



Vesa Karvonen

# Yhden piirilevyn tietokone teollisen internetin rajapintana tuotanto- ja prosessilaitteissa

Metropolia Ammattikorkeakoulu

Insinööri (YAMK)

Älykäs teollisuus

Insinöörityö

19.11.2023

# Tiivistelmä

Tekijä:	Vesa Karvonen
Otsikko:	Yhden piirilevyn tietokone teollisen internetin rajapintana tuotanto- ja prosessilaitteissa
Sivumäärä:	51 sivua + 1 liitettä
Aika:	19.11.2023
Tutkinto:	Insinööri (YAMK)
Tutkinto-ohjelma:	Älykäs teollisuus
Ammatillinen pääaine:	
Ohjaajat:	Lehtori Matti Välikylä Yliopettaja Jarno Varteva

---

Teollisuuden tiedonkeruu laitteista ja koneista ei ole uusia asia, mutta teollisen internetin ja tiedolla johtamisen suuntauksen myötä tiedonkeruun tarve laitteista on lisääntynyt. Tietoa halutaan perinteisen tuotannon seurannan lisäksi, ympäröivistä olosuhdemuutoksista, instrumenttien kunnosta, käyttäjien toimenpiteistä tuotantoprosessin aikana tai vaikka koneen käyttöönnoton aikaisista prosessin käynnistysparametreistä.

Tiedonkeruun ja teollisten laitteiden verkottaminen on kuitenkin muodostunut haasteelliseksi, koska on vaikea löytää osaavaa henkilöstöä, joka pystyy tekemään perusautomaatiotyön lisäksi myös teolliseen internetiin liittyvää ohjelmointia. Tutkimuksesta käy myös ilmi, että pilvipalveluihin ja data-analytiikkaan keskittyneillä yrityksillä on pula osajista, joten on tarve löytää keinoja kompensoida työvoimapulaa yksinkertaisilla teknisillä ratkaisuilla.

Tämän työn tarkoituksena on löytää sopivan yksikertainen API-rajapinta ARM-alustaiselle yhden piirilevyn tietokoneelle, jossa on valmiudet tiedonkeruulle niin paikallisesti kuin pilvialustalle. Lisäksi tavoitteena on, että työ olisi mahdollista tehdä hyvin ohjeistettuna ilman suurempia IT-taitoja.

Avainsanat: IIoT, Teollinen Internet, Node-Red, Low-Cost devices, Open-source software, Raspberry Pi, matalakoodi

## Abstract

Author: Vesa Karvonen  
Title: A single board computer as an API interface in production and process devices  
Number of Pages: 51 pages + 1 appendices  
Date: 19 November 2023

Degree: Master of Engineering  
Degree Programme: Intelligent Industrial Solutions  
Professional Major:  
Supervisors: Matti Välikylä, Senior Lecturer  
Jarno Varteva, Principal Lecturer

---

Industrial data logging from devices and machines is not a new thing, but with the industrial internet and the trend of information management, the need for data collection from devices has increased. In addition to traditional production monitoring, information is wanted about changes in surrounding conditions, the condition of instruments, user actions during the production process or even about the process start-up parameters during the commissioning of the machine.

However, data collection and networking of industrial devices has become a challenge because it is difficult to find competent personnel who can do not only basic automation work but also programming related to the industrial internet. The research also shows that companies focused on cloud services and data analytics have a shortage of experts, so there is a need to find ways to compensate for the labor shortage with simple technical solutions.

The purpose of this work is to find a suitable simple API interface for an ARM-based single-circuit board computer, which has capabilities for data collection both locally and on a cloud platform. In addition, the goal is that it would be possible to do the work with good instructions without major IT skills.

Keywords: IIoT, Industrial internet, Node-Red, Low-Cost devices, Open-source software, Raspberry Pi, LowCode

# Sisällys

## Lyhenteet

1	Johdanto	1
1.1	Tutkimusongelma	1
1.2	Tutkimuksen rajaukset	2
1.3	Tutkimuskysymykset	2
2	Teollinen internet ja työllisyys	3
2.1	Teollisen internetin Piilaakso?	3
2.2	Teollisen internetin haasteet	5
2.2.1	Haasteisiin vastaaminen	5
2.2.2	Yritysten välinen yhteistyö	6
2.2.3	Teknologiset ratkaisut	6
2.2.4	Halvat ratkaisut ohjaamaan investointeja	7
3	Teollisen internetin työkalut	9
3.1	Low cost -teknologia	10
3.1.1	Low cost -tietokoneet	10
3.2	Soft-PLC	12
3.2.1	OpenPLC	13
3.2.2	Codesys	13
3.3	Ohjelmoitavat sovellusrajapinnat	13
3.3.1	ISO-on-TCP	14
3.3.2	OPC-UA	14
3.3.3	Modbus/TCP	15
3.3.4	MQTT	15
3.3.5	REST API	15
3.4	Ohjelmointialusta ja ohjelmointimetodit	16
3.4.1	Low-Code- ja No-Code-ohjelmointialustat	18
4	Tutkimusympäristö	19
4.1	Tiedonkeruun ohjelmointialustan valinta	19
4.2	API-rajapintavertailu	20
4.2.1	ISO-on-TCP	20

4.2.2	Modbus/TCP	25
4.2.3	MQTT	28
4.3	Tutkimusympäristön tietoturva	33
4.4	Rajapintojen yhteenveto	34
4.5	Tiedon tallennus paikallisesti vai verkkoon?	35
5	Tiedonkeruuympäristön ohjelmointi	36
5.1	PLC:n konfigurointi	37
5.1.1	PLC-ohjelma	38
5.2	SBC:n konfigurointi	40
5.2.1	Esiasettelut	40
5.2.2	Tutkimustyökalujen asennus	40
5.3	Node-RED	41
5.4	Yhteenveto tiedonkeruun ohjelmoinnista	47
6	Pohdinta	48
6.1	Jatkokehitys	50
7	Lähdeluettelo	52

## Liitteet

Liite 1: NodeRED-ohjelmakoodi S7-tiedonkeruuseen

## Lyhenteet

- API: *Application Programming Interface. Ohjelmointirajapinta.* Standardin mukainen käytäntö tai yhtymäkohta, joka mahdollistaa integraation eri ohjelmistojen ja/tai laitteiden välillä.
- Ascii: *American Standard Code for Information Interchange.* 7-bittinen tai 8-bittinen merkkikoodi standardi, joka sisältää tietokoneissa käytetyt merkit. 8-bittinen merkistö sisältää myös skandinaaviset merkit.
- HMI: *Human Machine Interface,* käyttöliittymä ihmisen ja koneen välillä
- IEC 61131: Kansainvälinen standardi ohjelmitavien logiikoiden ohjelmointikielille ja -arkkitehtuurille. Standardin viimeinen versio on IEC 61131-3 julkaistu vuonna 2013
- IIoT: *Industrial Internet of Things.* Teollinen esineiden internet. Tuotanto- ja prosessikoneisiin liitettynä tuottaa arvoa yritykselle
- IT: *Information Technology.* Nimitys tietotekniikan/informaatioteknologian järjestelmäympäristöstä
- Json: *JavaScript Object Notation.* JSON on ohjelmointikielestä riippumaton, avoin ja standardoitu tiedostoformaatti, jota voi käyttää useiden eri ohjelmointikielien kanssa. JSON-tiedostot ovat tekstimuotoisia ja ihmisten luettavissa (ja muokattavissa). (Aalto, 2019)
- OT: *Operation Technology.* Nimitys teollisuuden tuotantoympäristöstä
- OTS: Off-the-Shelf Solution, valmisjärjestelmä, järjestelmän käyttöönotto ei vaadi tai vaati vähäistä räätälöintiä.

PLC: *Programmable Logic Controller*, ohjelmoitava logiikkakontrolleri, koneautomaation ohjauspiiri

REST API: *Representational state transfer*. REST API (tunnetaan myös nimellä RESTful API) on ohjelmointirajapintojen toteuttamiseen tarkoitettu arkkitehtuurimalli, joka noudattaa REST-arkkitehtonisen tyylin rajoituksia ja mahdollistaa vuorovaikutuksen RESTful-verkkopalveluiden kanssa.

SCADA: *Supervisory Control And Data Acquisition*,

Soft PLC: *Soft Programmable Logic Controller*, ohjelmoitava logiikkakontrolleri, joka toimii ohjelmistona tietokoneessa tai sulautetussa järjestelmässä.

# 1 Johdanto

Teollisen esineiden internetin myötä tietoon perustuvan johtamisen tarve tuotanto- ja prosessilaitteiden ohjauksissa on lisääntynyt. Useat tuotantolaitokset päivittävätkin kone- ja laitekantojaan säännöllisesti omien vaatimustensa mukaisesti tehostaakseen tuotantoa tai parantaakseen prosessien laatua. Päivittäminen on kuitenkin kallista ja esimerkiksi kaupalliset tiedonkeruulaitteistot ovat monesti niin kalliita suhteessa koneen kokonaiskustannuksiin ja vaikuttavat etenkin pienten toimijoiden hankintoihin, että tiedonkeruulaitteistot jätetään hankkimatta. Lisäksi PLC-ohjelmoijien rajalliset tiedot IT-ohjelmoinnista vaikeuttavat koneiden päivittämistä ja tuovat myös lisäkustannuksia, koska IT-ohjelmointityö on usein erillään normaalista teollisuuden ohjelmointityöstä. Toisaalta taas IT-ohjelmoijien tuntemus teollisuuden automaatiosta ei välttämättä ole sillä tasolla, että automaatiolaitteiston kytkeminen teolliseen esineiden internettiin olisi ongelmattonta. Teollisuuden automaatoratkaisujen tuottaminen on kuitenkin niin kilpailtu ala, että automaatio-osaamisen laajentaminen teollisen esineiden internetin ratkaisuihin pienissä ja keskisuurissa yrityksissä lisäisi automaatiotuotteiden arvoa niin asiakkaalle kuin palvelujen tuottajalle.

## 1.1 Tutkimusongelma

Kuten edellisessä kappaleessa mainittiin, voi teollisen esineiden internetin maailma automaatio suunnittelijoille olla vielä vierasta ja informaatioteknologian alalla vastaavasti on automaatiotekniikka osin vierasta. Tällä yhtälöllä saadaan yleensä tuhlattua rahaa eikä lopputulokseen ole välttämättä toivotunlainen. Kaupalliset järjestelmät taas voivat olla liian laajoja ratkaisuja ja siten myös kalliita pienille ja keskisuurille yrityksille. Teollisuuden automaatoratkaisujen tuottajilla onkin haaste löytää hinnaltaan sopiva räätälöity ratkaisu jokaiselle IIoT-ratkaisuja tarvitsevalle.

Ohjelmoijat niin IT- kuin OT-puolella ovat tottuneet ohjelmointikielien standardeihin, mikä on vakiinnuttanut ohjelmien kirjoitusta ja edesauttanut ymmärtämään



muiden kirjoittamaa ohjelmakoodia. IT-ohjelmakoodi voi automaatio-suunnittelijoille kuitenkin olla sen verran vierasta, että rajapinnan tekeminen automaatiolaitteiston ja IIoT-laitteiston välille ei onnistu kovin helposti. Tämän takia joudutaan turvautumaan valmiisiin ratkaisuihin, jotka eivät ole kustannustehokkaita eivätkä välttämättä ominaisuuksiltaan yhtään parempia kuin esimerkiksi avoimeen lähdekoodiin perustuvat ohjelmistot.

## 1.2 Tutkimuksen rajaukset

Tutkimusongelma on hyvin laaja, mutta tässä tutkimuksessa keskitytään ARM-arkkitehtuuriin ja Linux-pohjaisen tiedonkeruualustan integroimiseen ohjelmoitavaan logiikkaan avoimeen lähdekoodiin perustuvien ohjelmistojen avulla. Tarkoituksena on selvittää ja etsiä paikalliseen tiedonkeruuseen sopivat, helpot ohjelmointialustat, joiden konfigurointi tiedonkeruualustalle onnistuu automaatio-suunnittelijoiden lisäksi myös vähemmän ohjelmointia osaavilta henkilöiltä. Teorian tukemiseksi tutkimuksessa rakennetaan simuloitu ”prosessikone”, joka tuottaa dataa tiedonkeruulaitteelle. Testikokoonpanoon kuuluu Siemens TIA-1500 PLC, Raspberry Pi4 -tietokone ja Tosibox-reititin.

## 1.3 Tutkimuskysymykset

Tutkimuksessa haetaan vastauksia seuraaviin tutkimuskysymyksiin:

1. Voidaanko low cost -ohjelmistoilla luoda ARM-arkkitehtuuriin perustuva tiedonkeruualusta, jonka integroinnissa ohjelmoitavaan logiikkaan teollisissa automaatiotratkaisuissa lähestytään valmisjärjestelmää (OTS)?
2. Millaisia etuja ja haasteita liittyy avoimen lähdekoodin ohjelmistojen käyttämiseen tiedonkeruualustan integroinnissa ja miten nämä vaikuttavat ratkaisujen kustannustehokkuuteen?
3. Voidaanko teknologiaa kehittämällä löytää ratkaisuja helpottamaan työvoimapulaa teollisen internetin kehityksessä?

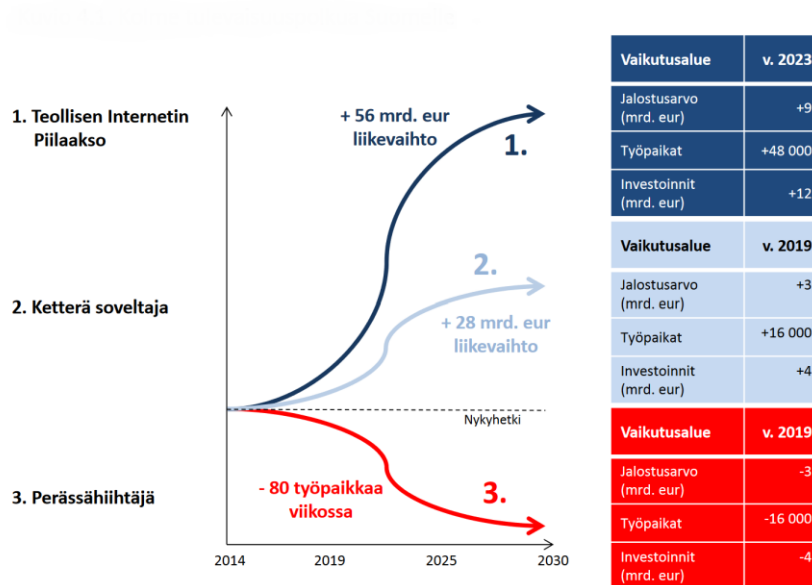
## 2 Teollinen internet ja työllisyys

Tiedonkeruu teollisuudessa ei ole uusi keksintö. Tuotanto- ja prosessilinjoista on kerätty tietoa automaattisesti tuotannon parantamiseen, ennakoivan huollon tueksi ja myös valmistuksen ja tuotannon ohjausjärjestelmiin vuosikymmeniä suurissa tuotantokoneissa. Voidaan sanoa, että ennen automaattista tiedonkeruuta tietoa kerättiin käsin, tuotantoparametrejä luettiin muistiinpanoista ja vaihdettiin aina tuotteen mukaisesti. Nykyään tuotantoa tehostavat toimet lasketaan PLC:ssä ja asetetut parametrit voivat olla PLC:n tai HMI:n muistissa valmiina. Vieläkin uudemmissa laitteissa käytetään hyväksi IloT:a, joka voi analysoida prosessidatan ja sillä perusteella myös muokata prosessiparametrejä tuotannon tehostamiseksi reaaliaikaisesti. IloT:n kasvun on ennustettu vaikuttavan räjähdysmäisesti yritysten markkinoihin ja aloittavan neljännen teollisen vallankumouksen aikakauden (Collin & Saarelainen, 2016a). Laitteiden päivitykset IloT-ympäristöön ovat kuitenkin melko kalliita ja teollisen internetin ammattilaisia on vähän, joten ainakaan toistaiseksi ei oikein voida puhua teollisesta vallankumouksesta vaan pikemminkin teollisen internetin aikakauden alkamisesta.

### 2.1 Teollisen internetin Piilaakso?

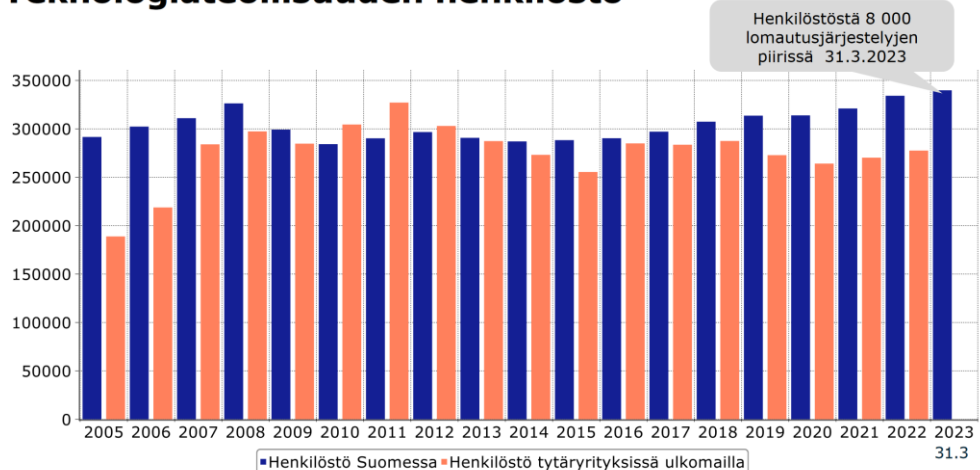
Vuonna 2015 valtioneuvoston kanslia julkaisi raportin hankkeesta, jonka tarkoituksena oli tutkia suomalaista teollista internettiä ja sen tuomia uhkia ja mahdollisuuksia, vaikutuksia työllisyyteen ja kehittämisen kannustumia ja esteitä (Valtioneuvosto, 2015a). Useista asiantuntijoista koostuva työryhmä visioi suomalaisen teollisuuden kehittyvän teollisen internetin mallimaaksi ja näin kasvattavan teollisuuden työpaikkojen määrää vuoteen 2023 mennessä 48000 henkilöllä ja lisäävän teollisuuden liikevaihtoa 56 miljardilla eurolla (Kuva 1). Teknologiateollisuuden toukokuussa 2023 julkaiseman väliraportin mukaan tavoite näyttäisi toteutuneenkin (Kuva 2). IT-osaajien määräkin on kasvanut tasaisesti vuodesta 2015, mutta toisaalta sähkö- ja elektroniikka-alan osaajien määrät ovat jopa laskeneet vuoden 2015 tasosta (Teknologiateollisuus, 2023). Osaltaan sähkö- ja elektroniikkaosaajien määrään vaikuttaa varmasti perusteollisuuden tuotteiden viennin väheneminen koronapandemian ja Ukrainan sodan myötä tyrehtyneen

Venäjän kaupan takia. Raportti ei kuitenkaan kerro, onko IT-ammattilainen pää-  
tynyt teollisen internetin osaajaksi vai joihinkin muihin teollisuuden IT-tehtäviin.



Kuva 1. Valtioneuvoston kanslian visio Teollisen Internetin kasvusta Suomessa (Valtioneuvosto, 2015a, s. 20).

## Teknologiäteollisuuden henkilöstö



1.5.2023

Teknologiäteollisuus

Lähde: Tilastokeskus, Teknologiäteollisuus ry:n henkilöstötiedustelu, Macrobond

2

Kuva 2. Teknologiäteollisuuden henkilöstömäärän kehitys 2023 (Teknologiäteollisuus, 2023, s. 2).

## 2.2 Teollisen internetin haasteet

Ei ole yllättävää, että Suomen johtavat teknologiayritykset ovat jo kehittäneet omia alustojaan tai ovat pitkällä omassa teollisen internetin kehitysprosessissaan. Pienemmille yrityksille teollisen internetin hyödyntäminen tuottavuuden lisäämiseksi oli vuoden 2015 raportin keskeisiä teemoja. Tärkeä kysymys oli, kuinka saada yritykset muuttamaan toimintatapojaan ja omaksumaan uudenlaisen palveluliiketoiminnan ajattelu. Valtioneuvoston raportista ilmeni, että yritykset, jotka eivät olleet valmiita muuttamaan toimintaansa uudenlaisen palveluliiketoiminnan suuntaan, ovat vaarassa jäädä jälkeen kilpailussa ja joutua tekemään leikkaustoimenpiteitä ja kustannussäästöjä (Valtioneuvosto, 2015b)

Haasteet eivät kuitenkaan rajoitu pelkästään muutoksen vastustukseen. Teollisen internetin asiantuntijoiden löytäminen osoittautuu vaikeaksi tehtäväksi, sillä IT-ala kärsii jatkuvasta työvoimapulasta eikä tilanne ole sen parempi teollisuuden automaatio-osaajienkaan osalta (TEM raportti, 2023). Automaatiosuunnittelijoiden IT-aidot ja teollisen internetin osaaminen vaihtelevat huomattavasti. Voidaan sanoa, että tietotekniikan taidot (IT) ovat heikommalla tasolla suunnittelijoilla, jotka ovat erikoistuneet ohjelmoimaan ohjausjärjestelmiä (PLC), kuin IT-suunnittelijoilla (Bolton, W, 2015). Osa automaatiosuunnittelijoista voikin kokea oman substanssiosaamisensa ulkopuolelle ulottuvat tehtävät vieraina, kun he kokevat osaamistasonsa heikoksi, mikä tekee tehtävistä taloudellisestikin kannattamattomia.

### 2.2.1 Haasteisiin vastaaminen

Pienten ja keskisuurten teknologiayritysten osalta, jotka ovat tämänkin tutkimuksen keskiössä, teollisen internetin mukanaan tuomat muutokset tuovat haasteita erityisesti hallinnollisella tasolla. On tärkeää rohkaista yrityksiä tekemään päätöksiä tuottavuuden parantamiseksi, vaikka riskinotto ja kustannukset saattavat huolestuttaa. Nopeasti muuttuvassa maailmassa pysyminen vaatii kuitenkin ketteryyttä ja valmiutta muutoksiin (VTT, 2018). Johtoportaan sitoutuminen ja tuki uusille hankkeille ja teknologisille muutoksille voi auttaa yritystä ylittämään

mahdolliset vastustukset ja haasteet. Kunnollinen viestintä organisaation sisällä on tärkeää, jotta kaikki ymmärtävät, miksi muutokset ovat tarpeellisia ja mitä niillä tavoitellaan (Valtioneuvosto, 2015c).

### 2.2.2 Yritysten välinen yhteistyö

Digitaalisen murroksen haasteisiin voidaan vastata myös yritysten välisellä yhteistyöllä, kuten IT- ja OT-palveluiden tuottajien tiiviillä yhteistyöllä. Tällainen yhteistyö voi avata uusia mahdollisuuksia teknologisten ratkaisujen kehittämisessä ja tarjota monipuolista asiantuntemusta eri aloilta. IT-palveluiden tarjoajat voivat tuoda yhteistyöhön digitaalisen osaamisensa ja teknologiset innovaationsa, kun taas OT-palveluiden tuottajilla on syvää tietämystä teollisuuden toimintaprosesseista ja vaatimuksista (Valtioneuvosto, 2015d) (Collin & Saarelainen, 2016b).

Yksi tapa hallita teollisen internetin murrosta on investoida osaamisen kehittämiseen. Suunnittelutoimistot voivat esimerkiksi tarjota työntekijöilleen koulutusta ja koulutusohjelmia, jotka auttavat heitä ymmärtämään paremmin IIoT:n mahdollisuuksia ja haasteita. Tämä voi auttaa yritystä omaksumaan uusia käytäntöjä ja hyödyntämään teknologian tarjoamia etuja. Monet suunnittelutoimistot ovatkin erikoistuneet teollisen internetin ratkaisuihin ja päivityksiin, mikä tekee niistä arvokkaita kumppaneita pienyrityksille ja miksei suuremmillekin yrityksille. Näillä toimistoilla on usein jo olemassa olevaa asiantuntemusta ja kokemusta erilaisten tuotanto- ja prosessilaitosten päivityksistä. Heidän osaamisensa avulla asiakasyritykset voivat saada tarvittavaa apua ja neuvontaa, kun he pyrkivät ottamaan käyttöön IIoT:n tuomia mahdollisuuksia.

### 2.2.3 Teknologiset ratkaisut

Digitaalisen murroksen haasteita voidaan myös hallita teknologian avulla. Avoin lähdekoodi ja standardit mahdollistavat eri järjestelmien yhteensopivuuden ja vähentävät suljettujen ekosysteemien aiheuttamaa kitkaa (L. O. Aghenta et al., 2019) (Martikkala et al., 2021a). Vakiintuneet työmenetelmät puolestaan tarjoavat tehokkaan tavan organisoitua ja toteuttaa digitaalisia muutoksia

(Automaatioseura, 2013). Kaikki nämä tekijät yhdessä helpottavat ja edistävät yritysten kilpailukykyä.

#### 2.2.4 Halvat ratkaisut ohjaamaan investointeja

VTT:n vuonna 2021 teettämän tutkimuksen mukaan pienet ja keskisuuret yritykset ovat haluttomia investoimaan uuteen teknologiaan. 200 pk-yritykselle tehdystä haastattelututkimuksesta vain 10 prosenttia aikoi investoida uusin digitalisaatiota hyödyntäviin teknologioihin (VTT verkkotiedote, 2023). On kuitenkin selvää, että tulevaisuudessa datan tarve ja mielenkiinto koneista tulevaan dataan lisääntyy entisestään. Data voidaan jalostaa informaatioksi analysoimalla datapaketteja. Näin datasta saadaan hyötyä reaaliaikaisesti esimerkiksi koneen säätöpiireille, kunnossapitoon tai laajempiin kokonaisuuksiin, kuten koneiden tuotannonohjaukseen ja yrityksen toiminnanohjaukseen. Prosessikoneen PLC:n tuottaman datan tallentaminen tietokantaan tai tiedostoon paikallisesti tai hajautetusti on teollisen internetin selkäranka. Kuten J. Collin et. al. Teollinen internet -kirjassa toteavat, "Ilman dataa ei ole mitään". Toisin sanoen data on raaka-ainetta, josta jalostetaan informaatiota hyödyttämään ja kehittämään liiketoimintaa (Collin & Saarelainen 2016c).

Erilaisia teknologioita, joiden avulla dataa voidaan tuottaa, on tarjolla laaja valikoima. Useissa tuotanto- ja prosessilaitteissa on jo valmiina prosessiin liittyvät anturit ja mittaukset, mikä saattaa vähentää tarvetta erillisille älykkäille antureille. Sen sijaan tarvitaan tapoja siirtää koneiden tuottamaa dataa tehokkaasti kohti teollisen internetin tuottamia palveluita, mikä vaatii investointeja ja ratkaisuja. Onkin pohdinnan arvoista, voisiko digitalisaation omaksumista edistää esimerkiksi valmisjärjestelmien (Off-the-shelf solution, OTS) avulla datan keräämiseen tuotanto- ja prosessilinjalta.

Taulukko 1. OTS-järjestelmän implementointivaihtoehdot (Perko, 2021).

<b>Valmisjärjestelmän implementointitapa</b>	<b>Kuvaus</b>
<b>"Out-of-the-box"</b>	Ohjelmisto asennetaan ja käytetään sellaisenaan
<b>Konfiguroitu</b>	Säädetään ohjelmiston asetuksia tarpeiden mukaan kirjoittamatta kuitenkaan uutta koodia
<b>Integroitu</b>	Integroidaan ohjelmisto johonkin olemassa olevaan järjestelmään. Vaatii yleensä kustomointia (muokkausta) liittymätyössä.
<b>Laajennettu</b>	Kehitetään lisätoiminnallisuutta kustomoimalla koodia laajentaen ohjelmiston kyvykkyyttä vastaamaan tarpeita.

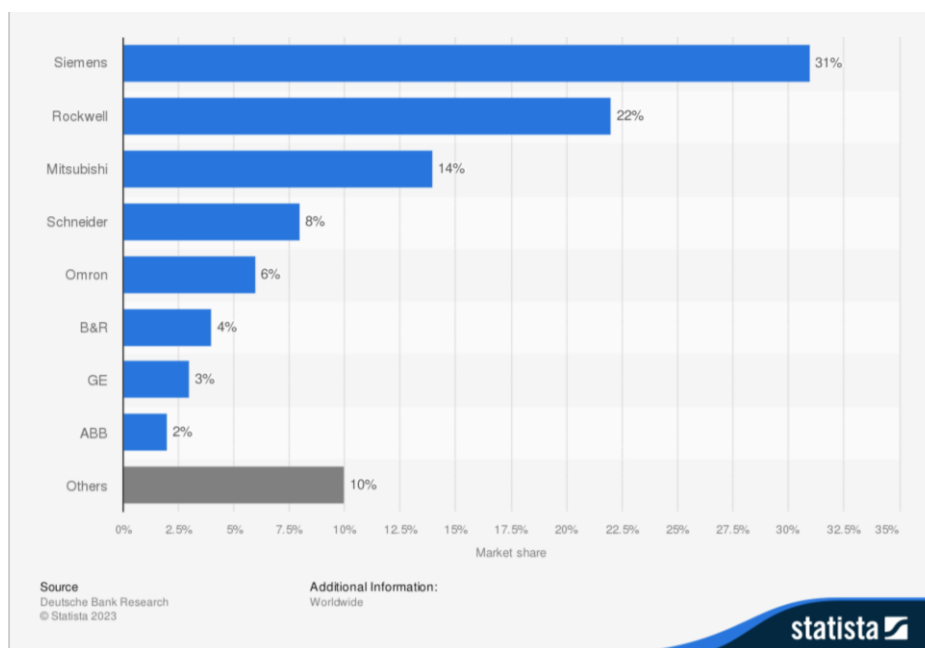
Tällainen valmiiksi esiohjelmoitu Taulukon 1 mukainen geneerinen datankeruusovellus voisi mahdollistaa digitaaliseen murrokseen vastaamiseen myös niissä yrityksissä, jotka ovat olleet epäröiviä investointien suhteen. Yhteistyössä IT- ja teollisuuspalveluiden tuottajien kanssa sekä hyödyntämällä avointa lähdekoodia ja standardeja yritykset voisivat löytää edullisia ja toimivia ratkaisuja, jotka helpottavat digitaalista murrosta ja tuovat lisäarvoa niin liiketoiminnalle kuin tuotannollekin.

### 3 Teollisen internetin työkalut

Suuret ohjelmistojen tuottajat tarjoavat laajoja ja monipuolisia ratkaisuja teollisen internetin työvälineiksi, kuten GE iFix, AVEVA Edge tai Siemens WinCC Professional. Edellä mainitut ovat SCADA-järjestelmiä, joissa on valmiina OPC UA-rajapinta, historiatyökalut datan tallennukseen, trendikuvaajat tiedon visualisointiin ja paljon muita lisäominaisuuksia. Usein hankintahinta käyttölisensseille näilläkin ominaisuuksilla on varsin huokea, mutta kehitysympäristöstä ohjelmistoyhtiöt perivät ohjelmistoista vuosittaisia lisenssimaksuja, jotka kohoavat useisiin tuhansiin euroihin. Esimerkiksi Aveva Edge:n kehityslisenssin hinnoittelu jaetaan kolmeen eri kategoriaan järjestelmän muuttujamäärien mukaisesti. Hinnot lisensseille syksyllä 2023 olivat halvimmasta kalleimpaan 4240 Yhdysvaltojen dollaria, 10070 Yhdysvaltojen dollaria ja 22425 Yhdysvaltojen dollaria (Avevaselect Edge, 2023). Tämä tulee taakaksi suunnittelutoimistojen asiakkaille kustannusten nousuna, kun pienenkin laitteiston rajapinnaksi vaaditaan teollisuus-PC tiedonkeruualustaksi ja kehityslisenssien kustannukset siirtyvät myös asiakkaalle suunnittelukustannuksina.

Puhuttaessa pienistä tuotanto- ja prosessikoneista ovat datan keruu, yksinkertainen visualisointi ja perustiedon tallennus ensimmäinen askel teollisen internetin suuntaan. Tässä tapauksessa suuret ja monimutkaiset SCADA-järjestelmät voivat olla ylimitoitettuja ja kalliita. Suurilla teollisuuden automaatiovalmistajilla, kuten Siemensillä, Rockwellillä ja Mitsubishiillä, oli vuonna 2017 kaksi kolmasosaa maailman PLC-markkinoista (Statista, 2017) (Kuva 3). Teollisen internetin kannalta tarkasteltaessa näiden toimijoiden järjestelmät ovat suljetumpia ja voivat olla hankalampia liittää avoimille tiedonkeruualustoille, mutta niillä on omat erityispiirteensä, vahvuutensa ja painopistealueensa, jotka voivat tehdä niistä houkuttelevia vaihtoehtoja joillekin asiakkaille ja sovellusratkaisuille. Valinta riippuu kuitenkin usein siitä, minkä tyyppisiä tarpeita yrityksellä on ja millainen automaatio- ja ohjausratkaisu parhaiten sopii kyseisiin tarpeisiin. On kuitenkin selvää, että näillekin alustoille tarvitaan jossain vaiheessa päivitys seuraavalle tasolle, teollisen internetin suuntaan.





Kuva 3. Suurimmat PLC-valmistajat 2017 (Statista, 2017).

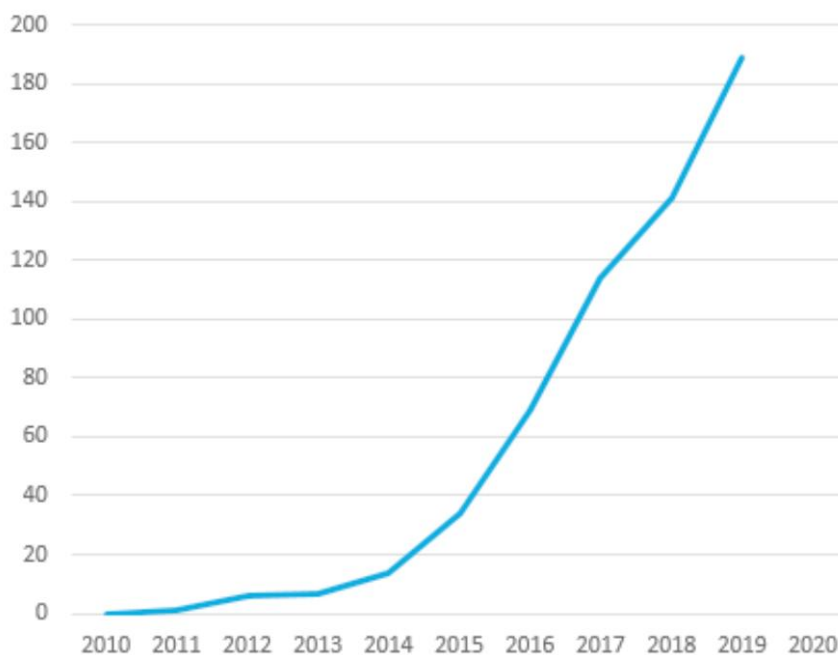
### 3.1 Low cost -teknologia

Low Cost -ratkaisut, kuten Raspberry Pi -pohjaiset järjestelmät tai muut edulliset automaatiotuotteet, voivat tarjota tarvittavat toiminnot huomattavasti alhaisemalla hinnalla. Pienten koneiden kohdalla ylläpito- ja lisenssikustannukset voivat muodostaa merkittävän osan kokonaiskustannuksista. Halpahintaiset ratkaisut voivat tarjota yksinkertaisemman ja edullisemman lisenssimallin sekä usein myös avoimen lähdekoodin vaihtoehdon, mikä voi vähentää vuosittaisia kustannuksia merkittävästi. Tämä on erityisen tärkeää pienille yrityksille ja suunnittelu-toimistoille, jotka usein joutuvat tarkkaan punnitsemaan laitekustannuksia.

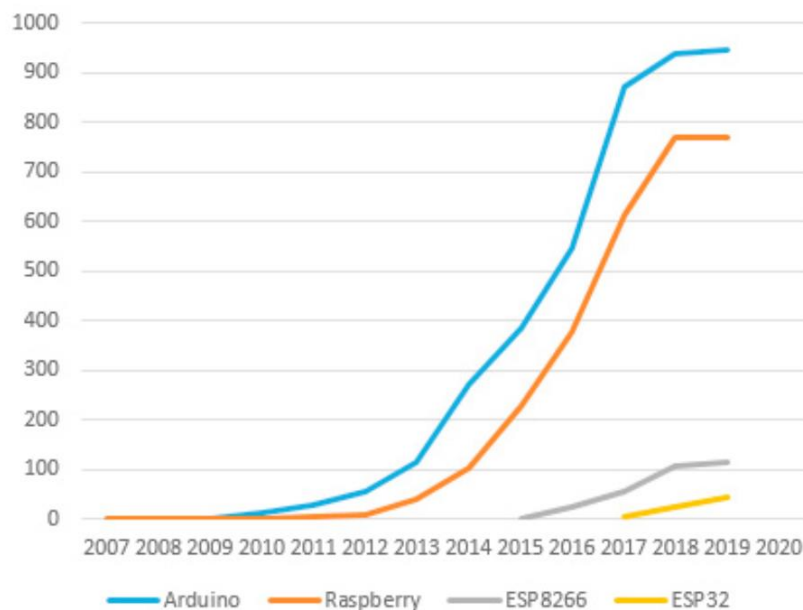
#### 3.1.1 Low cost -tietokoneet

Teollisuuden automaatiossa yksi huomattava tekijä low cost -ratkaisuihin ovat edulliset avoimen lähdekoodin Linux-käyttöjärjestelmäiset ARM-tietokonealustat. Nämä tietokoneet tarjoavat vaihtoehdon perinteisille PLC-järjestelmille ja ovat kasvattaneet suosiotaan teollisuuden digitalisaation tarpeissa, kuten myös yksityiskäytössä (Counterpoint, 2023). Martikkala et al. (2021) tutkivat

Tampereen ja Trondheimin yliopiston yhteistutkimuksessa halpateknologioiden ja avoimen lähdekoodin suuntauksia viime vuosien aikana. Tutkimuksen tuloksena nähtiin, kuinka valtavasti mielenkiinto avoimeen lähdekoodiin ja halpateknologioihin on lisääntynyt (Kuva 4 ja Kuva 5) (Martikkala et al., 2021b).



Kuva 4. Avoin lähdekoodi IoT:n kehityksessä (Martikkala et al., 2021b).



Kuva 5. Internet-haut halpateknologia-alustoista (Martikkala et al., 2021b).

Useat tunnetut valmistajat, kuten WAGO, Beckhoff, Bosch ja Phoenix Contact, ovat tuoneet markkinoille omia versioitaan näistä tietokoneista. ARM-pohjaiset prosessorit tunnetaan muun muassa energiatehokkuudestaan ja ne sopivat hyvin esimerkiksi mobiililaitteisiin, sulautettuihin järjestelmiin ja IoT-laitteisiin.

### 3.2 Soft-PLC

Soft-PLC on hyvä esimerkki teollisuuden sulautetusta järjestelmästä, kun teollisuustietokoneeseen ladataan PLC-ohjelmisto ja korvataan olemassa oleva erillinen PLC-laitteisto. Suuremmista PLC-valmistajista vain Siemensillä on ohjelmistopohjainen PLC saatavilla, mutta vain Windows-käyttöjärjestelmille (Siemens cpu150x, 2019). Vaikka Soft-PLC ei ole tämän tutkimuksen varsinaisena pääaiheena, on kuitenkin hyvä tiedostaa, että Soft-PLC-ratkaisujen tarjoajia on useita ja niillä voidaan tehdä kustannustehokkaasti linjapäivityksiä niin kuin tavallisilla ohjelmoitavilla logiikoilla. Seuraavassa on kaksi esimerkkiä Linux-alustalle saatavista sulautetun järjestelmän PLC-vaihtoehtoista.

### 3.2.1 OpenPLC

OpenPLC on ilmainen verkosta ladattava ilmainen avoimeen lähdekoodiin perustuva logiikkaohjain, joka tukee IEC-61131-3-kieliä. Tälle järjestelmälle on varmasti paikkansa, mutta ohjelma ei tue muita kaupallisia Ethernet-pohjaisia kommunikointiprotokollia kuin Modbus/TCP:tä. Muiden Ethernet-protokollien ja lisäksi Modbus:n toimintakoodien tuen puuttuminen (Alves & Morris, 2018) sulkee sen ulkopuolelle tuotantolinjojen IIoT:n jälkiasennusprojekteissa, joissa on esimerkiksi käytetty Profinet- tai EthernetIP-väyläliityntäistä hajautus-I/O:ta.

### 3.2.2 Codesys

Codesys on saksalaisen 3S-Smart Software Solutions:in kehittämä (Codesys, 2020), myös avoimeen lähdekoodiin perustuva Soft-PLC, joka tukee myös ohjelmistoillaan teollisuuden sulautettuja järjestelmäratkaisuja. IEC 61131-3-ohjelmointikielten lisäksi se tukee myös IT-ohjelmointikieliä. Codesys on perusversioltaan ilmainen, mutta siinä on saatavilla myös maksullisia lisäosia ja laajennuksia, jotka tarjoavat lisätoiminnallisuuksia ja työkaluja erilaisiin teollisuusautomaation projekteihin. Maksulliset lisäosat voivat kattaa esimerkiksi erilaisia kommunikointiprotokollia, tietoturvaomintoja tai muita erityistarpeisiin liittyviä ominaisuuksia. Edellä mainitut laitevalmistajat WAGO, Beckhoff, Bosch ja Phoenix Contact käyttävät PLC-alustanaan Codesys:stä räätälöityä ohjelmointialustaa (WAGO, 2023) (Beckhoff, 2023) (Bosch, 2023) (PhoenixContact, 2023). Näissä järjestelmissä on lisättyjä toiminnallisuuksia ja laitteiston erityispiirteitä, mutta niiden ohjelmoitava logiikka ja käyttöliittymä perustuvat Codesys:iin.

## 3.3 Ohjelmoitavat sovellusrajapinnat

Ohjelmoitava sovellusrajapinta (API) on keskeinen osa tiedonkeruuprosessia. Se toimii liitännänä, joka mahdollistaa eri järjestelmien ja ohjelmistojen välisen kommunikaation ja datan vaihdon reaaliajassa prosessikoneelta tietokantaan tai muuhun tallennuspaikkaan (Termipankki, 2023).

Tässä tutkimuksessa keskitytään low cost -alustalla toimivaan tiedonkeruuraikaisuun Siemens S7-1500 -PLC:n ja Raspberry Pi4:n välillä. API-rajapinnan ohjelmointiin on saatavilla runsaasti vaihtoehtoja, jotka ovat yhteensopivia teollisuuden laitteiden kanssa kommunikointiin. Seuraavassa kappaleessa vertailaan muutamia esimerkkejä API-rajapinnoista, joita voidaan käyttää tämän tutkimuksen tekemiseen.

### 3.3.1 ISO-on-TCP

ISO-on-TCP on avoimen lähdekoodin sovitin Siemens S7 -PLC-laitteiden kommunikointiin. Se käyttää kommunikoinnissa RFC1006-protokollaa ja Siemensin käyttäjille protokollan tutumpi nimi on Snap7. Se on suunniteltu selättämään OPC-palvelimien rajoitukset siirrettäessä suuria määriä nopeaa dataa teollisuuslaitoksissa, ja se skaalautuu myös hyvin pieniin Linux-pohjaisiin ARM- ja Mips-levyihin, kuten Raspberry Pi (myös 1 ja 2), BeagleBone Black, pcDuino, CubieBoard, UDOO ja ARDUINO YUN. Snap7 tukee myös useita ohjelmointikieliä, kuten C/C++, .NET/Mono, Pascal, LabVIEW, Python ja Node.js. (Snap7, 2023) ISO-on-TCP tuntuukin tämän tutkimuksen kannalta varsin hyvältä vaihtoehdolta. Vaikka tutkimuskohteena onkin Siemens S7 -PLC-laitteen kommunikointi, niin on hyvä ottaa huomioon muiden valmistajien teollisuuden Ethernet-protokollien liittäminen tiedonkeruualustaan.

### 3.3.2 OPC-UA

OPC-UA (Unified Architecture) on yhteensopivuusstandardi turvalliseen ja luotettavaan laitteiden väliseen kommunikointiin teollisuusautomaatioalueelle (OPC Foundation, 2023). Sitä voidaan käyttää Siemensin laitteiden ja Raspberry Pi -tietokoneiden välisen tiedonkeruun ja viestinnän toteuttamiseen. Tutkimuksessa hakukonehaut eivät tuottaneet varteenotettavia tuloksia, jossa Siemens S7-1500-sarjan logiikoille olisi Linux-pohjaisia low cost OPC-UA-server-ratkaisuja tuotantokoneisiin. OPC-UA-servereitä on ladattavissa simulointimahdollisuudella, mutta kaupallinen käyttö yleensä maksaa 500 eurosta ylöspäin.

### 3.3.3 Modbus/TCP

Modbus on Modiconin teollisuuden tarpeisiin kehittämä avoimen lähdekoodin väyläprotokolla. Se toimittaa viestejä kahden palvelimen ja asiakaslaitteen välillä molempiin suuntiin. Modbus/TCP on yleisesti käytössä teollisuuden laitteissa ja sen etuna onkin sen toimivuus useiden eri valmistajien laitteiden kanssa (ModbusTCP, 2023). Tutkimuksen kannalta tässä protokollassa on hyvänä puolena se, että lähes kaikki teollisuuden PLC-laitteet tukevat ModbusTCP:tä. Huonona puolena on, että tutkimuskohteena olevan Siemens S7:n vanhemmille 300-sarjan logiikoille tarvitaan maksullinen lisenssi (Siemens Modbus/TCP, 2016). Sen vuoksi kyseisissä päivitysprojekteissa on suotavaa harkita toisen kommunikointiprotokollan käyttöä.

### 3.3.4 MQTT

MQTT (Message Queuing Telemetry Transport) on avoimeen lähdekoodiin perustuva viestintäprotokolla. Tämän protokollan etuna on, että se pyrkii minimoimaan laitteiston resurssivaatimukset sekä käytettävän kaistanleveyden tiedonsiirrossa (MQTT, 2019). Lisäksi MQTT:n käyttö on sen energiatehokkuuden ansiosta lisääntynyt mobiililaitteiden IoT-sovelluksissa (Metropolia, 2016). Myös suurimmat PLC-valmistajat ovat lisänneet tämän protokollan IoT-kommunikoinnin vaihtoehdoksi, joten se on tutkimuksen kannalta myös hyvä vaihtoehto Siemens S7-1500 -kommunikoinnin rajapinnaksi.

### 3.3.5 REST API

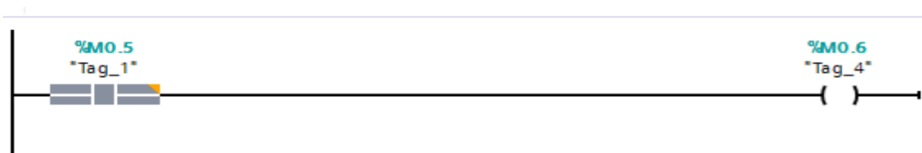
REST API on yleinen tapa tarjota ja kuluttaa palveluita verkon yli. REST-arkkitehtuurilla voidaan luoda rajapintoja sekä Siemensin että Raspberry Pi:n ohjelmistoissa ja käyttää niitä tietoturvallisesti tiedonkeruun ja viestinnän välineenä, käyttäen tiedonsiirtovälineenä HTTPS-protokollaa. REST API on kuitenkin enemmän suunnattu mobiili- ja verkkosovellusten käyttöön, kuten web-pohjaisiin käyttöliittymiin tai tiedonkeruuseen verkon yli (Siemens REST API, 2019). Tutkimuksen kannalta HTTPS-protokolla olisi hyvä vaihtoehto testaamiselle, jos

dataa siirretään internetin yli. Nyt käsitellään vain lähiverkossa reunapalvelimelle siirrettävää dataa, joten REST API jää tutkimuksen ulkopuolelle.

### 3.4 Ohjelmointialusta ja ohjelmointimetodit

Kun puhutaan reunatietokoneissa halvoista ratkaisuista, on muistettava, että suorituskyky koneissa ei ole paras mahdollinen. On myös otettava huomioon suunniteltaessa geneeristä tiedonkeruulaitteistoa, että eri taitotason omaavilla suunnittelijoilla, käyttöönottajilla tai asennusvalvojilla, on vain vähäiset tai ei välttämättä lainkaan ohjelmointitaitoja. Koska IT-ohjelmointikielien taitajia on vähän, on myös oletettava, että ohjelmoijien saaminen käyttöön on haastavaa. Pitäisi siis löytää ohjelmointikieli, joka sopii vähemmän ohjelmointia tehneelle.

IEC 61131-3 -standardiin kuuluvat graafiset PLC-ohjelmointikieliset, kuten tikapuukaavio, toimilohkokaavio ja sekvenssivuokaavio, ovat tuttuja useimmille PLC-ohjelmoijista. Graafisia ympäristöjä pidetään tehokkaampina ja helpompina ymmärtää kuin tekstipohjaisia. IT-alalla tätä asiaa on tutkittu muun muassa opetusmenetelmissä. Jussi Nikanderin tutkimuksessa ”Visuaalinen ohjelmointi tietotekniikan perusopetuksessa” (2006) vertailtiin eri mittarein kolmen visuaalisen ohjelmointiympäristön soveltuvuutta opetusympäristöön. Nämä mittarit olivat: helppojen virheiden välitön huomiointi tai estäminen, vasta-alkajalle ymmärrettävä, yksinkertainen syntaksi, ja välitön palaute koodin kirjoittamisen aikana. (Nikander, 2006). Näitä mittareita voidaan soveltaa myös työympäristöön, oli kyseessä sitten PLC-ohjelmointiympäristö tai IT-ohjelmointiympäristö. Helpot virheet ja yksinkertainen syntaksi ainakin PLC-ympäristössä voidaan huomata vertaamalla graafista ladder-koodia (tikapuukaavio) ja struktuuritekstiä. Alla kuvissa näkyy esimerkki molemmista koodeista (Kuva 6, Kuva 7). Pitää kuitenkin muistaa, että tekstipohjaisilla kielillä voidaan kirjoittaa huomattavasti monimutkaisempia ohjelmarakenteita ja ne ovat myös pohja graafiselle ohjelmoinnille. Nämä eivät saa siis olla toisiaan poissulkevia vaihtoehtoja vaan ohjelmointitapa on hyvä valita tapauskohtaisesti.



Kuva 6. Ladder-ehdolause.

```

1 IF "Tag_1" THEN
2     "Tag_4" := TRUE;
3 ELSE
4     "Tag_4" := FALSE;
5 END_IF;

```

Kuva 7. Struktuuriteksti-ehdolause.

Graafisia ohjelmointiympäristöjä löytyy myös IT-ohjelmointiympäristöistä. Lasten ja nuorten opetuslustoat Scratch tai Codester ovat myös hyviä alustoja kaikenikäisten ohjelmoijien ohjelmistokehityksen harjoitteluun. Mutta myös ammattilliseen tekemiseen löytyy runsaasti erilaisia vaihtoehtoja liike-elämän ja organisaatioiden käyttöön, ohjaamaan liiketoimintaprosesseja, tekemään data-analyysiä, raportointia tai asiakashallintaa. Näitä ohjelmia ovat esimerkiksi Microsoft Power Apps, Zoho Creator ja Appian, vain muutamia mainittuna.

Hämeen ammattikorkeakoulun (HAMK) lehtori Tommi Saksa kertoo artikkelissaan HAMK:n lähestymistavasta ohjelmoitiin, jossa se suosii ohjelmoinnin aloituskielesi C#:tä, Javaa, JavaScriptia ja Pythonia. Alakouluissa suositetaan visuaalisia ohjelmointialustoja, kuten Scratchia, Codesteria tai Legoa. Visuaalisissa kielissä hyvänä puolena on, ettei tarvitse mennä syvälle koodauksen saloihin. Heikkoutena on, että niistä ei välttämättä löydy kaikkia tarvittavia ominaisuuksia. Yksinkertaisista ohjelmointikielistä Saksa mainitsee Pythonin, jolla on vahva jalansija työelämän ohjelmistoautomaatiikassa, datatieteissä ja koneoppimisessa. Pythonin heikkoudeksi voidaan lukea erilainen ohjelmarakenne verrattuna C#- ja Java-kieliin, mikä vaatii uuden oppimista näiltä osaajilta (Saksa, 2020). Toisaalta Python.org-sivuston lainausosiossa Journyx:n toimitusjohtaja Curt Finch kehuu Pythonia kymmenen kertaa Javaa ja sata kertaa C++ -ohjelmointia



tuottavammaksi (Python.org, 2023). Tätä väitettä on tosin vaikea todistaa, mutta myös monilla muilla it-alan yrityksillä on samankaltaisia ajatuksia: esimerkiksi intialainen ohjelmistoyritys PeakIn on samoilla linjoilla Journyx:n kanssa (PeakIn, 2023).

Pythonin lisäksi perinteisistä ohjelmointikielistä rajapintojen ohjelmointiin on hyvä vaihtoehto Javascript-notaatioon perustuva dataformaatti, Json, on kehitetty tiedonvälitykseen verkkosovellusten ja palvelinten välille. Json onkin suosittu API:n rakentamisessa. (Json-ohjelmointikieli, 2023)

Yhdistämällä visuaaliset ohjelmointikielet sekä teollisuudessa suosittu Pythonin ja API:en rakentamisessa suosittu Json:n voisi löytyä tehokas ratkaisu teollisuuden sovellusten kehittämiseen ja tiedonkeruuseen. Silloin vaikeammat rakenteet, jotka vaativat enemmän IT-taitoja, olisi valmiiksi ohjelmoitu, kun taas helpommat konfigurointitehtävät voisi tehdä vähäisemmilläkin ohjelmointitaidoilla.

### 3.4.1 Low-Code- ja No-Code-ohjelmointialustat

Graafisia ja puoliksi graafisia ohjelmointialustoja kutsutaan Low-Code- ja No-Code-ohjelmointialustoiksi, joiden tarkoitus on nopeuttaa ohjelmointia. LCNC-alustojen graafiset komponentit eivät ainoastaan nopeuta kehitysprosessia vaan myös helpottavat ohjelman ymmärtämistä. Henkilöille, joilla on vain vähän ymmärrystä ohjelmoinnista, graafiset komponentit tarjoavat selkeämmän kuvan siitä, mitä ohjelma tekee. Tämä voi helpottaa tiimityötä ja kommunikointia eri sidosryhmien välillä. Huonona puolena LCNC-alustoista voidaan todeta, että valmiisiin komponentteihin on vaikeampi tehdä muutoksia.

Teollisuuden sovellusrajapintana toimivia LCNC-alustoja ei ole tarjolla kuin avoimeen lähdekoodiin perustuva ilmaisohjelma Node-RED. Se on Low-Code-alusta, johon voidaan lisätä myös perinteisellä tavalla kirjoitettua koodia, valmiiden kirjastokomponenttien lisäksi.

Node-RED:n hyväksi voidaan sanoa, että kaikki sovellusrajapinnat, jotka esiteltiin kappaleessa 3.3, voidaan ladata ohjelmistoon valmiina kirjastokomponentteina. Tutkimuksen kannalta tämä myös helpottaa API:en vertailua, koska kaikki voidaan testata yhdellä valmiilla alustalla.

## 4 Tutkimusympäristö

Tutkimuksen pyrkimyksenä oli siis löytää yksinkertainen tiedonkeruualusta, joka on helppo integroida ohjelmitavaan logiikkaan. Tällä helposti integroitavalla datankeruuympäristöllä pyrittiin löytämään kustannustehokas ratkaisu tuotekehitykselle ja käyttöönoton tueksi, samalla haluttiin tuoda vaihtoehto pk-yrityksille vanhojen tuotantokoneiden päivittämisen kautta teollisen internetin hyödyntämiseen. Tutkimusympäristöön valittiin Raspberry Pi4:lle sopiva ohjelmointialusta, sovellusrajapinta Siemensin PLC:n ja RPi:n välille ja tiedonkeruumenetelmä paikalliseen tiedonkeruuseen.

### 4.1 Tiedonkeruun ohjelmointialustan valinta

Tiedonkeruun ohjelmointialustaksi valikoitui perinteisten ohjelmointikielien sijaan Node-RED, sillä sen valmiiden API-kirjastojen monipuolisuuden ansiosta pystytään hoitamaan suurin osa teollisuuden kommunikointirajapinnoista. Node-RED:n solmujen välinen tiedonsiirto perustuu json-notaatioon ja sen funktiosolmu mahdollistaa minkä tahansa Javascript-koodin suorittamisen (Node-RED, 2023). Se siis tukee myös perinteisellä ohjelmointityylillä kirjoitettua ohjelmakoodia, josta kerrottiin kappaleessa 3.4. Toinen syy on vakiintunut ja visuaalinen ohjelmointiympäristö. Kun ohjelmointi tehdään yhdellä alustalla, laitteiston toiminnot ja komponentit tulevat tutummiksi, mikä taas helpottaa ja nopeuttaa työn valmistumista ja laitteiston ylläpitoa. Rajapinta on tarkoitus luoda kappaleessa 2.2.4 esitellyn OTS-järjestelmän konfiguroitavalle tai integroitavalle tasolle.

Kolmas Node-RED:n käyttöä tukeva ominaisuus on alustariippumattomuus. Node-RED:iä voidaan käyttää Windows-, Mac- sekä Linux-pohjaisilla alustoilla.

Selainpohjaisen ohjelmointiympäristön ansiosta kaikille alustoille samanlainen ohjelmointiympäristö tukee myös työn tehokkuutta.

## 4.2 API-rajapintavertailu

PLC-laitteeseen integroitava API-rajapinta toimisi yhdyskäytävänä paikalliselle tiedonkeruulle ja antaisi optiona mahdollisuuden liittää reunapalvelin pilvipalveluun. Tämän rajapinnan tulisi soveltua sekä käytössä oleviin että uusiin tuotantokoneisiin siten, että integrointi kohdelaitteisiin onnistuisi vakiintuneella ja yhteisellä toimintatavalla. Ennalta ohjelmoidun ja helposti konfiguroitavan API-rajapinnan käyttöönotto voi olla ratkaiseva tekijä digitalisaation omaksumisessa ja sen hyötyjen maksimoinnissa. Tässä tutkimuksessa on tarkoituksena tehdä mahdollisimman laaja yksipiirilevytietokoneisiin perustuva tiedonkeruurajapinta, joka muuttuu ketterästi eri laitteille sopivaksi.

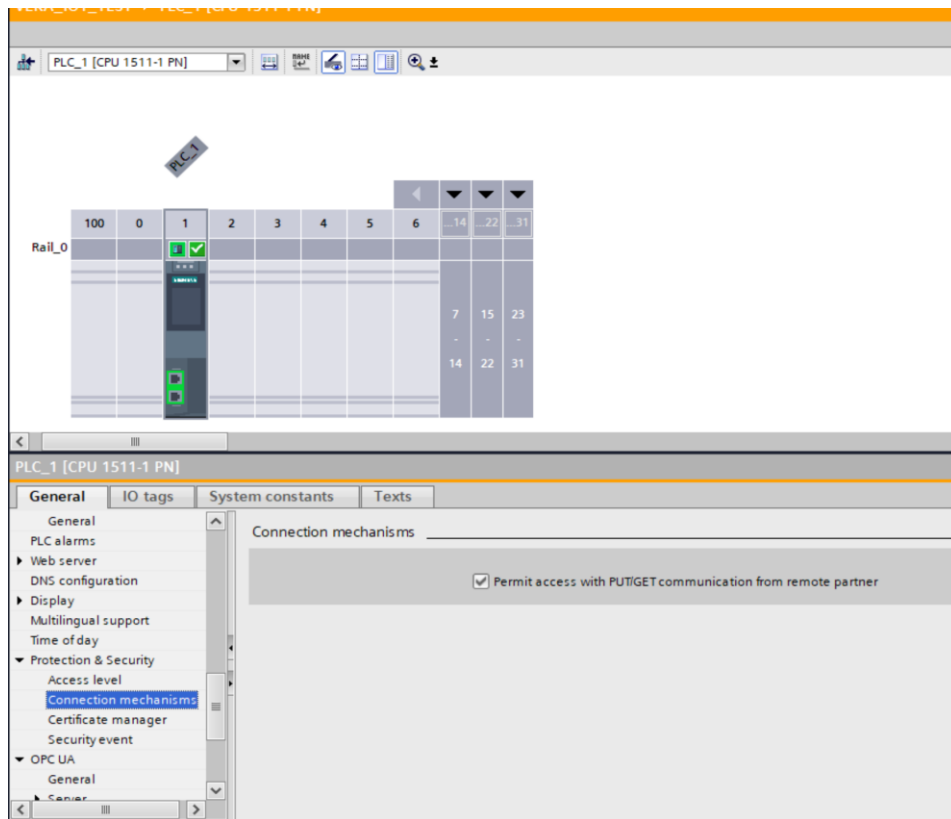
Kappaleessa 3.3 esitellyistä vaihtoehtoista ei pienellä tutkimisella löytynyt suuria eroja, joten tämä vertailu oli tehtävä käytännössä.

Siemensin OPC-UA Server -ohjelmisto on maksullinen, eikä Node-RED:stä löydy ilmaisvaihtoehtoa, joten sen käyttö ei tue tutkimuksen teemaa löytää kustannustehokasta ratkaisua, vaikka OPC-UA:n käyttö voi joissain tapauksissa on jopa järkevämpää. Toinen tutkimuksen ulkopuolelle jäävä rajapinta on REST API, joka tulkittiin sopivammaksi hieman erilaiseen käyttötarkoitukseen.

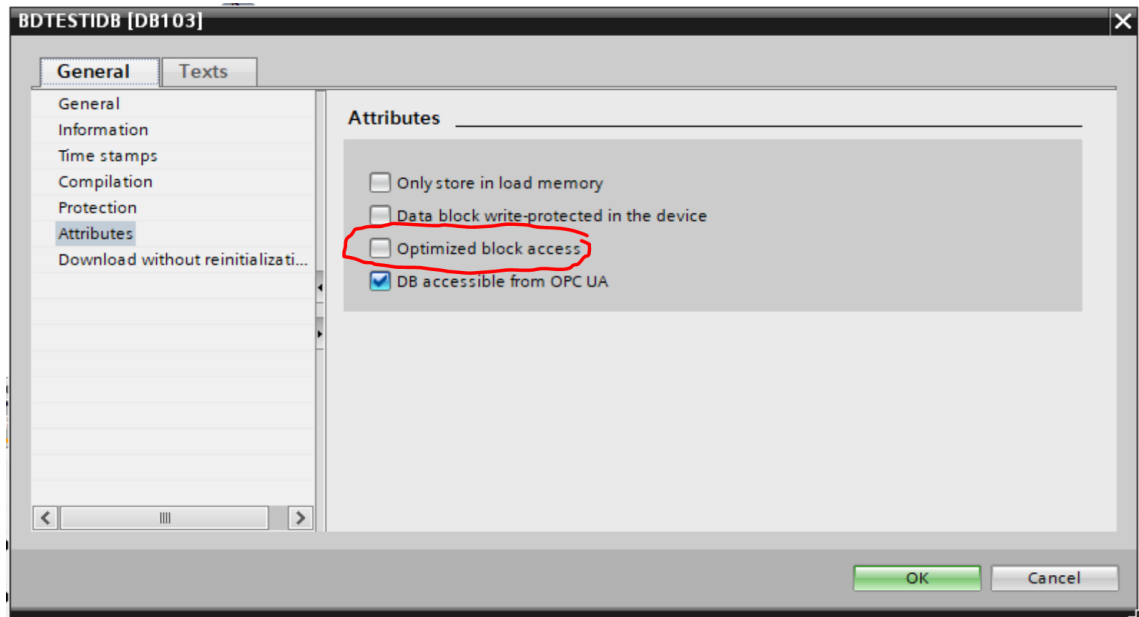
### 4.2.1 ISO-on-TCP

Ensimmäinen testattava API oli nimenomaan Siemensin kommunikointiin tarkoitettu ISO-on-TCP. Se olikin odotusarvon mukaisesti helpoin konfiguroitava tiedonkeruuseen. Käytännössä tämä Siemensin legacy-protokolla ei vaadi erillisiä ohjauslohkoja datan vaihtoon. Node-RED-kommunikointia varten on aktivoitava ominaisuus, joka sallii etäyhteyden kumppanilta, PUT/GET-kommunikointi (Kuva 8) ja 1500-sarjan logiikoiden ominaisuus datalohkojen optimoinnin poisto (Kuva 9). Datalohkojen optimoinnin poisto pitää tehdä, koska Node-RED:n API

ei pysty lukemaan symbolisia osoitteita vaan tarvitaan numeeriset osoitukset datapaikan osoitukseen.

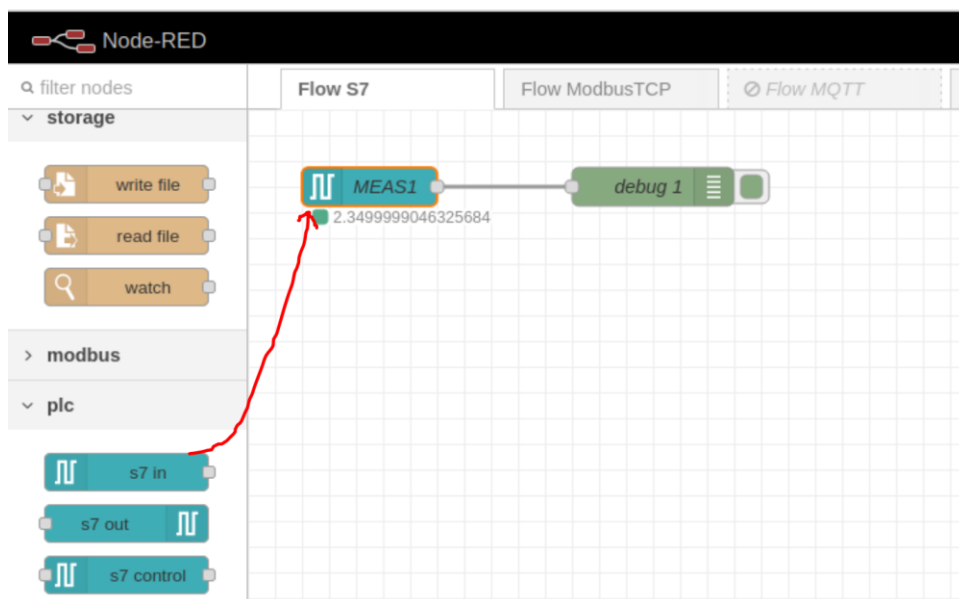


Kuva 8. PUT/GET aktivointi.

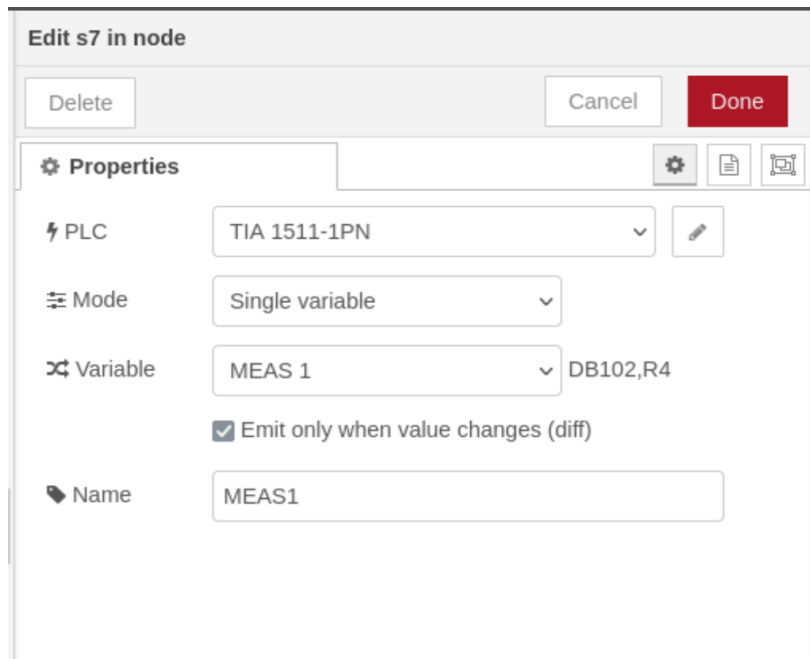


Kuva 9. Optimoidun datalohkon deaktivointi.

Node-RED:n osalta datan käsittely, niin sanotun Flow-kaavion lisääminen, onnistui myös melko helposti. S7-kommunikointi, ISO-on-TCP, vaatii kommunikointimoduulien asentamisen ja rajapinnan rakentamisen. Alla olevassa kuvassa "S7 in"-kommunikointisolmu lisätään Flow-ikkunaan (kuva 10).



Kuva 10. ISO-on-TCP kommunikointisolmu.



Kuva 11. S7-solmun (ISO-on-TCP) asetussivu.

Kuvassa 11 näkyy solmun asetteluikkunan perusnäky, josta päästään PLC:n yhteysasetuksiin (Kuva 12). Kommunikoinnin aktivointi vaatii siirtoprotokollan valinnan, IP-osoitteen ja kommunikointiportin valinnan sekä kommunikointipisteen eli fyysisen PLC:n paikan. PLC-räkissä uudemmat 1500-sarjan PLC:t ovat aina paikassa 0 ja Ethernet-portin paikka sijaitsee paikassa (slot) 1. Vanhemmissa 300-sarjan logiikoissa Ethernet-portin paikka on 2. Vaihtoehtona on myös erillinen kommunikointikortti, jolloin Rack ja Slot muuttuvat portin sijainnin mukaan. Variables-välilehdellä määritellään kaikki halutut mitattavat muuttujat (Kuva 13).

Edit s7 in node > **Edit s7 endpoint node**

Delete Cancel Update

**Properties**

Connection Variables

Transport Ethernet (ISO-on-TCP)

Address 10.20.0.3 Port 102

Mode Rack/Slot

Rack 0 Slot 1

Cycle time 1000 ms

Timeout 2000 ms

Name TIA 1511-1PN

Kuva 12. S7-yhteysasetukset.

Edit s7 in node > **Edit s7 endpoint node**

Delete Cancel Update

**Properties**

Connection Variables

Variable list

DB102,X0.0	Motor1OnSt	x
DB102,X0.1	Motor1FaultSt	x
DB102,X0.2	Motor2OnSt	x
DB102,X0.3	Motor2FaultSt	x
DB102,X0.4	Motor3OnSt	x
DB102,X0.5	Motor3FaultSt	x

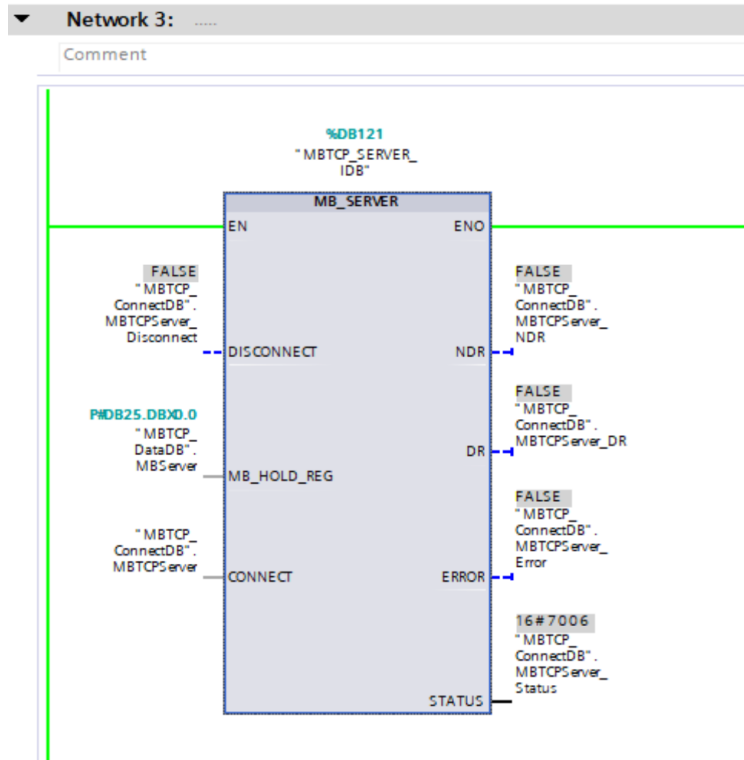
+ Add Remove all Import Export

Kuva 13. Variables-välilehti

Muuttujia voidaan lisätä suoraan taulukkoon, mutta suuremmilla muuttujamäärillä Excel tai muut taulukkolaskentatyökalut ovat hyviä apuvälineitä muuttujalistan muotoiluun.

#### 4.2.2 Modbus/TCP

Modbus/TCP on suhteellisen helposti liitettävä API Siemensin PLC:hen. Kuten kappaleessa 3.3.3 jo kerrottiin, vanhempiin PLC:hin tarvitaan maksullinen lisenssi, mutta uudemmissa Modbus/TCP-tuki on valmiina. Siemensin ohjelmointiympäristössä TIA portalissa on valmiit kommunikointiin tarvittavat komponentit Ladder- ja Function Block Diagram -ohjelmointiympäristöille kommunikoinnin rakentamista varten. Testilaitteessa käytettiin Modbus-Server-toimilohkoa sillä oletuksella, että tiedonkeruulaitteisto tarjoaa dataa ulkoiselle datapalvelimelle, joka toimii asiakkaana. Kuvassa 14 näkyy Modbus/TCP-palvelin toiminnassa. Vasemmanpuoleiset liityntäoksat antavat lähetettävän datan (MB\_HOLDING\_REG) ja liityntäparametrit (CONNECT) vastaanottavalle laitteelle.



Kuva 14. Modbus-palvelin Siemens S7-1500.



Kuvassa 15 näkyvät liityntäparametrit API:in. InterfaceID on PLC:n Ethernet-portin moduulinumero, joka on Profinetin erikoisuuksia, mutta se vaaditaan myös Modbus/TCP:n asetteluissa. ID-numerolla valitaan asiakaspari, jonka kanssa kommunikoidaan. Yhteystyyppi (Connection Type) valitaan TCP/IP:n ja UDP:n väliltä sekä valitaan tässä tapauksessa passiivinen yhteys laiteparien välille. RemoteAddress asetetaan asiakaslaitteiston IP-osoitteen mukaiseksi. MB\_HOLDING\_REG-siirtorekisteriin voidaan laittaa tarvittava määrä dataa. Datat siirto kohdelaitteelle tapahtuu 16-bittisillä kokonaisluvuilla. Jos lähetettävä lukuarvo on reaaliarvo, joka on tavallinen mittausten esitysmuoto, joudutaan Node-RED:ssä tämä lukuarvo kirjoittamaan takaisin reaalityyppiä erillisellä muunnosfunktiolla. Siemens-ympäristössä siirtorekisteriin (holding register) luettavat tiedot muuttuvat lohkon sisällä 16-bittiseen muotoon automaattisesti, mikä helpottaa reaalityyppien lähettämistä PLC:stä.

MBTCP_ConnectDB											
Name	Data type	Start value	Monitor value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervis...	Comment	
1	Static										
2	MBTCPServer	TCON_IP_v4									
3	InterfaceId	HW_ANY	64								
4	ID	CONN_OUC	1							HW-identifier of IE-interface submodule connection reference / identifier	
5	ConnectionType	Byte	16#0B							type of connection: 11=TCP/IP, 19=UDP (17=TCP/IP)	
6	ActiveEstablished	Bool	false							active/passive connection establishment	
7	RemoteAddress	IP_V4								remote IP address (IPv4)	
8	ADDR	Array[1..4] of Byte									
9	ADDR[1]	Byte	16#0A							IPv4 address	
10	ADDR[2]	Byte	16#14							IPv4 address	
11	ADDR[3]	Byte	16#00							IPv4 address	
12	ADDR[4]	Byte	16#08							IPv4 address	
13	RemotePort	UInt	0							remote UDPTCP port number	
14	LocalPort	UInt	502							local UDPTCP port number	
15	MBTCPServer_Disconn...	Bool	false								
16	MBTCPServer_NDR	Bool	false								
17	MBTCPServer_DR	Bool	false								
18	MBTCPServer_Error	Bool	false								
19	MBTCPServer_Status	Word	16#0								

Kuva 15. Modbus-server-liityntäparametrit.

Node-RED:ssä API muodostuu lisäkirjastona ladattavasta ModbusTCP-lohkosta. IP-asetukset laitetaan vastaamaan PLC:ssä olevan Server-laitteen eli PLC:n asetuksia. Unit-ID on 1 ja PLC:n IP-osoite ja kommunikointiportti 10.20.0.3:502 (Kuva 16). Kuvassa 17 määritellään datatyyppi, jota PLC:n Serverin rekisteristä halutaan lukea.

Edit Modbus-Read node > **Edit modbus-client node**

Delete Cancel Update

⚙️ Properties ⚙️ 📄

Settings Queues Optionals

Name TIA1500PLC

Type TCP ▾

---

Host 10.20.0.3

Port 502

TCP Type DEFAULT ▾

---

Unit-Id 1

Timeout (ms) 1000

🔌 Reconnect on timeout

Reconnect timeout (ms) 2000

Kuva 16. ModbusTCP-verkkoasetukset.

Edit Modbus-Read node

Delete Cancel Done

⚙️ Properties ⚙️ 📄 🖨️

Settings Optionals

Name TIA reaaliarvo

Topic real

Unit-Id 10

FC FC 3: Read Holding Registers ▾

Address 60

Quantity 20

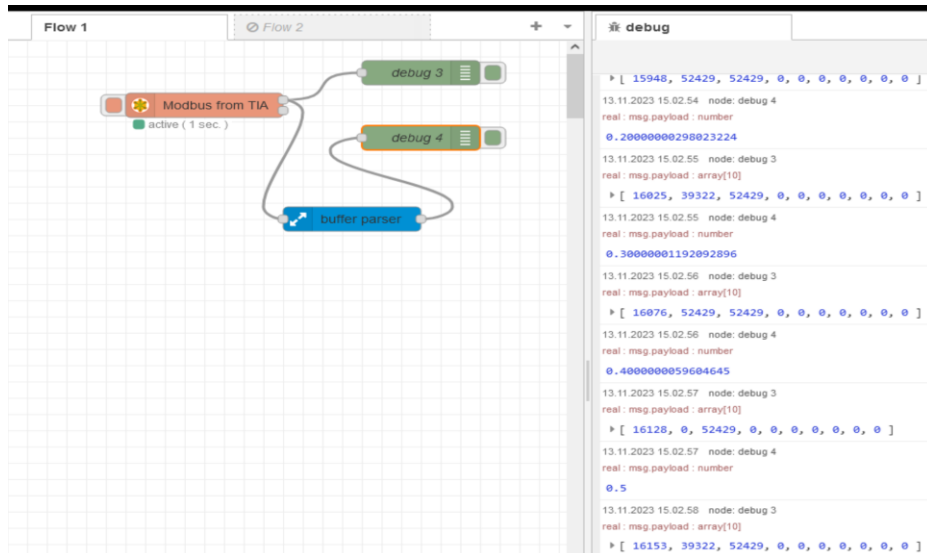
Poll Rate 1 ▾ second(s) ▾

🕒 Delay to activate input

Server TIA1500PLC ▾ ✎

📄  Enabled

Kuva 17. Asetukset Modbus-asiakkaan pyynnöstä Modbus-palvelimelle.



Kuva 18. ModbusTCP-tiedonsiirto PLC:stä käytössä.

### 4.2.3 MQTT

MQTT:n käyttö on IoT:n myötä lisääntynyt teollisuuden laitteissa sen keveyden ja turvallisen datansiirron vuoksi. Tämä julkaisija-tilaaja-viestiprotokolla kuormittaa vain vähän verkkoa ja siirtää datan laitteelta toiselle merkkijonoina, vaikka ei välitäkään lähetettävistä datatyypeistä (MQTT, 2019). Lisäksi julkaisijan ja tilaajan välille tarvitaan välittäjä eli Broker.

Node-RED-konfigurointi ei MQTT-testissä olennaisesti muuttunut muihin viesti-protokolliin verrattuna. Toisin kuin Modbus/TCP:ssa, MQTT-kommunikoinnissa Node-RED ja Siemens S7 toimivat molemmat client-roolissa ja voivat sekä julkaista että tilata dataa toiselta client-partnerilta. Välittäjäpalvelimena toimi testissä tietokoneelle asennettu Mosquitto-Broker. MQTT:n testiympäristössä siirrettiin vain yhtä aihetta PLC:n ja Node-RED:n välillä MQTT-solmun Topic:n ”a/b” kautta (Kuva 19, s. 28). MQTT valitseekin kommunikointipartnerin aiheen (Topic) mukaisesti.

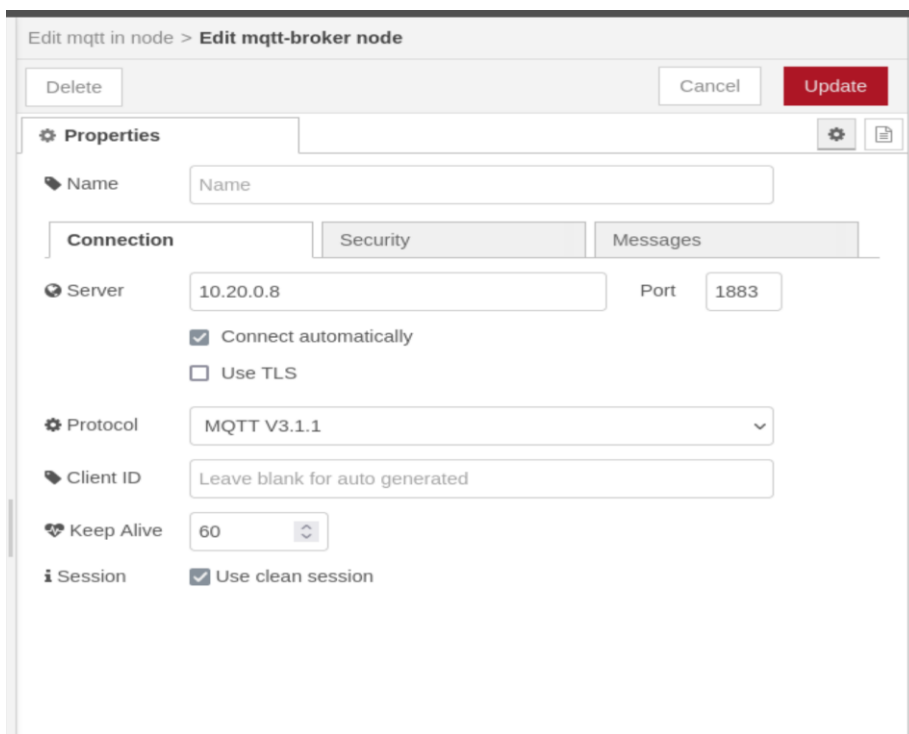
The image shows a configuration window titled "Edit mqtt in node". At the top, there are three buttons: "Delete", "Cancel", and "Done". Below this is a "Properties" section with several fields:

- Server:** A dropdown menu showing "TIA1511-1MQTT".
- Action:** A dropdown menu showing "Subscribe to single topic".
- Topic:** A text input field containing "a/b".
- QoS:** A dropdown menu showing "2".
- Output:** A dropdown menu showing "auto-detect (parsed JSON object, string or buffer)".
- Name:** A text input field containing "siemens".

At the bottom left of the window, there is a checkbox labeled "Enabled" which is currently unchecked.

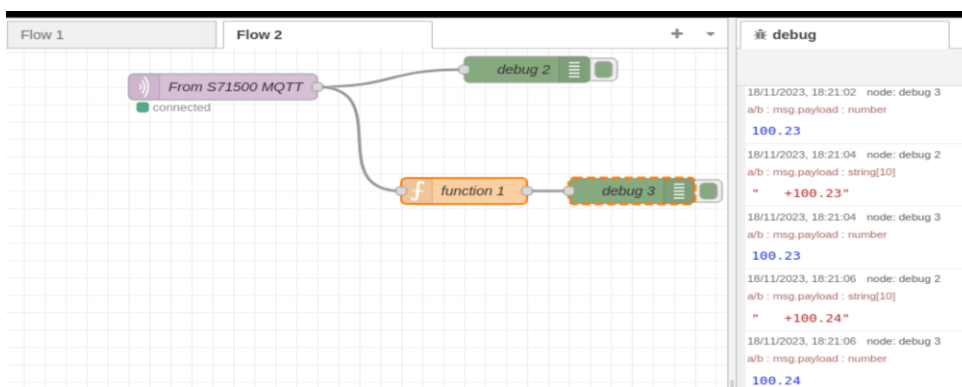
Kuva 19. MQTT-solmun perusasetukset.

Yhteysasetuksista valitaan MQTT-välittäjäsolmun IP osoite, toisin kuin muissa vertailuun valituissa rajapinnoissa (Kuva 20).



Kuva 20. MQTT-välittäjäsolmun kommunikointiasetukset.

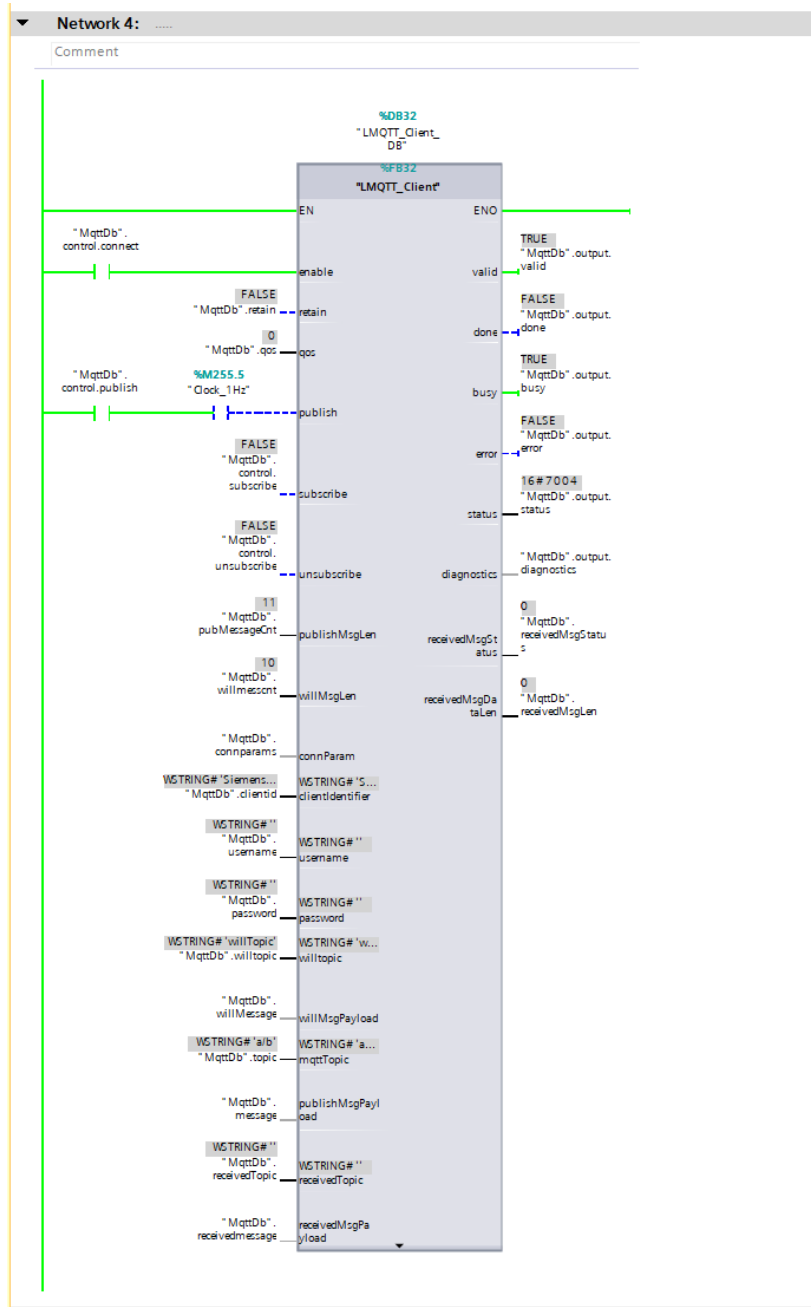
Kuvassa 21 näkyy Node-RED:n MQTT-rajapinta käytössä. Ylempi Debug-solmu palauttaa PLC:stä lähetetyn merkkijonon, ja alemmassa merkkijono on muunnettu muunnosfunktiolla reaalityyppiseksi.



Kuva 21. MQTT-tiedonsiirto PLC:stä käytössä.

PLC:n asetukset olivat aavistuksen monimutkaisempia aiempiin viestiprotokoliin verrattuna. Siemensin TIA portal -versiosta V16 löytyy valmis LMQTT-Client

-kirjasto, jonka avulla kommunikoinnin parametointi onnistui suhteellisen helposti. Kuvassa 22 näkyy MQTT-kommunikoinnin toimilohko käytössä. Tiedon siirtoa helpotti myös se, että tutkimuksen aiheena on pelkkä tiedonkeruu ja näin ollen datan vastaanottoa ei tarvinnut ottaa huomioon.



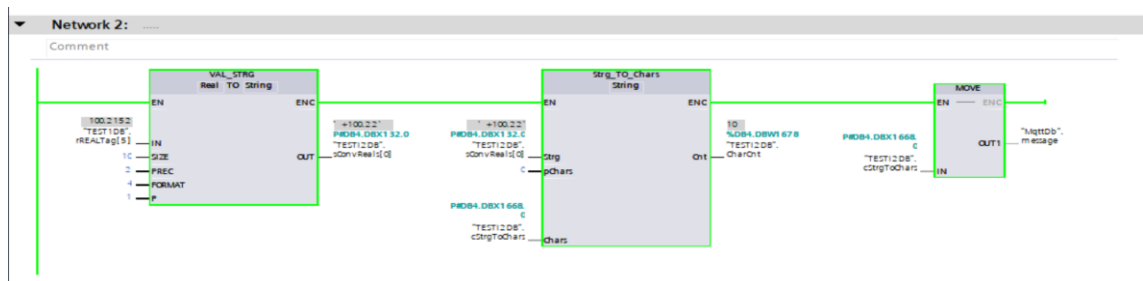
Kuva 22. TIA portalin MQTT-ohjauslohko.

Kaikki asetusrivot oli myös hyvin ryhmitelty. Kuvassa 23 näkyvät korostettuina kommunikoinnin asetukset, jotka Siemens-ympäristössä olivat merkitseviä.

Name	Data type	Start value	Monitor value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervis...	Comment
1	Static									
2	control	Struct								
3	connect	Bool	false	TRUE						Input of the FB
4	publish	Bool	false	TRUE						Set TRUE, to establish a connection - set to F...
5	subscribe	Bool	false	FALSE						Rising Edge: Sends a message
6	unsubscribe	Bool	false	FALSE						Rising Edge: Unsubscribes to a topic
7	output	Struct								Output Information of the FB
8	conparams	*LMQTT_typeConn...								Connection Parameters
9	hwid	HW_ANY	64	64						Hardware Id of the interface - default can be 0
10	connid	CONN_OUC	16#10	16#0010						Connection Id to be used
11	mqttBrokerAddress	Struct								Information on the mqtt broker
12	tls	Struct								When using tls for security
13	keepAlive	UInt	60	60						Keepalive mechanism for mqtt in seconds. 0 a
14	clientId	Wstring[20]	WSTRING#'Siem...	WSTRING#'Siemens...						ClientId with one or more characters
15	username	Wstring[20]	WSTRING#''	WSTRING#''						Username used for connect
16	password	Wstring[20]	WSTRING#''	WSTRING#''						Password used for connect
17	willtopic	Wstring[20]	WSTRING#'will...	WSTRING#'willTopic'						Topic where the last will message is sent
18	willMessage	Array[0..99] of Byte								Last will message Data - valid bytes in this arr
19	willMessageCnt	UInt	10	10						Valid number of elements in Will Message Da...
20	qos	USInt	0	0						Sets QoS 0, 1 or 2 for the Publish and Will Mes...
21	topic	Wstring[100]	WSTRING#'a/b'	WSTRING#'a/b'						Mqtt Topic to Publish / Subscribe / Unsubscribe
22	message	Array[0..9] of ...								Message Data to Publish - valid bytes in this a...
23	pubMessageCnt	UDInt	10	10						Valid number of elements in publish messag...
24	retain	Bool	false	FALSE						TRUE: Published Message will be set to retain...
25	receivedTopic	Wstring[200]	WSTRING#''	WSTRING#''						Mqtt Topic of received message
26	receivedMessage	Array[0..999] of Byte								Mqtt message data that was received - valid b...
27	receivedMsgLen	UDInt	0	0						Amount of Bytes: received in receivedMessag...

Kuva 23. Siemens MQTT data block.

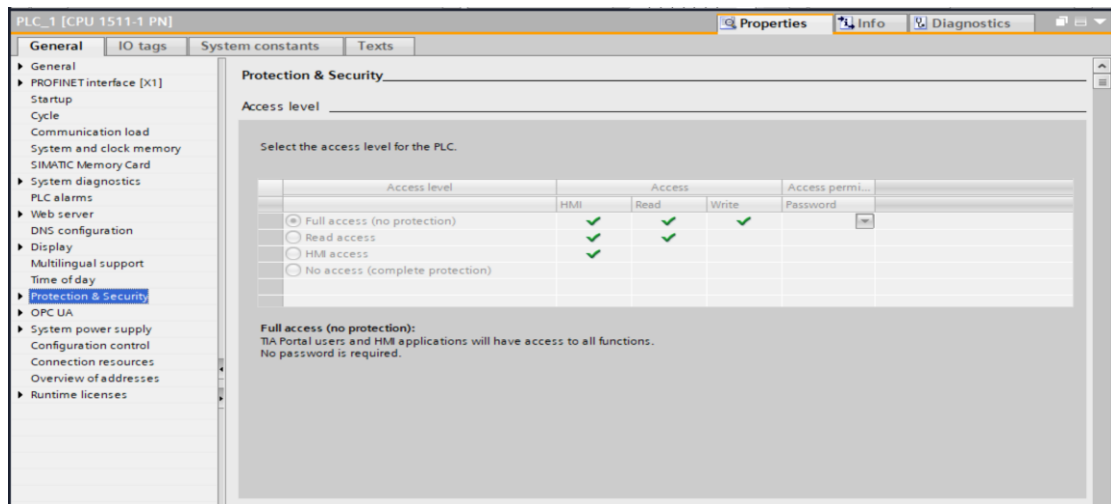
Kuvan 23 "message" on muuttujaryhmä, "Array of [0..9 ] Bytes", joka lähetetään välittäjäpalvelimen kautta tiedonkeruuseen. Testiympäristössä siirrettiin yhtä reaaliuuttujaa, joka oli siis muutettava tavuiksi (Bytes), jotta se pystyttiin lähettämään eteenpäin. Alla olevassa kuvassa 24 näkyy muuttujan muunnos reaalitylvusta tavumuotoiseksi Ascii-koodiksi. Siinä näkyy monivaiheinen muunnos reaalitylvusta tavumuotoiseksi ryhmäksi, joka ei näytä kovin tehokkaalta tavalta rakentaa useamman muuttujan siirtoa. Tässä tapauksessa tiedonsiirto olisi järkevämpää kirjoittaa struktuuritekstillä, jossa myös siirrettävän datalohkon vakioitu koko nopeuttaisi rajapinnan rakentamista.



Kuva 24. S7-muunnosfunktio reaalitylvusta Ascii-merkkijonoksi.

### 4.3 Tutkimusympäristön tietoturva

Tietoturvariskien tunnistaminen ja niiden torjuminen ovat avainasemassa, kun varmistetaan turvallinen ja luotettava tiedonvaihto PLC:n ja reunapalvelimen välillä. Yksinkertaisimmat tietoturvatoinenpiteet ovat luonnollisesti vahva salasanasuojaus reunapalvelimelle kirjautumiseen, vaihtoehtoisten kirjautumistapojen käytöstä poisto ja langattomien yhteyksien poistaminen käytöstä sekä fyysiset lukot reunapalvelimen ja PLC:n väliseen kommunikointiin. Tässä tapauksessa SSH- ja etätyöpöytäyhteys ovat välttämättömiä, jotta laitteen konfigurointi on ylipäätään mahdollista. Siemensin PLC:ssä on useita eri suojausmahdollisuuksia ja PLC:lle kirjautumisessa on neljä eri tasoa (Kuva 25).



Kuva 25. PLC:n suojausasetukset.

Ohjelmalohkoihin pääsy voidaan estää erillisellä Know-how protection -salasalla (Siemens Know-how, 2019), mutta datalohkoihin pääsyä tällä suojauksella ei voi estää. Siemensin PLC:llä on myös mahdollisuus luoda CA-sertifikaatti, jolla voidaan luoda turvallisuusvaltuutus kommunikointiin laitteelle, johon sertifikaatti on ladattu (Siemens sertifikaatti, 2019). Tämän tutkimuksen tapauksessa autentikoitu yhteys muodostettiin siis Raspberry Pi:n ja PLC:n välille.

Verrattaessa tutkimuksen kommunikointiprotokollia MQTT nousee selvästi tietoturvasuosimmaksi vaihtoehdoksi. ISO-on-TCP:n ja Modbus/TCP:n kommunikoinnin suojataso on käytännössä olematon. MQTT:n etuna on viestiliikenteen



salaminen SSL/TLS-suojausprotokollalla. Se ei myöskään tarvitse erillisiä autentikointisertifikaatteja, koska autentikointi tehdään protokollan omilla suojausmekanismeilla. Myös protokollan tekemä viestin eheystarkistus julkaisijan ja tilaajan välillä estää esimerkiksi viestiliikenteen väärentämisen.

Toisaalta taas CA-suojaussertifikaatilla on mahdollisuus suojata dataliikenne siirtotasojen välillä, joten suojattu dataliikenne saadaan tehtyä myös muihin tutkimuksen rajapintoihin. Teollisuusympäristössä olevien koneiden turvatasoa tarkasteltaessa pitää muistaa ulkoisten yhteyksien suojaaminen. Tutkimuksessa käytettiin etäyhteyden muodostukseen reititintä, joka salli liikenteen verkon yli vain fyysisellä USB-porttiin liitettävällä suojausavaimella. Tiedonkeruussa kommunikointiprotokollan valinta onkin tehtävä aina tapauskohtaisesta ja datan kriittisyys huomioiden. Kun ohjelmoitava logiikka ja reunapalvelin ovat samassa tilassa ja verkkoyhteydet ulkoisille palvelimille on suojattu, on tärkein tehtävä suojata fyysisesti laitteet niin, että niihin pääsevät käsiksi paikallisesti vain henkilöt, joilla on asianmukaiset valtuutukset ja oikeudet.

#### 4.4 Rajapintojen yhteenveto

Ohjelmointirajapintojen lisääminen Siemens S7 -logiikan ja Raspberry Pi edge-palvelimen välille oli suhteellisen helppoa. Haasteellisin vaihe oli reunatietokoneen perusasetuksiin ja laitteiston päivitykseen liittyvät ongelmat, jotka ovat myös vartenotettava asia, kun tehdään tiedonkeruuseen liittyvää projektia. Tutkimusaiheena olevien ohjelmointirajapintojen toiminnan kannalta Siemensin ISO-on-TCP-protokolla oli yksikertainen verrattuna muihin testattuihin protokolleihin, vaikka ero ei ollutkaan kovin suuri. Suurin etu, joka ISO-on-TCP:stä löytyy, on sen kyky lähettää eri muotoista dataa reunapalvelimelle. Modbus/TCP:n 16-bittisten sanojen ja MQTT:n merkkijonomuotoinen data joudutaan muuntamaan numeeriseksi ennen siirtoa tietokantaan tai csv-tiedostoon. Muunnosfunktiot voivat olla työllistäviä ja hitaita tehdä, mikä vaikuttaa kustannustehokkuuteen. Myös toimilohkojen konfiguroinnin datalohkot ovat aavistuksen työllistävempiä Modbus/TCP:n ja MQTT:n kanssa, mikä voi lisätä ohjelmoinnissa virheen mahdollisuutta.

Modbus/TCP on perinteinen ja yleinen kommunikointiprotokolla, joten se kannattaa pitää ISO-on-TCP:n rinnalla vaihtoehtona tiedonkeruuseen tai laitteistoihin, joissa kootaan useiden eri ohjelmoitavien logiikoiden (PLC) dataa samalle reunapalvelimelle. Myös MQTT on tiedonsiirron keveytensä ja tietoturvasa ansiosta kehitettävä alusta ohjelmoitavana rajapintana, etenkin mobiilisovelluksissa, joissa tiedonsiirron kaistanleveys on rajallinen.

Tutkimuksen ulkopuolelta on mainittava Allen-Bradleyn käyttämä Ethernet/IP-protokolla. Tässä datan siirto on hyvin samankaltainen kuin ISO-on-TCP:ssä, kun dataa siirretään PLC:n ja Node-RED:n välillä. Kommunikointi tarvitsee rajapinnan luomiseksi palvelimelle vain IP-osoitteen ja portin numeron sekä muuttujan osoitteen, joka on Allen-Bradleyn PLC:ssä symbolinen toisin kuin Siemensin PLC:ssä.

Aiemmin mainitun tilaston mukaan Siemens, Allen-Bradley ja Schneider ovat kolme suurinta PLC-valmistajaa, joiden tuotteet kattavat noin kaksi kolmasosaa markkinoilla olevista PLC-laitteista, joten tutkimuksen kohteena olevista kommunikointiprotokollista on jo saatu varsin kattava otanta teollisen internetin rajapinnaksi. Tutkimustyön kautta varmistui, että yleensä ulkoisten laitteiden liittäminen helpoin ja monipuolisin tapa on PLC-laitteiston mukainen liitäntästandardi, mikäli se on mahdollista valita rajapinnan vastaparina toimivasta laitteistosta.

#### 4.5 Tiedon tallennus paikallisesti vai verkkoon?

PLC:n mittausdatasta tuottama historiadata on yleensä aikasarjadataa eli ajallisesti järjestettyä dataa, joka sisältää aikaleimalla varustettuja mittaussuureita (Stenfors R., 2021), kuten lämpötila, paine, virtaus ja muita vastaavia suureita, joita prosessi- tai tuotantokoneesta voidaan mitata. Esimerkkinä voidaan olettaa pienen tuotantokoneen tuottavan 30 erilaista mittaussuuretta ja lisäksi niiden rinnalle jokaisella mittauksella on oma asetusarvonsa. Näin ollen datan keruu tapahtuu jokaisen PLC:n ohjelmakierroksen välein. Hypoteettisesti ajateltuna 100 millisekunnin välein syntyy tallennettavaksi dataa 600 mittauksen verran joka sekunti, mikä kuulostaa varsin mittavalta datan määrältä. Yhden

liukulukumuuttujan koko on 4 tavua, joten teoreettisesti tarkastellen 600 mittauksen tallennus tuottaisi 2,4 kilotavua (kB) dataa sekunnissa. Vuorokauden aikana datamäärä nousisi 3456 kilotavuun eli noin 3,5 megatavuun (MB), ja kuukauden aikana noin 104 megatavuun.

Tämä datamäärä ei enää kuulostakaan niin suurelta tallennustilaa ajatellessa. Miten tämä tieto tallennetaan ja missä se säilytetään? Erityisesti kun otetaan huomioon yhden piirilevyn tietokoneen (Single Board Computer, SBC) rajoitetumpi suorituskyky verrattuna perinteiseen tietokoneeseen (PC), tietokannan valinta saa entistäkin suuremman merkityksen. SBC:n suorituskyky riippuu sen prosessorin tehosta, muistista ja tallennustilasta. SBC:lle paikallisesti tallennettu data voi pahimmassa tapauksessa rikkoa koko tietokoneen, kun tallennustila täyttyy. On siis tärkeää optimoida talletettavan datan määrä tai tallennustiheys sopivaksi, jos dataa ei tallenneta ulkoiselle medialle tai sitä ei poisteta riittävän usein.

Tietokannan valinnassa on otettava huomioon myös se, millaisia tietokyselyjä tarvitaan ja miten tietoa tullaan hyödyntämään. Tutkimuksen tarkoituksena oli löytää ketterä tapa tiedonkeruuseen pk-yrityksille tai suunnittelun ja tuotekehityksen tukivälineeksi eikä varsinaisten tietokantakyselyiden kohdetta ole määriteltä. Tämä tukee yksinkertaisen datavaraston luomista paikalliseen verkkoon, ja näin ollen csv-tiedosto vaikuttaisi parhaalta vaihtoehdolta tähän tilanteeseen, vaikka se ei tue monimutkaisempia tietorakenteita tai suhteita, kuten oikeat tietokannat. Csv-tiedostomuotoa tukee myös sen yksinkertaisuus ja helppous ymmärtää henkilöille, jotka eivät ole perehtyneet tietokantojen kanssa työskenteleeseen. Csv:n käsittely onnistuu helposti yleisimmillä taulukkolaskentatyökaluilla, jotka ovat helposti jokaisen saatavilla.

## **5 Tiedonkeruuympäristön ohjelmointi**

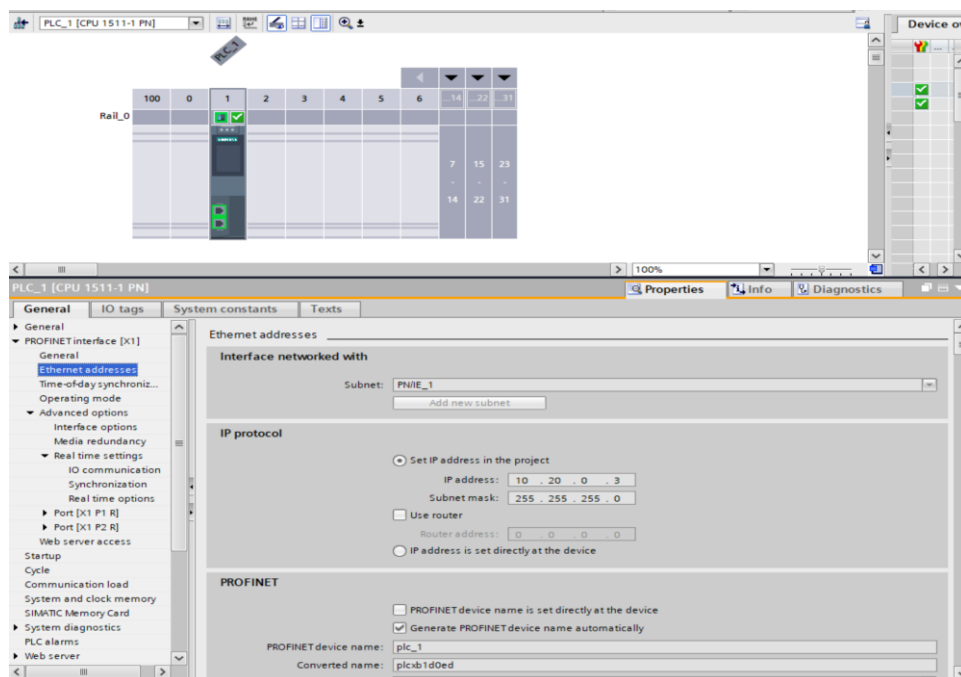
Tutkimustyön empiirisessä osuudessa rakennettiin tutkimusympäristö, johon kuului yhden piirilevyn tietokone Raspberry Pi4, Siemens S7-1511-PN -ohjelmoitava logiikka ja Tosibox Lock 150 -teollisuusreititin. Tällä

simulointiympäristöllä pyrittiin saamaan todellisuutta kuvaava ympäristö, johon voidaan ottaa tietoturvallinen yhteys julkisen verkon yli.

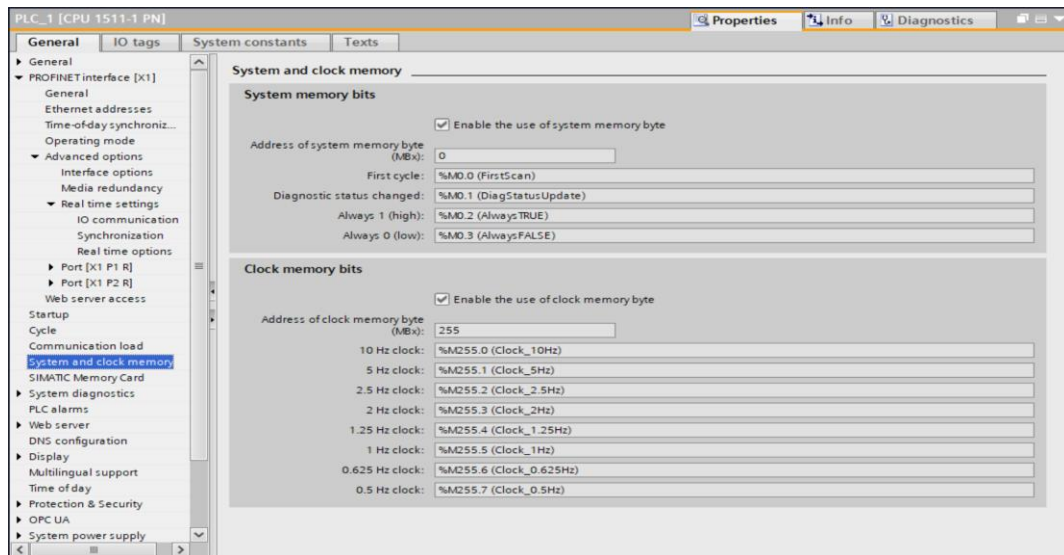
## 5.1 PLC:n konfigurointi

Testiympäristön PLC:lle tehtiin erityinen projekti, johon luotiin kaikille valituille kommunikointirajapinnoille vaadittavat konfigurointilohkot, datalohkot ja tietoa tuottavat ohjelmat eli toimilohkot. Toimilohkoissa tuotettiin dataa simuloimalla binäärilukujen tilamuutoksia ja kokonais- ja liukulukujen muutokset askeltaen sekä kiinteänä arvona.

PLC-laitteistokonfiguraatiossa asetettiin kiinteä IP-osoite ja aliverkon peite (Kuva 26). Lisäksi asetettiin järjestelmän sisäisiä muistibittejä ja eri taajuuksilla toimivia järjestelmän kellobittejä ohjelman aputoimintoja varten (Kuva 27).



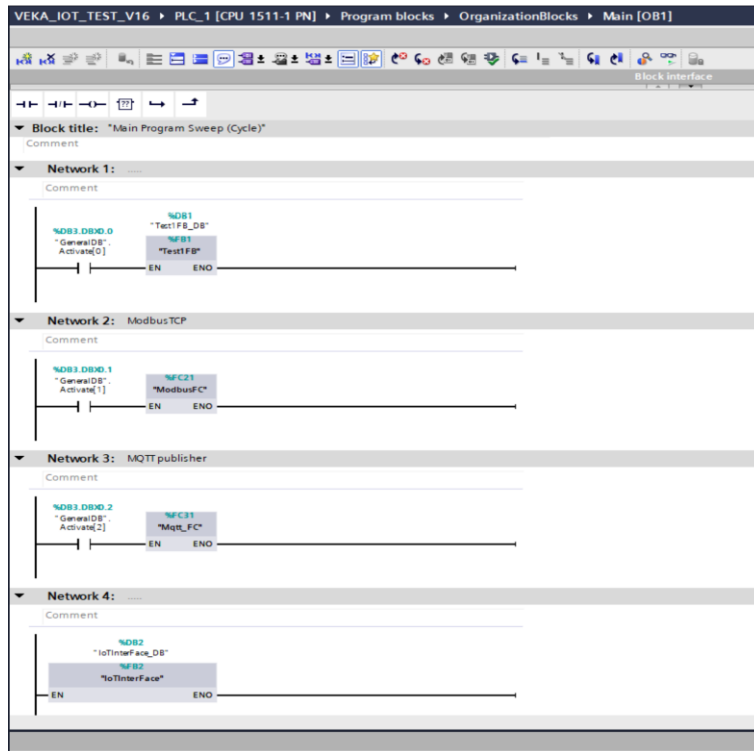
Kuva 26. PLC:n verkkoasetukset.



Kuva 27. Testilaitteiston S7-ohjelman järjestelmäbittien asettelut.

### 5.1.1 PLC-ohjelma

PLC-ohjelma oli tarkoitus pitää yksinkertaisena, jotta nähtiin selkeästi datansiirron toimivuus. Testausympäristössä ei olisi tarvittu lainkaan toimilohkojen kutsuja, koska Node-RED lukee datan suoraan halutusta datalohkosta. Toimilohkokutsu lisättiin kuitenkin ohjelmakiertoon simuloimaan oikean ohjelman suoritustapaa. Ohjelma suoritettiin OB1-ohjelmalohkossa, jossa kutsuttiin tarpeelliset ohjelmarakenteet (Kuva 28). OB1-lohkon suoritusjärjestyksessä viimeisenä on datasiirron funktiolohko, "IoTInterface" (Kuva 29). Käytännön merkitystä tällä rakenteella ei ole, koska Node-RED pystyy lukemaan muuttujat ohjelmalohkosta riippumatta, kunhan datalohko on "Global"-tyyppiä eli datalohko näkyy kaikille. Datasiirtolohkoon tuotiin muista toimilohkoista simuloitua tila- ja mittaustietoa. Muuttujat, jotka luettiin Node-RED:iin, olivat "IoTDB"-datalohkossa ja Node-RED:ssä luettiin sama määrä muuttujia, jolloin konfigurointia ei tarvinnut välttämättä muuttaa, vaikka PLC:stä olisi vaihdettu kerättävä muuttuja toiseen.



Kuva 28. OB1, Main block.

```

VEKA_IOT_TEST_V16 | PLC_1 [CPU 1511-1 PN] | Program blocks | TEST1_1_S7Comm | IoTInterFace [FB2]
Block interface
IF... CASE... FOR... WHILE... REGION
OP... TO DO... DO... (*..*)
1 //Process Status////////////////////////////////////
2 //Process On, On=1
3 "IoTDB".IoT_status[0] := "TESTIDB".bTag[0]; //motor 1 on
4 "IoTDB".IoT_status[1] := "TESTIDB".bTag[1]; //motor 1 fault
5 "IoTDB".IoT_status[2] := "TESTIDB".bTag[2];
6 "IoTDB".IoT_status[3] := "TESTIDB".bTag[3];
7 "IoTDB".IoT_status[4] := "TESTIDB".bTag[4];
8 "IoTDB".IoT_status[5] := "TESTIDB".bTag[5];
9 "IoTDB".IoT_status[6] := "TESTIDB".bTag[6];
10 "IoTDB".IoT_status[7] := "TESTIDB".bTag[7];
11 "IoTDB".IoT_status[8] := "TESTIDB".bTag[8];
12 "IoTDB".IoT_status[9] := "TESTIDB".bTag[9];
13 "IoTDB".IoT_status[10] := "TESTIDB".bTag[10];
14 "IoTDB".IoT_status[11] := "TESTIDB".bTag[11];
15 "IoTDB".IoT_status[12] := "TESTIDB".bTag[12];
16 "IoTDB".IoT_status[13] := "TESTIDB".bTag[13];
17 "IoTDB".IoT_status[14] := "TESTIDB".bTag[14];
18 "IoTDB".IoT_status[15] := "TESTIDB".bTag[15];
19 //////////////////////////////////Process Commands////////////////////////////////////
20
21 "IoTDB".IoT_Command[0] := "TESTIDB".bTag[16];
22 "IoTDB".IoT_Command[1] := "TESTIDB".bTag[17];
23 "IoTDB".IoT_Command[2] := "TESTIDB".bTag[18];
24 "IoTDB".IoT_Command[3] := "TESTIDB".bTag[19];
25 "IoTDB".IoT_Command[4] := "TESTIDB".bTag[20];
26 "IoTDB".IoT_Command[5] := "TESTIDB".bTag[21];
27 "IoTDB".IoT_Command[6] := "TESTIDB".bTag[22];
28 "IoTDB".IoT_Command[7] := "TESTIDB".bTag[23];
29 "IoTDB".IoT_Command[8] := "TESTIDB".bTag[24];
30 "IoTDB".IoT_Command[9] := "TESTIDB".bTag[25];
31 "IoTDB".IoT_Command[10] := "TESTIDB".bTag[26];
32 "IoTDB".IoT_Command[11] := "TESTIDB".bTag[27];
33 "IoTDB".IoT_Command[12] := "TESTIDB".bTag[28];
34 "IoTDB".IoT_Command[13] := "TESTIDB".bTag[29];
35 "IoTDB".IoT_Command[14] := "TESTIDB".bTag[30];
36 "IoTDB".IoT_Command[15] := "TESTIDB".bTag[31];
37 "IoTDB".IoT_Measurement[0] := "TESTIDB".rREALTag[0];
38 "IoTDB".IoT_Measurement[1] := "TESTIDB".rREALTag[1];
39 "IoTDB".IoT_Measurement[2] := "TESTIDB".rREALTag[2];
40 "IoTDB".IoT_Measurement[3] := "TESTIDB".rREALTag[3];
41 "IoTDB".IoT_Measurement[4] := "TESTIDB".rREALTag[4];
42
43 "IoTDB".IoT_Setpoint[31] := "TESTIDB".rREALTag[4];
44

```

Kuva 29. Datansiirtofunktio.

## 5.2 SBC:n konfigurointi

SBC:n käyttöjärjestelmäksi valittiin Raspberry Pi:n 64-bittinen Bullseye (desktop) -käyttöjärjestelmä. Valinta perustui usean verkkosivuston arvioon, mutta loppujen lopuksi Raspberry Pi:n Tips & Tricks -sivuston arvio vakaammasta ympäristöstä ja sopivuus vähemmän kokemusta omaaville henkilöille vaikutti päätökseen Raspberry OS:n eduksi muihin käyttöjärjestelmiin verrattuna. Käyttöjärjestelmän työpöytäversion valinnalla haettiin helppokäyttöisyyttä, mutta se oli myös välttämätön käytettäessä Node-RED:n graafista ohjelmointiympäristöä.

### 5.2.1 Esiasettelut

Käyttöjärjestelmä asennettiin Raspberry Pi Imager -työkalulla 32 Gt:n MicroSD-muistikortille. Tutkimuksen Raspberry Pi:lle ei ollut erillistä näyttöä, joten ensimmäiset ohjelmistopäivitykset ja asennukset tehtiin SSH-yhteydellä PuTTY:lla.

SSH-yhteydellä aktivoitiin VNC-palvelinyhteys, jotta graafiseen käyttöliittymään pystyttiin luomaan etäyhteys. VNC-server ladattiin myös SSH-yhteydellä, Windows-tietokoneeseen yhdistäminen vaati myös RPi:lle Xrdp-lisäosan asennuksen, jotta etäyhteys toimisi oikein. Etäyhteyden muodostaminen vaati myös käyttäjän ja etäkäyttöoikeuksien lisäämisen. Tutkimusympäristöön liitetyille etäyhteyskäyttäjille vaadittiin täydet järjestelmävalvojan oikeudet, kuten järjestelmän pääkäyttäjälläkin on, jotta tietokoneen konfigurointi onnistui myös etätyöpöydän kautta.

### 5.2.2 Tutkimustyökalujen asennus

Tietokoneen perusasetusten ja päivitysten jälkeen asennettiin tutkimuksen kannalta vaadittavat ohjelmistot eli Node-RED ja ohjelmointiympäristöksi Firefoxin selain. Firefox valittiin useilla verkkosivuilla olevien suositusten vuoksi: muun muassa tietokantoja ja pilvipalveluita tarjoava Influxdata piti Firefoxia parhaana ohjelmointiympäristönä Node-RED:lle (Influxdata, 2023).

Node-RED:n asennus tehtiin Node-RED:n verkkosivuston ohjeilla. Komentokotteen (Kuva 30) annetulla käskyllä ohjelmisto valitsi sopivimman version Node-RED:stä ja npm:stä, Node.js:n paketinhallintatyökalusta. Npm on ohjelmistorekisteri, jonka kautta kehittäjät lataavat valmiita ohjelmia tai ohjelmistokoodin pätkiä omiin ohjelmiinsa (npm, 2023).

```
bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)
```

Kuva 30. Node-RED-ohjelman asennuskomento.

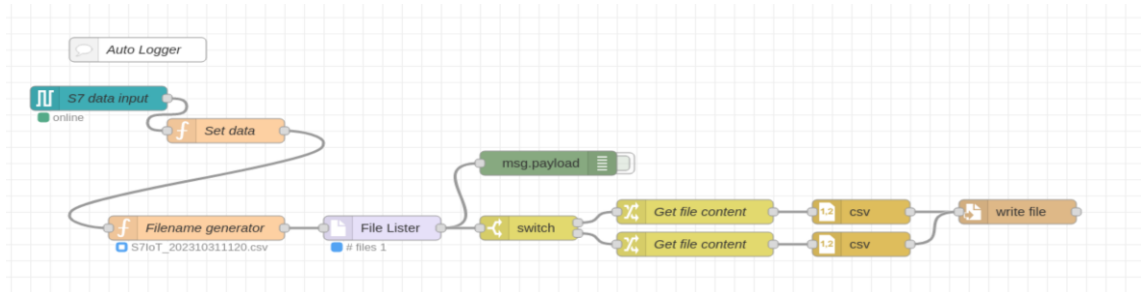
Kappaleessa 4.2 tiedonkeruun API:en vertailussa MQTT-protokollaa varten asennettiin myös Mosquitto MQTT-broker, viestinvälittäjä, jotta nähtiin kommunikoinnissa olevat virheet ja varmistettiin kommunikoinnin toimivuus. Tätä ei kuitenkaan lopullisessa tiedonkeruuympäristössä tarvittu.

### 5.3 Node-RED

Testirajapinnaksi valitun "S7-comm":n ISO-on-TCP-protokollan käyttö on yleistä Siemens-ympäristössä IoT-projekteissa. Tämä näkyi verkkohakuja tehdessä, kun etsittiin sopivaa pohjaprojektia työlle. ISO-on-TCP-protokollan käyttöä tukee avoimen lähdekoodin Siemensille suunnattu Snap7-kirjasto, joka mahdollistaa rajapinnan luonnin usealla ohjelmointikielellä Siemensin PLC-laitteille. Myös Node-RED:n Node.js kuuluu tähän joukkoon. Node-RED-alustalle ei kuitenkaan löytynyt valmista tiedonkeruuratkaisua. Toki internet on pullollaan tekstejä, kuvia, videoita ja ohjeita, joissa Siemens-muuttuja tuodaan Node-RED:iin ja nähdään ohjelman toiminnallisuus, mutta ei vaikuta löytyvän kokonaista PLC:stä tuotua datavirtamallia, jossa datavirta siirretään pakettina tallennusmedialle. Node-RED:n sivustolta kuitenkin löytyi Csongor Vargan tekemä projekti, jossa sääanturidataa siirrettiin Raspberryn Pi:n muistiin, ja josta Varga oli tehnyt myös monipuolisen ohjevideon (NodeRed nygma2004, 2023). Tästä pohjaohjelmasta oli sitten lähdettävä hakemaan ratkaisua Siemens-ympäristöön. Kuvassa 31



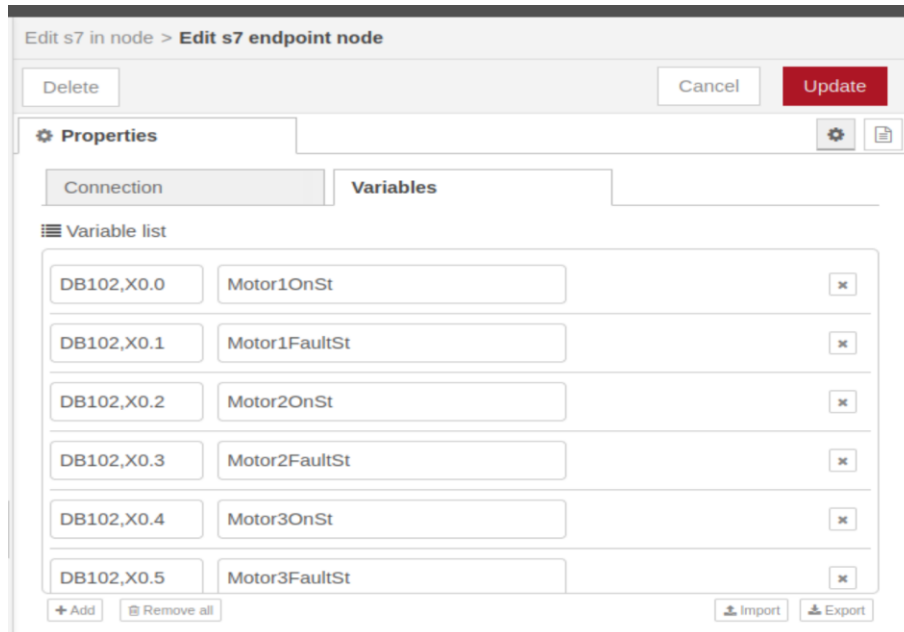
näkyä valmis Node-RED-koodi, jolla tiedot siirrettiin PLC:stä ja tallennettiin csv-tiedostoon.



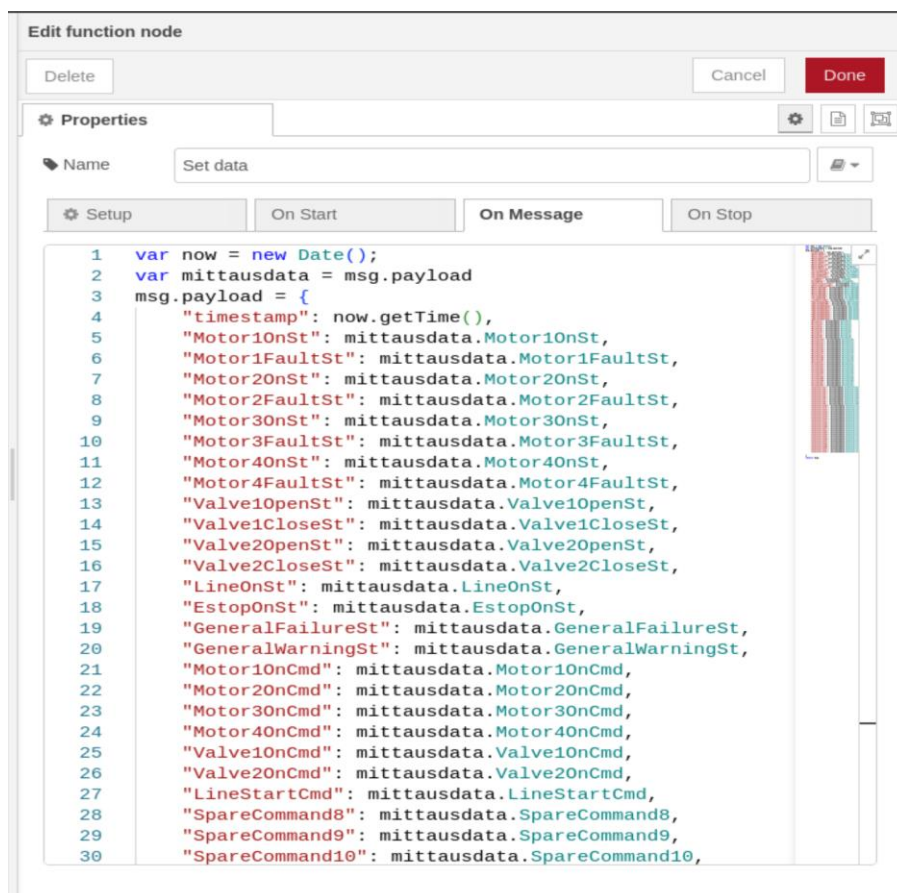
Kuva 31. S7-dataa csv-tiedostoon.

”S7 data input” -solmussa haluttu dataliikenne vastaanotetaan Node-RED:iin käsiteltäväksi (kuva 32). Tämä solmu oli mielenkiintoinen, koska jo kappaleessa 4.2.1 esitelty kommunikointirajapinnan muuttujien lisääminen on työlästä yksitellen, mutta ikkunan Import/Export-toiminnalla muuttujat voidaan viedä Exceliin käsiteltäviksi, mikä helpottaa muuttujien lisäämistä, uudelleen nimeämistä ja poistamista.

”Set data” -solmussa lisätään viestipakettiin aikaleima ja vastaanotetut tiedot muokataan oikeaan muotoon uudeksi msg-payload-paketiksi csv-tiedoston kirjoitusta varten (kuva 33). Myös tässä solmussa Excelin käyttö helpottaa muuttujien nimeämisessä (Kuva 34).



Kuva 32. Muuttujat s7-solmussa.



Kuva 33. "Set data" -funktio.

Column1	Column2	Column3	Sarake1	Column4
DB102.X0.0	Motor1OnSt	Motor1OnSt	Motor1OnSt: mittausdata.Motor1	mittausdata.Motor1OnSt
DB102.X0.1	Motor1FaultSt	Motor1FaultSt	Motor1FaultSt: mittausdata.Motc	mittausdata.Motor1FaultSt
DB102.X0.2	Motor2OnSt	Motor2OnSt	Motor2OnSt: mittausdata.Motor2	mittausdata.Motor2OnSt
DB102.X0.3	Motor2FaultSt	Motor2FaultSt	Motor2FaultSt: mittausdata.Motc	mittausdata.Motor2FaultSt
DB102.X0.4	Motor3OnSt	Motor3OnSt	Motor3OnSt: mittausdata.Motor3	mittausdata.Motor3OnSt
DB102.X0.5	Motor3FaultSt	Motor3FaultSt	Motor3FaultSt: mittausdata.Motc	mittausdata.Motor3FaultSt
DB102.X0.6	Motor4OnSt	Motor4OnSt	Motor4OnSt: mittausdata.Motor4	mittausdata.Motor4OnSt
DB102.X0.7	Motor4FaultSt	Motor4FaultSt	Motor4FaultSt: mittausdata.Motc	mittausdata.Motor4FaultSt
DB102.X1.0	Valve1OpenSt	Valve1OpenSt	Valve1OpenSt: mittausdata.Valve	mittausdata.Valve1OpenSt
DB102.X1.1	Valve1CloseSt	Valve1CloseSt	Valve1CloseSt: mittausdata.Valve	mittausdata.Valve1CloseSt
DB102.X1.2	Valve2OpenSt	Valve2OpenSt	Valve2OpenSt: mittausdata.Valve	mittausdata.Valve2OpenSt
DB102.X1.3	Valve2CloseSt	Valve2CloseSt	Valve2CloseSt: mittausdata.Valve	mittausdata.Valve2CloseSt
DB102.X1.4	LineOnSt	LineOnSt	LineOnSt: mittausdata.LineOnSt	: mittausdata.LineOnSt
DB102.X1.5	EstopOnSt	EstopOnSt	EstopOnSt: mittausdata.EstopOnSt	: mittausdata.EstopOnSt
DB102.X1.6	GeneralFailureSt	GeneralFailureSt	GeneralFailureSt: mittausdata.Ger	mittausdata.GeneralFailureSt
DB102.X1.7	GeneralWarningSt	GeneralWarningSt	GeneralWarningSt: mittausdata.G	mittausdata.GeneralWarningSt
DB102.X2.0	Motor1OnCmd	Motor1OnCmd	Motor1OnCmd: mittausdata.Mot	mittausdata.Motor1OnCmd
DB102.X2.1	Motor2OnCmd	Motor2OnCmd	Motor2OnCmd: mittausdata.Mot	mittausdata.Motor2OnCmd
DB102.X2.2	Motor3OnCmd	Motor3OnCmd	Motor3OnCmd: mittausdata.Mot	mittausdata.Motor3OnCmd
DB102.X2.3	Motor4OnCmd	Motor4OnCmd	Motor4OnCmd: mittausdata.Mot	mittausdata.Motor4OnCmd
DB102.X2.4	Valve1OnCmd	Valve1OnCmd	Valve1OnCmd: mittausdata.Valve	mittausdata.Valve1OnCmd
DB102.X2.5	Valve2OnCmd	Valve2OnCmd	Valve2OnCmd: mittausdata.Valve	mittausdata.Valve2OnCmd
DB102.X2.6	LineStartCmd	LineStartCmd	LineStartCmd: mittausdata.LineSt	mittausdata.LineStartCmd
DB102.X2.7	SpareCommand8	SpareCommand8	SpareCommand8: mittausdata.Sp	mittausdata.SpareCommand8
DB102.X3.0	SpareCommand9	SpareCommand9	SpareCommand9: mittausdata.Sp	mittausdata.SpareCommand9
DB102.X3.1	SpareCommand10	SpareCommand10	SpareCommand10: mittausdata.S	mittausdata.SpareCommand10
DB102.X3.2	SpareCommand11	SpareCommand11	SpareCommand11: mittausdata.S	mittausdata.SpareCommand11
DB102.X3.3	SpareCommand12	SpareCommand12	SpareCommand12: mittausdata.S	mittausdata.SpareCommand12

Kuva 34. Excel-muokatut muuttujat.

”Filename generator” -funktioilla annettiin csv-tiedostolle nimi (Kuva 35).

```

Edit function node
Delete Cancel Done
Properties
Name: Filename generator
Setup On Start On Message On Stop
1 // Get the current time and convert it to text
2 var now = new Date();
3 var yyyy = now.getFullYear();
4 var mm = now.getMonth() < 9 ? "0" + (now.getMonth() + 1)
5 var dd = now.getDate() < 10 ? "0" + now.getDate() : now.
6 var hh = now.getHours() < 10 ? "0" + now.getHours() : now.
7 var mmm = now.getMinutes() < 10 ? "0" + now.getMinutes()
8 var ss = now.getSeconds() < 10 ? "0" + now.getSeconds()
9
10 // Generate out file name pattern
11 //msg.fname = "S7IoT_" + yyyy + mm + dd + ".csv";
12 msg.fname = "S7IoT_" + yyyy + mm + dd + hh + mmm + ".csv"
13 // Full filename with path for the file node later
14 msg.filename = "/home/pi/datalog/" + msg.fname;
15
16 // We save the current payload into a different place on
17 msg.filecontent = msg.payload;
18
19 // We are passing the file name search pattern to fs node
20 msg.payload = {"pattern":msg.fname};
21
22 node.status({fill:"blue", shape:"ring", text:msg.fname});
23 return msg;

```

Kuva 35. Filename generator.

"FileLister" -solmu taas vastaanottaa paketin ja luo kansion tiedoista, mikäli sitä ei ole vielä luotu. Jos datamalli on jo valmiina, "FileLister" lisää uuden datan saman kansiorakenteen sisään.

"FileLister" lähettää paketin seuraavaksi "switch"-solmulle vertailtavaksi ja valitsee polun sen mukaan onko tallennuskansio jo luotu vai ei. Seuraavassa vaiheessa "Get File content" -solmu asettaa datan csv-tiedostolle valmiiksi. Kyseinen solmu on samanlainen molemmissa "switch"-solmun haaroissa.

"Csv"-solmu tekee datapaketista lopullisen tiedoston. "Switch"-solmun ylemmästä haarasta lähtee alla olevan kuvan 36 mukainen otsikkorivi csv:n ensimmäiseksi riviksi, kun taas seuraavassa kuvassa 37 näkyy kuvakaappaus alemmasta haarasta.

**Edit csv node**

Delete Cancel Done

**Properties**

Columns timestamp, Motor1FaultSt, Motor2OnSt, Motor2Fau

Separator comma

Name Name

---

CSV to Object options

Input Skip first 0 lines

first row contains column names

parse numerical values

include empty strings

include null values

Output a message per row

---

Object to CSV options

Output always send column headers

Newline Linux (\n)

Kuva 36. "Csv"-solmu, joka lisää otsikon mukaan tiedostoon.

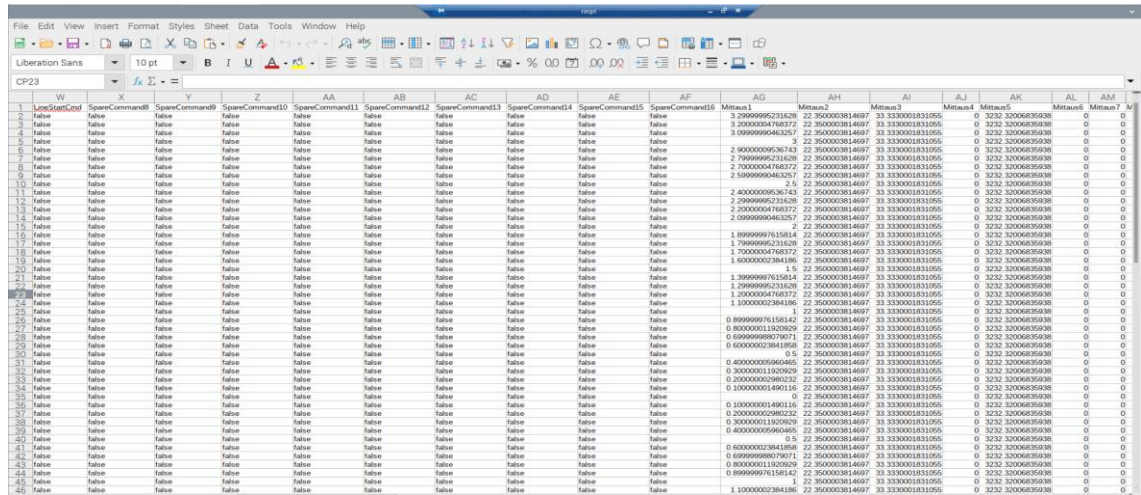
Object to CSV options

Output never send column headers

Newline Linux (\n)

Kuva 37. Alempi haara, ei luo otsikkoriviä.

Viimeisessä vaiheessa "Write File" -solmu luo tiedoston polkuun, joka määritettiin "Filename generator" -solmussa. Lopullisen tiedon voi lukea vaikka suoraan Raspberry Pi:n etätyöpöydältä Libre Office -ohjelmalla (Kuva 38).



Kuva 38. Libre Office -näkyvä.

### 5.4 Yhteenveto tiedonkeruun ohjelmoinnista

Tiedonkeruun simulointiympäristön ohjelmointi onnistui melko helposti. Node-RED pystyy lukemaan Siemensin PLC:stä helposti tiedot datatypistä ja -alueesta riippumatta. NodeRED:n selkeä käyttöliittymä ja monipuoliset ohjeet helpottivat tiedonkeruuohjelman tekemistä. Raspberry Pi -ympäristössä asennuksen haasteet olivat suurin ongelma laitteiston käyttöönotossa, vaikka tukiyhteisöjä ja ohjeita löytyy internetistä runsaasti. Suurimpana ongelmana oli Raspberry Pi:hin asennettujen ohjelmien käyttöoikeuksien jakaminen: ohjelmien asentaminen etäkäyttäjänä vaati lähes kaikkien oikeuksien jakamista kaikille käyttäjille. Samoin NodeRED:n kirjoittaminen Raspberry Pi:n kansiorakenteisiin ei onnistunut ilman kansiorakenteiden käyttöoikeuksien vapautusta kaikille käyttäjille. Oikeuksien jakaminen laskee laitteiston tietoturvan tasoa, koska se voi mahdollistaa mahdollisten haitallisten ohjelmien tai käyttäjien pääsyn järjestelmään. Raspberry Pi:n ohjelmoijille onkin hyvä tehdä tarkka ohjeistus ohjelmien ja etäkäyttäjien oikeuksien määrittelyä, jottei tietoturva vaarannu.

Toinen tietoturvaan liittyvä seikka oli laitekannan päivittäminen asiakaskäytössä. Tiedonkeruulaitteisto vaatii luonnollisesti päivittämistä, jotta tietoturvan taso verkkoympäristössä olisi riittävä. Testilaitteistoa käytettäessä päivitykset asennettiin testiaikana, mutta asiakasrajapinnassa tämä ei välttämättä ole mahdollista. Ongelman ratkaisemiseksi on asiakasprojekteihin hyvä tehdä ohjelma, jonka avulla laitekanta voidaan päivittää automaattisesti tietyinä aikana vuorokaudesta tai käyttäjän määrittelemänä aikana.

Pääosin Raspberry Pi:n asennukset menivät sujuvasti eikä suurempia haasteita ollut. Testiympäristöä sovellettaessa projekteille on hyvä kehittää tarkat ohjeet käyttöoikeuksien hallintaan, jotta tietoturva ei vaarannu. Laitekannan päivityksen osalta on suositeltavaa kehittää RPi-ympäristöä niin, että laitteisto mahdollistaa automaattiset tai puoliautomaattiset päivitykset.

## 6 Pohdinta

Tutkimuksen tarkoituksena oli tarkastella teknologisia ratkaisuja helppoon ja kustannustehokkaaseen tiedonkeruuseen teollisuuden laitteisiin vastauksena työvoimapulaan automaatio- ja IT-alalla sekä löytää vaihtoehtoisia ratkaisuja erityisesti pieniin teollisuuden tuotantolaitteistoihin, säilyttäen samalla tiedonkeruun tehokkuus ja luotettavuus. Samalla saatiin vastaus alussa asetettuihin tutkimuskysymyksiin: ARM-arkkitehtuuri teollisuuden automaatiotratkaisuissa? Avoimen lähdekoodin ohjelmistot tiedonkeruualustan integroinnissa? Teknologian rooli työvoimapulan helpottamisessa?

Vaikka INTEL:n prosessorit ovat vakiinnuttaneet asemansa teollisuudessa, ARM-arkkitehtuuriin perustuvat ratkaisut nousivat esille kustannustehokkuutensa ja alhaisen energiankulutuksensa ansiosta. Tutkimuksessa kävi ilmi, että vaikka ARM-alustat tarjoavat kilpailukykyisen vaihtoehdon, sen suorituskyky- ja toiminnallisuusrajoitukset on huomioitava sovelluskohtaisesti. Toisena aiheena tutkitut avoimen lähdekoodin ohjelmistot nousivat esiin kustannustehokkaina ja joustavina vaihtoehtoina. Niiden käyttöön liittyy kuitenkin haasteita erityisesti

ylläpidon ja päivitysten osalta. Tutkimus osoitti, että avoin lähdekoodi voi olla voimavara, mutta sen käytössä on huomioitava mahdollisesti työlääät ylläpitotehtävät. Työvoimapulan helpottajana voi teknologialla olla merkittävä rooli. Henkilöt, joilla ei ole syvällisiä ohjelmointitaitoja, voivat hyödyntää visuaalisen oppimisen kurssuja, joiden pohjalta he voivat myös jatkokouluttautua syvemmälle ohjelmoinnin saloihin. Avoimen lähdekoodin ohjelmistot helpottavat oppimista, ja projektityöskentely visuaalisessa ympäristössä nopeuttaa ja helpottaa työntekoa.

Laitteita vertailevassa tutkimusosuudessa rakennettiin halpuihin ratkaisuihin perustuva Siemensin PLC:hen liitettävä tiedonkeruujärjestelmä, joka soveltuu tuotekehitykseen ja käyttöönottajien apuvälineeksi ja samalla tarjoaa myös hyvän pohjan tuleville asiakasprojekteille. Tämä vaatii kuitenkin suunnittelijoilta ja suunnittelutoimistoilta perehtymistä ratkaisun toteuttamiseen. Tutkimusympäristönä ollut Siemensin S7-1500 PLC-laitteisto tarjosi monipuolisen valikoiman tiedonkeruurajapintoja ja tutkimuksessa testattiin kolme toimivaa tiedonkeruurajapintaa: ISO-On-TCP, ModbusTCP ja MQTT. Nämä protokollat osoittautuivat luotettaviksi ja laajasti käytetyiksi kommunikointirajapinnoiksi. Tutkimuksen empiiriseen osuuteen valittiin ISO-On-TCP sen kyvyn vuoksi siirtää eri muotoista dataa suoraan kohdepalvelimelle ilman datamuunnoksia. Lisäksi sen yksinkertainen ja nopea konfigurointi helpotti järjestelmän käyttöönottoa. Siemens-ympäristö edusti tutkimuksessa niin sanottua tutumpaa osaa, eikä ympäristön kanssa ollut ongelmia. Ohjelmointiympäristö ja PLC olivat sen verran vanhoja, että kaikkea haluttua ei pystytty tekemään.

Tiedonkeruun rajapintana toimineen Raspberry Pi4:n Linux-pohjaisen Debian-käyttöjärjestelmän käyttöönotto tuotti jonkun verran haasteita tutkimuksen aikana. Tutkimuksessa kävikin selväksi, että käskykannat ja ohjaukset aiheuttavat vaikeuksia, mikäli käyttäjä ei ole perehtynyt tarpeeksi Linuxin toimintaperiaatteisiin. Tähän haasteeseen voidaan vastata käyttämällä ohjeistusta ja lisäämällä koulutusta Linux-toimintaympäristöissä, jos suunnittelutoimisto on aikeissa valita Linux-pohjaisen tietokoneen palveluiden tuottamiseen. On myös



muistettava, että tutkimuksen ulkopuolelle jätetty sertifioitu OPC-UA-kommunikointirajapinta voi olla joissain tapauksissa järkevämpi valinta rajapinnaksi.

Tiedonkeruun ohjelmointiympäristö Node-RED osoittautui erittäin joustavaksi ja helppokäyttöiseksi tiedonkeruutyökaluksi. Esimerkiksi valmis tiedonkeruusovellus oli helppo siirtää Linux-ympäristöstä Windows-ympäristöön ilman ongelmia. Valmiita ratkaisuja oli runsaasti saatavilla ja ne voitiin helposti konfiguroida vastaamaan tämän tutkimusprojektin tarpeita. Vaikka Node-RED:n low-code-ohjelmointitapa nopeuttaakin ohjelmointityötä, on kirjastokomponenttien toimintaan syytä perehtyä, jotta kaikki ohjelmointimahdollisuudet osataan käyttää hyödyksi, kun suunnitellaan kommunikointia teollisuuden laitteisiin. On myös tiedostettava, että Raspberry Pi:n ja Node-RED:n kirjastojen ylläpito sekä laitteita integroivien henkilöiden perehdyttäminen vaativat resursseja, jotta OTS-järjestelmä pysyy kunnossa ja edelleen kustannustehokkaana.

## 6.1 Jatkokehitys

Tiedonkeruualustan vakaa toimiminen Siemens-ympäristössä rohkaisee jatkaa tuotekehitystä myös muiden perinteisten logiikkavalmistajien tuotteille. Aiemmin mainitulle Allen-Bradleyn Ethernet/IP-kommunikoinnille voisi löytyä helppo ratkaisu Siemensin rinnalle. Myös Node-RED:n omilla työkaluilla rakennettava käyttöliittymä voisi helpottaa operointia tiedonkeruulaitteistossa. Laajempi kehitysprojekti voisi olla sulautettu järjestelmä, jossa tiedonkeruulaitteistoon lisätään Soft-PLC, jolla voidaan päivittää vanha kone nykypäivän vaatimustasoa vastaavaksi. Myös jo valmiiksi saadun ympäristön liittäminen pilvipalveluun voisi luoda vaihtoehtoja tiedonkeruuseen ja tiedon jalostamiseen.

Yhteenvetona voidaan todeta, että Raspberry Pi:n alustalle rakennettu tiedonkeruu on kustannustehokas. Kustannustehokkuutta tukevat avoimen lähdekoodin ohjelmistot, jotka voivat vähentää lisenssikustannuksia. Lisäksi tutkimus luo perustan monille tulevaisuuden mahdollisuuksille, kuten laajemmalle tiedonkeruulle, analyytikalle ja teollisen internetin sovelluksille. Yksinkertaisilla

tiedonkeruuratkaisuilla voidaan luoda lisäarvoa teollisuuden prosesseille myös pk-yrityksissä.

## 7 Lähdeluettelo

Aalto, 2019. Verkkosivu. <https://mycourses.aalto.fi/mod/book/view.php?id=442395&chapterid=2510&lang=fi> (haettu 19.11.2023).

Alves & Morris, 2018. Thiago Alves, Thomas Morris 2018, OpenPLC: An IEC 61131–3 compliant open source industrial controller for cyber security research <https://www.sciencedirect.com/science/article/pii/S0167404818305388> (haettu 8.9.2023).

Avevaselect Edge, 2023. Verkkosivusto. <https://california.avevaselect.com/licenses/licenseinfo.aspx> (haettu 6.9.2023).

Automaatioseura, 2013. Verkkokirja, s.11. <https://www.automaatioseura.fi/site/assets/files/1426/standardikirja.pdf> (haettu 6.9.2023).

Beckhoff, 2023. Verkkosivusto. <https://www.beckhoff.com/fi-fi/industries/measurement-and-testing-technology/scalable-iiot-systems/> (haettu 6.11.2023).

Bolton, William, 2015. E-kirja. Programmable Logic Controllers, 6th Edition, s. 5. Newnes (haettu 6.5.2023).

Bosch, 2023. Verkkosivusto. <https://apps.boschrexroth.com/microsites/ctrlx-automation/en/ctrlx-world/> (haettu 6.11.2023).

Codesys, 2020. Verkkosivusto <https://www.codesys.com/news-events/press-releases/article/3s-smart-software-solutions-gmbh-becomes-codesys-gmbh.html> (haettu 8.9.2023).

Collin & Saarelainen, 2016a. Collin, Jari & Saarelainen, Ari 2016, Teollinen internet, s. 54. E-kirja. Helsinki: Talentum. [https://bisneskirjasto-almatalent.fi.ezproxy.metropolia.fi/teos/BAFBIXC-TEB#/kohta:I\(\(20\)OSA\(\(20\)Suuri\(\(20\)mahdollisuus/piste:t98](https://bisneskirjasto-almatalent.fi.ezproxy.metropolia.fi/teos/BAFBIXC-TEB#/kohta:I((20)OSA((20)Suuri((20)mahdollisuus/piste:t98) (haettu 15.8.2023).

Collin & Saarelainen, 2016b. Collin, Jari & Saarelainen, Ari 2016, Teollinen internet, s. 50. E-kirja. Helsinki: Talentum. [https://bisneskirjasto-almatalent-fi.ezproxy.metropolia.fi/teos/BAFBIXC-TEB#/kohta:l\(\(20\)OSA\(\(20\)Suuri\(\(20\)mahdollisuus/piste:t98](https://bisneskirjasto-almatalent-fi.ezproxy.metropolia.fi/teos/BAFBIXC-TEB#/kohta:l((20)OSA((20)Suuri((20)mahdollisuus/piste:t98) (haettu 15.8.2023).

Collin & Saarelainen 2016c. Collin, Jari & Saarelainen, Ari 2016, Teollinen internet, s. 48. E-kirja. Helsinki: Talentum. [https://bisneskirjasto-almatalent-fi.ezproxy.metropolia.fi/teos/BAFBIXC-TEB#/kohta:l\(\(20\)OSA\(\(20\)Suuri\(\(20\)mahdollisuus/piste:t98](https://bisneskirjasto-almatalent-fi.ezproxy.metropolia.fi/teos/BAFBIXC-TEB#/kohta:l((20)OSA((20)Suuri((20)mahdollisuus/piste:t98) (haettu 15.8.2023).

Counterpoint, 2023. Verkkosivusto. <https://www.counterpointresearch.com/insights/arm-based-pcs-to-nearly-double-market-share-by-2027/> (haettu 7.9.2023).

Influxdata, 2023. Verkkosivusto. <https://www.influxdata.com/blog/guide-installing-node-red/> (haettu 29.10.2023).

Json-ohjelmointikieli, 2023. Verkkosivusto. <http://netisto.fi/oppaat/js2/?id=05>, (haettu 6.10.2023).

L. O. Aghenta et al., 2019. L. O. Aghenta and M. T. Iqbal, "Low-Cost, Open Source IoT-Based SCADA System Design Using Thingier.IO and ESP32Thing", s.19. [https://www.researchgate.net/publication/334661815\\_Low-cost\\_open\\_source\\_IoT-based\\_SCADA\\_system\\_design\\_using\\_thingierIO\\_and\\_ESP32\\_thing#fullTextFileContent](https://www.researchgate.net/publication/334661815_Low-cost_open_source_IoT-based_SCADA_system_design_using_thingierIO_and_ESP32_thing#fullTextFileContent) (haettu 6.9.2023).

Martikkala et al., 2021a. Antti Martikkala, Joe Davida, Andrei Lobovb, Minna Lanza, Iñigo Flores Ituarte, Trends for Low-Cost and Open-Source IoT Solutions Development for Industry, s. 299. [https://www.researchgate.net/publication/355893498\\_Trends\\_for\\_Low-Cost\\_and\\_Open-Source\\_IoT\\_Solutions\\_Development\\_for\\_Industry\\_40](https://www.researchgate.net/publication/355893498_Trends_for_Low-Cost_and_Open-Source_IoT_Solutions_Development_for_Industry_40) (haettu 7.9.2023).

Martikkala et al., 2021b. Antti Martikkala, Joe Davida, Andrei Lobovb, Minna Lanza, Iñigo Flores Ituarte, Trends for Low-Cost and Open-Source IoT

Solutions Development for Industry, s.301, s. 302. <https://www.researchgate.net/publication/355893498> Trends for Low-Cost and Open-Source IoT Solutions Development for Industry 40 (haettu 7.9.2023).

Metropolia, 2016. Verkkosivusto. <https://wiki.metropolia.fi/pages/viewpage.action?pagelid=138445135> (haettu 16.11.2023).

ModbusTCP, 2023. Modbus organization. <https://modbus.org/faq.php> (haettu 13.9.2023).

MQTT, 2019. MQTT Version 5.0 OASIS Standard, s 2. <https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.pdf> (haettu 13.9.2023).

Nikander, 2006. Visuaalinen ohjelmointi tietotekniikan perusopetuksessa, s. 1-2. <https://www.researchgate.net/publication/242660845> Visuaalinen ohjelmointi tietotekniikan perusopetuksessa (haettu 2.11.2023).

Node-RED, 2023. Verkkosivusto. <https://nodered.org/docs/user-guide/messages#understanding-msgparts> (haettu 19.11.2023).

NodeRed nygma2004, 2023. Verkkosivusto. <https://flows.nodered.org/flow/4f0653905120f422cc4f37cd5512ee14> (haettu 31.10.2023).

Npm, 2023. Verkkosivusto. <https://docs.npmjs.com/about-npm> (haettu 16.11.2023).

OPC Foudation, 2023. Verkkosivusto. <https://opcfoundation.org/about/what-is-opc/> (haettu 13.9.2023).

PeakIn, 2023. Verkkosivusto. <https://peakin.in/importance-of-python-for-data-science-career/> (haettu 16.11.2023).

Perko, Minna, 2021. Opinnäytetyö: OTS-Järjestelmän implementointivaihtoehdot, s.17. [https://www.theseus.fi/bitstream/handle/10024/427088/Perko\\_Minna.pdf](https://www.theseus.fi/bitstream/handle/10024/427088/Perko_Minna.pdf) (haettu 25.9.2023).

Phoenix, 2023. Verkkosivusto. <https://www.phoenixcontact.com/en-us/us-lp-plc-next-technology> (haettu 6.11.2023).

Python.org, 2023. Python quotes. <https://www.python.org/about/quotes/> (haettu 6.10.2023).

Saksa, Tommi, 2020. Verkkójulkaisu. Paras ohjelmointikieli ohjelmoinnin opiskelun aloittamiseen. *HAMK Unlimited Journal* 17.4.2020. (haettu 6.10.2023).

Siemens cpu150x, 2019. Siemens cpu150x datasheet, s. 31. [https://support.industry.siemens.com/cs/attachments/109760977/s71500\\_cpu150xs\\_v26\\_product\\_information\\_x\\_en-US.pdf](https://support.industry.siemens.com/cs/attachments/109760977/s71500_cpu150xs_v26_product_information_x_en-US.pdf) (haettu 11.9.2023).

Siemens Know-how, 2019. Setting up block know-how protection. <https://support.industry.siemens.com/cs/mdm/109773506?c=95119589259&lc=en-WW> (haettu 29.10.2023).

Siemens Modbus/TCP, 2016. Verkkosivusto. [https://support.industry.siemens.com/cs/document/109741480/licensing-modbus-tcp-blocks-for-s7-300-s7-400-\(video\)?dti=0&lc=en-ID](https://support.industry.siemens.com/cs/document/109741480/licensing-modbus-tcp-blocks-for-s7-300-s7-400-(video)?dti=0&lc=en-ID) (haettu 1.11.2023).

Siemens REST API, 2019. Library for LHTTP Communication, s. 4. [https://cache.industry.siemens.com/dl/files/879/109763879/att\\_987991/v2/109763879\\_LHTTP\\_DOC\\_V10\\_en.pdf](https://cache.industry.siemens.com/dl/files/879/109763879/att_987991/v2/109763879_LHTTP_DOC_V10_en.pdf) (haettu 1.11.2023).

Siemens sertifikaatti, 2019. Using certificates with TIA-portal, <https://support.industry.siemens.com/cs/document/109769068/using-certificates-with-tia-portal?dti=0&lc=en-US>, (haettu 29.10.2023).

Snap7, 2023. Verkkosivusto. <https://snap7.sourceforge.net/> (haettu 12.9.2023).

Statista, 2017. Programmable logic controllers global manufacturer marketshare. <https://www.statista.com/statistics/897201/global-plc-market-share-by-manufacturer/> (haettu 20.9.2023).

Stenfors R., 2021. Prosessiautomaation aikasarjadataan tallentamisratkaisu s.14, <https://urn.fi/URN:NBN:fi:tuni-202102242236> (haettu 3.7.2023).

Teknolohiateollisuus, 2023. Teknolohiateollisuus 05-2023 Powerpoint esitys, s. 18, s. 10. <https://teknolohiateollisuus.fi/sites/default/files/2023-05/Henkil%C3%B6st%C3%B6.pdf> (haettu 12.9.2023).

TEM raportti, 2023. Työvoiman saatavuus, työvoimapula ja kohtaanto-ongelmat vuonna 2022, s. 16, s. 234. [https://julkaisut.valtioneuvosto.fi/bitstream/handle/10024/164550/Ty%C3%B6voimatiekartat\\_hankkeen%20loppuraportti\\_tarkistettu.pdf](https://julkaisut.valtioneuvosto.fi/bitstream/handle/10024/164550/Ty%C3%B6voimatiekartat_hankkeen%20loppuraportti_tarkistettu.pdf) (haettu 6.9.2023).

Termipankki, 2023. Ohjelmoitava rajapinta. <https://termipankki.fi/tepa/fi/haku/ohjelmointirajapinta> (haettu 4.10.2023).

Valtioneuvosto 2015a. Valtioneuvoston selvitys- ja tutkimustoiminnan julkaisusarja 4/2015, Suomi - Teollisen Internetin piilaakso. <http://urn.fi/URN:ISBN:978-952-287-174-9> (haettu 10.8.2023).

Valtioneuvosto, 2015b. Valtioneuvoston selvitys- ja tutkimustoiminnan julkaisusarja 4/2015, Suomi - Teollisen Internetin piilaakso, s. 32. <http://urn.fi/URN:ISBN:978-952-287-174-9> (haettu 10.8.2023).

Valtioneuvosto, 2015c. Valtioneuvoston selvitys- ja tutkimustoiminnan julkaisusarja 4/2015, Suomi - Teollisen Internetin piilaakso, s. 21, s. 25. <http://urn.fi/URN:ISBN:978-952-287-174-9> (haettu 10.8.2023).

Valtioneuvosto, 2015d. Valtioneuvoston selvitys- ja tutkimustoiminnan julkaisusarja 4/2015, Suomi - Teollisen Internetin piilaakso, s. 29.

<http://urn.fi/URN:ISBN:978-952-287-174-9> (haettu 10.8.2023).

VTT, 2018. Teollisen internetin palvelut valmistavassa teollisuudessa, VTT, s. 3, 2018. [https://publications.vtt.fi/julkaisut/muut/2018/OA\\_Teollisen\\_internetin\\_palvelut\\_valmistavassa\\_teollisuudessa.pdf](https://publications.vtt.fi/julkaisut/muut/2018/OA_Teollisen_internetin_palvelut_valmistavassa_teollisuudessa.pdf) (haettu 7.9.2023).

VTT verkkotiedote, 2023. <https://www.vttresearch.com/fi/uutiset-ja-tarinat/pk-yritykset-haluttomia-investoimaan-uuteen-teknologiaan-vetoapua-tarjolla> (haettu 15.8.2023).

WAGO, 2023. Verkkosivusto [https://www.wago.com/fi/ecockpit-suunnitteluohjelmisto?gad=1&qclid=Cj0KCQiAuqKqBhDxARIsAFZELmKfiX2T30YQKO\\_zvB-Z3j-B5b5uZGMgLera2p26nYzpUUCXHFpJOBkaAi4VEALw\\_wcB](https://www.wago.com/fi/ecockpit-suunnitteluohjelmisto?gad=1&qclid=Cj0KCQiAuqKqBhDxARIsAFZELmKfiX2T30YQKO_zvB-Z3j-B5b5uZGMgLera2p26nYzpUUCXHFpJOBkaAi4VEALw_wcB) (haettu 6.11.2023).



## Liite1: NodeRED-ohjelmakoodi S7-tiedonkeruuseen

```
{
  "id": "f79c29b7.055068",
  "type": "comment",
  "z": "62d8dcd694d13129",
  "name": "Auto Logger",
  "info": "",
  "x": 150,
  "y": 420,
  "wires": [],
  "id": "4c8ae605.33be88",
  "type": "function",
  "z": "62d8dcd694d13129",
  "name": "Set data",
  "func": "var now = new Date();\nvar mittausdata = msg.payload\nmsg.payload = {\n  \"timestamp\": now.getTime(),\n  \"Motor1OnSt\": mittausdata.Motor1OnSt,\n  \"Motor1FaultSt\": mittausdata.Motor1FaultSt,\n  \"Motor2OnSt\": mittausdata.Motor2OnSt,\n  \"Motor2FaultSt\": mittausdata.Motor2FaultSt,\n  \"Motor3OnSt\": mittausdata.Motor3OnSt,\n  \"Motor3FaultSt\": mittausdata.Motor3FaultSt,\n  \"Motor4OnSt\": mittausdata.Motor4OnSt,\n  \"Motor4FaultSt\": mittausdata.Motor4FaultSt,\n  \"Valve1OpenSt\": mittausdata.Valve1OpenSt,\n  \"Valve1CloseSt\": mittausdata.Valve1CloseSt,\n  \"Valve2OpenSt\": mittausdata.Valve2OpenSt,\n  \"Valve2CloseSt\": mittausdata.Valve2CloseSt,\n  \"LineOnSt\": mittausdata.LineOnSt,\n  \"EstopOnSt\": mittausdata.EstopOnSt,\n  \"GeneralFailureSt\": mittausdata.GeneralFailureSt,\n  \"GeneralWarningSt\": mittausdata.GeneralWarningSt,\n  \"Motor1OnCmd\": mittausdata.Motor1OnCmd,\n  \"Motor2OnCmd\": mittausdata.Motor2OnCmd,\n  \"Motor3OnCmd\": mittausdata.Motor3OnCmd,\n  \"Motor4OnCmd\": mittausdata.Motor4OnCmd,\n  \"Valve1OnCmd\": mittausdata.Valve1OnCmd,\n  \"Valve2OnCmd\": mittausdata.Valve2OnCmd,\n  \"LineStartCmd\": mittausdata.LineStartCmd,\n  \"SpareCommand8\": mittausdata.SpareCommand8,\n  \"SpareCommand9\": mittausdata.SpareCommand9,\n  \"SpareCommand10\": mittausdata.SpareCommand10,\n  \"SpareCommand11\": mittausdata.SpareCommand11,\n  \"SpareCommand12\": mittausdata.SpareCommand12,\n  \"SpareCommand13\": mittausdata.SpareCommand13,\n  \"SpareCommand14\": mittausdata.SpareCommand14,\n  \"SpareCommand15\": mittausdata.SpareCommand15,\n  \"SpareCommand16\": mittausdata.SpareCommand16,\n  \"Mittaus1\": mittausdata.Mittaus1,\n  \"Mittaus2\": mittausdata.Mittaus2,\n  \"Mittaus3\": mittausdata.Mittaus3,\n  \"Mittaus4\": mittausdata.Mittaus4,\n  \"Mittaus5\": mittausdata.Mittaus5,\n  \"Mittaus6\": mittausdata.Mittaus6,\n  \"Mittaus7\": mittausdata.Mittaus7,\n  \"Mittaus8\": mittausdata.Mittaus8,\n  \"Mittaus9\": mittausdata.Mittaus9,\n  \"Mittaus10\": mittausdata.Mittaus10,\n  \"Mittaus11\": mittausdata.Mittaus11,\n  \"Mittaus12\": mittausdata.Mittaus12,\n  \"Mittaus13\": mittausdata.Mittaus13,\n  \"Mittaus14\": mittausdata.Mittaus14,\n  \"Mittaus15\": mittausdata.Mittaus15,\n  \"Mittaus16\": mittausdata.Mittaus16,\n  \"Mittaus17\": mittausdata.Mittaus17,\n  \"Mittaus18\": mittausdata.Mittaus18,\n  \"Mittaus19\": mittausdata.Mittaus19,\n  \"Mittaus20\": mittausdata.Mittaus20,\n  \"Mittaus21\": mittausdata.Mittaus21,\n  \"mittaus22\": mittausdata.Mittaus22,\n  \"mittaus23\": mittausdata.Mittaus23,\n  \"mittaus24\": mittausdata.Mittaus24,\n  \"mittaus25\": mittausdata.Mittaus25,\n  \"mittaus26\": mittausdata.Mittaus26,\n  \"mittaus27\": mittausdata.Mittaus27,\n  \"mittaus28\": mittausdata.Mittaus28,\n  \"mittaus29\": mittausdata.Mittaus29,\n  \"mittaus30\": mittausdata.Mittaus30,\n  \"mittaus31\": mittausdata.Mittaus31,\n  \"mittaus32\": mittausdata.Mittaus32,\n  \"asetusarvo1\": mittausdata.Asetusarvo1,\n  \"asetusarvo2\": mittausdata.Asetusarvo2,\n  \"asetusarvo3\": mittausdata.Asetusarvo3,\n  \"asetusarvo4\": mittausdata.Asetusarvo4,\n  \"asetusarvo5\": mittausdata.Asetusarvo5,\n  \"asetusarvo6\": mittausdata.Asetusarvo6,\n  \"asetusarvo7\": mittausdata.Asetusarvo7,\n  \"asetusarvo8\": mittausdata.Asetusarvo8,\n  \"asetusarvo9\": mittausdata.Asetusarvo9,\n  \"asetusarvo10\": mittausdata.Asetusarvo10,\n  \"asetusarvo11\": mittausdata.Asetusarvo11,\n  \"asetusarvo12\": mittausdata.Asetusarvo12,\n  \"asetusarvo13\": mittausdata.Asetusarvo13,\n  \"asetusarvo14\": mittausdata.Asetusarvo14,\n  \"asetusarvo15\": mittausdata.Asetusarvo15,\n  \"asetusarvo16\": mittausdata.Asetusarvo16,\n  \"asetusarvo17\": mittausdata.Asetusarvo17,\n  \"asetusarvo18\": mittausdata.Asetusarvo18,\n  \"asetusarvo19\": mittausdata.Asetusarvo19,\n  \"asetusarvo20\": mittausdata.Asetusarvo20,\n  \"asetusarvo21\": mittausdata.Asetusarvo21,\n  \"asetusarvo22\": mittausdata.Asetusarvo22,\n  \"asetusarvo23\": mittausdata.Asetusarvo23,\n  \"asetusarvo24\": mittausdata.Asetusarvo24,\n  \"asetusarvo25\": mittausdata.Asetusarvo25,\n  \"asetusarvo26\": mittausdata.Asetusarvo26,\n  \"asetusarvo27\": mittausdata.Asetusarvo27,\n  \"asetusarvo28\": mittausdata.Asetusarvo28,\n  \"asetusarvo29\": mittausdata.Asetusarvo29,\n  \"asetusarvo30\": mittausdata.Asetusarvo30,\n  \"asetusarvo31\": mittausdata.Asetusarvo31,\n  \"asetusarvo32\": mittausdata.Asetusarvo32\n}\n\nreturn
```

```

msg;,"outputs":1,"timeout":"","noerr":0,"initialize":"","finalize":"","libs":[],"x":240,"y":520,"wires":[["f46f569e.ff1a08"]],{"id":"f46f569e.ff1a08","type":"function","z":"62d8dcd694d13129","name":"Filename generator","func":"// Get the current time and convert it to text\nvar now = new Date();\nvar yyyy = now.getFullYear();\nvar mm = now.getMonth() < 9 ? \"0\" +\n(now.getMonth() + 1) : (now.getMonth() + 1); // getMonth() is zero-based\nvar dd = now.getDate() < 10 ? \"0\" +\nnow.getDate() : now.getDate();\nvar hh = now.getHours() < 10 ? \"0\" + now.getHours() : now.getHours();\nvar mmm = now.getMinutes() < 10 ? \"0\" + now.getMinutes() : now.getMinutes();\nvar ss = now.getSeconds() < 10 ?\n\"0\" + now.getSeconds() : now.getSeconds();\n\n// Generate out file name pattern\n//msg.fname = \"S7IoT_\" +\nyyyy + mm + dd + \".csv\";\nmsg.fname = \"S7IoT_\" + yyyy + mm + dd + hh + mmm + \".csv\";\n// Full filename\nwith path for the file node later\nmsg.filename = \"/home/pi/datalog/\" + msg.fname;\n\n// We save the current\npayload into a different place on the msg object\nmsg.filecontent = msg.payload;\n\n// We are passing the file\nname search pattern to fs node to tell us if the file exists or not\nmsg.payload = {\"pat-  
tern\":msg.fname};\n\nnode.status({fill:\"blue\",shape:\"ring\",text:msg.fname});\nreturn msg;","outputs":1,"timeout":"","noerr":0,"initialize":"","finalize":"","libs":[],"x":210,"y":640,"wires":[["fa6936f.7758dc8"]],{"id":"a96494a5.b0f4c8","type":"debug","z":"62d8dcd694d13129","name":"","active":false,"tosidebar":true,"console":false,"tostatus":false,"complete":false,"x":570,"y":560,"wires":[],"id":"3db97537.b889ea","type":"switch","z":"62d8dcd694d13129","name":"","property":"$count(msg.payload)","propertyType":"jsonata","rules":[{"t":"eq","v":"0","vt":"num"},{"t":"else"}],"checkall":"true","repair":false,"outputs":2,"x":550,"y":640,"wires":[["3cb887e8.e533a8"],["f3ba1c7a.54ef5"]],{"id":"fe6f74d5.674ec8","type":"csv","z":"62d8dcd694d13129","name":"","sep":",","hdrin":"","hdrout":"all","multi":"one","ret":"\\n","temp":"timestamp,Motor1FaultSt,Motor2OnSt,Motor2FaultSt,Motor3OnSt,Motor3FaultSt,Motor4OnSt,Motor4FaultSt,Valve1OpenSt,Valve1CloseSt,Valve2OpenSt,Valve2CloseSt,LineOnSt,EstopOnSt,GeneralFailureSt,GeneralWarningSt,Motor1OnCmd,Motor2OnCmd,Motor3OnCmd,Motor4OnCmd,Valve1OnCmd,Valve2OnCmd,LineStartCmd,SpareCommand8,SpareCommand9,SpareCommand10,SpareCommand11,SpareCommand12,SpareCommand13,SpareCommand14,SpareCommand15,SpareCommand16,Mittaus1,Mittaus2,Mittaus3,Mittaus4,Mittaus5,Mittaus6,Mittaus7,Mittaus8,Mittaus9,Mittaus10,Mittaus11,Mittaus12,Mittaus13,Mittaus14,Mittaus15,Mittaus16,Mittaus17,Mittaus18,Mittaus19,mittaus20,mittaus21,mittaus22,mittaus23,mittaus24,mittaus25,mittaus26,mittaus27,mittaus28,mittaus29,mittaus30,mittaus31,mittaus32,asetusarvo1,asetusarvo2,asetusarvo3,asetusarvo4,asetusarvo5,asetusarvo6,asetusarvo7,asetusarvo8,asetusarvo9,asetusarvo10,asetusarvo11,asetusarvo12,asetusarvo13,asetusarvo14,asetusarvo15,asetusarvo16,asetusarvo17,asetusarvo18,asetusarvo19,asetusarvo20,asetusarvo21,asetusarvo22,asetusarvo23,asetusarvo24,asetusarvo25,asetusarvo26,asetusarvo27,asetusarvo28,asetusarvo29,asetusarvo30,asetusarvo31,asetusarvo32","skip":"0","strings":true,"include_empty_strings":false,"include_null_val-  
ues":false,"x":890,"y":620,"wires":[["3a8ebdf3.772cc2"]],{"id":"3a8ebdf3.772cc2","type":"file","z":"62d8dcd694d13129","name":"","filename":"filename","filenameType":"msg","appendNewline":false,"createDir":true,"overwriteFile":false,"encoding":"none","x":1050,"y":620,"wires":[],"id":"f226a86d.9bdfb8","type":"csv","z":"62d8dcd694d13129","name":"","sep":",","hdrin":"","hdrout":"none","multi":"one","ret":"\\n","temp":"timestamp,Motor1FaultSt,Motor2OnSt,Motor2FaultSt,Motor3OnSt,Motor3FaultSt,Motor4OnSt,Motor4FaultSt,Valve1OpenSt,Valve1CloseSt,Valve2OpenSt,Valve2CloseSt,LineOnSt,EstopOnSt,GeneralFailureSt,GeneralWarningSt,Motor1OnCmd,Motor2OnCmd,Motor3OnCmd,Motor4OnCmd,Valve1OnCmd,Valve2OnCmd,LineStartCmd,SpareCommand8,SpareCommand9,SpareCommand10,SpareCommand11,SpareCommand12,SpareCommand13,SpareCommand14,SpareCommand15,SpareCommand16,Mittaus1,Mittaus2,Mittaus3,Mittaus4,Mittaus5,Mittaus6,Mittaus7,Mittaus8,Mittaus9,Mittaus10,Mittaus11,Mittaus12,Mittaus13,Mittaus14,Mittaus15,Mittaus16,Mittaus17,Mittaus18,Mittaus19,mittaus20,mittaus21,mittaus22,mittaus23,mittaus24,mittaus25,mittaus26,mittaus27,mittaus28,mittaus29,mittaus30,mittaus31,mittaus32,asetusarvo1,asetusarvo2,asetusarvo3,asetusarvo4,asetusarvo5,asetusarvo6,asetusarvo7,asetusarvo8,asetusarvo9,asetusarvo10,asetusarvo11,asetusarvo12,asetusarvo13,asetusarvo14,asetusarvo15,asetusarvo16,asetusarvo17,asetusarvo18,asetusarvo19,asetusarvo20,asetusarvo21,asetusarvo22,asetusarvo23,asetusarvo24,asetusarvo25,a

```

```
setusarvo26,asetusarvo27,asetusarvo28,asetusarvo29,aset","skip":"0","strings":true,"include_empty_strings":false,"include_null_values":false,"x":890,"y":660,"wires":{"3a8ebdf3.772cc2"}},{"id":"3cb887e8.e533a8","type":"change","z":"62d8dcd694d13129","name":"Get file content","rules":{"t":"set","p":"payload","pt":"msg","to":"filecontent","tot":"msg"},"action":"","property":"","from":"","to":"","reg":false,"x":720,"y":620,"wires":{"fe6f74d5.674ec8"}},{"id":"f3ba1c7a.54ef5","type":"change","z":"62d8dcd694d13129","name":"Get file content","rules":{"t":"set","p":"payload","pt":"msg","to":"filecontent","tot":"msg"},"action":"","property":"","from":"","to":"","reg":false,"x":720,"y":660,"wires":{"f226a86d.9bdfb8"}},{"id":"fa6936f.7758dc8","type":"fs-file-lister","z":"62d8dcd694d13129","name":"","start":"/home/pi/datalog/","pattern":"","folders":"","hidden":true,"lstype":"files","path":true,"single":true,"depth":0,"stat":true,"showWarnings":false,"x":400,"y":640,"wires":{"a96494a5.b0f4c8","3db97537.b889ea"}},{"id":"235d2de2a61caf67","type":"s7 7 in","z":"62d8dcd694d13129","endpoint":"78207719fbe80f42","mode":"all","variable":"","diff":true,"name":"S7 data input","x":110,"y":480,"wires":{"4c8ae605.33be88"}},{"id":"78207719fbe80f42","type":"s7 endpoint","transport":"iso-on-tcp","address":"10.20.0.3","port":"102","rack":"0","slot":"1","localtsaphi":"01","localtsaplo":"00","remotetsaphi":"01","remotetsaplo":"00","connmode":"rack-slot","adapter":"","busaddr":"2","cycletime":"1000","timeout":"2000","name":"","vartable":{"addr":"DB102,X0.0","name":"Motor1OnSt"},{"addr":"DB102,X0.1","name":"Motor1FaultSt"},{"addr":"DB102,X0.2","name":"Motor2OnSt"},{"addr":"DB102,X0.3","name":"Motor2FaultSt"},{"addr":"DB102,X0.4","name":"Motor3OnSt"},{"addr":"DB102,X0.5","name":"Motor3FaultSt"},{"addr":"DB102,X0.6","name":"Motor4OnSt"},{"addr":"DB102,X0.7","name":"Motor4FaultSt"},{"addr":"DB102,X1.0","name":"Valve1OpenSt"},{"addr":"DB102,X1.1","name":"Valve1CloseSt"},{"addr":"DB102,X1.2","name":"Valve2OpenSt"},{"addr":"DB102,X1.3","name":"Valve2CloseSt"},{"addr":"DB102,X1.4","name":"LineOnSt"},{"addr":"DB102,X1.5","name":"EstopOnSt"},{"addr":"DB102,X1.6","name":"GeneralFailureSt"},{"addr":"DB102,X1.7","name":"GeneralWarningSt"},{"addr":"DB102,X2.0","name":"Motor1OnCmd"},{"addr":"DB102,X2.1","name":"Motor2OnCmd"},{"addr":"DB102,X2.2","name":"Motor3OnCmd"},{"addr":"DB102,X2.3","name":"Motor4OnCmd"},{"addr":"DB102,X2.4","name":"Valve1OnCmd"},{"addr":"DB102,X2.5","name":"Valve2OnCmd"},{"addr":"DB102,X2.6","name":"LineStartCmd"},{"addr":"DB102,X2.7","name":"SpareCommand8"},{"addr":"DB102,X3.0","name":"SpareCommand9"},{"addr":"DB102,X3.1","name":"SpareCommand10"},{"addr":"DB102,X3.2","name":"SpareCommand11"},{"addr":"DB102,X3.3","name":"SpareCommand12"},{"addr":"DB102,X3.4","name":"SpareCommand13"},{"addr":"DB102,X3.5","name":"SpareCommand14"},{"addr":"DB102,X3.6","name":"SpareCommand15"},{"addr":"DB102,X3.7","name":"SpareCommand16"},{"addr":"DB102,R4","name":"Mittaus1"},{"addr":"DB102,R8","name":"Mittaus2"},{"addr":"DB102,R12","name":"Mittaus3"},{"addr":"DB102,R16","name":"Mittaus4"},{"addr":"DB102,R20","name":"Mittaus5"},{"addr":"DB102,R24","name":"Mittaus6"},{"addr":"DB102,R28","name":"Mittaus7"},{"addr":"DB102,R32","name":"Mittaus8"},{"addr":"DB102,R36","name":"Mittaus9"},{"addr":"DB102,R40","name":"Mittaus10"},{"addr":"DB102,R44","name":"Mittaus11"},{"addr":"DB102,R48","name":"Mittaus12"},{"addr":"DB102,R52","name":"Mittaus13"},{"addr":"DB102,R56","name":"Mittaus14"},{"addr":"DB102,R60","name":"Mittaus15"},{"addr":"DB102,R64","name":"Mittaus16"},{"addr":"DB102,R68","name":"Mittaus17"},{"addr":"DB102,R72","name":"Mittaus18"},{"addr":"DB102,R76","name":"Mittaus19"},{"addr":"DB102,R80","name":"Mittaus20"},{"addr":"DB102,R84","name":"Mittaus21"},{"addr":"DB102,R88","name":"Mittaus22"},{"addr":"DB102,R92","name":"Mittaus23"},{"addr":"DB102,R96","name":"Mittaus24"},{"addr":"DB102,R100","name":"Mittaus25"},{"addr":"DB102,R104","name":"Mittaus26"},{"addr":"DB102,R108","name":"Mittaus27"},{"addr":"DB102,R112","name":"Mittaus28"},{"addr":"DB102,R116","name":"Mittaus29"},{"addr":"DB102,R120","name":"Mittaus30"},{"addr":"DB102,R124","name":"Mittaus31"},{"addr":"DB102,R128","name":"Mittaus32"},{"addr":"DB102,R132","name":"Asetusarvo1"},{"addr":"DB102,R136","name":"Asetusarvo2"},{"addr":"DB102,R140","name":"Asetusarvo3"},{"addr":"DB102,R144","name":"Asetusarvo4"},{"addr":"DB102,R148","name":"Asetusarvo5"},{"addr":"DB102,R152","name":"Asetusarvo6"},{"addr":"DB102,R156","name":"Asetusarvo7"},{"addr":"
```

```
DB102,R160", "name": "Asetusarvo8"}, {"addr": "DB102,R164", "name": "Asetusarvo9"}, {"addr": "DB102,R168", "name":  
"Asetusarvo10"}, {"addr": "DB102,R172", "name": "Asetusarvo11"}, {"addr": "DB102,R176", "name": "Asetusarvo12"}, {"  
addr": "DB102,R180", "name": "Asetusarvo13"}, {"addr": "DB102,R184", "name": "Asetusarvo14"}, {"addr": "DB102,R188  
", "name": "Asetusarvo15"}, {"addr": "DB102,R192", "name": "Asetusarvo16"}, {"addr": "DB102,R196", "name": "Asetusar  
vo17"}, {"addr": "DB102,R200", "name": "Asetusarvo18"}, {"addr": "DB102,R204", "name": "Asetusarvo19"}, {"addr": "DB  
102,R208", "name": "Asetusarvo20"}, {"addr": "DB102,R212", "name": "Asetusarvo21"}, {"addr": "DB102,R216", "name":  
"Asetusarvo22"}, {"addr": "DB102,R220", "name": "Asetusarvo23"}, {"addr": "DB102,R224", "name": "Asetusarvo24"}, {"  
addr": "DB102,R228", "name": "Asetusarvo25"}, {"addr": "DB102,R232", "name": "Asetusarvo26"}, {"addr": "DB102,R236  
", "name": "Asetusarvo27"}, {"addr": "DB102,R240", "name": "Asetusarvo28"}, {"addr": "DB102,R244", "name": "Asetusar  
vo29"}, {"addr": "DB102,R248", "name": "Asetusarvo30"}, {"addr": "DB102,R252", "name": "Asetusarvo31"}, {"addr": "DB  
102,R256", "name": "Asetusarvo32"}]]]
```