

PROSESSIDATAN KERÄÄMINEN PLC-LAITTEELTA PILVIPALVELUUN

OperonWay-digitalisaatioprojekti

Satta Simo

Opinnäytetyö

Sähkö- ja automaatiotekniikka
Insinööri (AMK)

2024

Sähkö- ja automaatiotekniikka
Insinööri (AMK)

Tekijä	Simo Satta	Vuosi	2024
Ohjaaja	Ari Afflekt		
Toimeksiantaja	Operon Group Oy		
Työn nimi	Prosessidatan kerääminen PLC-laitteelta pilvipalveluun		
Sivumäärä	53 + 5		

Tämän opinnäytetyön tavoitteena oli tutkia, miten PLC-prosessidataa voidaan lähettää automaatiokeskukselta pilvipalveluun monitoimittajaympäristössä. Tämä työ tarjoaa yhden ratkaisun datan keräämiseksi laitetasolla. Ratkaisun ei kuitenkaan ole tarkoitus olla ainoa tapa, jolla dataa voidaan kerätä.

Opinnäytetyön teoria käsitteli ratkaisun kannalta tärkeimpien tiedonsiirtoprotokollien eli Modbusin, Profibusin ja Profinetin ominaisuuksia ja toimintaa. Koska työn tavoitteena oli lähettää prosessidataa pilvipalveluun, työssä tutustuttiin tarvittavilta osin myös pilvilaskennan yleisiin määritteisiin ja teoriaan. Lisäksi työssä käytiin läpi tyypillisiä ohjausjärjestelmiin vaikuttavia häiriötekijöitä.

Ohjelmoitavien logiikoiden tavassa käsitellä ja tallentaa dataa oleellinen osa on kenttäväylien toiminta. Niinpä tässä opinnäytetyössä esitetyt ratkaisut perustuvat kenttäväylien ominaisuuksiin ja toimintaan.

Tämän opinnäytetyön tulokset pohjautuvat erityisesti Modbusin käyttämään muuttujien rekisteritaulukkoon ja sen tulkitsemiseen JSON-koodilla. Mikäli samanlainen ratkaisu toteutettaisiin kohteeseen, jonka kenttäväylä ei ole Modbus, laitteisto ja JSON-koodi jouduttaisiin mukauttamaan kohteessa olevaan väylään sopiviksi. Vaihtoehtoinen ratkaisu olisi käyttää väylämuunninta muuntamaan kyseinen väylä Modbusiksi, jolloin voitaisiin käyttää samaa laitteistoa ja koodia.

Tutkimuksen tuloksista eniten hyötyä on kohteissa, joissa datan kerääminen on käsin tapahtuvaa tai muuten laadultaan heikkoa ja joissa toimintaa halutaan kehittää enemmän dataan nojautuvan päätöksenteon suuntaan.

Electrical and Automation
Engineering Bachelor of Engineering

Author	Simo Satta	Year	2024
Supervisor	Ari Afflekt		
Commissioned by	Operon Finland Oy		
Title	PLC process data integration to cloud computing		
Number of pages	53 + 5		

The aim of this thesis was to investigate how to send gathered process data from a PLC device to a cloud service in an environment with many vendors. This work provides a single, usable solution to collect data at the device level as there are many ways to accomplish this.

The theory of the thesis handles the properties of the most important communication protocols, i.e. Modbus, Profibus and Profinet. Since the goal of the work was to send process data to a cloud service, also the general attributes and theory of cloud computing have been familiarized in this work. In addition, typical error factors affecting control systems were reviewed in this work.

An essential part of how programmable logics process and store data is the operation of field buses. Therefore, the solutions presented in this thesis are based on the characteristics and operations of these buses.

The results of this thesis are based especially on the register table of variables used by Modbus and interpreting the tables with JSON code. If a similar solution were to be implemented for a target with different field bus the hardware and JSON code would have to be adapted to fit the target's bus. An alternative solution would be to use a bus converter for the bus in question, in which case the same hardware and code could be used.

The results of the research are most useful in areas where data collection is done manually or is otherwise of poor quality and where there is a desire to develop operations more in the direction of data-based decision-making.

Keywords Data management, programmable logics, cloud computing

SISÄLLYS

1	JOHDANTO	7
2	TARKOITUS, PÄÄMÄÄRÄT JA TAVOITTEET	8
3	DATAN KERÄÄMINEN PLC-LAITTEELTA PILVEEN.....	9
3.1	Datan kerääminen	9
3.2	Tunnisteet ja kenttäväylän asettaminen.....	10
3.3	Datan lähettäminen.....	10
3.3.1	Pilvipalvelut	10
3.3.2	Pilvipalvelun hyödyt.....	11
3.3.3	Pilvipalvelun haitat.....	11
3.3.4	IoT-yhdyskäytävän yhdistäminen pilveen	12
3.4	Tiedonsiirtoprotokollat.....	13
3.4.1	Modbus.....	14
3.4.2	Profibus	16
3.4.3	Profinet.....	20
3.5	Protokollan valintaan vaikuttavat tekijät.....	22
3.6	Häiriölähteet ohjausjärjestelmissä	22
4	PLC-PILVIPALVELUN LINKITYKSEN SUUNNITTELU.....	24
4.1	Automaation lähtötilanne	24
4.2	Projektin vaatimukset.....	24
4.3	Tietoliikenneyhteyden osat	25
5	TIETOLIIKENNEYHTEYDEN TOTEUTUS	27
5.1	Prosessiautomaatiokeskus	27
5.2	PLC-linkki	28
5.3	PLC-linkin asennus.....	29
5.4	IoT-yhdyskäytävä	32
5.4.1	RUT-yhdyskäytävän asennus.....	33
5.4.2	Käyttöönotto	34
5.5	DMZ.....	39
5.6	Pilvipalvelu.....	39
5.7	Toteutetun PLC-linkin vaatimustenmukaisuustarkastelu	40
6	ONGELMANRATKONTA JA VIANSELVITTELY	44

7 TULOKSET JA TARKASTELU	48
7.1 Datan keräys ja lähetys	48
7.2 Linkin käytännön toteutus ja laajennettavuus	48
8 POHDINTA.....	49
LÄHTEET.....	51
LIITTEET	53

KÄYTETYT MERKIT JA LYHENTEET

API	ohjelmarajapinta, Application Programming Interface
DMZ	demilitarisoitu alue, Demilitarized Zone
DoS	palvelunestohyökkäys, Denial of Service
IoT	esineiden internet, Internet of Things
IP	internetprotokolla, Internet Protocol
MQTT	viestijonon telemetrian siirto, Message Queue Telemetry Transport
PLC	ohjelmoitava logiikka, Programmable Logic Controller
SSL	porttien salausprotokolla, Secure Slots Layer
TCP	tietoliikenneprotokolla, Transmission Protocol
VPN	virtuaalinen, yksityinen verkko, Virtual Private Network

1 JOHDANTO

Tässä opinnäytetyössä käsitellään prosessiautomaation digitalisaatioprojektia. Tarkoituksena on luoda kokonaisratkaisu, jolla voidaan tuottaa prosessidataa pilvipalveluun, josta datan analysointi eri sovellusten avulla on mahdollista. Nykyisellään monen pienen jätevedenpuhdistamon prosessidatan keräys on vaatimatonta, ja pahimmillaan data voi olla jopa täysin käsin keräyksen varassa. Suurinta osaa pienistäkin puhdistamoista ohjataan kuitenkin ohjelmoitavien logiikoiden avulla, joita voidaan käyttää myös prosessidatan keräykseen ilman suuria muutoksia. Tämän opinnäytetyön tarkoituksena on esittää yksi tapa, jolla automatisoitu prosessidatan keräys voidaan toteuttaa.

Vaikka prosessidatan kerääminen ja tallentaminen tietokantaan ei välttämättä tarvitse pilvipalvelua – data kun olisi mahdollista tallentaa myös paikalliselle palvelimelle –, tässä työssä lähtökohdaksi on asetettu nimenomaan datan vienti pilveen. Työssä tutkitaan teoriaa siitä, miten dataa yleensä kerätään teollisuusympäristöissä. Koska prosessia ohjaavien laitteiden, kuten antureiden, toimilaitteiden yms., tiedonkeräys on lähestulkoon synonyymi kenttäväylän käytölle, teoriaosassa tutustutaan kenttäväylien ominaisuuksiin ja toimintaperiaatteisiin. Kenttäväylät ovat jo itsessään varsin laaja aihe, minkä vuoksi tässä työssä keskitytään tarkastelemaan muutamaa yleisesti käytössä olevaa väyläprotokollaa.

Opinnäytetyön mukainen datan keräys tehdään pienelle jätevedenpuhdistamolle, jossa on käytössä rinnakkaissaostusprosessi. Koska työssä keskitytään prosessidatan pilvitallennukseen automaation ja laitetekniikan kannalta, sitä on mahdollista hyödyntää sellaisenaan missä tahansa vastaavassa ympäristössä toteutettavassa tiedonkeräyksessä.

Työ suoritetaan monitoimittajaympäristössä, mutta tässä opinnäytetyössä eri toimittajiin viitataan vain niiden toimialan perusteella.

2 TARKOITUS, PÄÄMÄÄRÄT JA TAVOITTEET

Tämän opinnäytetyön tarkoitus on tutkia, miten PLC-laitteelta saatavaa prosessidataa siirretään pilvipalveluun, ja toteuttaa käytännön ratkaisu todellisella laitteistolla. Työn rajaus keskittyy pienen, alle $250 \text{ m}^3/\text{n/d}$ tulovirtaaman rinnakkaissaostuslaitoksen prosessidatan keräykseen. Soveltaen tätä opinnäytetyötä voisi kuitenkin käyttää myös muunlaisissa prosesseissa, joiden kokoluokka ja muut olosuhteet ovat samankaltaiset.

Työn päämääränä on suunnitella käytännön ratkaisu, joka olisi hyvin skaalautuva ja jota tulisi voida käyttää mahdollisimman vähäisin muutoksin myös muissa saman kokoluokan tai hieman laajemmissa ympäristöissä.

Opinnäytetyössä käydään läpi, miten tiedonsiirtolaite tai -laitteisto on suunniteltu sekä miten se asennetaan ja otetaan käyttöön. Lisäksi keskeisenä tutkimusaiheena on, miten toteutetaan kustannustehokas prosessidatan kerääminen PLC-laitteelta pilvipalveluun monitoimittajaympäristössä. Se ei saisi vaatia suuria muutoksia prosessia ohjaavaan logiikkaan, ja sen tulisi olla konseptiltaan mahdollisimman helppo soveltaa myös muissa automaatiokohteissa.

3 DATAN KERÄÄMINEN PLC-LAITTEELTA PILVEEN

3.1 Datan kerääminen

Ohjelmoitavan logiikan data on informaatiota, joka kerätään digitaalisista tai analogisista tuloista ja jota käsitellään, tallennetaan ja muokataan PLC:ssä. Tätä dataa PLC käyttää tulojen valvontaan, ohjelman suorittamiseen ja lähtöjen ohjaamiseen (Trout Software 2023).

Normaalisti data kirjoitetaan uudelleen jokaisella ohjelmakierrolla, mutta se voidaan myös tallentaa PLC:n käyttömuistiin. Tällöin dataa voidaan lähettää ohjaus- tai tiedonsiirtotarkoituksessa muille samassa verkossa oleville laitteille, jotka käyttävät samaa tiedonsiirtoprotokollaa. (Fogg 2022.)

PLC-laitetasolla tuotetun prosessidatan lukeminen voidaan toteuttaa kahdella tavalla: siihen voidaan käyttää joko standardoitua protokollaa, kuten MQTT:tä, tai IoT-yhdyskäytävää. (Fogg 2022.)

Standardoitu pilviprotokolla toimii yleensä TCP/IP-ympäristössä, mutta sitä voidaan käyttää missä tahansa kaksisuuntaisessa yhteydessä. Protokolla mahdollistaa viestinnän anturin tai PLC-laitteen ja välittäjän välillä. Välittäjä toimii yhdyskäytävänä, joka vastaanottaa kaikki viestit ja ohjaa ne oikeaan kohteeseen, kuten pilveen tai toiselle laitteelle. (Fogg 2022.)

Standardoidun protokollan käytön etuna on, että se on hyvin määritelty ja säännönmukainen ja mahdollistaa yksinkertaisen siirron pilveen. PLC:n määrittäminen pilviprotokollaan voi kuitenkin olla hankalaa. Lisäksi tiedot voivat kadota, jos yhteys jostakin syystä katkeaa. (Fogg 2022.)

Useat kenttäväylät, kuten Modbus, eivät sovellu suoraan yhdistettäväksi pilveen, sillä niiden avoimuus aiheuttaa tietoturvallisuusriskin. Sen sijaan käytetään IoT-yhdyskäytävää, joka kerää datan PLC- ja muilta laitteilta ennen pilvipalveluun lähettämistä. (Fogg 2022.)

Yhdyskäytävän käyttö tuo mukanaan myös puskurointikyvyn, joten data ei katoa yhteyskatkoksen vuoksi vaan se voidaan välitallentaa, kunnes se on saatu lähetettyä pilveen. (Fogg 2022.)

IoT-reitittimen valinta perustuu sekä käytettävään kenttäväylään että valittuun pilvipalveluun, joten on varmistuttava siitä, että yhdyskäytävä tukee käytössä olevaa väylää ja pilven vaatimaa protokollaa (vNode 2021).

3.2 Tunnisteet ja kenttäväylän asettaminen

Datan kerääminen edellyttää, että jokaiselle datapisteelle, joka halutaan kerätä raakadatasta, on määriteltävä tunnisteet tai tagit. Nämä tunnisteet voivat sisältää esimerkiksi erilaisten antureiden syötteitä, kuten lämpötilan, kosteuden tai pinnankorkeuden, tai käyntitiedon. Jotkin laitetoimittajat tarjoavat tähän nopeita ratkaisuja, mutta yleensä tagit on määriteltävä manuaalisesti. (Fogg 2022.)

PLC- tai muut laitteet, jotka lähettävät mittausdatan, on asetettava valmistajan ohjeiden mukaisesti. Tämä tarkoittaa useimmiten kenttäväylän konfiguroimista. Pilveen lähetettävän datan on oltava standardoitua. Taulukko 1 nähdään esimerkki datan standardoimisesta. (Fogg 2022.)

Taulukko 1. Datan standardoinnilla tarkoitetaan sen yhdenmukaista nimeämistä (Fogg 2022)

DATA POINT	MACHINE A	MACHINE B	STANDARDIZED DATA
Date	December 27.2015	12/27/2015	12/27/15
Part Count	Part_cl	Part:count	PartCount
Machine Alarm	estop	Alarm:Estop	EmergencyStop

3.3 Datan lähettäminen

3.3.1 Pilvipalvelut

Yleisesti pilvipalveluilla tai pilvilaskennalla (cloud computing) tarkoitetaan palveluiden jakamista käyttäjille verkon välityksellä. Pilvipalvelun käytön keskeisimpiä hyötyjä on joustavuus käyttöliittymän ja resurssien hallinnassa eri käyttäjätyypeille (Rouhiainen 2015, 11). Resursseja ovat esimerkiksi hallinta ja lukuoikeudet.

Pilvipalveluita on kolmea päätyyppiä: yksityinen pilvi, julkinen pilvi ja hybridi, joka on kahden edellisen yhdistelmä. Yksityinen pilvi on esimerkiksi yhden yrityksen omistuksessa oleva palvelin. Fyysisesti palvelin voi sijaita kyseisen yrityksen tiloissa. Julkinen pilvi on kolmannen osapuolen tarjoama palvelu, jota voivat käyttää useammat käyttäjät samanaikaisesti. Hybridi yhdistelee yksityistä ja julkista pilvipalvelua sellaisen tekniikan avulla, joka mahdollistaa tiedonsiirron niiden välillä. (Rouhiainen 2015, 11; Azure 2023.)

3.3.2 Pilvipalvelun hyödyt

Pilvilaskennan käyttö voi alentaa kustannuksia, sillä taustalla olevasta infrastruktuurista vastaa palveluntarjoaja. Pilvilaskennan palvelimet mielletään yleisesti turvallisemmaksi ratkaisuksi kuin esimerkiksi yrityspalvelinkeskus, koska se on pilvipalveluntarjoajan pääliiketoimintaa. Pilvipalvelutarjonnan keskittämisen periaatteella voidaan sitoa enemmän resursseja, koska keskittäminen suuriin datakeskuksiin nähdään kustannustehokkaampana ratkaisuna. (Preimesberger 2021.)

Pilvi on myös maailmanlaajuinen, erinomaisesti skaalautuva ja helposti saatavilla, mikä vähentää ohjelmistosovellusten luomiseen ja käyttöönottoon kuluva aikaa. Periaatteessa pilvipalvelun avulla käytettävissä on aina riittävästi tilaa datalle. Tarvittaessa myös lisätilan käyttöönotto on vaivattomampaa ja edullisempaa hankintakulujen osalta kuin esimerkiksi fyysisen tallennustilan lisääminen paikalliseen palvelimeen. Lisäksi tiedostojen varmuuskopiointi on pilviympäristöissä yleensä automaattista, joten käyttäjän ei tarvitse huolehtia siitä. (Preimesberger 2021.)

3.3.3 Pilvipalvelun haitat

Vaikka pilvilaskennan käyttö tarjoaa merkittäviä etuja, se tuo myös omat haasteensa ja vaatimuksensa. Niistä ilmeisin on vaatimus päästä internetiin koko palvelun käytön ajan. Myös datan tallennus- ja tuhoamissäännöt voivat aiheuttaa ongelmia. Onkin tärkeää tutustua kunnolla palveluntarjoajan sääntöihin ja asetuksiin, sillä pahimmassa tapauksessa datan tallentaminen voi olla helppoa mutta sen palauttaminen ei. (Preimesberger 2021.)

Pilvipalveluntarjoajan jakamiseen liittyy tiettyjä riskejä, kuten luvaton pääsy väärin pääsynvalvontatoimintojen kautta ja valtuustietojen väärinkäyttö. Tutkimusten mukaan luvaton käyttö ja turvattomat API:t ovat ykkössijalla pilven suurimpana yksittäisenä haavoittuvuutena. Muita ovat esimerkiksi internetprotokollan ja tietojen palautuksen haavoittuvuudet, laskutuksen kiertäminen, toimittajan tietoturvariskit sekä vaatimustenmukaisuus- ja juridiset riskit. (Preimesberger 2021.)

Käyttämällä kolmannen osapuolen datapalvelinta asiakas antaa tiedot kyseisen palveluntarjoajan haltuun, jonka tehtävänä on varmistaa, että asiakkaan tietoja ei hävitetä tai vuodeta ulkopuolisille. Tietojen häviämällä tarkoitetaan arkaluonteisten tietojen ei-toivottua poistamista joko tietojärjestelmävirheen tai tietomurron seurauksena. Tietovuodot taas ovat arkaluonteisten tietojen luvatonta paljastumista esimerkiksi digitaalisen ympäristön haavoittuvuuksien kautta. Tietovuodot tapahtuvat yleensä verkon ja sähköpostin kautta, mutta niitä voi tapahtua myös mobiilitietojen tallennuslaitteiden, kuten optisten tietovälineiden, USB-avaimien ja kannettavien tietokoneiden, kautta. (Preimesberger 2021.)

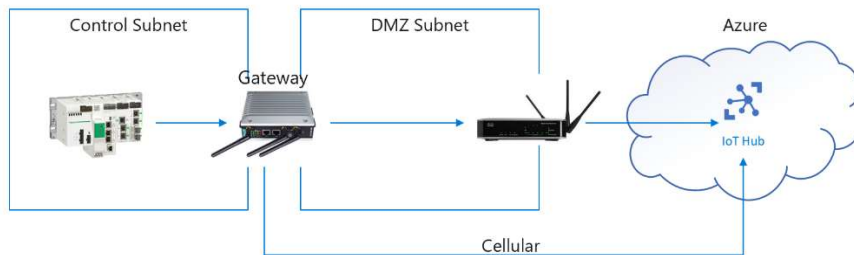
Palveluntarjoaja voi myös joutua palvelunestohyökkäyksen kohteeksi. Tällöin tarkoituksena on sammuttaa palvelin tai verkko, jolloin käyttäjät eivät pääse sen resursseihin. DoS-hyökkäys toteutetaan kuormittamalla kohdetta sellaisella liikenteellä, joka aiheuttaa ”kaatumisen”. Pilvipalvelun käyttäjillä ei ole juurikaan mahdollisuutta vaikuttaa DoS-hyökkäyksiin. (Preimesberger 2021.)

3.3.4 IoT-yhdyskäytävän yhdistäminen pilveen

Kun datapisteiden tagit ja laitteiden kenttäväyläasetukset on asetettu, IoT-yhdyskäytävä voidaan yhdistää internetiin, jonka kautta on pääsy pilvipalvelimelle. Tietoturvaa voidaan parantaa siten, että pääsy laitteen hallintasivuille ja dataan on palomuurin takana. Lisäksi voidaan järjestää etähallinta esimerkiksi VPN-yhteydellä.

IoT-reitittimen yhteyden kannalta tärkeimmät asetukset ovat pilvipalvelun asetukset. Esimerkiksi Azure käyttää URL-muotoista pääsyavainta, joka on

määritettävä reitittimelle (Fogg 2022). Kuvio 1 on esitetty datalähetyksen yksinkertaistettu periaate IoT-yhdyskäytävää käyttäen.



Kuvio 1. Datan lähetys pilvipalveluun yksinkertaistettuna (Mandin 2019)

3.4 Tiedonsiirtoprotokollat

Ohjelmoitavien logiikoiden ja kenttälaitteiden välisessä viestinnässä kommunikointi pohjautuu erilaisiin tiedonsiirtoprotokolliin eli kenttäväyliin. Protokollia on useita erilaisia, ja niiden ominaisuudet riippuvat fyysisestä ja protokollakerroksesta. Esimerkiksi CANopen ja DeviceNet käyttävät samaa laitteistoa mutta eroavat protokollatasolla. Toisaalta Modbus ja Modbus/TCP käyttävät samaa protokollaa, vaikka tiedonsiirto tapahtuu erilaisilla medioilla. (Nanotec 2023.)

Tarjolla olevien järjestelmien ominaisuudet vaihtelevat suuresti. Vanhemmat protokollat poistuvat hitaasti, sillä laitteiden käyttöikä on pitkä. Lisäksi osa protokollista on suoraan sidottu laitevalmistajaan, kun taas toiset väylät pyrkivät yhteensopivuuteen eri valmistajien laitteiden välillä. (Inst Tools 2023.) Yleisiä käytössä olevia kenttäväyliä ovat esimerkiksi Profinet, Profibus ja Modbus (Rouhiainen 2015, 7).

Taulukko 2 on vertailtu yleisiä kenttäväyliä, niiden ominaisuuksia ja tyypillisiä käyttökatteita. Kuten Taulukko 2 huomataan, valittavina on lukuisia erilaisia väyliä. Tässä työssä keskitytään kuitenkin edellä mainittuihin kolmeen tunnetuimpaan väylään.

Taulukko 2. Kenttäväylien ominaisuuksia (Inst Tools 2023)

Fieldbus Protocol	Speed	Max Distance	Max Nodes	Topology	Use Cases
Modbus	31.25 kbit/s to 1 Mbit/s (RS485)	1.2 km (RS485), Depends on media (TCP/IP)	247 (RS485), 253 (TCP/IP)	Bus	Wide range of industrial control systems
Profibus	9.6 kbit/s to 12 Mbit/s	100 m to 9.6 km (depends on speed)	127	Tree, Star	Factory automation, process automation
DeviceNet	125 kbit/s to 500 kbit/s	500 m (at 125 kbit/s)	64	Trunk/Drop	Automotive industry, industrial automation
EtherCAT	100 Mbit/s	100 m (between devices)	65535	Line	Real-time industrial control systems
Ethernet/IP	10 Mbit/s to 1 Gbit/s	100 m (Copper Ethernet), up to 40 km (Fiber Optic)	Unlimited	Star, Tree, Line	Various automation and process control applications
CAN (Control Area Network)	50 kbit/s to 1 Mbit/s	40 m (at 1 Mbit/s) to 1 km (at 50 kbit/s)	127	Bus, Star	Automotive, Industrial control systems
Profinet	100 Mbit/s to 1 Gbit/s	100 m (Copper Ethernet), up to 40 km (Fiber Optic)	Unlimited	Line, Star, Ring	Industrial automation, process control
AS-Interface (AS-i)	167 kbit/s	100 m (extendable with repeaters)	62	Line, Tree, Star	Simple or binary devices, e.g. switches, sensors
CC-Link	10 Mbit/s	100 m (extendable to 1.2 km with repeaters)	64	Line, Star, Ring	Mitsubishi PLCs, Asian markets
Foundation Fieldbus H1	31.25 kbit/s	1900 m	32	Bus, Star, Tree	Process automation, intelligent devices
SERCOS III	100 Mbit/s	100 m (Copper Ethernet), up to 80 km (Fiber Optic)	511	Ring	Servo drives, Machine tools
POWERLINK	100 Mbit/s	100 m (Copper Ethernet), up to 40 km (Fiber Optic)	240	Line, Ring	Real-time I/O, drives
Modbus-Plus	1 Mbit/s	1.6 km	64	Peer to Peer	Modicon PLCs

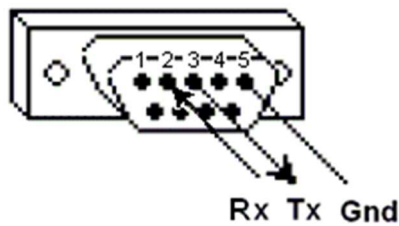
3.4.1 Modbus

Modbus on yksi vanhimmista ja yksinkertaisimmista käytössä olevista kenttäväyläprotokollista. Se kehitettiin vuonna 1979 teollisuusautomaatiojärjestelmiin, erityisesti ohjelmoitavien logiikoiden käyttöön. Nykyisin Modbus on Schneider Electric Groupin tuotemerkki, jota yhtiö myös ylläpitää. (Hersent, Boswarthick & Elloumi 2012, 79–83.) Modbus perustuu yksinkertaiseen master-slave-järjestelmään, joka käyttää tiedonsiirtoon sarjaliikennettä (RS-232/RS-485) tai Ethernetiä (Modbus TCP/IP). (Inst Tools 2023.)

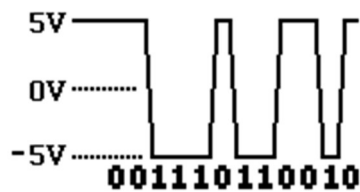
Yksinkertaisuutensa ja sen myötä helppokäyttöisyytensä vuoksi Modbusia käytetään laajalti teollisuuden elektronisten laitteiden liittämiseen. Se soveltuu hyvin analogisen ja I/O-datan siirtoon sekä ohjaus- ja instrumentointilaitteiden välillä tapahtuvan tiedonsiirron rekisteröintiin. (.)

Modbusin toiminta perustuu laitteiden väliseen sarjakaapeleissa tapahtuvaan jännitteen vaihteluun (Kuvio 2). Kuten Kuvio 3 nähdään, eri jännitetasot

vastaavat bitin eri arvoja siten, että 0-bitti lähetetään positiivisena 5 V:n jännitteenä ja 1-bitti negatiivisena jännitteenä (–5 V). (Simply Modbus 2020.)

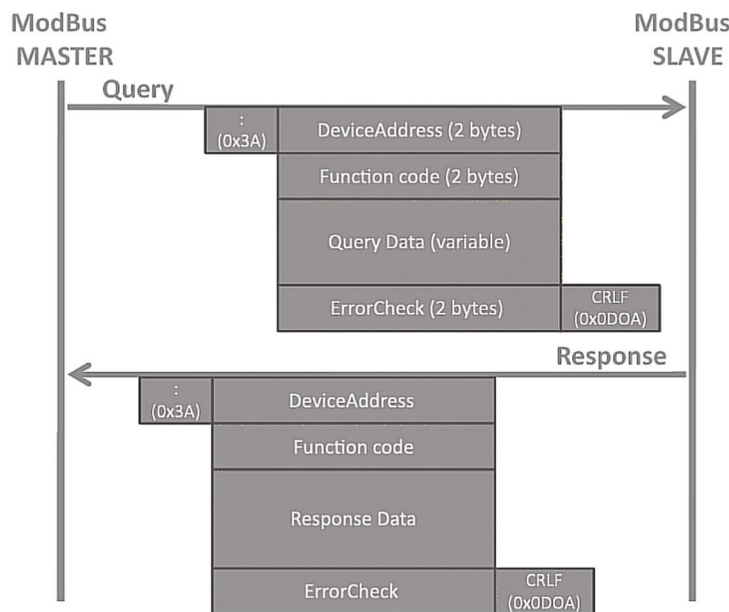


Kuvio 2. Modbus-sarjaliikenneportin periaate (Simply Modbus 2020)



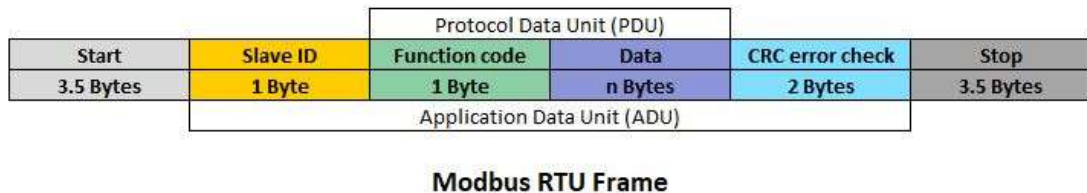
Kuvio 3. Modbusissa käytetyt jännitetasot tulkitaan bitin eri arvoiksi (Simply Modbus 2020)

Modbusin lähetysmoodeja on kahta eri tyyppiä: ASCII- ja RTU-moodit. Kuvio 4 on esitetty ASCII-lähetysmoodin periaate. Sen mukaan jokainen bitti vastaa kahta ASCII-merkkiä, jotka lähetetään erillisinä viesteinä. Aluksi lähetetään aloitusbitti, sen jälkeen 7 bitin dataosio ja mahdollinen pariteettibitti. Niitä seuraa lopuksi yksi tai kaksi lopetusbittiä. (Hersent ym. 2012, 79–83.)



Kuvio 4. Modbusin ASCII-viestin kehysrakenne (Hersent ym. 2012)

RTU-moodissa (Remote Terminal Unit) viesti lähetetään jatkuvana lähetyksenä. Kuten Kuvio 5 nähdään, jokainen 8-bittinen tavu on kehystetty aloitusbitillä, jota seuraavat 8 bitin data ja mahdollinen pariteettibitti. Tämän jälkeen seuraavat lopetusbitit. Viestit erotellaan 3,5 merkin pituisilla ”hiljaisilla” ajanjaksoilla. (Hersent ym. 2012, 79–83.)



Kuvio 5. Modbusin RTU-moodin viestirakenne (Simply Modbus 2020)

Modbus-osoitteisto koostuu neljästä muistitaulusta, joihin erityyppiset data-arvot tallennetaan tai luetaan datatyyppin mukaan. Viestin kehysrakenne on muotoa osoitekenttä – funktiokoodi – data ja virheentarkastus (Kuvio 5).

Taulukko 3 nähdään, kuinka erilaisia osoitteita käytetään. Esimerkiksi osoitteet 00001–09999 on varattu lähdöille ja niitä voidaan sekä lukea että kirjoittaa, kun taas osoitteet 10001–19999 on varattu tuloille ja niitä voidaan ainoastaan lukea. Analogisten tulojen rekisteröintiin käytetään osoitteita 30001–39999, ja osoitteita 40001–49999 voidaan käyttää arvojen rekisteröintiin esimerkiksi datankeräystoiminnoissa. (Simply Modbus 2020.)

Taulukko 3. Modbus-osoitteet (Inst Tools 2023)

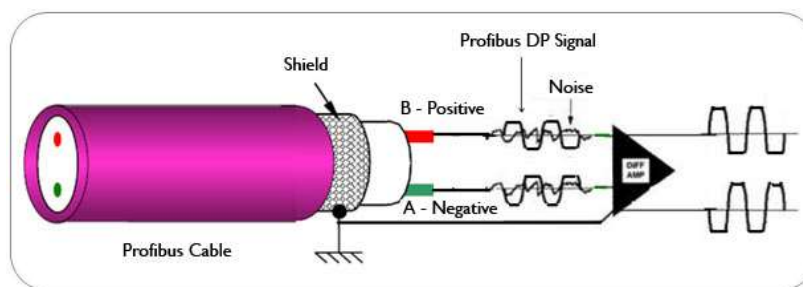
Modbus codes	Address range	Purpose
01, 05, 15	00001 to 09999	Discrete outputs (“coils”), <i>read/write</i>
02	10001 to 19999	Discrete inputs (“contacts”), <i>read-only</i>
04	30001 to 39999	Analog input registers, <i>read-only</i>
03, 06, 16	40001 to 49999	“Holding” registers, <i>read/write</i>

3.4.2 Profibus

Profibus (Process Field Bus) on alun perin Siemensin kehittämä kenttäväyläprotokolla, mutta nykyisellään sitä hallinnoi Profibus & Profinet

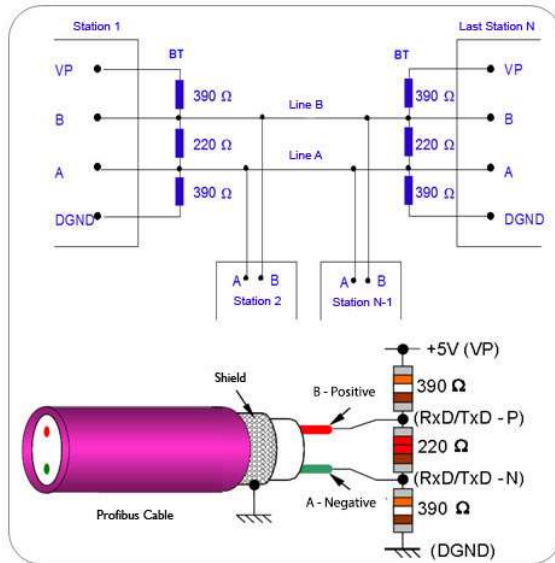
International. Sitä käytetään erityisesti tehdasautomaation sovelluksissa, kuten prosessiautomaatiossa ja erilaisissa liikkeenohjausjärjestelmissä. Profibusista on kaksi pääversiota: Profibus-DP ja Profibus-PA. Profibus-DP on nopeaan tiedonsiirtoon tarkoitettu, Profibus-PA taas erityisesti turvalaitteiden tiedonsiirtosovelluksissa käytetty protokolla. (Cassiolato 2011.)

Profibusissa dataliikenne tapahtuu Kuvio 6 mukaisesti kahden johtimen kautta, jotka toimivat itsenäisinä kanavina. Kanava A on tyypillisesti väriltään vihreä (negatiivinen), -5 – 5 V tai -3 – 3 V. Vastaavasti B-kanava on varattu samansuuruiselle mutta polariteetiltaan vastakkaiselle jännitteelle. Vaikka jännitteet ovat päinvastaiset, johtimet eivät kuulu samaan virtapiiriin vaan jännitteen tasaus tapahtuu kolmannen johtimen tai suojakuoren kautta. Sama signaali kulkee siis molemmissa kanavissa samaan aikaan mutta jännitteen kannalta peilattuna. (Cassiolato 2011.)



Kuvio 6. Profibus-kaapelin rakenne ja toimintaperiaate (Cassiolato 2011)

Signaalin toistaminen vastakkaisella polariteetilla on edullista siksi, että varsinainen signaali peilataan eri kanavissa, kun taas kaapeleissa esiintyvä kohina pysyy identtisenä. Tämän vuoksi oikeasta signaalista voidaan erotella häiriöt ja suodattaa ne pois. Toiminnan kannalta johtimen on oltava suojakuorellinen, ja sen maadoituksesta on huolehdittava. Lisäksi jokaisen Profibus-verkon päässä on oltava päätyvastus (Kuvio 7), joka tasaa ja harmonisoi verkon impedanssia. Tällä tavoin voidaan estää signaalin vääristyminen, jota esiintyy erityisesti pitkillä kaapeleilla. (Cassiolato 2011.)

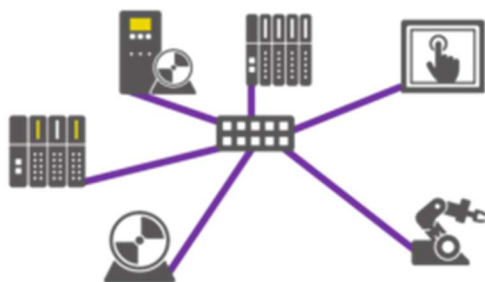


Kuvio 7. Profibus-päätyvastuksen periaate (Cassiolato 2011)

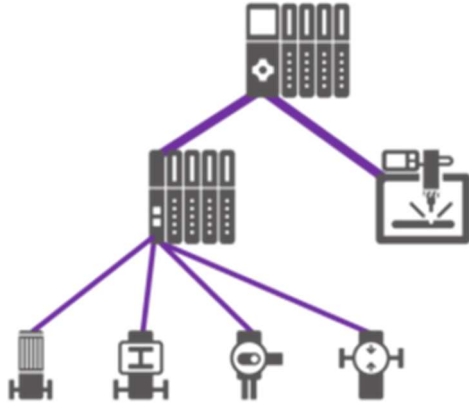
Modbusin tavoin myös Profibus toimii master-slave-periaatteella, mutta Profibus mahdollistaa usean isännän eli masterin käytön. Tämän myötä kenttäväylästä on mahdollista muodostaa erilaisia topologioita, kuten Kuvio 8–Kuvio 10 nähdään. (Inst Tools 2023.)



Kuvio 8. Suora eli linjatopologia (Bowne 2020)



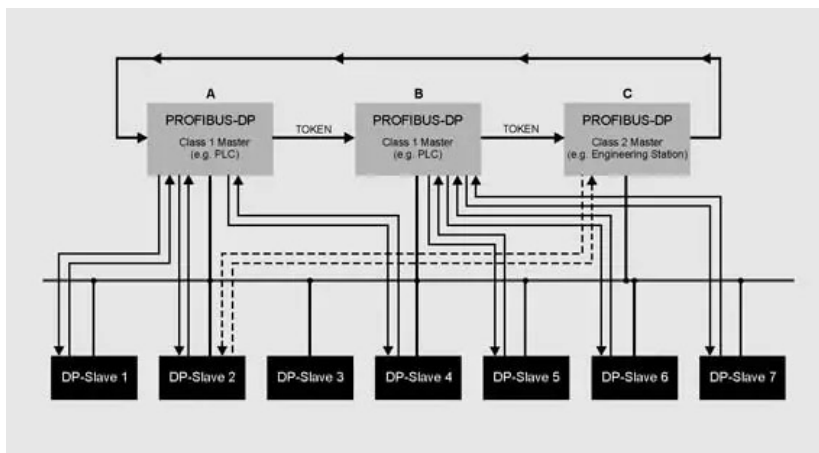
Kuvio 9. Tähtitopologia (Bowne 2020)



Kuvio 10. Puutopologia (Bowne 2020)

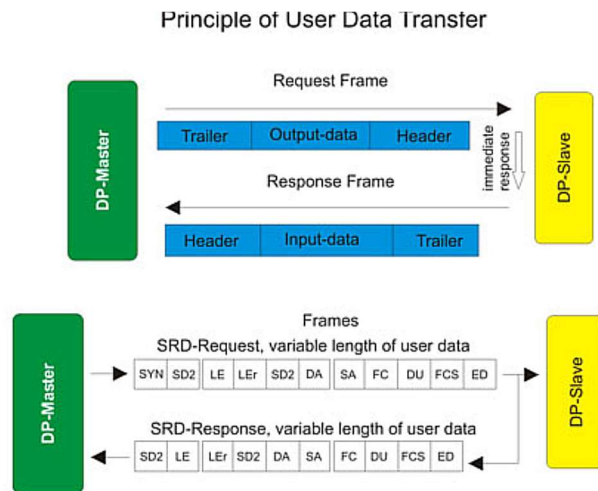
Käynnistysvaiheessa Profibus-laitteet suorittavat sekvenssin, jonka aikana laitteet liittyvät määriteltyyn verkkoon. Profibus-verkossa jokaisella laitteella on ajastin. Laite menee vikasetotilaan, mikäli viestiä masterilta ei saada. Kun verkossa käytetään lisäksi valvonta-ajastinta, aikamääre voidaan varmistaa jokaisella tiedonsiirtosykliä. (Inst Tools 2023.)

Kuviosta Kuvio 11 nähdään, miten useamman masterin Profibus-verkko toimii. Master A saa väylän hallintaoikeuden, jolloin se suorittaa tiedonsiirron kaikkien sille määriteltyjen slave-asemien kanssa vuorotellen. Tämän jälkeen hallinta luovutetaan seuraavalle masterille, joka käy samalla tavalla läpi oman tiedonsiirtonsa ja luovuttaa hallinnan edelleen. Kun kaikki masterit ovat käyneet vuoronsa läpi, hallinta luovutetaan takaisin ensimmäiselle masterille, minkä jälkeen sama prosessi toistuu. (Inst Tools 2023.)



Kuvio 11. Profibus-verkon skannaussykli (Inst Tools 2023)

Profibusin tiedonsiirto on standardimuotoista, joten sekä pyyntö (request) että vastaus (response) noudattavat standardoitua rakennetta. Tämä helpottaa suunnittelua, sillä funktiot ja toiminnot ovat samat laitevalmistajasta riippumatta. (Inst Tools 2023.)



Kuvio 12. Profibus-liikenteen kehysrakenne (Inst Tools 2023)

Kuviossa Kuvio 12 on kuvattu kaksi erilaista kehysmallia, joita käytetään Profibus-viestinnässä. Ylempänä on määrätyn datan kehysmalli, alempana vaihtelevan datamäärän malli. Alemmassa mallissa

- SYN on synkronointiaika
- SD2 on aloitusmerkki
- LE on pituus
- LEr on toistettu pituus
- DA on osoitekenttä (kohde)
- SA on osoitekenttä (lähde)
- FC on toimintokoodi
- DU on data
- FCS on kehyksen tarkistussekvenssi
- ED on loppumerkki.

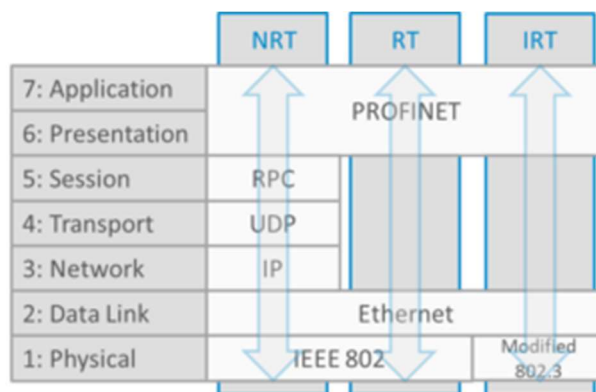
3.4.3 Profinet

Profinet (Process Field Net) on teolliseen Ethernetiin perustuva kenttäväyläprotokolla, joka on suunniteltu tietojen keräämiseen ja hallintaan teollisuusjärjestelmien laitteista. Sen parhaimpia ominaisuuksia on tiedonsiirto tiukkojen aikarajoitusten sisällä. (Inst Tools 2023.) Lisäksi se perustuu Ethernet-teknologiaan, minkä ansiosta kommunikointi on kaksisuuntaista. Tämä tarkoittaa

sitä, että nopeuden kasvun lisäksi topologiamahdollisuuksia on useita, kuten viiva, tähti, puu ja rengas. Lisäksi langattomat yhteydet solmujen välillä ovat mahdollisia. (Bowne 2020.)

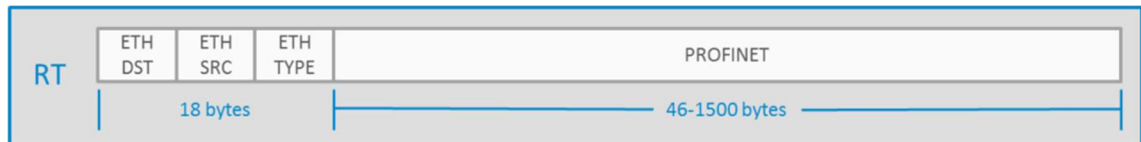
Profinet on rakennettu samalla periaatteella kuin Profibus mutta toisin kuin Modbus TCP/IP, joka on pohjimmiltaan Modbus RTU kapseloituna TCP/IP-paketiksi. Profinet suunniteltiin hyödyntämään Ethernetiä ja mahdollistamaan Profibus-toimintojen, kuten Profisafen, käytön. (Inst Tools 2023.)

Profinet pohjautuu IEEE 802 -protokollaan, joka on synonyymi useiden laitteiden, myös kuluttajalaitteiden, internetyhteydelle. Kuten kuviosta Kuvio 13 nähdään, Profinet käyttääkin tiedon siirtämiseen useampaa protokollaa riippuen siitä, tapahtuuko kommunikointi reaaliajassa (RT, Real Time), isokronisessa reaaliajassa (IRT, Isochronous Real Time) vai ei-reaaliajassa (NRT, Non Real Time). (Harrington, Contreras & Swindal 2023.) Isokroninen reaaliaika ja reaaliaika eroavat toisistaan siten, että isokronisessa reaaliajassa tietyn toiminnon on tapahduttava asetetun aikarajan puitteissa, reaaliajassa taas ennen aikarajaa (Julku 2021).

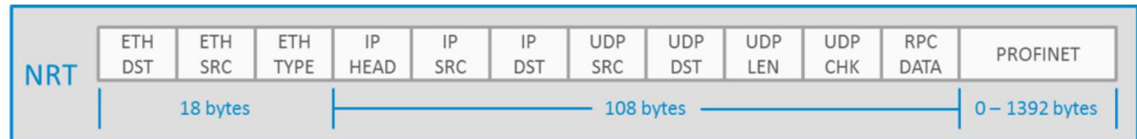


Kuvio 13. Profinetin eri kommunikointikanavat (Harrington ym. 2023)

Viipeiden vähentämiseksi käytettävässä reaaliaikakanavassa ohitetaan kapselointivaiheet verkko-, kuljetus- ja istuntotasoissa. Haittapuolena on, että tällöin ei myöskään käytetä IP-osoitetta, jolloin pakettia ei voida reitittää eri lähiverkkojen välillä. (Harrington ym. 2023.) Kuvioissa Kuvio 14 ja Kuvio 15 on esitetty Profinetin reaali- ja ei-reaaliaikaisen kanavan kehysrakenteet.



Kuvio 14. Reaaliaikaisessa kanavassa on karsittu rakenteita kommunikoinnin nopeuttamiseksi (Harrington ym. 2023)



Kuvio 15. Ei-reaaliaikaisen kanavan kehysrakenteessa käytetään IP- ja UDP-protokollia (Harrington ym. 2023)

3.5 Protokollan valintaan vaikuttavat tekijät

Kenttäväylän valintaan vaikuttavat useat tekijät, sillä jokainen kenttäväylä on kehitetty vastaamaan tiettyyn tarpeeseen. Oikean protokollan valinta ei ole täysin yksiselitteistä, koska valittavina on useita ominaisuuksiltaan toisistaan hyvin poikkeavia vaihtoehtoja. Vaikuttavia tekijöitä ovat teknisten ominaisuuksien lisäksi laitteiden saatavuus, markkina-asema, ohjelmointityökalut ja turvallisuus. (Pyyskänen 2007, 150–153.)

Erityisesti laajennettavuuden kannalta on otettava huomioon asioita, jotka eivät välttämättä ole vaikuttavia tekijöitä juuri sillä hetkellä mutta voivat muuttua merkittäviksi esimerkiksi silloin, jos laitekanta laajenee. Nämä seikat voivat vaikuttaa väylätopologian monimutkaistumisen tai välimatkojen kasvun kautta.

Kaikista valintatekijöistä merkittävin on kuitenkin olemassa oleva väylä. Tämä tarkoittaa sitä, että väylää ei kannata lähteä muuttamaan toiseksi, ellei siihen ole erittäin painavaa syytä. Mikäli kohteessa on käytössä jokin väylä mutta esimerkiksi laitteiden tarjonta ei ole yhteensopiva sen kanssa, voidaan harkita väylämuunnoksia, joihin on olemassa omat laitteistonsa.

3.6 Häiriölähteet ohjausjärjestelmissä

Teollisuusoloissa on laitteita, jotka aiheuttavat sähköisiä ilmiöitä ja altistuvat niille. Tämä voi aiheuttaa ei-toivottuja ominaisuuksia johtimissa ja laitteissa. Pahimmillaan se voi aiheuttaa virhetoimintoja (Keinänen & Sumujärvi 2019, 256).

Liian lähekkäin sijaitsevat kaapelit voivat aiheuttaa kapasitanssin vuoksi häiriövirtaa, jota kutsutaan myös "ylikuulumiseksi". Yleensä ilmiön haittavaikutukset havaitaan pitkällä, yli 100 metrin johtimilla. (Keinänen & Sumujärvi 2019, 256.)

Magneettikenttiä syntyy sähkövirran käämeistä, joita on muun muassa moottoreissa, kontaktoreissa, magneettiventtiilien keloissa, sähkömagneeteissa ja invertterikäytöissä. Niiden aiheuttamia häiriöitä kutsutaan myös transienttihäiriöiksi, ja niitä voidaan estää RC-piirin avulla. (Keinänen & Sumujärvi 2019, 257.)

Säteilyä esiintyy tietoliikenne- ja mittalaitteistoissa. Teollisuusympäristöön suunniteltujen laitteiden säteily pyritään pitämään vähäisenä, joten se ei yleensä aiheuta virhetoimintoja tai häiriötä ohjauksiin. Käyttämällä suojattuja kaapeleita voidaan vähentää säteilyn aiheuttamia häiriöitä. Myös koteloinnilla voidaan vaikuttaa säteilyn vaikutuksiin. (Keinänen & Sumujärvi 2019, 257.)

4 PLC-PILVIPALVELUN LINKITYKSEN SUUNNITTELU

Tehtävänä oli suunnitella laite tai useammasta laitteesta koostuva laitteisto, joka keräisi automaatiokeskuksen PLC-laitteelta saatavan automaatiodatan, muuttaisi sen pilvipalvelun ymmärtämään muotoon ja lähettäisi sen pilvipalveluun. Projektille asetettiin vaatimukset, jotka on määritelty luvussa 4.2.

4.1 Automaation lähtötilanne

Prosessiautomaatiokeskus (Kuvio 17) on prosessia ohjaava kokonaisuus. Sen perustana on PLC-laite, joka kerää tiedot antureilta ja muilta kenttälaitteilta. Prosessiautomaation PLC kerää tiedot I/O- ja analogisista mA- tai jänniteviesteistä.

Projektin lähtötilanteessa automaation prosessidataa ei kerätty riittävän laajasti. Ainoastaan tiettyjen muuttujien, kuten joidenkin pinnanmittausten ja virtausmittareiden, arvoista muodostettuja trendejä voitiin tarkastella jälkikäteen. Lisäksi ennalta määrättyjä raportteja voitiin tarkastella järjestelmästä saatavista .csv-tiedostoista, joiden datapisteet olivat määrätty. Tarkastelun tarkkuus voitiin valita ainoastaan tunti-, vuorokausi-, kuukausi- tai vuositasolla.

4.2 Projektin vaatimukset

Suunniteltavan tietoliikenneyhteyden tuli kyetä yhdistämään prosessiautomaatiojärjestelmä pilviympäristöön. Taulukosta Taulukko 4 nähdään vaatimukset, jotka suunniteltavalle ratkaisulle asetettiin. Vaatimusluokkaa on kahta tyyppiä: vähimmäisvaatimus (VV) ja lisätoive (LT).

Toteutettavan linkin tulisi kyetä lukemaan automaatiosta Modbus-muotoista dataa, muuntaa se pilvipalvelun vastaanottoon soveltuvaan formaattiin ja toimittaa se ennalta määritettyyn päätepisteeseen pilviympäristössä. Pilvipalvelun kautta ei saisi pystyä ohjaamaan prosessia, joten yhteyden automaatiokeskukselta olisi oltava ainoastaan lähetysmuotoista.

Lisäksi linkin tulisi kytkeytyä paikalliseen verkkoon fyysisiä kaapeleita käyttäen. Tietoturvasyistä linkissä ei saisi käyttää langattomia komponentteja.

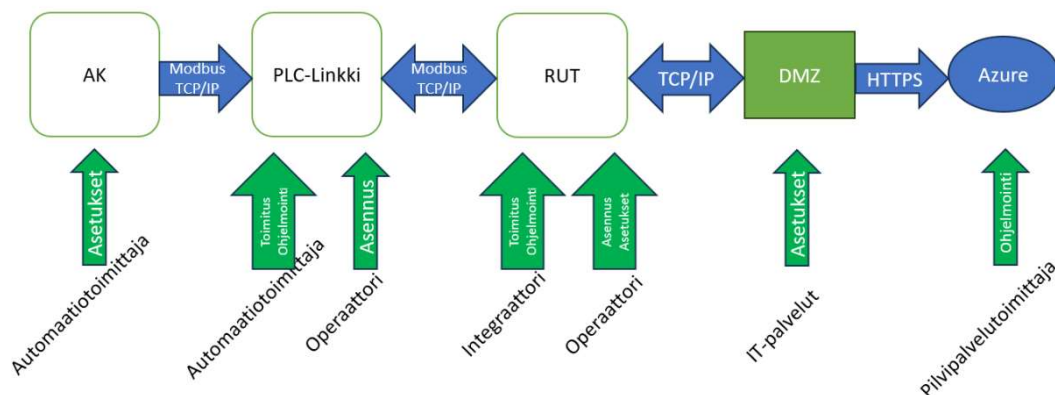
Lisätoiveena oli, että taajuuden, jolla laitteisto lukee dataa prosessiautomaatiosta ja lähettää sitä eteenpäin paketteina pilviympäristöön, tulisi olla käyttäjän säädettävissä. Lisäksi toivottiin, että konsepti olisi helposti siirrettävissä ja laajennettavissa muille laitoksille. Kaikki vaatimukset tulisi toteuttaa kustannustehokkaasti, mutta erityistä ylärajaa kustannuksille ei asetettu.

Taulukko 4. Projektin vaatimuslista

Vaatus	Luokka
Linkin tulee olla yhdensuuntainen (automaatiosta Azureen); linkin kautta ei saa ohjata automaatiojärjestelmää	VV
Linkin tulee kyetä lähettämään data standardimuotoisena (MQTT tms.) viestinä ennalta määriteltyyn end-pointiin Azuren päässä	VV
Linkkiin ei tule päästä käsiksi langattomasti mistään pisteestä; kaiken liikenteen tulee tapahtua laitoksella fyysisiä kytkentöjä pitkin toimistoverkon kautta	VV
Tiedon lähetystaajuuden tulee olla käyttäjän valittavissa	LT
Konseptin tulee olla helposti sovellettavissa toisilla laitoksilla, joiden lähtötiedoista ei ole tietoa (eli kenttäväyläformaatti voi olla jotakin muutakin)	LT
Linkin tulee olla mahdollisimman kustannustehokas	LT

4.3 Tietoliikenneyhteyden osat

Vaatimusten mukainen tietoliikenneyhteys on periaatteeltaan Kuvio 16 mukainen kokonaisuus. Se on toteutettu useasta osasta, joiden toimituksesta vastaavat eri tahot. Kokonaisuuden asennuksesta ja käyttöönotosta vastaa operaattori.



Kuvio 16. Tiedonsiirtolinkin kokoonpano – tietoliikenneyhteys koostuu useista osista

Automaatiokeskus ja toimiston VPN-yhdyskäytävä ovat kohteessa jo valmiina, joten tässä työssä käsitellään ainoastaan niille tarpeellisia muutoksia.

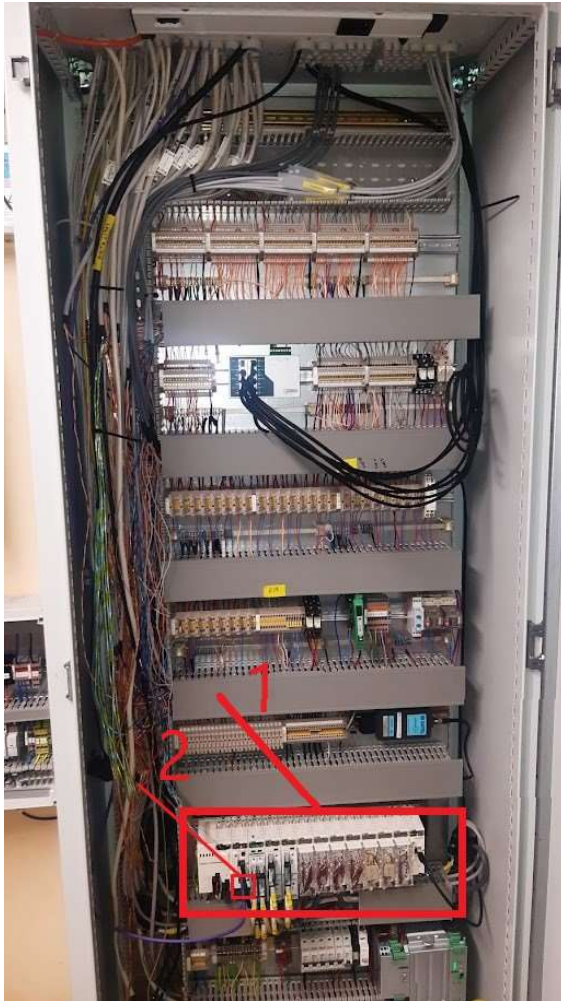
5 TIETOLIIKENNEYHTEYDEN TOTEUTUS

Tässä luvussa kuvataan toteutetun tiedonsiirtoyhteyden laitteita ja osakokoonpanoja. Kuten Kuvio 16 periaatekuvasta nähdään, kokonaisuus koostuu viidestä osasta: prosessiautomaatiokeskus, ohjauskeskus (PLC-linkki), IoT-yhdyskäytävä ja VPN-yhdyskäytävä ja Azure-tietokanta.

5.1 Prosessiautomaatiokeskus

Prosessidatan lähetyksen kannalta PLC:n dataliikenne on määriteltävä valmistajan ohjeiden mukaisesti. Käytännössä tämä tarkoittaa kenttäväylän määrittämistä. Koska kaikki ohjausautomaatiikka on prosessiautomaation toimittajan hallinnassa, tämä toteutetaan tilaustyönä.

Automaatiokeskuksen logiikka, joka on merkitty Kuvio 17 osoitinnumerolla 1, on Schneider Electricin valmistama Modicon-sarjan M340 PLC, joka käyttää Modbus TCP/IP -kenttäväylää, joten protokollan valinta määräytyy sen mukaan. Mikäli haluttaisiin käyttää jotakin muuta protokollaa, tulisi valita sopiva väylämuunnin, jonka avulla muunnos voitaisiin toteuttaa. Modbus-väylän RJ45-liitäntä on osoitettu kuviossa Kuvio 17 numerolla 2.



Kuvio 17. Prosessiautomaatiokeskus

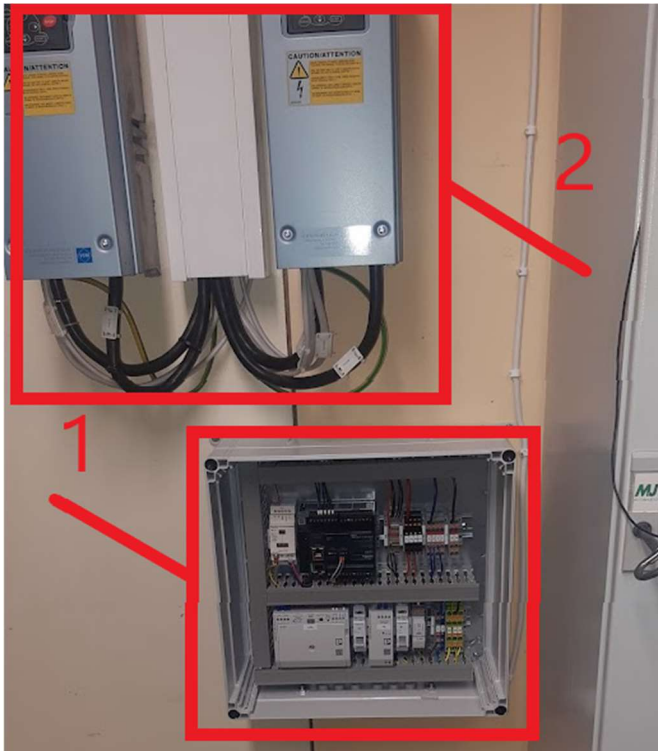
5.2 PLC-linkki

Ohjauskeskus eli PLC-linkki ohjaa automaatiokeskukselta tulleen datan väylän seuraavalle laitteelle eli RUT-yhdyskäytävälle. Lisäksi PLC-laite toimii tietoturvalaitteena muodostamalla yhden lisälaitteen. Sen tarkoitus on estää pääsy RUT-yhdyskäytävältä automaatiokeskuksen PLC-laitteen ohjausdataan.

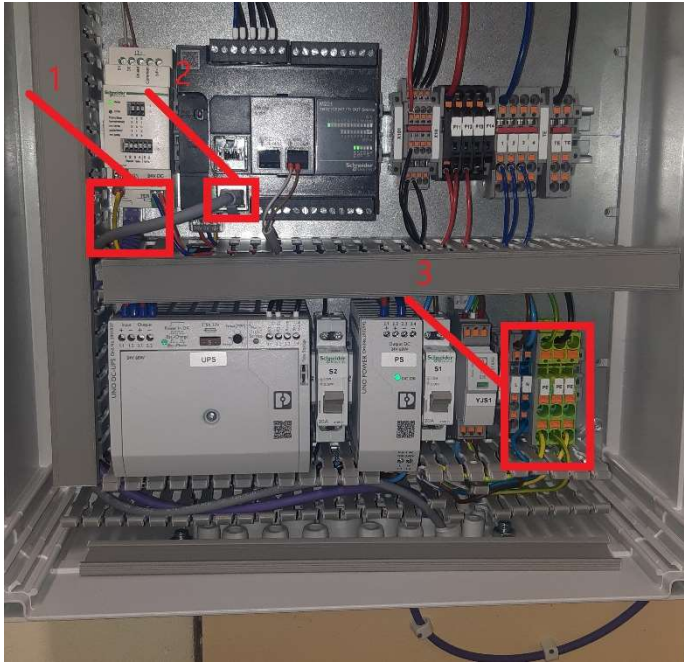
PLC-linkki toimii nimensä mukaisesti siten, että ohjauskeskuksen keskeisenä laitteena on Schneider Electricin valmistama Modicon T221, jossa datan siirtoon käytetään Modbus TCP/IP:tä kuten automaatiokeskuksen M340-mallin logiikassakin. Lisäksi keskuksessa on sarjaliikenneportin eristin (Kuvio 22), joka on malliltaan TWDXCAISO. Keskus on varustettu UPS-laitteistolla, joten lyhyet sähkökatkot eivät haittaa laitteen toimintaa.

5.3 PLC-linkin asennus

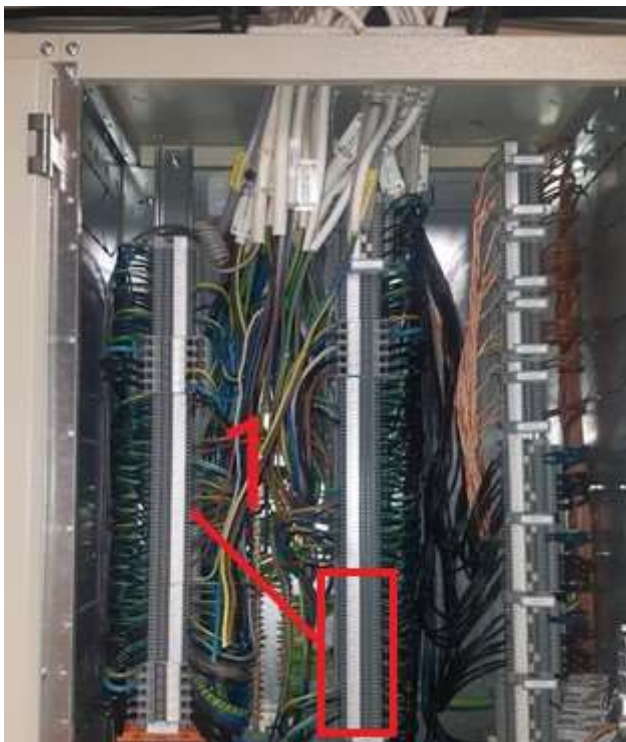
PLC-linkki (Kuvio 18, osoitusviiva 1) asennetaan automaatiokeskuksen ulkopuolelle. Jätevedenpuhdistamolla sopiva asennuskohta löytyi sähkötilasta automaatiokeskuksen vierestä. Riittävästi tilaa on jätettävä taajuusmuuttajille, joiden sijainti on merkitty osoitinviivalla 2. Kuvio 19 näkyvän PLC-keskuksen ohjausjännite otetaan sähkökeskuksen vaiheelta L1 (Kuvio 20). Sähkökuvien (liite 1) mukaisesti ohjausjännitteen johdonsuojakatkaisijan on oltava C6. Se löytyy keskuksen riviliittimeltä X4 (liitin 21). N- ja PE-liittimet on sijoiteltu suoraan vaiheen alapuolelle (Kuvio 19, osoitinviiva 3). Kun johdin on kytketty, johdonsuoja merkitään selvästi (Kuvio 21).



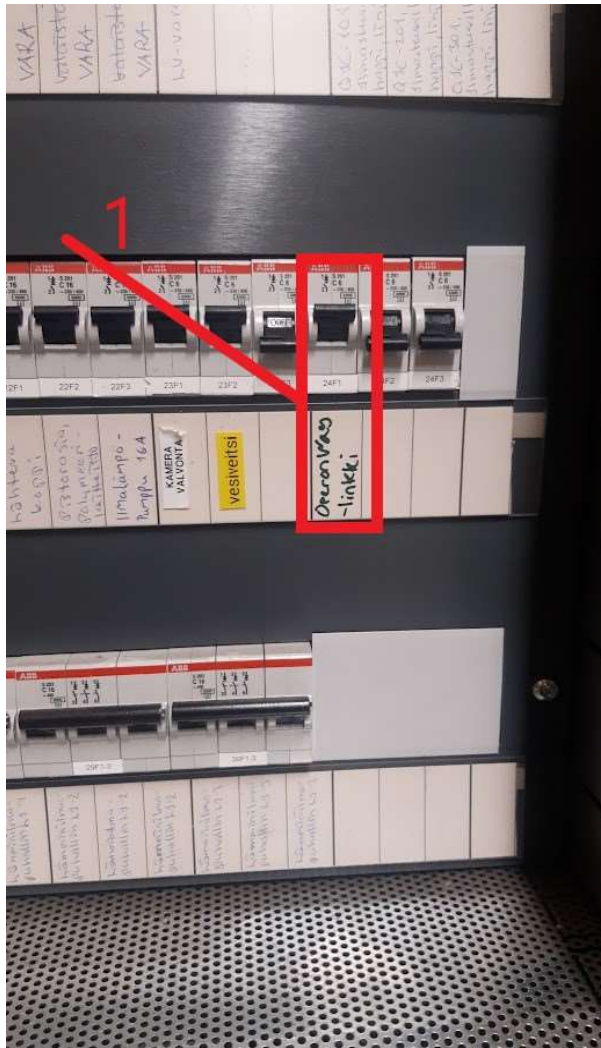
Kuvio 18. PLC-linkin sijoittaminen



Kuvio 19. PLC-linkki; laitteiston jännite on varmennettu UPS-akkuvarmenteella



Kuvio 20. Sähkökeskuksen riviliittimet mahdollistavat helpon, nopean asennuksen



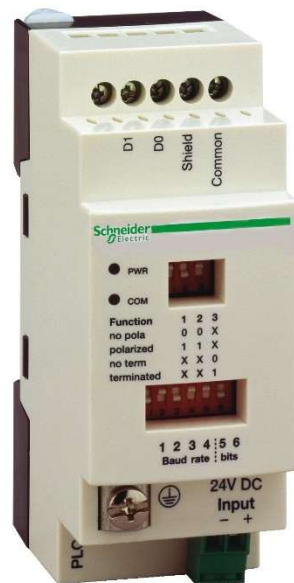
Kuvio 21. Johdonsuojakytkimen merkitseminen

PLC-linkin ja automaatiokeskuksen välinen kommunikointi tapahtuu Modbus TCP/IP -protokollaa käyttäen. Kytkeä tapahtuu RJ45-pääteillä varustetulla Cat 6 -tietoliikennekaapelilla. Kuten Kuvio 23 nähdään, PLC-linkki kytkeytyy prosessiautomaatiokeskukseen osoitinnuolella 1 osoitetun Modbus-portin kautta.

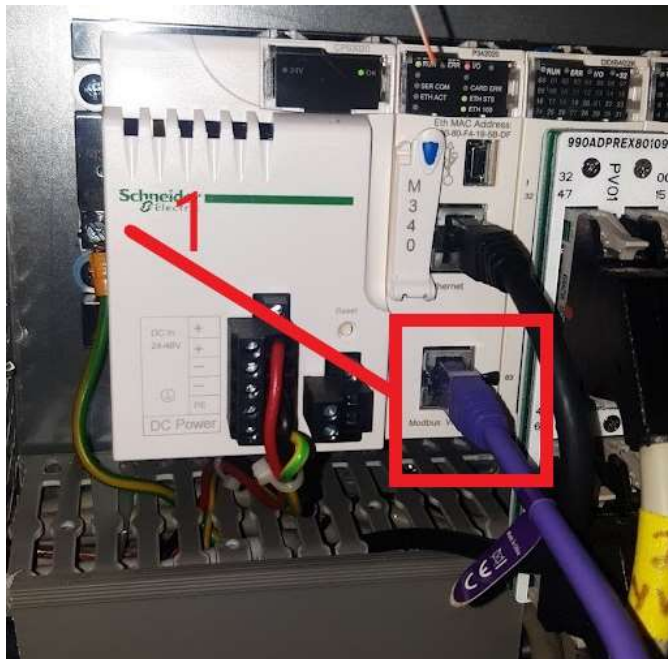
Tässä tapauksessa käytettiin häiriöltä suojaamatonta kaapelia. Kaapelin mitta on 5 metriä, eikä suuria häiriötekijöitä ole lähettyvillä. Niitä olisivat esimerkiksi taajuusmuuttajien kaapelit tai muut voimakkaan sähkökentän aiheuttavat kaapelit, kuten suuret voimakaapelit. Vaativammissa kohteissa suositellaan käytettäväksi Cat 6A -kaapelia.

Datakaapelia ei kytketä suoraan logiikoiden välille, vaan niiden välissä on Kuvio 22 esitetty Schneider Electricin TWDXCAISO-eristin. Kyseessä on

tiedonsiirron valvontaan ja tarkkailuun tarkoitettu lisälaite, joka lisäksi parantaa tietoturvaa eristysominaisuutensa vuoksi.



Kuvio 22. TWDXCAISO-tiedonsiirtolisälaite (Schneider Electric, 2023)



Kuvio 23. Automaatiokeskuksen logiikkaan kytkeydytään Modbus-liittimen kautta

5.4 IoT-yhdyskäytävä

IoT-reitittimeksi valittiin RUTX08. Sen tehtävänä on muuntaa PLC-linkiltä tuleva Modbus-data TCP/IP-liikenteeksi ja välittää se pilvipalveluun tämän

ymmärtämässä muodossa. Yhdyskäytävä valikoitui sen vuoksi, että se tukee useita tiedonsiirtoprotokollia, kuten Modbus TCP/IP:tä, HTTPS:ää ja SSH:ta, sekä mahdollistaa räätälöidyn Python-ohjelman käyttämisen datan käsittelyssä.

Datan muuntaminen toteutetaan käyttäen JSON-kieltä, joka poimii Modbus-rekisteritaulusta datan. Kuten Kuvio 24 nähdään, kun Modbus-datan toimintokoodi tulkitaan JSON-kielellä string-tyyppiseksi muuttujaksi, käytetään hyväksi Taulukko 3 rekisteriä. Ohjelmointi tilataan palveluna integraattorilta.

```

6  def function_code_to_string(function_number:int):
7
8      if function_number == 1:
9          return "read_coils"
10     elif function_number == 2:
11         return "read_input_coils"
12     elif function_number == 3:
13         return "read_holding_registers"
14     elif function_number == 4:
15         return "read_input_registers"
16     elif function_number == 5:
17         return "set_single_coil"
18     elif function_number == 6:
19         return "set_single_holding_register"
20     elif function_number == 15:
21         return "set_multiple_coils"
22     elif function_number == 16:
23         return "set multiple holding registers"

```

Kuvio 24. Esimerkki Modbus-rekisteritaulukon käsittelystä JSON-kielellä.

Datan käsittelyn jälkeen se lähetetään pilveen, johon IoT-yhdyskäytävä kytkeytyy VPN-reitittimen kautta.

5.4.1 RUT-yhdyskäytävän asennus

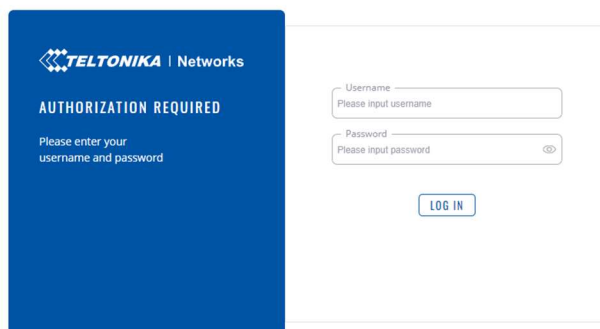
Asennusten osalta yhdyskäytävä on hyvin yksinkertainen ja vastaa hyvin paljon tavallisten, kuluttajille suunniteltujen laitteiden asennusta. Pakkauksessa mukana tullee muuntajavirtalähteellä tuotetaan käyttöjännite, ja kuten Kuvio 25 nähdään, PLC-linkiltä tuleva Ethernet-kaapeli kytketään mihin tahansa laitteen neljästä LAN-portista. WAN-kaapeli eli ulkoverkon kaapeli kytketään RUT-yhdyskäytävän sille varattuun, merkittyyn porttiin. WAN-kaapelin toinen pää kytketään Cison reitittimen LAN-porttiin, joka hoitaa reitityksen Azure-palvelimelle.



Kuvio 25. RUTX08-yhdyskäytävän asennus

5.4.2 Käyttöönotto

Reitittimen asennuksen jälkeen on tehtävä käyttöönotto, jossa määritellään asetukset. Tämä tapahtuu reitittimen asetussivujen (Kuvio 26) kautta. Pääsy sinne tapahtuu LAN-portin kautta kirjoittamalla selaimen osoitekenttään Teltonika WebUI:n IP-osoite. Tarvittaessa voidaan käyttää Windowsin PowerShellin komentoa `ipconfig /all`, mikäli laitteen tehdasasetusten mukainen IP-osoite ei ole tiedossa.



Kuvio 26. RUT-yhdyskäytävän hallintasivu

Yhdyskäytävää voidaan hallita myös ulkoverkon kautta, tai tässä tapauksessa VPN:ää käyttäen, kirjautumalla Operonin SSL-verkkoon. Jotta yhdyskäytävä olisi saavutettavissa virtuaalisen sisäverkon kautta, sille on asetettava oikeat ulkoverkon asetukset (Kuvio 27).

~ INTERFACES: WAN

GENERAL SETTINGS
ADVANCED SETTINGS
PHYSICAL SETTINGS
FIREWALL SETTINGS

Protocol: Static

IPv4 address: [redacted]

IPv4 netmask: [redacted]

IPv4 gateway: [redacted]

IPv4 broadcast: [redacted]

DNS servers: [redacted] +

~ DHCP SERVER

GENERAL SETUP
IPv6 SETTINGS

Enable DHCP: Disable

Kuvio 27. RUTX08:n IP-asetukset

Teltonikan WebUI:n lisäksi integraattori on ladannut laitteeseen asetustietokantaeditorin, jonka kautta voidaan ottaa käyttöön tarvittavat Modbusin ja Azure-tietokannan asetukset. Editori on jaettu kolmeen osaan: modbus-data.php, errors.php ja modbus.php. Varsinainen editori on modbus.php, ja muut sivut ovat "vain luku" -tyyppisiä tietosivuja. Editorissa voidaan muokata Modbusin tunnisteita (Kuvio 30), Azure-tietokantaan liittymistä ja kansiorakennetta (Kuvio 29). Yhdistäminen Azure-tietokantaan tapahtuu siten, että Modbus-asetuksissa asetetaan URL-muotoinen connection_string-polku.

Kertyvän datan määrää voi tarkastella modbus-data.php-sivun kautta, ja kuten Kuvio 28 nähdään, samalla voidaan arvioida lähetyksen toimivuutta. Data-sarake kertoo, onko dataa kertynyt. Sent-sarake osoittaa, onko data saatu lähetettyä pilveen. Tässä sarakkeessa arvo 0 tarkoittaisi, että dataa ei ole lähetetty pilveen tai ei ole otettu vastaan. Processed-sarake näyttää, onko data prosessoitu eli muutettu sopivaan muotoon, jotta se voidaan lähettää pilveen.

Select: modbus_data

New item

Search

Limit

50

Action

Select

Modify	timestamp	tag	data	sent	calculate_statistics	processed	rarely_changing
☐ edit	1700486000.297	rejektivesiallas_[m]	1834	1	0	1	0
☐ edit	1700486000.359	purkupumppaamo_[m]	2007	1	0	1	0
☐ edit	1700486000.424	linja_2_[m3_h]	0	1	0	1	0
☐ edit	1700486000.489	linja_3_[m3_h]	0	1	0	1	0
☐ edit	1700486000.551	saatimen_asetusarvo	2500	1	0	1	0
☐ edit	1700486000.613	Pinnankorkeus__MW812	0	1	0	1	0
☐ edit	1700486000.68	Tuleva_39200A0	4327	1	0	1	0
☐ edit	1700486000.742	linja_1_venttiili_auki	0	1	0	1	0
☐ edit	1700486000.807	Lahteva_39206A0	335	1	0	1	0
☐ edit	1700486000.872	linja_2_venttiili_auki	0	1	0	1	0
☐ edit	1700486000.934	linja_3_venttiili_auki	0	1	0	1	0
☐ edit	1700486000.996	LINJA1_39203A0	4616	1	0	1	0
☐ edit	1700486001.063	LINJA2_39204A0	4968	1	0	1	0
☐ edit	1700486001.128	LINJA3_39205A0	0	1	0	1	0
☐ edit	1700486001.207	happisaadin_asetusarvo_MW210	2500	1	0	1	0
☐ edit	1700486001.376	ilmastusallas_linja_2_happi_MW8	5023	1	0	1	0
☐ edit	1700486001.441	ilmastuskompressori_tjm102_olotaajuus_MW22	4781	1	0	1	0
☐ edit	1700486001.503	ilmastusallas_linjan_2_paine_MW24	4006	1	0	1	0
☐ edit	1700486001.568	happisaadin_asetusarvo_MW230	2600	1	0	1	0
☐ edit	1700486001.633	ilmastusallas_linja_3_happi_MW9	9170	1	0	1	0
☐ edit	1700486001.695	ilmastuskompressori_tjm103_olotaajuus_MW23	0	1	0	1	0
☐ edit	1700486001.757	ilmastusallas_linjan_3_paine_MW25	4237	1	0	1	0

Page

1 2 3 4 5 ... 472

Whole result

☐ 23,560 rows

Selected

☐ Edit
 ☐ Clone
 ☐ Delete

Export

Kuvio 28. Modbus-datataulukon tarkastelu

Uuden tagin lisääminen tapahtuu tagilistan (Kuvio 31) "New item" -kohdasta. Kuvio 30 nähdään valikko, johon määritellään kaikki tarvittavat tiedot.

Jokaisen tunnisteiden luominen käsin on kuitenkin kohtalaisen aikaa vievää. Tämän vuoksi modbus.php-sivulle on rakennettu tuo/vie-toiminto, jolla voidaan lisätä useita tageja .csv-tiedostosta nopeasti. On kuitenkin huomattava, että .csv-tiedoston muokkaus taulukko-ohjelmistolla, esimerkiksi Microsoft Excelillä, saattaa luoda tagilistaan muotoilutietoja, joiden tulkitsemista Modbus-asetusten tuonti ei tue. Tämän vuoksi suositellaan käyttämään muotoilun suhteen neutraalimpia ohjelmia, kuten Notepad tai Notepad++, joka on ilmaisohjelma.

Edit: modbus_settings

gateway_id	*	RUTX08
gateway_purpose	*	
is_active	*	1
connection_string		
data_url		
data_statistics_url		
data_rarely_changing_url		
errors_connection_string		
modbus_errors_url		
system_errors_url		
data_upload_interval	*	5
logs_upload_interval	*	5
data_upload_row_limit	*	1000
database_trim_interval	*	15
database_trim_cutoff	*	240

Kuvio 29. Modbus-asetukset

Server_ip on sen palvelimen osoite, jolta data luetaan. Jos kytkettyinä on useita PLC-laitteita, niiden vastaavat osoitteet tulisi määritellä tässä. Server_port on portti, jota Modbus käyttää, ja oletuksena sen arvo on 502. Slave_id on jokaiselle orjalle annettava yksilöllinen tunnistusnumero. Name on yksilöivä nimi positiolle, jolta data kerätään.

Nimeämiskäytännöissä kannattaa ottaa huomioon pilvipalvelun tietokannan käyttämä merkistö. Käytännössä tämä voi tarkoittaa erikoismerkkien ja tiettyjen aakkosten, kuten ä:n, ö:n ja å:n, välttämistä. Lisäksi yksikköjen tulkinta voi aiheuttaa ongelmia, mikäli IoT-reitittimellä on eri merkistöt kuin pilvipalvelussa. Tämän vuoksi merkistö standardoidaan siten, että yksiköt yksinkertaistetaan. Esimerkiksi lämpötila muutetaan merkinnästä °C muotoon C ja tilavuudet esitetään muodossa m3. Välimerkit esitetään alaviivoina, eli esimerkiksi pinta 1 merkittäisiin pinta_1. Vaikka nämä asiat voitaisiinkin ratkaista pilvipalvelun tietokannassa, joka tapauksessa pyritään työmäärän vähentämiseen. Nämä seikat voivat myös lisätä kustannuksia, koska tietokannan rakentaminen kuuluu tilattaviin palveluihin ja sitä kautta vaikuttaa suoraan kustannuksiin.

Register_address on osoite, johon muuttuja tallennetaan. Kuten luvussa 3.4.1 todettiin, Modbusin tapa tallentaa muuttujat perustuu muistipaikkojen 40001–49999 käyttöön. Tätä hyödynnetään rekisterin lukemiseen JSON-ohjelmalla, sillä ainoa tällä hetkellä käytössä oleva funktio on read_holding_register.

Period-arvolla voidaan määrittää, kuinka usein rekisteröitävä arvo päivitetään. Arvon yksikkö on sekunti, joten oletuksena arvot päivitetään minuutin välein. On huomattava, että datan keräystaajuus ei vaikuta lähetystaajuuteen, joka määritellään Modbus-asetuksissa kohdassa data_upload_interval (Kuvio 29).

Muuttujat rarely_changin, change_treshold ja calculate_statistics ovat lisämääritteitä, joilla voidaan määrittää, onko arvo harvoin muuttuva, kuten pumpun käyntitieto, joka on muotoa 0–1. Change_tresholdilla taas voidaan määrittää, kuinka paljon arvon tulee muuttua ennen kuin se päivitetään ja lähetetään pilveen. Tällä voidaan vähentää pilven tietokannan päivittämistarvetta. Toimintoa voidaan verrata hystereesitoimintoon. Calculate_statistics on nimensä mukaisesti arvojen tilastointia varten asetettava muuttuja, eli sillä voidaan määrittää, tilastoidaanko arvo vai ei.

Insert: modbus_tags

server_ip	*	
server_port	*	
slave_id	*	1
name	*	
enabled	*	1
register_address	*	
function	*	READ_HOLDING_REGISTER
period	*	60
rarely_changing	*	0
change_threshold	*	1
calculate_statistics	*	0

Kuvio 30. Modbus-tagin luominen

New item

Search (anywhere) Limit 50 Action Select

Modify	server_ip	server_port	slave_id	name	enabled	register_address	function	period	rarely_changing	change_thres
☐ edit	192.168.1.2	502	1	linja_1_[mg_l]	1	40001	READ_HOLDING_REGISTER	60	0	0
☐ edit	192.168.1.2	502	1	linja_2_[mg_l]	1	40002	READ_HOLDING_REGISTER	60	0	0
☐ edit	192.168.1.2	502	1	linja_3_[mg_l]	1	40003	READ_HOLDING_REGISTER	60	0	0
☐ edit	192.168.1.2	502	1	yliaamallettevirtaus_MW2	1	40004	READ_HOLDING_REGISTER	60	0	0
☐ edit	192.168.1.2	502	1	pH_lahteva_vesi	1	40005	READ_HOLDING_REGISTER	60	0	0
☐ edit	192.168.1.2	502	1	linja_44593_keskiarvo	1	40006	READ_HOLDING_REGISTER	60	0	0
☐ edit	192.168.1.2	502	1	linja_44621_keskiarvo	1	40007	READ_HOLDING_REGISTER	60	0	0
☐ edit	192.168.1.2	502	1	linja_44622_keskiarvo	1	40008	READ_HOLDING_REGISTER	60	0	0
☐ edit	192.168.1.2	502	1	linja_37653_keskiarvo	1	40009	READ_HOLDING_REGISTER	60	0	0
☐ edit	192.168.1.2	502	1	ilmastuspaine_[bar]	1	40010	READ_HOLDING_REGISTER	60	0	0
☐ edit	192.168.1.2	502	1	kompr_tjm_olotaj[Hz]	1	40011	READ_HOLDING_REGISTER	60	0	0
☐ edit	192.168.1.2	502	1	tuleva_vesi_[C]	1	40012	READ_HOLDING_REGISTER	60	0	0
☐ edit	192.168.1.2	502	1	ilmastus_1_vesi_[C]	1	40013	READ_HOLDING_REGISTER	60	0	0
☐ edit	192.168.1.2	502	1	lahteva_vesi_[C]	1	40014	READ_HOLDING_REGISTER	60	0	0
☐ edit	192.168.1.2	502	1	kompurahuone_[C]	1	40015	READ_HOLDING_REGISTER	60	0	0
☐ edit	192.168.1.2	502	1	saostuskem.salilo_[m]	1	40016	READ_HOLDING_REGISTER	60	0	0
☐ edit	192.168.1.2	502	1	polymeerisalilo_[m]	1	40017	READ_HOLDING_REGISTER	60	0	0
☐ edit	192.168.1.2	502	1	lietevarasto_[m]	1	40018	READ_HOLDING_REGISTER	60	0	0
☐ edit	192.168.1.2	502	1	rejektivesiallas_[m]	1	40019	READ_HOLDING_REGISTER	60	0	0
☐ edit	192.168.1.2	502	1	purkupumppaamo_[m]	1	40020	READ_HOLDING_REGISTER	60	0	0
☐ edit	192.168.1.2	502	1	linja_2_[m3_h]	1	40021	READ_HOLDING_REGISTER	60	0	0
☐ edit	192.168.1.2	502	1	linja_3_[m3_h]	1	40022	READ_HOLDING_REGISTER	60	0	0
☐ edit	192.168.1.2	502	1	saatimen_asetusarvo	1	40023	READ_HOLDING_REGISTER	60	0	0

Page 1 2 Whole result Selected ☐ 95 rows Edit Clone Delete Export CSV, Export

Kuvio 31. Modbus-tagilista

5.5 DMZ

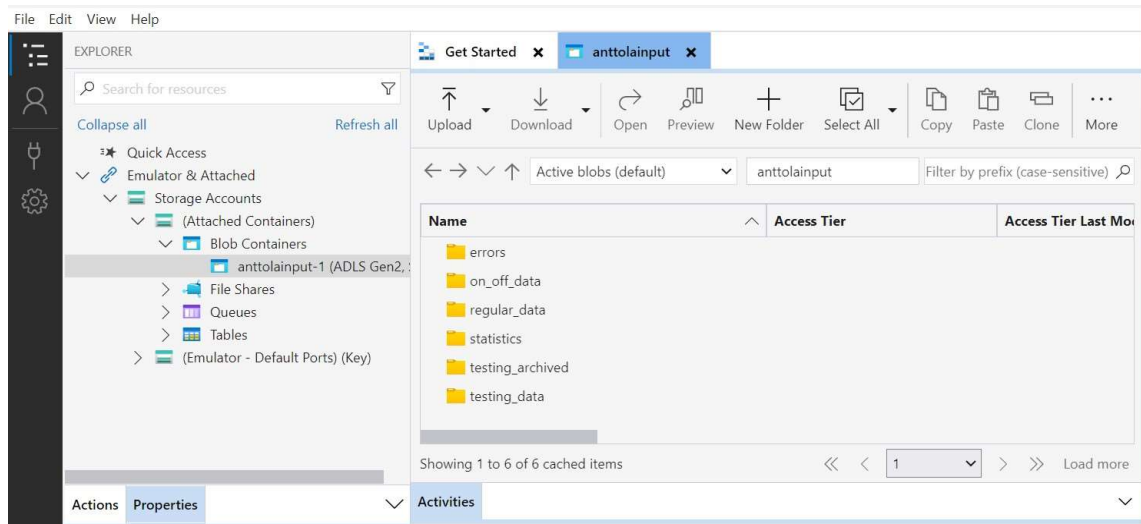
Koska tiedonsiirron pilvipalveluun on tapahduttava internetin välityksellä, laitoksen toimistoverkko ja datasiirron paikallisverkko on erotettava toisistaan demilitarisoidulla vyöhykkeellä.

Laitetasolla tämä toteutetaan Ciscon palomuuuri/ yhdyskäytävällä, jonka ylläpidosta vastaa laitoksen operaattoria palveleva IT-taho. Kuten kaikki yhteydet, liitäntä IoT-yhdyskäytävältä DMZ:lle tapahtuu soveltuvaa dataliikennekaapelia käyttäen. Kuten Kuvio 25 nähdään, tähän käytetään IoT-yhdyskäytävän WAN-porttiin kytkettävää normaalia TCP/IP-kaapelia. Kaapelin toinen pää tulee kytkeä Ciscon palomuurin porttiin, jolle voidaan määritellä DMZ-asetukset. Palomuuuri tarjoaa myös VPN- ja SSL-palvelut.

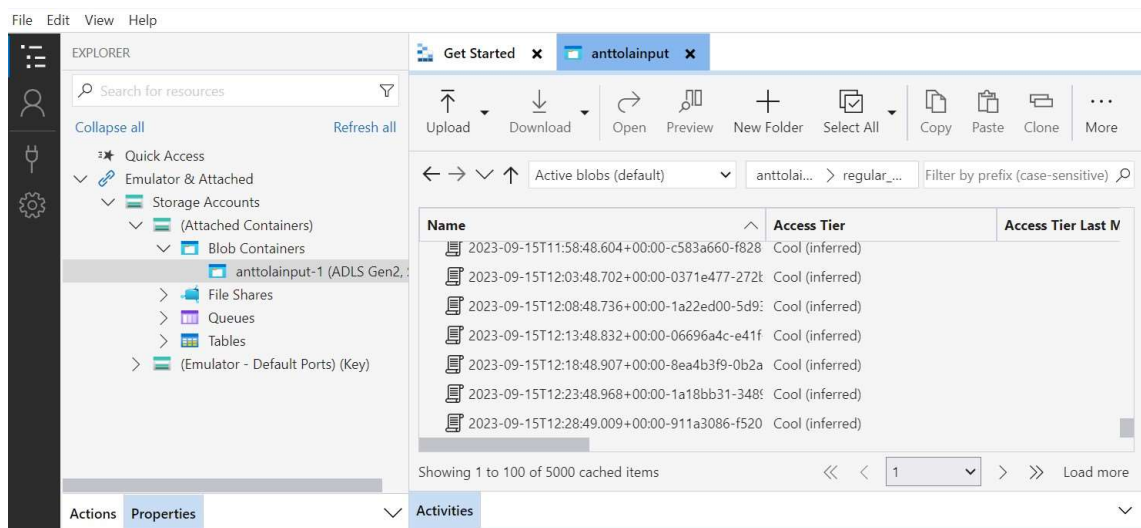
5.6 Pilvipalvelu

Pilvipalveluna käytetään Azure Blobia. Azuren käyttöön päädyttiin, koska se on Microsoft Corporationin palvelu, joka on integroitu muihin palveluihin. Tästä syystä pääsyoikeuksien hallinta on verrattain helppoa, sillä se tapahtuu Microsoft-tilien kautta.

Kuten Kuvio 32 nähdään, Azure Browser -ohjelmistolla voidaan tarkastella datan kerääntymistä ja kansiorakenteen muodostumista. Itse datan kirjautuminen nähdään Kuvio 33, jossa paketit on erotettu toisistaan aikaleimalla. Blobin datasta voidaan rakentaa erilaisia mittareita ja visualisointeja.



Kuvio 32. Azure Browserin näkymä: lähetetty data kertyy niille määrättyihin kansioihin



Kuvio 33. Datat näkymä Azure Browserissa

5.7 Toteutetun PLC-linkin vaatimustenmukaisuustarkastelu

Tässä luvussa tarkastellaan projektin lopputuloksia ja verrataan toteutusta Taulukko 54 esitettyihin vaatimuksiin. Tiedonkeräys on toteutettu varsin tavanomaisella, teoriaosassa esitettyllä kokoonpanolla, mutta varsinaista standarditapaa tiedonsiirtoon ei ole määritetty.

Kuten taulukosta 5 voidaan todeta, projektin lopputulos täyttää kaikki sille asetetut vähimmäisvaatimukset. Myös lisätoivomukset täyttyvät, ainakin osittain. Sovellettavuuskonsepti ei täyty täysin, sillä laitevalinnoissa ja asetuksissa tulee tehdä oleellisia muutoksia, mikäli esimerkiksi käytettävä kenttäväylä on jokin muu kuin Modbus. Yhteensopivuus voidaan kuitenkin toteuttaa vaikkapa väylämuuntimella, jolloin yhdyskäytävänä käytettävää laitetta ei tarvitse vaihtaa.

Taulukko 5. Vaatimuslistan toteutuminen

Vaatusmus	Luokka	Toteutu minen
Linkin tulee olla yhdensuuntainen (automaatiosta Azureen); linkin kautta ei saa ohjata automaatiojärjestelmää	VV	Kyllä
Linkin tulee kyetä lähettämään data standardimuotoisena (MQTT tms.) viestinä ennalta määriteltyn end-pointiin Azuren päässä	VV	Kyllä
Linkkiin ei tule päästä käsiksi langattomasti mistään pisteestä; kaiken liikenteen tulee tapahtua laitoksella fyysisiä kytkentöjä pitkin toimistoverkon kautta	VV	Kyllä
Tiedon lähetystaajuuden tulee olla käyttäjän valittavissa	LT	Kyllä
Konseptin tulee olla helposti sovellettavissa toisilla laitoksilla, joiden lähtötiedoista ei ole tietoa (eli kenttäväyläformaatti voi olla jotakin muutakin)	LT	Osittain
Linkin tulee olla mahdollisimman kustannustehokas	LT	Osittain

PLC-linkin laitteistossa olevan erottimen ansiosta Modbus-liikennöinti automaatiokeskuksen ja itse linkin PLC-laitteen välillä on yhdensuuntaista. Lisäksi IoT-yhdyskäytävän laite käsittelee ainoastaan Modbusin rekistereitä 40001–49999 ja niitäkin ainoastaan ”read”- eli ”luku”-komennolla. Näin ollen automaatiokeskuksen ohjaaminen muuttamalla muita rekistereitä IoT-laitteen välityksellä ei ole mahdollista.

IoT-yhdyskäytävä muuttaa Modbus-datan Azure Blobin käyttämään HTTPS-protokollamuotoon. Tämä toteutetaan räätälöidyllä JSON-ohjelmistolla. Linkin laitteistossa ei käytetä missään vaiheessa langattomia komponentteja, vaan kaikki yhteydet on toteutettu datansiirtokaapeleita käyttäen. Lähetystaajuus, jolla dataa lähetetään pilveen, voidaan valita modbus_settings.php-sivulla Data_Upload_Interval-arvolla. Oletusasetuksena on 5 minuuttia.

Sovellettavuus muille laitoksille riippuu täysin käytettävästä kenttäväylästä. Mikäli tämän opinnäytetyön mukaista ratkaisua halutaan soveltaa muilla kenttäväylillä,

on huomioitava laitteiden yhteensopivuus. Tämä koskee erityisesti sopivan IoT-yhdyskäytävän valitsemista, sillä laitteiston tulee olla yhteensopiva kenttäväylän ja pilvipalvelun käyttämän protokollan kanssa. Lisäksi IoT-yhdyskäytävän JSON-ohjelma ei ymmärrä muita kenttäväyliä, joten kenttäväylän ollessa jokin muu sille täytyy tehdä ohjelmointi. Vaihtoehtoisesti voidaan soveltaa kenttäväylämuunninta, mikä toisaalta lisää järjestelmän monimutkaisuutta yhden lisälaitteen verran.

Lisäksi tulee huomioida käytetyn väylän rajoitukset, jotka koskevat esimerkiksi solmujen määrää, väylän topologiamahdollisuuksia ja kaapelointia. Tässä tapauksessa käytössä olevaan RUTX08-IoT-yhdyskäytävään voidaan liittää neljä sisäverkon laitetta, joten jokaiseen yhdyskäytävään voitaisiin liittää neljä automaatiokeskusta. Koska yhdyskäytävä muuttaa datan TCP/IP-protokollaksi, teoriassa yhdysväyliä rakentamalla voidaan ylittää Modbus TCP/IP -protokollan 253 solmun raja. Huomioitavaa toki on, että yhtä väyläsegmenttiä rajoittaa edelleen sama solmumäärä, joten laajemman kokonaisuuden täytyisi koostua useammasta segmentistä.

Linkin toteutuksen kustannustehokkuutta arvioitaessa tulee huomioida sekä hankinta-, asennus- että käyttökustannukset. Tässä opinnäytetyössä kustannusten tarkempi käsittely on jätetty pois, mutta yleisellä tasolla tarkasteltuna voidaan sanoa, että pääpiirteiltään laitteisto on varsin kustannustehokas. Esimerkiksi IoT-yhdyskäytävänä käytetty RUTX08 on hyvin kustannustehokas laite, jossa on tarpeeksi toimintoja, mutta silti se on riittävän yksinkertainen valittuun tarkoitukseen.

Ainoastaan PLC-linkki on siinä mielessä ylimääräinen komponentti, että tiedonsiirtoon ei välttämättä tarvita ylimääräistä PLC-laitetta. Prosessiautomaatiokeskuksen PLC voitaisiin kytkeä suoraan IoT-yhdyskäytävään. Automaation ylläpidosta vastaa kuitenkin ulkoinen osapuoli, ja ehtona yhteistyölle oli ohjauskeskuksen lisääminen automaation ja IoT-yhdyskäytävän välille.

Tämä toki myös nostaa tietoturvan tasoa, sillä automaatiota ei voi ohjata verkon kautta, koska data saadaan ohjauskeskukseen erottimen kautta. Ilman erotinta verkon kautta ohjaaminen vaatisi sen, että ensin olisi päästävä murtautumaan

VPN-verkkoon ja sen jälkeen vielä laitteen hallintapaneeliin admin-tunnuksin. Tämän jälkeen JSON-ohjelmaan olisi lisättävä kirjoitusominaisuus kaikille Modbusin rekisteritauluille, sillä tällä hetkellä ohjelma pystyy ainoastaan lukemaan holding_register-taulun.

6 ONGELMANRATKONTA JA VIANSELVITTELY

Koska projekti oli luonteeltaan pilotti, sen käyttöönottovaiheessa ilmeni erilaisia haasteita. Tässä luvussa käydään läpi projektiin liittynyttä ongelmien ratkontaa ja vian selvittelyä.

Ensimmäisellä yrityksellä dataa ei kertynyt rekisteriin, mikä voitiin päätellä siitä, että Data-sarake näytti nollaa, kuten Kuvio 34 nähdään. Syynä tähän olivat virheelliset Modbus-asetukset automaatiokeskuksessa. Tästä syystä keskuksen PLC ei tallentanut dataa oikeaan rekisteriin (vrt. Taulukko 3).

Modify	timestamp ↓	tag ↓	data ↓	sent ↓	calculate_statistics ↓	processed ↓	rarely_changing ↓
☐ edit	1689062239.175	101	0	0	0	0	0
☐ edit	1689062179.152	101	0	0	0	0	0
☐ edit	1689062119.129	101	0	0	0	0	0
☐ edit	1689062059.109	101	0	0	0	0	0
☐ edit	1689061999.071	101	0	0	0	0	0
☐ edit	1689061939.01	101	0	0	0	0	0
☐ edit	1689061878.998	101	0	0	0	0	0
☐ edit	1689061818.935	101	0	0	0	0	0
☐ edit	1689061758.873	101	0	0	0	0	0
☐ edit	1689061698.811	101	0	0	0	0	0
☐ edit	1689061638.749	101	0	0	0	0	0
☐ edit	1689061578.692	101	0	0	0	0	0
☐ edit	1689061518.631	101	0	0	0	0	0
☐ edit	1689061458.569	101	0	0	0	0	0
☐ edit	1689061398.555	101	0	0	0	0	0
☐ edit	1689061338.494	101	0	0	0	0	0
☐ edit	1689061278.432	101	0	0	0	0	0
☐ edit	1689061218.37	101	0	0	0	0	0
☐ edit	1689061158.35	101	0	0	0	0	0
☐ edit	1689061098.289	101	0	0	0	0	0
☐ edit	1689061038.27	101	0	0	0	0	0
☐ edit	1689060978.239	101	0	0	0	0	0
☐ edit	1689060918.186	101	0	0	0	0	0
☐ edit	1689060858.124	101	0	0	0	0	0
☐ edit	1689060798.063	101	0	0	0	0	0

Page: 1 2 3 4 5 6
Whole result: ☐ 252 rows
Selected:
Export: CSV,

Kuvio 34. Dataa ei kerry rekisteriin

Modbus-asetusten korjaamisen jälkeen taulukon (Kuvio 28) Data-sarake indikoi datan kerääntymisestä, mutta Sent- ja Processed-sarakkeet näyttivät 0 (Kuvio 35). Tämä tarkoitti sitä, että dataa ei saatu lähetetyksi pilveen. Kuten Kuvio 36 nähdään, myöskään Azure Browserilla tarkasteltuna dataa ei ollut kertynyt palvelimelle.

Search

data

>=

1

(anywhere)

Limit

50

Action

Select

Modify	timestamp	tag	data	sent	calculate_statistics	processed	rarely_changing
☐ edit	1689071969.355	101	7057	0	0	0	0
☐ edit	1689072028.413	101	7087	0	0	0	0
☐ edit	1689072088.475	101	7100	0	0	0	0
☐ edit	1689072148.49	101	7110	0	0	0	0
☐ edit	1689072208.552	101	7124	0	0	0	0
☐ edit	1689072268.614	101	7129	0	0	0	0
☐ edit	1689072328.669	101	7150	0	0	0	0
☐ edit	1689072388.731	101	7130	0	0	0	0
☐ edit	1689072448.793	101	7178	0	0	0	0
☐ edit	1689072508.85	101	7156	0	0	0	0
☐ edit	1689072568.912	101	7174	0	0	0	0
☐ edit	1689072628.97	101	7219	0	0	0	0
☐ edit	1689072689.03	101	7192	0	0	0	0
☐ edit	1689072749.091	101	7224	0	0	0	0
☐ edit	1689072809.149	101	7209	0	0	0	0
☐ edit	1689072869.209	101	7206	0	0	0	0
☐ edit	1689072929.271	101	7209	0	0	0	0
☐ edit	1689072989.332	101	7224	0	0	0	0
☐ edit	1689073049.394	101	7195	0	0	0	0
☐ edit	1689073109.458	101	7180	0	0	0	0
☐ edit	1689073169.519	101	7194	0	0	0	0
☐ edit	1689073229.58	101	7223	0	0	0	0
☐ edit	1689073289.64	101	7178	0	0	0	0
☐ edit	1689073349.7	101	7187	0	0	0	0
☐ edit	1689073409.762	101	7184	0	0	0	0

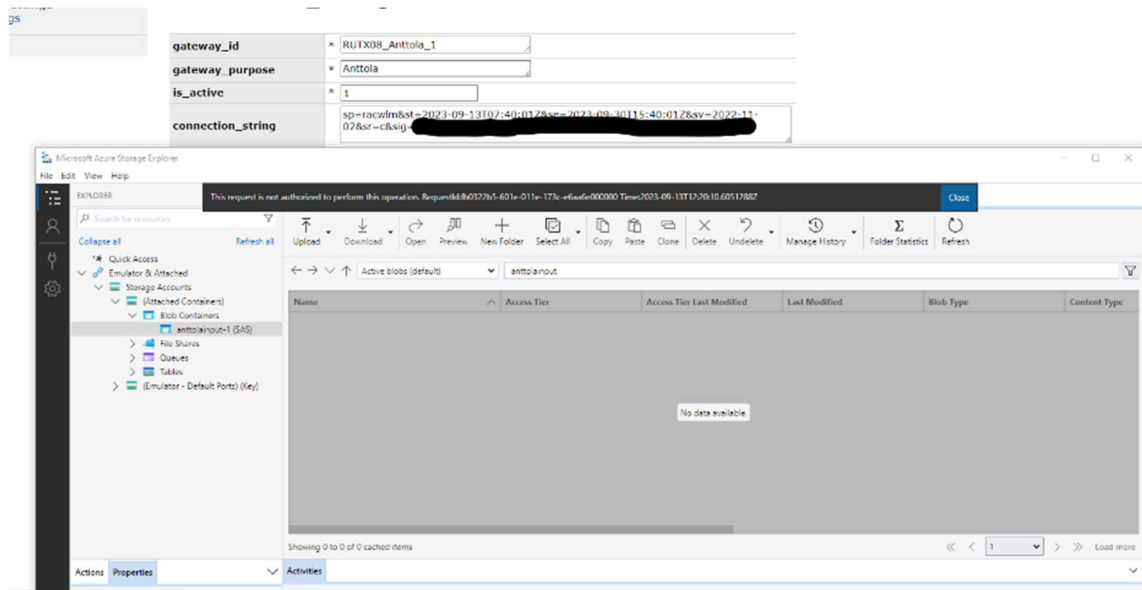
Whole result

Selected

Export

☐ 26 rows

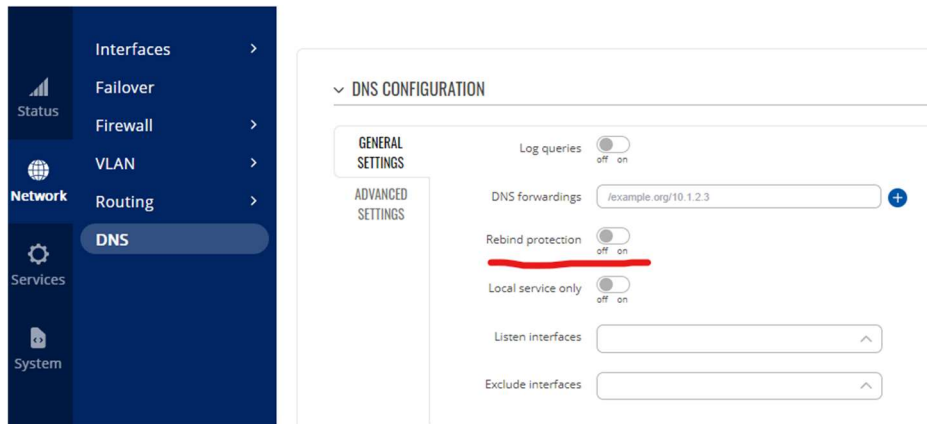
Kuvio 35. Dataa kerätään IoT-reitittimelle mutta ei lähetetä eteenpäin



Kuvio 36. Azure Browserin no data -ongelma

Lopulta syyksi paljastui RUT-yhdyskäytävän Rebind protection -turva-asetus (Kuvio 37), joka ei sallinut liikennettä eteenpäin. Tässä tilanteessa suojauksen poistaminen ei heikennä tietoturvaa merkittävästi, sillä suojauksen on katsottu olevan riittävä myös ilman rebind-hyökkäykseltä suojaavaa toimintoa. Tämä johtuu siitä, että yhteys pilveen muodostuu VPN:n kautta, joten DNS-kysely ei

tapahdu julkista verkkoa pitkin. Mikäli DNS-haku tapahtuisi julkista nimipalvelun hakua käyttäen, suojausta tulisi kuitenkin käyttää.



Kuvio 37. Rebind protection -asetuksen tulee olla poissa päältä

Testausvaiheessa ilmeni, että IoT-yhdyskäytävän aika-asetuksia ei oteta huomioon koodissa vaan UTC +00:00 -aika-alue on kovakoodattu ohjelmaan. Azuren kannalta tulisi kuitenkin käyttää oikeaa aikamuotoa, jotta aikaleima, jolla paketit erotetaan toisistaan, toimisi halutulla tavalla. Tämä voidaan toki hoitaa myös Azuren puolella, mutta ohjelman muuttaminen IoT-yhdyskäytävään on yksinkertaisempi toimenpide.

Käyttäjän näkökulmasta paras tapa olisi parametroida UTC-aika koodiin, jolloin asetukset huomioitaisiin. Tämä vaatisi muutostilauksen, sillä alkuperäisissä tilauksen määrittelyissä asiaa ei ollut osattu ottaa huomioon. Lisäksi parametroinnissa huomioitavaa on laitevalmistajan tuki, eli voidaanko jotakin tiettyä parametria ylipäättään lukea räätälöidyllä ohjelmistolla.

Kaikki laitteet, jotka yhdistetään Azuren pilvipalveluun, vaativat URL-muotoisen pääsyavaimen. Microsoft suosittelee avainten uudistamista tietyin väliajoin, ja tällöin kaikille laitteille on päivitettävä sama pääsyavain. Tämä on pikemminkin ominaisuus kuin varsinainen vika, sillä avaimen voimassaolopäiväys on tiedossa ja myös URL-muodossa nähtävissä. Uusiminen on kuitenkin tehtävä käsin, eli laitteille on syötettävä uusi avain asetusten kautta. Tämä kerryttää ns. hiljaista tietoa, jota tulisi aina välttää.

URL-muotoisen pääsyavaimen automaattinen päivitys vaatisi lisää ohjelmointia sekä IoT-laitteen että Azuren puolelle. Lisäksi olisi luotava turvallinen, luotettava kanava, jonka kautta pääsyavaimet laitteille voidaan jakaa.

7 TULOKSET JA TARKASTELU

7.1 Datan keräys ja lähetys

Prosessidata, joka tuotetaan PLC-laitteella, vaatii kenttäväylän määrittelyn. Dataa voidaan siirtää kenttäväylällä, ja esimerkiksi Allen Bradley käyttää MQTT-protokollaa, jolla dataa voidaan siirtää suoraan Azure-palvelimelle.

Tietoturvasyistä automaation kenttäväylän suositellaan olevan DMZ-ympäristössä. Mikäli kenttäväylää ei voida yhdistää suoraan pilveen, lähetykseen käytetään IoT-yhdyskäytävää, jossa kenttäväylän data muutetaan sopivaksi pilvipalvelulle. IoT-yhdyskäytävään on määritettävä kenttäväylän tunnisteet, jotta datan lähetys voi toimia.

7.2 Linkin käytännön toteutus ja laajennettavuus

Linkki tulee sijoittaa siten, että taajuusmuuttajat tms. eivät häiritse datan lähetystä. Automaatiokeskukselta kerättävä data siirretään Modbusilla RUTX08-IoT-yhdyskäytävään. Siellä data muutetaan JSON-koodilla toteutetun ohjelman avulla TCP/IP-protokollan mukaiseksi viestiksi, jonka pilvipalvelu Azure Blob kykenee tunnistamaan. IoT-yhdyskäytävä kytketään DMZ-yhdyskäytävän kautta internetiin ja pilveen.

Kohteessa on ainoastaan yksi prosessiautomaatiokeskus, jossa yksi PLC-laite ohjaa prosessia. Toteutetulla laitteistolla IoT-yhdysväylään voidaan suoraan kytkeä kolme PLC-laitetta, mutta keskittimillä se voidaan laajentaa vielä useamman PLC:n järjestelmäksi. Rajoittava tekijä on kuitenkin kenttäväylä, mutta mikäli laitteet ovat eri verkoissa, laajennettavuutta rajoittaa ainoastaan IP-osoiteavaruus.

Mikäli kenttäväylä on jokin muu kuin Modbus, laitteisto on vaihdettava sitä tukevaan malliin ja JSON-koodin on oltava käytettävän väylän mukainen, sillä eri kenttäväylissä rekisterit ovat erilaisia. Vaihtoehtoisesti voidaan käyttää kenttäväylämuunninta, jolla väylä muunnetaan Modbusiksi, ja hyödyntää suunniteltua laitteistoa.

8 POHDINTA

Kokonaisuudessaan projekti oli hyvin onnistunut, sillä kaikki sen vähimmäisvaatimukset saavutettiin. Lisäksi kaikki lisätoivomukset saavutettiin ainakin osittain.

Ainoa hieman ristiriitainen kohta on kustannustehokkuus, sillä toteutukseen tuli automaation toimittajaosapuolelta vaatimus erillisen PLC-laitteen käytöstä. Datan siirto ei nimittäin välttämättä tarvitsisi kyseistä ylimääräistä laitetta, vaan yhteys Azuren pilvipalveluun olisi voitu toteuttaa suoraan automaatiokeskukselta IoT-yhdyskäytävälle. Toisaalta tämän ratkaisun taustalla vaikuttivat tietoturvavaatimukset ja erityisesti datalähetyksen yksisuuntaisuuden vaatimus. Automaatiota ylläpitävä taho halusi varmistua, ettei prosessia voida ohjata pilvipalvelun kautta, ja tämän vuoksi päädyttiin ”ylimääräiseen” PLC-keskukseen automaatiokeskuksen ja IoT-yhdyskäytävän välillä. Lisäksi sopimus automaation ylläpidosta ei velvoita kytkemään ”ulkoisia” laitteita automaatioon, joten projektin lopputulos oli erittäin suuressa määrin riippuvainen myös yhteistyöstä automaatiotoimittajan kanssa. Luonnollisesti tämä seikka vaikutti ainakin alkukustannuksiin, sillä PLC-ohjauskeskuksen tilaus oli ns. normaalitoimituksen ulkopuolinen työ, josta koitui kustannuksia ylimääräisen tilaustyön mukaisesti.

Alun perin tähän opinnäytetyöhön suunniteltiin myös kustannustarkastelua, mutta lopulta se rajattiin ulkopuolelle työn laajuuden vuoksi. Toisaalta kustannuskysymykset ovat aina ajankohtaisia, joten jatkotutkimusta olisi syytä harkita ainakin siltä osin. Myös toteutus muilla kenttäväylillä olisi varsin mielenkiintoinen tutkimusaihe. Erityisen kiinnostava olisi tutkimus siitä, kuinka laajamittaisen datankeräyksen ja sen visualisoinnin toteuttaminen tämän opinnäytetyön kuvaamalla tavalla olisi vielä kustannustehokasta ja työmäärään nähden järkevää. Datan keräyksen kokonaistoimitus on nimittäin mahdollista teettää kokonaan ulkopuolisena palveluna, ja näitä palveluja tarjoavia tahoja on olemassa. Yleensä palvelut käsittävät kaiken datankeräyksen laitekannasta aina datan soveltamiseen saakka, kuten mittarit, seurantajärjestelmät ja visualisoinnit.

Lisäksi erittäin hyvä tutkimusaihe olisi pääsyavainten automaattinen päivittäminen. Laitekannan kasvaessa manuaalinen päivitys ei nimittäin liene järkevin vaihtoehto, vaikka avainten kelpoisuusaika olisikin varsin pitkä.

Datan laajamittaisen keräyksen hyödyntäminen on merkittävää, sillä nykyisellään erityisesti vedenkäsittelyprosesseja ajetaan ”varman päälle”. Esimerkiksi kemiallisissa prosesseissa käytettyjä kemikaaleja syötetään niin paljon, että määrä varmasti riittää tiettyyn puhdistustulokseen. Tarpeettoman osan poistaminen tästä annostelumäärästä – tai edes marginaalin kaventaminen – johtaisi välittömiin säästöihin. Tämä ei ilmenisi ainoastaan yhden puhdistamon kemikaalilaskuissa, vaan lisäksi voitaisiin vähentää ympäristön kuormitusta globaalisti.

Lähitulevaisuudessa parhaiten ehkä kuitenkin korostuu käyttökelpoisen datan merkitys. Siitä syystä näkisin datan keräyksen automatisoinnin ennen kaikkea sijoituksena tulevaisuuteen. Yksi pääsystä tähän on, että tulevaisuuden näkymät tekoälyn monipuolistumisen ja käytettävyyden kannalta ovat viime vuosina kasvaneet suorastaan räjähdysmäisesti. Prosessiteollisuudessa tekoälyn käyttöönottoa on suunniteltu jo pidempään ja jossakin määrin sitä on jo käytettykin menestyksekkäästi. Vedenpuhdistusalalla tekoälyä ei kuitenkaan juuri hyödynnetä. Toisaalta uudet puhdistustekniikat ovat vaativia, ja operaattorin vaikutus puhdistustulokseen on suurin merkittävä tekijä.

Nykyiset tekoälysovellukset ovat huomattavan monipuolisia. Esimerkiksi oppivan tekoälyn käyttö prosessinohjauksessa toisi mielenkiintoisia näkökohtia prosessin optimoimiseen sekä puhdistustuloksen että kustannusten kannalta. Tekoälyn kannalta tärkein yksittäinen tekijä on datan määrä, ja määrää vieläkin tärkeämpi ominaisuus on datan laatu. Toisaalta useimmat vedenpuhdistusprosessit nojaavat automaation käyttöön, joten laadukasta dataa olisi hyvin saatavilla, mikäli sen keräys voitaisiin automatisoida. Toki laadultaan hyvä data on myös sinällään arvokas asia minkä tahansa prosessin hallinnassa, olipa päätöksen tekijänä sitten inhimillinen tekijä tai koneäly.

LÄHTEET

Azure 2023. What is cloud computing? A beginner's guide. Viitattu 3.11.2023 <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-cloud-computing/>.

Bowne, M. 2020. The difference between Profibus and Profinet. PI North America 10.6.2020. Viitattu 8.10.2023 <https://us.profinet.com/the-difference-between-profibus-and-profinet/>.

Cassiolato, C. 2011. Tips for trouble shooting on the PROFIBUS-DP. Viitattu 25.10.2023 <https://www.smar.com/en/technical-article/tips-for-trouble-shooting-on-the-profibus-dp>.

Fogg, E. 2022. PLC to cloud. Using IoT to read data from a PLC. MachineMetrics 14.4.2022. Viitattu 12.11.2023 <https://www.machinemetrics.com/blog/plc-data>.

Harrington, H., Contreras, M. T. & Swindal, J. 2023. PROFINET communication channels. PROFINET University. Viitattu 31.10.2023 <https://profinetuniversity.com/profinet-basics/profinet-communication-channels/>.

Hersent, O., Boswarthick, D. & Elloumi, O. 2012. The internet of things. Key applications and protocols. Hoboken: John Wiley & Sons.

Inst Tools 2023. All about fieldbus protocols. Viitattu 2.10.2023 <https://instrumentationtools.com/fieldbus-protocols/>.

Julku, V.-V. 2021. Automaatioliikennesimulaattori. Kandidaatintyö, Tampereen yliopisto. Viitattu 3.11.2023 <https://urn.fi/URN:NBN:fi:tuni-202108246741>.

Keinänen, T. & Sumujärvi, M. 2019. Automaatiotekniikka. Helsinki: Sanoma Pro.

Mandin, T. 2019. Securely connect an industrial PLC to Azure. LinkedIn 21.1.2019. Viitattu 1.11.2023 <https://www.linkedin.com/pulse/securely-connect-industrial-plc-azure-terry-mandin>.

Nanotec 2023. Fieldbus systems. Viitattu 2.10.2023 <https://en.nanotec.com/knowledge-base-article/fieldbus-systems>.

Preimesberger, C. 2021. Cloud computing pros and cons. The good, the bad, and the gray areas. ZCNET 20.9.2021. Viitattu 12.12.2023 <https://www.zdnet.com/article/cloud-computing-pros-and-cons/>.

Pyyskänen, S. 2007. Teollisuuden laiteverkot. Johdatus väylätekniikkaan. Helsinki: Suomen Automaatioseura.

Rouhiainen, O. 2015. Logiikkaohjatun datan keräyksen integroiminen pilvipalveluun. Diplomityö, Tampereen teknillinen yliopisto. Viitattu 13.12.2023 <https://urn.fi/URN:NBN:fi:tty-201511241769>.

Schneider Electric 2023. Tuotetiedot. Modbus Tp. TWDXCAISO. Viitattu 17.10.2023 https://www.se.com/fi/fi/product/download-pdf/TWDXCAISO?filename=Schneider+Electric_Zelio-Logic_TWDXCAISO.pdf

Simply Modbus 2020. Data communication test software. Simply the easiest way to test and debug Modbus systems. Viitattu 10.10.2023 <https://simplymodbus.ca/index.html>.

Tools 2023. All about fieldbus protocols. Viitattu 2.10.2023 <https://instrumentationtools.com/fieldbus-protocols/>.

Trout Software 2023. Harnessing the power of PLC Data. Medium 21.8.2023. Viitattu 5.12.2023 <https://medium.com/@trout.software/harnessing-the-power-of-plc-data-a702a775e40f>.

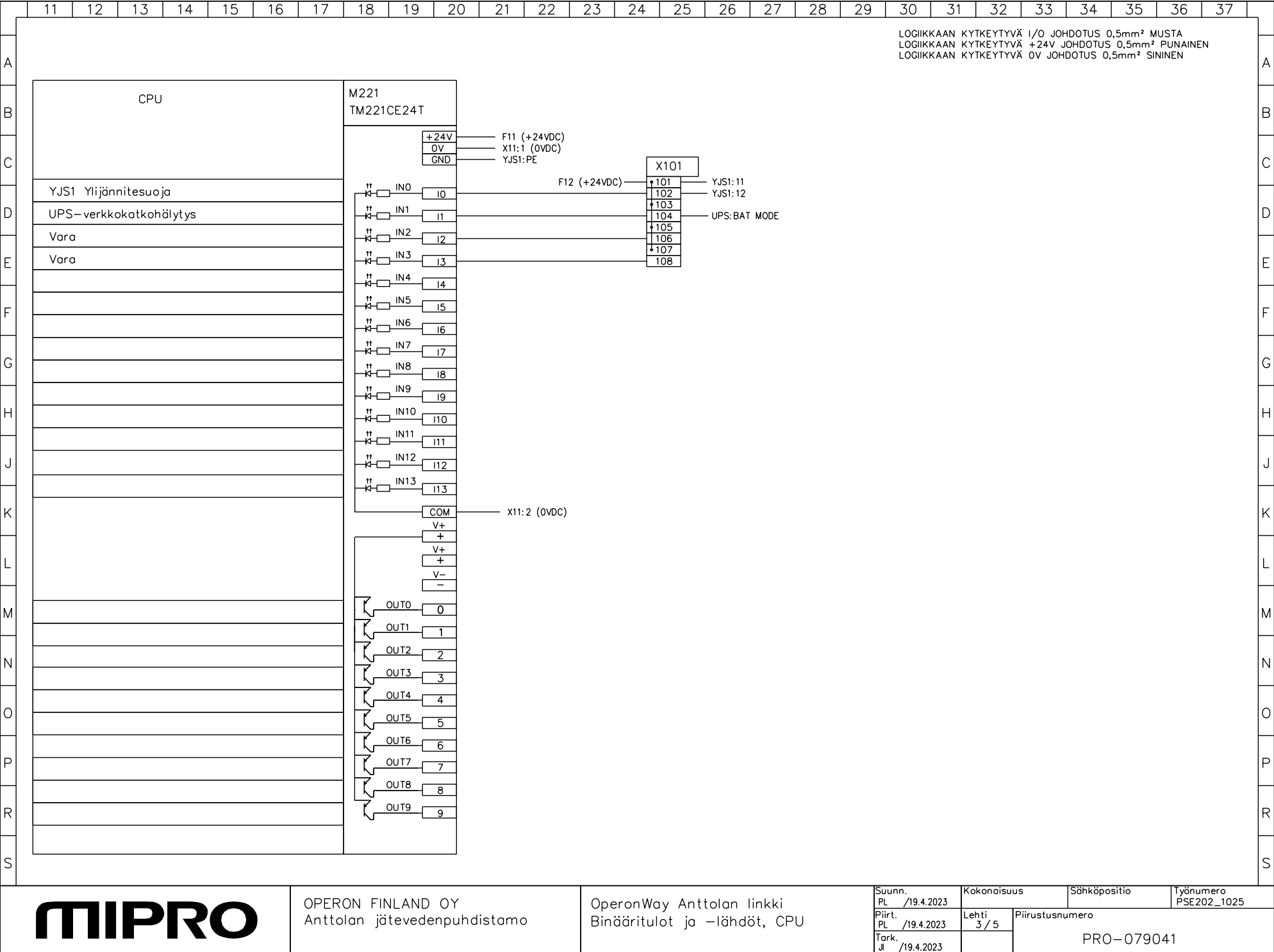
vNode 2021. How to connect my PLCs to Microsoft Azure IOT? 4.1.2021. Viitattu 12.11.2023 <https://vnodeautomation.com/how-to-connect-my-plcs-to-microsoft-azure-iot/>.

LIITTEET

Liite 1. Sähkökuvat

[illegible]

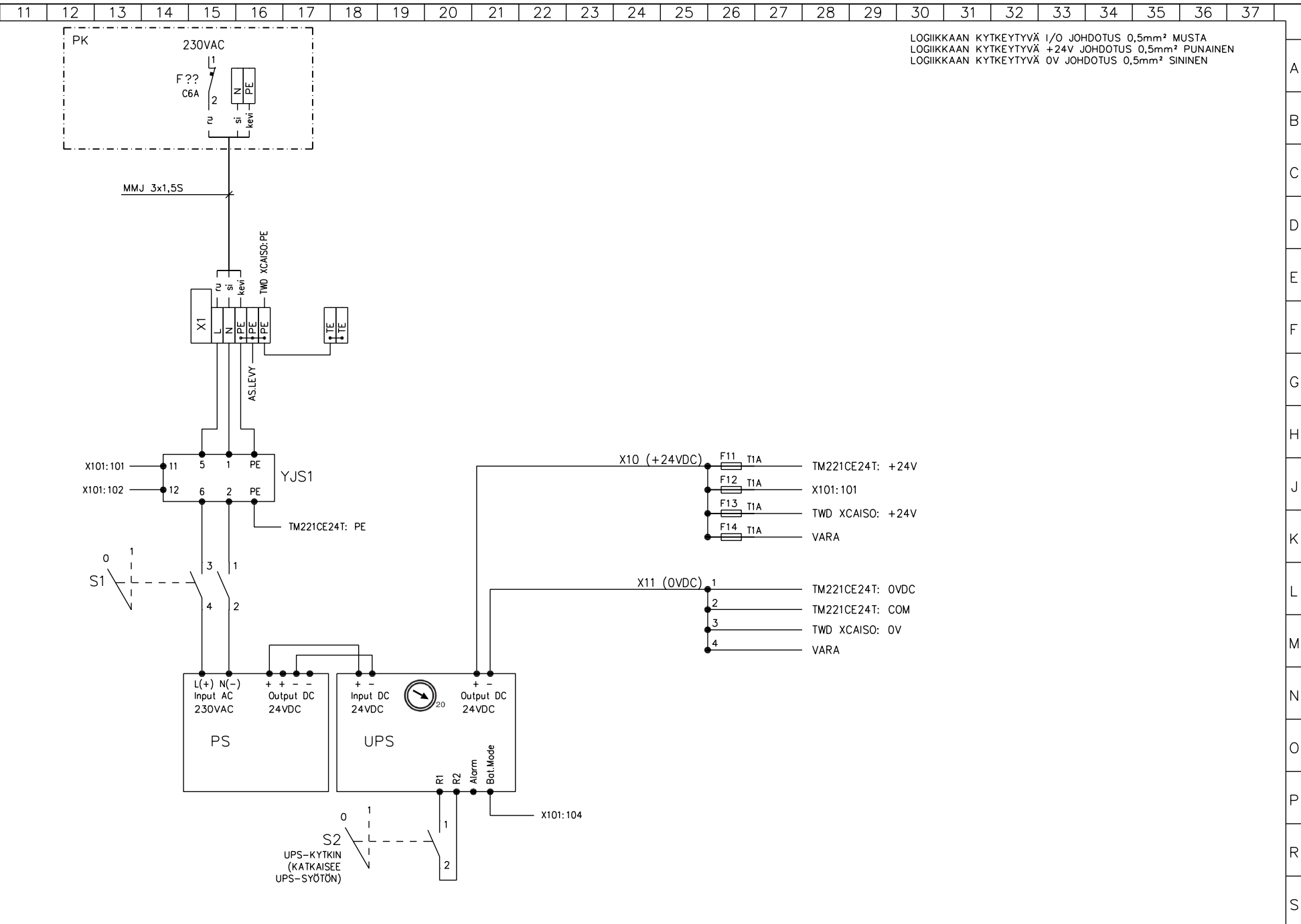
D muutos	E muutos	F muutos	21.4.2023	Työpiirustus	C muutos



OPERON FINLAND OY
Anttolan jätevedenpuhdistamo

OperonWay Anttolan linkki
Binääritulot ja -lähdt, CPU

Suunn. PL /19.4.2023	Kokonaisuus	Sähköpositio	Työnumero PSE202_1025
Piirt. PL /19.4.2023	Lehti 3 / 5	Piirustusnumero PRO-079041	
Tark. Jl /19.4.2023			



MIPRO

OPERON FINLAND OY
Anttolan jätevedenpuhdistamo

OperonWay Anttolan linkki
Jännitteenjako

Suunn. PL /19.4.2023	Kokonaisuus	Sähköpositio	Työnnumero PSE202_1025
Piirt. PL /19.4.2023	Lehti 4 / 5	Piirustusnumero	PRO-079041
Tark. J /19.4.2023			

[illegible]