



# **ABB IRB 140 robottisolun käyttöönotto**

Mikko Lindholm

Opinnäytetyö  
Maaliskuu 2011  
Kone- ja tuotantotekniikka  
Kone- ja laiteautomaatio  
Tampereen ammattikorkeakoulu



TAMK UNIVERSITY OF APPLIED SCIENCES

Mechanical and industrial engineering

Machine automation

Writer Mikko Lindholm

Thesis Introduction of ABB IRB 140 robot cell

Engineering thesis 69 pages + 19 appendices

Graduating time April 2011

Thesis supervisor Olavi Kopponen

Co-operating Company Tampere University of Applied Sciences

---

## **Abstract**

The target of my master's thesis was to build working robot cell for the students to practice with. The robot was disassembled and needed some changes. The main changes to the robot cell were building new tools, adding some safety equipment and removing unnecessary parts. Finally the robot cell was suitable to the studying environment.

My thesis is in two parts. The first part is a theory, which deals with the programming of the robot, and the structures of commands and program used. The section also focuses on their use through examples and one of the main move commands is opened as syntax. It gives the opportunity to see how the robot handles the commands.

The second part of my work deals with the changes to the robot and problems that occurred during the process. The main problems occurred in robots safe handling and the changes that occurred while editing the robot cell.

---

Keywords ABB, robot, robotics, introduction

## Symboliluettelo

I/O	In/Out Put Sisä/ulos tulot
DI	Digital input Tulo
DO	Digital output Lähtö
°	Asteen merkintätyyli

## Sanasto

### Resistiivinen kosketusnäyttö

Resistiivinen kosketusnäyttö on tarkempi kuin kapasitiivinen, mutta ei mahdollista moni kosketusta. Lisäksi resistiivistä kosketusnäyttöä voidaan käsitellä myös hanskat kädessä tai kynsillä, kun taas kapasitiivinen tarvitsee ihokosketuksen tai erillisen ohjaintikun.

### Servo

Asemointiin tarkoitettu toimilaitteen ohjauspiiri, jossa on takaisinkytkentä\_aseamanturiin. Mitä kauempana servo-ohjattu laite on tavoiteasemasta, sitä suuremmalla voimalla se pyrkii siihen.

### Syntaksi

Lauseoppi, johon sisältyy varattujen sanojen tunnistus.

## **Alkusanat**

Opinnäytetyön tekeminen Tampereen ammattikorkeakoululle oli minulle luonnollinen valinta, koska sain mielenkiintoisen ja haastavan aiheen. Aihe oli minulle henkilökohtaisesti kiinnostava, koska olen ollut aina kiinnostunut tekniikasta ja nyt pääsin suoraan tutustumaan robottiin alusta alkaen. Suuret kiitokset Tampereen ammattikorkeakoulun osaaville opettajille, Trimaster OY:lle ja ABB:n Juha Mainiolle, joilta kaikilta sain tukea ja inspiraatiota.

25.03.2011

Mikko Lindholm

## Sisällysluettelo

Johdanto .....	9
2. Robottijärjestelmä .....	10
2.1 Robotti IRB-140 .....	11
2.1.1 Tekniset tiedot Irb-140 .....	13
2.2 Ohjausmoduuli .....	14
2.2.1 Käsiäjotila .....	15
2.2.2 Testaustila (100% nopeus) .....	16
2.2.3 Automaattiajotila (tuotantotila) .....	17
2.2.4 Liitännät ja käyttöpainikkeet .....	19
2.3 Käsiohjain .....	21
2.4 Robotin käsiajo .....	24
2.4.1 Lineaarinen käsiajo .....	26
2.4.2 Askellusliike (inkrementti) .....	27
3. Ohjelmointi .....	29
3.1 RAPID-ohjelmointi .....	29
3.1 Varatut sanat .....	31
3.2 Ohjelman rakenne .....	32
3.2.1 Pääohjelma .....	33
3.2.2 Aliohjelma .....	33
3.2.3 Ohjelmadata .....	34
3.2.4 Käskyt .....	34
3.2.5 Moduilit .....	34
3.2.6 Järjestelmämoduuli .....	35
3.2.7 Ohjelmamoduuli .....	35
3.5 Ohjelman lataaminen .....	36

3.5.1 Ohjelman lataaminen vaiheittain .....	36
4 Datatyypit .....	39
4.1 Muuttuja .....	39
4.1.1 Muuttujan ominaisuudet .....	40
4.1.2 Muuttujan määrittely .....	40
4.1.3 Muuttujan arvojen antaminen .....	40
4.2 Pysyvä muuttuja .....	41
4.2.1 Pysyvän muuttujan määrittely .....	41
4.3 Vakio .....	42
4.3.1 Vakioarvon käyttö .....	43
5. Käskyjen rakenne .....	44
5.1 Liikekäskyn rakenne .....	44
5.1.1 Termien selitykset .....	45
6 Käyttöönotto .....	48
6.1 Robottisolun kokoaminen .....	49
6.2 Työkalunvaihtaja .....	52
6.3 Työkalut .....	53
6.3.1 Kaksoistarrain .....	53
6.3.2 Kääntyvätarrain .....	55
6.3.2 Työkalunvaihtajan teline .....	57
6.3.3 Paineilmaliitännän lisääminen .....	58
6.4 Turvaratkaisut .....	61
6.4.1 Valoverho .....	61
6.4.2 Turvarele .....	62
6.4.3 Ovien rajakytkimet .....	63
6.5 Robotin ohjelmiston päivittäminen .....	64
7 Havaintoja ja ongelmia .....	65

8 Yhteenveto.....	67
Lähteet .....	68
Liitteet .....	69



## Johdanto

Tämän opinnäytetyön tarkoituksena oli ottaa käyttöön Tampereen ammattikorkeakoulun robottiluokassa sijaitseva ABB-robotin robottisolun. Solu oli alun perin käytössä messuesittelyissä ja se muokattiin opetuskäyttöön sopivaksi.

Alkuperäisessä sovelluksessa robotti poimi hinnakuljettimelta muistitikun, siirsi sen lasertulostinyksikköön, jossa siihen tulostettiin asiakkaan haluama teksti. Tämän jälkeen muistitikku otettiin lasertulostusyksiköstä pois ja siirrettiin asiakkaan otettavaksi. Kappaleen paikantaminen kuljetinhihnalta toteutettiin konenäköohjelmiston ja pulssianturin avulla.

Aikaisempaan kokoonpanoon jouduttiin tekemään huomattavia muutoksia, mm. konenäkö ominaisuuden poisjättämisen ja robotin työkalujen muuttamisen takia. Myös robottisolun rakenteita muutettiin huomattavasti.

Robottijärjestelmää tullaan käyttämään opetustarkoituksessa, laboratoriotöiden yhteydessä.

## 2. Robottijärjestelmä

Robottijärjestelmä on kokonaisuus, jossa robotti poimii tasolta tai kuljettimelta kappaleita ja siirtää ne haluttuun paikkaan. Tätä kokonaisuutta on mahdollista tulevaisuudessa laajentaa toisella robotilla tai muilla laitteilla. Tässä opinnäytetyössä robotti toimii omana yksikkönään kappaleenkäsittelijänä joka poimii kappaleita kuljettimelta palettiin erilaisten vaihdettavien työkalujen avulla.

Robotissa on käytössä kaksi erilaista paineilmakäyttöistä tartuntatyökalua, joiden avulla saadaan toteutettua kappaleiden siirtäminen erilaisiin asentoihin ja paikkoihin harjoitustöiden vaatimalla tavalla. Järjestelmää on mahdollista laajentaa myös kokenäköominaisuudella, mutta järjestelmän laajuuden takia se jätettiin tästä opinnäytetyöstä pois. Kappaleiden ominaisuuksien ja paikkatiedon määrittämiseksi järjestelmään voidaan kehittää lisäämällä erilaisia kappaleen ominaisuuksia tai paikkaa ilmaisevia antureita, mutta ne jätettiin tuleviin kehityshankkeisiin toteutettavaksi.



**Kuvio 1.** Robottijärjestelmä

Robottijärjestelmä koostuu neljästä osasta (Kuvio 1): Robotti, ohjainyksikkö, käsiohjain ja tietokone.

Ohjainyksikkö ja tietokone sijaitsevat sivummalla, eivätkä siksi näy Kuviossa. Opetuskäyttöä varten tähän robottiin hankittiin työkalunvaihtaja ja rakennettiin kaksi erilaista paineilmakäyttöistä tartuntatyökalua joiden avulla robotti pystyy poimimaan tavaroita kuljettimelta tai tasolta.

## 2.1 Robotti IRB-140

IRB-140 (Kuvio 2) on ABB:n toiseksi pienin kappaleen käsittelyyn tarkoitettu 6-akselinen robotti. Robottia valmistetaan sekä viiden, että kuuden kilogramman nos-

tokyvyyllä. Koululle hankitun robotin nostokyky on kuusi kilogrammaa. Robotin kaikkia akseleita voidaan liikuttaa samaan aikaan. Tämä mahdollistaa tehokkaat ja sulavat liikkeet. Robottia voidaan ajaa käsiajolla joko akseli kerrallaan tai antamalla paikkapiste muuttamalla XYZ-koordinaatteja.

Robotille voidaan myös opettaa reittipisteet (flyby), joiden kautta robotti kulkee määrättyssä järjestyksessä. Tässä tapauksessa on kuitenkin huomioitava, että käyttää oikeaa työkalua määrityksissä kolarin välttämiseksi. Pisteet voidaan ohjelmoida syöttämällä ne käsin, RobotStudio ohjelman avulla tai käyttämällä opetustoimintoa, jolloin robotti ajetaan haluttuun asentoon ja paikkaan sekä tallennetaan paikkapiste.

Nämä paikkapisteet tallentuvat ohjelmadataan ja ovat kaikkien rutiinien käytettävissä.



**Kuvio 2.** ABB IRB-140 robotti

### 2.1.1 Tekniset tiedot Irb-140

Vapausasteita: 6 (Kuvio 3)

Suurin kuormitettavuus: 6 kg

Toistotarkkuus:  $\pm 0.03mm$

Ulottuvuus: 810 mm

Maksiminopeus:  $2500 \frac{mm}{s}$

Liikeradat:

1-akseli 360°

2-akseli 200°

3-akseli 280°

4-akseli 400°

5-akseli 240°

6-akseli 800°

Paino: käsivarren 98 kg ja koko yksikön 250 kg

IP luokitus: 67

Käyttölämpötila: 5-45°C

Käyttökosteus: 95 %

Maksimi kiihtyvyys:  $20 \frac{m}{s^2}$

Kiihtyvyyisaika 0-1  $\frac{m}{s}$  : 0.15 s

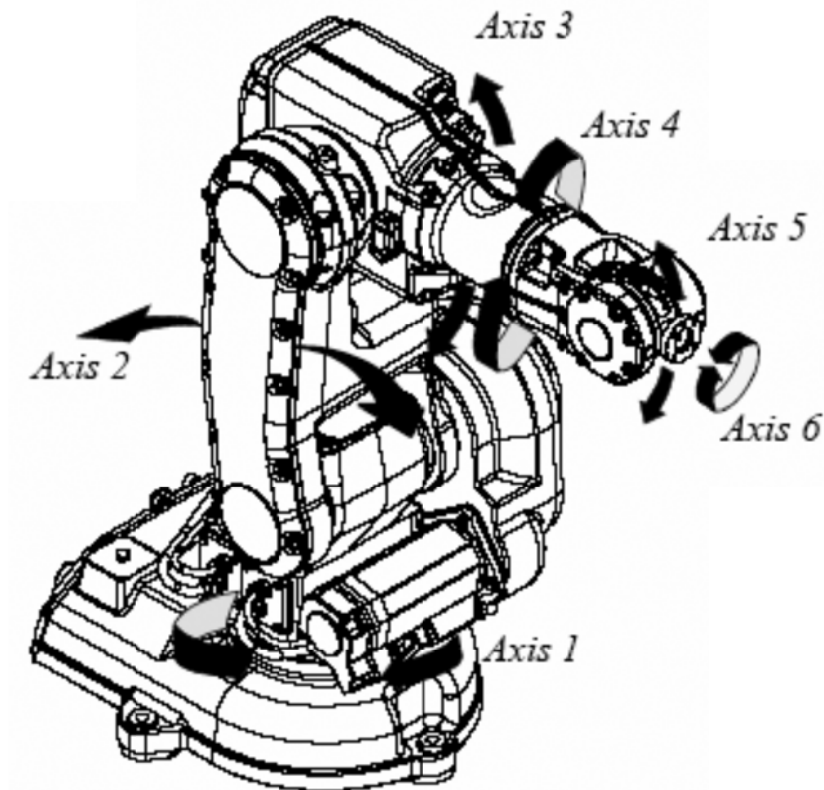
Verkkojännite 400 AC

Muuntaja 260 VAC

Tasasuuntaaja 260 VAC ja 370 VDC

Virtalähde 370 VDC

Robottimoottoreiden virtalähde 370 VDC



**Kuvio 3.** Robotin vapausasteet

## 2.2 Ohjausmoduuli

Ohjausmoduuli on robotin ohjausyksikkö (Kuvio 4). Tässä osassa tapahtuu varsinainen robotin ohjaus. Tietokoneella tai käsiohjaimella tehdyt ja muokatut ohjelmat lähetetään ohjausyksikölle, jonne ne tallentuvat. Ohjausmoduuli sijaitsee robotin alla olevassa kaapissa. Ohjausmoduulin etupaneelissa on tärkeimmät painikkeet ja avain, millä vaihdetaan ohjausyksikön tilaa. Käytettävissä ovat seuraavat kolme tilaa: Käsi-ajotila, testaustila ja automaattiajotila.



**Kuvio 4.** Robotin ohjausyksikkö

### 2.2.1 Käsiäjotila

Käsiäjotilassa robotti voi liikkua vain käyttämällä alennettua ja turvallista nopeutta (25% maksimi nopeudesta). Tässä tilassa käyttö tapahtuu ainoastaan käsin ohjaten. Käsiäjotila valitaan ohjainyksikön käyttötapakytkimestä (Kuvio 5). Käsinajoa käytetään ohjelmien luomiseen ja robottijärjestelmä käyttöönottoon.



**Kuvio 5.** Käyttötapakytkin

Robotin servomoottorit aktivoidaan painamalla sallintakytkintä (Kuvio 6). Käsiäjotilaa käytetään useimmiten ohjelmaa luotaessa ja robottijärjestelmän käyttöönotossa.

Käsiäjotilassa voidaan:

- Peruuttaa robottia ohjelmapolkua pitkin hätäseis-tilanteen jälkeen
- Muuttaa I/O-signaalien arvoja vikatilanteiden jälkeen
- Luoda ja editoida RAPID-ohjelmia
- Muuttaa ohjelmoitua paikkapisteitä

/1, s 43/



**Kuvio 6.** Sallintakytkin ("kuolleen miehen kytkin")

### 2.2.2 Testaustila (100% nopeus)

Täyden nopeuden käsiäjotila käytetään ohjelman testaamiseen luonnollisessa tilanteessa, verrattuna käsiäjotilan normaaliin alennettuun nopeuteen. Tämä tila on käyttökelpoinen robotin ja käsittelylaitteen välistä synkronointia testatessa. Testaustila valitaan ohjainyksikön käyttötapakytkimestä (Kuvio 7)



Testaustilassa voi ajaa täydellä nopeudella ja käytössä on myös ohjelmaeditorin antama virheiden poistomahdollisuus, mitä tuotantotilassa ei ole.

## VAARA

Testaaminen täydellä nopeudella on vaarallista. Varmista, että turva-alueella ei ole ihmisiä käynnistäessäsi ohjelman. Ulkoiset turvalaitteet, kuten valoverho, ovirajat, ynnä muut **eivät ole** käytössä ohjelman suorittamisen aikana. /1, s. 43/



**Kuvio 7.** Testaustilan valinta

### 2.2.3 Automaattiajotila (tuotantotila)

Automaattiajossa sallintakytkin (Kuvio 6) on toimeton, jolloin robotti voi liikkua ilman ihmisen ohjaamista. Tuotannossa robottijärjestelmä toimii yleensä automaattitilassa. Robottijärjestelmää voidaan ohjata kauko-ohjauksella, esim. käyttämällä ohjaimen I/O signaaleja. Tulosignaali voi esimerkiksi käynnistää tai pysäyttää RAPID-ohjelman, aktivoida robottijärjestelmä moottoreita, ja niin edespäin. Tuotantotila otetaan käyttöön valitsemalla käyttötapakytkimestä tuotantotilan asento (Kuvio 8).

Automaattiajotilassa käytetään monia lisäturvalaitteita, kuten valoverhoa ja rajakytkimiä turvallisuuden lisäämiseksi. Nämä lisäturvalaitteet eivät ole käytössä käsiajotiloissa. /1, s.44/

## Rajoituksia

Käsiäjo ei ole mahdollinen automaattitilassa.

Automaattiajossa voidaan:

- Käynnistää ja pysäyttää prosesseja
- Ladata, käynnistää ja pysäyttää RAPID-ohjelmia
- Peruuttaa robottia ohjelmapolkua pitkin hätäseis-tilanteen jälkeen
- Ottaa järjestelmästä varmuustallennuksia
- Palauttaa varmuustallennuksia
- Puhdistaa työkaluja
- Valmistella ja vaihtaa työkohteita
- Suorittaa muita prosessiin liittyviä tehtäviä

## VAARA!

Automaattitilassa robotti voi liikkua ilman ennakkovaroitusta

**Varmista, että turva-alueella ei ole ihmisiä**

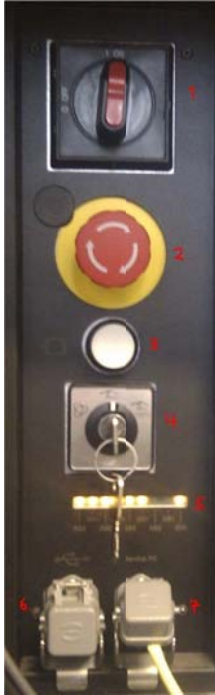


**Kuvio 8.** Tuotantotilan valinta

## 2.2.4 Liitännät ja käyttöpainikkeet

Ohjausyksikössä löytyvät seuraavat kytkimet ja käyttöpainikkeet (Kuvio 9):

1. Päävirtakytkin
  - Tästä kytketään robottiin virrat päälle
2. Hätäseis nappi
  - Häätätilanteessa robotin saa pysäytettyä
3. Kuittausnappi
  - Kuitataan, että kaikki on valmista automaattiajoa varten
4. Valintakytkin
  - Kytkimestä valitaan haluttu käyttötapa kolmesta eri vaihtoehdosta. Vaihtoehdot ovat käsiajo, käsiajo tuotantonopeudella ja automaattiajo
5. USB-liitin
  - varmuuskopioiden ottamista ja ohjelmien siirtämistä varten
6. ”Service pc” ethernet liitin,
  - Liitetään robotti online-ohjelmointia varten tietokoneeseen. Tämän liitännän kautta myös Picmaster konenäköohjelma käsittelee ja antaa komentoja robotille. /1, s. 4/



**Kuvio 9.** Liitännät ja käyttöpainikkeet



**Kuvio 10.** Robotin jarrujen vapautus

Robotti on myös varustettu jarrujen vapautus painikkeella (Kuvio 10). Tämän painikkeen avulla voidaan vapauttaa robotin kaikkien akselien jarrut. Tämä mahdollistaa hankalaan asentoon jumittuneen robotin liikuttamisen lihasvoimilla. Nappia painettaessa on huomioitava, että robotin käsivartta ei saa päästää hallitsemattomasti putoamaan. Pudottaminen voi aiheuttaa henkilö tai kalustovahinkoja. /1, s. 13/

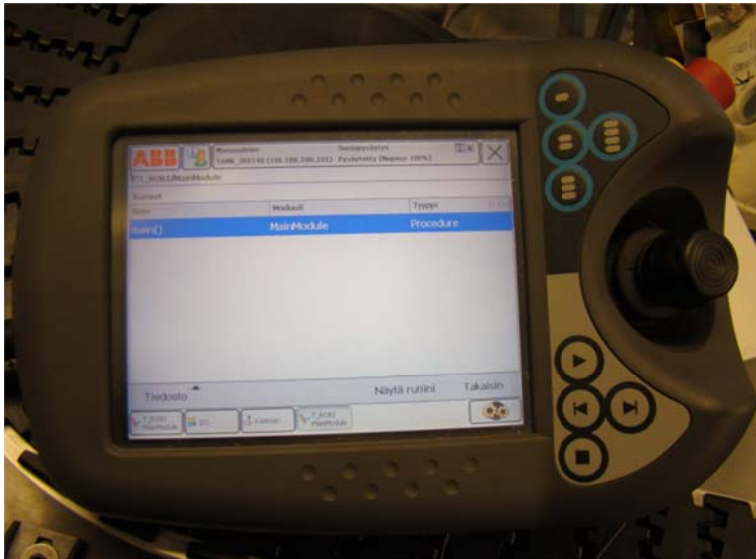
## 2.3 Käsiohjain

Ohjausyksikköön liitettiin myös käsiohjain, eli kädessä pidettävä opetusyksikkö (Kuvio 11). Ohjaimessa on suurikokoinen resistiivinen kosketusnäyttö (Kuvio 12) jonka avulla pystytään määrittämään tai muuttamaan robotille ohjelmoituja paikkapisteitä. Ohjaimessa on myös kolmisuuntainen ohjainsauva (Kuvio 12), jonka avulla voidaan ohjata robottia manuaalisesti ja samanaikaisesti useiden akselien liikettä.

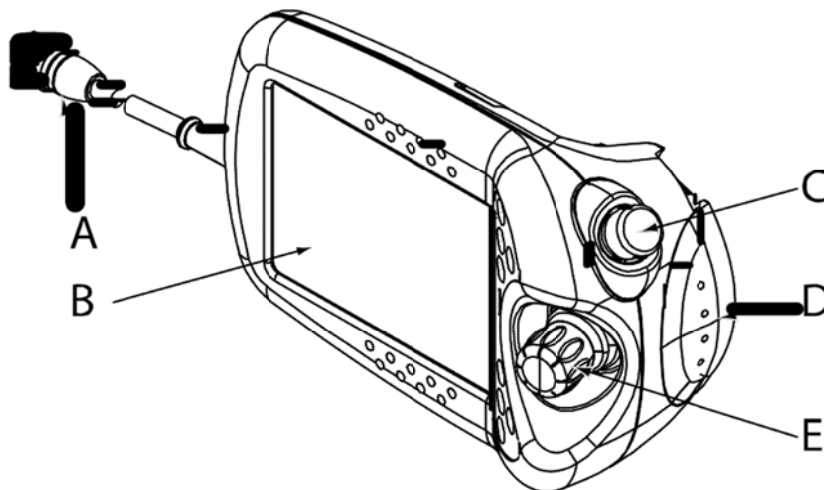
Käsiohjainta käytetään muun muassa paikkapisteiden määrittämiseen, robotin liikeratojen ja ohjelman toimivuuden testaamiseen, tarvittaessa virhetilanteiden korjaamiseen suoraan robotin vieressä. Tällainen tilanne on esimerkiksi vuorovaikutteinen ohjelman kierto. Tällöin käsiohjaimen näytön avulla tiedustellaan operaattorilta, montako kappaletta tuotetaan. Käsiohjain on erityisesti kätevä silloin, kun tarvitsee päästä kohteen lähelle tarkkailemaan samalla kun ohjataan robottia. /1, s. 8/

Käsiohjaimella tapahtuvaa robotin ajoa varten tulee avain kääntää ohjainyksikössä käsiajotilaan asentoon (Kuvio 5) Tämän lisäksi tulee ohjaimesta painaa kolmiasentoinen sallintakytkin (Kuvio 6) keskiasentoon, jolloin servot käynnistyvät. Sallintakytkin on robottia ohjattaessa pidettävä jatKuvio 6) keskiasennossa, koska muuten robotin servot sammuvat. Sallintakytkin on varoimenpide, jolla estetään ohjaavan henkilön puristumien robotin ja jonkin rakenteen väliin. Samalla varmistetaan se, että robotti ei liiku vahingossa painettaessa laitepainikkeita (Kuvio 12).

Lisäksi ohjaimesta löytyy neljä esiohjelmoitavaa nappia, joilla saa valittua lähtöjä päälle tai pois. Näitä nappeja on kätevä käyttää mm. kappaleiden noukkimiseen oikean paikkapisteen löytämiseksi (Kuvio 13). /2, s. 44/



**Kuvio 11.** Käsiohjain



**Kuvio 12.** Ohjausyksikön laitepainikkeet

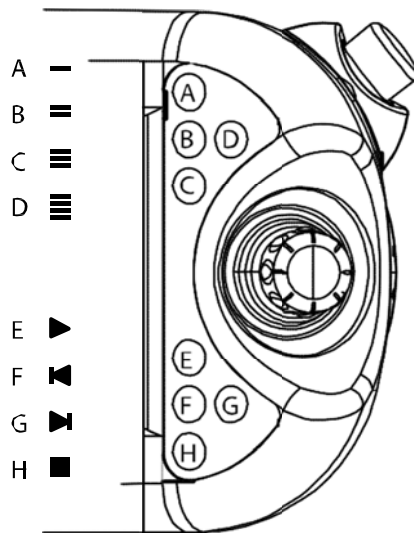
A Ohjausyksikön liitin

B Resistiivinen kosketusnäyttö

C Häätaseis kytkin

D Kuolleenmiehen kytkin

E Kolmisuuntainen ohjaussauva



**Kuvio 13.** Ohjausyksikön ohjelmapainikkeet

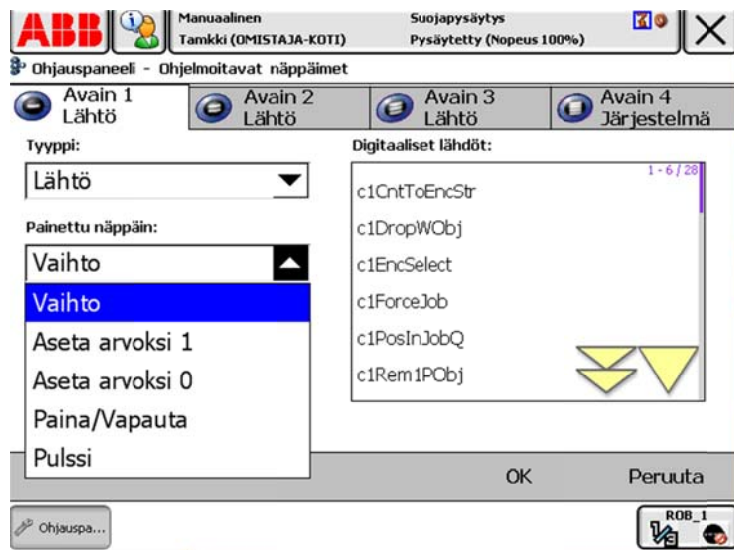
A-D	Ohjelmoitavat pikapainikkeet 1-4
E	Aja painike
F	Askellus eteen
G	Askellus taakse
H	Seis painike

Ohjelmoitavien pikapainikkeiden määrittäminen tapahtuu ABB valikon Ohjauspaneeli välilehdeltä. Lehdeltä valitaan ohjelmoitavat näppäimet valikko, jossa voidaan määrittää neljälle pikanäppäimelle haluttu tyyppi: Ei mitään, tulo, lähtö tai järjestelmä.

Mikäli valitaan tyyppi lähtö, voidaan valita seuraavista tapahtumavaihtoehtoista (Kuvio 14):

1. Vaihto
  - Vaihtaa signaalin arvon toiseksi. Esim. 1 0 ➔
2. Aseta arvoksi 1
  - Asettaa lähdölle arvon 1

3. Aseta arvoksi 0
  - Asettaa lähdölle arvoksi 0
  
4. Paina/Vapauta
  - Painettaessa vaihtaa signaalin arvoksi 1 (Huomioi, että invertoitu signaali antaa arvon 0).
  
5. Pulssi
  - Signaalin arvo tekee yhden pulssin



**Kuvio 14.** Pikanäppäinten ohjelmointi

## 2.4 Robotin käsiajo

Robottia voidaan ohjata ohjausyksikön avulla pitämällä sitä joko vasemmalla tai oikealla kädellä. Oikeanlaiset ohjausyksikön käsittelyotteet näkyvät kuvioista 15 ja 16.

Käsiajo tarkoittaa robotin tai ulkoisten akselien ohjaamista ohjausyksikön kolmisuuntaisella ohjaussauvalla. Käsiajo on mahdollista vain käsiajo-tiloissa, mutta ei samanaikaisesti kun robotti suorittaa ohjelmaa. Yleensä käsiajo suoritetaan suorina liikelinjoina, esimerkiksi lineaariliikkeenä paikkapisteestä A paikkapisteeseen B.

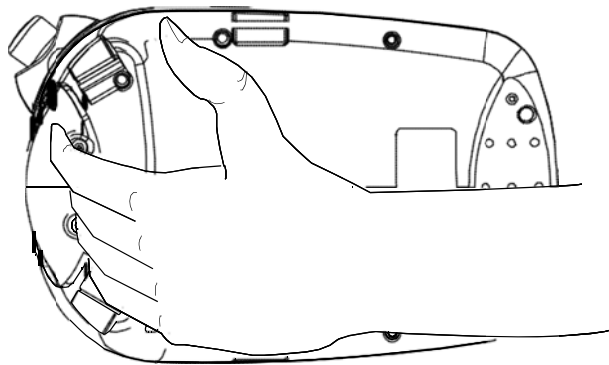


Lineaariliike suhteutuu valittavaan koordinaatistoon (maailma-, työkalu-, perus-, käyttäjä-, ohjelmasiirros tai kohdekoordinaatisto). Valittava koordinaatisto määräytyy valittavan tehtävän mukaisesti. /2, s. 42/

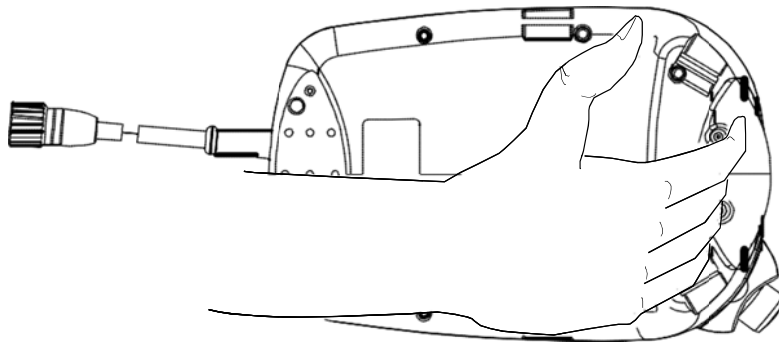
Voit myös liikuttaa robottia yksi akseli kerrallaan. Akseli kerrallaan liike ei kuitenkaan kohdistu mihinkään koordinaatistoon. Tällöin menetetään työkalun kohdistustiedot, mikä voi vaikuttaa myöhempisiin toimintoihin.

**HUOM!**

Robotin liikerata käsiajossa ei ole yleensä sama kuin ohjelman ohjaama rata. Ohjelma ohjaa robottia käyttämään nopeinta ja lyhintä suoraa tai kaarevaa liikerataa kahden paikkapisteen välillä. Tämä on huomioitava mahdollisia törmäyksien välttämiseksi. /1, s. 15/



**Kuvio 15.** Ohjausyksikön pitäminen vasemmalla kädellä

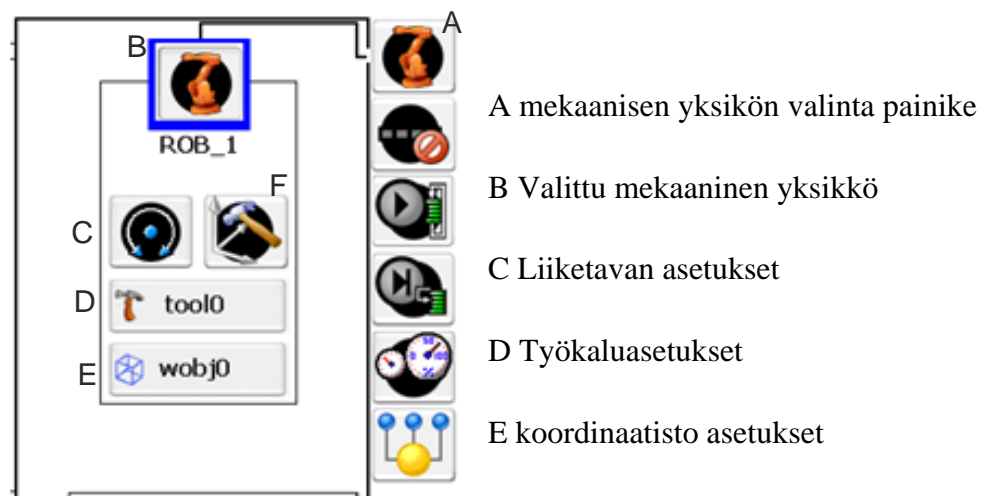


**Kuvio 16.** Ohjausyksikön pitäminen oikealla kädellä

### 2.4.1 Lineaarinen käsiajo

Valittu liike ja/tai koordinaatisto määrittää robotin liikkeen.

Lineaariliikkeessä työkalupiste liikkuu suoria viivoja pitkin avaruudessa esimerkiksi pisteen A ja B välin. Työkalupiste liikkuu valitun koordinaatiston akselien mukaisesti. /1, s. 16/



**Kuvio 17.** Liiketavan valinta

Ohjaussauvan toiminta määräytyy valitun liiketavan mukaan (Kuvio 17). Vaihtoehtoina ovat (Kuvio 18):

1. akselit 1-3
2. akselit 4-6
3. Lineaarinen
4. Uudelleen orientointi

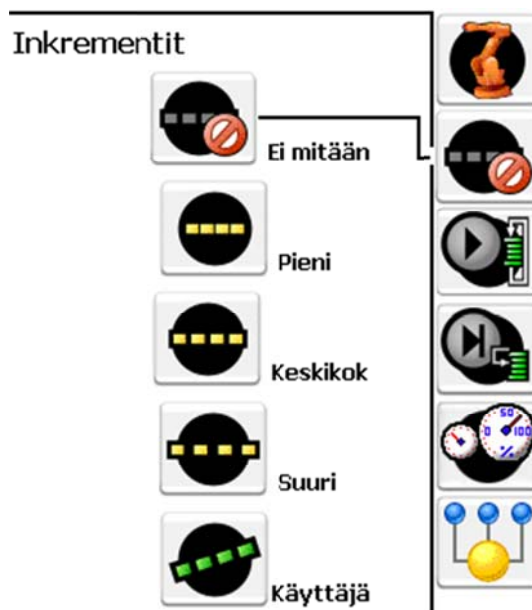


**Kuvio 18.** Liikuteltavien akselien valitseminen

Ohjaussauvan liikkeiden toiminta näkyy ikkunassa *ohjaussauvan suunnat* valinnan jälkeen. Huomioi kuitenkin, että ohjaussauvan suunta ei aina vastaa mekaanisen yksikön liikesuuntaa. Varmista aina liikesuunnan oikeellisuus käyttämällä lyhyitä ohjaussauvan liikkeitä, varmistaaksesi mekaanisen yksikön todellisen liikesuunnan. /1, s. 17/

#### 2.4.2 Askellusliike (inkrementti)

Askellusliikettä voidaan käyttää robotin paikantamiseen. Aina kun ohjaussauvaa käännetään, robotti siirtyy yhden askeleen (inkrementti). Mikäli ohjaussauvaa käännetään sekunnin tai pidempään, niin robotti askeltaa (10 askelta sekunnissa) niin kauan kuin ohjaussauvaa pidetään käännettynä, robotin liikerata saavuttaa rajansa tai tulee osuma rakenteisiin. /1, s. 19/



**Kuvio 19.** Inkrementit

Käytössä olevat askelkoot liittyen kuvioon 19.

### 3. Ohjelmointi

Tässä kappaleessa käsitellään Rapid-ohjelmointiin liittyvää ohjelmarakennetta. Rakenteeseen kuuluu oleellisesti varattuja sanoja, joita ei tule käyttää ohjelman osien nimeämiseen. Lisäksi käsitelen muita robotin ohjelmoinnin mahdollistavia ohjelmia ja niiden puutteita verrattuna RAPID editoriin.

#### 3.1 RAPID-ohjelmointi

Mikäli haluat tietokoneen tekevän jotain, tarvitaan ohjelma. RAPID on ohjelmakieli, kuten C-kieli, joita käytetään ohjelman kirjoittamiseen. Tietokoneen perusohjelma sisältää vain ykkösiä ja nollia. Tätä ohjelmatyyppiä on ihmisten mahdotonta ymmärtää. Tämän vuoksi tietokoneiden ohjelmoinnin helpottamiseksi on kehitetty erilaisia korkeantason ohjelmointi kieliä. RAPID on yksi tällainen korkeantason ohjelmointikieli. Se sisältää joitakin englanninkielisiä sanoja, kuten (IF ja FOR), jotta siitä on saatu helpommin ymmärrettävä. /3/ s.11

RAPID ohjelmointia on mahdollista tehdä usealla eri ohjelmalla. Kaikkein helpointa se on RobotStudion liitettyllä RAPID Editor nimisellä ohjelmaeditorilla. Tätä käytettäessä editori näyttää esimerkiksi MoveL käskyn jälkeen sallitut lisäkomennot (Kuvio 20). Toinen yksinkertainen tapa muokata ohjelmaa on käyttää käsiohjaimen ohjelmaeditoria (Kuvio 21), jolloin ohjelma itsessään pitää huolen muotokielen oikeellisuudesta.

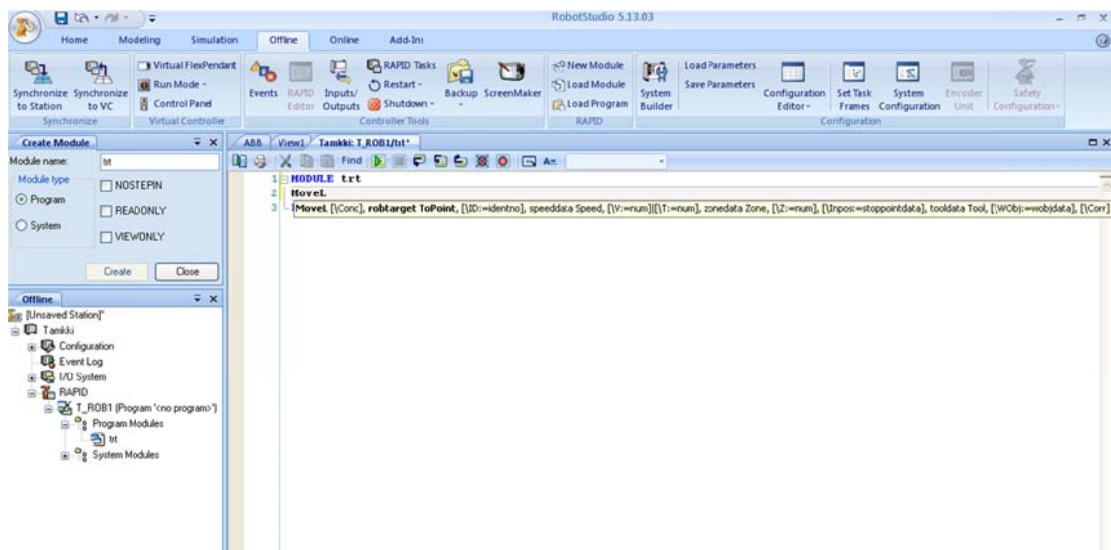
Molemmat ABB:n ohjelmaeditorit ehdottavat seuraavaa käskyyn liittyvää parametria, mitä tarvitaan käskyn suorittamiseen ja sisältävät myös virheenkorjaus toiminnon, joka näyttää missä kohdassa ohjelmaa on muotokielen (syntaksin) vastainen osa.

RobotStudiota käytettäessä on myös mahdollista simuloida robotin toimintaa ja liikkeitä kuten myös tehdä törmäystarkastelu, mikäli ohjelmaan on tuotu esimerkiksi

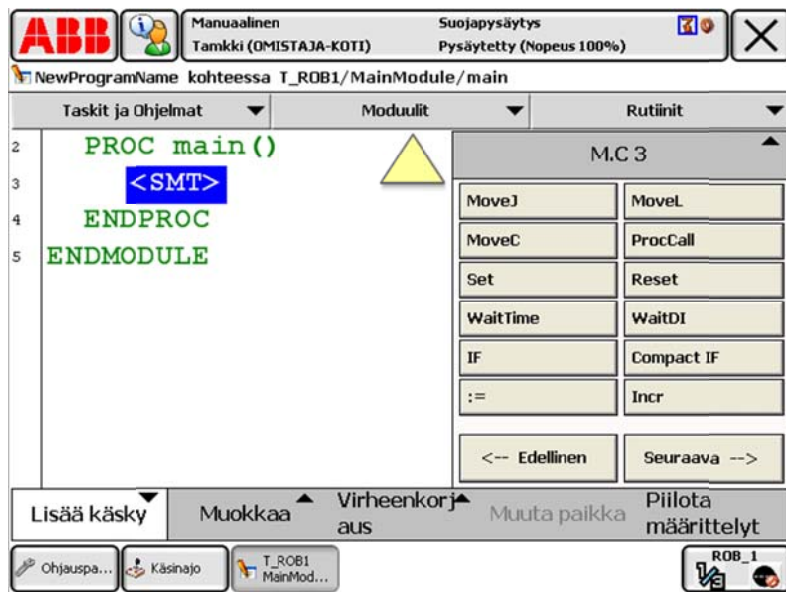
autocad:illä tehtyjä kappaleita. Tämä mahdollistaa tuotannon koko prosessivaiheen simuloinnin ja työkierron hiomisen mahdollisimman hyväksi.

RobotStudioon avulla on myös mahdollista tehdä Online ohjelmointia, eli robottiin ollaan yhteydessä ethernet yhteyden avulla. Tämä kuitenkin edellyttää, että robottiin on aktivoitu kyseinen ominaisuus. Jokaiseen ABB:n valmistamaan robottiin saa tämän ominaisuuden maksua vastaan käyttöön. Online-ohjelmointi mahdollistaa ohjelmoinnin, ilman että operaattori on fyysisesti läsnä robotin luona.

Ohjelmia on mahdollista kirjoittaa ja muokata myös perustason tekstieditorilla kuten Notepad. Tällöin pitää olla tarkkana tekstin muotopuolesta (syntaksi), koska RAPID ohjelmointikieltä ymmärtävää virhekorjausta ei ole käytettävissä normaalissa tekstieditorissa. Notepad on hyvin suosittu tapa editoida yksinkertaisia kappaleen siirto-ohjelmia, jossa useissa soluissa käytetään saman ohjelman pohjaa ja vain joitakin arvoja muutellaan. /5/



**Kuvio 20.** RAPID Editor



**Kuvio 21.** Käsiohjaimen ohjelmaeditori

### 3.1 Varatut sanat

Alla olevaan listaukseen on kerätty sanat, jotka ovat varattuja robotin ohjausjärjestelmässä. Niillä on erityinen tarkoitus RAPID ohjelmointikielessä, eikä niitä saa käyttää identifioimiseen.

Listassa on myös joukko ennalta määrättyjä nimiä eri datatyypeille, systeemi datalle, funktioille ja ohjeille. Näitä ei saa missään tapauksessa käyttää nimeämiseen. Näiden sanojen käyttäminen muissa, kuin sanoille tarkoitetuissa toiminnoissa, aiheuttaa virhe- tai jopa vaaratilanteita ohjelmaa suoritettaessa. /4, s. 17/

ALIAS	AND	BACKWARD	CASE
CONNECT	CONST	DEFAULT	DIV
DO	ELSE	ELSEIF	ENDFOR
ENDFUNC	ENDIF	ENDMODULE	ENDPROC
ENDRECORD	ENDTEST	ENDTRAP	ENDWHILE
ERROR	EXIT	FALSE	FOR
FROM	FUNC	GOTO	IF
INOUT	LOCAL	MOD	MODULE
NOSTEPIN	NOT	NOVIEW	OR
PERS	PROC	RAISE	READONLY
RECORD	RETRY	RETURN	STEP
SYSMODULE	TEST	THEN	TO
TRAP	TRUE	TRYNEXT	UNDO
VAR	VIEWONLY	WHILE	WITH
XOR			

### 3.2 Ohjelman rakenne

Ohjelma koostuu robottia ja oheislaitteita määrytällä tavalla ohjaavasta RAPID-kielellä ohjelmoiduista käskyistä ja datasta.

Ohjelma muodostuu yleensä neljästä erilaisesta osasta (Kuvio 22):

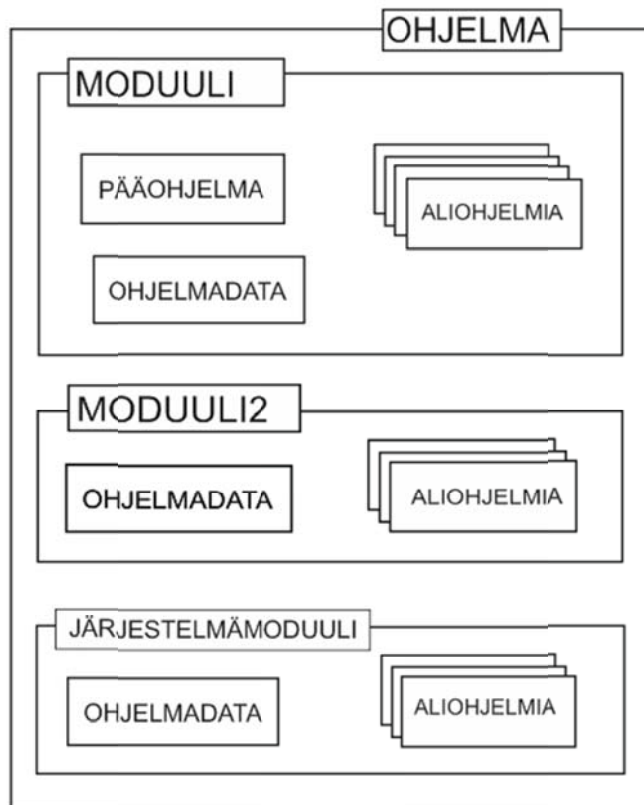
Päärutiini (pääohjelma)

Useita alirutiineja (aliohjelmia)

Ohjelmadata

Järjestelmämoduleista





**Kuvio 22.** Ohjelman rakenne

### 3.2.1 Pääohjelma

Ohjelman suoritus alkaa aina päärutiinista. Päärutiini on hyvä pitää mahdollisimman yksinkertaisena ja helposti luettavana, eli tehdä kaikki ohjelmat aina alirutiineiksi. Tällöin ohjelmaa on yksinkertaisempi lukea ja muokata virhetilanteiden sattuessa.

Päärutiini koostuu useista alirutiineista, jotka vastaavat robotin työkiertoa.

### 3.2.2 Aliohjelma

Aliohjelmaa käytetään ohjelman jakamiseksi pienempiin ja helpommin luettaviin osiin, jolloin muodostuu helposti luettava modulaarinen ohjelma. Alirutiineja kutsutaan päärutiinista tai muista alirutiineista. Kun rutiini on suoritettu loppuun, ohjelman suoritus palaa kutsuneen rutiinin seuraavaan käskyyn.

### 3.2.3 Ohjelmadata

Dataa käytetään sijaintien, numeraalisten arvojen (rekisterien, laskurien, ym.), koodinaattijärjestelmien, jne. määrittämiseen. Dataa voidaan muuttaa manuaalisesti, mutta se voi muuttua myös ohjelman toimesta (esimerkiksi laskurit). Datatyyppejä on useita, joista tärkeimmän esittely kappaleessa 3.3 Datatyypit.

### 3.2.4 Käskyt

Käsky määrittää tietyn toiminnan, joka täytyy tapahtua käskyä suorittaessa. Esimerkiksi robotin liikuttaminen (MoveJ), lähtöarvon asettaminen (reg1:=5;), datan muuttaminen, jne. Ohjelman käskyt suoritetaan yksi kerrallaan siinä järjestyksessä, missä ne on ohjelmoitu.

### 3.2.5 Moduulit

Moduulia voi ajatella eräänlaisena hakemistona tai kansiona, joka sisältää ohjelmia ja ohjelmadataa. Moduuleita voidaan tallentaa erikseen ja tarpeettomat moduulit voidaan poistaa robotin käyttömuistista. Jos moduuleita ei ole erikseen tallennettu ja ne poistetaan muistista, häviävät moduulit pysyvästi.

### 3.2.6 Järjestelmämoduuli

Robotin muistiin on tallennettu järjestelmämoduuli, joka sisältää järjestelmässä tarvittavia toistuvia toimintoja. Näitä toimintoja ovat muun muassa Työkalujen määrittelyt, kotiasemaan ajot, työkalun esimerkiksi tarraimen ohjaus ja työkalunvaihtorutiinit. Nämä tiedot eivät häviä robotin muistista ohjelmia poistettaessa.

### 3.2.7 Ohjelmamoduuli

Ohjelmamoduuleita voi olla ohjaimen ladattuna useita samalla hetkellä, mutta vain yksi voi olla nimeltään main, eli pääohjelma. Myöskään muiden moduulien välillä ei saa esiintyä nimiristiriitoja, eli samoja nimiä kahteen tai useampaan kertaan. Ohjelmamoduuli sisältää ohjelmadataa ja ohjelmia.

Robotin ohjelma voidaan jakaa ohjelmamoduuleihin, joista jokainen sisältää joukon rutiineja ja dataa. Tämän lisäksi voidaan käyttää aina läsnäolevia järjestelmämoduuleita.

Moduulit voivat sisältää esimerkiksi:

Yleisiä rutiineja monille eri asennuksille

CAD:lla tuotettuja sijainteja

Oheislaitteille, esimerkiksi työkappaleen käsittelijälle, tarkoitettuja rutiineja.

Järjestelmämoduulit voivat sisältää kaikkien robotin ohjelmien käyttämää dataa (työkaludata).

Sekä järjestelmä että ohjelmamoduulit toimivat samalla tavalla, kun ne on ladattu muistiin. Kaikkia moduuleita voidaan normaalisti editoida ohjelmointiyksiköstä,

mutta järjestelmämoduulin ollessa normaalisti kirjoitussuojattu, täytyy suojaus ensin poistaa.

### 3.5 Ohjelman lataaminen

Tässä luvussa kerrotaan, kuinka ohjelma ladataan robotille. Ohjelma muodostuu yleensä kolmesta osasta: päärutiinista (mikä on aina mukana), joukosta alirutiineja ja ohjelmadatasta. Ohjelmassa voi olla vain yksi päärutiini.

#### 3.5.1 Ohjelman lataaminen vaiheittain

Seuraavaksi kuvataan aiemmin luodun ohjelman lataaminen vaihe vaiheelta.

ABB-valikossa paina *Ohjelmaeditori*

Paina *Taskit ja ohjelmat*

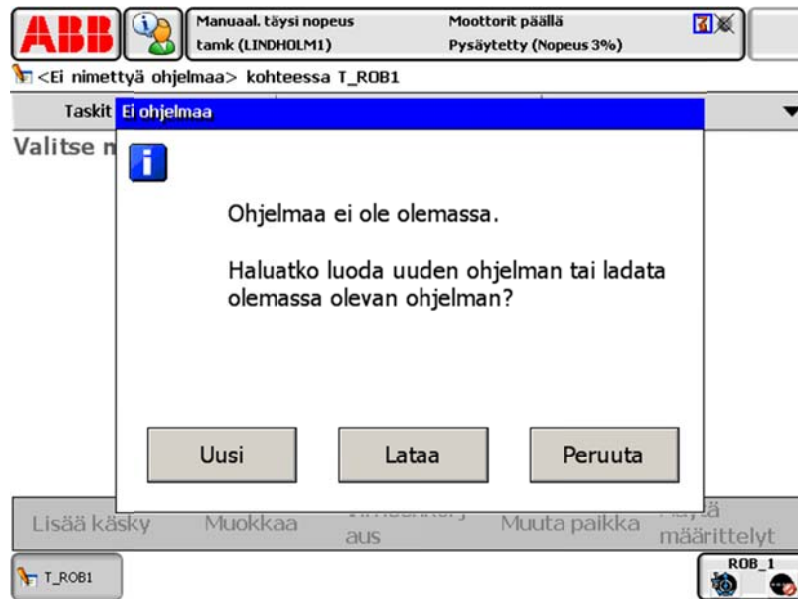
Mikäli tehtäväikkunassa ei ole ladattua ohjelmaa, dialogi-ikkuna avautuu (Kuvio 23).

Paina *Uusi*, ellei ohjelmaa ole ja uusi ohjelma luodaan.

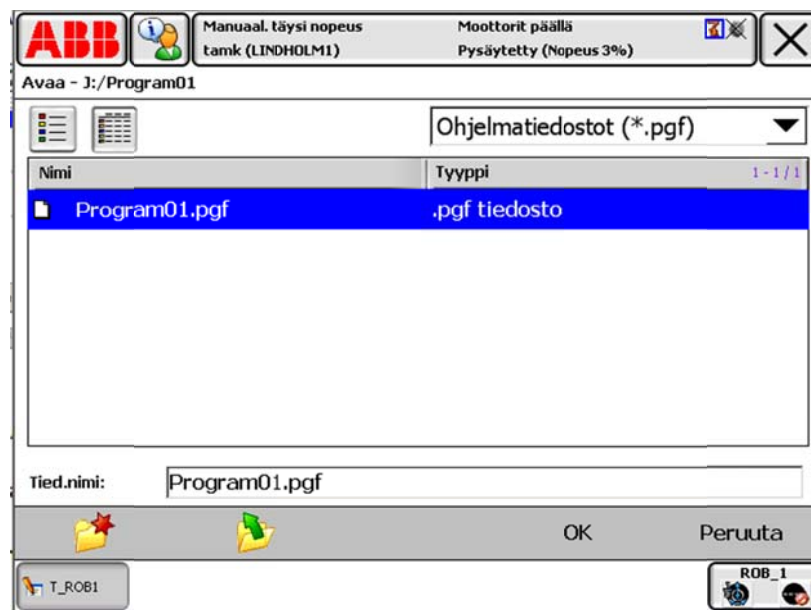
Paina *Lataa*, jos aikaisemmin kirjoitettu ohjelma käytetään.

Tilan tekemiseksi uudelle ohjelmalle, vanha ohjelma on ensin poistettava.

Käytä tiedostonhaku työkalua ladattavan ohjelmatiedoston hakemiseen (Kuvio 24).



**Kuvio 23.** Ohjelman lataaminen



**Kuvio 24.** Ohjelman lataaminen 2

Paina *Ok*.

Kun ohjelma on latautunut, näytössä näkyy esimerkiksi seuraavaa (Kuvio 25).

```
2  PROC main()  
3  Nollaus;  
4  kysely;  
5  kotiasema;  
6  FOR i FROM 1 TO n_vastaus_kiertoon DO  
7    haku;  
8    vienti_conv;  
9    haku_imu_conv;  
10   vienti;  
11  ENDFOR
```

**Kuvio 25.** Ohjelma ladattuna

## 4 Datatyypit

Seuraavissa kappaleissa on esitetty tärkeimmät datatyypit. Datatyypeistä käydään läpi niiden ominaisuuksia ja käyttötarkoituksia.

### 4.1 Muuttuja

#### Numeraaliset (Num)

Voivat sisältää kokonaislukuja tai desimaalilukuja. Esimerkiksi 10 tai 3.14159

#### Jono (String)

Tekstijono esim. "tämä on merkkijono". Voi sisältää korkeintaan 80 merkkiä

#### Boolean muuttuja (Bool)

Boolean muuttuja. Voivat saada vain arvot TRUE (tosi) tai FALSE (epätosi)

Kaikki robottiohjelmoinnissa käytettävät muut datatyypit (Liite 1) perustuvat näihin kolmeen eri datatyyppiin (Num, String ja Bool). Mikäli ymmärrät näiden datatyyppien käyttämisen ohjelmoinnissa, mahdollistaa se muiden mutkikkaampien datatyyppien käytön. /3, s. 12/

### 4.1.1 Muuttujan ominaisuudet

Muuttuja sisältää data-arvon. Jos ohjelma on pysäytetty ja aloitettu uudelleen, muuttuja pitää arvonsa. Mutta mikäli ohjelman osoitin on siirretty päärutiniin, niin olemassa ollut data arvo menetetään. Esimerkiksi työpäivän aikana valmiiksi saatujen kappaleiden lukumäärä menetetään ja laskuri aloittaa laskemisen taas alusta. /3, s. 12/

### 4.1.2 Muuttujan määrittely

Muuttujaa määriteltäessä on tapana määrittää nimi ja datatyyppi, jota muuttuja käyttää. Muuttuja määritetään käyttämällä avainsanaa VAR, välttääksemme ohjelmavirheitä. /3, s. 12/

*VAR datatyyppi nimi*

Esimerkki

VAR num length;

VAR string name;

VAR bool finished;

### 4.1.3 Muuttujan arvojen antaminen

Arvo annetaan muuttujalle käyttämällä käskyä :=

length := 10;

name := "John"

finished := TRUE;



Huomaa, että `:=` ei ole matematiikassa käytetty yhtä suurta tarkoittava merkki. Se tarkoittaa, että lauseke oikealla ohjaa muuttujaa vasemmalla. `:=` merkin vasemmalla puolella voi olla vain muuttuja.

Esimerkiksi, alla olevat ovat virheettömiä RAPID koodeja jolla `reg1` saa arvon 3.

```
reg1 := 2;
```

```
reg1 := reg1 + 1;
```

Määrittely voidaan myös tehdä samaan aikaan muuttujan määrittelyn kanssa.

```
VAR num length := 10;
```

```
VAR string name := "John";
```

```
VAR bool finished := TRUE;
```

## 4.2 Pysyvä muuttuja

Pysyvä muuttuja (Persistent variables) on yksinkertaisuudessaan samanlainen kuin tavallinen muuttuja. Tärkein ero on, että tämä muuttuja muistaa arvonsa, vaikka ohjelma lopetettaisiin ja aloitettaisiin taas alusta pääruutiinista. /3, s. 13/

### 4.2.1 Pysyvän muuttujan määrittely

Pysyvä muuttuja on eritelty avainsanalla `PERS`. Käskyssä alkuperäinen arvo on ilmoitettava

Esimerkki.

Tarkastele seuraavaa koodiesimerkkiä.

```
PERS num nbr := 1;
```

```
PROC main()
```

```
nbr := 2;
```

```
ENDPROC
```

Jos tämä ohjelma suoritetaan, olemassa oleva arvo on vaihdettu kahdeksi (2). Seuraavan kerran kun ohjelma suoritetaan, ohjelman koodi tulee näyttämään tältä:

```
PERS num nbr := 2;
```

```
PROC main()
```

```
nbr := 2;
```

```
ENDPROC
```

### 4.3 Vakio

Vakio (Constants) sisältää arvoja, kuten muuttuja, mutta arvo on aina annettu määrittelyinä (esim. 3.14 tai 9.81), arvoa ei voi sen jälkeen enää muuttaa ohjelman aikana. Vakioarvoa voidaan käyttää muuten ohjelmassa samalla lailla kuin muuttujaa. /3, s. 14/

Vakioarvo on määritetty avainsanalla CONST, jota seuraa datatyyppi, identifiointi ja määritetty arvo.

```
CONST num gravity := 9.81;
```

```
CONST string greating := "Hello"
```

### 4.3.1 Vakioarvon käyttö

Käyttämällä vakioarvoa muuttujana, varmistetaan siitä, että arvo ei muutu jossakin vaiheessa ohjelmassa. On parempi käyttää pysyvää muuttujaa, mikäli ohjelmaa tarvitsee päivittää jollakin muulla pysyvän muuttujan arvolla. Tällöin tarvitsee muuttaa vain yhteen paikkaan arvo ja voit olla varma, että et ole unohtanut päivittää arvoa minnekään muualle ohjelmaan. /3, s. 14/

## 5. Käskyjen rakenne

Tässä kappaleessa käsittelen liikekäsken rakennetta hieman perusteellisemmin. Kappaleessa on avattu käsken syntaksi ja kerrottu mitä käsky pitää sisällään robotin ohjelmoinnin näkökulmasta.

### 5.1 Liikekäsken rakenne

Esimerkissä käyn läpi robotin yhdistetyn liikekäsken syntaksin eli muotokielen.

#### **MoveJ – Liikuttaa robottia yhdistetyllä liikkeellä**

```
MoveJ [\Conc] ToPoint [\ID] Speed [\V] | [\T] Zone [\Z] [\Inpos]
Tool [\WObj]
```

Yläpuolella oleva teksti on robotin liikekäsken malli, minkä mukaan robotin operaattori määrittää haluamansa toiminnot. Seuraavaksi teksti kertoo mitä mikäkin käsken osa pitää sisällään robotin kannalta (syntaksi). Suluissa lihavoituna olevan osan jälkeen näkyy muuttajan tallennusmuoto. /6, s.249/

MoveJ

[ '\ Conc ' , ' ]

[ ToPoint ' := ' ] < expression (**IN**) of robtargget >

[ '\ ID ' := ' < expression (**IN**) of identno > ] , ' ,

[ Speed ' := ' ] < expression (**IN**) of speeddata >

[ '\ V ' := ' < expression (**IN**) of num > ]

| [ '\ ' := ' < expression (**IN**) of num > ] ' , ' ,

[ Zone ' := ' ] < expression (**IN**) of zonedata >

[ '\ Z ' := ' < expression (**IN**) of num > ]

[ '\ Inpos ' := ' < expression (**IN**) of stoppointdata > ] ' , ' ,

[ Tool' :=' ] < persistent (**PERS**) of tooldata >

[ '\` WObj' :=' < persistent (**PERS**) of wobjdata > ] ','

### 5.1.1 Termien selitykset

#### **[\Conc] (Rinnakkainen suoritus)**

#### **Datatyyppi switch**

Myöhemmät käskyt toteutetaan, robotin yhä ollessa liikkeessä. Argumenttia ei yleensä käytetä. Sitä voidaan kuitenkin käyttää, välttääkseen ei haluttuja pysähdyksiä. Nämä pysähdykset johtuvat robotin prosessorin ylikuormituksesta, kun käytetään liikeratapisteitä. Tämä komento on käytännöllinen, kun ohjelmoidut pisteet ovat lähellä toisiaan ja robotti liikkuu kovalla nopeudella. Toinen käyttökohte on, kun kommunikoidaan ulkoisten akselien kanssa ja synkronointi ulkoisen akselin ja robotin välillä ei ole välttämätöntä.

Kun käytetään argumenttia \Conc, jaksossa olevat liikekäskyt ovat rajattu viiteen. Ohjelmaisuus, joka sisältää Storepath-Restorepath argumentteja, \Conc argumentin käyttö on kielletty.

Jos tämä argumentti on jätetty pois ja kohdepiste ei ole pysähdyspaikka, niin seuraavat käskyt suoritetaan hieman ennen, kuin robotti on saavuttanut ohjelmoidun alueen (zone).

#### **ToPoint (kohdepiste)**

#### **Datatyyppi: Robtarget**

Kohdepiste robotille ja ulkoisille akseleille. Se on määritetty nimettynä kohteena (esim. p\_koti) tai ohjelmoitu suoraan ohjeesta (merkattu \*)

**[ID] (Synkronointi ID)****Datatyyppe: identno**

Tätä argumenttia täytyy käyttää Moniajo systeemissä (Useamman robotin välisessä), mikäli koordinoidaan synkronoitua liikuttelua. Ei ole sallittu missään muussa tapauksessa.

Määrätty ID numero pitää olla sama kaikissa yhteistyö ohjelma taskeissa. Id numero takaa sen, etteivät liikkeet eivät sekoitu ajon aikana.

**Speed (nopeus)****Datatyyppe: speeddata**

Nopeusdata, jota sovelletaan liikkeeseen. Nopeusdata määrittää työkalun keskipisteen nopeuden, työkalun uudelleen orientoitumiselle ja ulkoisille akseleille.

**[V] Velocity****Datatyyppe: num**

Tätä argumenttia on käytetty määrittämään nopeutta TCP:lle (Tool Center Point/työkalupiste) suoraan ohjeissa. Se korvaa siten nopeuden, joka on määritetty nopeusdatassa (speed data).

**[T] Time****Data type: num**

Tätä argumenttia käytetään määrittämään kokonaisaika sekunteina jonka robotti käyttää liikkumiseen. Se korvaa siten nopeuden nopeusdatassa (speeddata).

**Zone alue****Data type: zonedata**

Aluedata liikkeelle. Aluedata Kuvioa generoidun liikeradan tarkkuutta (minkä alueen sisällä robotti tekee halutun liikkeen liikepolulla).

**[Z]****Data type: num**

Argumenttia käytetään spesifioimaan paikan virheettömyys TCP:lle suoraan ohjeis-  
sa. Säde on annettu millimetreinä ja se vastaa aluetta joka on määrätty alue datassa  
(zone data). Esimerkiksi Z50 tarkoittaa että robotin työkalupiste kulkee 50 mm sisäl-  
lä ohjelmoidusta pisteestä.

**[Inpos] *In position* (kohteessa)****Data type: stoppointdata**

Argumenttia käytetään spesifioimaan lähestymiskriteeriä robotin TCP:lle pysäytys  
pisteessä. Pysäytys pistedata (stop point data) vastaa alueparametria.

**Tool (työkalu)****Data type: tooldata**

Työkalu, jota käytetään robotin liikkussa. Työkalun keskipiste liikkuu määrättyyn  
kohdepisteeseen.

**[WObj] *Work Object* (työkohde)****Data type: wobjdata**

Työkohde (koordinaatisto systeemi), johon robotin asento ja kohdistus liittyy. Tämä  
argumentti voidaan jättää pois, jolloin asento on yhteydessä maailmakoordinaatis-  
toon. Mikäli käytetään liikkumatonta työkalupistettä (TCP), tai koordinoitua ulkoiset  
akselit ovat käytössä, tämä argumentti pitää olla määritelty.

## 6 Käyttöönotto

Tämä osio pitää sisällään varsinaisen työn osuuden, jossa vanhaa järjestelmää lähdetään muokkaamaan koulun tarkoitukseen sopivaksi. Osio pitää sisällään myös huomioita ja ongelmia mitä projektin aikana ilmaantui.

Tampereen ammattikorkeakoulun ostama käytetty robottisolu oli purettuna Trimaster Oy:n hallissa (Kuvio 26). Solun alkuperäisenä tarkoituksena oli toimia demonstraatio ympäristönä erilaisissa messutapahtumissa. Tämän jälkeen robotti oli purettu solusta, koska sitä oli testattu erilaisissa kappaleenkäsittely tarkoituksissa Trimaster Oy:ssä.

Kehityshanke alkoi osaltani Trimaster Oy:n hallissa, missä robottisolu oli purettuna ja toimimattomana.



**Kuvio 26.** Robotti solusta purettuna





**Kuvio 27.** Tuleva alusta

## 6.1 Robottisolun kokoaminen

Robottisolun kokoamisen aloitin raivaamalla kaikki irrotetut johdot, pultit ynnä muut irtotavarat solun sisäpuolelta pois (Kuvio 27). Tämän jälkeen solun yksi hajanainen seinämä purettiin ja tarpeettomia tukirakenteita poistettiin. Hajanaisen seinän tilalle päätettiin asentaa valoverho esteettömän koulutusikäytön mahdollistamiseksi. Solusta poistettiin myös viisi halogeenivalaisinta ja aikaisemmin käytössä olleita työtasoja.

Robotti oli hallissa pakattuna siirtolavalle (Kuvio 26). Myös robotin ohjauskaappi ja soluosa olivat irrotettu toisistaan (Kuvio 28). Robottisolun kokoamisen aloitin nostamalla hallinosturin avulla robottisolun ohjainkaapin päälle. Tämän jälkeen Itse

robotti nostettiin paikoilleen soluun (Kuvio 29), tehtiin tarvittavat sähkökytkennät ja jatkettiin ylimääräisten osien poistamista (Kuvio 30).

Koska robotti oli ollut käytössä aikaisemmissa projekteissa ja nimenomaan konenäkösovelluksissa, ei I/O kytkentöjä tehty aluksi mitenkään koulua ajatellen loogisemmaksi. Kaikki johdot kytkettiin vanhoihin I/O paikkoihin ja robotin testauksessa käytettiin vanhoja tulojen ja lähtöjen nimiä.

Koululle tullessa robotisolussa oli vielä käytössä konenäkötoiminto, mutta se jätettiin pois tästä työstä aiheen laajuuden takia.



**Kuvio 28.** Robotin ohjauskaappi



**Kuvio 29.** Robotti kiinnitetynä alustansa



**Kuvio 30.** Solun siistimistä

## 6.2 Työkalunvaihtaja

Robottisolun monikäyttöisyyden parantamiseksi robottisoluuun päätettiin hankkia työkalunvaihtaja (Kuvio 31). Tämä RSP robot system:in 20 kg:n nostokyvyn omaava vaihtaja (M0107-1) mahdollistaa useiden erilaisten työkalujen käytön robotisolussa. Työkalunvaihtaja sisältää mahdollisuuden neljälle paineilmalähdölle (M5 liitäntä) ja neljälle elektroniselle liitännälle (M8). Järjestelmässä itsessään on kuusi paineilmalähtä, mutta kaksi niistä on varattu työkalun kiinnittämiseen / irrottamiseen. /7, s. 5/

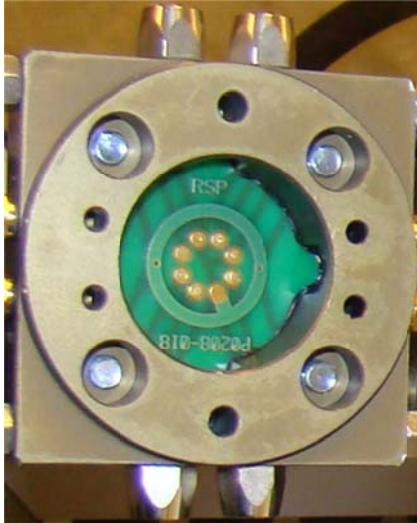


**Kuvio 31.** Työkalunvaihtaja

Robottisoluuun hankittiin neljä adapteriosaa (Kuvio 32), joihin jokaiseen voidaan rakentaa oma työkalu liitäntöjen mahdollistavalla tavalla. Työkaluja suunniteltaessa on huomioitava paineilmalähtöjen toiminta. Liitännät toimivat pareittain, jolloin käy-



tettäessä esimerkiksi vain kahta paineilimaliitintä on kaksi jäljelle jäänyttä paikkaa tukittava, paineilman karkaamisen estämiseksi. Työkalunvaihtajan mitat ja tiedot löytyvät liitteestä 2.



**Kuvio 32.** Työkalunvaihtajan adapteriosa

## 6.3 Työkalut

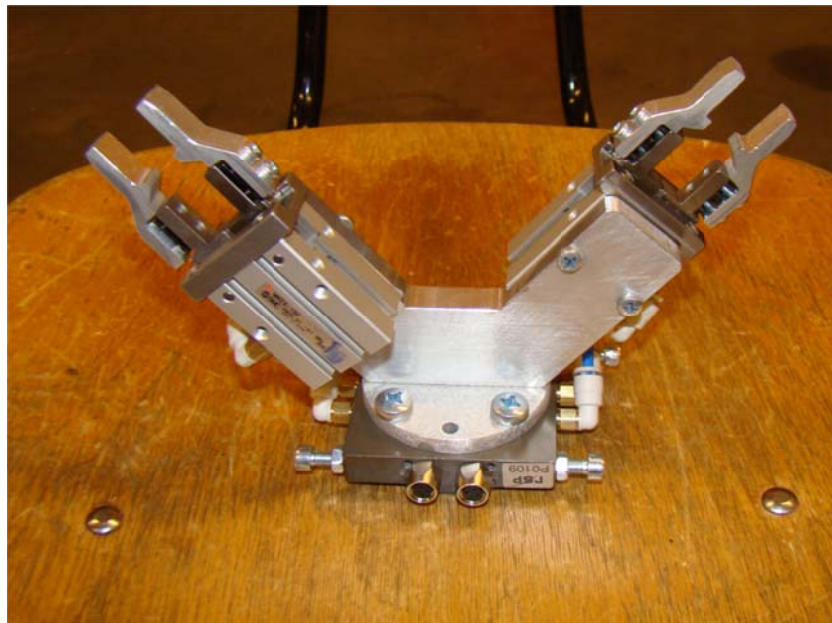
Robottiin suunniteltiin kokonaan uudet työkalut, koska vanhassa rakenteessa oli kiinteästi asennettuna kääntyvävartinen tarrain. Tämän työkalun käyttö työkalunvaihtajan kanssa oli mahdotonta, koska vanha rakenne ei mahdollistanut paineilman liittämistä työkaluun.

### 6.3.1 Kaksoistarrain

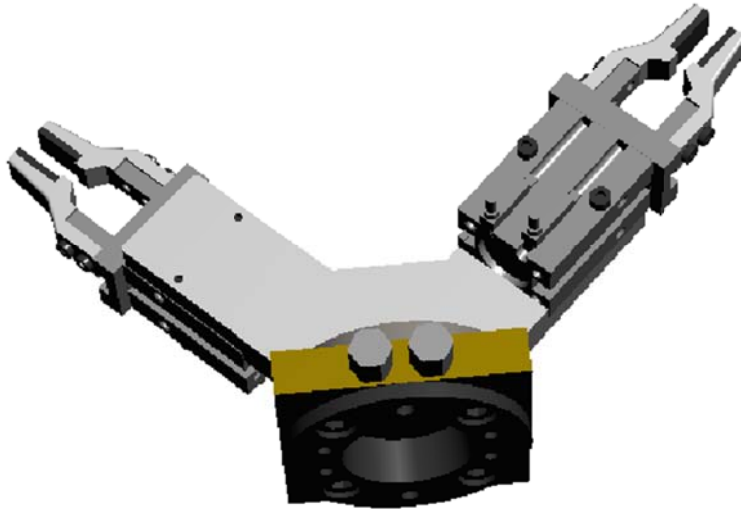
Kaksoistarraimessa (Kuvio 34) tarttujina käytettiin SMC:n MHZ2-16D2 (Kuvio 33) mallisia paineilmatoimisia tarttujia. Kahden tarttujan rakenne työkalussa mahdollistaa opetuskäytössä monipuolisien liikeratojen ja tehokkaan objektien poimimisen harjoittelun oppimisympäristössä. Kaksoistarraimen runko (Kuvio 35) teetettiin laserleikattuna HT-laserilla ja koottiin koululla. Tarttujan tiedot löytyvät liitteestä 3.



**Kuvio 33.** Tarttuja MHZ2-16D2



**Kuvio 34.** Kaksoistarrain



**Kuvio 35.** Kaksoistarraimen suunnitelma

### 6.3.2 Kääntyvätarrain

Robottisolussa oli alun perin vain yksi työkalu, joka oli kiinteästi asennettu kääntyvä tarrain. Tätä työkalua päätettiin käyttää hyväksi niin paljon kuin mahdollista, vaikka sen alkuperäinen rakenne ei suoraan mahdollistanut työkalun käyttämistä työkalunvaihtajan kanssa.

Alkuperäinen kääntyvä tarrain piti aluksi purkaa kokonaan ja suunnitella tarttujan kiinnitys työkalunvaihtajaan sopivaksi. Ratkaisuna päädyttiin tekemään siihen sopivan adapterin (Kuvio 36), mikä mahdollisti vanhan työkalun rungon kiinnittämisen työkalunvaihtajan adapteriosaan. Tämä uusi työkalu (Kuviot 37 ja 38) mahdollistaa esineiden poimimisen erilaisista asennoista verrattuna kiinteään kaksoistarraimeen.

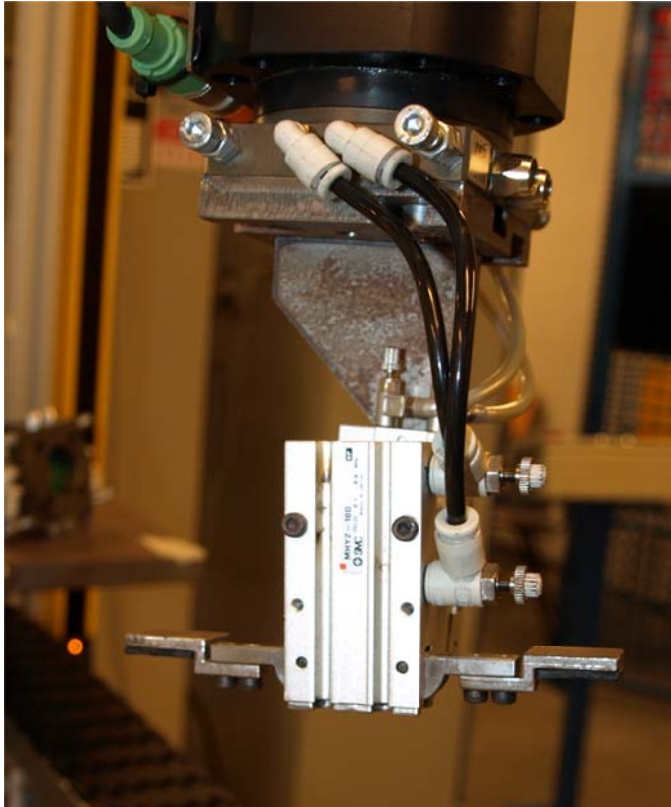


**Kuvio 36.** Adapteri



**Kuvio 37.** Kääntyvä tarrain sivusta

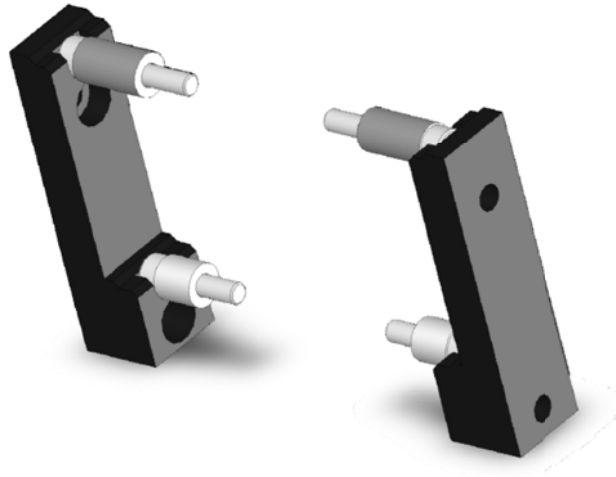




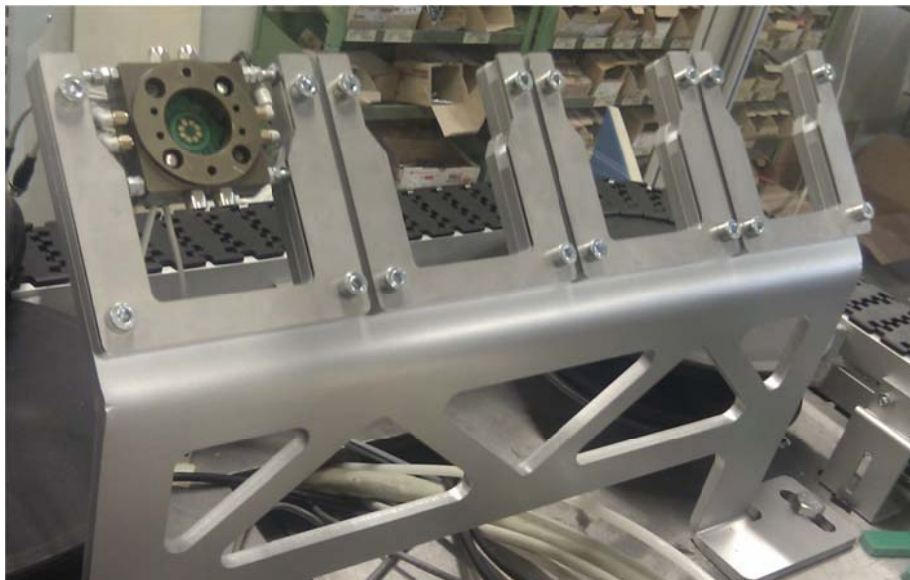
**Kuvio 38.** Kääntyvä tarrain edestä

### 6.3.2 Työkalunvaihtajan teline

Robottisoluun suunniteltiin työkalunvaihtajaa varten teline, johon työkalunvaihtajasta voidaan varastoida erilaisia työkaluja. Suunnittelussa käytettiin hyväksi työkalunvaihtajaan hankittavaksi tarkoitettujen työkalunpitimien rakennetta (Kuvio 39), jonka pohjalta robottisoluun suunniteltiin oma nelipaikkainen työkaluteline (Kuvio 40). Telineen koon määräsi valmiiden kiinnitysreikien paikka robottisolun pohjalevyssä.



**Kuvio 39.** Alkuperäinen työkaluteline



**Kuvio 40.** Nelipaikkainen työkaluteline

### 6.3.3 Paineilmaliitännän lisääminen

Koska robottisolussa ei ollut mitään mielekkäästi toteutettua paineilmaliitäntää, oli soluun mietittävä liitännälle hyvä paikka ja toteutus. Solun ulkopuolella löytyi ohjausyksikön ovien vierestä hyvä paikka paineilmaletkun kiinnittämiseksi. Solun seinään porattiin reikä ja kiinnitettiin sopiva pikaliitin (Kuvio 41). Tämä paikka oli paras, koska se mahdollisti luontevimman kiinnityksen huoltoyksikölle.



**Kuvio 41.** Paineilmaletkun kiinnitys

Solun ohjauskaapin sisäpuolelle kiinnitettiin huoltoyksikkö. Huoltoyksikkö sisältää ilmankuivaimen ja paineenalennusventtiilin (Kuvio 42), joidenka tarkoituksena on estää ylipaineen muodostumisen ja kosteuden kertymisen robotin paineilmajärjestelmään. Paineeksi säädettiin 7 baarin paine. Huoltoyksiköstä johdettiin käyttöpaine robotin paineilmaliihtäntään (Kuvio 43).



**Kuvio 42.** Huoltoyksikkö



**Kuvio 43.** Paineilmaliitântä robotille

## 6.4 Turvaratkaisut

Robottisolun turvallisuusasiat ovat ensiarvoisen tärkeitä, jotta vältytään robotin käsittelyn aikana onnettomuuksilta. Robottisolun turvallista käyttöä lisääviä turvaratkaisuja ovat muun muassa: turvareleet, valoverhot, rajakytkimet ja muut robotin liikealueelle menemistä estävät esteet. Tässä kappaleessa käsitellään robottisolussa käytettäviä turvaratkaisuja.

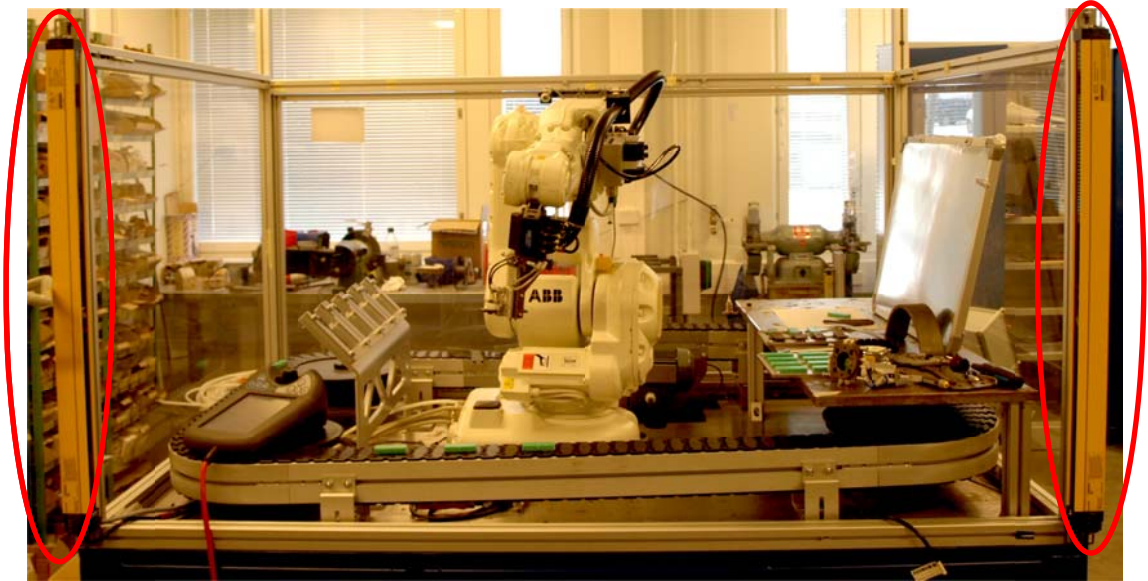
### 6.4.1 Valoverho

Koska robottisolusta poistettiin yksi robotin työskentelyalueelle pääsyn estävä muoviseinä, päätettiin robotin työskentelyalueelle pääsyä ja koulutuskäyttöä helpottamaan hankkia turvallisuusratkaisuksi valoverho (Kuvio 44).

Erilaisia vaihtoehtoja katsottaessa ja robottisolun hankintaan varatussa budjetissa päädyimme rahavaroja hallinnoivan tahon kanssa Omronin MS-4800 malliseen valoverhoon. Tämä turvakategorian luokan neljä valoverhona (sormisuojaus) on riittävä tämän robottisolun suojaamiseksi henkilövahingoilta (maksimi etäisyys 7m, resoluutio 14mm). Valoverhon kytkentäkaavio löytyy liitteestä 4.

**HUOM!**

**Valoverho ei ole käytössä käsiajotilassa**



**Kuvio 44.** Valoverho

#### 6.4.2 Turvarele

Robottisoluun hankittiin ensin Omronin G9SB-200-d mallinen kapea turvareleyksikkö (Kuvio 45). Yksikköä liitettäessä robottiin ongelmaksi muodostui robotin toisen AS (Amergy Stop) piirin laukeaminen. Koska kyseisellä turvareleellä ei saatu molempia AS-piirejä laukaistua samaan aikaan, jätettiin kyseinen turvarele pois tästä työstä. Turvarele lisättiin toisena oppilastyönä robottisoluun. Käytössä olleen turvareleen tiedot löytyvät liitteestä 5.



**Kuvio 45.** Omronin turvareleleysikkö

### 6.4.3 Ovien rajakytkimet

Alkuperäiseen robottisolun oviin oli kytkettynä kaksi ovikytkintä. Näiden tarkoituksena on estää henkilövahingot, mikäli ovet avataan kesken ohjelman suorittamisen. Ovikytkimet jätettiin robottisoluun toimivana turvaratkaisuna. Ovikytkimien mallina on Omron D4DS-15FS (Kuvio 46). Ovirajojen toimintakaavio löytyy liitteestä 6.





**Kuvio 46.** Ovien rajakytkimet

## 6.5 Robotin ohjelmiston päivittäminen

Robotin ohjelma päivitettiin uusimpaan versioon 5.12.1021.01 koska robottisolu oli hankittu sillä edellytyksellä, että siinä on uusimmat saatavilla olevat ohjelmat. Robotin ohjelmaversio oli hankittaessa 5.08.820.

Päivitys toteutettiin yhteistyössä ABB:n Juha Mainion kanssa puhelintukena. Saman päivityksen kanssa robottiin lisättiin uutena optiona worldzones, mikä estää ohjelmallisesti robotin kolaroimisen rakenteiden kanssa. Tällöin robotille määritetään alue, jonka ulko-, tai sisäpuolelle robotin käsivarsi ei saa mennä missään tilanteessa.

Aluksi vanhasta kokoonpanosta otettiin varmuuskopio, jotta ongelmatilanteiden ilmaantuessa saataisiin solu taas toimimaan vanhalla ohjelmistolla.

Tämän jälkeen puhelinohjeistuksen kanssa loin robot studio System Builder kohdan avulla uuden järjestelmä kokoonpanon. Uuden aktivointiavaimen sain puhelimen välityksellä. Uuden järjestelmän latauksen jälkeen robotti käynnistettiin I-startilla, jolloin robotin uusi ohjelmisto tuli toimivaksi. Päivitetyn ohjelmiston tiedot löytyvät liitteestä 7. /8/



## 7 Havaintoja ja ongelmia

Robottisolun käyttöönotossa ilmeni useita ongelmia. Ongelmat johtuivat pääsääntöisesti siitä, että käyttöönotossa pyrittiin käyttämään mahdollisimman paljon olemassa olevia asetuksia ja välineitä.

Robottisolussa oli oikaistu useissa I/O kytkennöissä, koska robottisolua oli käytetty pääsääntöisesti messuesittelyissä. Ovien turvakytkimet olivat kytketty sarjaan, vaikka molempien ovien kytkimet olisi pitänyt kytkeä erikseen robotille. Näin oli toimitettu sen takia, että robotille ei tarvinnut asentaa muun muassa turvarelettä ja robottisolua oli saatu nopeasti toimintakuntoon messuesittelyjä varten. Koska robottisolussa ei ollut muita turvallisuuteen vaikuttavia kytkimiä, oli robotti silloin helppo ohjelmoida toimimaan oikein.

Nämä kytkennät aiheuttivat ongelmia turvareleen kanssa, kun robottisoluun oli lisätty valoverho. Kun ovi tai valoverho laukaisi hälytyksen kesken robotin automaattiajon, laukaisi se vain toisen AS (Amergy stop) piirin. Tällöin robotti ilmoitti virheilmoituksena että vain toinen piiri on lauennut, vaikka molemmat näyttivät olevan lauenneet (Kuvio 47). Syy ilmoitukseen oli, että molempien piirien piti laueta samalla hetkellä.

Tätä ongelmaa ei saatu korjattua ensimmäiseksi ostetulla turvareleellä, koska molemmat AS piirit tulisi laukaista yhtä aikaa, riippumatta siitä aiheuttiko ovet vai valoverho hälytyksen. Hankittu turvarele sisälsi vain yhden kanavan, jolloin molempien AS piirien laukaisemista ei pystytty toteuttamaan minkäänlaisella kytkennällä kyseisellä releellä (Kuvio 48). Turvarele korvattiin oppilastyönä tehtynä ohjelmoitavalla turvareleellä.

Toinen ongelma oli robottisolussa käytetty konenäköominaisuus. Robotissa oli aikaisempien ohjelmointien seurauksena melkein kolmekymmentä hankalasti nimettyä I/O:ta, joiden nimeämistä ei voinut muuttaa opetuskäyttöön sopiviksi, koska silloin konenäkö ei toiminut oikein. Tämän takia konenäköominaisuus jätettiin tästä työstä pois ja samalla I/O:t saatiin nimettyä opetuskäyttöön paremmin sopiviksi. Ko-

nenäköohjelmisto määräsi tarkasti miten eri laitteet pitää nimetä, jotta se olisi toimiva kyseisellä kokoonpanolla.



**Kuvio 47.** AS piirien merkkivalot



**Kuvio 48.** AS piirien kytkentä paikka

## 8 Yhteenveto

Työn valmistuttua voidaan sanoa, että se oli huomattavasti haasteellisempi, kuin osasin odottaa. Aihe oli minulle entuudestaan vähän vieras ja manuaaleista opiskelu ei heti tuottanut toivottua tulosta. ABB:n Juha mainion pidettyä koulutuspäivän robotin toiminnoista ja ohjelmoinnin peruskomentojen selvittyä alkoi työ vasta toden teolla edistyä.

Työn olisi voinut toteuttaa muillakin tavoilla, mutta koska tarkoitus oli saada robotisolu toimimaan opetusympäristössä käyttäen mahdollisimman paljon robottisolun vanhoja tarvikkeita päädyin nykyiseen ratkaisuun.

## Lähteet

/1/ IRC5 ohjelmoinnin perusteet

/2/ IRCwithFlexPendant

/3/ Introduction to RAPID

/4/ Technical reference manual

/5/ Keskustelut Trimaster Oy:n henkilöstön kanssa

/6/ Rapid reference manual

/7/ Product Manual Swivel with Tool Changer 20 kg

/8/ Puhelinkeskustelu ABB:n Juha Mainion kanssa

## Liitteet

1. Muut käytettävissä olevat datatyypit
2. Työkalunvaihtaja
3. Tarttujan tiedot
4. Valoverhon kytkentä
5. Turvareleen tiedot
6. Ovirajojen toiminta ja tiedot
7. Robotin ohjelmiston tiedot

## Liite 1: Muut käytettävissä olevat datatyypit

aiotrigg - Analog I/O trigger condition

btnres - Push button result data

busstate - State of I/O bus

buttondata - Push button data

byte - Integer values 0 - 255

clock - Time measurement

confdata - Robot configuration data

corrdescr - Correction generator descriptor

datapos - Enclosing block for a data object

dionum - Digital values (0 - 1)

dir - File directory structure

dnum - Double numeric values

errdomain - Error domain

errnum - Error number

errstr - Error string

errtype - Error type

event\_type - Event routine type

exec\_level - Execution level

extjoint - Position of external joints

handler\_type - Type of execution handler

icondata - Icon display data

identno - Identity for move instructions

intnum - Interrupt identity

iodev - Serial channels and files

iounit\_state - State of I/O unit

3.27 jointtarget - Joint position data

listitem - List item data structure

loaddata - Load data

loadidnum - Type of load identification

loadsession - Program load session

mecunit - Mechanical unit

motsetdata - Motion settings data

opcalc - Arithmetic Operator

opnum - Comparison operator

orient - Orientation

paridnum - Type of parameter identification

paridvalidnum - Result of ParIdRobValid

pathrecid - Path recorder identifier

pos - Positions (only X, Y and Z)

pose - Coordinate transformations

progdisp - Program displacement

rawbytes - Raw data

restartdata - Restart data for trigg signals

rmqheader - RAPID Message Queue Message header

rmqmessage - RAPID Message Queue message

rmqslot - Identity number of an RMQ client

robjoint - Joint position of robot axes

robtargt - Position data

shapedata - World zone shape data

signalxx - Digital and analog signals

socketdev - Socket device

socketstatus - Socket communication status

speeddata - Speed data

stoppointdata - Stop point data

stringdig - String with only digits

switch - Optional parameters

symnum - Symbolic number

syncident - Identity for synchronization point

System data - Current RAPID system data settings

taskid - Task identification

tasks - RAPID program tasks

testsignal - Test signal

tooldata - Tool data

tpnum - FlexPendant window number

trapdata - Interrupt data for current TRAP

triggdata - Positioning events, trigg

triggios - Positioning events, trigg



triggstrgo - Positioning events, trigg

tunetype - Servo tune type

uishownum - Instance ID for UIShow

wobjdata - Work object data

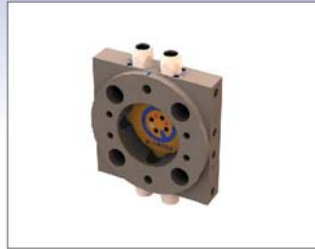
wzstationary - Stationary world zone data

wztemporary - Temporary world zone data

zonedata - Zone data

## Liite 2: Työkalunvaihtaja

## Small Swivels with Tool Changers



### Swivels with Tool Changers

The swivels with Tool Changer combines the best of both worlds to maximize the flexibility and productivity. Proven and robust design ensures the highest possible reliability without limiting the robots performance or working range. These products are the obvious choice in applications that involve welding guns or material handling tools and a lot of robot movement.

### Advantages:

- No limitation of robot working range
- Easier to reach tight positions thanks to fixed hosing and slim design
- Low maintenance
- Forgiving locking device
- Fast docking due to smart design
- Easy to switch connecting pins

### Technical data

Tool Changer	Part no.	Handling capacity (kg)	Electrical channels	Air channels	Weight (kg)	Corresponding Tool Attachment	Part no.	Weight (kg)	Max moment (Nm)
STC20-4	P0106	20	-	4	0.8	TA20-4	P0102	0.3	150
STC20-4E	P0108	20	6	4	1.0	STA20-4E	P0109	0.4	150

P0141 Square tool attachment bracket

### Electrical channels

Max voltage	60V *
Max current	1 A *
Connections	M12 / Cable

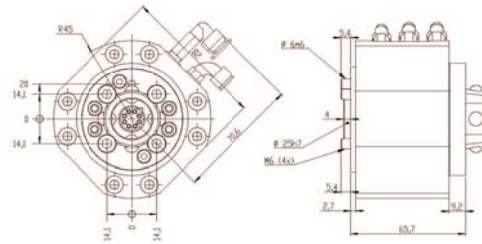
\* Other spec. available on request.

### Compressed air

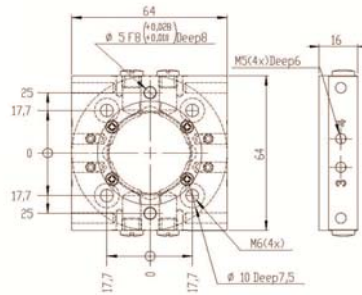
Max. compressed air	-1 to 10 bar
Tool change function	6-10 bar
Air flow, unregulated outflow	150 l/min, free, Pin = 6.0 bar
Air quality	Oil-clean and waterless filtered air, max 25 µm
Air connections	M5

## Dimensional drawings

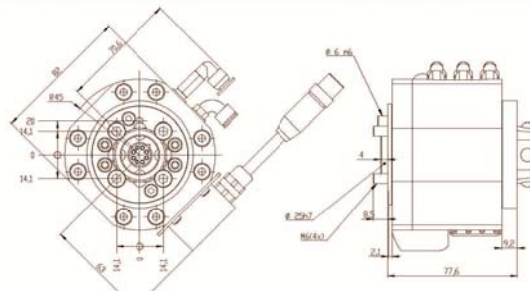
STC20-4:



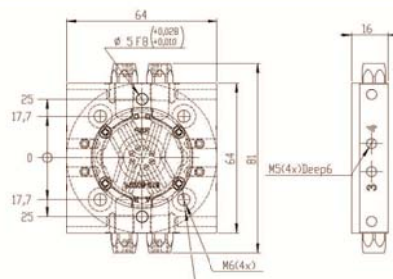
TA20-4:



STC20-4E:



STA20-4E:



B0102-1, Jan 2006

RobotSystem Products AB offers complete, standardized media supply solutions for most robot-based automation. We design, manufacture, and market a widerange of standard robot peripherals such as tool changers, swivels, hose packages, vacuum grippers and die-castings systems throughout the world. Visit us at [www.robotssystemproducts.com](http://www.robotssystemproducts.com)

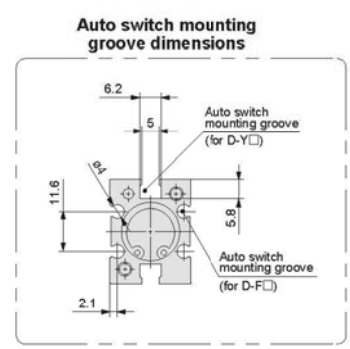
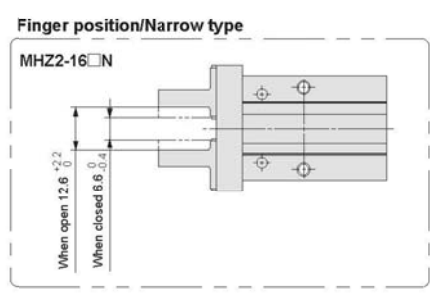
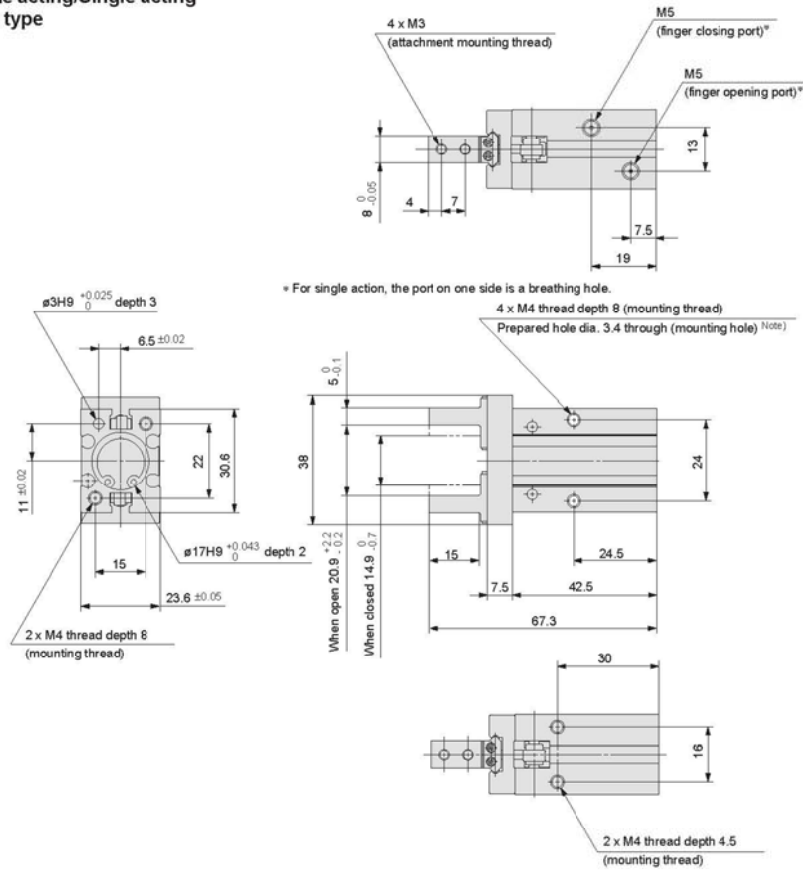
Liite 3: Tarttujan tiedot

Series **MHZ2**

Dimensions

MHZ2-16□  
 Double acting/Single acting  
 Basic type

Scale: 65%



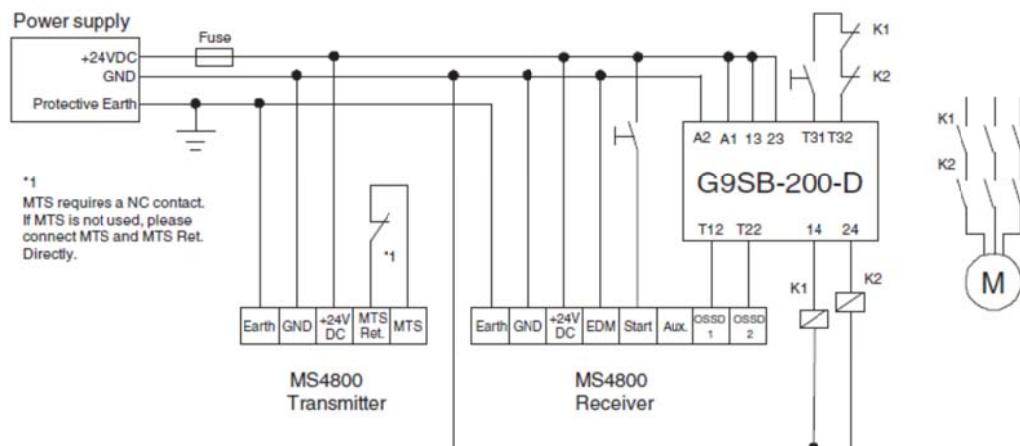
Note) When using D-Y59, D-Y69 and D-Y7 type auto switches, through hole mounting is not possible.

## Liite 4: Valoverhon kytkentä

Receiver cable			
Pin No. 1	Input power	Brown	+24VDC
Pin No. 2	Input power	Blue	GND
Pin No. 3	Input power	Green	Functional Earth
Pin No. 4	Output signal	White	OSSD2
Pin No. 5	Input signal	Yellow	Start
Pin No. 6	Input signal	Red	EDM
Pin No. 7	Output signal	Pink	Aux
Pin No. 8	Output signal	Black	OSSD1

Transmitter cable			
Pin No. 1	Input power	Blue	GND
Pin No. 2	Input power	Brown	+24VDC
Pin No. 3	Input test	White	MTS
Pin No. 4	Input test	Black	MTS Return
Pin No. 5	Input power	Green	Functional Earth



## Liite 5: Turvareleen tiedot

OMRON

## Safety Relay Unit

**G9SB****Ultra Slim Safety Relay Unit**

- Models of width 17.5 mm available with 2 or 3 safety contacts. Models of width 22.5 mm with 3 safety contacts and auxiliary contact are also available.
- Conforms to EN standards. (TÜV approval)
- DIN track mounting possible.
- Slim size controller dedicated for safety sensors F3SN, F3SH, F3S-B, F3S-TGR, F3SL, F3S-J



## Ordering Information

Main contacts	Auxiliary contact	Number of input channels	Reset mode	Input type	Rated voltage	Model	Category (EN954-1)	Size
DPST-NO 2 safety contacts	None	2 channels	Auto-reset	Inverse	24 VAC/VDC	G9SB-2002-A	4	17.5 mm
		1 channel or 2 channels		+ common		G9SB-200-B		
		2 channels	Manual-reset	Inverse		G9SB-2002-C		
		1 channel or 2 channels		+ common		G9SB-200-D		
3PST-NO 3 safety contacts	SPST-NC	None (direct breaking)	Auto-reset	---	24 VDC	G9SB-3010	3	17.5 mm
		2 channels		Inverse	24 VAC/VDC	G9SB-3012-A	4	22.5 mm
		1 channel or 2 channels		+ common		G9SB-301-B		
		2 channels	Inverse	G9SB-3012-C				
		1 channel or 2 channels	Manual-reset	+ common		G9SB-301-D		

## Model Number Legend

G9SB-□□□□□□ □  
1 2 3 4 5 6

**1. Function**

None: Emergency stop

**2. Contact Configuration (Safety Output)**

2: DPST-NO

3: 3PST-NO

**3. Contact Configuration (OFF-delay Output)**

0: None

**4. Contact Configuration (Auxiliary Output)**

0: None

1: SPST-NC

**5. Input Configuration**

None: 1-channel or 2-channel input possible

0: None (direct breaking)

2: 2-channel input

**6. Miscellaneous**

A: Auto-reset, inverse input

B: Auto-reset, + common input

C: Manual-reset, inverse input

D: Manual-reset, + common input

G9SB

G-123

G9SB

## Specifications

## Ratings

## Power Input

Item	G9SB-200□□	G9SB-3010	G9SB-301□□
Power supply voltage	24 VAC/VDC; 24 VAC, 50/60 Hz, or 24VDC 24 VDC: 24 VDC		
Operating voltage range	85% to 110% of rated power supply voltage		
Power consumption	1.4 VA/1.4 W max.	1.7 W max.	1.7 VA/1.7 W max.

## Inputs

Item	G9SB-200□□	G9SB-3010	G9SB-301□□
Input current	25 mA max.	60 mA max. (See note.)	30 mA max.

Note: Indicates the current between terminals A1 and A2.

## Contacts

Item	G9SB-200□□	G9SB-3010	G9SB-301□□
	Resistive load (cos $\phi$ =1)		
Rated load	250 VAC, 5 A		
Rated carry current	5 A		

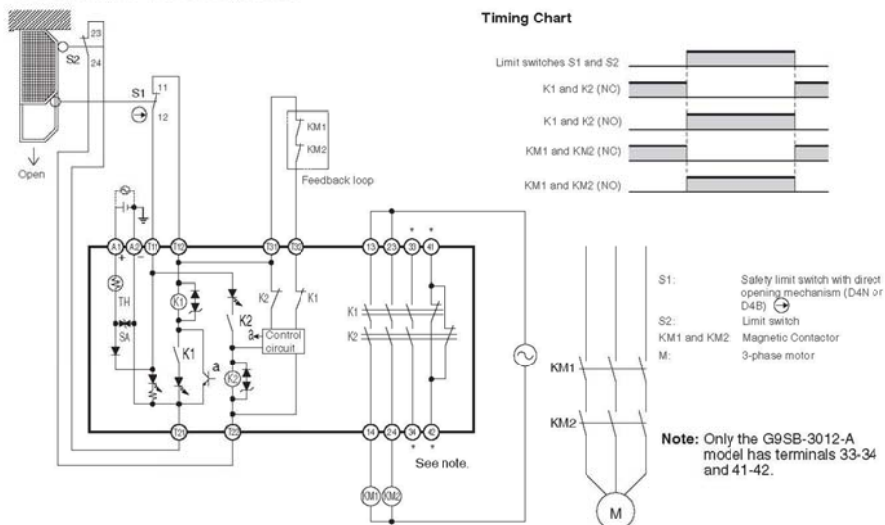
## Characteristics

Item	G9SB-200□□	G9SB-3010	G9SB-301□□
Contact resistance (See note 1.)	100 m $\Omega$		
Operating time (See note 2.)	30 ms max.		
Response time (See notes 2 and 3.)	10 ms max.		
Insulation resistance (See note 4.)	100 M $\Omega$ min. (at 500 VDC)		
Dielectric strength	Between different outputs	2,500 VAC, 50/60 Hz for 1 min	
	Between inputs and outputs		
	Between power inputs and outputs		
Vibration resistance	10 to 55 to 10 Hz, 0.375-mm single amplitude (0.75-mm double amplitude)		
Shock resistance	Destruction	300 m/s <sup>2</sup>	
	Malfunction	100 m/s <sup>2</sup>	
Durability	Mechanical	5,000,000 operations min. (at approx. 7,200 operations/hr)	
	Electrical	100,000 operations min. (at approx. 1,800 operations/hr)	
Minimum permissible load (reference value)	5 VDC, 1 mA		
Ambient operating temperature	-25 to 55°C (with no icing or condensation)		
Ambient operating humidity	35% to 85%		
Terminal tightening torque	0.5 N·m		
Weight	Approx. 115 g	Approx. 135 g	Approx. 120 g
Approved standards	EN954-1, EN60204-1, UL508, CSA C22.2 No. 14		
EMC	EMI: EN55011 group 1 class A EMS: EN50082-2		

- Note: 1. The contact resistance was measured with 1 A at 5 VDC using the voltage-drop method.  
 2. The bounce time is not included in the figure for operating time.  
 3. The response time is the time it takes for the main contact to open after the input is turned OFF.  
 4. The insulation resistance was measured with 500 VDC at the same places that the dielectric strength was checked.

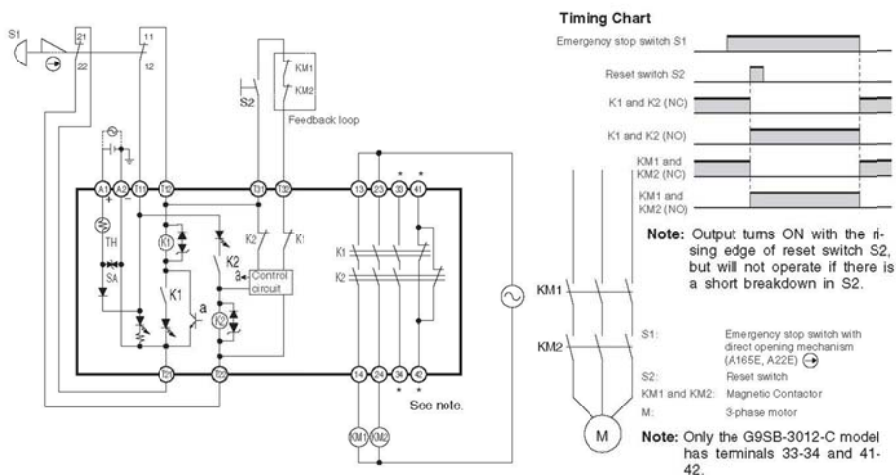
Application Examples

G9SB-2002-A (24 VAC/VDC) or G9SB-3012-A (24 VAC/VDC) with 2-channel Limit Switch Input/Auto-reset



Note: 1. External connections and timing charts for G9SB-200-B/301-B models are the same as those for G9SB-2002-A/3012-A models.  
 2. This circuit conforms to EN954-1 Safety Category 4.

G9SB-2002-C (24 VAC/VDC) or G9SB-3012-C (24 VAC/VDC) with 2-channel Emergency Stop Switch Input/Manual-reset

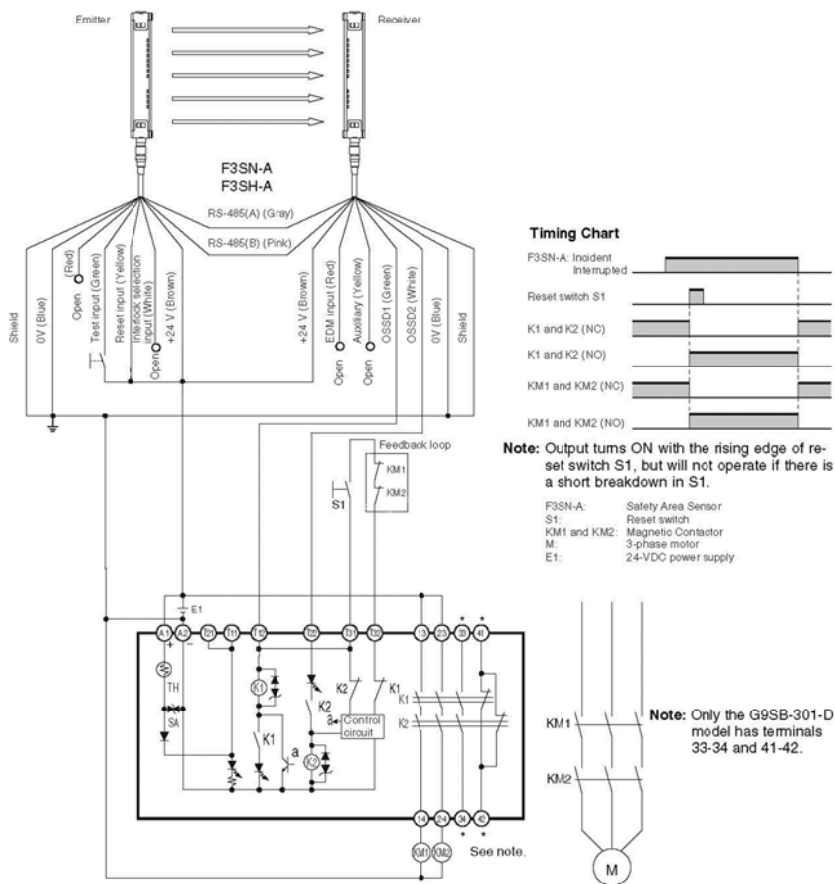


Note: 1. External connections and timing charts for G9SB-200-D/301-D models are the same as those for G9SB-2002-C/3012-D models.  
 2. This circuit conforms to EN954-1 Safety Category 4.

G9SB

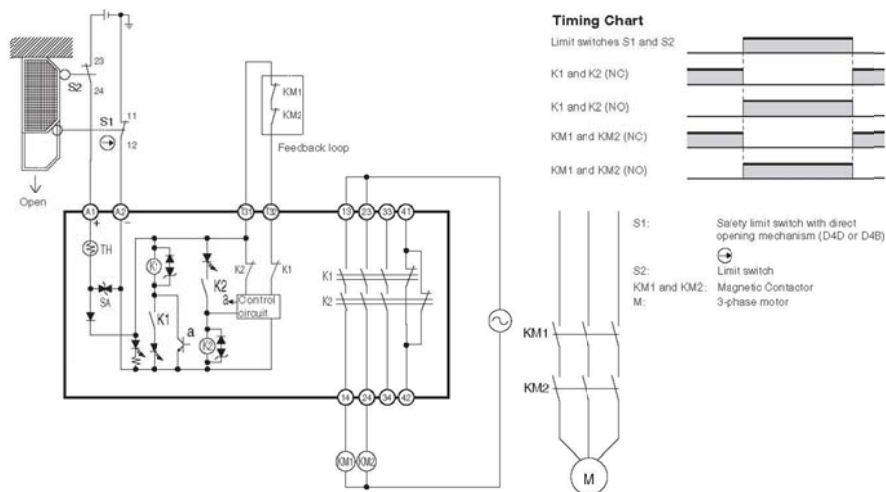


G9SB-200-D (24 VAC/VDC) or G9SB-301-D (24 VAC/VDC) with 2-channel Safety Area Sensor/Manual-reset



Note: This circuit conforms to EN954-1 Safety Category 4.

G9SB-3010 (24 VDC) with 2-channel Limit Switch Input/Auto-reset

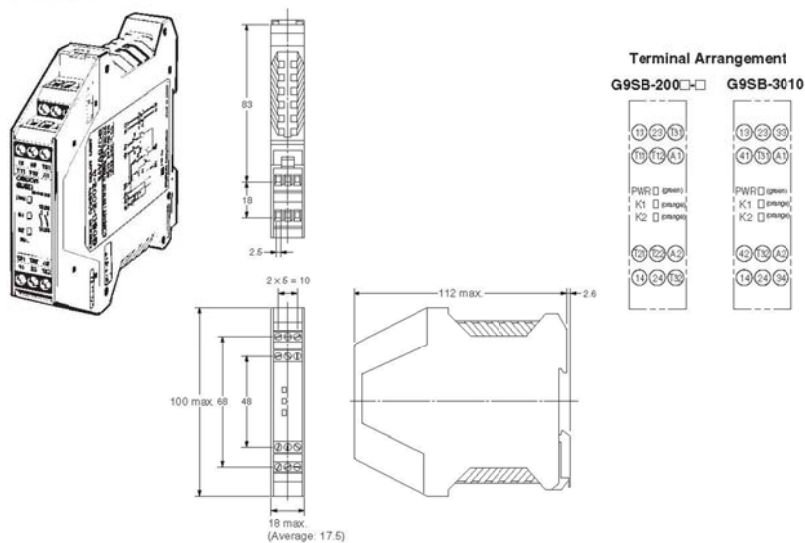


Note: This circuit conforms to EN954-1 Safety Category 3.

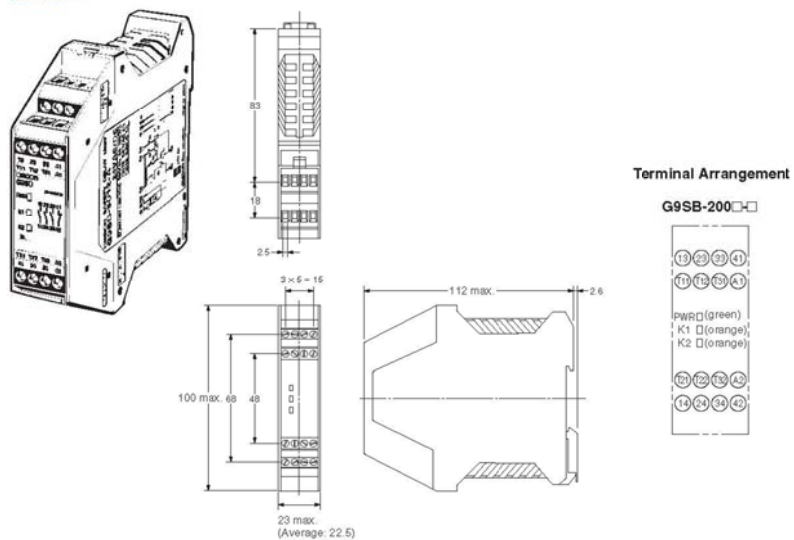
G9SB

Dimensions

G9SB-200□□  
G9SB-3010



G9SB-301□□

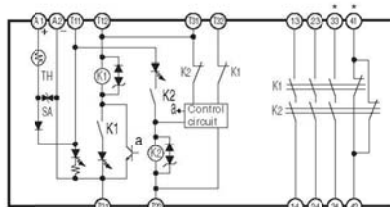


## Installation

## Internal connections

G9SB-2002-A/C (24 VAC/VDC)

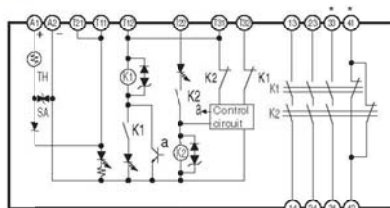
G9SB-3012-A/C (24 VAC/VDC)



See note 2.

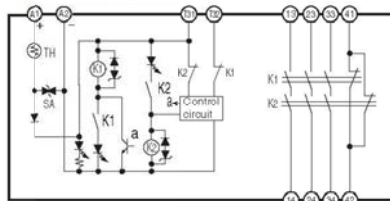
G9SB-200-B/D (24 VAC/VDC)

G9SB-301-B/D (24 VAC/VDC)



See note 2.

G9SB-3010 (24 VDC)



Note: 1. For 1-channel input with G9SB-□□□-B/D models, short terminals T12 and T22.  
It is not possible to wire G9SB-□□□2-A/C models for 1-channel input.

2. Only G9SB-301□□ models have terminals 33-34 and 41-42.

## Precautions

### Wiring

Turn OFF the G9SB before wiring. Do not touch the terminals of the G9SB while the power is turned ON, because the terminals are charged and may cause an electric shock.

Use the following to wire the G9SB.

Stranded wire: 0.2 to 2.5 mm<sup>2</sup>

Solid wire: 0.2 to 2.5 mm<sup>2</sup>

Tighten each screw to a torque of 0.5 to 0.6 N·m, or the G9SB may malfunction or generate heat.

External inputs connected to T11 and T12 or T21 and T22 of the G9SB must be no-voltage contact inputs.

### Applicable Safety Category

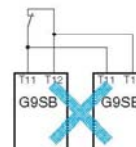
G9SB-200□□/301□□ meet the requirements of Safety Category 4 of the EN954-1 standards when they are used as shown in the examples provided by OMRON. Relays may not meet the standards in some operating conditions. The G9SB-3010 can be applied to Safety Category 3 of the EN954-1 using double breaking. The applicable safety category is determined from the whole safety control system. Make sure that the whole safety control system meets EN954-1 requirements.

### Mounting Multiple Units

When mounting multiple Units close to each other, the rated current will be 3 A. Do not apply a current higher than 3 A.

### Connecting Inputs

If using multiple G9SB models, inputs cannot be made using the same switch. This is also true for other input terminals.



### Earth Short

A positive thermistor is built into the G9SB circuits, so you can detect earth short breakdowns and breakdown shorts between channel 1 and channel 2. (Detection of breakdown shorts between channel 1 and channel 2 is supported for G9SB-2002□□/3012□□ models only.)

Note: In order to detect earth short breakdowns, connect the minus side of the power supply to ground.

ALL DIMENSIONS SHOWN ARE IN MILLIMETERS.

To convert millimeters into inches, multiply by 0.03937. To convert grams into ounces, multiply by 0.03527.

## Liite 6: Ovirajojen toiminta ja tiedot

## Operation

## ■ CONTACT FORM (DIAGRAMS SHOW STATE WITH KEY INSERTED)

Model	Contact	Diagrams	Remarks	
D4DS-□5□S	1NC/1NO			Only NC contact 11-12 has an approved positive opening mechanism. The terminals 11-12 and 23-24 can be used as unlike poles.
D4DS-□A□S	2NC			NC contacts 11-12 and 21-22 have an approved positive opening mechanism. The terminals 11-12 and 23-24 can be used as unlike poles.

Note: Terminals are numbered according to EN50013 and contacts are marked according to EN60947-5-1.

## ■ CHARACTERISTICS

Degree of protection (see note 1)	IP65 (EN60947-5-1)
Life expectancy (see note 2)	Mechanical: 1,000,000 operations min. Electrical: 150,000 operations min.
Operating speed	0.1 m/s to 0.5 m/s
Contact gap	2 x 2.0 mm min.
Operating frequency	30 operations/minute min.
Positive opening force (see note 3)	58.84 N min.
Positive opening travel (see note 3)	10 mm min.
Insulation resistance	100 MΩ min. (at 500 VDC) between terminals of same polarity, between terminals of different polarity, between each terminal and ground, and between each terminal and non-current-carrying metal part
Contact resistance	25 mΩ max. (initial value)
Dielectric strength	$U_{imp}$ 4 kV between terminals of same polarity, between terminals of different polarity, and between each terminal and non-current-carrying metal part (EN60947-5-1)
Rated insulation voltage ( $U_i$ )	400 V (EN60947-5-1)
Conditional short-circuit current	100 A (EN60947-5-1)
Short-circuit protective device (SCPD)	10 A, fuse type gI or gG (IEC269)
Switching overvoltage	1,500 V max. (EN60947-5-1)
Pollution degree (operating environment)	3 (EN60947-5-1)
Conventional enclosed thermal current ( $I_{the}$ )	10 A (EN60947-5-1)
Protection against electric shock	Class II (double insulation)
Vibration resistance	Malfunction: 10 to 500 Hz, 0.65-mm single amplitude with an imposed acceleration of 100 m/s <sup>2</sup> max.
Shock resistance	Destruction: 1,000 m/s <sup>2</sup> min. Malfunction: 300 m/s <sup>2</sup> min.
Ambient temperature	Operating: -30°C to 70°C (with no icing)
Ambient humidity	Operating: 95% max.
Weight	Approx. 76 g (for D4DS-15FS)

Note: 1. Although the Switch casing resists dust, oil, and water, make sure that the keyhole on the head is free from dust, oil, water, and chemical, or the D4DS may wear out, break, or malfunction.

2. Life expectancy values are calculated at an operating temperature of 5°C to 35°C, and an operating humidity of 40% to 70%. Contact your OMRON sales representative for more detailed information on other operating environments.

3. These figures are minimum requirements for safe operation.

Liite 7: Robotin ohjelmiston tiedot

**System Name:** TAMK\_IRB140

**Serial Number:** 14-50779

**Using Media:**

**Media:**

Name: RobotWare

Version: 5.12.1021.01

**Control Module:**

Key: 4UEEEEPIc4PSEIEUVEJ-kn

Key Signature: 89

Additional Languages:

644-11 Finnish

645-8 Swedish

**Options:**

RW Control module key

RobotWare OS and English

644-11 Finnish

- 645-8 Swedish

709-x DeviceNet

- 709-1 Master/Slave Single

642-1 Prepared for PickMaster

- 1 conveyor

- 1 IO board

608-1 World Zones

613-1 Collision Detection

**Drive Module:**

Key: B8Ie3SzKV23

Key Signature: 90

Manipulator:

ABB standard manipulator

Add axes IRB/drive module 140/1400

IRB 140 Type B

**Options:**

RW Drive module 1 key

ABB standard manipulator

- *IRB 140 Type B*

- *140 Type B*

Add axes IRB/drive module 140/1400

- *RC1 No add drive*

**Additional Options**