



AGCO Powerin uuden työntekijän koulutustehtävät ABB IRB 140 S4C+ -robotin käytön opeteluun

Benjamin Hyyppä

Opinnäytetyö
Helmikuu 2023

Konetekniikan tutkinto-ohjelma
Koneautomaatio

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Konetekniikan tutkinto-ohjelma
Koneautomaatio

HYYPPÄ, BENJAMIN:

AGCO Powerin uuden työntekijän koulutustehtävät ABB IRB 140 S4C+
-robotin käytön opetteluun

Opinnäytetyö 25 sivua, joista liitteitä 9 sivua
Helmikuu 2023

AGCO Powerin edustajat halusivat tehdä uusia työntekijöitä ja opiskelijoita varten Robotic Training Center -huoneen, jossa voidaan harjoitella ABB-robotin ja Fanuc-robotin käyttöä. Opinnäytetyön tavoitteena oli tehdä harjoitustehtävät ABB-robotille. Tehtävien suunnittelu aloitettiin tekemällä kysely muutamalle työntekijälle, jotka työskentelevät robottien parissa AGCO Powerin tiloissa. Kyselyn vastausten perusteella tehtiin harjoitustehtävät Robotic Training Centerin tiloihin. Harjoitustehtävien haasteet liittyivät ABB-robotin vanhaan ohjelmointijärjestelmään, jonka takia kaikki käskyt eivät toimi S4C+-ohjaintabletissa.

Opinnäytetyössä tehtiin neljä erilaista harjoitustehtävää: piirto-, pick and place-, MoveOffset- sekä I/O-kommunikaatiotehtävä. Kaikki nämä tehtävät tehdään S4C+-ohjaintabletilla. Tulevaisuudessa tehtäviä todennäköisesti kehitetään ja tehdään uusia, jotka liittyvät konenäköön ja ohjelmointiin tietokoneen avulla.

Haastavinta tässä työssä oli lähteiden löytäminen, sillä robotit ja niiden ohjelmat ovat nykyään parempia kuin vuoden 2005 aikoihin. Lähteinä hyödynnettiin opettajia sekä AGCO:n kokeneita robotin ohjelmoijia, joilla on kokemusta senaikaisten robottien ohjelmoinnista.

ABSTRACT

Tampere University of Applied Sciences
Degree Programme in Mechanical Engineering
Machine Automation

Hyypä Benjamin
Instructional Exercises for ABB IRB 140 S4C+ Robot for New Employees at
AGCO Power

Bachelor's thesis 25 pages, appendices 9 pages
February 2023

AGCO Power wanted to create a Robotic Training Center for their new employees and students to practice using the ABB robot and Fanuc robot. The objective of the thesis was to make training exercises for the ABB robot. The process was started by conducting a survey of a few employees who work with robots at the facilities of AGCO Power. Based on the answers, training exercises were developed for the Robotic Training Center room. The challenge with doing the training exercises was the old programming system of the ABB robot, which is why all the commands did not work on the S4C+ controller tablet.

The thesis consists of four different exercises: a drawing exercise, a pick and place exercise, a MoveOffset exercise and an I/O communication exercise. All these tasks were done with the S4C+ controller tablet. In the future, tasks will most likely be improved, and new ones will be created related to machine vision and computer based programming. The most challenging part of the thesis was finding sources since robots and their programs have improved from 2005.

Key words: abb-robot, training exercise, robotic training

SISÄLLYS

JOHDANTO	6
1 YLEISTÄ	7
1.1 Historiaa	7
1.2 AGCO Power	8
1.3 Robotic Training Center	9
1.4 Tavoite	10
2 ABB-ROBOTTI	12
2.1 ABB IRB 140 -robotti	12
2.2 Robotin kunnostus ja käyttöönotto	12
2.3 S4C+-ohjain	14
2.4 RAPID-ohjelmointikieli	16
2.4.1 Funktiot	16
2.4.2 Parametrit	16
2.4.3 Keskeytykset	17
3 HARJOITUSTEHTÄVÄT	18
3.1 Kysely	18
3.1.1 Vastaukset	19
3.1.2 Harjoitustehtävät	20
4 JOHTOPÄÄTÖKSET JA POHDINTA	24
LÄHTEET	25
LIITTEET	26
Liite 1. Turvallisuus	26
Liite 2. Uuden ohjelman aloitus	26
Liite 3. Paikkojen määrittäminen	27
Liite 4. Jatkoa paikkojen määrittämiselle	27
Liite 5. Ohjelman testaus	28
Liite 6. Lisähuomioita kappaleen käsittelyyn	28
Liite 7. Termejä	29
Liite 8. Piirtotehtävä	29
Liite 9. Vinkkejä piirtotehtävään	30
Liite 10. Tarvittavat lähdöt/tulot I/O-tehtävään	30
Liite 11. Esimerkkejä eri käskyistä	31
Liite 12. Esimerkkejä tarttuja ohjaukseen aliohjelmalla	31
Liite 13. Pick and Place -tehtävänanto	32
Liite 14. Miksi offset? Esimerkki	32
Liite 15. Offset-tehtävänanto	33

Liite 16. Apukuvia.....	33
Liite 17. Lisää funktioita	34

JOHDANTO

AGCO Powerista saatiin syksyllä 2021 opinnäytetyöaiheeksi suunnitella harjoitustehtäviä ABB- ja Fanuc-roboteille. Tätä projektia tehdään suurimmilta osin yhteistyössä TAMKin, Tredun ja AGCO Powerin kanssa. Tredun osuutena olivat Fanuc-robotin harjoitteet, ja tässä opintonäytetyössä tehtiin ABB-robotin harjoitustehtävät. Rajoituksina tehtäviin oli, että ne suunnitellaan vanhalle vuosimallia 2005 olevalle S4C+-ohjaimelle, minkä takia kaikki nykyiset RAPID-koodikielen käskyt eivät toimi, ja ettei tietokonetta tai konenäköä käytetä tehtävissä. Vaatimuksena on, että pysytään ainoastaan S4C+-ohjaimessa.

1 YLEISTÄ

1.1 Historiaa

ABB on yksi maailman johtavista robottivalmistajista. ABB tunnetaan innovatiivisista, luotettavista ja älykkäistä teollisuusroboteista, robottityösoluista ja ohjelmistoista. ABB on lähtöisin Sveitsistä, mutta nykyisin sen toiminta on laajentunut yli sataan maahan. (Robots Done Right 2022a.)

ABB tunnettiin ennen nimellä ASEA. Vuonna 1974 ASEA valmisti ensimmäisen robottinsa, IRB 6:n. IRB 6 oli maailman ensimmäinen robotti, jota ohjattiin sähköisellä mikrotietokoneella. Seuraavana vuonna 1975 ASEA automatisoi onnistuneesti kaarihitsaussovellukset IRB 6:lle. Kaarihitsausmenestyksen jälkeen ASEA kiinnitti huomionsa pistehitsausrobotin suunnitteluun, ja vuonna 1979 ASEA julkaisi IRB 60:n, josta tuli ensimmäinen sähköinen pistehitsauskone. (Robots Done Right 2022a.)

Laajennuttuaan Euroopassa 1970-luvulla ASEA siirtyi myös Japanin robottimarkkinoille vuonna 1982. Seuraavana vuonna ASEA julkaisi uuden robottiohjausjärjestelmän, S2:n, joka kykeni ohjaamaan jopa seitsemää akselia. Vuonna 1986 IRB 2000 julkaistiin. Tuolloin tämä nivelrobotti oli ainutlaatuinen, koska se käytti vaihtovirtamoottoreita. Vuonna 1988 ASEA yhdistyi BBC Brown Boveri Ltd:n kanssa ja perusti Asea Brown Boveri Ltd:n eli ABB:n. (Robots Done Right 2022a.)

ABB laajeni Yhdysvaltain robottimarkkinoille ostamalla Cincinnati Milacronin. Tämä mahdollisti ABB:n laajentumisen autoteollisuuteen pistehitsausrobottien avulla. Vuonna 1991 kehitettiin IRB 6000, josta tuli tuolloin tarkin ja nopein teollisuusrobotti pistehitsaukseen. Muutamaa vuotta myöhemmin vuonna 1998 ABB kehitti nopean FlexPicker-deltarobotin automaattisin keräily- ja robottipakkaussovelluksin. Samana vuonna ABB mullisti robottiohjelmoinnin RobotStudio-ohjelmistollaan, joka mahdollisti offline-simuloinnin. (Robots Done Right 2022a.)

ABB laajensi robottiautomaatiota 2000-luvulla julkistamalla useita menestyneimmistä malleistaan. Vuonna 2001 IRB 7600 julkaistiin, ja sillä oli tuolloin suurin hyötykuorma: 500 kg. Vuonna 2002 ABB halusi laajentaa robottityöskentelyä IRB 6600 -mallillaan, joka pystyy taipumaan täysin taaksepäin. Vuonna 2004 esiteltiin IRC5-ohjain, jolla voidaan ohjata neljää ABB-robottia samanaikaisesti. Vuosi 2005 oli yksi ABB:n suurimmista vuosista, kun se esitteli yli viisikymmentä uutta tuotetta mukaan lukien IRB 1600:n. Vuosikymmenen päätteeksi julkaistiin yksi ABB:n tunnetuimmista kuusiakselisista roboteista, IRB 120. (Robots Done Right 2022a.)

ABB toi markkinoille vuonna 2010 IRB 460:n, joka on suunniteltu erityisesti robottipakkaus käyttöön. Vuonna 2013 kehitettiin IRB 6700 -sarja ja vuonna 2015 esiteltiin ABB:n YuMi-yhteistyörobotti. YuMi on ensimmäinen yhteistyörobotti, jolla on kaksi käsivartta. (Robots Done Right 2022a.)

1.2 AGCO Power

Nokian Linnavuoreen rakennettiin lentomoottoritehdas vuonna 1942. Tehtaan päätehtävä oli lentomoottoreiden korjaustyö.

Ensimmäinen työ oli 30 kappaletta VL Myrsky II -hävittäjäkoneen rungon koonpano. Tämän jälkeen käynnistyi lentokonepotkureiden kunnostus, ja kesän lopussa toiminnan aloitti laakerivalimo. Tehtaan alkuperäiseen tarkoitukseen, lentomoottoreiden korjaustyöhön, päästiin vasta jatkosodan jälkeen vuonna 1947. (AGCO Power n.d.)

Sodan jälkeen tehdas siirtyi lähes täysin sotakorvaustuotantoon. Linnavuoreessa valmistettiin melkein kaikki sotakorvauksiin kuuluvat dieselmoottorit ja ilmakompressorit. Sotakorvauksia varten tehtiin 60 kappaletta June Munktell -raakaöljymoottoreita. Skandiaverken-lisenssillä valmistettiin dieselmoottoreita vetureita varten ja Atlas Dieselin lisenssillä neljää eri kompressorimallia. Sotakorvaustuotteista ainoastaan kompressoreita tuotettiin 1960-luvulle saakka. (AGCO Power n.d.)

Omaan tuotekehitykseen perustuvien dieselmoottorien valmistus aloitettiin vuonna 1946, jolloin tehtiin pieniä sarjoja 440- ja 648-mallin moottoreita lähinnä meri- ja generaattorikäyttöön. Osa niistä ehti jo sotakorvaustoimituksiin, ja sen jälkeen niitä toimitettiin vielä Neuvostoliittoon myytyihin laivoihin. Vuonna 1947 aloitettiin nelisylinterisen ottomoottorin tuotekehitys voimanlähteeksi Valmetin traktoreihin, jonka polttoaineena oli joko bensiini tai petroli. (AGCO Power n.d.)

1940-luvun loppupuolella oli aloitettu suuremman dieselmoottorin tuotekehitys Valtion Rautateille, jolla oli tarve kevyelle henkilöjunalle, johon tarvittiin dieselmoottori. Tämän moottorin toimituksen teki Linnavuoren tehdas. Tuohon käyttöön kehitettiin suora 12 litran kahdeksansylinterinen moottori, joka saavutti 180 hevosvoiman tehon. Jatkokehityksessä tästä tehtiin varavoimakäyttöön välijäähdyttimellä ja kahdella ahtimella varustettu moottori, joka saavutti 250 hevosvoiman tehon. (AGCO Power n.d.)

AGCO Power on nykyään johtava moottoriteknologian toimittaja, joka toimittaa moottoreita monille maailman johtaville traktoreiden ja muiden maatalouskoneiden valmistajille. AGCO Power valmistaa robotiikan avulla moderneja moottoreita kymmeniätuhansia vuodessa.

1.3 Robotic Training Center

AGCO Powerin ensimmäisessä tapaamisessa puhuttiin mahdollisesta huoneesta, jossa tulevat työntekijät ja Tredun tai muiden koulujen opiskelijat voisivat käydä opettelemassa robotin käyttöä simuloitussa työpaikkatilassa. Tapaamisessa jaettiin huoneen ympäristön suunnittelu ja itse harjoitustehtävien suunnittelu TAMKin opiskelijoille. Tätä koulutustilaa, jossa harjoitellaan ABB- ja Fanuc-robotin käyttöä, kutsutaan Robotic Training Center -huoneeksi. Kuvassa 1 on keskeneräinen Robotic Training Center.



Kuva 1: Keskeneräinen Robotic Training Center -koulutustila

1.4 Tavoite

Päätavoitteena oli suunnitella harjoitustehtävät ABB-robotille, joita tulevat koulutusryhmät voivat tarvittaessa jalostaa. Työhön kuitenkin myös kuului vanhan ABB IRB 140 -robotin huoltaminen ja uudelleenkäyttöönnotto. Robottia puhdistettiin pölystä ja muusta liasta, vanhat katkotut johdot vaihdettiin ja muodostettiin yhteys M2000-ohjausyksikön I/O-signaalien, S4C+-ohjaimen ja itse robotin välillä.

Harjoitustehtäviä tehtiin uusille työntekijöille, ja muu ryhmä keskittyi tekemään tehtäviä opiskelijoille. Harjoitustehtävät on tehty osittain avoimesti sen takia, että uudet työntekijät, jotka osaavat jo perusteet, voivat tehdä myös omia muutoksia ohjelmointitehtäviin. Ihan uusille ABB-käyttäjille on taas annettu ohjeistuksia ja kuvia avuksi. Harjoitustehtäviä rajoitettiin sillä tavalla asiakkaan pyynnöstä, että keskitytään ainoastaan S4C+-ohjaimen käyttöön. Konenäkötehtäviä ja tietokonetta ei haluttu ainakaan vielä alkuvaiheessa mukaan.

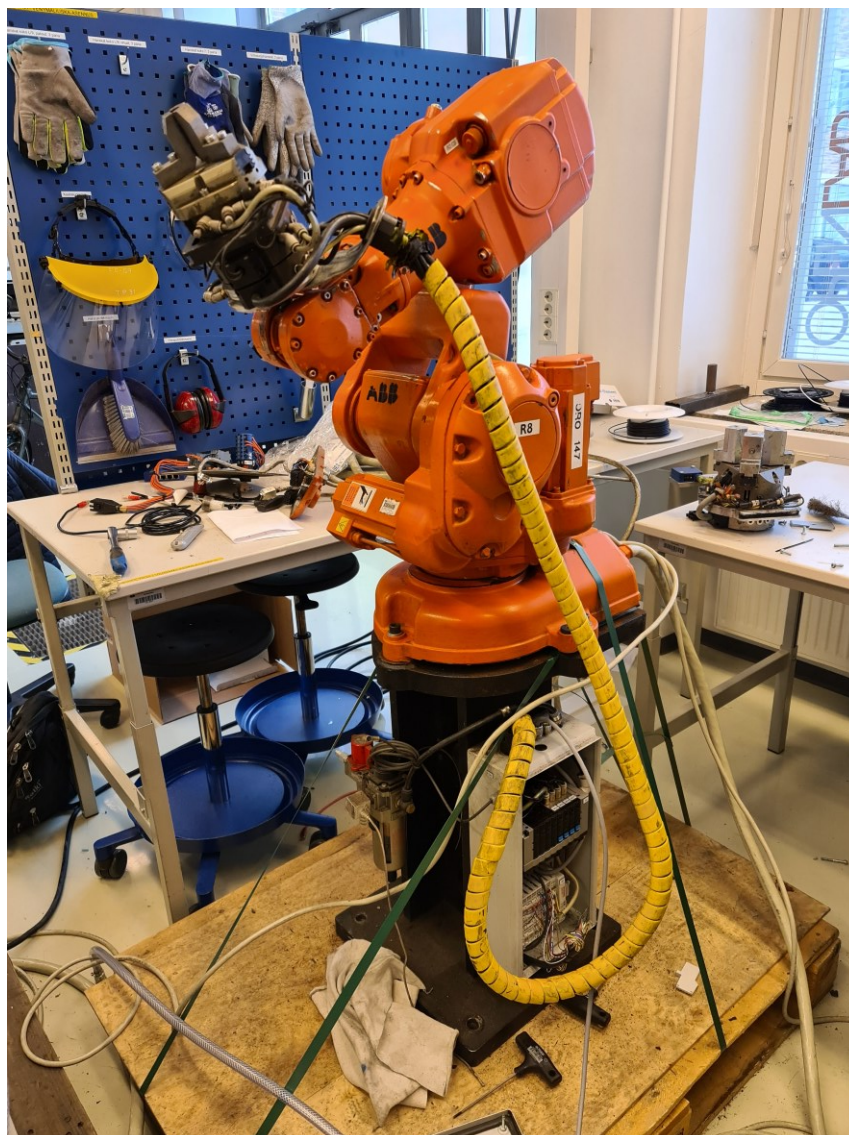
2 ABB-ROBOTTI

2.1 ABB IRB 140 -robotti

IRB 140 on 6-akselinen teollisuusrobotti, jonka hyötykuorma on 5 kg. Se on suunniteltu valmistusteollisuudelle, joka käyttää joustavaa robottipohjaista automaatiota. Robotilla on rakenne, joka on sovitettu joustavaan käyttöön ja voi kommunikoida laajasti ulkoisten järjestelmien kanssa. (Product specification – IRB 140 2022.)

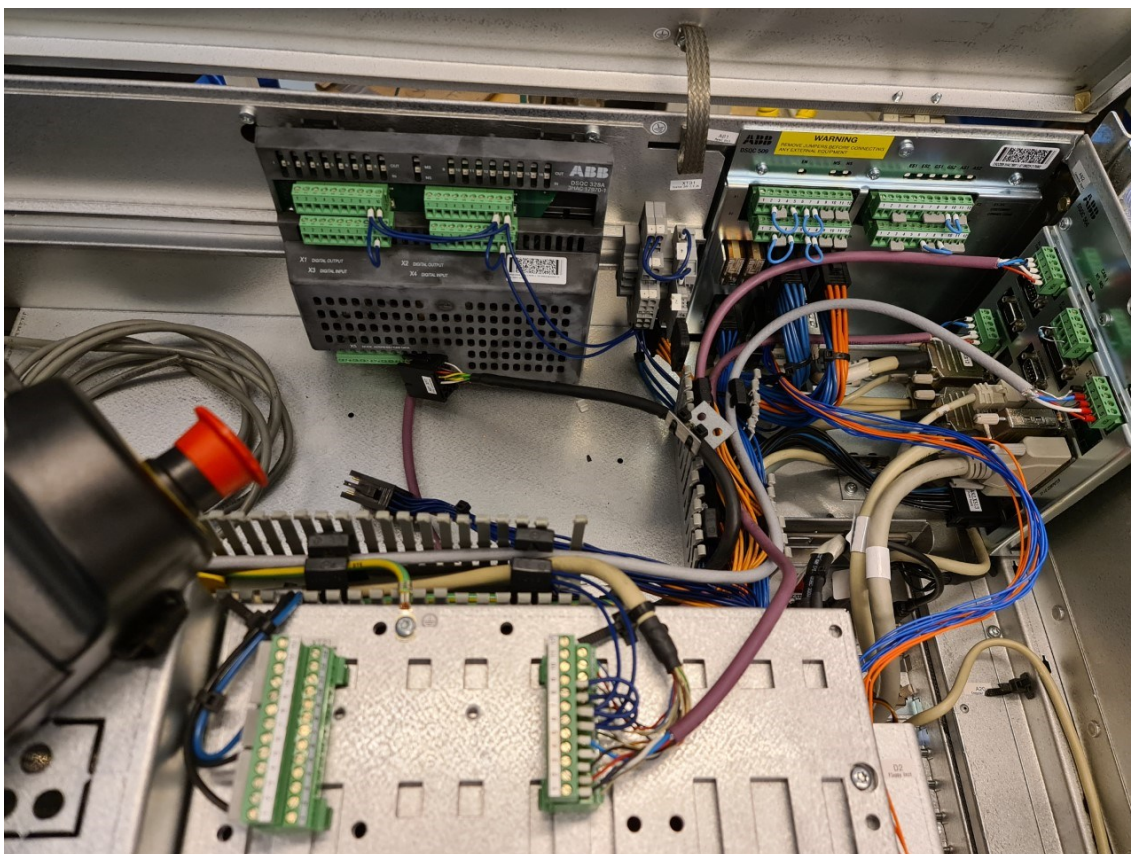
2.2 Robotin kunnostus ja käyttöönotto

Robottia oli ennen käytetty teollisuuslinjassa. Robotista alkuperäiset kaapelit oli katkottu leikkaamalla. Robotti siivottiin pölystä ja muusta liasta kunnostuksen aikana. Vaihdetut kaapelit olivat Device Net -kaapeli ja CAN2-väyläkaapeli, jolla ohjattiin lähtö- ja tuloarvoja robotissa. Tämän jälkeen piti saada I/O-arvot ja väyläkommunikaatiot toimimaan. Kuvassa 2 on juuri siivottu robotti liasta ja menossa on kaapeleiden ja tarttujan asettaminen.



Kuva 2: ABB-robotin kunnostus

Ensimmäinen ongelma tuli robotin uudelleenmäärittelyn aikana, kun yritettiin saada sitä toimimaan S4C+-ohjaimen avulla. Robotilla pystyi liikuttamaan kaikkia akseleita, mutta tarttujaa ei saatu suljettua ja avattua, mikä tarkoitti ongelmaa I/O-kommunikaatiossa robotin ja ohjauksen välillä. Ongelma löytyi johdotuksesta, ja pari johtoa oli vain asetettu väärinpäin, eikä robotista sen takia löytynyt I/O-arvoja. Kuvan 3 oikealla puolella on M2000:een asetettu violetti Device Net -kaapeli ja harmaa kaapeli, jolla saadaan yhteys I/O-arvoihin.



Kuva 3: Uudet kaapelit asetettu M2000:een

2.3 S4C+-ohjain

ABB S4C+ -ohjain on teollisuusrobottiohjain, jota käytettiin vuosina 2000–2004 robottien ohjaukseen. S4C+-ohjaimesta käytetään myös nimitystä ABB M2000 -ohjain. S4C+-ohjaimella voidaan ohjata myös vanhempia robotteja, jotka on yhdistetty aikaisemman sukupolven ohjaimiin. Käytetyt ABB S4C+ -robotit ovat edullinen tapa integroida sovelluksia, kuten kaarihitsaus tai materiaalinkäsittely. S4C+-ohjaimella saadaan lähetettyä ulkoisen ohjaimen M2000 avulla käskyt ABB-robottiin. (Robots Done Right 2022b.) Kuvassa 4 on S4C+-ohjain, jota tul- laan käyttämään harjoitustehtävien aikana. Kuvassa 5 on ulkoinen ohjain M2000, jossa S4C+-ohjain ja ABB IRB 140 -robotti ovat kiinni.



Kuva 4: S4C+-ohjain



Kuva 5: Ulkoinen ohjain M2000

2.4 RAPID-ohjelmointikieli

RAPID on korkean tason ohjelmointikieli, jota käytetään ABB:n teollisuusrobottien ohjaamiseen. ABB otti RAPIDin käyttöön yhdessä S4-ohjausjärjestelmän kanssa vuonna 1994.

2.4.1 Funktiot

RAPID-kielessä on integroituja toimintoja, joita robotti voi käyttää. Näiden integroitujen funktioiden avulla robotti voi liikkua eri paikkoihin. Esimerkiksi funktio "MoveL" on lineaarinen siirto ohjeistettuun pisteeseen. Osa funktioista suorittaa matemaattisia laskutoimituksia tai säätelee sitä, kuinka herkkä robotti on ulkoisille voimille. Näitä funktioita kutsutaan samalla tavalla kuin muita ST- tai RAPID-funktioita, joissa pääjuurifunktiota ei voida poistaa vaan sitä käytetään muiden funktioiden kutsumiseen. Voidaan myös tehdä omia funktioita, jotka palauttavat tietyn datatyypin arvon. (Dietrich 2022.)

2.4.2 Parametrit

Funktion tai metodin kutsun parametri on muuttuja, joka lähetetään kutsun mukana, kuten esimerkiksi "CalculateNum(Num1, Num2)". Num1 ja Num2 ovat parametreja; niiden arvoa ja tietotyyppejä käytetään CalculateNum-funktiossa. Parametrit ovat käytettävissä RAPID-ohjelmoinnissa, ja ne voivat pienentää ohjelman kokoa, kun yksi muuttuja voi palvella useampaa kuin yhtä tarkoitusta. (Dietrich 2022.)

Esimerkkinä voidaan mainita sovellus, joka vaatii useita opetettuja kantoja, mutta jossa käytetään kerrallaan vain yhtä sijaintia. Funktio laskee, mitä sijaintia on käytettävä, ja kutsuu sitten liikemenetelmää, jonka parametrina on sijainti. Kun tätä metodia kutsutaan, robotti siirtyy parametriin tallennettuun sijaintiin. Parametri kierrätetään uutta metodikutsua varten, kun uusi sijainti on laskettu. Kun

käytetään yhtä metodia ja lähetetään sijainti parametrin kautta, robotin liikuttamiseen tarvittava koodi on hyvin tiivistetty. (Dietrich 2022.)

2.4.3 Keskeytykset

Keskeytys on ohjelmoinnissa signaali tai tapahtuma, joka käskee prosessoria suorittamaan toisen tehtävän, kun nykyinen tehtävä on suoritettu loppuun, riippumatta siitä, mikä vaihe ohjelmassa on menossa. RAPID-ohjelmassa ohjelmointikatkoksia käytetään, kun robotti tulee alueelle tai kun tarttujasta tulee signaali. Näillä signaaleilla voidaan suorittaa lyhyt erityinen koodi tavanomaisen syklisen tehtävän ulkopuolella. (Dietrich 2022.)

3 HARJOITUSTEHTÄVÄT

3.1 Kysely

Runsaiden rajoituksien takia kyseessä on laadullinen tutkimus eikä määrällinen. Kun harjoitustehtävät tulivat aiheeksi ensimmäisessä tapaamisessa AGCO Powerin kanssa, aloitettiin kehittämään ideoita mahdollisista harjoitustehtävistä, kunnes tultiin siihen tulokseen, että olisi paljon parempi vain tehdä kysely AGCO Powerin työntekijöille, jotka työskentelevät robottien parissa. Ensimmäiseksi päädyttiin kysymään, mitä toivomuksia heillä olisi harjoitustehtävistä ja mitä haasteita heillä on robottien ohjelmoinnissa.

Ennen harjoitusten suunnittelua toteutettiin kysely seitsemälle työntekijälle, jotka työskentelevät ainakin osittain robottien parissa. Heidän vastauksiensa mukaan suunniteltiin tarvittavat harjoitustehtävät ABB-robotille.

Kysymykset olivat:

1. Käytätkö työssäsi robottia säännöllisesti? Hallitsetko robotin käytön nykytyössäsi?
2. Mitä toimenpiteitä teet robotin käsiohjaimella?
3. Tarvitsetko lisäkoulutusta ja jos kyllä niin kuvaile mahdollisimman kattavasti minkälaista. Alla listattuna apusanoja aiheesta:

Turvallinen käyttö	Robotin rakenne	Koordinaatistot	Ohjelmarakenne ja käskyt
Ohjelman valinta ja vaihtaminen	Pienet ohjelmanmuutokset	Uuden ohjelman tekeminen	Ongelmatilanteista toipuminen

4. 5-halliin valmistuu vuonna 2022 robotiikan koulutustila. Mitä odotuksia sinulla on tältä tilalta ja koulutuksen sisällöltä?
5. Muuta viestiä työnantajalle koskien robottikoulutusta?

Kaksi viimeistä kysymystä olivat yrityksen haluamia, kun taas kolme ensimmäistä kysymystä antoivat harjoitteiden tekemiseen liittyvää materiaalia. Tosin yksi vastaajista vastasi hyvin 4. kysymykseen, josta oli hyötyä harjoitteiden tekemiseen, kun taas eräs toinen vastasi haluavansa pulla ja kahvi -tarjoilut koulutustilan valmistuttua.

3.1.1 Vastaukset

Vastauksien analysointi lähti liikkeelle vastausten kokoamisesta muistioon. Vastauksista selvitettiin, mitkä ongelmat robotin ohjauksessa nousevat esille. Ongelmakohtista suunniteltiin harjoitustehtävät.

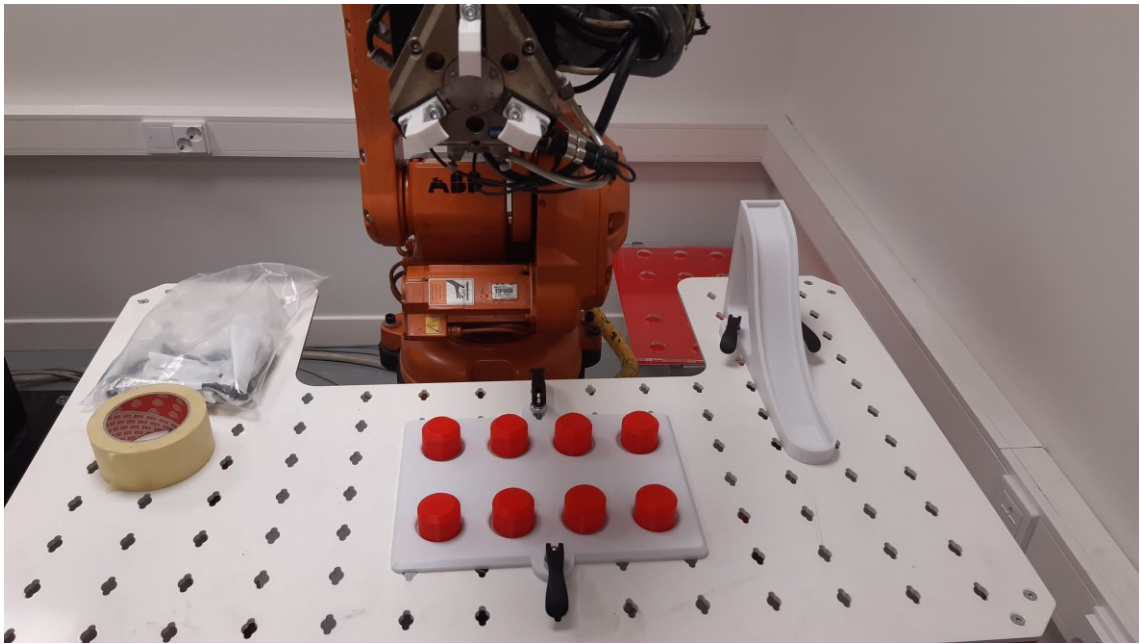
Kaikkia kyselyn vastauksia ei voinut tässä työssä hyödyntää. Seuraavaan alla olevaan taulukkoon on kirjattu vastaukset, joista oli hyötyä harjoitteiden tekoon. Vastaukset on esitetty samalla tavalla, miten vastaajat olivat kirjoittaneet kyselypaperiin. Vastaukset on esitetty taulukossa 1.

TAULUKKO 1: Vastaukset kyselyyn

<p>2.</p> <ul style="list-style-type: none"> - Häiriöiden purku + tulotietojen muutokset - 3x (Ajaa robottia virhetilanteista pois häiriökohdista käsiohjauksella) yms, muuttamalla parametrilukuja ja 1/0 käskyjä. 1/0 in/out tarkasteluja/käyttöjä, normitilanteessa datamuutoksia ja ohjelman vaihtoja. - Laskureiden nollaus <p>3.</p> <ul style="list-style-type: none"> - Koordinaatistot, 3x (pienet ohjelmanmuutokset), ongelmatilanteista toimuminen, ohjelmarakenne ja käskyt (ympyröity) - Jos menee "jatkuvasti" päin tai väärin ym. miten muutos tehdään. Siinä samalla ohjelman rakenne - Jos robotin paikka tai tarttujen liikku, tai muuta vastaava ja haku/vientipaikat muuttuvat. Olisi kätevää itse osata säätää paikkoja. <p>4.</p> <ul style="list-style-type: none"> - Koulutuksesta, että olisi mahdollisimman paljon, joita tehdään tehtaalla (liittyvää) - Hieman syventävää peruskäytön opetusta sekä pientä ohjelmanmuutosten tekoa.

3.1.2 Harjoitustehtävät

AGCO Powerin edustajan kanssa päätettiin, että kaikki harjoitustehtävät laetaan Powerpoint-dioihin. Powerpoint-diat ovat nähtävillä liitteissä 1–17. Kuvassa 6 on ABB-robotin koulutus piste, jossa harjoitustehtäviä tehdään.



Kuva 6: ABB-robotin koulutuspaikka

Ohjelmantekodiat tehtiin sellaisille käyttäjille, jotka eivät ole koskaan käyttäneet robottia tai eivät ole ennen käyttäneet S4C+-ohjainta (liitteet 2–6). Seuraavassa diassa (liite 7) käydään lyhyesti läpi termejä ja datatyyppejä, joita käytetään usein robotin ohjelmoinnin aikana ja seuraavissa tehtävissä.

Piirtotehtävän tarkoituksena on tutustua liikekäskyihin, jotta saadaan kuva siitä, mitä niillä voidaan tehdä. Piirtotehtävä on ensimmäisenä, koska sillä ei kuormiteta opiskelijaa liikaa teorialla vaan pyritään saamaan koulutukselle mukava alku. Tehtävässä on tunnettu työkalupiste ja koordinaatisto, joita ei tässä tarvitse vielä tehdä itse. Harjoitustehtävässä on mallikuva, mutta halutessaan siinä voi myös piirtää oman kuvion. MoveC-käskyä käytetään kaarien ohjelmointiin. MoveOffs-funktiota käytetään piirtämisen kanssa avuksi. Rata-ajo-ohjelmissa kannattaa käyttää in-line-pisteitä, jotka merkitään funktioon asteriskilla (*) (liite 8). Tästä tehtävästä saadaan hyvä käsitys robotin liikuttamisesta sekä liikefunktioista ja MoveOffs-funktiosta.

Seuraavassa diassa (liite 9) käydään läpi tussin työkalupisteen määrittely ja käyttäjäkoordinaatiston määrittely. Tooldata on työkaludata (esimerkiksi tarttuja), ja se määritellään ohjelmadatasta. Käyttäjäkoordinaatisto on Wobjdata eli work object data ja sijaitsee samassa valikossa. Työkalupisteellä näytetään kaksi pistettä X-akselilta ja yksi piste Y-akselilta. Työkohteessa on kaksi sisäkkäistä

koordinaatiota: UserFrame ja ObjectFrame. Oikean työkalun valintaan tulee kiinnittää huomiota.

I/O-kommunikaatio ei ole suoranaisesti harjoitustehtävä robotin käyttöön, mutta kyseinen taito on hyvä hallita. Sillä voidaan hallita tarttujan toimintoja sekä valo- ja, joilla annetaan käyttäjälle merkki, onko ulkoinen tarttuja vapaa. Tällä tarttujalla simuloidaan työstöpisteitä AGCO Powerin työpaikoilla.

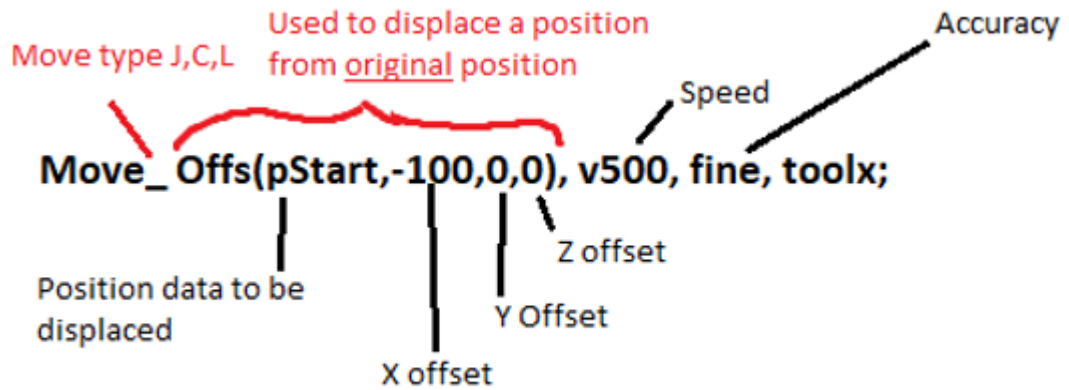
Liitteessä 10 on esitetty tarvittavat lähdöt ja tulot I/O-kommunikaatiotehtävään. Lähtöarvoista tarvitaan robotin tarttuja auki- ja robotin tarttuja kiinni -tilanteet. Tosin tämä riippuu siitä, onko robotissa jousipalautteinen venttiili 5/2 vai kaksikelainen venttiili, joka tarvitsee kahta signaalia toimiakseen.

Liitteessä 11 käydään läpi tehtävänanto ja annetaan esimerkkejä käskyistä. Reset-käskyä käytetään digitaalisen lähtösignaalin arvon nollaamiseen ja set-käskyä lähtösignaalin asettamiseen arvoksi yksi. SetDO-käskyä käytetään digitaalisen lähtösignaaliarvon muuttamiseen joko aikaviiveellä, synkronoinnilla tai ilman kumpaakaan näistä. WaitDI- eli wait digital input -käskyä käytetään, kun halutaan odottaa, kunnes digitaalitulo on asetettu.

Liitteessä 12 tarttujan ohjaus -dia on pieni teoria- ja esimerkkiosuus aliohjelmas- ta. Yhtenä esimerkkinä käytetään irrotusohjelmaa, jossa odotusaika on vakio (1 sekunti), ja toisena esimerkkinä käytetään tilannetta, jossa käyttäjä saa määritellä tartunta-ajan rivikohtaisesti.

Pick and place -tehtävästä saa hyvät perustiedot ABB-robotin ohjelmointiin (liite 13). Tehtävässä käytetään manuaaliohjausta ja tutustutaan, miten robotti käyt- täytyy käsiohjauksella. Tässä tehtävässä käytetään MoveJ- tai MoveL-käskyjä kappaleiden siirtämiseen. Aikaa kuluu eniten robotin siirtoon, koska tämä pitää tehdä käyttäen ohjaimen sauvaohjainta ja käsiajokoordinaatioita. Harjoitustehtävä on suositeltavaa suorittaa useita kertoja. Tämä tehtävä oli tarkoitus laittaa ensimmäiseksi harjoitustehtäväksi, mutta koska teoriaa tuli paljon, järjestystä vaihdettiin.

Offset-tehtävässä simuloidaan tilannetta, jossa robotin sijainti tai kappaleiden siirtopaikka vaihtuu, jolloin tehdään nopea muutos ohjelmaan. Tarkoituksena on käydä läpi offset-funktio, josta voi olla apua tällaisissa tilanteissa. Liitteessä 14 annetaan pitkä esimerkki kyseisestä tilanteesta ja tämän jälkeen esimerkkifunktorivit. Tämän jälkeen annetaan offset-tehtävänanto. Malliksi on annettu kuva 7, jossa käydään läpi argumentit.



Kuva 7: Mallikuva MoveOffs-funktiosta.

Tehtävänä tässä on, että robotti vie ja palauttaa koko paletillisen kappaleita pakkaan ja takaisin. Koska mitat tulee ohjelmaan asettaa millimetreinä, jonkinlaista mittaaakin on suositeltavaa käyttää. Kappaleet on kuitenkin asetettu säännöllisesti siten, että kahden vierekkäisen kappaleen välisen etäisyyden mittaaminen riittää, minkä jälkeen selviää yksinkertaisella matematiikalla. Lopuksi diassa on annettu pieniä vinkkejä (Liite 15). Viimeisessä MoveOffs-harjoitustehtävädiassa on apukuvia harjoittelijoille (Liite 16).

Viimeisessä diassa (Liite 17) on käskyjä, joista voi olla apua työssä. Nämä käskyt on koottu sen takia, että kiinnostuneet voivat käydä ne läpi harjoituksissa. Enimmäkseen ne on kuitenkin tehty harjoitustehtävien jatkokehitystä varten.

4 JOHTOPÄÄTÖKSET JA POHDINTA

Tässä työssä harjoitustehtävät tehtiin hyvin kyselyn vastauksien mukaan. Opiskelijoille ja uusille työntekijöille, jotka eivät ennen ole työskennelleet robottien parissa, harjoitustehtävät antavat hyvät alkutiedot robotin ohjelmoinnista. Aikaisempia harjoitustehtäviä ei ollut, joten laadittiin seuraavat harjoitustehtävät: piir-totehtävä, pick and place -tehtävä, jossa siirretään kappaleita manuaalisesti, I/O-kommunikaatio tehtävä, jossa harjoitellaan lähtö- ja tuloarvojen käyttöä ja asettamista, ja viimeisenä MoveOffset-tehtävä, jossa simuloidaan työpaikan tilanteita, joissa robotin paikka vaihtuu tai kappaleiden siirtopaikka vaihtuu. Ongelmana työn aikana oli harjoitusten käytännön testaamisen puuttuminen. Harjoitustehtävien testaaminen käytännössä ei ollut vielä mahdollista, sillä Robotic Training Center -tila ei ollut valmis eikä TAMKIN tiloissa voi uudempia robotteja käyttää, sillä niissä on uudempia ja parempia ohjelmointijärjestelmiä. Robottien parissa työskentelevä opettaja kävi tehtävät läpi.

Työn loppuvaiheessa tuli tieto, että uusi opiskelijaryhmä jatkaa siitä, mihin tässä työssä jäätiin. Kyseinen ryhmä todennäköisesti kehittää uusia tehtäviä ja jaloistaa tässä työssä esiteltyjä tehtäviä. Näissä tehtävissä saatetaan käydä läpi konenäköä ja robotin ohjelmoinnista tietokoneen kautta. Loppujen lopuksi työ onnistui hyvin, vaikka projektissa kului paljon aikaa kommunikaation puutteen takia, mutta näissä isoissa projekteissa tämä on yleisestään antaa hyvää esimerkkiä arkipäivien työpaikkojen projektitoista.

LÄHTEET

ABB. 2022. Product specification – IRB140.

<https://library.e.abb.com/public/a7121292272d40a9992a50745fdaa3b2/3HAC041346%20PS%20IRB%20140-en.pdf>

ABB Robotics. 2010. Technical reference manual RAPID Instructions, Functions and Data types.

https://library.e.abb.com/public/688894b98123f87bc1257cc50044e809/Technical%20reference%20manual_RAPID_3HAC16581-1_revJ_en.pdf

AGCO Power. n.d. Historia. Viitattu 12.1.2023.

<https://www.agcopower.com/fi/yritys/historia/>

Dietrich, S. 2022. Introduction to ABB Robot Programming Language. Control Automation 22.2.2022. Viitattu 4.2.2023

<https://control.com/technical-articles/introduction-to-abb-robot-programming-language/>

Robots Done Right. 2022a. History of ABB robots. Viitattu 28.10.2022.

<https://robotsdoneright.com/Articles/history-of-abb-robots.html>

Robots Done Right, 2022b. ABB S4C+ Controller. Viitattu 4.2.2023

<https://robotsdoneright.com/ABB/ABB-S4C+-Controller.html>

LIITTEET

Liite 1. Turvallisuus

TURVALLISUUS

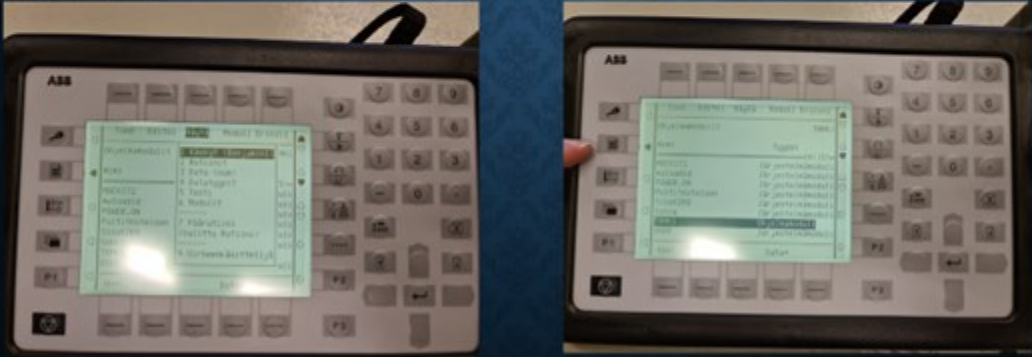
- Hätäseis painike löytyy M2000:sta. Punainen painike (Kuva)
- Kun työskentelet robottien kanssa niin pidä hyvä turvaväli roboteista.



Liite 2. Uuden ohjelman aloitus

UUSI MODULI JA RUTIINI

- Tee uusi moduli.
- Paina näytä > 6 moduliit
- Kun uusi moduli on tehty. Avaa se ja tee uusi rutiini.



Liite 3. Paikkojen määrittäminen

PAIKKOJEN MÄÄRITYS

- Valitse, mitä haluat liikuttaa: tarttujaa, vartta vai koko robottia (kuvassa punaisella ympyröity)
- Liikuta robotti haluamaasi koordinaattiin ja aseta se ohjelmaan.

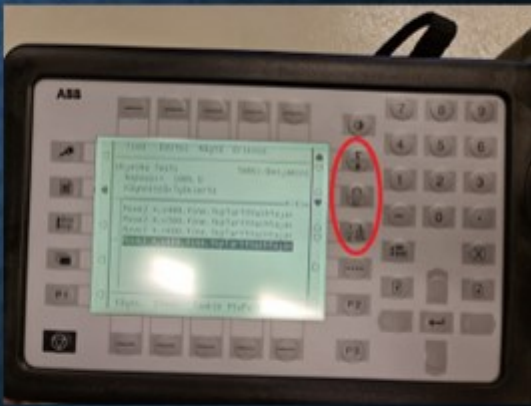
Suurin ero MoveJ, MoveL ja MoveC liikekäskyissä on:

- MoveJ: ei-lineaarinen liike. Käytetään yleensä ensimmäisenä liikkeenä
- MoveL: Lineaarinen liike
- MoveC: Pyöreä liike

MoveC ja MoveL käytetään silloin, kun robotti tekee sen päätehtävänsä.

Esim. MoveJ ToJointPos, Speed, Zone, Tool

- Näyttää suunnilleen tältä MoveJ (paikan nimi), v1000, z100, (Tool nimi)

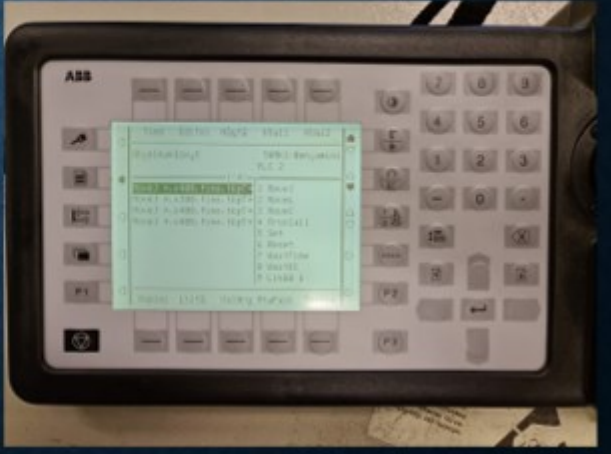


Liite 4. Jatkoa paikkojen määrittämiselle

- Set ja Reset Käytetään robotin lähtöjen ohjaamiseen ja näitä voidaan myös käyttää tarttujan liikkeisiin. Käytetään aliohjelmia tarttujan ohjaamiseen.

(Set=Kiinni, Reset=Auki).

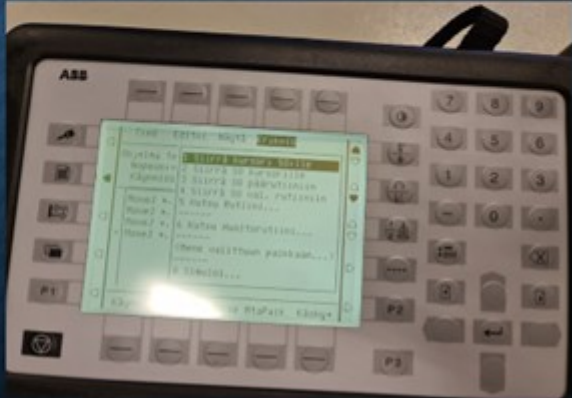
- Valitsemalla käsky komentorivistä niin voit säätää robotin vauhtia ja tarkkuutta.



Liite 5. Ohjelman testaus

OHJELMAN TESTAUS

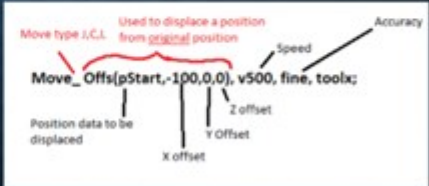
- Kun ohjelma on valmis testattavaksi, valitsemalla **Erikois >** siirrä kursori S0:lle eli ohjelman alkuun.
- Aja ohjelma ensin läpi askel kerrallaan, jos ei ollut ongelmia, niin aja sen jälkeen koko ohjelma putkeen.



Liite 6. Lisähuomioita kappaleen käsittelyyn

KAPPALEEN NOSTOSTA HUOMIOITA

- Kuvasta näkee Accuracy "fine" kohdan. Sitä yleensä kannattaa käyttää kappaletta nostaessa.
- Ellei sitä käytä niin tarttuja menee kiinni asentoon ennen kuin kappale on nosto asemassa. Tarttuja osuu kappaleeseen kiinni asennossa, mikä saattaa aiheuttaa vahinkoa kappaleeseen.



Liite 7. Termejä

TERMIT

- Rivipiste (inline) : tarkoittaa sitä, että on komennon mukana piste/pose
- Nimetty piste: on ennalta määrätty/asetettu pose. Esimerkiksi: p10.

Datatyypit

- Bool: Arvot voivat olla TOSI tai EPÄTOSI.
- Num: Käytetään numeroarvoille; esimerkiksi laskurit.

Liite 8. Piirtotehtävä

PIIRTOTEHTÄVÄ

- Piirrä haluamasi kuvio. Mallina kuva:
- Käytä MoveC funktiota, sillä saadaan kulmat pyöreiksi kuviossa.
- Käytetään Move_Offsia ja määritetään sen avulla koordinaatteja, jotta kynä pysyy paperissa.
- Rata-ajo ohjelmissa kannattaa käyttää * (inline) pisteitä
- Seuraavalla sivulla: Tussin työkalupisteen määrittäminen ja miten tehdään käyttäjäkoordinaatisto.

Muita!
Kaareissa (MoveC) kaksi pistettä samalla rivillä

Lähtö

Tee nämä pisteet ensin fine tarkkuudella.
Muekaa testauksen jälkeen fine-9z10.

Testauksen jälkeen kopio rutiini uudelle nimelle ja testaa pisteiden paikan päivittämistä

Liite 9. Vinkkejä piirtotehtävään

PIIRTOTEHTÄVÄ

Tussin työkalupisteen määrittäminen ja tehdään käyttäjäkoordinaatisto.

- Tooldata on työkalu (esim. tarttuja) ja se määritellään ohjelmadatassa. tapana voi olla 4-6 pistettä. 4 pistettä siirtää vaan paikkaa, 4+Z määrittää pisteen lisäksi Z akselin suunnan ja 4+Z+X saa määritettyä työkalun täydellisesti.
- Käyttäjäkoordinaatisto on Wobjdata (Wobj=work object) ja löytyy samasta valikosta. Työkalupisteellä näytetään 2 pistettä X akselilta ja yksi piste Y akselilta. Työkohteessa on kaksi sisäkkäistä koordinaatiosta. UserFrame ja ObjectFrame.
- Funktion perässä, kun ontodata ja wobjdata. Kannattaa tarkistaa, että onko valittuna oikea työkalu. Esimerkiksi:
 - `MoveJ *, v1000, z50, toolTarttuja1\WObj=wobj;`

Liite 10. Tarvittavat lähdöt/tulot I/O-tehtävään

I/O KOMMUNIKAATIO

Tarvitaan seuraavat lähdöt ja tulot:

Lähdöt

- Robotin tarttuja auki ja tarttuja kiinni. Riippuu onko 5/2 jousipalautteinen venttiili vai kahdella kelalla toimiva venttiili.
- "Sorvi" auki ja kiinni
- 1-2 valoa, joita voi vapaasti robotin ohjelmasta ohjata.

Tulot

- 1-2 painonappia, joita voi vapaasti robotin ohjelmasta ohjata.
- Tarttujan tilatieto (saattaa olla valmiina)
- Sorvin tilatieto eli punainen/vihreä valo (saako tuoda kappaleen)

Liite 11. Esimerkkejä eri käskyistä

I/O KOMMUNIKAATIO

- Tehtävänä on asettaa punainen/vihreä valo näyttämään ulkoisen tarttujan "sorvin" tilaa. Näyttää punaista valoa, kun "sorvi" on varattuna ja vihreää, kun se on vapaa.
- Tehtävänä on myös saada robottiin kytketyt ulkoiset laitteet toimimaan. S4C+ ohjaimella ohjelmaan tulee lisätä käskyjä lähtöjen ohjaukseen, esimerkiksi: Set ja Reset, SetDO (Tarttuja auki ja tarttuja kiinni) Sitten, että robotti osaa odottaa tai lukea antureiden tiloja. WaitDI. (Onko ulkoinen "sorvi" vapaa?)

Käskyt

- Set ja Reset Resettoimintoa käytetään digitaalisen lähtösignaalin arvon nollaamiseen ja Set-toimintoa käytetään digitaalisen lähtösignaalin asettamiseen arvoksi yksi.
 - Set do15; signaali do15 asetetaan arvo 1.
 - Reset do15; signaali do15 asetetaan arvo 0.
- SetDO käytetään digitaalisen lähtösignaali arvon muuttamiseen joko aikaviveellä, synkronoinnilla tai ilman.
 - SetDO do15, 1; signaali do15 asetetaan arvo 1.
- WaitDI (Wait Digital Input): käytetään, kun halutaan odottaa kunnes digitaalirulo on asetettu
 - WaitDI di4, 1; Ohjelman suoritus jatkuu vasta, kun di4-tulo on asetettu.

Liite 12. Esimerkkejä tarttuja ohjaukseen aliohjelmalla

TARTTUJAN OHJAUS

- Tarttujaa ohjataan aliohjelmilla.
- Tuliaan tarvitsemaan lista Inputeista ja Outputeista.

Tässä on esimerkki irrotusohjelmasta, jossa odotusaika on vakio 1s.

```
PROC Irrota()
  Set do15Tarrain1; Tähän se lähtö, joka on oikeassa robotissa.
  WaitTime 1;
ENDPROC
```

- Tässä esimerkki tartuntaohjelmasta, jossa käyttäjä saa määrittellä tartunta -ajan riviakohtaisesti

```
PROC Tartu(num Viiveaika)
  Reset do15Tarrain1; ! Tähän se lähtö, joka on oikeassa robotissa.
  WaitTime Viiveaika;
ENDPROC
```

Liite 13. Pick and Place -tehtävänanto

PICK AND PLACE TEHTÄVÄ

- Tässä tehtävässä olisi hyvä olla kappale paletti tai kaksi, mikä helpottaisi kappaleitten asettamista oikeisiin paikkoihin.
- Tee uusi moduuli/rutiini.
- Tee ohjelma, missä nostat ja siirrät kappaleet paikasta A paikkaan B. Liikuta robottia manuaalisesti paikkoihin ja aseta pisteet/poset perusliike käskyillä MoveJ ja MoveL. Ei tarvitse tehdä koko palettilisen määrä kappaleita. Tee tämä 2-4 kertaa oman tuntemuksen mukaan.
- Käytä aliohjelmaa tarttujan avaamiseen ja sulkemiseen.

Liite 14. Miksi offset? Esimerkki

MIKSI OFFSET

- MoveOffs käskyjä käytetään yleensä aina pick and place tapaisissa tehtävissä. Sillä se on nopeampaa säätää pisteitä numeerisesti ja kopiaamalla MoveOffs funktio kuin manuaalisesti ohjaamalla robotti poiminta/laskenta paikkoihin monta kertaa.
- Esimerkiksi: Ota ensin kopio välipiste MoveOffs funktiasta (1-3 cm kappaleen yläpuolella (Kappaleen nosto ja laskenta erikseen). Pisteet otetaan siitä tilanteesta, kun tarttuja pääsee tarttumaan kiinni kappaleeseen tai päästää irti kappaleesta. Otetaan siis pisteet ennen nosto/lasku funktiota ylös ja tehdään siitä kopio ja tehdään sama myös kappaleen laskenta offsetilla. Kun kopioit Offsetteja niin muista laittaa ne oikeisiin kohtiin ohjelmassa. Muista myös asettaa "fine" tarkkuudeksi välttääksesi tarttuja ja kappale vahinkoja.

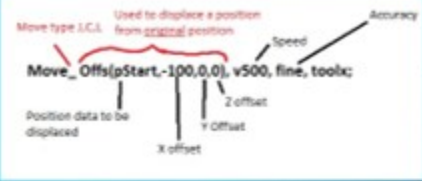
```

MoveLOffs(p10, 0, 0, 0), v500, fine, toolx; [Välipiste nostamiselle]
MoveLOffs(p10, 0, 0, -25), v500, fine, toolx; [Tarttuja on asemassa, voidaan tarttua kappaleeseen]
TarttujaKiinni (aliohjelma)
MoveLOffs(p10, 0, 0, 100), v500, fine, toolx; [välipiste korkealle varmuuden vuoksi]
MoveLOffs(p20, 0, 0, 0), v500, fine, toolx; [Välipiste kpl:n laskuun]
MoveLOffs(p20, 0, 0, -25), v500, fine, toolx; [Tarttuja on asemassa, voidaan aukaista tarttuja]
TarttujaAuki (aliohjelma)
MoveLOffs(p20, 0, 0, 100), v500, fine, toolx; [Välipiste korkealle varmuuden vuoksi]
MoveLOffs(p10, 50, 0, 0), v500, fine, toolx; [Välipiste nostoon. Muutetaan x ja/tai y koordinaatteja toisen kappaleen nostoon]
MoveLOffs(p10, 50, 0, -25), v500, fine, toolx; [Tarttuja on asemassa, voidaan tarttua kappaleeseen]
TarttujaKiinni (aliohjelma)..... Tämän logiikan mukaan tehdään näin seuraaville kappaleille paletissa.

```

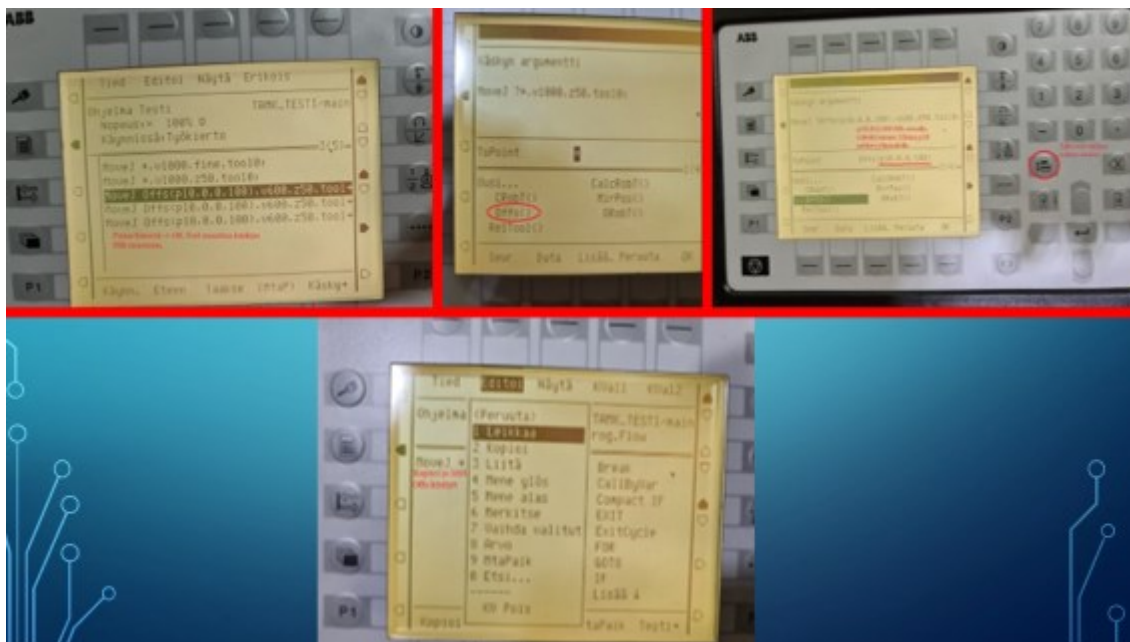

Liite 15. Offset-tehtävänanto

OFFSET_TEHTÄVÄ

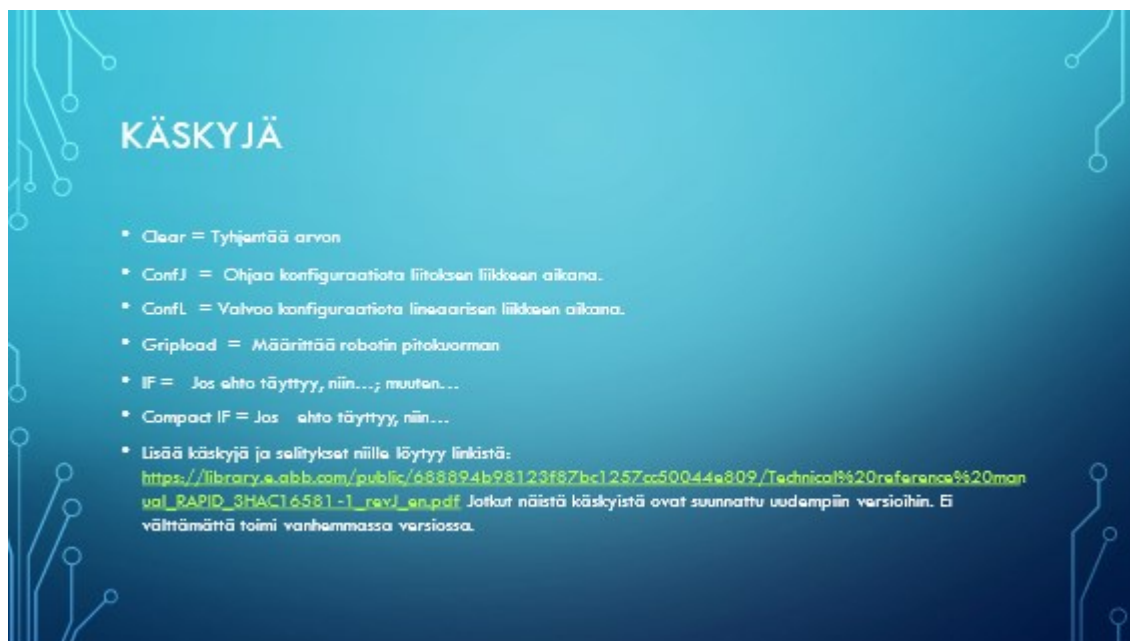


- Tehtävässä voi käyttää mittanauhaa. Kappaleitten paletti paikka myös vaihdetaan - Tällä simuloidaan mahdollisia nouto ja/tai vientipaikan vaihtoja.
- Tehtävä tässä osiossa on, että robotti vie ja palauttaa koko paletillisen kappaleita pakkaan ja takaisin.
- Käytä Move(C/J/L)Offs komentoa. Käytä mittanauhaa apuna offset koordinaattien määrittämisessä.
- Pitää tarttujan Offset korkeuden samana, koska sitä ei tarvitse muuttaa. Kopiointia + liittämistä kannattaa käyttää.
 - X,Y,Z offset mitat ovat millimetreinä. Z on korkeus.
 - Seuraavassa diassa apukuvia.

Liite 16. Apukuvia



Liite 17. Lisää funktioita



KÄSKYJÄ

- Clear = Tyhjentää arvon
- ConfJ = Ohjaa konfiguraatiota liitoksen liikkeeseen aikana.
- ConfL = Valvoo konfiguraatiota lineaarisen liitoksen aikana.
- Gripload = Määrittää robotin pitokuvaman
- IF = Jos ehto täyttyy, niin...; muuten...
- Compact IF = Jos ehto täyttyy, niin...
- Lisää käskyjä ja selitykset niille löytyy linkistä:
https://library.e.abb.com/public/688894b98123f87bc1257cc50044e809/Technical%20reference%20manual_RAPID_3HAC16581-1_rev1_en.pdf Jotkut näistä käskyistä ovat suunnattu uudempiin versioihin. Ei välttämättä toimi vanhemmassa versiossa.