

Nelli Hintsala

PROSESSIOHJAUKSEN AUTOMAATIOJÄRJESTELMÄN VAIHTO

PROSESSIOHJAUKSEN AUTOMAATIOJÄRJESTELMÄN VAIHTO

Nelli Hintsala
Opinnäytetyö
Syksy 2023
Sähkö- ja automaatiotekniikan tutkinto-
ohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Sähkö- ja automaatiotekniikan tutkinto-ohjelma, Automaatiotekniikka

Tekijä: Nelli Hintsala

Opinnäytetyön nimi: Prosessiohjauksen automaatiojärjestelmän vaihto

Työn ohjaaja: Manne Tervaskanto

Työn valmistumislukukausi ja -vuosi: Syksy 2023

Sivumäärä: 46 + 4 liitettä

Opinnäytetyön tavoitteena oli toteuttaa prosessiohjauksen automaatiojärjestelmän vaihto. Työn tilaajana toimii Apex Automation Oy. Toteutus liittyy tilaajan asiakasprojektiin. Projektissa nykyisen järjestelmän kaksi logiikkaa sekä näiden ja hajautus-I/O:n väliset väyläkommunikaatiotyökalut korvataan uusilla nykyaikaisilla komponenteilla. Väyläkommunikaatiotyökalujen vaihton myötä kommunikointiprotokolla vaihtuu, mikä vaatii uuteen väyläratkaisuun perehtymistä sekä sen testausta. Työssä perehdytään siihen, mitä prosessiohjauksen automaatiojärjestelmän vaihto vaatii toteutukselta, laitteistolta sekä ohjelmistoilta.

Teoriaosuudessa esitellään käytetyt ohjelmistot, nykyisen järjestelmän laitteet sekä niiden korvaavat laitteet. Perehdytään myös nykyisen ja uuden kokoonpanon väyläratkaisuihin. Nykyisen järjestelmän laitteisto koostuu ABB:n sekä Alfa Lavalin tuotteista ja uudet korvaavat laitteet tulevat Siemensiltä sekä Allen-Bradleyltä.

Toteutusosa on jaettu kahteen osaan: esisuunnitteluun ja kommunikoinnin testaukseen. Esisuunnitteluvaiheessa perehdytään toteutettavaan työhön, kerätään lähtötiedot, pohditaan projektin haastavia kohtia sekä tarkistetaan valittujen komponenttien yhteensopivuus. Kommunikoinnin testauksessa testataan testikokoonpanon avulla korvaavien laitteiden välisen kommunikoinnin ja datansiirron toimivuus.

Edellä mainituissa testauksissa ilmeni haasteita, joiden seurauksena Ethernet/IP-protokollaa tutkittiin syvällisemmin kuin projektin alussa ajateltiin. Testien tulosten analysoinnissa avataan käytettävää protokollaa kommunikoinnin näkökulmasta sekä vertaillaan datapaketteja ja sanomarakenteita eri testauskokoonpanojen välillä.

Testauksessa Siemensin Ethernet/IP-lohkokirjastosta löytyi datasegmenttipuutos, joka vaikutti uuden modulaarisen kokoonpanon kommunikoinnin toimimiseen. Tähän puutokseen on kehitteillä uusi lohkokirjasto, joka on tarkoitus julkaista seuraavan vuoden sisällä applikaatioesimerkinä. Aikataulusiististä päivitetty kirjastoversio ei kuitenkaan ehdi tämän projektin tarpeisiin, jonka vuoksi korvaavan logiikan tuotemerkkiä jouduttiin vaihtamaan.

Tämän opinnäytetyön aikataulun puitteissa siis ehdittiin keräämään tarvittavat lähtötiedot ja toteutamaan kommunikoinnin testaukset sekä ratkaistua siinä ilmenneet haasteet. Varsinainen toteutus eli suunnittelu- ja ohjelmointityö jäi myöhempään ajankohtaan. Kuitenkin tämän seuraavan työväiheen työjärjestys on kuvattu opinnäytetyön viimeisessä luvussa.

Asiasanat: automaatio, automaatiojärjestelmät, ohjelmoitavat logiikat, väyläkommunikaatio

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Electrical and Automation Engineering, Option of Automation Engineering

Author: Nelli Hintsala
Title of thesis: Process automation system replacement
Supervisor: Manne Tervaskanto
Term and year when the thesis was submitted: Autumn 2023
Number of pages: 46 + 4 appendices

The objective of this thesis is to design and accomplish a replacement of the process automation system. There are two PLC (Programmable logic controller) in current assembly which are replaced by one new PLC. In addition to this, bus communication is changed.

At the beginning of the project, initial data is collected from the current system documents and new communication bus testing is prepared. Different softwares for hardware configuration, programming and data analysis is used in testing. Testing had to be extended due to incompatible communication node and configuration library for Ethernet/IP.

As a result of the testing founded one missing data segment which send configuration to the node in Siemens Ethernet/IP scanner library. That simple data segment is optional but in case like this where we have a modular assembly that data segment is needed.

Due to the tests there was not enough time to complete all the objectives of the thesis. The actual programmig part of the project will be moved to a later date.

Keywords: automation, automation system, bus communication

SANASTO

CIP	Common Industrial Protocol
EDS file	Electronic Data Sheet file = Standardisoitu laitekuvaus-tiedosto
FBD	Function Block Diagram = Graafinen ohjelmointikieli
GSD file	General Station Description file
HMI	a Human-Machine Interface
HW	Hardware configuration
IEEE	Institute of Electrical and Electronics Engineers
LAD	Ladder logic = Tikapuulogiikka ohjelmointikieli
MAC	Media Access Control
OSI	Open System Interconnection = OSI-malli
PLC	Programmable logic controller = Ohjelmoitava logiikka-ohjain
SCL	Structured Control Language = Struktuuriteksti ohjel-mointikieli
TCP	Transmission Control Protocol

SISÄLLYS

1	JOHDANTO	7
2	PERUSTIEDOT PROJEKTISTA.....	8
2.1	Apex Automation Oy	8
2.2	Järjestelmävaihdon tavoitteet	8
2.3	Käytettävät ohjelmistot	9
2.3.1	TIA Portal V17	9
2.3.2	RSLogix5000	10
2.3.3	Wireshark.....	11
3	AUTOMAATIOJÄRJESTELMÄN LAITTEET.....	13
3.1	Projektin ohjelmoitavat logiikat	13
3.2	Käytetyt kenttäväylät	16
3.3	Hajautetut I/O-yksiköt	18
4	ESISUUNNITTELU.....	21
4.1	Lähtötietojen kerääminen	21
4.1.1	I/O-lista	21
4.1.2	Apumuistipaikat.....	22
4.1.3	Ohjelmalohkot ja niiden toiminnallisuudet	22
4.2	Haastavien kohteiden kartoitus ennen suunnittelutyön aloitusta	24
4.3	Laitetietojen tarkistus.....	25
5	KOMMUNIKOINNIN TESTAUS	27
5.1	Testauksen valmistelu	27
5.2	Testit	27
5.3	Testien tulokset	32
5.3.1	Ethernet/IP-protokollan sisältö kommunikoinnin näkökulmasta	32
5.3.2	Protokollan datapakettien vertailut	34
6	JÄRJESTELMÄVAIHDON TOTEUTUS	41
6.1	Suunnittelu ja ohjelmointi.....	41
6.2	Jatkotoimenpiteet	42
7	YHTEENVETO	43
	LÄHTEET.....	44
	LIITTEET	46

1 JOHDANTO

Tämän opinnäytetyön tilaajana toimii Apex Automation Oy. Opinnäytetyön toteutus liittyy Apex Automation Oy:n asiakasprojektiin. Työn tavoitteena on toteuttaa prosessiohjauksen automaatiojärjestelmän vaihto asiakkaan tuotantolaitokseen. Käytännössä nykyiset logiikat ja väyläyhteydet korvataan nykyaikaisilla laitteilla, koskematta olemassa olevaan hajautettuun I/O:hon. Tämä tarkoittaa sitä, että ohjelma kirjoitetaan uusiksi sekä väyläyhteys testataan toimivaksi uuden logiikan ja nykyisen hajautus-I/O:n välillä.

Nykyisessä järjestelmässä on kaksi logiikkaa, jotka halutaan tuoda yhden logiikan alle hajautus-I/O:n avulla. Kahden logiikan korvaaminen yhdellä edellyttää ohjelmien yhdistämistä ja erityistä tarkkuutta sekä I/O:n että apumuistipaikkojen kanssa, jotta päällekkäisyyksiä ei tule uudessa ohjelmassa. Varsinaisen ohjelman lisäksi myös valvomosovellus uusitaan. Nykyisen järjestelmän paneeli, kytkimet ja valvomo kootaan yhteen samaan olemassa olevaan valvomoon, jotta operointi olisi selkeää. Asiakkaan toiveena oli siirtyä Siemens-järjestelmään, joten nykyisen järjestelmän logiikat sekä operointipaneeli korvataan uusilla Siemens-tuotteilla.

2 PERUSTIEDOT PROJEKTISTA

Työn tilaajan yritysesitys sekä opinnäytetyön tavoitteet ja käytettävät ohjelmistot määritellään seuraavissa luvuissa.

2.1 Apex Automation Oy

Apex Automation Oy on kokkolalainen automaatio- ja sähkösuunnitteluun erikoistunut insinööritoimisto, joka on perustettu vuonna 1993. Tällä hetkellä yritys työllistää noin 60 alan ammattilaista, joiden osaaminen ja tietotaito pohjautuu vuosien varrella tehtyihin monipuolisiin asiakasprojekteihin. Palveluihin kuuluvat automaatio-, sähköistys- ja robotiikkaratkaisut sekä koneturvallisuuden asiantuntijapalvelut ja ratkaisut. Asiakaskunta koostuu teollisuuden ja energia-alan yrityksistä sekä laite- ja järjestelmätoimittajista. (1.)

2.2 Järjestelmävaihdon tavoitteet

Järjestelmävaihtoa lähdettiin toteuttamaan vanhan laitekannan vuoksi, ja sen isoimpana tavoitteena on saada nykyiselle laitteistolle ainakin viisi vuotta lisää elinkaarta. Nykyisessä järjestelmässä on kaksi logiikkaa: Sattcon05 Slimline ja Sattcon 200 – CPU 30/20. Näiden elinkaari on siirtynyt rajoittuneeseen vaiheeseen yli kymmenen vuotta sitten, jonka vuoksi varaosia tai uusia samanlaisia komponentteja ei ole enää saatavilla. I/O-kortit ovat myös vanhoja, mutta asiakkaalla on itsellään varastossa uusia kortteja mahdollisten rikkoontumisien varalle, joten koko I/O-korttikannan vaihto ei ole tarpeellinen. Käyttöönoton tiukka aikataulu puoltaa myös laitekannan osittain uusimista, koska koko I/O-korttikannan vaihdon vaatima kytkentätyö olisi liian iso aikatauluun nähden.

Nykyisen kokoonpanon kaksi logiikkaa halutaan uudessa kokoonpanossa korvata yhdellä logiikalla hajautus-I/O:n avulla. Tämä tarkoittaa molempien logiikoiden sisältämien ohjelmien yhdistämistä yhdeksi kokonaisuudeksi. Logiikanvaihdon lisäksi yksi Alfa Lavalin HMI-ohjelmointipaneeli vaihdetaan uudeksi Siemensin paneeliksi. Lisäksi nykyiset paneeliohjelmat, kytkimet sekä valvomo koostaan yhdeksi valvomokokonaisuudeksi, jotta ohjelmointi olisi selkeää. Valvomon uusimisen myötä kommunikointi kentälaitteiden ja valvomon välillä täytyy saada toimivaksi.

Väyläkommunikointi on vanhassa järjestelmässä toteutettu Alfa Lavalin sekä ABB:n 200-ACN ControlNet -adaptoreilla. Tavoitteena on korvata ne Allen-Bradleyn Ethernet/IP-adaptoreilla, mikä asettaa monta huomioitavaa asiaa projektin edetessä.

2.3 Käytettävät ohjelmistot

Työn toteutukseen käytetään eri ohjelmistoja riippuen työ- ja testausvaiheista. Seuraavissa luvuissa on esitetty keskeisimmät ohjelmistot projektissa eli TIA Portal V17, RSLogix5000 sekä Wireshark. Edellä mainittujen ohjelmistojen lisäksi opinnäytetyön aikana käytetään muutamia muita ohjelmistoja apuna testauksessa. BootP-DHCP-Tool -ohjelmistoa käytetään Allen-Bradleyn Ethernet/IP-adapterin IP-osoitteen asettamiseen, RSLinx-ohjelmaa adapterin EDS-tiedoston lataamiseen sekä Microsoftin Exceliä lähtötietojen keräämiseen.

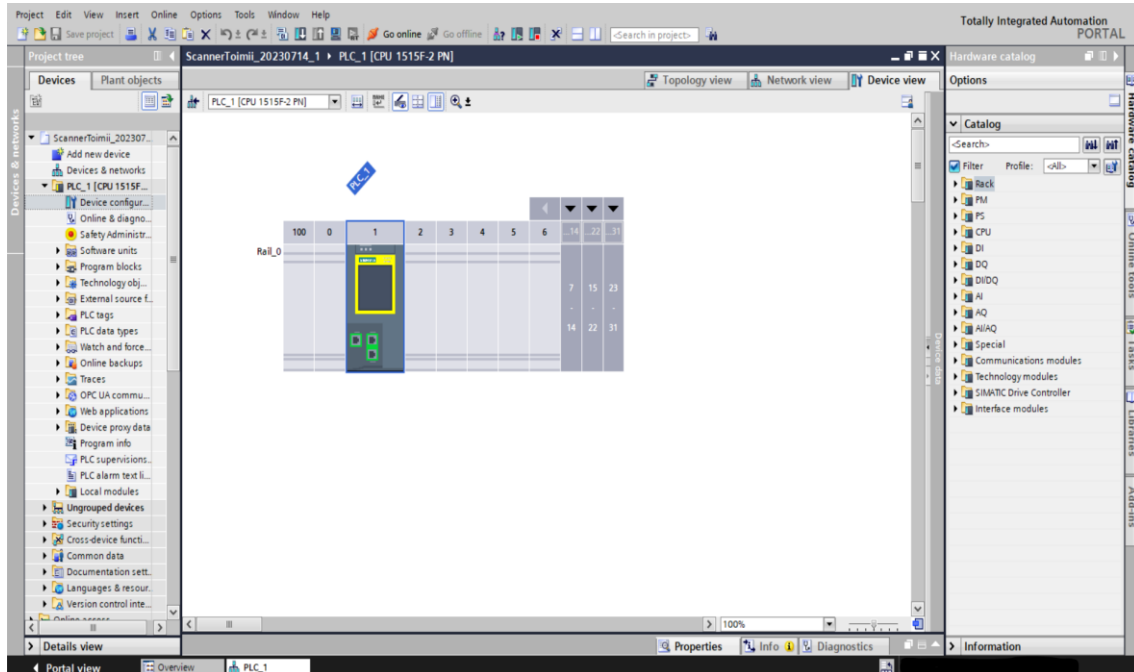
2.3.1 TIA Portal V17

TIA Portal eli Totally Integrated Automation Portal on Siemensin kehittämä ohjelmistoympäristö automaatiojärjestelmien konfigurointiin, suunnitteluun, ohjelmointiin ja ylläpitoon. Ohjelmisto tarjoaa samassa ohjelmassa PLC-ohjelmoinnin, HMI-suunnittelun, taajuusmuuttajien konfiguroinnin sekä liikkeenohjauksen. Ensisijaisesti TIA Portaalia käytetään Siemens-sarjan tuotteille, mutta ohjelmisto mahdollistaa myös ulkopuolisten laitteiden tuomisen ohjelmaan GSD-tiedostojen avulla.

TIA Portal -ohjelmisto tukee monia eri ohjelmointikieliä, joita ovat funktiolohkokaavio (FBD), tika-puulogiikka (LAD) ja struktuuriteksti (SCL). Siemens on kehittänyt laajat ohjelmalohkokirjastot ja -mallit sekä mahdollistaa myös omien kirjastojen luonnin. Tämä edesauttaa ohjelmakoodin uudelleenkäyttöä muissa projekteissa. Kirjastojen lisäksi yksi merkittävistä ominaisuuksista TIA Portalissa on sisäänrakennettu diagnostiikka sekä simulointimahdollisuus. Tämä mahdollistaa ohjelman testauksen ennen käyttöönottoa sekä helpottaa vianmääritystä. Muita ominaisuuksia ovat muun muassa online-muokkaus, käyttäjähallinta, etäyhteyksimahdollisuus sekä Multiuser-toiminto.

(2.)

Tässä opinnäytetyön projektissa TIA Portaalia käytetään väyläkommunikoinnin testauksessa sekä projektin myöhemmässä vaiheessa myös itse ohjelmoinnissa. Kuvassa 1 on TIA Portal V17 -ohjelmointiympäristö.



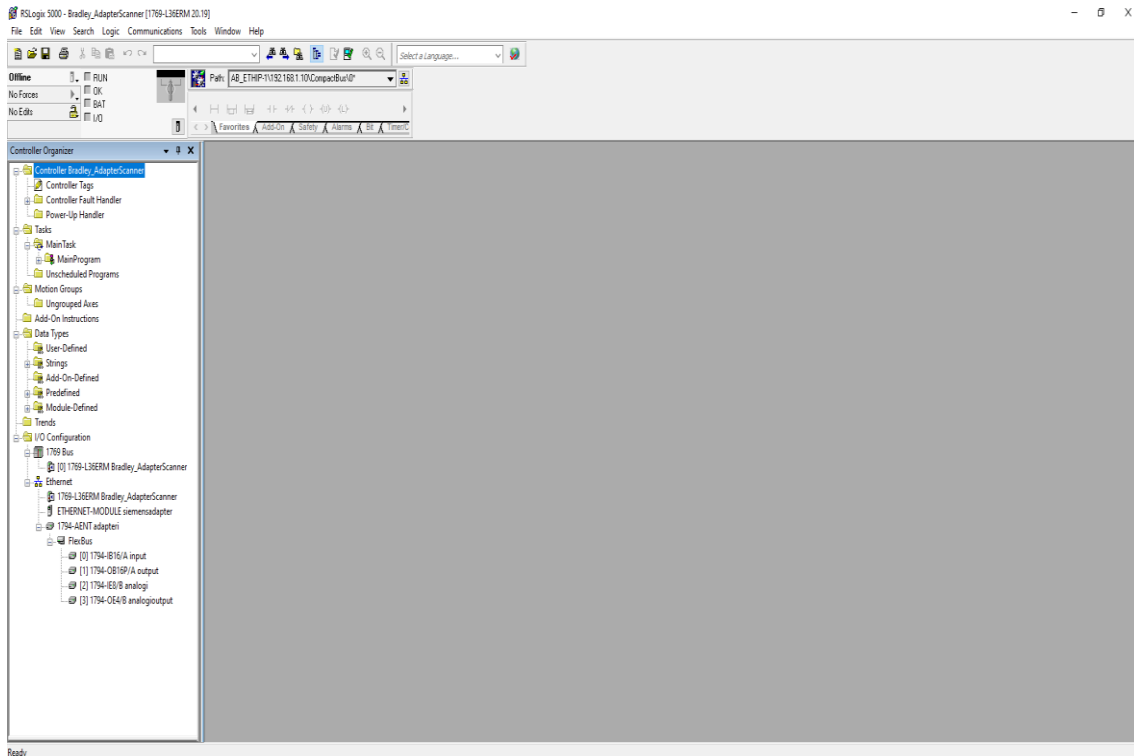
KUVA 1. TIA Portal V17

2.3.2 RSLogix5000

RSLogix5000 on Rockwell Automationin kehittämä ohjelmointiympäristö teollisten automaatiojärjestelmien ohjelmoitavien logiikoiden konfigurointiin sekä ohjelmointiin. Ohjelmisto tukee Logix5000-sarjan logiikoita eli ControlLogix, CompactLogix sekä SoftLogix.

RSLogix5000 mahdollistaa ohjelmoinnin funktiokaavio (FBD), tikapuulogiikka (LAD) ja struktuuriteksti (SCL) ohjelmointikielillä. Ohjelmisto sisältää perusominaisuuksien lisäksi, samoin kuin TIA Portal, diagnostiikka- ja simulointityökalut. RSLogix5000 myös tukee TIA Portaalin tavoin online-muokkausta eli ohjelmaa pystytään muokkaamaan käynnissä olevaan logiikkaohjelmaan pysäyttämättä prosessia. Se parantaa järjestelmän ylläpidettävyyttä sekä joustavuutta.

Tässä opinnäytetyön projektissa RSLogix5000-ohjelmistoa käytetään väyläkommunikoinnin testauksessa. Kuvassa 2 on RSLogix 5000 -ohjelmointiympäristö.



KUVA 2. RSLogix 5000

2.3.3 Wireshark

Wireshark on avoimen lähdekoodin analysointiohjelmisto tietoverkoille. Se ilmaisee yksityiskohtaisesti tietoverkkoliikenteen reaaliajassa. Liikenteen lukemisen pystyy pysäyttämään, jolloin saadaan tallennettua tiedosto ja analysoitua sekä tarkasteltua tätä myös myöhemmin. Data-analyysi näyttää protokollan tiedot, dataa lähettävien ja vastaanottavien laitteiden IP-osoitteet, datapaketit sekä niiden yksityiskohtaisen sisällön sekä tarjoaa käyttäjälleen monipuoliset suodatus- ja hakuominaisuudet. Ohjelmiston käyttökohteita ovat muun muassa vianetsintä ja analysointi tietoliikenteestä tai laitteiden välillä kulkevan datan selvitys. Wireshark tukee satoja eri protokollia sekä käyttöjärjestelmiä, joten se on monipuolinen ohjelmistotyökalu niin verkkoyhteyksien ylläpitäjille kuin tietoturva-verkkojen asiantuntijoille. Ohjelmisto on monien tietoliikenneasiantuntijoiden ylläpitämä, jonka vuoksi sen kehitys jatkuu koko ajan. (3.) Kuvassa 3 on yleisnäky Wireshark-ohjelmasta.

The screenshot displays the Wireshark interface with a list of network packets. Packet 321 is highlighted, showing details for Spanning-tree (for...) STP. The packet length is 60 bytes. The bottom pane shows the raw packet data in hexadecimal and ASCII format.

No.	Time	Source	Destination	Protocol	Length	Info
308	17.091416	192.168.1.10	192.168.1.222	CIP I/O	82	Connection: ID=0x00290101, SEQ=0000012053, O->T
309	17.091420	192.168.1.10	192.168.1.222	CIP I/O	82	Connection: ID=0x00290101, SEQ=0000012053, O->T
310	17.092790	192.168.1.20	192.168.1.10	TCP	60	44818 → 52870 [ACK] Seq=1 Ack=25 Win=8168 Len=0
311	17.092799	192.168.1.20	192.168.1.10	TCP	60	[TCP Dup ACK 310#1] 44818 → 52870 [ACK] Seq=1 Ack=25 Win=8168 Len=0
312	17.124234	192.168.1.55	192.168.1.10	TCP	60	44818 → 60033 [ACK] Seq=1 Ack=25 Win=8192 Len=0
313	17.124238	192.168.1.55	192.168.1.10	TCP	60	[TCP Dup ACK 312#1] 44818 → 60033 [ACK] Seq=1 Ack=25 Win=8192 Len=0
314	17.186971	192.168.1.222	192.168.1.10	CIP I/O	82	Connection: ID=0x92160000, SEQ=0000000001, T->O
315	17.186976	192.168.1.222	192.168.1.10	CIP I/O	82	Connection: ID=0x92160000, SEQ=0000000001, T->O
316	17.202794	192.168.1.10	192.168.1.222	CIP I/O	82	Connection: ID=0x00290101, SEQ=0000012054, O->T
317	17.202797	192.168.1.10	192.168.1.222	CIP I/O	82	Connection: ID=0x00290101, SEQ=0000012054, O->T
318	17.202798	192.168.1.10	192.168.1.222	CIP I/O	82	Connection: ID=0x00290101, SEQ=0000012055, O->T
319	17.202799	192.168.1.10	192.168.1.222	CIP I/O	82	Connection: ID=0x00290101, SEQ=0000012055, O->T
320	17.243666	192.168.1.10	192.168.1.222	TCP	60	62892 → 44818 [ACK] Seq=197 Ack=244 Win=8192 Len=0
321	17.243668	192.168.1.10	192.168.1.222	TCP	60	[TCP Dup ACK 320#1] 62892 → 44818 [ACK] Seq=197 Ack=244 Win=8192 Len=0
322	17.243678	Siemens_S8:e3:36	Spanning-tree (for...) STP	80 RST	Root = 32768/0/38:4b:24:58:e3:35 Lost = 0 Port = 0x0001	
323	17.287152	192.168.1.222	192.168.1.10	CIP I/O	82	Connection: ID=0x92160000, SEQ=0000000002, T->O
324	17.287165	192.168.1.222	192.168.1.10	CIP I/O	82	Connection: ID=0x92160000, SEQ=0000000002, T->O
325	17.295109	Siemens_S8:e3:3a	LLDP Multicast	LLDP	288	LA/kytkin LA/port-005 20 SysId=siylame Not Set SysD=Siemens, SPMATIC NET, SCALANCE XB200, 6GKS 208-0BA00-2TB2, Hw: Version 3, Fh
326	17.295121	192.168.1.10	192.168.1.222	CIP I/O	82	Connection: ID=0x00290101, SEQ=0000012056, O->T
327	17.295126	192.168.1.10	192.168.1.222	CIP I/O	82	Connection: ID=0x00290101, SEQ=0000012056, O->T
328	17.295856	192.168.1.10	192.168.1.222	CIP I/O	82	Connection: ID=0x00290101, SEQ=0000012057, O->T
329	17.295861	192.168.1.10	192.168.1.222	CIP I/O	82	Connection: ID=0x00290101, SEQ=0000012057, O->T
330	17.389979	192.168.1.222	192.168.1.10	CIP I/O	82	Connection: ID=0x92160000, SEQ=0000000003, T->O
331	17.389982	192.168.1.222	192.168.1.10	CIP I/O	82	Connection: ID=0x92160000, SEQ=0000000003, T->O

Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{41695D31-C970-4209-9F00-D4...} IEEE 802.3 Ethernet Logical-Link Control Spanning Tree Protocol

```

0000 01 00 c2 00 00 00 38 4b 24 58 e3 3d 00 27 42 42 .....8K 5X = 'BB
0010 03 00 00 02 02 7e 00 00 38 4b 24 58 e3 35 00 00 ..... 8K5X 5--
0020 00 00 00 00 38 4b 24 58 e3 35 00 00 00 00 14 00 .....8K5X 5-----
0030 02 00 0f 00 00 00 00 00 00 00 00 00

```

KUVA 3. Wireshark

Jokainen rivi yleisnäkymässä vastaa yhtä datapakettia. Datapaketin syvällisempää analysointia varten datapaketti voidaan avata napauttamalla riviä, jolloin sen tarkemmat tiedot avautuvat sivun alareunaan. Vasemmassa alareunassa paketti on avattu sanallisesti ja oikeanpuoleisessa reunassa taas heksadesimaalijärjestelmän mukaisesti. (LIITE 1)

Projektissa Wireshark-ohjelmisto on suuressa osassa Ethernet/IP-väyläkommunikoinnin testausta. Ohjelmistolla saadaan vertailtavaa ja analysoitavaa dataa kättelyistä sekä kommunikoinneista eri laitteiden välillä.

3 AUTOMAATIOJÄRJESTELMÄN LAITTEET

Nykyinen kokoonpano koostuu SattCon 200-CPU 30/40 ja Sattcon05 Slimline -logiikasta sekä Alfa Lavalin ja ABB:n ControlNet-adaptoreista sekä hajautetuista I/O-yksiköistä. Näiden kahden eri tuotemerkin käytön samassa järjestelmässä selittää yritysten yhteinen historia.

Alun perin Alfa Laval Automation on perustettu vuonna 1955 Ruotsissa, mutta aikojen saatossa se on kasvanut yritysfuusioiden seurauksena ja sillä on ollut monta eri nimeä (Satt Electronics, Ahlsell Control, SattControl sekä Alfa Laval Automation). Vuonna 1998 ABB osti Alfa Laval Automationin, jonka teknologioissa Alfa Lavalin innovaatio, kekseliäisyys ja tietotaito vaikuttaa vielä tänä päivänäkin. (4.) Tämän vuoksi myös tässä järjestelmässä käytetyt ABB:n ja Alfa Lavalin I/O-kortit ovat keskenään täysin samanlaisia toiminnoiltaan ja ulkomuodoltaan merkistä riippumatta. Kyseiset I/O-kortit ja -pohjat pohjautuvat Allen-Bradleyn Flex I/O-ratkaisuun.

Logiikoiden ja hajautetun I/O:n lisäksi nykyisessä järjestelmässä on valvomo, Alfa Lavalin operointipaneeli sekä -kytkimiä.

3.1 Projektin ohjelmoitavat logiikat

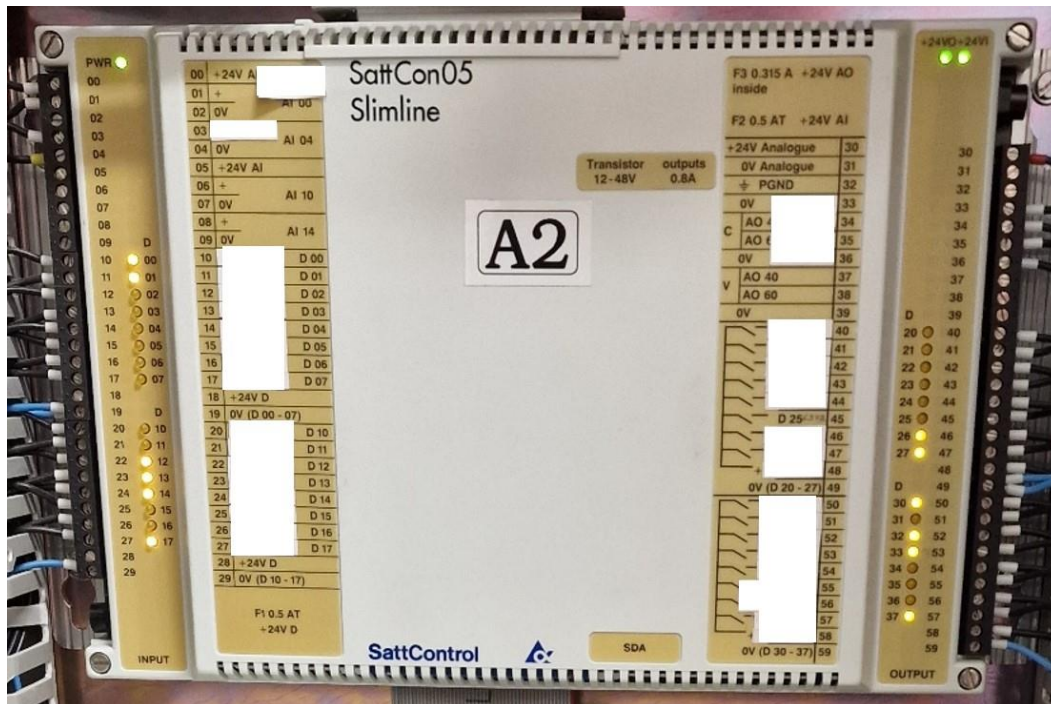
Järjestelmän päälogiikka on ABB:n SattCon 200-CPU 30/20 (kuva 4). Se on prosessiohjaintyökalu keskisuurille ja suurille järjestelmille. 200-sarjan logiikka sisältää monia vakio toiminnallisuuksia, kuten 16 erillistä ohjelmoitavien logiikkaohjainten ohjelmamoduulia, säätöpiiriominaisuudet, reseptien ja hälytysten käsittelyn sekä integroidun tiedonsiirron. (5.) SattCon 200 -sarja on siirtynyt elinkaarensa aktiivisesta vaiheesta klassiseen vuonna 2017. Ohjelmointiympäristönä SattCon 200-CPU 30/20 -logiikalle toimii DOX 10.



KUVA 4. SattCon 200-CPU 30/20

Toinen logiikka nykyisessä järjestelmässä on SattCon05 Slimline (kuva 5). Se on ohjelmoitava logiikkaohjain, jota voidaan käyttää erillislaitteena tai suuremman sovelluksen osana ohjauksessa. Se sisältää teollisuudelle tärkeitä ominaisuuksia, joita ovat muun muassa tiedonsiirto, laskenta, laskurit ja ajastimet, sekvenssin ohjaukset sekä näyttökehotteet. SattCon05 Slimline -logiikan vahvuuksia ovat erittäin kompakti koko, iso muisti, laajat yhdistelymahdollisuudet sekä integroitu tiedonsiirto. (6.)

Kyseisen logiikan käyttö alkaa kuitenkin olemaan hyvin riskialtista nykypäivänä tuotteen elinkaaren ja varaosien heikon saatavuuden vuoksi. SattCon05 Slimline on siirtynyt klassiseen vaiheeseen vuonna 2004 ja rajoitettuun vaiheeseen vuonna 2007. (6.) Kun tuotteen elinkaari on siirtynyt vanhentuneen tuotteen vaiheeseen yli kymmenen vuotta sitten, sen rikkoutuessa varaosia ei ole saatavilla tai, jos on, ne ovat hyvin kalliita. Ohjelmointiympäristönä tällekin logiikalle toimii DOX 10.



KUVA 5. SattCon05 logiikka

Edellä mainitut logiikat korvataan Siemensin 1515SP PC2 F -logiikalla (kuva 6). 1515SP-sarjan logiikassa on kaikki tarvittavat ominaisuudet, mitä nykyisenkin järjestelmän logiikoissa on. Näiden ominaisuuksien lisäksi valittu logiikka tukee myös turvaohjelmointia, jota ilmaisee tuotekoodissa oleva F-kirjain. Turvaominaisuudet voi asettaa joko aktiiviseksi tai käyttää ilman tätä ominaisuutta niin sanottuna normaalina logiikkana. PC 2 -merkintä taas ilmaisee logiikan olevan PC-pohjainen automaatiolaite, jota käytetään ohjaus- ja visualisointitarkoituksiin (7, s. 21). Muita tämän logiikan ominaisuuksia ovat sen kompakti koko sekä kestävä rakenne (7, s. 22). Ohjelmointiympäristönä tälle logiikalle toimii Siemensin TIA Portal.



KUVA 6. Siemens 1515SP PC2 F (8)

3.2 Käytetyt kenttäväylät

Nykyisessä järjestelmässä tiedonsiirtoprotokollana toimii ControlNet. ControlNet on kehitetty alun perin teollisuuden automaatio- ja ohjausjärjestelmien tiedonsiirtoon. Se tarjoaa suuren tiedonsiirtonopeuden aikakriittisten I/O:n sekä viestidatan siirroille sisältäen ohjelmien latauksen ja konfiguroinnin sekä vertaisverkkoa käyttävien fyysisten yhteyksien linkit. ControlNet tarjoaa kolme eri topologiavaihtoehtoa: lineaarisen, tähti- sekä puutopologian. Eri topologiatyyppejä voidaan yhdistellä tarpeen mukaan järjestelmässä.

ControlNet on osa CIP-protokollaa ylätasolla käytävien protokollien joukkoa. CIP eli Common Industrial Protocol tarjoaa kattavan määrän viestejä sekä palveluita erilaisille automaatiosovelluksille, joita ovat konfigurointi ja tiedonsiirto, synkronointi, turva ja ohjaus. Kuten kaikki CIP-protokollaa käyttävät tietoliikenteet, ControlNet mukaillee OSI-mallia eli Open System Interconnection. OSI-malli määrittelee verkkoliikenteen kehykset seitsemään eri kerrokseen, joita ovat: fyysinen, siirto, verkko, kuljetus, istunto, esitystapa sekä sovellus. (9, s. 2.)

Näistä fyysinen kerros määrittelee yhteyden fyysisen rakenteen eli kaapeloinnin, keskittimet sekä vahvistimet. Verkkoyhteyden maksimipituuteen vaikuttaa laitteiden ja vahvistimien määrä sekä kaapelityyppi. ControlNetin kaapelointiin käytetään joko koaksiaalista kaapelia tai optista kuitua. Koaksiaalinen kaapeli on yleisesti käytetty myös antennikaapelina. Optinen kuitu on hyvin häiriösuojattua sekä immuuni potentiaalieroille maadoituksessa. Optisen kuidun käyttö vaatii lähetinvas-taanotin parin toimiakseen. Redundanttikaapelointia eli molempia kaapelointityyppejä käytettäessä

jokainen laite arvioi signaalin laadun itse ja valitsee dynaamisesti käytettävän signaalin. Control-Net-kaapeloinnissa käytetään TNC- tai BNC-liittimiä riippuen ympäristöstä. (9, s. 3.)

Nykyisin kuitenkin ControlNetiä käytetään harvemmin Ethernet-pohjaisissa kommunikoinneissa. Ethernet-pohjaisista kommunikoinneista Ethernet/IP tarjoaa suuremmat ja mukautettavat tiedonsiirtonopeudet sekä yhteensopivuuden tavallisten verkkolaitteiden kanssa, jonka vuoksi sen suosio on kasvanut teollisuusautomaatiojärjestelmissä. Tässäkin projektissa ControlNet korvataan Ethernet/IP:llä.

Ethernet/IP käyttää myös ylätasojen CIP-protokollaa sekä mukailee OSI-mallia. Protokolla käyttää tiedonsiirrossa TCP/IP- sekä UDP-protokollia. Tämän lisäksi se tarjoaa käyttäjälleen standardinmukaisen Ethernet-tekniikan IEEE 802.3 yhdistettynä TCP/IP:hen. Ethernet/IP-standardi sallii käyttäjän valita itse tiedonsiirtonopeuden sekä yhteensopivuuden kaikkien Ethernet-asennusvaihtoehtojen kanssa. Ethernet/IP-protokolla on myös yhteensopiva Internet-protokollien kanssa eli esimerkiksi HTTP:n ja DHCP:n sekä teollisten protokollien kanssa kuten OPC UA. Muita ominaisuuksia ovat muun muassa yhteensopivuus Ethernet-APL fyysisen kerroksen kanssa, PRP-tuki, QuickConnect-ominaisuus sekä CIP Safety-, Sync- ja Motion-ominaisuudet. Topologiavaihtoehdot ovat tähti-, lineaari- sekä rengastopologia. (10, s. 2.)

Ethernet-kaapelointiin käytetään kierrettyä paria (CAT) tai optista kuitua. Koaksiaalinen kaapelointi on myös yksi vaihtoehto, mutta se on vanhempaa tekniikkaa, jota käytetään enää vain olemassa olevissa järjestelmissä. Optista kuitua käytettäessä tarvitaan lähetin ja vastaanotin, jotta yhteys toimii. Kyseistä kaapelointia käytetään järjestelmissä, jotka ovat alltiita sähkömagneettiselle säteilylle sekä järjestelmissä, jotka tarvitsevat suurempia tiedonsiirtonopeuksia kuin mihin parikaapelilla päästään.

Kierretty parikaapeli on yleisimmin käytetty kaapelointityyppi Ethernet-pohjaisissa järjestelmissä. Tämä kaapelityyppi koostuu yleensä kahdesta kierretystä parista, jotta tiedonsiirto saadaan kulkemaan molempiin suuntiin yhtä aikaa (full-duplex). Parikaapeloinnissa käytetään suojattuja kaapelityyppejä riippuen järjestelmän ympäristön häiriövaatimuksista. Ethernet/IP:n kaapeloinnissa yleensä käytetään RJ45-liitintä. (10, s. 3.)

3.3 Hajautetut I/O-yksiköt

Olemassa oleva hajautettu I/O koostuu ABB:n ja Alfa Lavalin S200-sarjan I/O-korteista sekä 200-ACN ControlNet -adaptereista. Järjestelmässä on digitaalisia tulo- ja lähtökortteja sekä analogisia tulo- ja lähtökortteja. Nämä kortit jäävät sellaisenaan järjestelmään ja vain väyläyksikkö vaihdetaan. Nykyisen järjestelmän väyläyksikkö on kuvassa 7.



KUVA 7. Nykyisen järjestelmän ControlNet-adapteri

Digitaalinput DI16-kortteja on yhteensä 14 kappaletta ja digitaaloutput DO16-kortteja 13 kappaletta. Näiden lisäksi analogiatulo AI8-kortteja on 6 kappaletta ja analogialähtö AO4-kortteja 3 kappaletta. Järjestelmässä on lisäksi muutama ABB:n I/O-kortti, jotka tullaan korvaamaan Siemensin ET 200SP-sarjan hajautus-I/O:lla samoin kuin nykyisen järjestelmän toinen logiikka. Kuvassa 8 on mallikuva I/O-korteista sekä liitinpohjasta.



KUVA 8. Liitinpohja sekä DO16- ja AI8-kortit

Nykyisen järjestelmän ControlNet-adapterit korvataan Allen-Bradleyn 1794-AENTR Ethernet/IP -adaptereilla. Yhteensä Ethernet/IP-adaptereita tulee uuteen kokoonpanoon 7 kappaletta. Adapterissa on kaksi kommunikointiporttia ja yhden adapterin perään voi liittää enintään kahdeksan liitinpohjaa. Liitinpohjiin yhdistettävät I/O-kortit voivat koostua joko yhdestä korttityypistä tai eri korttityyppien yhdistelmistä. 1794-AENTR adapteri on elinkaarensa aktiivisessa vaiheessa. Kuvassa 9 näkyvää adapteria 1794-AENT käytetään kommunikoinnin testauksessa. Se on valitusta adapterista vanhempi versio yhdellä kommunikointiportilla.



KUVA 9. Allen-Bradleyn vastaava Ethernet/IP-adapteri

Näiden lisäksi uudessa kokoonpanossa tulee olemaan Siemensin ET 200SP hajautettu I/O. Tällä korvataan nykyisen järjestelmän toisen logiikan I/O sekä muutama ABB:n I/O-kortti. ET 200SP-sarjasta löytyy tavallisia tulo- ja lähtökortteja sekä turvakortteja. Kortteja voidaan käyttää sellaisenaan hajautus-I/O:na tai integroidun logiikkaohjaimen yhteydessä. Integroitu logiikkaohjain kyseisiin kortteihin on S7-1500-sarjan logiikka, kuten tässäkin projektissa tulee olemaan.

Modulaarinen ET 200SP I/O-järjestelmä mahdollistaa ominaisuuksiensa ansiosta monimutkaisten kokoonpanojen toteuttamisen. Näitä ominaisuuksia ovat muun muassa vakio- ja turvatoiminnot, teknologiafunktio ja -mittaukset sekä kestävä rakenne. Kuvassa 10 on esimerkki ET 200SP-järjestelmästä ja I/O-korteista. (11.)



KUVA 10. ET 200SP (11)

4 ESISUUNNITTELU

Ennen suunnittelutyön aloitusta on esisuunnitteluvaihe. Esisuunnittelun tarkoitus on perehtyä tehtävään työhön, kerätä tarvittavat lähtötiedot, tarkistaa sekä varmistaa valittujen komponenttien ja työkalujen yhteensopivuus. Esisuunnittelun avulla saadaan hyvä käsitys projektikonaisuudesta ja reunaehdoista sekä mahdollisista selvityksiä tarvitsevista kohdista.

4.1 Lähtötietojen kerääminen

Järjestelmävaihdon aloittamiseksi tarvitaan lähtötietoja. Asiakas on lähettänyt heiltä löytyviä dokumentteja, joita ovat erilaiset manuaalit, sähkökuvat sekä nykyisen järjestelmän ohjelmien ohjelmakopiot. Ohjelmat toimitettiin meille Word-tiedostoina, joista saadaan kerättyä tarvittavat lähtötiedot. Edellä mainittujen dokumenttien lisäksi olemme saaneet valokuvia järjestelmästä ja laitteista.

Valokuvien ja sähkökuvien avulla saadaan laskettua I/O:den kokonaismäärä. Tulojen ja lähtöjen määrällä on merkitystä korvaavaa logiikkaa valitessa. Korvaavasta logiikasta täytyy löytyä tarpeeksi tehoa korvaamaan nykyiset käytössä olevat logiikat sekä muutama hajautetun I/O:n kortti. Kun tiedossa on järjestelmässä olevat ja korvattavat laitteet, saadaan määriteltyä projektista kokonaiskuva sekä arvioitua tarvittava työmäärä ja aikataulu.

4.1.1 I/O-lista

Ensimmäisenä kerätään I/O-lista. Tulot ja lähdöt saadaan kerättyä hyödyntämällä sähkökuvia, ohjelmakopiota sekä valokuvia keskuksista. On mahdollista, että kaikkea, mitä ohjelmaan on lisätty tai muokattu, ei ole välttämättä päivitetty sähkökuviin, joten kaikkien saatujen dokumenttien tarkastelu ja vertailu keskenään on hyvä tehdä, jotta varmistetaan todellinen I/O-määrä. I/O-lista kerätään Excel-taulukkoon nykyisten ja korvaavien osoitteiden selkeän esittämisen vuoksi.

Apexilta löytyy I/O-listan mallipohja, jota hyödyntämällä saadaan koottua selkeä kokonaisuus. I/O-listaan kerätään tulojen ja lähtöjen nimet, nykyiset osoitteet sekä sijainnit. Koska järjestelmäkokoisuus hieman muuttuu uudessa kokoonpanossa, lisätään I/O-listaan mahdolliset uudet osoitteet ja sijainnit.

4.1.2 Apumuistipaikat

Seuraavana vaiheena on ohjelmakopion läpikäyminen ja apumuistipaikkojen kerääminen Excel-
taulukoksi. Varsinaisen logiikkaohjelman sisäiset apumuistipaikat kerätään omaan taulukkoon, toi-
sen logiikkaohjelman omaansa sekä HMI-muistipaikat omaansa. Apumuistipaikkojen taulukointi
helpottaa ohjelman kirjoitusta uudelle logiikalle, jonka ohjelmointityö voidaan tehdä saadun ohjel-
makopion avulla.

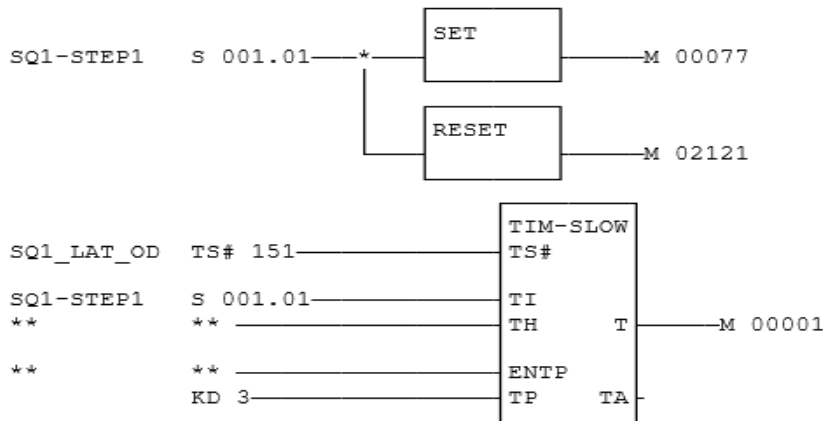
Apumuistipaikkojen suuren määrän vuoksi osoitteiden kanssa tulee olla tarkkana niin kerätessä
kuin uutta ohjelmaa tehdessä, jotta päällekkäisyyksiä ei tule. Taulukossa 1 on nykyisessä ohjel-
massa käytettyjen muistipaikkojen merkintöjä ja selityksiä.

TAULUKKO 1. Nykyisen ohjelman muistipaikkojen merkinnät

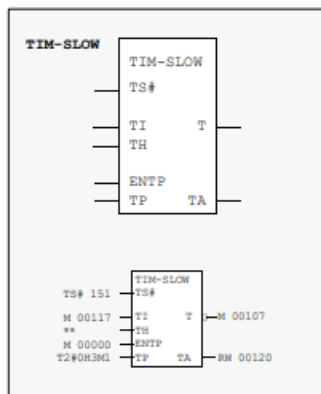
S	Sequence register	0.0 - 255.99	
M	Memory bit	0 - 37777	0 - 7777 = Working memories
RW	Register words	0 - 16383	0 - 99 = System functions 100 - 3999 = User register
KD	Decimal constant	0 - 65535	
L	Label	0 - 1999	0 - 499 User labels
SYS	System status	0 - 99	

4.1.3 Ohjelmalohkot ja niiden toiminnallisuudet

Ohjelman toiminnallisuuden selvitys aloitetaan nykyisessä ohjelmassa käytettyjen lohkojen toimin-
nan selvittämisellä. Aluksi kerätään kaikki käytetyt lohkot yhteen dokumenttiin. Kun lohkot on ke-
rätty, DOX10-ohjelmointimanuaalista löytyy lohkojen toimintakuvaukset. Manuaalin lohkokuvaus-
sissa lohkojen toiminta on avattu perusteellisesti, mikä helpottaa vastaavan toiminnan lohkojen
etsimistä uutta ohjelmaa varten. Kuvassa 11 on esimerkki ohjelmakopion ohjelmalohkoista ja ku-
vassa 12 on yhden käytetyn ohjelmalohkon toiminta manuaalissa avattuna.



KUVA 11. Esimerkki saadusta ohjelmakopion lohkoista



Updating of the timers is not synchronized with the PLC execution, consequently the real delay time may vary from the defined.

Slow timer function

Name	Variables	Function	Dup.	Inv.	Conn.
TS#	TS#, **	Timer number			
TI	B0	Timer input (start)		I	
TH	B0, **	Timer hold		I	
ENTP	B0, **	Enable timer preset input		I	
TP	W0, T2#, **	Timer preset			
T	B1	Timer output		I	N
TA	W2	Timer accumulated			N

Function: SattCon 200 contains 1000 timers for time delays in the application program. Slow timers are identified by numbers between 150 and 999. The corresponding preset time range and associated variable type are presented below.

Variables	Timer number	Resolution	Variable	Preset time range
TS#	150 - 999	1 s	T2#	0H0M01 - 18H12M15

Timers are controlled by the following inputs and outputs:

- TS# is the timer number.
- TI is the timer input for activation of the timer. When TI is FALSE, T is set to FALSE and TA is set to 0.
- TH is the hold input.
- ENTP is the enable input for TP
- TP is the preset input.
- T is the timer output. T is set when the preset time is reached.
- TA is the accumulated (elapsed) time.

If a double star is connected to TS# the timer is generated by DOX 10.

If TP is changed while TT is active, the new value will be taken directly. This can cause T to be changed in either direction.

Inputs TH and TP can be disabled by connecting a double star (**). Outputs T and TA need not be connected.

Associated function number is FUNC F# 201.

Refer to chapter "Timers" for further information.

Caution: Updating of the timers is not synchronized with the PLC execution, consequently the real delay time may vary from that defined.

Example: TIM-SLOW function TS#151 is used.

The timer is activated when M117 is TRUE.

The hold function (TH) is not used.

The timer is preset to 3 minutes and 15 seconds (T2#0H3M15) when M0 is TRUE.

M107 is FALSE when the preset time is reached.

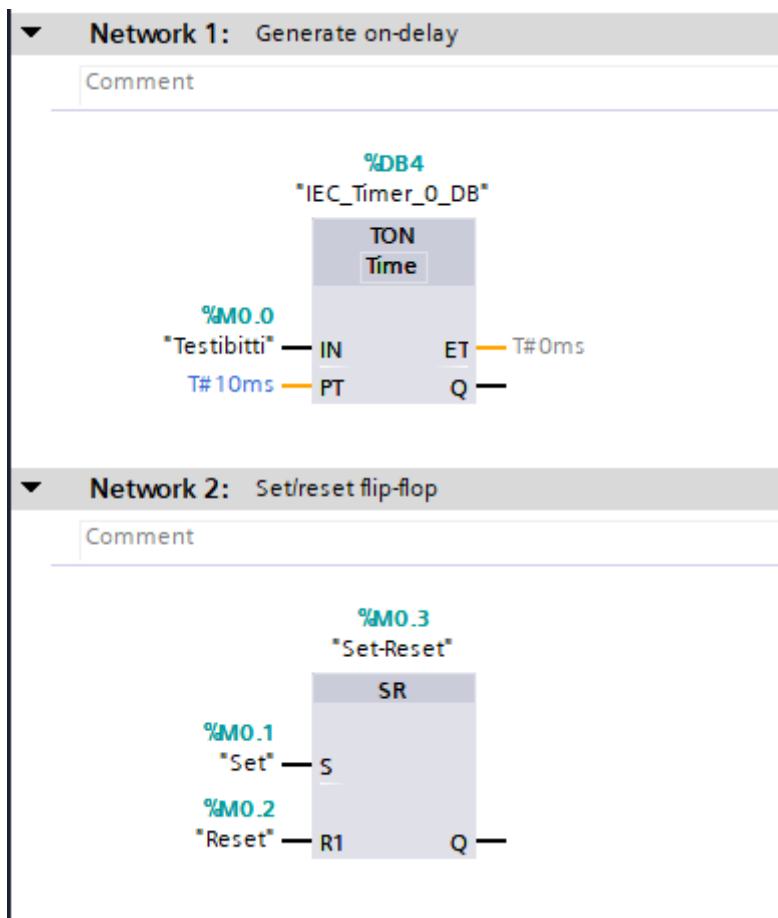
Register RW120 contains the elapsed time value.

Note. TS# can be displayed and changed via the terminal.

KUVA 12. Esimerkki manuaalin lohkokuvauksesta (12)

Kaikkien lohkojen toimintaa ei kuitenkaan löydy manuaalista, joten tulee selvittää millä keinolla saadaan selville näiden lohkojen toiminnallisuus. Näitä epäselviä lohkoja ovat kaikki sekvensseihin liittyvät lohkot.

Seuraavana vaiheena on löytää Siemens-ohjelmoinnin vastaavat toimilohkot. Suurimmalle osalle lohkoista löytyy suoraan vastine TIA Portalista, mutta osan joutuu tekemään itse yhdistelemällä eri lohkoja. Monet yksinkertaiset toimilohkot, kuten or-, and-, sr- tai r-trig-lohkot löytyvät suoraan TIA Portalista. Aikaisempien kuvien vastaavat lohkot TIA Portalissa on kuvassa 13.



KUVA 13. Esimerkkilohkot TIA Portalissa

4.2 Haastavien kohteiden kartoitus ennen suunnittelutyön aloitusta

Ennen varsinaista suunnittelutyötä kartoitetaan haastavat kohdat, jotta pystytään reagoimaan niihin varhaisessa vaiheessa. Haastavien kohteiden kartoitus selkeyttää kokonaiskuvaa sekä nostaa esille mahdolliset ongelmakohdat, jotka vaikuttavat työn etenemiseen ja aikataulutukseen.

Ohjelmakopion ollessa Word-tiedostona se tuottaa omia haasteita. Kaikkien tagien osoitteet tai kommentit eivät näy kokonaan, sekä ohjelmarakennetta on haasteellisempi hahmottaa. Ohjelmalohkojen toiminnot löytyvät ohjelmointiympäristön manuaalista, mutta muutamia puutteita näistäkin löytyy. Näiden epäselvien lohkojen toiminnan selvittämiseksi tarvitsemme tukiresursseja. Vanhan ohjelmointikoneen saaminen lainaan tai pääsy paikan päälle monitoroimaan ohjelmaa voisi auttaa näiden epäselvyyksien ratkaisemisessa.

Hajautetun I/O:n pysyessä ennallaan se vaatii liitinpohjan kanssa yhteensopivan väyläyksikön. Väyläyksikön valinnassa täytyy huomioida yhteensopivuus, elinkaari ja hinta. Allen-Bradley on ainoa, jonka valikoimasta löytyy elinkaarivaatimuksia vastaava väyläyksikkö, joka sopii olemassa olevien hajautus-I/O-korttien sekä liitinpohjien kanssa yhteen. Vaihtoehtoina tässä vaiheessa on Profibus ja Ethernet/IP. Profibus-protokolla on tuettu Siemensin logiikoissa oletuksena lisäväyläkortin avulla. Profibus-väyläyksikön hinta on kuitenkin huomattavasti suurempi kuin Ethernet/IP-adapterin hinta, koska kyseinen väyläyksikkö on siirtynyt elinkaarensa vuoksi varaosatuotteeksi. 1794-AENTR Ethernet/IP -adapteri on sen elinkaarensa aktiivisessa vaiheessa sekä hinta Profibus-väyläyksikköön verrattuna on huomattavasti pienempi, joten kyseinen adapteri on sekä elinkaareltaan että toimitusajaltaan paras ja kustannustehokkain.

Siemensin S7-sarjan logiikat tukevat Profibus- ja Profinet-protokollia oletuksena, mutta eivät Ethernet/IP-protokollaa. Siemens on kuitenkin tehnyt kyseiselle protokollakommunikoinnille omat kirjastot, joiden avulla pystytään käyttämään Ethernet/IP-protokollaa sekä konfiguroimaan kokoonpanossa käytettävät laitteet. Kirjaston toimivuus valitulla kokoonpanolla voi tuottaa haasteita ja siksi se täytyy testata. Valitulla tai vastaavanlaisella kokoonpanolla tehdystä järjestelmästä ei löydy tietoa, joten tiedon niukkuus voi tuottaa haasteita mahdollisten ongelmien esiintyessä.

4.3 Laitetietojen tarkistus

Siemensin logiikan, Allen-Bradleyn adapterin sekä hajautus-I/O:n yhteensopivuus tarkistettiin laitetuottajalta. Juuri kyseisellä kokoonpanolla ei ole ennen tehty järjestelmää, mutta teoriassa sen pitäisi toimia. Siemensin Ethernet/IP Scanner -lohkon kanssa yhteensopivien logiikoiden mallit ja versionumerot löytyvät kyseisen kirjaston manuaalista.

Testilogiikkoina käytettiin Siemensin 1515F-sarjan logiikkaa skannerina, joka on myös uuteen kokoonpanoon tuleva logiikkamalli sekä Siemensin 1215C-sarjan logiikoita adaptereiksi konfiguroituna. Uuteen järjestelmään tulee 1794-AENTR-adapterit, joiden toimitusaikojen vuoksi testausta varten jouduttiin tilaamaan vastaava yksittäinen käytetty 1794-AENT-adapteri. Testejä varten hankittu adapteri on elinkaarensa lopussa eli vanhempaa sarjaa kuin uuteen järjestelmään tuleva adapteri. Muuta toiminnallista eroa näillä adaptereilla ei kuitenkaan ole, kuin että vanhemman mallin adapterissa on yksi kommunikointiportti kahden sijaan.

5 KOMMUNIKOINNIN TESTAUS

Kommunikoinnin toimivuuteen Siemensin logiikan ja Allen-Bradleyn Ethernet/IP-adapterin välillä kohdistui suurimmat epäilyt projektissa, joten sen toimivuus tuli testata ja varmistaa ennen varsinaista logiikkakoodin kirjoitusta. Siemensin S7-logiikat eivät alun perin tue Ethernet/IP-kommunikointia, joten Siemens on tehnyt lohkokirjastot, joiden avulla pystytään käyttämään Ethernet/IP-protokollapohjaista kommunikointia. Projektia, jossa olisi käytetty juuri samoja komponentteja kuin tässä projektissa, ei kuitenkaan ole tiedossa.

5.1 Testauksen valmistelu

Kommunikoinnin testausta varten tarvittiin testilaitteet. Logiikka löytyi Apexin varastosta ja I/O-kortit saimme lainaan testausta varten asiakkaalta. Väyläyksikköjen toimitusaika laitetoimittajalta oli pitkä, joten täytyi selvittää, mistä saataisiin yksi väyläyksikkö testejä varten nopeammin. Nettikauppoja selaillessa löydettiin yksi vastaava käytetty väyläyksikkö nopeammalla toimitusajalla. Näiden laitteiden avulla voitiin lähteä testaamaan kommunikointia.

Testilaitteiden lisäksi tarvittiin ohjelmointiohjelmisto ja mahdolliset lisäkirjastot kommunikointiprotokollalle. Siemensin tekemät Scanner- ja Adapter-lohkokirjastot, niiden manuaalit sekä malliprojektit ovat ladattavissa Siemensin sivuilta. Uusimmalle TIA Portal versiolle eli V18 ei ole vielä tehty kuin Ethernet/IP-adapter kirjasto, joten projektin ohjelmaversioksi valikoitui TIA Portal V17.

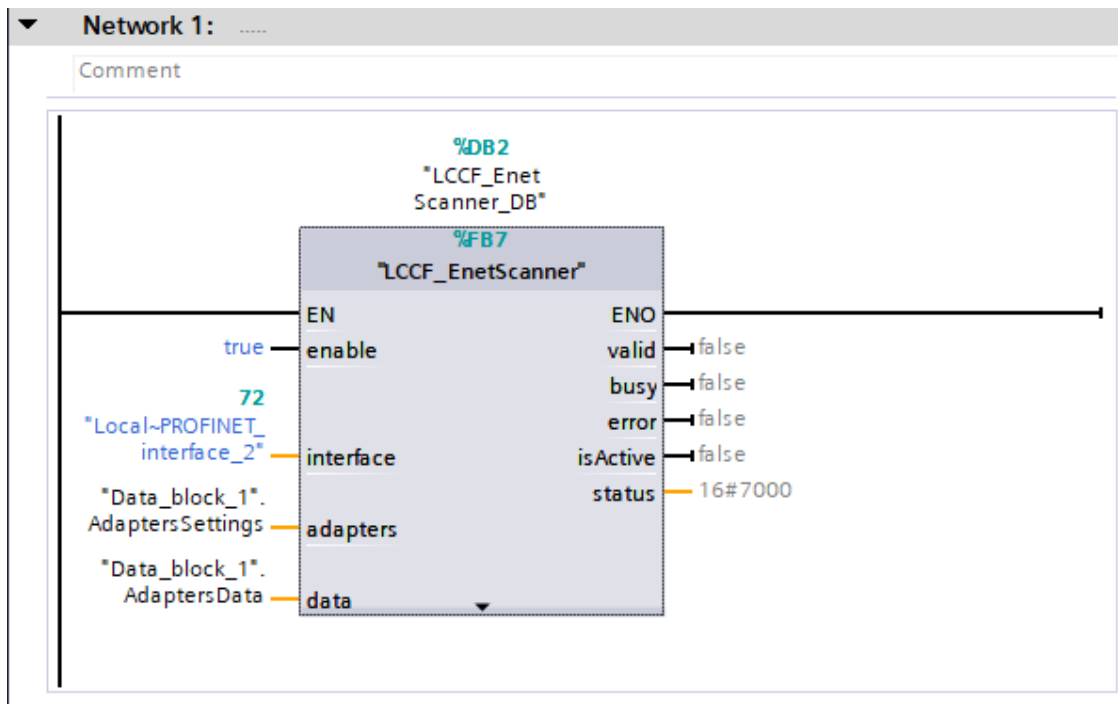
5.2 Testit

Testaus aloitettiin lataamalla tarvittavat kirjastot sekä perehtymällä niiden toimintaan ja käyttöön. Tämän jälkeen konfiguroitiin testilaitteet. Logiikka konfiguroitiin TIA Portaaliin, jossa tälle asetettiin IP-osoite. Allen-Bradleyn Ethernet/IP-adapterille IP-osoitteen voi asettaa joko laitteessa olevien painonappien avulla tai BootP-DHCP-Tool -sovelluksella. Näiden jälkeen TIA Portaalin luotiin datatyypit adapterin konfiguraatiolle sekä data-alueille (kuva 14). Näistä datatyypeistä konfiguraatioon määriteltiin adapterin teknisiä tietoja, joita olivat IP-osoite, tyyppinumero, tuotekoodi, versio, päivitysväli sekä kommunikointiporttien tunnukset ja koot. Nämä tiedot löytyivät laitteen EDS-tiedostosta, joka saatiin ladattua Allen-Bradleyn laitteesta RSLinx-ohjelmalla. Data-alueeseen määriteltiin

tarpeeksi suuret data-alueet niin lähtevälle kuin tulevalle datalle. Nämä datatyypit yhdistettiin Scanner-lohkokutsuun (kuva 15). Edellä mainittujen tietojen lisäksi kyseiseen lohkokutsuun määriteltiin käytettävä liitäntäportti.

	Name	Data type	Start value	F
[-]	Static			
[-]	[-] AdaptersSettings	Array[0..2] of "LCCF_typeEnetAdaptConfig"		
[-]	[-] [-] AdaptersSettings[0]	"LCCF_typeEnetAdaptConfig"		
[-]	[-] [-] AdaptersSettings[1]	"LCCF_typeEnetAdaptConfig"		
[-]	[-] [-] [-] AdaptersSettings[2]	"LCCF_typeEnetAdaptConfig"		
[-]	[-] [-] [-] address	IP_V4		
[-]	[-] [-] [-] vendorIdentifier	UInt	1	
[-]	[-] [-] [-] productType	UInt	12	
[-]	[-] [-] [-] productCode	UInt	90	
[-]	[-] [-] [-] revisionMajor	USInt	4	
[-]	[-] [-] [-] revisionMinor	USInt	1	
[-]	[-] [-] [-] assemblyInstan..	UInt	23	
[-]	[-] [-] [-] packetInterval	Time	T#100MS	
[-]	[-] [-] [-] conPointOutpu...	UInt	2	
[-]	[-] [-] [-] conPointOutpu...	UInt	16	
[-]	[-] [-] [-] conPointInputl...	UInt	3	
[-]	[-] [-] [-] conPointInputS...	UInt	20	
[-]	[-] [-] [-] configDataSize	UInt	0	
[-]	[-] [-] AdaptersData	Array[0..2] of "LCCF_typeEnetAdaptData"		
[-]	[-] [-] [-] AdaptersData[0]	"LCCF_typeEnetAdaptData"		
[-]	[-] [-] [-] AdaptersData[1]	"LCCF_typeEnetAdaptData"		
[-]	[-] [-] [-] [-] AdaptersData[2]	"LCCF_typeEnetAdaptData"		
[-]	[-] [-] [-] [-] inputs	Array[0..501] of USInt		
[-]	[-] [-] [-] [-] outputs	Array[0..497] of USInt		
[-]	[-] [-] [-] [-] configs	Array[0..399] of USInt		

KUVA 14. Datatyypit TIA Portalissa

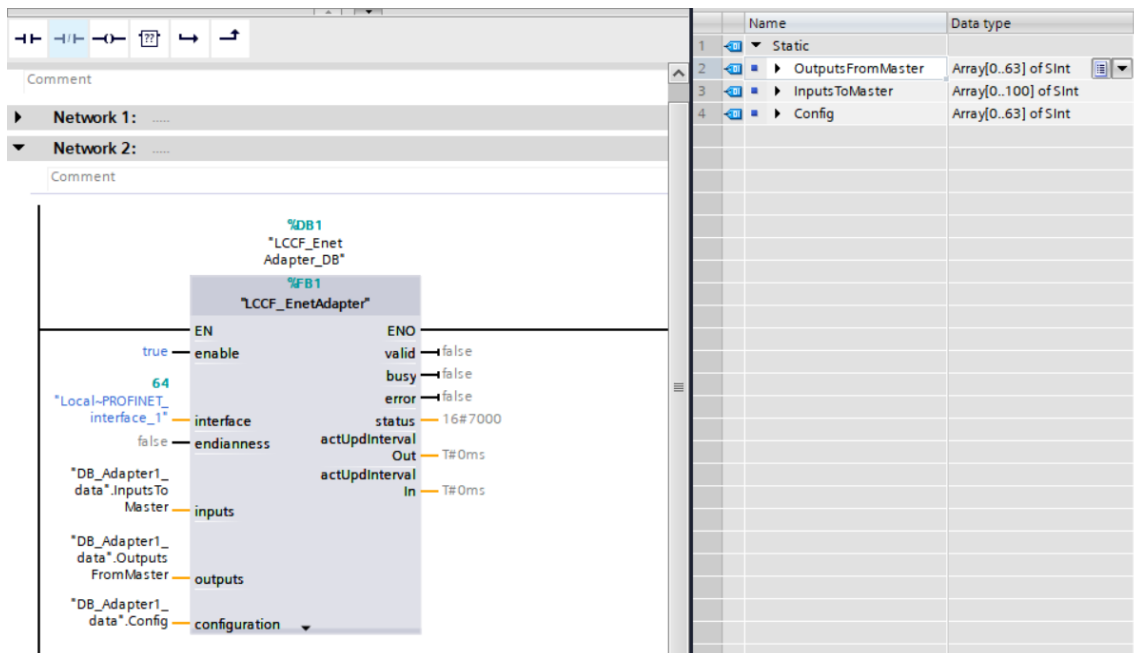


KUVA 15. Lohkokutsu TIA Portaalisssa

Ensimmäisellä testauksella kokeiltiin suoraan kommunikoida logiikan ja adapterin välillä. Kuitenkin tässä vaiheessa huomattiin, että kyseiselle kommunikoinnille 10 ms:n päivitysväli on ihan liian tiheä, jonka vuoksi kommunikointilohko menee häiriötilaan. Pidentämällä päivitysväliä yhteys saatiin toimimaan, eli logiikka tunnisti adapterin ja toisinpäin.

Seuraavana vaiheena oli lisätä adapterin perään I/O-kortteja ja tutkia, miten data näkyy TIA Portaalisssa. Tässä vaiheessa kuitenkin ongelmia alkoi ilmaantumaan. Logiikka ja scanner-lohko tunnistivat adapterin, mutta data ei alkanut näkymään TIA Portaalisssa. Adapterin konfigurointia kokeiltiin saada toimivaksi eri arvoilla, kuten esimerkiksi data-alueen pituuden muutoksella. Näillä muutoksilla ei kuitenkaan ollut mitään vaikutusta toimivuuteen.

Kun kommunikointi ja datansiirto ei lähtenyt toimimaan suunnitellusti, tarvitsimme toimivan testilaittekokoonpanon vertailukohdaksi. Siemensin omien laitteiden välisistä kommunikoinneista, joissa on käytetty Ethernet/IP-lohkokirjastoja, löytyi malliesimerkkejä netistä. Lisäsimme testiympäristöömme siis adapteriksi asetetun Siemensin S7-1200-logiikan. Kuvassa 16 on määritelty S7-1200-logiikka TIA Portaalisssa adapteriksi Ethernet/IP Adapter -kirjaston avulla. Tämän testin avulla nähtiin, miten lohko toimii, kun I/O-data liikkuu molempiin suuntiin. Liitteestä 3 löytyy S7-1200-logiikan parametrit EDS-tiedostossa sekä Scanner-lohkokirjaston konfiguroinnissa.



KUVA 16. Ethernet/IP Adapter -lohko

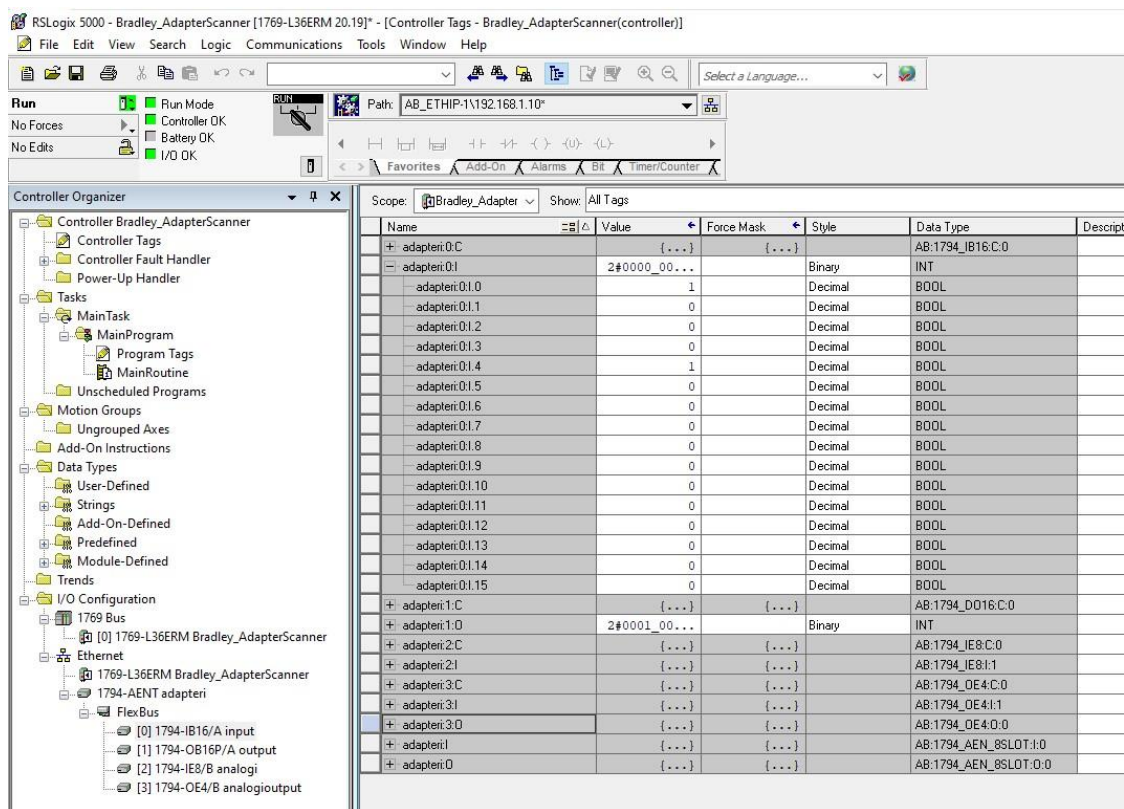
Siemensin laitteiden välisessä kommunikointitestauksessa huomattiin, että määritelty data-alueen pituus ja liikkuvan datan datatyyppi vaikuttaa kommunikoinnin toimimiseen. Kun määritelty data-alue skannerin päässä on pienempi kuin adapterin päässä, lohko menee häiriölle. Todellisuutta suuremmaksi määritellyllä data-alueella ei ole merkitystä I/O-datasiirron toimimiseen. Adapteri lähettää vain sen arvoissa määritellyn data-alueen mittaisia datapaketteja. Data-alueiden pituuksien asettamisen lisäksi oikein määritelty datatyyppi vaikuttaa datasiirron toimimiseen. Datatyyppien tulee olla molemmissa laitteissa sama.

Tässä vaiheessa testauksia tiedetään, miltä toimiva yhteys ja dataliikenne näyttää TIA Portaalissa. I/O-data tulee näkyviin luodun data-alueen ensimmäisiin paikkoihin. Näistä testeistä saaduilla tiedoilla testattiin vielä muutamilla eri konfigurointiparametreilla Allen-Bradley adapterin ja Siemensin logiikan välistä kommunikointia, mutta I/O-data ei vieläkaan tullut logiikalle. Kun tehdyillä testauksilla kommunikointia ei saatu toimivaksi, seuraava vaihe oli ottaa Wireshark-ohjelma avuksi ja alkaa tutkimaan laitteiden välillä liikkuvien datapaketien sisältöjä.

Siemensin laitteiden välillä kommunikointi toimi, joten käytetyt parametrit olivat oikeat. Testiadapterina olleen S7-1200-logiikan parametrit saatiin laitteen EDS-tiedostosta. Liitteessä 3 on toimivan Siemensin laitteiden välisen kommunikoinnin määritellyt parametrit. Kun parametrit tiedettiin, alettiin tutkimaan CIP I/O ja CIP CM datapaketien sisältöä. Näistä paketeista etsittiin kohdat, josta käytetyt parametrit löytyvät.

Seuraava epäily kohdistui Allen-Bradleyn adapterin konfigurointiin. Adapteria laitteena ei pysty konfiguroimaan erikseen, joten tarvitsimme lisää testilaitteita eli logiikan, johon pystyy ohjelmallisesti lisäämään kyseisen adapterin sekä I/O-kortit. Tilasimme laitetoimittajaltamme Allen-Bradleyn logiikan lainaan testejä varten.

Testeihin Allen-Bradleyn laitteiden välillä käytettiin RSLogix5000-ohjelmaa. Allen-Bradleyn testilogiikan malli oli 1769-L36ERM. Laitekonfiguraation tekeminen onnistui lisäämällä ohjelmaan kokonaisuudessaan käytetyt laitteet, ja data alkoi liikkumaan. Kuvassa 17 on yhden testikokoonpanon HW-konfiguraatio RSLogix5000-ohjelmassa. Tässä vaiheessa pystyttiin tarkistamaan Siemens-Bradley välisissä testeissä käyttämämme parametrit Bradleyn laitteiden välisen testauksen datapaketeista. Siemens-Bradley välisen testauksen parametrusointi oli määriteltä oikein, joten selittävä syytä ei löytynyt kommunikoinnin toimimattomuudelle käytetyistä parametreista.



KUVA 17. Allen-Bradleyn testikokoonpanon HW-konfiguraatio RSLogix5000-ohjelmassa

Adapterista ladattiin seuraavaksi EDS-tiedosto uudestaan, jotta nähdään, ovatko parametrit muutuneet tiedostossa. Mitään muutosta ei kuitenkaan ollut tiedostossa eikä varsinaisessa dataliikenteessä (CIP I/O), joten voitiin päätellä, että logiikka lähettää adapterille tarvittavat konfigurointitiedot. Näiden konfigurointitietojen perusteella adapteri osaa lähettää I/O-dataa logiikalle.

Allen-Bradleyn Ethernet/IP-adapteri vaatii siis konfiguroinnin, jossa määritellään kokoonpanossa olevat I/O-kortit erikseen. Seuraavana vaiheena oli selvittää, miten pystytään konfiguroimaan kokoonpano TIA Portaaliin. Datapakettien sisältöjen tutkimisessa löydettiin Bradleyn laitteiden välisestä kommunikoinnista datasegmentti, joka puuttuu Siemensin laitteiden välisistä kommunikoinneista. Kyseinen datasegmentti sisältää konfigurointitiedon kohteelle eli adapterille. Asiaa selvitettiin tämän jälkeen Siemensin kanssa ja saimme varmistuksen, että tässä projektissa käytettävä modulaarinen kokoonpano vaatii lohkokirjastolta korttikohtaisen konfigurointimahdollisuuden, jota tämänhetkinen kirjastoversio ei tue.

Datansiirtoa luettiin Wireshark-ohjelmalla jokaisessa testivaiheessa. Tämän avulla saatiin vertailtavaa dataa toimivan kokoonpanon sekä ei-toimivan kokoonpanon välillä. Näitä vertailuja sekä analysointeja avataan lisää luvussa Testien tulokset.

5.3 Testien tulokset

Testausta laajennettiin suunniteltua syvemmälle kommunikoinnin toimimattomuuden vuoksi. Helppo tapa olisi ollut vain todeta, että kyseisellä kokoonpanolla ei saada toimivaa järjestelmää, kun huomattiin adapterin ja logiikan välisen datansiirron toimimattomuus. Kuitenkin tilannetta, jossa kokoonpanoa ei vain saataisi kommunikoidaan ilman kunnon syytä, ei haluttu hyväksyä. Jos datansiirtoa ei saada toimimaan, ainakin oli löydettävä syy, miksi kyseisen kokoonpanon kommunikointi ei toimi. Myöskään asiakkaan toiveesta Siemensin laitekantaan siirtymisestä ei haluttu luopua.

Testauksen edetessä kommunikoinnin opiskeluun käytettiin paljon aikaa internetissä ja YouTubeissa. Tietoa ei paljoa löytynyt, joten Wireshark-tiedostoja alettiin pienistä palasista itse tutkimaan ja analysoimaan. Näiden lisäksi epäselviä asioita selvitettiin tukipyyntöjen ja kyselyiden avulla laitetoimittajien kanssa.

5.3.1 Ethernet/IP-protokollan sisältö kommunikoinnin näkökulmasta

Ethernet/IP:n datapakettien lähetyksen datalinkkikerroksessa määrittelee IEEE 802.3 -standardi. Kyseinen standardi on luotu lähiverkkotekniikkaa varten määrittelemään langallisen Ethernet-protokollayhteyden fyysisen kerroksen ja MAC-alikerroksen. IEEE 802.3 -standardi käyttää CSMA/CD-

yhteyttä (Carrier Sense Multiple Access with Collision Detection). Tämä mahdollistaa samaa fyysistä verkkoa käyttävien laitteiden tiedonsiirtonopeuden mukauttamisen sekä määrittää, kuinka laitteet havaitsevat ja reagoivat datan törmäyksiin. Kommunikointityyppi on full-duplex eli kommunikointi ja tiedonsiirto voi olla joko yksi- tai kaksisuuntaista. (10.)

Laitteiden kommunikoinnin Ethernet-verkossa mahdollistaa jokaisen yksilöllinen MAC-osoite. Kyseinen osoite koostuu kuudesta tavusta, joista kolme ensimmäistä on laitteen valmistajan oma etuliite ja kolme jälkimmäistä juokseva sarjanumero (13). MAC-osoite on yleensä jo tehtaalla valmiiksi asetettu, mutta joissakin tapauksissa sen voi myös vaihtaa ohjelmallisesti jälkikäteen. Osoitteen avulla dataliikenteessä identifioidaan, mistä laitteesta kyseinen datapaketti on tullut ja mihin se on menossa.

Jokaisella sovelluksella on omat vaatimuksensa datamäärän suhteen. Enimmillään yksi Ethernet/IP:n datapaketti voi sisältää jopa 1500 tavua dataa, jonka mahdollistaa TCP/IP- sekä UDP-protokollien hyödyntäminen eri tiedonsiirron vaiheissa. Protokollatyyppien yhdistäminen mahdollistaakin Ethernet/IP-protokollalle sen vahvuudet eli suuren datamäärän, ison suorituskyvyn sekä konfiguroinnin ja käytön helppouden. (10.)

TCP/IP on yhdistelmä monesta eri tietoliikenneprotokollasta. Tämä yhdistelmä käyttää IP-protokollaa (Internet Protocol), joka vastaa datapaketien reitittämisestä verkossa sekä laitteiden osoitteista. TCP-protokolla taas vastaa kahden laitteen välisestä kommunikoinnista ja datansiirrosta sekä hukkuneiden pakettien uudelleen lähettämisestä. TCP/IP:n käyttö tarjoaa Ethernet/IP-protokollalle toimivan kommunikoinnin ominaisuudet eli konfiguroinnin, osoitejärjestelmän sekä mekanismit yhteyden luomiseen laitteesta toiseen.

Ylemmillä tasoillaan Ethernet/IP käyttää CIP-protokollaa (Common Industrial Protocol), joka on yhteispohjainen protokolla. CIP viestityyppejä ovat explicit ja implicit. Eksplisiittisen viestin siirtoprotokolla on TCP/IP. Kyseinen viestityyppi sisältää niin pyyntö- kuin vastaustapahtumia ei-aikakriittisille datansiirroille, kuten konfigurointiparametrien lähetyksen. Implisiittisen viestin protokollana on UDP/IP. Tämä viestityyppi sisältää I/O-tiedonsiirron reaaliaikaiselle I/O-datalle, esimerkiksi kenttälaitteiden ohjaukset.

Kommunikointi CIP-protokollassa tapahtuu konfiguroitujen yhteyksien kautta, joten on määritelty, miten yhteys muodostetaan laitteiden välillä, jotka eivät vielä ole yhteydessä. Yhdistäminen tapahtuu UCMM eli Unconnected message manager -funktion avulla. CIP-yhteys muodostetaan lähettämällä UCMM Forward-open -pyyntö. Kyseinen pyyntö vaaditaan kaikilta laitteilta, jotka käyttävät Ethernet/IP-kommunikointia. (14.)

CIP CM Forward-open-datapaketti on siis yksi kommunikoinnin kättelypaketeista. Tämä datapaketti sisältää kaiken tiedon, mitä tarvitaan kommunikoinnin luontiin lähettäjäkohde välillä. Forward-Open pyynnön datapaketti sisältää pakollisia sekä valinnaisia tietoja. Pakollisia tietoja ovat määrittely-yhteyden aikakatkaisusta sekä CID-verkkoyhteyksistä lähettäjäkohde ja kohdevälittäjä välille, lähettäjälaitteen tunnus ja sarjanumero, suurin mahdollinen datakoko, määrittely yksi- tai monilähettyksestä, laukaisumekanismit (esimerkiksi syklinen tai tilanmuutos) sekä yhteyspolku sovellusobjektidatalle. Näiden lisäksi on valinnaisia tietoja, joita voidaan konfiguroida kommunikoinnille. Valinnaisia tietoja ovat elektroninen avain, reititystiedot sekä yksi datasegmentti. Elektronisen avaimen avulla kohde voi tarkistaa yhteensopivuuden lähettäjän kanssa. Reititystiedot määrittellään kommunikointiin, jos yhteyden on tarkoitus kattaa useampi kuin yksi verkko. Valinnainen datasegmentti sisältää kohteen konfigurointitiedot. (14.)

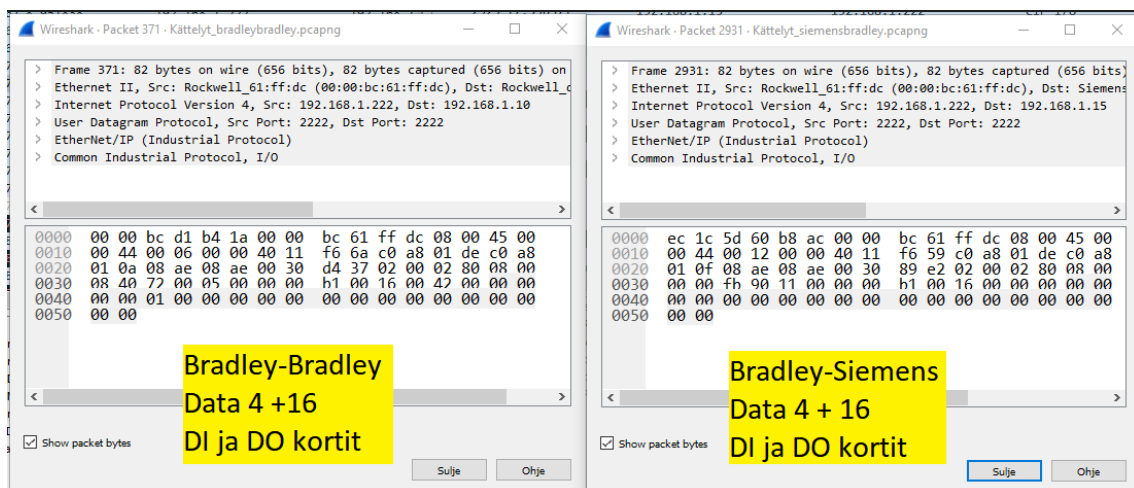
5.3.2 Protokollan datapakettien vertailut

Ethernet/IP-kommunikoinnissa on seuraavia datapaketteja: ARP, STP, LLDP, COTP, CIP, CIP CM, CIP I/O, TCP sekä ENIP.

Aluksi eroavaisuuksia lähdettiin etsimään CIP I/O -datapakettista, jossa varsinainen I/O-data liikkuu. CIP I/O -datapaketti on implisiittistä viestiä ja protokollana on UDP/IP. Kyseessä oleva datapaketti on tarkoitettu prosessidatan siirrolle. Näiden I/O-tietojen lisäksi paketti sisältää CIP-protokollan vaatimat perustiedot laitteista sekä kommunikoinnista kuten MAC- ja IP-osoitteet. Kuvassa 18 on esitetty datapaketin sisältö.

Tämän avulla pystyimmekin päättämään, minkä mittaisia datapaketteja Allen-Bradleyn Ethernet/IP-adapteri lähettää. Tämän tiedon avulla voitiin todeta, että pituus meidän testien konfiguraatiossa adapterin ja Siemensin logiikan välillä oli ollut oikea (LIITE 4). Laitekohtaiset tiedot olivat toki jokaisessa omansa, mutta varsinaiseen datan liikkumiseen vaikuttavia eroja ei löytynyt.

Alla olevassa kuvassa 20 on vertailussa CIP I/O -paketti toimivan ja ei-toimivan kokoonpanon välillä. Molemmista kokoonpanoissa Ethernet/IP:n adapteriin on kytketty yksi digitaalitulokortti sekä digitaalilähtökortti. Testauksessa on laitettu yksi tulokanava aktiiviseksi, jotta nähdään, miten data näkyy paketin sisällä. Varsinaisessa datansiirrossa on ensin neljä vakiotavua, jonka jälkeen I/O-data alkaa. Kuten kuvasta näemme Bradleyn laitteiden välillä aktiivinen tulo näkyy numerona 1 dataosion ensimmäisessä tavussa. Bradley-Siemens välinen testi on tehty juuri samanlaisella kokoonpanolla, mutta kuten näemme kuvasta, I/O-dataosio näyttää nollaa.



KUVA 20. CIP I/O datansiirron vertailu

Seuraavaksi lähdettiin tutkimaan muita kommunikoinnissa liikkuvia datapaketteja. CIP CM eli connection manager datapaketti lähetetään laitteiden kättelyissä. Kuvassa 21 on yhden CIP CM -datapaketin yleisnäkymä.

```

> Frame 85: 176 bytes on wire (1408 bits), 176 bytes captured (1408 bits) on interface \Device\NPF_{41695D31-C970-4209-9FB0-D477227B94AB}, id 0
> Ethernet II, Src: Rockwell_d1:b4:1a (00:00:bc:d1:b4:1a), Dst: Rockwell_61:ff:dc (00:00:bc:61:ff:dc)
> Internet Protocol Version 4, Src: 192.168.1.10, Dst: 192.168.1.222
> Transmission Control Protocol, Src Port: 55992, Dst Port: 44818, Seq: 1, Ack: 1, Len: 122
> EtherNet/IP (Industrial Protocol), Session: 0x0E020200, Send RR Data
> Common Industrial Protocol
> CIP Connection Manager

0000 00 00 bc 61 ff dc 00 00 bc d1 b4 1a 08 00 45 6c
0010 00 a2 9a 02 40 00 40 06 1b af c0 a8 01 0a c0 a8
0020 01 de da b8 af 12 a0 e4 6e 8b 7c 89 b2 32 50 18
0030 20 00 a5 a3 00 00 6f 00 62 00 00 02 02 0e 00 00
0040 00 00 00 00 00 00 07 00 00 80 00 00 00 00 00 00
0050 00 00 00 00 02 00 00 00 00 00 b2 00 52 00 54 02
0060 20 06 24 01 05 9c 00 00 00 00 08 40 72 00 07 80
0070 01 00 cb 12 83 00 01 00 00 00 20 4e 00 00 16 48
0080 20 4e 00 00 16 48 81 14 34 04 01 00 0c 00 5a 00
0090 84 01 70 04 24 03 7c 01 7c 07 80 0a 01 00 03 00
00a0 02 00 10 00 7d 0c 03 00 01 00 10 00 7d 0a 00 00

```

Show packet bytes

Sulje Ohje

KUVA 21. CIP CM datapaketti

Alla olevassa kuvassa 22 on CIP CM -paketin kolme ensimmäistä osiota tarkemmin avattuna. Ensimmäinen osio sisältää datapaketin perustiedot kuten päivämäärän ja kellonajan, paketin ja yhteyden muita aikatietoja, paketin pituuden, saadun datan pituuden sekä paketin sisältämät protokollatyypit. Toisessa osiossa on lähettäjän ja kohteen MAC-osoitteet ja kolmannessa lähettäjän ja kohteen IP-osoitteet, kokonaispituus, identifiointitiedot, protokollatyyppi sekä tarkistussumma.

```

▼ Frame 85: 176 bytes on wire (1408 bits), 176 bytes captured (1408 bits) on interface \Device\NPF_{41695D31-C970-4209-9FB0-D477227B94AB}
  Section number: 1
  > Interface id: 0 (\Device\NPF_{41695D31-C970-4209-9FB0-D477227B94AB})
  Encapsulation type: Ethernet (1)
  Arrival Time: Jul 27, 2023 12:08:01.010197000 Suomen kesäaika
  [Time shift for this packet: 0.000000000 seconds]
  Epoch Time: 1690448881.010197000 seconds
  [Time delta from previous captured frame: 0.033583000 seconds]
  [Time delta from previous displayed frame: 0.033583000 seconds]
  [Time since reference or first frame: 1.931845000 seconds]
  Frame Number: 85
  Frame Length: 176 bytes (1408 bits)
  Capture Length: 176 bytes (1408 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
  [Protocols in frame: eth:ethertype:ip:tcp:enip:cip:cipcm]
  [Coloring Rule Name: TCP]
  [Coloring Rule String: tcp]
  ▼ Ethernet II, Src: Rockwell_d1:b4:1a (00:00:bc:d1:b4:1a), Dst: Rockwell_61:ff:dc (00:00:bc:61:ff:dc)
    > Destination: Rockwell_61:ff:dc (00:00:bc:61:ff:dc)
    ▼ Source: Rockwell_d1:b4:1a (00:00:bc:d1:b4:1a)
      Address: Rockwell_d1:b4:1a (00:00:bc:d1:b4:1a)
      .... ..0. .... = LG bit: Globally unique address (factory default)
      .... ..0 .... = IG bit: Individual address (unicast)
      Type: IPv4 (0x0800)
  ▼ Internet Protocol Version 4, Src: 192.168.1.10, Dst: 192.168.1.222
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    > Differentiated Services Field: 0x6c (DSCP: Unknown, ECN: Not-ECT)
    Total Length: 162
    Identification: 0x9a02 (39426)
    > 010. .... = Flags: 0x2, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: TCP (6)
    Header Checksum: 0x1baf [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.1.10
    Destination Address: 192.168.1.222

```

KUVA 22. CIP CM sisällön kolme ensimmäistä osiota

Alla olevassa kuvassa 23 on kolme seuraavaa osiota. Ensimmäinen näistä on TCP-protokollaa, joka esimerkiksi sisältää lähettäjän ja kohteen porttiosoitteet, sekvenssinumeron sekä kuittausnumeron. Toinen osio on Ethernet/IP-protokollan tietoja ja kolmas CIP-protokollan tietoja sisältäen esimerkiksi kaksi määrittelevää datasegmentiä.

```

v Transmission Control Protocol, Src Port: 55992, Dst Port: 44818, Seq: 1, Ack: 1, Len: 122
  Source Port: 55992
  Destination Port: 44818
  [Stream index: 1]
  [Conversation completeness: Incomplete (12)]
  [TCP Segment Len: 122]
  Sequence Number: 1 (relative sequence number)
  Sequence Number (raw): 2699325067
  [Next Sequence Number: 123 (relative sequence number)]
  Acknowledgment Number: 1 (relative ack number)
  Acknowledgment number (raw): 2089398834
  0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x018 (PSH, ACK)
  Window: 8192
  [Calculated window size: 8192]
  [Window size scaling factor: -1 (unknown)]
  Checksum: 0xa5a3 [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  > [Timestamps]
  > [SEQ/ACK analysis]
  TCP payload (122 bytes)
  [PDU Size: 122]
v Ethernet/IP (Industrial Protocol), Session: 0x0E020200, Send RR Data
  > Encapsulation Header
  > Command Specific Data
v Common Industrial Protocol
  > Service: Unknown Service (0x54) (Request)
  Request Path Size: 2 words
  v Request Path: Connection Manager, Instance: 0x01
    > Path Segment: 0x20 (8-Bit Class Segment)
    > Path Segment: 0x24 (8-Bit Instance Segment)

```

KUVA 23. CIP CM sisällön kolme seuraavaa osiota

Viimeinen osio eli varsinainen CIP CM -osio avattuna näkyy kuvassa 24. Alussa määritellään, mikä paketin tarkoitus on, eli tässä tapauksessa Forward-open-pyyntö. Tämän jälkeen alkaa paketin varsinainen sisältö. Paketti määrittelee molempien datasuuntien verkkoyhteyden tunnistetiedot, lähettäjälaitteen sarjanumeron, toimittajan tunnisteen, datan lähetysaikavälin ja datasegmentit. Näistä kommunikoinnin testauksen vuoksi tarkastelussa on erityisesti datapaketissa liikkuvat data-segmentit.

```

CIP Connection Manager
  > Service: Forward Open (Request)
  > Command Specific Data
    ...0 .... = Priority: 0
    .... 0101 = Tick time: 5
    Time-out ticks: 156
    Actual Time Out: 4992ms
    O->T Network Connection ID: 0x00000000
    T->O Network Connection ID: 0x00724008
    Connection Serial Number: 0x8007
    Originator Vendor ID: Rockwell Automation/Allen-Bradley (0x0001)
    Originator Serial Number: 0x008312cb
    Connection Timeout Multiplier: *8 (1)
    Reserved: 0x000000
    O->T RPI: 20,000ms
  > O->T Network Connection Parameters: 0x4816
    T->O RPI: 20,000ms
  > T->O Network Connection Parameters: 0x4816
  > Transport Type/Trigger: 0x81, Direction: Server, Trigger: Cyclic, Class: 1
  Connection Path Size: 20 words
  > Connection Path: [Key], Assembly, Instance: 0x03, Connection Point: 0x01, Connection Point: 0x02, [Data]
    > Path Segment: 0x34 (Electronic Key Segment) (VendorID: 0x0001, DevTyp: 0x000C, 4.1)
    > Path Segment: 0x20 (8-Bit Class Segment)
    > Path Segment: 0x24 (8-Bit Instance Segment)
    > Path Segment: 0x2c (8-Bit Connection Point Segment)
    > Path Segment: 0x2c (8-Bit Connection Point Segment)
    > Path Segment: 0x80 (Simple Data Segment)
  [CIP Connection Index: 0]

```

KUVA 24. CIP CM viimeinen osio

Kuvassa 24 näkyy CIP CM -datapaketin Connection path -datasegmentin sisältö. Siemensin ja Allen-Bradleyn omien laitteiden välisessä kommunikaatiossa on viisi samaa segmenttiä. Bradleyn tuotteiden välisessä on näiden viiden segmentin lisäksi yksi valinnainen datasegmentti eli kuvassa 25 näkyvä Simple Data Segment. Kyseinen segmentti sisältää adapterin eli kohteen konfigurointi-tiedot eli määrittelee tässä tapauksessa adapterin perässä olevat I/O-kortit. Tätä pakettia ei lähe-tetä Siemensin laitteiden välillä, mistä löytyykin selitys projektin kokoonpanon kommunikoinnin toi-mimattomuudelle. Siemensin väliset laitteet eivät vaadi kyseistä pakettia, koska molemmat ovat älylaitteita itsessään sekä laitteet voidaan konfiguroida TIA Portaalin avulla oikein.

Bradley-Bradey

- ∨ Connection Path: [Key], Assembly, Instance: 0x03, Connection Point: 0x01, Connection Point: 0x02, [Data]
 - > Path Segment: 0x34 (Electronic Key Segment) (VendorID: 0x0001, DevTyp: 0x000C, 4.1)
 - > Path Segment: 0x20 (8-Bit Class Segment)
 - > Path Segment: 0x24 (8-Bit Instance Segment)
 - > Path Segment: 0x2c (8-Bit Connection Point Segment)
 - > Path Segment: 0x2c (8-Bit Connection Point Segment)
 - > Path Segment: 0x80 (Simple Data Segment)

Bradley-Siemens

- ∨ Connection Path: [Key], Assembly, Instance: 0x17, Connection Point: 0x02, Connection Point: 0x03
 - > Path Segment: 0x34 (Electronic Key Segment) (VendorID: 0x0001, DevTyp: 0x000C, 4.1)
 - > Path Segment: 0x20 (8-Bit Class Segment)
 - > Path Segment: 0x24 (8-Bit Instance Segment)
 - > Path Segment: 0x2c (8-Bit Connection Point Segment)
 - > Path Segment: 0x2c (8-Bit Connection Point Segment)

Siemens-Siemens

- ∨ Connection Path: [Key], Assembly, Instance: 0x68, Connection Point: 0x66, Connection Point: 0x65
 - > Path Segment: 0x34 (Electronic Key Segment) (VendorID: 0x04E3, DevTyp: 0x000C, 1.4)
 - > Path Segment: 0x20 (8-Bit Class Segment)
 - > Path Segment: 0x24 (8-Bit Instance Segment)
 - > Path Segment: 0x2c (8-Bit Connection Point Segment)
 - > Path Segment: 0x2c (8-Bit Connection Point Segment)

KUVA 25. Connection path datasegmentit

6 JÄRJESTELMÄVAIHDON TOTEUTUS

Kun testaukset on tehty ja väyläkommunikointi todettu toimivaksi, voidaan aloittaa ohjelman tekeminen esisuunnitteluvaiheessa kerättyjen lähtötietojen avulla. Suunnittelun ja ohjelmoinnin kulku on esitetty seuraavassa luvussa. Sitä ei kuitenkaan tämän opinnäytetyön aikataulun puitteissa ehditty tekemään.

6.1 Suunnittelu ja ohjelmointi

Ohjelmoinnin aluksi luodaan HW-konfiguraatio projektille. Tämä tarkoittaa kaikkien komponenttien lisäämistä ohjelmaan eli logiikan, väyläyksiköiden, I/O-korttien sekä muiden mahdollisten laitteiden kuten HMI-paneelin. Laitteet sijoitetaan ohjelmaan oikeaan järjestykseen, luodaan väyläkaavio sekä asetetaan IP-osoitteet logiikalle, väyläyksiköille sekä paneelille. Näiden lisäksi konfiguroidaan koko järjestelmän I/O. I/O:n konfigurointi tarkoittaa kaikkien tulo- ja lähtökanavien osoitteiden antamista sekä nimeämistä.

Kun konfiguraatio ohjelmassa on valmis, voidaan aloittaa koodin kirjoittaminen. Tässä projektissa tiedetään, kuinka paljon apumuistipaikkoja on käytetty nykyisen ohjelman ohjelmakopion perusteella, joten ne voidaan lisätä vielä ohjelmaan ennen varsinaista koodin kirjoitusta. Kun kaikki tarvittavat tagit on tehty valmiiksi, ohjelman kirjoitus helpottuu huomattavasti. Koska tässä projektissa on nykyisestä ohjelmasta saatu ohjelmakopio käytettävissä, voidaan uusi ohjelma kirjoittaa sitä mukailleen.

Viimeisenä tehdään valvomo- ja HMI-suunnittelu. Logiikan sekä valvomon ja paneelin välillä siirtyville tiedoille luodaan omat tagit. Tagien luonnin jälkeen tehdään valvomo- ja paneelikuvat.

Kaikkien edellä mainittujen vaiheiden jälkeen voidaan tehdä ohjelmallinen FAT-testi, jossa testataan simulointiympäristössä ohjelman toimivuus. Kun ohjelma on testattu simuloimalla onnistuneesti, voidaan tehdä varsinainen käyttöönotto. Tässä projektissa käyttöönoton aikataulu on tiukka, joten kaikki testataan, mitä voidaan testata toimivaksi ennen varsinaista käyttöönottoa.

6.2 Jatkoimenpiteet

Tämän opinnäytetyön aikataulun puitteissa varsinaista toteutusosaa ei ehditty tekemään. Siemensin kanssa käytiin keskustelua epäselvyyksistä sekä löytämistämme puutteista koko testauksen ajan. Siemens totesi myös heidän kirjastostaan löytyvän puutoksen liittyen modulaaristen kokoonpanojen konfiguraatioon, jotka tarvitsevat korttikohtaisen konfiguroinnin.

Ethernet/IP-lohkokirjastosta on tällä hetkellä kehitteillä uusi ohjelmakirjasto modulaarisia konfiguraatioita varten, joka on tarkoitus julkaista applikaatioesimerkkinä seuraavan vuoden aikana. Aikataulusyistä päivitetty kirjastoversio ei kuitenkaan ehdi tämän projektin tarpeisiin. Tämän vuoksi joudumme vaihtamaan projektissa käytettävän logiikan tuotemerkkiä. Uusi logiikka tulee olemaan Allen-Bradleyn tuotemerkiltä, joka on vastaavanlainen kuin kommunikointitestauksessa käyttämämme testilogiikka.

Kommunikointi uuden logiikan ja adapterin välillä on testattu toimivaksi aikaisemmissa esitetyissä testeissä, joten yhteensopivuuden kanssa ei tule ongelmia projektin myöhemmässä vaiheessa. Vaikka logiikkamerkki vaihtui, suunniteltu Siemensin HMI-paneeli sekä valvomosovelluksen päivitys tulee pysymään ennallaan. HMI-paneelin ja uuden logiikan välinen kommunikointi on testattu toimivaksi, mutta valvomosovelluksen ja uuden logiikan välinen kommunikointi on vielä testattava.

7 YHTEENVETO

Opinnäytetyön tavoitteena oli toteuttaa prosessiohjauksen automaatiojärjestelmän vaihto. Aikaisempaa kokemusta järjestelmävaihdosta ei juurikaan ollut, joten tartuin mielenkiinnolla projektiin. Projektin alku sisälsi aiheeseen perehtymistä saatujen aineistojen avulla sekä lähtötietojen keräämistä. Lähtötietojen keräämisen jälkeen alkoivat kommunikoinnin testaukset.

Kommunikointiprotokollat olivat ennestään tuttuja yleisellä tasolla, mutta testauksen edetessä ilmaantuneiden ongelmien ratkaisujen löytymiseksi pääsin perehtymään tarkemmin etenkin Ethernet/IP-protokollaan ja sanomarakenteisiin. Protokollan tarkempi sisältö ei ollut ennestään tuttua, joten aikaa vierähti huomaamatta aiheen opiskeluun ja tutkimiseen. Kommunikointiprotokollien ja niiden sanomarakenteiden sisällön ymmärtämisestä on varmasti hyötyä myös tulevilla projekteilla.

Aikataulullisesti projekti viivästyi suunniteltua pidempään, mikä johtui juuri kommunikoinnin testauksen syventymisestä, kun kaikki suunnitelmat eivät toimineetkaan odotetusti. Aikaa kului niin ilmenneiden ongelmien ratkaisujen etsimiseen kuin varmistuksien ja selvitysten odotteluun. Opinnäytetyön aikataulun puitteissa testaukset saatiin suoritettua ja löydettiin kommunikointiongelmiin selitys. Alkuperäiseen suunnitelmaan tuli tämän myötä muutoksia uuden kokoonpanon logiikan tuotemerkkiin liittyen.

Suurimpaan osaan asetetuista tavoitteista kuitenkin päästiin viivästyksistä huolimatta: lähtötiedot saatiin kerättyä nykyisestä järjestelmästä, korvaavat komponentit ja väyläratkaisu valittua sekä testattua kommunikoinnin toimivuus uuden kokoonpanon komponenttien välillä. Varsinainen ohjelmointityö kuitenkin jäi myöhempään ajankohtaan opinnäytetyön aikataulun vuoksi, mutta tämän opinnäytetyön aikana kerättyjen pohjatietojen avulla ohjelmointityön teko helpottuu.

LÄHTEET

1. Apex Automation 2023. Yritys. Hakupäivä 13.10.2023. <https://apexautomation.fi/yritys/>.
2. Siemens 2023. Totally Integrated Automation Portal – Always ready for tomorrow. Hakupäivä 14.10.2023. <https://www.siemens.com/global/en/products/automation/industry-software/automation-software/tia-portal.html>.
3. Wireshark 2023. About Wireshark. Hakupäivä 14.10.2023. <https://www.wireshark.org/about.html>.
4. ABB 2023. Alfa Laval Automation. Hakupäivä 13.10.2023. <https://global.abb/group/en/about/history/heritage-brands/alfa-laval-automation>.
5. ABB 2023. SattCon 200 hajautettuun Satt-ohjausjärjestelmään. Hakupäivä 13.10.2023. <https://new.abb.com/control-systems/fi/palvelut/j%C3%A4rjestelmille/satt/ohjaimet/satt-con-200>.
6. ABB 2023. SattCon 05 ja SattCon 05 Slimline hajautettuun Satt-ohjausjärjestelmään. Hakupäivä 29.9.2023. <https://new.abb.com/control-systems/fi/palvelut/j%C3%A4rjestelmille/satt/ohjaimet/sattcon-05>.
7. Siemens 03/2023. Operating instructions Simatic ET 200 SP Open Controller. Hakupäivä 16.10.2023. https://cache.industry.siemens.com/dl/files/551/109758551/att_966545/v2/cpu_1515sp2_pc_manual_en-US_en-US.pdf.
8. Siemens 2023. 6ES7677-2SB42-0GB0. Hakupäivä 16.10.2023. <https://mall.industry.siemens.com/mall/en/WW/Catalog/Product/6ES7677-2SB42-0GB0>.
9. ODVA 2023. ControlNet, CIP on CTDMA Technology. Hakupäivä 10.11.2023. <https://www.odva.org/wp-content/uploads/2020/05/PUB00200R1-Tech-Series-Control-Net.pdf>.

10. ODVA 2023. Ethernet/IP, CIP on Ethernet Technology. Hakupäivä 10.11.2023. https://www.odva.org/wp-content/uploads/2021/05/PUB00138R7_Tech-Series-EtherNetIP.pdf.
11. Siemens 2023. ET 200SP. Hakupäivä 16.11.2023. <https://mall.industry.siemens.com/mall/en/WW/Catalog/Products/10170367>.
12. ABB Satt AB 1997-1999. SattCon 200; Programmer's Manual; SattCon 200 Version 2.5.
13. TechTarget 2023. MAC address (media access control address). Hakupäivä 16.11.2023. <https://www.techtarget.com/searchnetworking/definition/MAC-address>.
14. Hilscher 02/2020. Capturing Ethernet/IP with Wireshark manual.

LIITTEET

LIITE 1 WIRESHARK YLEISNÄKYMÄ DATAPAKETEISTA

LIITE 2 CIP CM -DATAPAKETIN SISÄLTÖ

LIITE 3 S7-1200-LOGIIKAN PARAMETRIT ETHERNET/IP TESTAUKSESSA

LIITE 4 ALLEN-BRADLEY ETHERNET/IP 1794-AENT -ADAPTERIN PARAMETRIT

CIP I/O																			
Destination MAC address							Source MAC address						IP type		IPv4 header length	IPv4 diff. service field			
hex	0	0	bc	61	ff	dc	ec	1c	5d	60	b8	ac	8	0	45	0			
dec	0	0	188	97	225	220	236	28	93	96	184	172	8	0	69	0			
Total IPv4 lenght		IPv4 identific.		IPv4 flags		IPv4 time to live	IPv4 protocol UDP	IPv4 hader checksum		Source IP address				Destination -					
0 44		3d 65		0 0		40	11	b9 0b		c0 a8		1 0a		c0 a8					
68		15717		0 0		64	17	47371		192 168		1 10		192 168					
IP address		UDP source port		UDP dest. Port		UDP lenght		UDPchecksum		EIP item count		EIP type ID		EIP lenght					
1 de		8 ae		8 ae		0 30		ac 34		2 0		2 80		8 0					
1 222		2222		2222		0 48		44084		2 0		0x0280		8 0					
EIP connection ID				EIP encapsulation sequence number				EIP type ID		EIP type ID lenght		EIP sequence count		Data (32-bit header)					
81 3 27 0				1e 20 5 0				b1 0		16 0		1e 20		1 0					
0x00270381				335902				0x00b1		22 0		8222		1 0					
Data																			
0 0		0 0		0 0														
0 0		0 0		0 0															
vakio = 2 tavua																			
<table border="1" style="width:100%; border-collapse: collapse;"> <tr> <td style="width: 33%;">Sequence count value</td> <td style="width: 33%;">Real-time header</td> <td style="width: 33%;">I/O data</td> </tr> </table>																	Sequence count value	Real-time header	I/O data
Sequence count value	Real-time header	I/O data																	
<table border="1" style="width:100%; border-collapse: collapse;"> <tr> <td>32-bit Header 0 -> T originator to target data only</td> </tr> </table>																	32-bit Header 0 -> T originator to target data only		
32-bit Header 0 -> T originator to target data only																			
<table border="1" style="width:100%; border-collapse: collapse;"> <tr> <td>EIP type ID 1. Sequenced address item 2. Connected data item</td> </tr> </table>																	EIP type ID 1. Sequenced address item 2. Connected data item		
EIP type ID 1. Sequenced address item 2. Connected data item																			

[Device]

```
VendCode = 1251;
VendName = "Siemens AG";
ProdType = 12;
ProdTypeStr = "Communications Adapter";
ProdCode = 1200;
MajRev = 1;
MinRev = 4;
ProdName = "SIMATIC S7-1200 Controller";
Catalog = "6ES7 21x-1xx40-0XB0";
Icon = "1200.ico";
```

```
Param101 =
0, $ reserved, shall equal 0
,, $ Link Path Size, Link Path
0x0000, $ Descriptor
0xC7, $ Data Type
2, $ Data Size in bytes
"inputDataSize", $ name
"Sint", $ units
"Input Data Size in Sint", $ help string
0,502,64, $ min, max, default data values
,,,, $ mult, div, base, offset scaling
,,,, $ mult, div, base, offset links
,,,, $ decimal places
Param102 =
0, $ reserved, shall equal 0
,, $ Link Path Size, Link Path
0x0000, $ Descriptor
0xC7, $ Data Type
2, $ Data Size in bytes
"outputDataSize", $ name
"Sint", $ units
"Output Data Size in Sint", $ help string
0,502,64, $ min, max, default data values
,,,, $ mult, div, base, offset scaling
,,,, $ mult, div, base, offset links
,,,, $ decimal places
```

```
Param104 =
0, $ reserved, shall equal 0
,, $ Link Path Size, Link Path
0x0000, $ Descriptor
0xC7, $ Data Type
2, $ Data Size in bytes
"configDataSize", $ name
"Sint", $ units
"Configuration Data Size in Sint", $ help string
0,502,64, $ min, max, default data values
,,,, $ mult, div, base, offset scaling
,,,, $ mult, div, base, offset links
,,,, $ decimal places
```

AdaptersSettings[0]	"LCCF_typeEnetAdaptConfig"		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		contains required information about the EtherNet/IP adapte
address	IP_V4		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		IP v4 address of the adapter to access
vendorIdentifier	UInt	1251	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		ODVA assigned Vendor ID (from EDS files)
productType	UInt	12	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		ODVA assigned ProductType number (from EDS files)
productCode	UInt	1200	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		ODVA assigned ProductCode (from EDS files)
revisionMajor	USInt	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		MajorRevision
revisionMinor	USInt	4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		MinorRevision
assemblyInstan...	UInt	104	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Instance the Assembly can be reached at (from EDS file)
packetInterval	Time	t#100ms	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Requested Packet Interval/ Update Rate
conPointOutput...	UInt	102	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Connection Point Identifier O->T direction
conPointOutput...	UInt	64	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		[Bytes] Connection Point Size in Bytes O->T direction
conPointInput...	UInt	101	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Connection Point Identifier T->O direction
conPointInputS...	UInt	64	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		[Bytes] Connection Point Size in Bytes O->T direction
configDataSize	UInt	64	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		[Bytes] Configuration data size of the assembly instance

```
[Device]
VendCode      = 1;
VendName      = "Allen-Bradley";
ProdType      = 12;
ProdTypeStr   = "Communication Adapter";
ProdCode      = 90;
MajRev        = 4;
MinRev        = 1;
ProdName      = "1794-AENT FLEX I/O Ethernet Adapter";
Catalog       = "1794-AENT";
Icon          = "1794acn.ico";
1_1794_Legacy = Yes;

Param2 =
0,          $ first field shall equal 0
,,         $ path size,path
0x0000,    $ descriptor
0xC7,     $ data type : 16-bit Unsigned Integer
2,        $ data size in bytes
"Output Size", $ name
" ",      $ units
" ",      $ help string
0,16,16,  $ min,max,default data values
,,,      $ mult,dev,base,offset scaling not used
,,,      $ mult,dev,base,offset link not used
0;       $ decimal places not used
```

```
Param3 =
0,          $ first field shall equal 0
,,         $ path size,path
0x0000,    $ descriptor
0xC7,     $ data type : 16-bit Unsigned Integer
2,        $ data size in bytes
"Input Size", $ name
" ",      $ units
" ",      $ help string
4,20,20,  $ min,max,default data values
,,,      $ mult,dev,base,offset scaling not used
,,,      $ mult,dev,base,offset link not used
0;       $ decimal places not used
```

```
Param23 =
0,          $ first field shall equal 0
,,         $ path size,path
0x0210,    $ descriptor (read-only, EDS internal-only)
0xC7,     $ data type : 16-bit Unsigned Integer
2,        $ data size in bytes
"size of config data", $ name
" ",      $ units
" ",      $ help string
0,65535,0, $ min,max,default data values
,,,      $ mult,dev,base,offset scaling not used
,,,      $ mult,dev,base,offset link not used
0;       $ decimal places not used
```

AdaptersSettings[2]		*LCCF_typeEnetAdaptConfig*		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
address	IP_V4			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	contains required information about the EtherNet/IP adapte
vendorIdentifier	UInt	1		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	IP v4 address of the adapter to access
productType	UInt	12		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	ODVA assigned Vendor ID (from EDS files)
productCode	UInt	90		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	ODVA assigned ProductType number (from EDS files)
revisionMajor	USInt	4		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	ODVA assigned ProductCode (from EDS files)
revisionMinor	USInt	1		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	MajorRevision
assemblyInstance	UInt	23		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	MinorRevision
packetInterval	Time	T#100MS		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Instance the Assembly can be reached at (from EDS file)
conPointOutputIdentifier	UInt	2		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Requested Packet Interval/ Update Rate
conPointOutputSize	UInt	16		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Connection Point Identifier O->T direction
conPointInputIdentifier	UInt	3		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	[Bytes] Connection Point Size in Bytes O->T direction
conPointInputSize	UInt	20		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Connection Point Identifier T->O direction
configDataSize	UInt	0		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	[Bytes] Connection Point Size in Bytes O->T direction
				<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	[Bytes] Configuration data size of the assembly instance