

Eero Eriksson

# MODBUS TCP/IP -PROTOKOLLAN KÄYTTÖ TOIMILAITTEIDEN OHJAUK- SESSA

Kandidaatintyö  
Tekniikan ja luonnontieteiden tiedekunta  
Tarkastaja: Mikko Salmenperä  
Maaliskuu 2024

# TIIVISTELMÄ

Eero Eriksson: Modbus TCP/IP -protokollan käyttö toimilaitteiden ohjauksessa  
Kandidaatintyö  
Tampereen yliopisto  
Teknisten tieteiden kandidaatin tutkinto-ohjelma  
Maaliskuu 2024

---

Tiedonsiirtotekniikassa on käytettävissä lähes rajaton määrä mahdollisuuksia toteuttaa jopa maailmanlaajuisia verkostoja. Tässä opinnäytetyössä tutkittiin muutamaa maailmalla laajassa käytössä olevaa tekniikkaa, joiden pohjalta toteutettiin toimilaitteita ohjaava ohjelma. Ohjelman tekemisen avulla saatiin käytännön kokemusta mahdollisesta teollisuuden käyttökohteesta.

Tämä työ on jaettu kahteen osaan. Ensimmäisenä osana on teoria osuus, jossa perehdytään kenttäväylätekniikkaan ja Modbus-protokollaan, joiden avulla saadaan siirrettyä dataa tietoliikenneverkossa. Näiden kahden protokollan pohjalta tehtiin käytännön työ, jossa ohjataan kahden toimilaitteen toimintaa Modbus TCP/IP -verkossa.

Tiedonsiirtoon kenttälaitteiden välillä voidaan hyödyntää kenttäväyliä, joita käsitellään tämän työn ensimmäisessä teorialuvussa. Kenttäväylät ovat tietoliikenneprotokollia, jolla siirretään dataa verkossa ilman tarvetta erillisille I/O-alijärjestelmille. Moderneilla väylillä on useita hyötyjä ja ominaisuuksia tuotantoprosessin kehityksen eri vaiheissa, mitä esitellään tämän tutkimuksen aikana. Väylät voivat muodostaa erilaisia topologioita, joilla on erilaisia ominaisuuksia ja käyttökohteita.

Luvussa kolme tutustutaan Modbus-protokollan toimintaan, jota käytetään hyödyksi myös työn käytännön osuudessa. Modbus on yksi yleisimmistä kenttäväylien kanssa käytettävistä kommunikointiprotokollista. Protokollan suosio teollisuudessa perustuu yksinkertaisuuteen, monipuolisuuteen ja kustannustehokkuuteen. Tässä työssä perehdyttiin Modbusin toimintaan ja erityisesti sen käyttämien viestien rakenteisiin. Lisäksi funktio- ja poikkeuskoodien avulla sivuutetaan, millaiseen viestintään laitteiden välillä protokollaa voidaan hyödyntää.

Työn käytännön osuudessa luotiin ohjelma, jolla ohjataan kahden moottorin toimintaa ohjauspaneelin kautta. Ohjelma tehtiin käyttämällä Red Lionin luomaa Crimson 3.0 -ohjelmistoa, jolla ohjelmoitiin ohjauspaneelin käyttöliittymä ja kenttäväylän avulla siirrettävien pakettien rakenne Modbus-protokollaa hyödyntäen. Lopullisella ohjelmalla saatiin onnistuneesti siirrettyä dataa paneelilta toimilaitteille ja takaisin ilman toimintahäiriöitä. Näin saatiin havainnollistettua sekä kenttäväylän että Modbus-protokollan toimintaa käytännön sovelluksessa, jota voitaisiin soveltaa esimerkiksi teollisuuden tiedonsiirtotarpeissa.

Avainsanat: Modbus, kenttäväylä, Red Lion, HMI-paneeli

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck -ohjelmalla.

# SISÄLLYSLUETTELO

1. JOHDANTO .....	1
2. KENTTÄVÄYLÄTEKNIikka.....	2
2.1 Kenttäväylän toiminta .....	3
2.2 Kenttäväylien hyödyt.....	3
2.3 Verkon topologia.....	5
3. MODBUS-PROTOKOLLA .....	8
3.1 Toimintaperiaate.....	8
3.2 Viestin datakehys .....	10
3.3 Funktiokoodit .....	12
3.4 Poikkeuskoodit .....	13
4. KÄYTÄNNÖN TOTEUTUS.....	15
4.1 Toimilaitteisto.....	15
4.2 Ohjelmistoympäristö .....	17
4.3 Käyttöliittymä .....	18
4.4 Laitteiden kommunikointi .....	19
5. YHTEENVETO.....	22
LÄHTEET .....	23

# LYHENTEET JA MERKINNÄT

ADU	Application Data Unit
ASCII	American Standard Code for Information Interchange
HMI	Human Machine Interface
IEC	International Electrotechnical Commission
IP	Internet Protocol
MAC	Medium Access Control
MBAP	Modbus Application Protocol
OSI	Open Systems Interconnection
PDU	Protocol Data Unit
RS	Recommended Standard
RTU	Remote Terminal Unit
SAC	Standardization Administration of China
SEMI	Semiconductor Equipment and Materials International
TCP	Transmission Control Protocol

# 1. JOHDANTO

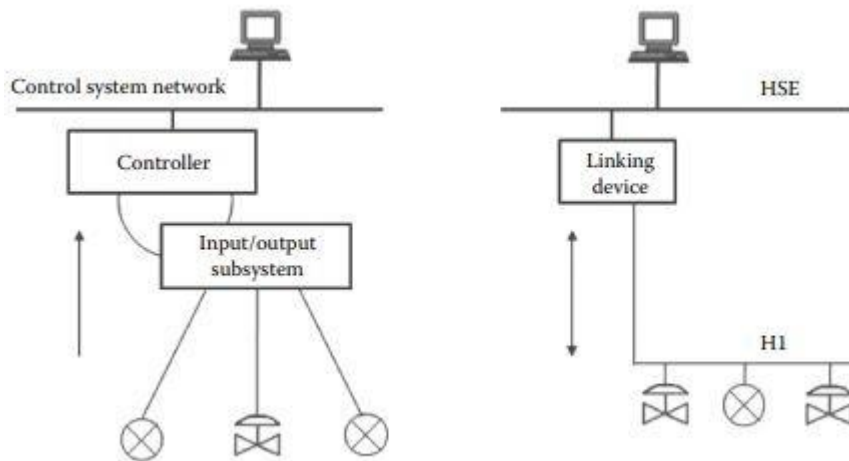
Tiedonsiirtotekniikka on aikojen saatossa muuttunut aina yksittäisistä analogisista signaaleista massiivisiin tietoliikenneverkkoihin, jotka ulottuvat ympäri maailmaa. Sekä taloudellinen että teknillinen kehitys on mahdollistanut lähes rajattoman määrän uusia innovaatioita, joista yhteen, jo valtavassa suosiossa olevaan teknologiaan, tutustutaan tässä työssä.

Tämä työ on jaettu kahteen osaan: teoria osuuteen ja käytännön osuuteen. Teoria osuudessa perehdytään, miten kenttäväylät yleisesti toimivat ja millaisia ne ovat rakenteeltaan. Näiden lisäksi tutustutaan kenttäväyliä hyötyihin ja siihen, miten ne jakautuvat väylän kehityksen ja käytön aikana. Kenttäväyliä yleisen osuuden jälkeen siirrytään yhteen tiettyyn kommunikaatioprotokollaan, joita kenttäväylissä käytetään. Tämä protokolla on Modbus. Modbusin kohdalla tutkitaan viestien toimintaa ja rakennetta väylällä, joita hyödynnetään työn jälkimmäisessä osassa.

Käytännön osuudessa tutkitaan, miten Modbus TCP/IP -protokollaa voidaan hyödyntää käytännön käyttökohteissa. Tässä tapauksessa protokollaa käytetään tiedonsiirtoon kahden taajuusmuuntajan ja ohjauspaneelin välillä. Osuuden pääpainona on ohjelma, jolla ohjataan kenttäväylän toimintaa. Tavoitteena on saada toimiva ohjelma ja käsitys Modbus-protokollan käytöstä ja ominaisuuksista etenkin teollisuuden käytössä.

## 2. KENTTÄVÄYLÄTEKNIikka

Kenttäväylä on kaksisuuntainen, kiertävä linjajärjestelmällinen tietoliikenneprotokolla, jolla voidaan yhdistää kentälaitteita toisiinsa [1, s. 81]. Protokolla on kehitetty korvaamaan perinteistä mallia, jossa kaikki laitteet yhdistetään ohjaavaan laitteeseen kahdella kaapelilla. Kiertävällä linjajärjestelmällä tarkoitetaan, että protokollalla luodaan paikallinen verkko, jossa toimilaitteet kommunikoivat toistensa kanssa yhteistä väylää pitkin ilman, että on tarvetta keskitetylle ohjausverkolla. Kaksisuuntaisuus muodostuu, koska laitteet pystyvät sekä lukemaan väylässä kulkevaa dataa että kirjoittamaan dataa väylään. Kuvassa 2.1 esitetään perinteisen mallin ja kenttäväylän ero käytännössä. Malleista huomataan, että mitä enemmän laitteita vanhan malliseen verkkoon tulee, sitä monimutkaisemmaksi se muuttuu. Kenttäväylän kanssa samaa ongelmaa ei ole. Kenttäväylän muista hyödyistä kerrotaan lisää alaluvussa 2.2.



**Kuva 2.1.** Vasemmalla vanhan mallin ja oikealla kenttäväylän kaavio [1, s. 86]

Kenttäväyliä käytetään pääosin teollisissa ympäristöissä, mutta käyttö ei rajoitu siihen. Väylää rakennettaessa on useita protokollavaihtoehtoja, joista voi valita tilanteeseen sopivimman, sillä jokaisella eri protokollalla on eri vahvuudet ja heikkoudet. Vaikka kenttäväylillä on paljon yhteisiä hyötyjä, tietyissä käyttökohteissa on tarkkoja rajoitteita, esimerkiksi reaaliaikaisuuden suhteen. Yleisimpiä kommunikointiprotokollia, joita käytetään kenttäväylien kanssa, on Foundation Fieldbus, Profibus, Modbus, HART ja Profinet [2]. Modbus-protokollaa tarkastellaan luvussa 3 tarkemmin.

## 2.1 Kenttäväylän toiminta

Kenttäväylien rakenne voidaan yleisesti yksinkertaistaa kolmeen eri kerrokseen: Fyysinen, siirtoyhteys- ja sovelluskerros [3, s. 1084]. Nämä kolme kerrosta on osa Open Systems Interconnection (OSI) -mallia. Fyysiseen kerrokseen kuuluu väylän toimilaitteet, kaapelit ja niiden muodostamat topologiat, joita käsitellään alaluvussa 2.3. Siirtoyhteyskerroksessa muodostetaan kenttäväylää pitkin siirrettävän paketin rakenne verkon tarvitsemaan muotoon ja järjestykseen. Tässä kerroksessa on myös Medium Access Control (MAC), jonka tehtävänä on estää väylän käyttö useamman laitteen toimesta samanaikaisesti [4, s. 214]. Sovelluskerroksessa on kentän käyttämiseen ja seuraamiseen tarvittavat sovellukset, joihin kuuluu esimerkiksi käyttöliittymät. Muut neljä OSI-mallin kerrosta ovat joko sisällytettyinä edellä mainittuihin kerroksiin tai ne eivät ole väylien muiden ominaisuuksien ansiosta tarpeellisia [3, s. 1085]. Tämä yleistys ei aina toteudu kaikissa sovelluskohteissa, mutta perusrakenne väylässä pysyy hyvin samanlaisena.

Toimilaitteiden välinen suhde toisiinsa riippuu kenttäväylään valitusta protokollasta. Yleisin malli on asiakas–palvelin -malli tai jokin kyseisen mallin muista variaatioista, jossa palvelin suorittaa asiakkaan pyytämän toiminnon. Luvussa 3 tarkasteltava Modbus-protokolla käyttää tätä mallia. Toinen yleinen malli on julkaisija–tilaaja -malli, jossa julkaisija jakaa pyydetyn tiedon sille määritetyille tilaajille. Tätä mallia käyttää esimerkiksi Foundation Fieldbus -protokolla. Muut mallit ovat enemmän käyttökohdekohtaisia kuin protokollakohtaisia, joten mahdollisia variaatioita on rajattomasti käytettävissä. [3, s. 1088–1090]

Kenttäväylän päätyihin sijoitetaan terminaattorit, joiden avulla vähennetään mahdollisia virheitä signaalissa. Näitä virheitä on esimerkiksi signaalin heijastumiset, jolloin signaali ei aina saavuta tavoiteltua kohdetta väylällä. Terminaattoreja ei lasketa väylässä oleviin toimilaitteisiin, koska ne eivät sinällään kommunikoi väylän kanssa, vaikkakin niiden olemassaolo on väylän kannalta olennaista. Kenttäväylään on mahdollista käyttää muitakin määriä terminaattoreja kuin kaksi, mutta yleensä se johtaa signaalien laadun heikkeneemiseen tai jopa pilaantumiseen. [1, s. 90]

## 2.2 Kenttäväylien hyödyt

Kenttäväylien kehitys voidaan karkeasti viiteen vaiheeseen. Nämä vaiheet ovat toteutusjärjestyksessä suunnittelu, asennus, operointi, kunnossapito ja uudistaminen [1, s. 91]. Suurimpina hyötyinä yritysten kannalta on huomattavasti pienemmät kustannukset kuin perinteisemmällä pisteestä pisteeseen (point-to-point) kytkennöillä, sekä se yksinkertaistaa verkon rakennetta, josta on hyötyä kaikissa viidessä vaiheessa.

Järjestelmää suunniteltaessa voi kulua pitkiä aikoja riippuen projektin laajuudesta. Suunnitteluun kuluva aikaa pystytään lyhentämään käyttämällä kenttäväyliä, koska suunniteltavien ja dokumentoitavien osien määrä pienenee huomattavasti, koska väylien avulla voidaan jättää jopa 80 % kaapeleista ja tarve I/O-alijärjestelmille poistuu [4, s. 210]. Tämä erityisesti pätee suurissa järjestelmissä. Myös projektin suunnittelun koordinointi ja aikatauluttaminen helpottuu työkuorman pienentyessä, jolloin myös suunnittelusta tulee kustannustehokkaampaa.

Järjestelmän suunnittelun jälkeen on järjestelmän asennusvaihe, jolloin projektin suunnitelma toteutetaan. Kenttäväyliä käyttämällä saadaan asennuksen materiaalikuluista suoraan vähennettyä edellä mainitut ylimääräiset kaapelit ja I/O-alijärjestelmät. Rakenteen yksikertaistumisella saadaan vähennettyä laitteiston ohjelmistojen ja ohjausten tuottamiseen kuluva aikaa ja kustannuksia. Kaapelien vähyyden ansiosta voidaan myös minimoida kytkennöissä tapahtuvia virheitä.

Kenttäväylien hyöty operointivaiheen aikana perustuu väylän toimintavarmuuteen. Yksittäisten laitteen viat tapahtuvat usein yhden syklin aikana. Väylä pystyy pitämään vian laitteen oman ohjauspiirin alueella minimoiden vaikutuksen muihin laitteisiin. Modernit laitteet ovat usein itsestään korjautuvia. Näin muu järjestelmä voi jatkaa toimintaansa normaalisti ja seuraavalla syklillä on palattu haluttuun tilaan. Koska kaikki laitteet ovat jaetussa väylässä, voidaan kaikki data siirtää mille tahansa laitteelle tarvittaessa, jolloin prosessien tarkkailu on luontevaa toteuttaa. Toisena hyötynä on, että väylä on monesti nopeampi ja tarkempi kuin joukko yksittäisiä laitekytkentöjä. Väylät ovat yleensä kokonaan digitaalisia, jolloin laitteiden välisiä muunnoksia analogisen ja digitaalisen signaalin välillä, joten turhia pyörityksiä voidaan välttää, ja aikaa ei kulu turhiin prosesseihin. [1, s. 91]

Seuraava vaihe on operointia tukeva kunnossapitovaihe. Kuten aiemmin mainittiin, modernit laitteet pystyvät korjaamaan osan pienemmistä vioistaan tekemällä itsediagnoosin ja uudelleen kalibroimalla tai syklin nollauksella [4, s. 210]. Jos laite ei pysty korjaamaan vikaansa itse, voidaan laite ottaa kunnossapidon ajaksi pois väyläpohjaisesta verkosta ilman, että muita prosesseja tarvitsee pysäyttää, jolloin seisakkaika saadaan minimoitua. Koska väylään tarvitsee vähemmän laitteistoa ja kaapelia, mekaanisten vikojen määrä pienenee. Mekaaniset viat vaativat lähes aina paikan päällä korjaamista, joka voi vaatia muiden prosessien sammuttamista. Tietotekniseen kunnossapitoon vaihtoehtona on kenttäväylien yksinkertaisesti mahdollistama etäkäyttö, joka on turvallisempi vaihtoehto kuin paikan päällä korjaaminen [3, s. 1078].



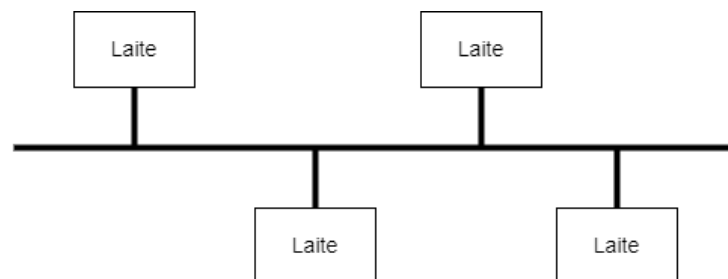
Viimeisenä vaiheena on uudistaminen. Kun osa verkosta ei ole enää kunnossapidettävissä, pitää tilalle vaihtaa uusi laite. Kuten kunnossapidettäessä, väylän modulaarisuus mahdollistaa laitteen vaihdon ilman suurempaa vaikutusta muuhun verkkoon [1, s. 91]. Laitteita voidaan vaihtaa tai päivittää uusiin, jos viestin muoto pysyy samana, joten laitevaihtoehtoja on runsaasti. Kenttäväylän runko on ainoa vaikeammin vaihdettava kohde, mutta hyvin suunniteltua ja toteutettua väylää harvemmin tarvitsee uusia. Väylän modulaarisuuden suurena etuna on, että osia pystytään vaihtamaan tai päivittämään yksi kerrallaan, jolloin investoinnit voidaan toteuttaa portaittain.

## 2.3 Verkon topologia

Kenttäväylän kytkentöjen rakenne on kuvattuna verkon topologiassa. Topologian valinta perustuu pääosin väylän ja siihen kytkettyjen laitteiden fyysisestä sijainnista kohteessa [4, s. 208]. Suuremmissa verkoissa voi olla useampi eri topologioita yhdistettynä varsinkin, jos verkko sisältää aliverkkoja. Yleisimpiä topologioita ovat pisteestä pisteeseen, väylä, tähti ja kehä.

Pisteestä pisteeseen -topologia on kaikkein yksinkertaisin kenttäväylän topologia. Väylä kulkee kahden laitteen välillä, jolloin muita laitteita väylään ei kytketä. Se on siis melkein identtinen perinteisen kytkentämallin kanssa, mutta väylä käyttää yhtä kaapelia kahden sijaan. Tämän topologian yleisin käyttökohte on tietokoneen kytkeminen ohjaamaan toisen laitteen toimintaa. [2]

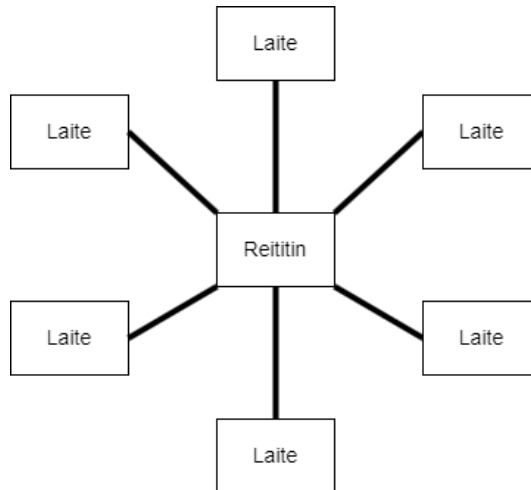
Väylätopologiassa laitteet ovat haaroitettu väylässä toimivasta pääkaapelista. Tieto siirtyy laitteelta haaran kautta pääväylään kaikkien muiden laitteiden käytettäväksi. Tätä topologiaa käytetään paljon varsinkin tehdasautomaatiojärjestelmissä ja pienissä verkoissa [5, s. 91]. Kuvassa 2.2 on esitettyä, miltä topologia näyttää.



**Kuva 2.2.** Laitteet kytketty väylätopologiseen verkkoon

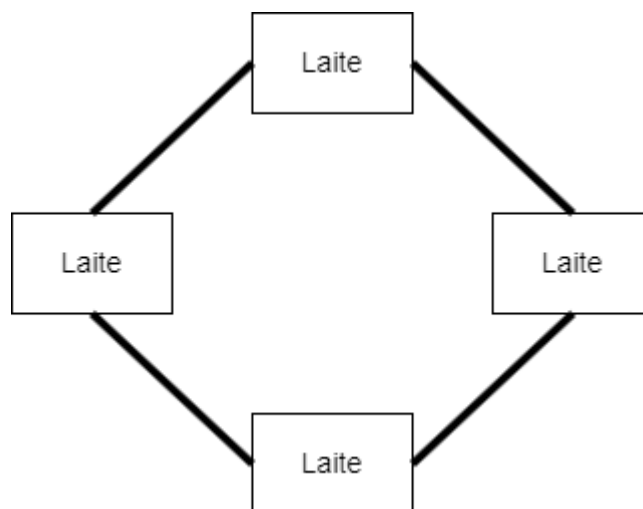
Verkossa olevat laitteet voidaan asentaa niin, että niistä lähtevät kaapelit kulkevat keskitetyksi yhden laitteen, kuten kytkimen tai reitittimen, kautta. Tätä topologiaa kutsutaan tähtitopologiaksi. Tähtitopologisen verkon etuna on, että se on kaikista malleista helpoin

muokata tai korjata, josta syystä se on laajasti käytössä [5, s. 94]. Se on myös viansietokyvyltään paras topologia. Kuvan 2.3 esimerkkimallista nähdään, millainen tähtitopologia on.



**Kuva 2.3.** Tähtitopologinen verkko

Kehätopologiassa laitteet ovat nimensä mukaisesti kytkettynä toisiinsa laitteesta laitteeseen, kunnes on palattu takaisin ensimmäiseen laitteeseen. Kuvassa 2.4 laitteet ovat kytkettynä toisiinsa kehänä [2]. Laitteiden viestit kulkevat tiettyyn suuntaa laitteista toiseen, kunnes ne saavuttavat halutun laitteen. Kehätopologian tehokkuutta voidaan tehostaa käyttämällä toista rinnalla olevaa kehää, jossa viestit kulkevat toiseen suuntaan, jolloin saadaan lisättyä sekä redundanssia että verkon kapasiteettia [5, s. 92–93].



**Kuva 2.4.** Laitteet kehätopologian mukaisesti

Kuten aiemmin mainittiin, verkkojen rakenteissa käytetään monesti useita eri topologioita. Tämä mahdollistaa kenttäväylien tehokkaan käytön lähes kaikissa järjestelmissä. Aliverkojen kytkemiseen sopii erityisesti tähti- ja väylätopologian yhdistelmät, koska niiden lisääminen reitittimen kautta toteutuu yksinkertaisesti.

## 3. MODBUS-PROTOKOLLA

Modbus-protokolla on Modicon Inc:n, nykyisin Schneider Electronics, vuonna 1978 kehittämä kommunikointiprotokolla, jonka tarkoituksena oli saavuttaa yksinkertainen keino siirtää dataa kahden laitteen välillä [6, s. 382]. Protokolla on saavuttanut niin suuren suosion, että nykyisin se mielletään teollisuusautomaation tiedonsiirron 'de facto' -standardina. Protokollan ovat 'de facto' -standardin lisäksi standardoineet International Electrotechnical Commission (IEC), Standardization Administration of China (SAC) ja osittain Semiconductor Equipment and Materials International (SEMI) toimesta [6, s. 382–385]. Vuonna 2004 Schneider Electronics siirsi protokollan Modbus Organizationin oikeudet omistukseen, minkä takia Modbus-protokolla ei sisällä lisensointikuluja [7].

Modbusin suosio johtuu suuresti protokollan yksinkertaisuudesta ja monipuolisuudesta. Modbus ei vaadi suurta määrää prosessointitehoa, jolloin verkostoon voidaan lisätä yksinkertaisiakin laitteita varsinkin, kun liitännävaihtoehtojakin on monenlaisiin standardeihin, kuten standardeihin RS232, RS422, RS485 ja TCP/IP [6, s. 384–385]. Varsinkin Modbus TCP/IP -protokollan hyötynä on se, että se on helposti yhdistettävissä Internetiin, jolloin sitä voidaan käyttää ja ylläpitää käytännössä mistä tahansa, mikä on tietyissä tilanteissa huomattavasti kannattavampaa tai turvallisempaa kuin fyysisesti paikalle meneminen [7]. Koska Modbus on pääosin avoimeen lähdekoodiin perustuva protokolla, on protokollan käyttö sekä pienten että isojen yritysten kannalta hyvin kustannustehokasta [8, s. 16–17].

### 3.1 Toimintaperiaate

Modbus-protokollan toiminta perustuu OSI-mallin sovelluskerroksessa tapahtuvaan palvelimen (server) ja asiakkaan (client) väliseen kommunikointiin. Modbus-protokollaa voidaan kuljettaa useiden erilaisten alla olevien kerrosten avulla, mikä mahdollistaa sen, että verkostossa olevat laitteet voivat toimia sekä asiakkaina että palvelimina ja asiakas ja palvelin voivat keskustella useamman vastakkaisen laitteen kanssa. Vaihtoehtona on myös, että asiakas ei pyydä vastausta palvelimelta, jolloin tieto liikkuu vain yhteen suuntaan laitteiden välillä, jolloin asiakas ei saa varmuutta, että palvelin on onnistuneesti käsitellyt pyynnön [6, s. 385–386].

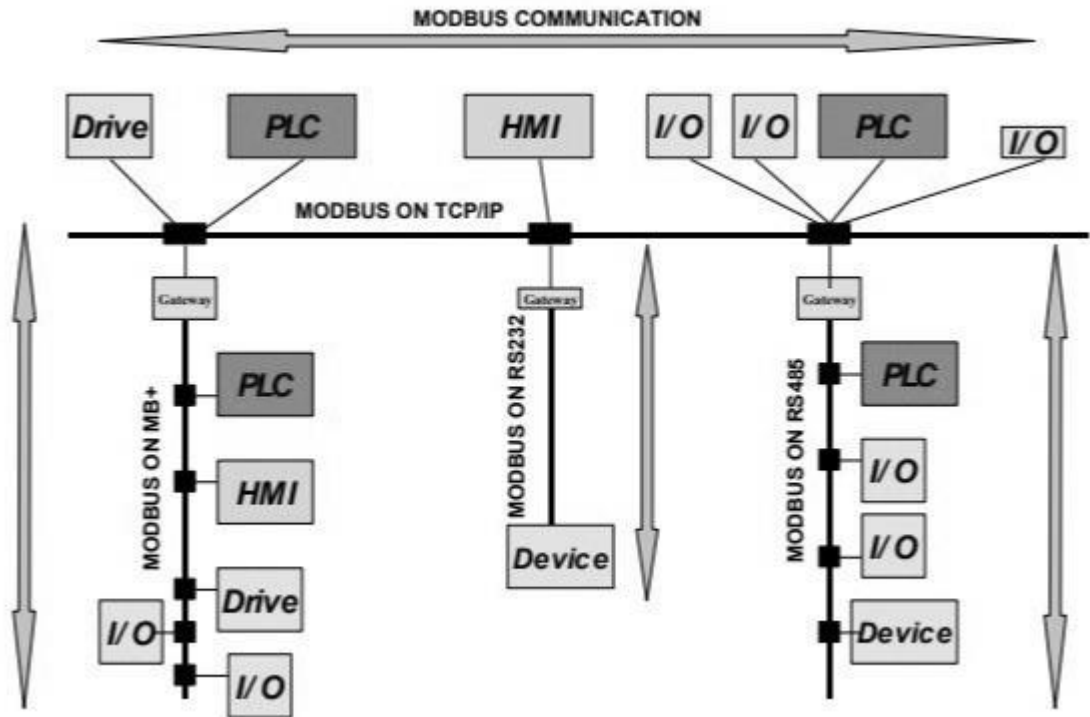
Yksittäinen keskustelu tapahtuu kuvan 3.1 mukaisesti, jossa asiakas lähettää pyynnön palvelimelle. Palvelin yrittää toteuttaa asiakkaan pyynnön ja palauttamaan siihen sopi-

van vastauksen takaisin. Asiakas tarkastaa saadun vastauksen varmistaakseen, että tapahtuma oli onnistunut. Jos palvelin ei pystynyt toteuttamaan pyyntöä, saa asiakas vastauksen sijasta tapahtunutta virhettä vastaavan virheilmoituksen. [9, s. 2] Kaikille verkkoon kuuluville laitteille on asetettu osoite, jotta laitteiden lähettämät pyynnöt ja vastaukset päätyvät oikeisiin laitteisiin. Järjestelmään on usein määritetty tietty aikaraja, jonka aikana haluttu toiminto tulee toteuttaa, jotta muut verkon toiminnot eivät jää turhaan odottamaan. Reaaliaikaisissa järjestelmissä tämän aikarajan vaikutus korostuu, jolloin voidaan asettaa aikarajat järjestelmätason sijaan pyyntökohtaisiksi prosessin laadun takaamiseksi. Tämäkin palautetaan asiakkaalle virheilmoituksena. Laitteiden välisten viestien rakennetta tarkastellaan tarkemmin alaluvussa 3.2.



**Kuva 3.1.** Palvelimen ja asiakkaan välinen kommunikointi [6, s. 383].

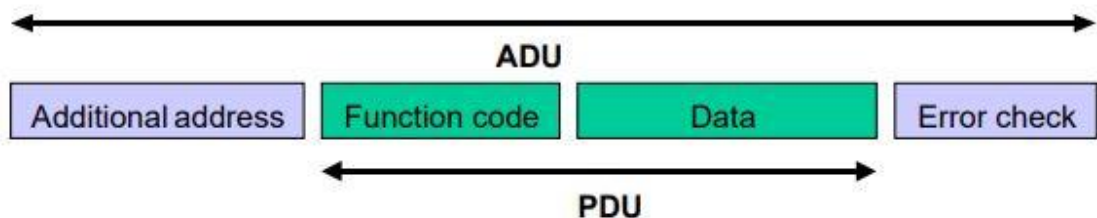
Modbus-verkot, kuten monet muutkin tietoliikenneverkot, mahdollistavat monien erilaisien laitteiden lisäämistä verkkoon ilman suurempi muutoksia. Kuvassa 3.2 nähdään malliesimerkki toimivasta Modbus TCP/IP -verkosta. Verkot voivat sisältää useita eri protokollia käyttäviä laitteita tai aliverkkoja käyttämällä yhdyskäytäviä tai reitittäjiä. Kuten kuvassa 3.2 nähdään, verkossa voi olla sekä useita asiakkaita että palvelimia, joilla on mahdollisuus kommunikoida keskenään. Järjestelmän PLC esimerkiksi usein toimii sekä asiakkaina korkeamman tason laitteille että palvelimena alemman tason I/O-laitteille.



Kuva 3.2. Esimerkki Modbus-verkosta [10, s. 3]

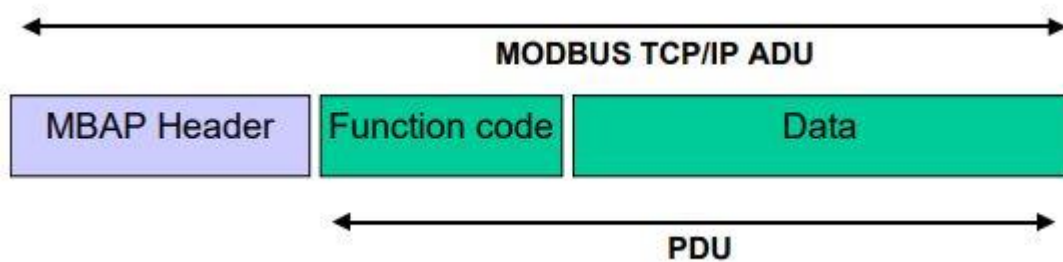
### 3.2 Viestin datakehys

Modbus-protokollan viesti on määritettynä asiakkaan tekemässä Application Data Unitista (ADU). ADU yleisesti koostuu kuvan 3.3 mukaisesti osoitteesta (address), Protocol Data Unitista (PDU) ja virheen tarkastuksesta (error check). Osoitekenttä sisältää viestin kohteen määränpäähän joko kahden merkin mittaisena American Standard Code for Information Interchange (ASCII) -viestinä tai kahdeksan bitin Remote Terminal Unit (RTU) -viestinä [11, s. 81]. Virheen tarkastus toimii hyvin samantyyllisesti kuin viestin osoite. Jos virhe tapahtuu viestin lähettämisessä tai vastaanottamisessa, tulee ADU:hun virheen poikkeuskoodi samassa muodossa kuin viestin osoite [10, s. 4].



Kuva 3.3. Modbus-viestin datakehys [9, s. 4].

Yleisestä mallista poiketen Modbus TCP/IP -protokollan ADU sisältää osoitteen ja virheetarkastuksen sijasta Modbus Application Protocol (MBAP) -otsikon kuvan 3.4 mukaisesti. MBAP-otsikko sisältää viestin tapahtuman tunnisteen, kohteen tunnisteen, käytetyn protokollan, joka on Modbusin kohdalla 0, ja viestin pituuden. Tapahtumaan lisätään tunniste, joka yksinkertaistaa viestin etenemisen seuraamista verkossa. Palvelimen tehtävänä on lisätä pyynnön tunniste myös vastaukseen, jolloin niistä muodostuu helposti tunnistettava pari. Kohteen tunniste on uniikki arvo, joka on määritetty jokaiselle verkossa olevalle laitteelle, jotta viestit päätyvät haluttuun päätepisteeseen. Nämä toimivat siis samantyyllisesti kuin yleisen mallin osoitteet. Viestin pituus on erikseen määritetty osaksi MBAP:ta, koska sen avulla voidaan tunnistaa, onko viesti jaettu useampaan osaan. Näin saadaan myös vähennettyä virheen mahdollisuutta sekä pyynnössä että vastauksessa [10, s. 5].



**Kuva 3.4.** Modbus TCP/IP -viestin datakehys [9, s. 4].

Sekä yleisessä että TCP/IP-protokollan mukaisessa ADU:ssa on samanlainen PDU, koska PDU ei ole riippuvainen OSI-mallin kommunikointikerroksista. PDU sisältää viestin olennaisimman osa eli laitteiden välillä siirrettävän datan, joka voi olla jaettuna yhteen tai useampaan viestiin riippuen siirrettävän datan koosta, koska datakentän maksimikoko on 253 tavua. Tämä tarkoittaa myös sitä, että yleisen mallin ADU:n maksimikoko on 256 tavua ja TCP/IP-protokollan 260 tavua [10, s. 5]. Datakentän käyttö ei ole välttämätöntä, vaan sen voi jättää tyhjäksi, jos niin halutaan. Tätä voidaan käyttää esimerkiksi siihen, että asiakas saa varmuuden, että viesti on vastaanotettu onnistuneesti ilman tarvetta saada muuta tietoa takaisin palvelimelta. Lähetettävän datan lisäksi PDU sisältää funktiokoodin, joka määrittää, miten palvelimen tulee käsitellä lähetettyä dataa ja mahdollisesti vastata asiakkaan pyyntöön.

### 3.3 Funktiokoodit

Funktiokoodien avulla järjestelmän osat osaavat käyttää viestin sisältöä oikealla tavalla. Funktiokoodit, kuten viestin osoite ja virheen tarkastus, on määritetty kahdella ASCII- tai kahdeksanbittisellä RTU-merkillä [11, s. 81]. Funktiokoodit voidaan jakaa kolmeen osaan määritetyillä väleillä. Julkiset funktiokoodit ovat yleisimmät koodit, joita käytetään, koska ne ovat käyttökohteesta riippumatta aina samoja ja valmiiksi dokumentoituja. Toisena osana on koodit, joita käyttäjä voi määrittää haluamaansa käyttöön, jolloin ei ole takuuta, että käytetty koodi on uniikki järjestelmässä. Viimeisenä osana on yritysten varaamia funktiokoodeja, jotka eivät ole julkisesti käytettävissä. [10, s. 10] Kuvassa 3.5 on määritettyä julkiset funktiokoodit käyttötarkoituksineen.

Function Code	Description
FC 2 (0x02)	Read discretes.
FC 1 (0x01)	Read coils.
FC 5 (0x05)	Write single coil.
FC 15 (0x0F)	Write multiple coils.
FC 5 (0x05) with unit ID = 0	Broadcast write single coil.
FC 15 (0x0F) with unit ID = 0	Broadcast write multiple coils.
FC 4 (0x04)	Read input registers.
FC 3 (0x03)	Read holding registers.
FC 6 (0x06)	Write single holding register.
FC 16 (0x10)	Write multiple holding registers.
FC 22 (0x16)	Mask write holding register.
FC 23 (0x17)	Read/write holding registers.
FC 24 (0x18)	Read FIFO.
FC 6 (0x06) with unit ID = 0	Broadcast write single holding register.
FC 16 (0x10) with unit ID = 0	Broadcast write multiple holding registers.
FC 20 (0x14)	Read file record.
FC 21 (0x15)	Write file record.
FC/subcode 43 (0x2B)/14 (0x0E)	Read device identification.
FC 7 (0x07)	Read exception status.
FC 8 (0x08)	Diagnostics.
FC 11 (0x0B)	Get com event counter.
FC 12 (0x0C)	Get com event log.
FC 17 (0x11)	Report server ID.
FC/subcode 43 (0x2B)/13 (0x0D)	CANopen general reference request and response PDU. Please see Note 4.

**Kuva 3.5.** Julkiset funktiokoodit [6, s. 394]

Joillekin funktiokoodeille on erikseen määritetty tiettyjä ehtoja muihin viestin osiin, kuten kuvan 3.5 koodista 5 ja 15 nähdään, että MBAP-otsikon kohteen tunniste on nolla, koska tieto kuulutetaan (broadcast) koko verkolle, jolloin tiettyä vastaanottajaa ei ole. Koodilla voi olla myös alakoodeja, jotka tarkentavat haluttua toimintoa, kuten koodilla 43 on alakoodi 13, joka viittaa CANopen-protokollaan. Taulukosta samalla näkee, minkälaisia



kommunikointimahdollisuuksia asiakkaan ja palvelimen välillä on käytettäessä Modbusin valmiita funktiokodeja, joiden lisäksi käyttäjällä on mahdollisuus luoda omia funktiokodeja käytännössä rajattomasti.

### 3.4 Poikkeuskoodit

Modbus-viestien lähettäminen ei aina toimi halutulla tavalla. Viestin lähettäjä saa vastauksen pyyntöönsä palvelimelta, joka sisältää saman funktiokoodin kuin lähetteessä oli. Jos tiedonsiirrossa tapahtuu jonkinlainen virhe, vastaus sisältää kaksi komponenttia. Ensimmäinen osa on tässäkin tapauksessa viestin funktiokoodi, mutta se on tasan 80 heksadesimaalia korkeampi kuin alkuperäinen funktiokoodi. Jälkimmäisenä osana on jokin Modbusin käyttämä poikkeuskoodi (exception code). Poikkeuskoodilla määritetään pyynnön tehneelle asiakkaalle, mikä tiedonsiirrossa on mennyt vikaan. [10, s. 47] Kuvassa 3.6 on esitettyä poikkeuskoodit, joita Modbusilla voidaan käyttää vian määrittämisessä.

Exception Code	MODBUS name	Comments
01	Illegal Function Code	The function code is unknown by the server
02	Illegal Data Address	Dependant on the request
03	Illegal Data Value	Dependant on the request
04	Server Failure	The server failed during the execution
05	Acknowledge	The server accepted the service invocation but the service requires a relatively long time to execute. The server therefore returns only an acknowledgement of the service invocation receipt.
06	Server Busy	The server was unable to accept the MB Request PDU. The client application has the responsibility of deciding if and when to re-send the request.
0A	Gateway problem	Gateway paths not available.
0B	Gateway problem	The targeted device failed to respond. The gateway generates this exception

**Kuva 3.6.** Poikkeuskoodit [9, s. 31]

Poikkeuskoodit jakautuvat karkeasti kolmeen kategoriaan. Kuvasta 3.6 huomataan, että monet poikkeukset liittyvät lähetetyn viestin rakenteeseen ja sisältöön. Lisäksi poikkeuskoodit viisi ja kuusi liittyvät palvelimen tilaan. Nämä eivät välttämättä liity suoraan viestinnässä olevaan vikaan, vaan siihen, että palvelin ei pysty sillä hetkellä käsittelemään pyyntöä, vaan pyytää asiakasta odottamaan. Myös verkon fyysisiin oleviin ongelmiin on käytössä muutamia koodeja. Modbus käyttää ennalta määritettyä odotusaikaa (timeout), jonka verran protokolla odottaa viestin vastausta [10, s. 47]. Jos viestin vastaus ei tule aikarajan sisällä, niin ohjelma ymmärtää vikatilanteen tapahtuneen ilman, että tarvitaan

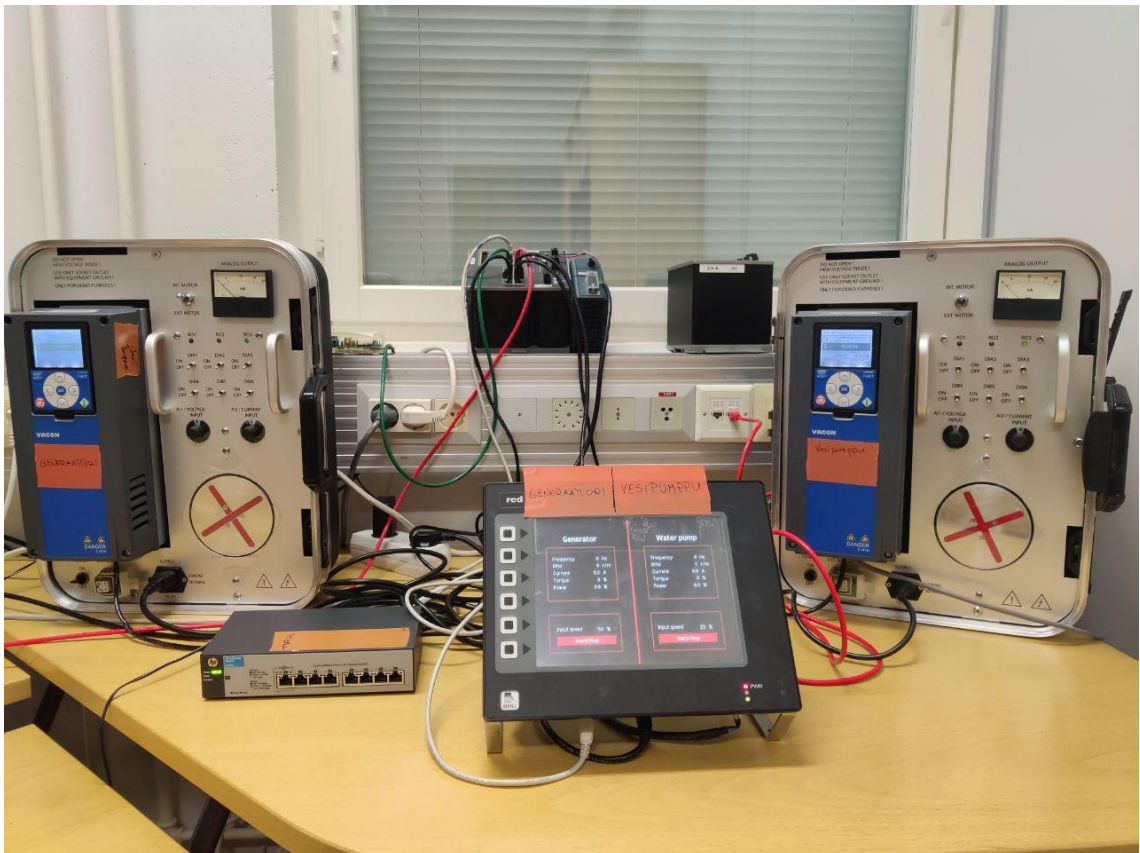
erillistä poikkeuskoodia. Tämä tieto usein näytetään myös ohjelman käyttäjälle, koska syitä siihen, että vastaus ei saapunut, on monia. Aikaraja viestien lähettämiseen on järjestelmäsuunnittelijan määritettävissä, mutta usein riittää, että käyttää TCP-protokollan vakioaikaa [9, s. 26].

## 4. KÄYTÄNNÖN TOTEUTUS

Tässä kappaleessa tarkastellaan työn käytännön osuutta. Tarkoituksena on toteuttaa ohjelma, jolla voidaan ohjata kahden moottorin toimintaa Modbus TCP/IP -protokollaa käyttämällä. Toteutuksella saadaan tarkastelua yhtä tekniikan mahdollista käyttökohdetta ja sitä, miten Modbusin ominaisuudet näkyvät käytännössä. Samalla nähdään, miten luvussa 2 esitellyt kenttäväylät käyttäytyvät.

### 4.1 Toimilaitteisto

Tässä työssä laitteistoon kuuluu kuvassa 4.1 näkyvät laitteet, jotka ovat kaksi moottoria kytkettyinä taajuusmuuntajiin, kytkin ja Human Machine Interface (HMI) -paneeli. Taajuusmoottorit ja HMI-paneeli ovat kytkettyinä toisiinsa kytkimen kautta, jolloin laitteisto muodostaa alaluvussa 2.3 mainitun tähtitopologian, vaikka laitteiden määrä onkin vähäinen. Topologia mahdollistaa laitteiden lisäämisen vaivattomasti kenttäväylään kytkemällä lisää laitteita kytkimeen, jolloin niitä voidaan ohjata käyttämällä HMI-paneelia.

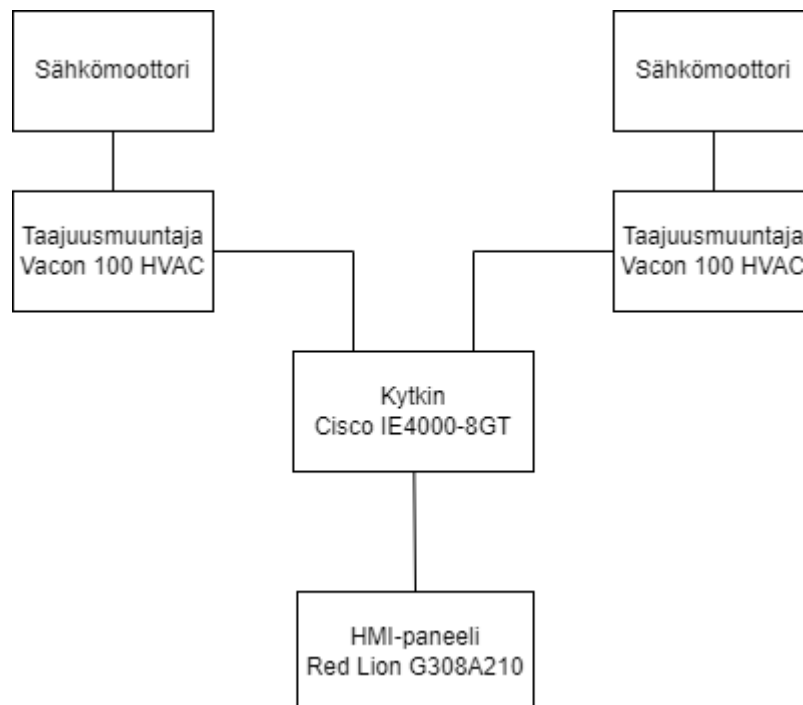


Kuva 4.1. Käytettävät toimilaitteet

HMI-paneelina toimii Red Lionin valmistama graafinen käyttöliittymäpääte G308A210, joka näkyy kuvassa 4.1 keskellä edessä. HMI-paneeli on 8,4-tuumainen, kosketusnäyttöinen paneeli, johon työssä käytettävä logiikka ladataan tietokoneelta. Laitteessa on useita liitännävaihtoehtoja, kuten RS-232, RS-422, RS-485 ja 10Base-T. [12] Tässä työssä tiedonsiirtoon käytetään RS-485 -porttia.

Ohjattavina toimilaitteina ovat kaksi Vacon 100 HVAC -taajuusmuuntajalla varustettua moottoria, jotka ovat kuvassa 4.1 merkattuna simuloimaan vesipumppua ja generaattoria. Taajuusmuuntajat ohjaavat moottorien pyörintää HMI-paneelilta tulevan etäohjauksen tai paikallishjauksen mukaan. Laite on kytkettynä kenttäväylään käyttämällä porttia, joka tukee RS-485 -standardia [13, s. 24].

HMI-paneeli ja taajuusmuuntajat kytkennässä on käytetty Cisco IE4000-8GT -kytkintä. Kytkin on suunniteltu teollisuuskäyttöön nopealla kaistanleveydellä ja monimuotoisilla käyttökohteilla suuren protokollavaliokoman vuoksi [14]. Vaikka kytkin on käyttökyvyiltään huomattavasti tehokkaampi kuin työssä olisi tarvetta, on olennaista, että kytkin tukee Modbus TCP -protokollan käyttöä ja siihen saa kytkettyä useamman laitteen samanaikaisesti. Kuvassa 4.2 on esitettyä laitteiston järjestelmäkaavio, josta nähdään paremmin toimilaitteet ja niiden muodostama väylä, ja myös edellä mainittu tähtitopologia.

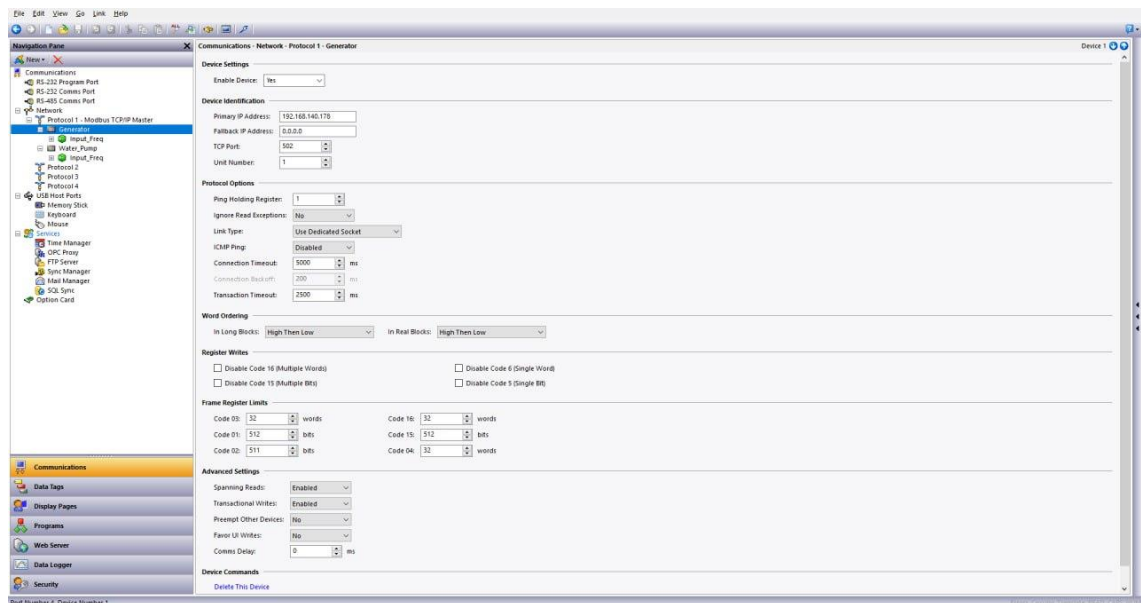


**Kuva 4.2.** Laitteiston järjestelmäkaavio

## 4.2 Ohjelmistoympäristö

Käytännön toteutuksen ohjelmistoympäristönä käytetään Red Lionin omaa Crimson 3.0 -ohjelmaa, joka on suunniteltu käytettäväksi edellä mainitun HMI-paneelin kanssa. Ohjelmassa on sisäänrakennettu tuki Modbus TCP/IP -protokollaa varten helpottaen ympäristön asennusta ja käyttöä. Ohjelman toiminta perustuu drag-and-drop -tyyppiseen käyttöliittymään ja datan sijoittamiseen tunnisteisiin, joiden avulla tieto käsitellään haluttuun muotoon ja osoitteeseen [15].

Ohjelman tärkeimpänä ominaisuutena on, että se tukee yli kolmea sataa eri protokollaa, jonka ansiosta sitä voidaan hyödyntää monenlaisissa käyttökohteissa [16]. Muita hyödyllisiä ominaisuuksia ohjelmalla on sen laaja kirjasto graafisia kuvakkeita, ja mahdollisuus kirjoittaa C-kielellä tarvittavia funktioita [15]. Kuvassa 4.3 nähdään, millainen ohjelmistoympäristö on.



**Kuva 4.3.** Modbus TCP/IP-protokollan asetukset Crimson 3.0:ssa

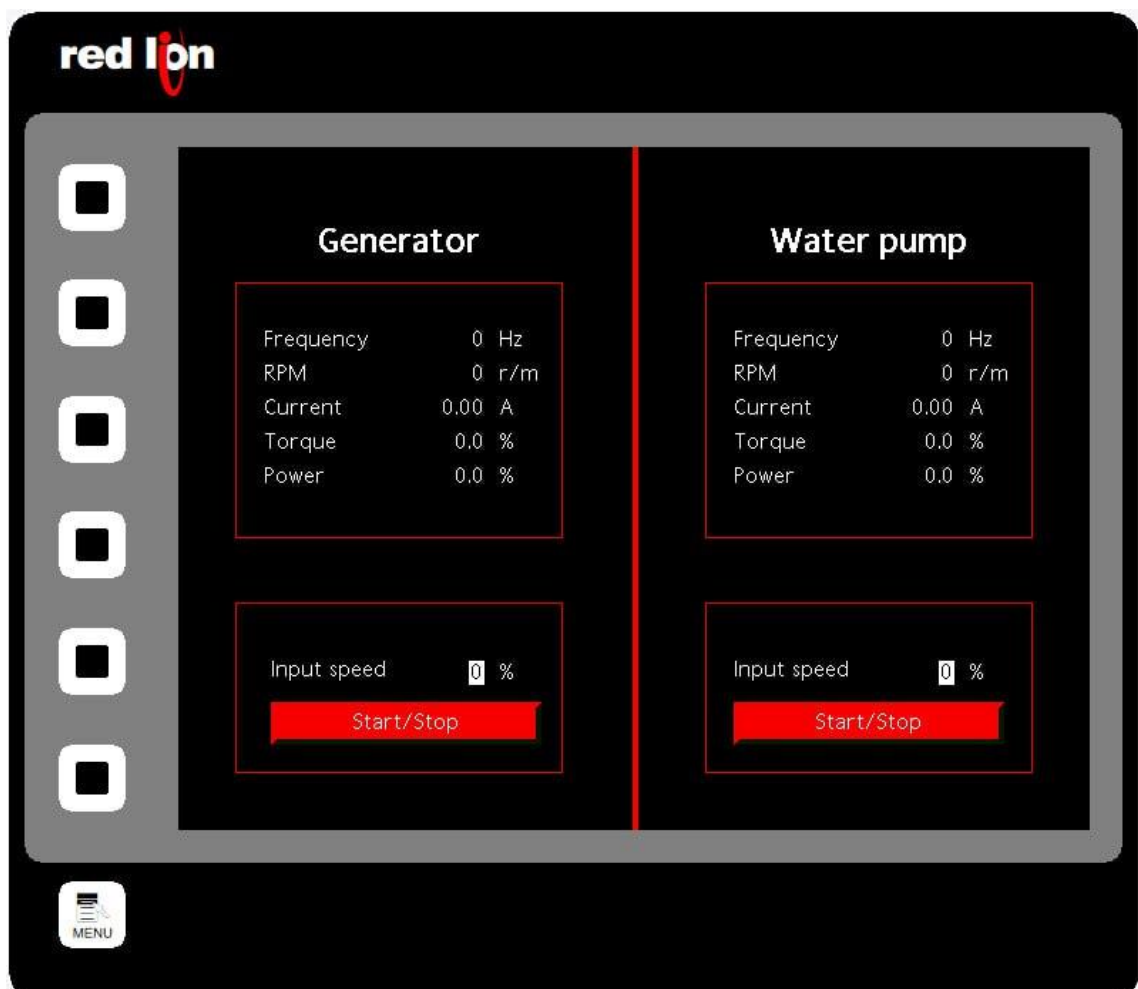
Crimsonissa on kolme välilehteä, joita pääosin tarvitaan työn suorittamiseen. Ensimmäinen niistä on Communications-välilehti, jossa määritetään Modbus-väylän ja porttien asetukset. Lähetettävien ja saapuvien pakettien data on muokattavissa Data tags -lehdellä, jolloin ne asetetaan käytettäviin tunnisteisiin, jotka toimivat muuttujina ohjelmassa. Viimeisenä on käyttöliittymän tekemiseen tarkoitettu Display pages -välilehti.

Crimson 3.0 tarjoaa mahdollisuuden standardin IEC 61131 mukaisen käytön, jolloin muita standardin kieliä voidaan käyttää toimilaitteen ohjelmoinnissa. [15] Täten vanhojen

ohjelmien integrointi onnistuu ilman tarvetta tehdä kaikkea uudelleen. Valtaosa ohjelmoitavan logiikan valmistajista käyttää tätä standardia, kuten myös Red Lionin Crimson -ohjelma [17, s. 12].

### 4.3 Käyttöliittymä

Käyttäjä ohjaa ja seuraa laitteiden toimintaa kuvassa 4.4 esitetyllä käyttöliittymällä, joka on asennettuna edellä mainittuun HMI-paneeliin. Käyttöliittymästä on tarkoituksella tehty mahdollisimman yksinkertainen, jotta väylän toimintaa pystytään seuraamaan helposti, joka on työn päämääränä.



Kuva 4.4. Ohjelman käyttöliittymä

Käyttöliittymä on jaettuna kahteen osaan, koska toimilaitteitakin on kaksi. Ylempiin laati-koissa näytetään laitteista ulostulevaa prosessidataa, joihin kuuluu viittä eri mittaussuu-retta: taajuus, kierrosnopeus, virta, vääntö ja teho. Alemmissa laatikoissa on laitteiden

ohjaus, joita käyttäjä käyttää. Haluttu nopeus asetetaan prosenttilukuna näytölle aukeavan numeronäppäimistön avulla. Painikkeella ”Start/Stop” annetaan toimilaitteelle lupa ottaa kenttäväylää pitkin ohjaus eli tässä tapauksessa käyttäjän asettama nopeusmäärä. Sivulla oleviin nappuloihin ei ollut tässä työssä tarpeellista laittaa toimintoja.

#### 4.4 Laitteiden kommunikointi

Laitteiden välinen kommunikointi tapahtuu Modbus TCP/IP -protokollaa käyttämällä. Laitteiden tärkeimpänä tunnisteena on niiden IP-osoite, jonka avulla oikeat paketit saavuttavat oikean määränpään. Laitteiden IP-osoitteet ovat taulukossa 4.1. Toimilaitteet käyttävät porttia 502, joka on Modbusin oletusporttina. Kuvasta 4.3. nähdään, miten nämä parametrit ovat määriteltynä toimilaitteelle 1 eli generaattorille Crimson-ohjelmaa käyttämällä.

**Taulukko 4.1.** Laitteiden IP-osoitteet

Laite	IP-osoite	Portti
HMI-paneeli	192.168.140.177	
Generaattori	192.168.140.178	502
Vesipumppu	192.168.140.179	502

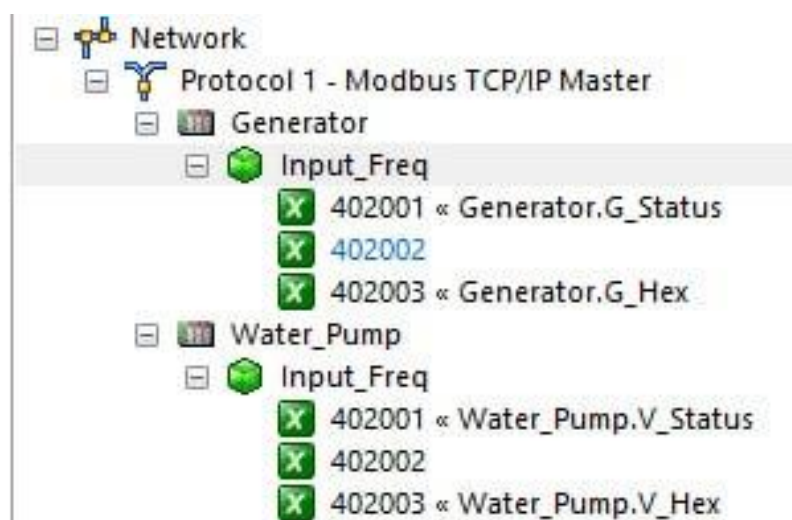
Taajuusmuuntajien asetukset ovat asetettu toimimaan kenttäväylää pitkin laittamalla parametri ”Rem. Ctrl. Place (P1.15)” muotoon ”FieldbusCTRL” [18, s. 18]. Lisäksi ohjaus on tässä tapauksessa asetettu ottamaan vastaan etäohjausta, jotta HMI-paneelilla voidaan lähettää myös haluttu ohjausta. Taajuusmuuntajat toimivat käyttämällä heksadesimaaleja, joten lähetettävät arvot ovat muunnettu oikeaan muotoon.

Molempiin suuntiin lähetettävien pakettien datat ovat tallennettuna säilytysrekistereihin (Holding register), jotka ovat 16-bittisiä arvoja, joihin voidaan sekä kirjoittaa että lukea. Aiemmin esitetyillä funktiokoodilla määritetään, mitä näillä säilytysrekistereille tehdään. [18, s. 30] Vaikka tässä tapauksessa viestien parametrit eivät suoraan vaikuta toisiinsa, on yksinkertaisempaa pitää datatyypit samantyyppisiksi molempiin suuntiin. Vacon taajuusmuuntajat ja Red Lion -paneeli mahdollistavat suuren määrän säilytysrekistereitä, joita voi käyttää, mutta tässä työssä ei tarvita montaa. Käytetyt rekisterit ovat kuvattuna taulukossa 4.2.

**Taulukko 4.2.** Työssä käytetyt säilytysrekisterit [18, s. 49]

HMI-paneeli - > Taajuusmuuntaja	
2001	Käyttölupa (0 tai 1)
2003	Ohjaustaajuus (0–100 %)
Taajuusmuuntaja - > HMI-paneeli	
2104	Ulostulos taajuus (0,01 Hz)
2105	Nopeus (1 rpm)
2106	Virta (0,1 A)
2107	Vääntö (0,1 %)
2108	Teho (0,1 %)

Taajuusmuuntajilta HMI-paneelille tulevien säilytysrekisterien sijoittamisella ei tässä tapauksessa ole merkitystä, koska niitä ei lähetetä mihinkään toimilaitteeseen, jonka takia ne ovat tallennettuna rekistereihin 0001–0005. Taajuusmuuntajassa on muuttujat ovat valmiiksi määriteltynä tunnisteisiin 1–5, jolloin ne tulevat samassa järjestyksessä myös HMI-paneelille. Toisin kuin paneelille tuleva data, paneelilta laitteille lähtevä ohjaus pitää olla asetettuna oikeaan rekisteriin ennen kuin se lähetetään. Tämä on kuvattuna kuvassa 4.5, josta nähdään, miten muuttujien data on siirrettynä oikeisiin rekistereihin Communications-välilehdellä. Numero neljä on rekisterin alussa, koska Crimson-ohjelma käyttää sitä datatyyppin määrittämiseen, joka viittaa siis siihen, että data on säilytysrekisterissä.

**Kuva 4.5.** Communications-välilehti



Kuvassa 4.5 nähdään, miten Modbus TCP/IP -protokolla on toteutettuna tasoittain. Ensin valitaan haluttu protokolla ja määritellään, millainen suhde laitteella on verkkoon. Tässä työssä HMI-paneeli on verkon asiakkaaksi. Sen alle on lisättyä molemmat toimilaitteet, joihin on tallennettuna taulukon 4.1 tiedot. Edellä mainitut säilytysrekisterit ovat laitekoh-  
taisesti palikoina, jolloin ohjelma tietää sen olevan yhtä viestiä. Lopulta ohjelma hoitaa  
lopullisen viestin rakentamisen, mikä tekee toimivan ohjelman tekemisestä yksinker-  
taista, kunhan tietää tarvittavat parametrit.

## 5. YHTEENVETO

Tämän opinnäytetyön käytännön osuus saatiin onnistuneesti tehtyä ja kaikki tavoitteet saatiin saavutettua. Ohjelma välittää kaiken tavoitellun datan laitteiden välillä halutulla tavalla. Ohjelma on toteutettu mahdollisimman dynaamisesti, että uusien laitteiden lisääminen verkkoon olisi tulevaisuudessa helppoa. Sekä muuttujat että kommunikoinnin ominaisuudet ovat jaettuna selkeisiin lohkoihin, jos ohjelman toimintaa tahdotaan tutkia myöhemmin. Käyttöliittymää tarvitsee tietenkin muokata, jos siihen haluaa lisäominaisuuksia tai laitteita.

Työssä toteutettua ohjelmaa voidaan vielä parantaa. Mahdollisia kehityskohteita on ainakin yksi. Ohjelma ei ota huomioon mahdollisia poikkeuskoodeja, koska oletuksena oli, että näin yksinkertaisen järjestelmän ei pitäisi tämän työn aikana aiheuttaa ongelmia. Crimson-ohjelma tarjoaa mahdollisuuden ottaa poikkeukset huomioon. Projektin aikana ilmaantui yksi ongelma. Taajuusmuuntaja ei ollut yhteydessä kenttäväylään, vaan se antoi vikakoodia 53. Vikakoodi 53 tarkoittaa, että kenttäväylässä on tietoliikennevika [13, s. 199]. Tämä vika korjaantui dokumentaation mukaisella toimenpiteellä eli uudelleen kytkemällä taajuusmuuntaja takaisin kytkimeen, eikä toistunut uudelleen. Vika oli siis suurella todennäköisyydellä fyysisissä kytkennöissä, joten se ei ollut olennainen tämän työn näkökulmasta.

Näin saatiin havainnollistettua sekä kenttäväylien että Modbus-protokollan toimintaa käytännössä. Teoria osuuksissa käydyistä asioista oli hyötyä yleisten toiminnallisuuksien ymmärtämisessä, ja varsinkin asiakaslaitteen eli HMI-paneelin parametrien asettelussa. Työssä esiteltyjä ja käytettyjä protokollia on käytössä laajasti teollisuudessa, joten tämän työn avulla saatiin alkukosketusta tiedonsiirtotekniikkaan, jota tullaan myös käyttämään tulevaisuudessa.

## LÄHTEET

- [1] S.K. Sen, *Fieldbus and Networking in Process Automation*, 1st Edition, CRC Press, 2014, 462 p. Saatavissa: <https://doi-org.libproxy.tuni.fi/10.1201/b16891>
- [2] O. Akande, *Industrial Automation from Scratch*, Packt Publishing, 2023, 492 p. Saatavissa: [https://learning.oreilly.com/library/view/industrial-automation-from/9781800569386/?sso\\_link=yes&sso\\_link\\_from=tampere-university](https://learning.oreilly.com/library/view/industrial-automation-from/9781800569386/?sso_link=yes&sso_link_from=tampere-university)
- [3] J.-P. Thomesse, *Fieldbus Technology in Industrial Automation: Proceedings of The IEEE*, vol.93, no.6, 2005, 1073–1101 p. Saatavissa: <https://ieeexplore-ieee-org.libproxy.tuni.fi/stamp/stamp.jsp?tp=&arnumber=1435740>
- [4] C. Dey, S.K. Sen, *Industrial Automation Technologies*, 1st Edition, CRC Press, 2020, 376 p. Saatavissa: <https://doi-org.lib-proxy.tuni.fi/10.1201/9780429299346>
- [5] P. Ciccarelli, C. Faulkner, *Networking Foundations*, John Wiley & Sons, Incorporated, 2004, 367 p. Saatavissa: <https://ebookcentral.proquest.com/lib/tampere/detail.action?docID=267311&pq-origsite=primo>
- [6] R. Zurawski, *Industrial Communication Technology Handbook*, 2nd Edition, BOCA RATON: CRC Press, 2017, 1757 p.
- [7] Modbus Organization Modbus FAQ, Modbus Organization. Saatavissa: <http://www.modbus.org/faq.php>.
- [8] M. Botto-Tobar, A. Díaz Cadena, O. S. Gómez, R. Rosero Miranda, *Advances in Emerging Trends and Technologies: Proceedings of ICAETT 2020*, Cham: Springer International Publishing AG, 2020, 347 p.
- [9] *Modbus Messaging on TCP/IP Implementation Guide V1.0b*, Modbus Organization, 46 p. Saatavissa: <http://www.modbus.org/specs.php>.
- [10] *Modbus Application Protocol Specification V1.1b3*, Modbus Organization, 50 p. Saatavissa: <http://www.modbus.org/specs.php>.
- [11] D. Boswarthick, O. Elloumi, O. Hersent, *The Internet of Things: Key Applications and Protocols*, Somerset: Wiley, 2011, 344 p.
- [12] G308A Data Sheet/Manual PDF, Saatavissa <https://www.redlion.net/files/213/5088>
- [13] *VACON 100 HVAC Sovelluskäsikirja*, Saatavissa <https://files.danfoss.com/download/Drives/Vacon-100-HVAC-Application-Manual-DPD01699K-FI.pdf>
- [14] *Cisco Industrial Ethernet 4000 Series Switches*, Saatavissa: <https://www.cisco.com/c/en/us/products/collateral/switches/industrial-ethernet-4000-series-switches/datasheet-c78-733058.pdf>
- [15] *Crimson® Website*, Saatavissa <https://www.redlion.net/support/software-firmware/red-lion-software/crimson>.

- [16] Crimson@3.0 Website, Saatavissa <https://www.redlion.net/red-lion-software/crimson/crimson-30>.
- [17] K.H. John, M. Tiegelkamp, IEC 61131-3: Programming Industrial Automation Systems Concepts and Programming Languages, Springer, 2010, 390 p. Saatavissa: [https://doi-org.libproxy.tuni.fi/10.1007/978-3-642-12015-2\\_1](https://doi-org.libproxy.tuni.fi/10.1007/978-3-642-12015-2_1)
- [18] VACON 100 Modbus User Manual, Saatavissa <https://files.danfoss.com/download/Drives/Vacon-100-Modbus-User-Manual-DPD00156C-UK.pdf>