

Jyri Lehtinen

KONENÄKÖÄ HYÖDYNTÄVÄN RO- BOTTIJÄRJESTELMÄN TOTEUTUS OPETUSKÄYTTÖÖN

Opinnäytetyö

Tekniikan ammattikorkeakoulututkinto

Robotiikan ja tekoälyn koulutus

2024



**Kaakkois-Suomen
ammattikorkeakoulu**

Tutkintonimike	Tekniikan ammattikorkeakoulututkinto, insinööri (AMK)
Tekijä/Tekijät	Jyri Lehtinen
Työn nimi	Konenäköä hyödyntävän robottijärjestelmän toteutus opetuskäyttöön
Toimeksiantaja	Kaakkois-Suomen Ammattikorkeakoulu Oy
Vuosi	2024
Sivut	50 sivua
Työn ohjaaja(t)	Teemu Jokela, Niko Puurtinen

TIIVISTELMÄ

Tämä opinnäytetyö käsittelee konenäköä hyödyntävän robottijärjestelmän toteutusta opetuskäyttöön Kaakkois-Suomen ammattikorkeakoululle. Työn tavoitteena oli luoda toimiva ja monipuolinen robottijärjestelmä, jota opiskelijat voivat hyödyntää robotiikan ja konenäön tutkimisessa sekä käytännön harjoitteissa ja projektitöissä. Työssä oli keskeistä selvittää, miten KUKA KR 6 900-2-robotti, Cognex 2001M-353-konenäköanturi ja Beckhoffin CX9020 sulautettu PC saadaan kommunikoidaan keskenään. Haasteena oli laitteiden erilaisten kommunikaatioprotokollien yhteensovittaminen.

Ratkaisuna käytettiin OPC UA -väylää konenäköanturin ja PLC:n välillä sekä EtherCAT-väylää PLC:n ja robotin välillä. Tämän mahdollistamiseksi järjestelmään lisättiin Beckhoffin PLC ja EtherCAT-yhdyskäytävä. Järjestelmällä toteutettiin yksinkertainen konenäkösovellus järjestelmän testaamiseksi. Sovelluksessa robotti poimii kappaleita pöydältä konenäköanturilta saatavien tietojen mukaan. Työssä piti hyödyntää monipuolisesti tutkimus- ja kehittämismenetelmiä, kuten suunnittelua, konfigurointia, kalibroida ja ohjelmointia. Työhön kuului myös hieman mekaanisia asennustöitä, lähiverkon kytkentöjä ja pie-nisjännitekytkentöjä.

Järjestelmän toteuttamiseksi piti konenäköanturi konfiguroida ja kalibroida, sekä opettaa sille tunnistettava kappale. PLC:lle piti tehdä käyttöönotto, konfigurointia ja ohjelmointia. Robotille piti tehdä kommunikaatioon liittyvää konfigurointia ja testausohjelma KUKAn KRL -ohjelmointikielellä.

Tuloksena saatiin toimiva robottijärjestelmä, joka täytti tilaajan asettamat tavoitteet. Työssä onnistuttiin yhdistämään konenäkö ja robotti ulkoisen logiikan avulla, sekä luotiin alusta opiskelijoille itsenäiseen työskentelyyn ja monipuolisten robotiikan sovellusten harjoitteluun sekä kehittämiseen. Työssä tunnistettiin myös useita jatkokehityskohteita, kuten järjestelmän laajentaminen Profinet-väylällä ja OPC UA:n Pub/Sub -ominaisuuden hyödyntäminen.

Asiasanat: konenäkö, robotiikka, kenttäväylät, tiedonsiirto

Degree title	Bachelor of Engineering
Author (authors)	Jyri Lehtinen
Thesis title	Implementation of a machine vision-based robot system for educational use
Commissioned by	South-Eastern Finland University of Applied Sciences
Time	2024
Pages	50 pages
Supervisor	Teemu Jokela, Niko Puurtinen

ABSTRACT

The focus of this thesis is the implementation of a machine vision-based robot system for educational purposes at South-Eastern Finland University of Applied Sciences. The goal was to create an operational and flexible robot system that students can utilise for robotics and machine vision research, practical exercises, and project work. The key part of the thesis was to establish communication between KUKA KR 6 900-2 robot, Cognex 2001M-353 machine vision sensor, and Beckhoff CX9020 embedded PC. The challenge was integrating the various communication protocols of the devices.

The solution used OPC UA for communication between the machine vision sensor and the PLC, while EtherCAT fieldbus was employed between the PLC and the robot. To enable this setup, a PLC and an EtherCAT bridge were needed. A simple machine vision application was implemented to test the system in which the robot picks objects from a table based on position information from the machine vision sensor. The project involved research and development methods such as design, configuration, calibration, and programming. Physical work was also involved, like mechanical installation, network connections, and low-voltage DC wiring.

The implementation included configuring and calibrating the machine vision sensor to recognize and measure the target object. The PLC was commissioned and programmed, while the robot was configured for communication and tested using a program developed with KUKA Robot Language (KRL).

The result was a functional robot system that meets the objectives of the client. I successfully integrated machine vision and the robot with a PLC. Resulting robot system works as a good platform for students to practice their skills and develop new applications. Several areas for further development were also identified: extending the system with a PROFINET network and making use of the Pub/Sub capabilities of OPC UA.

Keywords: machine vision, robotics, fieldbus, data communication

SISÄLLYS

1	JOHDANTO.....	6
2	YRITYSESITTELY.....	7
3	JÄRJESTELMÄN SUUNNITTELU.....	8
3.1	Työn määritelmä.....	8
3.2	Robotiikka.....	9
3.2.1	Koordinaatistot.....	9
3.2.2	Teollisuusrobotit.....	11
3.2.3	KUKA KR 6 R900-2	13
3.3	Konenäkö	14
3.3.1	Cognex konenäköanturi	15
3.4	Robottijärjestelmän kommunikaatio	16
3.4.1	EtherCAT	19
3.4.2	OPC UA.....	20
3.5	Ohjelmoitavat logiikat.....	21
4	JÄRJESTELMÄN TOTEUTUS	23
4.1	Järjestelmän suunnittelu	23
4.2	Järjestelmän asennus ja konfigurointi.....	26
4.2.1	Beckhoff CX9020 ja EL6695.....	26
4.2.2	Cognex konenäköanturin asennus	29
4.2.3	Robotin ja PLC:n yhdistäminen.....	30
4.3	Järjestelmän testaus.....	32
4.3.1	Konenäköanturin kalibrointi ja konfigurointi.....	34
4.3.2	PLC:n konfigurointi ja ohjelmointi.....	38
4.3.3	Robotin konfigurointi	42
4.4	Testaus ja tulokset.....	44
5	JOHTOPÄÄTÖKSET	46
	LÄHTEET.....	48

Lyhenteet ja merkinnät

DCS	Distributed Control System, hajautettu ohjausjärjestelmä
DOF	Degrees of freedom, vapausasteet
I/O	Input/Output, tulo- ja lähtötieto
PLC	Programmable Logic Controller, ohjelmoitava logiikka
PDO	Process Data Object, kenttäväylien tiedonsiirron perusyksikkö
KRL	KUKA Robot language, robottien ohjelmointikieli

1 JOHDANTO

Opinnäytetyön tilaajana on Kaakkois-Suomen ammattikorkeakoulu Oy (Xamk). Robotiikan ja tekoälyn koulutusohjelma alkoi Kouvolan kampuksella syksyllä 2021. Robotiikan laboratorion rakentaminen alkoi keväällä 2022 ja laboratorion kehittäminen on ollut jatkuva prosessi. Laboratoriolta tunnistettiin useampiakin kehittämisprojekteja, joissa olisi ainesta opinnäytetyöksi. Laboratoriolle oli hankittu Cognex-konenäköanturi sekä KUKA-käsivarsirobotti. Vastaavalla lehtorilla oli toiveena, että näistä voitaisiin koostaa robottijärjestelmä, jolla oppilaat voisivat tehdä erilaisia konenäköä hyödyntäviä robotiikkasovelluksia, kuten esimerkiksi esineiden poimintaa pöydältä konenäköanturin avulla. Konenäköanturi ja robotti eivät kuitenkaan olleet sellaisenaan yhteensopivia, joten järjestelmän rakentaminen vaatisi jatkoselvityksiä ja työtä.

Erilaiset konenäkösovellukset eivät ole varsinaisesti uusi aihe opinnäytetyöksi ammattikorkeakouluissa. Theseuksesta asiasanalla ”konenäkö” löytyy tätä kirjoittaessa yli 160 AMK- tai YAMK-opinnäytetyötä. Vanhimmat opinnäytetyöt Theseuksessa konenäköön liittyen ovat vuodelta 2013. Aihe on kuitenkin edelleen relevantti robotiikan ja automaation alalla. Tekniikan jatkuva kehittyminen mahdollistaa uudenlaisia sovelluksia ja käyttökohteita konenäköille (Nortio 2024). Konenäkö mainitaan myös International Federation of Roboticsin top-5 kehitystrendeissä osana yhteistyörobotiikan läpimurtoa (IFR 2024). Tästä syystä on erittäin tärkeää, että Xamkin robotiikan ja tekoälyn koulutusohjelmalla on käytössä laitteistoa, jolla oppilaat voivat tutkia aihetta.

Tarve tälle opinnäytetyölle nousee siitä, että robotiikan laboratoriolle hankittujen laitteiden yhdistämiselle ei ole olemassa valmista ohjeistusta ja aiheesta ei löytynyt aiempaa tutkimusta. Cognex 2000-sarjan konenäköanturin ja KUKA KR6 R900-2 käsivarsirobotin KR C5-ohjaimen välisen kommunikaation järjestäminen vaatii selvitystyötä useiden eri laitteiden ja teknologioiden osalta sekä mahdollisesti lisähankintoja.

Produktiivisen opinnäytteen kehittämistehtävänä on tehdä valmis järjestelmä, jolla opiskelijat voivat harjoitella erilaisten konenäköä hyödyntävien robotiikkasovellusten tekemistä. Lisäksi olisi eduksi, jos järjestelmä hyödyntäisi ulkoista

logiikkaa. Koulutusohjelman vastaavan lehtorin (tilaajan) toivomuksesta opin-
näytteen raportoinnin rajaamista tarkastellaan siitä näkökulmasta, että raportti
ei sisältäisi liian yksityiskohtaista toimintaohjetta laitteiston konfigurointiin, jotta
järjestelmää käyttäville opiskelijoille ei olisi liian suoraa ohjeistusta harjoitusten
tekemiseen.

Ojasalo ym. (2015, 65) mukaan konstrukttiivinen tutkimus on sopiva lähesty-
mistapa, kun kehittämistehtävänä on luoda konkreettinen tuotos. Pelkkä rat-
kaisun löytäminen tiedonsiirto-ongelmaan ei sinänsä siis vielä täytä hyvän
opinnäytetyön vaatimuksia. Tutkimuksen työstä tekee se, että pyrin perustele-
maan ratkaisuja olemassa olevalla teorialiedolla.

2 YRITYSESITTELY

Kaakkois-Suomen ammattikorkeakoulu Oy (Xamk) on korkeakoulu, joka tar-
joaa monipuolisia koulutusohjelmia ja tutkimusmahdollisuuksia. Xamk aloitti
toimintansa vuonna 2017, kun Kymenlaakson Ammattikorkeakoulu ja Mikkelin
Ammattikorkeakoulu yhdistyivät. Xamkilla on kampukset Kouvolassa, Kot-
kassa, Mikkelissä ja Savonlinnassa. Ammattikorkeakoulu tarjoaa koulutusta
useilla eri aloilla, kuten tekniikassa, liiketaloudessa, terveydenhuollossa ja so-
siaalialalla. Xamk työllistää noin 1 000 henkilöä, joista hieman yli kolmasosa
on opetushenkilöstöä ja loput TKI- tai tukihenkilöstöä.

Xamk panostaa tutkimukseen, kehittämiseen ja innovaatioihin (TKI-toiminta).
TKI-toiminta on laajaa ja kansainvälistä, ja se kattaa useita eri aloja, kuten di-
gitalisaation, ympäristötekniikan ja hyvinvoinnin kehittämisen. Xamk tekee
tiivistä yhteistyötä yritysten, julkisen sektorin ja muiden oppilaitosten kanssa.

Xamkin arvot perustuvat vastuullisuuteen, innovatiivisuuteen ja yhteisöllisyy-
teen. Ammattikorkeakoulun tavoitteena on vahvistaa Kaakkois-Suomen elin-
voimaa ja kansainvälistymistä, tarjota laadukasta koulutusta ja tukea opiskeli-
joiden ammatillista kasvua sekä edistää kestävästä kehitystä yhteiskunnassa.

3 JÄRJESTELMÄN SUUNNITTELU

Opinnäytteen kolmannessa luvussa käydään läpi työn tarkempi määritelmä, keskeisiä teorioita ja teknologioita sekä karkea suunnitelma järjestelmän toteuttamiseksi.

3.1 Työn määritelmä

Opinnäytetyön päätavoitteena on rakentaa opetuskäyttöön tarkoitettu robottijärjestelmä, jota on laajennettu ulkoisella logiikalla ja konenäköanturilla. Tällaisella robottijärjestelmällä pystytään opettamaan monia robotiikan kannalta oleellisia taitoja ammattikorkeakoulun opiskelijoille, kuten ohjelmointia, konenäön hyödyntämistä robotiikassa ja robotin integrointia muihin tehdasautomaatiolaitteisiin. Lisäksi tavoitteiden mukainen robottijärjestelmä tarjoaa hyvän pohjan täydentävien opintojen projektiopintoihin, joissa oppilaiden pitää tehdä itsenäisiä harjoitusprojekteja automaatioon, mekaniikkaan ja robotiikkaan liittyen. Oppilaat voivat esimerkiksi suunnitella työkaluja robotille, tehdä erilaisia konenäköä hyödyntäviä robottisovelluksia tai harjoitella robotin ohjaamista ulkoisella logiikalla.

Suunnitelmana oli ensimmäisenä tutkia käytössä olevien laitteiden ominaisuuksia ja teknologioita, sekä selvittää minkälainen ohjelmoitava logiikka järjestelmään kannattaisi lisätä ja tarvitaanko mahdollisesti muita hankintoja. Seuraavina vaiheina olisi laitteiden asennukset sekä konfigurointi siten, että niiden välinen kommunikaatio toimii luotettavasti. Viimeisenä vaiheena on järjestelmän testaus.

Työ on teknisesti vaativa, koska siinä pitää tutustua ja hallita ainakin seuraavia tietoja ja taitoja:

- Cognex-konenäkökameran liittäminen järjestelmään, konfigurointi, kalibrointi ja ohjelmointi
- Ohjelmoitavan logiikan asentaminen, konfigurointi ja ohjelmointi
- Robotin ohjaimen liittäminen ohjelmoitavaan logiikkaan ja kommunikaation konfigurointi sekä robotin ohjelmointi

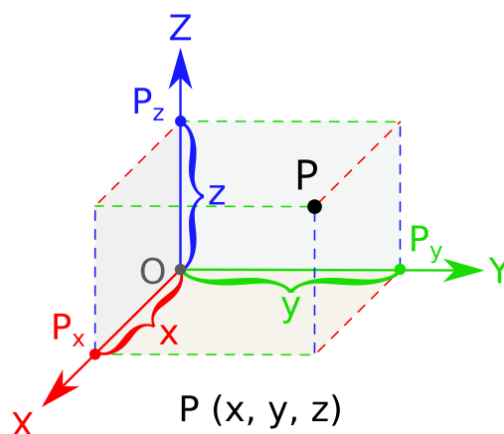
Ratkaisun toimivuus havainnollistetaan testaussovelluksella, jossa robotti osaa käydä viereiseltä pöydältä hakemassa kuution muotoisen kappaleen konenäköanturilta saatujen paikkatietojen perusteella.

3.2 Robotiikka

Tässä opinnäytetyössä ei syvennyttä varsinaisesti teollisuusrobotiikan tutkimiseen, vaan pääaiheena on toimivan konenäköä hyödyntävän robottijärjestelmän rakentaminen. Esittelen kuitenkin tässä luvussa muutamia työn kannalta oleellisia asioita teollisuusrobotiikkaan liittyen.

3.2.1 Koordinaatistot

Kolmiulotteisessa avaruudessa mikä tahansa piste voidaan esittää kolmella karteesisella koordinaatilla, jotka ovat etäisyydet kolmen keskenään kohtisuoran tason leikkauspisteestä. KUKA käyttää Z-akselia kuvaamaan ylös-alas-liikettä, y-akselia vasen-oikea-liikettä ja x-akselia eteen-taakse-liikettä (kuva 1). Eri robottivalmistajat saattavat käyttää x- ja y-akseleita kuitenkin eri tavalla. Robotiikassa pelkkä pisteen paikka ei kuitenkaan usein riitä, vaan tarvitaan tieto, missä asennossa pistettä halutaan robotilla lähestyä, tai missä asennossa pisteen kohdalla oleva kappale on. Tarvitaan siis lisäksi tieto pisteen tai kappaleen kierroista x-, y- ja z-akseleiden ympäri. (Billing 2024, luku 5.2.1.)



Kuva 1. Kolmiulotteinen avaruus (Pertsev s.a)

Robotissa käsivarren nivelet muodostavat kinemaattisen ketjun. Nivelien asentoja toisiinsa ja referenssikoordinaatistoon nähden voidaan kuvata Eulerin kulmilla. Ne kuvaavat 3D-kiertoa, joka tapahtuu vaiheittain jokaisen akselin

ympäri. Esimerkiksi 20 asteen kierto x-akselin ympäri merkitään $(20, 0, 0)$. Eulerin kulmat helpottavat robotin liikkeen hahmottamista ja ohjelmointia. (Billing 2024, luku 5.2.3.)

Aiempi esimerkki ei tosin aivan päde KUKAn roboteilla, joissa käytetään Eulerin kulmia järjestyksessä z, y, x, kun esimerkissä kierrot ovat järjestyksessä x, y, z. Rotaatioiden suhteen tällä on hyvinkin paljon merkitystä, koska rotaatiot toteutetaan tosiaan järjestyksessä. Esimerkiksi rotaatiot tehtynä järjestyksessä $(90, 0, 90)$ päätyvät täysin eri asentoon riippuen käytettävästä järjestyksestä.

Yleensä robotiikassa hyödynnetään useita koordinaatistoja, kuten robotin maailman koordinaatistoa tai peruskoordinaatistoa, itse kalibroitavia työkoordinaatistoja sekä työkalupistekoordinaatistoa. Robotti käyttää koordinaatistoja liikkeenohjaukseen. Työkalun paikka ja asento määritetään suhteessa robotin peruskoordinaatistoon, ja ohjelmointia helpottavat työkoordinaatistot. Myös työkalut ja ohjelmoidut paikkapisteet kuvataan koordinaatistoina. Koordinaatistot ovat dataa, jolla on paikka ja asento suhteessa referenssikoordinaatistoon. Työkalu- ja työkoordinaatistot voidaan määrittää numeerisesti tai opettaa robotin avulla. (Billing 2024, luku 5.2.4.)

Robotiikan tarkkuus on tärkeä ominaisuus, ja se jaotellaan kahteen päätyyppiin: toistotarkkuuteen ja paikoitustarkkuuteen. Toistotarkkuus kertoo, kuinka tarkasti robotti pystyy palaamaan samaan pisteeseen toistuvasti. Toisin sanoen, jos robotti ohjelmoidaan liikkumaan tiettyyn pisteeseen useita kertoja, toistotarkkuus kertoo, kuinka paljon nämä pisteet vaihtelevat toisistaan. Hyvä toistotarkkuus on tärkeää esimerkiksi kokoonpanotehtävissä, joissa robotin on toistettava sama liike useita kertoja. Paikoitustarkkuus puolestaan kertoo, kuinka tarkasti robotti pystyy liikkumaan tiettyyn pisteeseen absoluuttisessa koordinaatistossa. Toisin sanoen, jos robotille annetaan koordinaatit tietylle pisteelle, paikoitustarkkuus kertoo, kuinka lähelle tätä pistettä robotti todellisuudessa pääsee. Hyvä paikoitustarkkuus on tärkeää esimerkiksi työstökoneiden yhteydessä, joissa robotin on liikuttava tarkasti työkappaleen suhteen. Robotti laskee työkalun paikan teoreettisesti, mutta kuormat voivat aiheuttaa virheitä. Toistotarkkuus on yleensä parempi kuin paikoitustarkkuus, koska se ei ole riippuvainen absoluuttisista koordinaateista. Paikoitustarkkuutta voidaan

parantaa kalibroimalla robotti, jolloin mittavirheet voidaan minimoida. (Billing 2024, luku 5.2.8.)

3.2.2 Teollisuusrobotit

Ensimmäisenä teollisuusrobottina pidetään Unimate 1900 -robottia, joka otettiin ensimmäistä kertaa käyttöön General Motorsin tehtaalla vuonna 1961. 70-luvun alussa ruotsalainen ASEA valmisti ensimmäisen täysin elektronisesti ohjattavan nivelvarsirobotin, IRB-6:n. 70-luvun lopulla KUKA ja Unimation toivat markkinoille ensimmäiset kuuden akselin ja kuuden vapausasteen käsivarsirobotit. Tämä robottityyppi on säilynyt tähän päivään asti yleisimpänä teollisuusrobottina. Joissain sovelluksissa kuitenkin nopeus ja tarkkuus ovat tärkeämpiä ominaisuuksia kuin vapausasteiden määrä. Nopeutta vaativiin poiminta- ja asennustöihin kehitettiin 80-luvulla SCARA- ja deltarobotit. (Hägele ym. 2016, 1386–1388.)

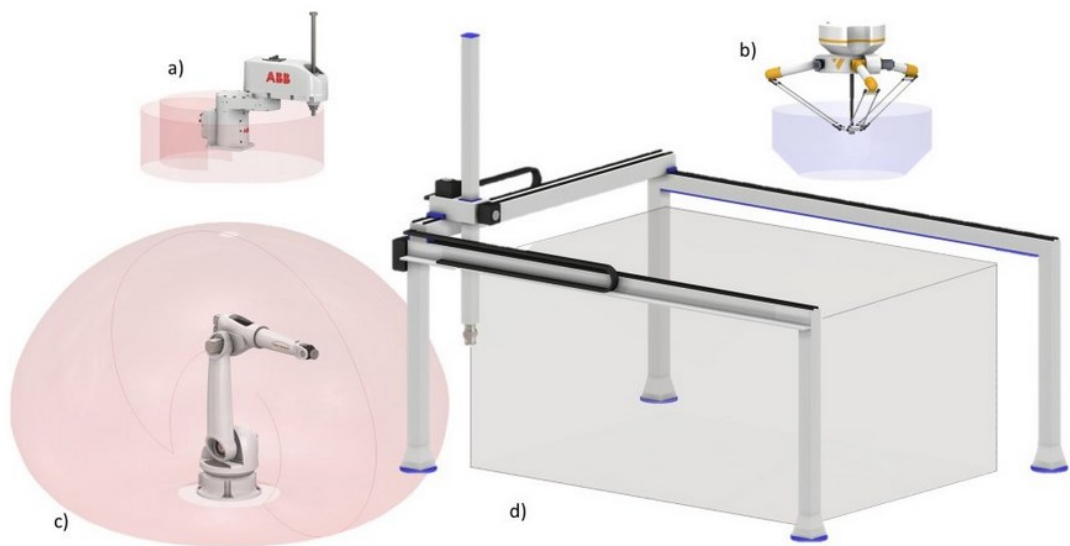
Nykyään termi teollisuusrobotti on määritelty standardissa ISO 8373:2021 kohdassa 3.6 tarkoittamaan teollisuusautomaation sovelluksissa käytettävää automaattisesti toimivaa, vähintään kolmella akselilla ohjelmoitavaa monikäyttöistä manipulaattoria, joka voidaan asentaa kiinteästi tai liikkuvalla alustalle. Standardin mukaan teollisuusrobottiin kuuluu manipulaattori, liikettä tuottavat toimilaitteet ja niitä ajava robotin ohjain, sekä laitteet ja ohjelmistot, jolla robottia voidaan opettaa tai ohjelmoida.

Yleisin teollisuusrobotti on kuuden vapausasteen käsivarsirobotti. Teollisuuden robotiikka -kirjan mukaan lähes 75 % teollisuusroboteista on sellaisia (Lempiäinen 2024, luku 1). Kuusi vapausastetta (DOF, degrees of freedom) tarkoittaa robotin kykyä liikkua yhteensä kuudessa erilaisessa suunnassa ja kiertosuunnassa. Tämä sisältää kolme lineaarista siirtymäliikettä (ylös/alas, vasemmalle/oikealle, eteen/taakse) ja kolme kiertoliikettä (pyöriminen näiden akselien ympäri). Tällainen robotti pystyy oman ulottuvuutensa puitteissa lähestymään haluttua pistettä lähes mistä suunnasta tahansa.

Muita yleisiä teollisuusrobotteja ovat SCARA-robotit, portaalirobotit sekä deltarobotit (kuva 2). SCARA-roboteissa (Selective Compliance Articulated Robot Arm) on yleensä neljä niveltä ja kolme vapausastetta (X, Y, Z). Rakenteensa

ansiosta ne ovat jäykkiä Z-akselin suuntaisesti ja notkeita XY-akseleilla. Yksinkertaisemman kinematiikan ansiosta ne ovat nopeampia ja tarkempia, kuin useamman vapausasteen robotit. SCARA-robotteja käytetään useimmin koonpanotehtävissä.

Portaalirobotit eli karteesiset robotit ovat kolmen vapausasteen robotteja, jotka liikkuvat lineaarisesti kolmella akselilla (X, Y, Z). Esimerkiksi suurin osa 3D-tuotimista ovat portaalirobotteja.



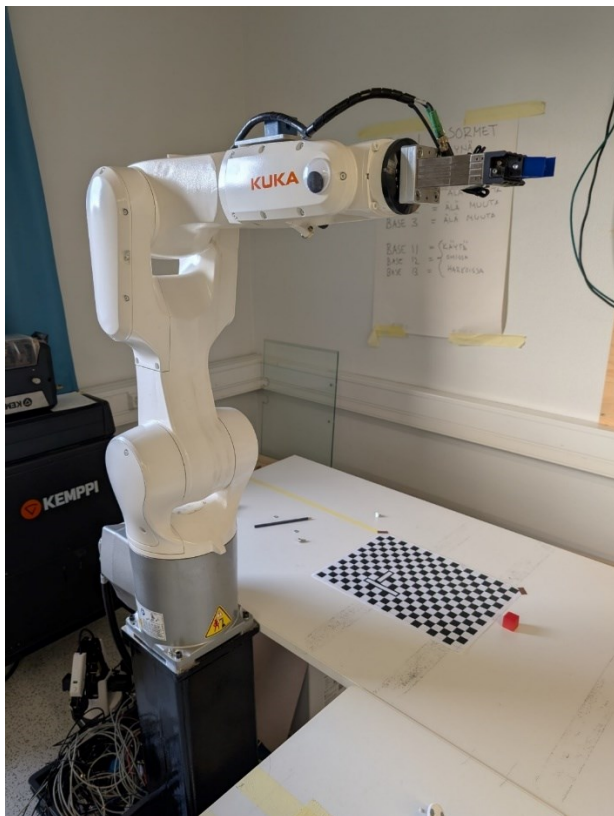
Kuva 2. a) SCARA-, b) Delta-, c) nivelvarsi- ja d) portaalirobotti (Lempiäinen 2024, luku 1.2).

Teollisuuden tarpeisiin on kehitetty myös nivelvarsirobotteja, jotka on tarkoitettu ihmisen ja robotin turvalliseen yhteistyöhön. Laitteita kutsutaan yhteistyöroboteiksi. Yhteistyörobotit ovat yleensä melko pieniä nivelvarsirobotteja, joiden turvaominaisuudet on kehitetty siten, että ihminen voi olla turvallisesti samassa työtilassa robotin kanssa. (Lempiäinen 2024, luku 1.6.)

Teollisuusrobottien ja -robottijärjestelmien turvallisuusvaatimuksia on määritetty standardissa SFS-EN ISO 10218-1.

3.2.3 KUKA KR 6 R900-2

Kuten teollisuusrobotit yleisesti, Xamkin robotiikan laboratorion robotti koostuu manipulaattorista, robotin ohjaimesta ja käsiohjaimesta. Robotin toimitukseen kuului myös Schunk EGP 50-N-N-B-sormitarttuja sekä turvalaitteita ja I/O-laajennus.



Kuva 3. KUKA KR 6 R900-2 robotiikan laboratorion laboratorion.

Robotilla on kuusi niveltä ja kuusi vapausastetta. Robotin ulottuvuus on 901 mm ja maksimikuorma 6,7 kg. Toistotarkkuudeksi luvataan +/- 0,02 mm (KUKA Deutschland GmbH 2024).

Robotti toimitettiin KR C5 micro -ohjaimella, joka on kompakti versio valmistajan tuoreimmasta ohjainmallista. Käyttöjärjestelmänä toimii KSS 8.7. Ohjain sisältää virtalähteen, servo-ohjaimet kuudelle akselille, sulautetun järjestelmän turvaohjaimella, Ethernet- ja USB-liitäntöjä, 16 kappaletta 24 V digitaalisia tuloja ja lähtöjä sekä tarvittavat liittimet robotille ja käsiohjaimelle. Ohjain tukee suoraan EtherCAT-väylää ja maksullisella lisäosalla Profinet- ja EtherNet/IP väyliä. (Knöpfle 2024).

3.3 Konenäkö

Opinnäytetyössä kytketään konenäköanturi osaksi robottijärjestelmää. Konenäön syvällinen tutkiminen ei kuitenkaan ole tämän opinnäytetyön ytimessä, joten käsittelen aihetta vain lyhyesti ja esittelen opinnäytetyössä käytettävän konenäköanturin.

Aiheesta puhuttaessa saattaa törmätä termeihin konenäkö (Machine Vision) ja tietokonenäkö (Computer Vision). Tietokonenäköllä tarkoitetaan yleisemmin näkökyvyn tieteellistä tutkimusta ja ohjelmistojen suunnittelua, kun taas konenäköllä tarkoitetaan sekä ohjelmistojen että laitteistojen ja kuvankaappaus-tekniikoiden tutkimusta todellisissa sovelluksissa. Konenäkö on siis enemmän insinööritieteisiin ja teollisuuteen painottunut aihe. Nykyään tietotekniikka on kehittynyt niin pitkälle, että suuri osa sovelluksista voidaan toteuttaa tavallisilla tietokoneilla. Sovellukset voivat toimia myös sulautetuissa järjestelmissä tai ohjelmoitavilla logiikoilla. Tämä on johtanut termien lähentymiseen, ja niitä käytetään usein lähes synonyymeinä. (Davies 2018, luku 1.4.)

Tähän työhön termi konenäkö sopii erinomaisesti, koska käsittelemme teollista robottijärjestelmää, johon integroidaan konenäköanturi. Termin käyttöä tukee myös se, että laitteen valmistaja Cognex, luokittelee laboratorion laitteen konenäköanturiksi (Machine vision sensor) (Cognex 2024b).

Aihetta käsiteltäessä huomaa myös nopeasti, että alalla käytetään termejä konenäkökamera ja konenäköanturi tarkoittamaan monesti samaa asiaa. Oletan, että teollisuuslaitteiden tuottajat, kuten Cognex, haluavat termillä ”konenäköanturi”, kertoa laitteen olevan paljon enemmän, kuin pelkkä tavallinen kamera. Anturilla voidaan tunnistaa, havaita ja analysoida, kun taas kameralla otetaan vain kuvia. Cognex myy myös teollisuuskameroita (Cognex 2024a). Niiden suurin ero heidän konenäköantureihinsa on se, että ne ovat tosiaan vain kameroita, joista kuvat pitää siirtää tietokoneelle prosessoitavaksi ja analysoitavaksi, kun taas konenäköantureissa nämä ominaisuudet ovat sulautettu yhteen laitteeseen. Tässä työssä käytän termiä konenäköanturi, koska se on laboratoriolle olevan laitteen virallinen nimi. Kamera-termiä käytettäessä viittaa spesifimmin konenäköanturin kameraan.

Konenäköä käytetään teollisuudessa yleisimmin tuotteiden tai kappaleiden tunnistamiseen, oikeellisuuden todentamiseen, mittaamiseen, määrän laskeamiseen sekä vikojen ja laatuvirheiden havaitsemiseen. Robottijärjestelmään integroituna konenäköä käytetään yleensä kohteiden luokitteluun, kohteen paikan ja asennon määrittämiseen ja tavoitemittaamiseen, eli muutetaan robotin liikeratoja konenäön havaintojen mukaan. (Siltala & Gautam 2024, luku 8.1.)

Konenäköanturin integrointi laboratorion robottijärjestelmään mahdollistaa esimerkiksi kappaleiden poiminnan konenäköanturilta saadun informaation perusteella.

3.3.1 Cognex-konenäköanturi

Laboratoriolle ostettu laite on tyyppimerkinnältään 2001M-353. Laite sisältää optisen moduulin, päämoduulin ja I/O-moduulin. Konenäköanturi on siis tarkoitettu olemaan itsenäinen laite, jolle määritellään tehtävä tai työ (job) Cognexin In-Sight EasyBuilder -ohjelmistolla. Ohjelma ladataan laitteelle, jonka jälkeen se toimii itsenäisesti ja voidaan määritellä antamaan tietoja otetuista kuvista joko laitteen tukemilla kommunikaatioväylillä tai 24 voltin lähdoilla. Käyttövalmis laite ei tarvitse olla kytkettynä tietokoneeseen, koska kuvien prosessointi tehdään laitteen omalla keskusyksiköllä.



Kuva 4. Cognex 2001M-353 konenäköanturi.

Optiseen moduuliin sisältyy LED-valo ja objektiivi, jota Cognex kutsuu termillä "liquid lens". Termillä tarkoitetaan lyhyesti sitä, että laite pystyy tarkentamaan kuvan automaattisesti, kun vastaavasti tavallisella optiikalla kuva pitää tarkentaa tiettyyn pisteeseen, jolloin etäisyyden suunnassa tapahtuvat muutokset johtavat kuvan epätarkkuuteen (Cognex 2024c).

Päämoduuliin kuuluu kuvakenno, käyttöpaneeli ja keskusyksikkö. Kuvakennon termi englanniksi on sensor, mutta kameratekniikassa käytetään yleisesti suomessa termiä kuvakenno, vaikka suora käänös olisi tässäkin tapauksessa anturi. Käyttöpaneelissa on indikaattorivaloja, sekä näppäin kuvan ottamiselle. Keskusyksiköstä Cognex ei anna tarkempia tietoja.

I/O-, eli liitäntämoduuliin kuuluu liittimet teolliselle Ethernet-kaapelille, sekä 12 kanavainen liitäntä johtosarjalle, jolla saadaan esimerkiksi virta kameralle, sarjaviestintä sekä yksi 24 V tulotieto ja neljä lähtöä (Cognex 2024d).

Laitteessa on 32MB haihtumatonta muistia ohjelmille ja kuville sekä 128MB SDRAM muistia kuvien prosessointia varten. Kuvien resoluutio on pienimmillään 640 x 480 pikseliä ja suurimmillaan 1280 x 960 pikseliä (Cognex 2024d). Kuvat ovat 8 bittisiä harmaasävykuvia, joka tarkoittaa sitä, että jokainen pikseli voidaan esittää 256:lla eri harmaan sävyllä.

Laite tukee seuraavia kommunikaatiotapoja; EtherNet/IP, Profinet, SLMP, Modbus TCP, TCP/IP, UDP, FTP ja Telnet (Cognex 2024d).

3.4 Robottijärjestelmän kommunikaatio

Koska työn suurin ongelma on saada konenäköanturi siirtämään tietoa robottijärjestelmään ja mahdollisesti saamaan tietoa tai käskyjä takaisin, on laitteiden välinen kommunikaatio tärkein yksittäinen aihe tässä opinnäytetyössä. Yleisesti ottaen automaatioissa yksittäisten laitteiden hyödyt saadaan parhaiten esiin vasta, kun laitteet saadaan kommunikoimaan keskenään. Mitä hyötyä on konenäköanturista, joka osaa ottaa kuvia ja analysoida niitä, mutta ei pysty kertomaan havainnostaan mihinkään?

Perinteisellä I/O:lla tarkoitetaan digitaalisia input (tulo) ja output (lähtö) -signaaleita. Digitaalinen I/O on fyysisesti johdinpari (+ ja -), jossa joko on jännite tai ei ole jännitettä. Yleisimmin käytetty ohjausjännite on 24 V tasavirtaa. Digitaalisessa maailmassa tämä vastaa yhtä bittiä, joka on arvoltaan 0 tai 1. Digitaaliset signaalit voivat toimia kahdella eri tavalla: ne voivat joko avata tai sulkea sähköisen piirin. Avautuva signaali (NC) tarkoittaa, että sähköpiiri on normaalisti suljettu, ja signaalin aktivoituessa piiri avautuu ja sähkövirta katkeaa. Sulkeutuva signaali (NO) puolestaan tarkoittaa, että sähköpiiri on normaalisti auki, ja signaalin aktivoituessa piiri sulkeutuu ja sähkövirta alkaa kulkea. Avautuvia signaaleja käytetään usein turvallisuuden kannalta kriittisissä sovelluksissa, kuten turvalaitekytkennöissä. Tämä johtuu siitä, että jos johto katkeaa tai signaalissa tapahtuu muu häiriö, piiri tulkitaan automaattisesti aktiiviseksi ja turvatoiminto käynnistyy. Näin järjestelmä pysyy turvallisena myös vikatilanteissa. (Liuha & Aro 2024.)

Perinteinen digitaalinen I/O soveltuu siis parhaiten välittämään tietoa asioista, jotka joko ovat tosia tai epätosia. Yleisimmin siis antureita, jotka havaitsevat onko kappale lähellä vai ei, tai onko painiketta painettu. Tai robotti voi kertoa toiselle robotille olevansa valmis toimimaan tai suorittamassa ohjelmaa.

Useita mitattavia määreitä, kuten esimerkiksi etäisyyksiä, on hankalampaa välittää digitaalisesti. Tällöin pitää ohjelmallisesti niputtaa bittejä binäärijärjestelmäksi tai käyttää analogista signaalia. Binäärijärjestelmä tarkoittaa kantalukujärjestelmää, jossa kantalukuja on kaksi, eli nolla ja yksi. Kahdella bitillä voidaan esittää yhteensä neljä tilaa; 00, 01, 10 ja 11. Tilat voidaan vastaavasti esittää myös desimaalilukuina. Esimerkiksi kahdeksalla bitillä, eli tavulla, voidaan esittää desimaalilukuja välillä 0...255.

$$11111111_2 = 2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = 255$$

16 bittisellä tiedolla voidaan esittää desimaalilukuja välillä 0...65535. Toisaan on tarve esittää myös negatiivisia desimaalilukuja, jolloin yleisimmin käytetään kahden komplementtia. Tällöin esimerkiksi 8 bittinen tieto voidaan esittää desimaaleina välillä -128...127. Positiiviset luvut 127 asti esitetään normaalisti binääreinä. Negatiiviset luvut esitetään kuten positiivinen luku, mutta kaikki bitit käännetään ja lopuksi lisätään yksi.

$$0000\ 0011_2 = 3$$

Kahden komplementtina

$$1111\ 1101_2 = -3$$

Analogista signaalia välitetään vaihtuvalla jännitteellä tai virralla. Teollisuudessa jänniteväli on sovittu olevan joko $-10\dots+10$ voltia tai $0\dots+10$ voltia. Virtaviestille on sovittu välit $0\dots20$ mA tai $4\dots20$ mA. Analogisessa viestissä signaali voi periaatteessa saada minkä tahansa arvon minimi- ja maksimiarvonsa välillä. Teoriassa siis analogisen viestin tarkkuus on rajaton, mutta käytännössä mittaustarkkuus rajoittaa sitä. Analogisessa viestissä siirtyy paljon enemmän tietoa kuin digitaalisessa, mutta signaali on muutettava digitaaliseen muotoon jatkokäsittelyä varten. Tässä muunnoksessa menetetään aina jonkin verran tietoa. (Liuha & Aro 2024.)

Perinteisen I/O:n suurin ongelma on se, että jokainen yksittäinen digitaalinen tai analoginen viesti tarvitsee yhden johtoparin. Joissakin tapauksissa virtapiirin saaminen toimivaksi onnistuu yhteisellä paluujohtimella, joka vähentää tarvittavien johtimien määrää huomattavasti (Liuha & Aro 2024, luku 7.1.1). Tällaisessa tapauksessa voisi esimerkiksi kaapeli, jossa on 18 johdinta, kuljettaa signaaleille virtapiirin tarvittavat yhteiset plus- ja miinus johtimet, sekä 16 signaalia.

Automaatiolaitteiden monimutkaistuessa tarvittavien johtimien ja liittimien määrä kasvoi niin suureksi, että niiden hallinta muodostui haastavaksi. Ensin kehitettiin sarjaliikenne, jossa lähetetään tietoa bitteinä peräkkäin, ikään kuin jonossa, yhdellä johtoparilla yhdestä laitteesta toiseen. Seuraavassa kehitysvaiheessa kehitettiin järjestelmiä ja protokollia, joiden avulla samaan verkkoon voitiin kytkeä useita laitteita. Viesteihin sisältyi varsinaisen viestin lisäksi tietoa lähettäjästä, vastaanottajasta ja virheiden tarkistuksesta. Näitä protokollia kutsutaan teollisuudessa myös kenttäväyliksi. Kenttäväylien yleistyminen vähensi tarvittavaa kaapelointia ja mahdollistaa suuremman tietomäärän liikuttamisen. (Liuha & Aro 2024, luvut 7.1.1–7.1.2.)

Nykyisin teollisuudessa on yleistymässä Ethernet-protokollaan perustuvat kenttäväylät, jotka tuovat jälleen uusia etuja perinteisiin kenttäväyliin verrattuna. Teollisen Ethernetin protokollat ovat nopeampia, pystyvät siirtämään

enemmän tietoa ja mahdollistavat huomattavasti suurempia laitekokonaisuuksia. Lisäksi teollisen Ethernetin protokollat ovat luonteeltaan deterministisiä, eli aikakriittisiä. Deterministisyys tarkoittaa, että käskyn toteutumiseen kuluva aika on ennalta tiedossa ja vakio. Käsky joko toteutuu määritellyssä ajassa tai siitä saadaan tieto, jos se ei toteudu. (Liuha & Aro 2024, luku 7.1.2)

Robottiikan laboratoriolle olevat laitteet tukevat useampaa teollisen Ethernetin kenttäväylää. Tässä työssä keskitytään niistä kahden tutkimiseen ja käyttöön tarkemmin. Laboratorion KUKA-robotit tukevat lisämaksutta EtherCAT-väylää ja konenäköanturin kommunikaatioon on alustavasti suunniteltu OPC UA -standardia.

3.4.1 EtherCAT

Koska laboratoriolle toimitettu robottijärjestelmä käyttää valmiiksi EtherCAT-kenttäväylää, on varmaa, että teknologiaa tullaan käyttämään työssä.

EtherCAT julkistettiin 2003. Protokollan kehityksen päätavoitteina olivat lyhyet sykliajat, vähäinen jitter ja matalat kustannukset. EtherCAT on alun perin Beckhoff Automation -yrityksen kehittämä ja nykyään sitä hallinnoi EtherCAT Technology Group -organisaatio. EtherCAT on kansainvälisesti standardoitu avoin teknologia, eli kuka tahansa voi käyttää sitä. (ETG 2024).

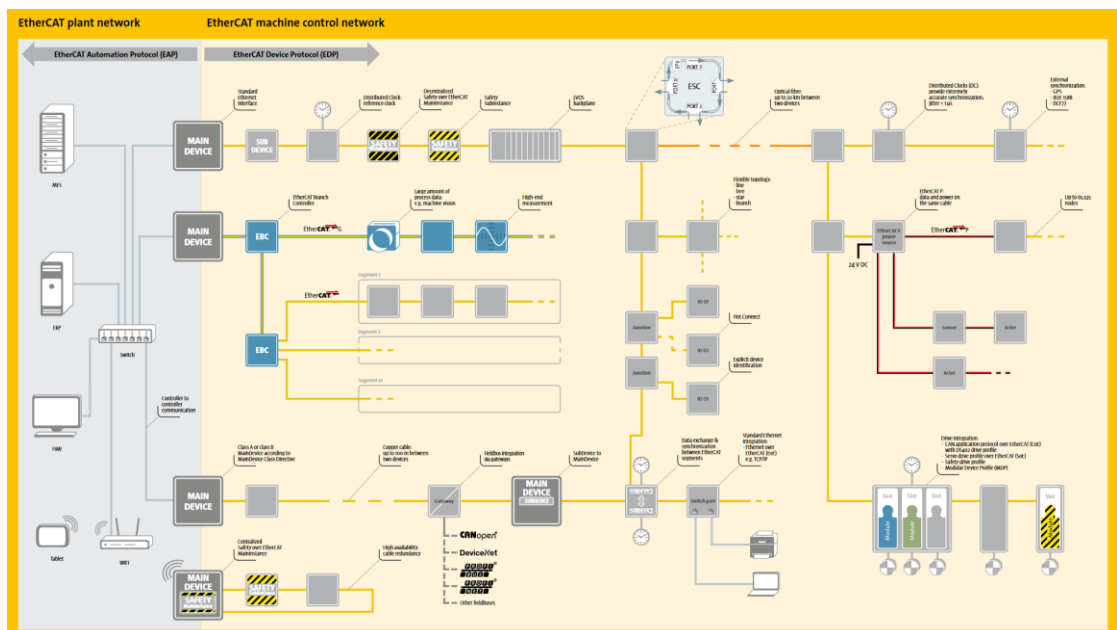
EtherCAT on teollisen Ethernetin protokolla, joka hyödyntää standardia Ethernet-teknikkaa (IEEE 802.3), mutta on samalla suunniteltu vastaamaan automaatioteollisuuden erityistarpeisiin. Automaatiossa kriittisiä ominaisuuksia ovat reaaliaikaisuus, deterministinen vasteaika ja suuri määrä solmuja (verkossa oleva laite), samoin kuin laitteistokustannusten minimointi.

Tavallinen Ethernet ei sovellu hyvin kenttätasolle, koska se on tehoton ja hidastaa prosessitiedon siirtämiseen. Jokainen viesti tavallisessa Ethernetissa on 84 tavua pitkä, josta vain 46 tavua on käytettävissä prosessitiedolle. Jos esimerkiksi taajuusmuuttaja lähettää ja vastaanottaa 4 tavua tietoa, tehokas tiedonsiirtonopeus on vain 4,8 %. Lisäksi taajuusmuuttajan reaktioaika ja protokollat, kuten IP ja TCP, hidastavat tiedonsiirtoa entisestään. (ETG 2024.)

Tiedonsiirto EtherCAT-protokollassa tapahtuu seuraavasti: EtherCAT-päälaitte lähettää datagrammin tai kehyksen, joka kulkee jokaisen solmun, eli laitteen,

läpi. Jokainen kenttälaite lukee sille osoitetun tiedon ja lisää samalla oman vastauksensa kehykseen sen kulkiessa eteenpäin. Viimeinen solmu havaitsee avoimen portin ja lähettää kehyksen takaisin päälaitteelle hyödyntäen Ethernetin full-duplex-ominaisuutta. Tällä menetelmällä saavutetaan erittäin tehokas tiedonsiirto. Datagrammien hyötysuhde on yli 90 %, ja täyden dupleksin, eli kaksisuuntaisuuden, ansiosta teoreettinen tiedonsiirtonopeus ylittää jopa 100 Mb/s. (ETG 2024.)

Verkon determinismi ja reaaliaikaisuus varmistetaan sillä, että vain päälaitte lähettää aktiivisesti EtherCAT-kehyksiä. Muut solmut ainoastaan lukevat ja kirjoittavat dataa kehykseen sen kulkiessa ohi. EtherCAT-laitteet käyttävät standardia Ethernet MAC-ohjainta, joten päälaitte voidaan toteuttaa millä tahansa laitteistoalustalla. Kenttälaitteet puolestaan hyödyntävät EtherCAT-alilaitteohjainta (ESC), joka käsittelee kehyksiä laitteistotasolla. Tämä takaa verkon ennustettavan suorituskyvyn riippumatta yksittäisten laitteiden toteutuksesta. (ETG 2024.)



Kuva 5. Esimerkki EtherCAT-järjestelmästä (ETG 2024).

3.4.2 OPC UA

OPC Unified Architecture (UA) on vuonna 2008 julkaistu tiedonsiirto-standardi, jota käytetään teollisuusautomaatiossa. Se on alustariippumaton ja toimii eri-

laisissa ympäristöissä sulautetuista mikrokontrollereista pilvipohjaisiin sovel-
luksiin. OPC UA sisältää myös turvallisuustoimintoja, kuten salauksen, toden-
nuksen ja auditoinnin. (OPC Foundation 2024a.)

OPC UA on standardoitu tiedonsiirtoprotokolla (IEC 62541). Sen avulla erilai-
set laitteet ja ohjelmistot voivat kommunikoida keskenään valmistajasta riippu-
matta. OPC UA:n kehityksestä vastaa OPC Foundation. Standardia hyödyn-
netään laajasti eri teollisuudenaloilla, ja sen käyttö on jatkuvassa kasvussa.
OPC UA:n avulla voidaan esimerkiksi yhdistää automaatiovalvomot (SCADA)
automaatio- ja ohjausjärjestelmiin (PLC/DCS). Tämä mahdollistaa tiedon ke-
räämisen ja laitteiden ohjaamisen keskitetysti. Myös valmistuksenohjausjärjes-
telmät (MES) ja toiminnanohjausjärjestelmät (ERP) hyödyntävät OPC UA:ta
yhä enemmän, mikä tehostaa tiedonkulkua ja parantaa tuotannon tehok-
kuutta. (Liuha & Aro 2024, luku 7.1.3.)

OPC UA kommunikaatio on pohjimmiltaan server-client-pohjainen. Kahden
laitteen kommunikoinnissa siis toinen laite on palvelin (server), josta konfigu-
roidaan tietoa esille. Toinen laite on taas asiakas (client), joka oikein konfigu-
roituna pääsee näkemään palvelimen tarjoavat tiedot. OPC UA ei sinänsä ole
reaaliaikainen, eikä deterministinen, joka saattaa haitata sen käyttöä teolli-
sissa sovelluksissa. Erityisesti näitä ominaisuuksia kaivataan kenttälaitteiden
kanssa. OPC UA kuitenkin kehittyy jatkuvasti. Siihen on jo lisätty publisher-
subscriber -malli, joka vastaa lähestulkoon reaaliaikaista väyläkommunikaa-
tiota. (Liuha & Aro 2024, luku 7.1.3.)

Suunnitteilla on tehdä PubSubista reaaliaikainen ja deterministinen TSN:n
(Time Sensitive Networking) avulla. Nimenomaan kenttälaitteikäyttöön OPC UA
on kehittämässä PubSub -tekniikkaan perustuvan UAFX (Field eXchange) -
spesifikaation. (OPC Foundation 2024b.)

3.5 Ohjelmoitavat logiikat

Ohjelmoitava logiikka (PLC, programmable logic controller) sai alkunsa rele-
pohjaisista ohjausjärjestelmistä. Ennen PLC:iden yleistymistä automaattinen
ohjaus hoidettiin releistä, kytkimistä, kelloista ja laskureista koostuvilla piireillä.

Mekaanisilla ohjauksilla oli monia haittoja. Ne veivät paljon tilaa ja niiden toteutus ja muuttaminen oli työlästä. Releohjaus koostui usein sadoista releistä ja johdoista, ja muutokset vaativat koko järjestelmän uudelleenjohtotuksen. Releet olivat myös alttiita vioille ja niiden testaus oli hankalaa. (Hanssen 2015, luku 1.)

Useiden yritysten pyrkimykset ratkaista mekaanisten ohjausten ongelmat johtivat ohjelmoitavan logiikan (PLC) kehittämiseen. Ensimmäinen kaupallinen PLC, MODICON 0844, hyödynsi relekaavioihin perustuvaa ohjelmointikieltä. PLC:iden ominaisuudet ovat kehittyneet ajan myötä, ja nykyään ne tarjoavat laajan valikoiman toimintoja, kuten laskureita, ajastimia ja analogisia I/O-liitäntöjä. Mikroprosessorien kehitys 1970-luvulla vauhditti PLC:iden kehitystä, ja nykyään ne sisältävät monipuolisia ohjelmistoja ja toimintoja, kuten tiedonsiirtoa ja säätöalgoritmeja. (Hanssen 2015, luku 1.1.)

Ohjelmointikielien, kuten LD (Ladder diagram), IL (Instruction list) ja SFC (Sequential function chart) sisällytettiin kansainväliseen IEC 61131-3 -standardiin. Standardi määrittelee myös graafiset FBD- (Function block diagram) ja tekstipohjaisen ST-kielen (Structured text). Ennen standardin käyttöönottoa PLC-valmistajien välillä oli suuria eroja. Nykyään valmistajat noudattavat standardia laajemmin, mikä helpottaa PLC-järjestelmien yhteensovittamista. (Hanssen 2015, luku 1.1.)

PLC:n pääosat ovat keskusyksikkö (CPU), muisti, virtalähde, I/O-yksiköt ja kommunikaatiomodulit. Useimmissa PLC:issä on myös merkkivaloja ja näyttöjä tilan ja tietojen näyttämiseen. PLC:n osat on yhdistetty väylillä, jotka ovat johtimia tai kupariliuskoja. Tyypillisiä väyliä ovat data-, osoite-, ohjaus- ja järjestelmäväylä. Dataa siirretään data- ja osoiteväylillä, ohjausväylä synkronoi ja ohjaa liikennettä, ja järjestelmäväylä hoitaa I/O-kommunikaatiota. (Hanssen 2015, luku 1.2.)

Yksinkertaistetusti PLC:t toimivat, kuten tietokoneet. PLC:lle kirjoitetaan ohjelma, joka yleensä ohjaa lähtöjä sisääntulojen perusteella. Tulotietoina voi olla digitaalisia tai analogisia tietoja ja lähtöinä voidaan ohjata esimerkiksi sähkömoottorin ajosuuntaa ja nopeutta.

4 JÄRJESTELMÄN TOTEUTUS

4.1 Järjestelmän suunnittelu

Laboratorion robottijärjestelmä tukee perinteistä I/O:ta ja EtherCAT-kenttäväylää. Lisämaksusta saa käyttöön myös Profinet- ja EtherNet/IP-kenttäväylät. Konenäköanturi tukee OPC UA-, EtherNet/IP-, Profinet-, SLMP- ja Modbus TCP -kenttäväyliä. Konenäköanturia voidaan pyytää ottamaan kuva perinteisellä I/O:lla, mutta takaisin saadaan vaan tieto, että onko kuva hyväksytty vai ei ohjelmoitujen ehtojen perusteella. Paikkatietoja ei pystytä välittämään perinteisellä I/O:lla.

Näin ollen Cognex 2000-sarjan konenäköanturia ja KUKA KRC5 robotin ohjainta ei ole mahdollista saada kommunikoimaan riittävällä tasolla ilman lisäkustannuksia.

Yksi mahdollisuus olisi ollut ostaa KUKAlta Profinet -optiopaketti, joka on ohjelmistopaketti, joka asennetaan robotin ohjaimelle. Profinet-paketin avulla robotin ohjain olisi voitu konfiguroida controlleriksi, joka ohjaa kommunikaatiota ja konenäköanturi olisi tällöin konfiguroitu laitteeksi, joka vastaanottaa käskyjä controllerilta ja palauttaa sille tietoa. Kommunikaatio olisi ollut mahdollista kiertää myös Siemensin ohjelmoitavan logiikan kautta. Laboratoriolla oli ennestään käytössä Siemensin ohjelmoitavia logiikoita, jolloin normaalissa teollisessa sovelluksessa tämä olisi saattanut olla loogisin vaihtoehto.

Laboratoriota ei haluttu kuitenkaan rakentaa yksinomaan yhden laitetoimittajan laitteiston pohjalle. Lisäksi robotin toimittaneen yrityksen suositus oli, että KUKAn robotin kanssa on helpointa käyttää pääsääntöisesti EtherCAT-väylää, koska toimitetun robottijärjestelmän kommunikaatio on rakennettu sen pohjalle. Tällöin oli loogista rakentaa järjestelmä Beckhoff Automation -yrityksen laitteilla.

Beckhoff Automationin myyntiedustaja suositteli laboratorion käyttötarpeeseen CX9020 sulautettua tietokonetta (Embedded PC) ja EL6695-yhdyskäytävää (EtherCAT bridge terminal). Beckhoffin termistön mukaisesti CX9020-moduuli on ohjelmoitava logiikka, kun sitä käytetään yhdessä yhtiön TwinCAT -ohjelmointiympäristön kanssa (Beckhoff 2024a). EL6695-yhdyskäytävä tarvitaan,

koska robotin ohjaimen on pakko olla EtherCAT-verkossa päälaitte (MainDevice, ennen master) ja meidän tapauksessamme on järkevää konfiguroida myös PLC päälaitteeksi, jotta voimme tulevaisuudessa tehdä myös sovelluksia, joissa PLC toimii järjestelmässä ohjaavana laitteena. EtherCAT-verkossa kaksi päälaitetta ei voi kommunikoida keskenään ilman yhdyskäytävää.

Toimitettuun robottijärjestelmään kuului vielä käsivarren, käsiohjaimen ja robotin ohjaimen lisäksi turvalaite- ja I/O-kaappi. Laitekaapissa on EK1100 EtherCAT -kytkentälaitte (Coupler) ja siihen liitettynä 16 kappaletta digitaalisia tuloja ja lähtöjä sisältävät terminaalit (Bus terminal). EK1100 on EtherCAT-verkossa alilaitte (SubDevice, ennen slave), joka on liitetty Ethernet -kaapelilla robotin ohjaimen XF8-liitäntään. XF8-liitäntä on tarkoitettu EtherCAT-verkon alilaitteille (KUKA 2023, luku 5.9.3.5). Koska EL6695-yhdyskäytävä näkyy robotille EtherCAT-verkon alilaitteena, se kannattaa ketjuttaa EK1100-kytkentälaitteelta eteenpäin kuuluvaksi samaan verkkoon. EL6695-yhdyskäytävä kiinnittyy suoraan CX9020 logiikkaan laitteiden kyljessä olevilla terminaaliliitännöillä.

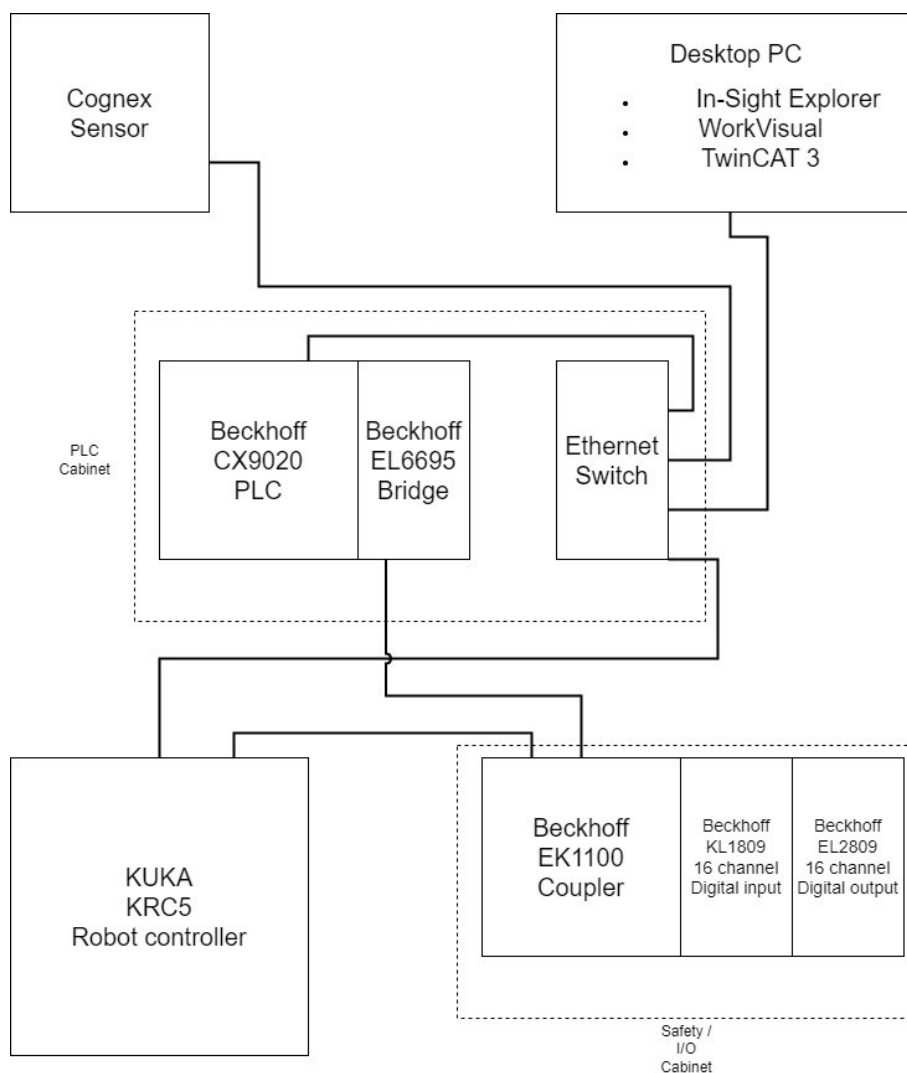
Järjestelmään on liitettävä tietokone, jolla konfiguroidaan ja ohjelmoidaan robotti, konenäkölaite ja PLC. Tämä oli helpointa toteuttaa siten, että laitteet kytketään verkkokytkimen kautta osaksi samaa Ethernet-verkkoa. Jokainen laite piti asettaa käyttämään kiinteää IP-osoitetta. Koska robotiikan laboratorion kaikki tietokoneet oli aiemmin määritelty käyttämään kiinteitä verkko-osoitteita samalla aliverkolla 192.168.100.x, se oli luonnollinen valinta myös tässä tapauksessa. Tietokoneen ja robotin ohjaimen yhdistäminen ei käy kuitenkaan EtherCAT-verkon kautta, joten verkkokytkimeltä pitää olla vielä kaapeli kytkettynä robotin ohjaimen porttiin XF5, joka on tarkoitettu ulkoisten laitteiden, kuten PLC:n tai tietokoneen liittämiseen. (KUKA 2023, luku 5.9.3.4.)

Verkko-osoitteiksi valikoituvat:

- Tietokone: 192.168.100.121
- Konenäkö: 192.168.100.32
- Robotin ohjain: 192.168.100.140
- PLC: 192.168.100.220

Tietokoneelle asennettiin KUKA WorkVisual 6.0 robotin konfigurointia ja ohjelmointia varten, Cognex In-Sight 6.5.0 konenäköanturin käyttöä varten sekä TwinCAT 3 logiikan konfigurointiin ja ohjelmointiin.

OPC UA oli hyvä valinta tässä vaiheessa konenäköanturin ja PLC:n kommunikaatioon, koska anturi osaa toimia OPC UA palvelimena (server) ja PLC voidaan säätää toimimaan OPC UA asiakkaana (client). Tämä tapahtuu TF6100-lisäosalla, joka on maksullinen OPC UA kirjasto TwinCAT -ohjelmointiympäristöön. TF6100 on Beckhoffin spesifikaation mukaan pääsääntöisesti tarkoitettu datan siirtämiseen tuotantotasolta ERP- tai pilvijärjestelmiin (Beckhoff 2024b). Tästä huolimatta OPC UA riittää luotettavuuden ja nopeuden osalta tällä hetkellä laboratorion käyttöön. Tiedonsiirron nopeuden kannalta saattaisi tulla haasteita, jos pitäisi tehdä sovellus, jossa robotille pitäisi saada konenäkölaitteelta viestejä useita kertoja sekunnissa.



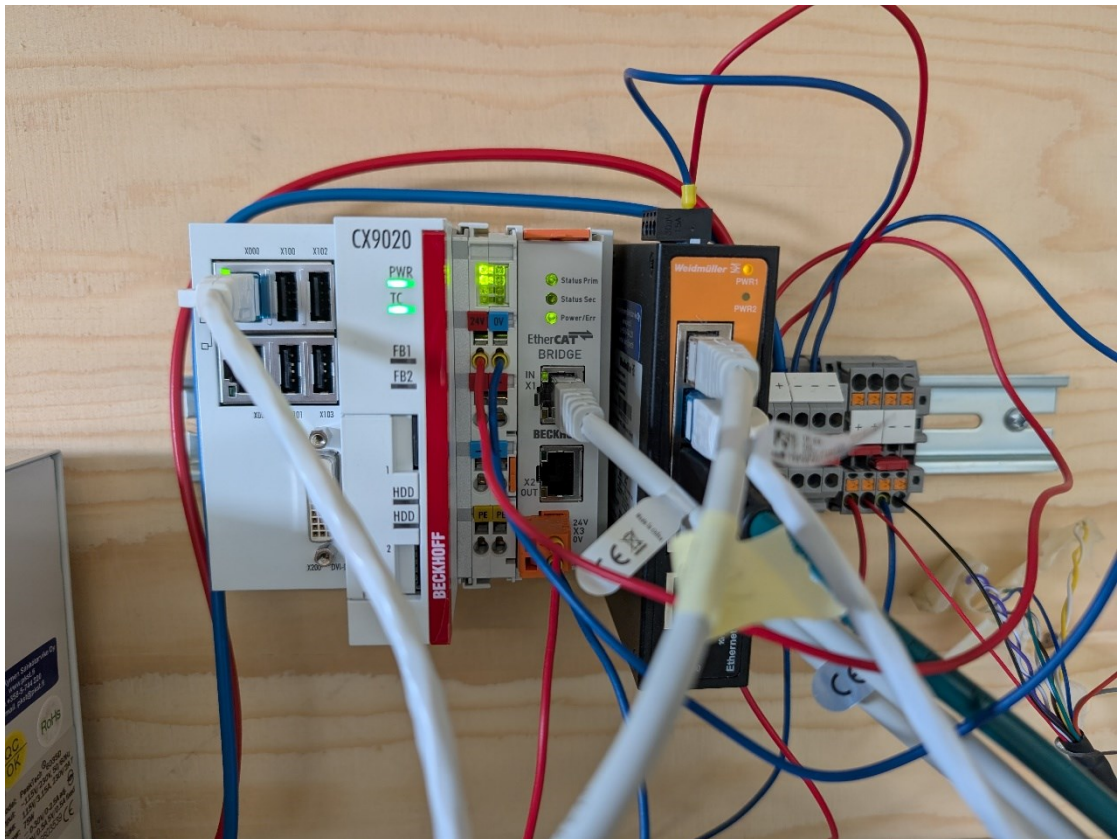
Kuva 6. Järjestelmän suunnitelma kaaviona

4.2 Järjestelmän asennus ja konfigurointi

Järjestelmän asennuksesta ja konfiguroinnista kerrotaan tässä luvussa vaiheittain. Osa vaiheista selostetaan hyvin karkealla tasolla, koska raportin ei ole tarkoitus toimia valmiina työohjeena mahdollisiin laboratorioharjoituksiin.

Ennen tämän projektin aloitusta robotti oli jo asennettu betonilattiaan kiinni. Robotin ohjain ja laitekaappi asennettiin seinälle robotin viereen. Robotin ja seinän välissä on 1500mm x 700mm pöytä, jota käytetään työtasona testiohjelmalle.

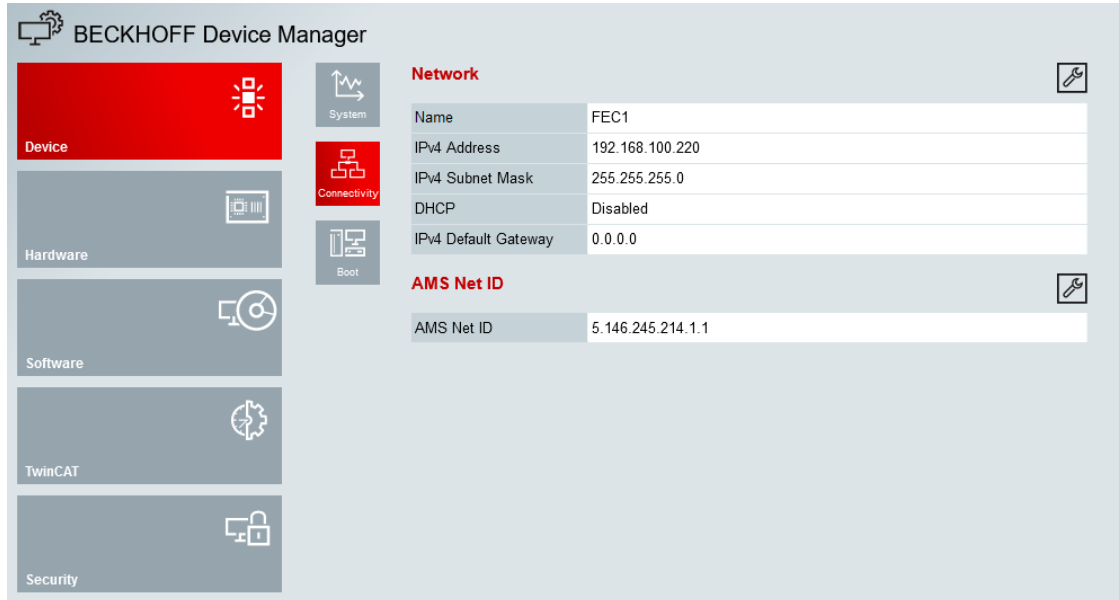
4.2.1 Beckhoff CX9020 ja EL6695



Kuva 7. Beckhoff CX9020, EL6695 ja verkkokytin.

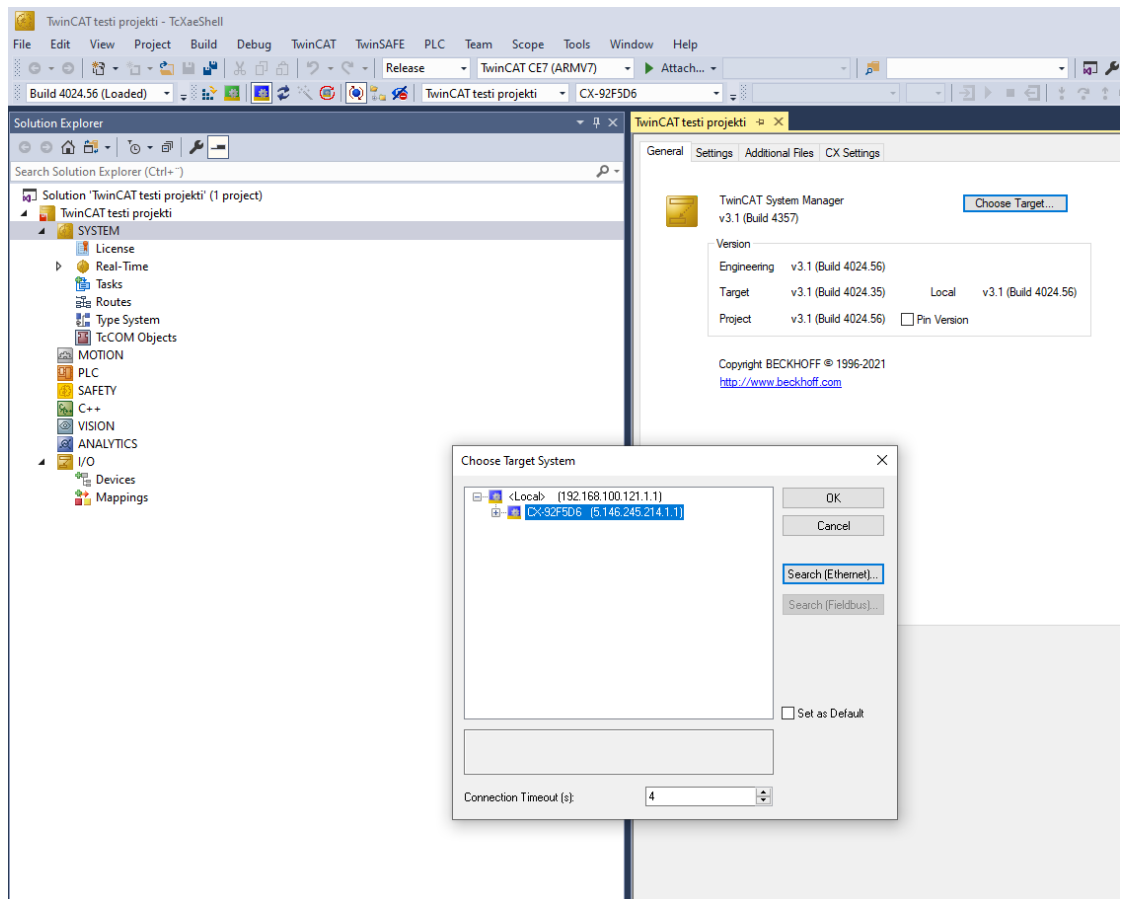
Ensimmäisenä asensin ohjelmoitavan logiikan ja yhdyskäytävän. Tässä vaiheessa projektia laitteet liitettiin toisiinsa ja asennettiin DIN-kiskolla robotin turvalaitekaapin yläpuolelle seinään kiinni. Samaan DIN-kiskoon tuli myös verkkokytin. Laitteille annetaan virta laboriovirtalähteellä. Laitteille on tarkoitus hankkia oma virtalähde ja laitekaappi myöhemmin sekä tehdä siistimmät kytkennät. CX9020 on liitetty verkkokyttimeen tavallisella Ethernet-kaapelilla.

Kun laitteessa on virrat, pääsee logiikan asetuksia muokkaamaan samaan verkkoon liitetyn tietokoneen verkkoselaimella. Oleellisinta on ottaa laitteen DHCP pois päältä ja asettaa kiinteä verkko-osoite (kuva 8).



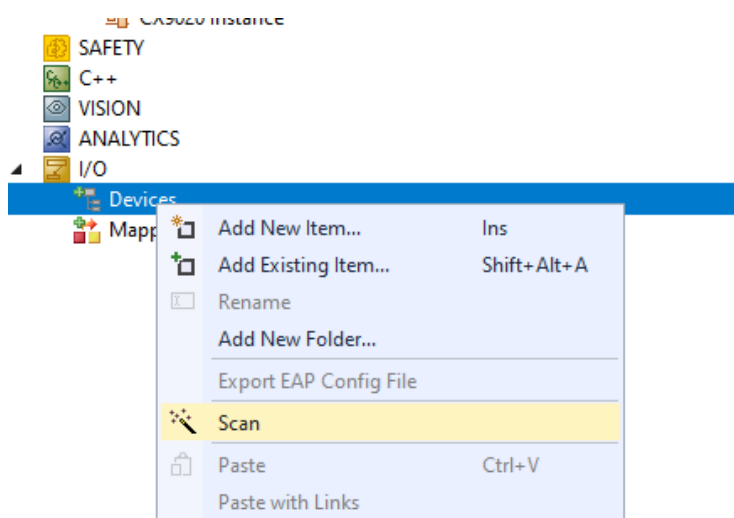
Kuva 8. Beckhoff device manager.

Laitteen käyttämiseksi pitää seuraavaksi käynnistää TwinCAT -ohjelmisto ja luoda sinne uusi projekti. Kun laitetta yhdistetään ensimmäistä kertaa, se pitää etsiä "choose target" -toiminnolla (kuva 9). Etsiminen voidaan tehdä verkko-osoitteen tai isäntänimen perusteella.



Kuva 9. TwinCAT 3.

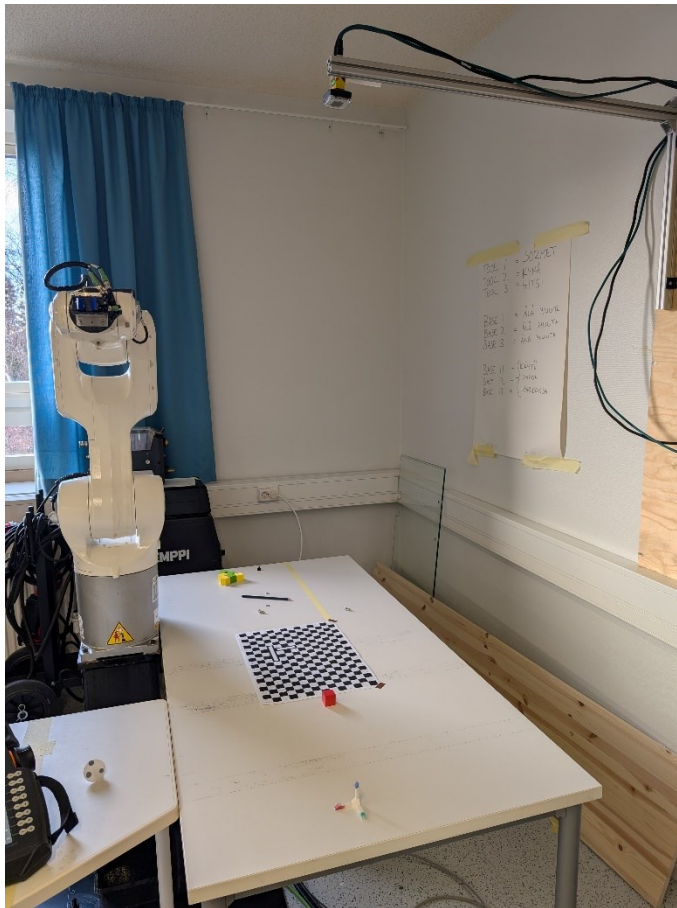
EL6695-yhdyskäytävä saadaan liitettyä projektiin valitsemalla I/O -> Devices ja "Scan" (kuva 10). Kysyttäessä valitaan ainoastaan EtherCAT. Tällöin TwinCAT yrittää etsiä kaikki yhteensopivat CX9020 -laitteeseen liitetyt terminaalit.



Kuva 10. TwinCAT 3, Scan devices.

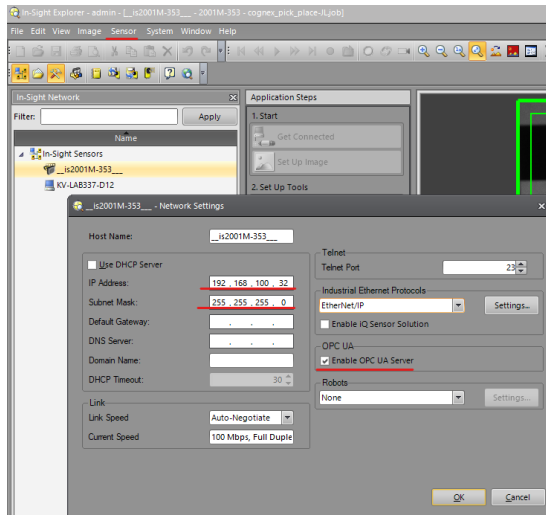
4.2.2 Cognex-konenäköanturin asennus

Konenäköanturille tein alumiiniprofiilista seinään kiinnitettävän puomin robotin vieressä olevan pöydän yläpuolelle. Puomin korkeus on säädettävissä, koska kameran kuvausalueen kokoa ei voi säätää muuten, kuin säätämällä kameran etäisyyttä kuvattavasta työalueesta (pöydän kannesta). Mittasin anturista kiinnitysruuvien paikat ja mallinsin sekä tulostin laboratorion 3D-tulostimella säädettävän kiinnikkeen konenäköanturin kiinnittämiseksi alumiiniprofiiliin. Anturi on liitetty verkkokaapelilla verkkokytkimeen, joka on PLC:n vieressä. Laite saa virran laboratoriovirtalähteeltä.



Kuva 11. Konenäköanturi ja robotti.

Laitteen konfiguraatio hoidetaan In-Sight Explorer -ohjelmistolla. Projektin aluksi on oleellista vaihtaa verkkoasetukset, siten, että DHCP otetaan pois päältä, verkko-osoitteeksi laitetaan aiemmin päätetty 192.168.100.32, ja aliverkon peitteeksi usein käytetty 255.255.255.0. Myös OPC UA palvelin pitää laittaa päälle, jotta se on käytettävissä myöhemmässä vaiheessa (kuva 12).



Kuva 12. In-Sight Explorer.

Laitteelta pitää myös säätää kameran asetukset vastaamaan valaistusolosuhteita ja vaihtaa kamera laukeamaan kenttäväylän komennoilla. Kameran kuvausalue säädetään puomin ja 3D-tulostetun kiinnikkeen avulla sopivalle etäisyydelle ja sopivan suuntaisesti pöydän kanssa. Kameraa on melko mahdollonta saada aivan suoraan kulmaan pöydän kannen kanssa ja joissain tapauksissa on jopa suositeltavaa, että kamera olisi kulmittain haitallisten heijastusten varalta. Kalibroinnin pitäisi pystyä korjaamaan linssin aiheuttamat vääristymät kuvaan, jotka voivat haitata kuvan analysointia.

4.2.3 Robotin ja PLC:n yhdistäminen

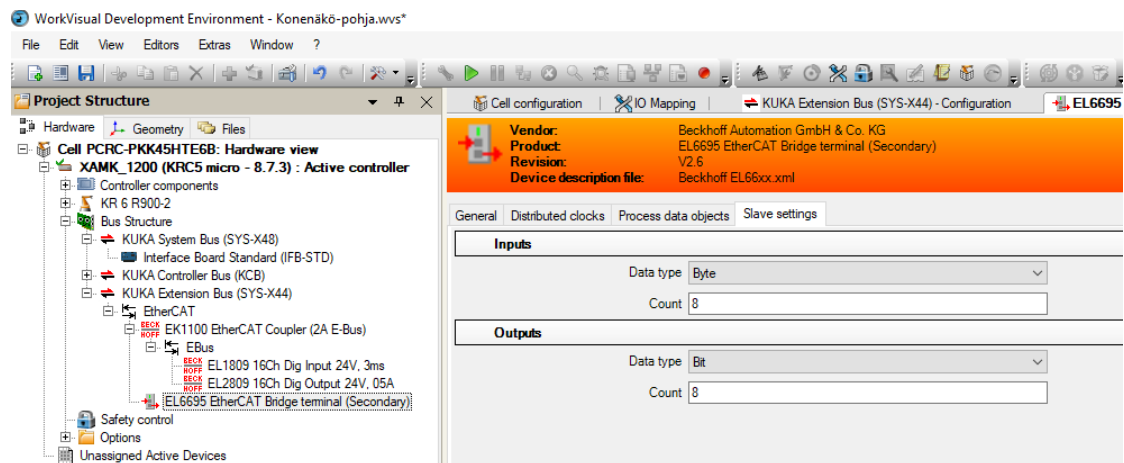
Robottijärjestelmän konfigurointi tapahtuu käyttäen tietokoneella WorkVisual -ohjelmistoa sekä robotin käsiohjainta. Robotille on jo asetettu aiemmin mainittu kiinteä verkko-osoite 192.168.100.140. Robotin ohjain on samassa lähiverkossa tietokoneen kanssa XF5-liitännän kautta, joten tietokoneelta päästään käsiksi robotin ohjaimen asetuksiin.

Robotin laitekaapissa on EK1100-kytkentälaitte, joka on yhdistetty robotin ohjaimen XF8-liitännään verkkokaapelilla. Lisäksi EK1100 on kytketty EL6695-yhdyskäytävään verkkokaapelilla.

WorkVisualissa aloitin uuden projektin tekemällä kopion toimivasta projektipohjasta, jolla robotti oli toimitettu. WorkVisualissa on samankaltainen "scan" -toiminto kuin TwinCAT-ohjelmassa, jolla voi automaattisesti etsiä EtherCAT-verkkoon asennettuja laitteita, mutta sen toiminta tuntui olevan epävarmaa.

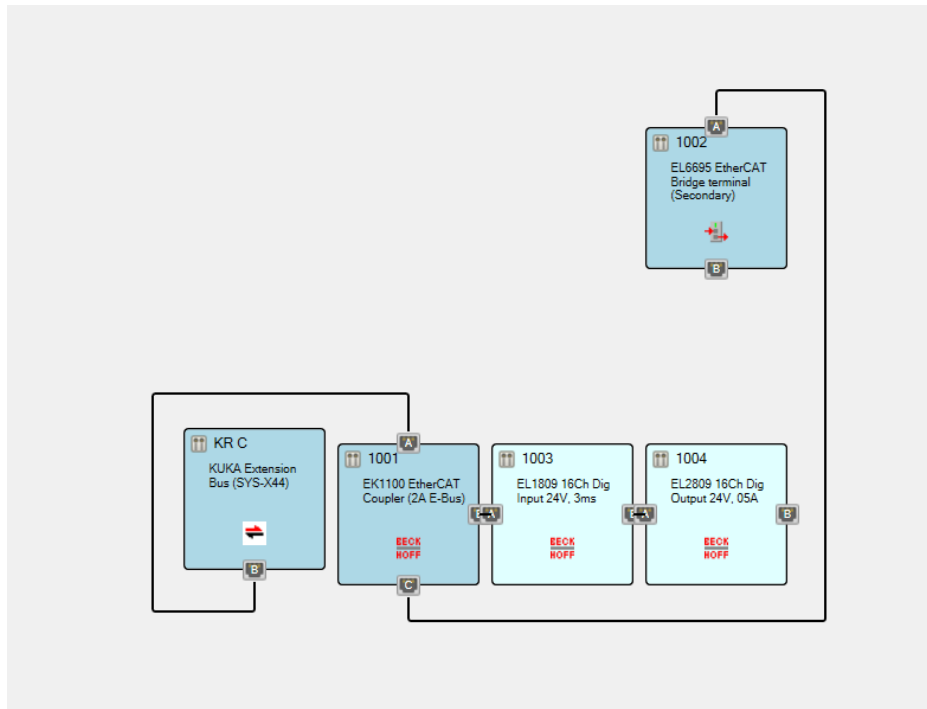
Toiminto jostain syystä päivitti myös EK1100-kytkentälaitteessa olevien liitinterminaalien tiedot ja teki sen valitettavasti väärin. Kun projekti vietiin robotin ohjaimelle WorkVisualin ”Deploy” -komennolla, ilmoitti käsiohjain ongelmia EtherCAT-verkossa. Tarkemmin etsittäessä viaksi löytyi liitinterminaalien väärät asetukset. Ongelma korjaantui tarkistamalla fyysisesti liitinterminaalien kyljestä tarkka versio ja asentamalla ne käsin oikeilla tiedoilla WorkVisualista. Koska automaattitoimintoa ei voinut käyttää, piti EL6695-yhdyskäytävälle tehdä täsmälleen sama prosessi.

Yhdyskäytävälle piti vielä konfiguroida, kuinka paljon dataa oletetaan laitteen lähettävän ja tuovan. Tässä vaiheessa arvioin, että kahdeksan tavua tulevaa tietoa pitäisi riittää ja ulospäin tarvitaan vain mahdollisesti muutamia bittejä. EL6695-manuaalin mukaan pienin mahdollinen tiedonsiirron perusyksikkö (PDO) on tavu, eli kahdeksan bittiä (Beckhoff 2024c, sivu 129). Tästä syystä valitaan lähteväksi tiedoksi kahdeksan bittiä. Jokaisen pienenkin muutoksen jälkeen konfigurointitietoihin, pitää tehdä uudelleen pari minuuttia kestävä ”Deploy”.



Kuva 13. WorkVisual, EL6695 -konfigurointi.

Robotin näkökulmasta WorkVisual-ohjelmassa valmiin järjestelmän EtherCAT-topologia näyttää kuvan 14 mukaiselta.



Kuva 14. WorkVisual, EtherCAT-verkon topologia.

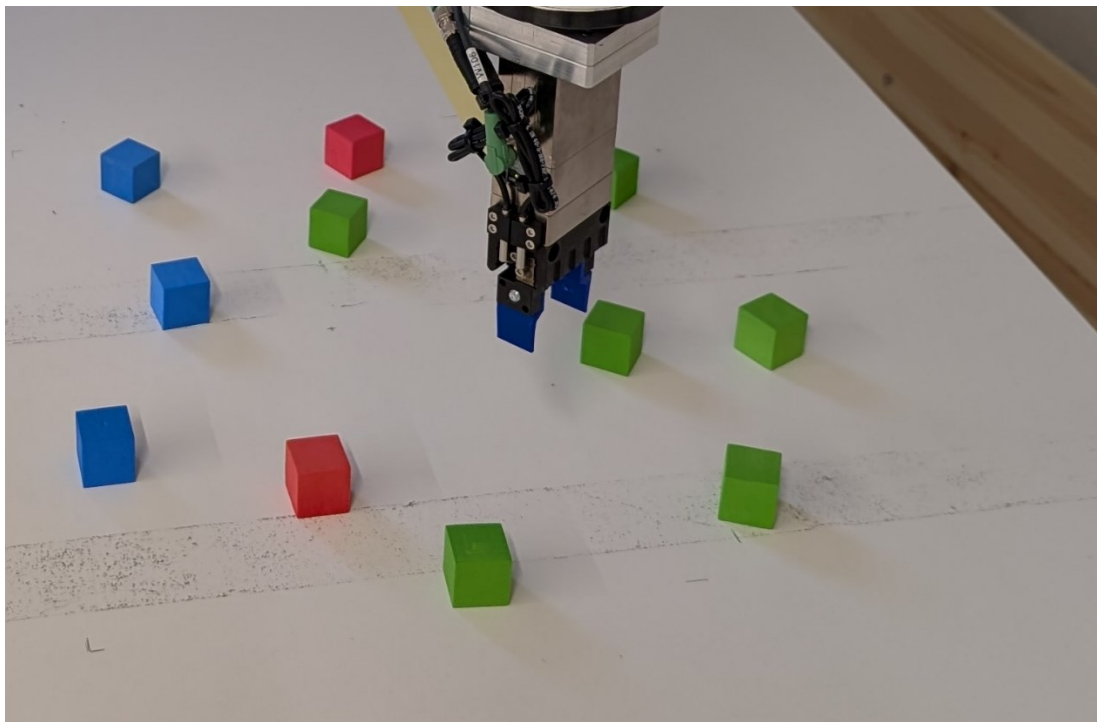
Ennen varsinaisen testausohjelman rakentamista testasin kommunikoinnin toimivuutta siten, että ohjelmoin PLC:n lähettämään EtherCATin välityksellä 16-bittisiä lukuja robotille ja robotti luonnollisesti konfiguroitiin ottamaan niitä vastaan. Käsiohjaimelta pystyi sen jälkeen tarkistamaan, saiko robotti tulotietoihin jotain bittejä päälle. Avaan konfigurointiprosessin vielä tarkemmin testausasetelmaa käsittelevässä luvussa. Kommunikointi ei lähtenyt toimimaan ensimmäisillä yrityksillä, vaan konfigurointia piti yrittää useamman kerran. Yhtenä ongelmana oli aluksi se, että yritin lähettää liian pientä määrää tietoa, vain yhtä bittiä sisään ja ulos, kunnes löysin ohjekirjasta, että pienin mahdollinen viesti on tavu. Ongelmia aiheutti myös "signal editor" ja "I/O mapping" -konfiguroinnit WorkVisualissa. Näillä konfiguroidaan robotin puolelta, miten I/O-tietoja sidotaan ja välitetään. Näistä kirjoitan lisää testausasetelmaa koskevassa luvussa.

4.3 Järjestelmän testaus

Testausasetelman perusajatuksena on rakentaa robotille sovellus, joka osaa poimia kappaleita robotin vieressä olevalta pöydältä. Kappaleiden paikkatiedot saadaan konenäköanturin avulla. PLC lukee konenäköanturilta paikkatietoja ja siirtää ne robotille. Konenäköanturi ja PLC kommunikoivat OPC UA:lla. PLC ja robotti kommunikoivat EL6695-yhdyskäytävän välityksellä EtherCAT-väylällä.

Pelkkä kommunikaation toimivuus voitaisiin todentaa yksinkertaisemmalla sovelluksella. Pienellä lisävaivalla sain todennettua myös sen, että järjestelmällä voidaan tosiaan tehdä konenäköä hyödyntäviä robottisovelluksia. Esimerkkitsovellus on epätäydellinen, mutta pyritty tekemään helposti käytettäväksi demonstraatioita varten. Sekä robotin ohjelma, että PLC:n ohjelma kommentoidaan hyvän ohjelmointitavan mukaisesti.

Poimittavaksi kappaleeksi laboratoriolta löytyy vaahtomuovisia kuutioita, jotka ovat särmältään 25 mm. Robotin tarttujalle on tulostettu sopivan kokoiset sormet kuutioiden poimintaan (kuva 15). Sormet on suunniteltu siten, että niiden ollessa auki, jää kärkien väliin noin 40 mm. Toleranssia paikoitusvirheille jää käytännössä siis noin 7 mm sivusuunnassa. Koska kuutiot ovat pehmeitä, sormilla saa niistä hyvän otteen vielä aivan reunasta. Täten toleranssia toiseen suuntaan on noin 5 mm.



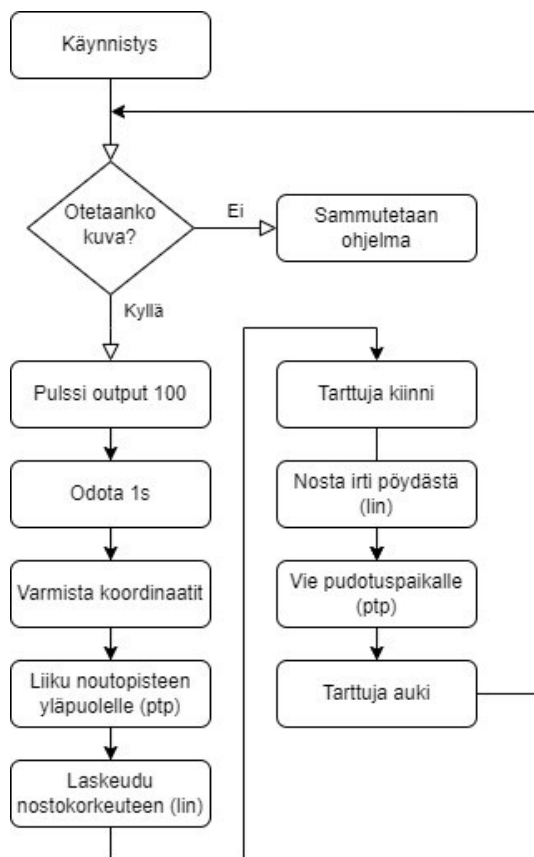
Kuva 15. Robotin tarttuja ja 3D-tulostetut sormet.

Poimintataso on pöydän kansi, eli kaksiulotteinen taso. Jotta robotti osaa hakea kaksiulotteiselta tasolta kuutioita, tarvitaan konenäköanturille ja robotille yhteinen origo sekä X- ja Y-akselit. Lisäksi tarvitaan tieto kuution rotaatiokulmasta Z-akselin ympäri.

Konenäköanturi pitää kalibroida ja sille pitää ohjelmoida tunnistettava kappale, sekä konfiguroida OPC UA-palvelin tarjoamaan haluttuja tietoja.

PLC:lle pitää tehdä projekti, johon konfiguroidaan logiikkaohjelma ja kommunikointiväylät konenäköanturille ja robotille. Logiikkaohjelman on tarkoitus olla sellainen, että logiikalle haetaan OPC UA-palvelimen X-, Y- ja rotaatio -tiedot ja ohjelma välittää niitä yhdyskäytävälle. Jos robotilta tulee pyyntö ottaa uusi kuva, välitetään laukaisupyyntö konenäköanturille.

Robotin ohjaimelle pitää konfiguroida I/O-kartoitus ja robotille pitää tehdä yksinkertainen ohjelma. Ohjelman hahmotelma vuokaaviona kuvassa 16.

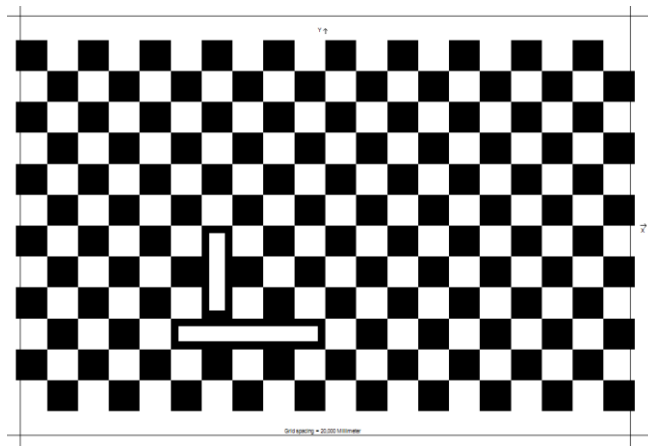


Kuva 16. Robotin ohjelman vuokaavio.

4.3.1 Konenäköanturin kalibrointi ja konfigurointi

Kamera voidaan nyt kalibroida ja ohjelmoida. Kalibroinnissa käytän A3-arkille tulostettua ruudukkoa, jossa on kohdistusmerkit (fiducial) osoittamaan X- ja Y- akselien suunnat (kuva 17). Kalibrointiruudukon saa tulostettavassa muodossa teetettyä In-Sight Explorer -ohjelmassa. Pidempi suorakulmio kertoo X-

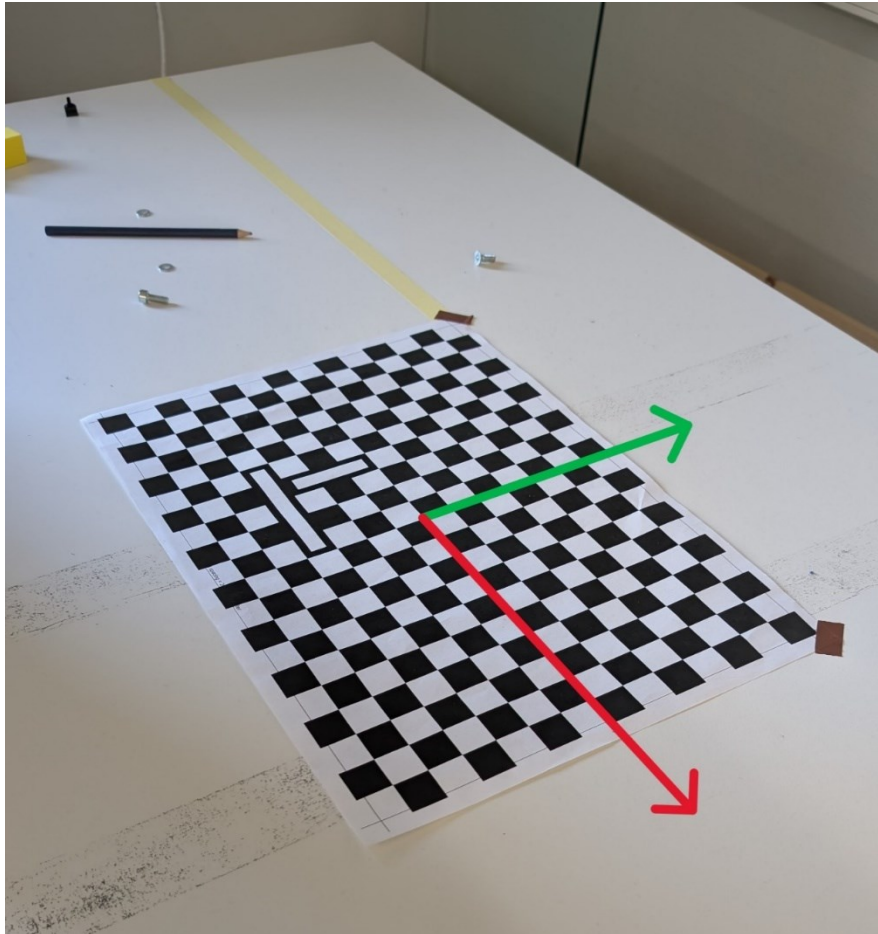
akselin suunnan. Origo määräytyy kohdistusmerkkien ulompien päiden leikkauspisteeseen, ellei sille erikseen määritetä offset -arvoa ohjelmassa.



Kuva 17. Kalibroitiruudukko.

Paperi asetellaan pöydälle halutulle työalueelle ja kohdistusmerkit sen suuntaisesti, miten akselit halutaan määritellä. Tällä on merkitystä myöhemmin, kun robotin työkoordinaatisto halutaan kalibroida samalle työalueelle. Kun kameralle kalibroitu origo ja akselien suunnat tiedetään, voidaan robotille määrittellä työkoordinaatisto, joka käyttää samaa origoa ja akseleiden suuntia, kuin anturikin.

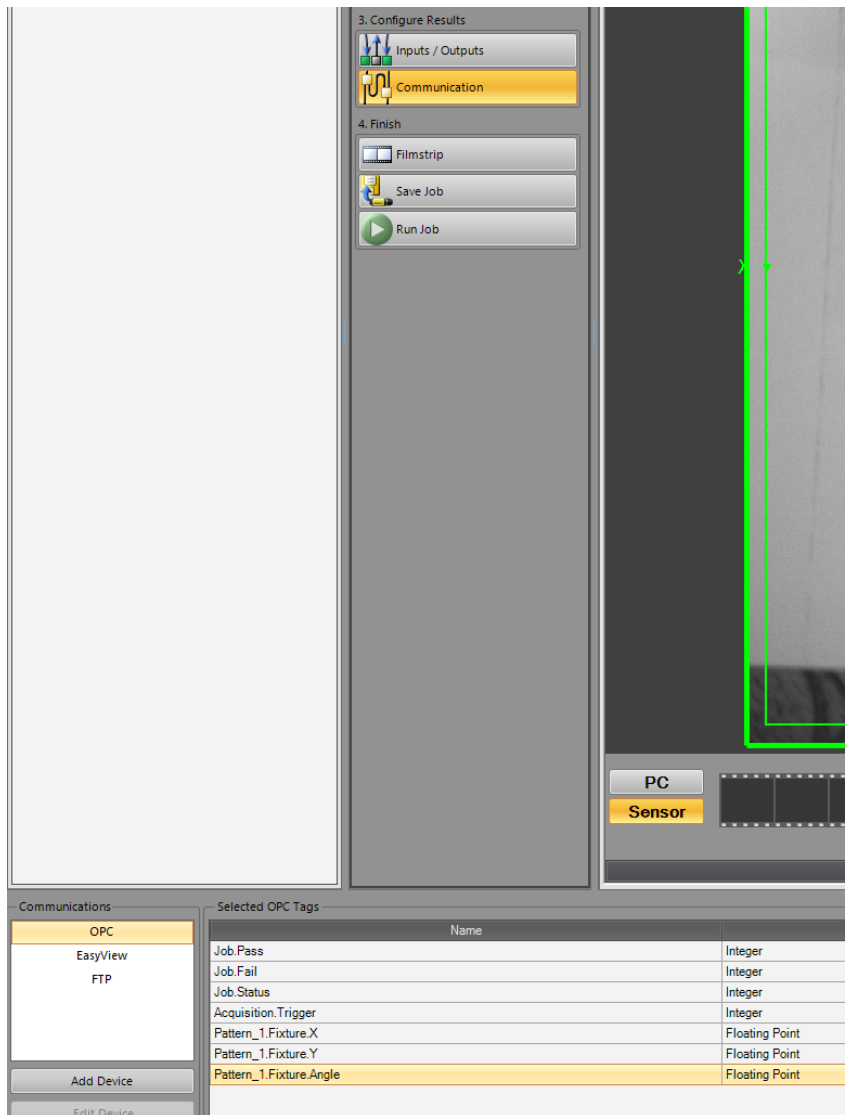
Kalibroitivaihetta demonstroivaan kuvaan olen lisännyt vihreän nuolen kuvaamaan Y-akselin positiivista suuntaa ja punaisen nuolen X-akselin positiivista suuntaa (kuva 18). Kun kamera on kalibroitu In-Sight Explorer -ohjelmassa, konenäköanturille voidaan opettaa tunnistettava kappale. Tällöin anturi osaa kappaleen löytäessään kertoa kalibroinnin perusteella kappaleen position X- ja Y-akseleilla suhteessa origoon, sekä rotaatiokulman. Anturin algoritmin ansiosta mittaustiedot kerrotaan millimetreissä, joka on juuri se tieto, jota tarvitaan myös robotin ohjaamiseen.



Kuva 18. Konenäköanturin kalibrointi, havainnekuva.

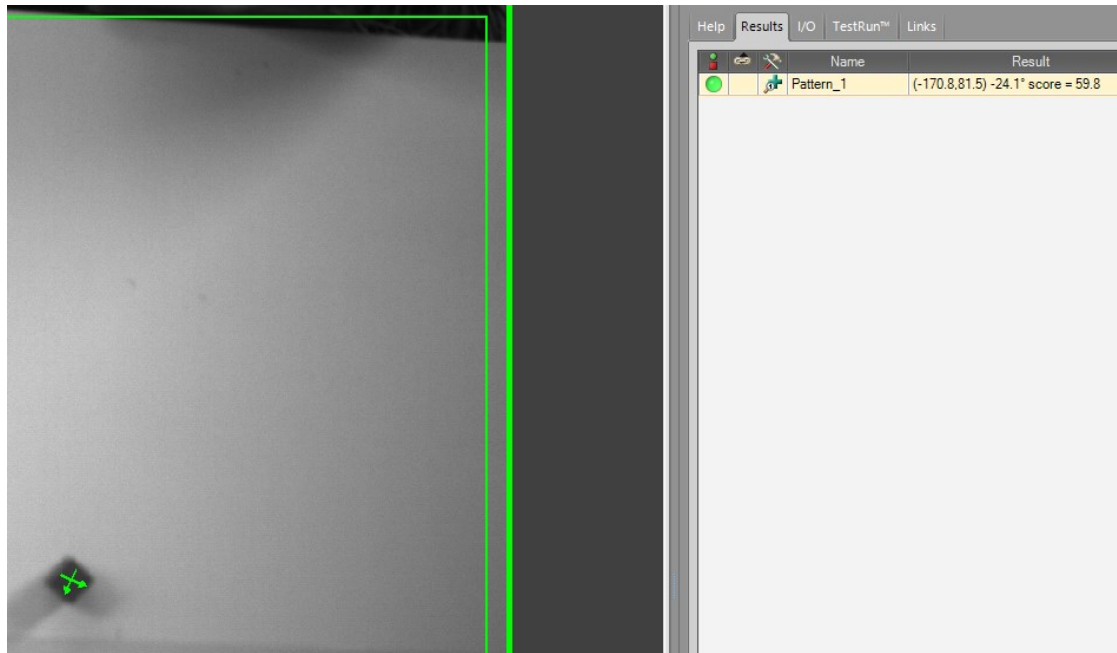
Kappaleen tunnistus määritellään myös In-Sight Explorer -ohjelmassa ja siinä käytetään Cognexin omaa algoritmia. Koska tämän raportin ei ole tarkoitus toimia suorana työohjeena, rajataan tarkempi työn kuvaus tämän osalta pois raportista.

Kun kalibrointi ja kappaleen tunnistus on tehty, siirrytään konfiguroimaan kommunikaatioasetukset. Laitteelle lisätään OPC UA -kommunikaatio ja valikosta lisätään haluttuja tietoja palvelimelle. Testin tapauksessa palvelimelle on laitettu esiin Integer-tyyppin tietoja työn onnistumisesta tai epäonnistumisesta ja Float-tyyppin tiedot tunnistettavan muodon sijoittumisesta X- ja Y-akseleilla origoon nähden, sekä rotaatio (kuva 19). Kuvassa näkyviä "Job Status"- ja "Acquisition.Trigger" -tietoja ei lopulta tarvittu.



Kuva 19. In-Sight Explorer, kommunikaatioasetuksia.

Kun kalibrointi ja konfigurointi on suoritettu, voidaan vielä tarkistaa, että annetut tiedot vaikuttavat oikeilta. Laitoin työalueelle kuution ja pyysin uutta kuvaa suoraan ohjelmasta käsin. Jos ruudulla on vihreät reunat ja ohjelma näyttää tietoja, voidaan olettaa, että samat tiedot ovat saatavissa anturin OPC UA -palvelimella (kuva 20). Tämä voitaisiin vielä tarkistaa esimerkiksi erillisellä UAExpert-ohjelmalla, jolla voitaisiin ottaa yhteys anturin OPC UA -palvelimeen ja tarkistaa siellä tarjolla olevia tietoja, mutta en nähnyt sitä tässä tarpeelliseksi, koska seuraavassa vaiheessa otetaan yhteys palvelimeen PLC:n ohjelmasta käsin, jolloin tiedot voi lukea PLC:ltä.



Kuva 20. In-Sight Explorer, havaitun kappaleen tiedot.

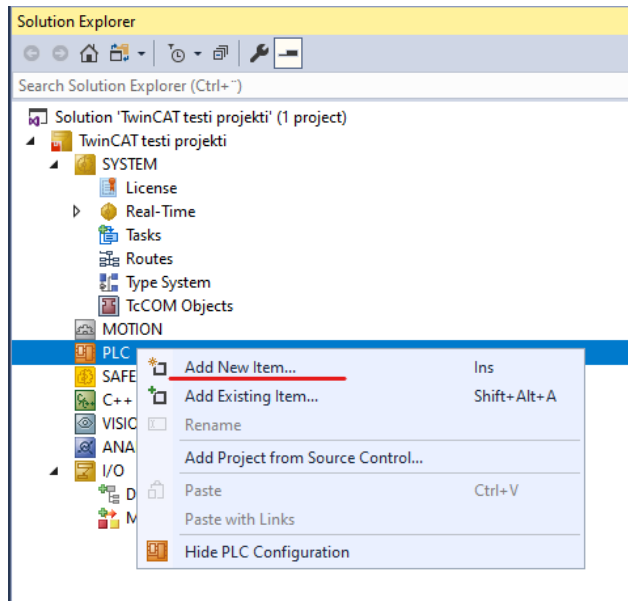
Kuvasta voidaan tulkita, että anturille opetetun muotoinen kappale on havaittu ja se on X-suunnassa -170,8 mm ja Y-suunnassa 81,5 mm origosta ja -24 asteen kulmassa.

Ennen kalibroinnin hyväksymistä tein muutamia testimittauksia, sijoittamalla kuution eri paikoille työaluetta ja vertailemalla itse käsin mitattuja tietoja anturin mittaamiin tietoihin. Mittaustarkkuuden tutkiminen ei kuitenkaan ollut tämän opinnäytteen ytimessä, joten tätä ei tehty tieteellisen tarkasti, vaan sillä ajatuksella, että tarkkuus on riittävä testausohjelman suorittamiseen. Koska työalue on valaistu vain tavallisilla luokahuoneen loisteputkilla ja ikkunasta tulevalla valolla, aiheuttavat varjot todennäköisesti enemmän ongelmia mittaustarkkuuden kanssa, kuin mikään muu tekijä. Varjot aiheuttavat sen, että anturin algoritmin voi olla vaikea päätellä, missä kappaleen reunat ovat. Tämä vaikuttaa erityisesti keskipisteen löytämiseen ja rotaatiokulman päättelyyn. Suurimmillaan käsin mitattujen ja kameran antamien tietojen välillä oli muutaman millimetrin eroja, jonka pitäisi riittää varsin hyvin. Rotaatioissa oli myös silmällä arvioiden pieniä virheitä.

4.3.2 PLC:n konfigurointi ja ohjelmointi

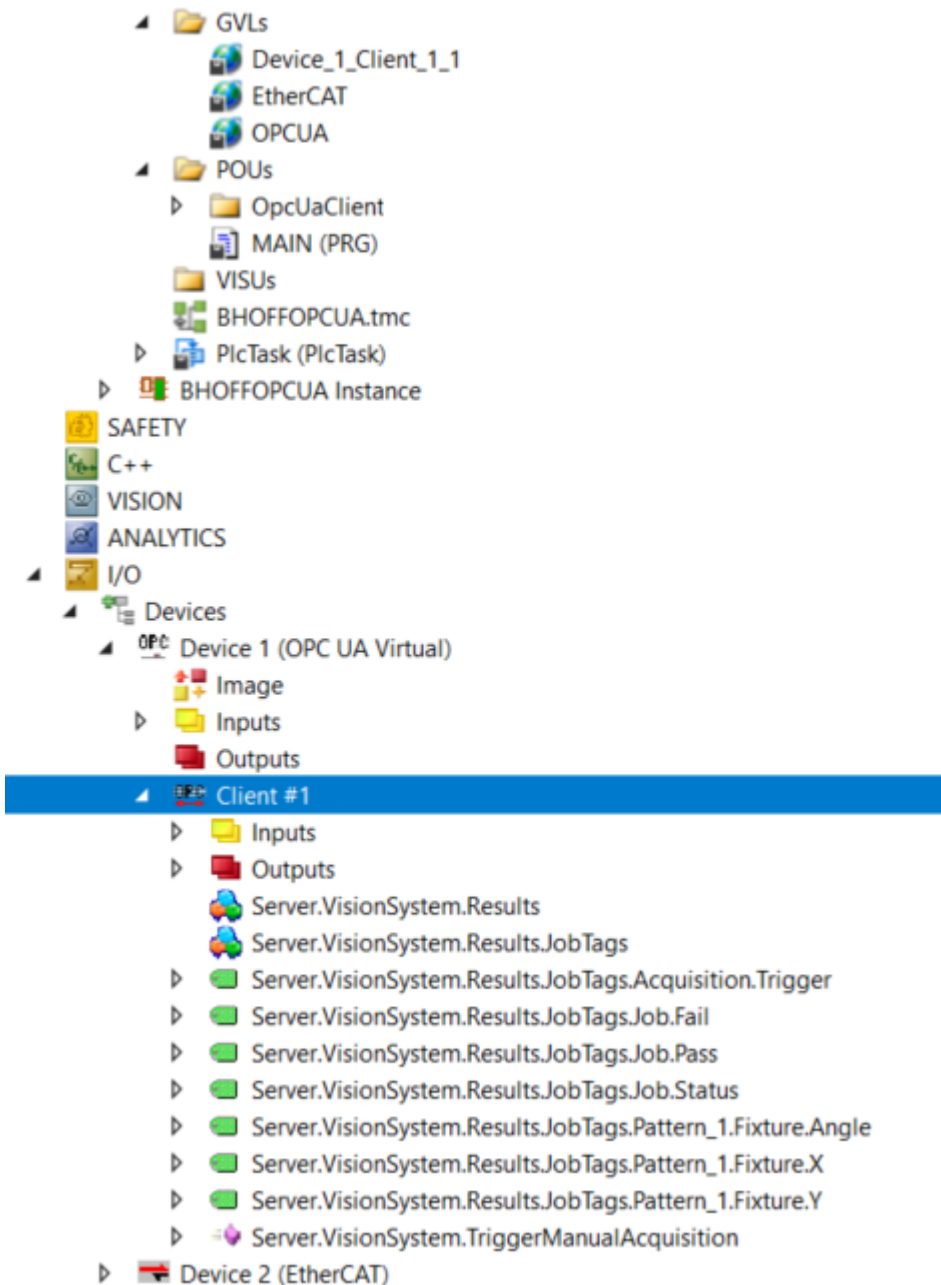
Laitteistoa asennettaessa tehtiin projektipohja, jolla varmistettiin PLC:n ja EL6695-yhdyskäytävän toimivuus. Aiemmin luotuun projektiin lisätään uusi

PLC-ohjelma, painamalla oikealla hiiren näppäintä PLC-tekstin kohdalla ja valitsemalla "Add new item..." (kuva 21).



Kuva 21. TwinCAT 3, Add new item.

Logiikkaohjelmointi ei ole sinänsä kovin oleellinen osa opinnäytteen varsinaista aihetta, mutta käsittelen tässä kuitenkin muutaman kommunikaation kannalta oleellisen asian. Projektipohjaan oli aiemmin lisätty automaattisella scan-toiminnolla EL6695-yhdyskäytävä. OPC UA -väylän käyttämiseksi, pitää projektiin konfiguroida vielä OPC UA client. TwinCAT-ohjelmistossa on tähän tarkoitukseen hyvät apuvälineet. Itse toteutin konfiguroinnin siten, että lopputuloksena TwinCAT määritteli automaattisesti logiikalle GVL-kansioon listan globaaleista muuttujista ohjelmointikäyttöön ja I/O-valikosta pystyy selaamaan konenäköanturilla olevan OPC UA -palvelimen tarjoamia muuttujia (kuva 22). X, Y ja Angle ovat anturilta tulevia oleellisia tietoja ja ne ovat 32-bittisiä REAL-muuttujia. REAL-muuttujia käytetään desimaalilukujen tai niiden likiarvojen esittämiseen. Tässä tapauksessa siis esimerkiksi aiemmassa kuvassa nähty X-akselin positiotieto oli -170,8 mm. Todellisuudessa OPC UA -palvelimella näkyvä luku on -170.75677. Tiedolla pystyttäisiin siis kertomaan positio lähes nanometrin tarkkuudella. Anturin algoritmi ja kalibrointi taas eivät todennäköisesti sellaiseen tarkkuuteen oikeasti pysty.



Kuva 22. TwinCAT 3, OPC UA palvelimen tietoja.

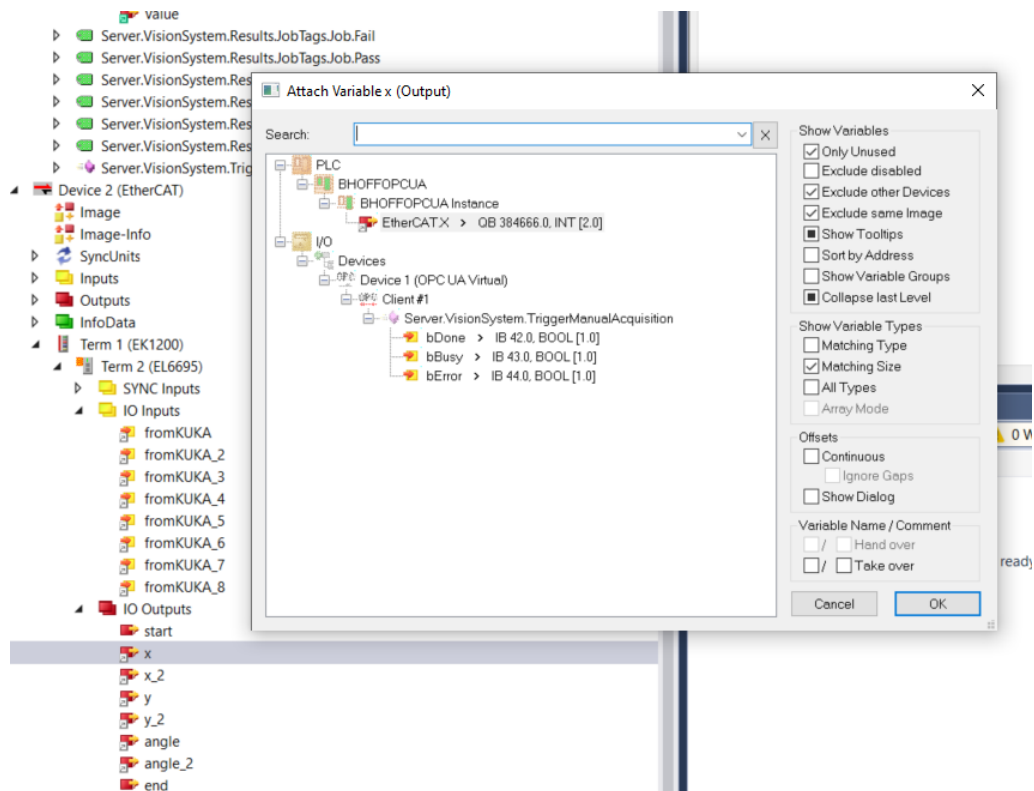
Päädyin sellaiseen ratkaisuun, että 16-bittinen Integer-tyyppi on riittävä esittämään positiotietoja tässä sovelluksessa. Tässä kohtaa tuli sellainen haaste, että KUKAn ohjelmointikielessä (KRL) 16-bittinen tieto nähdään aina etumerkittömänä kokonaislukuna, eli 0...65535. Konenäköanturin origo on kuitenkin kalibroitu siten, että tarvitsemme myös negatiivisia lukuja, jotta voimme käyttää työaluetta laajemmin. Ilman muutoksia, robotti ei osaisi tulkitä sille lähetettyjä negatiivista lukua oikein. Tämän ratkaisin siten, että logiikkaohjelmassa pyöristetään ensin OPC UA -palvelimelta saadut luvut kokonaisluvuksi ja siihen lisätään +32767. Tämä luku lähetetään robotille ja robotin ohjelmassa lu-

vusta vähennetään sama 32767. Tämä muuttaa käytettävissä olevan kokonaislukujen alueen välille -32767...32767, mutta robotille se näyttyy aina kokonaislukuna väliltä 0...65535. Tässä tapauksessa lopputulos on samankaltainen, kuin kahden komplementtia käytettäessä. Työaluetta tämä ei kohtuuttomasti rajoita. Tiedot ovat millimetreinä, joten käytettävä alue on -32,767 ... 32,767 metriä. Jos olisin halunnut tehdä positioiden määrytykset millimetrin kymmenyksien tarkkuudella, se olisi ollut mahdollista samalla 16-bittisellä INT-muuttujalla. Logiikkaohjelmassa olisi voitu kertoa robotille lähetettävä kokonaisluku kymmenellä, jolloin olisi lähetetty esimerkiksi;

$$-170,8 * 10 + 32767 = 31059$$

Tuosta luvusta olisi vähennetty taas robotin ohjelmassa 32767 ja jaettu kymmenellä, jolloin olisi saatu robotille tieto -170,8. Tämä olisi vähentänyt käytettävää aluetta, mutta olisi silti ollut yli kolme metriä suuntaansa, joka on huomattavasti suurempi alue, kuin mitä anturilla voidaan kuvata.

Jotta PLC toimisi oikein, pitää vielä linkittää I/O-tiedot OPC UA -clientilta ja EL6695 -yhdyskäytävältä PLC-ohjelman muuttujiin. Yhdyskäytävä käsittelee viestejä tavuina, joten yhdyskäytävän lähtötietoihin pitää tehdä jokaiselle 16-bittiselle tiedolle kaksi tavua, jotka linkitetään yhteen kuvan (kuva 23) mukaisesti. Aluksi yritin lähettää suoraan 16-bittisiä tietoja robotille, mutta huomasiin aika nopeasti, että muuttujien rekisterit menivät robotilla päällekkäin, jolloin ne eivät toimineet oikein. Nopein ratkaisu oli jakaa kommunikaatio tavuihin em. tavalla, mutta on hyvin mahdollista, että yhdyskäytävän voi konfiguroida lähettämään tietoa myös muuten, kuin tavuina. Tätä en kuitenkaan jäänyt tutkimaan sen enempää tähän opinnäytteeseen.



Kuva 23. TwinCAT, muuttujien linkitys.

4.3.3 Robotin konfigurointi

Kun konenäköanturille oli koordinaatisto kalibroitu, oli tarpeellista kalibroida myös robotille työkoordinaatisto saman suuntaisesti. Robotille tein työkoordinaatiston kalibroinnin robotin käsiohjaimelta.

Kun robotin ja PLC:n ohjelmat oli suunniteltu, tiedettiin mitä konfigurointeja pitää tehdä vielä robotille kommunikaation näkökulmasta. Robotilta halutaan lähettää yksi bitti, jolla pyydetään konenäköanturia ottamaan uusi kuva. Robotille halutaan tiedot havaitun kappaleen positiosta työalueella. Näille tiedoille tarvitsee määrittellä 16-bittiset muuttujat ohjelmointia varten. Määrittelin robotille globaalit muuttujat \$config.dat -tiedostoon (kuva 24).

```

;=====
; Userdefined Variables
; Omat globaalit muuttujat cognex - beckhoff - kuka -kommunikaatiota varten
; Beckhoff ohjeistuksen mukaan start ja end byte, vaikka niitä ei käytetä

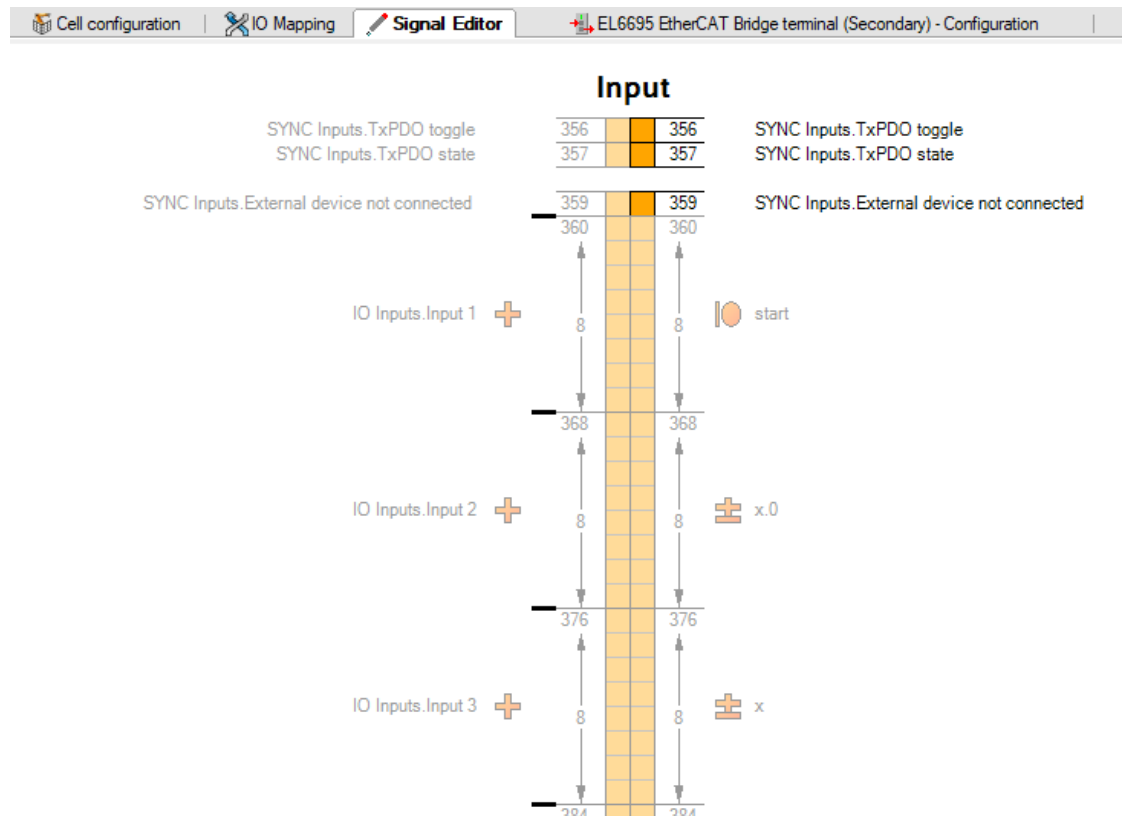
SIGNAL startti $IN[192] TO $IN[199]
SIGNAL position_x $IN[200] TO $IN[215]
SIGNAL position_y $IN[216] TO $IN[231]
SIGNAL angle $IN[232] TO $IN[247]
SIGNAL loppu $IN[248] TO $IN[255]

;=====

```

Kuva 24. WorkVisual, kuvakaappaus robotin \$config.data -tiedostosta.

Signaalieditorissa kerrotaan robotin ohjaimelle, että mitä tietoja odotetaan EL6695-yhdyskäytävän lähtöihin ja tuloihin. Tämä ei toiminut aivan kuten oletin. Oletin, että olisin voinut signaalieditorilla kertoa suoraan, että tulossa on 16-bittinen tieto, kun lähetin 16-bittisen tiedon PLC:ltä. Rekisterit menivät aina päällekkäin, enkä löytänyt asiaan muuta ratkaisua, kun jakaa tiedot tavuihin, kuten jo aiemmassa luvussa mainitsin.



Kuva 25. WorkVisual, signaalieditori.

Robotin päässä signaalit muotoilin siis siten, että ensiksi odotetaan yhtä tyhjää tavua, jonka jälkeen tulee 16 bittinen X-tieto kahtena tavuna ja niin edelleen. Tiedot on määritelty olemaan SINT, eli etumerkillisiä 8-bittisiä kokonaislukuja. Beckhoff suosittelee, että ongelmien välttämiseksi lähtömuuttujien alkuun ja loppuun lisätään tyhjä tavu. Syy tähän suositukseen liittyy EL6695:n sisäiseen muistinhallintaan. Kun laitteen viimeiseen tavuun kirjoitetaan dataa, se voi aiheuttaa siirtymisen seuraavaan EtherCAT-puskuriin. Tämä voi johtaa ongelmiin, jos ohjelma lukee dataa peräkkäin eri muistialueista ja viimeinen tavu luetaan. Tällöin puskuuri lukkiutuu eikä dataa voida enää lukea. (Beckhoff 2024c, 144.)

I/O-kartoituksella pitää vielä kertoa, mihin robotin digitaalisiin tuloihin nämä tiedot ohjataan. Haluan siis ohjata signaalieditorilla muotoillut tiedot aiemmin konfiguroituihin muuttujiin. Tämä tapahtuu siten, että WorkVisualissa ryhmitellään aiemmin määritellyt bittialueet tavun osissa ja yhdistetään tavut kuvan mukaisesti (kuva 26). Näin esimerkiksi yhdyskäytävältä tulevan 16-bittisen X-tiedon ensimmäiset 8 bittiä ohjataan robotin digitaalisiin tuloihin 200...207 ja seuraavat 8 bittiä tuloihin 208...215. Globaali muuttuja on määritelty sisältämään kaikki 16 bittiä välillä 200...2015, joten lukua voidaan jälleen käyttää 16-bittisenä.

The screenshot shows the WorkVisual I/O-mapping interface. It consists of two tables side-by-side, each with columns for Name, Type, Description, I/O, and Address.

Name	Type	Description	I/O	I/O	Name	Type	Address
SIN[192]#G	BYTE		←	← start	start	BYTE	
SIN[200]#G	SINT		←	← x.0	x.0	SINT	
SIN[208]#G	SINT		←	← x	x	SINT	
SIN[216]#G	SINT		←	← y.0	y.0	SINT	
SIN[224]#G	SINT		←	← y	y	SINT	
SIN[232]#G	SINT		←	← angle.0	angle.0	SINT	
SIN[240]#G	SINT		←	← angle	angle	SINT	
SIN[248]#G	BYTE		←	← end	end	BYTE	

Name	Type	Description	I/O	I/O	Name	Type	Address
SIN[190]	BOOL		←	← SYNC Inputs.TxPDO toggle	SYNC Inputs.TxPDO toggle	BOOL	
SIN[191]	BOOL		←	← SYNC Inputs.TxPDO state	SYNC Inputs.TxPDO state	BOOL	
SIN[192]#G	BYTE		←	← SYNC Inputs.External device ...	SYNC Inputs.External device ...	BOOL	
SIN[200]#G	SINT		←	← start	start	BYTE	
SIN[208]#G	SINT		←	← v.0	v.0	SINT	

Kuva 26. WorkVisual, I/O-mapping.

Robotin ohjelmoinnin suoritin käsiohjainta ja WorkVisualia käyttäen. Ohjelmointi ei ole tämän opinnäytetyön ytimessä, joten todetaan vain, että se on tehty alkuperäisen suunnitelman mukaisesti ja toimii.

4.4 Testaus ja tulokset

Ennen järjestelmän esittelyä tilaajalle, ajoin testiohjelman kymmeniä kertoja. Ohjelmassa on selkeitä puutteita, esimerkiksi se ei osaa ottaa huomioon tilannetta, jossa konenäköanturilta on tullut ilmoitus, että kappaletta ei ole havaittu. Ohjelmassa ei myöskään tehdä varmistusta, että anturi on oikeasti ottanut uuden kuvan. Tällaisissa tilanteissa robotti ajaa viimeksi OPC UA -palvelimelle tallennettuihin koordinaatteihin, eli paikkaan, jossa on viimeksi havaittu kappale tai sitten työalueen origoon. Toimiessaan ohjelma kuitenkin haki kappaleen aina oikeasta paikasta. Ohjelmalle piti alun perin tehdä ns. kättely, jolla robotti ja PLC olisivat vaihtaneet bitteinä tietoa, missä vaiheessa ohjelmaa mennään.

Opinnäytetyön tavoitteena ei ollut kuitenkaan tehdä täysin valmista tuotantokelpoista ohjelmaa, vaan konfiguroida järjestelmä käyttökuntoon ja varmistaa kommunikaation toimivuus sekä todentaa, että järjestelmä soveltuu konenäköä hyödyntävien robotiikka-sovellusten tekemiseen.

Virallinen testaus toteutettiin siten, että pöydälle aseteltiin satunnaisesti eri paikoille työaluetta kymmenen kuutiota ja ohjelmaa ajettiin niin kauan, että robotti oli poiminut kaikki kuutiot ja siirtänyt ne robotin vieressä olevaan laatikkoon. Testi toistettiin kolme kertaa, eli poimintoja yhteensä 30. Tämä todettiin tilaajan kanssa riittäväksi määräksi todentamaan luotettavasti, että kommunikaatio toimii ja järjestelmällä voidaan toteuttaa vastaavanlaisia monimutkaisempiakin sovelluksia.

Robotti epäonnistui kuution 11 kohdalla, koska anturi ei tunnistanut kappaletta ja robotti ajoi edellisiin koordinaatteihin. Muuten robotti onnistui hakemaan kaikki kuutiot pöydältä. Hakupaikka oli kaksi kertaa XY-suunnassa silmämääräisesti noin viisi millimetriä pielessä, mutta nosto onnistui silti. Viisi kertaa tarttujan ja kuution kulma oli hieman pielessä, mutta nosto onnistui. Robotille tulevat tiedot täsmäsivät kuitenkin konenäköanturin tarjoamiin tietoihin. Tästä voidaan päätellä, että ongelma ei ollut varsinaisesti kommunikaatiossa, vaan konenäkössä, logiikkaohjelmassa tai robotin ohjelmassa.

Yksinkertaisella testausohjelmalla ja havainnoimalla olen todentanut, että järjestelmä toimii tarkoituksenmukaisesti. Kommunikaatio robotin ja PLC:n välillä toimii EtherCAT-väylällä. Konenäköanturi kommunikoi PLC:n kanssa OPC UA:lla. PLC:hen voi liittää monenlaisia laitteita erilaisilla Ethernet pohjaisilla teollisuusväylillä tai perinteisellä I/O:lla. Lopputuloksena koulutusohjelmalla on käytettävissä robottijärjestelmä, johon on valmiit konfiguraatiot konenäköä tai muita laitteita hyödyntäviin robotiikan sovelluksiin.

Kun kommunikaation osalta pohjatyö on tehty, avautuu myös uusia mahdollisuuksia konenäön tai muiden laitteiden integraatiota tutkiviin projekti- tai opinnäytetöihin.

5 JOHTOPÄÄTÖKSET

Opinnäytetyön päätavoitteeksi oli määritelty robottijärjestelmän rakentaminen opetuskäyttöön, jota on laajennettu ulkoisella logiikalla ja konenäköanturilla. Käytössä oli KUKA KR 6 900-2 käsivarsirobotti, Cognex 2001M-353 konenäköanturi sekä työtä varten hankittu Beckhoffin CX9020 sulautettu PC, EL6695 -yhdyskäytävällä.

Toimivaan tulokseen pääsin siten, että kommunikaatio konenäköanturin ja PLC:n välillä käyttää OPC UA -väylää sekä kommunikaatio PLC:n ja robotin välillä EtherCAT -väylää. PLC ja robotti ovat molemmat EtherCAT-verkossa päälaitteita (MainDevice), jotka eivät voi kommunikoida keskenään ilman erillistä yhdyskäytävää. Koulutusohjelmalla on nyt käytössään monipuolinen robottijärjestelmä, joka on valmiiksi konfiguroitu erilaisia sovelluksia varten. Järjestelmään on integroitu konenäköanturi ja siihen voidaan integroida myös muita laitteita perinteisellä I/O:lla, EtherCAT-väylällä tai OPC UA-standardilla. Järjestelmää voidaan hyödyntää robotiikan opetuksessa ja tutkimuksessa, ja se tarjoaa opiskelijoille mahdollisuuden tutustua ja harjoitella monenlaisten robotiikkasovellusten parissa.

Robottijärjestelmään jäi vielä paljon kehitettävää. Laitteiden kotelointi ja siistimmät asennustyöt jäivät myöhempään ajankohtaan. Robotille voitaisiin hankkia KUKAn Profinet -optiopaketti, jolloin järjestelmä olisi mahdollista tehdä laboratorion Siemensin logiikoilla. Profinet ja EtherCAT pitäisi toimia samassa lähiverkossa, joten pitäisi olla myös mahdollista käyttää samassa järjestelmässä Siemensin ja Beckhoffin PLC:tä. Konenäköanturikin tukee Profinet-väylää. Olisi hyödyllistä vertailla konenäköanturin kommunikaation toimivuutta Profinet ja OPC UA -tekniikoiden välillä.

Beckhoffin PLC:lle voitaisiin hankkia analogisia tuloterminaleja, jotta robotin kanssa voitaisiin käyttää analogista tietoa antavia antureita. Robotille on jo suunniteltu laseretäisyysmittarin hankintaa, jonka käyttöä varten em. terminaalit tarvittaisiin. Samat terminaalit voitaisiin toki laittaa myös suoraan robotin EK1100-kytkentäterminaaliin. Beckhoffin PLC:tä voidaan myös laajentaa muita teollisen Ethernetin väyliä tukevilla yhdyskäytävillä. Järjestelmä voi myös välittää tietoa IoT-verkkoon, jos PLC on yhteydessä Internetiin. Tässä

opinnäytetyössä ei käsitelty ollenkaan kommunikaation turvallisuutta. Se on tietysti oleellinen osa kommunikaatiota, mutta aihe on niin laaja, että vaatisi kokonaan toisen opinnäytetyön. Ennen tiedon välittämistä IoT-verkkoon olisi syytä perehtyä kyberturvallisuuteen syvällisemmin, ettei altisteta robotin ohjausjärjestelmiä potentiaalisten hyökkääjien saataville.

Lisää tutkimustyötä kaipaisi erityisesti EtherCAT-väyläteknologian toiminta ja EL6695-yhdyskäytävän parempi konfigurointi. Tässä työssä ei onnistuttu selvittämään, voidaanko yhdyskäytävää konfiguroida 16- tai 32-bittisten arvojen suoraan käsittelyyn. Tämä tekisi yhdyskäytävän käyttämisestä ja robotin I/O-kartoituksesta mahdollisesti hieman helpompaa ja selkeämpää.

OPC UA -kommunikaatiosta jäi tutkimatta Pub/Sub -ominaisuus. Nyt PLC:lle ohjelmoitu OPC UA Client tekee tilakyselyn (polling) määrätyn aikavälein konenäköanturin OPC UA palvelimelle. PLC:lle konfiguroitu aika on 500 ms, eli käytännössä OPC UA Client pyytää kaksi kertaa sekunnissa palvelinta lähettämään tietoa. Tässäkään kohtaa ohjelma ei automaattisesti osaisi päätellä, että onko tieto uutta vai ei. Jatkuva kysely on todella tehoton tapa hoitaa tiedonvälitystä, erityisesti, jos verkossa on paljon laitteita. Tämä on ehkä suurin yksittäinen tekijä, jonka vuoksi OPC UA soveltuu huonosti teollisuuden kenttä-laitteiden käyttöön, vaikka tällaisen pienen robottijärjestelmän käytössä se saattaakin toimia kelpollisesti. Beckhoffilla on TwinCAT-ohjelmistolle TF6105-paketti, jolla voisi olla mahdollista saada käytettyä OPC UA:n Pub/Sub -ominaisuutta konenäköanturin kanssa. Tällöin PLC:n Clientilta määritellään tilaus palvelimelta haluttuihin tietoihin ja palvelin voidaan teoriassa konfiguroida lähettämään tietoa vain silloin, kuin tietoon tulee muutos. Cognex vihjailee tällaisen ominaisuuden olemassaolosta, mutta ei viittaa siihen suoraan virallisella PubSub -termillä (Cognex 2023).

Kokonaisuutena tämä opinnäytetyö onnistui tavoitteessaan, vaikka lisäselvityksien tarvetta jäi vielä paljon. Robottijärjestelmä kuitenkin toimii jo nykyisellään tilaajan asettamien tavoitteiden mukaisesti ja lisäselvityksillä sen toimivuutta ja ominaisuuksia voidaan parantaa sekä lisätä entisestään. Työssä pääsi tutustumaan monipuolisesti moniin menetelmiin, joita robottijärjestelmän suunnittelussa ja toteuttamisessa tarvitaan.

LÄHTEET

Beckhoff, 2024a. CX9020. WWW-dokumentti. Saatavissa: <https://www.beckhoff.com/fi-fi/products/ipc/embedded-pcs/cx9020-arm-r-cortex-r-a8/cx9020.html> [viitattu: 10.11.2024]

Beckhoff, 2024b. TF6100 TwinCAT 3 OPC UA. WWW-dokumentti. Saatavissa: <https://www.beckhoff.com/fi-fi/products/automation/twincat/tfxxx-twincat-3-functions/tf6xxx-connectivity/tf6100.html>? [viitattu 10.11.2024]

Beckhoff, 2024c. Documentation. EL6695 EtherCAT Bridge Terminal. PDF-dokumentti. Saatavissa: https://www.beckhoff.com/fi-fi/support/download-finder/search-result/?download_group=57249337&download_item=57249341 [viitattu 12.11.2024]

Billing, M. Teollisuusrobotti. Teoksessa: Välimäki, K (toim.) Teollisuuden robotiikka. Helsinki: Suomen Robotiikkayhdistys ry. E-kirja. Saatavissa: <https://kaakkuri.finna.fi/Record/kaakkuri.231319?sid=4840498519> [viitattu: 23.11.2024]

Cognex. 2023. OPC UA. Päivitetty 8.5.2023. WWW-dokumentti. Saatavissa: https://support.cognex.com/docs/is2d_2310/web/EN/InSight_Sheet/Content/Topics/Comm/OPCUA-Intro.htm?TocPath=Industrial%20Communications%7COPC%20UA%7C_0 [viitattu 23.11.2024]

Cognex, 2024a. COGNEX INDUSTRIAL CAMERAS (CIC). WWW-dokumentti. Saatavissa: <https://www.cognex.com/products/machine-vision/vision-software/vision-accessories/cic-cameras> [viitattu 2.11.2024]

Cognex, 2024b. In-Sight 2000 Vision Sensors. WWW-dokumentti. Saatavissa: <https://www.cognex.com/products/machine-vision/vision-sensors/in-sight-2000-vision-sensors> [viitattu 2.11.2024]

Cognex, 2024c. Liquid Lens technology. WWW-dokumentti. Saatavissa: <https://www.cognex.com/products/leading-technology/liquid-lens-technology> [viitattu 3.11.2024]

Cognex, 2024d. In-Sight 2000 Series Vision Sensor. Reference guide. Päivitetty 10.10.2024. PDF-dokumentti. Saatavissa: https://support.cognex.com/docs/is_651/ISE/EN/Manuals/is2000inst.pdf [viitattu 3.11.2024]

Davies, E. R. 2018. Computer Vision (Fifth Edition). Academic Press. E-kirja. Saatavissa: https://kaakkuri.finna.fi/Record/nelli29_mamk.4100000000972623?sid=4853858455

ETG (EtherCAT Technology Group). 2024. EtherCAT – The Ethernet Fieldbus. PDF-dokumentti. Saatavissa: <https://www.ethercat.org/en/technology.html> [viitattu 8.11.2024]

Hanssen, D. H. 2015. Programmable Logic Controllers. John Wiley & Sons, Incorporated. E-kirja. Saatavissa: https://kaakkuri.finna.fi/Record/nelli29_mamk.3710000000476533?sid=4863098191 [viitattu 7.11.2024]

Hägele, M., Nilsson, K., Pires, J & Bischoff, R. Industrial Robotics. Teoksessa Siciliano, B & Khatib, O (toim.). 2016. Springer Handbook of Robotics. Springer International Publishing. E-kirja. Saatavissa: https://kaakkuri.finna.fi/Record/nelli29_mamk.3710000000765075?sid=4844398798 [viitattu 18.10.2024]

IFR, 2024. Top 5 Robot trends 2024. IFR. WWW-dokumentti. Päivitetty 15.2.2024. Saatavissa <https://ifr.org/ifr-press-releases/news/top-5-robot-trends-2024> [viitattu 13.3.2024]

ISO 8373. 2021. Robotics – Vocabulary.

KUKA Deutschland GmbH, 2024. KUKA KR 6 R900-2. PDF-dokumentti. Päivitetty 7.3.2024. Saatavissa: <https://www.kuka.com/event/media?itemId=73A87D52D8394BDABD708B79C7B5E1BE> [viitattu: 23.10.2024]

KUKA Deutschland GmbH, 2023. KUKA KR C5 micro. Assembly instructions. PDF-dokumentti. Saatavissa: <https://xpert.kuka.com/service-express/portal/object/kukaid/PB11486> [viitattu 10.11.2024]

Knöpfe, A. 2024. KR C5 micro Product Presentation. PPTX-dokumentti. Päivitetty 20.6.2024. Saatavissa <https://my.kuka.com/sfc/servlet.shepherd/document/download/0691i000009xU3jAAE> [viitattu 24.10.2024]

Lempiäinen, J. 2024. Teollisuuden robotiikka Suomessa. Teoksessa: Välimäki, K (toim.) Teollisuuden robotiikka. Helsinki: Suomen Robotiikkayhdistys ry. E-kirja. Saatavissa: <https://kaakkuri.finna.fi/Record/kaakkuri.231319?sid=4840498519> [viitattu: 13.10.2024]

Liuha, A & Aro, J. 2024. Aistijärjestelmät ja kommunikointi. Teoksessa: Välimäki, K (toim.) Teollisuuden robotiikka. Helsinki: Suomen Robotiikkayhdistys ry. E-kirja. Saatavissa: <https://kaakkuri.finna.fi/Record/kaakkuri.231319?sid=4840498519> [viitattu: 4.11.2024]

Nortio, J. 2024. Teknologiatrendit Osa 2: Teknologioita trendikäyrän huipulta. WWW-dokumentti. Päivitetty 10.1.2024. Saatavissa: <https://www.automatiovayla.fi/artikkelit/teknologiatrendit-osa-2-teknologioita-trendikayran-huipulta/> [Viitattu 7.4.2024]

Ojasalo, K., Moilanen, T. & Ritalahti, K. 2015. Kehittämistyön menetelmät: Uudenlaista osaamista liiketoimintaan. 3.–4. painos. Helsinki: Sanoma Pro Oy. E-kirja. Saatavissa: <https://www.elibslibrary.com/fi/book/978-952-63-2695-5> [viitattu 1.8.2024].

OPC Foundation, 2024a. Unified Architecture. WWW-dokumentti. Saatavissa: <https://opcfoundation.org/about/opc-technologies/opc-ua/> [viitattu 20.11.2024]

OPC Foundation, 2024b. OPC UA Roadmap. WWW-dokumentti. Saatavissa: <https://opcfoundation.org/about/opc-technologies/opc-ua/opcua-roadmap/> [viitattu 21.11.2024]

Pertsev, A s.a. Coord planes color. CC-BY-SA-4.0. Wikimedia commons. WWW-dokumentti. Saatavissa: https://commons.wikimedia.org/wiki/File:Coord_planes_color.svg [viitattu 12.10.2024].

Siltala, N & Gautam M. 2024. Konenäkö. Teoksessa: Välimäki, K (toim.) Teollisuuden robotiikka. Helsinki: Suomen Robotiikkayhdistys ry. E-kirja. Saatavissa: <https://kaakkuri.finna.fi/Record/kaakkuri.231319?sid=4840498519> [viitattu: 3.11.2024]