



# Laadunvalvontajärjestelmän tiedonsiirtomenetelmien kehittäminen

Jarno Pasonen

Opinnäytetyö, AMK

Toukokuu 2021

Tekniikan ala

Automaatiotekniikan tutkinto-ohjelma



# jamk

**Pasonen, Jarno**

## **Laadunvalvontajärjestelmän tiedonsiirtomenetelmien kehittäminen**

Jyväskylä: Jyväskylän ammattikorkeakoulu. **Toukokuu 2021**, 39 sivua

Tekniikan ala. Automaatiotekniikan tutkinto-ohjelma. Opinnäytetyö AMK.

Julkaisun kieli: suomi

Verkkojulkaisulupa myönnetty: kyllä

### **Tiivistelmä**

Ethernet-pohjaiset väylät ovat yleistyneet myös teollisuudessa viime vuosina, ja korvanneet sekä vanhoja väyläratkaisuja, että perinteistä kovajohdotettua I/O:ta. Tutkimuksen toimeksiantajalla Mapvisionilla on tämän myötä syntynyt tarve uudistaa laitteidensa ohjausjärjestelmiä. Mapvision valmistaa mittalaitteita autoteollisuudelle mekaanisten osien laadunvalvontaan. Laitteiden ohjausrajapinta on toiminut alusta saakka perinteisellä I/O:lla, mutta sen rajoitteet ovat alkaneet tulla hiljalleen vastaan. Tämän lisäksi laitteiston sisäisiin komponentteihin on tehty päivityksiä, jotka vaativat vanhojen väyläratkaisujen uudelleenarviointia.

Tutkimus jaettiin kahteen erilliseen osioon, joista molemmat liittyvät samaan laitteistoon ja keskittyvät tiiviisti Ethernet-pohjaisiin teollisuusväyliin. Tutkimuksen ensimmäisen osion tavoitteena oli selvittää komponenttipäivityksestä seuranneita ongelmia Modbus-yhteydessä, sekä selvittää vaihtoehtoisia ratkaisuja kyseiselle väylälle. Toisessa osiossa puolestaan keskityttiin kehittämään toimivaa teolliseen Ethernet-väylään perustuvaa ohjausrajapintaa, perinteisen I/O rajapinnan korvaajaksi. Tavoitteena oli selvittää, millä tavoin rajapinta toteutettaisiin, ja millä protokollalla tiedonvälitys tapahtuisi.

Tutkimus tehtiin kvalitatiivisena tutkimuksena. Siinä selvitettiin eri Ethernet-pohjaisten teollisuusprotokollien sopivuutta toimeksiantajan tarpeisiin, sekä tapoja näiden protokollien käyttöönottoon käytännössä. Arvioinnit perustuivat protokollien omistavien tahojen viralliseen materiaaliin, kuten standardeihin, ohjeisiin ja esitteisiin.

Tutkimuksesta saatiin paljon tietoa eri vaihtoehtoista ja niiden yksityiskohdista, sekä siitä mitä näiden vaihtoehtojen toteutus käytännössä vaatii. Toimeksiantaja sai kehitettyä tuotteitaan eteenpäin tämän tutkimuksen avulla, ja tutkimus antaa myös hyvää lähtötietoa jatkokehitykselle.

Johtopäätös tutkimuksen ensimmäisestä osiosta on, että nykyinen Modbus-yhteys on toimiva ja riittävä käyttötarkoituksiin, ja että sen korvaaminen Profinetillä on mahdollista, mutta vaatii lisää kehitystyötä. Toisen osion johtopäätös on puolestaan se, että Ethernet-pohjainen tiedonsiirto voidaan toteuttaa ohjelmallisesti laitteistossa olevaan mittaustietokoneeseen. Modbus-protokolla on tähän sopiva ratkaisu, mutta myös muita vaihtoehtoja on olemassa, mikäli niitä halutaan tulevaisuudessa ottaa käyttöön.

### **Avainsanat (asiasanat)**

Ethernet, ohjausrajapinta, teollinen Ethernet, kenttäväylä, Modbus TCP/IP, Profinet, OPC UA, EtherNet/IP

### **Muut tiedot (salassa pidettävät liitteet)**

-

**Pasonen, Jarno**

### **Development of quality control system communication methods**

Jyväskylä: JAMK University of Applied Sciences, April 2021, 39 pages

Engineering and technology. Degree Programme in Automation Technology. Bachelor's thesis.

Permission for web publication: Yes

Language of publication: Finnish

### **Abstract**

Industrial Ethernet has gained popularity in recent years and have replaced older fieldbus solutions, as well as traditional hardwired I/O systems. Because of this, the assignor of the research, Mapvision, has slowly started to gain demand for upgrade of the control systems of their machines. Mapvision manufactures measuring machines for quality control of mechanical parts in automotive industry. From the beginning the machines have used traditional hardwired I/O as their control interface but it has started to reach its limits. Additionally, upgrades have been made to the internal components of the machine which has created a need for evaluation of the old data bus solutions.

The research is divided into two separate sections. Both of them are concerned about the same machine and both revolve tightly around industrial Ethernet protocols. The goal of the first section was to figure out the problems in the Modbus connection followed by the internal component upgrade and also find out possible alternatives for the Modbus connection. The second section focused on developing working solution to replace the traditional I/O control interface with industrial Ethernet. The goal was to find out how it can be realized and what protocols it should support.

The research was accomplished as a qualitative research. Different industrial Ethernet protocols were evaluated for suitability for the needs of Mapvision and it was also investigated how to implement them in practice. The evaluation was based in official materials of the protocol owners, like standards, manuals, and brochures.

A lot of information was gained of the different possibilities and details of those and also what is needed in practice to implement some of them. The assignor managed to develop their product further on with the results of this research and the research also gives good base information for the future developments.

Conclusion of the first section is that current Modbus connection is working and sufficient for the current use of it and that it is also possible to replace it with a Profinet connection but that would require more work. Conclusion of the second sections is that industrial Ethernet control interface is possible to implement programmatically into measuring computer in the measuring machine. Modbus was a suitable protocol for this but other protocols are also available in case there is need for them.

### **Keywords/tags (subjects)**

Ethernet, control interface, industrial Ethernet, fieldbus, Modbus TCP/IP, Profinet, OPC UA, EtherNet/IP

### **Miscellaneous (Confidential information)**

-

## Sisältö

<b>Käsitteet</b> .....	<b>3</b>
<b>1 Johdanto</b> .....	<b>4</b>
1.1 Toimeksiantaja .....	5
<b>2 Lähtökohdat</b> .....	<b>6</b>
2.1 Vaihtoehdot sisäiselle Modbus-yhteydelle .....	6
2.2 Ulkoisen rajapinnan toteutus kenttäväylänä .....	7
2.3 ET 200SP analogiakorttien lisääminen ilman lisätöitä .....	8
<b>3 Tutkimusasetelma</b> .....	<b>9</b>
3.1 Tutkimuskysymykset .....	9
3.2 Tutkimusote .....	9
3.3 Aineisto .....	9
<b>4 Ethernet-pohjaiset teollisuusprotokollat</b> .....	<b>10</b>
4.1 Modbus TCP/IP .....	11
4.2 Profinet.....	12
4.2.1 Profinetin ominaisuuksia.....	12
4.3 OPC UA .....	14
4.4 EtherNet/IP .....	16
<b>5 Vaihtoehdot sisäiselle Modbus-yhteydelle</b> .....	<b>18</b>
5.1 Vaatimukset .....	18
5.1.1 Mittauslaitteiston eri versiot.....	18
5.1.2 Olemassaolevat Profinet-laitteet .....	20
5.1.3 Vasteaika.....	20
5.1.4 Järjestelmä .....	20
5.2 Modbus-yhteyden ongelmat .....	20
5.3 Korvaaja Modbus-yhteydelle.....	21
5.3.1 Vaihtoehto 1: Siemensin Profinet ajuri .....	21
5.3.2 Vaihtoehto 2: Kolmannen osapuolen Profinet ajurit.....	22
5.3.3 Vaihtoehto 3: Pysytään Modbus-yhteydessä .....	22
5.3.4 Vaihtoehto 4: Snap7.....	23
5.3.5 Vaihtoehto 5: OPC UA .....	23
5.3.6 Vaihtoehto 6: Siemens Open Controller.....	23
5.4 Lopputulos.....	23

<b>6</b>	<b>Ulkoisen rajapinnan toteutus kenttäväylänä .....</b>	<b>24</b>
6.1	Kommunikaatiokirjasto .....	26
6.2	Testaus .....	28
6.3	Haasteet .....	28
6.4	Huomiot .....	29
6.4.1	Eroavaisuudet perinteisen I/O:n ja Ethernet-pohjaisen kommunikoinnin välillä ...	29
6.4.2	Osoiteavaruus .....	31
6.4.3	Tiedonsiirtotavat .....	32
6.4.4	Laajennettavuus .....	33
6.4.5	Lisenssi, testaaminen ja sertifiointi .....	33
<b>7</b>	<b>Pohdinta .....</b>	<b>34</b>
	<b>Lähteet .....</b>	<b>37</b>

## Kuviot

Kuvio 1.	Mapvisionin mittauslaitteisto .....	6
Kuvio 2.	Sisäinen Modbus-yhteys mittauslaitteistossa .....	7
Kuvio 3.	Ulkoinen rajapinta tehdasverkkoon on toteutettu perinteisellä I/O:lla .....	8
Kuvio 4.	Sisäinen Modbus-yhteys versiossa, jossa ei välttämättä tarvita ohjelmoitavaa logiikkaa	19
Kuvio 5.	Sisäinen Modbus-yhteys versiossa, jossa on mukana kappaleen syöttämiseen tarvittava laitteisto .....	19
Kuvio 6.	Vaihtoehto 1. Ethernet-väylän toteutukselle mittauslaitteistossa .....	25
Kuvio 7.	Vaihtoehto 2. Ethernet-väylän toteutukselle mittauslaitteistossa .....	25
Kuvio 8.	Kommunikaatiomodulin rakenne ja suhde muuhun mittausohjelmaan .....	26

## Käsitteet

**Perinteinen I/O:** Tiedonsiirtotapa, jossa tieto siirretään erillisillä johtimilla, joista jokaisessa kulkee yksi signaali.

**Väylä:** Tiedonsiirtotapa, jossa tietoa siirretään digitaalisesti useiden laitteiden välillä. Yhtä johdinta pitkin voidaan siirtää teoriassa rajaton määrä eri tietoja.

**Kenttäväylä:** Tiedonsiirtoväylä, joka on erityisesti suunniteltu käytettäväksi teollisuudessa laitteiden ohjaamiseen ”kentällä” eli esimerkiksi teollisuushallissa.

**OSI-malli:** Open Systems Interconnection Reference Model on ajatusmalli, joka jakaa tiedonsiirron seitsemään eri kerrokseen, joilla on jokaisella oma tehtävänsä. Sen voidaan ajatella olevan abstraktio protokollapinosta.

**Tiedonsiirtoprotokolla:** standardi tai muu käytäntö, jossa on sovittu millä tavalla kahden tietokoneen välinen tiedonsiirto tapahtuu.

**Protokollapino:** Joukko protokollia, jotka toimivat yhdessä ja joista jokaisella on oma tehtävänsä.

**Ethernet:** Tiedonsiirtoprotokolla, joka vastaa tiedon välittämisestä kahden samaan väylään kytketyn laitteen välillä, ja toimii OSI-mallin kerroksessa 2.

**IP:** Internet Protocol on protokolla, joka vastaa tiedon kulkemisesta oikeaan osoitteeseen laajemmassa verkossa, ja toimii OSI-mallin kerroksessa 3.

**TCP:** Transmission Control Protocol on protokolla, jonka tehtävä on pitää huoli tiedon välittämisestä perille luotettavasti, ja joka toimii OSI-mallin kerroksessa 4.

**UDP:** User Datagram Protocol on protokolla, jonka tehtävänä on siirtää tietoa mahdollisimman nopeasti. UDP toimii OSI-mallin kerroksessa 4, ja on yleinen vaihtoehto TCP:lle silloin kun halutaan varmuuden sijaa nopeutta.

**TCP/IP:** Protokollapino, jota käytetään yleisesti siirtämään tietoa internetissä. Pino sisältää muitakin protokollia kuin TCP ja IP, mutta nimi tulee siitä, että tämä on yleisin yhdistelmä.

**Tuottaja-kuluttaja-malli:** Tiedonsiirtomalli, jossa laite (tuottaja) lähettää etukäteen sovitusti tietoa toiselle laitteelle (kuluttaja), joka ainoastaan ottaa tietoa vastaan. Voidaan käyttää UDP-protokollan kanssa.

**Julkaisija-tilaaja-malli:** Sama kuin tuottaja-kuluttaja-malli, mutta tiedonsiirron aloitus tehdään erillisellä pyynnöllä. Käytännössä saattaa kuitenkin olla täysin identtinen.

**Asiakas-palvelin-malli:** Tiedonsiirtomalli, jossa laite (asiakas) tekee pyynnön toiselle laitteelle (palvelin), johon se reagoi jollakin tavalla ja lähettää takaisin vastauksen. Käytetään usein TCP-protokollan kanssa.

**HMI:** Human-Machine Interface on laitteen ja ihmisen välinen rajapinta, usein esimerkiksi kosketusnäyttö.

**Ohjelmoitava logiikka:** (engl. Programmable Logic Controller, PLC) on pienikokoinen tietokone, joka on suunniteltu erityisesti teollisuusprosessien ohjaamiseen.

## 1 Johdanto

Erilaiset kenttäväylät ovat olleet jo pitkään yleisessä käytössä teollisuudessa. Niissä on lukuisia etuja verrattuna perinteiseen ”kovajohdotettuun” I/O:hon, kuten laajennettavuus ja pienempi määrä johdottamista. Markkinoilla on kuitenkin useita keskenään kilpailevia, ja enemmän tai vähemmän epäyhteensopivia väyliä. Varsinkin aikaisemmin jopa väylän fyysinen kerros (kaapeli, jännitetasot, yms.) riippui valitusta väylätekniikasta, mutta sittemmin toimistomaailmastakin tuttu Ethernet on alkanut yleistymään myös teollisuusympäristöissä (Zhihong & Pearson 2018, 2 - 4). Ethernet yhtenäistää kenttäväylän fyysisen kerroksen ja helpottaa näin suuresti väylätekniikasta toiseen vaihtamista.



Toimijalla, jonka asiakkaita ovat useat eri tuotantolaitokset, tulisi olla ajantasainen ja universaali tapa liittyä tehtaiden ohjausjärjestelmiin. Perinteinen I/O on ollut toimiva ratkaisu, sillä sille löytyy tuki käytännössä mistä tahansa järjestelmästä. Samaan aikaan se on kuitenkin kankea, ja sen kautta siirrettävä tiedon määrä on varsin rajallinen. Tässä kohtaa Ethernet astuu kuvioihin. Sillä voidaan toteuttaa nykyaikainen, helposti laajennettavissa oleva ja kustannustehokas ohjausraja-pinta, joka pystyy tukemaan useita eri väyläteknikoita.

Tässä opinnäytetyössä vertaillaan eri Ethernet-pohjaisten teollisuusväyläprotokollien tuomia mahdollisuuksia ja rajoitteita toimeksiantajan järjestelmässä. Työ koostuu kahdesta löyhästi toisiinsa liittyvästä osiosta. Ensimmäisessä osiossa tarkastellaan mahdollista korvaajaa laitteiston sisäiselle Modbus-yhteydelle, ja toisessa osiossa keskitytään nykyaikaistamaan laitteiston ulkoinen perinteisellä I/O:lla toteutettu rajapinta.

## **1.1 Toimeksiantaja**

Opinnäytetyön toimeksiantaja on Mapvision oy. Mapvision valmistaa konenäköön perustuvia laadunvalvontajärjestelmiä erityisesti autoteollisuuden käyttöön. Yhtiön käyttämä teknologia kehitettiin alun perin VTT:llä Otaniemessä 1977. Mapvision yhtiönä perustettiin 1987. (About us 2020.)

Mittauslaitteisto (ks. kuvio 1) on ainoa maailmassa, joka kykenee mittamaan reaaliajassa 100 % kaikista osista suoraan tuotantolinjalla. Tämä mahdollistaa tarkan tuotantoprosessin valvomisen, ja näin parantaa ymmärrystä tuotannon toiminnasta sekä vähentää hävikin määrää ja virheitä. Mittaus perustuu fotogrammetriaan. Mittausprosessi on täysin automaattinen ja kestää noin 30 sekuntia. Laite kykenee mittaamaan kappaleen yksityiskohtien sijainnit, kulmat, etäisyydet, pintojen muodot ja tarkistaa onko kaikki vaaditut ominaisuudet olemassa. Mittauksen tarkkuus on  $\pm 0.02$  mm. Laitteisto myydään tilaajalle räätälöitynä pakettina. (Mapvision Quality Gate 2020.)



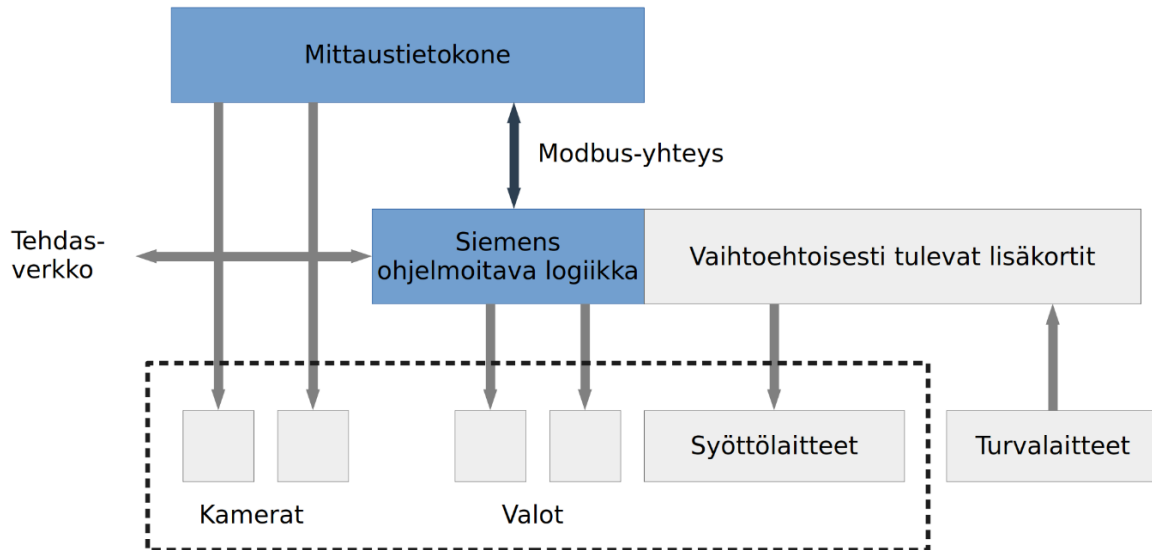
Kuvio 1. Mapvisionin mittauslaitteisto

## 2 Lähtökohdat

Työ koostuu kahdesta osittain toisiinsa liittyvästä osiosta. Tässä kappaleessa kerrotaan niihin liittyvistä pohjatiedoista ja kehitystarpeista.

### 2.1 Vaihtoehdot sisäiselle Modbus-yhteydelle

Mittauslaitteistosta on hiljattain kehitetty uudistettu versio, jossa vanha Modbusiin liitetty I/O-väyläsovitin on korvattu Siemensin ET 200SP-sarjan ohjelmoitavalla logiikalla (ks. kuvio 2). Uusi versio on vielä testausvaiheessa. Riippuen versiosta, logiikka toimii myöskin vain väyläsovittimena, tai se voi tämän lisäksi sisältää logiikkaohjelman, joka hoitaa mitattavan kappaleen syöttämiseen ja turvallisuuteen liittyviä asioita. Mittaustietokone on yhdistetty Siemensin logiikkaan täysin samalla Modbus-yhteydellä, kuin mitä vanhakin ratkaisu käytti. Modbus-yhteydellä ohjataan kappaleiden kuvauksessa käytettäviä valoja, merkkilamppuja sekä ulkoista rajapintaa, jolla mittauslaite liitetään osaksi tehtaan tuotantolinjaa. Ulkoisesta rajapinnasta kerrotaan lisää seuraavassa kappaleessa.

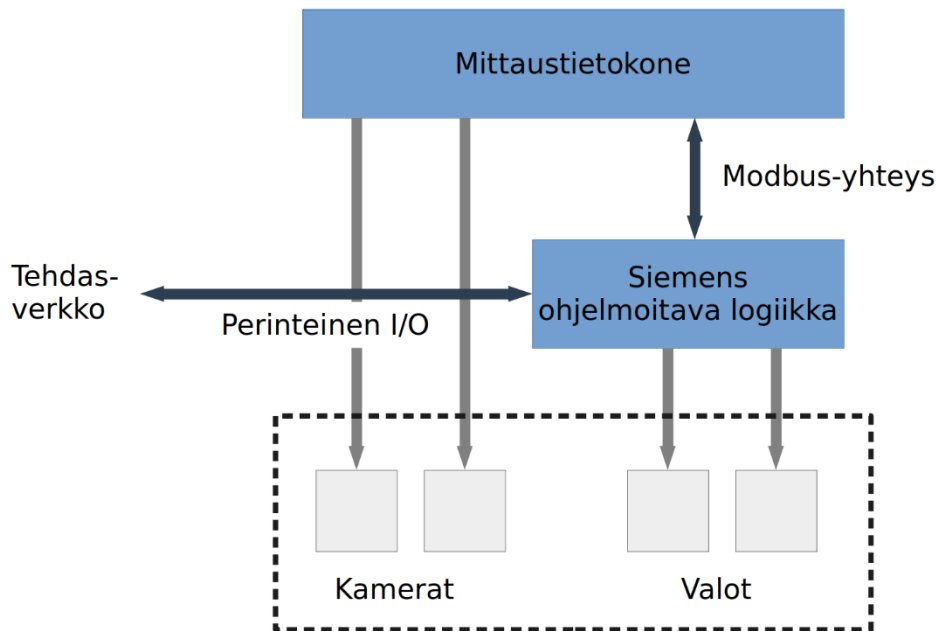


Kuvio 2. Sisäinen Modbus-yhteys mittauslaitteistossa

Uuden version kanssa on kaksi ongelmaa. Modbus-yhteys ei toimi luotettavasti Siemensin ohjelmoitavan logiikan kanssa, ja ohjelmoitava logiikka on merkittävästi vanhaa I/O-moduulia kalliimpi. Tämän lisäksi mittauslaitteiston joissakin versioissa on Modbusin lisäksi myös Profinet-yhteyksiä, joita sama ohjelmoitava logiikka hoitaa. Mikäli Modbusin voisi vaihtaa Profinetiksi, saataisiin kaikki mittauslaitteen osat samaan verkkoon, mikä yksinkertaistaisi järjestelmää. Näistä syistä lähdettiin etsimään vaihtoehtoisia ratkaisuja. Tahtotilana oli jatkaa nimenomaan Siemensin tuotteiden käyttämistä.

## 2.2 Ulkoisen rajapinnan toteutus kenttäväylänä

Mittauslaitteisto liitetään osaksi tehtaan tuotantolinjaa perinteisellä I/O:lla (ks. kuvio 3). Käytännössä tämä on iso liitin, jossa on koskettimet käyttöjännitteelle, 16 bitille ulospäin ja 8 bitille sisäänpäin. Ulospäin lähtevät bitit sisältävät tietoja kuten: ”mittaus valmis”, ”kappale läpäisi tarkistuksen”, ”kappale ei läpäissyt tarkistusta”. Sisään tulevat bitit puolestaan käsittävät tietoja kuten: ”aloita mittaus” ja ”osan valinta” (laite pystyy mittaamaan useanlaisia osia).



Kuvio 3. Ulkoinen rajapinta tehdasverkkoon on toteutettu perinteisellä I/O:lla

Tämä on toiminut hyvin, mutta tarvetta nykyaikaisemmalle ratkaisulle olisi, sillä osalla asiakkaista ei lähtökohtaisesti ole käytettävissä perinteistä I/O:ta. Liittimessä ei myöskään ole käytännössä enää yhtään tilaa lisätoiminnoille, joille saattaa tulevaisuudessa ilmentua tarvetta.

Tämän takia liittimen rinnalle, ja ehkä myöhemmin myös korvaamaan, halutaan tuoda tuki Ethernet-pohjaiselle ohjaukselle. Rajapinnan tulee mahdollistaa kaikki sama kuin perinteinen I/O-rajapinta. Tavoitteena on selvittää, miten tämä toteutetaan ja mitä protokollaa tai protokollia rajapinta tukee, ja tehdä toimiva toteutus tästä rajapinnasta. Protokollan tulee olla yleisesti asiakkaiden käyttämä, jotta laitteen liittäminen osaksi tehtaan järjestelmiä olisi mahdollisimman helppoa.

### 2.3 ET 200SP analogiakorttien lisääminen ilman lisätöitä

Työhön oli alun perin tarkoitus sisällyttää myös kolmas osio, jossa olisi tutkittu onko mahdollista lisätä ja poistaa I/O-kortteja Siemensin ET-200 SP järjestelmään ilman tarvetta konfiguraatiomuutoksille, tai ainakin mahdollisimman pienillä muutoksilla. Tästä osiosta kuitenkin luovuttiin aikatauluhaasteiden takia. Osion pois jääminen ei vaikuta kahden edellä mainitun osion tuloksiin.

## 3 Tutkimusasetelma

Tässä kappaleessa kerrotaan työn tutkimusotteesta ja tutkimuksen toteuttamistavoista.

### 3.1 Tutkimuskysymykset

- Osio 1: Vaihtoehdot sisäiselle Modbus-yhteydelle
  - Minkä takia Modbus-yhteys ei toimi kunnolla Siemensin ohjelmoitavan logiikan kanssa?
  - Mitä vaihtoehtoja on nykyiselle Modbus-yhteydelle siirtämään tietoa mittaustietokoneelta Siemensin I/O-väyläsovittimille tai ohjelmoitavalle logiikalle?
- Osio 2: Ulkoisen rajapinnan toteutus kenttäväylänä
  - Mikä tai mitä väyläprotokollia voidaan tukea?
  - Miten väyläohjaus toteutetaan näillä protokollilla?

### 3.2 Tutkimusote

Tutkimus on toteutettu käyttäen kvalitatiivista eli laadullista tutkimusmenetelmää. Tällä menetelmällä pyritään luomaan kattava ymmärrys ilmiöistä tai asiasta, ja muodostamaan tällä ymmärryksellä monipuolisia ja syvällisesti harkittuja tuloksia. Tilastolliset menetelmät tai muut määrällisen tutkimuksen keinot eivät tule kyseeseen työn luonteen takia. Työssä syvennytään rajattuun määrään erilaisia ratkaisuvaihtoehtoja, ja vertaillaan niiden soveltuvuutta toimeksiantajan tarpeisiin lukuisin, mutta vaihtelevin perustein. Toisin sanoen, vaihtoehtoja ei voi vertailla yksiselitteisesti.

### 3.3 Aineisto

Työn aineisto koostuu lähes yksinomaan eri teollisuusprotokollien spesifikaatioista ja muusta protokollan omistavan tahon virallisesta materiaalista. Virallinen aineisto on kerätty internetistä, sillä internetissä oleva materiaali on todennäköisemmin ajantasaista, eikä isoa osaa materiaalista edes julkaista fyysisenä kopiona.

Toimeksiantajan kanssa käydään aktiivista tiedonvaihtoa työn edistymisestä, sekä keskustellaan mihin suuntaan työssä halutaan edetä tehtyjen löydösten perusteella. Kommunikointi pidetään vuorovaikutteisena, sillä varsinkin työn alkuvaiheessa on paljon selvitettäviä asioita, ja joista pitää saada toimeksiantajan päätös, ennen kuin tutkimusta voidaan käytännöllisesti jatkaa.

## 4 Ethernet-pohjaiset teollisuusprotokollat

Työssä keskeisenä asiana ovat teollisuudessa käytetyt Ethernet-pohjaiset tiedonsiirto-protokollat. Tässä kappaleessa kerrotaan perustietoja neljästä eri protokollasta, joihin tämän opinnäytetyön yhteydessä perehdyttiin tarkemmin.

Ethernet on IEEE:n standardoima protokolla (virallisesti standardi 802.3). Se määrittelee OSI-mallin fyysisen kerroksen sekä siirtokerroksen. Näin ollen Ethernet-standardi määrittelee vain tiedon vaihdon kahden laitteen välisellä suoralla yhteydellä. Tiedonsiirtoon laajemmassa verkossa tarvitaan tämän lisäksi muita protokollia. Tavallisin tähän käytetty protokollapino toimisto- ja kotiverkoissa on TCP/IP tai UDP/IP. (Spurgeon 2000, luku 1.)

TCP-protokolla toteuttaa OSI-mallin kuljetuskerroksen. Sen tärkein tehtävä on varmistaa, että tieto tulee perille, ja tulee perille kokonaisuutena, oikeassa järjestyksessä ja virheettömänä. Tämä toteutetaan lähettäjän ja vastaanottajan välillä edestakaisin lähetettävillä varmistusviesteillä. TCP:tä kutsutaan myös yhteydelliseksi protokollaksi, koska jokaisen palvelimen ja asiakkaan välillä on oltava uniikki yhteys, jonka sisällä tietoa siirretään. Vaikka protokolla takaa hyvän luotettavuuden tiedonsiirrolle, se on toisaalta myös hidas kaikkine varmistuksineen. (Jiju 2018.)

UDP on puolestaan suunniteltu nopeus edellä. UDP on TCP:tä huomattavasti yksinkertaisempi ja nopeampi protokolla. Yksinkertaisuudessaan vastaanottaja lähettää pyynnön palvelimelle, jonka jälkeen palvelin alkaa lähettämään vastauksena paketteja vastaanottajalle. Mitään tarkistusta ei tehdä sen suhteen, saapuvatko paketit ikinä perille. Näin ollen UDP on yhteydetön. Tästä seuraa se, että UDP:ta voidaan käyttää myöskin ryhmälähetyksiin, mikä voi vähentää verkon kuormaa merkittävästi (Introduction to IP multicast 2006, 7). (Jiju 2018.)

Tässä työssä käsitellyistä teollisuusprotokollista kaikki hyödyntävät edellä kuvattuja tekniikoita ainakin jollakin tasolla. Kaikkien protokollien pohjalla on Ethernet, mutta protokollapinon ylempät osat voivat vaihdella protokollan sisälläkin käyttötapauksen mukaan. Esimerkiksi EtherNet/IP hyödyntää sekä TCP- että UDP-protokollia ja Profinetin syklinen tiedonsiirto toimii suoraan Ethernetin päällä.

## 4.1 Modbus TCP/IP

Modbus kehitettiin alun perin 1979 Modiconin (nykyään Schneider Electric) toimesta. Vuonna 1999 vanhan Modbus-standardin pohjalta kehitettiin uudempi Modbus TCP/IP. Modbusista on kehittynyt kaikkein yleisimmin tuettu protokolla teollisuudessa, pitkälti avoimuutensa ja yksinkertaisuutensa johdosta, mikä on myös sen vahvuus tämän opinnäytetyön kannalta – käytännössä kaikilta toimeksiantajan asiakkailta löytyy tuki Modbusille omissa järjestelmissään. (Modbus FAQ 2021.)

Modbusin kehitysvastuu siirtyi Schneiderilta voittoa tavoittelemattomalle Modbus Organization -järjestölle vuonna 2004. Järjestö koostuu lukuisista protokollan käyttäjistä ja laitteiden toimittajista. Modbus TCP/IP on täysin vapaa lisenssimaksuista ja protokollan spesifikaatiot ovat ilmaiseksi ladattavissa internetissä. (Modbus FAQ 2021.)

Modbus TCP/IP on pitkälti samanlainen protokolla kuin vanhempi Modbus, mutta osoite- ja tarkistussummakentät on korvattu uusilla otsikkokentillä ja data on paketoitu TCP- ja IP-otsikkokenttiin. Protokolla toimii OSI-mallin kerroksessa 7 (sovelluskerros) ja perustuu asiakas-palvelin-mallille, eli mallille, jossa asiakas tekee pyynnön ja palvelin vastaa pyyntöön yhdellä viestillä. TCP/IP on sama protokolla, jolla suurin osa tietokoneista liittyy internettiin. Näin ollen Modbusia pystyy käyttämään käytännössä millä tahansa tavallisella tietokoneella, ja jopa internetin yli. Protokolla ei lähtökohtaisesti takaa tiedonsiirron deterministisyyttä, mutta korkea luotettavuus voidaan saavuttaa estämällä ylimääräisten laitteiden pääsy Modbus-verkkoon. (Modbus FAQ 2021; Introduction to Modbus TCP/IP 2005, 4 - 7.)

Modbus TCP/IP sanoma koostuu pääpiirteittäin 7 tavun mittaisesta otsikkotiedoista, funktiokoodista sekä varsinaisesta datasta. Funktiokoodi (engl. function code) määrittää mitä halutaan tehdä (esim. lukea tai kirjoittaa rekistereitä) ja määrittää mitä tietoja sanomaan tulee sisällyttää, jotta haluttu toiminto voidaan suorittaa. Modbus tietomalliin kuuluu neljä eri tietotyyppiä, joita voidaan käsitellä. Nämä ovat binääriset tulot ja lähdöt, sekä 16 bittiset tulo- ja lähtörekisterit. (Introduction to Modbus TCP/IP 2005, 4 - 14.)

## 4.2 Profinet

Profinet on kenttäväyläprotokolla, joka pohjautuu vanhempaan Profibus-väylään. Merkittävin ero Profinetissä on, että se käyttää 100MB/s Ethernet-yhteyttä alustanaan, toisin kuin Profibus, joka perustuu täysin sitä varten kehitettyyn väyläkaapeliin ja -protokollaan. Näin ollen Profinet on täysin yhteensopiva muiden Ethernet-verkkojen kanssa, ja sitä on mahdollista käyttää samassa verkossa esimerkiksi Modbus TCP/IP- tai tavallisten toimistotietokoneiden kanssa. Profinetin hyötyihin kuuluu muun muassa universaali yhteensopivuus laitteiden ja järjestelmien välillä, laitteiden diagnostiikkatietojen välitys, turvallisuuden kannalta kriittinen tiedonsiirto (esim. hätä-seis) sekä reaaliaikainen ohjaus jopa alle 1  $\mu$ s tarkkuudella. (PROFINET System Description 2014.)

Profinetin toiminta perustuu tuottaja-kuluttaja-malliin, jossa on yleensä yksi, tai joskus myös useampi IO-kontrolleri, ja yksi tai useampi IO-laite sekä mahdollisesti IO-supervisor eli laite (yleensä PC-tietokone) vianselvitystä tai ohjelmointia varten. IO-kontrolleri pitää huolen koko järjestelmän toiminnasta lukemalla ja kirjoittamalla dataa IO-laitteisiin. Profinet sisältää useampia eri tapoja siirtää tietoa eri laitteiden välillä, näitä eri tekniikoita varten on laadittu luokat (engl. Conformance Classes) A-C. Luokkien perusteella näkee suoraan mitä Profinetin ominaisuuksia tietyn luokan vaatimukset täyttävä laite tukee, ja mitä asioita järjestelmän suunnittelussa tulee huomioida. Luokka CC-A sisältää perustoiminnallisuudet, ja luokat CC-B ja CC-C lisäävät tämän päälle muita ominaisuuksia. (PROFINET System Description 2014, 2.)

Profinet-standardi määrittelee myös, kuinka järjestelmä konfiguroidaan. Konfiguroinnissa määritellään kaikki verkossa olevat laitteet, niiden osoitteet ja asetukset. Konfiguraatioon kuuluu GSD-tiedostot, jotka ovat laitevalmistajien tarjoamia kuvauksia käytetyistä IO-laitteista. GSD-tiedoston avulla järjestelmän kehittäjä sekä IO-kontrolleri tietävät, missä muodossa IO-laitteet lähettävät ja vastaanottavat tietoja, mahdollistaen standardoidun tietomallin eri laitteiden välillä. Tämä on iso ero Modbus-protokollaan, joka ainoastaan määrittää miten tietoa siirretään, mutta ei sitä mitä tieto tarkoittaa. (PROFINET System Description 2014, 4.)

### 4.2.1 Profinetin ominaisuuksia

Tässä kappaleessa on kerrottu tarkemmin Profinetin eri ominaisuuksista.



Syklinen tiedonsiirto on ns. normaali tapa siirtää tietoa IO-kontrollerin ja IO-laitteiden välillä. Niimensä mukaisesti tiedonsiirto on syklistä, eli se suoritetaan säännöllisin väliajoin. Syklin pituus voidaan määrittää 250  $\mu$ s:sta 512 ms:iin. Eri IO-laitteille voidaan käyttää eri pituisia syklejä. Jokaisen syklin mukana siirtyy varsinaisen prosessidatan mukana myös tietoa datan eheydestä, diagnostiikkatietoa sekä syklilaskuri. Näiden perusteella kontrolleri voi esimerkiksi antaa hälytyksen, mikäli data ei tule ajoissa perille tai on viallista. Syklinen tiedonsiirto tapahtuu suoraan Ethernet-tasolla, eikä TCP/IP protokollia käytetä lainkaan. Syklinen tiedonsiirto on Profinetin perusominaisuuksia kuuluen CC-A luokkaan. (PROFINET System Description, Technology and Application 2014, 6 - 7.)

Syklisen tiedon lisäksi voidaan käyttää asyklistä tiedonsiirtoa. Tätä mekanismia käytetään satunnaisesti tapahtuvaan tiedonsiirtoon. Siirrettävän tiedon tyyppi on laitevalmistajan vapaasti määriteltävissä, mutta standardissa on määriteltä myös pakollisia määrämuotoisia tietoja, kuten diagnostiikkatiedot sekä tunniste ja huoltotiedot. Asyklinen data käyttää hyväkseen TCP/IP-tekniikkaa. Myös asyklinen tiedonsiirto on CC-A luokan ominaisuus. (PROFINET System Description, Technology and Application 2014, 7.)

Hälytyksiä voidaan antaa laitteen sisäisistä, anturin tms. vioista. Esimerkiksi anturin johdon katkeaminen voi laukaista hälytyksen. Myös sallittujen prosessiarvojen ylitys voidaan konfiguroida aiheuttamaan hälytys. Hälytyksiä on eri tasoisia vian vakavuuden ja korjauksen kiireellisyyden mukaan. Vikahälytyksiä voidaan antaa ennakoivasti ennen kuin laite on vielä lakannut toimimasta, ja näin parantaa järjestelmän käytettävyyttä. (PROFINET System Description, Technology and Application 2014, 7.)

Naapurilaitteiden tunnistus on CC-B ja CC-C luokkien ominaisuus, jonka avulla IO-laitteet voidaan löytää verkon rakenteen perusteella. Sen toiminta perustuu SNMP-protokollaan, jonka avulla pystytään lukemaan laitteiden tietoja Ethernet-portin (liittimen) perusteella, ja näin selvittämään mikä laite on kytkettynä mihinkin. Tämä mahdollistaa esimerkiksi verkon topologian muuttumisen tunnistamisen, sekä automaattisen laitteen nimeämisen laitetta vaihdettaessa. (PROFINET System Description, Technology and Application 2014, 8.)

Synkroninen reaaliaikainen tiedonsiirto (Profinet IRT) on luokan CC-C ominaisuus, joka mahdollistaa erittäin tarkan ja nopean tiedonsiirron. Sykliaika voi olla jopa 31,25  $\mu$ s alle 1  $\mu$ s tarkkuudella.

Tiedonsiirto perustuu reaaliaikaiselle tiedolle varattuun kaistaan. Kaikki samassa reaaliaikaisessa linjassa olevat laitteet tahdistetaan samaan kellotaajuuteen, jolloin tietty aika yhdestä kellosyklistä voidaan varata reaaliaikaisen tiedon siirtoon ja loppu aika kaikelle muulle liikenteelle. Yksi IO-kontrolleri voi olla yhdistettynä useampaan eri reaaliaikaiseen linjaan, mutta reaaliaikaisia ja ei-reaaliaikaisia laitteita ei voida kytkeä sekaisin, paitsi joissakin erikoistapauksissa. (PROFINET System Description, Technology and Application 2014, 9 - 11.)

Näiden lisäksi Profinet-standardi määrittelee muitakin ominaisuuksia ja toimintoja vaativampiin käyttötarkoituksiin. Nämä ovat niin sanotusti vapaaehtoisia ominaisuuksia, eli niiden toteuttaminen laitevalmistajalta ei ole välttämätöntä. Esimerkkeinä näistä on nopea uudelleenkäynnistys, käytössä olevan laitteen konfigurointi ja redundanttiset yhteydet. (PROFINET System Description, Technology and Application 2014, 11 - 14.)

### **4.3 OPC UA**

OPC UA (Open Platform Communications Unified Architecture) on vanhemman OPC Classicin pohjalta kehitetty standardi. OPC UA on monipuolinen standardi monine ominaisuuksineen. Se on alustariippumaton, ja se on suunniteltu soveltuvaksi moniin eri tarkoituksiin, niin kaupallisten tilastotietojen tuottamiseen järjestelmästä kuin yksittäisen kenttälaitteen ohjaamiseen. (Unified Architecture 2021.)

OPC UA:ta edeltänyt OPC Classic kehitettiin alun perin abstraktoimaan ja standardoimaan matalamman tason laitteiden viestintää HMI- ja SCADA-järjestelmien kanssa, toisin sanoen toimimaan standardoituna rajapintana ohjelmoitavan logiikan ja järjestelmänhallintatietokoneen välillä. OPC UA on OPC Foundationin omistama. Sen suunnittelu aloitettiin vuonna 2003 ja se julkaistiin 2008. Suunnittelun kulmakivinä oli toiminnallinen vastaavuus OPC Classicin kanssa, alustariippumattomuus, turvallisuus, laajennettavuus sekä tiedon semanttisuus. OPC UA:sta on tehty IEC-standardi (Unified Architecture 2021; Interoperability for Industrie 4.0 and the Internet of Things 2019, 18.)

Osoitteistus perustuu oliopohjaiselle hierarkiselle tietorakenteelle. Oliot siis muodostavat samankaltaisen rakenteen kuin kotitietokoneiden kansiot. Standardi pitää sisällään perustietotyyppejä,

joita pystyy käyttämään olioiden sisässä. Perustietotyypit voidaan jakaa kolmeen ryhmään: muuttajat (engl. variables), menetit (engl. methods) ja tapahtumat (engl. events). (Interoperability for Industrie 4.0 and the Internet of Things 2019, 22.)

Tietorakenteita voi luoda itse tai käyttää valmiita yleiseen käyttöön tarkoitettuja tietorakenteita (Companion specifications), joita on määritelty tiettyjen teollisuudenalojen ja laitteiden tarpeisiin. Esimerkkeinä näistä ovat CNC-koneet, painon mittaus, konenäkö ja ammattikeittiövälineet. Etuna tässä lähestymistavassa on se, että sillä voidaan standardoida eri valmistajien laitteiden rajapinnat, ja näin ollen ohjauslogiikan näkökulmasta ei ole väliä kenen valmistajan laite on käytössä. Idea on hieman vastaava kuin Profinetin GSD-määritelmätiedostot. (Interoperability for Industrie 4.0 and the Internet of Things 2019, 22 – 25; UA Companion Specifications 2021.)

Standardi on suunniteltu laajennettavaksi erilaisilla tavoilla, kuitenkin rikkomatta taaksepäin yhteensopivuutta. Standardissa on jätetty mahdollisuus uusille tavoille siirtää tietoa. Esimerkiksi jos keksitään uusia parempia tapoja suojata tietoja, voidaan uudet tekniikat ottaa OPC:ssä käyttöön rikkomatta jo olemassa olevia järjestelmiä. Tietoa voidaan myös lähettää erilaisissa muodoissa, kuten JSON tai OPC-binäärimuoto. Niin ikään standardin tietorakenteet (=oliot) on suunniteltu laajennettaviksi. Myös kaikki OPC Classicissa olleet toiminnot toimivat myös OPC UA:ssa. Järjestelmässä on pidetty erillään tapa siirtää tietoa, ja tiedon semantiikka eli merkitys. (Unified Architecture 2021.)

Tiedonsiirtoon on kaksi eri strategiaa. Asiakas-palvelin-malli toimii samaan tapaan kuin esimerkiksi Modbusissa, eli asiakas tekee pyynnön ja palvelin reagoi pyyntöön ja lähettää vastauksen. Toinen strategia on niin sanottu julkaisija-tilaaja-malli, jossa julkaisija lähettää viestejä väliohjelmistolle, josta vastaavasti tilaaja voi tilata haluamaansa tietoa. Tämä malli on suunniteltu erityisesti järjestelmiä varten, joissa tietoa halutaan siirtää monelta julkaisijalta monelle tilaajalle. Väliohjelmisto ei tosin ole pakollinen vaan julkaisija voi myös lähettää suoraan tietoa tilaajalle. (Interoperability for Industrie 4.0 and the Internet of Things 2019, 22 - 23.)

Tiedon salaus on yksi osa-alueista, johon OPC UA:ssa on panostettu. Tietoturva on jaettu kolmeen osa-alueeseen, jotka ovat käyttäjän, sovelluksen sekä tiedonsiirron turvallisuus. Kaksi ensimmäistä käytännössä autentikoivat käyttäjän, joka yrittää muodostaa yhteyttä ja auktorisoivat tälle sallitut

palvelut. Tiedonsiirron turvallisuus pitää sisällään tiedon eheyden tarkistuksen sekä tiedon salauksen. (Interoperability for Industrie 4.0 and the Internet of Things 2019, 30 - 31.)

Muita OPC UA:n ominaisuuksia ovat historiatietojen selaaminen, palvelimien automaattinen haku verkosta sekä tietorakenteiden selaaminen. (Unified Architecture 2021.)

#### **4.4 EtherNet/IP**

EtherNet/IP-protokolla (Ethernet Industrial Protocol) julkaistiin 2001 ja sen omistaa ODVA. Protokolla hyödyntää ODVAN toista, CIP-protokollaa (Common Industrial Protocol) OSI-mallin ylemmissä kerroksissaan. CIP on itsessään alustariippumaton ja EtherNet/IP on sen toteutus Ethernet-protokollan päälle. CIP on käytössä myös ODVAN muissa tuotteissa kuten DeviceNetissä ja CompoNetissä. (EtherNet/IP Technology Overview Series 2016, 2; Schiffer 2016, 7.)

CIP on oliopohjainen protokolla, hieman samaan tapaan kuin OPC UA. Tämä tarkoittaa, että jokaisessa CIP-laitteessa olevat tiedot on ryhmitelty olioihin. CIP määrittelee pakollisia olioita, jotka kaikkien laitteiden tulee sisältää, mutta tämän lisäksi valmistajat voivat myös määritellä omia olioitaan. CIP-olio voi pitää sisällään attribuutteja (esim. binäärinen tieto), palveluita (komento) sekä käytösmalleja (reaktiot tapahtumiin). Tämä muodostaa osoiterakenteen, jossa mikä tahansa tieto CIP-verkon sisällä on saatavilla, kun tiedetään laitteen, olion ja attribuutin osoitteet. Oliot luodaan luokista, jotka määrittelevät olion ominaisuudet, kuten sen sisältämät attribuutit. Yhdestä luokasta voidaan luoda useita samankaltaisia olioita. Attribuuteille ja olioille on määritetty palveluita, joiden perusteella vastaanotettuun tietoon reagoidaan, esimerkiksi tiedon luku tai kirjoitus. Palvelut ovat vastaava idea kuin Modbusin funktiokoodit. (Schiffer 2016, 12 – 14; EtherNet/IP Technology Overview Series 2016, 6.)

Tietoa voidaan siirtää kahdella tavalla. Aikakriittisiin ohjauksiin käytetään niin sanottua reaaliaikaista tiedonsiirtoa, joka tapahtuu UDP/IP-protokollan avulla. Tällöin käytetään tuottaja-kuluttajamallia, jossa tuottaja lähettää paketteja ja kuluttaja puolestaan ottaa niitä vastaan. Tieto siis liikkuu vain yhteen suuntaan. Kun taas halutaan varmistaa, että tieto menee varmasti perille, mutta se ei ole aikakriittistä, käytetään ns. täsmällistä TCP/IP-protokollaan perustuvaa siirtoa. Tällöin on puolestaan käytössä asiakas-palvelin-malli, jossa toinen asiakas tekee pyynnön ja palvelin lähettää siihen vastauksen. Reaaliaikaisessa tiedonsiirrossa on UDP:n tuoman nopeuden lisäksi etuna se,

että se tukee tiedon ryhmälähetystä. Tämän lisäksi EtherNet/IP määrittelee yhdistetyn ja yhdistämättömän tiedonsiirron. Yhdistämätön tiedonsiirto on tarkoitettu harvakseltaan tapahtuvalle kiireettömälle tiedonsiirrolle, esimerkiksi yhteyksien konfigurointiin. Yhdistetty tila soveltuu puolestaan säännöllisesti toistuvalla aikakriittiselle tai kiireettömälle tiedonsiirrolle. Tyypillisesti siis ensin avataan yhteys, jonka jälkeen prosessitietoja aletaan päivittämään yhdistetyn yhteyden ylitse (Schiffer 2016, 15). Nämä termit ovat EtherNet/IP:n omia ominaisuuksia, eikä niitä tule sekoittaa TCP- ja UDP-protokollien yhteyksiin. (EtherNet/IP Technology Overview Series 2016, 4-5.)

CIP-protokollassa on myös käsite laiteprofiilit (Device Profiles). Ne määrittelevät standardoituja laitetyppejä eli tietynlaisia CIP-olioita, jotka profiilia käyttävän laitteen tulee toteuttaa. Toisin sanoen niiden avulla voidaan luoda standardoituja rajapintoja eri valmistajien samaan tarkoitukseen tehdyille laitteille. Idea on hyvin saman kaltainen kuin OPC UA:n ”companion specifications”. (EtherNet/IP Technology Overview Series 2016, 6.)

Laitteet jaetaan kolmeen luokkaan tiedonsiirto-ominaisuuksien ja käyttötavan mukaan. Ensimmäinen luokka on ”Messaging Class” joka ei kykene aikakriittiseen tiedonsiirtoon, mutta voi lähettää ja vastaanottaa ei-kiireellistä tietoa. Esimerkiksi ohjelmointi-PC tai HMI-paneeli kuuluvat tähän ryhmään. Toinen luokka on ”Adapter Class” joka pystyy vastaanottamaan ja lähettämään aikakriittistä sekä kiireetöntä dataa, mutta vain jos toinen laite on ensin avannut yhteyden. Esimerkiksi I/O-kortit ja robotit voivat kuulua tähän luokkaan. Kolmas ryhmä on ”Scanner Class” joka toimii yhteyden avaajina ”Adapter Class” laitteille. Esimerkiksi ohjelmoita logiikka kuuluu yleensä tähän luokkaan. (Ethernet/IP Technology Overview Series 2016, 5.)

EtherNet/IP-verkko voidaan toteuttaa tähtitopologiana tai mikäli kenttälaitteissa itsessään on sisäänrakennettu kytkin, väylätopologiana. Protokolla tukee myös rengastopologiaa, kunhan verkkoon kytketyt laitteet tukevat DLR-teknologiaa (Device Level Ring). Fyysisenä siirtotienä voidaan käyttää mitä tahansa Ethernetin-tukemaa mediaa, kuten kuparijohto, lasikuitu tai langaton verkko. (EtherNet/IP Technology Overview Series 2016, 3.)

CIP-protokollan tietoturvalle ei ole vielä valmista standardia, mutta EtherNet/IP:lle on määritetty standardi (Ethernet/IP Confidentiality Profile) joka antaa osittaista suojaa täysin suojaamattomaan

verkkoon nähden. Se sisältää määrittelyjä laitteen tunnistamiseen ja tiedon eheyteen liittyen. Tietoliikenteen salausta EtherNet/IP ei tue. ODVA luottaa tietoturvan suhteen verkon eristämiseen luvattomalta käytöltä sekä fyysisesti että sähköisesti. (SecuringEtherNet/IP™ Networks 2011; Overview of CIP security, 7 - 8.)

ODVA on kehittämässä yhteistyössä OPC-säätiön kanssa kumppanispesifikaatiota, joka määrittelee kuinka tiedot saadaan siirrettyä CIP-protokollasta pilveen OPC UA-palvelimelle. (Market drivers, OPC UA 2021.)

## 5 Vaihtoehdot sisäiselle Modbus-yhteydelle

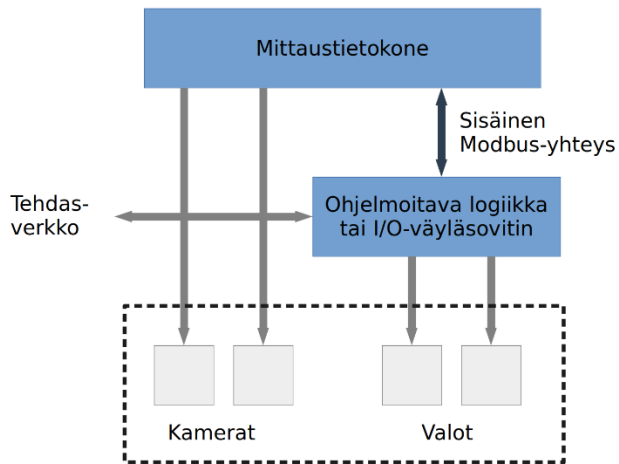
Tässä kappaleessa kuvataan tarkemmin opinnäytetyön ensimmäisen osion Modbus-yhteyden korvaamiseen liittyviä teknisiä vaatimuksia sekä selvitystyön tuloksia.

### 5.1 Vaatimukset

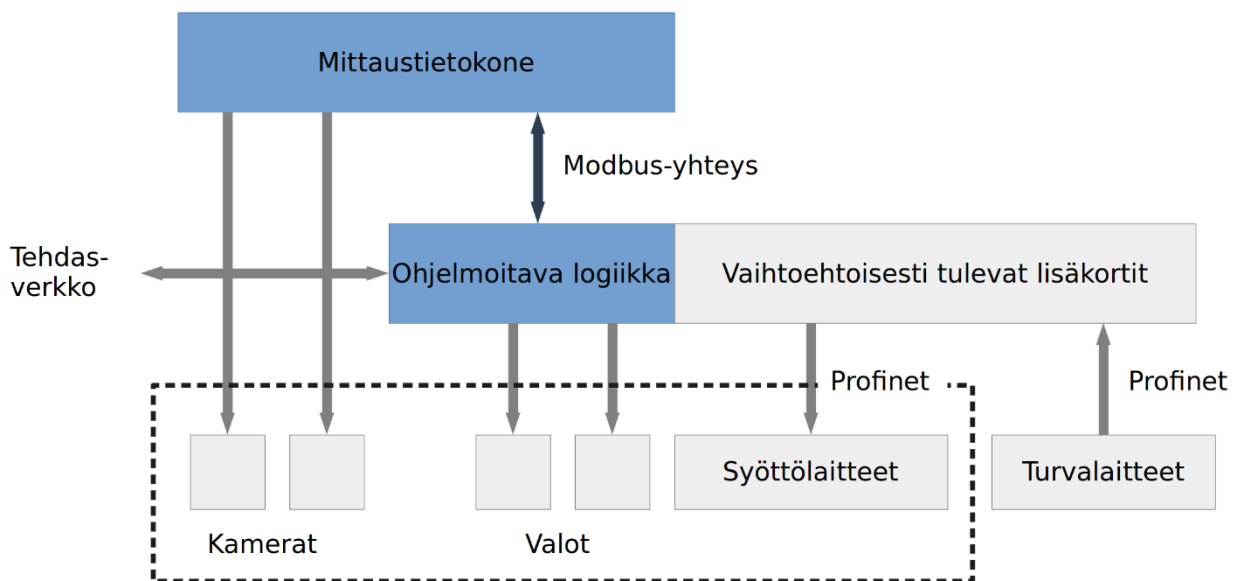
Väyläsovittimeen tulee voida liittää 48 analogista lähtöä, 12 digitaalista tuloa sekä 14 digitaalista lähtöä. ET 200SP-sarjan korteilla tämä tarkoittaa 15 korttia. Opinnäytetyön toisessa osiossa kuitenkin selvitetään ulkoisen rajapinnan perinteisen I/O:n korvaamista Ethernet-pohjaisella väylällä. Tämän myötä osa digitaalisista korteista voi jäädä tulevaisuudessa tarpeettomaksi. Tällöin tarvetta olisi 5 digitaaliselle lähdölle ja 2 digitaaliselle tulolle. Tässä tapauksessa tarvittaisiin kortteja 14.

#### 5.1.1 Mittauslaitteiston eri versiot

Mittalaitteistosta on olemassa kaksi versiota. Ensimmäisessä versiossa (ks. kuvio 4) toimitetaan ainoastaan mittalaitteisto, joka pitää sisällään ohjaustietokoneen, kamerat ja valot asennettuna mittauskammioon. Toisessa versiossa (ks. kuvio 5) on tämän lisäksi mukana järjestelmä kappaleen syöttämiselle ja poistamiselle mittauskammioista, sekä turvalaitteita. Jälkimmäiseen järjestelmään tulee mukaan useampia Profinetin yli ohjattavia laitteita, sekä lisää IO-kortteja ja ohjelmoitava logiikka hallitsemaan kappaleen siirtelyä. Näin ollen opinnäytetyössä huomio painottuu ensimmäiseen versioon, jossa ohjelmoitavaa logiikkaa ei välttämättä tarvittaisi.



Kuvio 4. Sisäinen Modbus-yhteys versiossa, jossa ei välttämättä tarvita ohjelmoitavaa logiikkaa



Kuvio 5. Sisäinen Modbus-yhteys versiossa, jossa on mukana kappaleen syöttämiseen tarvittava laitteisto

Koska ohjelmoitavalla logiikalla ei ole oikeastaan mitään muuta tehtävää, kuin muuntaa Modbus-viestit Siemensin muiden komponenttien ymmärtämään muotoon, ja koska ohjelmoitava logiikka on tähän tehtävään varsin kallis ja ylitehoinen komponentti (muistia: 750 kB, käsikyn suoritus aika maksimissaan: 115 ns) (6ES7510-1SJ01-0AB0 2015), herää luonnollisesti kysymys, että tarvitaanko sitä ollenkaan. Koska ET 200SP-sarja käyttää Profinettiä natiivina kommunikaatioprotokollanaan

(SIMATIC ET 200SP - The compact IO system for control cabinets), ja koska Profinet on avoin standardi, ajateltiin että logiikka olisi todennäköisesti mahdollista vaihtaa halvempaan saman sarjan väyläsovittimeen.

### **5.1.2 Olemassaolevat Profinet-laitteet**

Mittauslaitteistossa on myös käytössä Profinet-yhteyksiä jo vanhastaan. Jos Modbus-yhteys saataisiin muutettua Profinet-yhteydeksi, saataisiin kaikki laitteet samaan verkkoon, mikä yksinkertaistaisi laitteiston rakennetta, kun siitä saataisiin yksi muista poikkeava yhteys pois. Tämä voisi myös avata uusia mahdollisuuksia, sillä mittaustietokone ja jotkin Profinet-laitteet voisivat mahdollisesti viestiä suoraan keskenään. Nämä eivät ole välttämättömiä vaatimuksia, mutta toisivat toteutessaan lisää liikkumavaraa ja mahdollisuuksia laitteiston kehitykselle. Profinet-laitteita voi nykyisessä kokoonpanossa tulla yhteensä korkeintaan 9 kappaletta.

### **5.1.3 Vasteaika**

Yhteydellä ohjataan kappaleiden kuvauksessa käytettäviä valoja, ja kuvia otetaan nopealla tahdilla. Tästä syystä valojen on kyettävä reagoimaan nopeasti mittaustietokoneelta tuleviin käskyihin. Käytännössä tämä tarkoittaa maksimissaan noin 10 ms vasteaikaa. Tuossa ajassa ohjelmoitavan logiikan tai väyläsovittimen täytyy kyetä vastaamaan mittaustietokoneelle, että pyyntö on käsitelty.

### **5.1.4 Järjestelmä**

Mittaustietokoneessa on x86-64 arkkitehtuurin suoritin, ja siinä ajetaan Windows 7-käyttöjärjestelmää tai uudempaa Windowsia. Nykyinen ohjelmoitava logiikka on tällä hetkellä CPU 1510SP F-1 PN, mutta se voidaan korvata millä tahansa soveltuvalla ET 200SP-sarjan kanssa yhteensopivalla laitteella.

## **5.2 Modbus-yhteyden ongelmat**

Mittalaitteesta on olemassa testiversio, jossa vanha I/O-väyläsovitin on korvattu Siemensin ohjelmoitavalla logiikalla. Modbus-yhteyden kanssa on kuitenkin ollut ongelmia, eikä tieto aina siirry yhteyden yli. Ongelmaa selvitettiin lähettämällä mittaustietokoneelta paketteja sekä uudelle ohjelmoitavalle logiikalle että vanhalle väyläsovittimelle, ja tallentamalla verkkoliikennettä Wireshark-



ohjelmalla. Syntyneitä tietoliikennelokeja vertailtiin, ja havaittiin että vaikka paketit ovat sisällöltään identtisiä, on ohjelmoitavan logiikan vasteajat (noin 35 ms) ovat huomattavasti pidemmät kuin vanhan I/O-väyläsovittimen (noin 4 ms). Mittaustietokoneessa olevaan ohjelmaan on asetettu aikakatkaisuksi 12 ms, mikä on syynä ongelmaan.

Aikakatkaisun aikaa kasvatettiin, mutta uusi viive on silti liian pitkä, jotta laite voisi toimia luotettavasti. Tästä syystä logiikkaohjelmaan tehtiin muutos, jossa Modbus-palvelin siirrettiin 4 ms välein syklisesti suoritettavaan lohkoon. Tämä takaa sen, että Modbus-liikenteen hoitamiselle on aina tarpeeksi prosessointitehoa saatavilla. Muutoksen jälkeen viiveet ovat hyväksyttävällä tasolla, noin 10 ms. Tämän perusteella voidaan todeta, että Modbus-yhteys ja Siemensin ohjelmoitava logiikka ovat toimiva yhdistelmä, ja yksi mahdollinen vaihtoehto opinnäytetyön tämän osion ratkaisuksi.

### **5.3 Korvaaja Modbus-yhteydelle**

Tässä kappaleessa on listattu eri vaihtoehdot nykyiselle Modbus-yhteydelle ja niistä tehdyt havainnot. Vaihtoehtoja kartoitettiin tutkimalla Siemensin omia tuotteita, sekä hakemalla yleisesti internetistä tietoa.

#### **5.3.1 Vaihtoehto 1: Siemensin Profinet ajuri**

Siemensin Profinet ajuri on ohjelmisto ja kehitystyökalu, jonka avulla mistä tahansa tietokoneesta saadaan tehtyä Profinet-isäntälaitte. Käytännössä se on CD-levyllä toimitettu C/C++ lähdekoodi, joka voidaan sisällyttää mukaan kehityksen kohteena olevan projektin koodeihin. Ajuria on mahdollista käyttää ainakin Windows 7 ja Debian Linux käyttöjärjestelmillä. (Delivery Release of the PROFINET Driver for Controllers V1.1 2015.)

Ohjattavien laitteiden määrä on rajoitettu 16 laitteeseen Windowsilla, jolla ajuria tässä projektissa ajettaisiin. Raja ei nykyisellään ylity (taulukko 1), mutta saattaa tulla vastaan tulevaisuudessa, mikäli laitteeseen lisätään ominaisuuksia. Ajuri ei tue MRP-protokollaa eikä toimi älykkäänä laitteena (I-device). Näillä rajoitteilla ei kuitenkaan ole merkitystä tämän projektin osalta. Päivityssyklin pituus ajurilla on 32ms, ja dataa on mahdollista siirtää 4096 tavua sisään ja ulos, mikä riittää hyvin valojen ohjaukseen. 4096 saattaa (tosin epätodennäköisesti) muodostua ongelmaksi, mikäli lait-

teessa on kaikki lisäosat käytössä. Tämän varmistaminen vaatisi kattavaa selvitystä laitteen tietoliikenteestä. Ajurille on myös varattava oma Ethernet-adapteri tietokoneesta, mutta tämäkin toteutuu jo valmiiksi. (PN driver product information and support files; Delivery Release of the PROFINET Driver for Controllers V1.1 2015.)

Ajuri on testattu Siemensin puolesta ainoastaan Windows 7-käyttöjärjestelmällä, käyttäen WinPcap-kirjastoa, jonka kehitys on lopetettu vuonna 2018. Ainoa maininta asiasta on vuodelta 2013, joten asia on hyvinkin voinut muuttua. Koska Siemens ei ole julkistanut kattavaa versiohistoriaa ajurin kehityksestä, on hieman epäselvää, kuinka hyvä tuki ajurilla on uudemmissa ympäristöissä. (PN driver product information and support files.)

Ajuri vaatii myös kehityslisenssin sekä käyttölisenssejä jokaista versiota varten, jota ajetaan muulla kuin Simatic laitteistolla. Siemens ei itse jaa hintoja julkisesti, mutta tutkimalla usean verkkokaupan hintoja voidaan melko turvallisesti sanoa kehittäjälisenssin hinnan olevan noin 500 € ja käyttölisenssin noin 250 €. (Updating the PROFINET driver to V2.1 2018.)

Lopputulemana Profinet-ajuri on melko hyvä vaihtoehto, mutta riippuen lisenssien hinnasta, saattaa tuoda ylimääräisiä lisäkustannuksia. Vapaasti saatavilla oleva dokumentaatio ei myöskään ollut kovin kattava, joten tekniset yksityiskohdat tulisi vielä varmistaa.

### **5.3.2 Vaihtoehto 2: Kolmannen osapuolen Profinet ajurit**

Kolmannen osapuolen Profinet-ajureita tutkittiin vaihtoehtona, mutta yhtäkään sellaista ei löytynyt.

### **5.3.3 Vaihtoehto 3: Pysytään Modbus-yhteydessä**

Vaihtoehtojen kartoituksen aikana Siemensin omasta valikoimasta löytyi väyläsovitin (IM 155-6 MF HF), joka tukee natiivisti Modbusia. Modbusissa ei ole itsessään mitään varsinaista ongelmaa, miksi sen käyttöä ei voisi jatkaa. Profinetiä käytettäessä päästäisiin yhdestä ylimääräisestä protokollasta eroon ja Profinet tarjoaisi paremman skaalautuvuuden tulevaisuutta ajatellen.

#### 5.3.4 Vaihtoehto 4: Snap7

Snap7 on Davide Nardellan henkilökohtaisena projektina kehittämä vapaa kirjasto, joka hyödyntää Siemensin S7-protokollaa vaihtaakseen dataa Siemensin logiikoiden kanssa. Kirjasto on julkaistu GNU LGPL version 3 alla, ja on näin ollen vapaasti käytettävissä myös suljetussa ympäristössä. Koska kirjasto on yhden henkilön täysin vapaaehtoisesti kehittämä, on ymmärrettävää, että tekijä sanoutuu irti kaikesta mahdollisesta vastuusta, jota kirjaston käytöstä voi mahdollisesti seurata. Tästä syystä sen mahdollista käyttöä laitteessa, jossa voi olla terveydelle vaarallisia liikkuvia osia, ja josta riippuu kallis tuotantoprosessi, ei voida pitää järkevänä ratkaisuna. Mikäli lopputuloksen testaisi erittäin kattavasti, voisi kirjaston käyttö periaatteessa olla turvallista. (Nardella.)

#### 5.3.5 Vaihtoehto 5: OPC UA

Siemensin ohjelmoitava logiikka tukee myös OPC UA-protokollaa (6ES7510-1SJ01-0AB0 2015). Tämä ei kuitenkaan tuo mitään lisähyötyä verrattuna nykyiseen Modbus-ratkaisuun. Ottaen huomioon, että OPC UA on Modbusia monipuolisempi ja samalla monimutkaisempi protokolla, on mahdollista että se toimisi Modbusiakin hitaammin. Tätä pitäisi testata käytännössä, jotta asiaan voisi saada varmuuden, mutta kokeiluun menevä aika ja resurssit (ohjelmoitavan logiikan uudelleen konfigurointi ja ohjelmamuutokset sekä mittaustietokoneen OPC UA rajapinnan ohjelmointi) on todennäköisesti järkevämpää käyttää johonkin muuhun.

#### 5.3.6 Vaihtoehto 6: Siemens Open Controller

Siemensillä on myös PC-pohjainen kontrolleri. Käytännössä kyseessä on hieman ohjelmoitavaa logiikkaa muistuttavaan pakettiin rakennettu tietokone, jossa ajetaan Windows 10 käyttöjärjestelmää, ja joka sisältää valmiiksi asennettuna S7-1500 kontrolleriohjelmiston. Mapvisionilla on kuitenkin jo olemassa oleva tietokone, johon täytyy olla mahdollista asentaa lukuisia lisäkortteja, jota Siemensin tietokone ei jo fyysisen kokonsa vuoksi mahdollista. Näin ollen se ei ole käypä ratkaisu tähän työhön. (SIMATIC ET 200SP - The compact IO system for control cabinets.)

### 5.4 Lopputulos

Modbus-yhteydessä esiintyneet ongelmat saatiin korjattua, ja tämän johdosta lopulta päädyttiin jatkamaan nykyisen Modbus-yhteyden kanssa, koska se on todettu tarpeeksi hyvin toimivaksi, ja

muut vaihtoehdot tuovat mukanaan myös huomattavia kustannuksia ja riskitekijöitä. Toimeksiantajan laitteiston versioissa, jotka eivät tarvitse ohjelmoitavaa logiikkaa toimiakseen, voidaan käyttää Siemensin M 155-6 MF HF väyläsovittinta, sillä edellytyksellä että I/O-osoitteet vaihdetaan mitaustietokoneesta yhteensopiviksi Siemensin kanssa.

Siemensin Profinet ajuria saatetaan harkita tulevaisuudessa uudestaan, mutta sen käyttäminen vaatii kuitenkin suoria taloudellisia investointeja sekä kohtuullista määrää ohjelmistokehitystyötä.

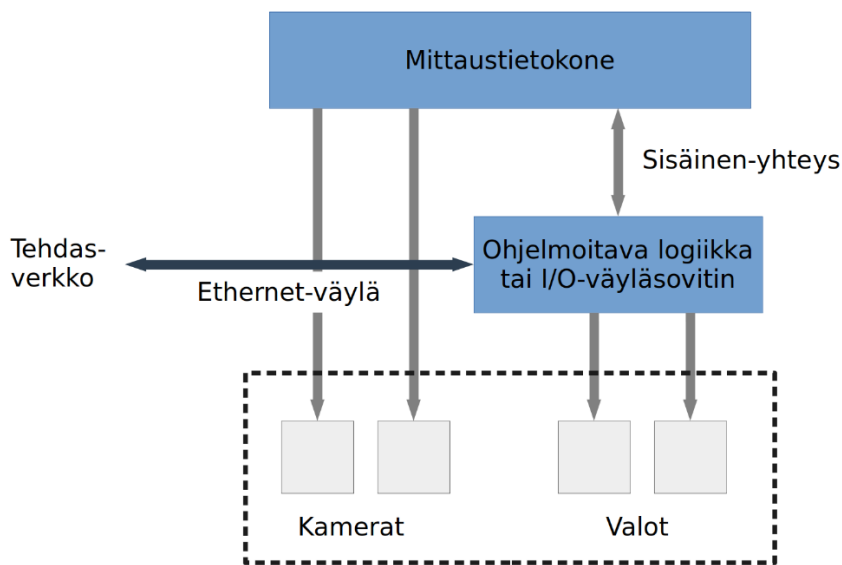
## 6 Ulkoisen rajapinnan toteutus kenttäväylänä

Tässä osiossa kerrotaan opinnäytetyön toisen osion havainnoista, joita ulkoisen ohjausrajapinnan suunnittelussa ja kehityksessä nousi esille. Vaatimuksena rajapinnalle on kohtuullinen hinta sekä hyvä yhteensopivuus ja helppo liitettävyyys useiden muiden järjestelmien kanssa. Rajapinnan toisella puolella voi käytännössä olla mikä tahansa järjestelmä, joten tiukkoja rajauksia vaatimuksille ei pysty tekemään. Tyypillinen käyttötapaus on ohjelmoitava logiikka, joka kommunikoi mittauslaitteen kanssa jotakin protokollaa hyödyntäen. Rajapinnan tulisi myös olla laajennettavissa mahdollisia tulevia toimintoja varten. Erityisen aikakriittinen rajapinta ei ole. Käytännössä esimerkiksi 100 ms vasteajat ovat riittävän nopeita.

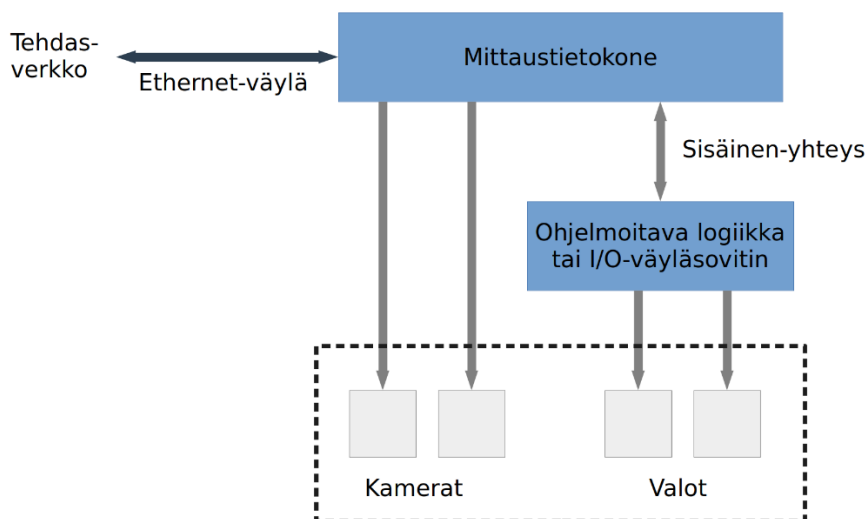
Heti työn alussa oli selvää toimeksiantajan toiveiden pohjalta, että vanha perinteisellä I/O:lla toteutettu rajapinta haluttaisiin korvata Ethernet-pohjaisella ratkaisulla. Laitteistoa ohjaamassa on tietokone, jossa on jo nykyisellään lukuisia verkkokortteja, joten uuden Ethernet-portin lisääminen on yksinkertaista ja edullista, esimerkiksi verrattuna Profibusiin, joka vaatisi erillisen sovittimen väylän ja tietokoneen välille. Ethernet on myös hyvin testattu ja aikaa kestänyt standardi, jonka käyttö myös teollisuudessa on yleistynyt viime aikoina (Zhihong & Pearson 2018, 4), joten asiakkaalla voidaan olettaa olevan laitteita, joista Ethernet rajapinta löytyy.

Ethernetin päällä toimivia teollisuusprotokollia on kehitetty lukuisia, kuten aiemmin tässä opinnäytetyössä on kuvattu. Ensimmäinen työ oli selvittää mitä protokollia haluttaisiin lähteä toteuttamaan ja miten. Vaihtoehtoja oli käytännössä kaksi. Ensimmäinen mahdollisuus olisi käyttää laitteessa olevaa Siemensin ohjelmoitavaa logiikkaa Ethernet-liikenteen käsittelyyn (ks. kuvio 6). Tämä kuitenkin rajoittaa tuetut protokollat niihin, mitä ohjelmoitava logiikkakin tukee. Toiseksi,

kuten työn ensimmäisessä osiossa todettiin, kaikkiin laitteisiin ohjelmoitavaa logiikkaa ei välttämättä haluta. Kolmanneksi tämä ratkaisu on kokonaisuudessaan monimutkaisempi kuin toinen vaihtoehto, joka on Ethernet-rajapinnan toteutus mittaustietokoneeseen (ks. kuvio 7). Tietokone on se jolle rajapinnasta tulevat käskyt joka tapauksessa lopulta päätyvät, joten laitteiston arkkitehtuurin kannalta on yksinkertaisempaa käsitellä Ethernet-liikenne suoraan tietokoneessa itsessään. Tämä myös mahdollistaa teoriassa tuen usealle eri protokollalle, joista toimeksiantajalla on täysi kontrolli. Asiakkaat voivat täten valita käyttöönsä itselleen parhaiten sopivan protokollan.



Kuvio 6. Vaihtoehto 1. Ethernet-väylän toteutukselle mittauslaitteistossa

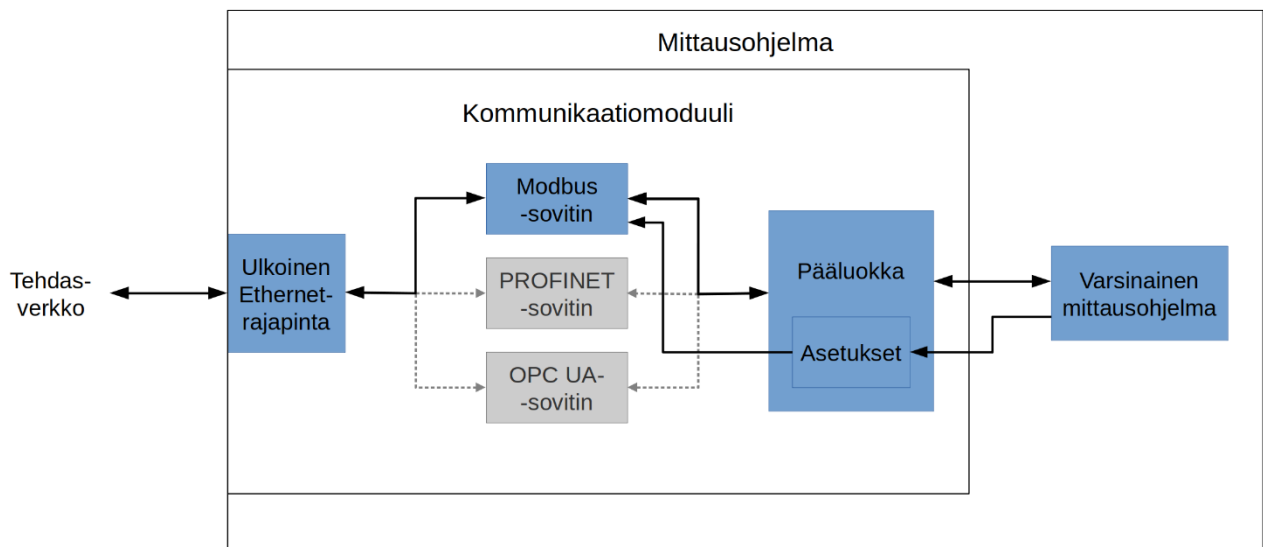


Kuvio 7. Vaihtoehto 2. Ethernet-väylän toteutukselle mittauslaitteistossa

## 6.1 Kommunikaatiokirjasto

Alun suunnittelun pohjalta päädyttiin ratkaisuun, jossa tietokoneelle ohjelmoidaan kirjasto, joka toimii varsinaisen mittausohjelman ja Ethernet-portin välissä. Kirjasto toteuttaa ohjelmointirajapinnan mittausohjelman puolelle, josta mittausohjelma voi kuunnella erilaisia käskyjä (esim. ”aloita mittaus”) sekä päivittää tietoja mittauksen etenemisestä ja tuloksista. Ethernet-portin puolelle toteutetaan useampia valittavissa olevia rajapintoja eri teollisuusprotokollille. Kirjaston nimeksi valittiin ”CommunicationModule” koska se hoitaa tiedon välitystä mittaohjelman ja Ethernet-rajapinnan välillä. Siihen viitataan myöhemmin tässä työssä nimellä ”kommunikaatiomoduli”.

Kirjasto koostuu pääluokasta sekä useammasta sovitinluokasta (ks. kuvio 8), jotka hoitavat viestintän Ethernet-portin suuntaan valitun protokollan mukaisesti. Pääluokan tehtäviä ovat rajapinnan abstraktointi, kirjan pitäminen mittaussyklin tiedoista ja puskurointi sovitimille sekä sovitimien käynnistäminen ja sammuttaminen asetusten mukaisesti. Sovitinluokkaa ajetaan omassa säikeessään, jotta se kykenee vastaamaan verkon kautta tuleviin pyyntöihin milloin tahansa.



Kuvio 8. Kommunikaatiomodulin rakenne ja suhde muuhun mittausohjelmaan

Abstraktoinnista on useanlaista hyötyä. Mikäli itse mittausohjelmaan tulee joskus muutoksia, jotka edellyttävät myös muutoksia mittaohjelman ja kommunikaatiomodulin rajapintaan, tarvitsee muutokset tehdä vain pääluokkaan, ja sovitinluokat voidaan pitää ennallaan. Toisaalta sama pätee

myös toiseen suuntaan. Jos sovittimiin täytyy tehdä muutoksia, voidaan rajapinta mittaohjelmistolle päin pitää ennallaan. Pääluokassa voidaan myös pitää toimintoja, jotka ovat kaikille sovittimille yhteisiä. Tämä helpottaa ylläpitoa ja vähentää virheiden mahdollisuutta. Mittaohjelmiston koodi pysyy myös siistimpänä, kun kaikki sovittimien hallintaan liittyvä koodi on eristettynä omaan luokkaansa.

Pääluokka pitää kirjaa mittasyklin etenemisestä sekä relevanttien muuttujien tiloista. Ratkaisu ei ole täysin optimaalinen, sillä nämä tiedot ovat osittain päällekkäistä mittaohjelmiston tilatietojen kanssa. Muuttujat ovat aikaisemmin olleet tallennettuna erillisen fyysisen Modbus-väylämoduulin rekistereihin, joten niiden pitäminen tallessa kommunikaatiomodulissa on välttämätöntä, koska Modbus-väylämoduulia ei tässä rajapinnassa ole käytössä. Tietojen hakeminen tarvittaessa suoraan mittaohjelmalta ei myöskään ole mahdollista, sillä mittaohjelma pyörii synkronisesti ja liikenne Ethernet-rajapintaan voi tulla milloin tahansa. Toisin sanoen rajapinta ei kykenisi vastaamaan pyyntöihin, jos mittaus on kesken.

Kommunikaatiomoduuili myös ylläpitää tietoa mittasyklin tilasta (esim. "odottaa aloitusta", "mittaus kesken", jne.). Tämäkin on perusteltua, sillä mittaohjelmassa tiedot eivät ole varsinaisesti tallennettuna mihinkään, vaan käytännössä tila määräytyy sen mukaan mitä osaa koodista ollaan parhaillaan suorittamassa. Kommunikaatiomoduulin sovittimet kuitenkin tarvitsevat tätä tietoa jälleen asynkronisesti, joten sen tallentaminen muuttujiin oli välttämätöntä.

Huono puoli tietojen kahdentamisessa on, että tehtäessä muutoksia mittaohjelmaan tai kommunikaatiomoduuliin on oltava erityisen tarkkana, että mittasyklissä tapahtuvat muutokset peilautuvat kommunikaatiomoduulin sisäisiin muuttujiin.

Koko kommunikaatiomoduuili on ohjelmoitu C++-kielellä. Verkkoliikenteen käsittelyyn käytetään Windowsin winsock2-kirjastoa. Tietojen vaihto sovittimen ja pääluokan välillä perustuu jaettuun muistiin sekä condition variable-muuttujiin. Sovitinluokkia toteutettiin lopulta kaksi. Modbus-palvelin ja Modbus-asiakasohjelma. Muita protokollia ei toteutettu toistaiseksi. Syistä on kerrottu lisää seuraavassa kappaleessa.

## 6.2 Testaus

Valmista kommunikaatiomoduulia on testattu simuloitussa ympäristössä. Mittausohjelmisto asetettiin käyttämään tekaistuja kuvia mittauksiin, ja Modbus-liikennettä voitiin lähettää ja vastaanottaa tarkoitusta varten ohjelmoidulla työkalulla. Mittausta on testattu erilaisilla asetuksilla, niin että kaikki vaaditut toiminnot on saatu testattua. Näitä toimintoja ovat mm. mittauksen aloitus, mittauksen pysäytys ja jatkaminen, tuotteen valinta, sarjanumeron lähetys, systeemivirhe sekä neljä skriptikomentoa. Sekä palvelimelle että asiakasohjelmalle on suoritettu samanlainen testaus. Testaus on suoritettu kokonaan manuaalisesti, ja automaattisia testejä kommunikaatiomoduulille ei ole tehty. Toimeksiantaja suorittaa itsenäisesti vielä kattavampia testejä todellisella laitteistolla, mutta niiden suoritus on tätä työtä kirjoittaessa vielä kesken. Testauksen jälkeen kommunikaatiomoduuli liitetään osaksi virallista mittausohjelmistoa.

## 6.3 Haasteet

Valittaessa mitä protokollia kommunikaatiomoduuli toteuttaisi, törmättiin nopeasti ongelmiin. Modbusin lisäksi ei löytynyt yhtäkään protokollaa, jonka toteutus olisi ollut mahdollista työn sallimien resurssien rajoissa. Työtä varten tutkittiin Modbusin lisäksi mahdollisuutta toteuttaa Profinet, OPC UA sekä EtherNet/IP protokollia. Kaikkien kohdalla kävi kuitenkin ilmi, että niiden toteutus vaatii kalliiden dokumentaatioiden tai lisenssien ostamista, tai rekisteröitymistä viralliseksi toimittajaksi. Kirjaston kehitys vei myös paljon aikaa, joten senkään puolesta muita protokollia ei olisi ehditty toteuttaa opinnäytetyön resursseilla. Lopputuloksena päädyttiin toteuttamaan vain Modbus, joka on vapaa lisensseistä. Kirjasto pyrittiin kuitenkin toteuttamaan niin että muita protokollia olisi helppo lisätä jälkikäteen.

Toteutuksen yhteydessä tuli myös odotettua enemmän ongelmia kirjaston sovittamisessa mittausohjelman kanssa. Mittausohjelmaa on kehitetty pitkään ja siihen on ajan kanssa kertynyt runsaasti ominaisuuksia, joihin perehtyminen vie aikaa, eikä kommunikaatiomoduulia voitu myöskään kehittää valmista rajapintaa vasten, koska vastaavaa ratkaisua ei ole mittausohjelmassa aiemmin ollut. Näin ollen rajapinta piti luoda kehittämisen yhteydessä tutkimalla mittausohjelman vanhaa koodia. Mittausohjelman ominaisuuksia ja koodipohjaa käytiin läpi ennen kommunikaatiomoduulin varsinaisen kehityksen aloittamista. Väärinkäsityksen johdosta, vaatimus pystyä ohjaamaan mitä tahansa lähtöä päälle tai pois jäi kuitenkin suunnittelussa huomiotta, ja tämä puute tuli esille vasta



kun kommunikaatiomoduulin kehitys oli jo pitkällä. Seurauksena kommunikaatiomoduuliin jäi useampia sisäisiä muuttujia, jotka esittävät samaa asiaa. Niiden siistiminen pois tulevaisuudessa on suositeltavaa. Pääluokan tietomalleja voi olla myös syytä hioa tämän vaatimuksen johdosta.

Oman haasteensa työhön toi eri protokollien spesifikaatioiden saatavuus. Käytännössä Modbus on protokollista ainoa, jonka spesifikaatiot ovat vapaasti ja ilmaiseksi jaossa. Muiden osalta itse spesifikaatioihin ei päästy tässä työssä käsiksi, vaan tässä työssä tehdyt havainnot perustuvat vapaasti saatavilla olevaan tietoon näistä protokollista. Onneksi tätä tietoa on runsaasti jaossa protokollien virallisissa lähteissä, joten itse spesifikaatioiden puute ei aiheuttanut esteitä työn loppuun viemiseksi.

## **6.4 Huomiot**

Toimiva Modbus-ohjausrajapinta saatiin ennakoidusti kehitettyä. Modbus-rajapintaa varten tehtiin kaksi sovitinluokkaa. Toinen luokka toimii Modbus-palvelimena ja toinen asiakkaana. Näistä voidaan valita yksi käyttöön kerrallaan, asiakkaan tarpeiden mukaan. Tässä kappaleessa kerrotaan sovittimien kehittämisen yhteydessä nousseista huomioista.

### **6.4.1 Eroavaisuudet perinteisen I/O:n ja Ethernet-pohjaisen kommunikoinnin välillä**

Mittausohjelman ulkoinen rajapinta on aikaisemmin perustunut perinteisille I/O-signaaleille. Perinteisen I/O:n toiminnassa on merkittävä ero Ethernet-pohjaiseen rajapintaan verrattuna – I/O-signaali voi muuttua milloin tahansa ja sitä lukevan pään vastuulla on huomata signaalin muutos. Ethernet-pohjaisessa ratkaisussa signaalin muutos puolestaan päivitetään kuuntelevalle osapuolelle, jonka ei tarvitse aktiivisesti seurata kyseisen signaalin tilaa. Tämä ero huomataan parhaiten pulssityyppisten signaalien kohdalla. Perinteisessä järjestelmässä pulssin keston tulee olla riittävän pitkä (esim. 1 sec), että sitä lukeva osapuoli ehtii varmasti rekisteröidä pulssin. Ethernetin yli pulssin pituudeksi taas riittää, että lähetetään yksi paketti, eli periaatteessa pulssilla ei tarvitse olla määritettyä pituutta ollenkaan.

Toinen huomioitava asia, jota Modbus-protokolla ei tosin tue, mutta esimerkiksi OPC UA tukee, on menet. Eli sen sijaan että vaihdettaisiin yhdenkään bitin tilaa, lähetetään suorituskäsky jollekin menetille. Tämän sovelluksen osalta mittauksen aloitus- ja jatkamiskäskyt sopisivat tähän malliin hyvin, sen sijaan, että vaihdetaan mittauksen aloitusbitti päälle ja taas pois, voitaisiin vain lähettää yksi pyyntö aloittaa mittaus.

Profinet ja OPC UA tukevat hälytyksiä. Nykyisessä järjestelmässä tähän sopivia tietoja olisi "system warning" ja "system error" bitit. Erityisesti Profinetin malliin nämä sopivat hyvin, sillä "system warning" bitti vastaa Profinetin hälytystasoa "huolto tarpeen" (engl. maintenance required) ja "system error" puolestaan vastaa hälytystasoa "huolto vaaditaan" (engl. maintenance demanded). Hälytyskanavien käyttö ei tietysti ole pakollista, mutta jos standardi sellaisen tarjoaa, on niiden käyttö suositeltua. (PROFINET System Description, Technology and Application 2014, 11; OPC UA Interoperability for Industrie 4.0 and the Internet of Things 2019, 23.)

Järjestelmä sisältää myös "live signal"-bitin, joka on käytännössä säännöllisin väliajoin päälle ja pois menevä bitti. Tämän avulla voidaan havaita, jos mittalaitteen ohjelma on jäänyt jumiin. Tätä ei ole tosin vielä toteutettu kommunikaatiomoduliin, sillä sen ei katsottu olevan välttämätön laitteen toiminnan kannalta. Tämänkin ominaisuuden toteutukseen voidaan harkita edellisessä kappaleessa käsiteltyjä hälytyksiä. Ratkaisu toimisi käytännössä niin, että mittaohjelma päivittäisi ohjelmiston sisäisesti väyläsovittimelle säännöllisen tiedon siitä, että ohjelma pyörii yhä. Tällöin mittaohjelman jäädessä jumiin, eri säikeessä ajettava väyläsovitin huomaa ohjelman jääneen jumiin, ja lähettää väylään varoituksen. Toisaalta jos väyläsovitin itsessään jää jumiin, syntyy ongelma. Modbusia käytettäessä tämä tulisi esille sillä, että laite lakkaa vastaamasta Modbus-pyyntöihin. Mutta yhteydettömien protokollien kohdalla (UDP tai Profinetin tapauksessa protokollan oma toteutus) ei väyläsovittimen jumiutumista tai kaatumista voida luotettavasti havaita. Parempi ratkaisu lienee siis toteuttaa "live signal"-bitti myös väyläsovitimiin.

Modbus-protokolla on pitkälti suunniteltu toimimaan perinteisen I/O:n kanssa, joten Modbus-sovitimet on myöskin ohjelmoitu vastaamaan mahdollisimman hyvin tätä käyttäytymistä, ja ovat pitkälti jo valmiita. Yllä olevat asiat on kuitenkin syytä ottaa harkintaan tulevaisuudessa kehitettäessä tukea uusille protokollille.

## 6.4.2 Osoitevaruus

Tässä kappaleessa kerrotaan protokollien erilaisten osoitevaruuksien tuomista haasteista, sekä mahdollisista ratkaisuista niihin. Eri protokollissa on hyvin erilainen lähestymistapa siihen, millälaisilla osoitteilla tietoja luetaan tai kirjoitetaan.

Ethernet-pohjaisina protokollina kaikki tässä työssä huomioidut protokollat tukevat laitekohtaisena osoitteena IP- ja/tai MAC-osoitteita. Laitteen sisäistä osoitevaruutta tarkastellessa protokollat voidaan puolestaan ajatella jaettavan kahteen ryhmään. Nämä ryhmät ovat juoksevaan numerointiin perustuvat (Modbus ja Profinet) ja oliopohjaiset protokollat (EtherNet/IP ja OPC UA). Juoksevana numerointina toteutettu osoitteistus aloittaa osoitteet nolasta ja jatkaa siitä ylöspäin. Oliopohjainen osoitteistus koostuu olioista, joilla on mielivaltainen osoite ja joka pitää sisällään lisää mielivaltaisesti määritettyjä osoitteita, joiden takana voi olla esimerkiksi arvo tai toinen olio. Oliot muodostavat siis samankaltaisen puurakenteen kuin kotitietokoneen kansiot.

Kaikki protokollat eivät myöskään tee eroa sisään tulevan ja lähtevän tiedon välillä, eivätkä erottele eri tietotyyppejä samalla tavalla. Esimerkiksi Modbusissa on omat osoitealueet yksittäisille bittien ja 16-bittisille arvoille, kun taas Profinetissä nämä voidaan määritellä korttipaikkakohtaisesti. Tämä johtaa väistämättä siihen, että perinteiseen I/O-ohjaukseen perustuvia osoitteita ei voi kaikkien protokollien kanssa käyttää, vaan osoitteet on määriteltävä protokollakohtaisesti.

Perinteisellä I/O:lla toteutetussa rajapinnassa oli hyvin rajallinen määrä signaaleja käytössä, vain 16 bittiä ulospäin ja 8 sisään. Tästä syystä mittaohjelmaan on jätetty vapaus ohjata toimeksiantajan omalla skriptikielellä mitä tahansa lähtöjä päälle tai pois, riippumatta onko kyseiselle lähdölle jo määritetty jokin vakio toiminto. Modbus-sovittimien kanssa tämä vaatimus ei ole ongelma, sillä olihan perinteinenkin I/O toiminut Modbus-yhteyden varassa. Tämä on kuitenkin hankalampi toteuttaa oliopohjaisten protokollien kanssa (OPC UA ja EtherNet/IP). Näillä protokollilla ei ole samanlaista numeerista osoitteistusta, vaan jokaiselle arvolle voidaan antaa lähes mielivaltainen osoitepolku. EtherNet/IP-standardi itseasiassa estää nykyisten osoitteiden käytön, sillä ne ovat jo varattu protokollan yleisille ominaisuuksille, eivätkä näin ollen ole käytettävissä (Schiffer 2016, 13).

Vapaasti ohjelmitaviin arvoihin liittyvä ongelma voidaan kiertää tekemällä osoitteenmuunnos skriptikielen osoitteesta protokollakohtaisiksi osoitteiksi. Tämä takaa hyvän taaksepäinyhteensopivuuden, mutta huonona puolena on järjestelmän muuttuminen monimutkaisemmaksi, koska tällöin yhdellä arvolla on sekä protokollakohtainen, että ns. perinteiseen I/O:hon pohjautuva osoite. Skriptin kirjoittaja joutuu siis ensin tarkistamaan mikä on protokollakohtainen osoite, ja muuntaamaan sen sitten skriptikielen osoitteeksi.

Toinen ratkaisumalli olisi ohjelmoida skriptikieleen tuki eri protokollien osoitteille. Tämän huono puoli on se, että tällöin skriptiohjelma tulee riippuvaiseksi valitusta protokollasta. Tällöin yhdelle protokollalle kirjoitettu skripti ei toimi suoraan toisen protokollan kanssa. Tämä myös sotii vastaan kommunikaatiomodulilla tavoiteltua hyötyä abstraktoida ohjausraja- pinta mittaohjelmalle. Toisin sanoen, mittaohjelmassa olisi otettava huomioon käytössä oleva protokolla.

Kolmas vaihtoehto on varata vapaasti ohjelmitaville arvoille oma osoitealueensa, ja estää vakio- toimintojen ylikirjoitus. Ratkaisu on varsin samanlainen ensimmäisen ratkaisun kanssa, mutta sillä erotuksella, että osoitemuunnoksista voidaan tehdä hieman käyttäjäystävällisempiä. Oliopohjaisiin protokollisiin voitaisiin esimerkiksi määrittellä yksi olio, joka pitää vain vapaasti ohjelmitavat arvot sisällään. Skriptiä kirjoittaessa ei tarvitsisi huomioida muita kuin nämä arvot, ja tahaton vakioarvon ylikirjoitus olisi mahdotonta. Tämän vaihtoehdon suuri miinus on rajapinnan taaksepäinyhteensopivuuden rikkoutuminen.

### **6.4.3 Tiedonsiirtotavat**

Modbusista poiketen muut protokollat perustuvat pääasiassa tuottaja-kuluttaja-mallille, vaikka kaikista myös löytyy jonkinlainen tuki asiakas-palvelin-mallille. Jälkimmäinen on kuitenkin tarkoitettu lähinnä konfigurointiin ja tilätietojen lukemiseen, ja esimerkiksi Profinetin tapauksessa, sillä ei edes muuta pysty tekemään (PROFINET System Description 2014, 7). Tämän ei kuitenkaan aiheuta esteitä näiden protokollien toteuttamiselle, vaan päinvastoin saattaa jopa yksinkertaistaa toteutusta. Tuottaja-kuluttaja-mallilla ei tule vastaavaa ongelmaa kuin Modbusin kanssa, missä tarvitaan omat sovitinvaihtoehdot palvelimelle ja asiakkaalle. Yhteyden molemmat päät toimisivat sekä tuottajana että kuluttajana. Mittalaitteisto lähettää ylemmälle ohjaimelle tietoa mittauksen tilasta ja tuloksista, ja ylempi ohjain lähettää mittalaitteistolle käskyn aloittaa mittaus ja muut tarvittavat parametrit.

#### 6.4.4 Laajennettavuus

Yksi syy Ethernet-pohjaiseen ohjaukseen siirtymiselle oli laajennettavuus. Lähes kaikki perinteisen I/O-liittimen koskettimet ovat jo käytössä, joten liitin pitäisi vaihtaa vielä isompaan, jos siihen haluttaisiin tulevaisuudessa jotain lisätä. Tätä ongelmaa ei kuitenkaan ole väyläprotokollien kanssa. Niiden rajapintoihin on vaivatonta lisätä uusia tietoja tulevaisuudessa. Oliopohjaisiin protokolleihin uusien tietojen lisääminen on erityisen helppoa, sillä uudet tiedot voidaan ryhmitellä mielivaltaisesti nimettyihin tai numeroituihin olioihin pitäen rajapinnan kokonaisuudessaan siistinä ja ymmärrettävänä. Modbusin ja Profinetin kohdalla etukäteen suunnittelu voi olla tarpeen, jotta toisiinsa liittyvät tiedot löytyvät loogisista osoitteista, eikä mistä sattuu, lisättyinä sitä mukaa kun ne ovat tulleet mieleen. Kummassakin tapauksessa rajapinta toimii tietysti, mutta rajapintaa käyttävän ohjelmoijan on helpompi hahmottaa kokonaisuus, mikäli tiedot ovat loogisesti ryhmitelty.

#### 6.4.5 Lisenssi, testaaminen ja sertifiointi

Modbusilla on matalin kynnys käyttöönotolle tässä työssä käsitellyistä protokollista. Sen spesifikaatiot voi ladata ilmaiseksi internetistä, eikä niiden käyttämiselle ole mitään erityisiä ehtoja (Modbus Organization End User License Agreement 2021). Modbus-organisaatio ei tarjoa virallista sertifiointiohjelmaa, mutta kehottaa silti testaamaan tuotteet virallisilla testiohjelmilla (Introduction to the Modbus TCP Conformance Testing Program 2021).

OPC on avoin standardi, eli kuka tahansa saa hyödyntää sitä haluamallaan tavalla (What is OPC? 2021). Myös OPC UA:n spesifikaatiot ovat ilmaiseksi ladattavissa internetistä. Niiden lataaminen tosin edellyttää OPC Foundation jäsenyyttä. Jäsenyyksiä on usean tasoisia, mutta alin niistä on maksuton, ja mahdollistaa OPC UA:ta hyödyntävien laitteiden kehityksen (Membership benefits 2021). OPC UA ei myöskään vaadi testausta ja sertifiointia laitteilta, mutta tällöin niitä ei saa kutsua OPC tuotteiksi, vaan niiden voidaan korkeintaan sanoa perustuvan OPC standardiin (Part 1: Overview and Concepts 2017, iv).

Profinet on vapaasti saatavilla IEC standardina (IEC-61158). IEC verkkokaupasta selviää, että standardi koostuu 88 osasta ja yhden osan hinta on keskimäärin noin 250 € (IEC 61158-1:2019 2019). Useilta tahoilta saa myös virallisen koulutuksen ja tukea PROFINET tuotteen kehitykseen, joka on

myös suositeltu tapa aloittaa Profinet-laitteen kehitys. Profinet velvoittaa, että kaikki laitteet on testattu ja sertifioitu virallisessa testilaboratoriossa (The Easy Way to PROFINET 2018, 4 - 6).

Niin ikään EtherNet/IP velvoittaa sitä käyttävien laitteiden testaamista ja sertifiointia ODVA:lla (Conformance Testing 2021). Tämän lisäksi valmistajan tulee tehdä käyttöehtosopimus ODVA:n kanssa saadakseen lisensoidun tuotetoimittajan statuksen, sekä ostaa toimittajatunnus (engl. Vendor ID), jota käytetään toimittajan yksilöimiseen tuotteissa. EtherNet/IP tuotteiden kehitys vaatii spesifikaatioiden ostamisen ODVA:lta (Getting Started 2021).

## 7 Pohdinta

Opinnäytetyössä oli tavoitteena kehittää mittauslaitteiston ohjauksia nykyaikaisemmiksi. Tämä jatkautui kolmeen osioon, joista yksi jouduttiin kuitenkin pudottamaan pois aikataulusyistä. Kahden toteutuneen osion osalta saatiin onnistuneita tuloksia, mutta jatkokehityksellekin jäi sijaa. Sekä mittauslaitteiston sisäisestä yhteydestä, että ulkoisesta yhteydestä saatiin lopputulokset, jotka tyydyttävät toimeksiantajan tarpeet.

Tutkimuksen teoriaosuus pohjautui lähes yksinomaan eri standardit omistavien tahojen viralliseen aineistoon. Itse standardien saatavuuden takia osan protokollista kohdalla tyydyttiin käyttämään muita saatavilla olevia materiaaleja, joita oli kuitenkin aivan riittävästi tarjolla tyydyttävien tulosten saamiseksi.

Kokonaisuutena opinnäytetyö oli laaja, ehkä turhankin laaja, sillä esimerkiksi ulkoisessa rajapinnassa olisi voinut jo yksinään olla tarpeeksi sisältöä yhteen opinnäytetyöhön. Aihe on myös luonteeltaan sellainen, ettei työn alussa ollut vielä kovin tarkkaa kuvaa, mitä kaikkea lopputulokseen pääseminen todellisuudessa vaatii. Tämän seurauksena työn määrä oli arvioitua suurempi.

Oman ongelmansa muodosti standardien hankala saatavuus. Tämä hankaloitti huomattavasti eri vaihtoehtojen arviointia. Työssä vertailtiin useita eri protokollia, ja osan kohdalla olisi ollut tarvetta saada vielä vähän syvempää ymmärrystä näiden toiminnasta, jotta tuloksia voisi pitää täysin kattavina. Tilanne on sinänsä erikoinen, että voidakseen tehdä päätöksen aloittaa jonkin protokollan

kehitys, pitäisi käytännössä sitoutua melko hintaviin kuluihin ja sopimuksiin. Paremmalla valmistumisella etukäteen olisi mahdollisesti voitu ehkäistä näitä ongelmia.

Tekemisen yhteydessä ilmenneistä ongelmista huolimatta, lopulta saatiin kuitenkin hyviä tuloksia. Sisäisen yhteyden osalta saatiin toimiva lopputulos, sekä pari vaihtoehtoa, joilla yhteyttä voidaan mahdollisesti tulevaisuudessa kehittää. Ulkoisen rajapinnan osalta saatiin niin ikään toteutettua toimiva kirjasto, joka on hyvä pohja jatkokehitykselle, niin itse ohjelman kehityksen, kuin myös tiedon ja ymmärryksen osalta.

Toimivan Modbus-yhteyden avulla toimeksiantaja pääsee jatkamaan laitteiston jo aloitettua päivitystä ja pitämään laitteiston komponentit ajan tasalla. Uusi ohjausrajapinta puolestaan mahdollistaa asiakkaille monipuolisemmat kytkeytymismahdollisuudet, sekä myös täysin uusien ominaisuuksien kehittämisen, jonka rajoitteena vanha liitin on ollut. Esimerkiksi teollisuus 4.0 tyyppinen datalouhinta ja -analyysi voisi olla jatkossa mahdollista, mikä saattaisi hyvinkin kiinnostaa asiakkaita, jotka haluavat seurata tuotantoprosessejaan tarkemmin. Ylipäätään jo toteutetut monipuolisemmat vaihtoehdot voivat helpottaa tuotteen myyntiä asiakkaille, ja vähintäänkin avaavat ovia uusille ominaisuuksille.

Työn tulokset ovat varsin luotettavia, erityisesti sisäisen yhteyden ja Modbus-ohjausrajapinnan osalta, sillä niiden toiminnasta on jo konkreettista näyttöä, joka on jo pitkälti testattukin. Toki on mahdollista, että toimeksiantajan omassa testauksessa löytyy vielä pieniä virheitä. Vaikka muita protokollia ohjausrajapinnassa ja sisäisen yhteyden muita vaihtoehtoja ei ole käytännössä testattu, pitäisi tulosten silti olla sinänsä oikein. Riskiksi jää lähinnä jokin huomiotta jäänyt protokollaspesifi tekninen rajoite, joka hankaloittaa tai jopa estää kyseisen protokollan käytön. Kaikkea ei ole mahdollista huomioida teoreettisesti tutkimalla, ja joskus ongelmat havaitaan vasta kun käytännössä yritetään saada asiaa toimimaan. Riski tällaiselle on kuitenkin pieni, sillä protokollien toiminta käytiin läpi niin tarkasti kuin se on mahdollista työn sallimissa rajoissa.

Jatkotutkimukselle jäi useita mahdollisuuksia. Esimerkiksi mihin tahansa ulkoisen rajapinnan toteuttamatta jääneistä protokollista voidaan lähteä perehtymään vielä syvemmin, ja suunnittelemaan niiden toteutusta ulkoiseen rajapintaan. Samassa yhteydessä nousisi varmasti myös havaintoja eri protokollien yhteensovittamisesta, joita tässä työssä ei ole tunnistettu. Myös sisäisen

yhteyden osalta voitaisiin mahdollisesti lähteä selvittämään Siemensin Profinet-ajurin käyttöönottoa, sillä sekään vaihtoehto ei ole toimeksiantajan puolelta täysin poissuljettu.



## Lähteet

Mapvision Quality Gate. 2020. Verkkosivu. Mapvision Ltd. Viitattu 26.4.2021. <https://www.mapvision.fi/mapvision-quality-gate>

About us. 2020. Verkkosivu. Mapvision Ltd. Viitattu 26.4.2021. <https://www.mapvision.fi/about-us>

Modbus FAQ. 2021. Verkkosivu. Modbus Organization. Viitattu 3.4.2021. <https://modbus.org/faq.php>

Introduction to Modbus TCP/IP. 2005. Tekninen referenssi. PDF-tiedosto. Acromag Incorporated.

Modbus Organization End User License Agreement. 2021. Verkkosivu. Modbus Organization. Viitattu 25.4.2021. <https://modbus.org/eula.php>

Introduction to the Modbus TCP Conformance Testing Program. 2021. Verkkosivu. Modbus Organization. Viitattu 25.4.2021. <https://modbus.org/certification.php>

PROFINET System Description, Technology and Application. 2014. Teknologiakuvaus. PDF-tiedosto. PROFIBUS Nutzerorganisation e.V. (PNO). Viitattu 16.4.2020. [http://us.profinet.com/wp-content/uploads/2012/11/PROFINET\\_SystemDescription\\_ENG\\_2014\\_web.pdf](http://us.profinet.com/wp-content/uploads/2012/11/PROFINET_SystemDescription_ENG_2014_web.pdf)

The Easy Way to PROFINET. 2018. Opas. PDF-tiedosto. PROFIBUS Nutzerorganisation e.V. (PNO). Viitattu 24.4.2021. <https://www.profibus.com/index.php?eID=dumpFile&t=f&f=85452&to-ken=b6aa92b46b7e239aa199e94421269a2a7acd5014>

IEC 61158-1:2019. 2019. Verkkokauppa. Verkkosivu. International Electrotechnical Commission. Viitattu 25.4.2021. <https://webstore.iec.ch/publication/59890>

OPC UA Interoperability for Industrie 4.0 and the Internet of Things. 2019. Esite. PDF-tiedosto. OPC Foundation. Viitattu 13.4.2020. <https://opcfoundation.org/wp-content/uploads/2017/11/OPC-UA-Interoperability-For-Industrie4-and-IoT-EN.pdf>

Unified Architecture. 2021. Verkkosivu. OPC Foundation. Viitattu 12.4.2021. <https://opcfoundation.org/about/opc-technologies/opc-ua/>

UA Companion Specifications. 2021. Verkkosivu. OPC Foundation. Viitattu 12.4.2021. <https://opcfoundation.org/about/opc-technologies/opc-ua/ua-companion-specifications/>

What is OPC? 2021. Verkkosivu. OPC Foundation. Viitattu 25.4.2021. <https://opcfoundation.org/about/what-is-opc/>

Membership benefits. 2021. Verkkosivu. OPC Foundation. Viitattu 25.4.2021. <https://opcfoundation.org/membership/benefits/>

Part 1: Overview and Concepts. 2017. Standardi. PDF-tiedosto. OPC Foundation. Release 1.04.

Ethernet/IP Technology Overview Series. 2016. Teknologiakuvaus. PDF-tiedosto. ODVA. PUB00138R6. Viitattu 15.4.2021. <https://www.odva.org/wp-content/uploads/2020/05/PUB00138R6-Tech-Series-EtherNetIP.pdf>

Overview of CIP security. Teknologiakuvaus. PDF-tiedosto. ODVA. PUB00319R1. Viitattu 15.4.2021. [https://www.odva.org/wp-content/uploads/2020/05/PUB00319R1\\_CIP-Security-At-a-Glance.pdf](https://www.odva.org/wp-content/uploads/2020/05/PUB00319R1_CIP-Security-At-a-Glance.pdf)

Schiffer, V. 2016. Common industrial protocol and the family of CIP networks. Teknologiakuvaus. PDF-tiedosto. ODVA. PUB00123R1. Viitattu 19.4.2021. [https://www.odva.org/wp-content/uploads/2020/06/PUB00123R1\\_Common-Industrial Protocol and Family of CIP Networks.pdf](https://www.odva.org/wp-content/uploads/2020/06/PUB00123R1_Common-Industrial Protocol and Family of CIP Networks.pdf)

SecuringEtherNet/IP™ Networks. 2011. Opas. PDF-tiedosto. ODVA. PUB00269R1.1. Viitattu 27.4.2021. [https://www.odva.org/wp-content/uploads/2020/05/PUB00269R1.1\\_ODVA-Securing-EtherNetIP-Networks.pdf](https://www.odva.org/wp-content/uploads/2020/05/PUB00269R1.1_ODVA-Securing-EtherNetIP-Networks.pdf)

Market drivers, OPC UA. 2021. Verkkosivu. ODVA. Viitattu 27.4.2021. <https://www.odva.org/technology-standards/market-drivers/process-automation/iiot-industry-4-0-2-opc-ua/>

Conformance Testing. 2021. Verkkosivu. ODVA. Viitattu 24.4.2021. <https://www.odva.org/technology-standards/conformance-testing/>

Getting Started. 2021. Verkkosivu. ODVA. Viitattu 25.4.2021. <https://www.odva.org/technology-standards/getting-started/>

Spurgeon, C. 2000. Ethernet: The Definitive Guide. O'Reilly Media, Inc. ISBN: 9781565926608.

Jiju N. 2018. TCP/IP vs UDP: What's the Difference? Artikkel. Verkkosivu. Colocation America. Viitattu 27.4.2021. <https://www.colocationamerica.com/blog/tcp-ip-vs-udp>

Introduction to IP multicast. 2006. PowerPoint-esitys. PDF-tiedosto. Cisco Systems, Inc. Viitattu 27.4.2021. [https://www.cisco.com/c/dam/en/us/products/collateral/ios-nx-os-software/ip-multicast/prod\\_presentation0900aecd80310883.pdf](https://www.cisco.com/c/dam/en/us/products/collateral/ios-nx-os-software/ip-multicast/prod_presentation0900aecd80310883.pdf)

Zhihong, L. & Pearson S. 2018. An inside look at industrial Ethernet communication protocols. PDF-tiedosto. Texas instruments. Viitattu 27.4.2021. <https://www.ti.com/lit/wp/spry254b/spry254b.pdf>

Delivery Release of the PROFINET Driver for Controllers V1.1. 2015. Tuotejulkaisu. Verkkosivu. Siemens AG. Viitattu 11.5.2020. <https://support.industry.siemens.com/cs/document/109480985/delivery-release-of-the-profinet-driver-for-controllers-v1-1?dti=0&lc=en-WW>

6ES7510-1SJ01-0AB0. 2015. Tuotetiedot. Verkkosivu. Siemens AG. Viitattu 27.4.2020. <https://support.industry.siemens.com/cs/pd/578289?pdtd=td&dl=en&pnid=13890&lc=en-WW>

SIMATIC ET 200SP - The compact IO system for control cabinets. Verkkosivu. Siemens AG. Viitattu 27.4.2020. <https://new.siemens.com/global/en/products/automation/systems/industrial/io-systems/et-200sp.html>

PN driver product information and support files. Verkkosivu. Siemens AG. Viitattu 16.4.2020. <https://support.industry.siemens.com/cs/document/86663853/pn-driver-product-information-and-support-files?dti=0&lc=en-WW>

Nardella D. Step7 Open Source Ethernet Communication Suite. Verkkosivu. Viitattu 27.4.2020. <http://snap7.sourceforge.net/>

Updating the PROFINET driver to V2.1. 2018. Verkkosivu. Siemens AG. Viitattu 11.5.2020. <https://support.industry.siemens.com/cs/document/109760216/updating-the-profinet-driver-to-v2-1?dti=0&lc=en-WW>