

Jesse Lassila

**OHJAUSJÄRJESTELMÄN SUUNNITTELU
BECKHOFF-YMPÄRISTÖSSÄ**

OHJAUSJÄRJESTELMÄN SUUNNITTELU BECKHOFF-YMPÄRISTÖSSÄ

Jesse Lassila
Opinnäytetyö
Kevät 2020
Konetekniikan tutkinto-ohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Konetekniikan tutkinto-ohjelma, koneautomaatio

Tekijä: Jesse Lassila

Opinnäytetyön nimi suomeksi: Ohjausjärjestelmän suunnittelu Beckhoff-ympäristössä

Opinnäytetyön nimi englanniksi: Control system development in Beckhoff environment

Työn ohjaaja: Juha Juntila

Työn valmistumislukukausi ja -vuosi: kevät 2020

Sivumäärä: 50 + 1 liitettä

Opinnäytetyö tehtiin Head Recycle Systems Oy:lle Oulussa vuonna 2019. Työssä suunniteltiin ja toteutettiin ohjausjärjestelmä automaattiselle muovinkierätyslaitteelle Beckhoff Automationin ohjaustekniikoilla. Työssä valittiin ohjausjärjestelmälle komponentit ja ohjelmoitiin toimiva logiikkaohjelma. Ohjausjärjestelmään integroitiin lähi-infrapuna-anturi ja koodilukkona toimiva matriisinäppäimistö. Lisäksi suunniteltiin ja valmistettiin käyttöliittymät huoltohenkilöille ja käyttäjille. Opinnäytetyössä tutkittiin myös jatkokehitysmahdollisuutena IoT-tietokannan yhdistämistä ohjausjärjestelmään.

Ohjausjärjestelmä suunniteltiin pääosin ennestään valitun lähi-infrapuna-anturin vaatimusten perusteella. Anturi oli laitteen tärkein osa, sillä anturin avulla laite pyrkii erottamaan kierrätettävästä muovista sekajätteen erilleen. Ohjausjärjestelmän täytyi olla yhteensopiva anturin kanssa, jotta laitteen toiminta oli mahdollista.

Ohjausjärjestelmäksi valittiin lähi-infrapuna-anturia tukeva Beckhoffin sulautettu tietokone. Sulautettuun tietokoneeseen valittiin I/O-aulukon mukaan tarvittavat modulaariset tulo- ja lähtöterminaalit. TwinSAFE-turvallisuuslogiikkakomponentit valittiin laitteen turvallisuusluokan perusteella. Kierrätyslaitteen toiminnasta laadittiin vuokaavio, jonka perusteella ohjelmoitiin logiikkaohjelma. Logiikkaohjelma ohjelmoitiin Structured Text -kielellä, johon yhdistettiin Function Block Diagram -kielisiä visuaalisia ohjelmayksiköitä. Turvalogiikka ohjelmoitiin sertifioitulla turvaohjelmointikielellä. IoT-tietokannan yhdistäminen ohjausjärjestelmään selvitetiin ja se todettiin mahdolliseksi Beckhoffin porttisovelluksella. Ohjausjärjestelmästä tuli toimiva ja se asennettiin yrityksen ensimmäiseen pilottilaitteeseen. Ohjausjärjestelmän kehittämistä jatketaan opinnäytetyön jälkeen pilottilaitteesta saatujen kokemusten avulla.

Asiasanat: ohjausjärjestelmä, IEC 61131-3, logiikkaohjelmointi

ABSTRACT

Oulu University of Applied Sciences
Degree programme of Mechanical Engineering, Machine Automation

Author: Jesse Lassila

Title of thesis: Control system development in Beckhoff environment

Supervisor: Juha Junntila

Term and year when the thesis was submitted: spring 2020

Pages: 50 + 1 appendices

This thesis was done for Head Recycle Systems Oy in Oulu. The project involved designing and implementing a control system for an automatic plastic recycling machine using Beckhoff Automations technologies. Control system's components was chosen and working PLC-program was programmed. Thesis also included integrating Near-infrared sensor and matrix keypad into the control system. In addition, user interfaces for maintenance personnel and users were designed and manufactured. The thesis also explored the possibility of connecting the IoT database to the control system as a further development opportunity.

The control system was mainly designed based on the requirements of a previously selected near-infrared sensor. The sensor was the most important part of the device, as the sensor tends to separate mixed waste from recyclable plastic. The control system had to be compatible with the sensor for the device to function.

Beckhoff's embedded computer was selected as the control system. Chosen embedded computer supported near-infrared sensor. Modular input and output terminals were selected according to the I/O list. TwinSAFE safety logic components were selected based on the safety class of the device. Devices operational function was drawn to flow chart. Based on flow chart PLC-program was programmed with Structured Text, which was combined with Function Block Diagram. The safety logic was programmed in a certified security programming language. The connection of the IoT database to the control system was investigated and proved possible with the Beckhoff port application. The control system became operational and was installed on the company's first pilot device. The development of the control system will continue after the thesis.

Keywords: PLC, IEC 61131-3, PLC-programming

ALKULAUSE

Opinnäytetyö tehtiin Head Recycle Systems Oy:lle Oulussa. Opinnäytetyön suunnittelu aloitettiin kesätöiden ohella kesällä 2019 ja se toteutettiin syksyllä 2019. Työn yhteyshenkilönä ja ohjaajana toimi yrityksen puolelta tuotekehityspäällikkö Jari Viitala. Oulun ammattikorkeakoulun puolelta ohjaajina toimivat lehtori Juha Junttila ja lehtori Tuija Juntunen.

Kiitos kaikille opinnäytetyön parissa toimineille henkilöille. Erityiskiitos Beckhoffin Oulun toimipisteen työntekijöille tuesta ja kaikista neuvoista. Haluan kiittää myös Head Recycle Systems Oy:n toimitusjohtaja Johnny Pehkosta ja tuotekehityspäällikkö Jari Viitalaa kesätyöpaikasta, joka johti opinnäytetyön tekemiseen ja työpaikkaan opinnäytetyön valmistuttua. Kiitos myös tyttöystävälleni ja perheeleni tuesta ja kannustuksesta.

Oulussa 12.2.2020

Jesse Lassila

SISÄLLYS

TIIVISTELMÄ	3
ABSTRACT	4
ALKULAUSE	5
SISÄLLYS	6
SANASTO	7
1 JOHDANTO	9
2 OHJAUSJÄRJESTELMÄT	10
2.1 Varhaiset ohjausjärjestelmät	10
2.2 Ohjelmitava logiikka	11
3 OHJAUSJÄRJESTELMÄN VALINTA JA SUUNNITTELU	14
3.1 Lähi-infrapuna-anturin vaatimukset ohjausjärjestelmästä	14
3.2 Komponentit	15
3.2.1 Ohjain	15
3.2.2 Tulo- ja lähtöterminaalit	17
3.2.3 Turvajärjestelmä	19
3.3 Logiikkaohjelmointikieli	20
4 OHJAUKSEN SUUNNITTELU JA TOTEUTUS	22
4.1 Vuokaavio	23
4.2 Muuttujalista	23
4.3 Laitteen ensisijainen logiikkaohjelma	24
4.4 Huollon logiikkaohjelma	29
4.5 Matriisinäppäimistön integroiminen	32
4.6 Turvajärjestelmän logiikkaohjelma	38
5 LÄHI-INFRAPUNA-ANTURIN INTEGROINTI PLC:HEN	41
6 JATKOKEHITYS	42
7 YHTEENVETO	45
LÄHTEET	47
LIITTEET	
Liite 1 Vuokaavio	

SANASTO

AMQP	Advanced Message Queuing Protocol, IoT-protokolla, jota käytetään vahvaa tietoturvaa tarvitsevilla sovelluksissa
CFC	Continuous Function Chart -logiikkaohjelmointikieli
CTU	Count up -toimilohko, nousevan reunan tunnistaessaan lisää laskuriin arvon yksi
EDM	External Device Monitoring, ulkoisten koskettimien valvonta
FBD	Function Block Diagram -logiikkaohjelmointikieli
IoT	Internet of Things, eli esineiden internet, on verkko, joka perustuu automaattiseen tiedonsiirtoon teknisten laitteiden välillä internetiä käyttäen
LD	Ladder Diagram -logiikkaohjelmointikieli
MQTT	MQ Telemetry Transport, yleinen avoimeen lähdekoodiin perustuva IoT-protokolla
NIR	Near Infrared Spectroscopy, lähi-infrapuna
pitopiiri	pitää arvon, kunnes se kuitataan. Voidaan toteuttaa mekaanisesti tai esimerkiksi RS-kiikun avulla.
PLC	Programmable Logic Controller, ohjelmitava logiikka
rajapinta	määritelmä, jonka avulla laitteet ja ohjelmat voivat vaihtaa tietoja keskenään
rele	sähköisesti ohjattava sähkömekaaninen kytkin
RS-kiikku	Reset/Set-kiikku, jossa reset-muuttuja on dominantti
SC	UML Statechart -logiikkaohjelmointikieli

SFC	Sequential Function Chart -logiikkaohjelmointikieli
ST	Structured Text -logiikkaohjelmointikieli
tilakone	ohjelmointimalli, jossa toiminnallisuudet jaetaan eri tiloihin, tiloja vaihdetaan ohjelmoitujen komentojen perusteella
TON	Timer on delay -toimilohko, viiveen mahdollistava toimilohko

1 JOHDANTO

Opinnäytetyössä suunnitellaan ohjausjärjestelmä Beckhoff-ympäristössä automaattiseen muovinkierrätyslaitteistoon. Työn tavoitteena on suunnitella ja toteuttaa laitteeseen toimiva ohjaus ja käyttöliittymä. Kierrätyslaitteisto tiivistää kaikki jätteet ja erottaa kierrätettävän muovin sekajätteestä. Työn toimeksiantajana toimii oululainen Head Recycle Systems Oy.

Head Recycle Systems Oy on Oulussa vuonna 2018 perustettu yritys, joka kuuluu Head Invest -konserniin. Yrityksen perustivat toimitusjohtaja Johnny Pehkonen ja teollisuusneuvos Veikko Lesonen. (1.) Yritys on keskittynyt muovin kierrätykseen ja siihen liittyviin teknologioihin.

Työn päätavoitteena on suunnitella, toteuttaa ja dokumentoida toimiva logiikka-ohjaus automaattiseen muovinkierrätyslaitteistoon. Työ sisältää myös käyttäjän ja huoltohenkilön käyttöliittymän suunnittelun ja toteutuksen. Tavoitteena on lisäksi integroida lähi-infrapuna-anturi ja koodinäppäimistö ohjausympäristöön. Logiikkaan tehdään myös tarvittavat turvatoiminnot Beckhoffin turvalogiikan avulla. Jatkokehityksenä tutkitaan mahdollisuutta käyttää IoT-tietokantaa Beckhoff-ympäristössä.

2 OHJAUSJÄRJESTELMÄT

Ohjausjärjestelmällä tarkoitetaan koneen tai toiminnon ohjaukseen käytettävää ohjausta. Yleisesti ohjausjärjestelmää käytetään ohjaamaan koneen toimintoja joko automaattisesti tai käyttäjän ohjaamana. Myös automaattisissa laitteissa täytyy olla jonkinlainen konekohtainen käyttöliittymä, jolla voidaan vaikuttaa manuaalisesti koneen toimintaan. Teollisuudessa yleisimpiä ohjausjärjestelmiä ovat ohjelmoitava logiikka eli PLC (Programmable Logic Controller), robotin ohjaus, PID-säädin (Proportional-Integral-Derivative -säädin) ja NC-ohjaus (Numerical Control). (2, s. 210.)

Arkielämän esimerkkejä ohjausjärjestelmistä ovat yksinkertaisimmillaan liiketunnistimilla toimivat ulkovalaisimet, joissa liiketunnistimen tunnistama liike kytkee valon päälle ja tietyn ajan kuluttua valo sulkeutuu. Monimutkaisia ohjausjärjestelmiä edustavat esimerkiksi bensa-automaatit. Yleisesti automaatit sisältävät käyttöliittymän, jota käyttämällä asiakas syöttää rahasumman, jolla haluaa bensiiniä, pankkikortin tunnusluvun ja mahdollisesti pumpun numeron. Tämän jälkeen asiakas ottaa bensiinipistoolin käteen ja automaattisesti tämän pistoolin pumppu käynnistyy. Pumppu sammuu automaattisesti, kun rahasumma, jonka asiakas on valinnut, tulee täyteen tai vaihtoehtoisesti kun esimerkiksi auton bensatankki on täynnä. (3, s. 6.)

2.1 Varhaiset ohjausjärjestelmät

Ennen tekniikan ja tietokoneiden kehittymistä sähköiset ohjausjärjestelmät valmistettiin käyttämällä releitä. Releitä käyttämällä pystytään toteuttamaan suuria ohjausjärjestelmiä. Kytkemällä releitä rinnan ja sarjaan pystytään toteuttamaan kaikki loogiset toiminnot, joita nykyäänkin käytetään. Tarvittavat ajastintoinnot voidaan toteuttaa aikareleillä, minkä takia relelogiikan toiminta ei juurikaan eroa nykyaikaisesta ohjelmoitavalla logiikalla tehdystä ohjausjärjestelmästä. Relelogiikan suurimpina ongelmina ovat järjestelmän monimutkaisuus ja muutoksien tekeminen alkuperäiseen toimintaan (kuva 1). Nykyään relelogiikoita käytetään ainoastaan erittäin yksinkertaisissa ohjauksissa. (2, s. 211.)



KUVA 1. Relelogiikkapaneeli (4)

2.2 Ohjelmoitava logiikka

Monimutkaisten johdotusten ja suurten komponenttimäärien takia autoteollisuudelta tuli toive 1960-luvulla ohjelmoitavasta logiikkaohjauslaitteesta, jolla voisi korvata nykyiset relelogiikat. Ensimmäiset laitteet tulivat markkinoille vuosina 1968–1969. Hieman myöhemmin Bedford Associates patentoi PLC:n. (5, s. 241–242).

PLC on yleisesti mikroprosessorilla varustettu pieni tietokone (2, s. 212), mutta nykyään on myös paljon PLC:itä, jotka ovat täysin PC-pohjaisia ja joissa on tehokas prosessori (6). Ohjelmoitavia logiikoita käytetään esimerkiksi koneiden, tuotantolinjojen ja yleisesti reaaliaikaisten automaatioprosessien ohjaukseen. Yhdellä PLC:llä on mahdollista korvata jopa tuhansia releitä, mikä yksinkertaistaa sähkösuunnittelua ja johdotuksia (kuva 2). PLC:tä käytettäessä tuotantolinjaston

muutokset pystytään tekemään helposti ja nopeasti ohjelmaa päivittämällä. Aiemmin releet jouduttiin johdottamaan aina uudelleen, jos tuotantolinjaan haluttiin muutoksia. (2, s. 212.)



KUVA 2. Modulaarinen PLC (7)

PLC:ssä on tulo- ja lähtöportteja, joihin kytketään kaikki tarvittavat kenttälaitteet, kuten moottorit, anturit ja painikkeet. Tuloportteihin kytketään komponentit, joiden antamien signaalien ja tietojen perusteella ohjataan järjestelmää. Lähtöportteihin kytketään sen sijaan toimilaitteet, joita halutaan ohjata, kuten moottorit ja venttiilit. Järjestelmää ohjataan logiikkaohjelmalla, joka tarkkailee kenttälaitteiden signaaleja ja tietoja. Signaalien ja tietojen perusteella ohjelma ohjaa lähtöportteja ja kytkee tarvittavat komponentit päälle ja pois päältä. (5, s. 242.) PLC:itä on moneen eri tarkoitukseen. Pienikokoisilla voidaan tehdä yksinkertaisia ON/OFF-kytkentöjä. Suurikokoisilla ja tehokkailla laitteilla voidaan ohjata kokonaisen tehtaan prosessia. (2, s. 212.)

Ohjelmoitavat logiikat on aikaisemmin jaettu askeltaviin ja vapaasti ohjelmoitaviin logiikoihin. Askeltavalla logiikalla pystytään tekemään pieniä relelogiikan korvaavia järjestelmiä. Vapaasti ohjelmoitavalla logiikalla pystytään tekemään suurempia järjestelmiä. Nykyään lähes kaikilla logiikkaohjaimilla voidaan toteuttaa askel-

tavia logiikoita ja vapaasti ohjelmoitavia logiikoita ohjattavan prosessin vaatimusten mukaisesti, ja siksi luokittelusta on luovuttu. Nykyisin PLC:t jaetaan kiinteällä I/O-määrällä oleviin ja modulaarisiin järjestelmiin. Kiinteitä järjestelmiä käytetään pieniin laitteisiin ja modulaarisia suurempien järjestelmien ohjaukseen. (2, s. 210–212.)

3 OHJAUSJÄRJESTELMÄN VALINTA JA SUUNNITTELU

Ohjausjärjestelmä suunniteltiin ja valittiin laitteen asettamien vaatimusten perusteella. Tärkeimpänä vaatimuksena oli, että kierrätyslaitetta pystytään ohjaamaan NIRONE Sensor S lähi-infrapuna-anturilla. Anturilla pystytään tunnistamaan esimerkiksi muovi sekajätteestä. Muita vaatimuksia olivat tulo- ja lähtöporttien modulaarisuus, jotta ohjausjärjestelmää voitaisiin tarvittaessa laajentaa. Lisäksi vaatimuksena oli ohjausjärjestelmän toimittajalta saatava laaja tuki, koska Head Recycle Systems Oy:llä ei ollut laajaa tietämystä ohjausjärjestelmistä ennestään.

Aikaisemman koulussa tehdyn tuotekehitysprojektin kokemuksen perusteella ja Head Recycle Systems Oy:n tuotekehitysjohtaja Jari Viitalan suosituksella ohjausjärjestelmän toimittajaksi valittiin Beckhoff Automation. Beckhoff Automationin ohjausjärjestelmä täytti kaikki edellä olevat vaatimukset, minkä lisäksi yrityksen paikallinen tuki Oulussa on erinomainen. Lisäksi valintaan vaikutti ilmainen kehitysympäristö TwinCAT 3, monipuolinen tuotevalikoima ja toimitusvarmuus.

Kierrätyslaitteesta suunniteltiin aluksi sähkömoottoreilla toimivaa, jolloin puristinruuvien puristus ja lajittelijan kääntö tapahtuisivat sähköisillä oikosulku tai servo-moottoreilla. Koska sähkömoottorit vaativat kalliit vaihteistot tarvittavan vääntömomentin saamiseksi, päädyttiin laitteesta tekemään hydraulikkatoiminen. Hydraulikka mahdollisti suuret voimat kohtuullisen edullisesti. Ohjausjärjestelmään tämä lisäsi jonkin verran tulo- ja lähtöporttien määrää, mutta kalliista moottoriohjaimista ja vaihteistoista päästiin eroon.

3.1 Lähi-infrapuna-anturin vaatimukset ohjausjärjestelmältä

Kierrätyslaitteen muovin tunnistus ja lajittelu tapahtuvat lähi-infrapuna-anturin eli NIR-anturin avulla, joten ohjausjärjestelmän piti pystyä lukemaan anturin arvot ja ohjaamaan näiden tietojen avulla PLC:n lähtöportteja. Kierrätyslaitteessa käytettiin suomalaissaksalaisen Spectral Enginen valmistamaa NIRONE Sensor S -anturia (kuva 3). Head Recycle Systems Oy oli valinnut jo aiemmin anturin, joten se määräsi pääosin ohjausjärjestelmän valinnan.



KUVA 3. NIRONET™ Sensor S (8)

Anturi on mahdollista liittää USB- tai RS232-kaapelilla järjestelmään. Ohjausjärjestelmän piti sisältää vähintään toinen näistä liittimistä ja lisäksi pystyä keskustelemaan anturin kanssa. Helpoimmalla anturin sai keskustelemaan järjestelmän kanssa, jos ohjausjärjestelmä tunnistaa anturin sarjaporttina. Tällöin anturin ja ohjausjärjestelmän rajapinta voidaan toteuttaa anturia tukevalla koodikielellä. (9.)

3.2 Komponentit

Ohjausjärjestelmän komponentit valittiin yhteistyössä Beckhoff Automationin Oulun toimipisteen työntekijöiden kanssa Beckhoffin laajasta tuotevalikoimasta. Beckhoff Automation on saksalainen perheyrittäjä, joka valmistaa edelleen kaikki tuotteensa Saksassa omissa tehtaissaan. Beckhoff valmistaa muun muassa teollisuustietokoneita, moottoreita, tulo- ja lähtöterminaaleja ja ohjelmistoja. Omissa tehtaissa valmistus mahdollistaa nopeat muutokset ja teknologian perässä pysymisen. (10, s. 3–8.)

3.2.1 Ohjain

Beckhoffin ohjaimet ovat täysin tietokonepohjaisia ja sisältävät Windows-käyttöjärjestelmän. Suurin etu Beckhoffin ohjaimissa onkin mahdollisuus hyödyntää tietokoneen ominaisuuksia. Beckhoffin ohjaimet jaetaan teollisuustietokoneisiin ja pienempiin sulautettuihin tietokoneisiin. Tietokone sisältää laitteen älyn, joka ohjaa niihin liitettäviä modulaarisia tulo- ja lähtöterminaaleja sekä kenttäväyläterminaaleja. Yhdessä nämä muodostavat ohjaimen, joka toimii PLC:n tavoin. (10, s. 3.)

Teollisuustietokoneiden tehovalikoima on laaja, ja niitä voidaan käyttää niin pienissä järjestelmissä kuin suurissa tehdasprosesseissa. Osa teollisuustietokoneista on rakennettu kosketusnäytön sisälle, mikä helpottaa ja yksinkertaistaa johdotuksia. Sulautetut tietokoneet ovat keskimäärin pienempiä kuin teollisuustietokoneet, ja ne on mahdollista kiinnittää DIN-kiskoon sähkökaapin sisään. Sulautettujen tietokoneiden teho vastaa parhaimmillaan pienempiä teollisuustietokoneita, mutta yleisesti niitä käytetään pienten laitteiden ja prosessien ohjaukseen. Sulautettuihin tietokoneisiin on mahdollista liittää sisäisen virtalähteen ansiosta suoraan tulo- ja lähtöterminaaleja, eikä erillistä kytkinlaitetta tarvita kuten teollisuustietokoneissa. (10, s. 8–9; s. 76.)

Sulautetut tietokoneet soveltuivat laitteeseen parhaiten pienen kokonsa ja laajennettavuutensa vuoksi. NIR-anturia testattiin eri tietokoneissa ja tunnistus onnistui helposti Windows 10 -käyttöjärjestelmän sisältävissä laitteissa, mutta Windows 10:tä vanhemmissa käyttöjärjestelmissä anturin tunnistaminen ei onnistunut ilman lisäohjelmia. Pienemmät ja tehottomammat tietokoneet olisivat riittäneet hyvin laitteen perustoimintoihin, mutta anturin vaatimuksien takia päädyttiin valitsemaan tietokoneeksi tehokas Windows 10 -käyttöjärjestelmäinen CX5130-0155 (kuva 4).



KUVA 4. CX5130 -sulautettu tietokone (10, s. 22)

Tietokoneen lisäksi valittiin CP6906-0001-0000 -kosketusnäyttöpaneeli ohjauspaneeliksi (kuva 5). Paneeliin pystyttiin tekemään graafinen käyttöliittymä laitteen

huoltajalle. Kosketusnäyttöpaneeli oli hyvä vaihtoehto perinteiselle painikepaneelille. Paneelilla saatiin laitteelle myös näyttävyyttä esimerkiksi laitetta esitettäessä messuilla.



KUVA 5. CP6906-0001-0000 -kosketusnäyttöpaneeli (11)

3.2.2 Tulo- ja lähtöterminaalit

Tulo- ja lähtöterminaalit valittiin laitteen tulo- ja lähtötaulukon eli I/O-taulukon mukaisesti (taulukko 1). Alun perin laitteen suunniteltiin tarvitsevan 13 digitaalista 24 V:n tuloporttia, minkä takia valittiin kaksi 8-kanavaista EL1008-tulotermiinaalia. Laitte vaati 15 lähtöporttia, mutta hydraulikkaventtiilien sisältämien 1,83 A:n ohjaukselojen takia tarvittiin viisi tehokkaampaa lähtöporttia. Yhtenä vaihtoehtona olisi ollut valita 8-kanavaiset 0,5 A:n EL2008-lähtöterminaalit ja ohjata niillä apureleitä. Apureleiden avulla 0,5 A:n virralla olisi voinut ohjata venttiilien tarvitsemää vahvempaa virtaa. Beckhoff Automationin tuotevalikoimasta löytyi 8-kanavainen 2 A antava lähtöterminaali EL2828, jolla pystyi ohjaamaan venttiileitä ilman releitä. EL2828:n avulla säästyttiin monimutkaisilta johdotuksilta. Tämän lisäksi valittiin käyttöpaneelin valoille ja muille vähemmän virtaa tarvitseville lähdöille 0,5 A antava lähtöterminaali EL2008.

TAULUKKO 1. Alkuperäinen tulo- ja lähtötaulukko

Tulot	Lähdöt
Start-painike	Venttiili 1 - Puristinruuvi +
Stop-painike	Venttiili 2 - Puristinruuvi -
Näppäimistö - Rivi 1	Venttiili 3 - Tykin kääntö +
Näppäimistö - Rivi 2	Venttiili 4 - Tykin kääntö -
Näppäimistö - Rivi 3	Venttiili 7 - Vapaakierto
Näppäimistö - Rivi 4	Lämpövastus - Rele
Koneikko - Öljyn pintahäly	Koneikko - Kontaktori
Painekeytkin	Punainen valo
Paikka 0 - Rajakytkin	Vihreä valo
Paikka 1 - Ind. anturi	Näppäimistö - Kolumni 1
Paikka 2 - Ind. anturi	Näppäimistö - Kolumni 2
Paikka 3 - Ind. anturi	Näppäimistö - Kolumni 3
Paikka 4 - Rajakytkin	Turvalukko 1
	Turvalukko 2
	Turvalukko 3
13	15

Kierrätyslaitteen ensimmäisten testausten yhteydessä huomattiin, että laitteeseen olisi hyvä tehdä eräänlainen eteen ja taakse vedettävä putki, jonka avulla voidaan säädellä tippuvaa jätettä. Putki nimettiin puristuskammiksi. Puristuskammio tarvitsee sylinterin ja suuntaventtiilin, mikä lisää lähtöjen määrän 17:ään. Alun perin suunniteltu huolto-oven mekaaninen lukko ei riittänyt turvaluokitukseen, minkä myötä siihen päädyttiin ostamaan sähköinen turvalukko. Tämä lisäsi lähtöporttien määrän jo 18:aan (taulukko 2). Näiden lisäyksien myötä ostettiin yksi EL2008-lähtöterminaali lisää. Onneksi kierrätyslaitteen asennuskoteloksi oli valittu riittävän iso kotelo, jotta lähtö- ja tuloterminaleja oli mahdollista lisätä jälkikäteen. Prototyypimäisen laitteen suunnittelussa ja valmistamisessa on hyvä varata tilaa mahdollisille muutoksille.

TAULUKKO 2. Lopullinen tulo- ja lähtötaulukko

Tulot	Lähdöt
Start-painike	Venttiili 1 - Puristinruuvi +
Stop-painike	Venttiili 2 - Puristinruuvi -
Näppäimistö - Rivi 1	Venttiili 3 - Tykin kääntö +
Näppäimistö - Rivi 2	Venttiili 4 - Tykin kääntö -
Näppäimistö - Rivi 3	Venttiili 5 - Sylinteri +
Näppäimistö - Rivi 4	Venttiili 6 - Sylinteri -
Koneikon öljyn pintahäly	Venttiili 7 - Vapaakierto
Painekeytkin	Lämpövastus - Rele
Paikka 0 - rajakytkin	Koneikko - Kontaktori
Paikka 1 - ind. anturi	Punainen valo
Paikka 2 - ind. anturi	Vihreä valo
Paikka 3 - ind. anturi	Näppäimistö -Kolumni 1
Paikka 4 - rajakytkin	Näppäimistö -Kolumni 2
	Näppäimistö -Kolumni 3
	Turvalukko 1
	Turvalukko 2
	Turvalukko 3
	Turvalukko 4
13	18

3.2.3 Turvajärjestelmä

Turvajärjestelmä on tärkein ominaisuus kaikissa koneissa ja teollisissa järjestelmissä. Standardissa SFS-EN 13849-1 on määritelty koneiden ohjausjärjestelmien yleiset suunnitteluperiaatteet koneturvallisuuden näkökulmasta (12, s. 1). Tämän standardin perusteella määriteltiin kierrätyslaitteen suoritustaso d-luokkaan. Turvatoimintojen suoritustaso, eli PL, arvioidaan a-, b-, c-, d- ja e-tasoihin, jossa a on matalin ja e korkein (13, s. 19).

Hätäpysäytys vaatii kahdennetun kytkennän, jotta se täyttää suoritustaso d:n vaatimukset. Kierrätyslaitteen ohjausjärjestelmässä täytyi olla yhteensä kolme turvatuloporttia, joista kaksi hätäpysäytykselle. Kolmas turvatuloportti tarvittiin tarkkailemaan koneikon apukosketinta EDM:n (External Device Monitoring) avulla. EDM:llä pystytään tarkkailemaan, että turvalogiikan ohjaamat koskettimet pystyvät toimimaan halutulla tavalla. Jos kontaktorin koskettimet ovat esimerkiksi sulaneet yhteen virtapiikin takia, EDM:n avulla tunnistetaan, ettei kontaktori toimi oikein, ja turvalogiikka pysäyttää laitteen. (14.)

Yksi turvalähtöportti vaadittiin sammuttamaan koneikko hätäpysäytystä käytettäessä. Näiden perusteella valittiin turvaohjaimeksi EL2911, joka täyttää SFS-EN ISO 13849-1 -standardin suoritustasoon PL e saakka ja SFS-EN ISO 61508 -

standardin SIL 3 -luokkaan asti (15). EL2911 sisältää neljä turvatuloporttia ja yhden turvalähtöportin. Turvaohjain EL2911 suunniteltiin sijoitettavaksi ennen EL2828-lähtöterminaalia. Turvaohjain syöttää jännitteen seuraaville terminaalille, jonka ansiosta hätäpysäytyspainiketta painettaessa myös hydraulikkaventtiilit sulkeutuvat.

3.3 Logiikkaohjelmointikieli

Logiikkaohjelmointi toteutettiin Beckhoffin omassa TwinCAT 3 XAE -kehitysympäristössä. TwinCAT 3 XAE on erittäin laaja kehitysympäristö, joka on integroitu Microsoftin kehittämään Visual Studio -ohjelmointikehitysympäristöön (16, s. 10). TwinCAT 3 tukee kaikkia standardin IEC 61131-3 -ohjelmointikieliä, joihin kuuluvat

- IL (Instructed List)
- FBD (Function Block Diagram)
- LD (Ladder Diagram)
- SFC (Sequential Function Chart)
- ST (Structured Text).

Näistä kielistä ST ja IL ovat tekstipohjaisia, kun taas FBD-, LD- ja SFC-kielet ovat graafisia. Tämän lisäksi TwinCAT 3 -kehitysympäristö sisältää graafisen CFC-kielen, joka ei sisälly standardiin IEC 61131-3. (17; 18.)

Ohjelmointikieleksi haluttiin valita standardiin IEC 61131-3 sisältyvä kieli, koska standardin mukaiset kielet ovat käytetyimpiä ja niiden tuki jatkuu myös tulevaisuudessa. Tämän vuoksi CFC rajautui pois heti alussa. Laitteen ohjelmointikielen tuli olla vapaasti ohjelmoitava, minkä takia askeltava SFC-kieli rajautui pois. LD-kieli rajattiin pois sen vanhanaikaisuuden takia.

Jäljelle jääneistä ohjelmointikielistä kieleksi valittiin ensin FBD-kieli, koska siitä oli jo ennestään hieman kokemusta koulussa käytyjen kurssien myötä. FBD-kieli on selkeä graafinen logiikkaohjelmointikieli, jota käytetään laajasti erilaisissa ohjausjärjestelmissä.

Selvitystyötä tehdessä kuitenkin huomattiin, että on järkevintä tehdä logiikkaohjelmointi ST-kielillä. ST-kieli on korkean tason ohjelmointikieli, mikä on lähellä

Pascal- ja C-kieltä. ST-kieli mahdollistaa esimerkiksi laskutoimitusten ja datan analysoinnin toteuttamisen helpommin verrattuna muihin IEC 61131-3 -standardin ohjelmointikieliin. (19, s. 5.) Näistä ominaisuuksista ajateltiin olevan hyötyä ohjausjärjestelmässä etenkin NIR-anturin kanssa. ST-kielen käyttöä suosittelivat myös ohjaava opettaja Juha Junntila ja Beckhoffin Oulun toimipisteen työntekijät. Kehitysympäristö mahdollistaa kuitenkin usean ohjelmointikielen käytön yhdessä projektissa, minkä takia graafista FBD-kieltä käytetään ST-kielen lisäksi helppojen ohjelmayksiköiden, kuten laskurien ja ajastimien tekemiseen. FBD-kielen funktiot toimivat myös tekstipohjaisina ST-kielessä. Tämän avulla on mahdollista käyttää aiempaa FBD-kielen kokemusta hyväksi logiikkaa ohjelmoidessa, mutta ohjelmassa saavutetaan silti ST-kielen monipuolisuus.

4 OHJAUKSEN SUUNNITTELU JA TOTEUTUS

Ohjauksen suunnittelu ja toteutus aloitettiin selkeyttämällä kierrätyslaitteen vaatimukset ja haluttu toiminta yhdessä Head Recycle Systems Oy:n toimitusjohtajan ja työntekijöiden kanssa. Ohjauksen osalta vaatimukset olivat alustavasti seuraavat:

- kaksi NIR-anturia ohjaamaan kierrätyslaitteen toimintaa
- kolme asentoa muovien pudottamiseen
- automaattinen käynnistys käyttöoven sulkeutuessa
- hätäpysäytys
- tyhjennystila, jolloin jäteastian tyhjentäminen on turvallista
- laitetta täytyy pystyä ajamaan manuaalisesti
- huoltotila, jolloin laitteen huoltaminen on turvallista.

Projektin ja suunnittelun edetessä huomattiin, ettei kyseinen ohjaus ole laitteeseen tehtyjen muutoksien jälkeen toimiva eikä se täyttäisi suoritustason PL d vaatimuksia. Laitteeseen lisätyn sylinterin myötä muovien pudotusasema siirtyi hie-
man kauemmaksi jätesäiliöstä, minkä vuoksi jouduttiin luopumaan kolmannesta pudotusasennosta. Suoritustason PL d myötä laitteeseen täytyi lisätä jokaiseen oveen sähköinen turvalukko, jotta laitteeseen ei ole mahdollista päästä käsiksi sen käytön aikana. Lisäksi laitetta saa käyttää vain sen käyttöön perehdytetyt henkilöt, jokaisen oven lähelle vaadittiin hätäpysäytyspainike ja laitteen käynnistys pitää tapahtua erillisestä painikkeesta automaattisen käynnistyksen sijasta. Ohjauksen osalta vaatimuslista muuttui seuraavaksi:

- yksi NIR-anturi ohjaamaan kierrätyslaitetta
- kaksi asentoa muovien pudottamiseen
- laitteen liikkuviin osiin ei saa päästä käsiksi laitteen ollessa käynnissä
- laitetta saa käyttää vain perehdytetyt henkilöt
- käynnistyspainike
- pysäytyspainike
- hätäpysäytys jokaisen oven lähetyville
- tyhjennystila, jolloin jäteastian tyhjentäminen on turvallista
- laitetta täytyy pystyä ajamaan manuaalisesti

- huoltotila, jolloin laitteen huoltaminen on turvallista.

4.1 Vuokaavio

Laitteen logiikan suunnittelu aloitettiin tekemällä laitteen toiminnasta vuokaavio. Vuokaavio on yleinen tapa selkeyttää prosessi tai ohjelma helpommin ymmärrettäväksi. Vuokaavio tehdään graafisena esityksenä, minkä symbolit ovat standardoitu standardissa SFS-ISO 5807. (20, s. 2–3.)

Vuokaavio tehtiin laitteen vaatimuksien ja halutun toiminnan perusteella Lucidchart web-sovelluksella. Vuokaavio jaettiin kahteen vaiheeseen. Ensimmäinen vaihe koostuu käyttäjän tekemistä alustavista toiminnoista ja toinen vaihe koostuu laitteen automaattisesta toiminnasta. Vuokaavio on liitteenä 1.

4.2 Muuttujalista

Muuttujat listattiin TwinCAT 3 -projektissa globaaliin muuttujalistaan (GVL), jotta niitä pystyttiin käyttämään helposti eri ohjelmayksiköissä. Globaali muuttuja eroaa paikallisesta muuttujasta sen perusteella, että siihen pääsee käsiksi kaikissa saman projektin sisälle luoduissa PLC-ohjelmissa (16, s. 6). Globaaliin muuttujalistaan lisättiin ensin muuttujat, jotka tiedettiin suunnitelmien ja vuokaavion perusteella. Logiikkaohjelman edetessä muuttujalista täydennettiin tarvittavilla muuttujilla.

Joitain muuttujia luotiin ohjelmien paikallisiksi muuttujiksi, koska niitä ei käytetty muissa ohjelmissa. Jos ohjelmaan luotua paikallista muuttujaa tarvitsi käyttää jossain muussa ohjelmassa, pystyi sitä kuitenkin kutsumaan toisen ohjelman nimellä (kuva 6). Muuttujat pyrittiin nimeämään mahdollisimman selkeästi ja kuvaavasti. Esimerkiksi kaikki boolean-muuttujat nimettiin aloittaen nimi b-kirjaimella ja int-muuttujat i-kirjaimella. Muuttujien nimeäminen toimintaa kuvaavasti on ohjelman luettavuuden kannalta tärkeää. Muuttujat olisi pitänyt nimetä vieläkin kuvaavammin ja selkeämmin, sillä ohjelman laajentuessa luettavuus muuttui haasteelliseksi.

```

43 //Huoltotila
44 IF Huolto.fbHuolto.Q1 THEN
45     iPutki := 5;
46 END_IF
47
48 //Tyhjennystila
49 IF Tyhjennys.fbTyhjennys.Q1 THEN
50     iPutki := 6;
51 END IF

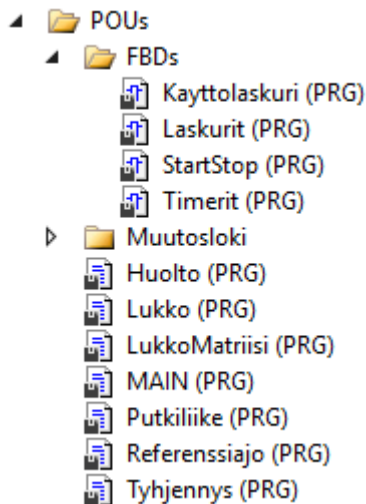
```

KUVA 6. Paikallisen muuttujan kutsuminen toisessa ohjelmassa

4.3 Laitteen ensisijainen logiikkaohjelma

Laitteen ensisijainen logiikkaohjelma tehtiin vuokaavion perusteella omaan ohjelmaan nimeltään Main. Main-ohjelmassa määritellään laitteen tärkeimmät toiminnot, kuten merkkivalon ohjaus, laitteen käynnistys, pysäytys, turvapiirin kuittaus, NIR-anturin lukeminen ja tilakone, joka ohjaa hydraulikkaventtiilien liikkeitä.

Alkuperäisenä suunnitelmana oli tehdä laitteen kaikki toiminnot Main-ohjelmaan mutta nopeasti päädyttiin jakamaan logiikkaohjelma useampaan ohjelmaan. Ohjelma jaettiin seitsemään ST-kieliseen ohjelmaan ja neljään FBD-kieliseen ohjelmaan (kuva 7), minkä avulla logiikkaohjelmasta saatiin helppolukuisempi. Varsinaiseen päälogiikkaan kuuluvat Main-ohjelman lisäksi Putkiliike- ja StartStop-ohjelmat. Selkeä ohjelmarakenne helpottaa mahdollisten ohjelmointivirheiden etsimistä ja koodin muokkaamista jälkikäteen. Main-ohjelman olisi voinut jakaa useampaankin ohjelmaan, jotta rakenteesta olisi saatu entistä selkeämpi. Kehitystyön edetessä esimerkiksi merkkivalon ohjaus tullaan muuttamaan omaksi ohjelmakseen. Nyt merkkivaloa ohjataan useasta eri ohjelmasta, minkä myötä ohjelmoinnilta vaaditaan erityistä tarkkuutta, ettei toinen ohjelma kumoa toisen ohjelman käskyjä.



KUVA 7. Ohjelmat

Main-ohjelmassa käytettiin useita FB-toimilohkoja ST-kielellä kirjoitettuna, kuten TON-ajastimia ja RS-kiikkuja. FB-toimilohkot sijoitettiin ohjelman alkuun lähelle toisiaan, jotta ne olisivat helposti löydettävissä ja ohjelmarakenne säilyisi selkeänä (kuva 8).

```

36 //LAITTEEN AJASTIN
37 fbTonRs (SET:=GVL_IO.bKansi, Reset1 := NOT GVL_IO.bKansi);
38 fbTon1 (IN:=fbTonRs.Q1, PT:=Aika2); //Ajastin kotiutukselle (ei enää käytössä)
39 fbTon2 (IN:=fbTonRs.Q1, PT:=Aikal); //Ajastin sammutukselle
40 //JOS NIR ON RIKKI/HÄIRIÖ -> AJASTIN
41 fbTon3 (IN:=fbTonRs.Q1, PT:=Aika2-T#15S);
42
43 //YLIPAINEN TRIGGAUS
44 fbYlipaine (IN:=fbYlipaineRs.Q1, PT:=T#3S);
45 fbYlipaineRs (SET:=GVL_IO.bYlipaine, RESET1:=fbYlipaine.Q);
46
47 //NIR AJASTUS
48 fbAjastusMuovi (IN:=Muovi, PT:=T#3000MS);
49 fbAjastusEiMuovi (IN:=EiMuovi, PT:=T#1000MS);
50
51 //NIRin tunnistuksien RS-kiikut
52 fbEiMuoviRs (SET:=fbAjastusEiMuovi.Q, RESET1:=PutkiLiike.fbPutkiliikeRs.Q1);
53 fbMuoviRs (SET:=fbAjastusMuovi.Q, RESET1 := PutkiLiike.fbPutkiLiikeRs.Q1);
54
55 //Liikkeen triggaukset
56 fbLiikeMyotRs (SET:=GVL_IO.bPaikka1, RESET1:= GVL_IO.bPaikka2);
57 fbLiikeVastRs (SET:=GVL_IO.bPaikka2, RESET1 := GVL_IO.bPaikka1);
58 fbLiikeKovaRaja4Rs (Set:= GVL_IO.bPaikka4, RESET1:=GVL_IO.bPaikka2 OR GVL_IO.bPaikka0);
59 fbLiikeKovaRaja0Rs (Set:= GVL_IO.bPaikka0, RESET1:=GVL_IO.bPaikka2 OR GVL_IO.bPaikka4);
60
61 //Valon ajastin
62 fbLuukunValoTon (IN:=fbLuukunValoRs.Q1, PT:=T#15S);
63 fbLuukunValoRs (SET:=Lukko.bKoodiOK, RESET1 := fbLuukunValoTon.Q);

```

KUVA 8. FB-toimilohkojen käyttöä

Main-ohjelman varsinainen toiminta koostuu kahdesta IF-lauseesta ja yhdestä ti-lakoneesta. Ensimmäinen IF-lause on hyvin yksinkertainen ja lyhyt, ja sillä ohjataan käyttöluukun avaamista. Käyttöluukun sähkölukko avataan numerokoodilla,

joka näppäillään näppäimistöllä. Numerokoodi kirjoittuu Lukko-ohjelmaan. Numerokoodin ollessa oikein sähkölukko avautuu. Ehtona on, etteivät huolto- tai tyhjennystila ole käynnissä. Käyttäjälle ilmaistaan oven avautuminen vihreällä valolla, jonka lisäksi apuvalaisin syttyy. Apuvalaisin valaisee käyttöluukun sisäpuolen, mikä helpottaa muovien laittamista kierrätyslaitteeseen. (Kuva 9.)

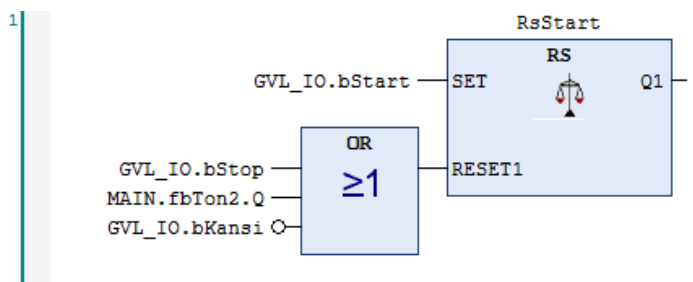
```

68 //AVATAAN LUKKO KOODILLA
69 IF Lukko.bKoodiOK AND NOT Tyhjennys.fbTyhjennys.Q1 AND NOT Huolto.fbHuolto.Q1 THEN
70     GVL_IO.bKansi := FALSE;
71     GVL_IO.bVihr := TRUE;
72     GVL_IO.bLuukunValo := TRUE;
73 ELSE
74     GVL_IO.bKansi := TRUE;
75 END_IF
76
77 IF fbLuukunValoTon.Q THEN
78     GVL_IO.bLuukunValo := FALSE;
79 END IF

```

KUVA 9. Käyttöluukun lukon avaus

Laitteen käynnistyksestä ja pysäytyksestä haluttiin mahdollisimman yksinkertainen ja helposti löydettävä, minkä takia siitä päätettiin luoda oma ohjelma FBD-kielellä (kuva 10). StartStop-ohjelmaan on luotu yksinkertainen RS-kiikku RsStart jota kutsutaan Main-ohjelmassa. RS-kiikun voi toteuttaa myös ST-kielellä, jolloin sen olisi voinut tehdä suoraan Main-ohjelmaan. ST-kielellä tehdyt RS-kiikut ovat vain yhden rivin pituisia, minkä takia ne saattavat olla vaikeita löytää laajasta ST-ohjelmasta.



```

55 RsStart(SET:=GVL_IO.bStart, RESET1 := GVL_IO.bStop OR NOT MAIN.fbTon2.Q OR NOT GVL_IO.bKansi);

```

KUVA 10. Laitteen ohjauksen RS-kiikku FBD- ja ST-kielellä

Ohjelman toisessa IF-lauseessa käynnistetään laitteen automaattinen sykli ja ohjataan tilakoneetta iState-muuttujan arvoilla. Tilakone on ohjelmointimalli, jossa edetään vaihe kerrallaan. Tietyn vaiheen ollessa valmis siirrytään seuraavaan tilaan.

Laitteen automaattisen syklin käynnistämiseen ja käynnissä pysymiseen täytyy täyttää tietyt ehdot. StartStop-ohjelman RsStart-pitopiirin on oltava tosi, minkä lisäksi käyttöluukku täytyy olla suljettuna ja fbTon2-ajastin täytyy olla kuitattu. Ajastin kuitataan avaamalla käyttöluukku numerokoodilla. Tämän takia laitetta ei pääse käyttämään ilman numerokoodia. Lisäksi huoltotila eikä tyhjennystila saa olla käynnissä. Laite ei käynnisty, jos jokin näistä ehdoista ei täyty, tai laite sammuu, jos jokin ehto muuttuu koneen käynnissä ollessa. Kun ehdot täyttyvät, hydraulikkakoneikko käynnistyy ja keltainen valo syttyy. (Kuva 11.)

```
80 //SYKLIN KÄYNNISTYS & NIRIN LUKEMINEN
81 IF StartStop.RsStart.Q1 AND GVL_IO.bKansi AND NOT fbTon2.Q AND NOT Tyhjennys.fbTyhjennys.Q1 AND NOT Huolto.fbHuolto.Q1 THEN
82
83     GVL_IO.bKoneikko := TRUE; //Koneikko päälle
84     GVL_IO.bKelt := TRUE; //Keltainen valo päälle
85
86     iState := 0; //TILAA 0 = PURISTUSPUTKI AJETAAN KESKIASENTOON JA AUGER KÄYNNISTETÄÄN
87
88     IF fbYlipaineRs.Q1 THEN //JOS YLIPAINE PURISTUKSESSA
89         iState := 15; //TILAA 15 = PYSÄYTTÄÄ AUGERIN JA AVAA PURISTUSKAMMION
90     END_IF
91
92     //NIRIN LUKU JA SILLÄ OHJAUS
93     IF rNIR1 >= [ ] AND rNIR1 <= [ ] THEN
94         EiMuovi := TRUE;
95         Muovi := FALSE;
96
97     ELSE EiMuovi := FALSE;
98     END_IF
99
100     IF fbEiMuoviRs.Q1 THEN
101         iState := 1; //TILAA 1 = JOSSA PURISTUSPUTKI LIIKUTETAAN PAIKKAAN 1
102         EiMuoviLaskuri := TRUE; //LASKURIIN +1
103         MuoviLaskuri := FALSE;
104     END_IF
```

KUVA 11. Main-ohjelman automaattisen syklin IF-lause

Toisen IF-lauseen sisällä on useita IF-lauseita, joiden avulla muutetaan iState-muuttujan arvoa. Arvoa muutetaan esimerkiksi NIR-anturin antamien tulosten perusteella. Muuttuja iStaten arvo muuttaa tilakoneen tilaa, jolla ohjataan kierrätyslaitteen hydraulikkaventtiileitä. Hydraulikkaventtiileillä ohjataan puristusputken, puristuskammion ja puristusruuvien liikkeitä. Tilakoneessa on myös tilat laitteen turvalliseen huoltotapahtumaan ja tyhjennykseen.

Kun NIR-anturi tunnistaa muovinkierrätykseen kuulumatonta jätettä, iState-muuttuja saa kokonaislukuarvon yksi. iState-muuttujan kokonaislukuarvo määrää tilakoneen tilan. Tilassa yksi puristusputki ajetaan vastapäivään sekajäteastian kohdalle. Sekajäteastian sijainti on nimetty PLC-ohjelmassa globaaliksi muuttujaksi bPaikka1 (kuva 12). Kun oikea asema on saavutettu, puristusputken liike pysähtyy ja globaali muuttuja bPutkiLiike käynnistää Putkiliike-ohjelmassa olevan tilakoneen, jolla pudotetaan sekajäte puristuskammioista.

```

192      1: //EI MUOVIA
193
194      GVL_IO.bPutkiLiike := FALSE;
195
196      IF GVL_IO.bPaikkal THEN //TÄMÄ ON OIKEA PAIKKA(SEKAJÄTE)
197          GVL_IO.bTykkiMyot := FALSE;
198          GVL_IO.bTykkiVast := FALSE;
199          GVL_IO.bPutkiLiike := TRUE;
200      END_IF
201
202      IF NOT Putkiliike.fbPutkiliikeRs.Q1 THEN
203          GVL_IO.bTykkiVast := TRUE;
204          GVL_IO.bTykkiMyot := FALSE;
205          GVL_IO.bAugerMyot := FALSE;
206          GVL_IO.bPutkiLiike := FALSE;
207
208          IF GVL_IO.bPaikkal THEN //TÄMÄ ON OIKEA PAIKKA
209              GVL_IO.bTykkiMyot := FALSE;
210              GVL_IO.bTykkiVast := FALSE;
211              GVL_IO.bPutkiLiike := TRUE;
212
213          ELSE GVL_IO.bPutkiLiike := FALSE;
214          END_IF
215
216          IF GVL_io.bPaikka2 THEN
217              GVL_IO.bTykkiMyot := FALSE;
218              GVL_IO.bTykkiVast := TRUE;
219          END_IF
220
221      END_IF
222
223      //JOS KOVARAJA KYTKEKYTTY
224      IF fbLiikeKovaRaja0Rs.Q1 THEN
225          GVL_IO.bTykkiMyot := TRUE;
226          GVL_IO.bTykkiVast := FALSE;
227          GVL_IO.bVapaaKierto := TRUE;
228      END_IF
229      IF fbLiikeKovaRaja4Rs.Q1 THEN
230          GVL_IO.bTykkiMyot := FALSE;
231          GVL_IO.bTykkiVast := TRUE;
232          GVL_IO.bVapaaKierto := TRUE;
233      END_IF

```

KUVA 12. Tilakoneen ensimmäinen tila

Main-ohjelman tilakoneessa on myös ylipainetila, jolloin puristuksessa tapahtuva jumiutuminen käynnistää Putkiliike-ohjelman. Putkiliike-ohjelma avaa puristuskammion ja pyrkii poistamaan tukoksen. Tilakoneessa olevat huolto- ja tyhjennystilat mahdollistavat turvalliset huolto- ja tyhjennystapahtumat. Lisäksi tilakoneessa on tila laitteen sammutukselle, jolla saadaan myös kaikki hydraulikka-venttiilit palautettua nolla-asentoihin.

Putkiliike-ohjelmassa ohjataan puristuskammion liikettä automaattisesti tilakoneen avulla. Tilakoneen tiloja ohjataan RS-kiikuilla ja TON-ajastimilla. Ohjelman tavoitteena on pudottaa tunnistettu muovi tai muu jäte puristuskammion. Puristuskammion liike käynnistyy, kun globaali muuttuja bPutkiLiike muuttuu todeksi. Laitteen täytyy olla käynnissä, eikä se saa olla huolto tai tyhjennystilassa, jotta automaattinen liike onnistuu. Putkiliike-ohjelman tilakone tekee seuraavan toiminnon:

- puristuskammio avautuu
- kun kammio on auki, puristusruuvi pyörii vastapäivään
- tietyn ajan kuluttua puristusruuvien pyörimissuunta vaihtuu
- pyöriminen loppuu ja puristuskammio sulkeutuu. (Kuva 13.)

```

CASE iPutki OF

0 : //PUTKI AUKAISTAAN (SYLINTERI SISÄÄB)
  GVL_IO . bSylSis := TRUE ;
  GVL_IO . bSylUlos := FALSE ;
  GVL_IO . bVapaaKierto := TRUE ;

  GVL_IO . bAugerMyot := FALSE ;
  GVL_IO . bAugerVast := FALSE ;

1 : //PYÖRITETÄÄN AUGERIA VASTAPÄIVÄÄN (SYLINTERI SISÄLLÄ)
  GVL_IO . bSylSis := FALSE ;
  GVL_IO . bSylUlos := FALSE ;
  GVL_IO . bVapaaKierto := TRUE ;

  GVL_IO . bAugerMyot := FALSE ;
  GVL_IO . bAugerVast := TRUE ;      //Pyöritetään augeria vastapäivään
hetki

2 : //PYÖRITETÄÄN AUGERIA MYÖTÄPÄIVÄÄN (SYLINTERI SISÄLLÄ)

  GVL_IO . bAugerMyot := TRUE ;      //Pyöritetään augeria myötäpäivään
  GVL_IO . bAugerVast := FALSE ;

  GVL_IO . bSylSis := FALSE ;
  GVL_IO . bSylUlos := FALSE ;
  GVL_IO . bVapaaKierto := TRUE ;

3 : //PUTKI KIINNI (SYLINTERI ULOS)

  GVL_IO . bSylSis := FALSE ;
  GVL_IO . bSylUlos := TRUE ;
  GVL_IO . bVapaaKierto := TRUE ;

  GVL_IO . bAugerMyot := FALSE ;
  GVL_IO . bAugerVast := FALSE ;

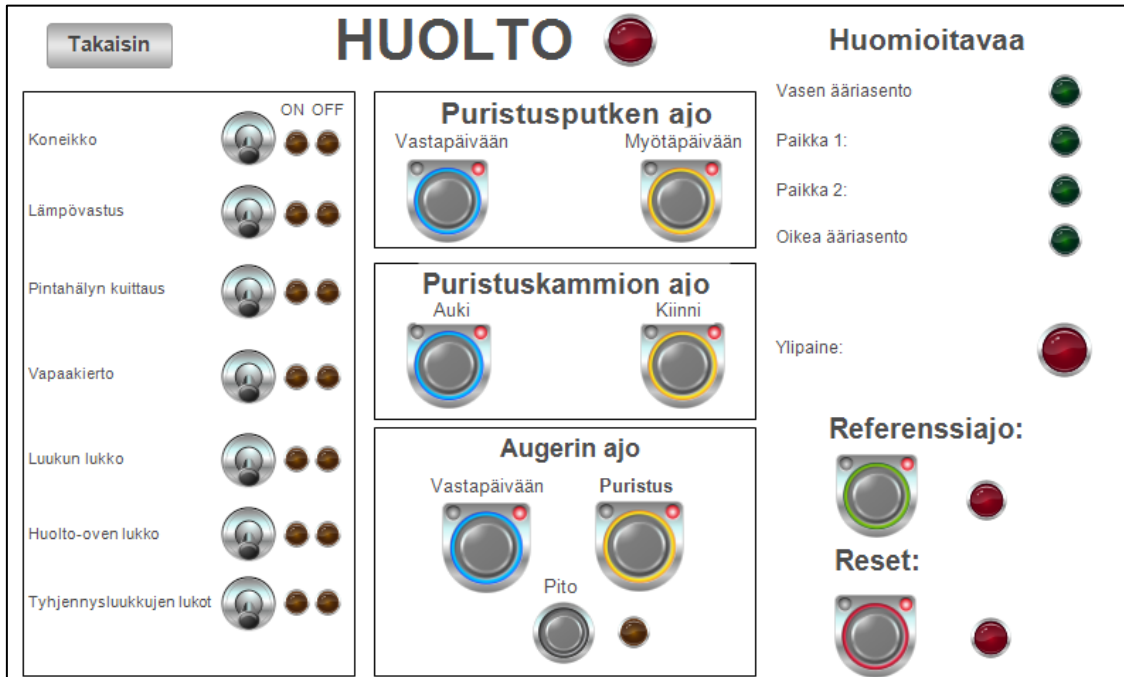
```

KUVA 13. Putkiliike-ohjelman tilakone

4.4 Huollon logiikkaohjelma

Huollon logiikkaohjelmaan haluttiin toimintoja, jotka avustavat huoltohenkilöä laitteen huollon aikana. Tärkein toiminto on huollon suorittamisen turvallisuus, joten laite ei saa käynnistyä vahingossa huoltotilan ollessa päällä. Tämä toteutettiin

muuttamalla kaikkien ohjelmien tilakoneet huoltotilaan, jolloin laitteen käyttäminen on mahdollista vain huollon kautta. Logiikkaohjelman lisäksi toteutettiin kosketusnäyttöpaneelille graafinen käyttöliittymä, joka oli opinnäytetyön yksi tavoitteista. Käyttöliittymällä voi ohjata laitetta huollon aikana manuaalisesti (kuva 14).



KUVA 14. Laitteen graafinen huoltokäyttöliittymä

Huollon graafisesta käyttöliittymästä tehtiin mahdollisimman yksinkertainen, koska kosketusnäyttöpaneeli oli vain 7-tuumainen. Näyttöön haluttiin pienestä koosta huolimatta ohjauspainikkeet puristusputkelle, puristuskammiolle ja puristusruuville. Näiden lisäksi haluttiin ohjata manuaalisesti koneikkoa, koneikon lämpövastusta, vapaakiertoa ja kaikkia sähkölukkoja. Painikkeet tarvittiin myös koneikon öljypinnan aiheuttaman hälytyksen kuittaukselle ja referenssiajolle. Reset-painikkeella on mahdollista palauttaa venttiilit nolla-asentoihinsa manuaalisesti. Painikkeiden lisäksi käyttöliittymään asetettiin ja ohjelmoitiin erilaisia huomiovaloja huollon helpottamiseksi.

Huollon logiikkaohjelma rakennettiin pääasiassa IF-lauseen avulla, koska sen toiminta on kohtuullisen yksinkertainen. Huoltotila käynnistetään syöttämällä huol-

tokoodi näppäimistöllä tai poikkeustilanteessa painamalla huoltopainiketta kosketusnäyttöpaneelista (kuva 15), mikä muuttaa globaalin muuttujan bHuolto todeksi.



KUVA 15. Käyttöliittymän aloitusnäky

Globaali muuttuja bHuolto kytkee huollon pitopiirin fbHuolto päälle. Kun fbHuoltopitopiiri on päällä, laitteen kaikki toiminnot pysähtyvät ja vain laitteen manuaalinen käyttö on mahdollista. Pitopiiri fbHuolto asettaa ohjelman kaikki tilakoneet huoltotilaan. Huoltotilan aloitustilanteessa käyttöluukun ja huolto-oven lukot avautuvat

IF-lauseen sisään tehtiin useita IF-lauseita (kuva 16), joilla pystytään luomaan poikkeuksia, jotka sallivat ensimmäisen IF-lauseen sisällä olevien muuttujien käytön, vaikka fbHuolto.Q1 olisikin tosi. Käyttöliittymän virtuaalinapeille luotiin omat boolean-muuttujat, joilla ohjattiin fyysiseen PLC-laitteeseen linkitettyjä muuttujia.

```

fbHuolto(SET:=GVL_IO.bHuolto, RESET:=GVL_IO.bKuittaus); //Huoltomoodi päälle, kuittaus kuittausnapilla sekä huoltonapilla

IF fbHuolto.Q1 THEN

    GVL_IO.bHuoltoLukko := FALSE;
    GVL_IO.bHuoltoValo := TRUE;

    GVL_IO.bKoneikko := FALSE;
    GVL_IO.bKannenLukko := FALSE;
    GVL_IO.bAugerMyot := FALSE;
    GVL_IO.bAugerVast := FALSE;
    GVL_IO.bTykkiMyot := FALSE;
    GVL_IO.bTykkiVast := FALSE;
    GVL_IO.bVapaaKierto := FALSE;
    GVL_IO.bSylSis := FALSE;
    GVL_IO.bSylUlos := FALSE;

    //GVL_IO.bOikeaTyhj := TRUE;
    //GVL_IO.bVasenTyhj := TRUE;

    //GVL_IO.bAugerMyot := FALSE;

    GVL_IO.bVihr := FALSE;

    IF bPunValo THEN //Merkkivalon ohjauslause
        GVL_IO.bPun := TRUE;
    ELSE GVL_IO.bPun := FALSE;
    END_IF

    IF GVL_IO.bKoneikkoOn THEN //Koneikon painike käyttöliittymässä
        GVL_IO.bKoneikko := TRUE;
    ELSE
        GVL_IO.bKoneikko := FALSE;
    END_IF

    IF GVL_IO.bHuoltoAjoMyot THEN //Tykin ajo myötöpäivään käyttöliittymästä
        GVL_IO.bTykkiMyot := TRUE;

```

KUVA 16. Ote huollon logiikkaohjelmasta

Jotta laitteen normaalikäyttö on taas mahdollista, täytyy huoltotilasta poistua. Huoltotilasta poistutaan kuittaamalla pitopiiri fbHuolto. Huoltotila kuitataan painamalla käyttöliittymän aloitussivun kuittauspainiketta tai syöttämällä näppäimistöllä huollon kuitauskoodi. Nämä toimenpiteet muuttavat globaalin muuttujan bKuittaus todeksi, joka resetoit fbHuolto-pitopiirin.

4.5 Matriisinäppäimistön integroiminen

Turvallisuusvaatimusten takia laitteen käyttö vaatii perehdytyksen, minkä myötä laitteessa täytyy olla lukitus. Laitteeseen mietittiin aluksi perinteistä lukkopesää ja avainta jokaiselle käyttäjälle. Koska laitteeseen oli jo hankittu suoritusaso PL d:n vuoksi sähköiset lukot jokaiseen oveen, päätettiin näitä käyttää hyväksi myös tässä tapauksessa. Asiakkaan puolelta tuli samaan aikaan toive koodilukosta, jonka saisi avattua asuntokohtaisella koodilla. Laitteeseen päätettiin hankkia Apem CCM 12 TR -matriisinäppäimistö (kuva 17).



KUVA 17. Apem CCM 12 TR -matriisinäppäimistö (21)

Näppäimistössä oli seitsemännapainen liitin. Navat 1–3 olivat pystyriveille ja navat 4–7 vaakariveille (kuva 18). Aluksi oletettiin, että näppäimistössä olisi ollut ylimääräinen liitinnapa virralle, jolloin pysty- ja vaakarivien navat olisi voitu kytkeä suoraan PLC:n tuloihin. Tässä tapauksessa yhtä näppäintä painettaessa kaksi tuloa olisi sammunut, jolloin jokainen näppäin olisi saatu luettua yksinkertaisella logiikkakoodilla.



KUVA 18. Näppäimistön kytkentämatriisi (22, s. 4)

Koska näppäimistössä ei ollut erillistä liitinnapaa virralle, jouduttiin logiikkaohjelma tekemään huomattavasti haastavammalla tavalla. Laajan selvitystyön jäl-

keen selvisi, että paras tapa on kytkeä pystyrivit PLC:n lähtöportteihin ja vaakarivit PLC:n tuloportteihin. Pystyrivien lähtöporttien päällä oloa vaihdellen ja näppäintä painamalla saadaan virta kulkemaan PLC:n tuloporttiin. Esimerkiksi kun virta tulee liittimen napaan 1 ja näppäintä 7 painetaan, virta kulkee liittinnavan 5 kautta PLC:n tuloporttiin. Näin saadaan näppäin 7 luettua omaksi näppäimeksi. (Kuva 19.)

```
67 //Määritetään näppäinarvot
68 IF GVL_IO.bColl AND GVL_IO.bRow1 THEN
69     GVL_IO.b1 := TRUE;
70 ELSE GVL_IO.b1 := FALSE;
71 END_IF
```

KUVA 19. Näppäinten määrittely

Lähtöporttien vaihtosykli toteutettiin tilakoneella ja TON-ajastimilla. Näppäimistön jokaiselle pystyriville, eli lähtöportille, luotiin oma tila, jonka sisällä määriteltiin näppäinten arvot. Lähtöporttien vaihtelu täytyi olla erittäin nopea, jotta oikea numero kirjoittui riittävän nopeasti numerokoodiin. Ensimmäisessä versiossa vaihteluaika oli 100 millisekuntia. 100 millisekunnin vaihteluvälillä näppäimen tunnistamiseen saattoi kulua huonoimmassa tapauksessa 300 millisekuntia. Vaikka 300 millisekuntia tuntui nopealta ajalta, testien jälkeen huomattiin, ettei se ole riittävän nopea. Käytettävyyden kannalta riittävän nopeaksi vaihteluajaksi osoitautui seitsemän millisekuntia. Tilojen vaihtojen väliin lisättiin viiveeksi viisi millisekuntia, koska ilman viivettä koodiin kirjoittui välillä väärä numero. (Kuva 20.)

```

1 //Ajastin jolla vaihdetaan pystyriivi (lähtö) jolle virta syötetään 43
2 TimerTon1(IN:=GVL_IO.bCol1, PT:=T#7MS); 44
3 TimerTon2(IN:=GVL_IO.bCol2, PT:=T#7MS); 45
4 TimerTon3(IN:=GVL_IO.bCol3, PT:=T#7MS); 46
5 47
6 //Viiveet 48
7 TimerTonViive1(IN:=Ton1Rs.Q1, PT:=T#5MS); 49
8 TimerTonViive2(IN:=Ton2Rs.Q1, PT:=T#5MS); 50
9 TimerTonViive3(IN:=Ton3Rs.Q1, PT:=T#5MS); 51
10 52
11 //Pitopiirit 53
12 Ton1Rs(SET:=TimerTon1.Q, RESET1:=TimerTonViive1.Q); 54
13 Ton2Rs(SET:=TimerTon2.Q, RESET1:=TimerTonViive2.Q); 55
14 Ton3Rs(SET:=TimerTon3.Q, RESET1:=TimerTonViive3.Q); 56
15 57
16 IF TimerTon1.Q THEN 58
17     iState := 1; //Pystyriivin 1 näppäinten määrittely 59
18 END_IF 60
19 61
20 IF TimerTonViive1.Q THEN 62
21     iState := 2; //Viive 63
22 END_IF 64
23 65
24 IF TimerTon2.Q THEN 66
25     iState := 3; //Pystyriivi 2 67
26 END_IF 68
27 69
28 IF TimerTonViive2.Q THEN 70
29     iState := 4; //Viive 71
30 END_IF 72
31 73
32 IF TimerTon3.Q THEN 74
33     iState := 5; //Pystyriivi 3
34 END_IF
35
36 IF TimerTonViive3.Q THEN
37     iState := 0; //Viive
38 END_IF

```

```

CASE iState OF
0:
    GVL_IO.bCol1 := TRUE; //Pystyriivi 1 tosi
    GVL_IO.bCol2 := FALSE; //Pystyriivi 2 false
    GVL_IO.bCol3 := FALSE; //Pystyriivi 3 false

    //Määritetään näppäinarvot
    IF GVL_IO.bCol1 AND GVL_IO.bRow1 THEN
        GVL_IO.b1 := TRUE;
    ELSE GVL_IO.b1 := FALSE;
    END_IF

    IF GVL_IO.bCol1 AND GVL_IO.bRow2 THEN
        GVL_IO.b4 := TRUE;
    ELSE GVL_IO.b4 := FALSE;
    END_IF

    IF GVL_IO.bCol1 AND GVL_IO.bRow3 THEN
        GVL_IO.b7 := TRUE;
    ELSE GVL_IO.b7 := FALSE;
    END_IF

    IF GVL_IO.bCol1 AND GVL_IO.bRow4 THEN
        GVL_IO.bCancel := TRUE;
    ELSE GVL_IO.bCancel := FALSE;
    END_IF

1: GVL_IO.bCol1 := FALSE;
   GVL_IO.bCol2 := FALSE;
   GVL_IO.bCol3 := FALSE;

```

KUVA 20. Tilojen vaihto ajastimien avulla ja tilakoneen 0-tila

Numerokoodin kirjoitukseen luotiin erillinen ohjelma, jossa koodi kirjoitetaan näppäinten painalluksien perusteella string-tietotyyppinä eli merkkijonona. TwinCAT 3 sisältää TC2_standard -kirjastossa string-funktiot, joiden avulla on mahdollista muokata merkkijonoja (23).

Numerokoodin lähtöarvoksi asetettiin 'xxxx', jota muokattiin replace-komennolla. Replace-komennolla on mahdollista korvata osa merkkijonosta toisella merkkijonolla (24). Haluttu numero kirjoitettiin oikealle paikalle CTU-laskurin arvon perusteella. Jokainen näppäimistön painallus lisää laskuriin luvun yksi, minkä avulla koodi saadaan kirjoitettua kokonaisuudessaan. Lisäksi logiikkakoodiin lisättiin TON-ajastin, joka varmistaa, että yhdellä painalluksella kirjoitetaan vain yksi kirjain koodiin. Näppäimistön ohjelmointi vaati paljon testailua ja eri versioiden ohjelmointia, minkä takia muuttujien nimeäminen ei pysynyt enää kuvaavana. Esimerkkinä voidaan pitää TON-muuttujaa fbTestiTon, joka ei kuvaa ajastimen tarkoitusta ollenkaan. (Kuva 21.)

```

20 //Kun laskuri on 0, määritetään koodin ensimmäinen numero JA kun nappi on pohjassa
21 IF Laskurit.LaskuriLukko.CV = 0 AND NOT fbTestiTon.Q THEN
22
23     bNoNappi := FALSE;
24
25     IF GVL_IO.b0 THEN Koodi := REPLACE(Koodi,s0,1,1); Laskurit.LaskuriLukko.CV := TRUE; //Kirjoittaa luvun 0 paikkaan 1
26     ELSIF GVL_IO.b1 THEN Koodi := REPLACE(Koodi,s1,1,1); Laskurit.LaskuriLukko.CV := TRUE; //Kirjoittaa luvun 1 paikkaan 1
27     ELSIF GVL_IO.b2 THEN Koodi := REPLACE(Koodi,s2,1,1); Laskurit.LaskuriLukko.CV := TRUE; //Kirjoittaa luvun 2 paikkaan 1
28     ELSIF GVL_IO.b3 THEN Koodi := REPLACE(Koodi,s3,1,1); Laskurit.LaskuriLukko.CV := TRUE; //Kirjoittaa luvun 3 paikkaan 1
29     ELSIF GVL_IO.b4 THEN Koodi := REPLACE(Koodi,s4,1,1); Laskurit.LaskuriLukko.CV := TRUE; //Kirjoittaa luvun 4 paikkaan 1
30     ELSIF GVL_IO.b5 THEN Koodi := REPLACE(Koodi,s5,1,1); Laskurit.LaskuriLukko.CV := TRUE; //Kirjoittaa luvun 5 paikkaan 1
31     ELSIF GVL_IO.b6 THEN Koodi := REPLACE(Koodi,s6,1,1); Laskurit.LaskuriLukko.CV := TRUE; //Kirjoittaa luvun 6 paikkaan 1
32     ELSIF GVL_IO.b7 THEN Koodi := REPLACE(Koodi,s7,1,1); Laskurit.LaskuriLukko.CV := TRUE; //Kirjoittaa luvun 7 paikkaan 1
33     ELSIF GVL_IO.b8 THEN Koodi := REPLACE(Koodi,s8,1,1); Laskurit.LaskuriLukko.CV := TRUE; //Kirjoittaa luvun 8 paikkaan 1
34     ELSIF GVL_IO.b9 THEN Koodi := REPLACE(Koodi,s9,1,1); Laskurit.LaskuriLukko.CV := TRUE; //Kirjoittaa luvun 9 paikkaan 1
35
36     ELSE Laskurit.LaskuriLukko.CV := FALSE;
37     END_IF
38
39 END_IF
40
41 //Kun laskuri on 1, määritetään koodin toinen numero JA kun mikään nappi ei ole ollut pohjassa tietyn ajan
42 IF Laskurit.LaskuriLukko.CV = 1 AND fbTestiTon.Q THEN
43
44     Koodi := Replace(Koodi, 'xxx',3,2); //Poistaa mahdolliset ylimääräiset numerot aiemmasta painalluksesta
45
46     IF GVL_IO.b0 THEN Koodi := REPLACE(Koodi,s0,1,2); Laskurit.LaskuriLukko.CV := TRUE; //Kirjoittaa luvun 0 paikkaan 2
47     ELSIF GVL_IO.b1 THEN Koodi := REPLACE(Koodi,s1,1,2); Laskurit.LaskuriLukko.CV := TRUE; //Kirjoittaa luvun 1 paikkaan 2
48     ELSIF GVL_IO.b2 THEN Koodi := REPLACE(Koodi,s2,1,2); Laskurit.LaskuriLukko.CV := TRUE; //Kirjoittaa luvun 2 paikkaan 2
49     ELSIF GVL_IO.b3 THEN Koodi := REPLACE(Koodi,s3,1,2); Laskurit.LaskuriLukko.CV := TRUE; //Kirjoittaa luvun 3 paikkaan 2
50     ELSIF GVL_IO.b4 THEN Koodi := REPLACE(Koodi,s4,1,2); Laskurit.LaskuriLukko.CV := TRUE; //Kirjoittaa luvun 4 paikkaan 2
51     ELSIF GVL_IO.b5 THEN Koodi := REPLACE(Koodi,s5,1,2); Laskurit.LaskuriLukko.CV := TRUE; //Kirjoittaa luvun 5 paikkaan 2
52     ELSIF GVL_IO.b6 THEN Koodi := REPLACE(Koodi,s6,1,2); Laskurit.LaskuriLukko.CV := TRUE; //Kirjoittaa luvun 6 paikkaan 2
53     ELSIF GVL_IO.b7 THEN Koodi := REPLACE(Koodi,s7,1,2); Laskurit.LaskuriLukko.CV := TRUE; //Kirjoittaa luvun 7 paikkaan 2
54     ELSIF GVL_IO.b8 THEN Koodi := REPLACE(Koodi,s8,1,2); Laskurit.LaskuriLukko.CV := TRUE; //Kirjoittaa luvun 8 paikkaan 2
55     ELSIF GVL_IO.b9 THEN Koodi := REPLACE(Koodi,s9,1,2); Laskurit.LaskuriLukko.CV := TRUE; //Kirjoittaa luvun 9 paikkaan 2
56
57     ELSE Laskurit.LaskuriLukko.CV := FALSE;
58
59     END_IF
60 END_IF

```

KUVA 21. Numerokoodin kirjoitus ST-kielillä

Numerokoodin ollessa oikein ja OK-näppäintä painettaessa, bKoodiOK-muuttuja muuttuu todeksi. Tämä muuttaa Main-ohjelmassa globaalin muuttujan bKansi epätodeksi, joka on linkitetty käyttöluukun sähköluukoon. Tällä tavoin sähköluukko avautuu ja käyttäjä voi syöttää muovit luukusta sisään. Numerokoodin avulla saadaan myös käynnistettyä huoltotila tai tyhjennystila. (Kuva 22.)

```

100 //Kun koodi on valmis ja oikein, painamalla OK painiketta saadaan lukko avattua.
101 IF GVL_IO.bOK THEN
102
103     GVL_IO.bVihr := TRUE;
104
105     IF Koodi = '███' OR Koodi = '███' OR Koodi = '███' OR Koodi = '███'
106
107     bKoodiOK := TRUE;
108
109     END_IF
110
111     IF Koodi = '███' THEN //HUOLLON KOODI -> KÄYNNISTÄÄ HUOLLON
112         GVL_IO.bHuolto := TRUE;
113         Laskurit.HuoltoKoodi.CU := TRUE;
114     ELSE GVL_IO.bHuolto := FALSE;
115     END_IF
116
117
118     IF Koodi = '███' THEN //HUOLLON KUITTAUS
119         GVL_IO.bHuolto := FALSE;
120         GVL_IO.bKuittaus := TRUE;
121     END_IF
122
123     IF Koodi = '███' THEN //TYHJENNYKSEN KOODI -> KÄYNNISTÄÄ TYHJENNYKSEN
124         GVL_IO.bTyhjennys := TRUE;
125     ELSE GVL_IO.bTyhjennys := FALSE;
126     END_IF

```

KUVA 22. Numerokodeilla ohjaus

Matriisinäppäimistö asennettiin käyttöpaneeliin. Käyttöpaneelissa on matriisinäppäimistön lisäksi merkkivalo, käynnistys- ja pysäytyspainike. Käyttöpaneeli muodostaa sen yläpuolella olevan hätäpysäytyspainikkeen kanssa fyysisen käyttöliittymän. Fyysisen käyttöliittymän suunnittelu ja toteuttaminen oli yksi opinnäytetyön tavoitteista. (Kuva 23.)



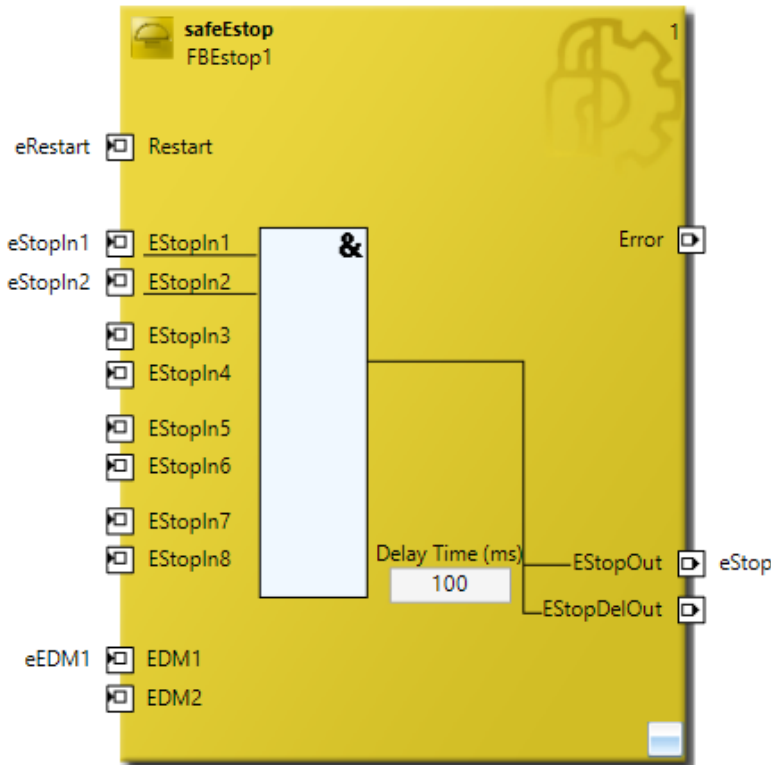
KUVA 23. Laitteen fyysinen käyttöliittymä

4.6 Turvajärjestelmän logiikkaohjelma

Turvaohjelman ohjelmointi poikkeaa perinteisestä PLC-ohjelmoinnista. Turvaohjelma ohjelmoidaan TwinCAT 3 XAE-kehitysympäristössä TwinSAFE-projektiin, joka on turvajärjestelmän äly. Logiikkaohjelma ohjelmoidaan sertifioidulla turvaohjelmointikielillä, joka muistuttaa FBD-kieltä. Turvaohjelmointikieli sisältää muun muassa hätäpysäytysfunktion, jolla voidaan luoda laitteen hätäpysäytys. Logiikkaohjelman ollessa valmis siirretään se fyysisessä PLC-järjestelmässä olevaan TwinSAFE-komponenttiin, jossa se pyörii itsenäisenä ohjelmana erillään varsinaisesta PLC-ohjelmasta. (25, s. 12.)

Laitteen turvaohjelmaa suunniteltiin pitkään. Laite tarvitsi jokaisen oven ja luukun läheisyyteen hätäpysäytyspainikkeen. Jokaisessa ovesa oli lisäksi turvaluokitellut sähkölukot. Koska ovien avautuessa laitteen haluttiin pysähtyvän, oli sähkölukkoja mahdollista käyttää hätäpysäytyksen kaltaisesti. Sähkölukkojen kielten ja lukituksen seurannan NC-koskettimet ketjutettiin hätäpysäytyspainikkeen NC-koskettimien kanssa. Tällä tavalla laite tulkitsee hätäpysäytyksen olevan päällä,

ja laite sammuu, jos mikä tahansa ovi tai luukku on avoinna. Ketjutettu hätäpysäytyspiiri kytkettiin EL2911-turvalogiikan tuloihin 1 ja 2. Tämän ansiosta pystyttiin ohjelmoimaan yksinkertainen hätäpysäytyslogiikka FBESTop1 (kuva 24).



KUVA 24. Laitteen hätäpysäytyslogiikka

Hätäpysäytyslogiikan muuttujat eStopIn1 ja eStopIn2 ovat hätäpysäytyspiirin ketjutetut koskettimet. Muuttuja eStopIn1 sisältää hätäpysäytyspainikkeen toisen koskettimen ja sähkölukkojen kielten seurannan koskettimet. Muuttuja eStopIn2 sisältää hätäpysäytyspainikkeen toisen koskettimen ja sähkölukkojen lukituksen seurannan koskettimet. Hätäpysäytyspiirin täytyi olla kahdennettu, jotta laite täyttää suoritustaso PL d:n vaatimukset. Logiikan eEDM1-muuttuja on kytketty hydraulikkakoneikkoa ohjaavan kontaktorin apukoskettimeen, mikä tarkkailee, ettei kontaktorin koskettimet ole esimerkiksi sulaneet yhteen. Muuttuja eRestart on turvapiirin kuittaamista varten. Jos hätäpysäytyspiiri katkeaa, täytyy piiri kuitata ennen kuin laitetta on mahdollista taas käyttää. Muuttuja eStop on kytketty hydraulikkakoneikkoa ohjaavan kontaktorin ohjausreleeseen. Kun hätäpysäytyspiiri katkeaa, ohjausreleen kosketin avautuu ja virta katkeaa koneikkoa ohjaavalta

kontaktorilta. Tämän myötä hydraulikkakoneikko sammuu ja välitöntä vaaraa aiheuttavat liikkeet pysähtyvät.

5 LÄHI-INFRAPUNA-ANTURIN INTEGROINTI PLC:HEN

Lähi-infrapuna-anturi NIRONE Sensor S integroitiin valittuun Beckhoffin sulautettuun tietokoneeseen CX5130-0155. CX5130-0155 tunnisti NIR-anturin sarjaporttina USB-liitännän kautta ilman lisäohjelmia, mikä helpotti anturin integrointia. Anturin rajapinta pystyttiin ohjelmoimaan sensoria tukevalla ohjelmointikielellä.

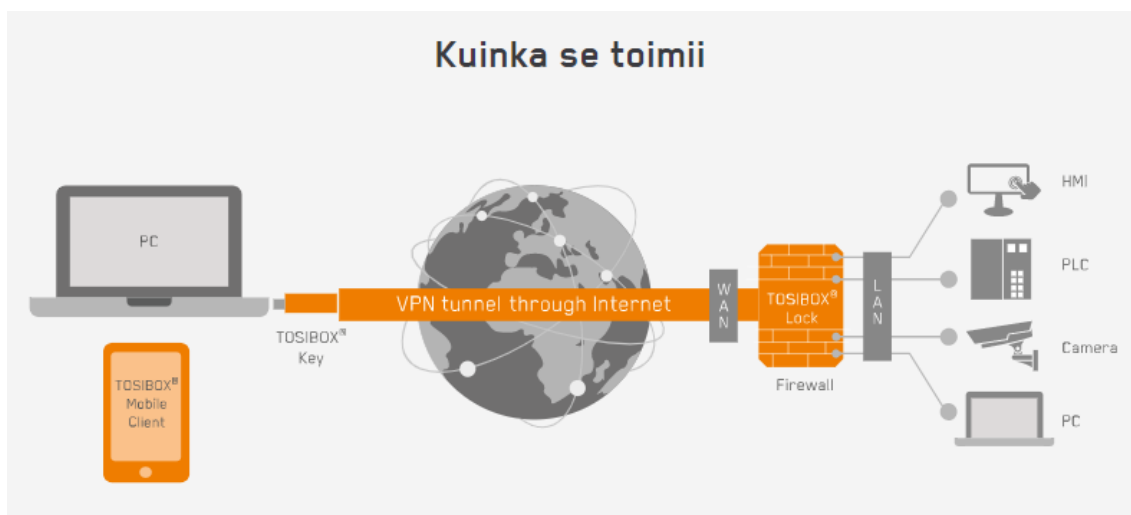
Ensimmäiseksi ohjelmoitiin Spectral Enginen ohjelmistokehityspaketin avulla toimiva ohjelma. Ohjelma käynnistää NIR-anturin ja ensimmäisenä kalibroi sen. Kalibroinnin jälkeen NIR-anturin valo syttyy ja anturi alkaa lukemaan jatkuvasti halettujen aallonpituuksien pisteitä. Ohjelmassa oleva algoritmi laskee tuloksen, jonka perusteella ohjelma tulostaa lukuarvon. Lukuarvon perusteella voidaan päätellä, onko anturin edessä muovia vai sekajätettä.

Valmis ohjelma integroitiin PLC:n kanssa. Beckhoffin Oulun toimipisteen teknisen tuen avulla NIR-anturi saatiin keskustelemaan PLC-ohjelman kanssa ADS-protokollan avulla. ADS-protokolla (Automation Device Specification) on laite- ja kenttäväyläriippumaton tiedonsiirtoprotokolla ja rajapinta, joka on Beckhoffin kehittämä. Beckhoff käyttää sitä omissa automaatio-sovelluksissaan esimerkiksi tiedonsiirrossa TwinCAT 3 -kehitysympäristön ja -runtimen välillä. (16, s. 17.)

6 JATKOKEHITYS

Opinnäytetyön yhtenä tavoitteena oli tutkia jatkokehitysideana IoT-tietokannan käyttöä Beckhoff-ympäristössä. IoT eli Internet of Things koostuu älykkäistä laitteista, jotka ovat yhdistetty toisiinsa. Laitteet keräävät ja käyttävät dataa suorittaakseen tehtäviä. (26.) Muovinkierrätyslaitteiston liittäminen IoT-pilvipalveluun ja IoT-tietokantaan parantaisi laitteen useita ominaisuuksia. Esimerkiksi NIR-anturi voisi lukea muovintunnistusalgoritmin datan IoT-pilvipalvelusta, joka mahdollistaisi huomattavasti suuremman datamäärän verrattuna paikalliseen datatiedostoon. Suurella datamäärällä saavutettaisiin entistä tarkempi muovintunnistusalgoritmi.

IoT-laite tarvitsee verkkoyhteyden, joka on turvallista luoda esimerkiksi Tosiboxin tuotteiden avulla. Tosibox on Oulussa perustettu yritys, joka suunnittelee ja valmistaa korkeimman luokan tietoturvasovelluksia muun muassa teollisuuteen (27; 28). Tosibox Lukko on etäyhteyslaite, johon liitetään fyysinen laitteisto. Liitettyihin laitteisiin on mahdollista päästä Tosibox Avaimella salatulla VPN-yhteydellä (kuva 25). (29; 30.)

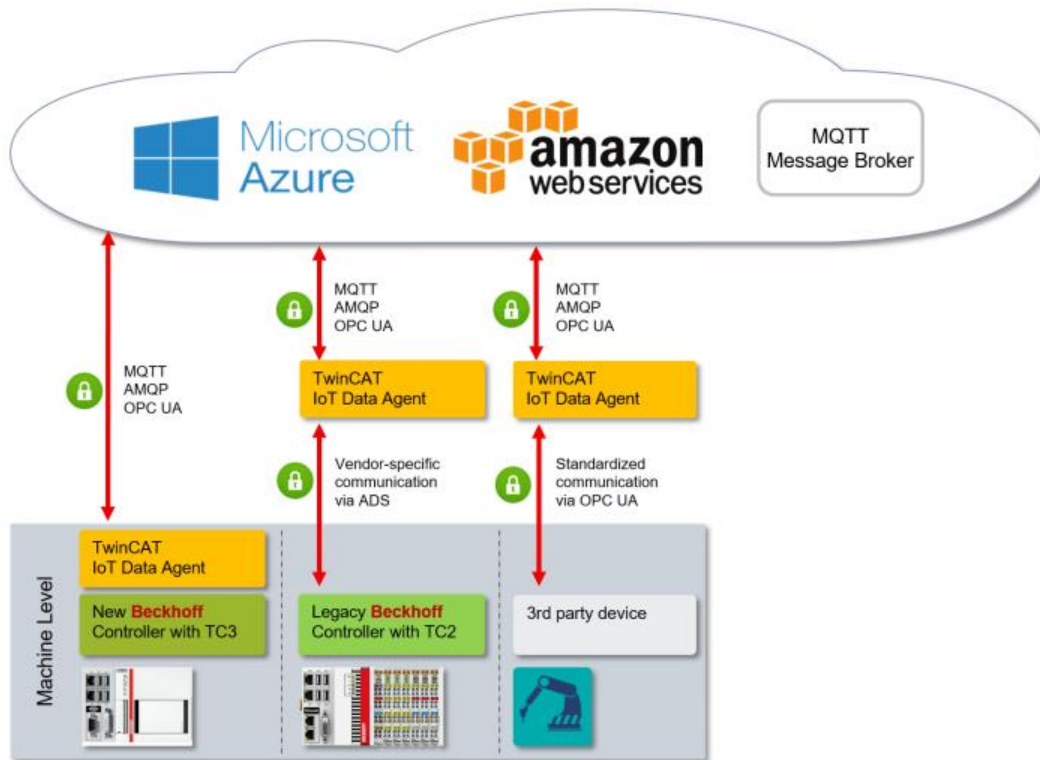


KUVA 25. Tosiboxin toimintaperiaate (26)

Tosibox-tuotteilla kierrätyslaitteistoa ohjaava CX5130-0155 saataisiin turvallisesti verkkoon, joka mahdollistaisi IoT-palveluiden käytön. Verkkoyhteys olisi mahdollista toteuttaa myös perinteisellä 4G-mokkulalla, mutta kierrätyslaitteeseen halutaan mahdollisimman turvallinen verkkoyhteys.

CX5130-0155 tukee suoraan IoT-mahdollisuuksia, eikä erillisiä IoT-terminaaleja tai -kytkimiä tarvita. TwinCAT 3 -kehitysympäristöön on saatavilla useita IoT-työkaluja. Perustoiminnallisuudet saadaan aikaan TC3 IoT Communication -työkälulla PLC-kirjastojen muodossa. TC3 IoT Communication -työkalu käyttää datan lähettämisessä ja vastaanottamisessa MQTT-protokollaa, joka on laajasti käytetty standardoitu tiedonvälitysprotokolla. (31.)

Monipuolisemmat IoT-toiminnot toteutetaan itsenäisellä TC3 IoT Data Agent -sovelluksella. Beckhoffin TC3 IoT Data Agent -sovellus on kaksisuuntaiseen IoT-kommunikointiin kehitetty porttisolvellus. Sovellus voidaan asentaa PLC-tietokoneeseen tai siihen yhdistettyyn tietokoneeseen. Sovellus poimii konfiguroidun datan ja välittää sen pilveen viestintä- tai datapalveluun. Pilvipalveluna voidaan käyttää yleisiä pilvipalveluita, kuten Microsoft Azurea tai AWS-palvelua (Amazon Web Services). Sovellus voi lähettää konfiguroidun datan myös MQTT- tai AMQP-välittäjälle. TC3 IoT Data Agent -sovellus tukee myös OPC UA -kommunikointiprotokollaa, joka mahdollistaa kolmannen osapuolen laitteiden yhdistämisen pilveen. Kuvassa 26 on selvitetty eri laitteiden yhdistäminen pilvipalveluun TC3 IoT Data Agent -sovelluksen avulla. (32, s. 7.)



KUVA 26. Eri laitteiden yhdistäminen IoT-pilvipalveluun (32, s. 7)

TC3 IoT Data Agent -sovellusta käyttämällä saataisiin muovinkierrätyslaite yhdistettyä IoT-pilvipalveluun ja IoT-tietokantaan. IoT-pilvipalveluksi soveltuisi esimerkiksi Microsoft Azure tai AWS. Molemmat näistä palveluista tukevat relaatiotietokantoja ja sisältävät myös koneoppimistyökaluja, jotka voivat olla tulevaisuudessa kierrätyslaitteelle hyödyllisiä ominaisuuksia (33). Tarkempaa selvitystä IoT-pilvipalveluista ja -tietokannoista ei tässä opinnäytetyössä tehdä.

7 YHTEENVETO

Opinnäytetyön päätavoitteena oli suunnitella ja toteuttaa ohjausjärjestelmä auto-maattiseen muovinkierrätyslaitteistoon Beckhoff-ympäristössä. Opinnäytetyö sisälsi myös laitteen käyttöliittymän suunnittelun ja toteutuksen käyttäjille sekä huoltohenkilöille. Lisäksi tavoitteena oli integroida ohjausympäristöön lähi-infrapuna-anturi ja koodilukkona toiminut matriisinäppäimistö. Ohjausjärjestelmään tehtiin myös vaadittavat turvatoiminnot Beckhoffin TwinSAFE-turvakomponenttien ja -ohjelman avulla. Opinnäytetyössä tutkittiin lisäksi kierrätyslaitteen jatkokehitystä varten mahdollisuutta käyttää IoT-tietokantaa Beckhoff-ympäristössä.

Opinnäytetyössä valittiin ohjausjärjestelmän komponentit jo aiemmin valitun NLRONE Sensor S -lähi-infrapuna-anturin perusteella. Ohjausjärjestelmän täytyi olla yhteensopiva anturin kanssa, minkä vuoksi ohjausjärjestelmän keskusyksiköksi valittiin Beckhoffin sulautettu tietokone CX5130-0155. Sulautettuun tietokoneeseen lisättiin pienikokoinen kosketusnäyttöpaneeli graafista käyttöliittymää varten. Modulaariset tulo- ja lähtöterminaalit valittiin I/O-taulukon mukaisesti. Tuloportteja tarvittiin 13 ja lähtöportteja 18. Turvalogiikaksi valittiin Beckhoffin EL2911, joka täytti vaadittavat turvastandardit ja se mahdollisti yksinkertaisen häätäpysäytyspiirin tekemisen.

Logiikkaohjelma suunniteltiin laitteen halutun toiminnan mukaisesti vuokaavioon. Vuokaavion perusteella logiikkaohjelma ohjelmoitiin TwinCAT 3 -kehitysympäristössä Structured Text ja Function Block Diagram -kielillä. Logiikkaohjelma jaettiin neljään FBD-kieliseen ohjelmaan ja seitsemään ST-kieliseen ohjelmaan. Matriisinäppäimistö integroitiin ohjausjärjestelmään kahdella ST-kielisellä ohjelmalla. Toisessa ohjelmassa matriisinäppäimistön näppäimet määriteltiin ja toisessa ohjelmassa määritellyt näppäimet kirjoitettiin koodiin string-tietotyyppinä.

NIR eli lähi-infrapuna-anturi integroitiin ohjausjärjestelmään Beckhoffin ADS-protokollaa käyttäen. NIR-anturille ohjelmoitiin ohjelma, jolla pystyttiin erottamaan muovi sekajätteestä. ADS-protokollan avulla luotiin keskusteluyhteys ohjausjärjestelmän ja ohjelman välille. Keskusteluyhteys mahdollisti laitteen ohjaamisen NIR-anturilta saatujen tietojen perusteella.

Jatkokehityksenä tutkittiin IoT-tietokannan liittämistä ohjausjärjestelmään. IoT-tietokanta ja -pilvipalvelu on mahdollista liittää valittuun sulautettuun tietokoneeseen ilman fyysisiä lisäkomponentteja. Verkkoyhteys voidaan toteuttaa Tosiboxin tuotteilla turvallisesti salatulla VPN-yhteydellä. Beckhoffin TC3 IoT Data Agent -porttisovelluksella voidaan järjestelmä yhdistää haluttuun pilvipalveluun standardeja protokollia käyttäen.

Suurimmat ongelmat koettiin matriisinäppäimistön integroinnissa. Matriisinäppäimistössä ei ollut erillistä liitinnapaa virralle, vaan virta piti syöttää näppäimistön pystyriveille. Pystyriivit liitettiin PLC:n lähtöportteihin ja lähtöjä vaihdeltiin seitsemän millisekunnin välein. Vaakarivit liitettiin tuloportteihin, jolloin näppäintä painaessa vaakarivin tulo syttyi, kun oikean pystyriivin lähtöportti meni päälle. Tällä tavalla oikea näppäin saatiin luettua PLC:hen ja kirjoitettua koodiin. Lisäksi valitun matriisinäppäimistön näppäintuntuma oli huono, mikä huononsi käytettävyyttä.

Opinnäytetyö onnistui mielestäni hyvin ja ohjausjärjestelmästä tuli toimiva. Oli mielenkiintoista huomata, kuinka paljon yhteistyötä ohjausjärjestelmän suunnittelu vaatii mekaniikkainsinöörien ja muiden projektiryhmän henkilöiden kanssa. Oma osaamiseni kehittyi valtavasti logiikkaohjelmoinnissa, ohjausjärjestelmissä ja sähköteknisissä asioissa. Ennen projektia osasin perusteet ohjausjärjestelmistä ja FBD-logiikkaohjelmoinnista, mutta projektin jälkeen tunnen olevani kokonaisvaltaisesti osaava ohjausjärjestelmien suunnittelija ja ST-kielen logiikkaohjelmoija.

LÄHTEET

1. Rintala, Milja 2018. Oululaismiljonääri kehittää muovista miljardibisnestä: Ensin laitteistoja, sitten kokonaisia tehtaita. Oulu: Ilta-Sanomat 31.8.2018. Saatavissa: <https://www.is.fi/oulun-seutu/art-2000005811009.html>. Hakupäivä 16.11.2019.
2. Keinänen, Toimi – Kärkkäinen, Pentti – Lähetkangas, Markku – Sumujärvi, Matti 2007. Automaatiojärjestelmien logiikat ja ohjaustekniikat. Helsinki: WSOY Oppimateriaalit Oy.
3. Frid, Johnny – Johnson, Jörgen 2005. Ohjaustekniikka: Oppikirja A. Iisalmi: IS-VET.
4. Relay Logic Vs Ladder Logic. 2019. Ladder Logic World. Saatavissa: <https://ladderlogicworld.com/relay-logic-vs-ladder-logic/>. Hakupäivä 24.10.2019.
5. Keinänen, Toimi – Kärkkäinen, Pentti – Metso, Tommi – Putkonen, Kari 2001. Koneautomaatio: 2, Logiikat ja ohjausjärjestelmät. Helsinki: WSOY.
6. Balance between latest PC technology and long-term availability. 2019. Beckhoff. Saatavissa: <https://www.beckhoff.fi/fi/default.htm?beckhoff/products.htm>. Hakupäivä 24.10.2019.
7. Embedded PC has Profinet interface. 2013. Beckhoff Automation. Saatavissa: <https://www.controleng.com/articles/embedded-pc-has-profinet-interface/>. Hakupäivä 24.10.2019.
8. NIRONE Sensor S. Spectral Engines. Saatavissa: https://cdn2.hubspot.net/hubfs/4905262/Assets/Brochures/NIRONE_Sensor_product_brochure_04_WEB.pdf?_hssc=192714997.4.1571997421927&_hstc=192714997.509e16df371bbebc1244dadb1abc2625.1557125457463.1571995500625.1571997421927.4&_hsfp=2267869807&hsCtaTracking=109067f2-

- [aed0-4b7c-b601-100a5e2c2c4f%7C2cf2ce31-1dbf-4735-975d-a012280494a6](https://www.beckhoff.com/Products/CP6906). Hakupäivä 25.10.2019.
9. Kantojärvi, Uula 2019. Chief Technology Officer. Spectral Engines Oy. Pala-
veri 8.5.2019.
10. Product Overview. 2018. Beckhoff Automation GmbH & Co. Saatavissa:
https://download.beckhoff.com/download/Document/Catalog/Beckhoff_Products_2018.pdf. Hakupäivä: 27.10.2019.
11. CP6906-0001-0000 | “Economy” built-in Control Panels with DVI/USB Ex-
tended interface. 2019. Beckhoff Automation. Saatavissa: <https://www.beckhoff.com/CP6906/>. Hakupäivä: 12.1.2020.
12. SFS-EN ISO 13849-1. 2015. Koneturvallisuus. Turvallisuuteen liittyvät oh-
jausjärjestelmien osat. Osa 1: Yleiset suunnitteluperiaatteet. Helsinki: Suo-
men Standardisoimisliitto SFS.
13. Artikkelisarja turvallisen koneen suunnittelusta, Osa 4: Turvallisuuteen liitty-
vien ohjausjärjestelmien suunnittelu. 2019. METSTA. Saatavissa:
http://www.metsta.fi/www/koneturvallisuuden_teemasivut/artikkelit/Turvalli-senkoneensuunnitteluosa4.Turvallisuuteenliittyvienohjausjrjestelmiensuun-nittelu.pdf. Hakupäivä: 27.10.2019.
14. External Decive Monitoring (EDM). 2019. Schneider Electric Saatavissa:
[https://product-help.schneider-electric.com/Machine%20Ex-pert/V1.1/en/tm3ioprg/tm3ioprg/TM3_Safety_Modules/TM3_Safety Mod-ules-6.htm](https://product-help.schneider-electric.com/Machine%20Expert/V1.1/en/tm3ioprg/tm3ioprg/TM3_Safety_Modules/TM3_Safety_Mod-ules-6.htm). Hakupäivä: 27.10.2019.
15. EL2911. 2019. Beckhoff Automation. Saatavissa: <https://www.beckhoff.com/EL2911/>. Hakupäivä 27.10.2019.
16. TwinCAT 3 Käsikirja. 2019. Versio 3.1.4022. Beckhoff Automation Oy

17. TC1200 | TC3 PLC. 2019. Beckhoff Automation. Saatavissa: https://infosys.beckhoff.com/english.php?content=../content/1033/tc3_base_overview/27021597874762635.html&id=7441776209718377895. Hakupäivä: 14.11.2019.
18. IEC 61131-3. 2013. PLCopen. Saatavissa: <https://plcopen.org/iec-61131-3/>. Hakupäivä: 14.11.2019.
19. Understanding the IEC61131-3 Programming Languages. 2009. Bosch Rexroth Corporation. Saatavissa: https://www.automation.com/pdf_articles/IEC_Programming_Thayer_L.pdf. Hakupäivä: 16.11.2019.
20. Merioja, Riikka 2019. Vuokaavio RAU-suunnittelun työkaluna datakeskusjärjestelmissä. Insinööriyö. Metropolia Ammattikorkeakoulu, sähkö- ja automaatiotekniikka. Saatavissa: https://www.theseus.fi/bitstream/handle/10024/261688/Merioja_Riikka.pdf?sequence=2&isAllowed=y. Hakupäivä: 5.2.2020.
21. CCM12TR - Kalvonäppäimistö 12 painokytintä, Apem. 2020. Elfa Distrelec. Saatavissa: <https://www.elfadistrelec.fi/fi/kalvonaeppaemistoet-12-painokytintae-apem-ccm12tr/p/13567971>. Hakupäivä: 10.1.2020.
22. Produktinformation. 2002. Elfa. Saatavissa: <https://www.elfadistrelec.fi/Web/Downloads/79/55/03567955.pdf?pid=13567971>. Hakupäivä: 10.1.2020.
23. Overview. 2020. Beckhoff Automation. Saatavissa: https://infosys.beckhoff.com/english.php?content=../content/1033/tcplclib_tc2_standard/9007199328716043.html&id=2674467248723055811. Hakupäivä: 5.1.2020.
24. Replace. 2020. Beckhoff Automation. Saatavissa: https://infosys.beckhoff.com/content/1033/tcplclib_tc2_standard/9007199329162763.html?id=5603890705187125585. Hakupäivä: 5.1.2020.

25. Operating Instructions for EL2911. 2019. Beckhoff. Saatavissa: <https://download.beckhoff.com/download/document/automation/twinsafe/el2911en.pdf>. Hakupäivä: 5.1.2020.
26. IoT. 2020. TOSIBOX. Saatavissa: <https://www.tosibox.com/fi/iot/>. Hakupäivä: 7.1.2020.
27. Lyhyesti yrityksestä. 2020. TOSIBOX. Saatavissa: <https://www.tosibox.com/fi/lyhyesti-yrityksesta/>. Hakupäivä: 9.1.2020.
28. Helppoa IoT-tietoturvaa. 2020. TOSIBOX. Saatavissa: <https://www.tosibox.com/fi/iot-turvallisuus/>. Hakupäivä: 9.1.2020.
29. TOSIBOX® Avain. 2020. TOSIBOX. Saatavissa: <https://www.tosibox.com/fi/tuote/avain/>. Hakupäivä: 7.1.2020.
30. TOSIBOX® Lukko 100. 2020. TOSIBOX. Saatavissa: <https://www.tosibox.com/fi/tuote/lock-100/>. Hakupäivä: 9.1.2020.
31. TF6701 | TC3 IoT Communication (MQTT). 2019. Beckhoff Automation. Saatavissa: <https://www.beckhoff.com/english.asp?twincat/tf6701.htm>. Hakupäivä: 11.1.2020.
32. TC3 IoT Data Agent. 2018. Beckhoff. Saatavissa: https://download.beckhoff.com/download/document/automation/twincat3/TF6720_TC3_IoT_Data_Agent_EN.pdf. Hakupäivä: 7.1.2020.
33. Carey, Scott 2019. AWS vs Azure vs Google: What's the best cloud platform for enterprise? Saatavissa: <https://www.computerworld.com/article/3429365/aws-vs-azure-vs-google-whats-the-best-cloud-platform-for-enterprise.html> Hakupäivä: 11.1.2020.

