



SEINÄJOEN AMMATTIKORKEAKOULU
SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

Riku Kallioniemi

Putkentyöstökoneiden ohjelmiston kehittäminen

Opinnäytetyö

Kevät 2022

Automaatiotekniikan tutkinto-ohjelma



SEINÄJOEN AMMATTIKORKEAKOULU

Opinnäytetyön tiivistelmä

Tutkinto-ohjelma: Automaatiotekniikka

Suuntautumisvaihtoehto: Sähköautomaatio

Tekijä: Riku Kallioniemi

Työn nimi: Putkentyöstökoneiden ohjelmiston kehittäminen

Ohjaaja: Niko Ristimäki

Vuosi: 2022

Sivumäärä: 46

Liitteiden lukumäärä: 0

Opinnäytetyön tavoitteena oli tutkia ja kehittää tiedonkeruuohjelmistoa T-Drill-putkentyöstökoneille. Tavoitteena oli kehittää tiedonkeruuohjelmisto, joka olisi yhteensopiva kaikille yrityksen valmistamille putkentyöstökoneille. Ohjelmiston tuli kerätä hyödyllistä dataa yritykselle muodossa, jota voidaan lähettää yrityksen käyttämään pilvipalveluun Beckhoffin ohjaimelta. Lisäksi tavoitteena oli kehittää putkentyöstökoneiden huoltoilmoituksia.

Työssä selvitettiin helpoin toteutustapa tiedonkeruuohjelmistolle käyttäen kvantitatiivista menetelmää funktioiden testaamiselle ja tutkimiselle. Työssä selvitettiin laitteiden kommunikointiyhteydet, jotta ymmärrettiin, missä muodossa tietoa voitiin lähettää pilvipalveluun.

Opinnäytetyön tuloksena kehitettiin toimilohko Beckhoffin ohjaimelle, josta ulostulevat tiedot ovat sellaisissa muodoissa, joita voidaan lähettää Noux Noden pilvipalveluun. Toimilohkon koodi dokumentoitiin ja toimilohkosta tehtiin lyhyt käyttöohje yrityksen työntekijöille. Lisäksi kehitettiin putkentyöstökoneiden kosketusnäyttöpaneelien huoltoilmoituksille lisäinformaatio sivut.

¹ Asiasanat: TwinCAT, PLC, tiedonkeruu, toimilohko, pilvipalvelu

SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

Thesis abstract

Degree programme: Automation Engineering

Specialisation: Electrical Automation

Author: Riku Kallioniemi

Title of thesis: Development of Software for Tube and Pipe Processing Machines

Supervisor: Niko Ristimäki

Year: 2022

Number of pages: 46

Number of appendices: 0

The goal of the project was to develop a data collection program for T-Drill tube and pipe processing machines which would send data to the cloud from Beckhoff's controller. The data collection program needed to receive measured data from sensors and send it to the cloud via Beckhoff's own data bus. The data collection program also needed to be compatible with all the client's machines.

The thesis contained a lot of testing with different kinds of methods and functions and the aim was to find the most optimized settings for the data collection program. To find functions worth testing, a quantitative method was used in this project to compare numerous functions. The study focused on the communication connections between devices to understand in which form data can be sent to the cloud service.

As the result of the thesis project there was a ready to use function block, whose output signals are in a format that can be transmitted from Beckhoff's controller to Noux Node's cloud. The function block includes necessary information which demonstrates the purpose of its use. Also, additional information pages were provided for the maintenance notifications of the touch screen panel of the tube and pipe processing machines.

¹ Keywords: TwinCAT, PLC, data gathering system, function block, cloud service

SISÄLTÖ

| | |
|--|----|
| Opinnäytetyön tiivistelmä | 1 |
| Thesis abstract | 2 |
| SISÄLTÖ | 3 |
| Kuva-, kuvio- ja taulukkoluetelo | 5 |
| Käytetyt termit ja lyhenteet..... | 7 |
| 1 JOHDANTO | 8 |
| 1.1 Työn tausta | 8 |
| 1.2 Työn tavoite..... | 8 |
| 1.3 Työn rakenne | 8 |
| 1.4 Yritysesittely | 9 |
| 2 LAITTEISTO..... | 10 |
| 2.1 PLC | 10 |
| 2.1.1 Soft PLC..... | 10 |
| 2.2 HMI-paneeli..... | 12 |
| 3 YMPÄRISTÖ | 14 |
| 3.1 Ohjelmoitavan logiikan ohjelmointialustat..... | 14 |
| 3.2 TwinCAT | 14 |
| 3.3 ADS..... | 15 |
| 3.4 IEC 61131-3 | 16 |
| 3.4.1 FBD..... | 16 |
| 3.4.2 ST | 16 |
| 3.5 Yhteydet..... | 17 |
| 4 PILVIPALVELUT | 19 |
| 4.1 Termistö | 19 |
| 4.2 Palvelumallit | 21 |
| 4.3 Toteutustavat..... | 22 |
| 4.4 Pilvilaskenta ja sen hyödyt | 23 |
| 5 SERVOTEKNIikka | 25 |

| | |
|--|----|
| 5.1 Servojärjestelmä..... | 25 |
| 6 SUUNNITTELU | 27 |
| 7 TOTEUTUS..... | 28 |
| 7.1 Toimilohkon teko | 28 |
| 7.2 Testaus | 34 |
| 7.3 Huoltoilmoitusten lisäinformaatiosivujen teko..... | 36 |
| 8 POHDINTA JA YHTEENVETO..... | 42 |
| LÄHTEET | 44 |

Kuva-, kuvio- ja taulukkoluetelo

| | |
|---|----|
| Kuva 1. Beckhoff CX5130 -teollisuus-PC..... | 11 |
| Kuva 2. Beckhoff CP2916 -HMI-paneeli | 13 |
| Kuva 3. Putkentyöstökoneiden yhteyksien infrastruktuuri | 17 |
| Kuva 4. Servojärjestelmän rakenne (mukaihen Fonselius ym, 1998, s. 8). | 25 |
| Kuva 5. Servomoottorin ohjaus..... | 26 |
| Kuva 6. Projektin valinta ja nimeäminen. | 28 |
| Kuva 7. PLC-alustan lisääminen projektiin..... | 29 |
| Kuva 8. Kirjastojen lisääminen projektiin..... | 29 |
| Kuva 9. Toimilohkon lisääminen 1. | 30 |
| Kuva 10. Toimilohkon lisääminen 2. | 31 |
| Kuva 11. Akselin referenssidatan sisääntulo toimilohkolle | 31 |
| Kuva 12. Toimilohkon ulostulot akselin referenssidatalle..... | 32 |
| Kuva 13. Akselin referenssidatan linkitykset omille muuttujille..... | 32 |
| Kuva 14. Toimilohkon muuttujan lisääminen toimilohkoon..... | 32 |
| Kuva 15. Toimilohkon sisääntulomuuttuja FB_SoEReadAmplifierTemperature- toimilohkon kytkimeksi. | 33 |
| Kuva 16. Toimilohkon ulostulomuuttuja FB_SoEReadAmplifierTemperature- toimilohkolta tulevalle tiedolle. | 33 |
| Kuva 17. Toimilohkon kutsu toimilohkossa IF-lauseen sisällä. | 34 |
| Kuva 18. FB_Blink-toimilohkon kutsu ja käyttö IF-lauseessa..... | 35 |

| | |
|---|--|
| Kuva 19. Servovahvistimen lämpötilan mittaus ja korkeimman lämpötilan tallennus. | 35 |
| Kuva 20. Lämpötilamuuttujien nollaus. | 36 |
| Kuva 21. Toimilohko kutsuttuna FBD-ohjelmointikielellä..... | 36 |
| Kuva 22. Kuvien lisääminen projektiin. | 37 |
| Kuva 23. Visualisaation luominen. | 38 |
| Kuva 24. Visualisaation koko. | 39 |
| Kuva 25. Kuvan lisääminen visualisaatioon. | 40 |
| Kuva 26. Sommitelma valmiista huoltoilmoituksen lisäinformaatiosivusta. | 40 |
| Kuva 27. Valmis huoltoilmoituksen lisäinformaatiosivu kopioitavaksi. | 41 |
| | |
| Kuvio 1. Teollinen internet ja kuluttajien internet (mukaillen Collin & Saarelainen 2016, s. 31)..... | 20 |
| Kuvio 2. Palvelumallien kolme pääluokkaa. | 21 |
| Kuvio 3. Pilvirakenteet. | 23 |
| | |
| Taulukko 1. Taulukon otsikko (viite)..... | Virhe. Kirjanmerkkiä ei ole määritetty. |

Käytetyt termit ja lyhenteet

| | |
|----------------------------|--|
| IP-osoite | Internet Protocol address eli internetin protokollaosoite on laitteen yksilöllinen osoite, jolla laite tunnistetaan paikallisesta verkosta tai internetistä. |
| Kytkin | Kytkin on tietotekniikassa laite, jolla liitetään tietokoneet yhteen. Kytkimen tehtävä on vastaanottaa ja lähettää tietoa vastaanottille laitteille. Kytkin tunnistaa ja ohjaa kulkevan tietoliikenteen MAC-osoitteen avulla oikeaan porttiin. |
| Käyttöliittymä | Käyttöliittymä on käyttöjärjestelmän, ohjelman tai laitteen osa, josta voidaan lukea sekä syöttää tietoa. |
| MAC-osoite | Verkkoportin yksilöivä osoite. Käytetään myös nimillä kiinteä osoite, rautaosoite tai Ethernet-osoite. |
| Protokolla | Tietoliikenteessä protokolla on käytäntö tai standardi. Protokolla mahdollistaa tai määrittelee yhteydet ja tiedonsiirron laitteiden tai ohjelmien välille. |
| Reititin | Reititin on laite, jolla yhdistetään kaksi verkkoa toisiinsa, jolloin on mahdollista siirtää tietoa näiden kahden verkon välillä. Reititin käyttää laitteiden tunnistukseen IP-osoitetta. |
| Tietoliikenneverkko | Tietoliikenneverkko tarkoittaa verkon laitteita, jotka ovat viestintäyhteyksissä toisiinsa. Kutsutaan tietotekniikassa yleisesti myös nimillä verkko tai tietoverkko. |

1 JOHDANTO

1.1 Työn tausta

Työ käsittelee Etelä-Pohjanmaalla Laihialla sijaitsevan yrityksen T-Drillin valmistamien putkentyöstökoneiden automaation kehittämistä. Uusien koneiden kehittäminen on vienyt paljon aikaa ja kaikkiin koneisiin on tehty omat ohjausohjelmistot erikseen. Yrityksessä ei ole valmiita yleispäteviä ohjelmistomoduuleita putkentyöstökoneille. Yrityksen ongelmana on ollut valmiiden aliohjelmien puuttuminen. Aliohjelmat nopeuttaisivat ja helpottaisivat putkentyöstökoneiden ohjelmointia. Koneista saatua dataa lähetetään pilvipalveluun, josta voidaan seurata kaikkien koneiden tilatietoja. Tämän opinnäytetyön aihealue käsittelee koneista dataa keräävän ohjelmiston kehittämistä. Huoltoilmoituksissa on ollut ongelmana myös vähäinen tieto huollon tarpeista ja huollon ohjeistuksesta, joten aihealue käsittelee myös huoltoilmoitusten ja huolto-ohjeistusten modernisointia.

1.2 Työn tavoite

Työn tavoitteena oli kehittää yritykselle tiedonkeruuohjelmistoa. Tiedonkeruuohjelmisto kerää tarvittavaa dataa ja tilatietoja yrityksen putkentyöstökoneista. Tavoitteeksi ohjelmiston kehitykselle asetettiin yhteensopivuus kaikkiin yrityksen putkentyöstökoneisiin. Lisäksi tavoitteena oli parantaa putkentyöstökoneiden kosketusnäyttöpaneelille tulevien huoltoilmoitusten ohjeistuksia.

1.3 Työn rakenne

Luvussa yksi on johdanto. Johdannossa ensimmäisessä luvussa kerrotaan opinnäytetyön tausta, jonka jälkeen käydään läpi työn alussa asetetut tavoitteet, työn rakenne ja toimeksiantajayritys. Luvussa kaksi käydään läpi työssä tarvittuja laitteistoja. Luvussa kolme kerrotaan ympäristöstä eli sovelluksesta, ohjelmoinnista ja niiden standardeista, tarvittavista yhteyksistä sekä protokollista. Luvussa neljä käydään läpi pilvipalveluita. Luvussa viisi kerrotaan yleisesti servotekniikasta. Kuudennessa luvussa esitellään työn suunnitelma. Seitsemännessä luvussa on työn toteutus ja viimeisissä luvuissa käydään läpi tulokset, pohdinta ja yhteenveto.

1.4 Yritysesittely

Opinnäytetyön toimeksiantajana toimii T-Drill Oy. T-Drill Oy on suomalainen yritys, joka on perustettu vuonna 1980 Laihialla (T-Drill, i.a.). T-Drill Oy kuuluu osaksi Leinolat Groupia. Yrityksen päätoimiala on putkentyöstökoneet. Yritys valmistaa, suunnittelee ja myy erilaisia koneita ja laitteita metallisten putkien työstämiseen, esimerkiksi haaroitukseen, katkaisuun, laipoitukseen ja muotoiluun. Putkentyöstökoneita yritys valmistaa erikokoisia aina käsikäyttöisistä parin kilon koneista isoihin ja painaviin teollisuuskoneisiin. Yrityksellä on kolme toimipistettä, joista kaksi sijaitsee Suomessa ja yksi Yhdysvaltojen Atlantassa. Suomen toimipisteet sijaitsevat Laihialla sekä Isossakyrössä. Markkinat ulottuvat suurimmaksi osaksi ulkomaille ja koneita on myyty yli 70 maahan. Asiakastiedon (2020) mukaan työntekijöitä yrityksellä vuoden 2020 taloustietojen mukaan on 67 henkilöä ja liikevaihtoa on 9,8 miljoonaa euroa.

2 LAITTEISTO

2.1 PLC

Programmable Logic Controller on ohjelmoitava logiikka, lyhennettynä sitä kutsutaan myös nimellä PLC (Singh & Deswal, 2015, s. 9). PLC on kuin tietokone, jota käytetään ohjaamaan teollisuudessa käytettäviä koneita. Koneita, joita PLC:llä voidaan ohjata, on erilaisia, kuten kuljettimet, elintarvikkeiden jalostuskoneet ja kokoonpanolinjat. PLC auttaa yksinkertaistamaan vanhoja teollisuudessa käytettyjä ohjausjärjestelmiä. Vanhat teollisuuden ohjausjärjestelmät on rakennettu johdottamalla kaikki järjestelmässä olevat laitteet yhteen ja käskysignaalit luodaan analogisesti, näin kaikki vanhat ohjausjärjestelmät on räätälöity toimimaan vain kyseisessä järjestelmässä. Vanhoissa ohjausjärjestelmissä johdotetaan kaikki laitteet keskenään yhteen, PLC:tä käytettäessä kuitenkin johdotetaan laitteet vain PLC:lle. Näin laitteita voidaan ohjata PLC:lle ladattavalla ohjelmalla.

Rakenteeltaan PLC koostuu suorittimesta, virtalähteestä, ohjelmointilaitteesta, muistista, tulo- ja lähtömoduuleista ja viestintärajapinnoista (Bolton, 2006. s. 5). Suoritin on laite, joka sisältää mikroprosessorin, jonka avulla suoritin ohjaa lähtösignaaleja käyttämällä tulosignaaleja muistiin tallennetulla ohjelmalla. Virtalähde muuntaa vaihtovirran tasavirraksi, jota koko PLC käyttää. Ohjelmointilaitteella ohjelmoidaan tarvittava ohjelma ja siirretään kyseinen ohjelma PLC:n muistiin, nykyisin ohjelmointilaitteina toimivat tavalliset tietokoneet, joihin on asennettu ohjelmointisovellus, mutta ohjelmointilaitte voi olla myös kädessä pidettävä laite tai erillinen työpöytäkonsoli. Muisti on paikka, johon tallennetaan ohjelma, joka sisältää mikroprosessorin suorittamat ohjaustoiminnot, ja johon tallennetaan tulosten tiedot käsittelyä ja tulosignaaleja varten. Tulo- ja lähtömoduulit vastaanottavat tulosignaaleja ulkoiselta laitteelta ja lähettävät ne käsiteltäviksi suorittimelle ja suoritin lähettää takaisin lähtösignaaleja moduulille ja sitä kautta taas ulkoisille laitteille. Viestintärajapintaa käytetään tiedon vastaanottamiseen ja lähettämiseen verkkoon muille laitteille.

2.1.1 Soft PLC

Soft PLC on teknologia, joka käyttää tietokonetta ohjelmoitavana logiikkana ohjelmiston avulla (Liang, 2011, s. 1). Tällä ohjelmistolla saadaan tietokoneeseen perinteisen PLC:n

toimintojen lisäksi tietokoneen omat edut. Soft PLC jaetaan kahteen osaan, ohjelmointipuoleen ja suorittavaan puoleen (suorittava puoli on kuvassa 1). Suorittava puoli on soft PLC:n selkäranka, sillä se suorittaa ohjelman samalla tavalla kuin perinteinen PLC, mutta käyttäen tietokoneen laitteistoa.

Ohjelmointipuolella puolestaan voidaan ohjelmoida ohjelma suorittavalle puolelle, ja suorittavaa puolta voidaan simuloida tietokoneella. Tämä tarkoittaa sitä, että tietokoneella voidaan ohjelmoida sekä ajaa ohjelmaa ilman fyysistä suorittavaa teollisuus-PC:tä. Näin ohjelmointi nopeutuu ja helpottuu kun ohjelmaa ei tarvitse aina siirtää jokaisen testauksen välissä teollisuus-PC:lle. Myös turvallisuus parantuu, kun teollisuus-PC:lle kytketyt laitteet eivät pääse aiheuttamaan vahinkoa, mikäli ohjelmasta on löydetty kriittinen vika tietokoneella ajettaessa ja simuloidessa.

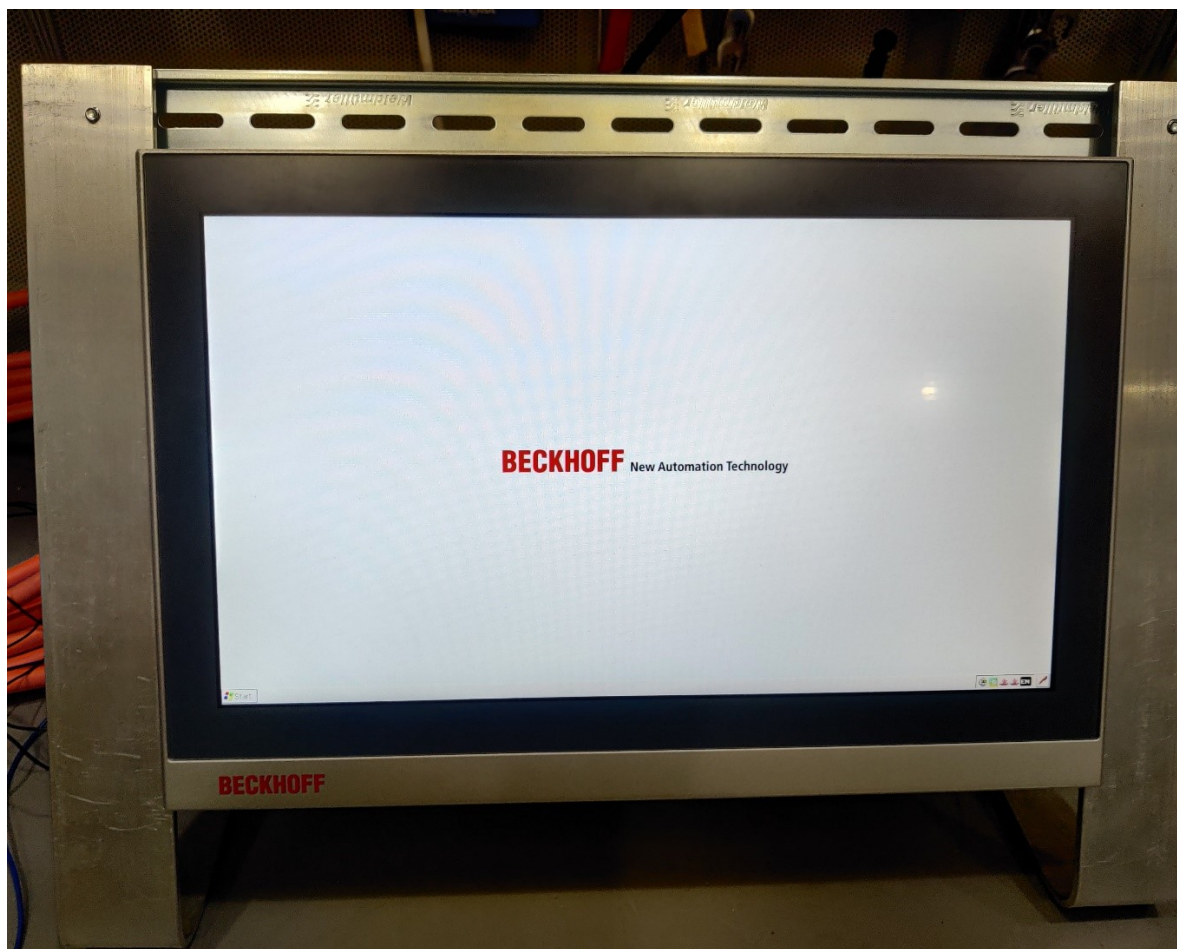


Kuva 1. Beckhoff CX5130 -teollisuus-PC

2.2 HMI-paneeli

HMI on lyhenne sanoista Human Machine Interface ja tarkoittaa suomeksi käyttöliittymää. HMI on laitteisto tai ohjelmisto, jonka kautta käyttäjä on vuorovaikutuksessa laitteen tai sovelluksen ohjaukseen (Falco ym, i.a., s. 11). HMI voi vaihdella fyysisestä ohjaimesta, jossa on painonappeja ja merkkivaloja, aina värinäytöllisiin teollisuuskoneisiin, joissa on erillinen HMI-ohjelmisto.

HMI-paneeli on teollisuudessa paljon käytetty käyttöliittymäratkaisu PLC:lle. Nykyisin paneelit ovat lähestulkoon kaikki kosketusnäyttöpaneeleita. Paneeli on itsessään käyttöliittymä ihmisen ja koneen väliseen kommunikointiin, mutta paneeli ei tyypillisesti sisällä valmista käyttöliittymäohjelmaa, vaan sille täytyy ohjelmoida erillinen käyttöliittymäohjelma. On myös olemassa tietokonepohjaisia paneeleita, joissa on käyttöliittymä, mutta näihinkin paneeleihin täytyy ohjelmoida erillinen käyttöliittymäohjelma. HMI-paneeleita on olemassa eri kokoisia ja eri ominaisuuksilla varusteltuja.



Kuva 2. Beckhoff CP2916 -HMI-paneeli

3 YMPÄRISTÖ

3.1 Ohjelmoitavan logiikan ohjelmointialustat

PLC:n ohjelmointi tapahtuu yleensä tietokoneella erillistä sovellusta käyttäen. Tällaisia sovelluksia ovat esimerkiksi Beckhoffin TwinCAT, Siemensin TIA Portal sekä Schneiderin EcoStruxure. On myös olemassa käsiohjaimia, joissa on sisäänrakennettu käyttöliittymä ja muisti, joilla voidaan ohjelmoida tiettyjä ohjelmoitavia logiikoita. Tässä työssä käytetään kuitenkin TwinCAT-sovellusta. Lisäksi kaikki edellä mainitut sovellukset on luotu käyttämään IEC 61131-3-standardia, joten järjestelmät toimivat samalla toimintaperiaatteella. Eroavaisuudet löytyvät suurimmalta osalta kehitysympäristön käyttöön liittyvistä toiminnoista, kuten asetuksista ja valikoista.

3.2 TwinCAT

Beckhoff on kehittänyt oman PC-pohjaisen ohjaustekniikan vuonna 1986 (Beckhoff, i.a. -a). Ohjelmiston puolella TwinCAT-kehitysympäristö muodostaa ohjausjärjestelmän ytimen. TwinCAT on lyhenne sanoista "The Windows Control and Automation Technology" ja tarkoittaa suomennettuna Windows-pohjaista ohjaus- ja automaatiotekniikkaa. TwinCAT-ohjelma tekee PC-pohjaisesta järjestelmästä reaaliaikaisen ohjausjärjestelmän, jolla voidaan toteuttaa useita erilaisia ohjausjärjestelmiä, kuten esimerkiksi ohjelmoitavia logiikoita, servoakseleita tai robotiikan ajonaikaisen järjestelmän. TwinCAT on sovellussarja, jonka uusin sovellusversio kantaa nimeä TwinCAT 3.

TwinCAT 3 koostuu kehitysympäristöstä (TwinCAT3 Engineering, XAE) ja ajonaikaisesti suorittavasta puolesta (TwinCAT 3 Runtime, XAR) (Beckhoff, i.a. -b). TwinCAT 3 XAE sisältää kaikki komponentit ja tuotteet, joilla voidaan konfiguroida, ohjelmoida, simuloida, analysoida ja korjata virheitä käyttöliittymästä tai ohjelmasta (Beckhoff, i.a. -c).

XAE on soft PLC -teknologiassa ohjelmointipuoli, kun taas XAR on suorituspuoli, joka on suunniteltu PLC-käyttöön tietokoneelle. XAE asennetaan yleensä koneelle, jolla suoritetaan ohjelmointi, kun taas XAR asennetaan ohjaavalle koneelle. Tästä huolimatta molemmat voidaan asentaa myös samalle koneelle (Beckhoff, i.a. -b). Esimerkiksi

ohjelmointikoneella voidaan testata ohjelmaa samanaikaisesti suorituspuolella, ilman fyysistä ohjaavaa konetta.

TwinCAT 3:n integroiminen osaksi Visual Studio -ympäristöä mahdollistaa ohjelmistokomponenttien ohjelmoinnin IEC61131-3 -ja C- sekä C++-ohjelmointikielillä (Beckhoff, i.a. -a). Ohjelmistokomponentteja voidaan kutsua eri ohjelmointikielillä riippumatta siitä, millä ohjelmointikielillä komponentit on ohjelmoitu. Sovellukseen on sulautettu järjestelmän hallintatyökalu (TwinCAT System Management), jolla voidaan konfiguroida, syöttää parametrejä, ohjelmoida ja analysoida automaatiolaitteistoja.

Visual Studio voidaan integroida kahdella tavalla (Beckhoff, i.a. -a): normaalilla (TwinCAT Standard) tavalla, joka käyttää Visual Studion perusrakennetta, ja integroidulla (TwinCAT Integrated) tavalla, jossa TwinCAT integroi koko sovelluksensa Visual Studion kanssa. Integroidulla tavalla saadaan käyttöön C-, C++-, C#- ja VB.NET-ohjelmointikielet sekä MATLAB- ja Simulink-ohjelmat.

3.3 ADS

ADS on Beckhoffin kehittämä protokolla. ADS-lyhenne tulee sanoista "The Automation Device Specification", se on TwinCATin kommunikointiprotokolla (Beckhoff, i.a. -d). Tämä protokolla mahdollistaa tiedonkulun ja toimintojen ohjauksen. ADS kommunikoi verkon tai sarjaportin avulla ja se on täysin mediasta riippumaton protokolla.

TwinCAT-sovelluksen rakenne mahdollistaa erillisten ohjelmamoduulien toiminnan itsenäisinä laitteina, jokaiselle tehtävälle on oma ohjelmamoduuli, joka käyttäytyy joko palvelimena tai asiakkaana (Beckhoff, i.a. -e). Ohjelmamoduuleita ovat esimerkiksi TwinCAT-CNC, -PLC tai -NC. Näiden ohjelmamoduulien välillä kulkeva viestintä kulkee ADS-rajapinnan kautta. Palvelimien välinen rajapinta hallinnoi ja jakaa kaikkia viestejä järjestelmässä ja TCP/IP-tietoliikenneprotokollaa käyttävissä yhteyksissä. ADS-laitteita ovat kaikki laitteet, joissa on ADS-rajapinta, ja jotka tarjoavat palvelimen palveluja.

Palvelin toimii tietotekniikassa tarjoamalla palveluita asiakkaalle, englannin kielellä palvelinta kutsutaan nimellä server, suomessa palvelinta kutsutaan myös yleisellä tasolla

serveriksi. Asiakasta kutsutaan englannin nimellä client, suomeksi käännettynä tämä tarkoittaa asiakasta tai asiakasohjelmaa.

TCP/IP tulee sanoista Transmission Control Protocol / Internet Protocol (Fisher, 2019). TCP/IP on sarja standardoituja sääntöjä, jotka sallivat tietokoneiden kommunikoinnin verkossa kuten internetissä. TCP ja IP ovat kaksi erillistä tietokoneverkkoprotokollaa. IP määrittää, mihin tieto lähetetään, ja TCP vastaa tiedon lähettamisestä, kun IP-osoite on löydetty.

3.4 IEC 61131-3

IEC 61131-3 on kolmas osa IEC-61131-standardista (International Electrotechnical Commission, i.a.). Tämä osa standardia määrää ohjelmitavien ohjaimien yhtenäisen ohjelmointikielisarjan syntaksin ja semantiikan. Sarjaan kuuluu kaksi tekstipohjaista kieltä Instruction List (IL) ja Structured Text (ST), ja kaksi graafista kieltä Ladder Diagram (LD) sekä Function Block Diagram (FBD).

3.4.1 FBD

FBD tulee sanoista Function Block Diagram eli toimilohkokaavio. FBD käyttää virtapiirejä (Beckhoff, i.a. -f). Jokainen virtapiiri sisältää rakenteen, joka voi sisältää loogisia ja aritmeettisiä lausekkeita, toimilohkokutsuja, hyppyjä tai return-lauseita.

Toimilohkot toimivat samalla periaatteella kuin oliot olio-ohjelmoinnissa, eli ne sisältävät ohjelman, jota voidaan kutsua toisissa ohjelmissa. Toimilohkot ja muuttujat yhdistetään viivoilla (Beckhoff, i.a. -f). Verkkoa eli yksittäistä toimilohkoriviä luetaan vasemmalta oikealle ja verkkolistaa eli editoria luetaan ylhäältä alas alkaen ensimmäisestä verkosta.

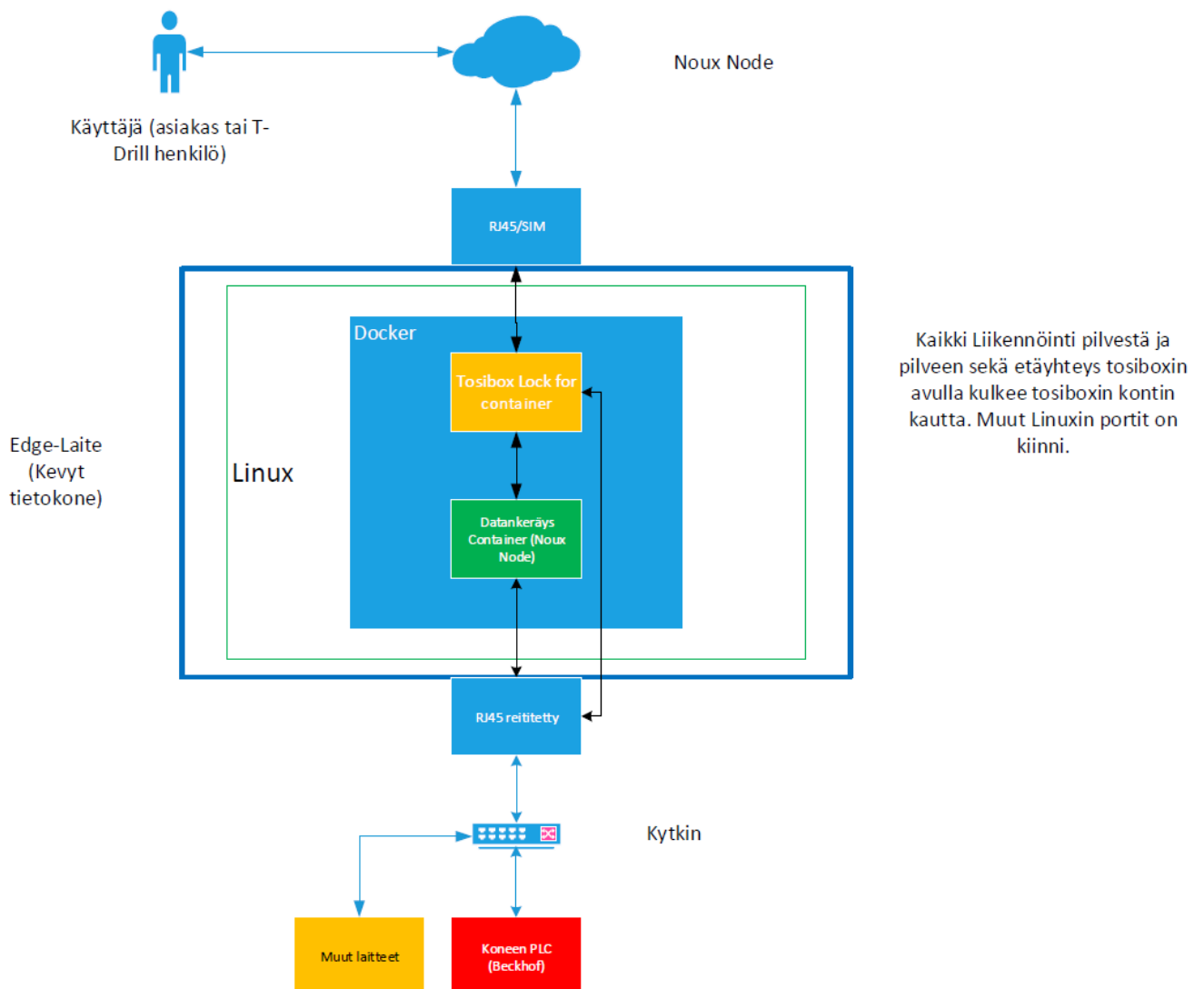
3.4.2 ST

ST tulee sanoista Structured Text eli rakenteinen teksti. ST on verrattavissa toisiin korkean tason ohjelmointikieliin, esimerkiksi C- tai PASCAL-ohjelmointikieliin, joissa voidaan kehittää monimutkaisia algoritmeja (Beckhoff, i.a. -g). Koodi rakentuu ohjeista ja

lausekeyhdistelmistä, jotka voivat olla ehtolausekkeita, esimerkiksi IF, THEN ja ELSE tai silmukoita, kuten WHILE ja DO.

Lauseke on rakenne, joka palauttaa lauseen suorittamisen lopussa saadun arvon, lauseke voi olla myös yhdistelmä operaattoreita ja operandeja (Beckhoff, i.a. -g). Lausekkeina voidaan käyttää myös tehtäviä. Operandi voi olla vakio, muuttuja, toimilohkokutsu tai jokin muu lauseke.

3.5 Yhteydet



Kuva 3. Putkentyöstökoneiden yhteyksien infrastruktuuri

Kuvassa 3 nähdään putkentyöstökoneiden verkkoyhteyksien infrastruktuuri. Beckhoffin ohjain sekä muut laitteet on kytketty verkkokaapeilla kytkimelle ja sitä kautta edge-laitteelle ja edge-laitteelta Noux Noden pilveen. Edge-laitteen sisällä on NouxCloud edge -sovellus ja Tosibox Lock for container -sovellus omissa konteissansa. Kaikki tiedonkulku pilven ja edge-laitteen välillä kulkee Tosibox Lock -kontin kautta. Beckhoff PLC:n ja NouxCloud edgen välinen kommunikointi tapahtuu ADS-protokollan kautta. NouxCloud egde on Noux Node Oy:n kehittämä IoT-alusta.

Edge-laitteeksi määritellään mikä tahansa laitteisto, joka ohjaa tiedonkulkua kahden verkon välillä. Tässä tapauksessa edge-laitteena toimii kyseiseen tehtävään suunniteltu pieni tietokone Linux-pohjaisella käyttöjärjestelmällä, jonka sisälle on asennettu Docker-virtuaalisointialusta, jossa voidaan käyttää kontteja.

Kontti on standardisoitu ympäristö, jossa voidaan ajaa ohjelmaa (Wallenius, 2022). Kun ohjelma on kontin sisällä, voidaan sitä siirrellä helposti eri ympäristöjen välillä, esimerkiksi ohjelmointikoneen, testiympäristön, tuotannon ja pilven välillä. Ohjelmien siirtäminen konttien sisällä on helpompaa, koska normaalisti ohjelmien koodissa on paljon koodin toimivuuden kannalta pakollisia lisäosia, kirjastoja ja asetuksia, jotka vaikeuttavat ohjelman siirtoa laitteelta laitteelle, näin konttien standardisoidulla pakkaustavalla voidaan ohjelma pakata muotoon, jolla sitä voidaan siirtää ja ajaa ilman ohjelmiston muokkausta.

Tosibox Lock on Tosibox Oy:n kehittämä älykäs etäkäyttö- ja verkkolaite, joka toimii suojattujen etäyhteyksien päätepisteenä (Tosibox, i.a.). Lukkoon yhdistetyt laitteet ovat käytävissä verkossa VPN-yhteyden kautta.

Virtual Private Network (VPN) eli virtuaalinen erillisverkko suojaa internetyhteyden ja yksityisyyden verkossa luomalla yksityisen verkon julkisesta verkosta (Norton, 2022). Tätä yksityistä verkkoa kutsutaan VPN-yhteydessä tunneliksi. Tunneliin ulkopuoliset laitteet eivät pääse lukemaan virtuaalisesti IP-osoitteita tai muitakaan tietoja, mitä laitteiden välillä liikkuu. Tämä tekee VPN-yhteydestä erittäin turvallisen.

4 PILVIPALVELUT

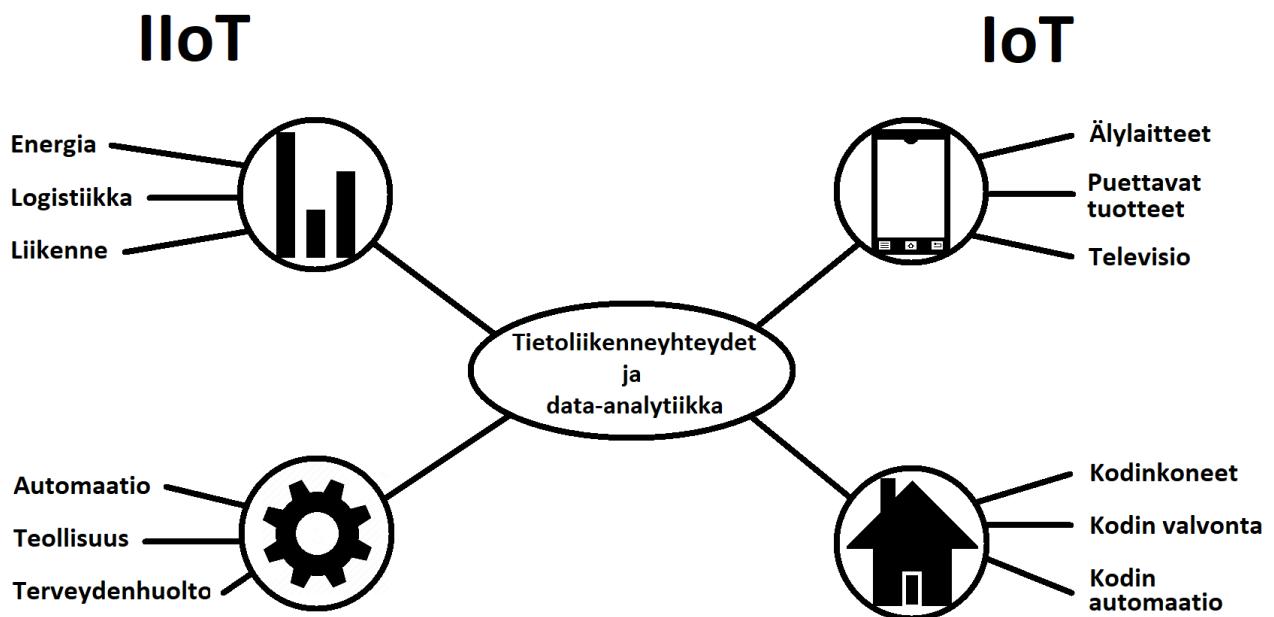
4.1 Termistö

Pilvipalvelu koostuu palvelimista, jotka sijaitsevat fyysisesti usein isoissa datakeskuksissa (Ranger, 2022). Palvelun tarjoaja tarjoaa asiakkaalle käytettäväksi palvelimien resursseja internetin välityksellä, näin asiakkaan ei tarvitse omistaa omia fyysisiä palvelimia. Palvelimilta voidaan tarjota asiakkaalle esimerkiksi tallennustilaa, datan analytiikkaa, pilvialustoja ja -palveluita sekä laskentatehoa pilvilaskentana. Usein sekoitetaan pilvialustat ja pilvipalvelut keskenään, koska niitä on vaikea erottaa toisistaan. Pilvipalvelulla tarkoitetaan valmista palvelua, jota voidaan tarjota suoraan asiakkaalle. Pilvialusta tarkoittaa siis palvelua, jossa asiakas voi itse rakentaa haluamansa palvelun tai sovelluksen alustan päälle. Pilvialustan käyttäminen vaatii asiakkaalta enemmän perehtyneisyyttä ja monesti myös ohjelmointitaitoja. Hyvänä esimerkkinä tavallisesta pilvipalvelusta ovat esimerkiksi Googlen Gmail ja Microsoftin Outlook sähköpostipalvelut, kyseisissä sähköposteissa käyttäjän sähköposteja säilytetään kyseisten yritysten tarjoamilla palvelimilla eikä sähköpostia käyttävän tarvitse ladata sähköposteja tietokoneen omaan muistiin, vaan sähköposteja voidaan tarkastella internetin kautta.

Internet of things eli esineiden internet on termi, jota käytetään konseptista, jossa yhdistetään kaikki laitteet internetiin (Clark, 2016). IoT on käytännössä valtava verkosto toisiinsa yhdistetyistä laitteistoista, jotka voivat kommunikoida keskenään, ja tarjota käyttäjille hyödyllistä dataa. IoT-laite voi olla mikä tahansa laite, josta saa mitattua jonkinlaista tilatietoa internetiin, yksinkertaisimmillaan laite voi olla tavallinen katkaisin, josta saadaan luettua, onko katkaisin päällä vai ei. Vaikka IoT kattaa kaikki internetiin yhdistetyt laitteet, niin IoT jaetaan kuitenkin vielä kuluttajien internetiin ja teolliseen internetiin (kuvio 1). Kuluttajien internetiä sanotaan myös esineiden internetiksi. Collinsin ja Saarelaisen, A. (2016, s. 32) mukaan esineiden internet viittaa ensisijaisesti yksityisille kuluttajille suunnattuihin tuotteisiin.

Industrial Internet of Things (IIoT). Esineiden internet teollisuudessa eli teollinen internet on sama kuin esineiden internet, mutta teollisuudessa (Collin & Saarelainen, 2016, s. 31). Teollista internetiä voidaan käyttää teollisuudessa esimerkiksi laitteiden etäkäyttöön, valvontaan, hallintaan, päivityksiin, optimointiin, ennakoivaan kunnossapitoon sekä laitteiden

väliseen kommunikointiin. Lisäksi uusia käyttötarkoituksia ja sovelluksia kehitetään jatkuvasti. Teollisuudessa Esineiden internetiä voidaan käyttää hyväksi IoT-alustoilla, joihin tuodaan tarkasteluun haluttuja tilatietoja laitteista. Alustat eivät rajoitu pelkästään tarkasteluun, vaan alustoilla voidaan myös ohjata alustalle linkitettyjä laitteita. IoT-alustoja on lukemattomia erilaisiin käyttötarkoituksiin. Esimerkiksi IoT-alustalle voidaan tuoda tietoja elokuvateattereiden varausjärjestelmästä, jolloin tietoja käsittelemällä saadaan selville elokuvateatterin halutuimmat istumapaikat.

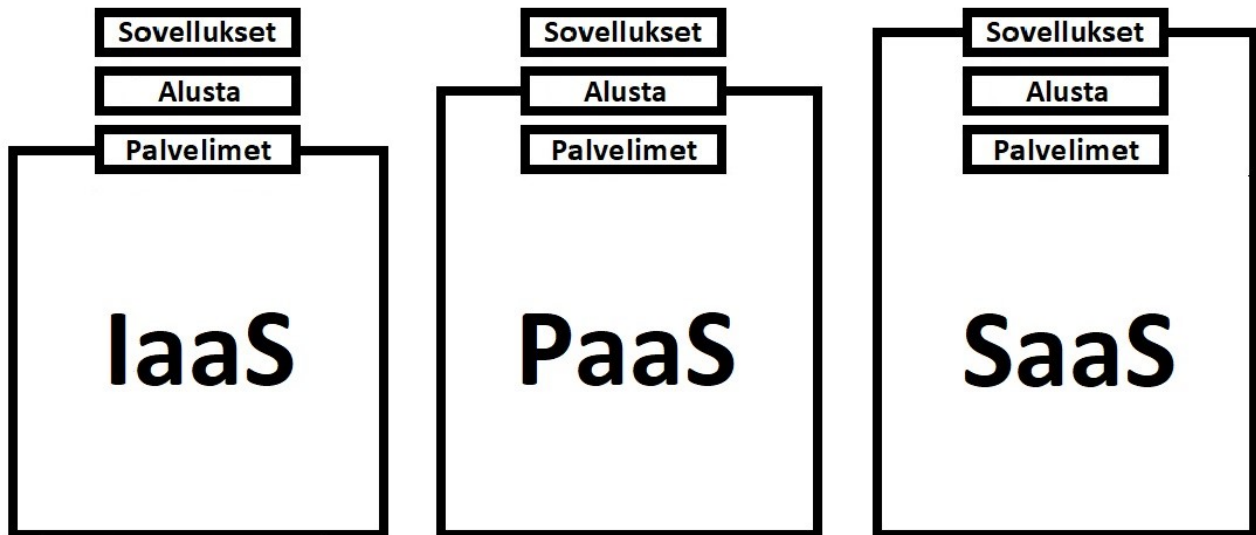


Kuvio 1. Teollinen internet ja kuluttajien internet (mukaillen Collin & Saarelainen 2016, s. 31).

Cyber Physical System (CPS). CPS on kyberfyysinen järjestelmä, joka on tullut käsitteeksi Yhdysvalloissa noin kymmenisen vuotta sitten (Collin & Saarelainen, 2016, s. 33). CPS käsite on lähestulkoon sama kuin teollinen internet. CPS-käsitteen määrittelyn alkuna on uuden sukupolven sulautetut järjestelmät, jotka käyttävät uutta IoT-teknologiaa. Tämä teknologia mahdollistaa liittymisen järjestelmiin, jotka muodostuvat viestivistä laitteista. Nämä laitteet ovat itsenäisesti, mutta verkossa toistensa kanssa sekä ihmisen kanssa viestiviä laitteita. Käsitteen avulla tavallisesti esitetään todella kattava järjestelmätaso. CPS syntyy, kun viestintäinfrastruktuurin kautta yhdistetään älylaitteet, ympäristöt ja ihmiset sekä sulautetut järjestelmät ja laitteet. Esimerkkeinä kyseisistä järjestelmistä ovat älytehdas,

sähköverkot, metropolit, älykodit, nykyaikaiset autot. Periaatteessa CPS kattaa saman kuin internetiin kytketyt älytuotteet ja palvelut sekä yleisesti esineiden internetin.

4.2 Palvelumallit



Kuvio 2. Palvelumallien kolme pääluokkaa.

Palvelumallien kolme pääluokkaa ovat IaaS, PaaS ja SaaS. Palvelumalleilla on isoja eroavaisuuksia esimerkiksi käytettävyydessä ja joustavuudessa. Alla käydään läpi palvelumallien kolme pääluokkaa aloittaen laajimmasta ja työllistävimmästä palvelumallista helpokäyttöisimpään. Viimeisenä alla käydään läpi palveliton laskenta, jota käytetään pilvilaskennassa.

Infrastructure as a Service (IaaS) (Wallenius, 2019). Tässä palvelumallissa palvelua käyttävä kehittäjä vastaa kaikesta muusta paitsi koneista, fyysisistä palvelimista ja virtualisointikerroksesta. Kehittäjä rakentaa itse kaiken käyttöjärjestelmästä lähtien, joten tämä palvelumalli on kaikista työllistävin, mutta samalla kaikista joustavin.

Platform as a service (PaaS) (Wallenius, 2019). Tässä palvelumallissa palveluntarjoaja hoitaa palvelun infrastruktuurin, virtuaalipalvelimet ja virtuaalipalvelimella olevasta alustasta. Tämä alusta voi olla jonkinlainen tietokanta tai sovelluskehitysympäristö. Palvelua

käyttävä kehittäjä vastaa, miten alustaa voidaan hyödyntää omiin tarpeisiin. Palvelumalli ei ole valmis paketti ja se vaatii kehittäjältä tietoa ja taitoa palvelun rakentamisen alustan päälle.

Software as a Service (SaaS) (Wallenius, 2019). Tässä palvelumallissa palveluntarjoaja hoitaa koko infrastruktuurin, palvelimet ja ohjelmistot, joten palvelua käyttävälle jää huolehtiminen ainoastaan järjestelmään syötettävästä datasta sekä käyttäjähallinnasta. Tämä palvelu on kaikista palveluista helpoin, mutta koska palvelua ei voi muokata, ei palvelu tarjoa hirveästi joustovaraa.

Serverless computing eli palveliton laskenta mahdollistaa sovellusten kehittämisen nopeasti johtuen siitä, ettei tässä tarvitse hallita palvelimen rakennetta (Microsoft, i.a. -c). Palvelittomilla sovelluksilla pilvipalvelun tarjoaja automaattisesti luo, skaalaa ja hallinnoi koodin suorittamiseen tarvittavaa infrastruktuuria. Tässä palvelumallissa on tärkeää ymmärtää, että tässä käytetään silti palvelimia. Nimi palveliton tulee siitä, että infrastruktuurin valmistelu ja hallintaan liittyvät tehtävät ovat näkymättömiä kehittäjille.

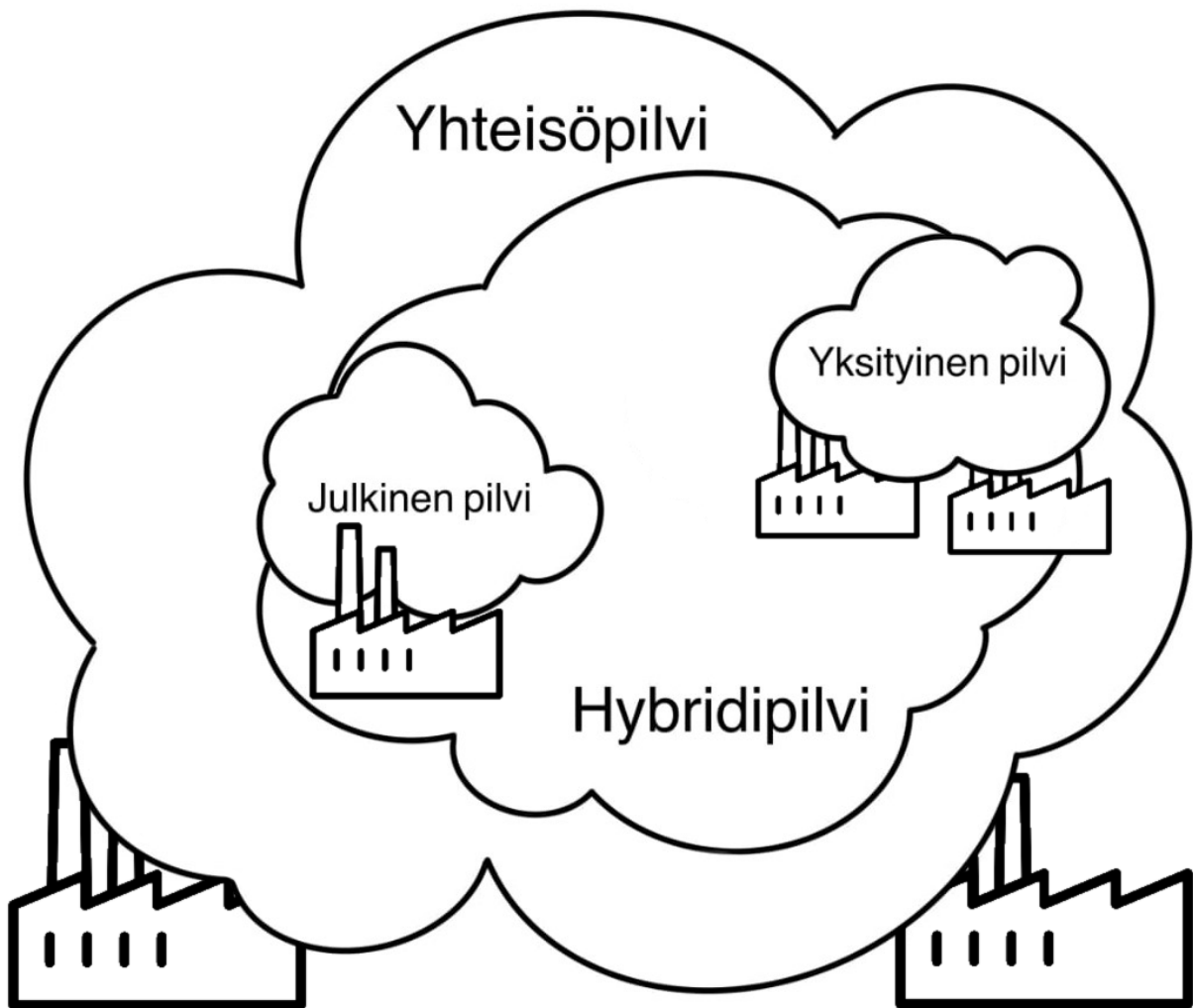
4.3 Toteutustavat

Julkinen pilvi on kolmannen osapuolen omistama ja ylläpitämä eli palvelualusta sijaitsee palveluntarjoajan omissa fyysisissä tiloissa (Microsoft, i.a. -a). Julkisen pilven omistaja omistaa kaikki laitteistot, ohjelmistot ja alustat sekä toimittaa laskentaresursseja internetin kautta kuten tallennustilaa ja palvelimia. Näihin palveluihin pääsee käsiksi verkkoselaimella palveluntarjoajan antamilla tunnuksilla.

Yksityinen pilvi on pilvi, jonka laskentaresursseja käyttää yksinomaan yksittäinen yritys tai organisaatio (Microsoft, i.a. -a). Yksityinen pilvi voi sijaita yrityksen tai organisaation fyysisessä datakeskuksessa. Yksityinen pilvi voi olla myös vuokrattava kolmannelta osapuolelta. Lyhyesti sanottuna yksityinen pilvi on pilvi, jossa palvelut ja infrastruktuuri ylläpidetään yksityisessä verkossa.

Hybridipilvi yhdistää julkiset ja yksityiset pilvet keskenään tekniikalla, joka mahdollistaa sovellusten, datan ja tietojen jakamisen näiden kahden pilven välillä (Microsoft, i.a. -b).

Yhteisöpilvi on pilvi, joka on tarkoitettu nimenomaan tietyn yhteisön käytettäväksi (Wallenius. 2019). Tällaisia yhteisöjä ovat esimerkiksi sellaiset, joiden asiakkaiden välillä on enemmän luottamusta verrattuna julkiseen pilveen, sillä julkisessa pilvessä kaikki voivat käyttää samaa ympäristöä.



Kuvio 3. Pilvirakenteet.

4.4 Pilvilaskenta ja sen hyödyt

Yksinkertaistettuna pilvilaskenta on tiedonkäsittelyjärjestelmä, johon sisältyvät palvelimet, varastotallennustilat, tietokannat, verkosto, ohjelmisto, analytiikka ja tekoäly. Nämä toimivat internetin kautta tarjotakseen nopeampia innovaatioita, joustavia resursseja ja

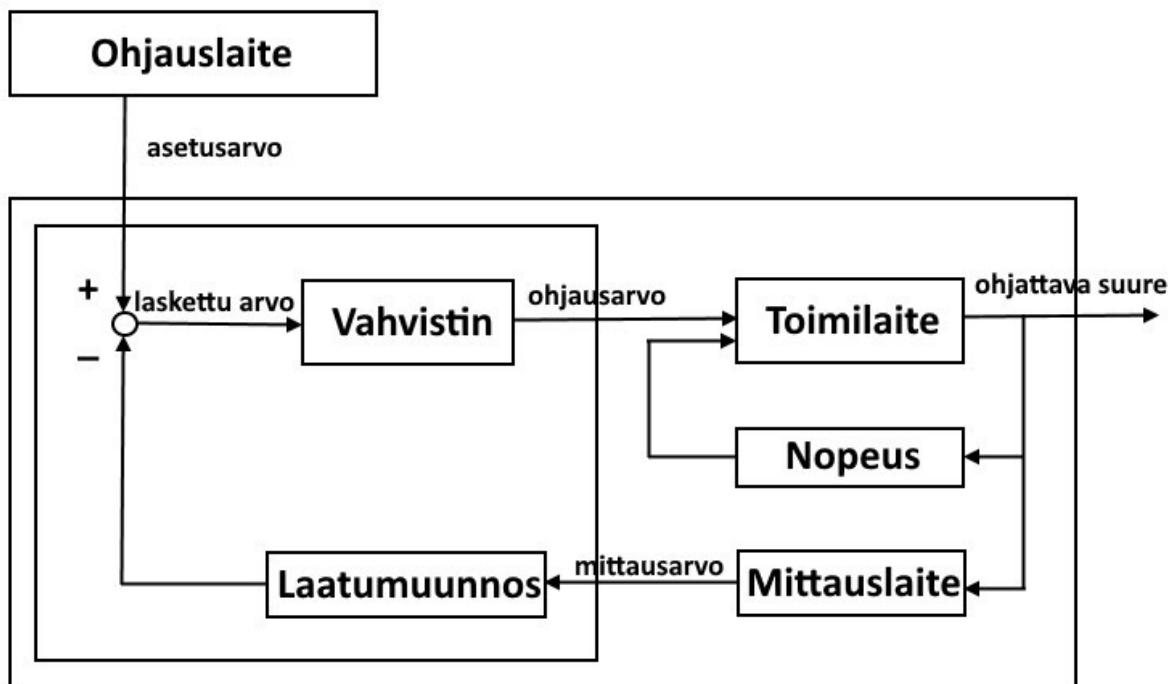
kustannustehokkuutta suurille datamäärille (Microsoft, i.a. -a). Pilvilaskennassa tyypillisesti maksetaan siitä resurssimäärästä, mitä käytetään, tämä auttaa laskemaan käyttökustannuksia, ylläpitämään infrastruktuuria tehokkaammin ja kasvattamaan pilvilaskennan käyttöä samassa suhteessa käyttötarpeiden kasvuun nähden. Pilvilaskennassa on monia hyötyjä verrattuna perinteisiin IT-järjestelmiin, kuten hinta, nopeus, tuottavuus, luotettavuus, turvallisuus ja skaalautuvuus. Hinnassa pilvilaskenta päihittää pääomakustannukset, jotka tulevat laitteiston ja ohjelmiston ostamisesta, palvelinkeskuksien perustamisesta ja käyttämisestä, virrankulutuksesta sekä IT-asiantuntijan ylläpitotyöstä.

Nopeudessa pilvilaskennassa voidaan tarjota valtavat määrät laskentaresursseja muutamassa minuutissa, tämä on mahdollista, koska suurin osa pilvilaskennasta tarjotaan tilauspalveluina. Tuottavuus lisääntyy, kun pilvilaskenta syrjäyttää kiinteät palvelimet, ohjelmiston päivitykset, korjaukset ja kaikki muut aikaa vievät IT-hallintatyöt. Yrityksen IT-henkilöt voivat keskittyä tärkeämpiin liiketoiminnan tavoitteisiin. Pilvilaskenta tekee tietojen varmuuskopioinnista, järjestelmän palauttamisesta ja liiketoiminnan jatkuvuudesta helpompaa ja halvempaa, koska tiedot voidaan peilata moniin muihin ylimääräisiin sivustoihin pilvilaskennan tarjoajan verkossa. Monet pilvilaskennan tarjoajat tarjoavat laajan joukon käytäntöjä, teknologioita ja kontroleja, jotka vahvistavat yleistä tietoturvaa ja auttavat suojaamaan tietoja, sovelluksia ja infrastruktuuria mahdollisilta uhilta. Myös yksi pilvilaskennan keskeisimmistä eduista on joustava skaalautuvuus. Tämä tarkoittaa tarpeellisen määrän IT-resurssien toimittamista pilveen, esimerkiksi enemmän tai vähemmän laskentatehoa, tallennustilaa ja kaistanleveyttä tarpeen mukaan pilvilaskennan tarjoajan optimaalisimmasta maantieteellisestä sijainnista.

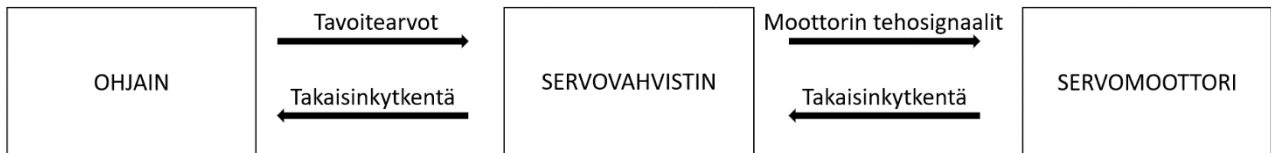
5 SERVOTEKNIikka

5.1 Servojärjestelmä

Fonselius ym. (1998, s. 7) mukaan servotekniikan tehtävä on ohjata kappaleen asema ja nopeus haluttuun arvoon. Servolaitteet jaetaan tyypillisesti kahteen luokkaan, moottoreihin ja sylintereihin. Suure, jota säädetään, ilmaisee, onko kyse asema-, nopeus-, voima- vai momenttiservosta. Servojärjestelmässä keskeinen asia on takaisinkytkentä, jolla mitataan suure alussa, tätä verrataan järjestelmälle syötettyyn arvoon. Tarkimmissa servojärjestelmissä takaisinkytkentöjä voi olla enemmänkin kuin yksi. Väärin säädetyt servojärjestelmät ovat alttiita värähtelylle. Nykyään servojärjestelmiä voidaan mallintaa ja tutkia tietokonesovelluksilla, mutta luotettavien tulosten saamiseksi on tunnettava oikeiden fyysisten järjestelmien ominaisuudet.



Kuva 4. Servojärjestelmän rakenne (mukaillen Fonselius ym, 1998, s. 8).



Kuva 5. Servomoottorin ohjaus.

Kuvassa 4 havainnollistetaan servojärjestelmän periaatteellista rakennetta. Ohjauslaitteelta annetaan asetusarvo erotuspisteelle, jossa verrataan takaisinkytkennältä saatua arvoa asetusarvoon (Fonselius ym, 1998, s. 8). Erotuspisteeltä lähtee laskettu arvo vahvistimelle, joka muuntaa lasketun arvon ohjausarvoksi, joka menee toimilaitteelle, jolloin toimilaitte liikkuu ohjausarvon mukaisesti.

Kuvassa 5 on yksinkertaistettu kaavio servojärjestelmästä servomoottorin ohjauksessa, jossa ohjain antaa tavoitearvot servovahvistimelle. Servovahvistin ohjaa servomoottoria, servomoottorilta tulee takaisinkytkentä servovahvistimelle ja servovahvistimelta menee takaisinkytkentä ohjaimelle.

Servovahvistimen tehtävä on noudattaa ohjaimelta saatuja käskyjä ja ohjata toimilaitetta vertailemalla näitä arvoja toimilaitteen takaisinkytkennältä saatuihin arvoihin (Omron. i.a. s. 3). Vertailun jälkeen vahvistin ohjaa toimilaitteen arvoja, kuten vääntöä, kierrosnopeutta tai paikkaa

Kaikista tyypillisimpiä servomoottoreita teollisuudessa ovat harjattomat moottorit (Omron. i.a. s. 3). Tällaisen moottorin roottorilla on tehokas kestmagneetti. Staattori on rakennettu useista keloista ja moottori saadaan pyörimään, kun keloihin syötetään virtaa määrätystä järjestyksessä. Roottorin liike määräytyy staattorin taajuudesta, vaihekulmasta, napaisuudesta ja virrasta, kun oikea määrä virtaa syötetään staattorin keloille oikeaan aikaan. Servomootorit erottuvat tavallisista moottoreista sillä, että niissä on enkooderi. Enkooderi mahdollistaa ohjauksen suurilla nopeuksilla ja tarkkuuksilla. Enkooderi on tärkeä laite, joka muodostaa servojärjestelmän ytimen, sillä se mahdollistaa takaisinkytkennän esimerkiksi vahvistimen ja moottorin välille. Monissa tapauksissa enkooderi on integroitu servomoottorin sisälle tai kiinnitetty servomoottorin perään. Joissakin järjestelmissä enkooderi voi olla erillisenä laitteena asennettuna kauemmaksi servomoottorista.

6 SUUNNITTELU

Työtä aloittaessa täytyi hankkia kattavasti tietoa yrityksen koneista, laitteistosta ja käytännöistä. Yritykseltä saatiin käyttöoppaat kolmelle eri putkentyöstökoneelle, joissa oli yhteensä noin kolmesataa sivua tietoa koneista. Alussa pidettiin myös palavereita, joissa keskusteltiin, minkälaista tietoa koneista haluttiin kerätä sekä huoltoon liittyvistä ilmoituksista. Käyttöoppaat antoivat hyvän pohjan ja paljon ideoita varsinkin huoltoilmoitusten modernisoinnille. Huoltoilmoitusten osalta suunnitelmaksi tuli tehdä jokaisesta huoltoilmoituksesta oma lisäinformaationsivu käyttöliittymälle pois lukien päivittäiset huollot, koska päivittäisissä huolloissa ei ollut koneiden toiminnan kannalta kriittisiä huoltoja. Päivittäisiä huoltoja oli esimerkiksi koneen ympäristön siisteys ja puhtaat pinnat. Huoltoilmoitusten lisäinformaationsivuilla olisi kuva huoltoa vaativasta koneen osasta ja sen sijainnista, sekä ohjeistus huollon tekemiseen ja siihen tarvittaviin työkaluihin ja tarvikkeisiin. Ideoita tuli myös itse huoltosivun modernisointiin, mutta nämä ideat priorisoitiin hautumaan työn loppuun, mikäli niiden toteuttamiselle jäisi aikaa.

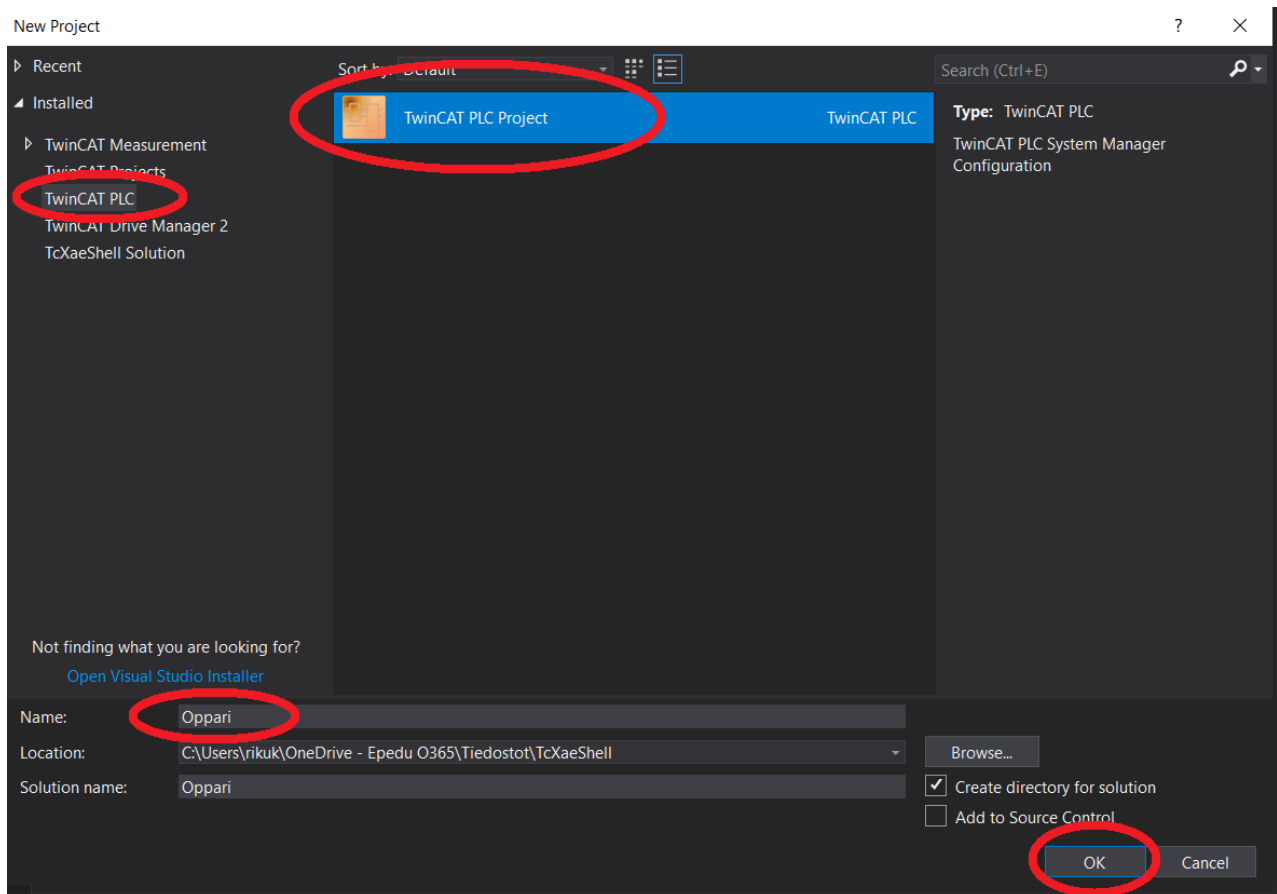
Tiedonkeruuhjelmistolle ei ollut varsinaista suunnitelmaa, vaan pikemminkin jatkuvaa tutkimista, mitä tietoja koneista halutaan ja miten ne voitaisiin saada näkyville mahdollisimman helposti. Näin ollen tiedonkeruuta tutkittiin ja testattiin monella eri tavalla. Työn edetessä alkoi hahmottua, miten tiedonkeruuhjelmisto toteutetaan. Loppujen lopuksi saatiin aikaan toimilohko, jonka ulostuloista saadaan koneiden hyödyllisiä tietoja muodoissa, joita voidaan lukea suoraan NouxCloud edge IoT -alustalla. Koska yritys käyttää Beckhoffin laitteistoja, kuten servomootoreita ja vahvistimia, tämä mahdollistaa TwinCATin kirjastoista löytyvien valmiiden mittauslohkojen käyttämisen. Beckhoffin nykyaikaisista servomootoreista ja vahvistimista on mahdollista lukea monenlaista tietoa, kuten esimerkiksi lämpötiloja, virtaa ja jännitettä. Kyseiset kirjastojen toimilohkot on suunniteltu suoraan tähän käytettäväksi, joten suunnitelmana on kutsua kirjastojen datanmittaustoimilohkoja itse tehdyn toimilohkon sisällä. Koska NouxCloud edgen ja Beckhoff PLC:n välinen kommunikointi tapahtuu ADS-protokollan kautta, NouxCloud edgeen voidaan lähettää normaaleja muuttujatyyppisiä, kuten BOOL tai LREAL. ItsetehdYN toimilohkon idea on sen helppo kopioitavuus muihin projekteihin käytettäväksi. Lisäksi toimilohkoon lisätään servoakselilta saatava referenssidata, josta saadaan luettua suoraan esimerkiksi akselin paikka, vääntö, nopeus ja tilatieto.

7 TOTEUTUS

7.1 Toimilohkon teko

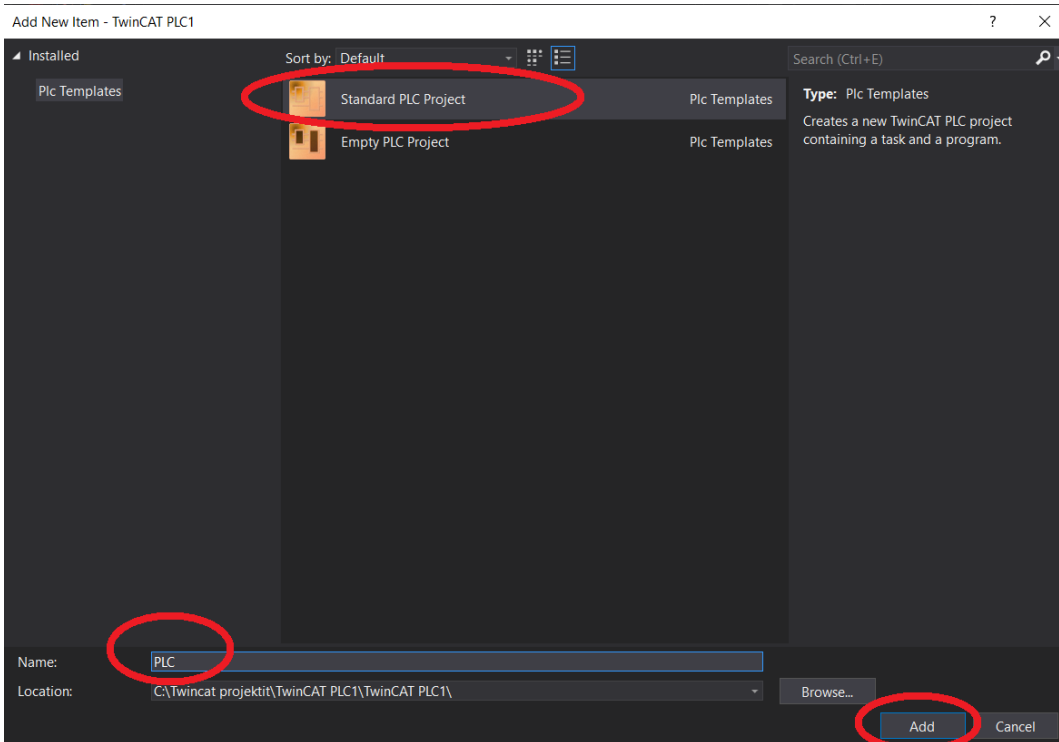
Toimilohkon teko aloitettiin tekemällä uusi TwinCAT-projekti. Projektiin lisättiin tarvittavat kirjastot. Tämän jälkeen luotiin toimilohko. Toimilohkoon lisättiin akselin referenssidatan tilatietoja ja kirjastojen toimilohkoja. Alla käydään läpi tarkemmin toimilohkon teon vaiheet.

Uusi TwinCAT-projekti luotiin valitsemalla vasemmalta ylhäältä File-valikosta New ja valitsemalla Project.... Tästä aukesi uusi ikkuna nimeltä New Project. Tästä ikkunasta valittiin TwinCAT PLC Project ja annettiin sille nimi, jonka jälkeen painettiin OK” (kuva 6).



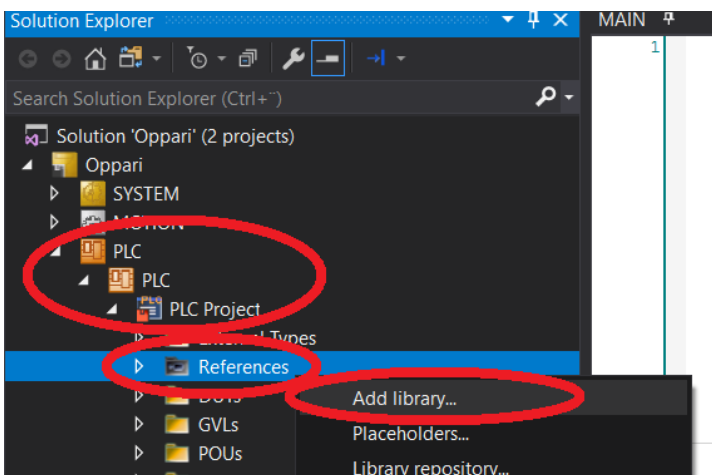
Kuva 6. Projektin valinta ja nimeäminen.

OK-painikkeen jälkeen aukesi uusi ikkuna nimeltä Add New Item. Ikkunasta valittiin Standard PLC Project ja annettiin sille nimeksi PLC ja painettiin ADD (kuva 7).



Kuva 7. PLC-alustan lisääminen projektiin.

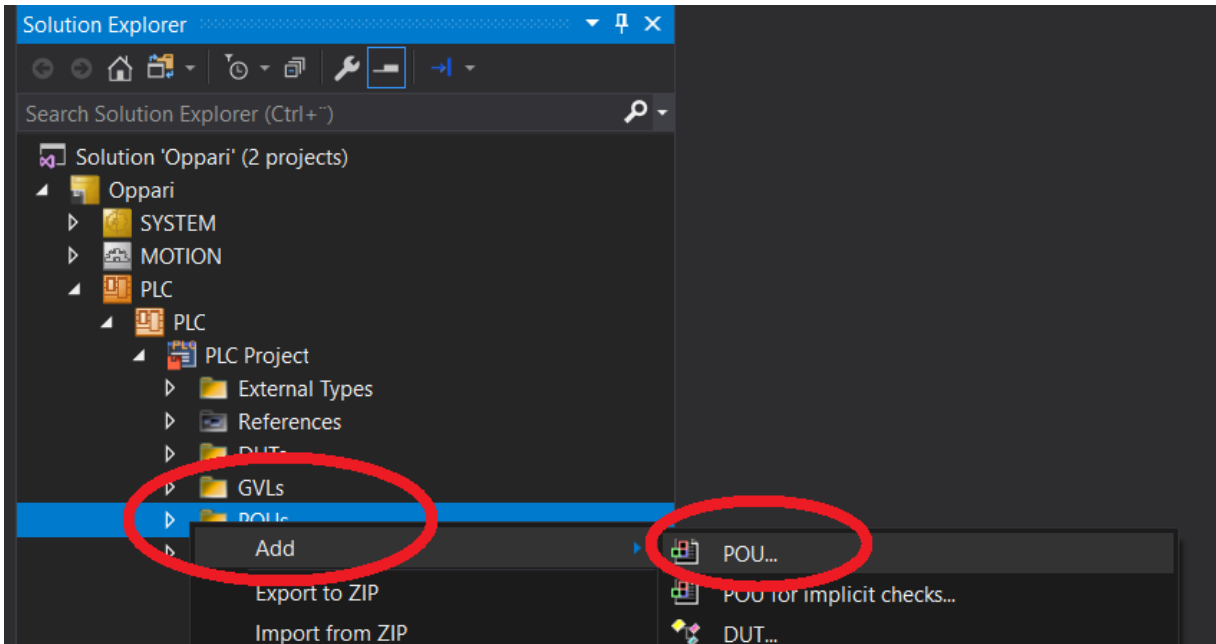
Projekti saatiin luotua, joten seuraavaksi tuotiin projektiin tarvittavat kirjastot. Kirjastoja voidaan tuoda projektiin PLC project -valikon alta klikkaamalla hiiren oikealla kohtaa References, josta edelleen valitaan Add library (kuva 8).



Kuva 8. Kirjastojen lisääminen projektiin.

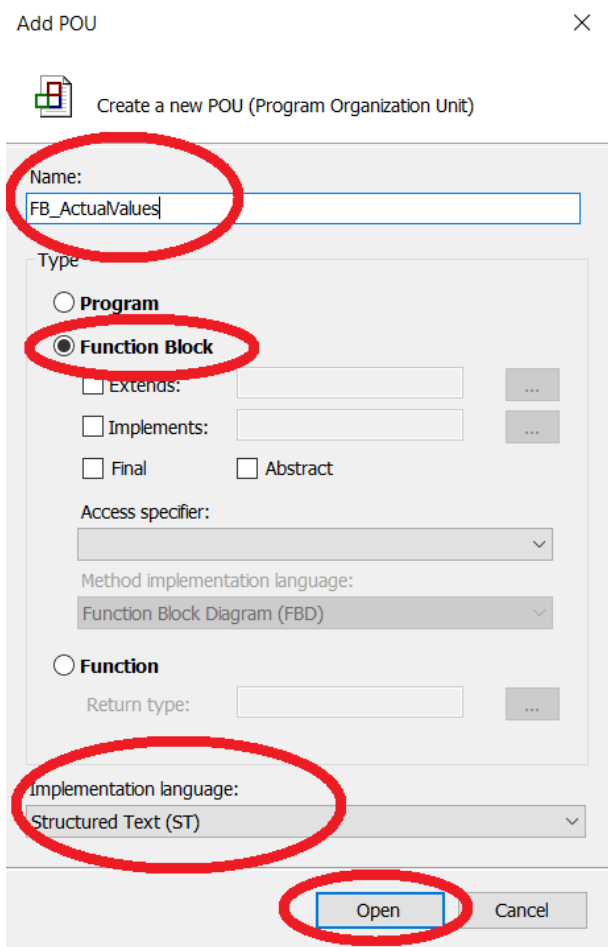
Add library -painikkeesta aukeaa ikkuna, josta voidaan valita tarvittavat kirjastot, kirjastot täytyy kuitenkin lisätä yksitellen.

Kirjastojen lisäämisen jälkeen päästiin tutkimaan kirjastojen sisältämiä toimilohkoja. Kirjastoja tutkiessa erityisen hyödylliseksi kirjastoksi ilmeni Tc2_MC2_Drive-kirjaston toimilohkot. Kyseisen kirjaston toimilohkoja päätettiin käyttää omassa toimilohkossa. Toimilohkon lisääminen projektiin tapahtuu klikkaamalla hiiren oikealla POUs-kohdasta ja siitä edelleen POU... (kuva 9).



Kuva 9. Toimilohkon lisääminen 1.

Tästä aukeaa uusi ikkuna, jossa valitaan POU-typiksi toimilohko Function Block. Sille annetaan nimi ja valitaan ohjelmointikieli. Tässä projektissa kieleksi valikoitui Structured Text (ST) (kuva 10).



Kuva 10. Toimilohkon lisääminen 2.

Itse ohjelmoinnissa ensimmäisenä lisättiin toimilohkoon akselin referenssidatalle sisään-tulo nimeltä io_Axis VAR_IN_OUT muuttujalistan alle ja annettiin tietotyyppi akselin referenssidatan tietotyyppiin AXIS_REF (kuva 11).

```

10 VAR_IN_OUT
11     io_Axis : AXIS_REF;
12 END_VAR

```

Kuva 11. Akselin referenssidatan sisääntulo toimilohkolle

Toimilohkon ulostulomuuttujalistalle (VAR_OUT) lisättiin akselin referenssidatasta saataville mittausarvoille omat muuttujat paikalle o_Axis_Position, väännölle o_Axis_Torque, nopeudelle o_Axis_Velocity ja tilalle o_Axis_State. Muuttujatyypeiksi annettiin AXIS_REF-tietotyyppiin käyttämät muuttujatyypit (kuva 12).


```

22 VAR_OUTPUT
23   o_Axis_Position:          LREAL; //AXIS_REF paikka
24   o_Axis_Torque:           LREAL; //AXIS_REF vääntö
25   o_Axis_Velocity:        LREAL; //AXIS_REF nopeus
26   o_Axis_State:            DWORD; //AXIS_REF tila
27   o_DcBusCurrent:         LREAL; //Tc2_MC2_Drive
28   o_DcBusVoltage:         LREAL; //Tc2_M_2_Drive
29   o_DcBusPower:           LREAL; //DoBusCurrent ja DoBusVoltage täytyy olla
30                           //päällä saadaksesi tämän arvon (WATTI)
31   o_ServoAmplifierTemperature: LREAL; //Tc2_MC2_Drive servovahvistin lämpötila
32   o_ServoAmplifierTemperature_Peak: LREAL; //Servovahvistin korkein lämpötila
33   o_ServoMotorTemperature: LREAL; //Tc2_MC2_Drive servomoottori lämpötila
34   o_ServoMotorTemperature_Peak: LREAL; //Servomoottori korkein lämpötila
35   o_AX5000MainVoltage:     LREAL; //Tc2_MC2_Drive soveltuu AX5000 sarjan
36                           //servo-ohjainten jännitteen mittaukseen.
37 END_VAR

```

Kuva 12. Toimilohkon ulostulot akselin referenssidatalle.

Lopuksi akselin referenssidatalta tulevat tiedot linkitettiin koodin puolelta omiin muuttujiinsa (kuva 13).

```

1  o_Axis_Position      := io_Axis.NcToPlc.ActPos;
2  o_Axis_State        := io_Axis.NcToPlc.AxisState;
3  o_Axis_Torque       := io_Axis.NcToPlc.ActTorque;
4  o_Axis_Velocity     := io_Axis.NcToPlc.ActVelo;

```

Kuva 13. Akselin referenssidatan linkitykset omille muuttujille.

Seuraavaksi lisättiin Tc2_MC2_Drive-kirjaston toimilohkoja toimilohkon sisälle. Kaikkien toimilohkojen lisääminen ja kutsuminen toimilohkon sisälle toimii samoilla periaatteilla, joten seuraavassa on esitetty yksi esimerkki toimilohkon lisäämisestä. Esimerkissä tehdään FB_SoEReadAmplifierTemperature-toimilohkon lisääminen. Toimilohkon muuttujalistaan (VAR) luodaan muuttuja, jolla toimilohkoa kutsutaan fb_SoEReadAmplifierTemperature : FB_SoEReadAmplifierTemperature (kuva 14).

```

38 VAR
39   fb_SoEReadAmplifierTemperature : FB_SoEReadAmplifierTemperature;
40   fb_SoEReadDcBusVoltage : FB_SoEReadDcBusVoltage;
41   fb_SoEReadMotorTemperature : FB_SoEReadMotorTemperature;
42   fb_SoEAX5000ReadActMainVoltage : FB_SoEAX5000ReadActMainVoltage;
43   r_ServoAmplifier_Peak: LREAL; //Korkein lämpötila
44   r_ServoMotor_Peak: LREAL; //Korkein lämpötila
45   fb_blink : fb_blink; //päivittää käytettyjä toimilohkoja
46   t_blinkTime : TIME := T#1S; //toimilohkojen päivitysten aikaväli
47 END_VAR

```

Kuva 14. Toimilohkon muuttujan lisääminen toimilohkoon.

Toimilohkosta tehdään yleispätevä, siitä voidaan valita arvo, mitä toimilohkolla halutaan mitata. Toimilohkon sisällä kutsuttaville toimilohkoille tehdään aktivointimuuttujat, joilla

toimilohkoja voidaan laittaa päälle tarvittaessa. Tämä optimoi toimilohkon käyttämiä PLC:n laskentaresursseja ja muistia. Tämä tehdään lisäämällä toimilohkoon BOOL-tyyppiset sisääntulomuuttujat, joilla ohjataan sisällä kutsuttuvia toimilohkoja päälle tai pois. Sisääntulomuuttujan nimeksi annetaan tässä tapauksessa `i_Enable_ServoAmplifierTemperature` (kuva 15). Sisääntulomuuttujiksi annetaan myös toimilohkon käyttämät Timeout- ja NetId-tulot. Timeout määrittelee ADS-yhteyden aikakatkaisun ajan, ja NetId on laitteen IP-osoite. Toimilohkon ulostulomuuttujalle annetaan nimeksi `o_ServoAmplifierTemperature` ja tietotyyppi `FB_SoEReadAmplifierTemperature`-toimilohkon käyttämä ulostulon tietotyyppi eli LREAL (kuva 16). Näin voidaan myöhemmin linkittää `FB_SoEReadAmplifierTemperature`-toimilohkolta tuleva tieto toimilohkon ulostulomuuttujalle.

```

12  VAR_INPUT
13  i_t_Timeout: TIME := DEFAULT_ADS_TIMEOUT; //DEFAULT_ADS_TIMEOUT = 5s
14  i_NetId : T_AmsNetId := ''; //IP-osoite
15  i_Enable_DcBusCurrent: BOOL; //on/off
16  i_Enable_DcBusVoltage: BOOL; //on/off
17  i_Enable_ServoAmplifierTemperature: BOOL; //on/off
18  i_Enable_ServoMotorTemperature: BOOL; //on/off
19  i_Enable_AX5000Voltage: BOOL; //on/off
20  i_Reset_Peaks: BOOL; //on/off
21  END_VAR

```

Kuva 15. Toimilohkon sisääntulomuuttuja `FB_SoEReadAmplifierTemperature`-toimilohkon kytkimeksi.

```

22  VAR_OUTPUT
23  o_Axis_Position: LREAL; //AXIS_REF paikka
24  o_Axis_Torque: LREAL; //AXIS_REF vääntö
25  o_Axis_Velocity: LREAL; //AXIS_REF nopeus
26  o_Axis_State: DWORD; //AXIS_REF tila
27  o_DcBusCurrent: LREAL; //Tc2_MC2_Drive
28  o_DcBusVoltage: LREAL; //Tc2_M 2_Drive
29  o_DcBusPower: LREAL; //DcBusCurrent ja DcBusVoltage täytyy olla
30  //päällä saadakseen tämän arvon (WATTI)
31  o_ServoAmplifierTemperature: LREAL; //Tc2_MC2_Drive servovahvistin lämpötila
32  o_ServoAmplifierTemperature_Peak: LREAL; //Servovahvistin korkein lämpötila
33  o_ServoMotorTemperature: LREAL; //Tc2_MC2_Drive servomoottori lämpötila
34  o_ServoMotorTemperature_Peak: LREAL; //Servomoottori korkein lämpötila
35  o_AX5000MainVoltage: LREAL; //Tc2_MC2_Drive soveltuu AX5000 sarjan
36  //servo-ohjainten jännitteen mittaukseen.
37  END_VAR

```

Kuva 16. Toimilohkon ulostulomuuttuja `FB_SoEReadAmplifierTemperature`-toimilohkolta tulevalle tiedolle.

Ohjelmakoodissa sisääntulomuuttujalla ohjataan kutsuttavaa toimilohkoa päälle tai pois, joten toimilohkoa kutsutaan IF-lauseeseen sisällä. Toimilohkoa kutsutaan sille aikaisemmin annetulla muuttujanimeillä, jonka perään laitetaan sulkumerkit `fb_SoEReadAmplifierTemperature()`;. Sulkujen sisällä määritellään toimilohkon käyttämiseen tarvittavat muuttujat ja

linkitetään FB_SoEReadAmplifierTemperature-toimilohkon ulostulo aikaisemmin luotuun toimilohkon ulostuloon kuten kuvassa 17 on toteutettu.

```

6 //Servovahvistimen lämpötila
7 IF i_Enable_ServoAmplifierTemperature THEN
8   fb_SoEReadAmplifierTemperature(           //To2_MC2_Drive
9     Axis      := io_Axis,                   //AXIS_REF
10    Execute   := TRUE,                      //Käynnistää toimintalohkon
11    Timeout  := i_t_Timeout,                //Toimintalohkon suorittamiseen sallittu enimmäisaika
12    AmplifierTemperature=>o_ServoAmplifierTemperature //Linkkaus blokin ulostuloon
13  );
14 ELSE   fb_SoEReadAmplifierTemperature.Execute := FALSE;
15 END_IF

```

Kuva 17. Toimilohkon kutsu toimilohkossa IF-lauseen sisällä.

Samoilla periaatteilla toimilohkoon lisättiin servomoottorin lämpötilamittauslohko, AX5000-servovahvistinsarjan jännitteenmittauslohko sekä servovahvistimelta lähtevän tasavirtaväylän jännitteen mittauslohko ja tasavirtaväylän virran mittauslohko. Tasavirtaväylän teho lasketaan ohjelmakoodissa yksinkertaisesti tasavirtaväylien ulostuloista eli jännite kerrotaan virralla ja tulos linkitetään suoraan teholle luodulle toimilohkon ulostulomuuttujalle.

7.2 Testaus

Toimilohkoa testattiin myös oikeilla laitteilla T-Drillin toimitiloissa. Laitteistona toimi kuvan 1 mukainen Beckhoff CX5130 -teollisuus-PC, kuvan 2 mukainen Beckhoff CP2916 -HMI-paneeli, Beckhoff AX5201 -servovahvistin ja Beckhoff AM8033 -servomoottori. Testauksessa toimilohkolla saatiin luettua akselin tietoja sekä kutsutuilla toimilohkoilla saatiin luettua servovahvistimen ja servomoottorin lämpötiloja. Lisäksi saatiin luettua servovahvistimen jännitettä sekä servovahvistimelta lähtevän tasavirtaväylän jännitettä ja virtaa.

Testauksessa huomattiin ongelmaksi, että kutsutut toimilohkot mittasivat arvot vain kerran. Ongelmaan ei löytynyt suoraa ratkaisua, joten ongelmaa lähdettiin ratkaisemaan FB_Blink-toimilohkon avulla. FB_Blink-toimilohko on yrityksen itse kehittämä toimilohko, jolla voidaan vilkuttaa toimilohkon ulostuloa päälle ja pois annetun aikavälin mukaan. FB_Blink-toimilohkolle luotiin kutsumuuttuja nimeltä fb_blink ja aikavälimuuttuja nimeltä t_blinkTime muuttujalistaan. Koodin puolella sitä kutsutaan jokaisen IF-lauseen sisällä, jossa kutsutaan toimilohkoja. Jokaisen toimilohkon kutsun sisällä linkitetään FB_Blink-toimilohkon ulostulo Execute-muuttujaan. Execute-muuttuja kytkee toimilohkon päälle ja pois. Muuttujalistalla aikavälille annetaan ajaksi yksi sekunti, jolloin kutsutut toimilohkot menevät päälle ja pois

sekunnin välein. Näin ulostulot päivittyvät myös sekunnin välein. Kuvassa 18 nähdään, kuinka FB_Blink-toimilohkoa käytetään IF-lauseen sisällä.

```

6 //Servovahvistimen lämpötila
7 IF i_Enable_ServoAmplifierTemperature THEN
8   fb_blink(i_enable := i_Enable_ServoAmplifierTemperature, i_time := t_blinkTime); //päivittäjä
9   fb_SoEReadAmplifierTemperature( //To2_MC2_Drive
10    Axis := io_Axis, //AXIS_REF
11    Execute := fb_blink.o_Blink, //Käynnistää toimintalohkot nousevalla reunalla
12    Timeout := i_t_Timeout, //Toimintalohkon suorittamiseen sallittu enimmäisaika
13    AmplifierTemperature=>o_ServoAmplifierTemperature //Linkkaus blokin ulostuloon
14   );
15 ELSE fb_SoEReadAmplifierTemperature.Execute := FALSE;
16 END_IF

```

Kuva 18. FB_Blink-toimilohkon kutsu ja käyttö IF-lauseessa.

Testauksen aikana lisättiin toimilohkoon vielä servomoottorin ja servovahvistimien lämpötiloille muuttujat, joihin tallennetaan niiden korkeimmat lämpötilat. Muuttujille tehtiin myös ulostulomuuttujat. Tallennus muuttujaan ja linkitys ulostulomuuttujaan tapahtuu tekemällä toimilohkojen kutsujen alle uusi IF-lause, joka toteutuu, kun lämpötila ylittää muuttujan sen hetkisen arvon. Kuvassa 19 on esimerkki servovahvistimen korkeimman lämpötilan tallennuksesta.

```

6 //Servovahvistimen lämpötila
7 IF i_Enable_ServoAmplifierTemperature THEN
8   fb_blink(i_enable := i_Enable_ServoAmplifierTemperature, i_time := t_blinkTime); //päivittäjä
9   fb_SoEReadAmplifierTemperature( //To2_MC2_Drive
10    Axis := io_Axis, //AXIS_REF
11    Execute := fb_blink.o_Blink, //Käynnistää toimintalohkot nousevalla reunalla
12    Timeout := i_t_Timeout, //Toimintalohkon suorittamiseen sallittu enimmäisaika
13    AmplifierTemperature=>o_ServoAmplifierTemperature //Linkkaus blokin ulostuloon
14   );
15
16 IF r_ServoAmplifier_Peak < o_ServoAmplifierTemperature THEN // Pistetään suurin lämpötila muistiin
17   r_ServoAmplifier_Peak := o_ServoAmplifierTemperature;
18   o_ServoAmplifierTemperature_Peak := r_ServoAmplifier_Peak;
19 END_IF
20
21 ELSE fb_SoEReadAmplifierTemperature.Execute := FALSE;
22 END_IF

```

Kuva 19. Servovahvistimen lämpötilan mittaus ja korkeimman lämpötilan tallennus.

Korkeimmille lämpötilamuuttujille tehtiin myös resetointimuuttuja toimilohkon sisääntuloon, sillä saadaan nollattua korkeimpien lämpötilojen muuttujat tarvittaessa. Koodissa resetointi suoritetaan IF-lauseen sisällä, jossa jokaiselle lämpötilamuuttujille ja niiden ulostulomuuttujille annetaan arvoksi 0. IF-lause toteutuu, kun resetointimuuttuja saa arvon 1.

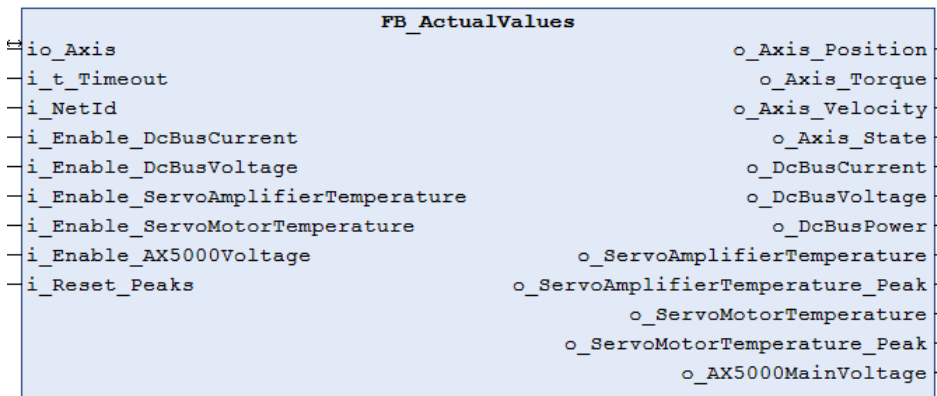
```

82 | //Resetoidaan korkeimmat lämpötilat toimilohkon muistista
83 | IF i_Reset_Peaks THEN
84 |     o_ServoAmplifierTemperature_Peak := 0;
85 |     o_ServoMotorTemperature_Peak := 0;
86 |     r_ServoAmplifier_Peak := 0;
87 |     r_ServoMotor_Peak :=0;
88 | END_IF

```

Kuva 20. Lämpötilamuuttujien nollaus.

Valmista toimilohkoa voidaan kutsua myös FBD-kielellä, jolloin toimilohko näyttää kuvan 21 mukaiselta.

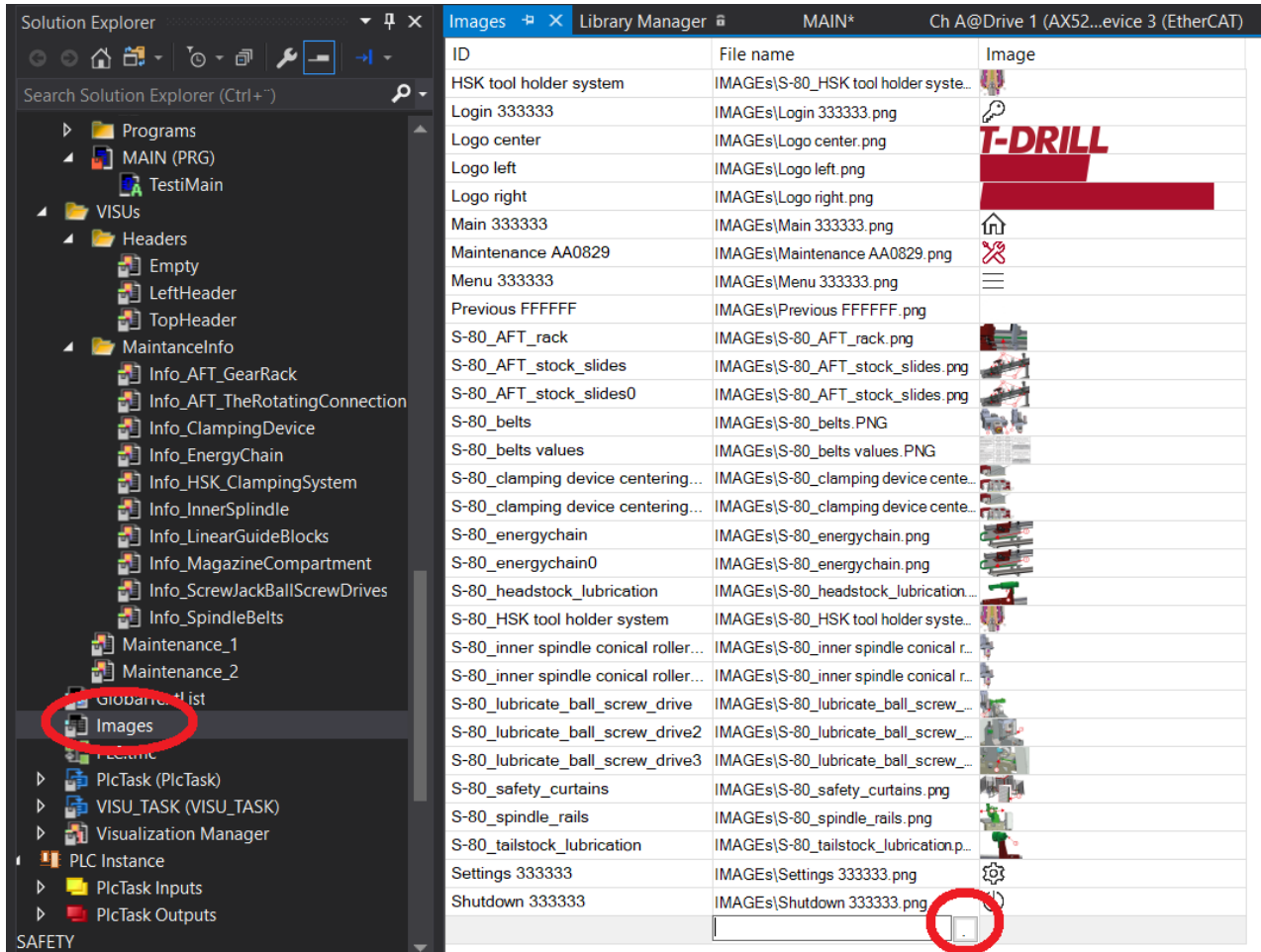


Kuva 21. Toimilohko kutsuttuna FBD-ohjelmointikielellä.

7.3 Huoltoilmoitusten lisäinformaatio sivujen teko

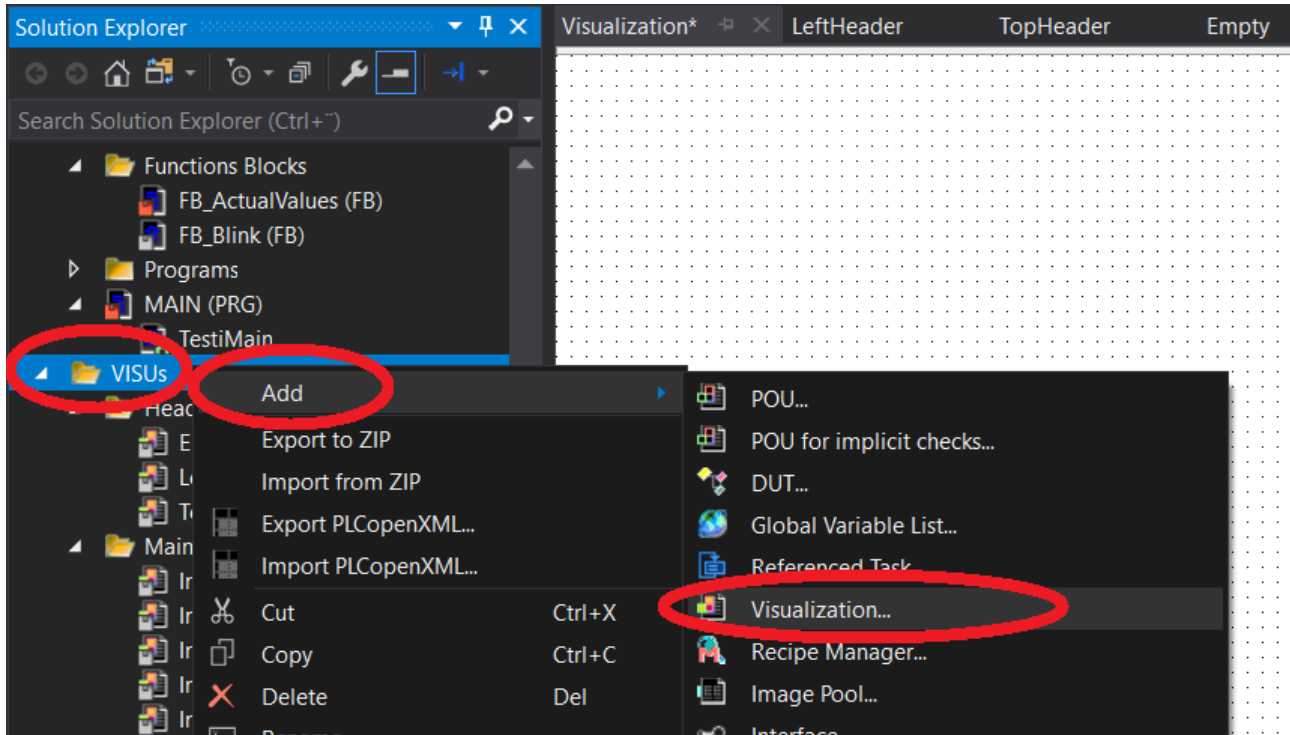
Huoltoilmoitusten lisäinformaatio sivujen tekeminen alkoi lukemalla koneiden käyttöohjeet läpi. Niistä selvisi koneille tarvittavat huollot, niiden aikavälit, työkalut sekä tarvikkeet. Kuvat huollettavista osista ja niiden sijainnista saatiin otettua yrityksen valmiista 3D-mallinuskuvista siten, että huollettava osa on kuvissa korostettu vihreänä. Osalle huoltoilmoituksista saatiin otettua kuvat käyttöohjekirjoissa olevista poikkileikkauskuvista. Kuviin piirrettiin tämän jälkeen nuoliviivoja ja ohjeistuksia huoltoon liittyen.

Kuvien lisääminen projektiin tapahtui viemällä kuvat projektikansion sisälle kansioon nimeltä IMAGES. Tämän jälkeen kuvat voidaan valita käytettäväksi projektin sisällä VISUs-kansion alta images-välilehdeltä (kuva 22). Välilehden sisällä klikataan pienestä napista, joka sijaitsee File name -sarakkeen jokaisella rivillä. Napista aukeaa IMAGES-kansio, josta valitaan haluttu kuva. Kuvat valitaan yksitellen jokaiselle riville.



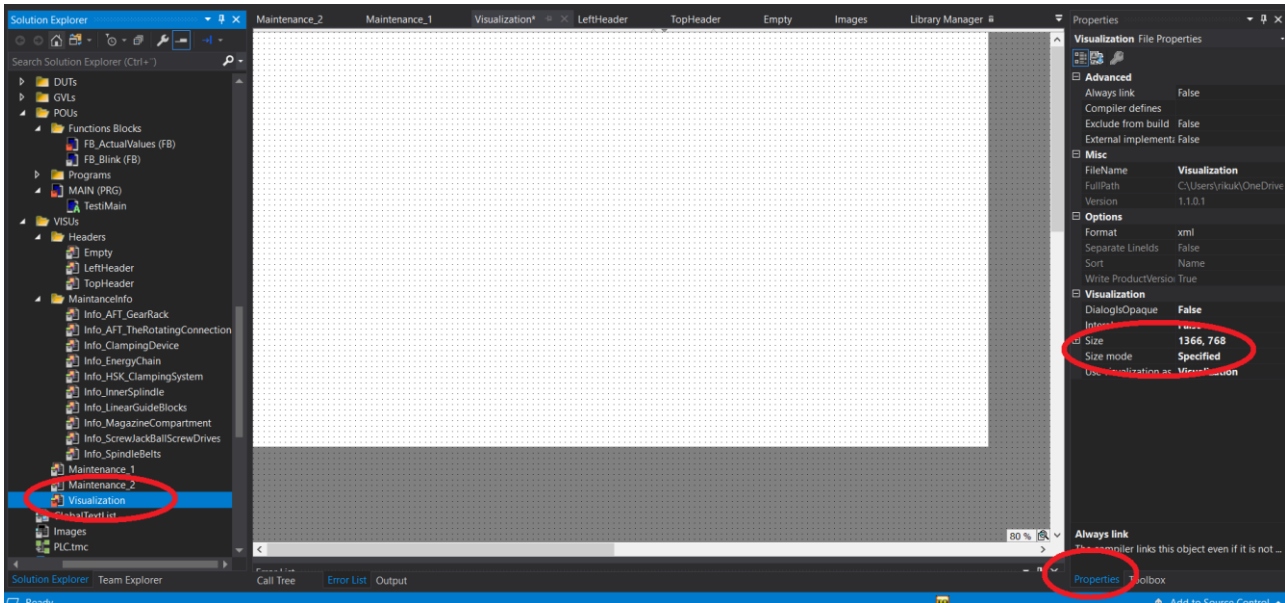
Kuva 22. Kuvien lisääminen projektiin.

Visualisaatiosivu saadaan lisättyä projektiin painamalla hiiren oikealla VISUs-kansiosta. Tämän jälkeen valitaan ADD-painikkeesta avautuvasta valikosta Visualisation... (kuva 23). Ruudulle avautuu ikkuna, jossa visualisaatiosivulle voidaan antaa nimi, jonka jälkeen painetaan open.



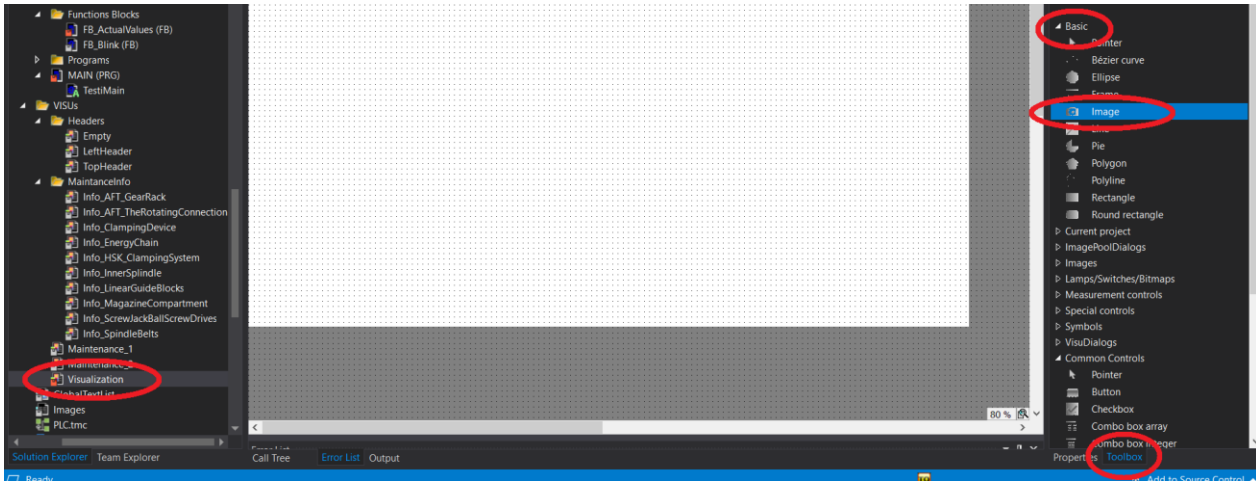
Kuva 23. Visualisaation luominen.

Tämän jälkeen muutetaan visualisaatiosivun koko yrityksen isoimman HMI-paneelin resoluution mukaiseksi, jolloin sivut saadaan skaalattua pienemmille paneeleille ilman sivujen objektien resoluutioiden venymisestä aiheutuvaa epätarkkuutta. Koko saadaan muutettua visualisaatiosivun Properties-valikosta, asettamalla Size mode arvoon Specified, jonka jälkeen voidaan asettaa Size-arvoksi resoluutio 1366, 768 (kuva 24).



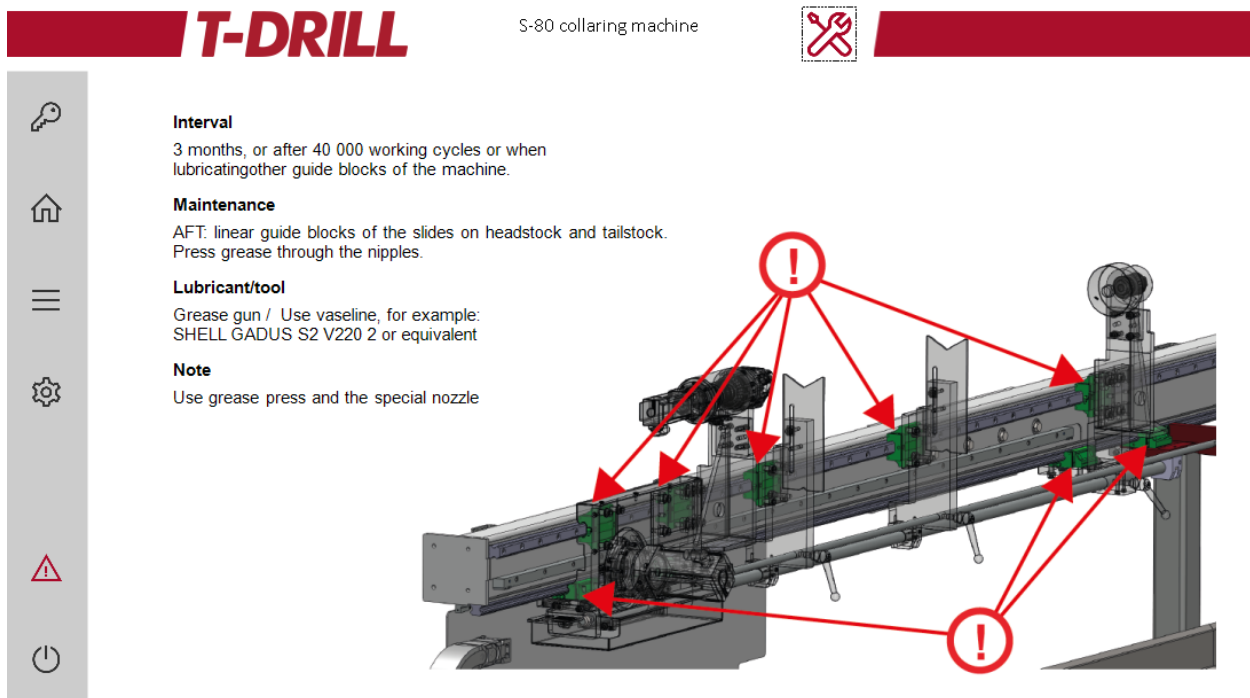
Kuva 24. Visualisaation koko.

Koon määrittelyn jälkeen projektiin tuotiin yrityksen edellisestä projektista mukaan sivu- ja ylävalikot, joiden avulla saatiin sommiteltua kuvat ja tekstit oikeisiin paikkoihin. Valikot tuotiin projektiin kuvina, koska kyseisille valikoille ei ollut projektissa tarvetta. Kuvia voidaan lisätä itse visualisaatiosivun Toolbox-valikosta vetämällä Basic-valikon alta Image sivun päälle (kuva 25). Tästä aukeaa uusi Input Assistant -niminen ikkuna, josta valitaan haluttu kuva ja painetaan oikealta alhaalta OK. Jotta kuvan kokoa voidaan muuttaa hiirellä ja säilyttää kuvasuhde, täytyy kuvan Scaling type käydä muuttamassa arvoon Isotropic, kuvan Properties-valikosta.



Kuva 25. Kuvan lisääminen visualisaatioon.

Kuvien lisäyksen jälkeen tarvittiin vielä informaatiotekstit. Tekstit saatiin lisättyä Toolboxista Common controlsin alta vetämällä label visualisaatiosivulle. Labeliin voidaan kirjoittaa haluttu teksti. Kuvassa 26 näkyy sommitelma valmiista lisäinformaationsivusta yhdestä huoltoilmoituksen lisäinformaationsivusta. Sommitelmasta poistettiin kuitenkin vielä ylä- ja sivuvalikkokuvat, jotta sivu voidaan kopioida yrityksen omiin projekteihin ilman ylimääräisiä kuvia. Valmis sivu on kuvassa 27.



Kuva 26. Sommitelma valmiista huoltoilmoituksen lisäinformaationsivusta.

Interval

3 months, or after 40 000 working cycles or when lubricating other guide blocks of the machine.

Maintenance

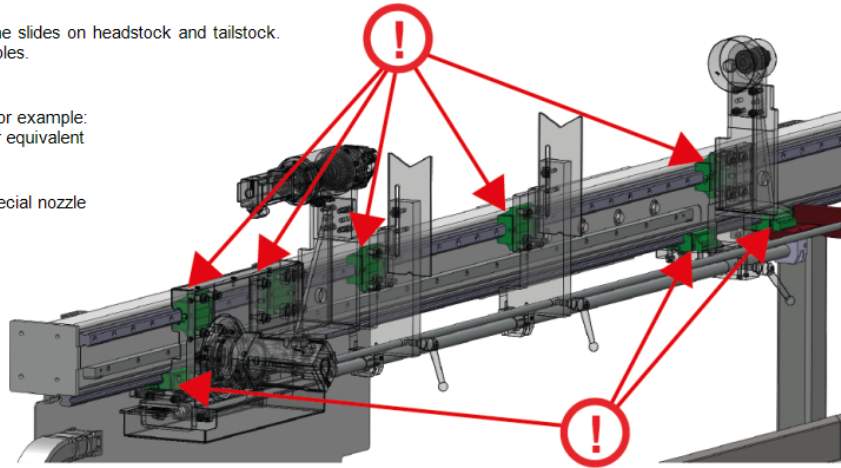
AFT: linear guide blocks of the slides on headstock and tailstock. Press grease through the nipples.

Lubricant/tool

Grease gun / Use vaseline, for example: SHELL GADUS S2 V220 2 or equivalent

Note

Use grease press and the special nozzle



Kuva 27. Valmis huoltoilmoituksen lisäinformaationsivu kopioitavaksi.

Samalla tavalla ja periaatteilla tehtiin yhdeksälle muulle huoltoilmoitukselle lisäinformaationsivu. Koko projekti jää yrityksen käytettäväksi, näin yrityksen työntekijöiden tarvitsee vain kopioida toimilohko ja lisäinformaationsivut sekä niiden kuvat tuleviin projekteihin.

8 POHDINTA JA YHTEENVETO

Tavoitteiksi opinnäytetyölle asetettiin tiedonkeruuhjelmiston kehitys putkentyöstökoneille siten, että sama ohjelmisto olisi kaikille putkentyöstökonemalleille sopiva. Lisäksi tavoitteena oli huoltoilmoitusten modernisointi.

Työ vaati paljon tuntemusta yrityksen koneista ja laitteistosta sekä niiden käyttämistä teknologioista. Palaverissa tuli paljon ideoita minkälaiseen suuntaan tavoitteita olisi mahdollista lähteä kehittämään, moni ideoista olisi vaatinut todella paljon enemmän aikaa toteuttamisen suhteen, eikä aika ei olisi näihin kaikkiin ideoihin riittänyt.

Tulokseksi saatiin toteutettua tiedonkeruutoimilohko ja huoltoilmoituksille lisäinformaatio sivut, joita yritys pystyy hyödyntämään heti projektin luovutuksesta alkaen. Tiedonkeruutoimilohkon mukana palautetaan erillinen käyttöohje. Tiedonkeruutoimilohkoa yritys voi helposti jatkokehittää, mikäli yrityksellä tulee uusia tiedonkeruutarpeita ja valmiita tapoja tiedon keräämiseen. Tiedonkeruulohkoa voisi jatkokehittää heti muokkaamalla `o_Axis_State` ulostulomuuttujalle ulostuloksi tekstiarvo. Tällä hetkellä ulostuloon tulee tilatieto lukuarvona, jonka merkitys selviää Beckhoffin verkkosivuilta, ellei sitä ulkoa muista. Lisäksi tiedonkeruulohkoa voisi kehittää siirtämällä `FB_Blink`-toimilohkon ominaisuudet tiedonkeruulohkon sisälle, jolloin tiedonkeruulohko ei tarvitse toimiakseen erillistä `FB_Blink`-toimilohkoa. Toimilohkoa tehdessä tuli vastaan myös ongelmatilanteita, joissa tiettyjä tietoja ei saatu jostakin tuntemattomasta syystä mitattua, vaikka niitä yritettiin lukea samalla tavalla kuin muitakin tietoja. Tulevaisuudessa olisi mielenkiintoista löytää näihin ongelmiin syy. Lisäksi vastaan tuli tiedonmittaustapoja, joita ei olisi saanut helposti toimilohkolla mitattua, vaan siihen olisi täytynyt tehdä konfiguroinnin puolella linkityksiä. Nämä tietojen mittaukset jätettiin pois, koska tiedot eivät olleet kriittisiä tietoja koneiden toiminnan kannalta. Lisäksi toimilohkon helppo kopioitavuus projektista toiseen olisi kärsinyt. Tähän olisi vielä mielenkiintoista selvittää, olisiko kyseisten tietojen mittaukseen olemassa muita helpompia tapoja.

Huoltoilmoitusten modernisoinnissa päädyttiin tekemään lisäinformaatio sivut huoltoilmoituksille. Aluksi suunnitteilla oli lisäksi modernisoida huoltoilmoitussivu visuaalisemmaksi, mutta suunnitelma jätettiin toteutettavaksi työn loppuun, mikäli sille jäisi aikaa. Tämä

päätettiin yhdessä yrityksen kanssa. Huoltosivu on näkyvillä ainoastaan T-Drillin työntekijöillä tai koulutetuilla henkilöillä, joten käytettävyys priorisoitiin visualisoinnin edelle, vaikka sen tekeminen visuaalisemmaksi olisi ollutkin mukavaa.

Työn tavoitteet täytettiin. Lisäksi tiedonkeruuhjelmistosta saatiin helposti muokattava tulevaisuuden laajennuksien varalle, sillä todennäköisesti tulevaisuudessa tiedonkeruun määrä ja merkitys kasvavat. Kerättyä tietoa voidaan käyttää esimerkiksi yritysten markkinoinnissa, ennakoivassa huollossa ja kehityksessä.

LÄHTEET

Asiakastieto. (2020). *T-Drill Oy*. <https://www.asiakastieto.fi/yritykset/fi/t-drill-oy/05487858/yleiskuva>

Beckhoff. (i.a. -a). *TwinCAT Automation software*. <https://www.beckhoff.com/fi-fi/products/automation/twincat/>

Beckhoff. (i.a. -b). *Introduction*. https://infosys.beckhoff.com/english.php?content=../content/1033/tc3_licensing/index.html&id=1162492240319917408

Beckhoff. (i.a. -c). *TExxxx | TwinCAT 3 Engineering*. <https://www.beckhoff.com/fi-fi/products/automation/twincat/te1xxx-twincat-3-engineering/>

Beckhoff. (i.a. -d). *TC1000 | TwinCAT 3 ADS*. <https://www.beckhoff.com/fi-fi/products/automation/twincat/tc1xxx-twincat-3-base/tc1000.html>

Beckhoff. (i.a. -e). *ADS introduction*. https://infosys.beckhoff.com/english.php?content=../content/1033/tcadscommon/html/tcadscommon_intro.htm&id=

Beckhoff. (i.a. -f). *FBD/LD/IL*. https://infosys.beckhoff.com/english.php?content=../content/1033/tc3_plc_intro/2526631051.html&id=2322164930080637687

Beckhoff. (i.a. -g). *Structured Text (ST), Extended Structured Text (ExST)*. https://infosys.beckhoff.com/english.php?content=../content/1033/tc3_plc_intro/2526757515.html&id=6803421220384787374

Bolton, W. (2006). *Programmable logic controllers*. Newnes.

Clark, J. (17.11.2016). *What is the Internet of Things (IoT)?* IBM. <https://www.ibm.com/blogs/internet-of-things/what-is-the-iot/>

Collin, J. & Saarelainen, A. (2016). *Teollinen internet*. Talentum.

Falco, J., Stouffer, K., Wavering, A., & Proctor, F. (i.a.). IT Security for Industrial Control Systems. National Institute of Standards and Technology. https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=821684

Fisher, S. (30.7.2019). *What is TCP/IP and How Does it Work?* Haettu 13.3.2022, <https://www.avast.com/c-what-is-tcp-ip>

Fonselius, J., Rinkinen, J., & Vilenius, M. (1998). *Servotekniikka*. Edita.

International Electrotechnical Commission (i.a.). *IEC-61131-3:2013*. <https://web-store.iec.ch/publication/4552>

Liang, Q. (2011). *The Study of Soft PLC Running System*. Elsevier. <https://doi.org/10.1016/j.proeng.2011.08.228>

Microsoft. (i.a. -a). *What is cloud computing?* <https://azure.microsoft.com/en-us/overview/what-is-cloud-computing/#benefits>

Microsoft. (i.a. -b). *What is hybrid cloud computing?* <https://azure.microsoft.com/en-us/overview/what-is-hybrid-cloud-computing/>

Microsoft. (i.a. -c). *Serverless computing*. <https://azure.microsoft.com/en-us/overview/serverless-computing/>

Norton. (24.2.2022). *What is a VPN?* Haettu 14.3.2022, <https://us.norton.com/internetsecurity-privacy-what-is-a-vpn.html>

Omron. (i.a.). *Technical Explanation for Servomotors and Servo Drives*. https://www.ia.omron.com/data_pdf/guide/14/servo_tg_e_1_1.pdf

Ranger, S. (25.2.2022). *What is cloud computing? Everything you need to know about the cloud explained*. <https://www.zdnet.com/article/what-is-cloud-computing-everything-you-need-to-know-about-the-cloud/>

Singh, J., & Deswal, M. (2015). *PLC and SCADA*. Laxmi Publications.

Tosibox. (i.a.). *Lock*. <https://helpdesk.tosibox.com/support/solutions/folders/2100002391>

T-Drill. (i.a.). *Yritys*. <https://t-drill.com/fi/yritysinfo/>

Wallenius, N. (6.11.2019). *Näin tunnistat aidon pilvipalvelun*. <https://niklaswallenius.fi/pilvipalvelu-maaritelmia/>

Wallenius, N. (23.2.2022). *Konttitekniologia – mitä kontit ovat ja mitä hyötyä niistä on?* <https://niklaswallenius.fi/konttitekniologia-mita-hyotya/>