



samk

Satakunnan ammattikorkeakoulu  
Satakunta University of Applied Sciences

MIKKO LEPPÄLÄ

# **Mobiilirobotin kuljettimen sähköis- täminen ja integrointi automaa- tiojärjestelmään**

SÄHKÖ- JA AUTOMAATIOTEKNIIKAN  
KOULUTUSOHJELMA  
2020

Tekijä(t) Leppälä, Mikko	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä 12/2021
	Sivumäärä 33	Julkaisun kieli Suomi
Julkaisun nimi <b>Mobiilirobotin kuljettimen sähköistäminen ja integrointi automaatiojärjestelmään</b>		
Tutkinto-ohjelma Sähkö- ja automaatiotekniikka		
<p>Tiivistelmä</p> <p>Mobiilirobotit ovat viime vuosina nostaneet päätään markkinoilla ja erilaisia teollisuuden sovelluksia ilmestyy enenevässä määrin. Vaikka mobiilirobotit eivät ainakaan vielä ole tyystin korvaamassa perinteisiä kuljettimia, ovat ne joustavuutensa ja muokattavuutensa takia erittäin hyvä vaihtoehto niille sopiviin sovelluksiin.</p> <p>Työn päämääränä oli rakentaa Kone Kuusisto Oy:n tiloihin laatikonkäsittelydemo, jossa Omron LD-60 -mobiilirobotti tuo tyhjiä laatikoita kuljettimelle, jossa Omron TM -yhteistyörobotti täyttää laatikot. Täydet laatikot mobiilirobotti vie painovoimaiselle varastoradalle. Kuljetin, jossa yhteistyörobotti täyttää laatikot sekä tyhjien laatikoiden varastokuljetinta ohjaa Siemensin PLC. Kommunikointi mobiilirobotin, yhteistyörobotin ja Siemensin PLC:n välillä kulkee Omron NX1P2 PLC:n kautta.</p> <p>Tämä työ keskittyy mobiilirobotin ohjelmointiin sekä sen kuljettimen sähkösuunnitteluun ja Omron NX1P2 PLC:n ohjelmointiin. Projekti onnistui muutamaa demoeefektiä lukuun ottamatta mainiosti. Valmista demoa käytiin katsomassa – koronarajoitusten puitteissa – useamman lähialueen yrityksen voimin.</p>		
<p><a href="#">Asiasanat</a>          mobiilirobotti, yhteistyörobotti, ohjelmoitava logiikka</p>		

Author(s) Leppälä, Mikko	Type of Publication Bachelor's thesis	Date 12/2021
	Number of pages 33	Language of publication: Finnish
Title of publication <b>Electric design of a conveyor on mobile robot and integration into automation system</b>		
Degree program Electric and automation technology		
<p>Abstract</p> <p>In the past few years, applications that utilize mobile robots in industrial setting have been steadily increasing. Even though mobile robots may not fully replace traditional conveyors, they are an excellent asset in any application which require flexibility and adaptability.</p> <p>The goal of this project was to build a functioning crate handling demo in which an Omron LD-60 mobile robot picks up empty crates from a storage conveyor and drops them off onto another conveyor. Both conveyors are controlled by a Siemens PLC. On the latter conveyor a collaborative robot – Omron TM – fills the crates with boxes. Full crates are then picked up by the mobile robot and transported onto a gravitational storage conveyor. Communication between the robots and the Siemens PLC is handled by Omron NX1P2 PLC.</p> <p>This thesis focuses on the electrical design of the conveyor that is on the mobile robot and the programming of the mobile robot and Omron NX1P2 PLC. All in all, the project was a success aside from a few hiccups during the presentations. Several companies from the nearby area came to see the final demo.</p>		
<p><u>Key words</u>          mobile robot, collaborative robot, programmable logic controller</p>		

# SISÄLLYS

1 JOHDANTO .....	6
1.1 Tausta .....	6
1.2 Tavoitteet.....	6
1.3 Kokoonpano .....	6
2 OMRON LD-60 -SÄHKÖJÄRJESTELMÄ .....	7
2.1 Yleistä .....	7
2.2 Rullakuljetin.....	7
2.3 Kuljettimen anturointi .....	9
2.4 LD-60 I/O -järjestelmä.....	10
2.4.1 I/O- eli tulo- ja lähtöjärjestelmä.....	10
2.4.2 LD-60:n tulot .....	12
2.4.3 LD-60:n lähdöt.....	13
3 MOBILEPLANNER JA ARCL-KOMENTOKIELI .....	15
3.1 Kartan teko .....	15
3.1.1 Skannaus .....	15
3.1.2 Maalit, odotuspisteet ja lataustelakka .....	15
3.2 Työt .....	16
3.3 Mobiilirobotin ohjelmointi.....	16
3.4 ARCL-komentokieli.....	19
4 OHJELMOITAVAT LOGIIKAT .....	20
4.1 IEC 61131 .....	20
4.1.1 Muuttujat.....	20
4.1.2 Program organization unit (POU).....	21
4.1.3 Ohjelmointikielet .....	22
4.2 Siemensin ohjelmoitava logiikka .....	24
4.3 Omron NX1P2 .....	24
4.3.1 Pääohjelma ja taskit .....	25
4.3.2 Mobiilirobotin ohjaukseen liittyvät ohjelmat .....	25
4.4 Omron TM .....	30
4.4.1 Modbus .....	30
4.4.2 Yhteistyörobotin kommunikointiin liittyvät ohjelmat .....	30
5 TULOKSET .....	33
LÄHTEET	

## SYMBOLI- JA LYHENNELUETTELO (EI PAKOLLINEN)

PLC	Programmable logic controller, ohjelmoitava logiikka
BOOL	Boolean, muuttujan datatyyppejä, jolla on yksi kahdesta arvosta, 0 tai 1
INT	Signed integer, muuttujan datatyyppejä, etumerkillinen kokonaisluku välillä -32768 - (+)32767
STRING	String, muuttujan datatyyppejä, johon voi tallentaa tekstiä
STUCT	Structure, kokoelma muuttujia ja niiden datatyyppejä
POU	Program organization unit, ohjelmakokonaisuus
PROG	Program, korkeimman tason ohjelmakokonaisuus
FB	Function block, ohjelmalohko
FUN	Function, ohjelmalohko, jossa on vain yksi lähtömuuttuja
Omron LD-60	Omron mobile robot with 60kg load capacity, Omronin toimittama mobiilirobotti, jonka kantokyky on 60kg
Omron TM	Omron collaborative robot, cobot, Omronin toimittama yhteistyörobottibrändi
I/O	Input/output, tulo- ja lähtöjärjestelmä
ARCL	Advanced Robotics Command Language, mobiilirobotin tekstipohjainen komentokieli

# 1 JOHDANTO

## 1.1 Tausta

Kone Kuusisto Oy on vuonna 2012 perustettu sähkö- ja automaatioalan yritys, joka työllistää Säkylässä opinnäytetyön tekohetkellä 8 henkilöä. Yritys tekee pääosin teollisuuskoneiden sähkösuunnittelua, ohjelmointia, sähköasennusta, robotiikkaa ja huoltoa. Tämän projektin myötä yritys saa ensikosketuksensa mobiilirobotteihin ja yhteistyörobotteihin.

## 1.2 Tavoitteet

Työn tavoitteena on sähköistää mobiilirobotin päälle rakennettava rullakuljetin ja ohjelmoida mobiilirobotti palvelemaan yhteistyörobottia tuomalla sille tyhjiä laatikoita ja viemällä täydet laatikot niin sanotulle varastoradalle. Valmista demoa on tarkoitus esitellä lähialueen yrityksille ja herätellä mahdollisia ideoita tulevaisuuden tuotoksille.

## 1.3 Kokoonpano

Koko projekti koostuu kahdesta ohjelmoitavasta logiikasta, mobiilirobotista, yhteistyörobotista ja kolmesta oikosulkumoottorikäyttöisestä kuljettimesta sekä yhdestä painovoimaisesta kuljettimesta. Painovoimainen kuljetin toimii varastoratana ja moottorikäyttöiset kuljettimet Omron TM -yhteistyörobotin laatikkoratana ja kappaletavararatana sekä yksi antaa tyhjiä laatikoita mobiilirobotille. Toinen ohjelmoitavista logiikoista, Siemensin logiikka, ohjaa kuljettimia ja Omron NX1P2 -logiikka mahdollistaa kommunikoinnin projektin eri laitteiden välillä sekä ohjaa mobiilirobotin sekvenssiin liittyviä ohjelmia. Omron LD-60 -mobiilirobotti vie täydet laatikot kuljettimelta varastoradalle ja tuo tyhjiä laatikoita kuljettimelle, jossa Omron TM täyttää niitä.

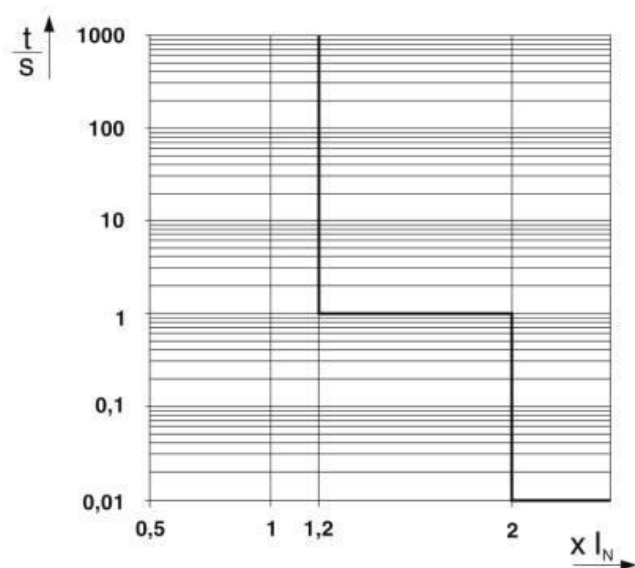
## 2 OMRON LD-60 -SÄHKÖJÄRJESTELMÄ

### 2.1 Yleistä

Omron LD-60 -mobiilirobotissa on 72 ampeeritunnin akku, josta riittää sähköä mobiilirobotille ja sen lisälaitteille 15 tunnin ajaksi. Akku toimii 22–30 voltin tasajännitteellä. LD-60:stä on saatavilla muutama eri jänniteulostulo mobiilirobotin lisälaitteille. Vakavoituja tasajännitteitä on tarjolla seuraavasti: 5 V, 12 V ja 20 V. Jokainen em. syötöistä kestää 1 ampeerin kuorman. Lisäksi on tarjolla raakaa akkuvirtaa 22–30 voltin tasajännitteellä, jota voi kuormittaa joko 4 ampeerin kuormalla tai 10 ampeerin kuormalla, ottopaikasta riippuen. Yksi 10 ampeerin syötöistä on turvapiirin takana, joka katkeaa, jos mobiilirobotin turvapiiri rikkoontuu. Tässä projektissa ei tarvita turvapiirin takaista syöttöä ja siitä olisi oikeastaan vain haittaa pitkällä aikavälillä (Omron industrial automation www-sivut, 2021.)

### 2.2 Rullakuljetin

24 voltin rullamoottoreita on saatavilla usealta eri valmistajalta. Tähän projektiin valittiin Interrollin RollerDrive EC5000 -rullamoottori ja sille DriveControl 54 -ohjain. DriveControl -ohjaimen käyttäjämateriaalista käy ilmi, että ohjain tarvitsee toimiakseen 19–26 voltin tasajännitettä. LD-60:n akusta saatava vakavoimattoman jännitelähdön jännitepiikit saattavat olla haitallisia ohjaimen herkille puolijohteille, siksi jännite ohjataan Siemensin DC/DC-konvertterin läpi rullakuljettimen ohjaimelle. DC/DC-konvertteriin voidaan syöttää 14–32 voltin tasajännitettä ja ulostulosta saadaan vakavoitua 24 voltin tasajännitettä. DC/DC-konvertteri halutaan myös suojata oikosulun varalta. Oikosulkusuojaksi tarvitaan nopeasti reagoiva suojalaite, näitä ovat Z-käyrälliset johdonsuojalaitteet ja elektroniset selektiivisuojalaitteet. Edellä mainitut komponentit varmistavat, että rullamoottorin ohjain saa tarpeeksi oikeanlaista jännitettä, eikä minkään pitäisi hajota oikosulun sattuessa (Interroll www-sivut, 2021.)



Kuva 1. Phoenix Contact PTCB -suojalaitteen laukaisukäyrä (Phoenix Contact www-sivut, 2021)

Kuvan 1 kuvaajassa vaaka-akselilla on nimellisvirran monikerrat ja pystyakselilla aika. Kuva on Phoenix Contactin elektronisesta selektiivisuojalaitteesta, jolla suojataan DC/DC-konvertteri ja jonka datalehti lupaa alle 10 millisekunnin poiskytkennän kaksinkertaisella nimellisvirralla. Suojalaitteen oikosulun katkaisukyky ylittää 300 ampeeriin saakka (Phoenix Contact www-sivut, 2021.)



### 2.3 Kuljettimen anturointi

Mobiilirobotin päällä olevalla kuljettimella on tarkoitus siirtää laatikoita kuljettimelta toiselle. Jotta tämä saadaan tapahtumaan mahdollisimman varmasti joka ikinen kerta, tarvitaan antureita. Mahdollista olisi myös mitata ajallisesti, kuinka kauan laatikon kyytiin otto tai purku vie aikaa ja sen perusteella tehdä ohjelma mobiilirobotin kuljettimelle. Tässä tapauksessa laatikon vinosyöttö, laitoihin takertelu tai mikä tahansa muu tekijä luultavasti tiputtaisi laatikon, ja sen sisällön, maahan robotin lähtiessä liikkeelle. Suunnitteluvaiheessa haluttiin tehdä kahdenlainen anturointi, yksi laatikoita varten ja toinen mobiilirobotin paikoittamista varten. Omronin kenttäsovellusasiantuntija vakuutti, että LD-60:n paikannus olisi tarpeeksi tarkka ja tarkkuutta voitaisiin parantaa entisestään mobiilirobotin kohteilla. Kohteet ovat L-kirjaimen mallisia peltejä, joista mobiilirobotti pystyy peilaamaan paikoituksen  $\pm 5$  mm tarkkuudella (Virmanen henkilökohtainen tiedonanto, 19.10.2020.)

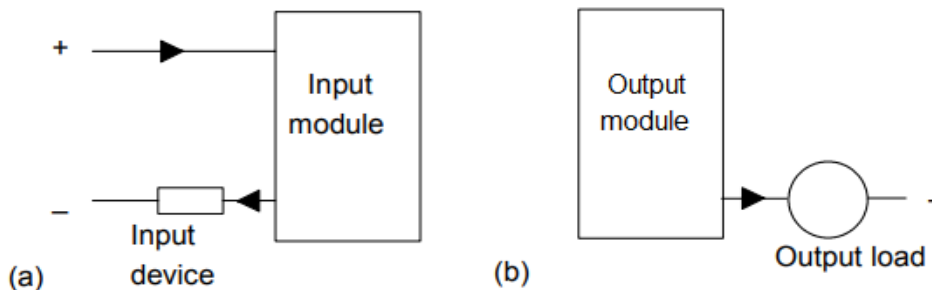
Mobiilirobotin kuljettimen anturointi tehdään kolmella peilistä heijastavalla optisella lähestymiskytkimellä eli valokennolla. Valokennosta lähtevä valo heijastuu peilin kautta takaisin anturille eli lähetin ja vastaanotin ovat samassa pakkauksessa. Valonsäteen katkeaminen luonnollisesti tarkoittaa sitä, että kohde on havaittu eli laatikko on kuljettimella. Yksi valokennoista sijoitetaan kuljettimen takaosaan, yksi keskelle ja yksi etuosaan. Kolmella valokennolla voidaan tarkkailla sitä, että laatikko pysyy kydyssä ja mahdollisesti korjata sen paikkaa tarpeen tullen.

## 2.4 I/O-järjestelmä

Koneiden ”aivot” – yleisesti ohjelmoitavat logiikat – saavat tietoa järjestelmän tilasta tuloporttien kautta ja lähtöporttien kautta ne voivat ohjata järjestelmän toimilaitteita. Tässä projektissa tuloja tarvitaan siihen, että saadaan mobiilirobotin kuljettimen anturinnista tulevat signaalit sekä Siemensin logiikalta tulevat kättelytiedot käsiteltyä. Lähtöjä taas tarvitaan kuljettimen käynnistämiseen, sen suunnan valitsemiseen ja kättelytietojen antoon Siemensin logiikalle.

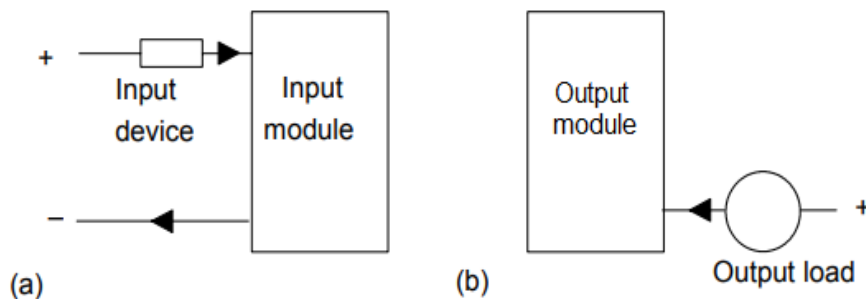
### 2.4.1 Tulo- ja lähtöjärjestelmätyypit

I/O-tyyppejä on kahdenlaisia, ns. positiivinen logiikka eli virtaa ottava (sinking) ja negatiivinen logiikka eli virtaa antava (sourcing). Virtaa antavassa järjestelmässä sähkö virtaa logiikan tulokortilta laitteelle, esim. anturille, eli tulokortti on silloin anturin virran lähde. Lähtöpuoli toimii samalla lailla eli kuorma saa virran logiikan lähtökortilta (Kuva 2).



Kuva 2. Sourcing eli virtaa antava järjestelmä (Programmable Logic Controllers 2006, 10)

Virtaa ottavassa järjestelmässä kentän laitteille vietään positiivinen jännite, joka sitten tuodaan ohjelmoitavalle logiikalle. Tällöin sähkö virtaa luonnollisesti laitteelta logiikan kortille (Kuva 3).

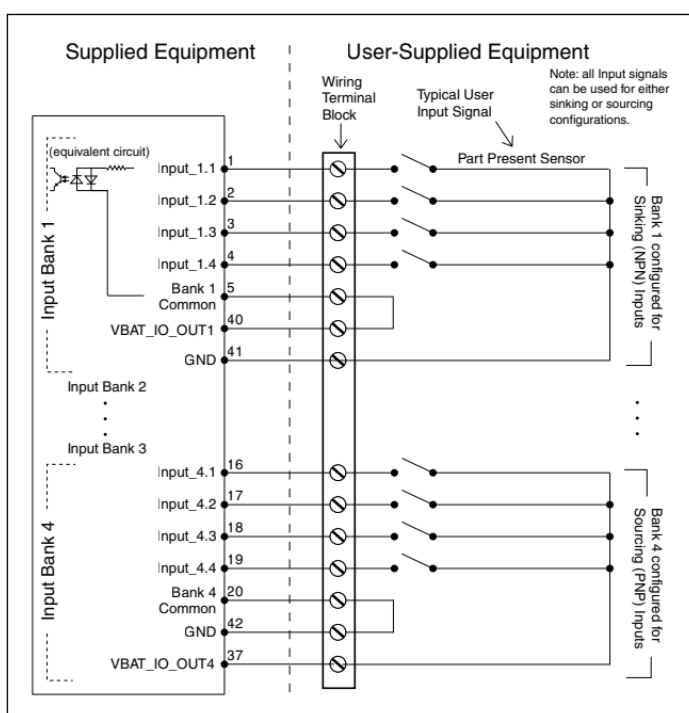


Kuva 3. Sinking eli virtaa ottava järjestelmä (Programmable Logic Controllers 2006, 10)

On tärkeää ottaa selvää sekä logiikan että kentän päässä käytettävän anturoinnin tulo-tyypeistä, jotta kytkennät voidaan tehdä oikein. Yleensä katsotaan logiikan päästä, kumpaan suuntaan sähkövirta kulkee valittaessa I/O-järjestelmän tyyppiä. Mikäli logiikan tulokortit ovat virtaa ottavia, pitää anturien olla PNP-tyyppiä. Jos taas logiikan tulokortit ovat virtaa antavia, tarvitaan NPN-tyyppisiä antureita (Programmable Logic Controllers 2006, 10.)

## 2.4.2 LD-60:n tulot

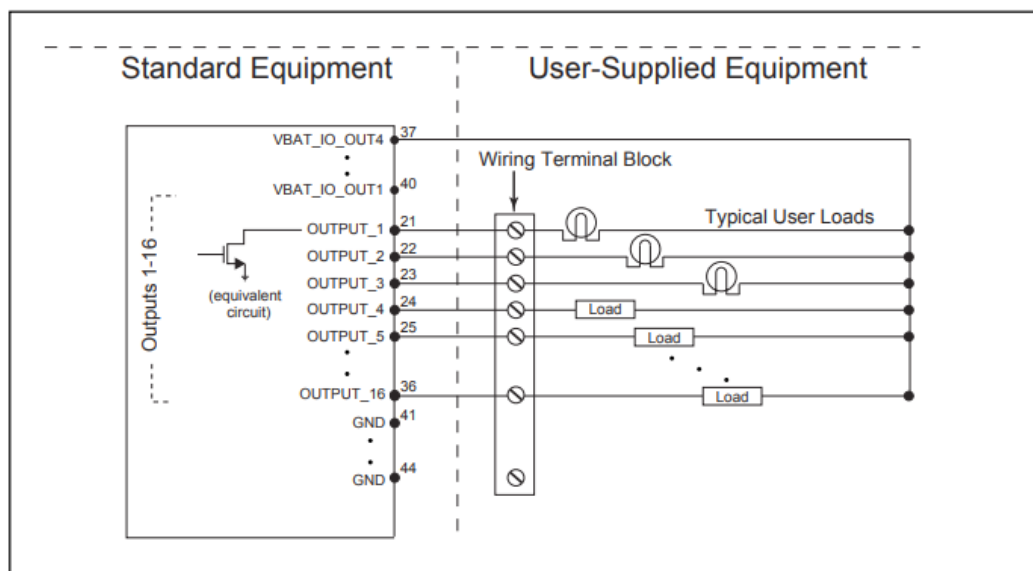
LD-60 -mobiilirobotissa on 16 digitaalituloja neljässä eri ryhmässä. Jokainen ryhmä voidaan kytkeä joko PNP- tai NPN- antureille sopivaksi. Tässä projektissa tuloportit kytketään kuvassa 4 näkyvän alemman esimerkin mukaan, jolloin valokennojen lähtöjen pitää olla PNP-tyyppisiä. Anturin kytkeytyessä päälle-tilaan, sähkö virtaa anturin kautta LD-60:n tulokortille. Myös DriveControl 54 häiriö -tieto tuodaan LD-60:n tulokortille. Luonnollisesti ryhmän COM-liitin kytketään tällöin nollaan (LD Platform User's Guide 2017, 63.)



Kuva 4. Esimerkki tulojen kytkennästä (LD Platforms User's Guide 2017, 63)

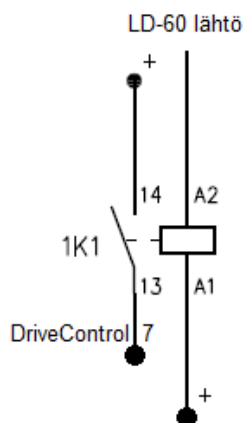
### 2.4.3 LD-60:n lähdöt

LD-60:llä on muutamia eri lähtötyyppejä: relelähtöjä, transistorilähtöjä ja triac-lähtöjä. LD-60:n lähtökortit ovat kuvassa 5 näkyviä ”sinking” eli virtaa ottavia ”open-drain” -lähtöjä. Kun logiikkaa käsketään laittamaan ko. lähtö päälle, transistorin hilalle tuodaan jännite, jolloin transistorin nielu maadoittuu lähteen kautta ja sähkö virtaa kuorman kautta lähtökortin nollaan. Tämän kaltainen kytkentä on yleistä Aasiassa tuotetussa elektroniikassa (LD Platform User’s Guide 2017, 64.)



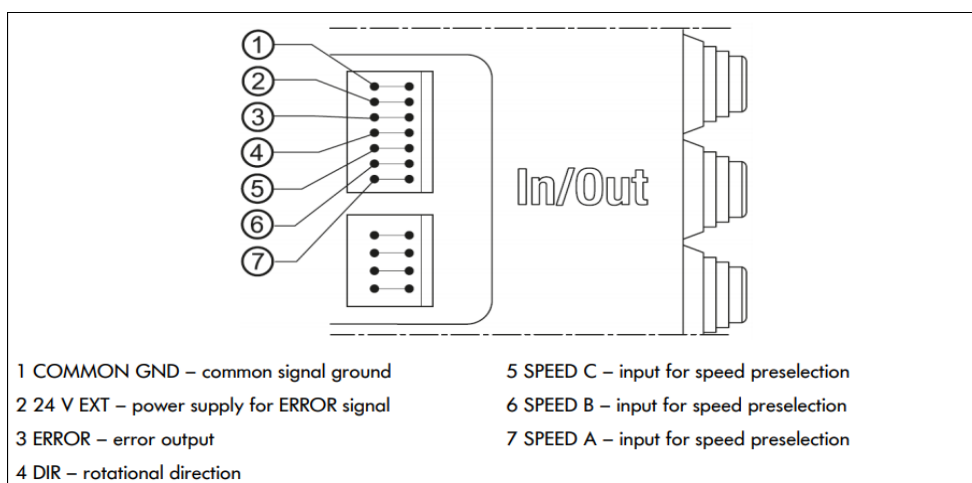
Kuva 5. Esimerkki lähtöjen kytkennästä (LD Platforms User’s Guide 2017, 64)

Tässä projektissa mobiilirobotin lähtöjä tarvitaan kuljettimen päälle laittamiseen ja kuljettimen suunnan valitsemiseen. Kuten edellisestä kappaleesta mainittiin, LD-60:ssä on virtaa ottavia lähtöjä ja Interroll DriveControl -manuaalista käy ilmi, että ohjaimen pinneihin pitää syöttää positiivista 24 voltin tasajännitettä. Tämä onnistuu kuvassa 6 näkyvällä apurelekytkennällä. Kuvan 6 ”+”-riviliittimet tulevat Siemensin DC/DC-konvertertilta, josta jännite tuodaan releen ”1K1”-kelalle ja sulkeutuvien koskettimien kautta DriveControl -ohjaimelle (Operating manual Interroll DriveControl 2019, 32.)



Kuva 6. Apurelekytkentä, jolla vaihdetaan virtaa ottava lähtö virtaa antavaksi.

Kuvasta 7 nähdään, että DriveControl-ohjaimen I/O-pinnit 5–7 ovat sekä nopeuden valitsemiselle, että ”päälle”-komentoa varten ja pinni neljä suunnan valitsemiselle. Jotta saadaan kuljetin käyntiin molempiin suuntiin, pinneihin 4 ja 7 tarvitaan jännitettä, eli tarvitsemme kaksi kuvan 6 mukaista kytkentää.



Kuva 7. DriveControl ohjaimen I/O-pinnit (Operating manual Interroll DriveControl 2019, 31.)

Ohjaimessa on I/O-pinnien lisäksi neljä dippikytkintä nopeuden valitsemiselle ja yksi käynnistys- ja pysäytysrampin asettamiselle sekä käyttöohjeessa taulukko nopeuden tarkkaan valintaan. Ramppi on asetettu kiinteästi niin, että dippikytkimen päällä ollessa, ohjain kiihdyttää kuljettimen maksiminopeuteen 0,39 sekunnissa. Ramppia ei tässä projektissa tarvita, koska laatikoiden massa on suhteellisen pieni (Operating manual Interroll DriveControl 2019, 37-41.)

## 3 MOBILEPLANNER JA ARCL-KOMENTOKIELI

MobilePlanner on Omron Adept Technologies -osakeyhtiön luoma välttämätön mobiilirobotin käyttöön liittyvä ohjelmisto. Ohjelmistolla mm. muunnetaan mobiilirobotin skannaama alue kartaksi, editoidaan luotua karttaa sekä ohjelmoidaan ja parametroidaan mobiilirobottia.

### 3.1 Kartan teko

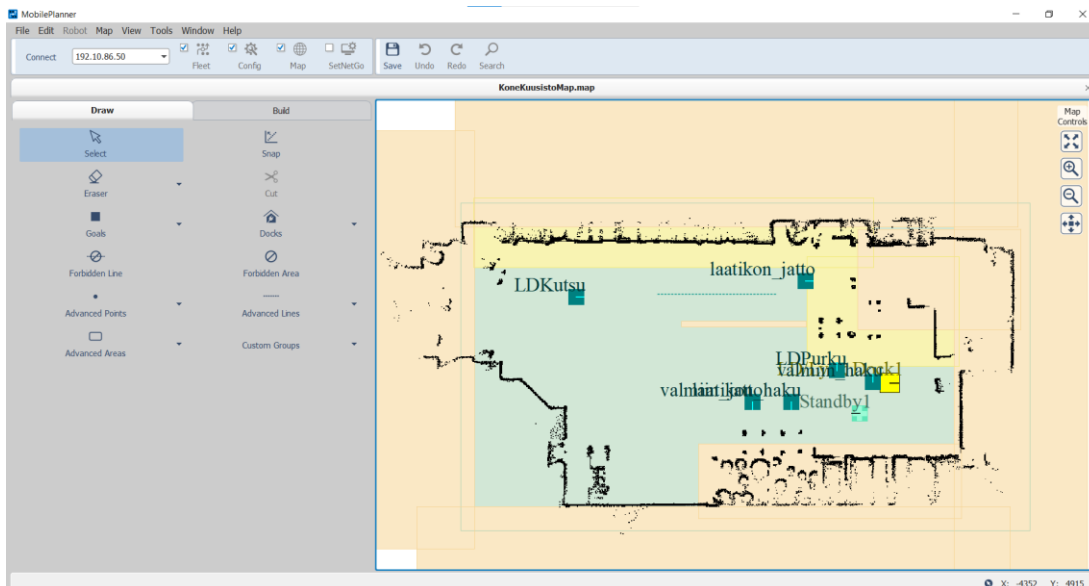
Mobiilirobotin kartan tekeminen tarkoittaa työalueen skannaamista ja erilaisten työpisteiden luomista kartalle. Seuraavissa kappaleissa em. vaiheet on kuvattu tarkemmin.

#### 3.1.1 Skannaus

Ennen kuin mobiilirobottia voidaan ohjelmoida, pitää työalue skannata ja skannaus muuttaa kartaksi. Skannaus tehdään niin, että mobiilirobottiin kiinnitetään langallinen joystick ja ajetaan hiljaa työalueen reunoja pitkin molempiin suuntiin, jolloin LD-60:n keulassa sijaitseva päälaser ”kartoittaa” työalueen. Tämän jälkeen skannaus ladataan mobiilirobotista PC:lle ja käännetään kartaksi MobilePlanner-ohjelmistolla.

#### 3.1.2 Maalit, odotuspisteet ja lataustelakka

Skannauksen jälkeen karttaan lisätään tarvittavat maalit eli ne pisteet, joihin mobiilirobotin halutaan menevän ohjelman aikana poimimaan tai antamaan laatikoita sekä odotuspaikka, johon mobiilirobotti menee odottamaan seuraavaa käskyä. Kartalle pitää myös asettaa mobiilirobotin lataustelakka. Maaleja pystyy myös luomaan kartalle skannauksen yhteydessä, mikä auttaa maalien tarkempaa sijoittamista isoissa tiloissa tai tiloissa, missä ei ole näkyviä kiintopisteitä. Kuvassa 8 on skannauksesta tehty kartta, johon on asetettu tarvittavat maalit, lataustelakka ja odotuspiste.



Kuva 8. Kuvankaappaus MobilePlanner -ohjelmiston käyttöliittymästä ja valmis kartta.

### 3.2 Työt

Tässä projektissa mobiilirobotti saa työkäskyjä Siemensin logiikalta. Yhteen työkäskyyn kuuluu esimerkiksi täyden laatikon nouto ja sen vienti vapaaseen varastopaikkaan tai tyhjän laatikon nouto ja sen vienti yhteistyörobotille.

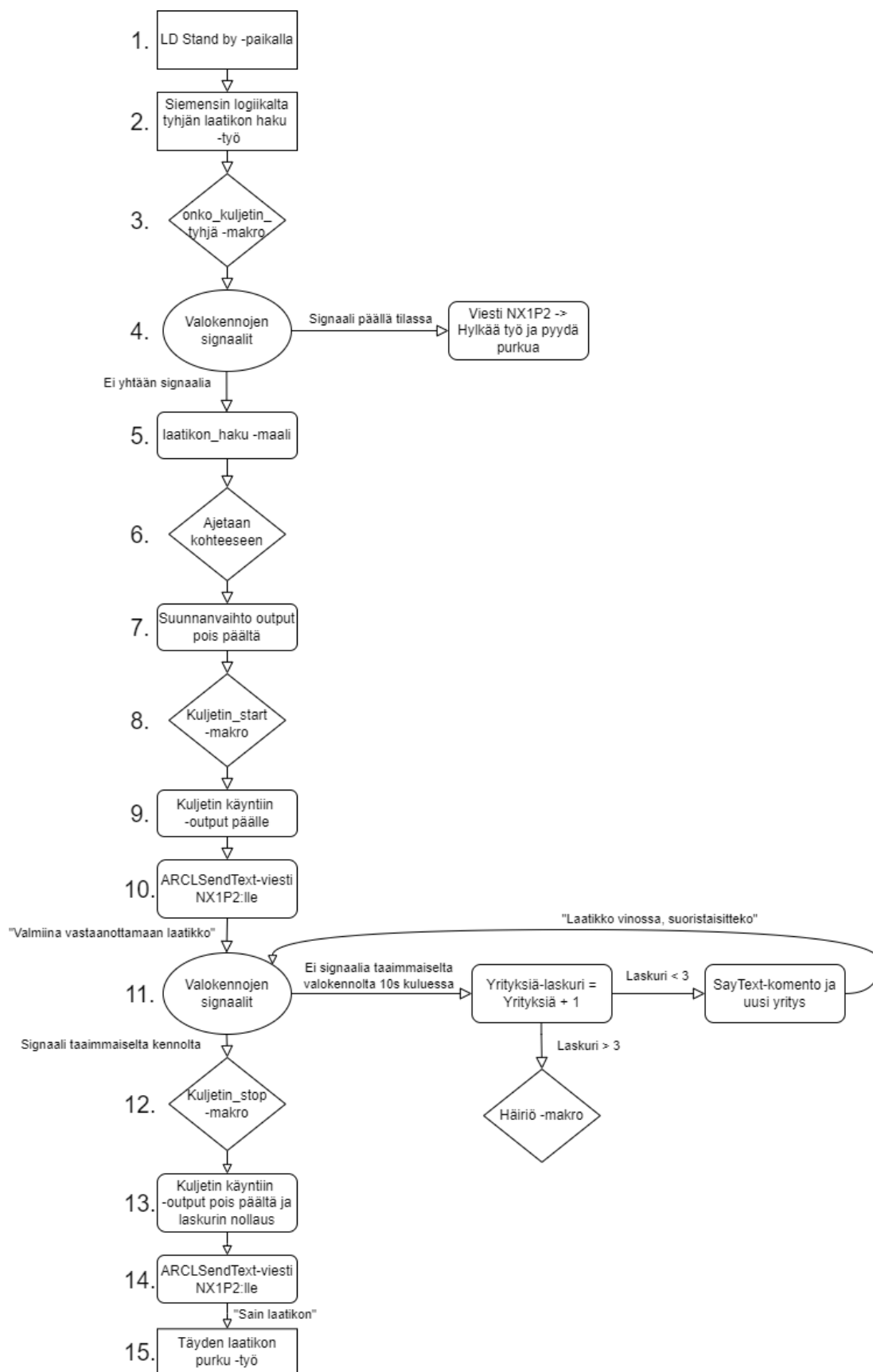
### 3.3 Mobiilirobotin ohjelmointi

Mobiilirobotin ohjelmointi on MobilePlanner -ohjelman graafisen käyttöliittymän kautta suhteellisen helppoa. Helpoin tapa on tehdä oma makro kaikille toiminnoille ja laittaa luotuja makroja maaleihin. Tässä projektissa oma makro luotiin mm. seuraaville toiminnoille:

- laatikon otto
- laatikon anto
- kuljetin käyntiin eteen
- kuljetin käyntiin taakse

sekä muutaman sekunnin väliajoin ajettava makro, jolla katsotaan, onko kuljettimella laatikkoa. Kuvassa 9 on vuokaavio yhdestä työkäskystä ja sen sisältämistä komennoista.





Kuva 9. Vuokaavio mobiilirobotin työkäskystä ja siinä käytettävistä makroista ja maaleista.

Kun mobiilirobotti on ollut toimeettomana kaksi minuuttia, se palaa odottelupisteeseen odottamaan töitä. Työkäskyn saapuessa Siemensin logiikalta NX1P2-logiikalle, käynnistää se sille työlle tarkoitettun ohjelmasekvenssin NX1P2-logiikassa. NX1P2-logiikan sekvenssi pitää huolen siitä, että kuljetaan mobiilirobotin työkäskyn ja Siemensin logiikan kanssa samassa linjassa eli Siemensin logiikalle kerrotaan, milloin mobiilirobotti on valmis vastaanottamaan laatikon ja milloin mobiilirobotti on valmis antamaan laatikon Siemensin logiikan ohjaamalle kuljettimelle.

Mobiilirobotin ensimmäinen tehtävä työkäskyn saapuessa on ajaa makro, joka tarkistaa, onko kyydissä laatikkoa. Mikäli jokin valokennoista on vaikuttuneena, mobiilirobotti lähettää NX1P2-logiikalle ARCL-viestin, joka aikaansaa työkäskyn perumisen ja laatikon purku -työn käynnistymisen. Jos mikään valokennoista ei ole vaikuttanut, mobiilirobotti lähtee kohti työn määräämää maalia.

Kuvan 9 kohdassa 7 ennen mobiilirobotin kuljettimen käynnistysmakroa kannattaa varmistaa, ettei suunnanvaihto ole jäänyt makaamaan päälle esimerkiksi virhetilanteen takia. Kun kuljetin on saatu käyntiin, mobiilirobotti lähettää NX1P2-logiikalle ARCL-viestin, että se on valmis vastaanottamaan laatikon. NX1P2 käyttää kättelytietoja hyväkseen ja ilmoittaa Siemensin logiikalle, että mobiilirobotti on paikalla kuljettimen päädyssä, oma kuljetin käynnissä valmiina vastaanottamaan laatikon, jolloin Siemensin logiikka laittaa kuljettimen käyntiin. Tämän jälkeen mobiilirobotti odottaa taaimaisen valokennon signaalia. Signaali aiheuttaa mobiilirobotin kuljettimen pysäytysmakron käynnistymisen. Mikäli signaalia ei tule 10 sekunnin kuluessa, mobiilirobotti kasvattaa ”yrityksiä” laskuria yhdellä, sanoo tarvitsevänsä apua ja yrittää uudelleen kaksi kertaa, jonka jälkeen menee häiriötilaan.

Kun laatikko on saatu mobiilirobotin kuljettimelle, mobiilirobotti lähettää ARCL-viestin NX1P2-logiikalle, joka ohjaa logiikan sekvenssiä eteenpäin askeleeseen, jossa kuitataan työ tehdyksi ja lähetetään laatikon purku -työ mobiilirobotille.

### 3.4 ARCL-komentokieli

ARCL-komentokieli tulee sanoista ”Advanced Robotics Command Language” ja se on yksinkertainen tekstipohjainen käsky- ja vastauskomentokieli. Se on kehitetty helpottamaan yhden tai useamman mobiilirobotin integrointia ulkoiseen automaatiojärjestelmään.

Komentokielen avulla mobiilirobottia ja siihen liitettyjä laitteita voidaan valvoa ja ohjata verkosta käsin. Komentoja voi lähettää mobiilirobotille esimerkiksi PuTTY-ohjelmalla tai komentokehoteesta Telnetillä (Adept ARCL Reference Guide 2016, 25).

Turvallisuuden vuoksi MobilePlanner-ohjelmasta täytyy parametroida mobiilirobotin ARCL- serveri päälle, antaa sille IP-osoite ja asettaa salasana, muutoin serverin porttia ei saa auki, eikä yhteyttä voida muodostaa (Adept ARCL Reference Guide 2016, 31).

## 4 OHJELMOITAVAT LOGIIKAT

Ohjelmoitava logiikka on teollisuuden tarpeisiin kehitetty pienikokoinen tietokone, jolla ohjataan ja valvotaan reaaliaikaisia automatisoituja prosesseja, kuten CNC-konetta tai kokoonpanolinjastoa. Tavalliseen pöytätietokoneeseen verrattuna ohjelmoitava logiikka on tehty kestävämpään teollisuuden vaativampiin olosuhteisiin kuten esimerkiksi värinää, kosteutta ja korkeampia lämpötiloja. Ne ovat myös, mallista ja merkistä riippuen, hyvin modulaarisia, mikä mahdollistaa tarvittavien komponenttien lisäämisen minimaalisella lisätyöllä (Programmable Logic Controllers 2006, 3.)

Ohjelmakierto tapahtuu ohjelmoitavassa logiikassa syklisesti. Syklin alussa luetaan tulokorttien muutokset ja tallennetaan ne tulojen prosessikuvaan, minkä jälkeen käydään logiikan muistissa oleva ohjelma läpi. Läpikäynnin jälkeen tehdään mahdolliset tulojen aiheuttamat muutokset lähtöihin ja tallennetaan lähtöjen prosessikuva, joka sitten tallennetaan lähtökorteille (Programmable Logic Controllers: A practical approach to IEC 61131-3 using CODESYS 2015, 39.)

### 4.1 IEC 61131

IEC 61131 on standardi ohjelmoitaville logiikoille. Sen kolmas osa, IEC 61131-3, määrittelee kaikki ohjelmoinnissa käytettävien muuttujien perusdatatyypit ja niiden koot, kolme eri ”ohjelmointikokonaisuutta” (Program organization unit tai POU), johon ohjelmakoodi kirjoitetaan ja viisi eri ohjelmointikieltä (Programmable Logic Controllers: A practical approach to IEC 61131-3 using CODESYS 2015, 199.)

#### 4.1.1 Muuttujat

Ohjelmoitavat logiikat käyttävät monenlaisia datatyyppejä muuttujissaan. Muuttujan käyttötarkoituksen mukaan valitaan muuttujalle sopiva datatyyppi. Esimerkiksi valokennon päällä/pois -signaalin datatyyppi on ”boolean (BOOL)”, koska sillä voi olla kullakin hetkellä vain yksi kahdesta arvosta, 0 tai 1, TOSI tai EPÄTOSI. Toinen hyvin yleinen datatyyppi on ”integer (INT)” eli kokonaisluku. Vaikka digitaaliset järjestelmät ymmärtävät parhaiten binääristä dataa, voi käyttäjän kannalta olla selvempää

käyttää joissain muuttujissa kymmenkantaista järjestelmää. INT-datatyyppissä on 16 bittiä, joista 15 bittiä esittää sen arvon ja yksi bitti etumerkin. Näin ollen INT voi olla arvoltaan  $2^{15}$  eli arvosta -32768 arvoon +32767. Tässä projektissa INT-tyyppiseen muuttujaan tallennetaan mm. mobiilirobotin sekvenssiohjelman askeleen arvo. Mikäli muuttujaan halutaan tallentaa tekstiä, on sillekin oma datatyyppi, ”STRING” (Practices Guide for NX1P Programming 2017, 23.)

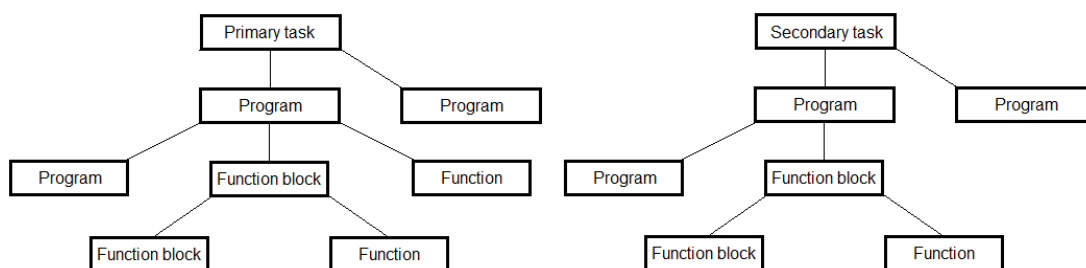
Suunniteltaessa ohjelman arkkitehtuuria, kannattaa miettiä millaisia toistuvia kokonaisuuksia projektissa on, koska erilaisten muuttujien tarve voi lisääntyä räjähdysmäisesti pienissäkin projekteissa. Kokonaisuuksia voisivat olla esimerkiksi kuljetin, pinoaja, mobiilirobotti jne. Näille kokonaisuuksille kannattaa tehdä oma ”Structure (STRUCT)” -datatyyppi, joka sisältää kaikki sille kokonaisuudelle tarvittavat muuttujat ja niiden datatyytit (Programmable Logic Controllers: A practical approach to IEC 61131-3 using CODESYS 2015, 244.)

#### 4.1.2 Program organization unit (POU)

”Program block (PRG tai PROG)” on korkeimman tason kokonaisuus, joka kutsuu alemman tason kokonaisuuksia ja sinne voidaan kirjoittaa mitä tahansa IEC 61131-3 -standardin määrittelemää ohjelmointikieltä. PROG on myös ainoa POU, johon voidaan määrittää globaaleita muuttujia. Globaaleiden muuttujien arvoa voi käyttää kaikki ohjelman osat (Programmable Logic Controllers: A practical approach to IEC 61131-3 using CODESYS 2015, 309.)

”Function block (FB)” on ohjelmalohko, jonka instansseja korkeamman tason kokonaisuus (PROG) kutsuu tarvittaessa. Function blokkiin kirjoitetaan esimerkiksi yksittäisen anturin tai toimilaitteen ohjelma. Jokainen FB:n instanssi varaa myös ohjelmoitavan logiikan muistista tarvittavan määrän tilaa paikallisille muuttujilleen ja parametreilleen, joita ovat mm. instanssissa käytettävät tulot, lähdöt ja apumuuttujat (Programmable Logic Controllers: A practical approach to IEC 61131-3 using CODESYS 2015, 295.)

”Function (F)” on ohjelmalohko, joka ei varaa omaa muistia ohjelmoitavalta logiikalta ja sillä on vain yksi lähtö. Lähdön arvo voi sisältää yhden arvon, mutta se voi myös sisältää matriisin arvoja, mikäli tuloparametrinä on matriisi arvoja. Funktiolla voidaan ohjelmoida esimerkiksi koneen tai laitteiston käyttötapa eli käsikäyttö tai automaatti. Kuvassa 10 on vielä esitetty ohjelmoitavan logiikan POU:t ja viivoilla merkityt mahdolliset ohjelmakutsut (Programmable Logic Controllers: A practical approach to IEC 61131-3 using CODESYS 2015, 269.)



Kuva 10. Program organization units eli ohjelmatyypit

#### 4.1.3 Ohjelmointikieliet

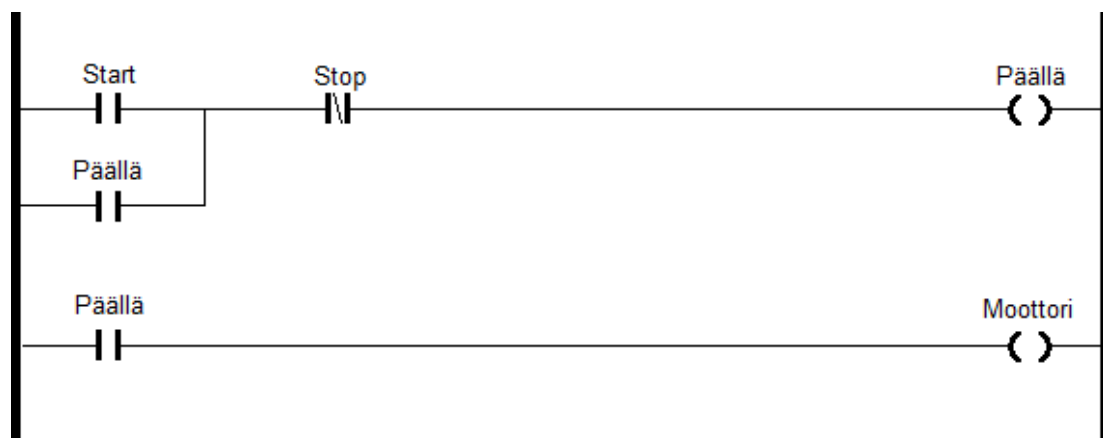
IEC 61131-3 -standardin määrittelemät ohjelmointikieliet ovat:

- Structured Text – ST
- Function Block Diagram – FBD
- Ladder Diagram – LD
- Instruction List – IL
- Sequential Function Chart – SFC

Kielistä FBD, LD ja SFC ovat graafisia ohjelmointikieliä ja ST ja IL ovat tekstipohjaisia ohjelmointikieliä. Tässä projektissa ohjelmointi on tehty suurimmaksi osaksi ST:llä ja LD:llä, joten käydään niiden toimintaa läpi sekä sivutaan FBD:tä, koska se on hyvin laajalle levinnyt ja yleisesti käytetty kieli (Programmable Logic Controllers: A practical approach to IEC 61131-3 using CODESYS 2015, 199.)

IEC 61131-3 -standardin tekstipohjaisista kielistä toinen, IL, on assembleria muistuttava alemman tason ohjelmointikieli. ST taas on korkean tason ohjelmointikieli, jossa monia toimintoja ja käskyjä voidaan antaa yhdellä ohjelmarivillä. ST muistuttaa syntaksiltaan C- ja Pascal-ohjelmointikieliä muista korkean tason ohjelmointikielistä. ST on kehitetty erityisesti monimutkaisten aritmeettisten funktioiden ohjelmoimiseen, taulukoiden manipuloimiseen ja tekstin, sekä tekstimuuttujien kanssa työskentelyyn (Programmable Logic Controllers: A practical approach to IEC 61131-3 using CODESYS 2015, 206.)

Tikapuukaavio eli ”Ladder Diagram” tai ”LD” on edelleen hyvin suosittu PLC-ohjelmoijien keskuudessa. Vaikka ST tai SFC olisi tehokkaampi valinta johonkin ohjelmointityöhön, saattaa ohjelmoijan valinta silti olla LD, koska se on helposti omaksuttava kieli ja sen graafinen esitystapa muistuttaa perinteistä sähkökaaviota. LD:llä tehty ohjelma koostuu ohjauksista (tuloista), jotka toteuttavat esimerkiksi matemaattisia operaatioita, logiikkaa tai laskureita ja asettavat sitten lähtöjä päälle tai pois päältä, riippuen siitä, mitä ohjelman on tarkoitus tehdä. Seuraavassa kuvan 11 esimerkissä on LD:llä toteutettu start/stop -toiminto (Programmable Logic Controllers: A practical approach to IEC 61131-3 using CODESYS 2015, 207.)



Kuva 11. Tikapuukaavio eli LD-esimerkki

Kuvan 11 esimerkissä ”Start” ja ”Päällä” ovat sulkeutuvia koskettimia ja ”Stop” on avautuva kosketin. ”Start”-nappulaa painettaessa ”Päällä”-kela aktivoituu ja sen koskettimet sulkeutuvat. ”Stop”-nappulaa painettaessa, sen koskettimet avautuvat ja pito-piiri päästää, jolloin ”Päällä”-kela kytkeytyy pois päältä. Nykyajan ohjelmoitavien

logiikkojen valmistajat mahdollistavat monimutkaisienkin funktioiden toteuttamisen LD:llä, tarjoamalla valmiita ohjelmalohkoja käyttäjilleen.

FBD eli ”Function Block Diagram” on toinen hyvin yleisesti käytetty graafinen ohjelmointikieli. Yksinkertaisimmillaan siinä yhdistellään funktioita tai function blockeja ja em. lohkot voivat sisältää loogisia operaatioita, laskureita tai ajastimia. Funktiot ja function blockit voivat myös olla itse rakennettuja, jolloin monimutkaisiakin toimintoja saadaan ohjelmoitua FBD:llä. FBD on myös erinomainen kieli aloittelijoille, koska sen graafinen esitystapa on LD:n tapaan yksinkertainen ja selkeä. FBD sopii myös pieniin projekteihin koko ohjauksen ohjelmointiin, mutta isommissa projekteissa rakenne muuttuu lopulta sekavaksi, jolloin ST on yleensä parempi vaihtoehto (Programmable Logic Controllers: A practical approach to IEC 61131-3 using CODESYS 2015, 206.)

#### 4.2 Siemensin ohjelmoitava logiikka

Demon varastokuljettimia ja robotin kuljettimia ohjaa Siemensin ohjelmoitava logiikka. Kyseinen logiikka myös jakaa töitä mobiilirobotille. Ajan puutteen vuoksi Siemensin ja Omronin logiikan välinen kommunikaatio tehtiin kättelytiedoilla. Kättelytieto tarkoittaa sitä, että muutama lähtö Siemensin logiikalta johdotetaan suoraan Omron NX1P2-logiikan tuloon ja toisinpäin. Näin ollen prosessin elintärkeitä anturitietoja saatiin vaihdettua logiikkojen kesken. Edistyneempi keino olisi ollut käyttää olemassa olevaa EthernetIP-väylää, mutta kommunikointiin ei ollut valmiita ohjelmalohkoja kummaltakaan valmistajalta, joten väylän kautta kommunikointi päätettiin jättää myöhemmälle ajankohdalle.

#### 4.3 Omron NX1P2

Omron NX1P2 on kompakti logiikka, jossa on neljätoista digitaalista tuloa ja kymmenen digitaalista lähtöä sekä kaksi Ethernet-porttia väyläkommunikointia varten. Logiikassa on myös 1,5 megatavua omaa muistia käyttäjän ohjelmia varten. (Omron industrial automation www-sivut 2021.)



Tämän logiikan tarkoitus projektissa on toimia tiedonvälittäjänä Siemensin logiikan ja mobiilirobotin sekä yhteistyörobotin välillä ja ohjata mobiilirobotin töihin liittyviä sekvenssejä. Ohjelmointi tapahtuu Sysmac Studio -ohjelmointiympäristössä.

#### 4.3.1 Pääohjelma ja taskit

NX1P2-pääohjelmassa pyörii tässä projektissa kaksi jaksollista taskia eli tehtävää, ns. primaarinen ja ei-primaarinen. Logiikan asetuksista voidaan määrittää, kuinka usein kukin ei-primaarinen tehtävä ajetaan läpi. Primaarisessa tehtävässä kutsutaan kahta eri ohjelmaa, ensimmäinen on nimeltään ”AIV1\_Sekvenssi”, jossa ohjataan mobiilirobotin sekvenssiä ja toinen on ”ComMuuttujat”, jossa luetaan logiikan tulot ja kirjoitetaan lähdöt. Primaarista tehtävää ajetaan 2 millisekunnin välein läpi. Logiikan ei-primaarisessa tehtävässä kutsutaan ohjelmaa, jossa on kaikki mobiili- ja yhteistyörobotin kommunikointiin tarvittavat FBt ja tätä tehtävää ajetaan 20 millisekunnin välein läpi. Kuvassa 12 näkyy ohjelmointiympäristöstä otettu kuvankaappaus, jossa eriteltyinä em. tiedot.

#### 1-6.Task Settings

Task Name	Period/Execution Conditions	Detailed Execution Conditions	Task Period Exceeded Detection	Task Detection Time	Timeout	Execution Priority	Variable Access Time [%]
<b>Primary periodic task</b>							
PrimaryTask	2ms		Detect	10ms		4	3
<b>Periodic task</b>							
PeriodicTasko	20ms		Detect	100ms		17	3
<b>Unit</b>				<b>Task Name</b>			
<b>EtherCAT Network Configuration</b>							
<b>CPU/Expansion Racks</b>							
	Built-in I/O	Built-in I/O Settings			PrimaryTask		
	OptionBoard	Option Board Settings					
	NXBusMaster	NX Bus Master					
<b>Task Name</b>		<b>Assigned programs</b>				<b>Initial status</b>	
<b>PrimaryTask</b>							
	1	AIV1_Sekvenssi				Run	
	2	ComMuuttujat				Run	
<b>PeriodicTasko</b>							
	1	Communications				Run	

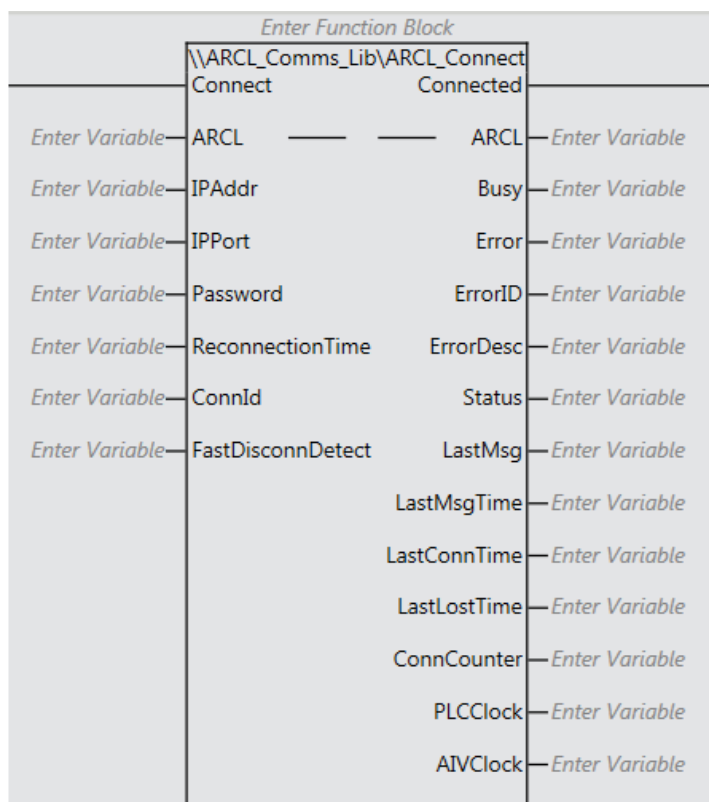
Kuva 12. Kuvankaappaus Sysmac Studion ”Task Settings” -kohdasta

#### 4.3.2 Mobiilirobotin ohjaukseen liittyvät ohjelmat

Ohjelmassa ”ComMuuttujat” asetetaan NX1P2-logiikan jokainen sisäänrakennettu tulo ja lähtö johonkin muuttujaan, jotta niihin voidaan viitata kuvaavimmilla nimillä muissa ohjelmissa. On paljon helpompaa ymmärtää, mihin tulomuuttujaa nimeltä

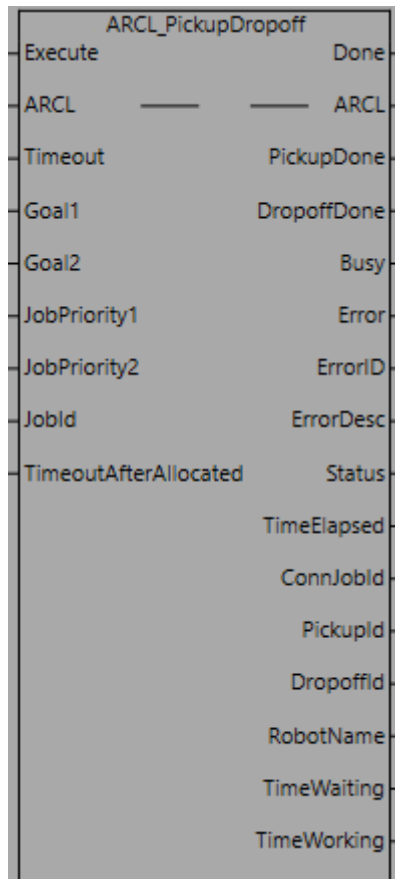
”Com.Control.RequestStart” käytetään kuin ”input08”, vaikka ne tekevätkin saman asian. Myös ohjelman virheenjäljitys helpottuu.

”Communications” ohjelmassa kutsutaan, asetetaan päälle ja parametroidaan kaikki FBt, mitä mobiili- ja yhteistyörobotin ohjaamiseen, valvomiseen ja niiden töiden antamiseen tarvitaan. Omron on tehnyt runsaan määrän valmiita ohjelmalohkoja mobiilirobotin ohjaamiseen ohjelmoitavalla logiikalla ja tärkein niistä löytyy heti ensimmäisenä ARCL-kirjastosta.



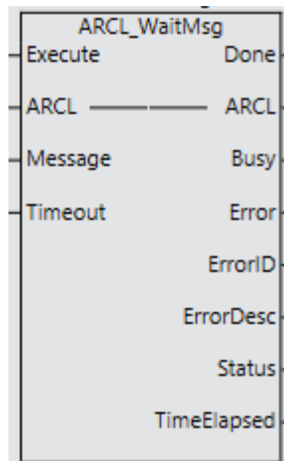
Kuva 13. ARCL\_Connect FB (Sysmac Function Block Library for Omron Mobile Robots 2019, 20)

Kuvan 13 ARCL\_Connect FB muodostaa yhteyden mobiilirobotin ja ohjelmoitavan logiikan välille. Tuloparametreinä FB haluaa mobiilirobotin IP-osoitteen, portin, salasanan, uniikin tunnisteiden yhteydelle sekä ajan, minkä jälkeen aloitetaan uudelleenyhdistäminen. IP-osoitteella, portilla ja salasanalla määritellään robotti tai laivueen ohjain, johon halutaan yhdistää ja ”ConnId” toimii tunnisteena komennoille, joita lähetetään tämän yhteyden yli. Lähtöparametrejä FB:stä löytyy myös iso kasa. Niistä suurin osa on tarkoitettu virheenjäljitykseen tai mobiilirobotin valvontaan (Sysmac Function Block Library for Omron Mobile Robots 2019, 20.)



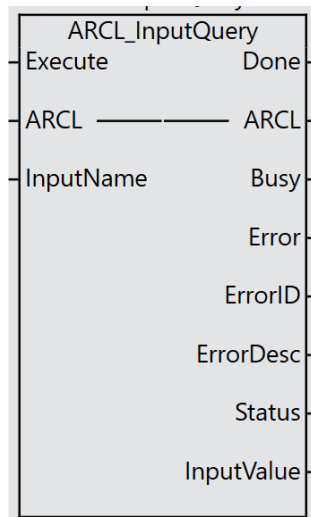
Kuva 14. ARCL\_PickupDropoff FB (Sysmac Function Block Library for Omron Mobile Robots 2019, 63)

ARCL\_PickupDropoff FB lähettää nimensä mukaisesti hae ja vie -tehtävän mobiilirobotille. Tuloparametreinä lohko vaatii kaksi maalia, toisessa otetaan laatikko kyytiin ja toisessa puretaan kyydistä sekä prioriteetin em. maaleille ja myös tunnisteen työlle. Ohjelman testausvaiheessa huomattiin, että kaikkien lohkojen käyttämättömille tuloparametreille kannattaa antaa joko oikeaa tyyppiä oleva tyhjä muuttuja tai jokin staattinen arvo, sillä muuten lohko antaa mystisen virheilmoituksen tai sitä ei suoriteta lainkaan. Kuvassa 14 näkyy ARCL-kirjastosta leikattu ARCL\_PickupDropoff-FB (Sysmac Function Block Library for Omron Mobile Robots 2019, 63.)



Kuva 15. ARCL\_WaitMsg FB (Sysmac Function Block Library for Omron Mobile Robots 2019, 91)

Kuvan 15 ARCL\_WaitMsg FB on yksi tärkeimmistä mobiilirobotin sekvenssiin liittyvistä lohkoista. Tälle lohkolle annetaan vain string-tyyppinen muuttuja ”Message”-tuloon. Kun tämä suoritetaan, lohko odottaa muuttujassa olevaa tekstiä mobiilirobotilta. Mobiilirobotin maaliin on kirjoitettu tehtäviä, joita se suorittaa järjestyksessä päästyään maaliin. Yksi näistä tehtävistä on ”SendMsg”, jolla saadaan viesti mobiilirobotilta NX1P2-logiikalle. Kun oikea viesti saapuu ohjelmoitavalle logiikalle, se kääntää tämän lohkon ”Done”-lähden päälle tilaan. Näin saamme tiedon esimerkiksi Siemensin logiikalle, että mobiilirobotti on valmiina vastaanottamaan laatikon, ja voimme hypätä seuraavaan askeleeseen sekvenssissä (Sysmac Function Block Library for Omron Mobile Robots 2019, 91.)



Kuva 16. ARCL\_InputQuery FB (Sysmac Function Block Library for Omron Mobile Robots 2019, 38)

Mobiilirobotin ohjelmassa on ajoittain toistettava makro, joka tarkkailee kuljettimen keskimmäisen valokennon signaalia. Kuvassa 16 olevalla ARCL\_InputQuery -lohkolla NX1P2 kysyy mobiilirobotilta em. signaalia. Lohko tarvitsee ”InputName”-tuloparametriksi string-tyyppisen muuttujan, johon kirjoitetaan sen tulon nimi, jonka arvo halutaan tietää. Mobiilirobotin käyttöönotossa käytettävät tulot ja lähdöt on nimetty, jotta niihin on helpompi viitata. Tällä tavoin NX1P2 saa välitettyä reaaliajassa tietoa Siemensin logiikalle siitä, onko mobiilirobotin kyydissä laatikko. Tieto siitä, onko kyydissä laatikko, on tärkeää esimerkiksi siinä vaiheessa, kun ohjelmassa tapahtuu jokin virhe ja järjestelmä joudutaan käynnistämään uudelleen. Silloin Siemensin logiikka ei voi antaa mobiilirobotille ”nouda laatikko” -tehtävää, ennen kuin kyydissä oleva laatikko on viety varastoon (Sysmac Function Block Library for Omron Mobile Robots 2019, 38.)

#### 4.4 Omron TM

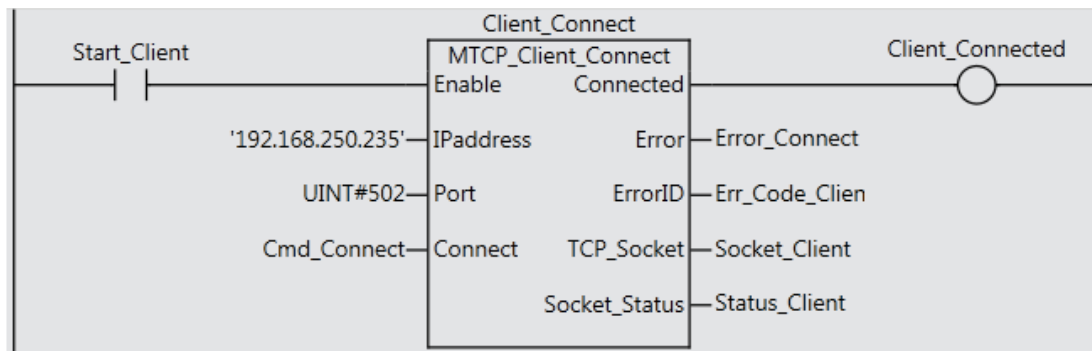
Omron TM on Omronin yhteistyörobottibrändi. Yhteistyörobotti eroaa tyypillisestä teollisuusrobotista siten, että sen ollessa yhteistyömoodissa ihminen voi työskennellä robotin työalueella robotin ollessa käynnissä. Tavanomainen teollisuusrobotti taas vaatii ympärilleen standardin mukaisen turvaratkaisun ja pääsy robotin työalueelle sen ollessa käynnissä evätään esimerkiksi ovesa sijaitsevalla turvaovikytkimellä. Kun ovi avataan, turvapiiri rikkoutuu ja robotti menee turvalliseen tilaan. Omron TM -yhteistyörobotti on pieni ja näppärä, joten se soveltuu erinomaisesti tähän projektiin. Yhteistyörobotissa on 700-1300 mm:n ulottuvuus ja 4-14 kg:n kuormankantavuus mallista riippuen sekä 16 digitaalista tuloa ja lähtöä (Omron industrial automation www-sivut, 2021.)

##### 4.4.1 Modbus

Modbus on alun perin Modicon yhtiön, nykyisin Schneider Electricin, vuonna 1979 luoma asiakas/palvelin -mallia noudattava sarjamuotoinen tiedonvälitysprotokolla ohjelmoitaville logiikoille. Myöhemmin Modbus-väylästä kehitettiin Ethernet-ympäristöön soveltuva versio, joka käyttää TCP/IP-protokollapinoa. Tässä projektissa tiedonvälitys NX1P2-logiikan ja Omron TM -yhteistyörobotin välillä tapahtuu Modbus/TCP:n avulla (modbus.org www-sivut, 2021.)

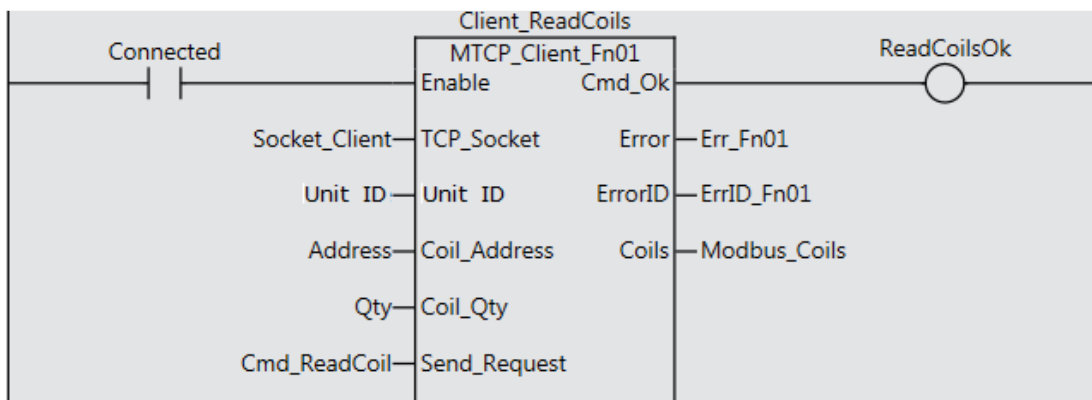
##### 4.4.2 Yhteistyörobotin kommunikointiin liittyvät ohjelmat

Kuten edellisestä kappaleesta kävi ilmi, kommunikointi NX1P2-logiikan ja yhteistyörobotin välillä tapahtuu TCP/IP:n yli Modbus-väylän avulla. Omron on tehnyt tähän tarkoitukseen valmiita ohjelmalohkoja, millä voi lukea ja kirjoittaa yhteistyörobotin keloja sekä rekistereitä. Tässä projektissa yhteistyörobotin pitää ilmoittaa Siemensin logiikalle, milloin robotti on täyttämässä kuljettimella seisovaa laatikkoa. Se onnistuu aiemmin mainituilla kättelytiedoilla, joista yksi on varattu kuljettimen lukitsemiseen. Yhteistyörobotti ”lukitsee” kuljettimen laatikon täytön ajaksi vaihtamalla yhden bitin muistirekisteristään ”TOSI”-tilaan, NX1P2 lukee tämän bitin ja peilaa sen Siemensin logiikalle.



Kuva 17. MTCP\_Client\_Connect FB (MTCP\_NJ\_Client 2016, 1)

Ennen kuin tiedonsiirto yhteistyörobotin ja Siemensin logiikan välillä on mahdollista, pitää yhteys ensin avata yhteistyörobotin ja NX1P2-logiikan välille. Kuvan 17 MTCP\_Client\_Connect -lohkolle annetaan tuloparametreina STRING-, UINT- ja BOOL-tyyppiset muuttujat, joihin kirjoitetaan verkossa sijaitsevan Modbus-palvelimen IP-osoite ja portti. Tässä tapauksessa Modbus-palvelin on yhteistyörobotti. BOOL-tyyppisellä muuttujalla pyydetään lupa yhteyden avaamiseen. Lähtöparametrejä lohkoissa on muutama häiriötilaa ilmaiseva muuttuja sekä STRUCT-tyyppinen muuttuja, mihin kirjoitetaan Modbus-asiakkaan eli Omron NX1P2:n IP-osoite ja portti (MTCP\_NJ\_Client 2016, 1.)



Kuva 18. MTCP\_Client\_Fn01 FB (MTCP\_NJ\_Client 2016, 1)

Yhteistyörobotin kelojen lukemiseen käytetään kuvan 18 lohkoa. TCP\_Socket -strukturi saadaan edellisestä MTCP\_Client\_Connect -lohkosta. TCP/Modbus-verkossa liikkuville sanomille täytyy antaa tunniste, jotta palvelimet tietävät viestien lähettäjän ja järjestyksen. ”Unit ID” on tavun pituinen (0–FF) heksaluku, joka toimii tunnisteena viesteille. Lohkolle tarvitsee myös kertoa ensimmäisen luettavan kelan osoite rekisterissä ja luettavien kelojen määrä kahden tavun pituisina heksalukuina. Lohkon

alimmainen BOOL-tyyppinen muuttuja "Send\_Request" lähettää palvelimelle rekisterin lukupyynnön. Palautuksena palvelin lähettää pyydettyjen kelojen arvoja taulukkomuodossa BOOL-tyyppisinä muuttujina (MTCP\_NJ\_Client 2016, 1.)



## 5 TULOKSET

Työn tarkoituksena oli luoda demo näytettäväksi lähialueen yrityksille ja tutustua mobiiliroboti- sekä yhteistyörobotiteknikkaan. Demossa mobiiliroboti haki tyhjiä laatikoita oikosulkumoottorikäyttöiseltä kuljettimelta ja toimitti niitä toiselle kuljettimelle, jossa yhteistyöroboti täytti ne. Täydet laatikot mobiiliroboti vei painovoimaiselle varastoradalle. Demon moottorikäyttöisiä kuljettimia ohjasi Siemensin logiikka ja kommunikointi tapahtui Omronin NX1P2 logiikan kautta. Siemensin ja Omronin logiikan välisessä kommunikaatiossa käytettiin kättelytietoja, yhteistyörobotin ja Omronin logiikan välinen kommunikointi tapahtui Modbus-väylän yli ja Omronin logiikan ja mobiilirobotin välinen kommunikointi taas Wifin yli ARCL-komentokieltä hyväksi käyttäen. Projekti onnistui aikatauluun nähden hyvin ja valmista demoa kävi katsomassa yllättävän moni yritys ja olisi käynyt useampikin, mikäli globaali pandemia-tilanne olisi sen sallinut.

Mobiilirobotin ohjelmointiin käytetystä ARCL-kirjastosta löytyi kaikki projektissa tarvittavat FB:t, mutta tässä demossa käytetty ARCL\_PickUpDropOff oli kankea ja tulen korvaaman sen tulevissa projekteissa FB:lla, joka antaa vain yhden työn mobiilirobotille. Kyseinen FB antoi – nimensä mukaisesti – kaksi työtä, joista toinen oli laatikon kyytiin otto ja toinen työ oli laatikon purku. FB toimii parhaiten täydellisessä maailmassa, jossa ei koskaan tule virhetilanteita. Ongelmaksi tuli virhetilanne kesken työn, koska ohjelman uudelleen aloituksessa oli pakko antaa mobiilirobotille kaksi työtä, vaikka toinen työ olisi ollut jo tehty. Sekä mobiilirobotin että Omronin logiikan ohjelmien virheen käsittely ja häiriöistä palautuminen jäi todella vajavaiseksi, mutta sain projektista erittäin hyviä ideoita, miten toteuttaa ongelmatilanteista palautuminen tulevia projekteja varten. Demon myötä Omronin logiikoiden ohjelmointiympäristö Sysmac Studio tuli tutummaksi ja opin myös Modbus-väylästä muutamia tärkeitä seikkoja. Esimerkiksi sen, että samalla logiikan ohjelmakerroilla ei kannata yrittää lukea ja kirjoittaa Modbus-rekistereitä.

## LÄHTEET

- Adept ARCL Reference Guide. Adept Technology, Inc. Viitattu 14.7.2021. [https://assets.omron.eu/downloads/manual/en/v11/i617\\_advanced\\_robotics\\_command\\_language\\_\(arcl\)\\_reference\\_manual\\_en.pdf](https://assets.omron.eu/downloads/manual/en/v11/i617_advanced_robotics_command_language_(arcl)_reference_manual_en.pdf)
- Dag H. Hanssen. Programmable Logic Controllers: A practical approach to IEC 61131-3 using CoDeSys. Wiley. Viitattu 15.7.2021
- ia.omron www-sivut. Viitattu 16.8.2021. <https://industrial.omron.fi/fi/products/collaborative-robots>
- ia.omron www-sivut. Viitattu 9.10.2020. <http://www.ia.omron.com/products/family/3664/specification.html>
- ia.omron www-sivut. Viitattu 29.11.2021. <https://www.ia.omron.com/products/family/3650/lineup.html>
- Interroll www-sivut. Viitattu 9.10.2020. [https://www.interroll.com/fileadmin/Downloads/User\\_Manuals/Controls/DriveControl/User\\_Manual\\_DriveControl\\_EN.pdf](https://www.interroll.com/fileadmin/Downloads/User_Manuals/Controls/DriveControl/User_Manual_DriveControl_EN.pdf)
- LD Platforms User's Guide. Omron Adept Technologies, Inc. Viitattu 14.7.2021. [https://assets.omron.eu/downloads/manual/en/i611\\_mobile\\_robots\\_oem\\_-\\_ld\\_platform\\_users\\_manual\\_en.pdf](https://assets.omron.eu/downloads/manual/en/i611_mobile_robots_oem_-_ld_platform_users_manual_en.pdf)
- modbus.org www-sivut. Viitattu 16.8.2021. <https://modbus.org/faq.php>
- Operating manual Interroll DriveControl. Interroll Engineering GmbH. Viitattu 14.7.2021. [https://www.interroll.com/fileadmin/user\\_upload/AMERICAS/User\\_Manuals/en\\_DriveControl\\_20-54\\_V21\\_EN.pdf](https://www.interroll.com/fileadmin/user_upload/AMERICAS/User_Manuals/en_DriveControl_20-54_V21_EN.pdf)
- Phoenix Contact www-sivut. Viitattu 9.10.2020. <https://www.phoenixcontact.com/online/portal/fi?uri=pxc-oc-itemdetail:pid=2908262>
- Practices Guide for NX1P Programming. Omron corporations. Viitattu 16.7.2021. <https://www.support-omron.fr/telechargements/documentations/2019-02-07%20-%2011-22-02%20-%20549749263/P122-E1-01%20Practices%20Guide%20for%20NX1P%20Programming.pdf>
- Sysmac Function Block Library for Omron Mobile Robots. Omron corporations. Viitattu 18.7.2021. [https://www.support-omron.fr/telechargements/documentations/2019-02-05%20-%2015-36-31%20-%2058916612/ARCL\\_Comms\\_Lib\\_1.0\\_RC7\\_StartingGuide.pdf](https://www.support-omron.fr/telechargements/documentations/2019-02-05%20-%2015-36-31%20-%2058916612/ARCL_Comms_Lib_1.0_RC7_StartingGuide.pdf)
- Virmanen, V. 2020. Field application engineer, Omron Electronics Oy. Säskylä. Haastattelu. 17.10.2020. Haastattelijana Mikko Leppälä. Muistiinpanot Haastattelijan hallussa.
- William Bolton. Programmable Logic Controllers. Newnes. Viitattu 13.7.2021.