



VAASAN AMMATTIKORKEAKOULU  
UNIVERSITY OF APPLIED SCIENCES

Joonas Perälä

# AC500 JA COMPACT HMI

Toiminnallisuuden selvitys

-----

Tekniikka  
2019

## TIIVISTELMÄ

Tekijä	Joonas Perälä
Opinnäytetyön nimi	AC500 ja Compact HMI
Vuosi	2019
Kieli	suomi
Sivumäärä	30
Ohjaaja	Olli Tuovinen

---

Tämän opinnäytetyön tarkoituksena on tutkia mahdollisuutta käyttää ABB:n AC500 ohjelmoitavaa logiikkaa yhdessä ABB:n CompactHMI-valvomojärjestelmän kanssa. Näin saavutettaisiin kompakti ja edullisempi ohjausjärjestelmä.

Tässä opinnäytetyössä pyrittiin löytämään ja tunnistamaan mahdolliset ongelmat ja esteet AC500:n ja CompactHMI:n käytölle, sekä selvittämään ne.

Käyttöönottaessa ja tutkiessa tätä AC500- ja CompactHMI-järjestelmää, tuli vastaan useita ongelmatilanteita ja haasteita, joiden ratkaiseminen vaikutti välillä lähes mahdottomalta. Keskeisiä haasteita työssä olivat erinäiset kommunikointiongelmat eri ohjelmien välillä.

Järjestelmä saatiin kuitenkin kokonaisuutena toimivaksi, ja voitiin todeta sen soveltuvan mainiosti pienempien projektien, sekä projektiosien toteuttamiseen.

## ABSTRACT

Author	Joonas Perälä
Title	AC500 and Compact HMI
Year	2019
Language	Finnish
Pages	30
Name of Supervisor	Olli Tuovinen

---

This thesis was done to examine the possibility to use ABB's AC500 PLC together with ABB's CompactHMI. AC500 is a programmable logic controller, which gathers and sends data to an automation system. This data is then processed and sent to CompactHMI, where it is displayed in user-friendly format. Using this combination would result in a compact and inexpensive solution. The purpose was to find and acknowledge the possible problems in using this combination, and to solve them.

AC500 and CompactHMI were connected using gateway software. In this setup, both AC500 and CompactHMI were running on the same computer, which made the system a bit more complex, because usually this kind of setup is built with servers. While trying to connect AC500 and CompactHMI, there were many problems, which seemed to be unsolvable. Mainly the problems were discovered while trying to communicate between different programs. Usually problems were solved by changing configuration settings.

However, this combination was found working, and suitable especially for smaller projects or project parts.

# SISÄLLYS

TIIVISTELMÄ

ABSTRACT

KUVALUETTELO

LYHENTEET JA MÄÄRITELMÄT

1	JOHDANTO .....	7
2	OHJELMAT .....	8
	2.1 CompactHMI .....	8
	2.2 Automation Builder .....	10
	2.3 CoDeSys .....	13
	2.4 Ohjelmien kommunikointi .....	<b>Virhe. Kirjanmerkkiä ei ole määritetty.</b>
	2.5 Taustaohjelmat .....	17
3	LAITTEISTO .....	20
4	TIETOTURVA .....	23
5	JÄRJESTELMÄKOKONAISUUS .....	25
	5.1 Valmis järjestelmä .....	25
6	VIKATILANTEET .....	27
7	LOPPUTULOS JA TULOSTEN VERTAILU .....	30
	7.1 Opinnäytetyön lopputulema .....	30
	7.2 Tulosten vertailu .....	30

LIITTEET

**KUVALUETTELO**

Kuva 1. CompactHMI:n Engineering workplace, jossa valvomo luodaan.....	10
Kuva 2. Automation Builderin käynnistysnäkyvä.....	11
Kuva 3. Automation Builderin projektinäkyvä.....	12
Kuva 4. Automation buidlerissa projektin muuttajat ”mäpättiin”, eli yhdistettiin oikeisiin tulo- ja lähtöliittimiin.....	12
Kuva 5. CoDeSys-ohjelmointinäkyvä.....	13
Kuva 6. OPC-konfigurointiohjelma.....	15
Kuva 7. AdvDSOPCClient-ohjelma.....	16
Kuva 8. AC500-Logiikkaohjain.....	21
Kuva 9. AX522Lkortti.....	21
Kuva 10. CI590-kortti.....	22
Kuva 11. Kortin vaihto pohjalevyyn.....	28
Kuva 12. Logiikka irrotettuna pohjalevystä.....	29

**LYHENTEET JA MÄÄRITELMÄT**

CHMI	-	Compact HMI, valvomo-ohjelmisto
AI	-	Analog input, analoginen tulokanava
AO	-	Analog output, analoginen ulostulo
DI	-	Digital input, digitaalinen sisääntulo
DO	-	Digital output, digitaalinen ulostulo
IAEN	-	ABB:n yksikkö

# 1 JOHDANTO

Tässä opinnäytetyössä tutkitaan mahdollisuutta käyttää ABB:n valmistamaa AC500-logiikkaohjainta, sekä Compact-HMI valvomojärjestelmää ABB:n IAEN-yksikön projekteissa. Lopputyön aihe tuli IAEN-yksiköstä, jossa oli mielenkiintoa tutustua aiheeseen ja sen käyttömahdollisuuksiin tarkemmin. IAEN on Vaasan Strömberg Parkissa sijaitseva ABB:n yksikkö, jossa toteutetaan pääasiallisesti vesivoimalaprojekteja Pohjoismaihin. Vesivoiman lisäksi yksikössä toteutetaan ydinvoimaprojekteja sekä dieselvoimalaitoksia.

Opinnäytetyön tavoite on selvittää, voidaanko edellä mainittua yhdistelmää käyttää kyseisen yksikön pienemmissä vesivoimaprojekteissa. Ennen lopputyön aloittamista saadun tiedon perusteella on AC500 sopimaton isompiin projekteihin, joihin siinä olevista puutteista verrattuna isompaan AC800M-ohjaimen. Compact HMI:n käyttö sen sijaan tiedetään olevan mahdollista, sillä se on vain yksinkertaisempi versio ABB:n pääasiallisesta valvomojärjestelmästä, 800xA:sta. Kuitenkaan ei tiedetä, voidaanko Compact HMI liittää ja saada toimimaan yhdessä AC500:n kanssa.

## 2 OHJELMISTOT

Työn aihe valittiin ja rajattiin sopivaksi yhdessä ohjaavan opettajan sekä ABB:n edustajan kanssa. Aihe oli kyseisellä osastolla ollut esillä, mutta sen käytännön toteutus, sekä yleinen tutkiskelu oli tekemättä. Kysymyksiä oli enemmän kuin vastauksia, ja näihin kysymyksiin tässä opinnäytetyössä pyritään vastaamaan. Kysymyksistä suurin liittyi kyseisen järjestelmän käyttökohteisiin ja siihen, onko tällä ABB:n osastolla tarpeellista ja hyödyllistä alkaa käyttämään tätä järjestelmää sen projekteissa.

### 2.1 CompactHMI

CompactHMI on ABB:n 800xA valvomojärjestelmään pohjautuva kompaktimpi ja rajatumpi valvomojärjestelmä. Se ei sisällä vain yhtä tiettyä ohjelmaa, vaan useita toisiaan tukevia ja toisistaan riippuvaisia ohjelmia ja työkaluja. Järjestelmän määrittelyt tehdään erilaisilla työkaluohjelmilla, joita järjestelmässä on useampi. Kuitenkin itse valvomon määrittelyt, luonti ja käyttö tehdään yhdestä, kahteen osaan jaetusta workplace-ohjelmasta.

Suunnitelman valmistuttua oli ensimmäinen askel selvittää ohjelmien toimivuus. Suunnitelman mukaan CHMI ja Automation builder asennetaan samaan Windows 10-käyttöjärjestelmän tietokoneeseen. Ongelmaksi kuitenkin muodostui se, että CHMI ei vielä ollut saanut virallista Windows 10-järjestelmää tukevaa päivitystä. Tästä huolimatta Windows 8.1 versiota tukevaa asennuspakettia alettiin asentamaan Windows 10 koneeseen.

Asennuspaketin mukana tulevat ohjelmistot ja ajurit, sekä itse CHMI asentuivat kaikki ongelmitta Windowsin uudempaan versioon huolimatta siitä, ettei tukea tälle uudemmalle versiolle oltu luvattu.

Askel eteenpäin oli otettu, mutta se muuttui äkkiä harppaukseksi taaksepäin, sillä jo valvomojärjestelmää käyttöönotettaessa tuli vastaan ongelma, jonka syy johtui siitä, ettei valvomojärjestelmä tukenut Windows 10-versiota. Ongelma oli, että määritettäessä tietokoneen nimeä, ei ohjelma pystynyt kyseisen nimistä tietokonetta



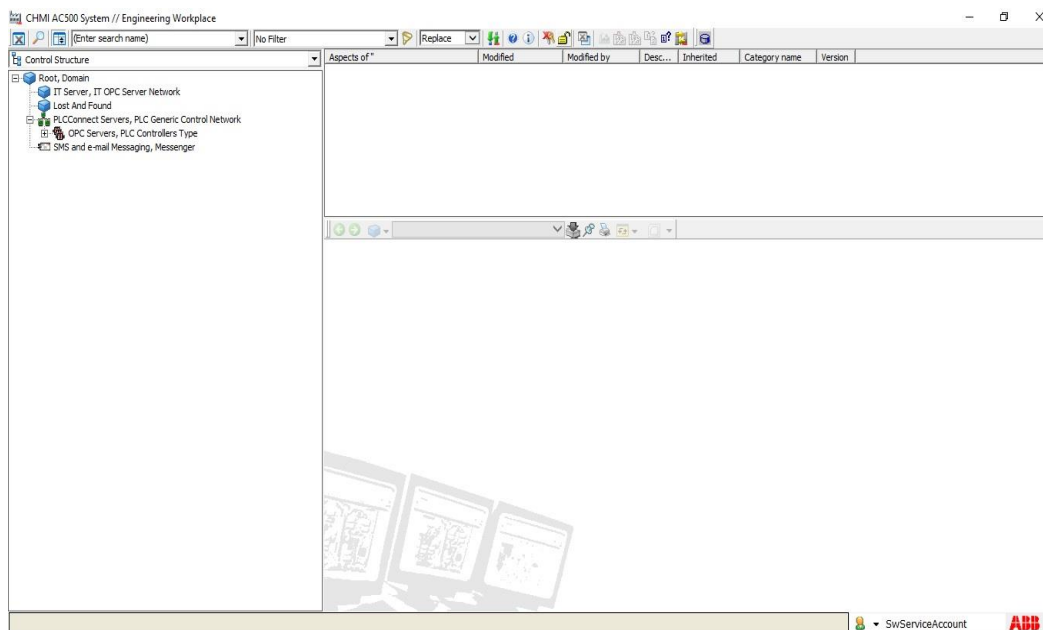
löytämään ja määrittämään ohjelmaan, vaikka nimi oli täysin oikein ja kyseessä tietokone, johon ohjelma oli asennettu.

Ongelman selvitys ja ratkaisun hakeminen aloitettiin välittömästi. Ratkaisua tähän ongelmaan haettiin ulkomaita myöten ja sitä etsiessä paljastui, että viikon sisällä tullaan julkistamaan CHMI:n versio, joka tukee Windows 10-versiota. Julkaisujankoha ei tämän opinnäytetyön kannalta olisi voinut olla parempi.

Uuden version asennuksen jälkeen voitiin tehdä valvomojärjestelmän määrittäminen loppuun. Alkuasetteluiksi täytyi Windowsiin määrittellä käyttäjäryhmät ja käyttäjät, joilla jokaisella oli omat tehtävät ja oikeudet. Näin pystytään estämään kokemattomien käyttäjien virheet, sekä vahingossa tehdyt muutokset, joilla voi olla vakavia seurauksia järjestelmän vakauteen.

Yksi yleisimmistä tapauksista täysin avoimessa ja hierarkiattomassa järjestelmässä on se, että käyttäjä luulee tietävänsä mitä on tekemässä, vaikkei todellisuudessa ymmärrä tekemänsä muutoksen seurauksia. Pahimmassa tapauksessa tällainen voi jopa kaataa koko järjestelmän ja aiheuttaa näin mittavia tappioita tuotantolaitoksessa, jossa jo tunnin pysähdyksen hintalappu voi nousta satoihin tuhansiin euroihin.

Kun määrittelyt oli tehty, ja valvomojärjestelmä saatu toimimaan, voitiin ohjelmaa alkaa tutkimaan tarkemmin. CHMI-ohjelman käyttäminen oli hyvin samanlaista kuin 800xA:n, mihin se myös pohjautuu. Tässä vaiheessa ei ollut vielä oikea aika alkaa rakentamaan valvomoympäristöä, sillä Automation builder-ohjelman asennus oli vielä edessä.



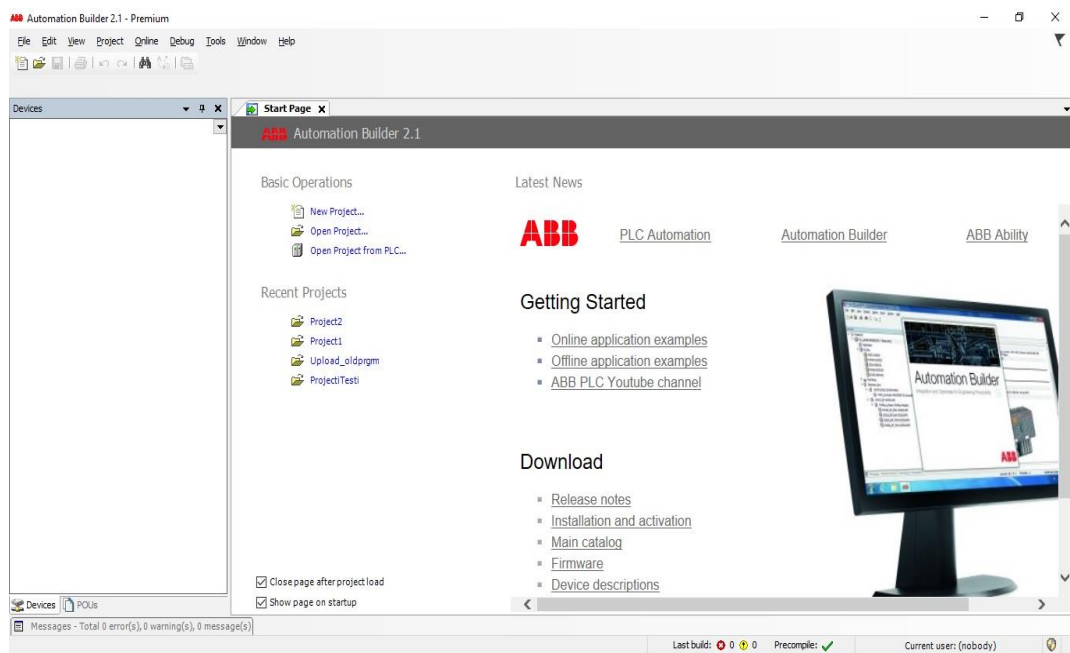
**Kuva 1.** CompactHMI:n Engineering workplace, jossa valvomo luodaan.

## 2.2 Automation Builder

Automation builder-ohjelma on tarkoitettu AC500-logiikan määrittelyyn ja ohjelmointiin. Sen päätarkoitus on määrittää logiikan oikea malli ja versio, käytettävien korttien mallit, sekä suorittaa päivitykset näille laitteille. Varsinainen ohjelmointi tehdään kuitenkin Automation builderin sisältä avattavalla CoDeSys-ohjelmalla.

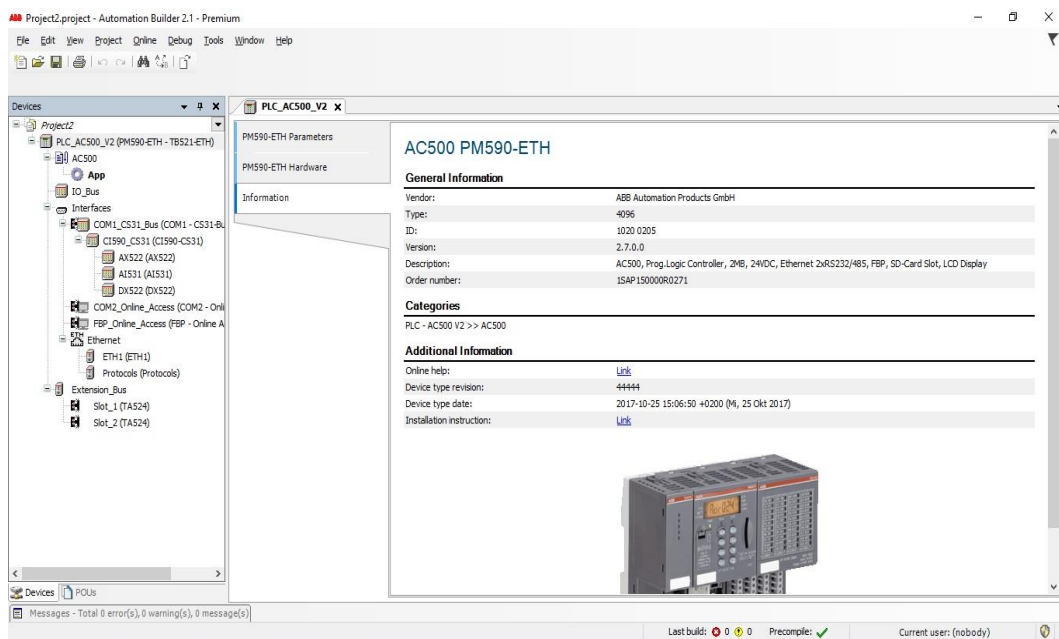
Automation builderin asennus sujui aikaisempiin haasteisiin verrattuna helposti, sillä Windows 10 tuen omaava versio löytyi jo valmiiksi.

Ohjelma oli erilainen yleisemmin käytettyyn control builderiin verrattuna, ja ohjelman käyttämisen aloittaminen vaati hieman tutustumista, sekä ohjekirjan lukemista.

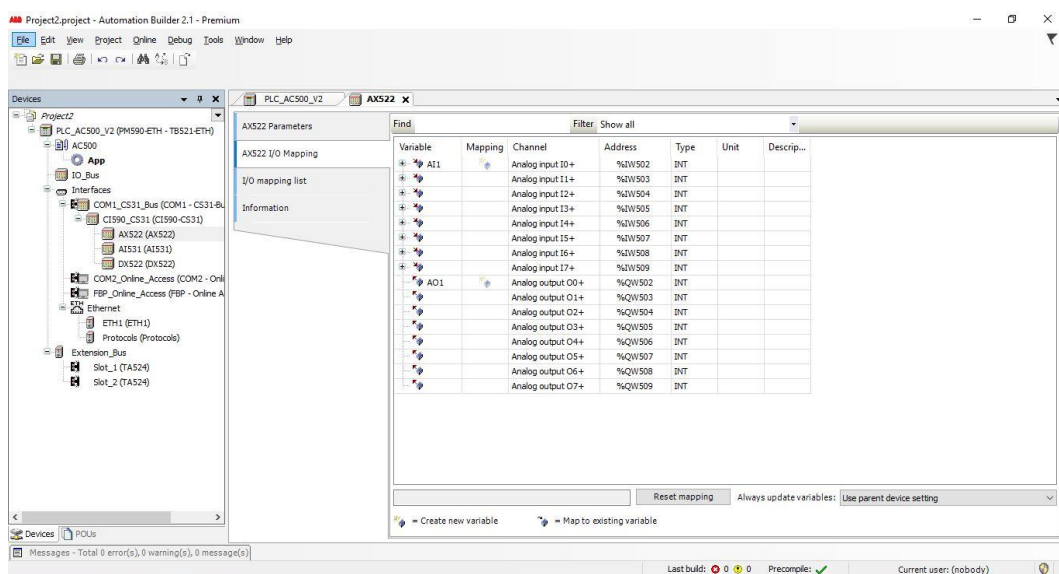


**Kuva 2.** Automation Builderin käynnistysnäky.

Tutustumisen jälkeen voitiin aloittaa itse työn toteutus. Aivan aluksi määritettiin oikea versio AC500-logiikasta, ja siihen liitetystä AI-, AO-, DI- ja DO-korteista. Kaikki nämä komponentit olivat helposti löydettävissä. Määriteltyjä asetuksia ladataessa logiikalle, ilmoitti ohjelma päivittävänsä logiikan ohjelmistoversion uusimpaan, jotta se olisi ajantasainen. Tämä ominaisuus on erinomainen, mikäli logiikan ohjelmisto halutaan pitää ajan tasalla.



**Kuva 3.** Automation Builderin projektinäkömä.

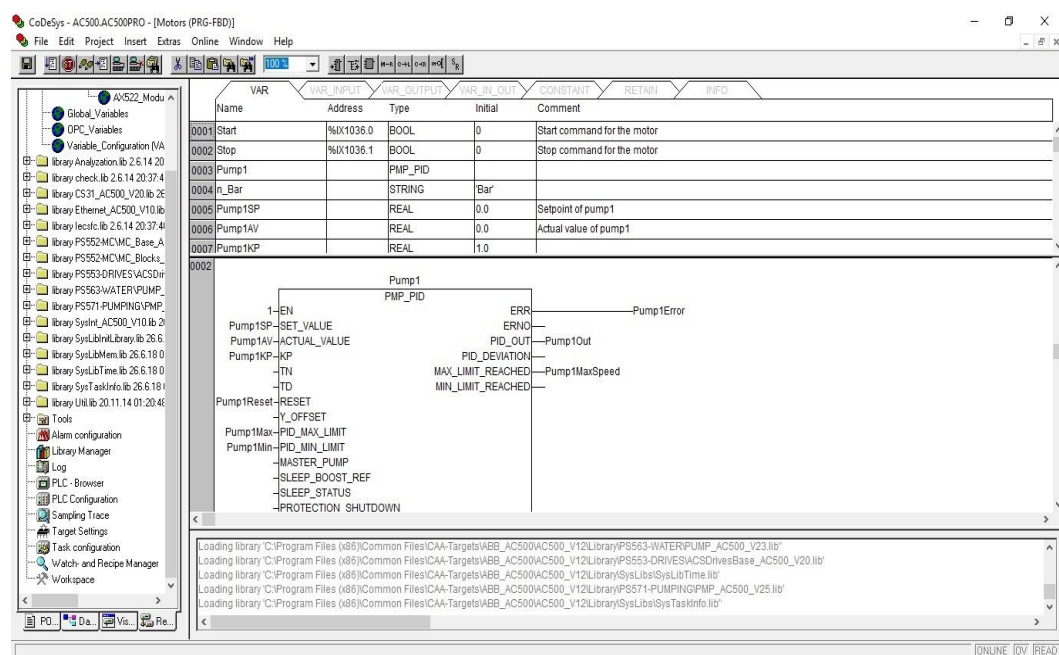


**Kuva 4.** Automation buidlerissa projektin muuttujat ”mäpättiin”, eli yhdistettiin oikeisiin tulo- ja lähtöliittimiin.

Itse ohjelmointi, eli logiikan suorittaman ohjelman luontia ei tehdä Automation builderista, vaan sen kautta avattavalla CoDeSys-ohjelmalla. CoDeSys-ohjelmointiympäristö oli hyvin erilainen kuin yleisessä käytössä olevat ABB:n control builder ja Siemensin Step7, jotka ovat CoDeSysiin verrattuna hyvin samankaltaisia keskenään. Siitä löytyy kuitenkin kaikki tarvittava ohjelmoinnin suorittamiseen, joskin sillä ohjelmointi vie huomattavasti enemmän aikaa ja vaatii totuttelua.

### 2.3 CoDeSys

CoDeSys on lisenssivapaa ohjelmointiympäristö, joka on sulautettu Automation builderiin, ja jolla tehdään AC500:n ohjelmointi. CoDeSysillä ohjelmointi on huomattavasti erilaisempaa kuin control builderilla, jolla hoidetaan AC800M:n ohjelmointi. Verrattuna control builderiin, sen ohjelmointiympäristö on paljon alkeellisempi ja sekavampi.

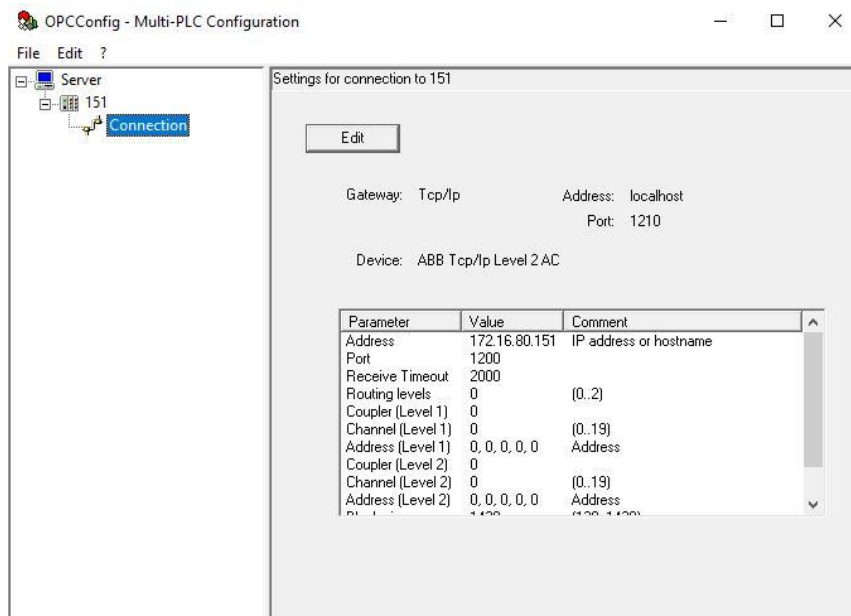


Kuva 5. CoDeSys-ohjelmointinäköymä.

## 2.4 Logiikan ja valvomon välinen kommunikointi

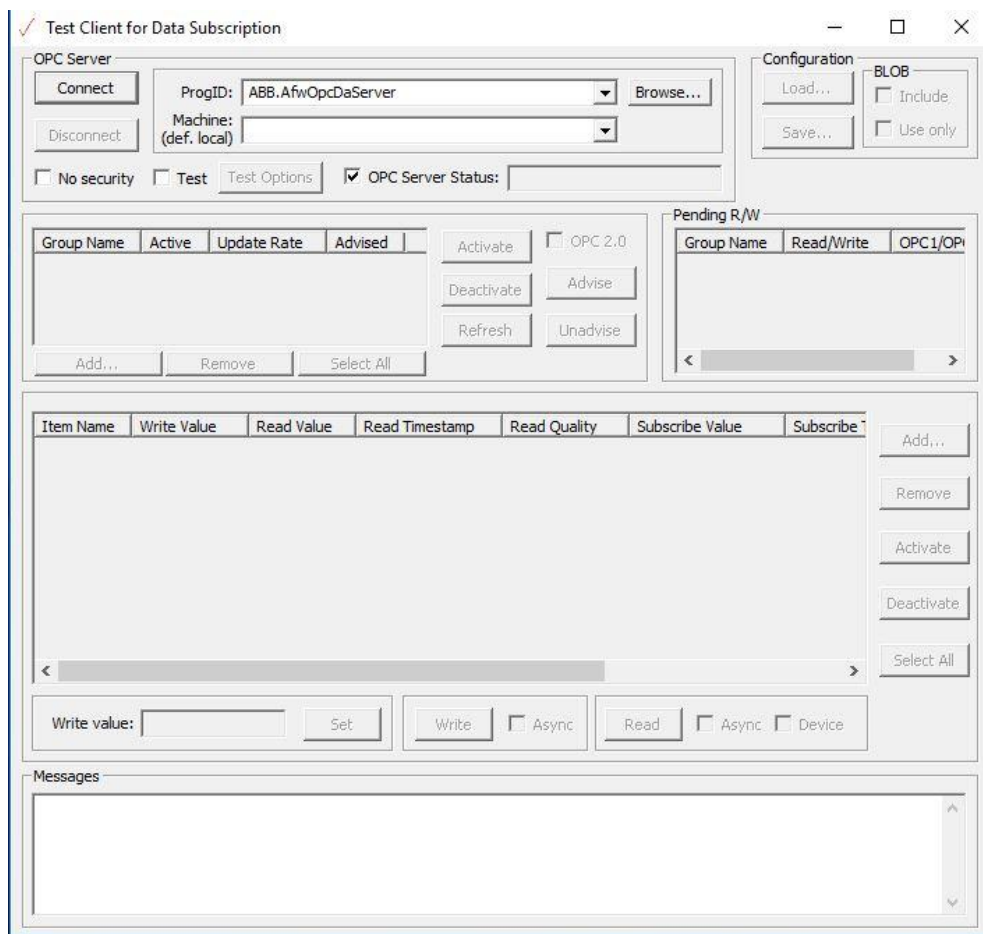
Kun kaikki asennuspaketit valvomo- ja ohjelmointiympäristölle oli asennettu ja todettu toimiviksi, voitiin aloittaa ohjelmien yhteensopivuuden selvittäminen. Tätä varten luotiin muuttujia ja lyhyt ohjelmapätkä logiikalle, sekä muutama indikoiva symboli valvomonäytölle.

Logiikan ja valvomon kommunikointiyhteyden luominen ei kuitenkaan ollut helppoin osa työtä. Yhteys vaati toimiakseen gateway-ohjelman, joka toimii porttina ohjelmien välillä. Tässä kyseisessä järjestelmässä porttina toimii CoDeSys Gateway, joita perusasennuksen yhteydessä tietokoneelle asentui 2 kappaletta, versio 2.3 ja versio 3. Näiden kahden ohjelman ero ei ollut nopealla silmäyksellä havaittavissa, vaan vaati syvempää tutkiskelua. Gateway-ohjelma valitaan sen perusteella, mitä perusversiota logiikka käyttää. Tässä tapauksessa logiikan perusversio oli 2.3, joten käytettäväksi porttiohjelmaksi tuli gateway 2.3. Gatewayn tulisi löytää logiikka automaattisesti, kun se on kytketty samaan verkkoon. Näin ei kuitenkaan käynyt, sillä molempien gateway-ohjelmien ollessa asennettuna ei yhteys toimi. Gateway 3:n asennus poistettiin, ja näin saatiin yhteys Automation builderin ja Gatewayn välille.



**Kuva 6.** OPC-konfigurointiohjelma.

Kun yhteys kahden edellä mainitun ohjelman välille oli saavutettu, jäi jäljelle yhteyden muodostaminen Gatewayn ja CompactHMI:n välille. Tämä olikin työn haastavimpia kohtia, sillä yhteyttä ei saatu muodostettua, vaikka kaikki tehtiin ohjekirjan mukaisesti. Useiden tuntien ponnisteluiden jälkeen vaikutti siltä, että näiden ohjelmien, ja koko järjestelmän yhdistäminen ei tule onnistumaan. CompactHMI:n puolella vaikutti siltä, että ohjelma ei osaa ollenkaan kommunikoida gatewayn kanssa. Asiaa edelleen tutkittaessa, löytyi CompactHMI asennuspaketin mukana tietokoneelle asentunut AdvDSOPCCClient-ohjelma. Kyseinen ohjelma oli hyvin piilotettu usean kansion sisälle. Tämä löytynyt ohjelma täytyi vielä käynnistää Windows 7 yhteensopivuustilassa, jotta se suostui toimimaan.



**Kuva 7.** AdvDSOPCCClient-ohjelma.

AdvDSOPCCClient-ohjelman avulla pystyttiin havaitsemaan, että CompactHMI:n yhteyden luomiseen käyttämä OPC connector havaitsee ja tunnistaa CoDeSysin OPC-serverin, ja tästä voitiin päätellä, että yhteyden luominen CompactHMI:n puolelle on mahdollista. CompactHMI:ssä etsitään CoDeSysillä luotuja muuttujia CHMI OPC Browserilla, ja kun kaikki asetukset ja yhteydet oli kohdallaan, löysi se muuttujat, joita nyt voitiin käyttää valvomon päässä. Yhteyden toiminnan varmistus tehtiin vielä antamalla muuttujille eri arvoja, ja toteamalla niiden näkyminen niin ohjelmointiympäristössä kuin myös valvomossa.



## 2.5 Taustaohjelmat

Edellä mainittujen ohjelmien lisäksi järjestelmäkokonaisuus sisältää useita ohjelmia, jotka toimivat taustalla. Nämä ohjelmat ovat asentuneet asennuspakettien mukana ja ovat joillekin toiminnoille välttämättömiä. Lisäksi joukossa on toisia ohjelmia tukevia ohjelmakokonaisuuksia.

Asennuspakettien mukana tulevat ohjelmat eivät kaikki ole tarpeellisia kaikissa tilanteissa, mutta niitä ei ole suositeltavaa kuitenkaan poistaa, sillä niiden poissaolo saattaa haitata järjestelmäkokonaisuuden toimintaa tulevaisuudessa. Tällainen tilanne saattaa tulla, kun järjestelmäkokonaisuuden toimintaa päivitetään ja muutetaan, esimerkiksi muuttuneen laitekannan takia. Poistettujen ohjelmien takaisin saaminen saattaa joissain tilanteissa olla haastavaakin, ja pahimmassa tapauksessa voidaan järjestelmä joutua asentamaan uudelleen alusta alkaen.

Lisäksi on huomioitava sellaiset ohjelmat, joiden asennus saattaa haitata jonkin toisen ohjelman toimintaa. Tällaisia tapauksia saattaa ilmaantua, jos samasta ohjelmasta asennetaan useampi kuin yksi versio. Tällaisesta tilanteesta hyvänä esimerkkinä toimii tässä opinnäytetyössä vastaan tullut tapaus, jossa asennuspaketin vakioasennuksessa asentui kaksi Gateway-ohjelmaa. Nämä päällekkäisesti toimivat ohjelmat estivät toinen toistaan toimimasta. Tästä tilanteesta ei kumpikaan ohjelma kuitenkaan antanut minkäänlaista varoitusta ja päällepäin vaikuttikin siltä, että ohjelmat toimisivat aivan kuten niiden kuuluukin.

Lisäksi taustaohjelmista saattaa löytyä sellaisia ohjelmia, joiden tarkoituksena on ratkaista ongelmia, tai ainakin olla apuna ongelmia ratkottaessa. Näiden ohjelmien hyödyllisyys ei välttämättä tule esille vielä asennus- tai käyttöönottovaiheessa. Toisaalta näitä ohjelmia voidaan joutua käyttämään heti alusta alkaen, jotta järjestelmä saadaan toimivaksi. Tässäkin opinnäytetyössä jouduttiin turvautumaan taustaohjelmaan, joka auttoi ratkaisemaan ongelman. Tämä ohjelma oli AdvDSOPCCClient, joka löytyi CompactHMI-järjestelmän mukana asentuneesta kansioista. Ohjelma oli piilotettu usean kansion taakse. Ohjelmalla päästiin tutkimaan järjestelmän OPC-yhteyksiä, ja todentamaan yhteyksien olemassaolo ja toiminta.

Ongelmien ilmetessä voidaan myös joutua turvautumaan kolmansien osapuolien tarjoamiin ohjelmiin, joilla voidaan ongelma ratkaista, tai ainakin edesauttaa niiden ratkeamista. Osa tällaisista ohjelmista on ilmaisohjelmia, kun taas toiset vaativat lisenssin. Tällaisia ohjelmia saattaa myös löytyä julkisen jakelun ulkopuolelta. Tällaiset ohjelmat ovatkin yleensä tarkoitettu yrityksen sisäiseen käyttöön, eikä niitä anneta ulkopuolisille käyttöön.

## 2.6 Virtuaalitietokone

Tässä opinnäytetyössä käytettiin tietokoneeseen asennettua virtuaalitietokonetta, jotta järjestelmän asennus voidaan suorittaa puhtaaseen tietokoneeseen. Virtuaalitietokoneen käyttämiseen käytettiin Oraclen VM Virtualbox-ohjelmaa, johon asennettiin Windows 10-käyttöjärjestelmä.

Virtuaalitietokoneen käyttäminen tällaisessa tutkimukseen ja testaamiseen tähtäävässä projektissa on hyvä menetelmä, sillä testauksen tai tutkimuksen valmistuttua voidaan virtuaalitietokone tuhota ilman, että siitä jää jälkeäkään tietokoneeseen. Mikäli tässä opinnäytetyössä tutkimisen aiheena olleen järjestelmän laajuinen kokonaisuus yritettäisiin hävittää tietokoneesta poistamalla se, jäisi siitä kuitenkin erinäisiä ohjelmia ja ajureita sekä asetuksia Windowsiin. Tämä saattaisi aiheuttaa tulevaisuudessa uusissa asennettavissa ohjelmissa erinäisiä ongelmia ja ristiriitoja. Tällaiset ongelmat voisivat pahimmassa tapauksessa väärentää tutkimusten ja testausten lopputuloksia.

Ennen virtuaalitietokoneen käyttöönottoa tulee sen toimintaan kuitenkin ensin perehtyä. Virtuaalitietokone sisältää myös paljon erilaisia asetuksia ja säätöjä, joiden toiminta ja tarkoitukset olisi hyvä ymmärtää ennen virtuaalitietokoneen käyttöä. Asetuksia voidaan myös joutua säätämään ja vaihtamaan virtuaalitietokoneen ollessa käytössä. Tällaisesta tapauksesta toimii esimerkkinä tätä työtä tehdessä vastaan tullut tilanne. Siinä virtuaalitietokoneen verkkoasetuksia tuli muuttaa niin, että asennusvaiheessa virtuaalikone käyttää pohjatietokoneen langatonta verkkokorttia internetyhteyden luomiseksi, ja järjestelmän käyttöönotto ja testausvaiheessa virtuaalitietokone käyttää pohjatietokoneen Ethernet-porttia yhteyden saamiseksi ohjelmoitavalle logiikalle.

Myös virtuaalitietokoneen tallennus ja siirtotarpeet tulee huomioida ja esimerkiksi USB-muistitikkujen käyttö vaati virtuaalitietokoneen asetuksiin kajoamista. Tiedostojen siirto virtuaalitietokoneen ja pohjatietokoneen välillä ei asetusten oikeellisuudesta huolimatta aina toimi sujuvasti. Tällöin on helpointa käyttää juurikin USB-muisteja tiedostojen siirtoon.

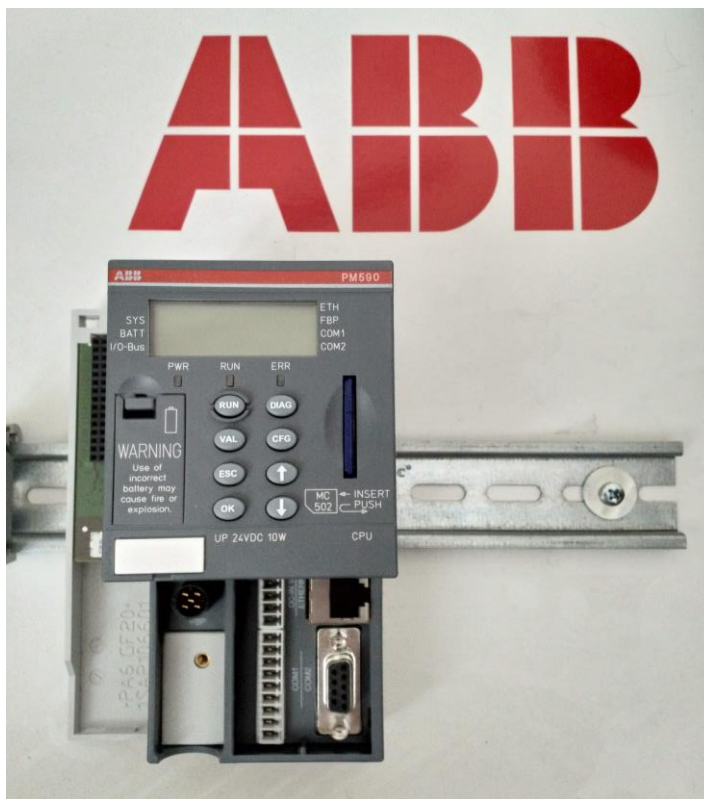
Virtuaalitietokoneiden käytössä on myös muita hyötyjä. Luotu virtuaalitietokone voidaan tarvittaessa siirtää helposti myös toiseen pohjatietokoneeseen. Tästä saadaan hyötynä esimerkiksi se, että mikäli tutkimusympäristö halutaan siirtää toiseen tietokoneeseen tai sen fyysistä sijaintia halutaan vaihtaa, voidaan se toteuttaa ilman suurempia ponnisteluja. Lisäksi virtuaalitietokonetta voidaan käyttää suoraan ulkoisesta muistista, esimerkiksi ulkoiselta kovalevyltä. Näin se voidaan siirtää helposti ja nopeasti uuteen paikkaan ja uuteen pohjatietokoneeseen.

### 3 LAITTEISTO

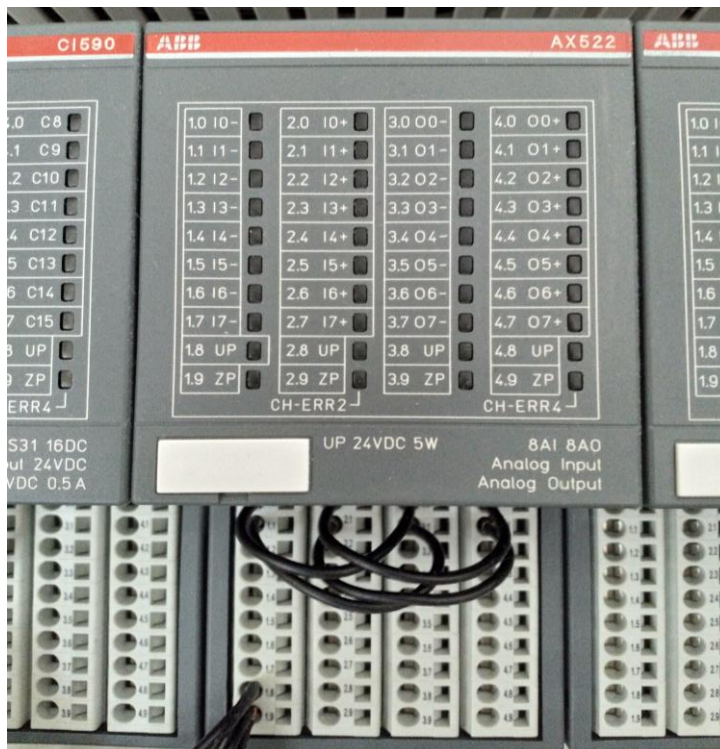
Ohjelmistot eivät yksinään saa laitosta toimimaan, vaan siihen tarvitaan myös laitteisto, jonka kautta ohjaukset välittyvät prosessilaitteille, kuten venttiileille ja moottoreille. Nämä ohjauslaitteet valitaan aina sen mukaan, mitä projektilta vaaditaan. Isommassa projektissa on useampia laitteita.

Tässä opinnäytetyössä laitteistona toimii ABB:n AC500 ohjelmoitava logiikka, sekä siihen liitetyt CI590- ja AX522-kortit (**Kuvat 9 ja 10.**). Nämä riittävät varsin hyvin tällaiseen pieneen projektiin, jossa ei tarvita useaa tulo- ja lähtöliitintä tai toimintoa. Ohjattavia laitteita ei tässä työssä ole, joten korttien lähdöt on kytketty saman kortin tuloliittimiin. Kokonaisuus on asennettu purulevyyn, jotta sen liikuteltavuus olisi helppoa. Lisäksi pohjalevyssä on myös 24VDC virtalähde logiikalle ja korteille sekä lisäkorteja, mikäli niille olisi ollut tarvetta.

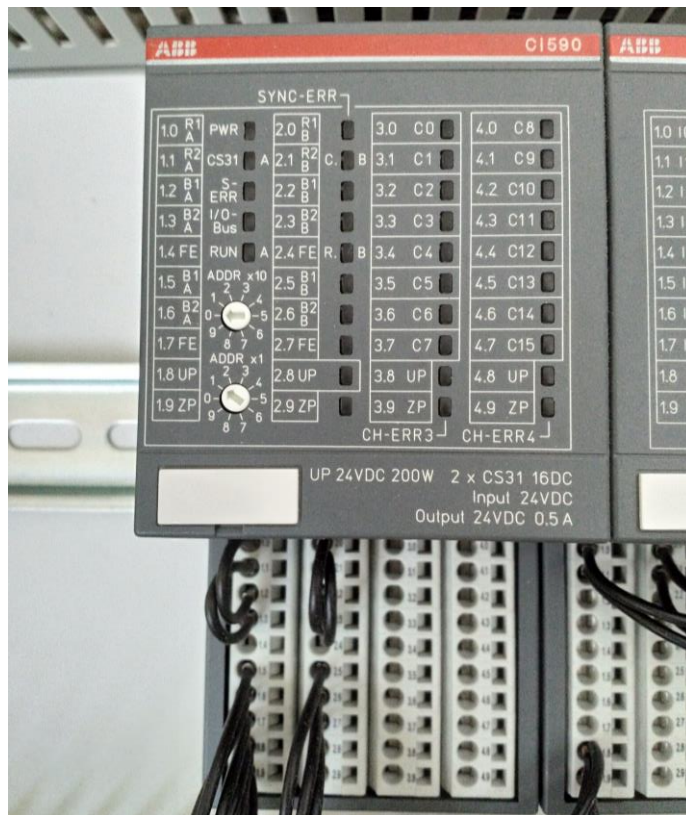
AC500 on hyvin pieneen tilaan mahtuva edullisempi vaihtoehto yleisemmin käytetylle AC800M-ohjaimelle. AC500 sopii paremmin pienempiin projekteihin, joskin sillä on myös toteutettu laajempia kokonaisuuksia, joissa on mukana useampia AC500-logiikoita, ja paljon IO:ta. AC500 soveltuu mainiosti esimerkiksi erilaisiin simulointi- ja testaustehtäviin. Tällaisiin projekteihin sitä on myös useasti käytetty.



**Kuva 8.** AC500-logiikkaohjain.



**Kuva 9.** AX522-analogiakortti.



**Kuva 10.** CI590-digitaalikortti.

## 4 TIETOTURVA

Nykyaikana, kun lähes kaikki laitteet pyritään saattamaan saavutettavaksi verkon välityksellä, tulee muistaa suojautumisen tärkeys. Kun jokin laite saatetaan saavutettavaksi verkon kautta, tarkoittaa se sitä, että myös muut henkilöt tai ryhmät voivat saavuttaa tämän laitteen verkon välityksellä. Tämä asia ei aina kuitenkaan ole toivottua, etenkin jo henkilöllä tai ryhmällä on mielessään laitteen väärinkäyttö. Tällöin laitteen asianmukainen käyttö saatetaan estää, tai saatetaan asettaa laite toimimaan oikeaa haltijaansa vastaan. Tähän ongelmaan on kuitenkin kehitetty ja jatkuvasti kehitetään erilaisia keinoja suojautua. Yleisimpiä näistä suojauksista ovat palomuurit ja viruksentorjuntaohjelmat.

Isommat järjestelmät pyritään myös saamaan verkkoon, jotta niiden käyttäminen helpottuisi. Tällaisessa järjestelmässä kuitenkin on erittäin toivottavaa, ettei siihen pääse käsiksi verkon kautta sellaiset henkilöt tai ryhmät, jotka haluavat aiheuttaa vahinkoa järjestelmään tai sen ohjaamaan laitokseen. Tällaisia hyökkäyksiä on myös tapahtunut, kun suojaus ei ole ollut kunnossa. Mikäli kyseessä olisi laitos, joka tuottaa sähköä tai ohjaa sen jakelua, saattavat verkkohyökkäyksen vahingot ulottua paljon laajemmalle kuin vain itse laitokseen.

Kun kyseessä on teollisuuslaitos, on sen suojaaminen viety myös korkeammalle tasolle, ja näin väärinkäyttöjen aikaansaamista vaikeutettu. Teollisuuteen on saatavilla erilaisia ratkaisuja verkkojen suojaamiseen, mutta niitä ei tässä opinnäytetyössä käydä läpi. Tämän luvun tarkoituksena on käsitellä tietoturva-asiaa vain pintapuolisesti, ja herättää lukija tutkimaan asiaa tarkemmin. Tässä opinnäytetyössä ei myöskään asennettu tai toteutettu minkäänlaista suojausta, sillä tätä järjestelmää ei tulla ottamaan käyttöön, eikä liittämään verkkoon, joten se katsottiin tarpeettomaksi.

Kun laitos on asianmukaisesti suojattu, voidaan sitä pitää verkossa ilman huolta siitä, että väärinkäytöksiä esiintyisi. Kuitenkin koko ajan kehittyvien hyökkäystapojen kehittäminen on syytä muistaa, että myös suojauksen on sen mukana kehityttävä. Tämä tarkoittaa sitä, että myös laitoksen suojauksen on kehityttävä ja sitä on päivitettävä. Laitoksen suojausta suunniteltaessa onkin syytä pohtia sitä, milloin

laitoksen suojaus tarkastetaan ja mahdollisesti päivitetään seuraavan kerran. Suojauksen ajan tasalla pitämiseen onkin syytä luoda jonkinlainen suunnitelma.



## 5 JÄRJESTELMÄKOKONAISUUS

Isot tuotantolaitokset vaativat toimiakseen ohjausta ja säätöä. Ennen nämä säädöt tehtiin käsin, mutta nykyään siihen käytetään automaatiojärjestelmiä. Automaatiojärjestelmä hoitaa suurimman osan säätämisestä ja ohjauksista itsenäisesti, sille ohjelmoidulla tavalla. Kuitenkin joitain ohjauksia ja säätöjä halutaan aina, tai tiettyissä tapauksissa tehdä manuaalisesti. Jotta tämä olisi mahdollista, täytyy automaatiojärjestelmään luoda lisäksi joku järkevä esitystapa sen tekemille toimille. Tähän käytetään valvomojärjestelmiä.

Valvomojärjestelmällä on paljon muitakin ominaisuuksia kuin pelkästään manuaalisten ohjausten tekeminen. Valvomojärjestelmällä voidaan valvoa esimerkiksi lopputuotteen laatua ja määrää sekä seurata laitoksen eri laitteiden tilaa. Esimerkiksi viallisesta laitteesta voidaan ohjata valvomoon ilmoitus, ja näin saattaa vika tietoon. Vian laadun mukaan voidaan ilmoitukselle asettaa erilaisia huomiota herättäviä toimintoja, kuten valo- ja äänimerkkejä. Näiden toimintojen lisääminen nopeuttaa vian huomaamista ja korjaamista.

Tässä opinnäytetyössä esitetyillä laitteilla, työkaluilla ja menetelmillä voidaan luoda järjestelmäkokonaisuus, jolla voidaan käyttää, ohjata ja valvoa kokonaista laitosta tai sen osaa.

### 5.1 Valmis järjestelmä

Valmiissa järjestelmässä kaikki sen osat toimivat yhtenä kokonaisuutena, johon optimaalisessa tilanteessa ei käyttöönoton jälkeen tarvitse tehdä muutoksia. Käyttäjä, joka tällaisessa ympäristössä on yleisesti valvomohenkilökunta, käyttää järjestelmässä pääasiallisesti ainoastaan graafista valvomoympäristöä. Graafinen ympäristö suunnitellaan siten, että sen käyttäminen olisi mahdollisimman selkeää ja helppoa, mutta kuitenkin asiakkaan toiveita kuunnellen.

Yleisesti graafinen ympäristö vastaa hyvin paljon tehtaan tai laitteen ympäristöä, jolloin myös vianetsintä kentällä helpottuu. Lisäksi graafisessa ympäristössä

navigointi pyritään toteuttamaan siten, että se vastaisi todellisuutta, ja olisi näin looginen ja helppokäyttöinen.

Koska asennettu järjestelmä on laaja ja monimutkainen, voidaan se määrittää niin, että valvomohenkilökunnan kirjautuessa sisään omalla käyttäjällään, avautuu heille suoraan näkymä valvomojärjestelmään. Näin heidän ei tarvitse käyttää aikaa oikean ohjelman löytämiseen, jonka lisäksi estetään tahattomat väärin ohjelmien aukaisut. Tässä opinnäytetyössä ei kuitenkaan luotu erikseen valvomohenkilöille omaa käyttäjätiliä, sillä sitä ei katsottu tarpeelliseksi.

## 6 VIKATILANTEET

Tällaisessa järjestelmässä sellaiset vikatilanteet, jotka johtuvat jonkin komponentin rikkoutumisesta ovat harvassa. Kuitenkin sellaisia voidaan olettaa tapahtuvan jossain vaiheessa järjestelmän elinkaarta. Yleisempiä ovat viat, jotka johtuvat väärästä toimintatavasta tai käyttäjän virheestä. Näitä ongelmia ei voida poistaa, mutta niitä voidaan pyrkiä vähentämään sekä estämään. Lisäksi niiden vaikuttavuutta voidaan vähentää, esimerkiksi logiikan ja sen korttien kohdalla nopeuttamalla komponenttien vaihtotyötä.

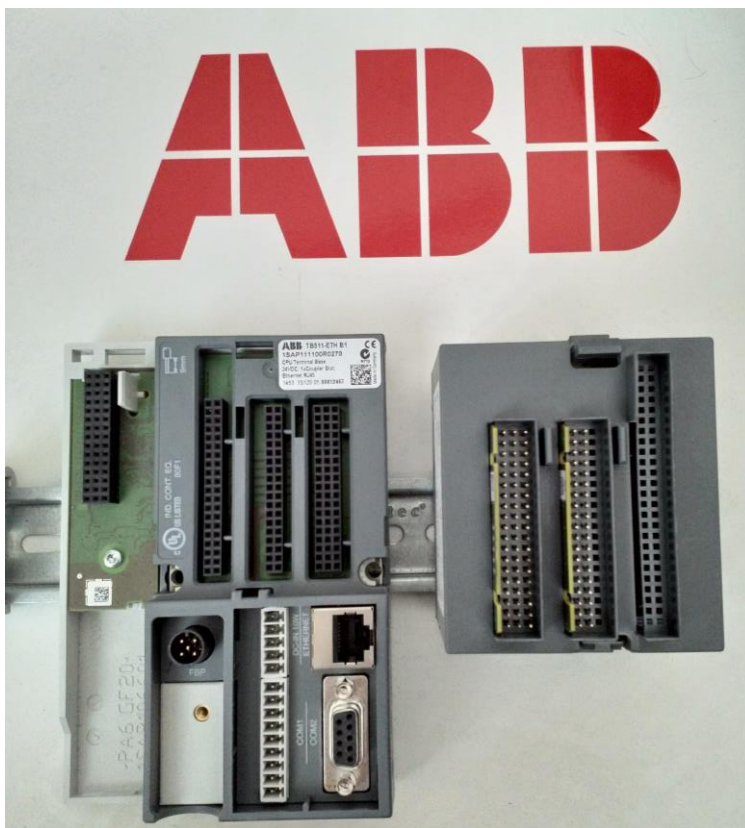
AC500-logiikan tai sen korttien vioittuessa voidaan ne korvata ilman suurempia purkutöitä, sillä logiikka ja kortit asennetaan pohjalevyille, joita ei tarvitse poistaa vaihdosta tehdessä. Tämä nopeuttaa huomattavasti vaihtotyötä ja näin lyhentää vian kestoaikaa.

AC500-logiikkaan liitetyt kortit voidaan vian tullessa korvata ilman, että niille tulevaa johdotusta tarvitsee purkaa. Tämä nopeuttaa ja helpottaa korttien vaihtotyötä huomattavasti.



**Kuva 11.** Kortin vaihto pohjalevyyn.

Myös logiikan vaihtaminen on tehty helpoksi ja nopeaksi. Mikäli logiikka vioittuu ja täytyy vaihtaa, voidaan siitä irrottaa SD-kortti, joka sisältää ohjelmat, joita logiikka ajaa. Kun vanha logiikka on irrotettu ja uusi asennettu tilalle, voidaan SD-kortti syöttää takaisin logiikkaan ja näin logiikka alkaa jälleen suorittaa ohjelmaa joka kortilla on. Tämä tietenkin edellyttää sitä, että alkuperäistä ohjelmaa tehdessä se talletetaan myös SD-kortille, eikä pelkästään logiikan sisäiseen muistiin. SD-kortin käyttämisestä saadaan muitakin hyötyjä kuten se, että ohjelma voidaan tarpeen vaatiessa nopeasti vaihtaa logiikalle vain SD-korttia vaihtamalla.



**Kuva 12.** Logiikka irrotettuna pohjalevystä.

## **7 LOPPUTULOS JA TULOSTEN VERTAILU**

### **7.1 Opinnäytetyön lopputulema**

Ensimmäisen palaverin jälkeen muotoutunut suunnitelma ei tässä opinnäytetyössä toteutunut. Suurin syy siihen oli yksittäisten ongelmien vaatima suuri työmäärä. Ongelmien ratkominen kuitenkin on oleellinen osa oikeaa työelämää, ja tältä kantilta katsottuna opinnäytetyötä tehdessä vastaan tulleet ongelmat olivat ainoastaan työelämään valmistavia haasteita. Näistä haasteista huolimatta opinnäytetyö pystyttiin saattamaan järkevään lopputulokseen, jota voidaan työn tilanteessa yksikössä hyödyntää heidän parhaaksi katsomallaan tavalla.

Vaikka opinnäytetyössä tehtävät tutkimukset ja tarkastelut eivät laajuudessaan vastaa ensimmäisiä luonnoksia, ei työtä välttämättä kannata katsoa epäonnistuneeksi, sillä tuntemattoman asian laajuutta arvioidessa osuu arvio usein väärään. Tämän opinnäytetyön laajuutta arvioitaessa ei osattu ottaa huomioon niitä haasteita, jotka söivät eniten aikaa työtä tehdessä.

### **7.2 Tulosten vertailu**

Opinnäytetyössä saatuja tuloksia voidaan pitää selvinä. AC500-logiikkaa verrattaessa AC800M-ohjaimen voidaan todeta, että AC500:n ohjelmoiminen on huomattavasti työläämpää kuin AC800M:n. Tämä johtaa siihen, että projektien ohjelmointiajat pitenevät, mikä taas lisää kustannuksia. Toisaalta, AC500 ei sovi suurin projekteihin, joissa on suurempia vaatimuksia ja odotuksia logiikalta. Näin ollen pienemmissä projekteissa ohjelmoinnin vaatiman ajan erot pienevät. Näiden kahden laitteen ohjelmoinnin nopeusero johtuu siitä, että niiden ohjelmointiin käytettävät ohjelmistot ovat täysin erilaisia.

AC800M:lle tarkoitettu control builder on AC500:lle tarkoitettuun codesysiin verrattuna huomattavasti käyttäjäystävällisempi ja selkeämpi. Control builderissa kaikki tarvittava on yhdessä paikassa ja pelkällä vilkaisulla voi saada hyvän kuvan laitteiston ja ohjelmien määrästä ja tyyppistä. Codesysissä taas kaikki on hajautettu,

eikä esimerkiksi samalla vilkaisulla voi saada kattavaa kuvaa sekä laitteistosta että ohjelmista.

Lisäksi AC800M:n kannalta parempi ominaisuus on se, että ohjeet ovat huomattavasti yksityiskohtaisempia sille kuin AC500:lle. Niitä vertaamalla vaikutti myös siltä, että yhteyden muodostaminen AC800M:lle olisi yksinkertaisempaa. Tämä puolestaan helpottaisi ja nopeuttaisi projektien käyttöönottoa.

Eroavaisuudet eivät kuitenkaan rajoitu pelkästään ohjelmallisiin, vaan myös fyysisiin eroihin. AC500 on selkeästi pienemmän ja siromman kokoinen, mikä taas tarkoittaa sitä, ettei sille tarvitse varata yhtä suurta tilaa kuin AC800M:lle. Kokoero ei rajoitu ainoastaan itse logiikkaan, vaan myös siihen liitettävät kortit ovat huomattavasti isompaa mallia pienempiä joka suunnassa. Kuitenkin laitteisto sijoitetaan yleensä sille varattuun erilliseen kaappiin, ja taloudellisesti ajateltuna ei AC800M:n vaatiman isomman kaapin hintaero ole kokonaisprojektin mittakaavassa huomattava.