

Sami Nyrhinen

**DIGITAALISEN
KÄYTTÄMINEN**

SUUNNITTELUTIEDON

UUDELLEEN

**Opinnäytetyö
CENTRIA-AMMATTIKORKEAKOULU
Tuotantotalouden koulutusohjelma
Helmikuu 2018**

TIIVISTELMÄ OPINNÄYTETYÖSTÄ

Yksikkö	Aika	Tekijä/tekijät
Ylivieska	2018	Sami Nyrhinen
Koulutusohjelma		
Tuotantotalous		
Työn nimi		
DIGITAALISEN SUUNNITTELUTIEDON UUELLEENKÄYTTÖ		
Työn ohjaaja	Sivumäärä	
Jari Kaarela, Sakari Pieskä	28	
Työelämäohjaaja		
Henri Kneckt		
<p>Tämän työn tarkoituksena oli selvittää, voidaanko digitaalista suunnittelutietoa hyödyntää automaatiolaitteen suunnittelun aikana eri ohjelmien välillä. Työn aikana suunniteltiin asiakkaalle testilaitteisto, jossa ohjelmoitava logiikka ohjaa prosessia. Työssä suunniteltiin logiikkaohjelma sekä piirikaaviot helpottamaan laitteiston valmistamista. Työn aikana kokeiltiin, kuinka digitaalisen suunnittelutiedon uudelleen käyttäminen onnistuu käytettyjen ohjelmien kesken. Työssä käytettiin yrityksen käytettävissä olevia ohjelmia eikä näin ollen oteta kantaa siihen, miten digitaalisen suunnittelutiedon uudelleenkäyttö toimii muiden ohjelmistojen kesken.</p> <p>Työn teoriaosiossa esitellään lyhyesti digitaalisen suunnittelutiedon uudelleenkäyttö, sähkösuunnittelun perusteet sekä logiikka ja sen ohjelmointikielet.</p>		

Asiasanat
Ohjelmoitava logiikka, sähkösuunnitelma, ohjelmointi, automaatio

CENTRIA UNIVERSITY OF	Date	Author
APPLIED SCIENCES	2018	Sami Nyrhinen
Ylivieska		
Degree programme		
Industrial Management		
Name of thesis		
RE-USE OF DIGITAL DESIGN MATERIAL		
Instructor		Pages
Jari Kaarela, Sakari Pieskä		28
Supervisor		
Henri Knecht		
<p>The purpose of this thesis was to find out whether it is possible to use the same digital design material in different software during electrical- and automation planning work. During this customer work I implemented a program for programmable logic and I did draw electrical design to help to manufacture this test equipment. I tried how round-trip engineering works between the used software. The Software that were used are already owned by the company, so the thesis does not discuss how the reuse works with other programs.</p> <p>The theoretical part briefly discusses work flow of round-trip engineering, electrical design and programmable logic.</p>		

Key words
Programmable logic, electrical design, programming, automation

SISÄLLYS

1 JOHDANTO.....	4
2 DIGITAALISEN SUUNNITTELUTIEDON UUELLEEN KÄYTTÖ	5
3 OHJELMOITAVA LOGIIKKA	6
3.1 OHJELMOINTIKIELET	7
3.1.1 FBD (Function block diagram).....	7
3.1.2 LD (Ladder diagram)	7
3.1.3 IL (Instrucion list).....	8
3.1.4 ST (Structured text)	8
3.1.5 SFC (Sequential function chart)	9
3.1.6 CFC (Continuous function chart).....	9
4 SÄHKÖSUUNNITTELU	10
4.1 Pääkaavio	10
4.2 Piirikaavio	10
4.3 Valitun ohjelman esittely	10
5 TYÖSSÄ KÄYTETTY LOGIIKKA JA SEN OHJELMOINTI	12
5.1 LOGIIKKA.....	12
5.1.1 CPU-yksikkö PM573-ETH.....	12
5.1.2 I/O-moduuli CD522.....	12
5.1.3 I/O moduuli DC523	13
5.1.4 I/O moduuli AO523.....	13
5.1.5 I/O-moduuli AI523.....	13
5.2 OHJELMOINTI.....	14
5.3 PROJEKTIN LUOMINEN	15
5.3.1 Logiikan konfigurointi.....	15
5.3.2 Ohjelmointiympäristö.....	17
6 DIGITAALISEN SUUNNITTELUTIEDON UUELLEEN KÄYTTÖ KÄYTÄNNÖSSÄ	19
6.1 I/O-tiedon siirtäminen logiikkaohjelmoinnista sähkösuunnitteluun.....	19
6.2 Suunnittelutiedon tuonti sähkösuunnitteluohjelmaan.....	22
6.3 Esimerkkitiedosto ja PLC I/O Utility.....	22
7 POHDINTA.....	26
LÄHTEET	27

KUVAT

KUVA 1. Function block diagram -esimerkki. (Codesys Online help 2017.).....	7
KUVA 2. Ladder diagram -esimerkki. (Codesys Online help 2017.).....	7
KUVA 3. Instruction list -esimerkki. (Codesys Online help 2017.).....	8
KUVA 4. Structured text -esimerkki. (Codesys Online help 2017.)	8
KUVA 5. Sequential function chart -esimerkki. (Codesys Online help 2017.).....	9
KUVA 6. Continuous function chart -esimerkki. (Codesys Online help 2017.)	9
KUVA 7: Logiikan keskusyksikkö. (ABB 2018a.)	12
KUVA 8: Projektinäkymä.....	15
KUVA 9: Moduulin konfigurointinäkymä	16
KUVA 10. Codesys-ohjelmointinäkymän resurssit-välilehti.	18
KUVA 11. Ohjelmalohkon lisääminen, tyyppin ja ohjelmointikielen valinta.....	18
KUVA 12. Asetussivu.....	19
KUVA 13. I/O-tiedon vienti	20
KUVA 14 CSV-tiedosto esimerkki.....	21
KUVA 15. PLC I/O drawings from spreadsheets. (AUTODESK 2018f.)	22
KUVA 16. Osa käytetyn Excel-taulukon sisällöstä	23
KUVA 17. Valitaan avattava tiedosto.....	24
KUVA 18. Sarakkeiden valinta	24
KUVA 19. Valmis, Excel-taulukosta muodostettu piirikaavio.....	25

1 JOHDANTO

Opinnäytetyö juontaa juurensa entiseen työpaikkaani ja asiakastyöhön, jonka yhteydessä tarjosimme kokonaistoimituksen testausjärjestelmän mekaniikka-, sähkö ja automaatio suunnittelusta sekä logiikkaohjelmoinnista. Samalla päätimme miettiä, kuinka voisimme hyödyntää digitaalista suunnittelutietoa siten, että suunnittelun aikana ei tarvitsisi syöttää samoja tietoja uudestaan ohjelmasta toiseen siirryttäessä. Työn aikana tarvitaan mm. logiikan I/O-tietoja logiikkaohjelmoinnissa sekä sähkö- ja automaatio suunnittelussa, kun piirretään kytkentäkuvia.

Digitaalisen suunnittelutiedon siirtämisellä ohjelmasta toiseen pyritään estämään kirjoitus- ja ajatusvirheiden syntyminen ja toisaalta nopeuttamaan suunnittelutyötä. Työvaiheet tein mekaniikkasuunnittelua lukuun ottamatta itse, joten opinnäytetyössä ei oteta kantaa ryhmätyöskentelyyn ja sen asettamiin vaatimuksiin. Pääpaino opinnäytetyössä oli kokeilla sitä, kuinka digitaalinen suunnittelutieto saadaan siirrettyä ohjelmasta toiseen ja onko siitä merkittävää hyötyä niillä ohjelmilla, jotka olivat, sillä hetkellä yrityksen käytettävissä.

Automaatio- ja sähkösuunnittelu toteutettiin Autodesk Electrical 2016-ohjelmistolla, jonka ohjelmistolisenssiä voitiin lainata yrityksen emoyhtiön lisenssipankista. Logiikkaohjelmointi suoritettiin käyttäen logiikkatoimittajan omaa ohjelmaa, ABB Automation builder 1.2.

Opinnäytetyön lähteinä käytettiin pääsääntöisesti ohjelmistovalmistajien verkkosivuja. Opinnäytetyössä ei tulla esittelemään asiakastyötä kokonaisuudessaan vaan keskitytään digitaalisen suunnittelutiedon uudelleen käyttöön siltä osin kuin se tässä työssä toteutui.

2 DIGITAALISEN SUUNNITTELUTIEDON UUELLEEN KÄYTTÖ

Digitaalinen suunnittelutieto ja sen siirtäminen ohjelmien välillä (Round-trip engineering) tarkoittaa yleisesti ottaen sitä, että yhdistetään useiden eri työkalujen tuottama aineisto yhteiseksi kokonaisuudeksi, jotta vältytään saman asian uudelleen tekemiseltä. Pitkälle vietyä digitaalisen suunnittelutiedon uudelleen käyttö on parhaimmillaan ryhmätyöskentelyssä, jossa samaan projektiin osallistuu useita henkilöitä. Vielä pidemmälle vietyä tämä tarkoittaa sitä, että useat eri yritykset voivat liittyä samaan, yhteen isoon projektiin, jakaen digitaalista suunnitteluaineistoa reaaliajassa. Näin voidaan varmistua siitä, että kaikilla on aina oikea tieto käytettävissä eikä esim. kirjoitus- tai ajatusvirheitä pääse syntymään. Laajoissa, eri yritysten muodostamissa isoissa projekteissa, digitaalisen suunnittelutiedon jakaminen kaikkien kesken ei yksin riitä. Suurissa projekteissa myös projektin johto ja tiedottaminen ovat tärkeässä roolissa.

Tämän työn perimmäinen tarkoitus oli tutustua siihen, että voidaanko pienen yrityksen sisällä siirtää kertaalleen tehtyä suunnittelutietoa ohjelmien välillä. Tämän työn yhteydessä kokeiltiin, että voiko logiikkaohjelmoinnissa saatua tietoa siirtää ja käyttää suoraan hyödyksi sähkösuunnittelussa, niillä työkaluilla, joita yrityksessä on käytettävissä.

3 OHJELMOITAVA LOGIIKKA

Ohjelmoitava logiikka (Programmable Logic Controller) eli PLC tai logiikka on pieni tietokone, jota käytetään tosiaikaisten automaatioprosessien ohjauksessa, kuten esimerkiksi NC-koneen tai tehtaan kokoamislinjan ohjaamisessa. Ohjelmoitava logiikka pitää sisällään yhden tai useamman mikroprosessorin.

Ohjelmoitavat logiikat otettiin käyttöön alun perin autoteollisuudessa, missä ohjelmistopäivitykset korvasivat ohjausjärjestelmien uudelleenjohtotukset. Ohjelmoitavan logiikan toiminnallisuus on vähitellen kasvanut perinteisestä releiden korvaajasta ohjauskeskukseksi, joka hallitsee kehittyneen liikkeen ohjauksen, prosessin säädön, hajautetut hallintajärjestelmä ja tietoverkot.

Erilaisten tulo-, lähtö- ja väyläyksiköiden kautta logiikat voidaan liittää erityyppisiin ja -suuruisiin prosessisignaaleihin. Signaalit ovat pääsääntöisesti binäärisiä kaksitilaohjauksia, mutta myös analogisia standardiviestejä ja pulssijonoja käytetään. Viime vuosina erilaiset kenttäväyläratkaisut ovat lisänneet sarjaliikenteen käyttöä eri laitteiden viestinnän sekä hajautetun I/O:n parissa. Ohjelmoitavan logiikan idea on siis toteuttaa ohjaukset sen sovellusohjelmaan sekä prosessitietoihin perustuen. Tätä varten tarvitaan tuloliitännät prosessitietojen lukemiseen, lähtöliitännät prosessin ohjauksiin sekä muisteja ohjelmaa ja muuttujia varten. Muistit jaotellaankin karkeasti ohjelmamuistiin ja muuttujamuisteihin. Sovellusohjelma käyttää muuttujamuistipaikkoja arvojen lukemista ja kirjoittamista varten.

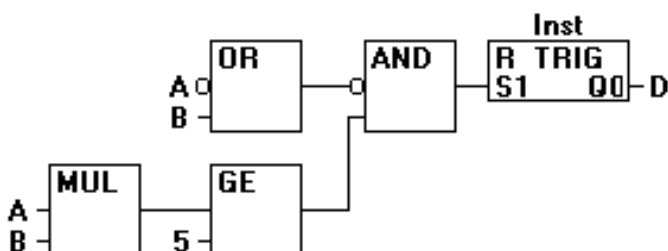
Nykyiset ohjelmoitavat logiikat toimivat syklisellä ohjelmankäsittelyperiaatteella, jossa ohjelmakierto eli sykli koostuu tietyistä rutiineista. Tyypillisesti yhden syklin aikana tutkitaan suorittimen eli CPU:n ja oheislaitteiden tilaa, suoritetaan sovellusohjelmaa rivi riviltä sekä päivitetään tulot ja lähdöt. Syklistä voidaan poiketa mm. keskeytystapahtumilla, jolloin kesken suoritettavan ohjelman väliin tulee tärkeämpi suoritettava ohjelmalohko. Näitä voivat olla esimerkiksi erilaiset hätätilanteet kuten hätäseis-napin painaminen. (Omron CX-One logiikkakoulutusmateriaali 2016).

3.1 OHJELMOINTIKIELET

ABB Automation builder pohjautuu Codesys-ohjelmointityökaluun. Codesys on IEC 61131-3 -standardin mukainen ohjelmointityökalu, joka tukee kuutta eri IEC 61131-3 -ohjelmointikieltä. Näitä ohjelmointikieliä ovat FBD (Function block diagram), LD (Ladder diagram), IL (Instruction list), ST (Structured text), SFC (Sequential function chart) ja CFC Continuous function chart.

3.1.1 FBD (Function block diagram)

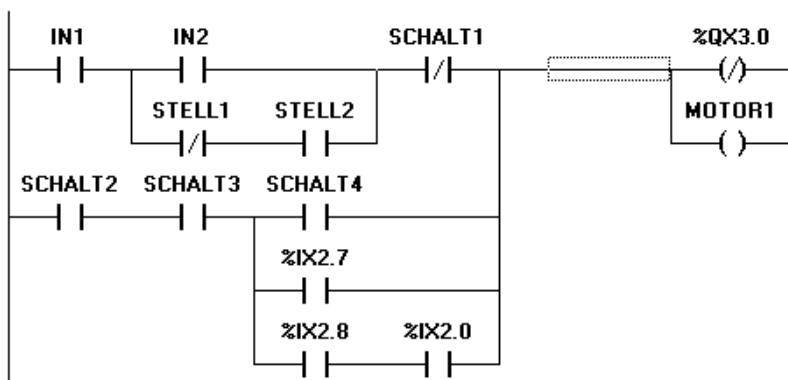
Function block diagram on graafinen ohjelmointikieli, jossa ”koodi” muodostuu laatikoista ja niiden väliin piirretyistä viivoista. Valmiit laatikot tarjoavat erilaisia toimintoja kuten vertailua, AND- ja OR-toimintoja. Koodi suoritetaan vasemmalta oikealle ja sitä on mahdollista jakaa omiin koodilohkoihin (network) vaakaviivoilla.



KUVA 1. Function block diagram -esimerkki. (Codesys Online help 2017.)

3.1.2 LD (Ladder diagram)

Ladder diagram on myös graafinen ohjelmointikieli, joka kuitenkin poikkeaa Function block -diagrammin esitystavasta. Ladder diagram koostuu vasemman ja oikean laidan pystyviivoista sekä niiden välissä olevista keloista ja kontakteista. Kontakteilla tarkoitetaan esimerkiksi tuloja ja kelalla lähtöä.



KUVA 2. Ladder diagram -esimerkki. (Codesys Online help 2017.)

3.1.3 IL (Instrucion list)

Instruction list on tekstipohjainen ohjelmointikieli, joka koostuu yksinkertaisista komennoista. Tämä ohjelmointikieli muistuttaa tietokoneen ohjelmoinnissa käytettyä konekielistä ohjelmointia (Assembly).

```
LD 17
ST lint (* Kommentar *)
GE 5
JMPC next
LD idword
EQ istruct.sdword
STN test
next:
```

KUVA 3. Instruction list -esimerkki. (Codesys Online help 2017.)

3.1.4 ST (Structured text)

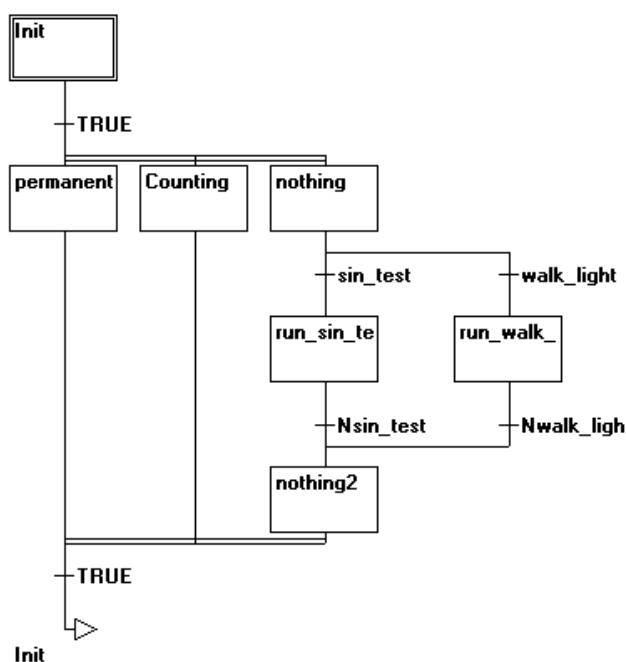
Structured text on myös tekstipohjainen ohjelmointikieli, joka muistuttaa ulkoasultaan perinteistä ohjelmointia (esim. C-kieli).

```
IF value < 7 THEN
  WHILE value < 8 DO
    value:=value+1;
  END_WHILE;
END_IF;
```

KUVA 4. Structured text -esimerkki. (Codesys Online help 2017.)

3.1.5 SFC (Sequential function chart)

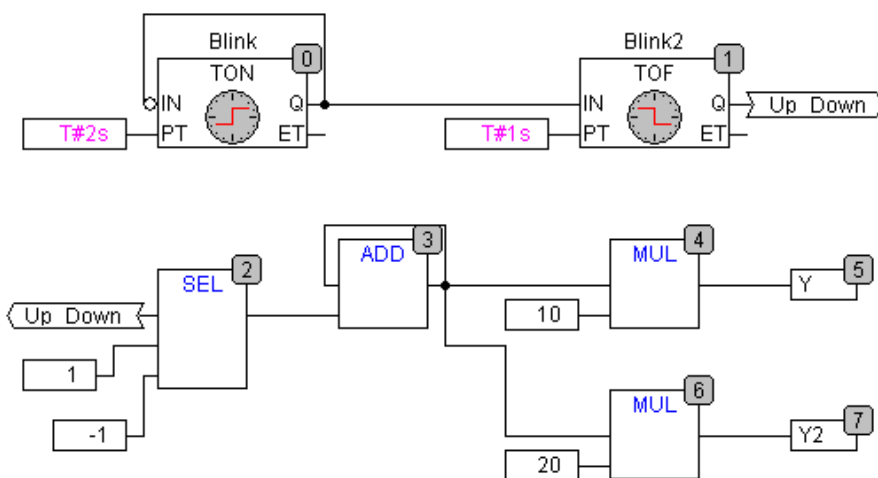
Sequential function chart on myös graafinen ohjelmointikieli, joka mahdollistaa kronologisen järjestyksen eri toimintojen välillä.



KUVA 5. Sequential function chart -esimerkki. (Codesys Online help 2017.)

3.1.6 CFC (Continuous function chart)

Continuous function chart perustuu function block diagram -ohjelmointikieleen. CFC:ssa ei kuitenkaan eroteta operaatioita eri lohkoihin (network) vaan ne voidaan sijoittaa vapaasti ja niiden välillä voidaan siirtyä alla olevan kuvan mukaisesti.



KUVA 6. Continuous function chart -esimerkki. (Codesys Online help 2017.)

4 SÄHKÖSUUNNITTELU

Sähkö- ja automaatiolaitteiden valmistus vaatii sähkö- ja automaatio suunnitelman, jotta laite voidaan valmistaa, testata ja sitä voidaan käyttää turvallisesti. Toisaalta suunnitelmat auttavat myös huolto- ja vikatilanteissa ymmärtämään laitteen toimintaa.

Sähkösuunnitelma jaetaan tyypillisesti kolmeen eri kaavioon, jotka ovat pää-, piiri- sekä johdotuskaavio. Suunnitelmaan voidaan näiden lisäksi täydentää myös erilaisilla kaavioilla, jotka eivät suoraan liity sähkösuunnitteluun, mutta auttavat kuitenkin hahmottamaan järjestelmän toimintaa tai laajuutta. Tällaisia kaavioita voivat olla mm. johdotuskaavio, jossa esitetään johdotusreitit keskuksessa tai yleiskaavio, jossa voidaan kuvata miten eri laitteet tai laitekokonaisuudet ovat kytkeytyneenä toisiinsa.

4.1 Pääkaavio

Pääkaaviossa tai toiselta nimeltään yleiskaaviossa esitetään järjestelmän päävirtapiirit. Yleiskaavio on suhteellisen yksinkertainen, usein yksiviivaista esitystä käyttävä kaavio, joka esittää järjestelmän, osajärjestelmän, asennuksen, osan, laitteen tai ohjelman isommasta kokonaisuudesta. Yleiskaavio voi olla lähtökohtana myöhemmälle suunnittelutyölle, esim. yksityiskohtaisempien kaavioiden kuten toimintakaavioiden ja piirikaavioiden laatimiselle. (Pere 1998.)

4.2 Piirikaavio

Piirikaavio on kaavio, joka esittää järjestelmän, osajärjestelmän, asennuksen, osan, laitteen, ohjelman jne. piirit toteutettuna. Siinä esitetään osat ja liitännät piirrosmerkeillä, jotka on järjestelty esittämään toiminnat ottamatta välttämättä huomioon yksiköiden fyysistä kokoa, muotoa ja sijaintia. Piirikaavio on sähköpiirustuksien perusdokumentti. Paitsi sitä, että piirikaavio toimii itsenäisenä sähköpiirustuksena, se on useiden muiden sähköpiirustuksien lähtökohtana (mm. liitäntäkaaviot ja kaapeliluettelot). (Pere 1998.)

4.3 Valitun ohjelman esittely

Autodesk Electrical on sähkösuunnitteluun tarkoitettu suunnitteluohjelmisto. Electrical pitää sisällään mm. layoutsuunnitteluun sekä piirikaavioiden suunnitteluun tarvittavat osiot sekä mahdollisuuden yhdistää 3D-mallinnus ja sähkösuunnittelu siten, että johtoreitit ovat nähtävissä 3D-mallissa. Tämä ominaisuus vaikuttaa erittäin hyvältä, koska näin johdoille tulee varattua reitit jo ennen prototyypin valmistamista. (AUTODESK 2017a.)

Electrical pitää sisällään kirjaston, josta löytyy valmiiksi eri valmistajien komponentit. Kirjasto ei ole kuitenkaan täydellinen ja esimerkiksi perusasennus antoi vain muutamien logiikkavalmistajien logiikat käytettäväksi. Komponenttivalmistajilla voi olla tarjolla kirjastoja, mutta valitettavasti kaikilta ei löydy sopivia kirjastoja kaikille mahdollisille CAD-ohjelmille. Maksullisia kirjastoja voidaan tarjota myös ohjelmistomaahantuuojien tai koulutusta tarjoavien yritysten toimesta. (AUTODESK 2017b.)

Electrical tarjoaa erilaisia mahdollisuuksia kuvien oikeellisuuden varmistamiseksi. Electrical antaa automaattisesti johdoille numerot sekä jokaiselle komponentille oman TAG-numeron. Näin varmistutaan siitä, että mm. komponenttiluettelo on oikein eikä esim. samaa osaa ole käytetty kahdesti eri paikoissa. Ohjelma osaa varoittaa erilaisista virheistä esim. johdotuksen ollessa poikki tai jos jotain pistettä ei ole johdotettu ollenkaan. (AUTODESK 2017c.)

Electrical pitää lukua myös releiden ja kontaktorien kelojen ja koskettimien käytöstä. Näin ei pääse syntymään tilannetta, että on käytetty samat koskettimet useaan kertaan. (AUTODESK 2017d.)

Electricalissa on mahdollista tehdä myös omia sisäisiä piirikaavioita, bloqueja, joita on helppo käyttää uudelleen eri suunnitelmassa. Tämä nopeuttaa työtä erityisesti silloin kun tuotteista tehdään erilaisia versioita, jotka kuitenkin pitävät sisällään samoja osioita. (AUTODESK 2017e.)

Electrical tarjoaa myös mahdollisuuden logiikkakaavioiden luomiseen mm. Excel-taulukon avulla. Tämä ominaisuus mahdollistaa digitaalisen suunnittelutiedon tuonnin toisesta ohjelmasta, joko suoraan tai muokattuna. Taulukossa voidaan määritellä mm. se mihin riviliittimiin logiikan tulot- ja lähdöt ovat kytkettyinä. Tästä ominaisuudesta kerrotaan lisää luvussa [6.2](#).

5 TYÖSSÄ KÄYTETTY LOGIIKKA JA SEN OHJELMOINTI

Ohjelmoitavaksi logiikaksi valikoitui asiakkaan toivomuksesta ABB AC 500. AC500-logiikkajärjestelmä koostuu CPU-yksiköstä, johon voidaan liittää eri käyttötarkoituksiin soveltuvia kommunikointi- ja I/O-moduuleita. Eri yksiköt kytketään toisiinsa käyttäen alustoja (terminal), jotka sisältävät moduuleiden liitososan sekä johdotusta varten kytkentäosan. Yksiköitä voidaan kytkeä toisiinsa enintään kymmenen.

5.1 LOGIIKKA

5.1.1 CPU-yksikkö PM573-ETH

CPU-yksiköitä on tarjolla useita erilaisia, joista valitaan kulloiseenkin tarpeeseen sopiva mm. muistin ja prosessoritehon osalta. Käytimme tässä asiakastyössä PM573-ETH-mallia, jossa on mm. Ethernet-verkkoliitäntä, muistia 512kB, I/O-määrät seuraavasti 160analogista tuloa, 160 analogista lähtöä, 320 digitaalista tuloa tai 240 digitaalista lähtöä.



KUVA 7: Logiikan keskusyksikkö. (ABB 2018a.)

5.1.2 I/O-moduuli CD522

CD522-laajennusmoduuliin voidaan ohjelmoida seuraavat vaihtoehtoiset toiminnot

- 2 kpl enkooderituloja
- 2kpl PWM (pulse width modulation)-lähtöä
- 6kpl digitaalista sensorituloa
- 10 digitaalituloa tai 10 digitaallilähtöä (24VDC). (ABB 2018b.)

5.1.3 I/O moduuli DC523

DC523-laajennusmoduuliin voidaan ohjelmoida yhteensä 24 digitaalista I/O-liitäntää, jotka voivat olla joko tuloja tai lähtöjä. (ABB 2018c.)

5.1.4 I/O moduuli AO523

AO523-laajennusmoduuli on 16-kanavainen ja jokainen kanava voidaan ohjelmoida toimimaan joko virta- tai jännitelähtöinä. Virtalähdöt voivat olla 0-20mA tai 4-20mA, mutta kuitenkin enimmillään 8 yhtäaikaista kanavaa. Jännitelähtö on -10 ... +10VDC. (ABB 2018d.)

5.1.5 I/O-moduuli AI523

AI523-laajennusmoduuli on 16-kanavainen ja jokainen kanava voidaan ohjelmoida toimimaan joko virta- tai jännitetulona. Virtatulot ovat 0-20mA tai 4-20mA ja jännitetulot -10...+10VDC tai 0-10VDC. Lisäksi tulot voivat olla mm. PT100, PT1000 tai NI1000 ja niitä voidaan lukea kuten digitaalituloa. (ABB 2018e.)

5.2 OHJELMOINTI

Logiikan ohjelmointiin käytetään ABB:n kehittämää ABB Automation builder v.1.2-ohjelmaa. ABB Automation Builder on ohjelmistoalusta, jonka avulla koneiden ja laitteistojen rakentajat voivat automatisoida koneet ja järjestelmät tehokkaasti. Ohjelmisto yhdistää automaation kokoonpanon, ohjelmoinnin, virheidenpoiston ja ylläpidon projektityökalut yhden ja saman selkeän käyttöliittymän alle. Automation Builder on valmis ratkaisu nykypäivän teollisten automaatioprojektien suurimpaan yksittäiseen kustannustekijään: ohjelmistoihin. (ABB 2016f.)

Automation builder 1.2 on saatavissa ilmaiseksi (Basic versio) ja käytimme ilmaisversiota tässä asiakasprojektissa. Asiakasprojektin kertaluonteisuuden vuoksi pyrimme tekemään työn ilmaisella versiolla. Työn laajuus oli sellainen, että ominaisuuksien puolesta Basic-versio oli riittävä.

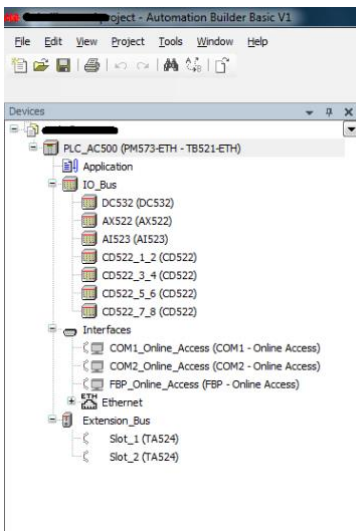
Automation builder 1.2 sisältää mahdollisuuden digitaalisen suunnittelutiedon uudelleen käytölle (ECAD round-trip engineering), mutta vain maksullisessa Premium-versiossa. Siinäkin tuki on suoraan vain muutamalle sähkösuunnitteluohjelmistolle kuten EPLAN ja Zuken E3. ABB tukipalvelusta sain kuitenkin tiedon, että digitaalinen suunnittelutieto on jossain määrin mahdollista siirtää myös Basic-versiosta, joten päätin kokeilla sitä. Luvussa 6 kerron tästä lisää. (ABB 2018g.)

5.3 PROJEKTIN LUOMINEN

Käytin työssä ST (Structured text) -ohjelmointitapaa, joka valikoitui alihankkijamme aiemmin samalle asiakkaalle tekemän projektin perusteella. Strukturoitu teksti oli myös minulle tutuin ohjelmointitapa aiemman työhistoriani perusteella.

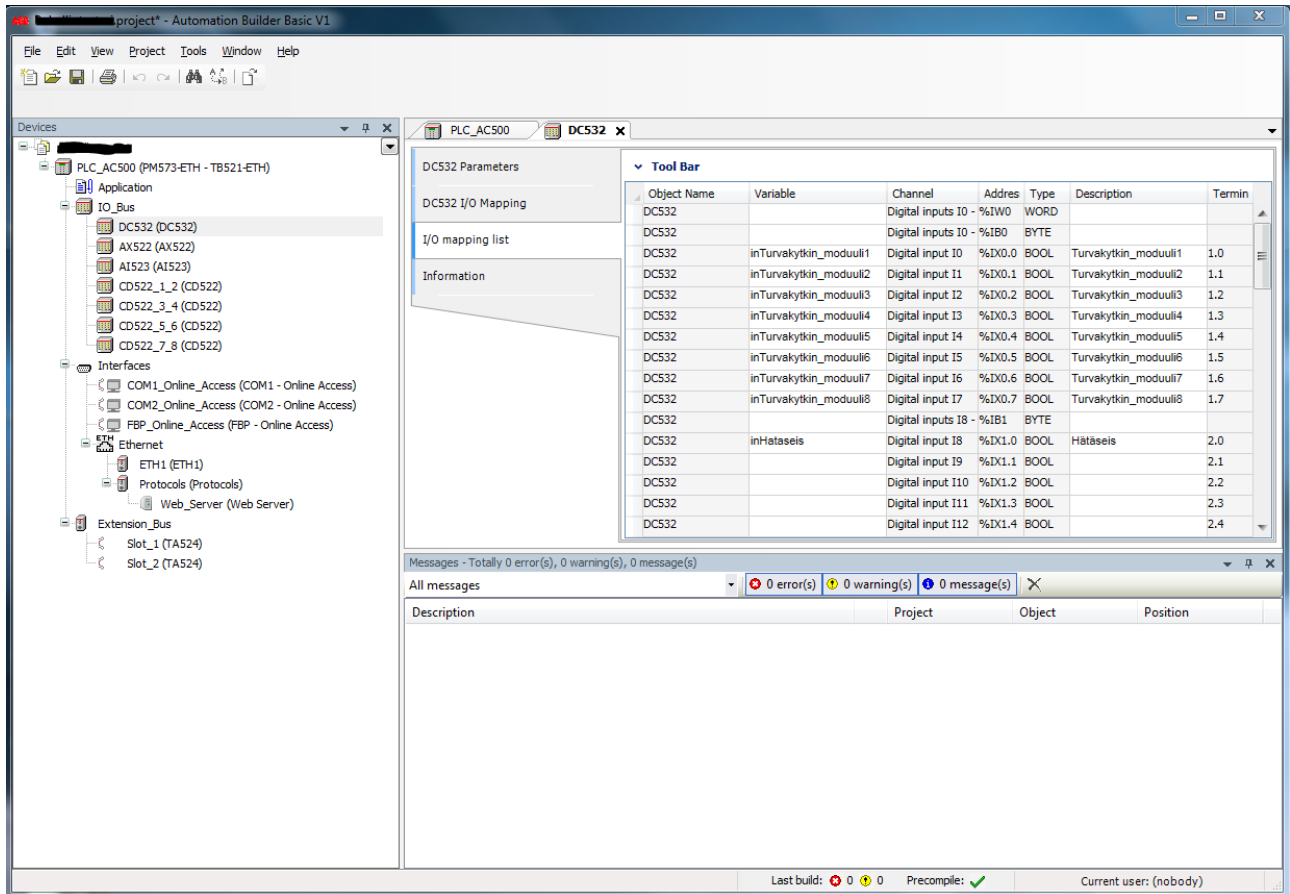
5.3.1 Logiikan konfigurointi

Automation builder 1.2 sisältää ABB-logiikoidenkonfigurointiosion sekä Codesys-pohjaisen ohjelmointiympäristön. Kun aloitetaan uuden projektin luominen, valitaan siihen käytettävät komponentit kuten CPU:n tyyppi ja käytettävät I/O-moduulit. Projektinäkymässä (KUVA 8) näkyy luotuun projektiin lisätyt laajennokset mm. IO_Bus ja CPU-yksikkö, joka on ylinnä. Tässä osiossa valitaan myös käytettävät kommunikointiväylät. Tässä työssä käytettiin Ethernet-liitäntää. Ethernet-liitännän alla on määrittelyt mm. verkkopalvelimelle.



KUVA 8: Projektinäkymä

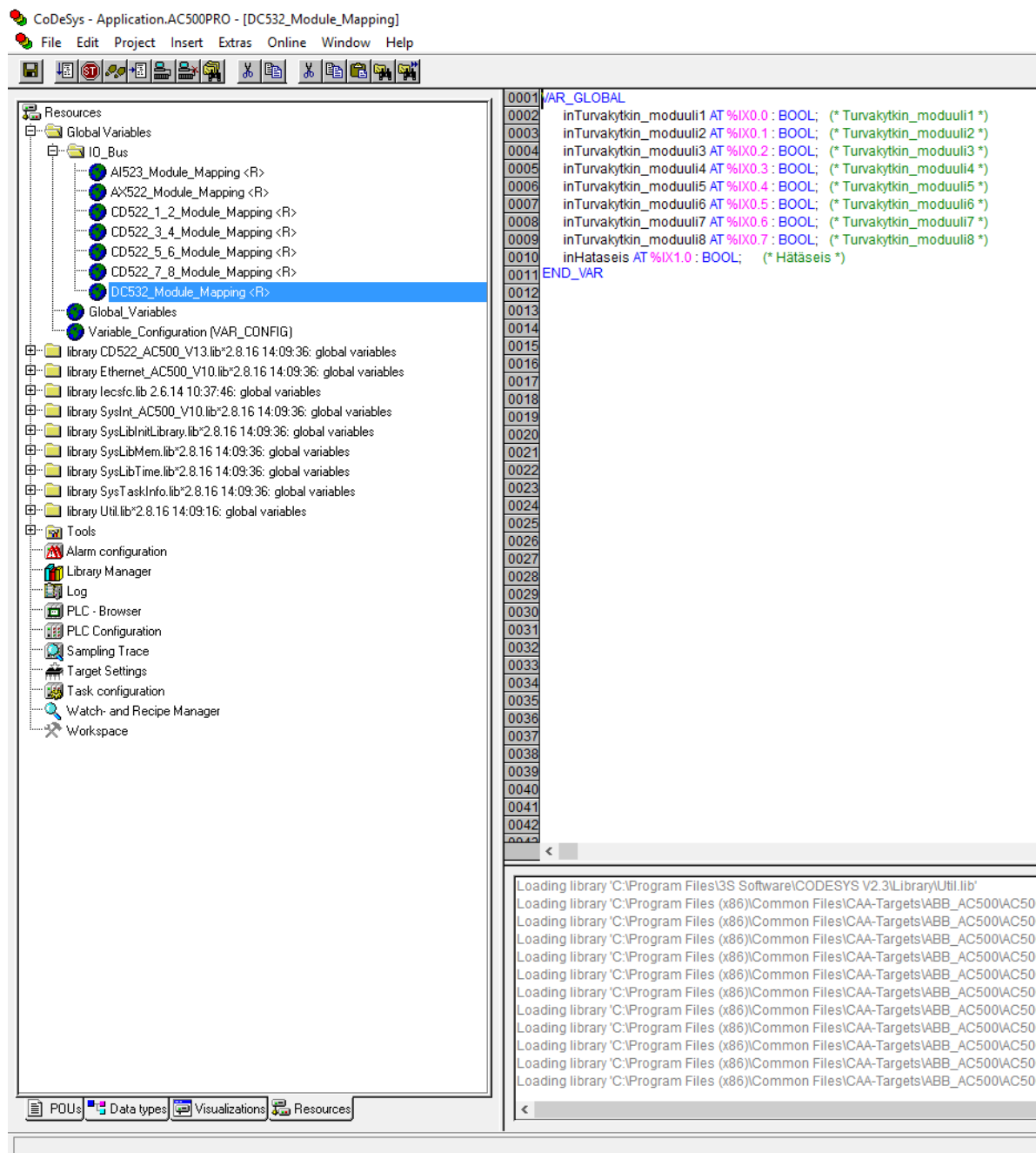
Konfigurointi osiossa (kuva 9) näkyy mm. IO moduuleihin tehdyt määritykset ja lisätyt muuttuja nimet. Tässä tehdyt määrittymisen ja muuttujanimet siirtyvät logiikkaohjelmaan, joka tehdään Codesys-ympäristöllä.



KUVA 9: Moduulin konfigurointinäkymä

5.3.2 Ohjelmointiympäristö

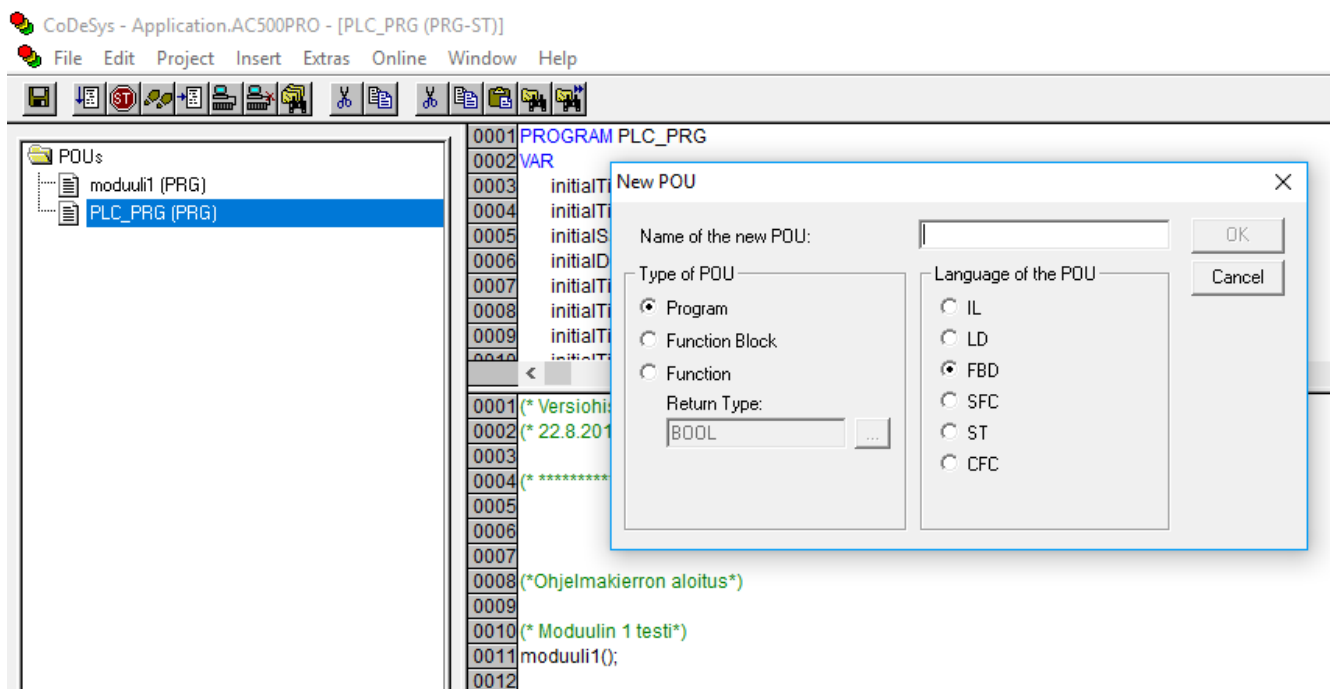
Codesys-ympäristö (KUVA 10), joka avautuu projektinäkömän ”Application”-kohtaa tuplaklikkaamalla. Ikkunan vasemmassa laidassa on jaoteltuna eri välilehdille mm. ohjelmat, datatyypit, visualisointi sekä resurssit, jotka ovat käytettävissä ohjelman tekemiseen. Kuvassa 4 on avattuna resurssit välilehdeltä moduulin DC532 muuttujamäärittelyt, jotka siirtyivät Codesys-puolelle ABB Automation builder -ohjelman määrittelyksistä (KUVA 3). Nämä muuttujanimet ovat VAR_GLOBAL -esittelyn alla, jotta ne näkyvä kaikille koodilohkoille samalla tavalla ja niihin viittaaminen on koodista helpompaa.



KUVA 10. Codesys-ohjelmointinäköymän resurssit-välilehti.

Ohjelmointi suoritetaan POUs-välilehden kautta lisäämällä PLC_PRG -niminen pääohjelma, joka on tyyppiä Program. Tästä ohjelmasta suoritetaan kutsuja muihin ohjelmiin, jotka voivat olla tyypiltään Program, Function Block tai Function. Jokainen uusi POU voi olla tyypiltään joku näistä kolmesta. Tyypin lisäksi valitaan ohjelmoinnissa käytettävä kieli, joka puolestaan voi olla joku kuudesta, luvussa 3 mainitusta kielistä (IL, LD, FBD, SFC, ST tai CFC) (KUVA 11).

Tämän pidemmällä tämän työn puitteissa ei ole tarkoitus mennä itse ohjelmoinnissa. Todettakoon kuitenkin sen verran kuvaan 11 liittyen, että siinä näkyy ohjelman PLC_PRG rakenne. Ylemmässä lohossa on ohjelman nimi PROGRAM PLC_PRG ja VAR-osiossa tämän ohjelman sisällä käytettävät muuttujat. Alempaan lohkoon kirjoitetaan varsinainen ohjelmakoodi, joka on tässä tapauksessa kutsu moduuli1()-nimiseen aliohjelmaan. Tämä aliohjelma puolestaan sisältää kaiken koodin kyseistä moduulia kohti.



KUVA 11. Ohjelmanlohkon lisääminen, tyypin ja ohjelmointikielen valinta

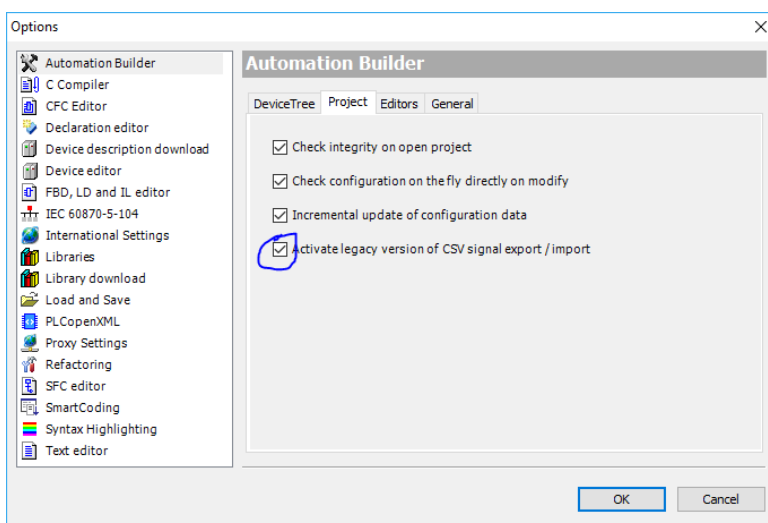
6 DIGITAALISEN SUUNNITTELUTIEDON UUELLEEN KÄYTTÖ KÄYTÄNNÖSSÄ

Digitaalinen suunnittelutieto ja sen siirtäminen ohjelmien välillä (Round-trip engineering) tarkoittaa yleisesti ottaen sitä, että yhdistetään useiden eri työkalujen tuottama aineisto yhteiseksi kokonaisuudeksi, jotta vältytään saman asian uudelleen tekemiseltä. Pitkälle vietyä digitaalisen suunnittelutiedon uudelleen käyttö olisi parhaimmillaan ryhmätyöskentelyssä, jossa samaan projektiin osallistuu useita henkilöitä ja jopa eri toimipisteistä. Näin voitaisiin varmistua siitä, että kaikilla on aina oikea tieto käytettävissä eikä esim. kirjoitus- tai ajatusvirheitä pääsisi syntymään.

Tämän työn perimmäinen tarkoitus oli tutustua siihen, että voidaanko yrityksen sisällä siirtää kertaalleen tehtyä suunnittelutietoa ohjelmien välillä. Tämän työn yhteydessä kokeiltiin, että voiko logiikkaohjelmoinnissa saatua tietoa siirtää ja käyttää suoraan hyödyksi sähkösuunnittelussa, niillä työkaluilla, joita yrityksessä on käytettävissä.

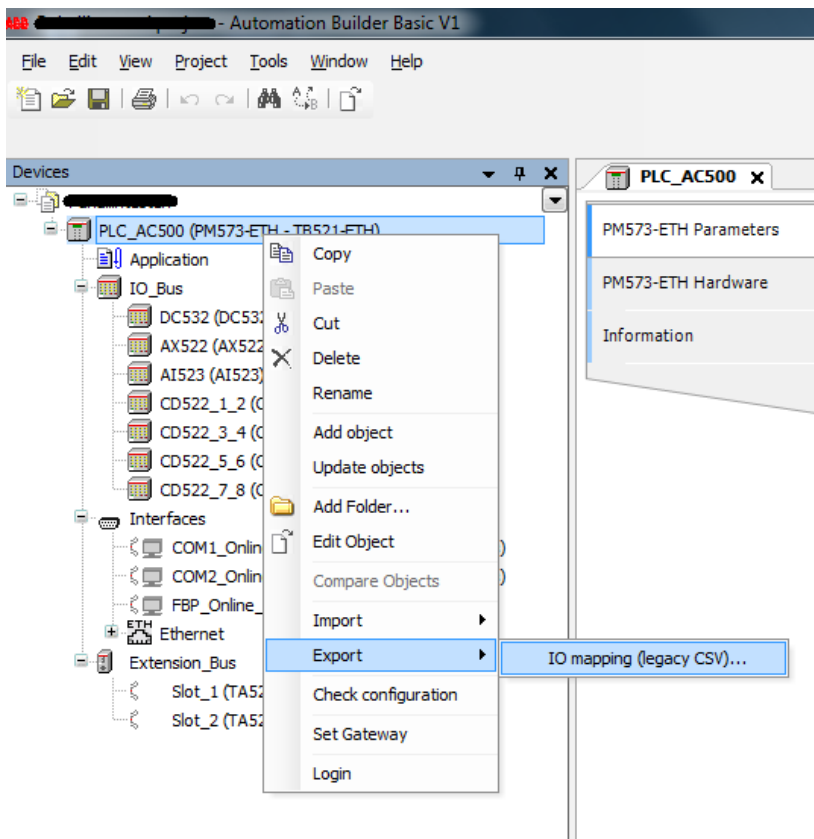
6.1 I/O-tiedon siirtäminen logiikkaohjelmoinnista sähkösuunnitteluun

ABB Automation Builder-ohjelmassa luotu I/O-tieto voidaan siirtää sähkösuunnitteluohjelmaan usealla eri tavalla. Paras tapa siirtää tieto on käyttää siihen maksullisen version omaa siirtotyökalua. Tässä projektissa meillä oli käytettävissä ABB Automation builderin ilmainen Basic-versio, joten jouduimme käyttämään ”Legacy”-versiota, joka käyttää taulukkomuotoista CSV-tiedostoa tiedon siirtoon. Legacy-version tuki joudutaan konfiguroimaan erikseen päälle Automation builderin asetuksista (KUVA 12).



KUVA 12. Asetussivu

Kun tuki on asetettu päälle, ilmestyy projektin valikon alle mahdollisuus I/O-tiedon viemiselle CSV-tiedostoon (KUVA 13).



KUVA 13. I/O-tiedon vienti

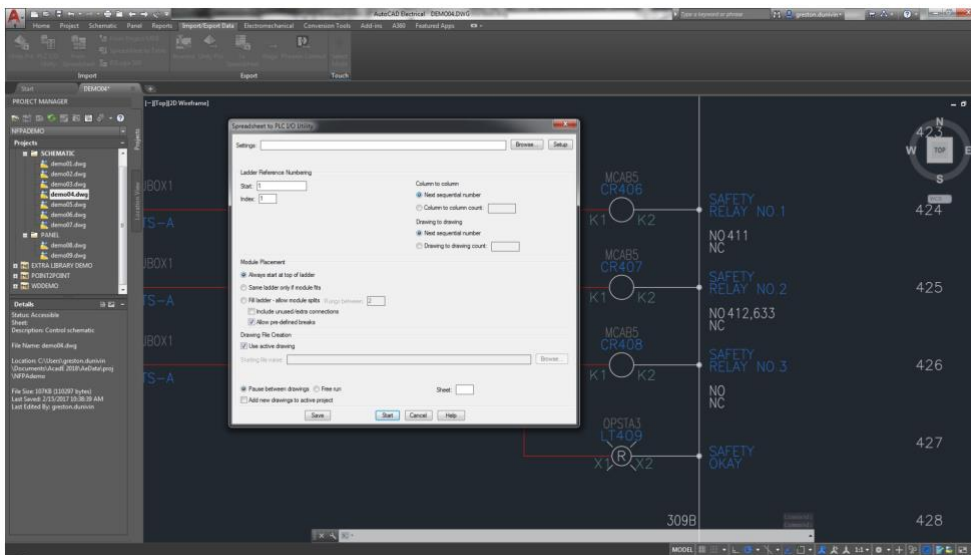
Export-toiminto tuottaa CSV-muodossa olevan I/O-tiedoston, jota voidaan tarkastella esim. Microsoft Excel-taulukkolaskentaohjelmalla. CSV-tiedostossa (KUVA 14). on mm. jokaiselle I/O-pisteelle annettu nimi ja osoite sekä laitteen nimi, jotka vastaavat kuvissa 9 ja 10 olevia tietoja.

	A	B	C	D	E	F	
1	//CoDeSys Mapping Export V1.2						
2	//Mapped variable	//Paramete	//Unit	//Description	//IEC addre	//Device name	
3	//Important: change only first, third or fourth column in Excel or add variable name before first						
4		Digital inputs I0 - I15			%IW0	DC532	
5		Digital inputs I0 - I15@1			%IW0	DC532	
6		Bytes			%IB0	DC532	
7		Digital inputs I0 - I7			%IB0	DC532	
8	inTurvakytkin_moduuli1	Digital input I0		Turvakytkin_moduuli1	%IX0.0	DC532	
9	inTurvakytkin_moduuli2	Digital input I1		Turvakytkin_moduuli2	%IX0.1	DC532	
10	inTurvakytkin_moduuli3	Digital input I2		Turvakytkin_moduuli3	%IX0.2	DC532	
11	inTurvakytkin_moduuli4	Digital input I3		Turvakytkin_moduuli4	%IX0.3	DC532	
12	inTurvakytkin_moduuli5	Digital input I4		Turvakytkin_moduuli5	%IX0.4	DC532	
13	inTurvakytkin_moduuli6	Digital input I5		Turvakytkin_moduuli6	%IX0.5	DC532	
14	inTurvakytkin_moduuli7	Digital input I6		Turvakytkin_moduuli7	%IX0.6	DC532	
15	inTurvakytkin_moduuli8	Digital input I7		Turvakytkin_moduuli8	%IX0.7	DC532	
16		Digital inputs I8 - I15			%IB1	DC532	
17	inHataseis	Digital input I8		HÄtÄseis	%IX1.0	DC532	

KUVA 14 CSV-tiedosto esimerkki

6.2 Suunnittelutiedon tuonti sähkösuunnitteluohjelmaan

Suunnittelutieto tuodaan sähkösuunnitteluohjelmaan Electrical:n Import/Export Data-valikon kautta. Valikosta löytyy PLC I/O Utility-työkalu, jossa annetaan tuontia varten määrittelyt ja valitaan luettelosta haluttu PLC.



KUVA 15. PLC I/O drawings from spreadsheets. (AUTODESK 2018f.)

Autodeskin tukisivustolta löytyvää ohjetta seuraamalla nähdään, kuinka logiikkaan liittyvät piirikaaviot luodaan automaattisesti, taulukossa annettujen arvojen perusteella. (AUTODESK 2018g.)

Koska meillä oli tehtynä vain Electrical:n perusasennus, ei kirjastosta löytynyt käyttämäämme logiikkaa. Käyttämämme logiikka sisältyy ylläpitosopimuksen piirissä olevaan kirjastoon, jonka tarjoaa Future Group. Tämä kirjasto on erittäin hyödyllinen ja se kannattaa hankkia, mikäli ohjelmiston käyttö on jatkuvaa. Hyvin alkanut työ ei näin ollen meidän osalta voinut toteutua. Toinen ongelma tässä menetelmässä on se, että automaattisesti tuotu tieto ei konfiguroinneista huolimatta vastannut aiemman projektin ulkoasua eikä näin ollen soveltunut tähän työhön.

6.3 Esimerkkitiedosto ja PLC I/O Utility

Koska Automation builderin luoman tiedoston käyttäminen ei tässä tapauksessa onnistunut, esitän luvussa 6.2 esitetyn toiminnon Electrical:n mukana tulleen esimerkki tiedoston avulla.

Tiedosto voi olla ABB Automation builderin luoma tai se voidaan tehdä käsin. Siihen voidaan lisätä tietoja kuten tässä esimerkkitapauksessa. Esimerkissä tiedostoon on lisätty mm. kenttäkotelointi sekä painonappeja, jolloin myös ne tulevat valmiina piirikaavioon.

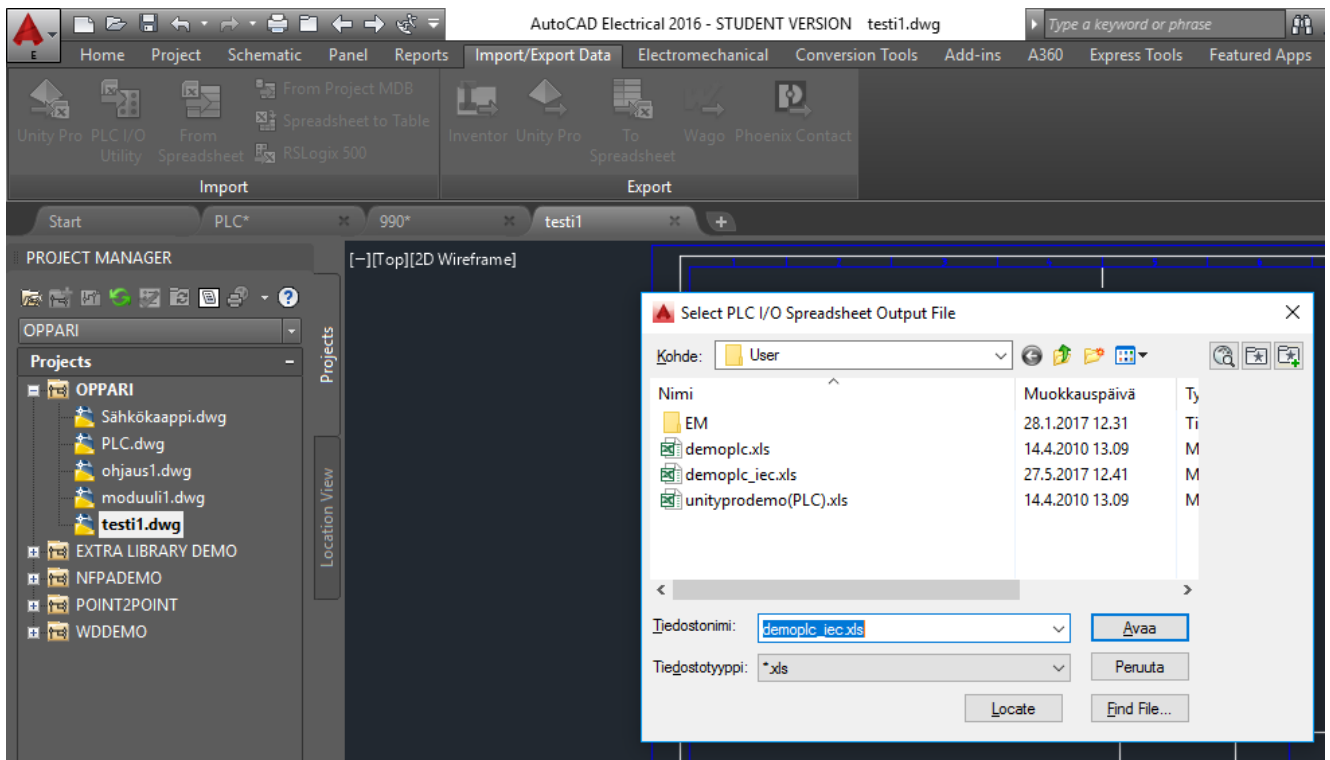
Taulukossa on hyödynnetty sarakkeiden reunaviivoja kuvaamaan kytkentäpisteitä tai sijoittumista.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1	CODE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
2	11THAD	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
3																									
4																									
5																									
6																									
7																									
8																									
9																									
10																									
11																									
12																									
13																									

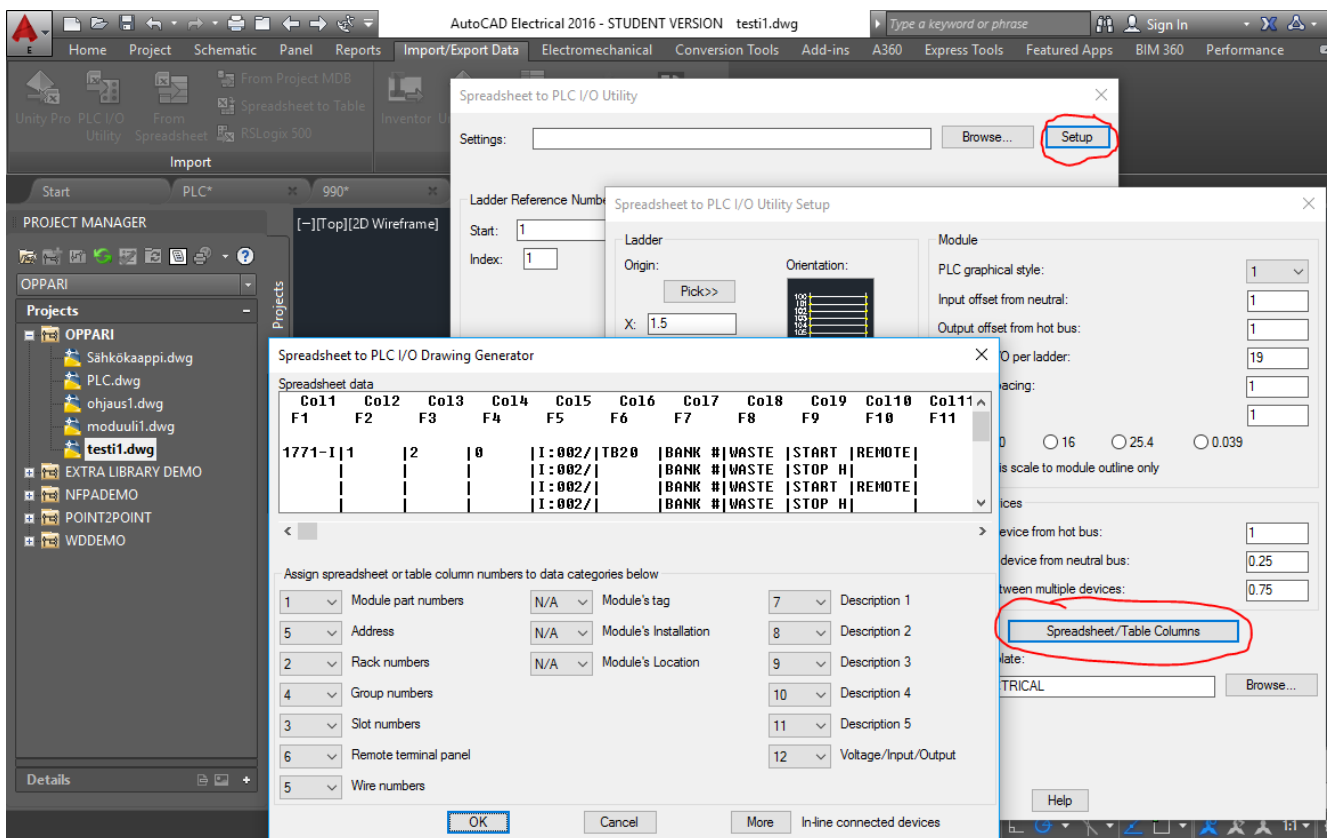
KUVA 16. Osa käytetyn Excel-taulukon sisällöstä

Toisessa vaiheessa valitaan piirikaavion luontiin halutut asetukset kuten riviväli, tiedoston nimi yms. Lisäksi kuvassa 18 näkyvien ympyröityjen painikkeiden kautta voidaan määrittää, mitkä kaikki Excel-tiedostossa olevista sarakkeista otetaan mukaan piirikaavion luontiin. Tämän toiminnon avulla voidaan eri muodoissa olevat tiedostot sovittaa import-työkaluun.

Kolmannessa vaiheessa (KUVA19) nähdään osa luodusta piirikaaviosta. Kaikki kuvassa esitetyt tiedot ovat tulleet suoraan Excel-taulukosta eikä yhtään komponenttia tai viivaa ole tarvinnut piirtää käsin. Tämä pienentää inhimillisten erehdyksen määrää ja mahdollistaa vakiomuotoisten piirikaavioiden tuottamisen, vaikka sivuja olisi useampia.



KUVA 17. Valitaan avattava tiedosto.



KUVA 18. Sarakkeiden valinta



KUVA 19. Valmis, Excel-taulukosta muodostettu piirikaavio

7 POHDINTA

Idea digitaalisen suunnittelutiedon uudelleen käytöstä on erittäin hyvä. Isoissa projekteissa suunnittelutietoa on paljon ja sen jatkuva pitäminen ajan tasalla on erittäin työlästä ja aikaa vievää. Valitettavasti, ainakin omalta kohdalta, täytyy todeta, että hienosta ajatuksesta huolimatta työ ei onnistunut kuten olin sen alun perin ajatellut.

Kun eletään vuotta 2018, olisin voinut kuvitella, että digitaalinen suunnittelutieto saataisiin muodostettua yleiskäyttöiseen muotoon niin, että se olisi helposti siirrettävissä avoimen standardin mukaisesti. Tällä hetkellä avointa standardia ei ole, joten ohjelmistovalmistajat tekevät asiat niin kuin itse haluavat. Tämä näkyy mm. siinä, että esimerkiksi ABB Automation builder tukee vain kahta sähkösuunnitteluohjelmistoa ”natiivisti”. Toinen merkittävä puute on siinä, että eri suunnitteluohjelmistot käyttävät omia kirjastojaan. Näin ollen eri komponenttivalmistajat joutuvat tekemään omista komponenteistaan useita eri versioita. Tämä on kallis ja aikaa vievä toimenpide, joten aina jää jokin ohjelma vähemmälle huomiolla. Työtä tehdessä törmäsin siihen, että Electrical-ohjelmasta ei löytynyt sopivaa logiikkakomponenttia, joten työtä ei ollut mahdollista tehdä automaattisesti, ainakaan tässä aikataulussa.

Mielestäni ohjelmistovalmistajien pitäisi istua saman pöydän ääreen ja sopia yhteinen standardi mm. kirjastoille ja erilaisille tiedoille, joita voidaan siirtää ohjelmista toisiin. Toisaalta tämän työn otanta eri ohjelmistojen suhteen on kuitenkin varsin pieni, joten voi olla mahdollista, että tämä asia on hoidettu paremmin muiden ohjelmien kanssa.

Valitettavasti aika ei riittänyt testaamaan, olisiko ABB Automation builder toiminut maksullisella Premium-versiolla paremmin esimerkiksi Eplan- tai E3-sähkösuunnitteluohjelmistojen kanssa. Tulevaisuudessa jos ja kun yrityksiin valitaan ohjelmistoja, tulisi valinnassa ottaa paremmin huomioon eri ohjelmistojen toimiminen keskenään. Paremmalla ohjelmistojen yhteen sovituksella on mahdollista saada todellisia säästöjä sekä parantaa suunnitelmien laatua.

LÄHTEET

ABB. Detailed information for: PM573-ETH. 2017a. Saatavilla: <http://new.abb.com/products/1SAP130300R0271/pm573-ethac500-prog-logic-controller-512kb24vdc-ethernet-2xrs232-485-fbpsd-car>. Luettu 3.2.2018.

ABB. Detailed information for: CD522. 2017b. Saatavilla: <http://new.abb.com/products/ABB1SAP260300R0001>. Luettu 3.2.2018.

ABB. Detailed information for: DC532. 2017c. Saatavilla: <http://new.abb.com/products/ABB1SAP240100R0001>. Luettu 3.2.2018.

ABB. Detailed information for: AO523. 2017d. Saatavilla: <http://new.abb.com/products/ABB1SAP250200R0001>. Luettu 3.2.2018.

ABB. Detailed information for: AI523. 2017e. Saatavilla: <http://new.abb.com/products/ABB1SAP250300R0001>. Luettu 3.2.2018.

ABB. ABB Automation builder. 2016f. Saatavilla: <http://new.abb.com/plc/fi/automation-builder>. Luettu 3.2.2018.

ABB. Automation Builder 1.2 ominaisuudet. 2017g. Saatavilla: <http://new.abb.com/plc/fi/automation-builder/alusta/ohjelmisto>. Luettu 3.2.2018.

AUTODESK, ”videos.autodesk.com. 2017a. Saatavilla: <http://videos.autodesk.com/zencoder/content/dam/autodesk/www/collections/product-design/videos/electrical-design-video-1920x1080.mp4>. Luettu 14.1.2017.

AUTODESK. Autodesk Electrical - Overview. 2017b. Saatavilla: <http://www.autodesk.com/products/autocad-electrical/overview>. Luettu 14.1.2017.

AUTODESK. Autodesk Electrical - Features. 2017c. Saatavilla: <http://www.autodesk.com/products/autocad-electrical/features/electrical-schematic-design/auto-wire-numbering-component-tagging>. Luettu 14.1.2017.

AUTODESK. Autodesk Electrical - Features. 2017d. Saatavilla: <http://www.autodesk.com/products/autocad-electrical/features/electrical-controls-design/real-time-coil-contact-cross-referencing>. Luettu 14.1.2017.

AUTODESK. Autodesk Electrical - Features. 2017e. Saatavilla: <http://www.autodesk.com/products/autocad-electrical/features/electrical-schematic-design/circuit-design-reuse>. Luettu 14.1.2017.

AUTODESK. PLC-IO-DRAWING. 2017f. Saatavilla: <https://damassets.autodesk.net/content/dam/autodesk/www/products/autodesk-autocad-electrical/fy18/overview/images/plc-i-o-drawings-spreadsheets-large-1920x1080.jpg>. Luettu 16.2.2018.

AUTODESK. Autodesk Knowledge Network. 2018g. Saatavilla: <https://knowledge.autodesk.com/support/autocad-electrical/learn-explore/caas/CloudHelp/cloudhelp/2018/ENU/AutoCAD-Electrical/files/GUID-454FFEAB-D09E-46EF-BD90-8B65D73BF742-htm.html>. Luettu 16.2.2018

Codesys. Codesys Online help. 2017. Saatavilla: <https://help.codesys.com>. Luettu 14.1.2017.

Omron, CX-One ja logiikkaohjelmointi, 2016. Koulutusmateriaali NRO: 041-FIN-16

Pere, A. 1998. Sähköpiirustus. Viiden painos toim. Kirpe Oy. Helsinki: Offsetpiste