

Sami Martinmäki

## **KENTTÄVÄYLÄRAJAPINTA ETÄMONITOROINTIJÄRJESTELMÄLLE**

# KENTTÄVÄYLÄRAJAPINTA ETÄMONITOROINTIJÄRJESTELMÄLLE

Sami Martinmäki  
Opinnäytetyö  
Kevät 2019  
Tietotekniikan tutkinto-ohjelma  
Oulun ammattikorkeakoulu

# TIIVISTELMÄ

Oulun ammattikorkeakoulu  
Tietotekniikan tutkinto-ohjelma, laite- ja tuotesuunnittelu

---

Tekijä: Sami Martinmäki

Opinnäytetyön nimi: Kenttäväylärajapinta etämonitorointijärjestelmälle

Työn ohjaajat: Eino Niemi, Hannu-Pekka Niemelä

Työn valmistumislukukausi ja -vuosi: Kevät 2019

Sivumäärä: 37 + 2 liitettä

---

Opinnäytetyö tehtiin öljy- ja kaasuteollisuudessa käytettävän mittaus- ja monitorointitekniikan kehittämiseen erikoistuneelle Creowave Oy:lle. Työn lähtökohtana oli potentiaalisen asiakkaan esittämä toivomus mahdollisuudesta liittää Creowaven C-Site-etämonitorointijärjestelmä PROFIBUS-kenttäväylän kautta valmiiksi käytössä olevaan automaatiojärjestelmään.

Työn toteutus aloitettiin tutkimalla ja vertaamalla erilaisia laite- ja ohjelmistopohjaisia ratkaisuja, joiden avulla C-Site-järjestelmään kuuluvaan M7-palvelinohjelmistoon voitaisiin toteuttaa PROFIBUS-liitettävyyttä. Arvioinnin perusteella valitut ratkaisut joko hankittiin tai toteutettiin itse osana opinnäytetyötä.

Tuloksena syntyneessä ratkaisussa käytettiin PROFIBUS – Modbus TCP/IP -yhdyskäytävälaitteen kanssa ohjelmallista Modbus TCP/IP -protokollapinoa ja M7-rajapintaa. Ratkaisun avulla saatiin testiympäristössä siirrettyä onnistuneesti dataa C-Site-järjestelmän ja PROFIBUS-master-laitteen välillä.

---

Asiasanat: Modbus, PROFIBUS, kenttäväylät, Java, Creowave, C-Site, etämonitorointi

## ABSTRACT

Oulu University of Applied Sciences  
Degree programme in Information Technology, option of Device and Product Design

---

Author: Sami Martinmäki

Title of thesis: Fieldbus Interface for a Remote Monitoring System

Supervisors: Eino Niemi, Hannu-Pekka Niemelä

Term and year when the thesis was submitted: Spring 2019      Number of pages: 37 + 2 attachments

---

This thesis was commissioned by Creowave Oy, a technology company based in Oulu. Creowave specializes in the development and manufacturing of measuring and monitoring technology used in the oil and gas industries. The premise was a request by a potential client for the ability to integrate Creowave's C-Site remote monitoring system via PROFIBUS fieldbus to an existing factory automation system.

Thesis work began by researching and evaluating different hardware- and software-based solutions that could be used for implementing PROFIBUS connectivity to the C-Site M7 server application. Solutions selected based on the evaluation were either procured or developed as part of the thesis.

The resulting combined solution used a PROFIBUS – Modbus TCP/IP gateway device in conjunction with software-based Modbus TCP/IP stack and M7 interface. This solution was successfully used in a test environment to transmit data between the C-Site system and a PROFIBUS master device.

---

Keywords: Modbus, PROFIBUS, fieldbus, Java, Creowave, C-Site, remote monitoring

# SISÄLLYS

TIIVISTELMÄ.....	3
ABSTRACT .....	4
SISÄLLYS.....	5
1 JOHDANTO .....	7
2 CREOWAVE C-SITE ETÄMONITOROINTIJÄRJESTELMÄ.....	8
2.1 Tausta.....	8
2.2 Creowave C-Site S7 -sensoriyksikkö.....	8
2.3 Creowave C-Site R7 -tukiasema.....	9
3 KENTTÄVÄYLÄT .....	10
3.1 Määritelmä ja käyttötarkoitus .....	10
3.2 OSI-viitemalli .....	10
3.3 KytKentätopologiat .....	12
3.3.1 Tähtikytkentä.....	12
3.3.2 Väyläkytkentä.....	12
3.4 Jaetun tiedonsiirtokanavan käyttäminen .....	13
3.4.1 Kiertokysely.....	13
3.4.2 Token passing.....	14
3.4.3 Aikaväli .....	14
3.4.4 Satunnainen pääsy .....	14
3.5 Modbus TCP/IP -kenttäväylä .....	15
3.5.1 Tausta.....	15
3.5.2 Toimintaperiaate .....	15
3.5.3 Modbus-viestin rakenne .....	16
3.5.4 Tiedon mallinnus.....	18
3.5.5 Modbus-transaktio .....	19
3.6 PROFIBUS-kenttäväylä.....	21
3.6.1 Tausta.....	21
3.6.2 Toimintaperiaate .....	21
3.6.3 PROFIBUS DP .....	22
3.6.4 Syklinen tiedonsiirto.....	22
3.6.5 Asyklinen tiedonsiirto.....	23

4	KÄYTETYT LAITTEET .....	24
4.1	HMS Anybus X-Gateway AB9001 PROFIBUS – Modbus TCP/IP -yhdyskäytävä ....	24
4.1.1	Käyttötarkoitus .....	24
4.1.2	Konfiguraatio .....	24
4.2	Kunbus DF PROFI II -PROFIBUS-laitekortti .....	26
4.2.1	Käyttötarkoitus .....	26
4.2.2	Konfiguraatio .....	26
5	TOTEUTUS.....	28
5.1	Tausta.....	28
5.2	Ohjelmointi .....	28
5.3	Modbus TCP/IP -protokollapino .....	29
5.3.1	TCP/IP-yhteydet.....	29
5.3.2	Slave-laite.....	30
5.4	M7-rajapinta .....	30
5.4.1	REST-rajapinta .....	30
5.4.2	Modbus-välimuisti .....	31
5.5	Yhdyskäytävän konfiguraatio .....	31
6	POHDINTA .....	33
	LÄHTEET.....	35
	LIITTEET .....	37

# 1 JOHDANTO

Creowave Oy on öljy- ja kaasuteollisuudessa käytettävien mittaus- ja monitorointiratkaisujen kehittämiseen erikoistunut teknologia-alan yritys. Creowaven perustivat vuonna 2004 yrityksen nykyinen toimitusjohtaja Jyrki Koski yhdessä varatoimitusjohtaja Teemu Vähäkankaan kanssa. Creowave Oy työllisti vuonna 2017 yhteensä 14 henkilöä ja sen liikevaihto oli samana vuonna noin 2,7 miljoonaa euroa (Suomen Asiakastieto Oy 2017).

Joulukuussa 2018 rekrytointiyritys A-Talent otti minuun yhteyttä ja kertoi Creowaven tarjoamasta opinnäytetyöpaikasta. Opinnäytetyön tarkoituksena oli selvittää, miten Creowaven valmistama C-Site-etämonitorointijärjestelmä voitaisiin liittää automaatiojärjestelmään PROFIBUS-kenttäväylän kautta.

Automaatiojärjestelmissä laitteet ovat yleensä kytkettyinä automaatiokeskukseen kenttäväylän avulla. Kenttäväylä voi olla esim. jaettu kuparikaapeli, jota pitkin automaatiolaitteet siirtävät vuorotellen tietoa automaatiokeskukselle. PROFIBUS-kenttäväylässä tiedonsiirtoon käytetään usein kierrettyä parikaapelia RS-485-standardin mukaisesti.

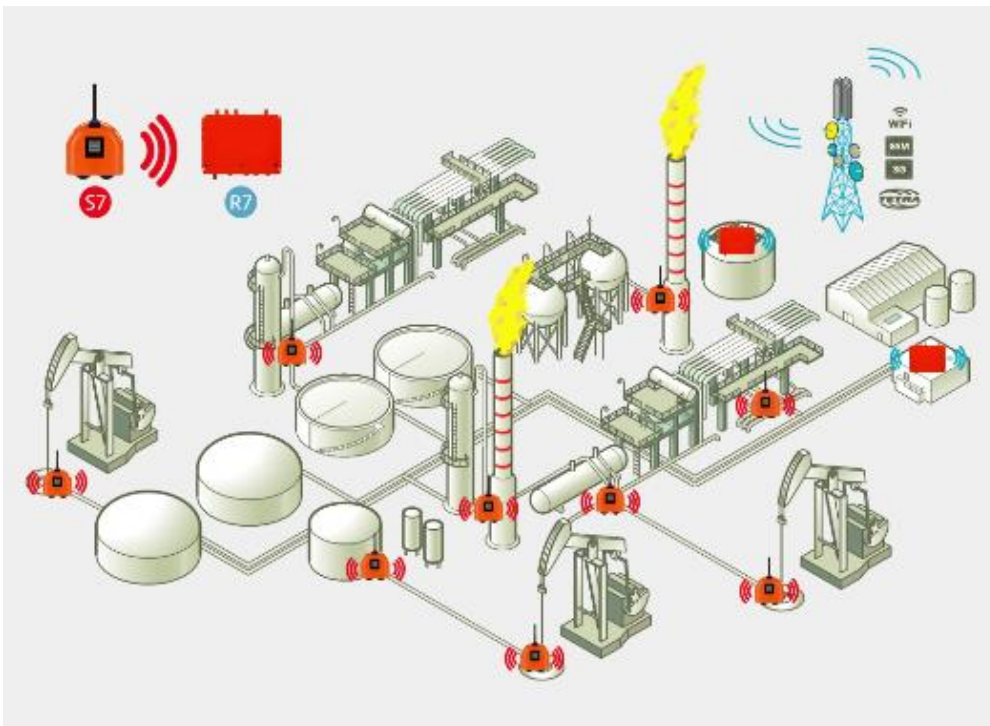
C-Site-etämonitorointijärjestelmässä mittaavat laitteet siirtävät tietoa langattomasti. Tieto kulkeutuu lopulta palvelinohjelmistolle, josta sitä voidaan lukea web-käyttöliittymän kautta. C-Site on täysimittainen järjestelmä, joka tarjoaa kaikki laitteiden monitorointiin ja ylläpitämiseen tarvittavat ominaisuudet. C-Site-järjestelmän laitteilla ei näin ollen ole lähtökohtaisesti tarvetta kenttäväyläliitettävyydelle.

Potentiaalinen asiakas oli kuitenkin kysynyt Creowavelta mahdollisuudesta yhdistää C-Site-järjestelmä valmiiksi käytössä olevaan automaatiojärjestelmään. Tähän toiveeseen perustuen aloin selvittämään ja toteuttamaan ratkaisuja C-Site-järjestelmän liittämiseksi PROFIBUS-kenttäväylään.

## 2 CREOWAVE C-SITE ETÄMONITOROINTIJÄRJESTELMÄ

### 2.1 Tausta

C-Site on Creowave Oy:n kehittämä ja valmistama etämonitorointijärjestelmä. Se koostuu kentälle asennettavista S7-sensoryyksiköistä ja R7-tukiasemista sekä M7-palvelinohjelmistosta (kuva 1). Palvelinohjelmiston avulla tukiaseman lähettämää reaaliaikaista sensoryyksiköiden mittausdataa voidaan lukea web-käyttöliittymän tai älypuhelinsovelluksen kautta. (Creowave 2015a.)



KUVA 1. Esimerkki C-Site-järjestelmästä öljykentällä (Creowave 2015a)

### 2.2 Creowave C-Site S7 -sensoryyksikkö

S7 on C-Site-järjestelmään kuuluva sensoryyksikkö (kuva 2). Siihen voidaan liittää erilaisia ulkoisia antureita konfiguroitavan anturiliitännän avulla. Anturin mittausväli voidaan säätää halutun pituiseksi ja mittatuloksille voidaan asettaa raja-arvoihin perustuvia hälytyksiä. Sensoryyksikön ominaisuuksiin kuuluvat lisäksi mm. RSSI/LQI-diagnostiikka, paristojännitteen monitorointi ja salattu tiedonsiirto. C-Site S7 on ATEX-sertifioitu tuote, joka mahdollistaa käytön räjähdysvaarallisissa tiloissa. (Creowave 2015b.)





KUVA 2. Creowave C-Site S7 -sensoriyksikkö (Creowave 2015b)

### 2.3 Creowave C-Site R7 -tukiasema

R7 on C-Site-järjestelmään kuuluva tukiasemalaite (kuva 3), johon kentällä olevat S7-sensoriyksiköt ovat yhteydessä. Sen ominaisuuksiin kuuluu mm. erilaiset diagnostikat, mittaustuloksien tallennus sekä kalibraatiotiedostojen lähetys sensoriyksiköille. R7-tukiasema on S7-sensoriyksikön tapaan ATEX-sertifioitu. (Creowave 2015b.)



KUVA 3. Creowave C-Site R7 -tukiasema (Creowave 2015b)

## 3 KENTTÄVÄYLÄT

### 3.1 Määritelmä ja käyttötarkoitus

Kenttäväylälle ei ole olemassa yhtä selkeää määritelmää. Yleisesti voidaan kuitenkin sanoa, että kenttäväylät ovat jaettuja, digitaalisia ja sarjamoitaisia tiedonsiirtoväyliä, joiden avulla älykkäitä automaatiolaitteita voidaan yhdistää toisiinsa. Prosessiteollisuudessa 1980-luvulla alkunsa saanut, saksankielinen termi "Feldbus" viittaa pääasiallisesti ns. prosessikenttään, joka kattaa tehdasalueella sijaitsevat hajautetut automaatiolaitteet, jotka ovat suoraan yhteydessä ohjattavan prosessin kanssa. Tällaisia automaatiolaitteita voidaan vastaavasti kutsua kenttälaitteiksi. (Sauter 2017, luku 1.1–1.2.)

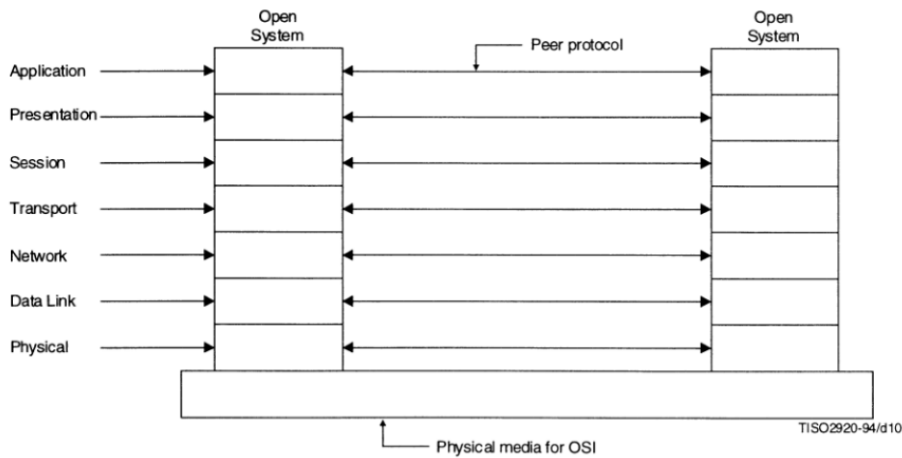
Ennen kenttäväylien kehittämistä automaatiojärjestelmien kaapeloinnista syntyvät kustannukset saattoivat kasvaa erittäin suuriksi. Yksi perimmäisistä motivaatioista kenttäväyläteknologian kehittämiseksi olikin juuri kaapelointikustannusten pienentäminen. Toinen merkittävä syy oli halu korvata perinteiset analogiset tiedonsiirtotekniikat (esim. 4–20mA virtasignaali) nykyaikaisemmalla digitaalisella tiedonsiirrolla, joka mahdollistaa suurempien datamäärien liikkumisen kenttälaitteiden ja ohjausjärjestelmien välillä. (Sauter 2017, luku 1.1–1.2.)

### 3.2 OSI-viitemalli

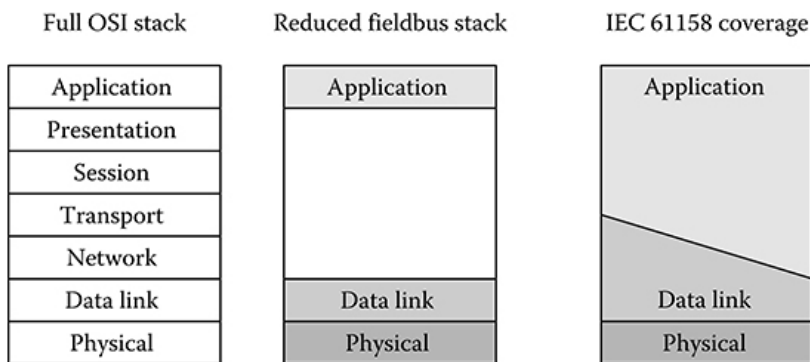
OSI (Open Systems Interconnection) on ISO:n (International Standards Organization) ja IEC:n (International Electrotechnical Commission) yhteistyössä kehittämä viitemalli, jonka tarkoituksena on tarjota yleinen lähtökohta avoimien tietojärjestelmien yhdistämiseen käytettävien teknologioiden standardoimiseksi. (ISO & IEC 1994, 1–4.)

Kenttäväyläprotokollat perustuvat OSI-viitemallin kerroksittaiseen arkkitehtuuriin (kuva 4). IEC 6158 kenttäväylästandardin mukaiset protokollat määrittelevät OSI-viitemallin seitsemästä kerroksesta vain kerrokset 7 (Application), 2 (Data link) ja 1 (Physical) (kuva 5). Täyden seitsemän kerroksen protokollapino vaatii useimmiten liikaa resursseja yksinkertaisilta kenttälaitteilta, eikä mo-

nissa varsin yksinkertaisissa kenttäväyläverkoissa ole tarvetta välissä olevien kerrosten määrittelylle ominaisuuksille. Tarpeen vaatiessa kerroksiin 2 ja 7 voidaan kuitenkin sisällyttää joitakin kerroksien 3–6 toiminnallisuuksia yksinkertaistetussa muodossa. (Sauter 2017, luku 1.5.2.)



KUVA 4. OSI-viitemallin kerrosarkkitehtuuri (ISO & IEC 1994, 28)

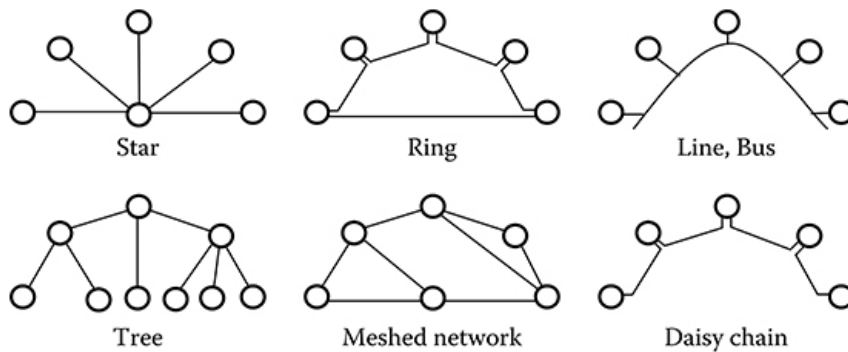


KUVA 5. OSI-viitemalli ja kenttäväylät (Sauter 2017, luku 1.5.2)

On kuitenkin olemassa kenttäväyläprotokollia, jotka määrittelevät osan tai kaikki OSI-viitemallin kerroksista 3–6. Tällaisia ovat esimerkiksi European Installation Bus ja BACnet, joita käytetään lähinnä rakennusautomaatiossa. Rakennusautomaatiossa väylään kytkettyjen laitteiden lukumäärä voi olla suuri, mistä syntyvä tarve hierarkkisesti jäsennellyille kytkentätopologioille tekee kerroksien 3–6 määrittelytarpeelliseksi. (Sauter 2017, luku 1.5.2.)

### 3.3 Kytkentätopologiat

Kytkentätopologia tarkoittaa loogista rakennetta, jonka mukaisesti samassa verkossa olevat laitteet ovat fyysisesti kytkettyinä toisiinsa. Kenttäväyläverkoissa käytettyihin kytkentätopologioihin kuuluvat mm. tähti-, rengas- sekä väylämuotoinen kytkentä (kuva 6).



KUVA 6. Yleisimmät kenttäväylissä käytetyt kytkentätopologiat (Sauter 2017, luku 1.5.3)

#### 3.3.1 Tähtikytkentä

Ennen kenttäväylien kehittämistä automaatiojärjestelmän laitteet kytkettiin yleensä tähtitopologian mukaisesti. Perinteisessä tähtikytkennässä jokainen hajautettu automaatiolaite on yhdistetty keskitettyyn, järjestelmää kontrolloivaan laitteeseen omalla kaapelillaan. Tästä syntyvät kaapelointikustannukset olivat yksi suurimmista syistä ensimmäisten kenttäväyläjärjestelmien kehittämiseen. Nykyisin tähtitopologia on kuitenkin tehnyt paluun teolliseen Ethernetiin perustuvissa järjestelmissä. (Sauter 2017, luku 1.5.3.)

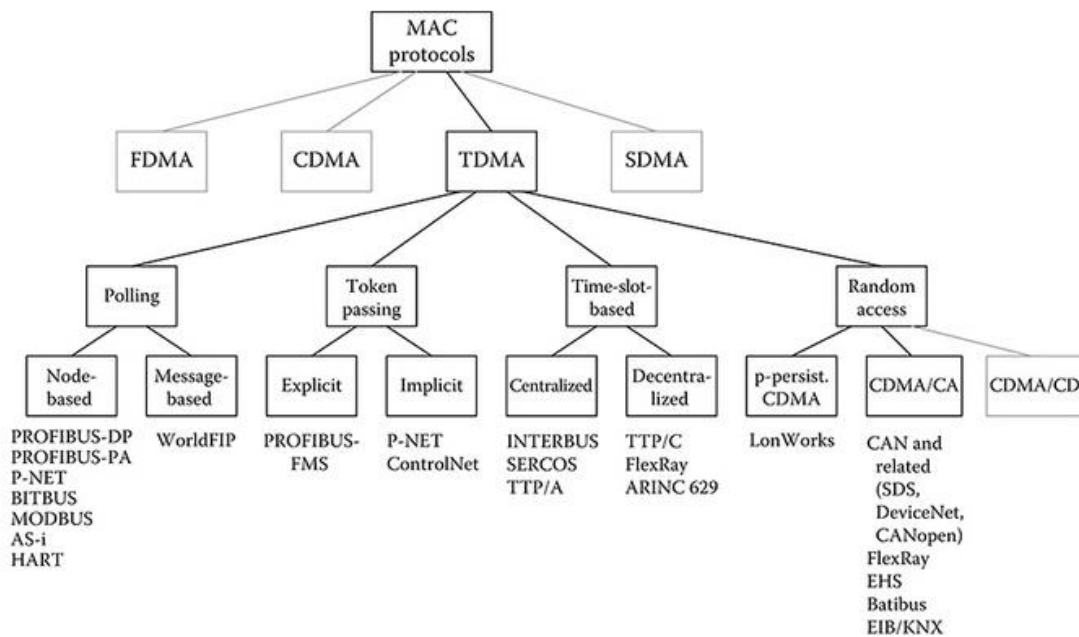
#### 3.3.2 Väyläkytkentä

Väylämuotoinen kytkentä on kenttäväyläjärjestelmissä yleisimmin käytetty topologia. Väyläkytkennässä jokainen hajautettu automaatiolaite on kytketty samaan kaapeliin, mikä teki väylätopologiasta kustannustehokkaan ja loogisen korvaajan tähtitopologialle. Laitteiden väyläkytkentä toteutetaan usein RS-485-standardin mukaisella liitännällä. (Sauter 2017, luku 1.5.3.)

### 3.4 Jaetun tiedonsiirtokanavan käyttäminen

Mikäli monta laitetta yrittää käyttää jaettua tiedonsiirtokanavaa (esim. väylä) samanaikaisesti, laitteiden lähettämät signaalit "törmäävät" ja sekoittuvat keskenään. MAC (Medium Access Control) tarkoittaa tekniikoita, joiden avulla tällaiset törmäykset voidaan joko välttää tai tiedonsiirto tehdä luotettavaksi niistä huolimatta. Tämä voi tarkoittaa esim. laitteelle annettua yksinoikeutta väylän käyttämiseen tietyinä ajanjaksona tai päällekkäisestä väylän käyttämisestä johtuvan törmäyksen havaitsemista ja viestin uudelleenlähettämistä myöhemmin.

Kenttäväyläjärjestelmissä käytetyt MAC-tekniikat perustuvat TDMA-periaatteeseen (Time Division Multiple Access), jossa laitteet käyttävät väylää vuorotellen jonkin ajanjakson verran (kuva 7). TDMA:han perustuvat MAC-tekniikat eroavat toisistaan siinä, millä tavalla (kiertokysely, aikaväli jne.) väylässä oleva laite saa tämän ajanjakson käyttöönsä. (Sauter 2017, luku 1.5.4.)



KUVA 7. MAC-tekniikat kenttäväylissä (Sauter 2017, luku 1.5.4)

#### 3.4.1 Kiertokysely

Kiertokyselyyn ("polling") perustuvassa MAC-tekniikassa on master/slave-periaatteen mukaisesti väylää kontrolloiva aktiivinen laite, joka antaa vuorotellen väylällä oleville passiivisille laitteille luvan väylän käyttämiseen. Kiertokyselyä voidaan käyttää myös järjestelmissä, joissa on useampia aktiivi-

visia laitteita, joskin tällaiset järjestelmät vaativat jonkin muun tekniikan (esim. token passing) käyttämisen väylän käyttöoikeuden jakamiseen aktiivisten laitteiden välillä. Kiertokyselyyn perustuvaa MAC-tekniikkaa käytetään mm. HART-, Modbus- ja PROFIBUS-kenttäväylissä. (Sauter 2017, luku 1.5.4.1.)

### **3.4.2 Token passing**

Token passing on MAC-tekniikka, jossa oikeus väylän käyttämiseen perustuu tasa-arvoisten laitteiden välillä kiertävän, väylän käyttöoikeutta kuvastavan informaation eli tokenin hallussapitoon. Tokenien siirto voidaan toteuttaa joko fyysisesti laitteelta toiselle lähetettävällä viestillä tai abstraktisti esimerkiksi laitteissa olevilla synkronoiduilla laskureilla. Token passing -tekniikkaa käytetään mm. PROFIBUS-kenttäväylässä master-laitteiden välillä. (Sauter 2017, luku 1.5.4.2.)

### **3.4.3 Aikaväli**

Aikaväleihin ("time-slot") perustuvassa MAC-tekniikassa jokaiselle väylään liitettylle laitteelle määritellään kaikkien laitteiden tiedonsiirrolle käytössä olevasta ajasta tietty jakso, jonka aikana laitteella on oikeus käyttää väylää. Toisin kuin kiertokyselyssä, aikaväliin perustuvassa järjestelmässä jokainen laite vastaa autonomisesti oman väyläaikansa käyttämisestä eikä tarvitse erillistä lupaa väylän käyttämiseen. Laitteille jaetut aikavälit voivat olla pituudeltaan erikokoisia tai niitä voidaan myös jakaa dynaamisesti tarpeen vaatiessa. Käytettäessä aikaväleihin perustuvaa MAC-tekniikkaa väylällä olevat laitteet tulee pitää synkronoituina. Laitteiden välinen synkronointi voidaan toteuttaa esimerkiksi master-laitteen lähettämällä viesteillä. Aikaväleihin perustuvaa tekniikkaa käytetään mm. SERCOS-kenttäväylässä. (Sauter 2017, luku 1.5.4.3.)

### **3.4.4 Satunnainen pääsy**

Satunnaiseen pääsyyn ("random-access") perustuvassa MAC-tekniikassa väylällä olevat laitteet saavat vapaasti halutessaan pyrkiä lähettämään tietoa väylää pitkin. Näin ollen laitteiden välisiä törmäyksiä ei voida kokonaan välttää, vaan MAC-tekniikan täytyy jollain tapaa käsitellä tapahtuneet törmäykset ja toimia niistä huolimatta. Tätä periaatetta kutsutaan nimellä CSMA (Carrier Sense

Multiple Access), josta on olemassa useita eri variaatioita, jotka eroavat toisistaan törmäysten käsittelyn suhteen. Satunnaiseen pääsyyn perustuvia MAC-tekniikoita käytetään mm. LonWorks- ja CAN-kenttäväylissä. (Sauter 2017, luku 1.5.4.4.)

### **3.5 Modbus TCP/IP -kenttäväylä**

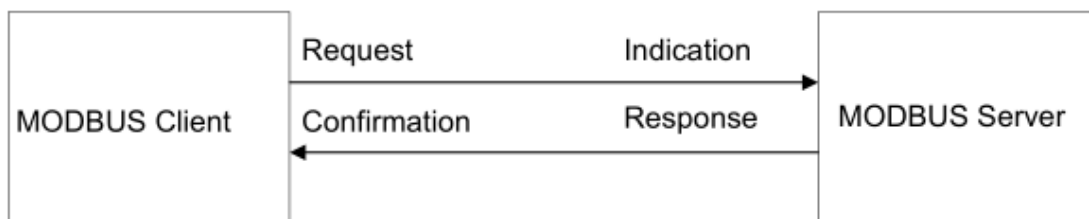
#### **3.5.1 Tausta**

Modbus on Modiconin (nyk. Schneider Electric) vuonna 1979 kehittämä avoin ja valmistajaneutraali viestintäprotokolla, jota käytetään laajasti mm. teollisuusautomaatiossa. Nykyisin Modbus-standardin ylläpitämisestä ja kehittämisestä vastaa voittoa tavoittelematon ja riippumaton järjestö Modbus Organization. Modbus-laitteita on arvioiden mukaan käytössä yli seitsemän miljoonaa kappaletta pelkästään Euroopassa ja Pohjois-Amerikassa. (Modbus Organization 2019.)

Modbus sijoittuu OSI-viitemallin 7. kerrokseen (Application Layer). Modbus-viestintä voidaan toteuttaa käyttäen useita erilaisia tiedonsiirtoprotokollia ja fyysisiä siirtoteitä. Modbusin TCP/IP-toteutuksessa Modbus-viestit liikkuvat Ethernet-verkossa TCP/IP-yhteyttä pitkin. TCP-portti 502 on varattu Modbus TCP/IP -liikenteelle. (Modbus Organization 2012, 2.)

#### **3.5.2 Toimintaperiaate**

Modbus TCP/IP on request/response tyyppinen protokolla (kuva 8). Aktiivinen master-laite ("client") lähettää kyselyn ("request") sitä kuuntelevalle passiiviselle slave-laitteelle ("server"), joka vuorostaan lähettää master-laitteelle vastauksen ("response") prosessoituaan kyselyn. Alkuperäisestä, jaettuun sarjaväylään perustuvasta toteutuksesta poiketen Modbus TCP/IP:ssä useita viestejä voi kulkea Ethernet-verkossa samanaikaisesti molempiin suuntiin, minkä takia Modbus TCP/IP:ssä ei ole tarvetta erilliselle MAC-protokollalle (Dutertre 2008, 4–5).



**A MODBUS Request** is the message sent on the network by the Client to initiate a transaction,

**A MODBUS Indication** is the Request message received on the Server side,

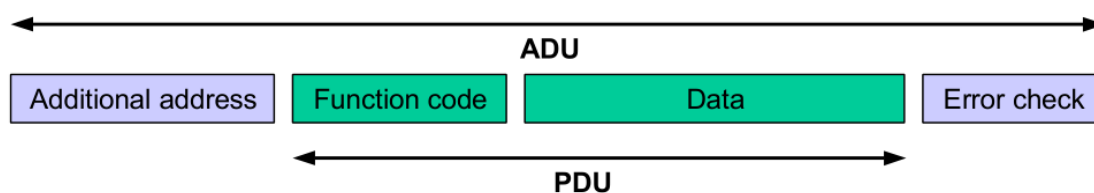
**A MODBUS Response** is the Response message sent by the Server,

**A MODBUS Confirmation** is the Response Message received on the Client side

KUVA 8. Modbus TCP/IP -kommunikaatiomalli (Modbus Organization 2006, 2)

### 3.5.3 Modbus-viestin rakenne

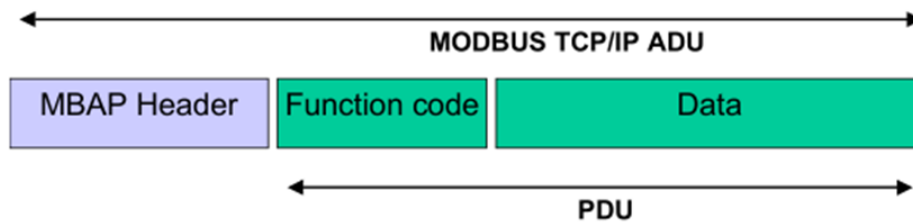
Modbus määrittelee kaikille sen toteutuksille ja tiedonsiirtotavoille yhteisen viestikehyksen Modbus PDU (Protocol Data Unit). Toteutuksen mukaan PDU:n yhteyteen voidaan lisätä erilaisia tietokenttiä (esim. laitetunniste tai virheentarkistussumma). Modbus PDU:n ja toteutuksesta riippuvaisten tietokenttien yhdistelmästä käytetään nimitystä Modbus ADU (Application Data Unit). (Kuva 9.) (Modbus Organization 2012, 2–3.)



KUVA 9. Modbus-viestin yleinen rakenne (Modbus Organization 2012, 3)

Modbus TCP/IP -toteutuksessa Modbus ADU koostuu Modbus PDU:n lisäksi sen eteen tulevasta MBAP (Modbus Application Protocol header) otsikkokehyksestä (kuva 10).





KUVA 10. Modbus TCP/IP -viestin rakenne (Modbus Organization 2006, 4)

MBAP-otsikkokehys on seitsemän tavun mittainen. Sen sisältämät kentät ovat Transaction Identifier, Protocol Identifier, Length ja Unit Identifier (kuva 11). Master-laite voi käyttää Transaction Identifier -kenttää samanaikaisten Modbus-transaktioiden erottamiseen toisistaan. Protocol Identifier -kenttä nimensä mukaisesti identifioi Modbus-protokollan ja sen tulee aina olla arvoltaan 0. Length-kenttä kertoo ADU:ssa sitä itseään seuraavien tavujen lukumäärän. Length-kentän arvo koostuu siis Unit Identifier -kentän sekä PDU:n yhdistetystä tavumäärästä. Unit Identifier -kentän tarkoitus on identifioida vastaanottava slave-laite, mikäli monta laitetta on saman IP-osoitteen takana esim. yhdyskäytävälaitteeseen kytkettynä. (Modbus Organization 2006, 5.)

Fields	Length	Description -	Client	Server
Transaction Identifier	2 Bytes	Identification of a MODBUS Request / Response transaction.	Initialized by the client	Recopied by the server from the received request
Protocol Identifier	2 Bytes	0 = MODBUS protocol	Initialized by the client	Recopied by the server from the received request
Length	2 Bytes	Number of following bytes	Initialized by the client ( request)	Initialized by the server ( Response)
Unit Identifier	1 Byte	Identification of a remote slave connected on a serial line or on other buses.	Initialized by the client	Recopied by the server from the received request

KUVA 11. MBAP-otsikkokehysten rakenne (Modbus Organization 2006, 5)

PDU:n pituus, joka periytyy alkuperäisestä Modbus-toteutuksesta, on enintään 253 tavua. PDU sisältää aina yhden tavun mittaisen funktiokoodin, mutta sitä ei välttämättä seuraa yhtään tavua dataa. Näin ollen Modbus TCP/IP ADU:n pituus tavuina sijoittuu aina välille

$$8 \leq (ADU = MBAP + PDU) \leq 260.$$

Master-laitteen lähettämässä kyselyssä funktiokoodi määrittelee toiminnon, jonka vastaanottavan slave-laitteen tulee pyrkiä toteuttamaan. Slave-laitteen lähettämän vastauksen funktiokoodi on virhetilanteita lukuunottamatta aina sama kuin sen vastaanottamassa kyselyssä. (Modbus Organization 2012, 4–5.)

PDU:ssa funktiokoodin lisäksi oleva data on funktiokoodista riippuvaista – esimerkiksi master-laitteen lähettämän funktiokoodin ollessa 3 (Read Holding Registers) datakentässä oleva rekisteriosoite ja rekisterien lukumäärä viittaavat Holding Register tietotyypin osoiteavaruuteen (kuva 12). Vastaavasti slave-laitteen lähettäessä vastauksen samalla funktiokoodilla ovat datakentän rekisteriarvot Holding Register tyyppisiä. (Modbus Organization 2012, 4–5.)

#### Request

Function code	1 Byte	0x03
Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity of Registers	2 Bytes	1 to 125 (0x7D)

KUVA 12. Request PDU:n rakenne, funktiokoodi Read Holding Registers (Modbus Organization 2012, 15)

### 3.5.4 Tiedon mallinnus

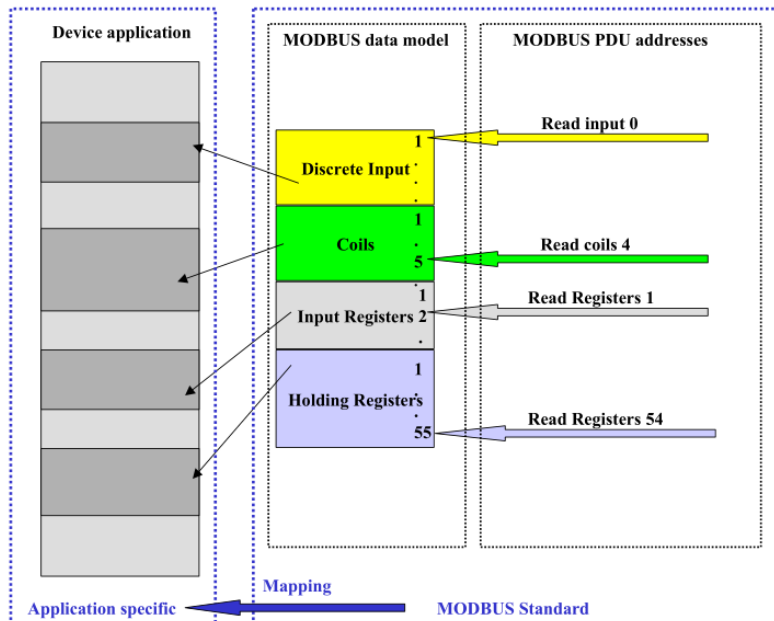
Modbus määrittelee neljä erilaista tietotyyppiä: Coil, Discrete Input, Holding Register ja Input Register. Coil ja Discrete Input ovat yhden bitin kokoisia, kun taas Holding Register ja Input Register ovat 16-bittisiä. Jokaiseen tietotyyppiin liittyvät myös tietynlaiset luku- ja kirjoitusoikeudet: Discrete Input- ja Input Register -tyypistä tietoa voidaan ainoastaan lukea, kun taas Coil- ja Holding Register -tyypistä tietoa voidaan lukemisen lisäksi muuttaa kirjoittamalla. (Kuva 13.)

Primary tables	Object type	Type of	Comments
Discretes Input	Single bit	Read-Only	This type of data can be provided by an I/O system.
Coils	Single bit	Read-Write	This type of data can be alterable by an application program.
Input Registers	16-bit word	Read-Only	This type of data can be provided by an I/O system
Holding Registers	16-bit word	Read-Write	This type of data can be alterable by an application program.

KUVA 13. Modbus-tietotyypit (Modbus Organization 2012, 6)

Jokaisella Modbus-tietotyypillä on oma 16-bittinen osoiteavaruutensa (0x0001–0xFFFF). Osoitteilla ei ole välttämättä mitään tekemistä tiedon fyysisen sijainnin kanssa. Sen sijaan esim. 32-

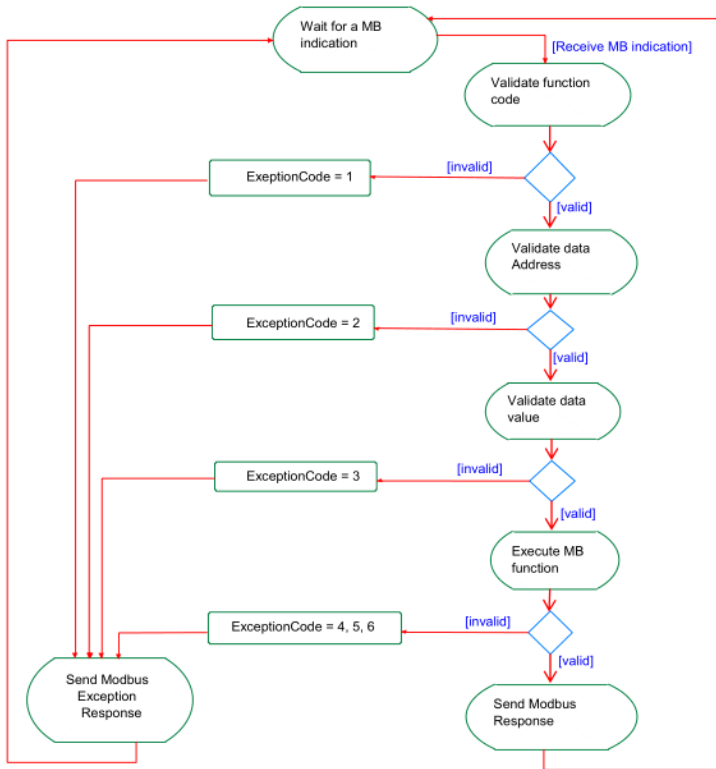
bittisessä laitemuistissa oleva tieto voidaan joutua assosioimaan jollain tapaa käytetyn Modbus-tietotyyppin osoitevaruuden kanssa. Tietomallista poiketen PDU:n sisältämien rekisteriosoitearvojen indeksointi alkaa nollassa, jolloin arvot ovat väliltä 0x0000–0xFFFE. (Kuva 14.) (Modbus Organization 2012, 6.)



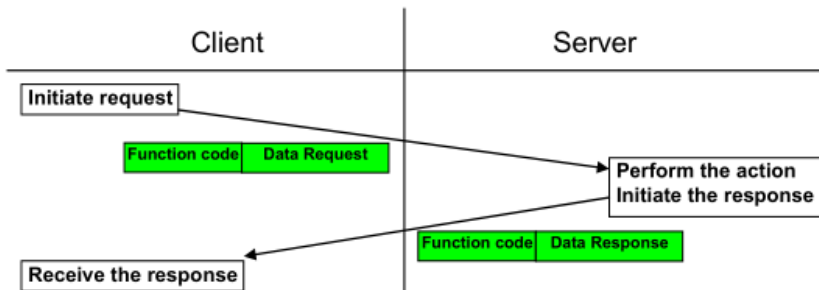
KUVA 14. Modbus-osoitemalli (Modbus Organization 2012, 8)

### 3.5.5 Modbus-transaktio

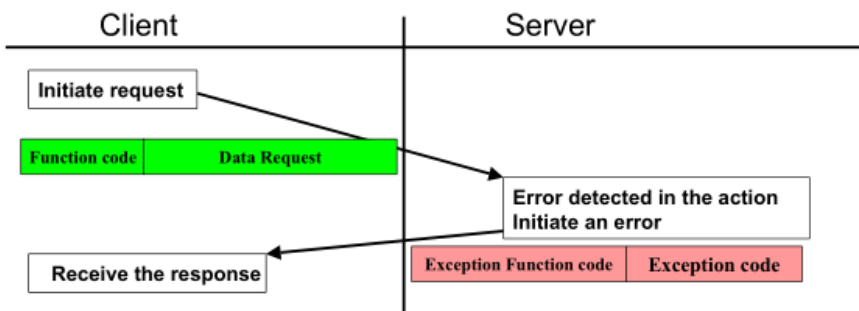
Jokainen Modbus-transaktio alkaa master-laitteen slave-laitteelle lähettämästä kyselystä. Slave-laite, jolle kysely on osoitettu, aloittaa verifioimalla kyselyn sisällön. Mikäli kysely sisältämä funktiokoodi, rekisteriosoite sekä rekisterien lukumäärä ovat standardinmukaisia, slave-laite jatkaa suorittamalla funktiokoodin ja rekisteriarvojen mukaisen toiminnon. Jos slave-laite onnistuu toiminnon suorittamisessa, se lähettää master-laitteelle vastauksen, joka sisältää alkuperäisen funktiokoodin sekä pyydyt rekisteriarvot. Epäonnistuttuaan joko kyselyn verifiointissa tai toiminnon suorittamisessa slave-laite vastaa master-laitteelle poikkeusviestillä, joka sisältää alkuperäisestä funktiokoodista muodostetun poikkeusfunktiokoodin ja datakentässä olevan poikkeuskoodin. Poikkeusfunktiokoodi muodostetaan summaamalla alkuperäinen funktiokoodi heksadesimaalisen lukuarvon 0x80 kanssa. Datakentän poikkeuskoodin arvo määräytyy sen mukaan, missä vaiheessa transaktiota virhe tapahtuu. (Kuvat 15–17.)



KUVA 15. Modbus-transaktion tilakaavio slave-laitteen näkökulmasta (Modbus Organization 2012, 9)



KUVA 16. Onnistunut Modbus-transaktio (Modbus Organization 2012, 4)



KUVA 17. Epäonnistunut Modbus-transaktio (Modbus Organization 2012, 4)

## 3.6 PROFIBUS-kenttäväylä

### 3.6.1 Tausta

PROFIBUS on Saksassa usean eri yrityksen ja instituution yhteistyössä 1980-luvun lopulla kehitetty avoin kenttäväylästandardi. Vastuu PROFIBUS-standardin ylläpitämisestä ja kehittämisestä siirtyi vuonna 1989 saksalaiselle PROFIBUS Nutzerorganisation e.V (PNO) -järjestölle, josta tuli vuonna 1995 osa tällöin perustettua kansainvälistä PROFIBUS and PROFINET International (PI) -järjestöä. PROFIBUS-laitteita on maailmalla käytössä yli 50 miljoonaa kappaletta, mikä tekee PROFIBUS-kenttäväylästä alan markkinajohtajan. PROFIBUS on osa IEC:n julkaisemia kenttäväylästandardeja IEC 61158 ja IEC 61784. (Profibus Nutzerorganisation e.V. 2016, 3–4, 7.)

PROFIBUS kattaa OSI-viitemallin kerrokset 1, 2 ja 7 (kuva 18). Erilaiset versiot PROFIBUS-tekniologiasta perustuvat sovellusprofiileihin, jotka jäävät OSI-viitemallin ulkopuolelle. PROFIBUS tukee erilaisia Layer 1 -tiedonsiirtoteknologioita kuten RS-485, MBP (Manchester Bus Powered) ja valokuitu. (Profibus Nutzerorganisation e.V. 2016, 3.)

	User program	Application profiles
7	Application Layer	PROFIBUS DP Protocol (DP-V0, DP-V1, DP-V2)
6	Presentation Layer	Not used
5	Session Layer	
4	Transport Layer	
3	Network Layer	
2	Data link Layer	Fieldbus Data Link (FDL): Master Slave principle Token principle
1	Physical Layer	Transmission technology
	OSI Layer Model	OSI implementation at PROFIBUS

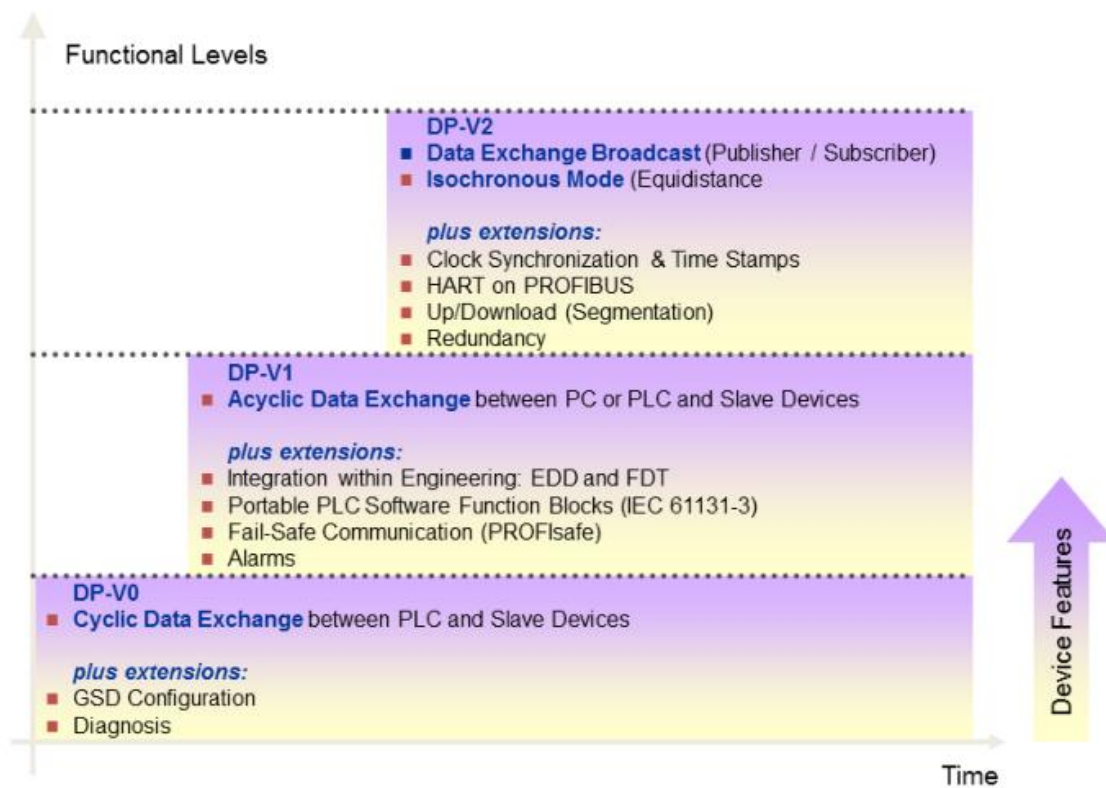
KUVA 18. PROFIBUS OSI-viitemallissa (Profibus Nutzerorganisation e.V. 2016, 3)

### 3.6.2 Toimintaperiaate

PROFIBUS toimii master/slave-periaatteen mukaisesti. Slave-laitteet ovat passiivisia ja ainoastaan vastaavat väylällä olevien aktiivisten master-laitteiden lähettämiin pyyntöihin. Mikäli väylällä on monta master-laitetta, oikeus väylän käyttöön perustuu token passing -periaatteen mukaisesti master-laitteiden välillä kiertävän "tokenin" hallussapitoon. (Profibus Nutzerorganisation e.V. 2016, 10.)

### 3.6.3 PROFIBUS DP

PROFIBUS DP (Decentralized Periphery) on kaikille PROFIBUS-sovelluksille yhteinen kommunikaatioprotokolla. PROFIBUS DP:stä on kolme eri versiota: DP-V0, DP-V1 sekä DP-V2. Jokainen uusi versio lisää ominaisuuksia edellisen version määrittelemien ominaisuuksien päälle. DP-V0 on yksinkertaisin ja vanhin versio protokollasta, joka mahdollistaa syklisen tiedonsiirron master- ja slave-laitteiden välillä. DP-V1 tarjoaa DP-V0 ominaisuuksien lisäksi mahdollisuuden asykliseen tiedonsiirtoon. DP-V2 taas mahdollistaa esimerkiksi slave-laitteiden välisen kommunikaation. (Kuva 19.) (Profibus Nutzerorganisation e.V. 2016, 10–11.)



KUVA 19. PROFIBUS DP -protokollan versiot (Profibus Nutzerorganisation e.V. 2016, 10)

### 3.6.4 Syklinen tiedonsiirto

Syklisessä tiedonsiirrossa master-laite pyytää prosessidataa tietyin määräajoin väylään kytketyiltä slave-laitteilta MS0-kanavaa pitkin. Mikäli väylällä on monta master-laitetta, voi jokainen niistä lukea dataa kaikilta väylällä olevilta slave-laitteilta. Ainoastaan yksi master-laite, jolle slave-laite on konfiguroitu, voi kirjoittaa dataa kyseiseen slave-laitteeseen. (Jaganmohan Reddy & Mehta 2014, luku 12.3; Profibus Nutzerorganisation e.V. 2016, 11.)

Master-laite muodostaa laitetietokannan väylään kytketyistä slave-laitteista GSD-laitekuvaustiedostojen avulla, joissa on määritelty mm. slave-laitteilta saatavan prosessidatan määrä. Laitetietokanta sisältää jokaisen slave-laitteen alustamiseen käytetyn parametri- ja konfiguraatiodatan, sekä muut syklisen tiedonsiirron edellyttämät tiedot. (Jaganmohan Reddy & Mehta 2014, luku 12.3.)

Master-laite aloittaa syklisen tiedonsiirron lähettämällä jokaiselle laitetietokantaan konfiguroidulle slave-laitteelle parametri- ja konfiguraatioviestin. Slave-laitteet tarkistavat vastaanottamansa konfiguraatiodatan vertaamalla sitä niiden omissa muisteissa oleviin konfiguraatioihin. Master-laite pyytää parametri- ja konfiguraatioviestien jälkeen slave-laitteelta diagnostiikkaviestin, jonka perusteella se voi päätellä, onnistuiko laitteen konfigurointi. Onnistuneen konfiguroinnin jälkeen master-laite aloittaa syklisen prosessidatan siirron slave-laitteiden kanssa. (Jaganmohan Reddy & Mehta 2014, luku 12.3; Profibus Nutzerorganisation e.V. 2016, 11–12.)

### **3.6.5 Asyklinen tiedonsiirto**

Master-laitteet käyttävät asyklistä tiedonsiirtoa slave-laitteiden laiteparametrien lukemiseen ja kirjoittamiseen. Asyklinen tiedonsiirto toimii kahdella eri kommunikaatiokanavalla: MS1 ja MS2. MS1 on kommunikaatiokanava slave-laitteen ja sen master-laitteen välillä, jonka kanssa slave-laite kommunikoi syklisesti. MS1-kanavaa käytetään syklisen tiedonsiirron alustamiseen tarvittavan parametrin lähettämiseen ja se avautuu laitteiden välille automaattisesti. MS2-kanavaa pitkin slave-laite voi viestiä usean master-laitteen kanssa. Master-laitteen tulee manuaalisesti avata MS2-kanava slave-laitteen kanssa ja se sulkeutuu itsestään, mikäli sitä ei käytetä aktiivisesti. MS2-kanavan tiedonsiirto tapahtuu syklisen tiedonsiirron syklien “välissä”, eli syklin viimeiselle slave-laitteelle lähetetyn kyselyn jälkeen ennen seuraavan syklin aloittamista. (Jaganmohan Reddy & Mehta 2014, luku 12.3; Profibus Nutzerorganisation e.V. 2016, 12.)

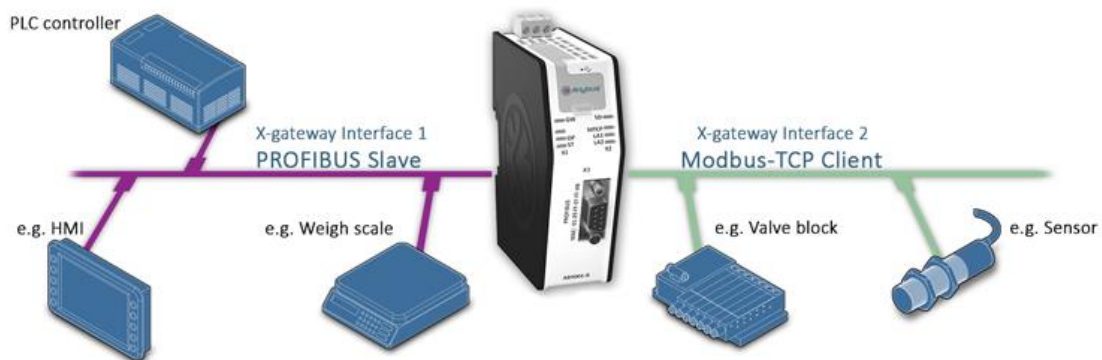
## 4 KÄYTETYT LAITTEET

### 4.1 HMS Anybus X-Gateway AB9001 PROFIBUS – Modbus TCP/IP -yhdyskäytävä

#### 4.1.1 Käyttötarkoitus

Anybus X-Gateway on ruotsalaisen automaatiovalmistaja HMS Industrial Networksin tuoteperhe, johon kuuluvilla yhdyskäytävälaitteilla voidaan yhdistää eri kenttäväyliä sekä niihin kytkettyjä laitteita toisiinsa. AB9001-yhdyskäytävä mahdollistaa Modbus TCP/IP -laitteiden kytkemisen PROFIBUS-järjestelmään. Se toimii PROFIBUS-järjestelmässä slave-laitteena ja Modbus TCP/IP -järjestelmässä master-laitteena. (Kuva 19.) (HMS Industrial Networks 2019.)

Yhdyskäytävään voidaan kytkeä jopa 64 Modbus TCP/IP -laitetta, joille voidaan yhteensä konfiguroida enintään 64 Modbus TCP/IP -transaktiota. Kytketyt slave-laitteet voivat siirtää yhdyskäytävän kanssa yhteensä 368 tavun verran tietoa, josta yhteen suuntaan voi liikkua enintään 244 tavua. (HMS Industrial Networks 2012.)



KUVA 19. Anybus X-Gateway AB9001 sovellusesimerkki (HMS Industrial Networks 2019)

#### 4.1.2 Konfiguraatio

Yhdyskäytävälaitteen konfigurointi tapahtuu laitteessa olevan HTTP-palvelimen web-käyttöliittymän kautta (kuva 20). Käyttöliittymän kautta voidaan määritellä yhdyskäytävään kytketyt Modbus TCP/IP -slave-laitteet ja niiden Modbus TCP/IP -transaktiot (kuva 21). Myös joitain PROFIBUS-



slave-laitteen ja Modbus TCP/IP -master-laitteen ominaisuuksia voidaan määrittellä, kuten esim. PROFIBUS-laitteen tunnistenumero ja Modbus TCP/IP -laitteen käyttäytyminen vikatilanteessa.

**Anybus X-gateway Modbus-TCP - PROFIBUS DP-V1**

**OVERVIEW**  
**Home**  
**CONFIGURATION**  
 Authentication  
**Modbus Client**  
 Modbus Servers  
**PROFIBUS DP-V1**  
**TOOLS**  
 X-gateway Management  
 Backup & Restore  
 Mapping Overview  
 Transaction Monitor

**Anybus X-gateway configuration and status web pages.** Welcome to the configuration interface of the Anybus X-gateway. Use the left side menu to navigate. Changes to the configuration do not take effect until the X-gateway is restarted from the X-gateway Management page. 'Network 1' represents the controlling network, where the X-gateway acts as a server. 'Network 2' represents the controlled network, where the X-gateway acts as a client.

Identification		Ethernet link status	
Product name:	Anybus X-gateway Modbus-TCP	Port 1:	
Firmware version:	1.13	Speed:	100 Mbps
Serial number:		Duplex:	Full Duplex
Uptime:	0 days, 5h:35m:31s	Port 2:	
CPU Load:	8% (auto updated every 5s)	Speed:	-
MAC ID (Modbus-TCP):		Duplex:	-

Operation Mode		Ethernet link statistics	
PROFIBUS DP-V1 (Network 1):	I/O data exchanged	In pkts:	292327 Errors: 1
Modbus-TCP (Network 2):	Run	Out pkts:	67041 Errors: 0

© 2011 HMS Industrial Networks - All rights reserved Support Connecting Devices™

KUVA 20. Kuvakaappaus yhdyskäytävälaitteen web-käyttöliittymästä, aloitussivu (HMS Industrial Networks 2019)

**Anybus X-gateway Modbus-TCP - PROFIBUS DP-V1**

**OVERVIEW**  
**Home**  
**CONFIGURATION**  
 Authentication  
**Modbus Client**  
 Modbus Servers  
**PROFIBUS DP-V1**  
**TOOLS**  
 X-gateway Management  
 Backup & Restore  
 Mapping Overview  
 Transaction Monitor

**Transactions configuration.** Add, edit or delete transactions used for a connections on this page. On each connection several transactions towards a Modbus server can be set up. Press 'Add transaction' button to add a new transaction, then edit it to set transaction properties. The global limit is 64 transaction and it is not possible to map more process data than the gateway can handle.

**Global configuration limits.**  
 Transactions: 2/64 | Minimum allowed scan time: 10 | I/O mapped input data: 40/244 bytes | I/O mapped output data: 0/244 bytes | I/O mapped input bits: 0/1024 | I/O mapped output bits: 0/1024 | Input data: 32/256 bytes | Output data: 0/256 bytes | Total I/O mapped data: 40/368 bytes |

Name	IP address	Port	Protocol
M7	192.168.200.25	502	TCP

#	Function	Encoding	Scan time	Timeout	UID	Address / Bit	Data Type	Elements	Registers	Action on no Network1 I/O	
S7_0	3	BBEWBE	1000	5000	1	1	uint16	8	8	N/A	Edit Delete
S7_1	3	BBEWBE	1000	5000	2	9	uint16	8	8	N/A	Edit Delete

Hovering mouse over an element where the cursor shows a question mark displays help.

[Back to server list](#) [Add new transaction](#)

Add/edit transaction

General transaction settings			
Function code	3-Read Holding Registers	Name	S7_0
Data encoding	Byte Big Endian, Word Big Endian	Timeout (ms)	5000
Trigger	Cyclic	Scan time (ms)	1000
I/O mapped	Yes	Unit Id	1
Read settings			
Starting register	1	Data type	uint16
Elements	8	Registers	8

[Cancel](#) [Ok](#)

© 2011 HMS Industrial Networks - All rights reserved Support Connecting Devices™

KUVA 21. Kuvakaappaus yhdyskäytävälaitteen web-käyttöliittymästä, Modbus TCP/IP -transaktion lisääminen (HMS Industrial Networks 2019)

## 4.2 Kunbus DF PROFI II -PROFIBUS-laitekortti

### 4.2.1 Käyttötarkoitus

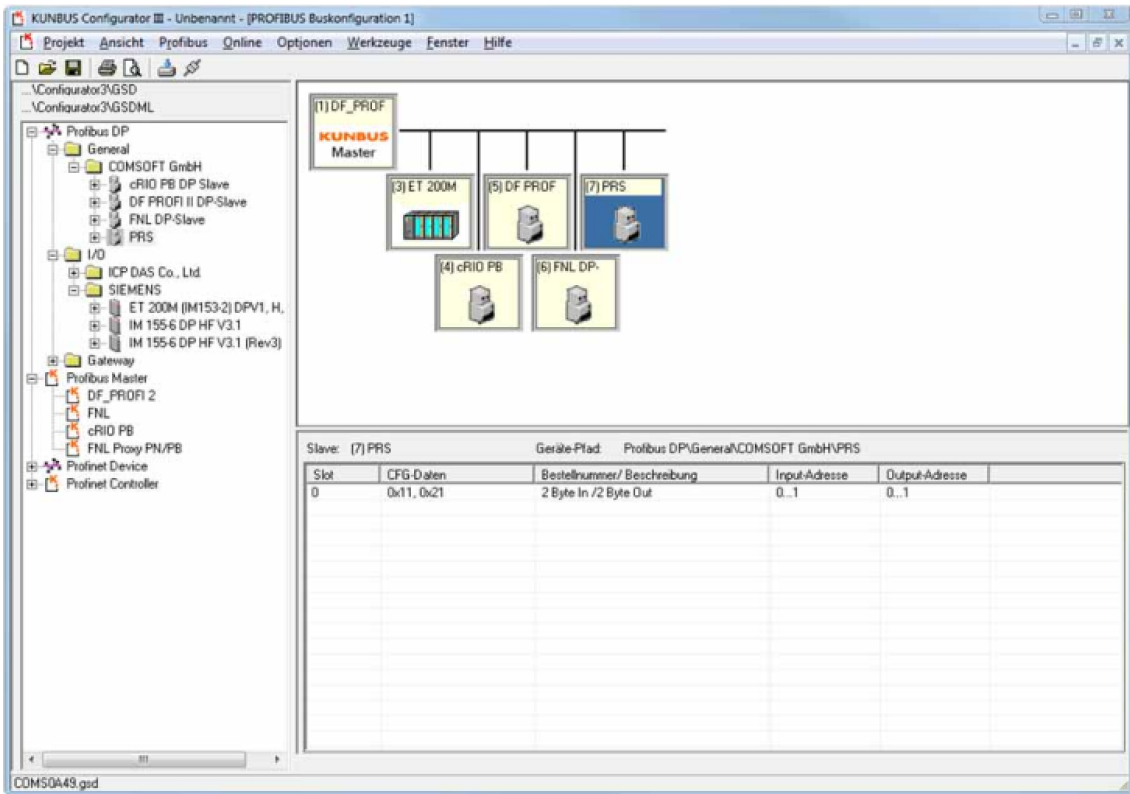
DF PROFI II on saksalaisen Kunbus GmbH:n (ent. Comsoft) valmistama, tietokoneeseen yhdistettävä PROFIBUS-laitekortti (kuva 22). Käytetty DF PROFI II yhdistettiin tietokoneeseen PCIe-väylän kautta, mutta kortista on saatavilla myös PCI ja CPCI yhteensopivat versiot. DF PROFI II -korttia käytettiin PROFIBUS-master-laitteena yhdessä laitteen mukana tulleen Configurator III -ohjelman kanssa järjestelmän testaamista varten.



KUVA 22. Kunbus DF PROFI II -laitekortti (Kunbus GmbH 2017, 1)

### 4.2.2 Konfiguraatio

DF PROFI II konfiguroitiin käyttäen laitteen mukana tullutta Configurator III -ohjelmaa (kuva 23). Anybus X-Gateway AB9001 -yhdyskäytävälaitteen GSD-laitekuvaustiedosto ladattiin ohjelmaan, jonka jälkeen muodostettiin itse PROFIBUS-väyläkonfiguraatio. Konfiguraatio muodostettiin valitsemalla halutut I/O-moduulit ohjelman GSD-tiedoston perusteella luomasta listasta, jonka jälkeen valitut moduulit yhdistettiin master-laitteeseen. Lopuksi muodostettu väyläkonfiguraatio ladattiin laitekortille, jonka jälkeen se oli valmis käytettäväksi tiedonsiirtoon yhdyskäytävän kanssa.



KUVA 23. Kunbus Configurator III -ohjelma (Kunbus GmbH 2017, 8)

## 5 TOTEUTUS

### 5.1 Tausta

C-Site-järjestelmään kuuluvan M7-palvelinohjelmiston liitettävyyden PROFIBUS-kenttäväylään päätettiin toteuttaa käyttämällä Modbus TCP/IP -protokollaa sekä yhdyskäytävälaitetta, joka muuntaa ja siirtää automaattisesti tietoa Modbus TCP/IP- ja PROFIBUS-verkkojen välillä (liite 1). Modbus TCP/IP:n valinta toteutettavaksi protokollapinoksi PROFIBUSin sijaan perustui Modbusin yksinkertaisuuteen, Ethernetin käyttämiseen fyysisenä siirtotienä sekä dokumentaation saatavuuteen.

Koko Modbus TCP/IP -spesifikaatio on 50 sivun mittainen ja saatavissa ilmaiseksi Modbus Organizationin nettisivuilta, kun taas PROFIBUS-standardin Layer 7 -määrittelydokumentit ovat yhteensä noin 1600 sivua ja 600 euroa IEC:n verkkokaupasta ostettuna. Modbus TCP/IP:n fyysisenä siirtotienä käyttämä Ethernet mahdollistaa paikallisesti asennetun palvelintietokoneen kytkemisen yhdyskäytävälaitteeseen normaalia RJ-45-verkkokaapelia käyttäen sekä useamman yhdyskäytävän samanaikaisen kytkennän palvelintietokoneeseen tavallisen Ethernet-kytkimen avulla.

Huomioonotettavaa on myös se, että yhdyskäytävälaitteet mahdollistavat palvelinohjelmiston liittämisen moneen eri kenttäväylään samalla ohjelmatoteutuksella, mikäli vastaava Modbus TCP/IP -master-laitteena toimiva yhdyskäytävälaitte on saatavissa halutulla kenttäväyläliitännällä. Mikäli liitettävyyden olisi toteutettu suoraan esim. PCIe-väylään kytkettävällä PROFIBUS-laitekortilla, ei siihen kuuluvaa ohjelmistototeutusta olisi voitu käyttää helposti (jos ollenkaan) muiden kenttäväyläjärjestelmien kanssa.

### 5.2 Ohjelmointi

Ohjelmisto toteutettiin käyttäen Java SE 8 -ohjelmointikieltä (liite 2). Javan valinta toteutuksessa käytettäväksi kieleksi perustui siihen, että C-Site-järjestelmän palvelinohjelmisto oli toteutettu suurimmaksi osaksi Javalla. Työn alkuvaiheessa ei vielä ollut tiedossa, miten toteutettava ohjelmisto tulisi integroimaan palvelinohjelmiston kanssa, minkä takia Java oli luonnollisin valinta käytettäväksi ohjelmointikieleksi.

Ohjelmistoon kuuluvien loogisten kokonaisuuksien välinen kommunikaatio toteutettiin tarpeen selaista vaatiessa luokkarajapintojen avulla. Luokkarajapinnoissa voidaan määrittellä, millaisia funktioita rajapinnan toteuttavan luokan tulee tarjota, mutta yleensä itse rajapinta ei sisällä minkäänlaista toteutusta sen määrittelemälle funktiolle. Luokkien toteutuksen ja niiden välisen kommunikaation erottaminen rajapintoja käyttämällä mahdollistaa luokkien uudelleenkirjoittamisen tai uusien luokkien käyttämisen tulevaisuudessa ilman huolta yhteensopivuuden rikkomisesta.

### **5.3 Modbus TCP/IP -protokollapino**

Modbus TCP/IP -protokollapinoon toteutettiin TCP/IP-yhteyksien hallinta sekä Modbus TCP/IP -slave-laitteen toiminnallisuus. Protokollapinossa päätettiin tukea ainoastaan tarpeelliseksi katsottua osaa käytetyn yhdyskäytävälaitteen tukemista funktiokoodista.

#### **5.3.1 TCP/IP-yhteydet**

Yhteyksien hallinta ja prosessointi toteutettiin säikeistetyksi. Yhteyksien hallinta käytti yhtä säiettä uusien yhteyksien jatkuvaan kuuntelemiseen. Uudet avatut yhteydet siirrettiin omiin säikeisiinsä niiden autonomista prosessointia varten.

Avatut yhteydet pyrkivät lukemaan ja verifioimaan saapuvia Modbus TCP/IP -viestejä. Mikäli vastaanotetussa viestissä havaitaan virhe tai uutta viestiä ei vastaanoteta tietyn ajan kuluessa, yhteys sulkeutuu. Yhteyden sulkeutumisen myötä sen käyttämä säie vapautuu uusien avattujen yhteyksien käytettäväksi.

Yhteys tarkistaa lukemastaan viestistä ensimmäiseksi Protocol Identifier-, Length-, Unit Id- ja Function Code -kenttien standardinmukaisuuden. Mikäli kyseisten kenttien tarkistus onnistuu, yhteys vertaa lopuksi Length-kentän lukuarvoa ja datakenttään luetun datan tavumäärää toisiinsa varmistakseen näiden yhdenmukaisuuden.

Yhteys lähettää hyväksymänsä Modbus TCP/IP -viestit eteenpäin slave-laitteen toiminnallisuudesta vastaavaan osaan ohjelmaa, joka rakentaa ja palauttaa sille vastausviestin. Yhteys lähettää vastauksen alkuperäisen viestin lähettäjälle, jonka jälkeen se jää odottelemaan seuraavaa viestiä protokollaspesifikaation mukaisesti.

### 5.3.2 Slave-laite

Slave-laite käsittelee TCP/IP-yhteyksiltä vastaanottamiaan viestejä. Slave-laite toteutettiin ilman säikeistystä siten, että kaikki rinnakkaisesti prosessoitavat yhteydet lähettävät kyselyjä samalle slave-laitteen instanssille. Tämä jaettu instanssi käsittelee aina yhden kyselyn kerrallaan muiden sinä aikana vastaanotettujen kyselyjen odottaessa omaa vuoroaan.

Slave-laite tarkastaa ensimmäiseksi protokollapinon tuen viestin sisältämälle funktiokoodille. Jos funktiokoodi on tuettu, slave-laite tarkastaa seuraavaksi viestin datakentästä löytyvät arvot. Nämä arvot ovat funktiokoodista riippuvaisia ja kuvaavat yleensä rekisteriosoitetta ja rekisterien lukumäärää, joista viestin lähettäjä haluaa lukea dataa.

Jos rekisterien lukumäärä on spesifikaation mukainen, rekisteriosoitteen ja rekisterien lukumäärän muodostama osoiteavaruus verifioidaan lähettämällä tarpeelliset viestistä poimitut arvot M7-rajapinnalle. Jos osoiteavaruuden verifiointi onnistuu, slave-laite pyytää M7-rajapinnalta sitä vastaavan datan ja sisällyttää sen palauttamansa vastausviestin datakenttään.

Mikäli viestin tarkastamisen tai datan hakeminen epäonnistuu, slave-laite muodostaa ja palauttaa yhteydelle epäonnistumisen vaiheesta riippuvaisen virheviestin. Virheviesti sisältää alkuperäisen viestin funktiokoodista muodostetun poikkeusfunktiokoodin sekä virhettä kuvaavaan poikkeuskoodin viestin datakentässä.

## 5.4 M7-rajapinta

M7-rajapinnan toteutus perustui HTTP-kyselyjen lähettämiseen M7-palvelinohjelmiston tarjoamalle REST-rajapinnalle. REST-rajapinnalta saatu data tallennettiin Modbus-välimuistiin, josta slave-laite pystyi M7-rajapinnan tarjoamien funktioiden avulla verifioimaan osoiteavaruuksia ja hakemaan niille assosioitua dataa.

### 5.4.1 REST-rajapinta

Data haettiin M7-palvelinohjelmistolta sen tarjoaman REST-rajapinnan kautta. REST-rajapinnalta saatu JSON-tyyppinen data muunnettiin luokkainstansseiksi FasterXML:n Jackson-kirjaston avulla.

REST-rajapinnan lähettämän JSON-datan rakenne määritteli muunnoksiin käytettävien luokkien rakenteen.

REST-rajapinnalle lähetettävät kyselyt konfiguroitiin JSON-tyyppisen tiedoston avulla ja kyselyjä suoritettiin tietyin väliajoin omassa säikeessään. Kyselyn päätteeksi Modbus-välimuistissa oleva data korvattiin rajapinnalta haetulla uudemmalla datalla.

#### **5.4.2 Modbus-välimuisti**

Palvelinohjelmistolta haetun datan tallentamiseen käytettiin välimuistia, josta dataa voitiin hakea Modbus TCP/IP -kyselystä poimittujen tietojen perusteella. Välimuisti on tyypiltään avain-arvo paria sisältävä assosiaatiotaulu, johon tallennettua dataa eli arvoja voidaan hakea avaimien avulla.

Taulun avaimien tietotyyppinä käytettiin luokkaa, jonka sisältämät kaksi kenttää vastaavat Modbus TCP/IP -viestin datakentästä löytyvää rekisteriosoitetta, sekä funktiokoodiin liittyvää Modbus-data-mallin mukaista tietotyyppiä (Coil, Discrete Input, Holding Register tai Input Register).

Arvojen tietotyyppinä käytettiin geneeristä luokkarajapintaa, jonka määrittelemän funktion avulla voidaan hakea toteuttavan luokan sisältämä rekisteridata tavujonoksi muunnettuna. Mikäli dataa pyydetään monesta rekisteriosoitteesta kerrallaan, rajapinta yhdistää kaikki haetuilta arvoilta saadut tavujonot ennen datan palauttamista.

#### **5.5 Yhdyskäytävän konfiguraatio**

Yhdyskäytävälaitteen konfiguraatio perustui laitteen konfiguroimiseen normaalisti käytettävän web-käyttöliittymän lähettämien HTTP-pyyntöjen ohjelmalliseen jäljittelyyn. Halutut yhdyskäytävälaitteen asetukset konfiguroitiin JSON-tyyppisen tiedoston avulla. Laitteen HTTP-palvelimen toiminnasta ei ollut saatavilla dokumentaatiota, joten sen käyttämät HTTP-pyyntöt jouduttiin selvittämään itse.

Selvitys tapahtui käyttämällä HTTP-palvelimen normaalia web-käyttöliittymää, jonka aikana web-käyttöliittymän lähettämät HTTP-pyyntöt kaapattiin Firefox-verkkoselaimen kehittäjätyökalujen

avulla (kuva 24). Ohjelmallisen toteutuksen lähettämät vastaavat HTTP-pyynnöt verifioitiin Wireshark-paketinkaappausohjelmaa käyttämällä (kuva 25).

```
▼ Form data
  function: edit
  nodeid: 0
  transactionid: 0
  alias: New_Trans1
  mbfunction: 3
  encoding: 1
  clienterroraction: 0
  delay: 250
  timeout: 5000
  unitid: 255
  transactiontrigger: 0
  pdmapped: 1
  startupmode: 1
  raddr: 0
  rtype: 0
  relements: 1
  safeelementvalue: 0
```

KUVA 25: Kuvakaappaus Firefoxin kehittäjätyökaluilla kaapatun HTTP-pyyntöä datasta (Mozilla 2019)

```
[Full request URI: http://192.168.200.35/conf/node.html]
[HTTP request 1/1]
[Response in frame: 631]
File Data: 218 bytes
▼ HTML Form URL Encoded: application/x-www-form-urlencoded
  > Form item: "function" = "edit"
  > Form item: "nodeid" = "0"
  > Form item: "transactionid" = "1"
  > Form item: "alias" = "S7_1"
  > Form item: "mbfunction" = "3"
  > Form item: "encoding" = "0"
  > Form item: "clienterroraction" = "0"
  > Form item: "delay" = "1000"
  > Form item: "timeout" = "5000"
  > Form item: "unitid" = "2"
  > Form item: "transactiontrigger" = "0"
  > Form item: "pdmapped" = "1"
  > Form item: "startupmode" = "1"
  > Form item: "raddr" = "8"
  > Form item: "rtype" = "0"
  > Form item: "relements" = "8"
  > Form item: "safeelementvalue" = "0"
```

KUVA 26: Kuvakaappaus Wireshark-työkalulla kaapatun, ohjelmallisesti tuotetun HTTP-pyyntöä datasta (Wireshark Foundation 2019)



## 6 POHDINTA

Opinnäytetyön tarkoituksena oli selvittää, miten Creowave C-Site -etämonitorointijärjestelmä voitaisiin liittää PROFIBUS-kenttäväylään, sekä toteuttaa selvityksen perusteella valitut ratkaisut. Työn tuloksena syntyneellä yhdistetyllä laite- ja ohjelmistoratkaisulla saatiin testiympäristössä siirrettyä onnistuneesti tietoa C-Site M7 -palvelinohjelmiston ja PROFIBUS-master-laitteen välillä.

Työn aihe oli haastava. Varsinkin työn alkuvaiheessa, kun en vielä tiennyt yhtään mitään kenttäväylistä tai C-Site-järjestelmän toiminnasta, en ollut erityisen itsevarma toimivan ratkaisun löytämisestä ja toteuttamisesta suunnitellussa aikataulussa. Työhön liittyi paljon niin suuria kokonaisuuksia kuin pieniä yksityiskohtia, joita tuli ottaa huomioon suunnittelun ja toteutuksen yhteydessä.

Käytetyn yhdyskäytävälaitteen konfiguraatio ohjelmallisesti on varsin kömpelöä, sillä laite on suunniteltu konfiguroitavaksi ainoastaan verkkoselaimen avulla. Yhdyskäytävälaitteen etukäteen määriteltyihin Modbus TCP/IP -transaktioihin perustuva toiminta myös teki koko ohjelmiston konfiguroitavuuden suunnittelemisesta haasteellista. Konfiguraatitiedoston pitäminen rakenteellisesti yksinkertaisena ja helposti kirjoitettavana vaati konfiguraatiomallin joustavuuden rajoittamista. Esimerkiksi jokaiselle yhdyskäytävän transaktiolle on vain yksi yhteinen konfiguraatio.

M7-rajapinnan suunnittelussa haasteeksi muodostui palvelinohjelmiston ja sen REST-rajapinnan dokumentaation saatavuus alihankkijalta. Palvelimen ja REST-rajapinnan toimintaa piti selvittää itse tutkimalla lähdekoodeja ja lähettämällä testipyyntöjä palvelimelle. Myös testipalvelimen kanssa käytetty tietokanta aiheutti jonkin verran päänvaivaa, sillä kantaan jouduttiin muun muassa manuaalisesti lisäämään simuloituja mittaustuloksia oikeiden laitteiden puuttumisen takia.

Yhdyskäytävälaitteen tavoin M7-rajapinnan konfiguraation suunnittelu oli haasteellista. Konfiguraatitiedoston pitämiseksi yksinkertaisena jouduttiin konfiguraatiomallista tekemään varsin rajoittunut. Esimerkiksi kaikille konfiguroiduille S7-laitteille on ainoastaan yhdet yhteiset Modbus- ja anturikonfiguraatiot. Mikäli konfiguraatiomallista tehdään vähemmän rajoittunut, tarkoittaa se todennäköisesti huomattavasti pidempiä ja hankalammin kirjoitettavia konfiguraatitiedostoja laitteiden määrän lisääntyessä.

Yhdyskäytävän ja M7-rajapinnan konfiguraatiomallit ovat lisäksi toisistaan riippuvaisia. Esimerkiksi yhdyskäytävälaitteen konfiguraatiossa määritellyn funktiokoodin tulee vastata M7-rajapinnan konfiguraatiossa määriteltyä Modbus-datatyyppeä. Tällainen riippuvaisuus ei ole suunnittelun näkökulmasta ideaalia, mutta konfiguraatioiden täydellinen erottaminen toisistaan käytettäessä valittuja yhdyskäytävälaitteita olisi haasteellista.

PROFIBUS-kenttäväylä itsessään asettaa myös tietynlaisia rajoitteita. Yksi PROFIBUS-slave-laite, tässä tapauksessa käytetty yhdyskäytävä, voi lähettää master-laitteelle ainoastaan 244 tavua dataa sykliä kohti. Tämä tarkoittaa sitä, että mikäli yhdellä yhdyskäytävälaitteella halutaan siirtää esim. 30:n kentällä olevan laitteen data PROFIBUS-väylälle, jokaiselta laitteelta voidaan ottaa ainoastaan kahdeksan tavun verran dataa. Jos taas yhden laitteen siirtämän datan määrää lisätään, laitteiden määrä yhdyskäytävää kohden romahtaa nopeasti – kuudellatoista tavulla maksimi on 15 laitetta, kun taas 32 tavua rajoittaa laitteiden määrän seitsemään.

Toteutetussa ratkaisussa havaittuja ongelmia voitaisiin mahdollisesti välttää esimerkiksi käyttämällä yhdyskäytävälaitteiden sijaan RS-485-ajuria ja ohjelmallisesti toteutettua PROFIBUS-protokollapinoa, joka simuloi useaa PROFIBUS-slave-laitetta ja on suoraan yhteydessä M7-rajapintaan ilman välissä tapahtuvaa Modbus TCP/IP -muunnosta. Tällainen toteutus, vaikkakin vähemmän rajoitettu konfiguraation näkökulmasta, olisi luultavasti huomattavasti hankalampi toteuttaa PROFIBUS-protokollan verrannollisen monimutkaisuuden vuoksi. Toisin kuin yhdyskäytävätoteutuksessa, jossa on aina käytössä sama Modbus TCP/IP -pino, ohjelmallista PROFIBUS-pinoa ei myöskään voisi hyödyntää muita kenttäväyläteknologioita käyttävissä ympäristöissä. Näin ollen valittu lähestymistapa PROFIBUS-rajapinnan toteuttamiseksi on mielestäni hyväksyttävä kompromissi.

## LÄHTEET

Creowave Oy 2015a. Products. Viitattu 24.4.2019, <http://www.creowave.com/products/>.

Creowave Oy 2015b. Technical specs. Viitattu 24.4.2019, <http://www.creowave.com/products/technical-specs/>.

Dutertre, B. 2008. Formal Modeling and Analysis of the Modbus Protocol. Teoksessa Goetz, E. & Shenoj, S. Critical Infrastructure Protection. Boston: Springer, 189–204. Viitattu 22.3.2019, [https://link.springer.com/content/pdf/10.1007%2F978-0-387-75462-8\\_14.pdf](https://link.springer.com/content/pdf/10.1007%2F978-0-387-75462-8_14.pdf).

HMS Industrial Networks 2012. Anybus X-Gateway Modbus-TCP – PROFIBUS DPV1 User Manual Rev 1.10. Viitattu 11.4.2019, [https://www.anybus.com/docs/librariesprovider7/default-document-library/manuals-design-guides/hms-hmsi-168-46.pdf?sfvrsn=f66baad6\\_4](https://www.anybus.com/docs/librariesprovider7/default-document-library/manuals-design-guides/hms-hmsi-168-46.pdf?sfvrsn=f66baad6_4).

HMS Industrial Networks 2019. Anybus X-Gateway – Modbus TCP Client – Profibus Slave. Viitattu 8.4.2019, <https://www.anybus.com/products/gateway-index/anybus-xgateway/modbus-tcp-detail/anybus-x-gateway---modbus-tcp-client---profibus-slave>.

ISO & IEC 1994. ISO/IEC 7498-1:1994(E) Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model. Viitattu 24.4.2019, [https://standards.iso.org/ittf/PubliclyAvailableStandards/s020269\\_ISO\\_IEC\\_7498-1\\_1994\(E\).zip](https://standards.iso.org/ittf/PubliclyAvailableStandards/s020269_ISO_IEC_7498-1_1994(E).zip).

Jaganmohan Reddy, Y. & Mehta, B.R. 2014. Industrial Process Automation Systems. Waltham, Massachusetts: Butterworth-Heinemann. Viitattu 25.4.2019, <https://learning.oreilly.com/library/view/industrial-process-automation/9780128009390/>.

Kunbus GmbH 2017. Getting Started Windows DF PROFI II. Viitattu 25.4.2019, [https://www.kunbus.de/files/media/bedienungsanleitungen/DF\\_PROFI\\_II\\_GettingStarted\\_Windows\\_E.pdf](https://www.kunbus.de/files/media/bedienungsanleitungen/DF_PROFI_II_GettingStarted_Windows_E.pdf).

Kunbus GmbH 2019. Kunbus DF PROFI II. Viitattu 25.4.2019, <https://www.kunbus.com/pc-cards-profibus-df-profi-ii.html>.

Modbus Organization 2006. MODBUS Messaging on TCP/IP Implementation Guide V1.0b. Viitattu 21.3.2019, [http://modbus.org/docs/Modbus\\_Messaging\\_Implementation\\_Guide\\_V1\\_0b.pdf](http://modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf).

Modbus Organization 2012. Modbus Application Protocol Specification V1.1b3 2012. Viitattu 21.3.2019, [http://modbus.org/docs/Modbus\\_Application\\_Protocol\\_V1\\_1b3.pdf](http://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf).

Modbus Organization 2019. Modbus FAQ. Viitattu 21.3.2019, <http://modbus.org/faq.php>.

Profibus Nutzerorganisation e.V. 2016. PROFIBUS System Description Technology and Application. Viitattu 5.4.2019, <https://www.profibus.com/index.php?eID=dumpFile&t=f&f=52380&token=4868812e468cd5e71d2a07c7b3da955b47a8e10d>.

Sauter, T. 2017. Fieldbus System Fundamentals. Teoksessa Zurawski, R. Industrial Communication Technology Handbook, 2nd Edition. Boca Raton, Florida: CRC Press, 1.1–1.7. Viitattu 18.4.2019, <https://learning.oreilly.com/library/view/industrial-communication-technology/9781482207323>.

Suomen Asiakastieto Oy 2017. Creowave Oy. Viitattu 24.4.2019, <https://www.asiakastieto.fi/yriyset/fi/creowave-oy/19299321/taloustiedot>.

## LIITTEET

Liite 1. Systemikaavio

Liite 2. Ohjelmakaavio

