

RAPID-ohjelmointikielen peruskäskyjä ja –rakenteita

Ohjelmamoduli, Mainmodule

Ohjelmadata

- Num Numeeriset arvot (rekisterit, laskurit)
- Bool Loogiset arvot (tosi/epätosi, true/false)
- Robtarget Paikoitusdata
- Tooldata Työkaludata, käytettävien työkalujen määrittelydata
- Wobjdata Työkohteet, käyttäjän koordinaatistot

Aliohjelmat (rutiini)

Kaikki ohjelmat kannattaa jakaa osiin ja luoda jokaiselle osatehtävälle oma ohjelma. Näitä ohjelmia kutsutaan rutiineiksi ja ne voidaan kutsua toteutettavaksi tarpeen mukaan.

Pääohjelma (Main, päärutiini)

Päärutiinissa on ainoastaan aliohjelmakutsuja, joten ohjelman seuraaminen on selkeää. Samankaltaiset toiminnot, kuten tarttujan ohjaus on hyvä tehdä omana rutiinina, jota kutsutaan tarvittaessa. Ohjelmien testiajo voidaan myös suorittaa rutiini kerrallaan.

Jokainen robottiohjelma on ohjelmamoduli, joka sisältää ohjelmairutiinit (main- ja alirutiinit) sekä Data-kentän.

Seuraavalla sivulla on esimerkki ohjelmamodulista, Mainmodule.

CONST robtarget

Target_60:=[[142.931537459,48.444245771,71.251698823],[0.068319388,0.379528605,-
0.921962152,0.035725181],[1,0,0,1],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Target_20:=[[145.000008361,100.000252151,-
0.000356583],[0.068319592,0.379528716,-

0.921962099,0.035725007],[1,0,0,1],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Target_30:=[[70.110213446,100.000272536,-
0.000314812],[0.068319554,0.37952868,-

0.921962116,0.035725005],[1,0,0,1],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Target_40:=[[70.110233003,48.444219569,-
0.00033104],[0.068319491,0.379528688,-

0.921962118,0.035724982],[1,0,0,1],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Target_50:=[[142.931523986,48.444221567,-
0.000309126],[0.068319467,0.379528651,-

0.921962133,0.035725043],[1,0,0,1],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget p10:=[[-130.190604554,643.824912005,187.820224077],[0.088799664,-
0.413015368,0.905434119,-0.041496774],[1,0,0,1],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Target_70:=[[142.93,48.44,71.25],[0.0683327,0.379533,-
0.92196,0.0357258],[1,0,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget p_home:=[[142.93,48.44,71.24],[0.0683304,0.379532,-
0.92196,0.0357285],[1,0,0,1],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget point:=[[143.24,49.21,130.52],[0.0683202,0.37953,-
0.921963,0.0357241],[1,0,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget point10:=[[142.56,47.54,0.36],[0.0683238,0.379535,-
0.92196,0.0357165],[1,0,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR num nappula;

VAR num kerrat;

PROC main()

MoveJ p_home,v300,z100,t_pencil\WObj:=w_paperi;

! viesti ja valinta käyttäjälle, riippuen mitä painiketta käyttäjä painaa saa toiminto

! arvon 1 – 5, joka tallentuu määritellylle muuttujanimelle, tässä nimenä ”nappula”,

! viesti ja painikkeiden päälle tulevat tekstit tulee olla lainausmerkkien sisällä

TPReadFK nappula,"VALITSE !","Nelio","Ympyra","Kolmio","","Lopetus";

! testataan annetun valinnan mukainen toiminta, CASE 1 toimii kun painike 1

! on valittu, CASE 2 liittyy painikkeeseen 2 jne. DEFAULT toimii, jos valitaan

! tyhjä painike, jolle ei ole määriteltä omaa CASEa

TEST nappula

CASE 1:

! kysymys käyttäjälle, vastaus tallentuu muuttujaksi ”kerrat” tässä tapauksessa

TPReadNum kerrat,"KUINKA MONTA ?";

FOR i FROM 1 TO kerrat DO

Nelio;

ENDFOR

CASE 2:

ympyra;

CASE 3:

kolmio;

Konetekniikka

DEFAULT:

TPWrite "Yrita uudelleen !";

WaitTime 3;

ENDTEST

ENDPROC

PROC Nelio()

MoveL Target_60,v300,fine,t_pencil\WObj:=w_paperi;

MoveL Target_20,v300,fine,t_pencil\WObj:=w_paperi;

MoveL Target_30,v300,fine,t_pencil\WObj:=w_paperi;

MoveL Target_40,v300,fine,t_pencil\WObj:=w_paperi;

MoveL Target_50,v300,fine,t_pencil\WObj:=w_paperi;

MoveL Target_20,v300,fine,t_pencil\WObj:=w_paperi;

MoveL Target_60,v300,z100,t_pencil\WObj:=w_paperi;

ENDPROC

PROC ympyra()

MoveL Target_60,v300,fine,t_pencil\WObj:=w_paperi;

MoveL Target_20,v300,fine,t_pencil\WObj:=w_paperi;

MoveC Target_30,Target_40,v300,fine,t_pencil\WObj:=w_paperi;

MoveC Target_50,Target_20,v300,fine,t_pencil\WObj:=w_paperi;

MoveL Target_60,v300,z100,t_pencil\WObj:=w_paperi;

ENDPROC

PROC kolmio()

MoveJ point,v200,z50,t_pencil\WObj:=w_paperi;

MoveL point10,v200,z50,t_pencil\WObj:=w_paperi;

MoveL Offs(point10,50,25,0),v200,z50,t_pencil\WObj:=w_paperi;

MoveL Offs(point10,25,-25,0),v200,z50,t_pencil\WObj:=w_paperi;

MoveL point10,v200,z50,t_pencil\WObj:=w_paperi;

ENDPROC

ENDMODULE

1. Liikekäskyt

*MoveJ *,v200,z10,kyna;*

Siirto epälineaarista rataa pitkin valittuun pisteeseen, robotti valitsee optimaalisen radan mahdollisimman nopean liikkeen aikaansaamiseksi. Käytetään aina kun ei tarvita täsmälleen lineaarista siirtymää.

a) Liikkeen kohdepiste ilmoitetaan:

1) *

- käytetään vain vähemmän merkitsevien välipisteiden opetuksessa
- sitä ei voida enää kutsua muissa käskyissä myöhemmin, koska sitä ei voida tunnistaa, tosin se voidaan kopioida (copy/paste) ja siten käyttää uudelleen mutta sotkeentumisen vaara muihin tähtipisteisiin on olemassa.

2) **pisteelle annetaan nimi**

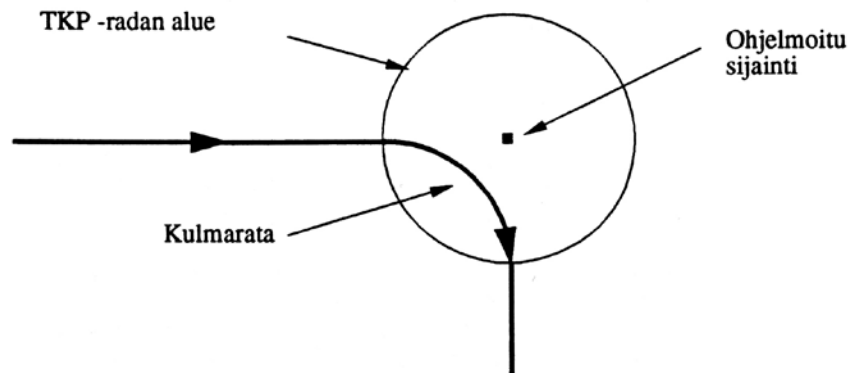
- nimen tulisi viitata kohteeseen (esim. p1, koti, alusta, nurkka, syl.nouto jne.)
- voidaan käyttää useammassakin liikelausekkeessa
- helppo hienosäätää paikoitusta jälkikäteen esim. lisäämällä z-koordinaattiin 0.6 mm, isoissa muutoksissa kannattaa opettaa uudelleen ModPos-komennolla
- Offs-lauseiden yhteydessä tarvitaan viitepisteinä

b) Nopeus v

- v200 (=200 mm/s) määrittää robotin nopeuden (TKP:n nopeus). Seuraavana on luettelo järjestelmämodulin BASE valikossa olevista arvoista, joista valitaan sopiva

<u>Nimi</u>	<u>TKP nopeus</u>	<u>Orientaatio</u>	<u>Lin. ulk. aks.</u>	<u>Pyör. ulk. aks.</u>
v5	5 mm/s	500°/s	5000 mm/s	1000°/s
v10	10 mm/s	500°/s	5000 mm/s	1000°/s
v20	20 mm/s	500°/s	5000 mm/s	1000°/s
v30	30 mm/s	500°/s	5000 mm/s	1000°/s
v40	40 mm/s	500°/s	5000 mm/s	1000°/s
v50	50 mm/s	500°/s	5000 mm/s	1000°/s
v60	60 mm/s	500°/s	5000 mm/s	1000°/s
v80	80 mm/s	500°/s	5000 mm/s	1000°/s
v100	100 mm/s	500°/s	5000 mm/s	1000°/s
v150	150 mm/s	500°/s	5000 mm/s	1000°/s
v200	200 mm/s	500°/s	5000 mm/s	1000°/s
v300	300 mm/s	500°/s	5000 mm/s	1000°/s
v400	400 mm/s	500°/s	5000 mm/s	1000°/s
v500	500 mm/s	500°/s	5000 mm/s	1000°/s
v600	600 mm/s	500°/s	5000 mm/s	1000°/s
v800	800 mm/s	500°/s	5000 mm/s	1000°/s
v1000	1000 mm/s	500°/s	5000 mm/s	1000°/s
v1500	1500 mm/s	500°/s	5000 mm/s	1000°/s
v2000	2000 mm/s	500°/s	5000 mm/s	1000°/s
v2500	2500 mm/s	500°/s	5000 mm/s	1000°/s
v3000	3000 mm/s	500°/s	5000 mm/s	1000°/s
vmax	5000 mm/s	500°/s	5000 mm/s	1000°/s

- c) Tarkkuus z
- **z10** on paikoitustarkkuus. Paikoituspisteen ympärillä on säteeltään 10 mm:n ympyrä, mikä tarkoittaa sitä, että tullessaan pisteeseen robotti alkaa jo 10 mm ennen pistettä suuntaamaan seuraavaan pisteeseen ellei kysymyksessä ole päätepiste.
 - **Fine (z0)** tarkoittaa, että robotti käy täsmälleen ohjelmoidussa pisteessä paikoitustarkkuuden puitteissa. Mitä suurempi z-alue on välipisteessä sen tasaisempi robotin liike saavutetaan. Siis välipisteisiin suuri z-arvo ja päätepisteeseen tarkoitusta vastaava arvo kuten *fine*.



Alla on luettelo z-arvoista jotka ovat robotin BASE modulissa:

Nimi	TKP-rata	TKP -liike			Ei TKP -liike	
		Orientaatio	Ulk. aks.	Orientaatio	Lin. aks.	Pyör. aks.
z1	1 mm	1 mm	1 mm	0.1 °	1 mm	0.1 °
z5	5 mm	8 mm	8 mm	0.8 °	8 mm	0.8 °
z10	10 mm	15 mm	15 mm	1.5 °	15 mm	1.5 °
z15	15 mm	23 mm	23 mm	2.3 °	23 mm	2.3 °
z20	20 mm	30 mm	30 mm	3.0 °	30 mm	3.0 °
z30	30 mm	45 mm	45 mm	4.5 °	45 mm	4.5 °
z40	40 mm	60 mm	60 mm	6.0 °	60 mm	6.0 °
z50	50 mm	75 mm	75 mm	7.5 °	75 mm	7.5 °
z60	60 mm	90 mm	90 mm	9.0 °	90 mm	9.0 °
z80	80 mm	120 mm	120 mm	12 °	120 mm	12 °
z100	100 mm	150 mm	150 mm	15 °	150 mm	15 °
z150	150 mm	225 mm	225 mm	23 °	225 mm	23 °
z200	200 mm	300 mm	300 mm	30 °	300 mm	30 °

- d) Työkalu
- kynä on käytettävän työkalun nimi. Nimi voidaan valita vapaasti aina kun uusi työkalu opetetaan robotille. Työkalutiedosto sijaitsee USER- modulissa. Työkalu voidaan opettaa myös pelkästään jonkin ohjelman työkaluksi jolloin se pysyy tallessa ohjelman mukana vaikka robotti välillä kylmäkäynnistettäisiin uudelleen. Silloin sitä ei voida käyttää muissa ohjelmissa hyväksi.

Seuraavaksi esimerkkejä joistain käskylauseista ja niiden merkityksestä:

MoveL c1,v500,z5,kyna;

- siirtää lineaarisesti työkalun kynä (TKP) pisteeseen c1 nopeudella 500 mm/s tarkkuudella 5 mm
- käytetään pääsääntöisesti silloin kun liikeradan lineaarisuudella on merkitystä

MoveL Offs(p2,0,0,10), v1000,z50,tool3

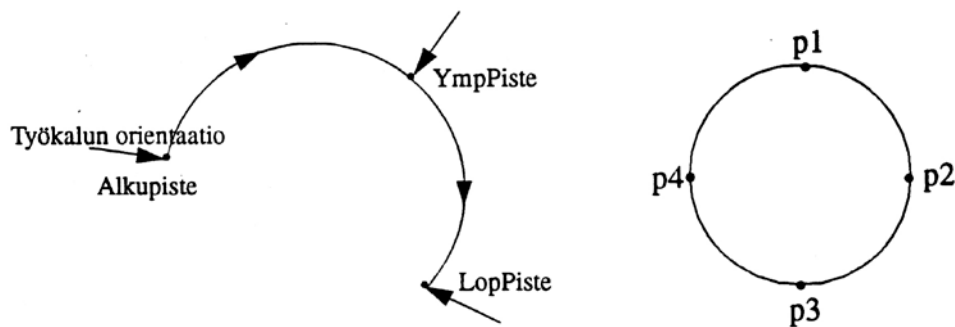
- robotti siirretään pisteeseen, joka on 10 mm pisteestä p2 (z-suunnassa)
- sopii esim. palettisovelluksiin joissa on indeksoituja paikkoja

MoveL Offs(OSA,xmitta,ymitta,0),v50,fine,tarttuja1\WObj:=paletti1;

- työkohde (Work Object) on tässä käskyssä paletti1, johon robotin sijainti on käskyssä suhteutunut
- käytetään silloin kun tarvitaan tiettyyn asentoon koordinaatistoa, esim. indeksipöytä

MoveC p2,p3,v200,z5,kyna;

- työkalu siirretään ympyrärataa pitkin pisteen p2 kautta pisteeseen p3 (puoliympyrä)
- täyden ympyrän suorittaminen tapahtuu seuraavan ohjelman mukaisesti:
MoveL p1,v500,z5,tool2; (robotti siirtää työkalun tool2 pisteeseen p1)
MoveC p2,p3,v200,z5,tool2;
MoveC p4,p1,v200,fine,tool2;



Seuraavaa tapaa voidaan käyttää kun ympyrärata ja siinä oleva piste zp1 tunnetaan tarkasti, näin vältetään opetuksen tuoma epätarkkuus:

MoveC Offs(zp1,20,20,0),Offs(zp1,0,40,0),v5,fine,kyna;
MoveC Offs(zp1,-20,20,0),Offs(zp1,0,0,0),v80,fine,kyna;

2. Aliohjelman rakenne ja kutsu

```
PROC circ()
  MoveJ home,v300,z10,kyna;
  MoveL c1,v500,z5,kyna;
  MoveC c2,c3,v200,z5,kyna;
  MoveC c4,c1,v200,z5,kyna;
  MoveL home,v500,z10,kyna;
  RETURN;
ENDPROC
```

- aliohjelma koostuu käskyjoukosta niin kuin pääohjelmakin mutta se päättyy aina RETURN komentoon jolla palataan takaisin pääohjelmaan. Aliohjelmia kannattaa rakentaa selkeille toimintokokonaisuuksille myös silloin kun rutiinia tarvitaan vain yhden kerran ohjelmakerrossa. Pääohjelma koostuukin lähinnä aliohjelmakutsuista sekä kommunikoinnista käyttäjän kanssa.

- aliohjelma kutsutaan pääohjelmassa nimellään, ja liitetään puolipiste perään

```
PROC main()
  zorr;
  circ;
ENDPROC
```

3. Muuttujat

Muuttujia on useita erilaisia tyyppisiä (numeerinen, boolean ...). Tavallisin muuttuja on numeerinen, ts. jollekin asialle ohjelmassa annetaan lukuarvo, joka vielä ohjelman edetessä saa uusia arvoja.

```
levy:=0;
```

Muuttuja levy on saanut arvon 0. Jos työkiertojen käsittelemiä levyjä lasketaan yksi kerrallaan niin käskyn muoto on seuraava:

```
levy:=levy+1;
```

Muuttuja voidaan sisällyttää myös Offset-funktioon:

```
xsiirto:=xsiirto+200
...
...
MoveJ Offs(qq,xsiirto,0,20),v200,z5,imukuppi;
```

Ohjelman käyttöä voidaan helpottaa ja monipuolistaa käsiohjaimen näytölle kirjoitettavilla viesteillä ja kysymyksillä sekä ohjelmoitavilla painikkeilla tehtävillä valinnoilla.



! viesti ja valinta käyttäjälle, riippuen mitä painiketta käyttäjä painaa saa toiminto
! arvon 1 – 5, joka tallentuu määritellylle muuttujanimelle, tässä nimenä ”nappula”,
! viesti ja painikkeiden päälle tulevat tekstit tulee olla lainausmerkkien sisällä
 TPreadFK nappula,"VALITSE !","Nelio","Kolmio","Ympyra"," ","Lopetus";
! testataan annetun valinnan mukainen toiminta, CASE 1 toimii kun painike 1
! on valittu, CASE 2 liittyy painikkeeseen 2 jne. DEFAULT toimii, jos valitaan
! tyhjä painike, jolle ei ole määritelty omaa CASEa

TEST nappula

CASE 1:

! kysymys käyttäjälle, vastaus tallentuu muuttujaksi “kerrat” tässä tapauksessa

TPreadNum kerrat,"KUINKA MONTA ?";

FOR i FROM 1 TO kerrat DO

nelio;

ENDFOR

CASE 2:

kolmio;

CASE 3:

! aliohjelmakutsu

ymp;

CASE 5:

koti;

DEFAULT:

TPWrite "Yrita uudelleen !";

WaitTime 5;

ENDTEST

ENDPROC

ENDMODULE

5. Ehtokäskyt

a) IF

```
IF osanum<4 THEN  
xmitta:=(osanum-1)*100;  
ymitta:=0;  
ELSE  
xmitta:=(osanum-4)*100;  
ymitta:=100;  
ENDIF
```

- muuttujat osanum, xmitta ja ymitta

- :=, antaa uuden arvon muuttujalle, arvo voi olla vakio tai aritmeettinen lauseke

b) FOR

- FOR:a käytetään kun tietty käsky on toistettava useita kertoja peräkkäin

```
FOR I FROM 1 TO 6 DO
```

```
osanum:=I;
```

```
xymitta;
```

```
vienti2;
```

```
ENDFOR
```

c) GOTO

- ehdoton hyppy haluttuun kohtaan

```
reg1:=1;
```

```
next:
```

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

```
IF reg1<=5 GOTO next;
```

- ohjelmasilmukka next suoritetaan viisi kertaa, reg1 on muuttuja

6. Muut käskyt

WaitTime 3;

- odottaa tietyn ajan, tässä kolme sekuntia. Sopii tilanteisiin missä halutaan varmistaa esim kappaleen kiinnittyminen tarraimen tai irtoaminen siitä

StopMove

- tällä käskyllä robotin liike pysähtyy. Sopii esimerkiksi työkierron päätteeksi tai keskeytusrutiiniin robotin väliaikaiseen pysäyttämiseen.

StartMove

- käynnistää liikkeen uudelleen
- esim. **StopMove**
 WaitDI ready_input,1 (kun tulo muuttuu 1:ksi niin liike käynnistyy)
 StartMove

VelSet 50, 800

- käskyä seuraavien paikoituskäskyjen nopeuksia vähennetään 50% käskyyn ohjelmoidusta arvosta. TKP:n nopeus enintään 800 mm/s

CRobT-funktio (robotin paikan luku).

CRobT (Current Robot Target) funktiota käytetään robotin senhetkisen sijainnin lukemiseen

Esimerkki

```
pAlku:=CRobT();
MoveL offs(palku,0,100,0)...;
RETURN;
```

- tallentaa ensin pAlku -paikkatietodataan robotin sijaintiargumentit, ja liikkuu sitten pAlku:n suhteen.

Ajan mittaaminen

ClkStart Käynnistää ajanottoon käytetyn kellon

ClkStop Pysäyttää ajanottoon käytetyn kellon

ClkReset Nollaa ajanottoon käytetyn kellon

Funtiot: ClkRead Laskee ajanottoon käytetyn kellon

Data: Clock Ajan mittaamiseen käytetty Data

```
ClkReset clock1;
ClkStart clock1;
MoveJ p1, v1000, fine, tool0;
MoveJ p2, v1200, fine, tool0;
ClkStop clock1;
aika:=ClkRead(clock1)
TPWrite "aikaa kului:"\Num:=aika;
```

7. I/O- käskyt

Set press2;

- asettaa signaalin press2 arvon ykköseksi

Reset press2;

- nolaa signaalin press2:n

8. Työkohdekoordinaatisto Wobj

Normaalisti robottiohjelma voidaan tehdä käyttäen robotin omaa peruskoordinaatistoa, *World*-koordinaatistoa. Liikekäskyissä ei silloin näy mitään erillistä koordinaatistomäärittelyä.

MoveL alku,v300,z30,imukuppi;

Erillinen itse määritelty *WorkObject*-koordinaatisto voidaan luoda mielivaltaiseen paikkaan ja asentoon robotin työalueelle. Valittaessa kyseinen koordinaatisto aktiiviseksi Jogging-ikkunassa robottia voidaan liikutella käsiohjaimella tämän koordinaatiston mukaisesti ja opetettaessa paikoituksia liikekäskyjä varten tulee koordinaatistomäärittely käskyn määrittelyyn mukaan:

MoveJ alku,v500,fine,imukuppi\WObj:=jigi1;