



Osaamista
ja oivallusta
tulevaisuuden
tekemiseen

Samir Nakroufi

Yhteistyörobotin automaattinen siirto ja telakointi mobiilirobotilla

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Sähkö- ja automaatiotekniikka

Insinöörityö

18.4.2021

Tekijä Otsikko	Samir Nakroufi Yhteistyörobotin automaattinen siirto ja telakointi mobiilirobotilla
Sivumäärä Aika	52 sivua 18.4.2021
Tutkinto	insinööri (AMK)
Tutkinto-ohjelma	sähkö- ja automaatiotekniikka
Suuntautumisvaihtoehto	automaatiotekniikka
Ohjaaja	Lehtori Reijo Leinonen
<p>Insinööriyön aiheena oli kehitystyö, joka tehtiin yritykselle GE Healthcare. Työ toteutettiin osana Reboot IoT Factoryn haastetta. Insinööriyön tavoitteena oli suunnitella ja luoda järjestelmä, joka automaattisesti siirtää ja telakoi yhteistyörobotin työskentelypöydästä toiseen.</p> <p>Insinööriyö jaettiin teoreettiseen sekä käytännön osioon. Työn teoriaosuudessa perehdyttiin järjestelmässä käytettyyn tekniikkaan, kilpaileviin vaihtoehtoihin ja kyseisen tekniikan historiaan. Teoreettisessa osiossa syvennyttiin mobiilirobotteihin, yhteistyörobotteihin, ohjelmistorajapintaan, REST-API:n ja telakointijärjestelmään.</p> <p>Työn käytännön osuudessa kehitettiin Reboot IoT Factory:n haasteen ensimmäistä vaihetta ja automatisoitiin sen prosessi. Työssä käytettiin mobiilirobottina MiR-mobiilirobottia ja yhteistyörobottina ABB:n YuMi-kobottia. Palvelin toteutettiin Microsoft Visual Studio- kehitysympäristössä C#-ohjelmointikielellä, jossa käytettiin web-rajapintaa, joka hyödynsi REST-arkkitehtuurimallia.</p> <p>Lopputuloksena saatiin järjestelmä, jolla automaattisesti siirrettiin MiR-mobiilirobotilla YuMi-kobottia yhdeltä työskentelypöydältä toiselle. Toteutuksen mahdollistamiseksi järjestelmälle luotiin palvelin, joka ohjaa ja ajoittaa robottien toiminnan toteutusta ja telakointi operaatioita. Järjestelmää käyttämällä vältetään jättämästä yhteistyörobottia toimettomaksi, kun sen toiminta on päättynyt. Vastaavaa järjestelmää ei ole valmiina saatavilla. Tulevaisuudessa vastaavanlaiset järjestelmät tulevat lisääntymään teollisuudessa huomattavasti.</p>	

Avainsanat	mobiilirobotti, yhteistyörobotti, kobotti, REST-API, robotit, YuMi, MiR,
------------	--

Author Title	Samir Nakroufi Automatic transportation and docking of collaborative robot by mobile robot
Number of Pages Date	52 pages 18 April 2021
Degree	Bachelor
Degree Program	electric and automation engineering
Specialisation option	automation technique
Instructor	Reijo Leinonen, Senior Lecturer
<p>The subject of this thesis was based around the product development process for a company called GE Healthcare. The work was carried out as part of the Reboot IoT Factory challenge. The purpose of this thesis was to design and create a system that would automatically move and dock a collaborative robot from a one workstation to another.</p> <p>The thesis was divided into two components: theoretical and practical. The theoretical aspect of the work was focused on researching the technology used in the system whilst comparing alternatives and the histories of the technologies that were studied. The research specifically delved into mobile robots, collaborative robots, software interfaces, REST APIs, and the docking system.</p> <p>For the practical segments of the thesis, it was concentrated on developing an automated process for the first phase of the Reboot IoT Factory challenge. MiR was used as the mobile robot in collaboration with ABB's YuMi cobot. The server was implemented via the Microsoft Visual Studio using the C# language. This was to ensure the REST architecture model was used within the web interface.</p> <p>In conclusion, the result was a system that automatically moved the YuMi cobot from one workstation to another using the MiR mobile robot. To enable implementation, a server was created for the system to control and schedule the implementation and docking of robotic operations. By using this system, the collaboration robot would not be left inactive after its operational task was complete.</p>	

There are no equivalent systems available currently. However, the potential use of similar systems in the industry sector will increase drastically in the near future.

Keywords	Mobile robot, Collaboration robot, REST-API, Robots, YuMi, MiR,

Sisällys

1	Johdanto	1
	GE Healthcare	1
2	Yhteistyörobotin ja mobiilirobotin toiminnan yhdistäminen	2
	2.1 Projektin taustaa ja vaatimuksia	2
	2.2 Projektin päämäärä	2
	2.3 Järjestelmän kokonaisuus	3
3	Järjestelmässä käytetyt ratkaisut, valinnat ja tekniikan taustaa.	4
	3.1 Robotit	4
	3.1.1 Mobiilirobotti	5
	3.1.2 Yhteistyörobotti	8
	3.2 Ohjelmistoarkkitehtuuri	14
	3.2.1 Asiakas-palvelin järjestelmä	14
	3.2.2 Ohjelmointiympäristö (IDE)	15
	3.2.3 Ohjelmointirajapinta	16
	3.3 Telakointijärjestelmä	18
	3.3.1 RSP (Robot System Products)	18
	3.3.2 ABB, Eden turva-anturi	19
4	Järjestelmän toteutus	20
	4.1 Lähtökohta	20
	4.2 Työskentelypöydän ja kuljetusalustan suunnittelu	21
	4.3 Robotit	22
	4.3.1 MiR-mobiilirobotti	23
	4.3.2 Yumi-kobotti	28
	4.4 Telakointijärjestelmä	30
	4.5 Yhteys	33
	4.6 Palvelin	36
	4.7 Järjestelmässä käytetty REST API	39
	4.7.1 MiR REST-API	39
	4.7.2 YuMi REST API	42
	4.7.3 YuMi:n telakointiohjelma	45
	4.8 Testaus	46
5	Yhteenveto	47

5.1	Lopputuloksen tarkastelu	47
5.2	Haasteet	48
5.3	Luetettavuus ja virheiden arviointi	48
5.4	Jatkokehitys pohdintaa	49
	Lähteet	50

Lyhenteet ja käsitteet

AGV	<i>Autonomous guided vehicles.</i> Vihivaunu.
AMR	<i>Autonomous mobile robot.</i> Itsenäinen mobiilirobotti.
API	<i>Application program interface.</i> Ohjelmointirajapinta.
Kobotti	<i>Collaborative robot.</i> Yhteistyörobotti.
GE	General electric.
GM	<i>General Motors</i>
HRC	<i>Human-robot collaboration.</i> Ihmisen ja robotin yhteistyöskentely.
HTTP	<i>Hypertext Transfer Protocol.</i> Hypertekstin siirtoprotokolla.
IMU	<i>Inertial Measurement Unit,</i> Elektronin laite, joka mittaa ja raportoi rakenteen orientoinnin kohdekoordinaatistossa.
IoT	<i>Internet of things.</i> Esineiden internet.
IP	<i>Internet Protocol address.</i> Internetin protokollaosoite tai yhteiskäyttöosoite
JSON	<i>JavaScript Object Notation.</i> JavaScript-pohjainen objekti esitystiedon välitykseen.
MiR	<i>Mobile industrial robot.</i> Teollinen mobiilirobotti
OSHA	<i>Occupational Safety and Health Administration.</i> Työterveys- ja työturvallisuusvirasto.
REST	<i>Representational State Transfer.</i> HTTP- protokollaan perustuva ohjelmistorajapintojen arkkitehtuurimalli.
ROS	Robot operating system. Ohjelmistojen luominen robottisovelluksia varten.

Solu Yhteistyörobotin työskentelypöytä.

TCP *Transmission Control Protocol*. Tietoliikenneprotokolla tietokoneiden väliseen tiedonsiirtoon.

1 Johdanto

Insinööriyön tavoitteena on suunnitella ja luoda järjestelmä, jossa automatisoidaan yhteistyörobotin siirto solusta toiseen. Järjestelmän tarkoituksena on siirtää yhteistyörobotti solusta toiseen, kun sen työskentely solussa on päättynyt. Solulla tarkoitetaan työskentelypöytää, jossa yhteistyörobotti toimii.

Toteutuksen tarkoituksena on ottaa hyöty yhteistyörobotin työskentelyn päättymisen jälkeisestä hukka-ajasta; yhteistyörobotti muutoin seisoi toimettona. Käyttämällä järjestelmää hyödynnetään kyseinen hukka-aika siirtämällä yhteistyörobotti toiseen soluun työskentelemään.

Insinööriyöprojektin järjestelmän on tarkoitus toimia automaattisesti. Yhteistyörobotin siirtämiseen käytetään mobiilirobotia. Kyseisessä projektissa käytetään yhteistyörobotina ABB:n YuMi-kobottia.

YuMi-kobotille on suunniteltu projektin yhteydessä työskentelypöytä ja kuljetusalusta. Kuljetusalustan päälle sijoitetaan YuMi-kobotti. Sen siirtäminen toteutetaan mobiilirobotilla. Projektissa käytetään mobiilirobotina MiR-mobiilirobotia. Kuljetusalustan ja telakointijärjestelmän avulla MiR-mobiiliroboti kykenee siirtämään YuMi-kobottia paikasta toiseen. Insinööriyöprojektin lopputuotoksena syntyi järjestelmä, joka siirtää automaattisesti YuMi-kobotin solulta toiselle käyttäen MiR-mobiilirobotia.

GE Healthcare

Tämä insinööriyö on kehitystyö, joka toteutetaan yksilötyönä yritykselle GE Healthcare. GE on yhdysvaltalainen monikansallinen yritys. Yrityksen tunnetuimpia työskentelyalueita ovat uudelleensyntyvä energia (renewable energy), energia teknologia (Power), ilmailu (Aviation) ja hyvinvointi (Healthcare).

Projekti toteutetaan GE Healthcaren toimitilassa, jossa valmistetaan potilasmonitoreja. Potilasmonitorien osien rakentamista ja kokoamista pyritään tehostamaan automatisoimalla toimipisteen toiminta. Automatisoinnilla pyritään nopeuttamaan tuotantoa ja pienentää virhemarginaalia monitorien osien rakentamis- ja kokoamisprosessissa. [36.]

Insinööriyö on kehitystyöprojekti, joka on tehty osana Reboot IoT Factory toista haastetta ”Robotic fusion”. Reboot IoT Factory tarkoituksena on edistää suomalaisten yritysten kilpailukykyä globaalilla skaalalla. Siihen ottavat osaa lukuisat yritykset, jotka jakavat tietotaitoaan keskenään toteuttaakseen haastetta. Haasteen tarkoituksena on käyttää hyödyksi robottien yhdistämisen teknologiaa. [35.]

2 Yhteistyörobotin ja mobiilirobotin toiminnan yhdistäminen

2.1 Projektin taustaa ja vaatimuksia

Projektin tavoitteena ja vaatimuksena oli integroida MiR-mobiilirobotti YuMi-kobotin ajoitettuun siirto-, telakoimis- ja irrotusoperaatioon eli luoda järjestelmä, joka automaattisesti siirtää yhteistyörobotin solusta toiseen mobiilirobottia käyttäen.

Insinööriyöprojektia oli edeltänyt haasteen ensimmäinen vaihe, jossa luotiin työskentelypöytä, kuljetusalusta YuMi-kobotille ja pneumaattinen telakointijärjestelmä, joka kiinnittää kuljetusalustan työskentelypöytään kiinni.

YuMi:n kuljetusalustan ja työskentelypöydän solun telakointiasema oli toteutettu haasteen ensimmäisessä vaiheessa pneumaattisella telakointijärjestelmällä ja projektissa päätettiin käyttää samaa telakointijärjestelmää haasteen toisessakin vaiheessa. Telakointijärjestelmää jatkettiin haasteen toisessa vaiheessa mobiilirobottiin ja sille luotiin langaton kommunikaatio palvelimeen.

2.2 Projektin päämäärä

Insinööriyöprojektissa päämääränä on luoda järjestelmä, jossa yhteistyörobottia siirretään automaattisesti työpisteestä toiseen. Siirto suoritetaan käyttäen mobiilirobottia.

Projektin järjestelmää käyttämällä yhteistyörobotin työskentely ei lopu työpisteen tehtävän päätyttyä, vaan se jatkuu automaattisesti toisessa työpisteessä. Ilman järjestelmää yhteistyörobotin työskentely pysähtyy, kun työtehtävät kyseisellä työpisteellä loppuvat. Käyttämällä järjestelmää vältytään jättämästä yhteistyörobottia toimettomaksi ja siirretään sitä sinne, missä sille on tarvetta ja käyttöä.

Projektin järjestelmää ohjaa palvelin, joka mahdollistaa yhteistyörobotin siirron automaattisesti toiselle työpisteelle. Palvelin hallinnoi, ohjaa robotteja ja telakointia. Fyysiset kiinteät osat on suunniteltu yhteensopiviksi järjestelmän kanssa; työskentelypöytä ("solu"), yhteistyörobotin kuljetusalusta, telakointijärjestelmä ja mobiilirobotti on suunniteltu keskenään yhteensopiviksi.

Kuljetusalustassa täytyy huomioida MiR-mobiilirobotin korkeus ja leveys, jotta se mahtuu sijoittumaan kuljetusalustan alle.

Järjestelmässä käytetään pneumaattista telakointijärjestelmää. MiR-mobiilirobotille on rakennettu oma pneumaattinen järjestelmä ja työskentelypöydällä on omansa. MiR-mobiilirobotille oli mietittävä mobiili ratkaisu pneumaattiselle toteutukselle.

2.3 Järjestelmän kokonaisuus

Reboot IoT Factory (Vaihe II, GC2 PoC 2.2 Docking) projektin tuottama järjestelmä, jossa tarkoituksena on siirtää ABB:n YuMi-kobotti solusta toiseen automaattisesti. Järjestelmä (Kuva 1) koostuu YuMi-kobotista, kuljetusalustasta, YuMi-kobotin työskentelypöydästä ("solu"), telakointijärjestelmästä, MiR200-mobiilirobotista ja palvelimesta.

YuMi:n automaattinen siirtojärjestelmä sisältää neljä komponenttia:

- MiR-mobiilirobotti, telakointiasema,
- YuMi-kobotti, kuljetusalusta, telakointiasema
- Työskentelypöytä, telakointiasema, VL-markkeri
- Langaton PC-palvelin, joka ohjaa koko järjestelmää.



Kuva 1. Työpöytä, YuMi, kuljetusalusta ja MiR.

3 Järjestelmässä käytetyt ratkaisut, valinnat ja tekniikan taustaa.

Järjestelmä on joukko elementtejä, jotka liittyvät toisiinsa tai ovat kanssakäymisissä toistensa kanssa noudattaen asetettuja sääntöjä, mitkä yhdistävät niitä toisiinsa [38]. Elementeillä tarkoitetaan kyseisen järjestelmän kohdalla robotteja (YuMi-kobotti ja mobiilirobotti), työskentelypöytää, kuljetusalustaa ja telakointijärjestelmää.

Tässä luvussa käydään läpi projektin järjestelmässä käytettyjen elementtien eri valinta vaihtoehtoja, niiden taustaa ja kilpailevia valintoja.

3.1 Robotit

Määritelmä robotille elää ja muuttuu tutkimusten ja teknologian kehittymisen myötä. Sana ”Robotti” tuli suosituksi tšekkiläisen näytelmäkirjailijan Karel Capek vuoden 1921 näytelmästä ”Rossum’s universal Robots” (R.U.R.). Robot-sana koostuu tšekkiläisistä sanoista ”rabota”, joka tarkoittaa ”pakollinen työ” ja sanasta ”robotnik”, joka tarkoittaa

suomeksi ”maajorja”. Yhden määritelmän mukaan sana robotti tarkoittaa: ”Robotti on itsenäinen systeemi, joka on fyysisesti olemassa ja pystyy aistimaan ympäristöään saavuttaakseen sille määritetyn päämäärän”. [5. s.1–2.]

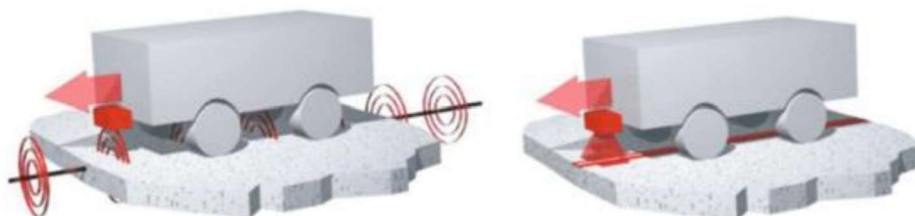
Nykyinen virallinen IFR:n hyväksymä määritelmä robotille on: ”Robotti on automaattisesti ohjattu ja uudelleen ohjelmitava, monikäyttöinen käsittelylaite, jolla on kolme tai useampi uudelleen ohjelmitavia akseleita, joka voidaan sijoittaa kiinteästi paikalleen tai liikkuviksi teollisuuden automaatiosovelluksissa”. Robotissa on kaksi pääosaa: ohjausjärjestelmä ja mekaaninen yksikkö. [4.]

Roboteilla vapautetaan ihmiset toistoa vaativista ja yksinkertaisista työnkuvista, jotta saadaan ihmisiä sijoitettua muihin tehtäviin, jotka ovat ihmisen terveyden kannalta parempia; resursseja vapautuu. Robottien lähtökohtainen tehtävä on auttaa, helpottaa ja tehostaa ihmisten elämää ja työntekoa. Robotteja on monenlaisia, mutta insinööriyöprojektissa käytetään mobiilirobottia (AMR) ja yhteistyörobottia.

3.1.1 Mobiilirobotti

Mobiilirobotti tarkoittaa liikkuva robotiikka. Mobiilirobotteja on kahdenlaisia AMR ”Autonomous mobile robot” ja AGV” Autonomous guided vehicles”. Insinööriyössä käytämme MiR-mobiilirobottia, joka on AMR. Mobiilirobotti voi kulkea sille ohjelmoidulla kartalla pitkin halleja ja käytäviä. [2.]

AGV eli vihivaunu on vanhempaa tekniikkaa harjoittava mobiilirobotti ja pohjautuu fyysisen ohjaukseen ennalta määritetyssä ympäristössä ja ennalta määritetyllä reitillä (Kuva 2). Se sopii toistuvaa toimenpidettä vaativan tehtävän suorittamiseen, kuten linjaa seuraava robotti. Yleensä AGV robotteja suunnitellaan spesifisteihin tehtäviin. [2.]



Kuva 2. AGV:n induktiivinen rataohjaus ja optinen rataohjaus [3. s.110].

AMR on uudempaa ja älykkäämpää tekniikkaa. Sitä käytetään laajasti teollisuudessa ja se alkaa korvaamaan AGV:tä.

Autonomisiin mobiilirobotteihin luetaan lukuisia erilaisia laitteita. Robotti-imurit, itseohjautuvat autot, nämä ja monet muut luetaan mobiilirobotiikkaan. [2.]

Kun sijoitetaan mobiilirobotteja yksinkertaisiin ja toistoa vaativiin siirto- ja kuljetustehtäviin, niin saadaan lisää resursseja ja tehoa yritykselle. Mobiilirobotteja käytetään monilla teollisuuden aloilla. Niiden käyttö ja sovellettavuus tulee lisääntymään tulevaisuudessa. Mobiilirobotteja sovelletaan esimerkiksi metsäteollisuudessa, maanviljelyssä, vesialueiden tutkimuksessa ja muissa toimialoissa. [2.]

AMR "Autonomous mobile robot"

Itsenäinen mobiilirobotti "AMR" on monella tapaa parempi ja päivittyneempi versio AGV-mobiilirobotista. Ero AGV-mobiilirobottiin korostuu etenkin AMR:n kyvyssä navigoida ennakoimattomassa ja osaksi tuntemattomassa ympäristössä. Tämä tarkoittaa sitä, että AMR kykenee arvioimaan ympäristöään ja toimimaan itsenäisesti sille määritetyssä infrastruktuurissa, edistyneiden näkyvyysantureiden ja koneoppimisen avulla. [8.]

Mobiilirobotteja ilmenee yhä useammilla eri sektoreissa (Kuva 3), kuten yrityksissä, sairaaloissa, instituutioissa, maataloudessa ja kodeissa parantamaan palveluita ja palvelemaan arkipäiväisissä aktiviteeteissä.

Teknisen kehityksen ansiosta mobiilirobottien suorittamien tehtävien ja palveluiden määrä on kasvanut huomasti: raskaiden esineiden kanto, tarkkailu, etsintä ja pelastus tehtävät.

AMR:n liikkeissä käytetään joko vähän ihmisen väliintuloa tai ei ollenkaan. Se on suunniteltu seuraamaan ennalta määritettyä reittiä, oli se sitten ulkona tai sisätiloissa. Sisätiloissa navigoidakseen AMR tarvitsee pohjapiirroksen kerroksesta, kaikuluotaimen ja IMU:n (Inertial Measurement Unit). [9.]



Kuva 3. Mobiilirobotin sovellukset [9.]

OMRON LD- Mobiilirobotti

OMRON LD on japanilaisen Omron yhtiön valmistava AMR. Mobiilirobotiikan lisäksi Omronilla on teollisuudessa muitakin tuotteita, kuten robotiikkaa, antureita, ohjauksjärjestelmiä sekä turvatuotteita.

Omronin LD-mobiilirobotti mallit (Kuva 4) pystyvät kantamaan seuraavanlaisia kuormia: LD-60 60 kg, LD-90 90 kg ja LD-250 250 kg. LD mallit on suunniteltu toimimaan itsenäisesti ihmisten parissa ja sisätiloissa. Omronin LD-mallien akkukapasiteetti on noin 15 tuntia täyteen ladattuna.



Kuva 4. Omronin valikoima mobiiliroboteista [10.]

MiR-mobiilirobotti

MiR-mobiilirobotti (Mobile industrial Robot) on suhteellisen nuori yritys, joka perustettiin vuonna 2011. Ensimmäisen mobiilirobotin (MiR100) yritys myi vuonna 2013. MiR yrityksenä on keskittynyt mobiilirobotiikkaan ja sen sovelluksiin. [40.]

MiR-mobiilirobotti malleja on muutamia: MiR100, MiR200, MiR250, MiR500 ja MiR1000. Erona mallien välillä on kantokapasiteetti. MiR100 pystyy kantamaan 100 kg, MiR200 200 kg, MiR250 250 kg, MiR500 500 kg ja MiR1000 1000 kg. MiR-mobiilirobotin akkukapasiteetti on noin 10 tuntia täyteen ladattuna. [41.]

MiR:llä on lisälaitteita tukemaan mobiilirobotiikan toimintoja. Yksi lisälaitteista on MiR Fleet, jolla voidaan ohjata yhdestä paikasta käsin MiR:n robotteja. MiR Fleet toimii kaikissa laitteissa, jossa on internet-selain. [42.]

3.1.2 Yhteistyörobotti

Yhteistyörobotti eli kobotti on suunniteltu toimimaan samassa työtilassa ihmisten kanssa. Teollisuuden robotit ovat yleensä eristetty ihmisten läheisyydestä, jotta välttäisiin tapaturmilta.

Professorien Michael Peshkin ja J.Edward Colgate tutkimustyö johti kobottien keksimiseen, jolloin se patentoitiin ensimmäisen kerran vuonna 1999.

Myöhemmin Colgaten and Peshkinin nosto systeemiin lisättiin antureita havainnoimaan siihen kohtaan, jossa käyttäjä haluaa siirtää kuorman, jonka jälkeen järjestelmä ottaa haltuun nostamisoperaation. Antureilla oli merkittävä rooli, kun tehtiin laitteista älykkämpiä ja yhteistyökelpoisempia.



Kuva 5. Linjastotyöntekijä ja kobotti vierekkäin työskentelemässä linjastossa. [13.]

Teknologian kehityksen edistyminen on luonut kasvavan potentiaalin turvallisesti yhdistää sekä robotin voiman ja tarkkuuden että ihmisen luovuuden ja ongelmanratkaisukykyyn. [11.]

Termin yhteistyörobotti voi helposti väärinymmärtää. Yhteistyörobotti termin väärinymmärrys ei johdu robotin tarkoituksesta tai sen käytöstä, vaan sen toiminnallisuudesta ja miten se toimii. Se pelkää, että robotti toimii ihmisten läheisyydessä ilman turvaita, ei tee siitä yhteistyörobottia. [12]

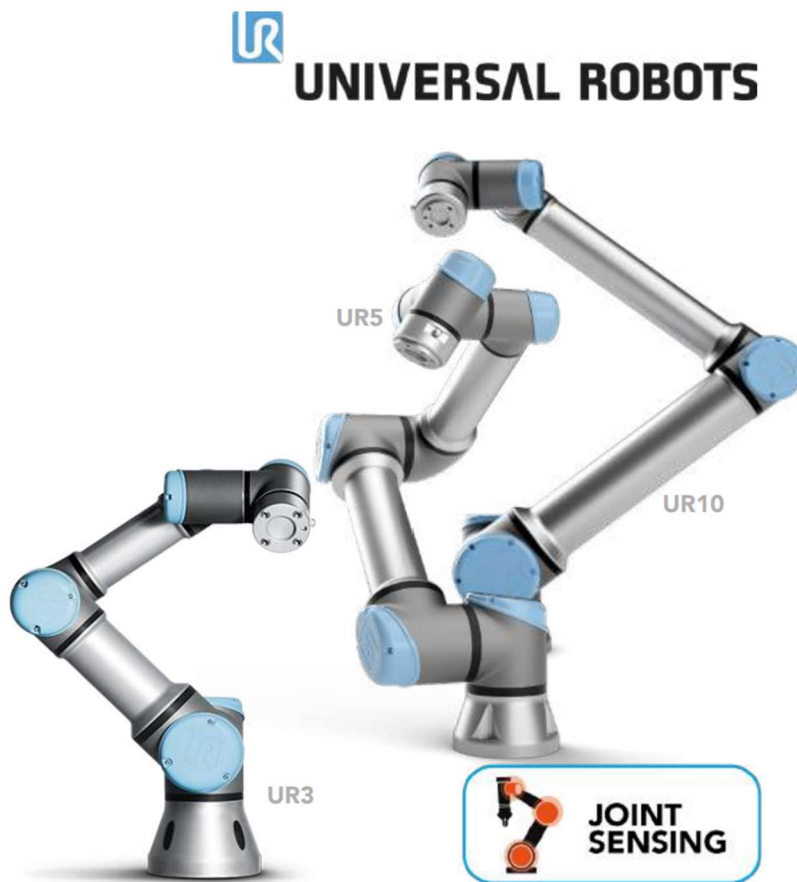
On olemassa monenlaisia yhteistyörobotteja, mutta vain yhtä tyyppiä voidaan käyttää ilman lisättyjä suojoimia. ISO 10218 osan 1 ja osan 2 mukaan robotti omaa neljä tekniikkaa, joita käytetään robotin ja ihmisen yhteistyön toteuttamiseksi: Turvaluokiteltu valvottu pysäytys, käsin ohjaaminen, nopeuden ja vähimmäisetäisyyden valvonta, tehon ja voiman rajoittaminen luontaisesti turvallisella tavalla tai ohjauksella. [12]

Teollisuusrobotin ja yhteistyörobotin erona on se, että teollisuusrobotit ovat isoja, vahvoja ja kestäviä, ja ne on suunniteltu spesifisteihin tarkoituksiin palvelemaan tiettyä tarkoitusta. Teollisuusrobotit ovat suojattu aidoilla, suojakaiteilla ja antureilla, jotta työskentely niiden kanssa olisi turvallista. [15]

Universal Robots, UR

Esben Østergaard, Kasper Støy, ja Kristian Kassow perustivat Universal Robots-yhtiön vuonna 2005. Heidän tavoitteenaan oli mahdollistaa yhteistyörobottien saanti pienille- ja keskikokoisille yrityksille.

Ensimmäisen kobotin UR5:n he myivät vuonna 2008, minkä koettiin olevan yksi robotiikka yhdistyksen merkittävimmistä teknologisista saavutuksista kymmeneen vuoteen. Universal Robots mallien (Kuva 6) kantokapasiteetti ilmenee robottien mallien numeroinnista seuraavasti: UR3 3 kg, UR5 5 kg, UR10 10 kg ja UR16 16 kg. [14.]



Kuva 6. Universal Robots kobotit UR3, UR5, UR10. [15.]

APPLICATIONS



finishing



packaging



machine
tending



assembly



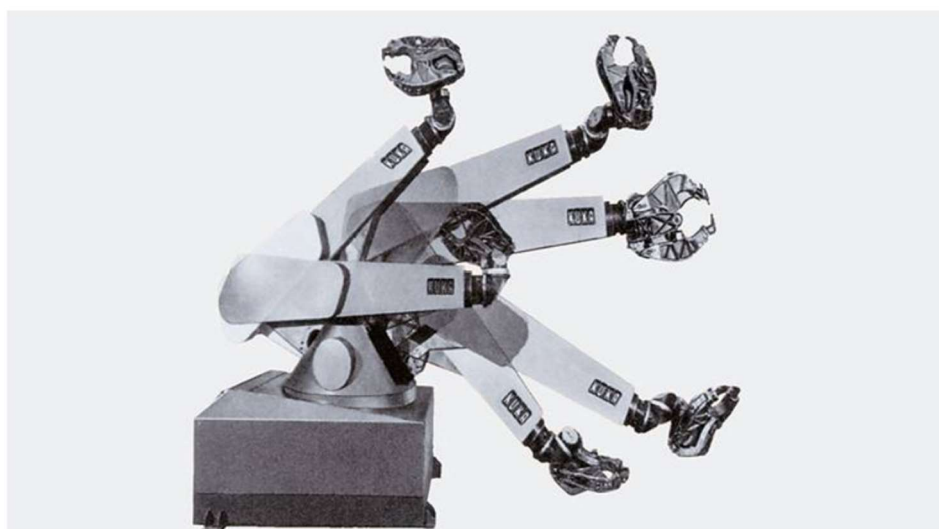
dispensing

Kuva 7. Universal Robots kobottien sovellettavuus [15.]

KUKA, LBR iiwa

Johann Joseph Keller ja Jakob Knappich perustivat KUKA:n Saksassa vuonna 1898. Yritys aloitti keksimällä asetyleenikaasulaitoksen, joka mahdollisti kustannustehokkaan kotitalouksien ja katujen valaistuksen. Pian tämän jälkeen yritys laajensi toimintaansa myös muilla toimialoilla.

Vuonna 1971 KUKA otti merkittävän askeleen robotiikassa ja rakennutti Euroopan ensimmäisen robottiohjatun hitsauskokoontuolinjan Daimler-Benzille. Yritys kehitti maailman ensimmäisen teollisuusrobotin vuonna 1973 (Kuva 8). Yritys oli myös ensimmäinen robottivalmistaja maailmassa, joka otti riskin siirtyä tietokonepohjaiseen ohjaukseen robotissa. [16.]



Kuva 8. Maailman ensimmäinen teollisuus robotti FAMULUS. [16.]

KUKA valmisti ensimmäisen yhteistyörobottinsa vuonna 2013. Se oli ensimmäinen ”aistiva” (sensitive) robotti, jonka hyväksyi HRC (Human-Robot Collaboration).

LBR iiwa on KUKA:n valmistama yhteistyörobotti sarja (Kuva 9). Sarjaan kuuluu seuraavat mallit ja niiden kantokapasiteetit: iiwa7 7 kg ja iiwa14 14 kg. [16.]



Kuva 9. LBR iiwa7 ja -8. [15.]

APPLICATIONS



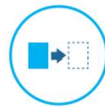
quality
testing



assembly



palletizing



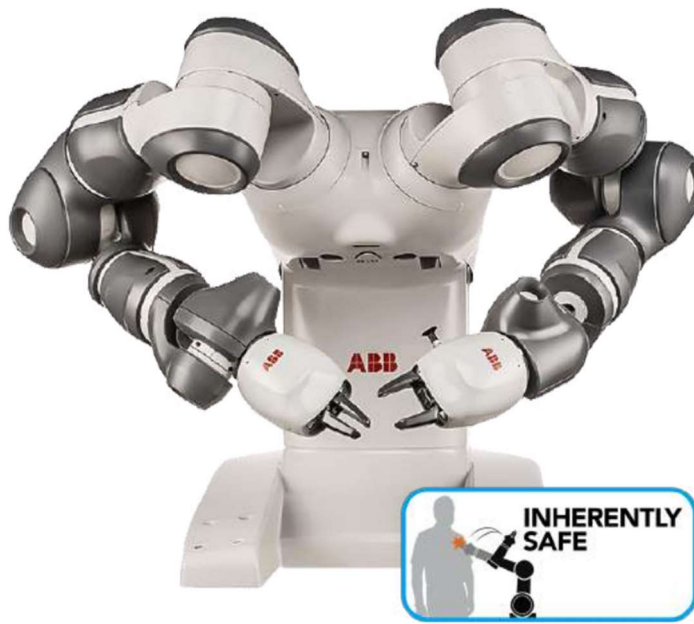
pick & place

Kuva 10. KUKA kobottien sovellettavuus [15.]

ABB, YuMi

ABB on kahden Euroopan tunnetuimpiin sähkötekniikkaa harjoittavan yrityksen yhdistymisen tuotos: ASEA:n ja BBC:n (Brown Boveri & Cie). ABB perustettiin vuonna 1988, se aikaisemmin tunnettiin nimellä ACEA Brown Boveri. Yrityksen päätoiminta alueet ovat sähkötuotanto, siirto ja jakelu. Sähköinen kuljetus. Teollisuusautomaatio ja robotiikka.

Vuonna 2015 ABB esitteli maailman ensimmäisen todellisen yhteistyörobotin YuMi:n. YuMi:sta teki erikoisen se, että se pystyi tekemään samaa tehtävää ihmisen kanssa ollessa sen vieressä. [17.]



Kuva 11. IRB 14000 YuMi. [15.]

ABB:n kobottimalleja ovat YuMi- IRB 1400, YuMi IRB 14050, GoFa- CRB 15000 ja SWIFTI- CRB 1100. YuMi- IRB 1400 on kahdella kädellä toimiva kobotti (Kuva 11) ja YuMi IRB 14050 yhden käden malli. [19.]

YuMi- mallit on tehty käsittelemään pieniä osia ja se on todella tarkka ja ketterä toiminoissaan. Sen kantokapasiteetti on 0,5 kg.

SWIFTI-malli tarjoaa teollisuustason tehokkuuden ja yhteistyörobotin turvallisuuden. Se on ominaisuuksiltaan nopeasti ja tarkasti toimiva malli, jonka kantokapasiteetti on 4 kg. GoFa- malli on nopein yhteistyörobotti omassa luokassaan. Se on ominaisuuksiltaan pitkälle ulottuva, malli omaa edistyneen ja laajan turvallisuuden. [18.]

Insinööriyön järjestelmässä päädyttiin käyttämään yrityksen puolesta YuMi- IRB1400-mallia. Valinnan perusteena toimi se, että kyseistä ABB YuMi malleja oli jo valmiiksi käytössä yrityksen toimitiloissa, johon insinööriyöprojekti tehtiin.

3.2 Ohjelmistoarkkitehtuuri

Ohjelmistoarkkitehtuurille löytyy monenlaista määritelmää. IEEE:n määritelmä termille on, että se on mikä tahansa järjestelmä, jossa ohjelmistolla on olennainen vaikutus suunnitteluun, järjestelmän rakentamiseen, käyttöönottoon ja ohjelman kehitykseen kokonaisuutena. [39.]

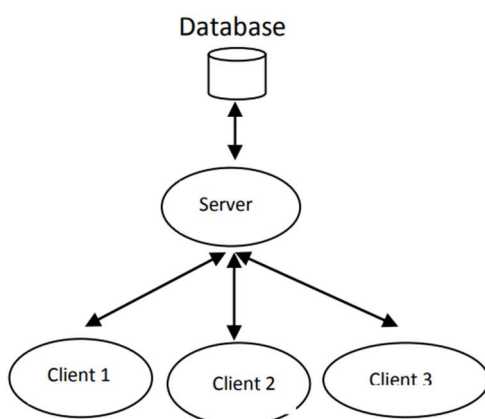
Tietotekniikan yhteydessä palvelimella (server) tarkoitetaan palvelinohjelmaa ja tietokonea, jossa kyseinen ohjelma pyörii. Palvelimen tarkoituksena on tarjota palveluita muille ohjelmille tai laitteille.

3.2.1 Asiakas-palvelin järjestelmä

Asiakas-palvelinarkkitehtuurilla tarkoitetaan sitä, että palvelin jakaa palveluita, kuten datan välittämistä asiakkaille eli muille ohjelmille tai laitteille. Asiakkaalla tarkoitetaan muita ohjelmia tai laitteita.[25.]

Asiakas-palvelinjärjestelmä on tullut suosituksi, koska sitä käytetään päivittäin erilaisiin sovelluksiin. Osa standardisoiduista protokollista, joita asiakas-palvelin järjestelmä käyttää ovat File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP) ja Hypertext Transfer Protocol (HTTP). Täten asiakas-palvelin järjestelmää voidaan määrittellä ohjelmistoarkkitehtuuriksi, joka sisältää molemmat, asiakkaan sekä palvelimen.

Asiakas-palvelinjärjestelmä tarjoaa prosessien välisen kommunikoinnin (Kuva 12). Siihen liittyy asiakkaan ja palvelimen välinen datan vaihto, jossa kumpikin suorittaa eri toimintoja. [26.]



Kuva 12. Asiakas-palvelinmalli [26]

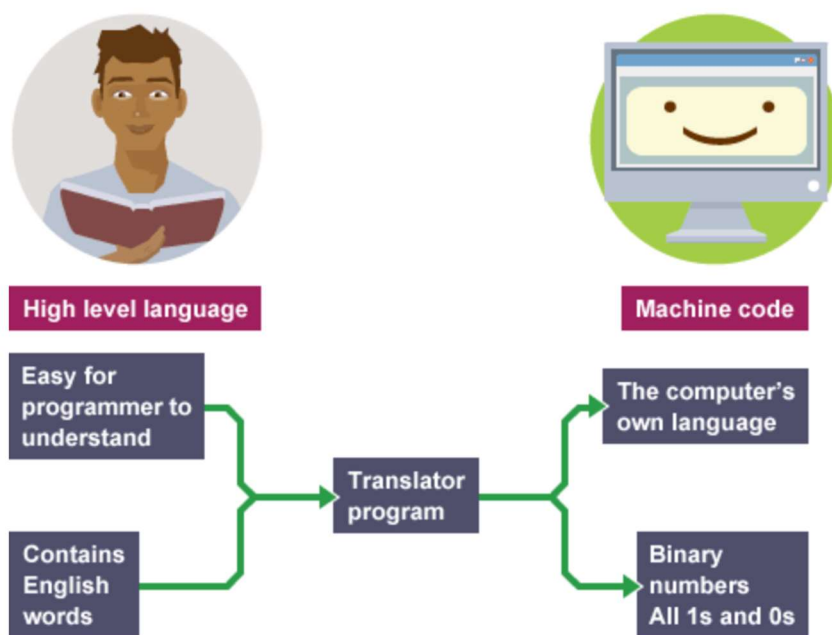
3.2.2 Ohjelmointiympäristö (IDE)

Ohjelmointiympäristö on ohjelma, joka helpottaa ohjelmien kehittämistä. Se on suunniteltu sisältämään kaikki ohjelmoinnin tehtävät ja yhdistämään ne yhdeksi ohjelmaksi: koodieditori (code editor), kääntäjä (compiler), virheen korjaaja (debugger) ja rakennetut automaatiotyökalut (build automation tools). [28.]

Ohjelmointikielien on jaettu kahteen ryhmään, korkean tason kielet (High level languages) ja matalan tason kielet (Low level languages) (Kuva 13). Matalan tason kielet kuvastavat paremmin CPU:n (Central prosessin unit; tietokoneen aivot) mekaanista toimintaa.

Korkean tason kieliin lukeutuvat Java, Javascript, C++, RUBY ja Python. Matalan tason kieliin kuuluu C, assembly ja konekielen (machine code).

Tietokoneen CPU suorittaa ryhmän binaarilukuja, eli kaikki ohjelmointikielien käännetään binaarikoodiksi. Alemman tason kieliä prosessoidaan nopeammin kuin korkean tason kieliä, mutta niitä on ihmisen vaikeampi kirjoittaa ja lukea. [27]



Kuva 13. Kääntäjä kääntää (compiler) ohjelmointikielen tietokoneelle ymmärrettävään muotoon. [27]

Ohjelmointiympäristöjä on monia, jotka tarjoavat kehittäjille monta erilaista tapaa työskennellä ja tuottaa erityyppisiä koodeja. On ohjelmointiympäristöjä, jotka on suunniteltu

toimimaan yhdellä ohjelmointi- kielellä, pilvi pohjaisia ohjelmistoympäristöjä, mobiili sovelluksille räätälöityjä ohjelmistoympäristöjä tai HTML:lle, ja ohjelmistoympäristöjä, jotka on tehty Apple alustalle tai Microsoft alustalle. [28.]

Eclipse, IntelliJ IDEA ja Visual Studio ovat vaihtoehtoisia valintoja ohjelmoijalle aloittamaan tekemään ohjelman kehitystyötä. Eclipse ja IntelliJ IDEA ovat ilmaisia ohjelmia. Visual Studio on maksullinen. Visual Studiolla on laajempi valikoima eri projektimalleja ja avustavia työkaluja. [29; 30; 31.]

3.2.3 Ohjelmointirajapinta

Ohjelmistorajapinta eli API (Application Programming Interface) on joukko protokollia ja määritelmiä sovellusohjelmien rakentamiseen ja integrointiin. API antaa tuotteen tai palvelun kommunikoida muiden tuotteiden ja palveluiden kanssa. Tämä yksinkertaistaa ohjelmien kehitystä, jolla säästetään ohjelmistokehittäjien aikaa, joka puolestaan säästää rahaa. [32.]

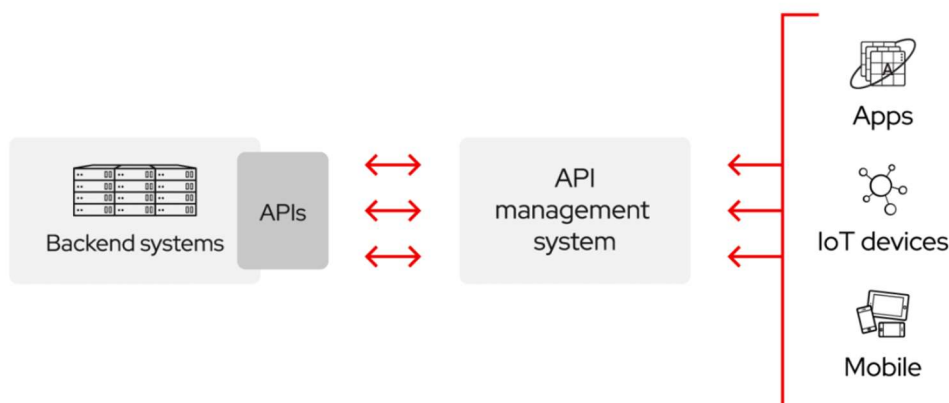
Käyttöjärjestelmällä tasolla toimivat API:t auttavat ohjelmia kommunikoimaan alemmalla olevien kerroksien kanssa seuraamalla rajapinnassa määritettyjä protokollia ja käyttämällä rajapinnan spesifikaatioita (Kuva 14).

Ohjelmistokirjastoilla on huomattava rooli järjestelmien yhteensopivuuksien kannalta. API:t määrittävät miten ohjelman kirjastot käyttäytyvät. Applikaation, jotka ovat vuorovaikutuksessa kirjaston kanssa, täytyy noudattaa API:n määrittämiä sääntöjä. Tämä lähestymistapa helpottaa ohjelmistokehittäjiä kirjoittamaan applikaatioita, jotka ovat vuorovaikutuksessa muiden kirjastojen kanssa, kunhan kaikki kirjastot noudattavat samaa API:a. Toinen hyöty tässä metodissa on, että voidaan käyttää samaa ohjelmistokirjastoa eri ohjelmistokielillä, kunhan kirjastoa lataava applikaatio on yhteensopiva API:n kanssa. [34.]

API:t avaa pääsyn resursseihin ja samalla ylläpitää turvallisuuden ja kontrollin. Se, kelle annetaan pääsy ja miten, on ohjelmoijan päätettävissä. On kolme lähestymistapaa API:n julkaisukäytännöissä:

- *Yksityinen*: API on vain sisäiseen käyttöön. Tämä antaa yrityksille suurimman osan kontrollista heidän API:in

- *Pari*: API on jaettu tietyn liikekumppanin kanssa. Tämä voi tarjota lisää tulovirtoja laadusta tinkimättä.
- *Julkinen*: API on saatavilla kaikille. Tämä antaa kolmansien osapuolien kehittää applikaatioita, jotka ovat vuorovaikutuksessa julkisen API:n kanssa. [32.]



Kuva 14. API:n toimintamalli [32.]

REST API

REST (representational state transfer) on yksi ohjelmistorajapinnan arkkitehtuurimalli, joka perustuu HTTP-protokollaan (Kuva 15). Se on standardi tapa resurssien saamiseen internetistä. Tämä prosessi vaatii kaksi osapuolta: asiakkaan (client) ja resurssi.

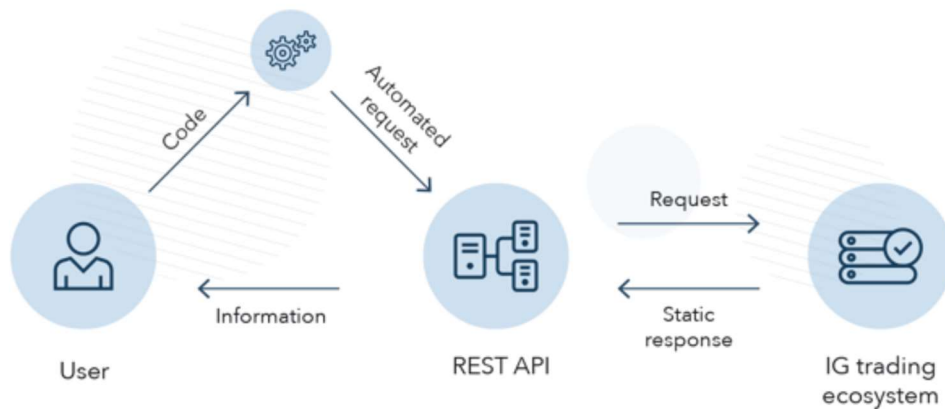
REST API vaatii automatisoidun pyynnön, joka on kirjoitettu koodiin, käynnistääkseen automaattisen staattisen vastauksen resurssilta. REST API saa vain kuvankaappauksen tiedoista sillä hetkellä, joten sillä on tapana kerätä erityisiä tietoja ja tehdä erilaisia toimintoja, kuten pääsy käyttötilin historiaan. REST API vaatii koodauksen ja ohjelmistokehityksen ymmärrystä.

On neljä operaatio luokitusta. HTTP-metodit ovat toimintoja, joita suoritetaan API:lla ("Application programming interface"):

- GET Mahdollistaa resurssien vastaanottamisen. Noudetaan tietoa verkkopalvelimesta

- POST mahdollistaa resurssien luomisen.
- PUT mahdollistaa resurssien korvaamisen.
- DELETE mahdollistaa resurssien poiston.

[33.]



Kuva 15. REST API-toimintamalli [33.]

3.3 Telakointijärjestelmä

Mobiilirobotille ei löytynyt valmista telakointijärjestelmää, joka telakoisi robotin ja kuljetusalustan, kuljetusalustan ja työskentelypöydän.

Insinööriyössä on sovellettu pneumatiikalla toimivaa RSP:n Tool changeria, jota käytetään järjestelmässä kiinnitystä ja telakointia vaativissa tehtävissä.

3.3.1 RSP (Robot System Products)

RSP osti itsensä ulos ABB Roboticsista vuonna 2003 ja julistautui omaksi erilliseksi yritykseksi. Yritys pohjautuu robotiikan oheislaitteiden tutkimukseen ja kehitykseen.

Yrityksen päämääränä on ratkaista teollisessa robotiikassa ilmeneviä haasteita.

RSP:llä on laaja valikoima oheislaitteita, joita ovat työkaluvaihtaja (tool changer, Kuva 16), tarttuja (gripper) ja monia muita. [20]



Kuva 16. Tool changer TC180-8E ja tool attachment TA180-8E [21]

3.3.2 ABB, Eden turva-anturi

ABB:n Eden on kosketukseton turva-anturi (Kuva 17), jota käytetään mm. lukitsemaan ovia ja turvallisen asennon valvonnassa. Eden sisältää kaksi osaa: Adam ja Eva. Adam tunnistaa Evan ilman minkäänlaista mekaanista kosketusta. [24.]



Kuva 17. Eden [24]

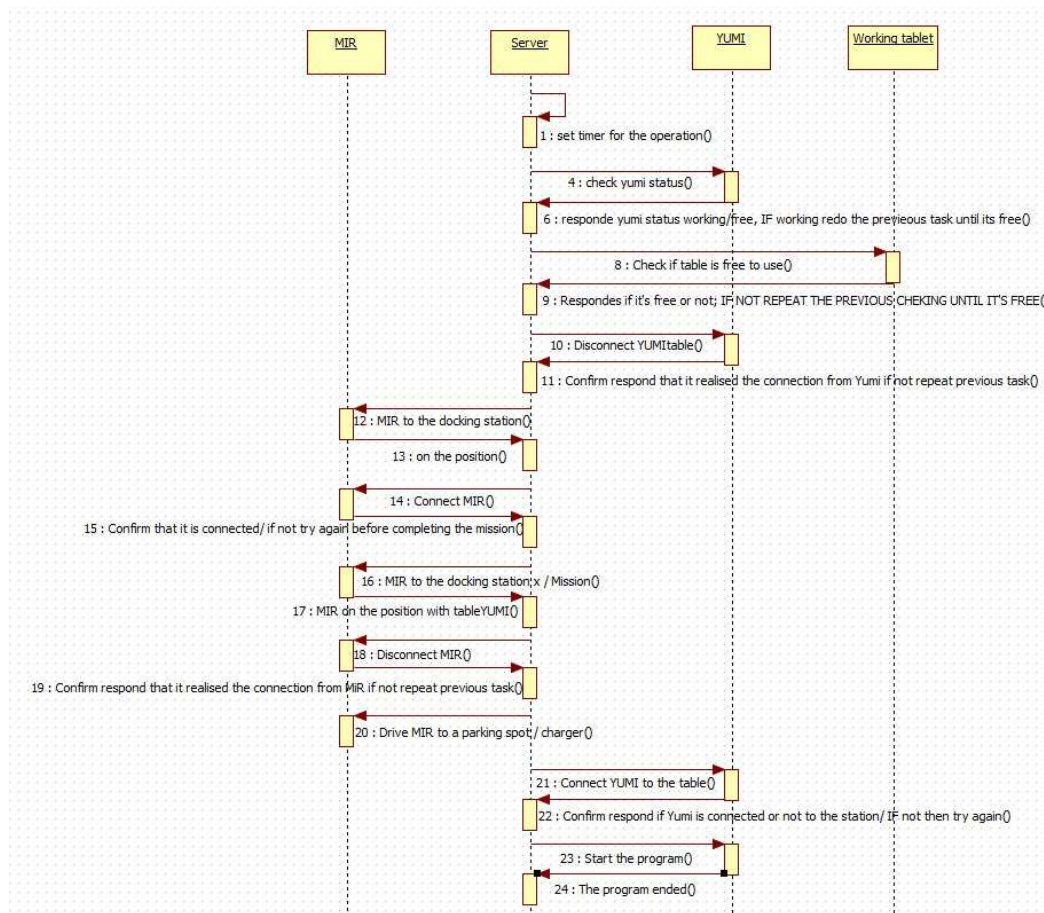
4 Järjestelmän toteutus

4.1 Lähtökohta

Projekti on haasteen Reboot IoT Factoryn (Vaihe II, Grand Challenge 2) toinen vaihe. Reboot IoT Factoryn (Vaihe I, Grand Challenge 2) ensimmäisessä vaiheessa toteutettiin YuMi-kobotin siirtäminen ihmisvoimin työskentelypöydältä toiselle sekä manuaalisestitelakoinnin toteutus ja irrotus.

Reboot IoT Factoryn toisessa vaiheessa luotiin järjestelmä, joka mahdollistaa YuMi-kobotin automaattisen siirtämisen työskentelypöydältä toiseen ja automaattisen telakoinnin työskentelypöytään.

Valmista järjestelmää ei ollut saatavilla, joten se täytyi rakentaa tyhjästä. Järjestelmälle luotiin palvelin, joka hallinnoi operaatioiden toteutuksista. Palvelimen käyttöliittymästä ajoitetaan haluttujen operaatioiden toteutukset.



Kuva 18. Alustavan suunnitelman hahmottelu

4.2 Työskentelypöydän ja kuljetusalustan suunnittelu

Järjestelmän Reboot IoT Factoryn (Vaihe I, Grand Challenge 2) ensimmäisessä vaiheessa YuMi-kobotille suunniteltiin ja rakennettiin sekä työskentelypöytä, että kuljetusalusta.

Haasteen toisessa vaiheessa samoja rakennelmia ei voinut käyttää, koska MiR-mobiiliroboti ei ollut yhteensopiva kuljetusalustan kanssa ja ei mahtunut ajamaan sen alle. Se, että kuljetusalustan ulottuvuuksia piti muuttaa sopimaan MiR-mobiilirobotin kanssa, niin täytyi myös työskentelypöydän ulottuvuuksia muuttaa sopimaan kuljetusalustalle tehtyjen muutosten kanssa.

Työskentelypöytä ja kuljetusalusta suunniteltiin uusiksi vastaamaan tulevan järjestelmän tarpeita ja ulottuvuuksia (Kuva 19). Suunnitteluohjelmalla käytettiin CREOa ja suunnittelupohjana haasteen ensimmäisen vaiheen rakennelmia, joita muokattiin sopimaan uuden järjestelmän kanssa.



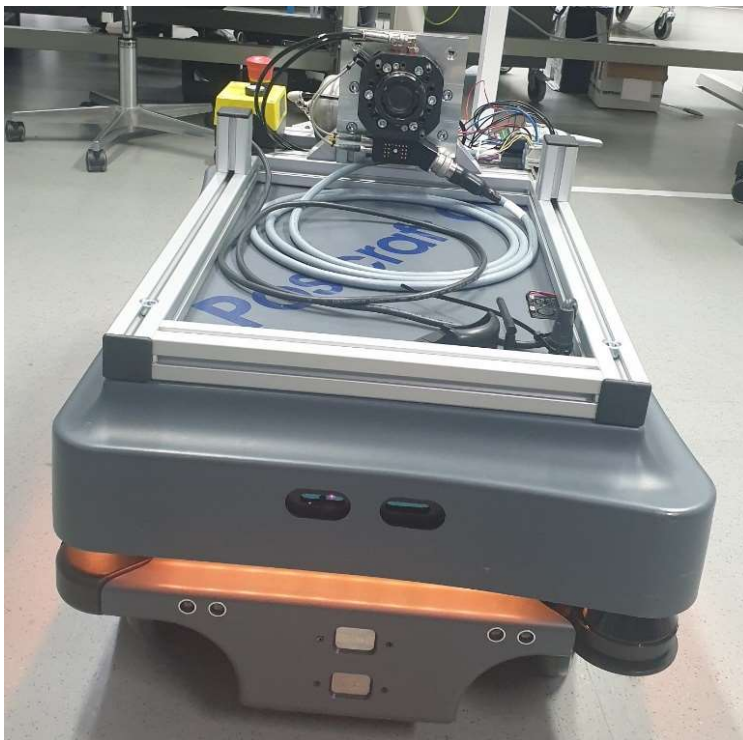
Kuva 19. Työskentelypöytä ja kuljetusalusta.

4.3 Robotit

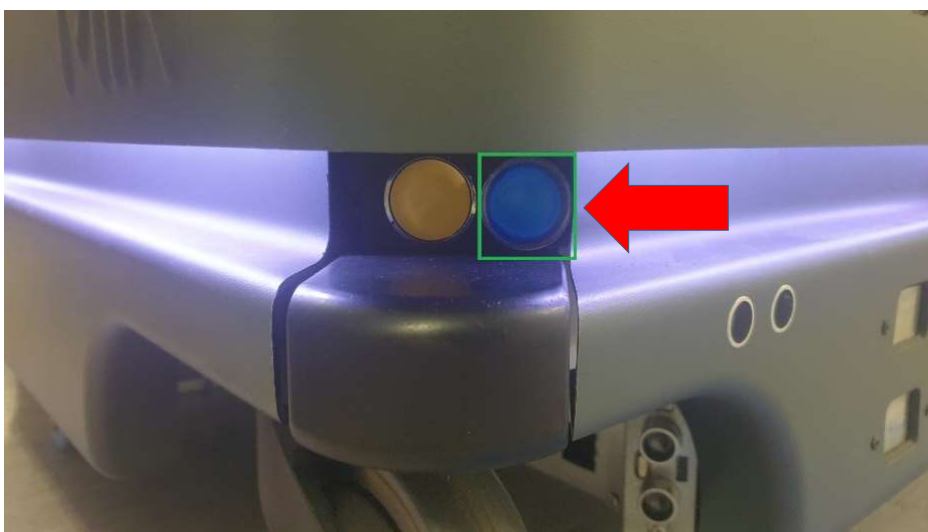
Järjestelmässä käytetään kahta robottia. Yhteistyörobottina ABBn YuMi-kobottia ja mobiilirobottina MiR-mobiilirobottia. Palvelin ohjaa robottien toimintaa, lähettää käskyjä roboteille ja vastaanottaa niiltä dataa.

4.3.1 MiR-mobiilirobotti

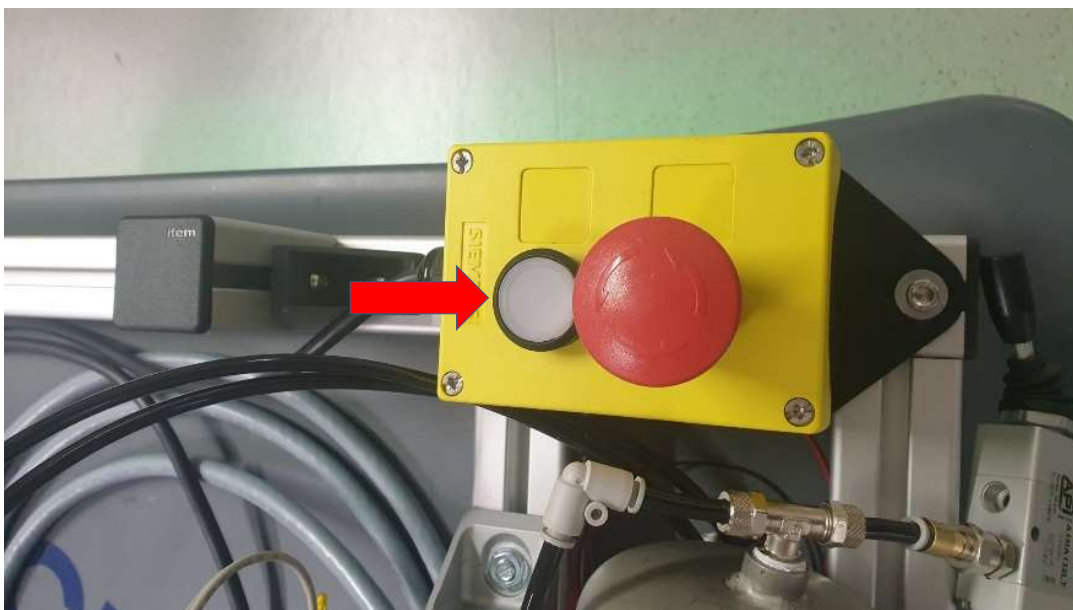
MiR-mobiilirobotin päälle on asennettu mobiilinen telakointijärjestelmä ja Siemensin hätä-seis/ kuittaus painike (Kuva 22). Mobiiliseen telakointijärjestelmään kuuluu telakointiaseman lukkiutuva puoli (Kuva 23, "E") ja sen vastakappale on kuljetusalustassa (Kuva 28, "D"), Painesäiliö (Kuva 33) ja Wise 4060 rele (Kuva 23).



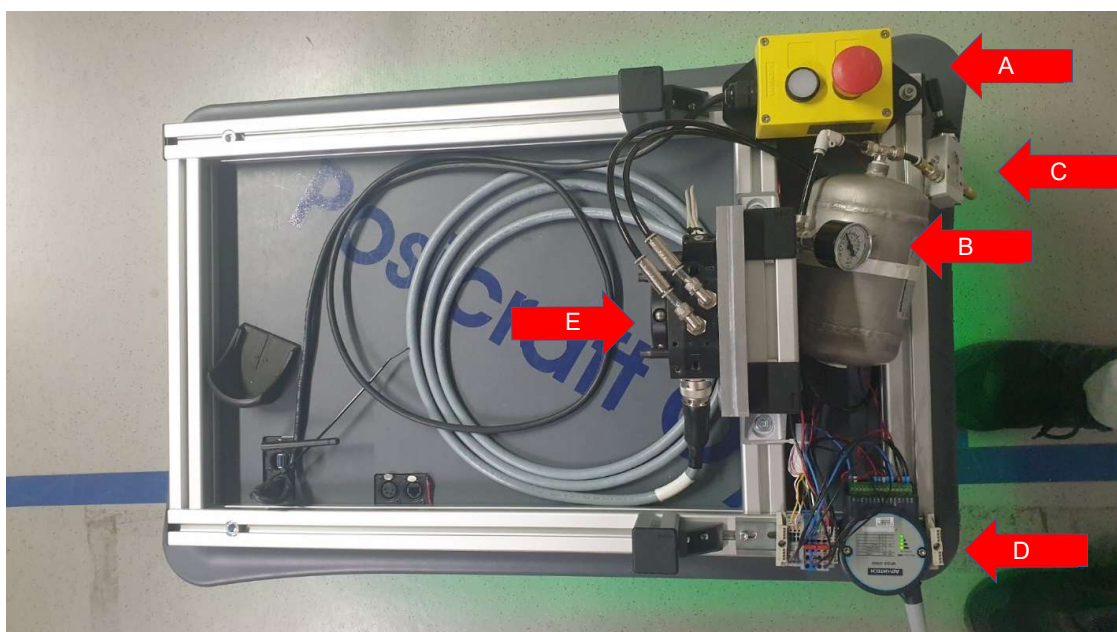
Kuva 20. MiR-mobiilirobotti ja telakointiasema.



Kuva 21. MiR-mobiilirobotin päävirtakytkin.



Kuva 22. Hätäseis-painike ja kuittausnappula.



Kuva 23. MiR-mobiilirobotti ylhäältä: (A) hätäseis-painike, (B) painesäiliö ja säiliönmittari, (C) painesäiliön täyttöventtiili, (D) WISE-4060-rele, (E) telakointiaseman lukitusosa.

MiR:n käyttöjärjestelmästä pystyy määrittämään robotin asetukset, kartat, tehtävät ja robotin ulottuvuudet. MiR-mobiilirobotin tehtävät määritetään MiR-mobiilirobotin hallinnointiohjelmistolla ja palvelin käyttää kyseisiä tehtäviä suorittaakseen operaatioita.

Järjestelmässä on käytetty yhden operaation suorittamiseen kahta erillistä tehtävää: Ensimmäisessä tehtävässä (Kuva 25) MiR-mobiilirobotti ajaa työskentelypöydälle, telakoi

itsensä YuMi- kobotin kuljetusalustaan kiinni ja siirtää kuljetusalustan "Home Point"-pisteeseen (Kuva 24, "C"). Toisessa tehtävässä (Kuva 26) MiR-mobiilirobotti siirtää YuMi-kobotin kuljetusalustan työskentelypöydälle, irrottaa telakoinnin ja siirtyy "Home Point"-pisteeseen (Kuva 24, "C") odottamaan uutta operaatiota.

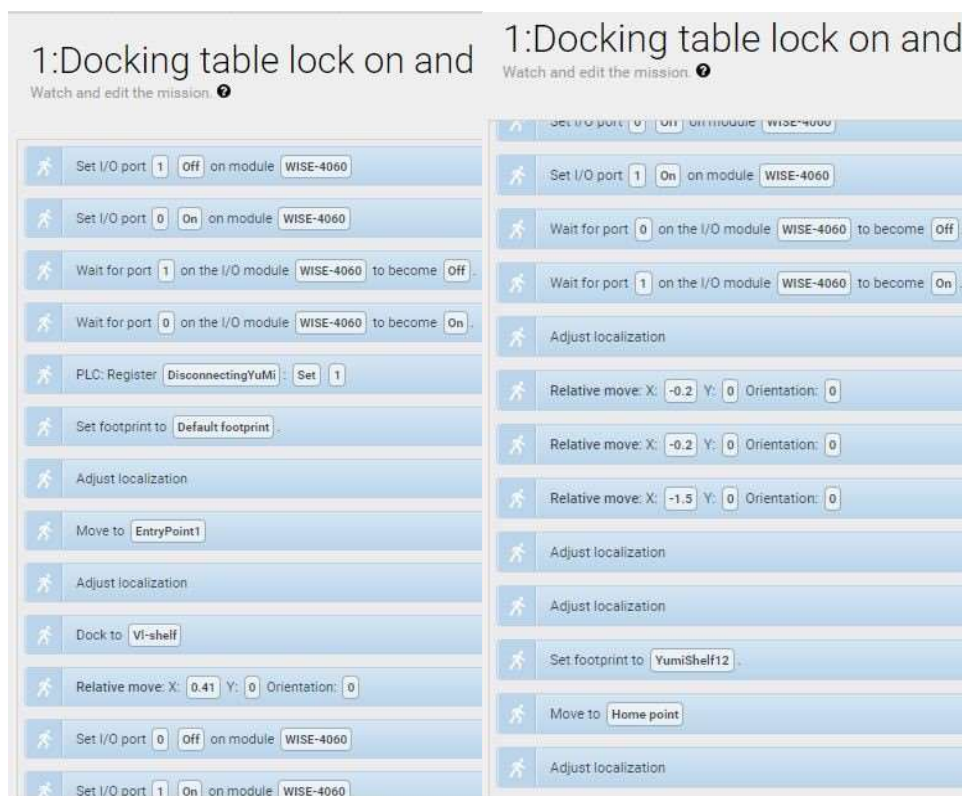
MiR- mobiilirobotti käyttää apunaan VL-markeria (Kuva 24, "A" ja Kuva 27) sijoittautumiseen työskentelypöydän ja kuljetusalustan alle. VL-markeri asetetaan kohtaan, jonka eteen MiR-mobiilirobotti halutaan ajavan. VL-markeria käytetään silloin, kun halutaan MiR-mobiilirobotin sijoittuvan tarkasti.



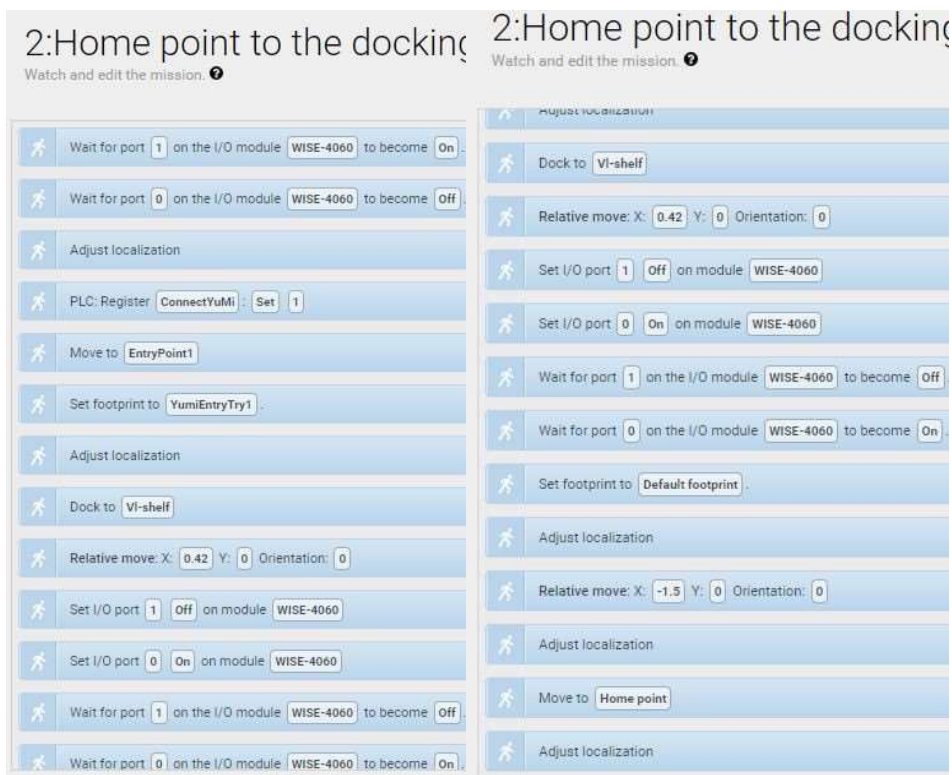
Kuva 24. MiR-mobiilirobotin kartta, jota se käyttää.

Taulukko 1. Selityksiä kartassa (Kuva 24) oleville nuolille.

Kirjaimet	Selitys
A	VL-marker. VL-marker sijoitetaan sijoitetaan siihen kohtaan missä halutaan, että MiR asetetaan kuljetusalustan alle (Kuva 1 ja Kuva 24).
B	EntryPoint1. Piste johon MiR voi kutsua.
C	Home Point. Piste johon MiR voi kutsua.
D	MiR-mobiilirobotti
E	Punaiset linjat kartalla osoittaa mitä MiR näkee, ja se että punaiset linjat ovat linjassa kartassa näkyvien seinien ja muiden mustalla merkittyjen rajojen kanssa tarkoittaa, että MiR:n sijainnin säätö on kohillaan.
Vihreä ja punainen alue	Punainen alue tarkoittaa että se on kiellettyä aluetta MiR-robotille ja vihreä että sitä aluetta suositetaan MiR-robotin käyttävän.



Kuva 25. MiR-mobiilirobotin tehtävä, jolla haetaan YuMi:n kuljetusalusta työskentelypöydältä.



Kuva 26. MiR-mobiilirobotin tehtävä, jolla viedään YuMi:n kuljetusalusta työskentelypöydälle.

Selityksiä MiR-tehtävissä käytetyistä elementeistä (Kuva 25 ja Kuva 26)

Nimike	Selitys
<i>Set I/O port</i>	<i>Asettaa MiR:n I/O arvon 1 tai 0</i>
<i>Wait for port</i>	<i>Odottaa, että MiR:n I/O arvo asettuu 1 tai 0</i>
<i>PLC register</i>	<i>PLC rekisteri, jota voidaan kutsua palvelimiestä, jotta osataan asettaa YuMi:n I/O arvot vastaamaan haluttua tilannetta.</i>
<i>Set a footprint</i>	<i>Määrittää miten MiR näkee omat ulottovuudet</i>
<i>Adjust localization</i>	<i>Tarkistaa ja tarkentaa omaa sijaintiaan kartalla.</i>
<i>Move to</i>	<i>Siirtyy valittuun pisteeseen</i>
<i>Dock to</i>	<i>Asettuu tarkasti valittuun kohtaan</i>
<i>Relative move</i>	<i>Liikkuu vain suoraan määritettyyn suuntaa ja sille määritetyn etäisyyden verran</i>

Edit marker

Name
VI-shelf

Type
VL-marker

Orientation from X-axis
-180.000

X coordinate in meters
75.126

Y coordinate in meters
13.158

X offset
-0.93

Y offset
0.087

Orientation offset
0.9

OK Detect marker Cancel

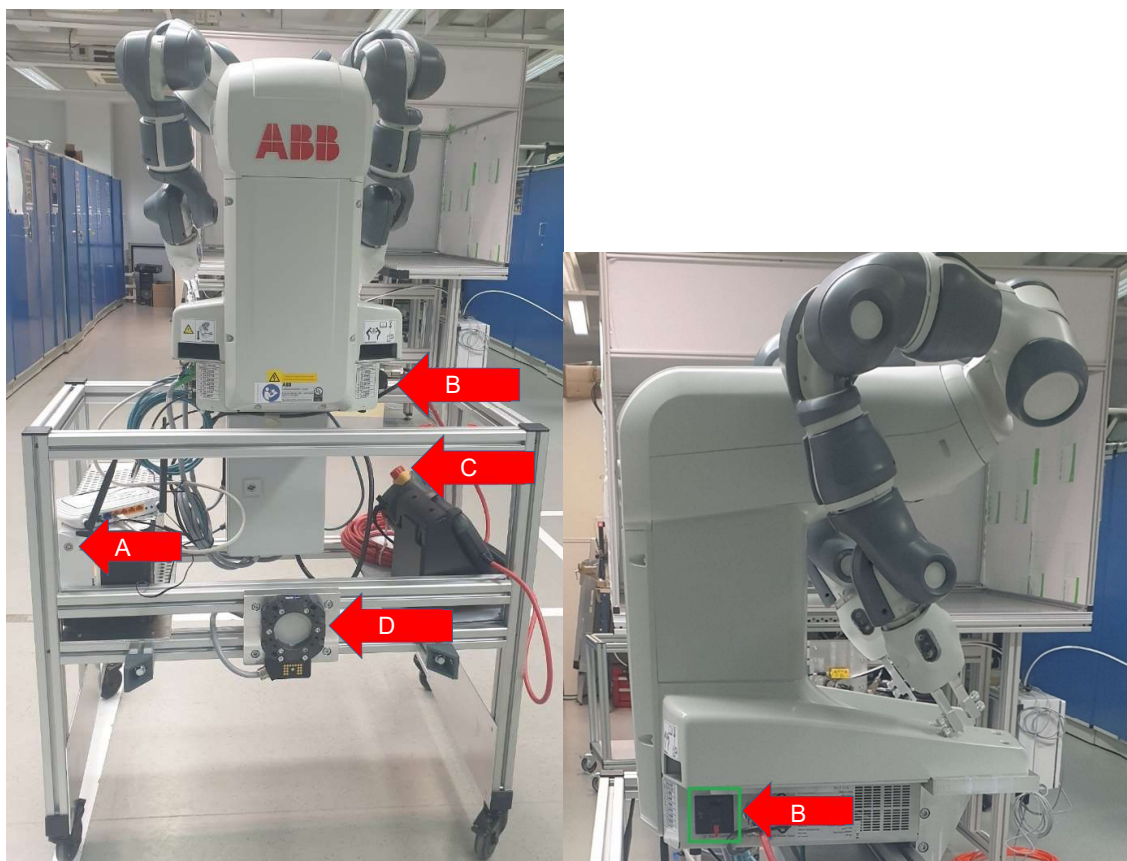
Kuva 27. VL-marker ja arvot, joita se sisältää.

4.3.2 Yumi-kobotti

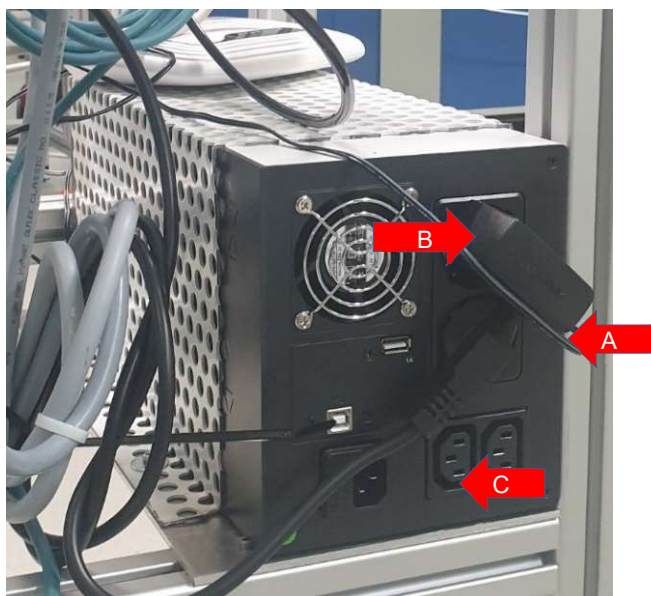
YuMi-kobotin liikuttaminen paikasta toiseen on mahdollistettu kuljetusalustalla ja siihen sijoitetuilla lisätarvikkeilla. Turva-anturi (ABB Eden) on sijoitettu työskentelypöydän ja YuMin-kuljetusalustan telakointijärjestelmän yhteyteen. Sen tehtävänä on tunnistaa työskentelypöytä ja telakointiasemanlukitus osan tarpeellinen etäisyys vastakappaleesta. Telakointiaseman vastakappale MiR-mobiilirobotille (Kuva 28, "D").

Langaton yhteys mahdollistettiin YuMi:lle sijoittamalla reititin kuljetusalustalle. Yhteys toteutettiin yhdistämällä YuMi:n WAN-portti modeemin LAN-porttiin.

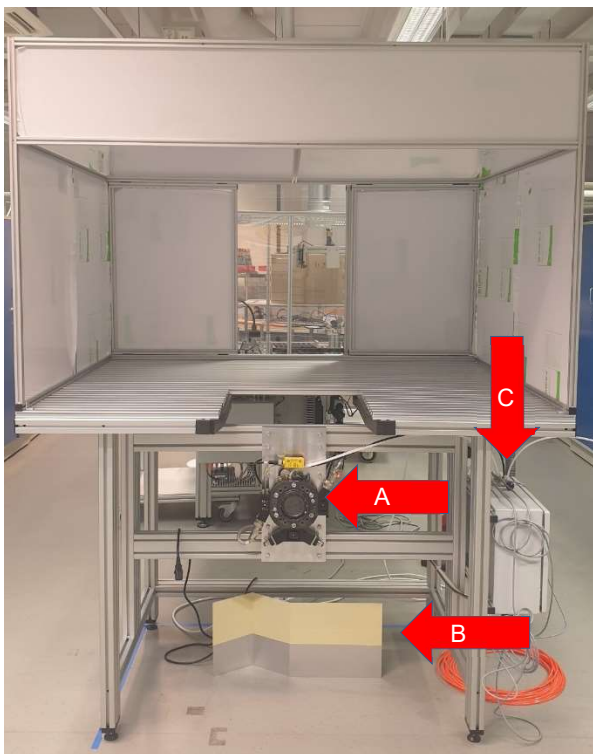
Virran saanti YuMi:lle kuljetuksen aikana toteutettiin asentamalla varavirtalähde kuljetusalustalle, jonka päällä YuMi-kobotti sijaitsee. Reititin saa virtansa myös kyseisestä varavirtalähteestä.



Kuva 28. YuMi kuljetinalustalla, (A) varavirtalähde, (B) YuMi:n päävirtakytkin, (C) Hätäseis painike Flexpendantin sivussa, MiR: ja YuMi:n kuljetusalustan välinen telakoinnin vastakappale (D).



Kuva 29. Varavirtalähteen takaosa, (A) YuMi:n virtajohto, (B) Modeemin virtajohto ja (C) varavirtalähteen latauskaapelin paikka.



Kuva 30. Työskentelypöytä, Telakointiasema (A), VL-Markkeri (B), logiikkakaappi ja paineen syöttö (C).

RobotStudio on ABB:n simulaatio ja ohjelmointiohjelma. Ohjelma tarjoaa digitaalisen jäljitelmän ”digital twin”-robotista ja sen sisältämästä ohjelmasta. Sitä käyttämällä päästään käsiksi YuMi-kobotin toimintoihin myös etäältä käsin. RAPID on ohjelmointikieli, jota käytetään hallinnoimaan ABB:n robotteja.

Järjestelmän työskentelypöydän ja kuljetusalustan telakoinnin ohjausta ohjaa YuMi-kobotti. Ohjelmoimalla RAPID-muuttujia toteuttamaan telakoinnin.

4.4 Telakointijärjestelmä

Telakointijärjestelmä on sekä MiR-mobiilirobotin ja kuljetusalustan välillä (Kuva 20 ja Kuva 28), että kuljetusalustan ja työskentelypöydän välillä (Kuva 32). Telakointijärjestelmä on pneumaattinen ja vaatii toimiakseen paineilmaa (Kuva 33 ja Kuva 31).

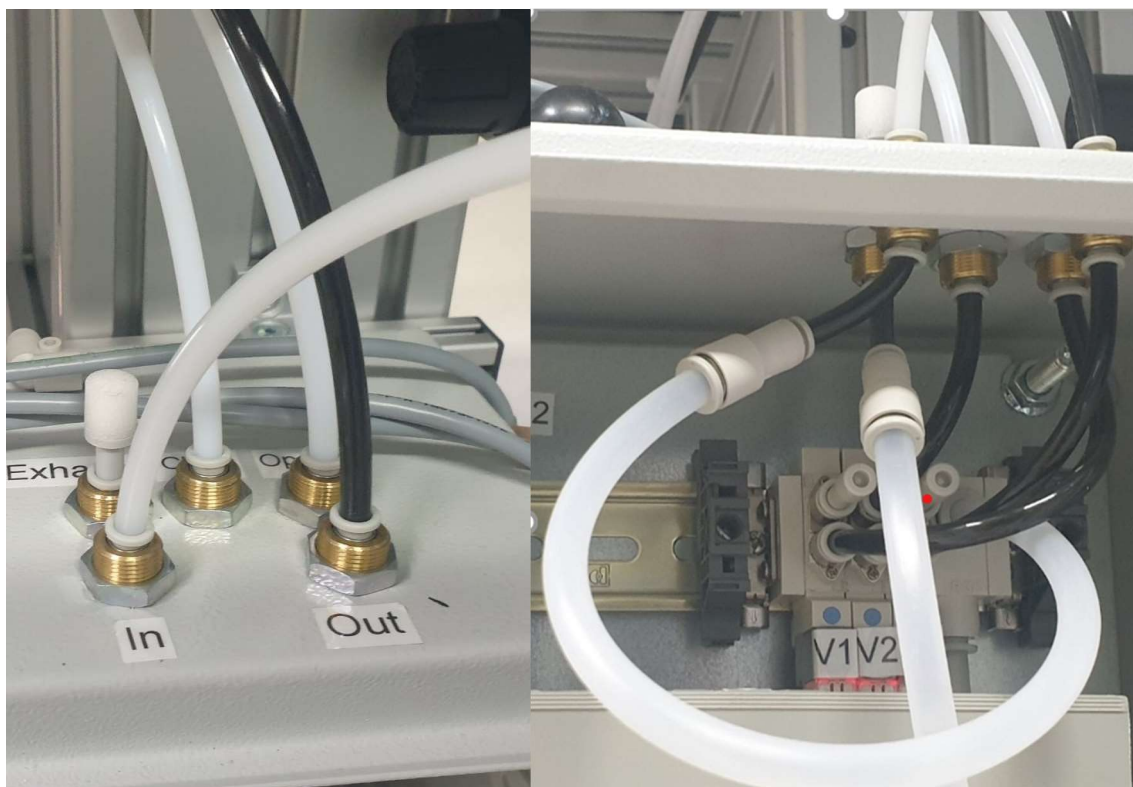
Telakointiin käytetään RSP:n (Robot System Product) telakointikombinaatiota. Osat telakointijärjestelmässä ovat ukitusosa mallia P1801 ja sen liitännäiset osat (Kuva 32). MiR-mobiilirobotin ja kuljetusalustan välistä telakointia ohjaa MiR-mobiilirobotti (Kuva 25 ja Kuva 26).

MiR- mobiilirobotin päälle on asennettu painesäiliö (Kuva 23), josta telakointijärjestelmä saa paineen ja WISE-4060-rele ohjaa telakointia (Kuva 34). WISE-4060-rele saa virtansa MiR-mobiilirobotista ja releen avulla MiR-mobiilirobotti ohjaa telakointi operaatiota.

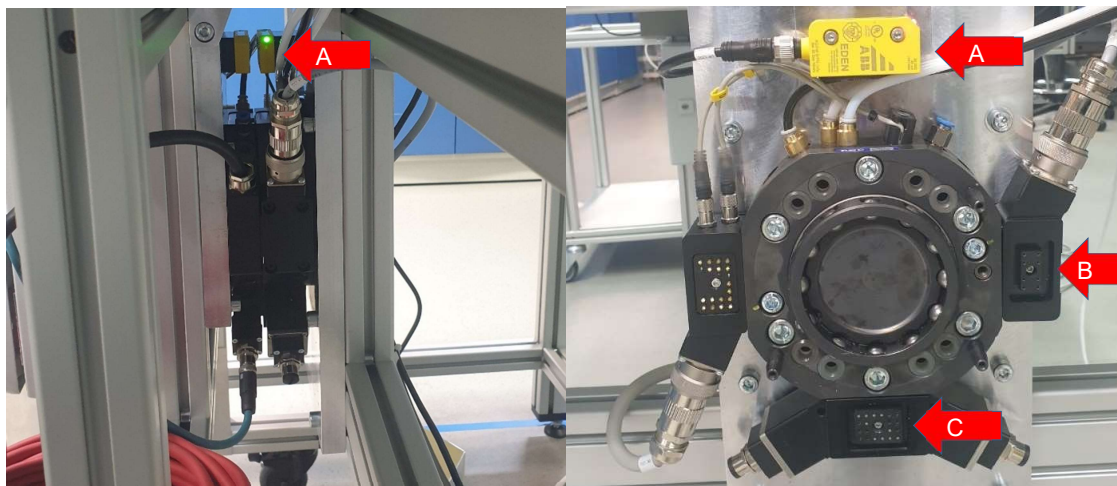
Kuljetusalustan ja työskentelypöydän välistä telakointia ohjaa YuMi-kobotti. Paineen syöttö kuljetusalustan ja työskentelypöydän telakoinnin välillä tulee työskentelypöydästä (Kuva 30, "C").

YuMi-kobotin ja työskentelypöydän välistä telakointiasemaa ohjaa palvelin, joka reagoi MiR-mobiilirobotin tehtävään ohjelmoituun PLC-rekisteriin. Palvelin noutaa ja reagoi PLC-rekisterin muuttuessa yhdestä nolaksi tai nolasta yhdeksi (Kuva 53).

Telakoinnin lukituksessa tulee YuMi-kobotin RAPID-muuttujan arvon olla "true", jonka jälkeen turva-anturi aktivoi telakointilukituksen automaattisesti. Turva-anturi aktivoituu, kun kuljetusalustan turva-anturi on tarpeeksi lähellä sen vastakappaletta työskentelypöydässä (Kuva 32).



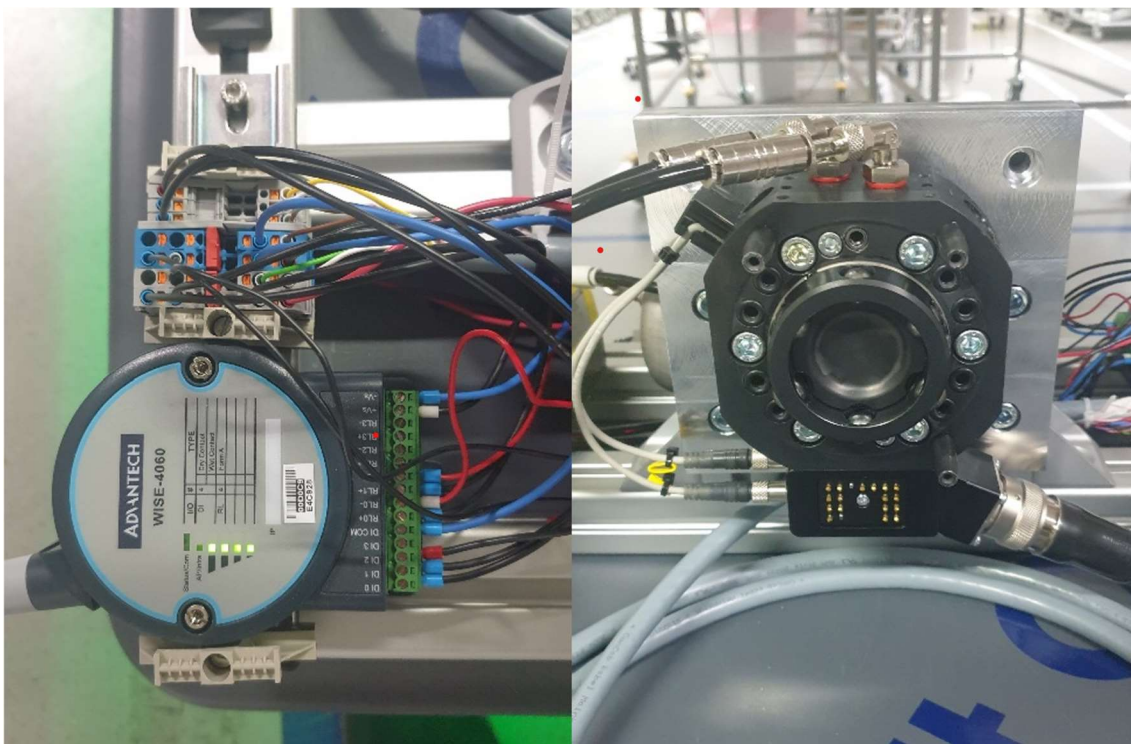
Kuva 31. Logiikka laatikon sisällä oleva pneumaattinen ohjaus ja paineen syöttö järjestelmään.



Kuva 32. Turva-anturi (A), Virransyöttö varavirtalähteelle (B), telakointiaseman kiinnityksen tarkistus I/O (C), LAN yhteys (D).



Kuva 33. Painesäiliö ja sen painemittari.



Kuva 34. WISE-4060-rele ja telakointilukitusosa.

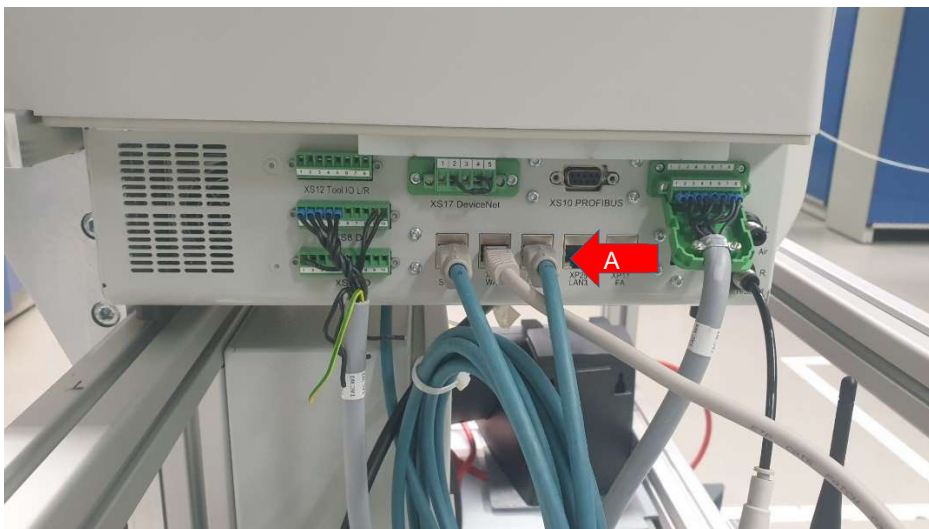
4.5 Yhteys

Robottien yhteyden yhdistäminen palvelimeen on toteutettava saman WiFi-verkon kautta. YuMi:lla ei ole sisäistä langatonta yhteyttä. Langaton yhteys YuMi:lle saadaan toteutettua yhdistämällä Ethernet johto WiFi-reitittimeen (*Kuva 36 ja Kuva 37*).

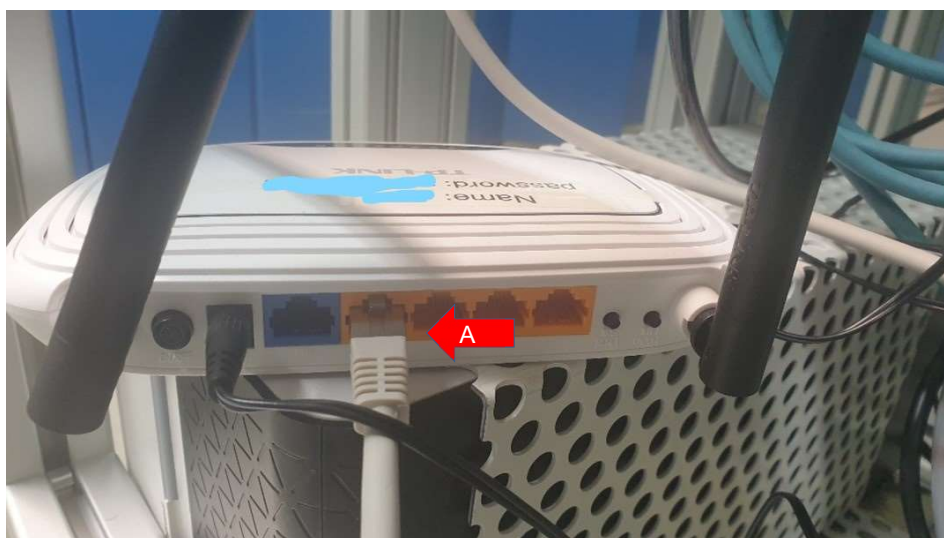
MiR-mobiilirobotti yhdistetään langattomasti YuMi:n kuljetusalustalla sijaitsevan reitittimen WiFi verkkoon (*Kuva 38*). Kirjautumalla reitittimen käyttöliittymään (*Kuva 39*) asetetaan YuMi:n ja MiR:n IP-osoitteet staattisiksi IP-osoitteiksi. MAC-osoitteelle pitää määrittää haluttu IP-osoite (*Kuva 40*).



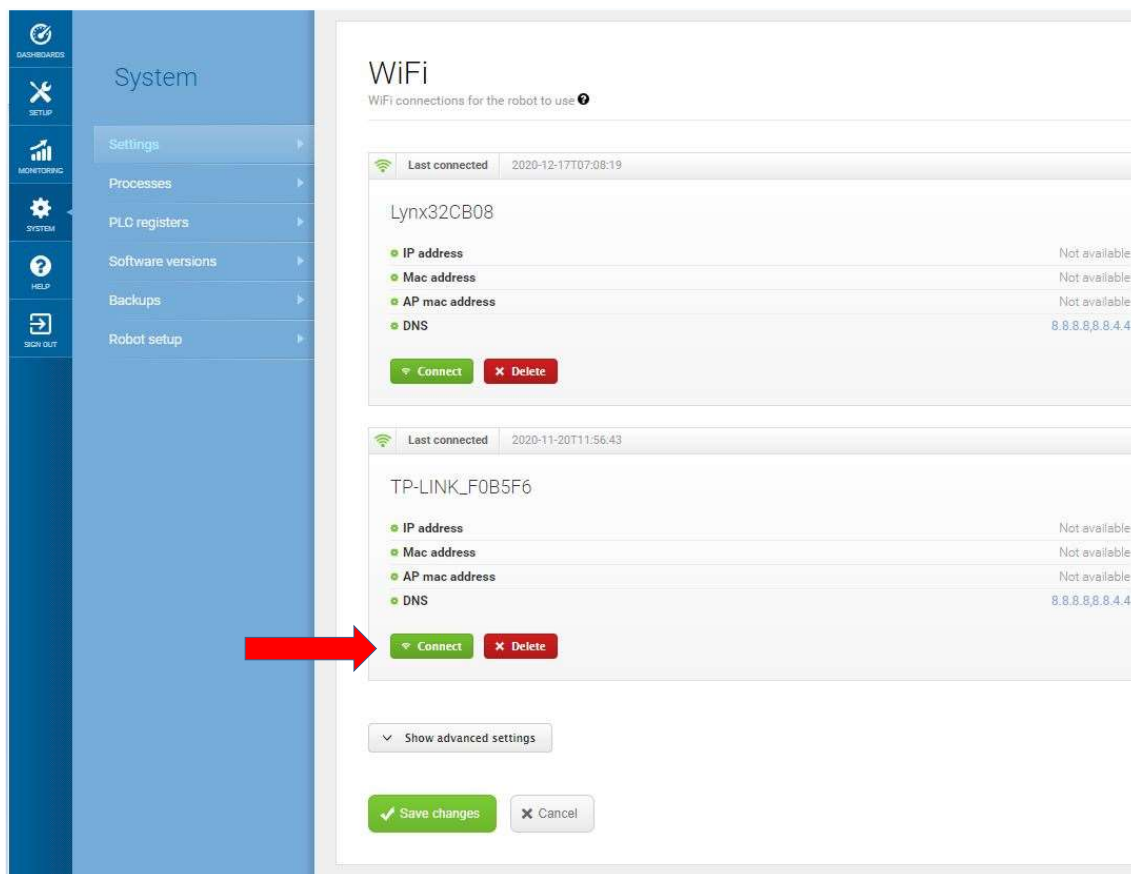
Kuva 35. TP-link TL-WR841N Wi-Fi-reititin.



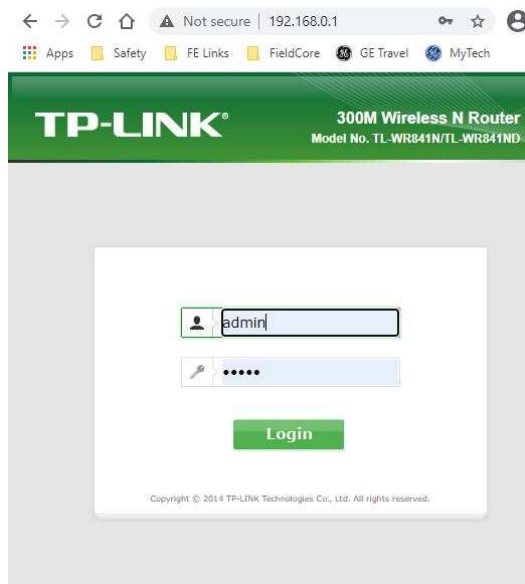
Kuva 36. YuMi-kobotin sivusta, (A) johto kiinnitettynä YuMi:n kyljessä sijaitsevaan WAN porttiin.



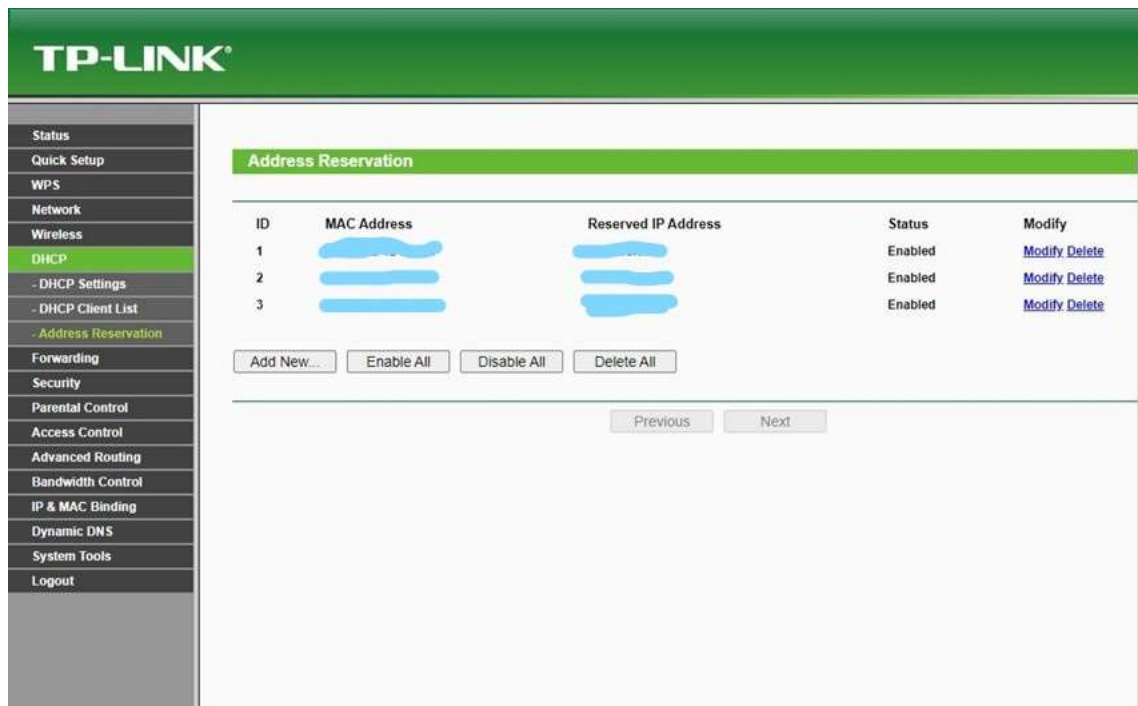
Kuva 37. Reitittimen takaosa, YuMi:sta tuleva (A) ethernet kaapeli LAN portissa.



Kuva 38. MiR käyttöliittymä System/Settings/WiFi.



Kuva 39. Reitittimen käyttöliittymän pääsivu.



Kuva 40. Reitittimen staattisen IP-osoitteen ja MAC-osoitteen määrittäminen.

4.6 Palvelin

Palvelimen luomiseen käytettiin Microsoftin Visual Studio- ohjelmankehitysympäristöä. Projektipohjana käytettiin Windows Forms App (.NET Framework). Ohjelmointikielenä toimi C#-ohjelmointikieli. Palvelimelle luotiin käyttöliittymä (Kuva 41), jonka toiminnosta on kerrottu enemmän kuvan yhteydessä.

Form1

File Edit Help

MiR Connection

Conne cted

Username: [redacted] Port: [redacted] Ip Address: [redacted] Status: Ready Battery: 73.1%

Password: [redacted] Connect

State: Waiting for new missions...

Error MiR

System.Collections.Generic.List`1 [project5.Models.ErrorModel]

YuMi Connection

Conne cted

Username: [redacted] Port: [redacted] Ip Address: [redacted] State: Connected

Password: [redacted] Connect RMMP Connect

Error YuMi

HPJ_OK

Tasking

Mission From: Move

Mission to: Move

10.12.2020 12:35

Tasking Tools

Add Task Edit Task

Refresh Task List Delete Selected Task

Task List

From: Missions	To: Missions	Date:	Status
1: Docking table lock on and to the...	2: Home point to the docking table...	09.12.2020 14.20	Pending

MiR Mission Queue Tools

Pause MiR Add selected

Refresh Queue Clear Queue

Delete Selected Mission

MiR Mission queue

Missions MissionState

YuMi Robot Tools

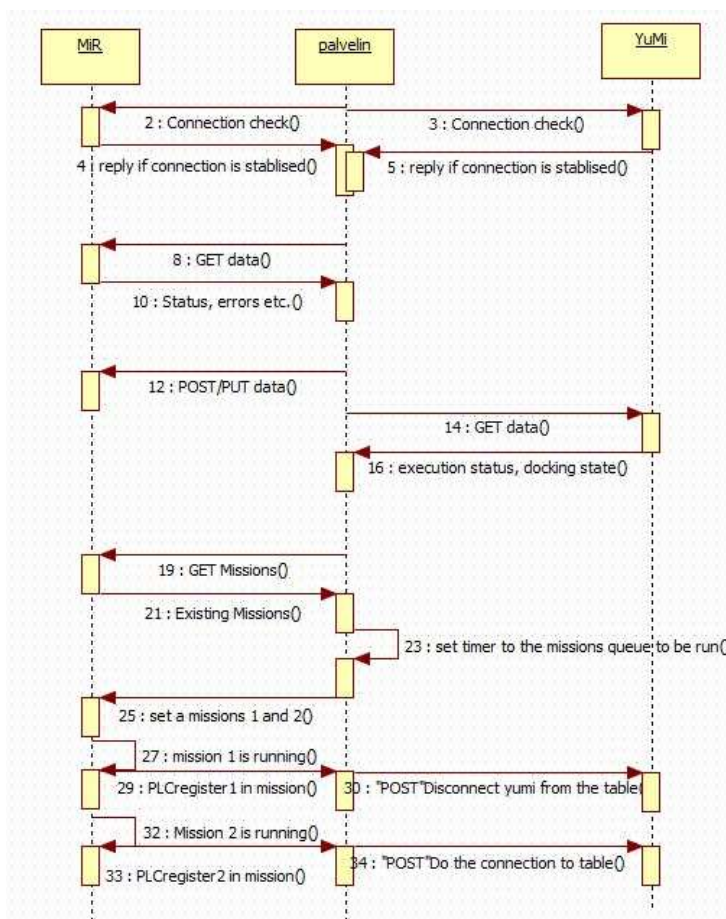
Refresh YuMi List Connect to table

Yumi Robot Online List

Kuva 41. Palvelimen käyttöliittymä.

Taulukko 2. Selityksiä käyttöliittymän (Kuva 41) painikkeille.

Nimi	Toiminto
MiR/ YuMi Connection	Kertoo onko robotti yhdistynyt, jos ei niin painikkeissa lukee "Disconnected, yhdistämisen määrittymiset voidaan muuttaa myös laatikoista (IP-osoite, Tunnuksset ja salasana).
RMMP Connect	kun palvelimen yhdistää uuden YuMi-robotin kanssa ensimmäistä kertaa, niin sen yhdistämisen hyväksyminen robotilla vaatii painikkeen painamista.
Status	Kertoo mikä tila MiR-Robotilla on käynnissä
Tasking	Tehdään tehtävän määrittymiset (mistä, mihin, milloin)
Tasking tools	Painikkeilla pystyy vaikuttamaan ja muuttamaan "Task list" olevia tehtäviä: lisäämään tehtävän jonoon "Add Task", muokkaamaan "Edit Task", poistamaan "Delete Task" ja päivittämään listan "Refresh list".
Task List	Tasking toolissa määritettyjen tehtävien jonotus paikka palvelimessa. Kun tehtävän aika koittaa, palvelin lähettää tehtävän suoritettavaksi
MiR Mission queue	Näkymä tehtäville jotka ovat MiR:n jonossa ja suoritetaan.
MiR Mission Queue Tools	MiR:ssä olevien tehtävien ja tilan vaikuttamiseen ovat painikkeet: Lisäämään suoritettavaksi suoraan "Task List" jonosta "Add selected", muuttaa MiR:n status tilaa ja muuttaa tekstiä statuksen muututtua (Kuvassa) "Pause MiR, päivittää "MiR Mission queue" listan, tyhjentää MiR:n jonosa, tai suoritettavien tehtävien listan "Clear Queue", poistaa valitun tehtävän listasta "Delete selected Mission"
YuMi Robot Online List	Ei toiminnassa, tarkoitus oli lisätä kaikki käytössä olevat YuMi robotit ja näyttää ne listassa ja niide tila.
YuMi Robot Tools	Yumi:n kuljetinalustan ja työskentelypöydän telakoinnin Connect ja Disconnect "Connect to table"
Error MiR/YuMi	Robottien virhetilat näkyvät teksti laatikoissa



Kuva 42. Palvelimen ohjelman toteutuksen lopputulos.

4.7 Järjestelmässä käytetty REST API

Ohjelmointikielenä toteutuksessa käytettiin C# ohjelmointi kieltä ja projektipohjana Windows Forms App (.NET Framework).

HTTP-metodit ovat toimintoja, joita suoritetaan API:lla ("Application programming interface"):

4.7.1 MiR REST-API

MiR-mobiilirobotin yhteys REST-API:n vaatii MiR:n verkkopalvelimen tunnuksien ja salasanan määrittämisen koodiin (Kuva 43).

Yhteyden luomiseksi salansa täytyi muuntaa koodikieleksi käyttämällä "BASE64", joka muuntaa sen MiR-verkkopalvelimelle ymmärrettävään muotoon (Kuva 44).

Ohjelmassa käytetyissä operaatioissa käytetään hyväksi jo luotua koodia (Kuva 44), joka yhdistää ohjelman MiR-verkkopalvelimeen ja lisää siihen operaatioissa tarvittavat osat (Kuva 45, Kuva 46, Kuva 47 ja Kuva 48).

```

3 references
public class SetAuthenticationValues
{
    1 reference
    public static string SetIP()
    {
        // Retrieve ip from file and set it to Form
        return "██████████";
    } /*"192.168.0.101"*/

    1 reference
    public static string SetUserName()
    {
        // Retrieve ip from file and set it to Form
        return "██████████";
    }

    1 reference
    public static string SetPassWord()
    {
        // Retrieve ip from file and set it to Form
        return "██████████";
    }
}

```

Kuva 43. MiR verkkopalvelimen tunnuksien ja salasanan määrittäminen.

```

private static HttpClient httpClient;

#region Connection/authentication
2 references
public static bool RestConn(string ipAddress, string userName, string password, bool reconnectMiR)
{
    if (reconnectMiR)
    {
        httpClient = new HttpClient();
        httpClient.Timeout = TimeSpan.FromMilliseconds(1000);

        var uri = "http://" + ipAddress + "/api/v2.0.0/";
        httpClient.BaseAddress = new Uri(uri);
        httpClient.DefaultRequestHeaders.Accept.Clear();
        httpClient.DefaultRequestHeaders.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));

        string passwordHashed;

        using (var sha256 = new SHA256Managed())
        {
            passwordHashed = BitConverter.ToString(sha256.ComputeHash(Encoding.UTF8.GetBytes(password))).Replace("-", "");
        }

        httpClient.DefaultRequestHeaders.Add("Authorization", "Basic " + System.Convert.ToBase64String(System.Text.Encoding.UTF8.GetBytes($"{userName}:{passwordHashed}")));
    }
    try
    {
        HttpResponseMessage response = httpClient.GetAsync("missions").Result;
        if (response.StatusCode.ToString() == "OK")
        {
            return true;
        }
        else
        {
            return false;
        }
    }
    catch (Exception ex)
    {
        //ex.Message.ToString()
        return false;
    }
}
#endregion

```

Kuva 44. MiR verkkopalvelimen yhteyden luominen ja sen tarkastaminen.

```

2 references
public List<MissionModel> GetMission()
{
    try
    {
        HttpResponseMessage response = httpClient.GetAsync("missions").Result;
        String restData = response.Content.ReadAsStringAsync().Result;

        List<MissionModel> myDeserializedClass = JsonConvert.DeserializeObject<List<MissionModel>>(restData);
        return myDeserializedClass;
    }
    catch (Exception ex)
    {
        return null;
    }
}

```

Kuva 45. MiR verkkopalvelimen GET- operaatio.

```

1 reference
public List<MissionQueueInfoList> GetMissionQueue()
{
    try
    {
        HttpResponseMessage response = httpClient.GetAsync("mission_queue").Result;
        String restData = response.Content.ReadAsStringAsync().Result;
        List<MissionQueueList> myDeserializedClass = JsonConvert.DeserializeObject<List<MissionQueueList>>(restData);

        //Create a new object of the Mission queue rest model to store the missions received from MiR
        List<MissionQueueInfoList> missionQueueInfoList = new List<MissionQueueInfoList>();
        // Loop through List for "Pending" and "Executing" mission queued
        foreach (MissionQueueList item in myDeserializedClass)
        {
            if (item.state == "Pending" || item.state == "Executing")
            {
                MissionQueueInfoList missionQueueInfoList1 = new MissionQueueInfoList();
                missionQueueInfoList1.MissionQueueId = item.id.ToString();
                missionQueueInfoList1.MissionState = item.state.ToString();
                missionQueueInfoList.Add(missionQueueInfoList1);
            }
        }

        // If there is no mission queue, the abort the operation
        if (missionQueueInfoList.Count == 0) return null;

        // Get the mission GUID and Mission Name for above
        foreach (MissionQueueInfoList item in missionQueueInfoList)
        {
            Rest a = new Rest();
            string guid = a.MissionInfo(int32.Parse(item.MissionQueueId));
            item.MissionGuid = guid;
            item.MissionName = a.MissionName(guid);
        }

        //missionQueueInfoList.Add(new MissionQueueInfoList()
        //{
        //    MissionGuid = item.id.ToString()
        //});

        return missionQueueInfoList;
    }
    catch (Exception ex)
    {
        return null;
    }
}

```

Kuva 46. MiR verkkopalvelimen GET- operaatio.

```

2 references
public static bool PostPLCRegister(string RegisterNo, int value, string label)
{
    var newPostPlcRegisterModel = new PostPlcRegisterModel();
    newPostPlcRegisterModel.value = value;
    newPostPlcRegisterModel.label = label;
    //newPLCRegisterModel.id = RegisterNo.ToString();

    var json = Newtonsoft.Json.JsonConvert.SerializeObject(newPostPlcRegisterModel);
    try
    {
        var response = httpClient.PostAsync("registers/" + RegisterNo, new System.Net.Http.StringContent(json, Encoding.UTF8, "application/json"));
        if (response.Result.StatusCode.ToString() == "OK")
        {
            return true;
        }
        else
        {
            return false;
        }
    }
    catch (Exception ex)
    {
        //ex.Message.ToString()
        return false;
    }
}

```

Kuva 47. PLC-rekisterin tilan muuttaminen POST operaatiota käyttämällä MiR-verkkopalvelimessa.

```

public static bool ChangeMiRStateToMove(int stateID, bool resetError)
{
    // Post data from JSON with
    //string json = "{ \"state_id\": 3 }";

    var settings = new JsonSerializerSettings
    {
        DefaultValueHandling = DefaultValueHandling.Ignore,
        NullValueHandling = NullValueHandling.Ignore
    };
    var newStatusID = new PutStatus();

    if (resetError == true)
    {
        newStatusID.ClearError = resetError;
    }

    else
    {
        newStatusID.StateId = stateID;
    }

    var json = Newtonsoft.Json.JsonConvert.SerializeObject(newStatusID, settings);
    try
    {
        var response = httpClient.PutAsync("status", new System.Net.Http.StringContent(json, Encoding.UTF8, "application/json"));
        //var response = httpClient.PostAsync("status", new System.Net.Http.StringContent(json, Encoding.UTF8, "application/json"));
        if (response.Result.StatusCode.ToString() == "OK")
        {
            return true;
        }
        else
        {
            return false;
        }
    }
    catch (Exception ex)
    {
        //ex.Message.ToString()
        return false;
    }
}

```

Kuva 48. MiR-mobiilirobotin tilan muuttaminen PUT-operaatiota käyttämällä.

4.7.2 YuMi REST API

Palvelimen yhteyden määrittämiseen YuMi-kobottin verkkopalvelimeen vaatii IP-osoitteen, tunnuksen ja salasanan (Kuva 49).

YuMi:n verkkopalvelin vaatii salasanan muuntamisen "application/x-www-form-urlencoded"-muotoon, jotta verkkopalvelin ymmärtää sen ja antaa pääsyn toteuttamaan PUT, POST ja DELETE operaatioita. GET operaatio ei vaadi salasanan muuntamista kyseiseen muotoon (Kuva 50).

YuMi:n verkkopalvelimeen suoritettavat operaatiot käyttävät yhteyden muodostamiseen tehtyä koodia hyväkseen suoriutuakseen (Kuva 52).

```
3 references
public class SetAuthenticationValuesYumi
{
    1 reference
    public static string SetIPYumi()
    {
        // Retrieve ip from file and set it to Form
        return "[REDACTED]";
    }

    1 reference
    public static string SetUserNameYumi()
    {
        // Retrieve ip from file and set it to Form
        return "[REDACTED]";
    }

    1 reference
    public static string SetPassWordYumi()
    {
        // Retrieve ip from file and set it to Form
        return "[REDACTED]";
    }
}
```

Kuva 49. YuMi verkkopalvelimen tunnuksien ja salasanan määrittäminen.

```

#region Connection/authentication
private static HttpClient httpClient;

3 references
public static bool RestConnYumi(string ipAddressYumi, string userNameYumi, string passWordYumi, bool reconnectMirYumi)
{
    if (reconnectMirYumi)
    {
        var handler = new HttpClientHandler
        {
            Credentials = new NetworkCredential(userNameYumi, passWordYumi),
            Proxy = null,
            UseProxy = false
        };

        httpClient = new HttpClient(handler);
        httpClient.Timeout = TimeSpan.FromMilliseconds(1000);

        var uri = "http://" + ipAddressYumi + "/";
        httpClient.BaseAddress = new Uri(uri);
        httpClient.DefaultRequestHeaders.Accept.Clear();
        httpClient.DefaultRequestHeaders.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));
        //httpClient.DefaultRequestHeaders.Add("Content-Type", "application/x-www-form-urlencoded");
        //httpClient.DefaultRequestHeaders.Accept.Add(new MediaTypeWithQualityHeaderValue("application/x-www-form-urlencoded"));
        //Content-type
    }
    try
    {
        HttpResponseMessage response = httpClient.GetAsync(@"rw/system?json=1").Result;
        String b = response.Content.ReadAsStringAsync().Result;

        if (response.StatusCode.ToString() == "OK")
        {
            return true;
        }
        else
        {
            return false;
        }
    }
    catch (Exception ex)
    {
        //ex.Message.ToString()
        return false;
    }
}
#endregion

```

Kuva 50. YuMi verkkopalvelimen yhteyden luominen ja sen tarkastaminen.

```

2 references
public static ExecutionRoot GetExecutionStateYumi()
{
    try
    {
        HttpResponseMessage response = httpClient.GetAsync(@"rw/rapid/execution?json=1").Result;
        String restData = response.Content.ReadAsStringAsync().Result;

        ExecutionRoot myDeserialiedClass = JsonConvert.DeserializeObject<ExecutionRoot>(restData);
        String b = response.Content.ReadAsStringAsync().Result;

        return myDeserialiedClass;
    }
    catch (Exception ex)
    {
        return null;
    }
}

```

Kuva 51. YuMi- verkkopalvelimen GET- operaatio.


```

public static bool PostUpdateRapidV(string uri1, Dictionary<string, string> keyValues)
{
    if (!uri1.EndsWith("?action=set"))
        uri1 += "?action=set";
    HttpResponseMessage response = new HttpResponseMessage();
    try
    {
        response = httpClient.PostAsync(uri1, new FormUrlEncodedContent(keyValues)).Result;
        //var encoding = new FormUrlEncodedContent(response);

        if (response.IsSuccessStatusCode == true)
        {
            return true;
        }
        else
        {
            return false;
        }
    }
    catch (Exception ex)
    {
        //ex.Message.ToString()
        return false;
    }
}
#endregion

#region Docking station True/false funktion
1 reference
public static bool ConnectYuMi()
{
    string uri1 = "rw/rapid/symbol/data/RAPID/T_CHANGER/ChangerData/bDisconnect";

    var keyValues = new Dictionary<string, string>();
    keyValues.Add("value", "false");
    bool succeeded = PostUpdateRapidV(uri1, keyValues);
}

```

Kuva 52. YuMi-verkkopalvelimen POST-operaatio.

4.7.3 YuMi:n telakointiohjelma

YuMi-verkkopalvelimen ohjaama telakoinnista on tehty kaksi erillistä koodia, jotka suorittavat telakoinnin irrotus- ja kiinnitysoperaatioita (Kuva 53).

koodeille on määrätty reitti, jotta se löytää oikean parametrin muuttaa oikeasta paikasta.

```

1 reference
public static bool ConnectYuMi()
{
    string uri1 = "rw/rapid/symbol/data/RAPID/T_CHANGER/ChangerData/bDisconnect";

    var keyValues = new Dictionary<string, string>();
    keyValues.Add("value", "false");
    bool succeeded = PostUpdateRapidV(uri1, keyValues);
    if (succeeded)
    {
        var newrabid = GetRabidData(uri1);

        if (newrabid.Embedded.State[0].Value == "FALSE")
        {
            var keyValues1 = new Dictionary<string, string>();
            keyValues1.Add("value", "true");
            uri1 = "rw/rapid/symbol/data/RAPID/T_CHANGER/ChangerData/bConnect";
            succeeded = PostUpdateRapidV(uri1, keyValues1);
            if (succeeded)
                return true;
        }
    }
    return false;
}

2 references
public static bool DisconnectYuMi()
{
    string uri1 = "rw/rapid/symbol/data/RAPID/T_CHANGER/ChangerData/bConnect";

    var keyValues = new Dictionary<string, string>();
    keyValues.Add("value", "false");
    bool succeeded = PostUpdateRapidV(uri1, keyValues);
    if (succeeded)
    {
        var newrabid = RestYumi.GetRabidData(uri1);

        if (newrabid.Embedded.State[0].Value == "FALSE")
        {
            var keyValues1 = new Dictionary<string, string>();
            keyValues1.Add("value", "true");
            uri1 = "rw/rapid/symbol/data/RAPID/T_CHANGER/ChangerData/bDisconnect";
            succeeded = PostUpdateRapidV(uri1, keyValues1);
            if (succeeded)
                return true;
        }
    }
    return false;
}

```

Kuva 53. Palvelimen kommunikointi YuMi-verkkopalvelimen kanssa.

4.8 Testaus

Järjestelmää testattiin ja hiottiin noin neljä viikkoa. Yhtenä ongelmana ilmeni robotin itsensä paikantaminen, joka saatiin korjattua lisäämällä MiR-mobiilirobotin suorittamiin tehtäviin useampi sijainnin säätökäskyjä (adjusting location).

Toisena asiana, jota saatiin hiottua ja sujumaan paremmin, oli telakointiosio. Aluksi kuljetusalustan ja työskentelypöydän telakointi oli ajoitettu. Se muutettiin lisäämällä PLC-reksiteri käskyn MiR-mobiilirobotin suorittamaan tehtävään. Palvelin saa tiedon siitä, kun

MiR:n tehtävän PLC-rekisterikäsky on suoritettu ja välittää tiedon YuMi-kobotille, joka valmistaa kuljetusalustan telakointiasemaa työskentelypöytää varten.

Testauksen aikana YuMi:n kuljetusalustan kuljetus sujui hyvin ja mobiilirobotti huomioi liikkuvat objektit ja ihmiset. Mobiilirobotin ja kuljetusalustan liikkuvuutta testattiin alustalla, joka oli yhtenäinen muovimatto ja ei päästy testaamaan sitä eri ympäristössä.

Järjestelmän testauksen aikana ei sen ihmeempiä ongelmia tullut vastaan. Telakointi ja tehtävien suoritus sujui mutkitta.

5 Yhteenveto

Insinööriyön aiheena oli Reboot IoT Factoryn (Vaihe I, GC2 PoC 2.2 Docking) tuotekehitys. Reboot IoT Factoryn (Vaihe II, GC2 PoC 2.2 Docking) tavoitteena oli suunnitella ja luoda järjestelmä, joka automaattisesti siirtää YuMi-kobotin työskentelypöydältä toiselle.

Työn teoriaosuudessa perehdyttiin järjestelmässä käytettyyn tekniikkaan, kilpaileviin vaihtoehtoihin ja kyseisen tekniikan historiaan. Työn käytännön osuudessa kehitettiin Reboot IoT Factoryn ensimmäistä vaihetta ja automatisoitiin prosessi.

Haasteen toinen vaihe koostui palvelimen ohjelmoinnista, robottien ohjelmoinnista, testauksesta, järjestelmän käyttöoppaan laatimisesta, toteutuksen suunnittelusta ja sen toteutuksesta.

5.1 Lopputuloksen tarkastelu

Työ oli laaja ja tavoite selkeä. Tavoitteena oli luoda järjestelmä, joka siirtää YuMi-kobotin työskentelypöydältä toiselle. Vastaavaa järjestelmää ei ollut valmiina saatavilla, joten täytyi luoda täysin uusi järjestelmä.

Lopputuotoksena saatiin järjestelmä toimimaan tavalla, mitä alussa suunniteltiin ja toteutuksen aikataulusta pidettiin kiinni.

Toteutuksen mahdollistamiseksi järjestelmälle luotiin palvelin, joka ohjasi ja ajoitti robottien toiminnan ja telakoinnin.

Insinööriyötä tehdessä käytiin läpi järjestelmän solun ja kuljetusalustan suunnittelua, jonka tekemiseen käytettiin Creo-suunnitteluohjelmaa.

Erityisesti työssä syvennyttiin ohjelmoinnin ja tietoliikenteen merkitykseen järjestelmän robotiikan automatisoinnissa. Lopuksi järjestelmälle tehtiin käyttöopas, jossa käydään läpi perusteellisesti järjestelmän käyttöohjeita, kuinka se toimii ja mahdollisten ongelmatausten ratkaisuja.

Insinööriyössä kehitetyn kaltaiset järjestelmät tulevat lisääntymään tulevaisuuden teollisuudessa huomattavasti.

5.2 Haasteet

Insinööriyönprojektin vaatimuksena oli luoda järjestelmä, joka siirtää yhteistyörobotin, kun se on lopettanut työskentelyn yhdessä solussa, ja siirtää se toiseen soluun. Vastavaa järjestelmää ei ole saatavilla valmiina, joten se täytyi rakentaa tyhjästä.

Projektissa käytettiin järjestelmälle suunniteltua yhteistyörobotin kuljetusalustaa ja telakointijärjestelmää. Mobiilirobotin ja yhteistyörobotin kuljetusalustan telakointi toteutettiin projektissa pneumaattisella telakointijärjestelmällä.

Mobiilirobotin ja yhteistyörobotin välinen kommunikaatio palvelimeen oli haasteellista toteuttaa ilman suuntaa antavaa pohjaa yhteyden muodostamiselle, datan välittämiseksi ja datan vastaanottamiselle palvelimessa. Ohjelmointirajapintana käytettiin REST API:a. Yhteistyörobotin ja mobiilirobotin REST API ohjelmointirajapinnoissa on hieman eroa.

5.3 Luetettavuus ja virheiden arviointi

Järjestelmän toimivuutta testattiin noin kolme viikkoa. Ongelma, joka ilmeni testauksien aikana, oli MiR-mobiilirobotin itsensä paikantaminen ja sijoittaminen kartalle tarkasti elävässä ympäristössä; tavaroita, lavoja ja karrjja liikuteltiin paikasta toiseen. Se saatiin korjattua lisäämällä MiR-mobiilirobotin suorittamiin tehtäviin useampi sijainnin säätöä (adjusting location) käsky. Ongelma saatiin korjattua lisäämällä MiR-mobiilirobotin suorittamiin tehtäviin useampi sijainnin säätöä (adjusting location) käsky.

MiR-mobiilirobotin kulkualustana testauksen aikana toimi yhtenäinen muovimatto, joka oli otollinen mobiilirobotin liikkuvuuden kannalta. Järjestelmän toimivuutta ei ole testattu muissa ympäristöissä, joten voi vain tehdä ennakoivaa riskienarviointia:

- Järjestelmän liikkuvuuden toimivuuteen liikkuvuudessa voi mahdollisesti ilmetä ongelmia, jos reitin varrella on kynnyksiä.
- Robottien ja palvelimen yhteyden vakaus ja vahvuus voi vaihdella, jos palvelimen ja robottien välille kertyy etäisyyttä ja esteitä, jotka heikentävät langatonta yhteyttä.

5.4 Jatkokehitys pohdintaa

Jatkokehitysmahdollisuuksia järjestelmässä on monia. Järjestelmä on suunniteltu ja kehitetty toimimaan vain järjestelmässä käytettyjen robottien kanssa. Tulevaisuudessa järjestelmää voidaan kokeilla soveltaa muiden robottien kanssa.

Projekti on toteutettu käyttäen yhtä mobiilirobottia, yhteistyörobottia ja työskentelypöytää, mutta se on silti suunniteltu tulevaisuutta katsoen, että siihen voidaan liittää myöhemmin useampia robotteja ja soluja.

Yksi kehitysalue on robottien ja palvelimen yhteyden järjestäminen silleen, että robotteja voidaan liittää useampia saman WiFi-verkon alaisuuteen, jotta saataisiin useampia operaatioita toimimaan saman järjestelmän alaisuuteen.

Järjestelmän painesäiliön täyttö on jätetty auki, mutta sille on mietitty muutamaa vaihtoehtoa: joko painesäiliön täyttö suoritetaan latauspisteellä ja suunnitellaan sinne automaattinen paineen täyttö järjestelmä tai telakoinninyhteydessä työskentelypöydällä.

Viimeisenä kehitysaiheena olisi selvittää ja testata, voiko järjestelmää soveltaa käyttämällä muitakin robotteja kuin MiR ja YuMi ja vertailla niiden toimivuutta keskenään.

Lähteet

1. Jokinen, Tapani. 2010. Tuotekehitys. Helsinki: Verkkoaineisto. Otatieto Oy. <<http://lib.tkk.fi/Reports/2010/isbn9789526033204.pdf>>. Luettu 3.3.2021.
2. Köseoğlu, Murat. Pektaş, Ömer & Çelik , Orkan Murat. 2017. Design of an autonomous mobile robot based on ROS. Verkkoaineisto. IEE. <<https://ieeexplore.ieee.org/abstract/document/8090199>>. Luettu 16.2.2021.
3. Ullrich, Günter. 2015. Automated Guided Vehicle Systems, A Primer with practical applications. Verkkoaineisto. Springer. <<https://link.springer.com/content/pdf/10.1007/978-3-662-44814-4.pdf>>. Luettu 17.2.2021.
4. International robot standardization within ISO. 2020. Verkkoaineisto. IFR. <<https://ifr.org/standardisation/>>. Luettu 25.2.2021.
5. Matarić, Maja J. 2007. The Robotics Primer. The MIT Press, London, England.
6. Goodwin, Damond. 2020. The Evolution of Autonomous Mobile Robots. Verkkoaineisto. CONTROL AUTOMATION. <<https://control.com/technical-articles/the-evolution-of-autonomous-mobile-robots/>>. Luettu 28.2.2021.
7. LeBouthillier, Arthur Ed. 2010. W. Grey Walter and his Turtle Robots. Verkkoaineisto. Robotics Society of Southern California. <<http://web.csulb.edu/~wmartinz/rssc/content/w-grey-walter-and-his-turtle-robots.html> >. Luettu 1.3.2021
8. The Evolution of Mobile Robotics Solutions. 2019. Verkkoaineisto. INVIA Robotics. <<https://www.inviarobotics.com/blog/evolution-mobile-robotics-solutions/> >. Luettu 3.3.2021
9. B. Alatise, Mary. P. Hancke, Gerhard. 2020. A Review on Challenges of Autonomous Mobile Robot and Sensor Fusion Methods. Verkkoaineisto. IEEEAccess. <<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9007654>>. Luettu 4.3.2021
10. LD Series Autonomous Mobile Robots. 2021. Verkkoaineisto. Omron, Automation. <<https://automation.omron.com/en/us/products/family/ld>>. Luettu 5.3.2021.
11. Lazarte, Maria. 2016. Robots and humans can work together with new ISO guidance. Verkkoaineisto. ISO. <<https://www.iso.org/news/2016/03/Ref2057.html>>. Luettu 7.3.2021.
12. Bélanger-Barrette, Mathieu. 2015. What Does Collaborative Robot Mean?. Verkkoaineisto. Robotiq. <<https://blog.robotiq.com/what-does-collaborative-robot-mean>>. Luettu 8.3.2021.
13. Pittman, Kagan. 2018. A History of collaborative robots: From intelligent lift assists to cobots. Verkkoaineisto. Engineering.com. <<https://www.engineering.com/story/a>>

- history-of-collaborative-robots-from-intelligent-lift-assists-to-cobots>. Luettu 8.3.2021
14. Our History "Make robot technology accessible to all". 2021. Verkkoaineisto. Universal Robots. <<https://www.universal-robots.com/about-universal-robots/our-history/>>. Luettu 9.3.2021.
 15. Collaborative robot buyer's guide, Ebook. 2020. Verkkoaineisto. Robotiq. <<https://blog.robotiq.com/hubfs/COBOT%20EBOOK%20FINAL.pdf>>. Luettu 9.03.2021.
 16. The History of KUKA. 2021. Verkkoaineisto. KUKA. <<https://www.kuka.com/en-se/about-kuka/history>>. Luettu 9.3.2021.
 17. History of ABB. 2020. Verkkoaineisto. ABB. <<https://global.abb/group/en/about/history>>. Luettu.9.3.2021
 18. Built to Work together with you. 2021. Verkkoaineisto. ABB. <<https://cobots.robotics.abb.com/en/>>. Luettu. 9.3.2021
 19. Built to Work together with you. 2021. Verkkoaineisto. ABB. <<https://new.abb.com/products/robotics/collaborative-robots>>. Luettu 9.3.2021
 20. About us. 2021. Verkkoaineisto. Robot System Products. <<https://www.robot-systemproducts.com/en/about-us>>. Luettu 11.3.2021
 21. Products. 2021. Verkkoaineisto. Robot System Products. <<https://www.robot-systemproducts.com/en/products/tool-changers/modular/tc180/p1803-tool-changer-tc180-8e-detail>>. Luettu 11.3.2021
 22. About Advantech. 2021. Verkkoaineisto. Advantech. <<https://www.advantech.com/about>>. Luettu 11.3.2021
 23. WISE-4060. 2021. Verkkoaineisto. Advantech. <https://www.advantech.com/products/4260f153-57cd-4102-81ea-7a0f36d9b216/wise-4060/mod_bb247acb-d538-4e6f-9402-6030fe8dbc31>. Luettu 11.3.2021.
 24. Eden. 2021. Verkkoaineisto. ABB. <<https://new.abb.com/low-voltage/products/safety-products/safety-sensors-switches-and-locks/eden>>. Luettu 11.3.2021
 25. Windows Server administration fundamentals: exam 98-365. 2011. Verkkoaineisto. Yhdysvallat, New Jersey: John Wiley & Sons Inc. <<https://archive.org/details/windowsserveradm00cour>>. Luettu 12.3.2021
 26. Oluwatosin, Haroon Shakirat. 2014. Client-Server Model. Verkkoaineisto. Malaysia, Kedah: IOSR Journal of Computer Engineering. <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1083.8741&rep=rep1&type=pdf>>. Luettu 13.3.2021
 27. Programming software and the IDE. 2021. Verkkoaineisto. United Kingdom: BBC. <<https://www.bbc.co.uk/bitesize/guides/zgmp82/revision/1>>. Luettu 13.3.2021

28. What is an Integrated development environment (IDE)?. 2021. Verkkoaineisto. Yhdysvallat, Massachusetts: Veracode. <<https://www.veracode.com/security/integrated-development-environment>>. Luettu 14.3.2021
29. Visual Studio 2019. 2021. Verkkoaineisto. Microsoft. <<https://visualstudio.microsoft.com/vs/>>. Luettu 14.3.2021
30. Community for Open Innovation and Collaboration. 2021. Verkkoaineisto. Eclipse Foundation. <<https://www.eclipse.org/>>. Luettu 14.3.2021
31. IntelliJ IDEA Capable and Ergonomic IDE for JVM. 2021. Verkkoaineisto. JetBrains. <<https://www.jetbrains.com/idea/>>. Luettu 14.3.2021
32. What is an API?. 2021. Verkkoaineisto. Red Hat. <<https://www.redhat.com/en/topics/api/what-are-application-programming-interfaces>>. Luettu 15.3.2021
33. How to use IG's trading APIs. 2020. Verkkoaineisto. IG. <<https://www.ig.com/au/trading-platforms/trading-apis/how-to-use-ig-api>>. Luettu 15.3.2021
34. Pedro, B. 2017. What are web APIs. Verkkoaineisto. Hackernoon. <<https://hackernoon.com/what-are-web-apis-c74053fa4072>>. Luettu 15.3.2021
35. Grand Challenge 2. 2020. Verkkoaineisto. <<https://rebootiofactory.fi/robotics-fusion/>>. Luettu 15.1.2021
36. About us. 2021. Verkkoaineisto. <<https://www.ge.com/about-us>>. Luettu. 15.1.2021
37. Automated guided vehicles (AGV). 2017. Verkkoaineisto. <<https://www.spscap.com/agv>>. Luettu 15.03.2021.
38. Backlund, Alexander. 2000. The definition of system. Verkkoaineisto. Emerald insight. <<https://www.emerald.com/insight/content/doi/10.1108/03684920010322055/full/html#loginreload>>. Luettu 10.4.2021.
39. IEEE Recommended Practice for Architectural Description of Software-Intensive Systems. 2000. Verkkoaineisto. Yhdysvallat, New York, IEE. <<https://ieeexplore.ieee.org/document/875998>>. Luettu 13.4.2021.
40. MiR History. 2020. Verkkoaineisto. MiR. <<https://www.mobile-industrial-robots.com/en/about-mir/mir-history/>>. Luettu 5.3.2021.
41. Optimize your workflows with the mobile robots from MiR. 2020. Verkkoaineisto. MiR. <<https://www.mobile-industrial-robots.com/en/>>. Luettu 05.03.2021.
42. Experienced computer scientist/software engineer. 2020. Verkkoaineisto. MiR. <<https://www.mobile-industrial-robots.com/en/career-old/experienced-computer-scientistsoftware-engineer/>>. Luettu 5.3.2021

