessä kommunikoinnissa; tagien lukemisessa ja kirjoittamisessa tapahtui useita virheitä ja ohjelman tulokset olivat epäloogisia. Virheiden aiheuttaja johtui keskeytyksellisestä ohjelmankäsittelystä, jolloin ohjelmablokkia ei suoriteta jokaisella ohjelmakierroilla. Siksi S7UnitTestiä käytettäessä testattavien ohjelmablokkien kutsu on oltava jatkuvassa ohjelmakierrossa.

S7UnitTestiä kokeiltiin vielä testaamalla jatkuvassa ohjelmakierrossa olevia blokkeja, jolloin testi toimi oikein. Työkalu soveltuu käytännössä yhden tai muutaman ohjelmablokin

samanaikaiseen testaukseen irrallaan koko logiikkaohjelmasta ja se on tarkoitettu lähinnä ohjelmablokkien standardoimiseen.

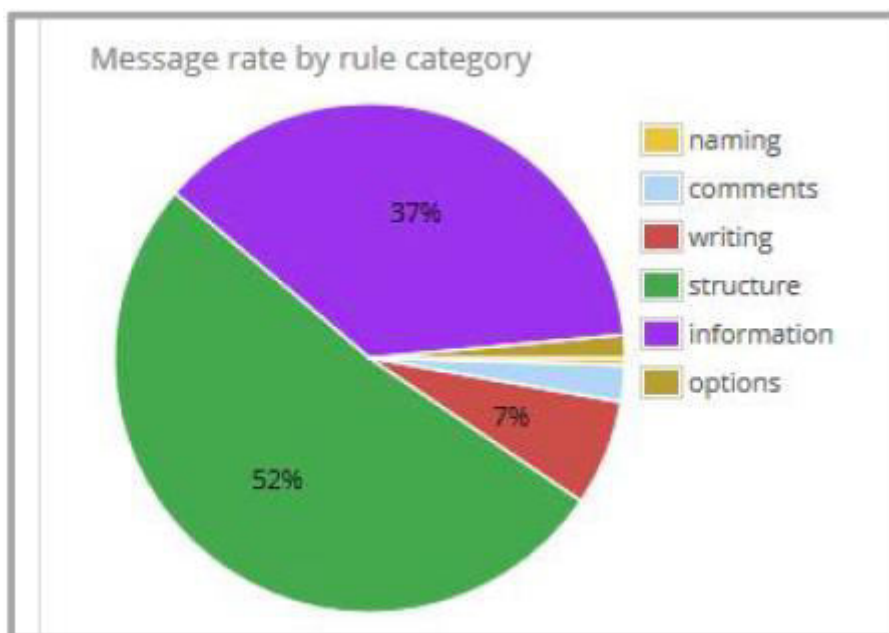
PLCSIM Advancedista ja ohjelmointirajapinnasta saisi parhaan hyödyn suunnittelemalla omaan käyttöön sopiva ohjelma. Yksi käyttötapa ohjelmalla voisi olla PLC:n ja HMI:n välisen kommunikoinnin testaaminen. Ohjelmalla voisi kirjoittaa datablokkiin turva-alueiden, anturien ja toimilaitteiden tilatietoja, jolloin simuloidun HMI-paneelin diagnostiikkasivulta näkisi mahdolliset tekstivirheet. Samalla tavalla voisi myös kirjoittaa hälytysbittejä hälytysdatablokkiin. Ohjelma olisi nopea ja helppokäyttöinen työkalu verrattuna siihen, että muuttaisi datablokin arvoja yksi kerrallaan TIA Portalissa.

5 PLC CHECKER

5.1 Yleistä

Yhtenä lisänä logiikkaohjelman testaukseen pohdittiin ranskalaisen Itris Automationin kehittämää PLC Checker-ohjelmaa. PLC Checker on logiikkaohjelman staattista analysointia varten kehitetty pilvipohjainen työkalu. Sen tarkoitus on varmistaa ohjelmakoodin laatu ennalta määriteltyjen ohjelmointisääntöjen avulla.

Ohjelma tukee Siemensin lisäksi tunnetuimpia PLC-valmistajia, kuten Rockwell ja Beckhoff ja kaikkia IEC 61131-standardissa määriteltyjä ohjelmointikieliä. Ohjelmakoodi käännetään Itris Automationin omalle ylemmän tason ohjelmointikielelle, GLIPS:ille. Ohjelmakoodi analysoidaan ja tuloksen perusteella nähdään, noudattaako ohjelma ohjelmointistandardeja ja samalla monet ohjelmavirheet pystytään havaitsemaan suunnitteluvaiheessa. Kuvassa 17 on osa ohjelman tekemän analyysin tuloksista. Tuloksista selviää jokaisen virheen yksilölliset tiedot sekä tilastotietoa analysoidusta ohjelmakoodista. (Itris Automation 2018.)



Kuva 17. Ohjelman tekemän analyysin tuloksia. (Itris Automation 2018.)

5.2 Ohjelmointisäännöstö

Ohjelmaan on mahdollista määritellä omia ohjelmointisääntöjä, joiden perusteella ohjelmasta etsitään virheitä. Oletuksena ohjelmointisäännöt noudattavat PLCopen-järjestön kehittämää 64:n ohjelmointisäännön kokoelmaa. Säännöt on jaettu seuraavanlaisiin kategorioihin:

- Nimeäminen
 - Varmistetaan, että kaikki ohjelman lohkot ja muuttujat noudattavat nimeämissääntöjä helpomman luettavuuden ja muokattavuuden saavuttamiseksi.
- Kommentointi
 - Jotta ohjelmakoodi olisi helposti ymmärrettävää, on ohjelman osat kommentoitava mahdollisimman tarkasti.
- Kirjoitus
 - Säännöillä varmistetaan, että muuttujia käytetään oikein ja oikeassa järjestyksessä
- Ohjelmarakenne
 - Havaitaan epäluotettavuutta aiheuttavat ohjelmarakenteet
- Yleishyödyllinen informaatio
 - Kerätään ohjelmakoodista yleistä tietoa
- Ohjelmointikieli
 - (Itrix Automation 2015.)

5.3 Soveltuvuus Rautelle

Ohjelmassa todettiin olevan hyviä sääntöjä, joista olisi apua ohjelmointitapojen yhtenäistämässä Rautella. Lisäksi PLC Checkeristä olisi apua joidenkin ohjelmointivirheiden havaitsemisessa. Tällaisia virheitä ovat esimerkiksi luettavan muuttujan kirjoittamatta jättäminen tai väliaikaisten testimuuttujien jättäminen ohjelmakoodiin.

6 YHTEENVETO

Tämän työn tavoitteena oli löytää keinoja logiikkaohjelman valmiusasteen toteamiseen Rauten automaatio suunnittelun tehostamiseksi. Opinnäytetyössä perehdyttiin viulun jatkamisprosessiin, Rauten viulun jatkoslinjaan ja ohjelman testausmenetelmiin. Työn tuloksena saatiin kartoitettua TIA Portalin ja Siemensin simulaatiotyökalujen ominaisuuksia sekä soveltuvuutta ohjelman testaukseen. Lisäksi kartoitettiin kaupallisten sovellusten tarjontaa ja soveltuvuutta Rauten käyttöön.

Työ aloitettiin tutkimalla viulun jatkoslinjan logiikkaohjelmaa TIA Portalissa ja tutustumalla Siemensin simulointityökalujen ominaisuuksiin. Alkuun kuului paljon tiedonkeruuta Rauten ja Siemensin teknisistä dokumenteista. Aikaa kului paljon myös teknisten ongelmien selvittelyyn simulaatiotyökalujen käytössä, jotka saatiin lopulta ratkaistua yhteistyössä Siemens Oy:n kanssa.

I/O-listojen vertailuohjelman tekeminen Excelissä onnistui mielestäni hyvin ottaen huomioon, että VBA-ohjelmointi oli minulle täysin uusi asia. Ohjelman tekeminen oli myös oppimisen kannalta hyvä ja mielenkiintoinen projekti.

Työn aihe oli haasteellinen ja logiikkaohjelman testaukseen liittyvät hankaluudet ovat yhä olemassa. Yhdeksi mielenkiintoiseksi ja varteenotettavaksi vaihtoehdoksi osoittautui TIA Portalin Openness-ohjelmointirajapinta, jota saatetaan hyödyntää tulevaisuudessa automaatio suunnittelun tehostamiseen.

LÄHTEET

Itris Automation 2015. PLC programs development guidelines [viitattu 31.3.2018].

Julkaisematon verkkodokumentti

Itris Automation 2018. PLC Checker [viitattu 31.3.2018]. Saatavissa: <http://www.itris-automation.com/plc-checker/>

Koponen, H. 1995. Puulevytuotanto. Helsinki: Opetushallitus.

Lahden kaupunginmuseo 2018. Lahtelainen metalliteollisuus ennen toista maailmansotaa [viitattu 15.3.2018]. Saatavissa: <http://www.lahdenmuseot.fi/kuka-mita-lahti/lahden-historia/teollisuuskaupunki/lahtelainen-metalliteollisuus-ennen-toista-maailmansotaa/>

Microsoft 2016. Getting Started with VBA in Office 2010 [Viitattu 9.4.2018]. Saatavissa: [https://msdn.microsoft.com/en-us/library/office/ee814735\(v=office.14\).aspx](https://msdn.microsoft.com/en-us/library/office/ee814735(v=office.14).aspx)

Raute Oyj 2017. Safety instructions and line overview, Scarf-jointing line [viitattu 24.3.2018]. Julkaisematon dokumentti.

Raute Oyj 2018a. Tietoa Rautesta [viitattu 15.3.2018]. Saatavissa: <http://www.raute.fi/fi/tietoa-rautesta>

Raute Oyj 2018b. Veneer scarf jointing [viitattu 23.3.2018]. Saatavissa: <http://www.raute.fi/fi/veneer-scarf-jointing>

Raute Oyj 2018c. Vuosikertomus 2011 [viitattu 15.3.2018]. Saatavissa: http://www.raute.fi/c/document_library/get_file?uuid=b85a85be-b9c9-4788-809b-2a7168eec862&groupId=24746

Raute Oyj 2018d. Vuosikertomus 2017 [viitattu 15.3.2018]. Saatavissa: http://www.raute.fi/documents/10157/882802/Raute+Oyj_Vuosikertomus+2017_painosta+01032018.pdf/b67087c6-fea5-479b-b2b5-35013f01c26c

Siemens AG 2008. PROFINET [viitattu 12.4.2018]. Saatavissa: http://www.siemens.fi/pool/products/industry/iadt_is/tuotteet/automaatiotekniikka/teollinen_tiedonsiirto/profinet/br_profinet.pdf

Siemens AG 2016a. SIMATIC ET 200SP CPU, The Controller for Distributed Intelligence [viitattu 12.4.2018]. Saatavissa: <https://www.industry.usa.siemens.com/automation/us/en/formsdocs/Documents/2016%20MIA-%2036%20SIMATIC%20ET200SP.pdf>

Siemens AG 2016b. SIMATIC S7-PLCSIM Advanced: Co-Simulation via API [viitattu 31.3.2018]. Saatavissa:

https://support.industry.siemens.com/cs/attachments/109739660/109739660_PLCSIM_Advanced_DOCU_V10_en.pdf

Siemens AG 2017a. Function Manual, PLCSIM Advanced [viitattu 26.3.2018]. Saatavissa:

https://cache.industry.siemens.com/dl/files/153/109739153/att_895955/v1/s7-plcsim_advanced_function_manual_en-US_en-US.pdf

Siemens AG 2017b. SIMATIC, Automating projects with scripts [viitattu 10.4.2018].

Saatavissa:

https://support.industry.siemens.com/cs/attachments/109477163/TIAPortalOpennessenUS_en-US.pdf?download=true

Siemens AG 2018a. Hajautettu I/O (ET 200) [viitattu 12.4.2018]. Saatavissa

http://www.siemens.fi/fi/industry/teollisuuden_tuotteet_ja_ratkaisut/tuotesivut/automaatiote_kniikka/hajautettu_io_et200.php

Siemens AG 2018b. S7UnitTest [viitattu 31.3.2018]. Saatavissa:

https://cache.industry.siemens.com/dl/files/405/109746405/att_919362/v2/109746405_S7_UnitTest_V10_en.pdf

Siemens AG 2018c. Turvalogiikat [viitattu 12.4.2018] Saatavissa:

http://www.siemens.fi/fi/industry/teollisuuden_tuotteet_ja_ratkaisut/tuotesivut/kone_ja_prosessiturvallisuus_seka_atex/koneturvallisuus/simatic_turva_automaatio/turvalogiikat.htm

Taanila, A. 2013. Excel-ohjelmointi [viitattu 8.4.2018]. Saatavissa: [http://myy.haaga-](http://myy.haaga-helia.fi/~taaak/vba/vba.pdf)

[helia.fi/~taaak/vba/vba.pdf](http://myy.haaga-helia.fi/~taaak/vba/vba.pdf)