ABB robotaren programazioa

ELHUYAR
edizioak

Jon Artola Mendiola

ABB robotaren programazioa

ELHUYAR
edizioak

Jon Artola Mendiola

Hezkuntza, Unibertsitate eta ikerketa Sailak onetsia: 2007-11-27

Azalaren diseinua: Miriam Arregi

Maketa: Miriam Arregi

Hizkuntza-zuzenketak: Elhuyar: Hizkuntza Zerbitzuak

© Edizio honena: Elhuyar fundazioa, Zelai Haundi, 3 Osinaldi industrialdea, 20170 Usurbil (Gipuzkoa)
E-mail:elhuyar@elhuyar.com. www.elhuyar.org

© Jon Artola Mendiola

2007ko EIMA III

Aurkibidea

1. ABB ROBOTA. EZAUGARRI NAGUSIAK

1.1. Sistemaren ezaugarri nagusiak 2

1.2. Sistema martxan jartzea 8

1.3.Leihoak erabiliz nola lan egin 9

1.4.Robota mugiaraztea aginte-palanka bidez 11

1.5. Programa-hautaketa 16

1.6.Programa martxan jartzea 20

1.7. Programa gelditzea 24

1.8. Modu automatikoa 24

1.9 Akatsak 25

1.10. Sistema gelditzea 26

1.11. Programa aldaketa 27

1.12. Programa disketean nola gorde 34

1.13. Sarrera/Irteerren seinaleak 39

2. OINARRIZKO INSTRUKZIOAK

2.1. Rapid progranazioaren-hizkuntzarako sarrera 46

2.2. Datu motak 53

2.3. Instrukzioak 65

3. BIBLIOGRAFIA 87

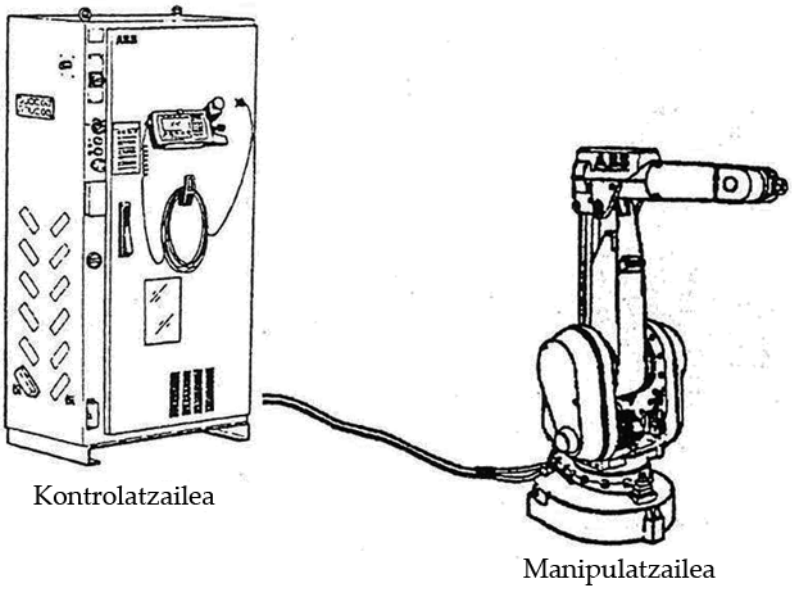
ABB robota
Ezaugarri nagusiak

1

1.1 SISTEMAREN EZAUGARRI NAGUSIAK

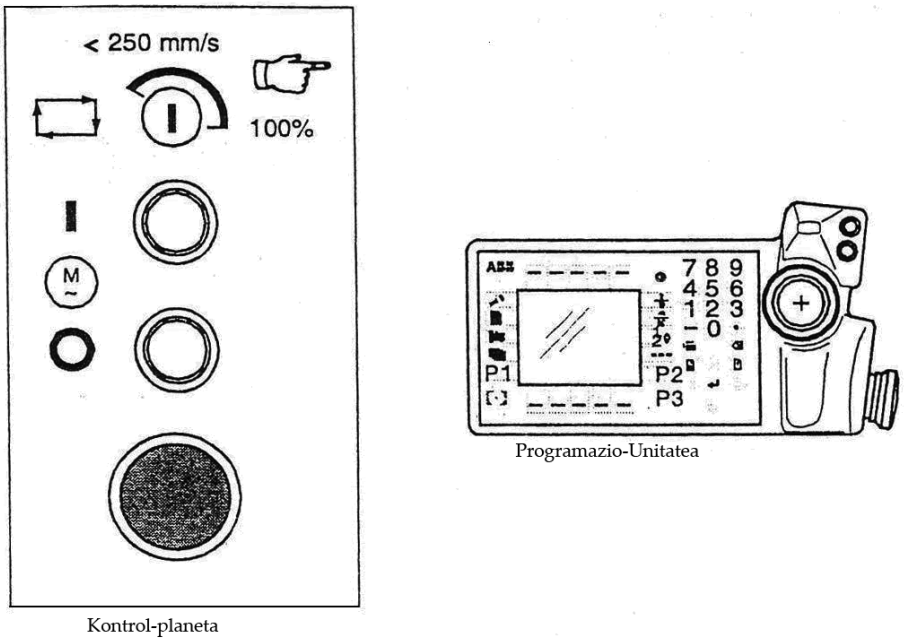
1.1.1 OROKORTASUNAK

Robot batek bi atal nagusi ditu: kontrolatzailea eta manipulatzailea.



1. irudia

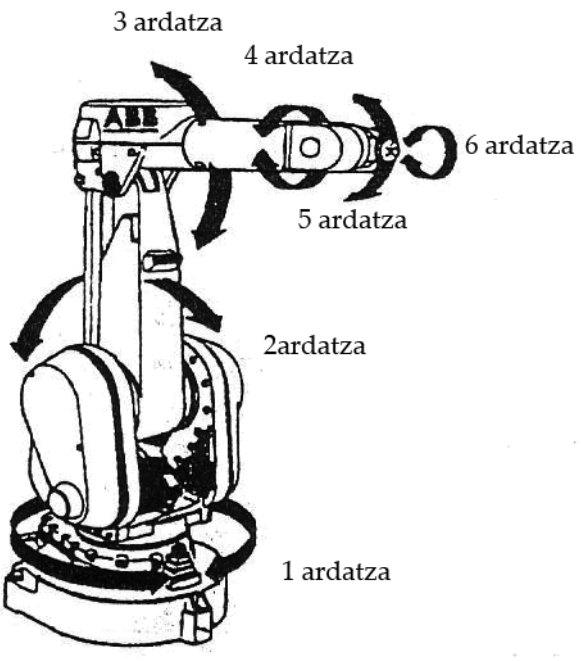
Erabiltzaileak robotarekin komunikatzeko bi era ditu: programazio-unitatearen bidez edota kontrolagailuan dagoen panelaren bidez.



2. irudia

1.1.2 MANIPULATZAILEA

Manipulatzailearen ardatzak zein noranzkotan mugi daitezkeen eta zer izen duten erakusten digu 3. irudiak.

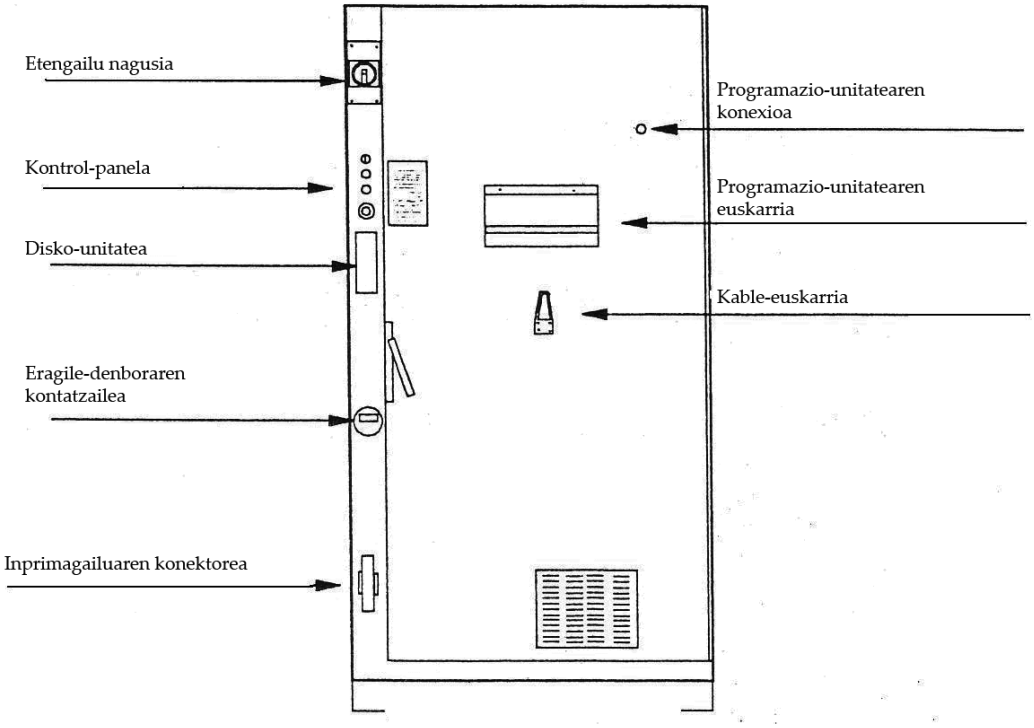


3. irudia

1.1.3 KONTROLATZAILEA

Hauek dira kontrolatzailearen atal nagusiak:

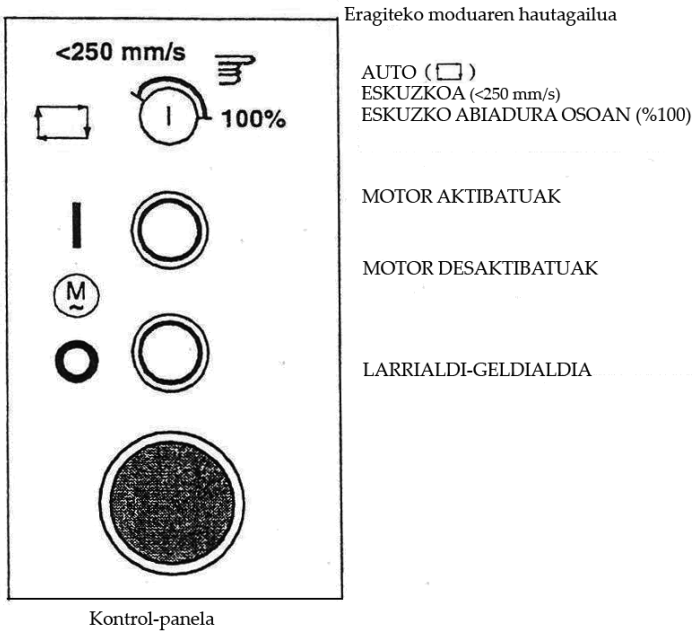
* etengailu nagusia
* kontrol-panela
* disko-unitatea
* eragite-denboraren kontatzailea
* inprimagailuaren konektorea
* programazio-unitatearen konexioa
* programazio-unitatearen euskarria
* kable-euskarria



4. irudia

1.1.4 KONTROL-PANELA

Kontrol-panelaren ikuspegi zehatza 5. irudian dugu. Erabiltzaileak aurrerago topatuko du pultsadore bakoitzaren azalpen laburra.



5. irudia

AUTO eran lan egitea (Produkzio era)

Produkziorako aurrez prestaturiko programekin erabiltzen da. Hala, erabiltzaileak aginte-palanka erabili behar du manipulatzailea mugitzeko. *Motor desaktibatuak* dioen pultsadore-adierazlearen argia etengabe dago piztuta.

ESKUZKO ABIADURA MURRIZTUA eran lan egitea (Programazio era)

Robotaren lan-esparruan ibili behar denean eta robota programatzeko garaian erabiltzen da.

ESKUZKO ABIADURA OSOA eran lan egitea (Baieztatze era)

Robotaren programa abiadura osoz exekutatu eta probak egiteko erabiltzen da.

Motor aktibatuak

Era honetan, robotaren motorrak aktibatuak daude, eta pultsadore-adierazlearen argia etengabe dago piztuta.

Motor desaktibatuak

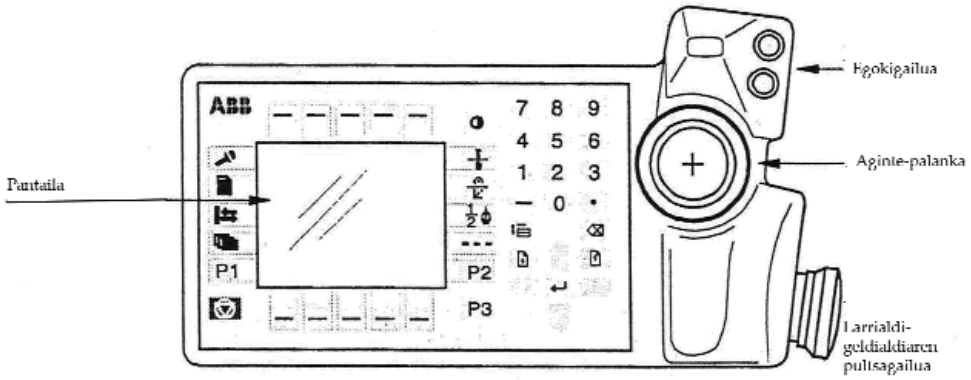
Era honetan, robotaren motorrak desaktibatuak daude, eta balaztak lanean ari dira. Sistema eskuzko eran dagoenean, pultsadoreak argi ez-jarraikia izango du.

Larrialdi-geldialdia

Larrialdi-pultsadorea sakatu orduko, robota gelditu egingo da. Ez du kontuan izango zer eratan ari zen lanean. Pultsadoreak sakaturik jarraituko du, eta *motor aktibatuak* erara itzultzeko, jatorrizko egoerara itzuli beharko du pultsadoreak.

1.1.5. PROGRAMAZIO-UNITATEA

Programazio-unitatearen ikuspegi orokorra 6. eta 7. irudietan dugu.



6. irudia

Larrialdi-geldialdia

Larrialdi-pultsadorea sakatzean, robota berehala gelditzen da, lanean zer egoeratan eta eratan ari zen kontuan izan gabe. Pultsadoreak sakaturik jarraituko du, eta motor aktibatuak erara itzultzeko, jatorrizko egoerara itzuli beharko du.

Egokigailua (eragiketa-segurtasuna bermatzeko)

Erabiltzaileak programazio-unitatean topatuko du palanka hori. Tarteko posizioan sakaturik dagoenean, *motor aktibatuak* pultsadorea aktibatu egingo da (baldin eta eskuzko lanerako bi eratakoren batean ari bagara). Egokigailua guztiz sakatuta edota sakatu gabe dagoenean, robota desaktibatu egingo da (motor desaktibatuak), eta motorrak gelditu egingo dira.

Egokigailua askatu eta segundo-erdiko tartean berriz sakatuz gero, robota ez da motor aktibatuak egoerara itzuliko. Hori gertatuz gero, egokigailua askatu eta berriz erdiko posizioan sakatu beharko da.

Robota mugitu nahi denean soilik aktibatuko da egokigailua, nahiz eskuz (palanka erabiliz), nahiz programa exekutatuz.

Aginte-palanka

Robota eskuz mugitzeko balio du; robota programatzeko garaian, adibidez.

Pantaila

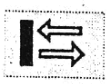
Programatzeko garaian, mota guztietako informazioa ikusteko, aldaketak egiteko eta abarrerako balio du. Pantailak 16 lerrotarako lekua du, eta lerro bakoitza 40 karakterekoa da.

Programazio-unitateko teklen izenak 7. irudian ikus ditzakegu.

- Leiho-teklak: pantailan erabili nahi den lan-leihoa aukeratzeko.

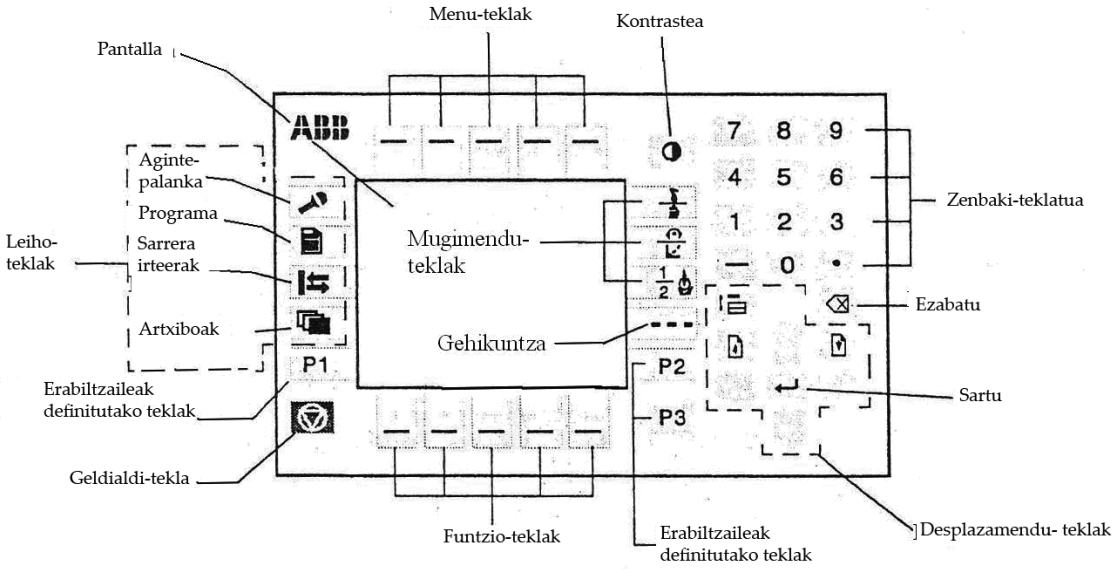
Aginte-palanka: robota mugitzeko.

Programa: programak egin eta testak egiteko.



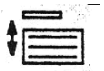
Sarrera/Irteerak: robotari konektaturiko sarrera eta irteerak funtzionarazteko.

Hainbat leiho: sistema-parametroak, zerbitzuak, produkzioa, artxibo-administratzailea, eta horien moduko zenbait leiho.



7. irudia

- Desplazamendu-teklak: pantaila barneko leihoetan, kurtsorea mugitzeko.



Zerrenda: leihoaren alde batetik bestera kurtsorea eramateko (marra bikoitzez bereiziak egon ohi dira).

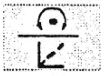
Aurreko/hurrengo orria: orri batetik bestera pasatzeko.

Desplazamendu bertikalerako geziak: kurtsorea gora edo behera mugitzeko.

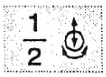
Desplazamendu horizontalerako geziak: kurtsorea ezker-eskuinmugitzeko.

- Mugimendu-teklak: eskuzko funtzionamenduan eta aginte-palanka erabiltzean, robota zein beste gailu periferikoren bat nola mugitu aukeratzeko.

Mugimendu-unitatea: robota zein beste unitate mekanikoak mugitzeko.

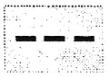


Mugimendu mota: robota nola mugitu aukeratzeko, hots, berrorientatuta edota mugimendu linealean.



Mugimendu mota: ardatzez ardatz erako mugimendua aukeratzeko.

* 1 = 1-3 ardatzak
* 2 = 4-6 ardatzak.



Gehikuntza: gehikuntza bidezko mugimendua aktibatzeko ala desaktibatzeko.

- Beste tekla batzuk

Geldialdi-tekla: programa gelditzeko.

Kontrastea: pantailaren kontrastea doitzeko.

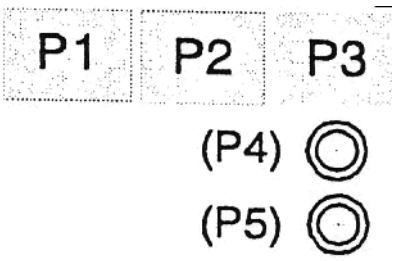
Menu-teklak: komandoek dituzten menuak ikusteko.

Funtzio-teklak: komandoak zuzenean aukeratzeko.

Ezabatu:pantailan aukeraturiko datuak ezabatzeko.

Sartu: datuak sisteman sartzeko.

-Tekla programagarriak



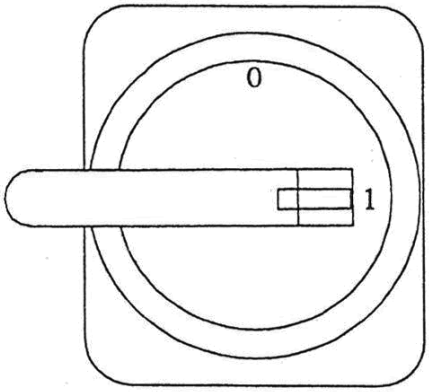
Erabiltzaileak definitu behar ditu horien funtzioak.

1.2. SISTEMA MARTXAN JARTZEA

Atal honetan, erabiltzaileak sistema konektatzeari ekingo dio; hau da, programatzeko, programak exekutatzeko eta abarrerako prest izango da.

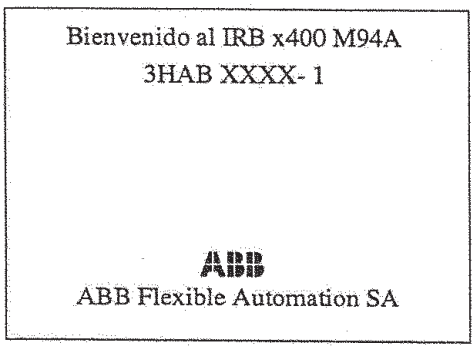
Sistema konektatu aurretik, ziurtatu robotaren lan-eremuan ez dagoela inor.

Itxi etengailu nagusia, 8. irudian ageri den moduan. Robotak sistemaren egiaztapen automatikoa egingo du.



8. irudia

Sistema egiaztatu eta akatsik aurkitu ezean, 9. irudiko ongietorri-mezua ageriko da pantailan.



9. irudia

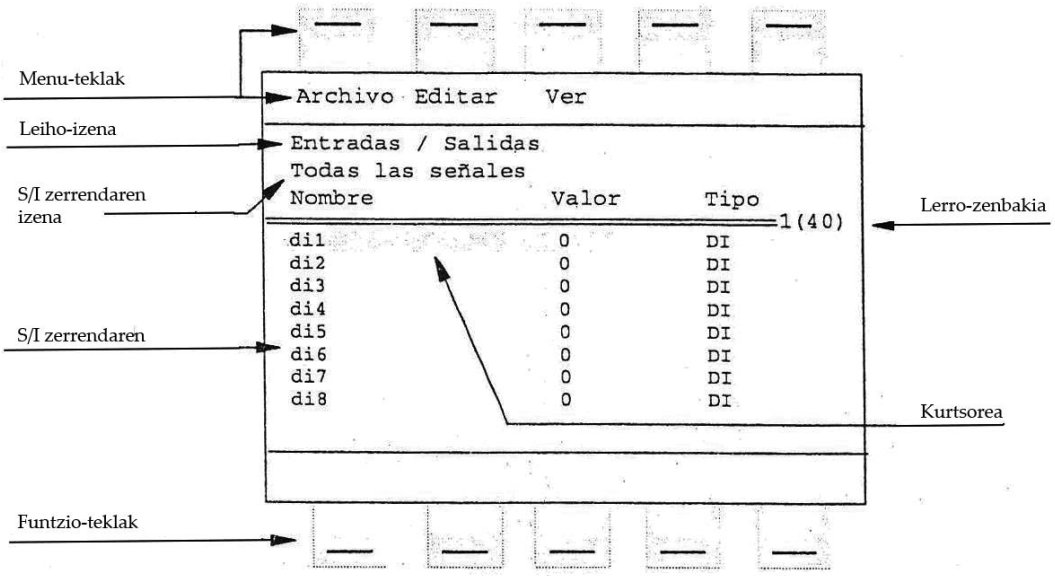
1.3. LEIHOAK ERABILIZ NOLA LAN EGIN

Atal honetan, leihoekin lan egiteko behar diren oinarriak topatuko dira. Ondoren, sarrera/irteeren eskuzko erabilera-leihoa ikusiko dugu.

1.- Sakatu Sarrera/Irteera izeneko tekla (S/I) (ikusi 10. irudia).

10. irudia

2.- Eskuzko S/I-ren leihoa agertuko da pantailan, 11. irudian ikusten den moduan. S/I-ren zerrendaren itxura aldatuko da, sistema-seinaleen definizioaren eta sistemak dituen S/I txartel kopuruaren arabera.



11. irudia

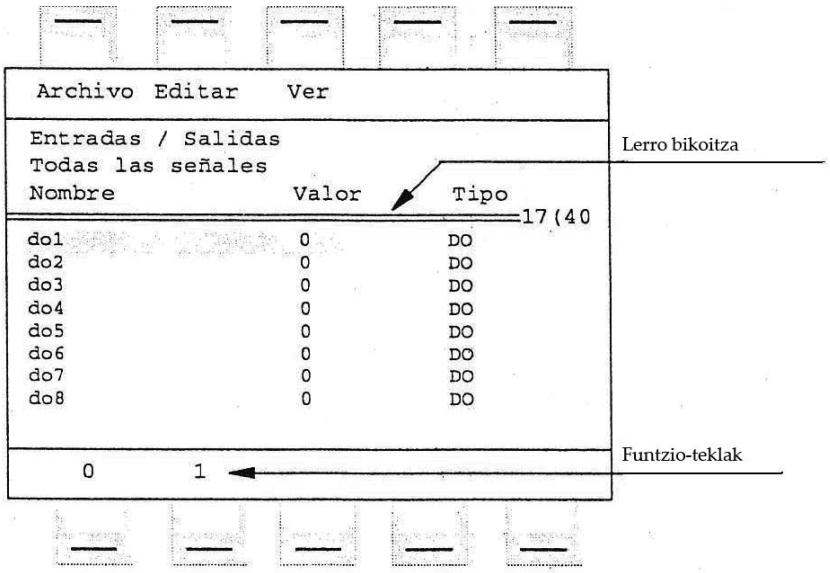
Irteera digitala aukeratuz gero, haren egoera alda daiteke funtzio-teklen bidez. Adibidez, *Test?* sakatuz gero, pantailan agertuko den beste leiho bati dei egiten zaio.

3.- Zerrendako ikurren bat aukeratuz gero (kurtsorea mugituz), era hauetako mugimenduak lortuko dira:

Ikurra	**Mugimendua**
Goranzko gezia	Lerro bat gora
Beheranzko gezia	Lerro bat behera
Menuko Editar aukeran, Ir al inicio	Zerrendako lehen lerrora
Menuko Editar aukeran, Ir al final	Zerrendako azken lerrora
Hurrengo orria	Hurrengo orrira
Aurreko orria	Aurreko orrira
Editar aukeran, idatzi lerro-zenbakia eta sakatu OK	Zerrendako lerro jakin batera

4.- Zenbait kasutan, leihoa marra lodi batez bitan zatitua ager daiteke (ikusi 12. irudia).

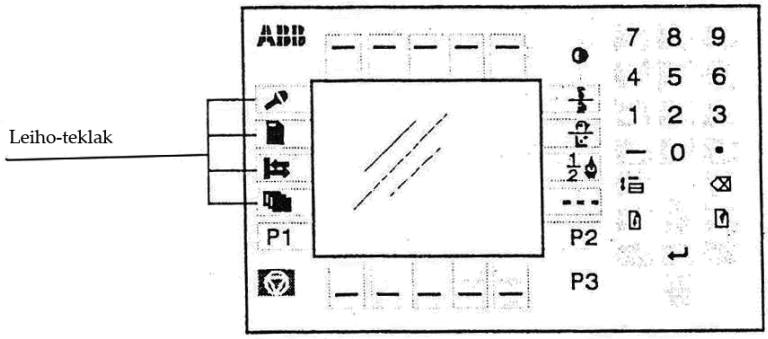
5.- Sakatu hurrengo orrira joateko tekla, pantailako lehen lerroan irteera bat agertu arte. Bi funtzio-tekla erabilgarri agertuko dira.



12. irudia

Zenbait leihotan, erabiltzaileak kurtsorea mugitu ahal izango du. Horietan, mugitu kurtsorea zerrenda-teklaren bidez.

6.- Programazio-unitateak lau leiho ditu, 13. irudian ikus daitekeenez.



13. irudia

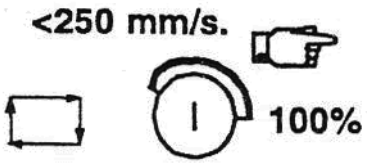
Leiho-teklaren bat sakatzean, ikusgai zegoena leiho berri baten atzean ezkutatuko da. Erabiltzaileak leiho bat aukeratzen duen bakoitzean, azken aldian utzita bezalaxe agertuko zaio.

1.4. AGINTE-PALANKA BIDEZ ROBOTA MUGIARAZTEA

Programazio-unitateko aginte-palankaren bidez, robota mugitu ahal izango da. Atal honetan, robota zuzenki (lerro zuzenean) eta urratsez urrats nola mugitu ikusiko da. Hala, robota guk nahi dugun lekuan zehaztasunez koka dezakegu. Gehikuntza bidezko mugimendu izenez ere ezagutzen da.

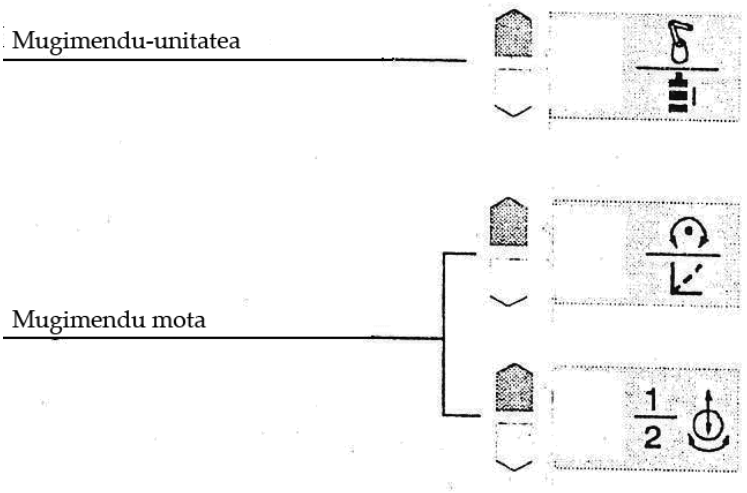
1.4.1. MUGIMENDU ZUZENA

Ondorengo azpiatal honen azalpena ariketa praktikoan oinarrituz egingo da.
1.- Ziurtatu eragiteko moduaren hautagailua <250 mm/s posizioan dagoela (ikusi 14. irudia).



14. irudia

2.- Ziurtatu robotaren mugimendu-unitatea eta mugimendu zuzeneko aukera hautatu direla.



15. irudia

Mugimendu-unitatearen tekla sakatzen bada, robota edota kontrolagailuari erantsitako beste edozein unitate funtzionaraztea hauta daiteke aginte-palankaren bidez. Hautatu robota mugimendu-unitatetzat, ariketa honetarako.

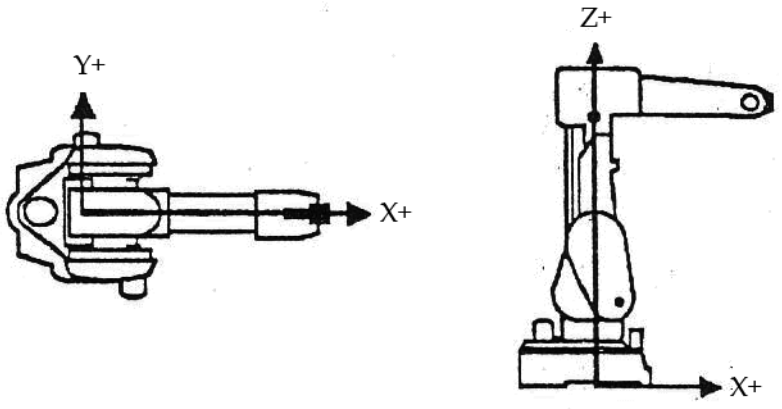
Mugimendu motaren teklaren eta aginte-palankaren bidez, robota eskuzko eran nola mugitu nahi den hauta daiteke.

Erabiltzaileak aukera hauek izango ditu:

- mugimendu zuzena
- amaierako gailu jakin bat berrorientatzea
- ardatzez ardatz erako mugimendua

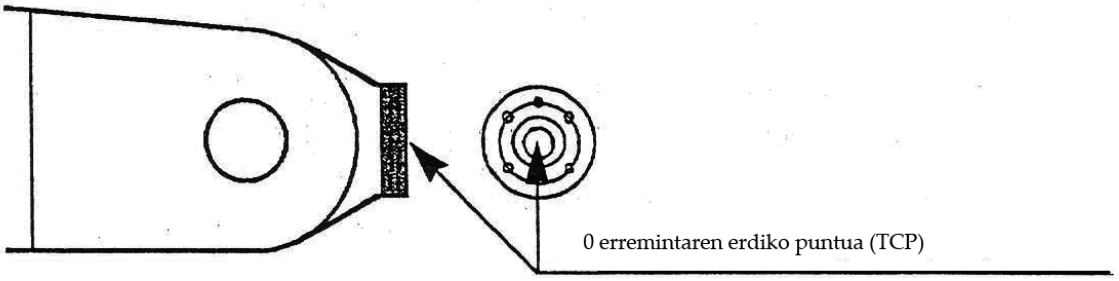
Aukeratu mugimendu zuzena, ariketa honetarako.

Esanda bezala mugimendu zuzena aukeratu bada, robota 16. irudiko azalpenen arabera mugituko da.



16. irudia

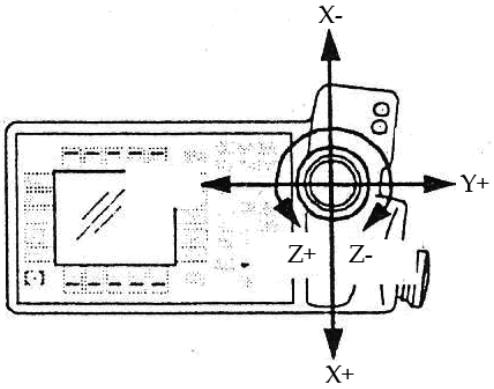
Aurreko irudiko koordenatu-sistemako ardatzen arabera, zuzenki mugitzen den puntuari erremintaren erdiko puntu (TCP: Tool Center Point) deritzo. Hori robotaren besoaren goiko muturrean dugu, robotaren aurreko planoaren erdi-erdian.



17. irudia

3.- Sakatu egokigailua motor aktibatuak posiziora iritsi arte.

4.- Orain robota mugi daiteke aginte-palanka erabiliz. Robotaren aurrean jarriz gero, aginte-palanka mugitzen den heinean, TCP 0 zuzenki mugituko da (X, Y, Z) ardatzetan.



18. irudia

Saiatu robota mugitzen aurreko irudiko (X, Y, Z) ardatzen norabideetan. Bestalde, aginte-palankaren mugimenduak konbinatuta, robota norabide anitzetan mugi daiteke. Ikusi robotaren abiadura aginte-palankari eragiteko eraren menpe dagoela. Zenbat eta gehiago eragin palankari, orduan eta azkarrago mugituko da robota.

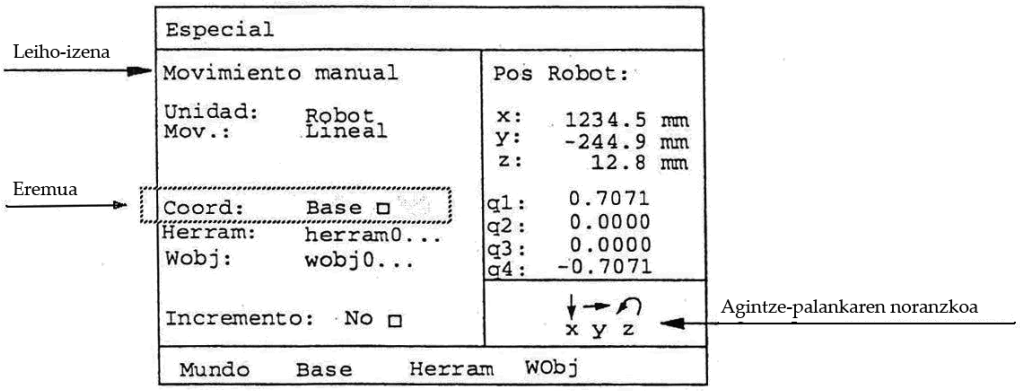
1.4.2. KOKAPEN ZEHATZA

Azpiatal honen azalpena ariketa praktikoan oinarrituz egingo da.

1.- Sakatu mugimenduaren leiho-tekla.

19. irudia

Leiho-tekla sakatuta, 20. irudian ageri den leihoaren antzekoa agertuko da.



20. irudia

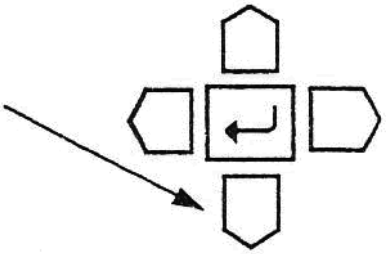
Leihoaren itxura aukeratutako leihoaren arabera aldatuko da, hau da, lortu nahi denaren arabera.

Menu aukerako teklek komando bat baino gehiago zuzentzen dituzte. Komando erabilgarrien zerrenda menuko edozein tekla sakatuz agertuko den goitik beherako menuan ikus daiteke.

Leihoan marra etenez mugaturiko azalerari eremu deritzo. Kolore griseko azalerak eremu-sarrera izena du, eta funtzio-teklen bidez funtzio berri bat hautatuz alda daiteke. Zenbait kasutan, programazio-unitateko mugimendu-teklak erabiliz ere egin daiteke aldaketa hori.

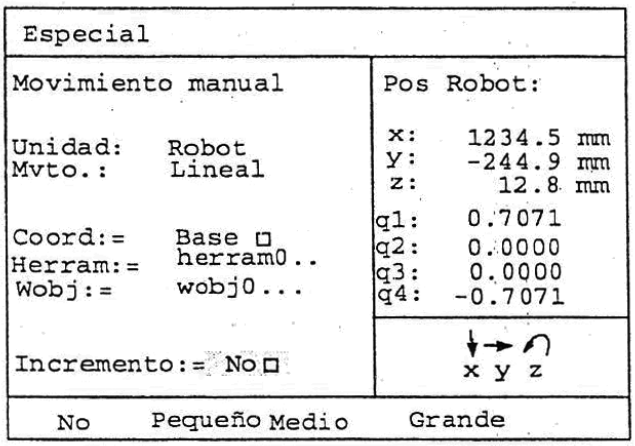
20. irudian hautaturiko eremu-sarrerak “?” ikurra du jarraian, eta horrek esan nahi du aukera *funtzio-teklaren* baten bidez egin dela.

2.- Mugitu kurtsorea kolore griseko eremutik incremento dioen eremura, beheranzko desplazamendu-gezia erabiliz.



21. irudia

3.- Kurtsorea gehikuntzaren eremura mugituz gero, gehikuntza bidezko mugimendua aukeratu ahal izango da, funtzio-teklaren bat sakatuta.



22. irudia

Pequeño, *Medio* edo *Grande* tekletakoren bat sakatzean, automatikoki “No” agertuko da gehikuntzaren eremuan.

Hala, aginte-palanka mugitutako aldi bakoitzean, robotak pauso bat emango du. Pausoaren tamaina erabiltzaileak hautatutako aukeraren (Pequeño, Medio edo Grande) araberakoa izango da.

tekla erabiliz, gehikuntza aktibatu edo desaktibatzen da.

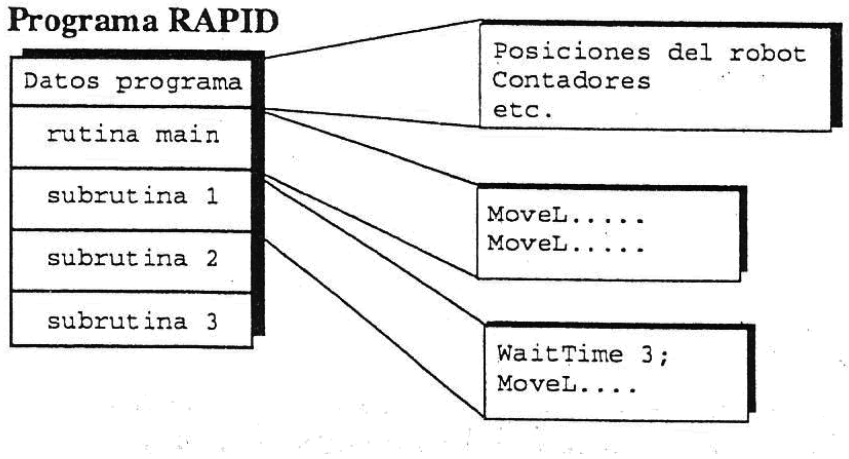
Saiatu robota mugitzen aginte-palanka erabiliz, eta ikusi nola mugitzen den.

Eskuzko erabilerari, koordenatu-sistemei eta abarri buruzko informazio gehiago nahi izanez gero, *Erabiltzaile-gidan*, begiratu mugimenduari buruzko atala.

Ariketa honekin jarraitu nahi ez bada, sistema deskonektatu beharko da, 1.10. ataleko (*Sistema gelditzea*) azalpenen arabera.

1.5. PROGRAMA-HAUTAKETA

Atal honetan, programa nola ireki azalduko da. Programak hiru atal izan ohi ditu: errutina (beti agerian), azpiprograma kopuru bat eta programa-datuak. Programa bakoitzeko errutina bakarra onartzen da.



23. irudia

Programa berri bat irekitzean, programa horrek robotaren memoria betetzen zuen programa ordezkatuko du. Orduan, errutina nagusia agertuko da pantailan, eta errutina nagusi horren lehen agindua gelditukо da hautaturik.

1.5.1. NOLA ERABILI ARIKETA-PROGRAMA

Ariketa-programa IRB2 sistemako disketean, “DEMO” direktorioan eta “EXERCISE” izenarekin jasoa dago.

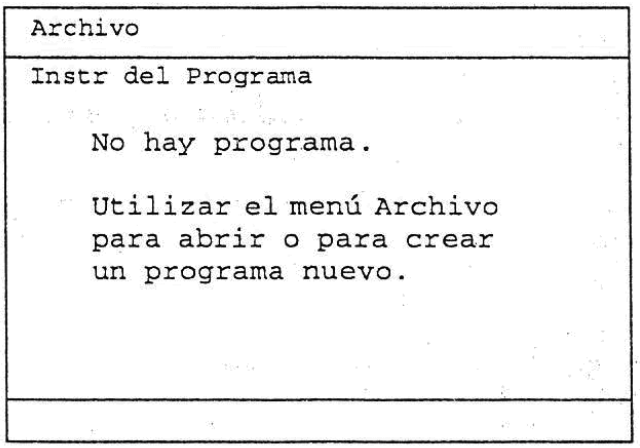
Erabiltzen hasteko:

1.- Kokatu eragiteko moduaren hautagailua <250 mm/s posizioan.

2.- Sakatu programaren leiho-tekla.

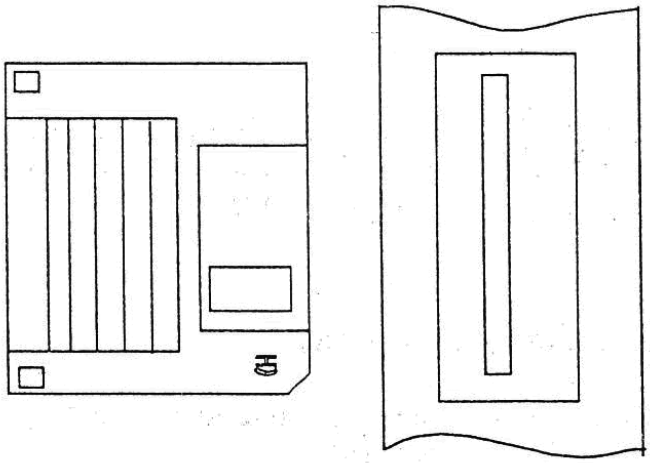
Robotaren memorian programarik ez badago, 25. irudiko mezua agertuko da pantailan; hala ez bada, robotaren memorian jasotako programa ikusiko da.

24. irudia



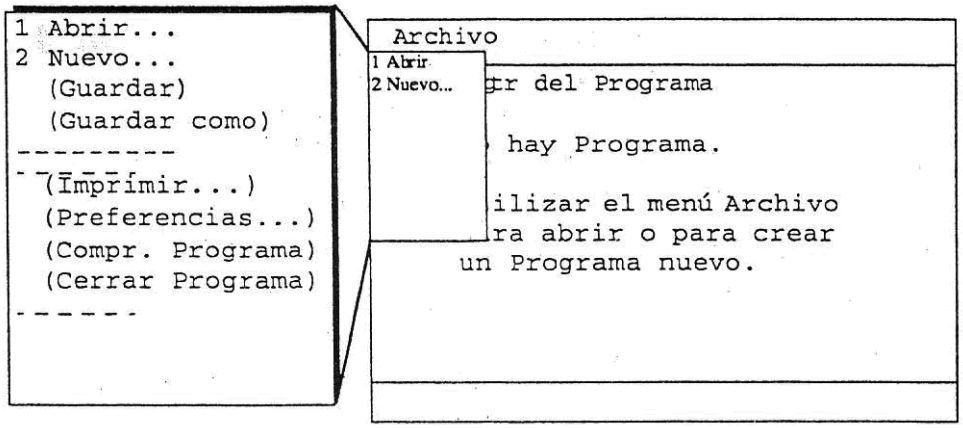
25. irudia

3.- Sartu instalazio-disketea (Set up) armairuak aurrealdean duen unitatean, 26. irudian ageri den moduan.



26. irudia

4.- Sakatu Archivo menua, 25. irudian azaldu bezala, eta 27. irudiko leihoa agertuko da.



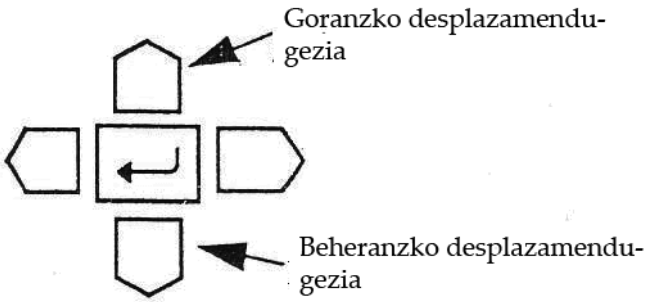
27. irudia

Pantailan goitik beherako menu bat azalduko da orain. Archivo menuko komando erabilgarri guztiak zerrendaturik agertuko dira (parentesi artean ageri diren komandoak hautatu ezinak dira). Menuko gainerako teklek berdin funtzionatzen dute.

Hemendik aurrera, *Archivo: Abrir, Archivo: Guardar,* eta abar idatziko dugu. Bi puntuen ezkerreko hitza menuaren izena da, eta eskuineko hitza, berriz, erabiltzaileak hautatu beharreko komandoari dagokio.

Menu-tekla sakatutakoan, goitik beherako menuko lehen funtzioa hautaturik agertuko da beti. Programazio-unitateko desplazamendu-geziak erabiliz, kurtsorea mugitu ahal izango da menu barnean (ikusi 28. irudia). Erabili nahi den komandoa hautatuta, sakatu Sartu tekla.

Zenbaki-teklatua erabiliz ere badago komandoak hautatzea; horretarako, komandoaren aurretik dagoen zenbakia sartu behar da.

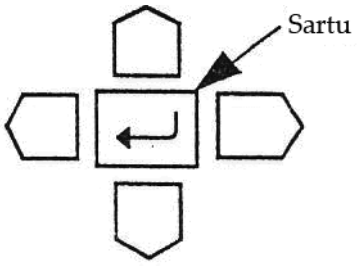


28. irudia

Komando-izenaren ondoren datozen hiru puntuek "…" adierazten dute komando hori hautatzean elkarrizketa-leiho bat agertuko dela.

Goitik beherako menua desagerrarazteko, sakatu menua irekitzeko erabili den tekla bera.

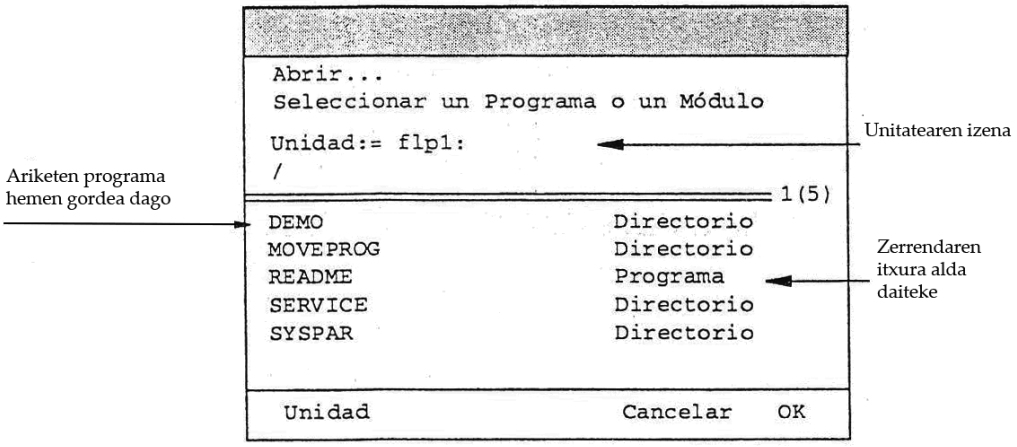
5.- Abrir… komandoa hautatuta, sakatu Sartu botoia (ikusi 29. irudia), eta komandoak bere funtzioa exekutatuko du. Komandoaren ondorengo hiru puntuek esan nahi dute agindua ez dela zuzenean beteko, informazio gehiago behar baita. Kasu horretan, ireki nahi den programa hautatu behar da.



29. irudia

Unidad funtzio-teklarekin, aldatu unitate batetik bestera, adibidez, robotaren barne-memoriatik (RAM) diskete-memoriara (flp1:) edota beste edozein biltegi-gailutara.

6.- Sakatu Unidad tekla. Ziurtatu flp1: adierazpena (1 disko malgua) Unidad:= adierazpenaren atzetik agertzen dela. Aipaturiko disketearen edukia erakusten digun elkarrizketa-leihoa azalduko da, 30. irudikoa.



30. irudia

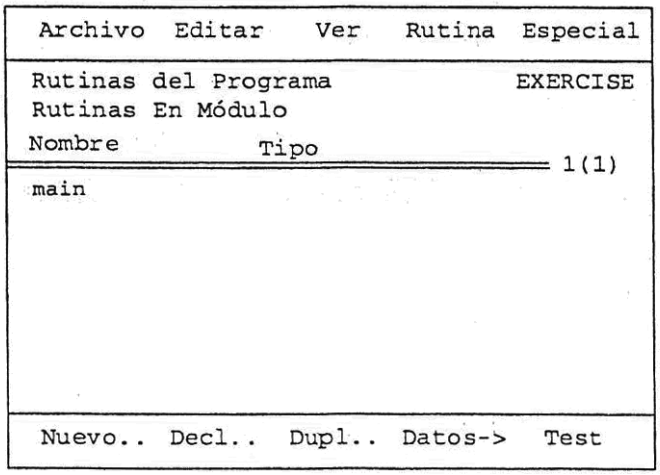
Elkarrizketa-leihoren bat (menurik ez duen leihoa) Cancelar teklaz itxitakoan, hautaturiko funtzioa ez da exekutatuko. OK tekla sakatuz gero, hautaturiko funtzioa exekutatu eta elkarrizketa-leihoa itxiko da.

7.- Hautatu *DEMO* funtzioa. Mugitu kurtsorea goranzko eta beheranzko desplazamendu-gezien bidez.

8.- Sakatu Sartu tekla.

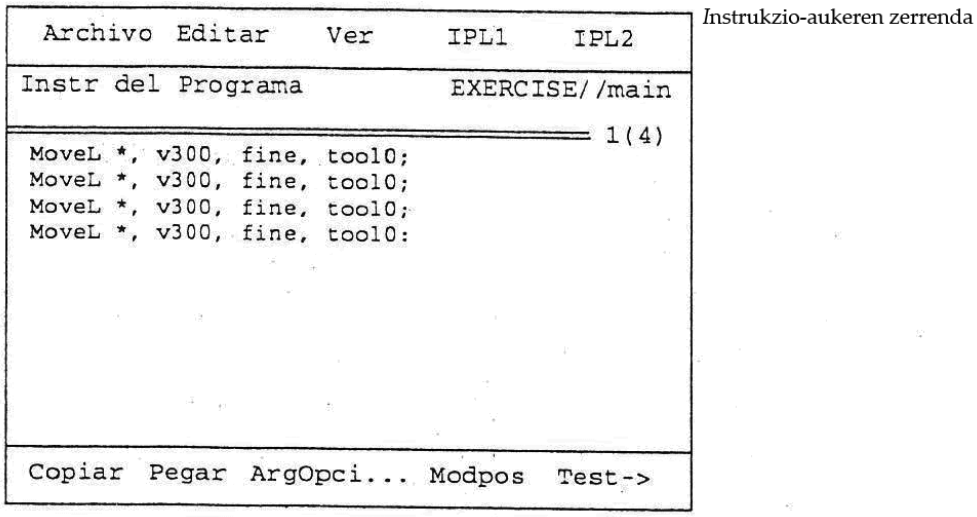
9.- Hautatu EXERCISE.

10.- Sakatu OK tekla, eta 31. irudiko leihoa agertuko da.



31. irudia

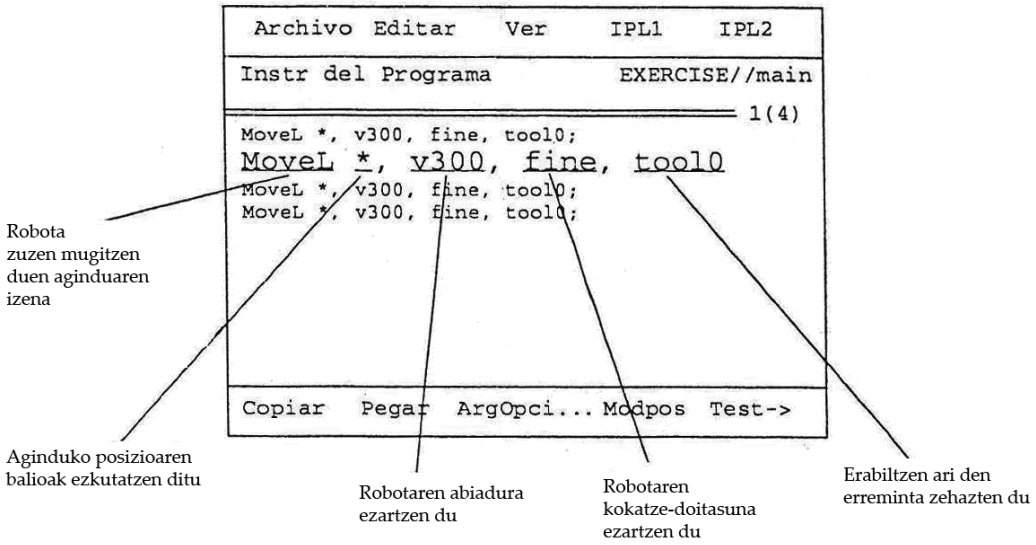
11.- Ondoren, sakatu Sartu tekla; 32. irudiko leihoa agertuko da.



32. irudia

Hau da ariketa-programen errutina nagusia. Lau mugimendu-aginduk osatzen dute (*move*).

Errutinak hainbat agindu motaz osaturik daude: mugimendu-aginduak, itxarote-aginduak, etab. Agindu bakoitzak bere arrazoiak ditu jarraian. Arrazoiak motaren arabera alda edo deusez daitezke. Agindu baten adibidea 33. irudian ikus dezakegu.

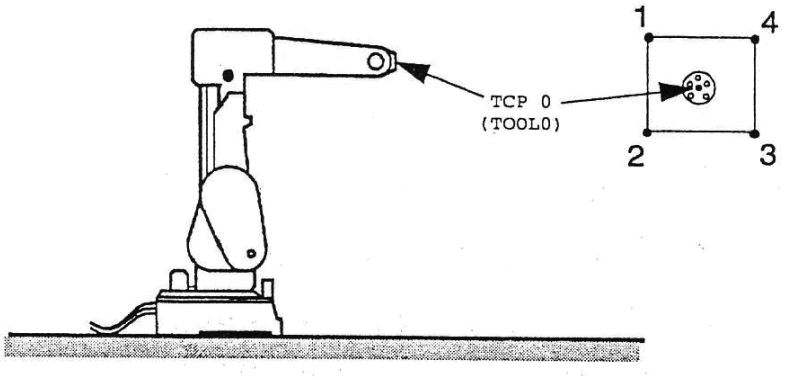


33. irudia

1.6. PROGRAMA MARTXAN JARTZEA

Erabiltzaileak aurreko atalean irekitako programa martxan jarri behar du orain. Lehenbizi, abiadura murriztuarekin, programa pausoz pauso exekutatu behar da. Gero, exekutatuko da era jarraituan.

Programak lau mugimendu-agindu ditu, eta robotaren *kalibraketa-puntutik* gertu dauden posizioak ditu barneraturik (ikusi 34. irudia).



34. irudia

Programa martxan jarri aurretik, aginte-palankaren bidez, 5. ardatza 45º-tik behera eskuz mugitu behar da (ikusi 3.2. atala).

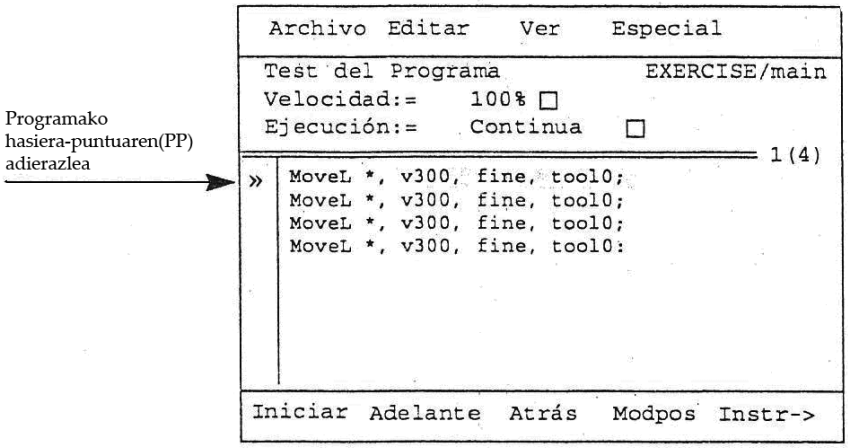
Hona hemen jarraitu beharreko aginduak:

1.- Sakatu Programaren leiho-tekla eta 35. irudiko pantailaren antzekoa agertuko da.

```
Archivo  Editar   Ver    IPL1    IPL2
Instr. del Programa      EXERCISE//main
                                   1(4)
MoveL *, v300, fine, tool0;
MoveL *, v300, fine, tool0;
MoveL *, v300, fine, tool0;
MoveL *, v300, fine, tool0:

Copiar  Pegar  ArgOpci... Modpos  Test->
```

35. irudia

2.- Sakatu Test funtzio-tekla, eta 36. irudiko leihoa agertuko da.



36. irudia

Funtzio-tekla hauek daude ikusgai:

- Iniciar: programa modu jarraituan exekutatzeko.

- Adelante: agindu bat aurrera egiteko.

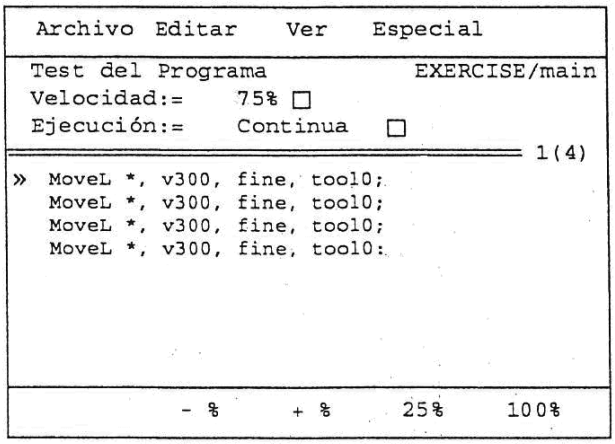
- Atrás: agindu bat atzera egiteko.

- Instr→: programaren agindu-leihoa hautatzeko.

Programaren hasiera-puntuak (PP) adierazten digu Iniciar, Adelante edo Atrás tekletakoren bat sakatzean programa zein aginduarekin hasiko den.

3.- Hautatu leihoaren goiko zatia, zerrenda-tekla sakatuta.

4.- Murriztu abiadura % 75era, -% funtzio-tekla erabilita. Abiadura % 5eko gehikuntzak eginez aldatzen da.



37. irudia

5.- Zerrenda-teklaren bidez, eraman kurtsorea berriz programaren lehen lerrora.

```
Archivo  Editar    Ver    Especial
Test del Programa            EXERCISE/main
Velocidad:=    75% □
Ejecución:=    Continua    □
                                    1(4)
»  MoveL *, v300, fine, tool0;
   MoveL *, v300, fine, tool0;
   MoveL *, v300, fine, tool0;
   MoveL *, v300, fine, tool0:

Iniciar  Adelante  Atrás  Modpos  Instr->
```

38. irudia

Programa abiatzeko prest dago. Ziurtatu orain robotaren inguruko babes-eremuaren barnean inor ez dagoela.

6.- Egokigailua eta Adelante funtzio-tekla sakatuta, ipini programa martxan.

Behin programa martxan jarrita, robotak agindu bat beteko du eta gelditu egingo da. Sakatu berriz Adelante hurrengo agindua martxan jartzeko, eta egin gauza bera gainerako guztiekin. Programa exekutatzen den bitartean, 39. irudiko leihoa agertuko da.

```
Ejecución del Test
Ejecutar prog.               EXERCISE//main
Velocidad:=    75% □
Ejecución:=    Continua    □
Registro eventos
                                    1(1)
Ejecutando
```

39. irudia

7.- Programa osatzen duten agindu guztiak banan-banan bete behar dira. Horretarako, sakatu Adelante tekla behin eta berriz, robota posizio berrira iristen den aldi oro.

8.- Azken agindura iritsitakoan, Adelante tekla sakatuz gero, programa abiapuntutik hasiko da berriz.

9.- Eraman robota 4. posizioraino (ikusi 34. irudia).

10.- Mugitu kurtsorea, lehen egindako eran, Ejecución eremura, eta aldatu Ciclo moduko exekuziora.

11.- Kokatu berriro kurtsorea programan.

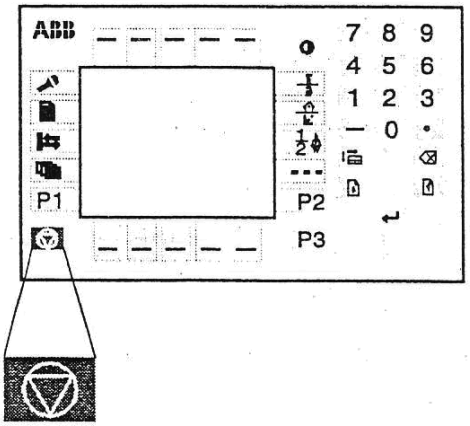
12.- Sakatu Adelante tekla, eta ipini programa martxan.

Ciclo modua hautatzean, exekutatu programa behin, eta, ondoren, 4. posizioan geldituko da.

13.- Hautatu berriz Continua exekuzio-modua.

1.7. PROGRAMA GELDITZEA

Programazio-unitateko geldialdi-tekla sakatuta, programa gelditu egingo da.

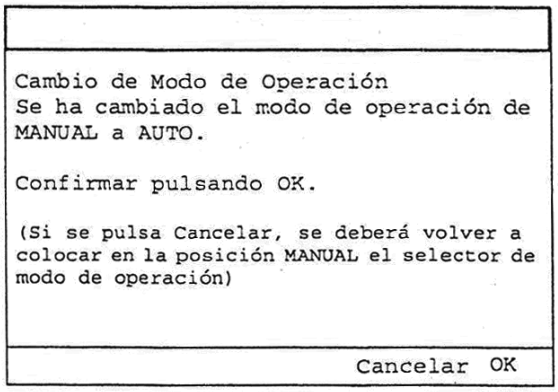


40. irudia

1.8. MODU AUTOMATIKOA

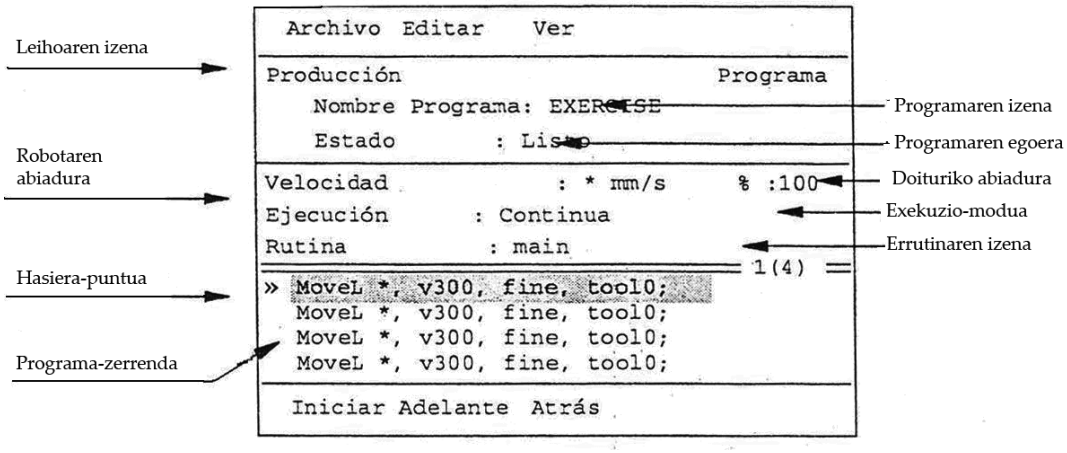
Modu automatikoa erabiltzeko prestatuta dauden programentzat da. Hona hemen modu automatikoa gauzatzeko bete beharreko aginduak:

1.- Kokatu Auto posizioan kontrol-paneleko hautagailua, eta 41. irudiko pantaila agertuko da.



41. irudia

2.- Sakatu OK tekla. Hala, era automatikoan eragiteko modura aldatu da, eta, pantailan, *Producción* (ekoizpena) leihoa ageri da (ikusi 42. irudia).



42. irudia

3.- Sakatu kontrol-paneleko *Motor aktibatuak* pultsadorea.

4.- Ipini programa martxan, *Iniciar* funtzio-tekla sakatuta.

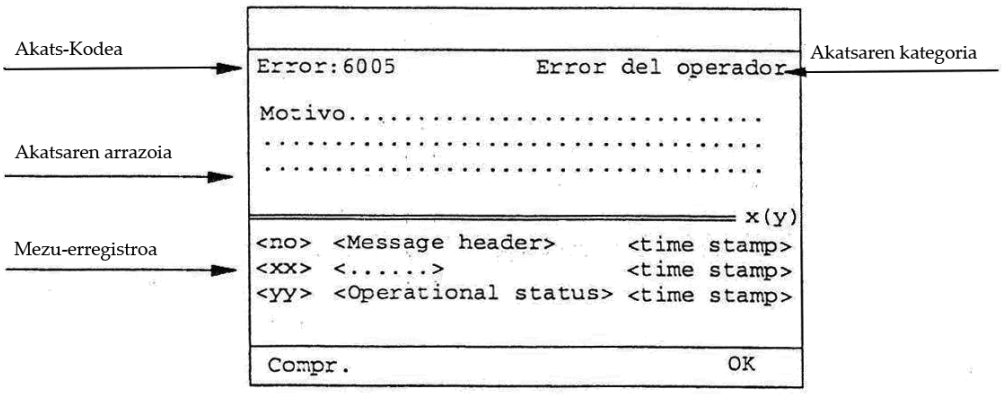
5.- Gelditu programa, programazio-unitateko *STOP* pultsadorearekin.

6.- Sakatu kontrol-paneleko Motor desaktibatuak tekla.

7.- Ipini berriro <250 mm/s.

1.9. AKATSAK

Sisteman akatsen bat gertatzen den bakoitzean, 43. irudikoaren itxurako akats-mezua agertuko da pantailan.



43. irudia

Akatsaren–kode-zenbakia

Akats bakoitzari zenbaki bakarra dagokio.

Akats-maila

Sistemak maila edo kategoriatan sailkatzen ditu akatsak. Maila bakoitzak bere kode segidak ditu. Hala, operatzailearen akats bakoitzari 6001-6999 tarteko zenbaki bat dagokio.

Arrazoia

Akats bat azaltzen den bakoitzean, akatsa zerk eragin duen azaldu behar da hizkuntza laburtu gabekoan. Informazio hori mezu-erregistroan geldituko da. Hardware-aren akatsei buruz gehiago jakin nahi izanez gero, kontsultatu produktuaren eskuliburua.

Mezu-erregistroa

Berriki izandako akatsen berri ematen du. Lehen lerroan ageri den akatsa leihoan dakusaguna da. Akats-zerrendak akatsaren kodea, deskribapen laburra eta sistemak akatsa zer ordutan erregistratu duen erakusten ditu. Kurtsorearekin zerrendako akatsen bat aukeratuz gero, leihoa eguneratu egingo da garrantzizko kode, arrazoi eta akats-mailarekin batera.

Akats jakinak prozesatzeko erari buruzko laguntza lortzeko, Compr. funtzio-tekla erabil liteke.

OK tekla sakatuz gero, agerian zegoen akats-mezuen leihoa desagertu egingo da. Aginte-palankaren laguntzaz, erabiltzailea robota eskuz eragiten eta lan-eremutik kanpo mugitzen saiatuko da. Berehala, akats-mezuaren adibidea agertuko da.

1.10. SISTEMA GELDITZEA

Gainerako ariketekin jarraitzeko asmoa izanez gero, atal hau alde batera utz liteke.

Robota deskonektatzean, irteerako seinale guztiak 0 egoerara itzuliko dira. Horrek manipulazio-pintza eta periferikoetan eragin dezake. Hori dela eta, lehenbizi ziurtatu talde osoa (sistema eta periferikoak), eta ziurtatu inor ez dagoela robotaren lan-eremuan.

Hori guztia jakinda, hauek dira sistema gelditzeko jarraitu beharreko pausoak:

1.- Programa lanean ari bada, sakatu programazio-unitatea gelditzeko tekla (Stop).

2.- Ondoren, sakatu kontrol-paneleko Motor desaktibatuak tekla.

3.- Hori egin eta gero, desaktibatu etengailu nagusia.

Robotaren memoriak ez du aldaketarik jasango, segurtasuneko elikadura-sistema bateriaduna baitu.

1.11. PROGRAMA-ALDAKETA

Hurrengo atalak programak sortu edota editatu nahi dituzten erabiltzaileei zuzenduta daude.

Atal honetan, aurrez ireki eta martxan jarritako programa aldatzeko hainbat era aurkituko dira. Urrats hauek egin behar dira:

- Exekutatu programa pausoz pauso aldatu nahi den posiziora iritsi arte.

- Aldatu posizioa.

- Aldatu agindu bateko arrazoia.

- Sartu posizio-agindu berria.

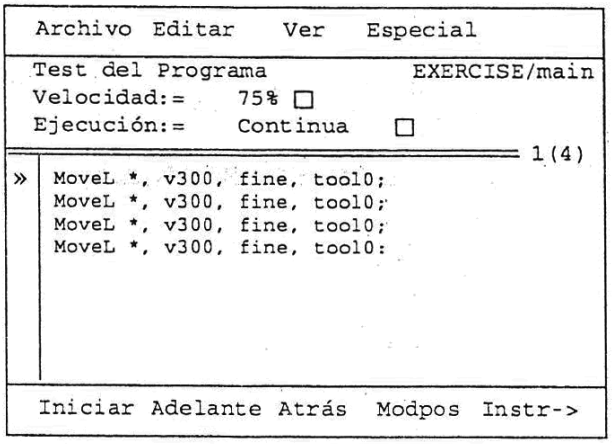
- Programatu itxarote-baldintza berria.

1.11.1. POSIZIO-ALDAKETAK

1.- Erabiltzaileak aurreko ariketak alde batera utzi baditu, Programa leiho-tekla sakatu behar du.

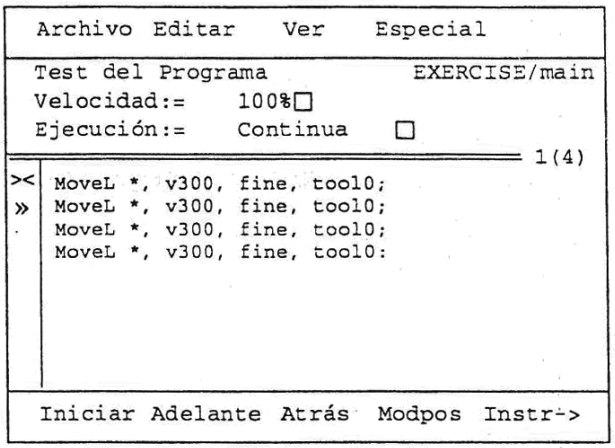
44. irudia

Pantailan 45. irudia agertuko da.



45. irudia

2.- Eragin egokigailuari, eta sakatu *Adelante* funtzio-tekla. Mugitu robota programaren lehen posiziora (lehenengo aginduak hautatuta egon behar du).



46. irudia

3.- Ondoren, aginte-palanka erabiliz, mugitu robota beste posizio batera.

4.- Sakatu Modpos funtzio-tekla. Aurrez izendatutako jatorrizko posizioa robotak orain duen posiziora aldatuko da.

5.- Eragin egokigailuari, eta sakatu berriz Adelante funtzio-tekla, robota hurrengo posiziora mugitzeko.

Egin berriro 3. puntutik 5. puntura bitarteko pausoak, eta pasatu ariketa-programako posizio guztietatik.

6.- Egiaztatu programa pausoz pauso abiadura gutxitu gabe. Gelditu programa edozein posiziotan, eta sakatu Instr funtzio-tekla (programaren testa amaitzeko). Beheko irudiko leihoaren antzekoa agertuko da.

```
Archivo  Editar    Ver     IPL1    IPL2
Instr del Programa          EXERCISE/main
                                     1(4)
MoveL *, v300, fine, tool0;
MoveL *, v300, fine, tool0;
MoveL *, v300, fine, tool0;
MoveL *, v300, fine, tool0:

Copiar  Pegar ArgOpci...  ModPos  Test->
```

47. irudia

1.11.2. ARGUMENTU-ALDAKETA

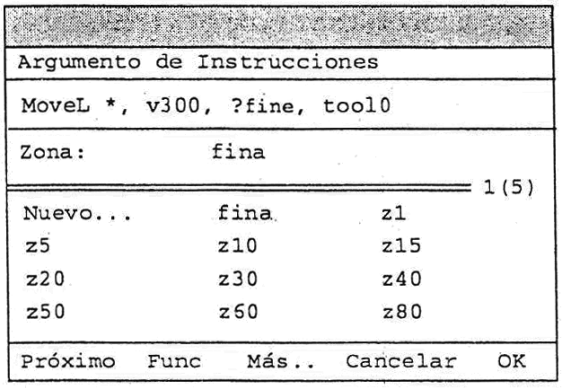
Hautaturik dagoen lehen mugimendu-aginduko (*MoveL*) argumentuetako bat aldatuko dugu ondoren. Urrats hauek egin behar dira:

1.- Hautatu FINE argumentua (ikusi 48. irudia). Mugitu kurtsorea eskuineranzko desplazamendu-gezia erabiliz.

```
Archivo  Editar    Ver     IPL1    IPL2
Instr del Programa          EXERCISE/main
                                     1(4)
MoveL *, v300, fine, tool0;
MoveL *, v300, fine, tool0;
MoveL *, v300, fine, tool0;
MoveL *, v300, fine, tool0:

Copiar  Pegar ArgOpci...  ModPos  Test->
```

48. irudia

2.- Sakatu Sartu tekla, eta pantailan, 49. irudiko leihoa agertuko da.



49. irudia

3.- Mugitu kurtsorea *z10* posiziora.

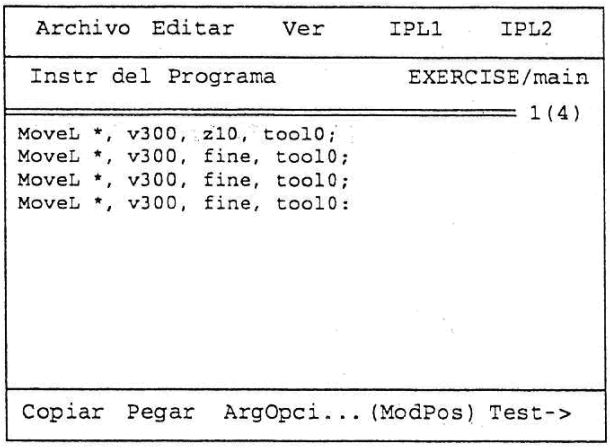
4.- Sakatu Sartu tekla, eta *FINE* argumentua *z10*-era pasatuko da.

5.- Sakatu *OK* tekla. Agindua *z10*-era aldatu da orain.

6.- Mugitu kurtsorea agindu osoa hautatzeko modura (ikusi 50. irudia).

1.11.3. AGINDUA GEHITZEA

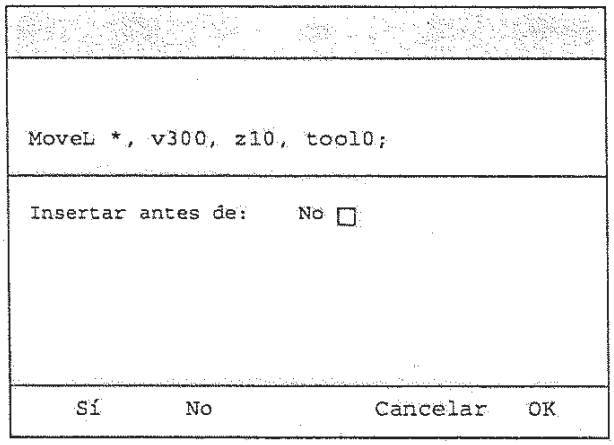
Lehen aginduaren ondoren, programari mugimendu-agindu bat gehitzea lortu nahi da. Pantailan *Instr del Programa* leihoa (ikusi 50. irudia) agertuko da.



50. irudia

1.- Sakatu *Copiar* funtzio-tekla, hautaturiko lehen agindua kopiatzeko.

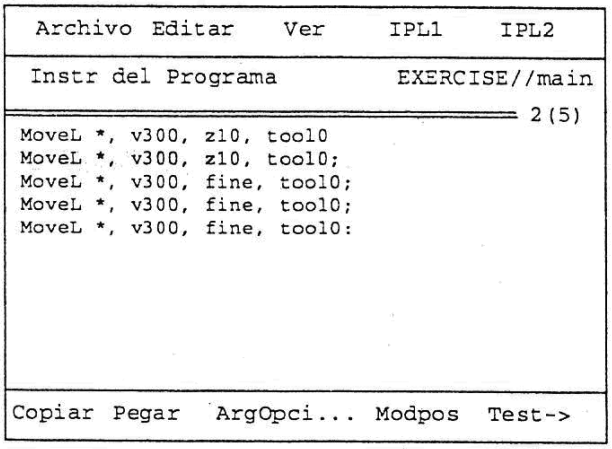
2.- Ondoren, sakatu *Pegar* tekla, eta 51. irudiko leihoa agertuko da. Programatik hautaturiko lehen agindua denez, agindu berria non sartu nahi dugun galdetuko digu.



51. irudia

3.- Hautatu *No*, eta sakatu *OK* tekla.

4.- Agindu berria aurrez hautaturiko aginduaren azpian txertatuko da, eta nabarmenduta agertuko da.



52. irudia

5.- Aginte-palankaren laguntzaz, mugitu robota nahi duzun posiziora.

6.- Sakatu Modpos (ikusi 52. irudia).

7.- Egiaztatu programaren exekuzioa modu jarraikiren bat erabiliz.

8.- Hautatu Test➔funtzio-tekla.

9.- Sakatu egokigailua.

10.- Sakatu *Iniciar* tekla.

1.11.4. ITXAROTE-BALDINTZAREN PROGRAMAZIOA

Itxarote-baldintza programatutakoan, robota denbora jakin batez itxaronaraziko dugu. Agindu berria laugarren aginduaren ondoren txertatzen da.

Hona hemen urratsak:

1.- Programaren exekuzioa amaitu ondoren, sakatu Instr funtzio-tekla, eta 53. irudiko leihoa agertuko da.

```
Archivo  Editar    Ver     IPL1     IPL2
Instr del Programa          EXERCISE//main
                                      1(5)
MoveL *, v300, z10, tool0
MoveL *, v300, z10, tool0
MoveL *, v300, fine, tool0
MoveL *, v300, fine, tool0
MoveL *, v300, fine, tool0

Copiar  Pegar  ArgOpci...  Modpos  Test->
```

53. irudia

2.- Desplazamendu bertikaleko gezien bidez (gorantz eta beherantz), eraman kurtsorea programako laugarren agindura. Hautaturiko aginduaren azpian, agindu berria txertatuko da.

3.- Hautatu *IPL1: Varios* (ikusi 54. irudia).

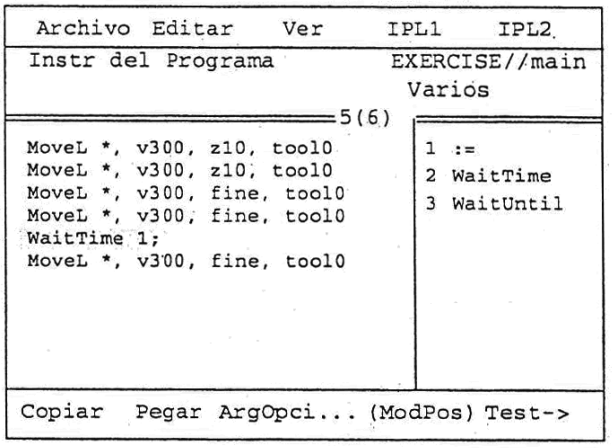
```
Archivo  Editar    Ver     IPL_1    IPL_2
Instr del Programa          EXERCISE//main
                              Varios
                                      4(5)
MoveL *, v300, z10, tool0          1 :=
MoveL *, v300, z10, tool0          2 Comment
MoveL *, v300, fine, tool0         3 WaitDI
MoveL *, v300, fine, tool0         4 WaitTime
MoveL *, v300, fine, tool0         5 WaitUntil

Copiar Pegar  ArgOpci...  ModPos  Test->
```

54. irudia

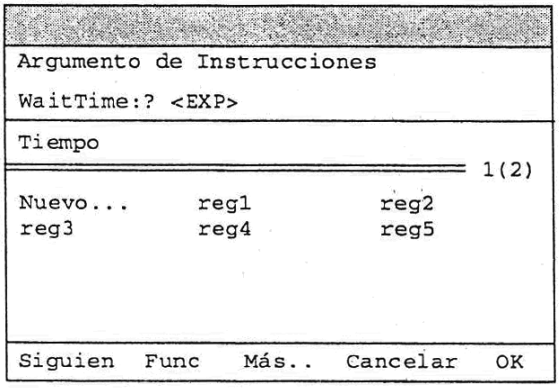
4.- Hautatu agindu-zerrendako agindu bat, bi era hauetakoren bat erabiliz:

- Zenbaki-teklatua erabiliz, sartu itxarote-aginduaren (WaitTime) aurretik ageri den zenbakia (4) (ikusi 54. irudia).

- Markatu hautatze-zerrenda zerrenda-tekla sakatuta. Ondoren, hautatu agindua, eta sakatu Sartu tekla (ikusi 55. irudia).



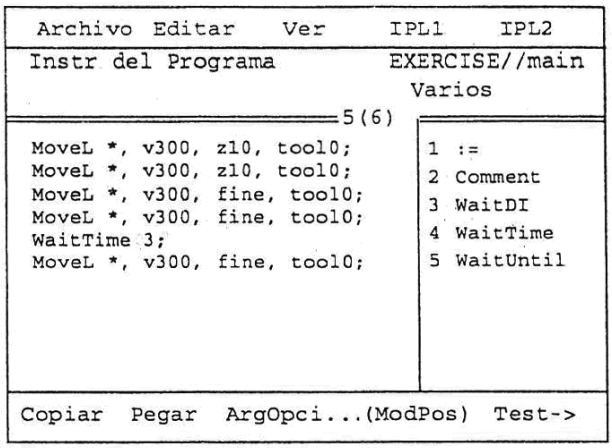
55. irudia



56. irudia

5.- Zenbaki-teklatua erabiliz, sartu 3 zenbakia, hiru segundoko itxarote-baldintza hautatzeko.

6.- Sakatu OK tekla.



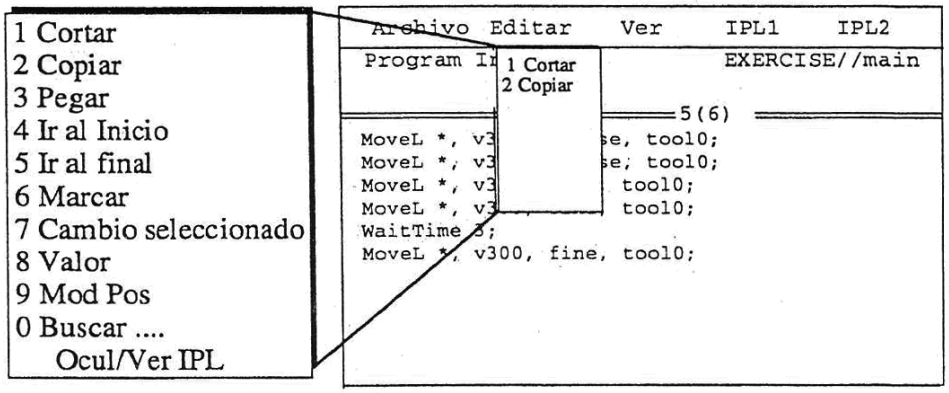
57. irudia

7.- Sakatu *Edit: OculILP* funtzio-tekla, hautatze-zerrenda desagerrarazteko.

8.- Orain, egiaztatu berriro programaren exekuzioa programako Test leihoaren bidez. Hautatu exekuzio jarraikia.

Oharra:

Editar menuak programa editatzeko erabiltzen diren hainbat funtzio ditu.



58. irudia

Sakatu berriz *Editar* menu-tekla menua desagerrarazteko.

1.12. PROGRAMA DISKETEAN NOLA GORDE

Eragiketa honen bidez, disketean kopiatu nahi da programa; eta, horretarako, dentsitate altuko 3.5” HD disketeak erabili behar dira.

Argi! Ez erabili instalazio-disketea (Set Up) programak gordetzeko.

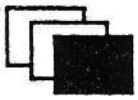
Diskete berria erabiliz gero, lehendabizi formateatu; eta, horretarako, jo *Varios* aukerara eta sakatu *Administrador de Archivos* tekla.

1.12.1 DISKETEA FORMATEATZEA

Oharra: informazioa duen diskete bat formateatuz gero, gordeta dagoen informazioa galdu egingo da.

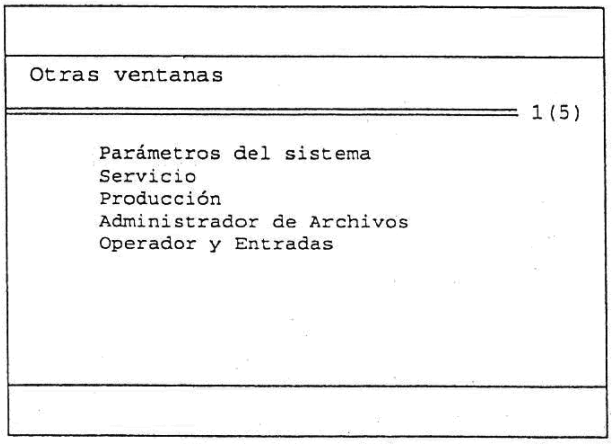
1.- Sartu disketea kontrol-armairuaren aurrealdean dagoen disko-unitatean.

2.- Hautatu Varios leihoa.



59. irudia

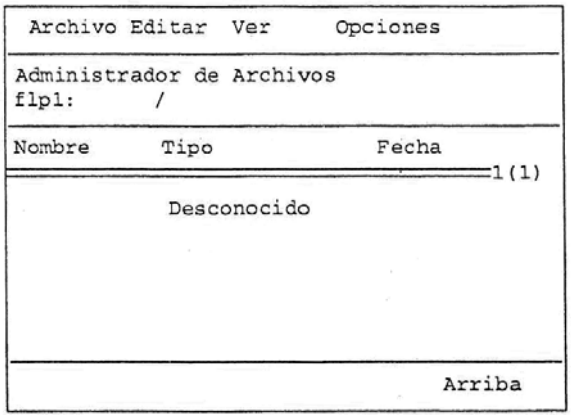
Leiho erabilgarrien zerrenda agertuko da pantailan (ikusi 60. irudia).



60. irudia

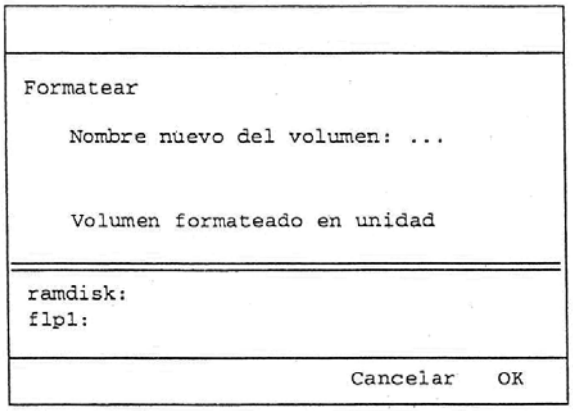
3.- Hautatu zerrendako *Administración* de *Archivos* aukera.

4.- Sakatu *Sartu* tekla.



61. irudia

5.- Hautatu Opciones: Formatear.



62. irudia

6.- Artxiboari izena jarri nahi izanez gero, mugitu kurtsorea leihoaren goiko aldera.

7.- Sakatu Sartu tekla.

8.- Sartu artxiboari dagokion izena.

9.- Kokatu berriro kurtsorea leihoaren beheko aldean.

10.- Hautatu *flp1* unitatea.

11.- Jarri martxan disketea formateatzeko prozesua; horretarako, sakatu OK tekla.

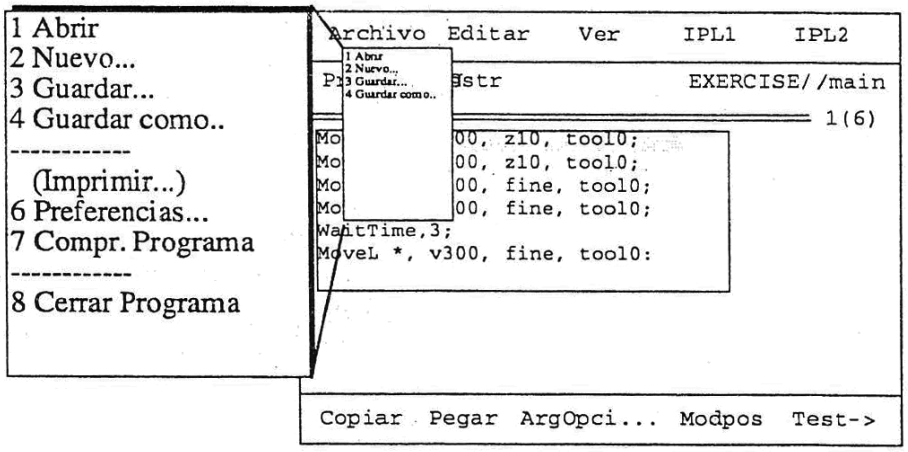
12.- Itxaron elkarrizketa-leihoa programazio-unitatetik desagertu arte.

1.12.2. DISKETEAN GORDETZEA

1.- Hautatu Instr del Programa funtzioa.

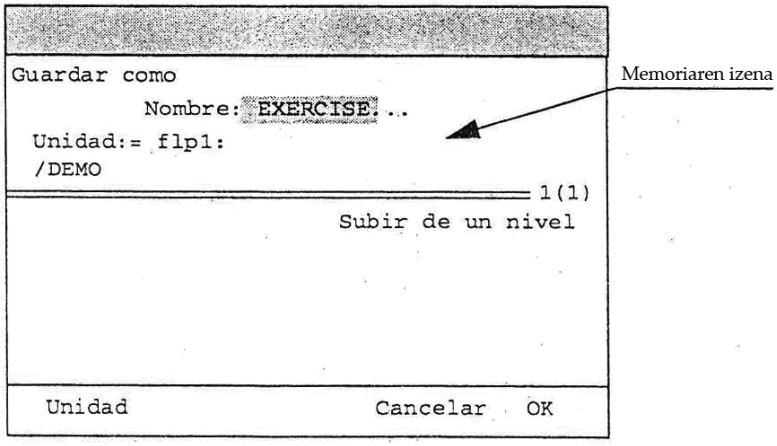
63. irudia

2.- Sakatu *Archivo* menu-tekla.



64. irudia

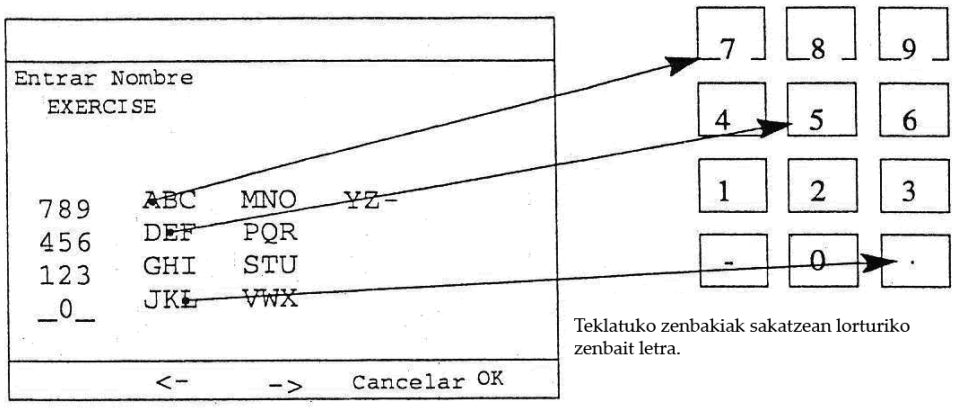
3.- Hautatu *Guardar como*..,. eta sakatu Sartu tekla.



65.irudia

4.- Sakatu *Unidad* funtzio-tekla, gordetzeko modua hautatzeko. Hala, leihoko 3. lerroan flp1 agertuko da (ikusi 65.irudia).

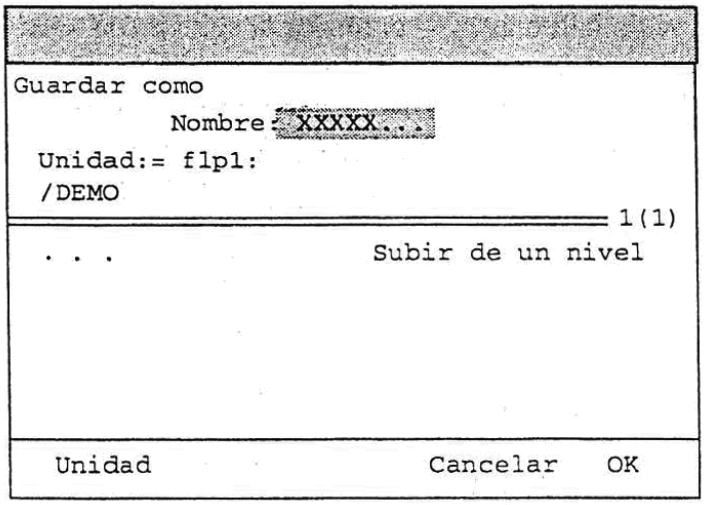
5.- Sakatu Sartu tekla, aukerako izena sartzeko.



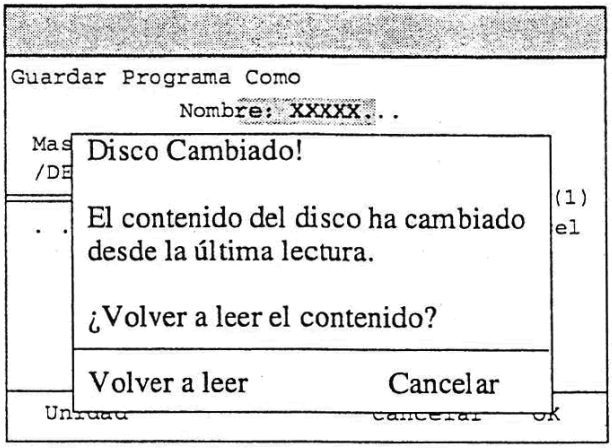
66. irudia

Bost karaktere-talde agertuko dira pantailan, eta talde bakoitza zenbaki-teklatuko tekla batek ordezkatzen du. Karaktere batetik bestera mugitzeko erabili ➔ eta ➔ funtzio-teklak.

Erabili *Ezabatu* tekla ⌫ ikusgai den izena edota oker sarturiko edozein datu ezabatzeko.
6.- Orain, programari izen berria eman behar zaio. Behin izen berria sarturik, sakatu OK tekla.



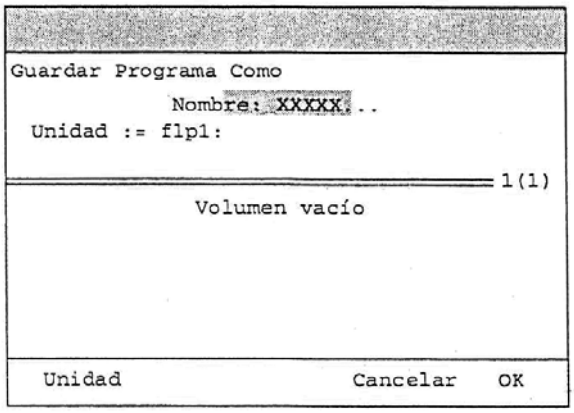
67. irudia



7.- Sakatu berriz OK tekla, eta informazio-leihoa agertuko da: diskoko informazioa aldatu egin dela jakinarazi, eta ikusi nahi al dugun galdetuko digu.

68. irudia

8.- Sakatu Sartu tekla, aldaketak ikusteko.



69. irudia

9.- Programa disketean gordeko da OK tekla sakatzean.

10.- Itxi elkarrizketa-leihoa.

1.13. SARRERA/IRTEEREN SEINALEAK

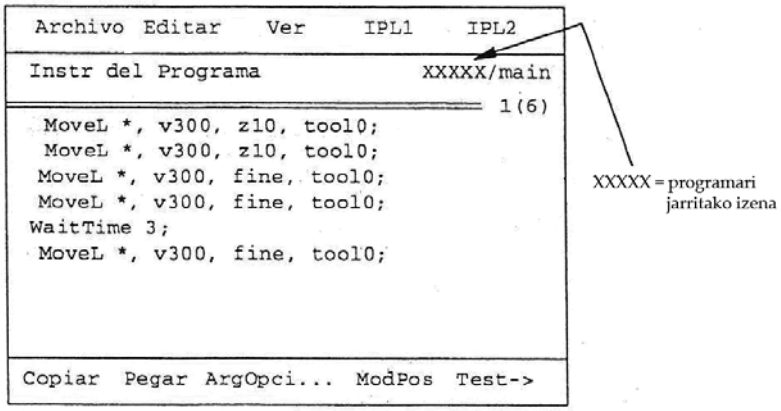
Atal honetan, irteera digitala aktibatuko duen agindua nola programatu azalduko da. Programa exekutatu ondoren, eskuz ireki S/I-ren zerrenda, eta bilatu seinalea.

1.13.1 S/I AGINDU PROGRAMAZIOA

1.- Hautatu *programa*-leihoa.

70. irudia

Beheko leihoa agertuko da pantailan, eta goiko eskuineko aldean, programari 1.12. atalean jarritako izena agertuko da.



71. irudia

Agindu berria (irteera baten aktibazioa) hautatutako aginduaren azpian txertatuko da zuzenean. Hautatu programako hirugarren agindua.

2.- Hautatu *IPL1: E/S*, eta leiho hau agertuko da:

```
Archivo  Editar   Ver    IPL1    IPL2
Instr del Programa            XXXXX/main
                              E/S
                        4(7)
MoveL *, v300, z10, tool0;     1 InvertDO
MoveL *, v300, z10, tool0;     2 PulseDO
MoveL *, v300, fine, tool0;    3 Reset
Set do4;                       4 Set
MoveL *, v300, fine, tool0;    5 SetAO
WaitTime 3;                    6 SetDO
MoveL *, v300, fine, tool0:    7 SetGO
                               8 WaitDI
Copiar  Pegar  ArgOpci... ModPos  Test->
```

72. irudia

Hautatu leiho horretako *Set* funtzioa, 1.11. atalean WaitTime itxarote-baldintzaren agindua hautatutako modu berean; hala, beste leiho bat agertuko da.

```
Argumento de Instrucciones
Set ? <EXP>
Señal
                                   1(11)
Nuevo...    d01        d02
d03         d04        d05
d06         d07        d08
d09         d010       do11
Siguien  Func  Más...  Cancelar  OK
```

73. irudia

3.- Ageri den zerrendatik hautatu *do4* eta sakatu OK; beraz, beste leiho hau azalduko da:

```
 Archivo  Editar    Ver     IPL1     IPL2
 Instr del Programa          EXERCISE//main
  E/S
                          3(7)
 MoveL *, v300, z10, tool0;      1 InvertDO
 MoveL *, v300, z10, tool0;      2 PulseDO
 MoveL *, v300, fine, tool0;     3 Reset
 MoveL *, v300, fine, tool0;     4 Set
 WaitTime 3;                     5 SetAO
 MoveL *, v300, fine, tool0;     6 SetDO
                                 7 SetGO
                                 8 WaitDI
 Copiar  Pegar  ArgOpci... ModPos  Test->
```

74. irudia

4.- Sakatu Edit: *OcullPL* funtzio-tekla, eta agindu-zerrenda desagertuko da. Horren ondorioz, leiho berria agertuko da.

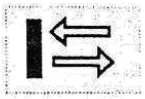
```
 Archivo  Editar    Ver     IPL1     IPL2
 Instr del Programa          XXXXX/main
                                      4(8)
 MoveL *, v300, z10, tool0;
 MoveL *, v300, z10, tool0;
 MoveL *, v300, fine, tool0;
 Set do4;
 MoveL *, v300, fine, tool0;
 WaitTime 3;
 MoveL *, v300, fine, tool0;

 Copiar  Pegar  ArgOpci... (ModPos) Test->
```

75. irudia

5.- Egiaztatu programa Test funtzioa erabilita, aginduz agindu, programak Set do4 irakurri arte.

6.- Orain, egiaztatu eskuz seinalearen egoera.

7.- Sakatu *Sarrerak/Irteerak* leiho-tekla.



Sarrera/irteera leiho-tekla

Aurreko orria

Hurrengo orria

76. irudia

8.- Topatu sarrera/irteeren zerrendan do4 seinalea, desplazamendu bertikalerako gezien laguntzaz, eta hautatu.

9.- Egiaztatu balioa. Hala nahi izanez gero, balioa alda daiteke pantailan ageri diren 0/1 funtzio-teklen bidez.

10.- Aldatu seinalearen balioa, eta sakatu Progr. funtzio-tekla.

11.- Exekutatu berriz programa, Test funtzioa erabilita, eta egiaztatu berriro seinalearen balioa.

Oharra: eskuzko sarrera/irteeren pantailako *Ver* menuko zerrenden bidez, pantailan ageri diren sarrerako seinaleak edota irteerako seinaleak baino ezin dira hautatu.

Oinarrizko aginduak

2

2.1. RAPID PROGRAMAZIO-HIZKUNTZA. SARRERA

2.1.1. SARRERA

ABB enpresak bere robot-generazio berriarentzat garaturiko programazio-hizkuntza dugu. 1994. urtean kaleratu zenetik, aurrerapauso handia izan da robotak programatzeko erreminten garapenean.

Maila altuko testu-hizkuntza da, eta zenbait ezaugarri garrantzitsu biltzen ditu:

- prozedura eta funtzioen erabilera
- parametrotan adieraz litezkeen errutinak erabiltzeko aukera
- programaren egitura guztiz modularra
- errutina eta datu orokor nahiz lekuzkoak programatzeko aukera

RAPID eran idatzitako programa robotaren lana azaltzen duten zenbait agindutan datza. Egin beharreko lana deskribatzen duten hainbat parametrorekin dago lotua agindu bakoitza, eta parametro horiek era askotakoak izan daitezke:

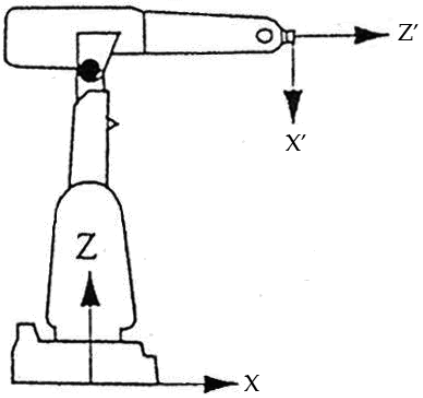
- zenbakizko balioak
- datu-erreferentziak
- datu-adierazpenak
- funtzioei deitzea

Programak programazio-paletan gidatutariko menu batzuen laguntzaz garatzen dira. Paletan kokaturiko *joystick*-aren eta gainerako botoien bidez (mugimendu mota, kanpo-ardatzak, S/I, etab.), badago datuak sartzea edota inguruan eragitea. Programak zuzenean idatz daitezke, ohiko testu-editorea duen PCan, eta, gero, diskete baten laguntzaz, robotaren memoriara pasatu.

Ohiko erabileretarako programazio-hizkuntzak gogora ekar ditzakeen programa da. *RAPID* eran garaturiko programei ataza (egitekoa) deritze, eta programa berarekin batera sistemaren zenbait modulu barneratzen dituzte. Programak edozein unetan erabil ditzakeen errutinak eta datu orokorrak dituzte barnean modulu horiek. Aldi berean, programa modulu ugaritan bana daiteke, eta batek nagusia izan behar du. Modulu horietako bakoitzak azpimoduluak ditu, hainbat exekuzio-errutinez gainera.

2.1.2. PROGRAMAREN EGITURA

Programa atal nagusi batez eta hainbat moduluz eta errutinaz osaturik dago (ikusi 77. irudia).



77. irudia

Hiru azpiprograma posible daude:

- *Prozedura*: inongo informaziorik itzultzen ez duen eta agindu gisa erabiltzen den errutina.

- *Funtzioa*: modu jakineko datua itzuli eta adierazpen gisa erabiltzen den errutina.

- *TRAP errutina*: etenaldiei lotzen zaizkien eta horiek aktibatzean exekutatzen diren errutinak dira. Ezin dira inoiz modu esplizituan deitu.

2.1.3. DATU MOTAK

Erabilera-datuak, osaeraren arabera, bi motatakoak izan daitezke:

- Atomikoak: ez dira beste funtzioen arabera definitzen eta ezin dira osagaietan banatu.

- Erregistrokoak: dagozkien izenekin ordenaturiko osagai zerrendaz osaturik daude. Aldi berean, osagaiak atomikoak edota erregistro motakoak izan daitezke.

Programan duten irismenaren arabera, erabilera-datuak bi motatakoak izan daitezke:

- Lekukoak: bere modulu edo azpiprograman bakarrik dira ezagunak.

- Orokorrak: modulu eta azpiprograma guztietan dira ezagunak.

Programan duten iraunkortasunaren arabera, datuak hiru motatakoak izan daitezke:

- Konstanteak: balio berririk eman ezin zaien balio finkoko datuak ordezkatzen dituzte.

- Aldakorrak: programaren exekuzioan balio berriak har ditzaketen datuak dira.

- Iraunkorrak: programaren exekuzioan balioa aldatzen zaien bakoitzean, hasierako balioa ere aldatzen zaien datuak dira.

Kualitatiboki hiru motatako datuak ditugu:

- Zenbakizkoak: *num*

- Logikoak: *bool*

- Kateatuak: *string*

2.1.4. AURREZ DEFINITUTAKO DATUEN EGITURAK

Garrantzitsua da aurrez definitutako atal ugariko datuen egiturek zer izaera duten nabarmentzea.

- **Loaddata**

* Deskribapena: robotaren eskumuturrean kokaturiko zama deskribatzen du.
* Atalak:
  a) *mass*: zamaren pisua, kg-tan.

  b) *cog*: zamaren grabitate-zentroa.

  c) *aom*: grabitate-zentroaren barnean inertzia-ardatzek duten orientazioa, *kuaternion* eran idatzita.

  d) *ix,iy,iz*: zamaren inertzia-momentuak *X*, *Y* eta *Z* ardatzekiko, kgm^2-tan adierazita.

* Adibidea:
  PERS loaddata pieza:=[5,[50,0,50],[1,0,0,0],0,0,0];

- **Tooldata**

* Deskribapena: erremintaren ezaugarriak adierazteko egitura.
* Atalak:
  a) *robhold*: robotak erreminta duen ala ez definitzen du.

  b) *trframe*: erremintaren koordenatu-sistema (posizioa eta orientazioa).

  c) *tload*: loadata motako datua.

* Adibidea: :
  PERS tooldata pinza:= [TRUE,[[97,0,220],[0.94,0,0.383,0]]

  [5,[-23,0,75],[1,0,0,0],0,0,0]

- **Robtarget**

* Deskribapena: robotaren eta muturreko ardatzen kokapena adierazteko erabiltzen da.
* Atalak:
  a) *trans*: koordenatu-sistemaren *X*, *Y* eta *Z* ardatzen gaineko mugimendua.

  b) *rot*: koordenatu-sistemaren kuaternio-errotazioa.

  c) *robconf*: robotaren konfigurazioa (cfl, cf4, cf6 eta cfx)

  d) *extax*: kanpo-ardatzen kokapena.

* Adibidea:
  VAR robtarget posicion:= [[600,500,225],[1,0,0,0],[1,0,0,0],

  [11,12,9E9,9E9,9E9,9E9]];

- **Motsetdata**

* Deskribapena: programaren kokapen-aginduei eragiten dieten zenbait mugimendu-parametro definitzeko erabiltzen da.
* Atalak:
  a) *vel.oride*: programatutakoaren abiaduraren ehunekoa.

  b) *vel.max*: abiadura maximoa, mm/s-tan.

  c) *sing.wrist*: eskumuturraren bakantasuna ekiditeko erremintaren orientazioaren desbideratzea.

  d) *sing.arm*: besoaren bakantasuna ekiditeko erremintaren orientazioaren desbideratzea.

  e) *conf.ax1*: 1 ardatzaren gehienezko desbideratzea.

f) *conf.ax4*: 4 ardatzaren gehienezko desbideratzea.

g) *conf.ax6*: 6 ardatzaren gehienezko desbideratzea.

h)…

2.1.5. AGINDUEN SAILKAPENA

Agindu aritmetikoak eta logikoak

=	Balioa eman
Abs0	Balio osoa lortu
Add	Zenbakizko balioa gehitu
Clear	Balioa ezabatu
Decr	Balioari unitate bat gutxitu
Incr	Balioari unitate bat gehitu

Edizio-aginduak

Comment	Iruzkina
Label	Lerroaren izena
TPErase	Programazio-paletako izena ezabatu
TPReadFK0	Programazio-paletako funtzio-teklak irakurri
TPWrite	Programazio-paletan idatzi

Programaren kontrol-aginduak

CompactIF	Baldintza betez gero, orduan… (agindua)
EXIT	Programaren exekuzioa amaitu
FOR	Hainbat aldiz (kopuru jakina) errepikatu
GOTO	Agindu berrira joan
IF	Baldintza betez gero, orduan…; bestela,…
ProcCall	Prozedura berriari deitu
RAISE	Akats-gidariari deitu
RETRY	Akatsaren ondoren, berriro hasi
RETURN	Errutinaren exekuzioa amaitu
Stop	Programaren exekuzioa gelditu
TEST	Adierazpenaren balioaren arabera…
WaitTime	Denbora jakin bat itxaron
WaitUntil	Baldintza bete arte itxaron
WHILE	Errepikatu… bitartean

S/I-ren kontrol-aginduak

Ainput0	Sarrera-seinale analogikoaren balioa irakurri

InvertDO	Seinale digitalaren balioa alderantziz jarri
PulseDO	Irteera-seinale digitalean pultsua sortu
Reset	Irteera digitala berrezarri
Set	Irteera digitala ezarri
SetAO	Irteera analogikoaren balioa aldatu
SetDO	Irteera digitalaren balioa aldatu
WaitDI	Sarrera digitalaren ezarpenari itxaron
Write	Serie-kanalean edota letra-fitxategian idatzi
WriteBin	Serie bitarreko kanalean idatzi

Tenporizazioak

ClkStart	Denbora neurtzeko erlojua martxan jarri
ClkStop	Denbora neurtzeko erlojua gelditu
GetTime0	Erlojua zenbakizko balio-eran irakurri
WaitTime	Denbora jakin bat itxaron

Robotaren konfiguratzeko eta mugitzeko aginduak

Accset	Azelerazioa murriztu
ConfJ	Mugimendu artikulatuan konfigurazioa kontrolatu
ConfL	Mugimendu zuzenean robotaren konfigurazioa pantailaratu
GripLoad	Robotaren zama definitu
HoldMove	Robotaren mugimendua gelditu
LimConfL	Robotaren konfigurazioan onartutako okerra definitu
MoveC	Robota mugimendu zirkularrean mugitu
MoveJ	Robotaren mugimendu artikulatua
MoveL	Robotaren mugimendu zuzena
Offs0	Robotaren posizio-desplazamendua
RelMove	Robotaren mugimenduarekin jarraitu
SingArea	Puntu bakanen inguruko interpolazioaren definizioa
VelSet	Programaturiko abiadura aldatu

Mugimenduak definitutako posizioen arabera programatzen dira; hau da, robotari dagoen lekutik posizio jakin batera mugitzeko agintzen zaio.

MoveC, *MoveJ* eta *MoveL* aginduek konfigurazio berezia dute:

- *[Conc]*: exekuzio-bateratzailea. Robota mugitzen den bitartean, aginduak exekutatzen jarraitzen dute.

-*Topoint*: robotaren helmuga-puntua (*robtarget* datu mota)

- *Speed*: mugimenduari eragiten dion abiadura (*speeddata* datu mota)

- *[\V]*: TCParen abiadura espezifikoa (*num* datu mota)

- *[\T]*: mugitzen pasatu beharreko denbora segundotan (*num* datu mota)

-*Zone*: ondorengo agindua betetzen hasteko mugimendu-eremua (*zonedata* datu mota)

- *[\Z]*: TCParen doitasuna (*num* datu mota)

- *Tool*: erabiltzen ari den erreminta (*tooldata* datu mota)

- *[\Wobj]*: mugimenduari loturiko lan-objektua (*wobjdata* datu mota)

(RAPID) Agindu zerrenda osoa

=	Balioa eman
Abs()	Balio osoa lortu
Ainput()	Sarrera-seinale analogikoaren balioa irakurri
Accset	Azelerazioa murriztu
Add	Zenbakizko balioa gehitu
Clear	Balioa ezabatu
ClkStart	Denbora neurtzeko erlojua martxan jarri
ClkStop	Denbora neurtzeko erlojua gelditu
Comment	Iruzkina
CompactIF	Baldintza betez gero, orduan… (agindua)
ConfJ	Mugimendu artikulatuan konfigurazioa kontrolatu
ConfL	Mugimendu zuzenean robotaren konfigurazioa pantailaratu
Decr	Balioari unitate bat gutxitu
EXIT	Programaren exekuzioa amaitu
FOR	Hainbat aldiz (kopuru jakina) errepikatu
GetTime()	Erlojua zenbakizko balio eran irakurri
GOTO	Agindu berrira joan
GripLoad	Robotaren zama definitu
HoldMove	Robotaren mugimendua gelditu
IF	Baldintza betez gero, orduan…; bestela,…
Incr	Balioari unitate bat gehitu
InvertDO	Seinale digitalaren balioa alderantziz jarri
Label	Lerroaren izena
LimConfL	Robotaren konfigurazioan onartutako okerra definitu
MoveC	Robota mugimendu zirkularrean mugitu
MoveJ	Robotaren mugimendu artikulatua
MoveL	Robotaren mugimendu zuzena
Offs0	Robotaren posizio-desplazamendua
Open	Serieko kanal edo fitxategia ireki
Present ()	Aukerako parametroa erabiltzen ari dela egiaztatu
ProcCall	Prozedura berriari deitu
PulseDO	Irteera-seinale digitalean pultsua sortu
RAISE	Akats-gidariari deitu

RelMove	Robotaren mugimenduarekin jarraitu
Reset	Irteera digitala berrezarri
RETRY	Akatsaren ondoren, berriro hasi
RETURN	Errutinaren exekuzioa amaitu
Set	Irteera digitala ezarri
SetAO	Irteera analogikoaren balioa aldatu
SetDO	Irteera digitalaren balioa aldatu
SetGO	Irteera digital multzo bati balioa aldatu
SingArea	Puntu bakanen inguruko interpolazioaren definizioa
Stop	Programaren exekuzioa gelditu
TEST	Adierazpenaren balioaren arabera…
TPErase	Programazio-paletako izena ezabatu
TPReadFK()	Programazio-paletako funtzio-teklak irakurri
TPWrite	Programazio-paletan idatzi
VelSet	Programaturiko abiadura aldatu
WaitDI	Sarrera digitalaren ezarpenari itxaron
WaitTime	Denbora jakin bat itxaron
WaitUntil	Baldintza bete arte itxaron
WHILE	Errepikatu… bitartean
Write	Serie-kanalean edota letra-fitxategian idatzi
WriteBin	Serie bitarreko kanalean idatzi

2.2. DATU MOTAK

2.2.1. ORIENTAZIOA

Orient

Errotazio eta orientazioetarako erabiltzen da (erreminta nahiz koordenatu-sistemen antzera).

Azalpena

Lau elementuz osaturiko kuaternion (laukote) gisa azaltzen da orientazioa: q_1, q_2, q_3 eta q_4. Zenbakizko datuak baino ez diren lau osagai dira.

q_1: Quaternion 1 Datu mota: *num*

q_2: Quaternion 2 Datu mota: *num*

q_3: Quaternion 3 Datu mota: *num*

q_4: Quaternion 4 Datu mota: *num*

Adibidea

VAR orient orient1;

Orient:= [1,0,0,0];

orient1 orientazioari q_1 = 1, q_2-q_4 = 0 balioa eman zaio; balio horiek ez dagozkio inongo errotaziori.

Mugak

Orientazioak normalizatua izan behar du; hau da, lauren baturak 1 balio behar du.

$$q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1$$

Zer da kuaterniona?

Koordenatu-sistemaren orientazioa (erremintarena, adibidez) errotazio-matrize gisa azaldu daiteke. Errotazio-matrize horrek koordenatu-sistemako ardatzen norabidea deskribatzen du erreferentzia-sistema batekiko (ikus 77. irudia).

Errotazioaren erreferentzia-sistemetako ardatzak (*X*, *Y*, *Z*) erreferentziazko koordenatu-sisteman honela adierazten diren bektoreak dira:

X= $= (x_1, x_2, x_3)$
Y= $= (y_1, y_2, y_3)$
Z= $= (z_1, z_2, z_3)$

Beraz, X bektorearen x osagaia erreferentziazko koordenatu-sisteman izango da, y osagaia , etab.

Hiru bektore horiek elkarrekin adieraz daitezke matrize bakar batean: errotazio-matrizean. Matrize horretan, bektore bakoitza zutabe bat izango da:

$$\begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{bmatrix}$$

Kuaterniona matrize hori bera adierazteko era zehatzagoa da; eta kuaternionak errotazio-matrizeko elementuetan oinarrituz kalkulatzen dira:

$$q_1 = \frac{\sqrt{x_1 + y_2 + z_3 + 1}}{2}$$

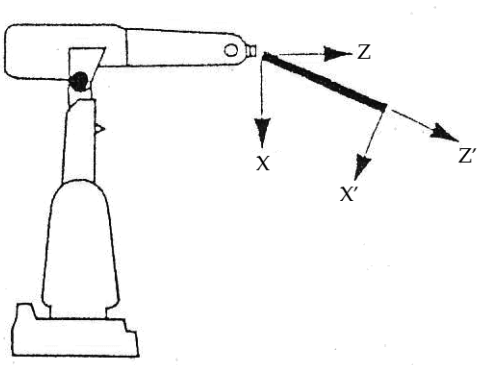
$$q_2 = \frac{\sqrt{x_1 - y_2 - z_3 + 1}}{2}$$ q_2-ren zeinua = ()-ren zeinua

$$q_3 = \frac{\sqrt{y_2 - x_1 - z_3 + 1}}{2}$$ q_3-ren zeinua = ()-ren zeinua

$$q_4 = \frac{\sqrt{z_3 - x_1 - y_2 + 1}}{2}$$ q_4-ren zeinua = ()-ren zeinua

1. adibidea

Erreminta bat bere *Z´* ardatza zuzen aurrera begira duela orientaturik dago (oinarriko erreferentzia-sistemako *X* ardatzaren norabide berean). Erremintaren *Y´* ardatza oinarriko erreferentzia-sistemako *Y* ardatzarekin bat dator.



78. irudia

Ardatzak honela erlazionaturik daude:

$$x' = -z = (0,0,-1)$$
$$y' = y = (0,1,0)$$
$$z' = x = (1,0,0)$$

Beraz, errotazio-matrize hau dagokio:

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}$$

Beraz, errotazio-matrizetik kuaternion hau atera daiteke:

$$q_1 = \frac{\sqrt{0+1+0+1}}{2} = \frac{\sqrt{2}}{2} = 0{,}707$$

$$q_2 = \frac{\sqrt{0-1-0+1}}{2} = 0$$

$$q_3 = \frac{\sqrt{1-0-0+1}}{2} = \frac{\sqrt{2}}{2} = 0{,}707$$

$$q_4 = \frac{\sqrt{0-0-1+1}}{2} = 0$$

2. adibidea

Erremintaren norabidea 30º biratu da eskumuturreko koordenatu-sistemarekiko X´ eta Z´ ardatzetan (ikusi 79. irudia). Nola definituko da erremintaren orientazioa erreminta-datuetan?

Ardatzak honela erlazionaturik daude:

$$x' = (\cos 30°, 0, -\sin 30°)$$
$$y' = (0,1,0)$$
$$z' = (\sin 30°, 0, \cos 30°)$$

Beraz, errotazio-matrize hau dagokio:

$$\begin{bmatrix} \cos 30° & 0 & \sin 30° \\ 0 & 1 & 0 \\ -\sin 30° & 0 & \cos 30° \end{bmatrix}$$

Errotazio-matrizetik kuaternion hau atera daitekeen:

$$q_1 = \frac{\sqrt{\cos 30° + 1 + \cos 30° + 1}}{2} = 0{,}965926$$

$$q_2 = \frac{\sqrt{\cos 30° - 1 - \cos 30° + 1}}{2} = 0$$

$$q_3 = \frac{\sqrt{1 - \cos 30° - \cos 30° + 1}}{2} = 0{,}258819$$

$$q_4 = \frac{\sqrt{\cos 30° - \cos 30° - 1 + 1}}{2} = 0$$

Egitura

<dataobject of orient>
<q1 of *num*>
<q2 of *num*>
<q3 of *num*>
<q4 of *num*>

2.2.2. POSIZIOAK

Pos

Posizioetarako (*x, y, z*) erabiltzen da.

Robtarget datu mota robotaren posizioa zehazteko erabiltzen da, eta erremintaren orientazioa eta ardatzen konfigurazioa barne hartzen ditu.

Deskribapena

Pos erako datuek (x, y, z) koordenatuen posizioa deskribatzen dute.

Osagaiak

x:	X ardatzeko posizioaren balioa.	Datu mota: *num*
y:	Y ardatzeko posizioaren balioa.	Datu mota: *num*
z:	Z ardatzeko posizioaren balioa.	Datu mota: *num*

Adibideak

VAR pos pos1;

pos1:=[500,0,940];

pos1 posizioari honako balio hauek esleitu zaizkio: x = 500 mm, y = 0 mm, z = 940 mm.

pos2:=pos1.x+50;

pos1 posizioa 50 mm desplazatu da X ardatzaren gainean.

Egitura

<dataobject of pos>
<x of *num*>
<y of *num*>
<z of *num*>

2.2.3. POSIZIO-DATUAK

Robtarget

Robotaren eta kanpo-ardatzen posizioa definitzeko balio du.

Deskribapena

Robotak eta kanpo-ardatzek hartu beharreko posizioa definitzeko balio dute.

Robota gai da posizio bera era askotara lortzeko; beraz, ardatzen konfigurazioa zehaztu behar da. Horrek ardatzen balioak definitzeko balio izango du anbiguotasun kasuetan. Adibidez:

- Robota aurreranzko edo atzeranzko posizioren batean badago

- 4 ardatza gora edo behera begira badago

- 6 ardatzak bira positibo edo negatiboa badu

Posizioa lan-objektuaren koordenatu-sisteman oinarrituz definituko da, programaren edozein desplazamendu barne dela. Aginduan erabilitakoa ez beste lan-objekturen batekin programatzen badugu posizioa, robota ez da mugituko espero den moduan. Ziurtatu posizio-aginduak programatzeko garaian lan-objektu bera erabiltzen dela. Erabilera okerrak kalteak sor ditzake langileengan edota ekipoan.

Osagaiak

trans Datu mota: *pos*

Erremintaren erdiko puntuaren posizioa adierazten du, (x, y, z) mm-tan adierazia.

Erabilitako objektuaren koordenatu-sistemaren arabera zehazten da posizioa, programaren desplazamendua barne dela. Lan-objekturik zehazten ez bada, koordenatu-sistema orokorra erabiliko da.

rot Datu mota: *orient*

Erremintaren orientazioa adierazten du kuaternion gisa.

Erabilitako objektuaren koordenatu-sistemaren arabera zehazten da orientazioa, programaren desplazamendua barne dela. Lan-objekturik zehazten ez bada, koordenatu-sistema orokorra erabiliko da.

robconf Datu mota: *confdata*

Robotaren ardatzen konfigurazioa (*cf1*, *cf4*, *cf6*, *cfx*) zehazten du. Erabilitako 1, 4 eta 6 ardatzen koadranteen arabera definitzen da. Lehen koadrante positiboa (0-90º) 0 moduan definiturik dago. Bestalde, *cfx* osagaia IRB5400 robot-modeloan soilik erabiltzen da.

extax Datu mota: *extjoint*

Kanpo-ardatzen posizioa zehazten du.

Posizioa ondoren erakusten den moduan definituko da ardatz bakoitzarentzat:

- Errotazio-ardatzetan, posizioa errotazio moduan definituko da, gradutan, kalibraketa-posiziotik abiatuta.

- Ardatz linealetan, posizioa distantzia moduan definituko da, mm-tan, kalibraketa-posiziotik abiatuta.

Eax_a… kanpo-ardatzak ardatz logikoak dira. Ardatz logikoen eta ardatz fisikoen erlazioa sistemaren parametroetan aurkituko dugu definituta.

9E9 balioa konektaturik ez dauden ardatzentzat definituko da. Eta posizio-datuetan definitutako ardatzak programa exekutatzeko konektaturik dauden ardatzekin bat ez badatoz, hauxe aplikatuko da:

- Kokapena ez badago posizio-datuetan definitua, 9E9 balioa ez da kontuan izango.

- Kokapena posizio-datuetan definitua badago eta ardatza konektatu gabe badago, 9E9 balioa ez da kontuan izango.

Adibideak

CONST robtarget p15:= [[600,500,225.3],[1,0,0,0],[1,1,0,0],[11,12.3,9E9,9E9,9E9,9E9]];

p15 posizioa honela definituko da:

- Robotaren posizioa: *x* = 600 mm, *y* = 500 mm, *z* = 225,3 mm objektuaren koordenatu-sisteman.

- Erremintaren orientazioa objektuaren koordenatu-sistemaren norabide berean.

- Robot-ardatzaren konfigurazioa: 1 eta 4 ardatzak, 90º-180º posizioan; 6 ardatza, 0-90º posizioan.

- Kanpo-ardatzen posizio logikoa: *a* eta *b* gradutan edo mm-tan azaldua (ardatz motaren arabera). *c* eta *f* ardatzak definitu gabe daude.

VAR robtarget p20;

P20:= CRobT( );
P20:= Offs (p20,10,0,0);

CRobT-ri dei eginez, *p20* posizioa aktibatuko da robotak une horretan duen posizio berean. Hortaz, posizioa 10 mm mugituko da *X* norabidean.

Mugak

Absolute Limit Modpos edizio-funtzio konfiguragarria erabiltzen denean, *robtarget* moduko datu-aren izenaren karaktere kopurua 14ra mugatua dago.

Egitura

<dataobjetct de *robtarget*>
    <trans de *pos*>
        <*x* de *num*>
        <*y* de *num*>
        <*z* de *num*>
    <*rot* de *orient*>
        <q_1 de *num*>
        <q_2 de *num*>
        <q_3 de *num*>
        <q_4 de *num*>
    <*robconf* de *confdata*>
        <*cf1* de *num*>
        <*cf4* de *num*>
        <*cf6* de *num*>
        <*cfx* de *num*>
    <*extax* de *extjoint*>
        <*eax_a* de *num*>
        <*eax_b* de *num*>
        <*eax_c* de *num*>
        <*eax_d* de *num*>
        <*eax_e* de *num*>
        <*eax_f* de *num*>

2.2.4. ABIADURA-DATUAK

Speeddata

Robota eta kanpo-ardatzak zer abiaduratan mugituko diren zehazteko balio du.

Deskribapena

Abiadura-datuek hau definitzen dute:

- erremintaren erdiko puntuaren abiadura
- erremintaren orientazio-abiadura
- kanpo-ardatz birakari nahiz linealen abiadura

Mugimendu mota bat baino gehiago konbinatzen direnean, sarritan abiaduretako batek mugatzen ditu mugimendu guztiak. Eta gainerako mugimenduen abiadurak mugimendu guztiak batera amaitzeko egokituko dira.

Robotaren ahalmenak ere mugatuko du abiadura, erabilitako robotaren eta ibilbidearen menpe baitago hori.

Osagaiak

v_tcp Datu mota: *num*

Erremintaren erdiko puntuaren (TCP) abiadura adierazten da, mm/s-tan.

Erreminta geldikorra edota kanpo-ardatz koordinatuak erabiltzen badira, abiadura lan-objektu-arekiko zehaztuko da.

v_ori Datu mota: *num*

Erremintaren erdiko puntuaren orientazio-abiadura, gradu/s-tan adierazita.

v_leax Datu mota: *num*

Kanpo-ardatz linealen abiadura, mm/s-tan adierazita.

v_reax Datu mota: *num*

Kanpo-ardatz birakarien abiadura, gradu/s-tan adierazita.

Adibidea

VAR speeddata vmedia:=[1000,30,200,15];

Vmedia abiadura-datua abiadura hauekin definituko da:

-1.000 mm/s TCParentzat

- 30 gradu/s erreminta orientatzeko

-z 200 mm/s kanpo-ardatz linealentzat

- 15 gradu/s kanpo-ardatz birakarientzat

Vmedia.v_tcp:=900;

TCParen abiadura 900 mm/s-ra aldatuko da.

Aurrez definitutako datuak

Zenbait abiadura-datu oinarrizko sistemaren moduluan definituta datoz:

Izena	*TCP abiad.*	*Orientazioa*	*Kanp-ard. linealak*	*Kanp-ard. birak.*
v5	5 mm/s	500º/s	5.000 mm/s	1.000º/s
v10	10 mm/s	500º/s	5.000 mm/s	1.000º/s
v20	20 mm/s	500º/s	5.000 mm/s	1.000º/s
v30	30 mm/s	500º/s	5.000 mm/s	1.000º/s
v40	40 mm/s	500º/s	5.000 mm/s	1.000º/s
v50	50 mm/s	500º/s	5.000 mm/s	1.000º/s
v60	60 mm/s	500º/s	5.000 mm/s	1.000º/s
v80	80 mm/s	500º/s	5.000 mm/s	1.000º/s
v100	100 mm/s	500º/s	5.000 mm/s	1.000º/s
v150	150 mm/s	500º/s	5.000 mm/s	1.000º/s
v200	200 mm/s	500º/s	5.000 mm/s	1.000º/s
v300	300 mm/s	500º/s	5.000 mm/s	1.000º/s
v400	400 mm/s	500º/s	5.000 mm/s	1.000º/s
v500	500 mm/s	500º/s	5.000 mm/s	1.000º/s

v600	600 mm/s	500º/s	5.000 mm/s	1.000º/s
v800	800 mm/s	500º/s	5.000 mm/s	1.000º/s
v1000	1.000 mm/s	500º/s	5.000 mm/s	1.000º/s
v1500	1.500 mm/s	500º/s	5.000 mm/s	1.000º/s
v2000	2.000 mm/s	500º/s	5.000 mm/s	1.000º/s
v2500	2.500 mm/s	500º/s	5.000 mm/s	1.000º/s
v3000	3.000 mm/s	500º/s	5.000 mm/s	1.000º/s
vmax	5.000 mm/s	500º/s	5.000 mm/s	1.000º/s

Egitura

<dato del tipo de *speeddata*>
< *v_tcp* de *num*>
< *v_ori* de *num* >
< *v_leax* de *num* >
< *v_reax* de *num* >

2.2.5. EREMU-DATUAK

Zonedata

Posizio batek nola amaitu behar duen zehazteko erabiltzen da; hau da, ardatzek, hurrengo posiziorantz abiatu aurretik, programaturiko posiziotik zer distantziara egon behar duten zehazteko.

Deskribapena

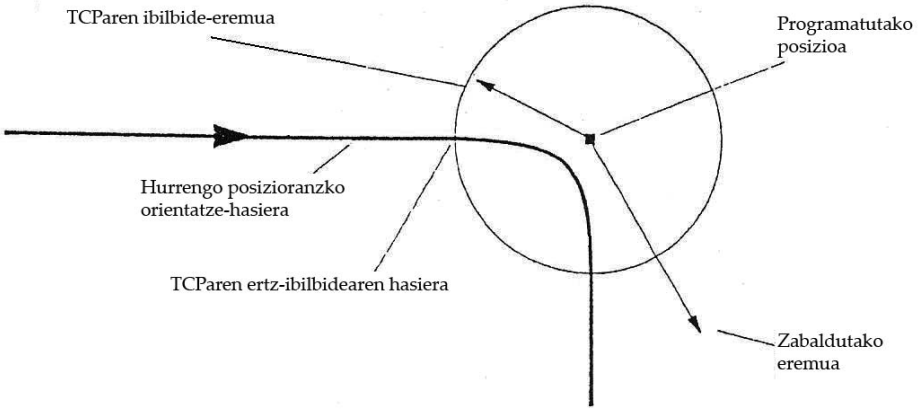
Posizio bat gelditze-punturen batean edota iragan-punturen batean amai daiteke.

Gelditze-puntua deskribatuz gero, nola robotak hala kanpo-ardatzak zehazturiko posiziora iritsi behar dute programaren exekuzioak hurrengo aginduarekin jarraitu aurretik.

Iragan-puntua deskribatuz gero, inoiz ez da programaturiko posiziora iritsiko. Horren ordez, mugimenduaren norabidea aldatzen da posiziora iritsi aurretik. Posizio bakoitzarentzat bi eremu defini daitezke:

- TCParen ibilbide-eremua

- erremintaren eta kanpo-ardatzen orientatze-eremua



79. irudia

Eremuek era berean funtzionatuko dute ardatzak mugitzen diren bitartean. Hala ere, eremuaren tamaina zertxobait alda liteke programatutakoarekiko.

Eremuaren tamaina ezin da inoiz izan gertueneko posizioaren erdia (aurrerantz ala atzerantz) baino handiagoa. Eremu handiagoren bat zehaztuz gero, robotak automatikoki gutxituko du.

Adibidez:

Trayec.pzone_tcp:=40;

TCParen ibilbide-eremua 40 mm-ra doitua dago.

Aurrez definitutako datuak

Badira oinarri-sistemaren moduluan dagoenekoz definiturik dauden zenbait eremu-datu.

Gelditze-puntuak

Izena

fine 0 mm

Iragan-puntuak

Izena	*TCP ibilb.*	*Orient.*	*Kanp.-ard.*	*Orient.Ard.lineal.*		*Errot. ard.*
z1	1 mm	1 mm	1 mm	0,1º	1 mm	0,1º
z5	5 mm	8 mm	8 mm	0,8º	8 mm	0,8º
z10	10 mm	15 mm	15 mm	1,5º	15 mm	1,5º
z15	15 mm	23 mm	23 mm	2,3º	23 mm	2,3º

z20	20 mm	30 mm	30 mm	3º	30 mm	3º
z30	30 mm	45 mm	45 mm	4,5º	45 mm	4,5º
z40	40 mm	60 mm	60 mm	6º	60 mm	6º
z50	50 mm	75 mm	75 mm	7,5º	75 mm	7,5º
z60	60 mm	90 mm	90 mm	9º	90 mm	9º
z80	80 mm	120 mm	120 mm	12º	120 mm	12º
z100	100 mm	150 mm	150 mm	15º	150 mm	15º
z150	150 mm	225 mm	225 mm	23º	225 mm	23º
z200	200 mm	300 mm	300 mm	30º	300 mm	30º

Egitura

<data object de *zonedata*>
<*finep* de *bool*>
<*pzone_tcp* de *num*>
<*pzone_ori* de *num*>
<*pzone_eax* de *num*>
<*zone_ori* de *num*>
<*zone_leax* de *num*>
<*zone_reax* de *num*>

2.3. AGINDUAK

2.3.1. ROBOTAREN MUGIMENDU BIRAKARIA

MoveC agindua

Agindu honek erremintaren erdiko puntua (TCP) helmuga jakin batera mugimendu birakariz eramateko balio du. Mugitzen den bitartean orientazioa konstante mantenduko da zirkuluarekiko.

Adibideak

MoveC p1,p2,v500,z30,herram2;

Erremintaren *herram2* TCPa mugimendu birakariz joango da *p2* posiziora, *v500* abiadura-datua eta *z30* eremu-datua erabiliz. Zirkulua hasiera-puntuaren, zirkuluko *p1* puntuaren eta *p2* helmuga-puntuaren arabera definitzen da.

MoveC,*,v500\T:=5,fine,pinza3;*

Erremintaren *pinza3* TCPa mugimendu birakariz joango da aginduan gorderiko puntu finkora. Zirkuluko puntua ere aginduan gorderik dago. Mugimendu guztia burutzeko 5 segundo beharko dira.

```
MoveL p1,v500,fine,herram1;
MoveC p2, p3,v500,z20,herram1;
MoveC p4,p1,v500,fine,herram1;
```

Zirkulu oso bat egiteko lau posizio behar dira.

Argumentuak

[\Conc] Datu mota: *switch*

Robota mugitzen den heinean, ondoren datozen agindu logikoak exekutatuz doaz. Argumentu horrek zikloaren denbora murrizteko balio du; adibidez, kanpo-atalen batekin komunikatzen ari denean eta sinkronizaziorik behar ez denean.

Conc argumentua erabiliz, gehienez, bost mugimendu-agindu erabil daitezke.

Argumentu hori aipatzen ez bada, hurrengo agindua aurrekoa amaiturikoan baino ez da exekutatuko.

PuntoCírculo Datu mota: *robtarget*

Robotaren zirkulu-puntua da. Zirkulu-puntua irteera- eta helmuga-puntuen arteko ibilbide zirkularreko posizio bat da. Doitasun handiagoa lortzeko, irteera- eta amaiera-puntuen erdibidean egon behar du, gutxi gorabehera. Muturreko punturen batetik gertuago egonez gero, robotak jakinarazi egingo du. Zirkulu-puntua posizio jakin bezala izendaturik egon daiteke, edota zuzenean aginduan sar daiteke "*" ikurrarekin adierazita. Kanpo-ardatzen posizioak ez dira erabiltzen.

AlPunto Datu mota: *robtarget*

Robotaren eta kanpo-ardatzen helmuga-puntua da. Posizio jakin gisa izendaturik egon daiteke edota zuzenean aginduan sar daiteke "*" ikurrarekin adierazita.

Velocidad Datu mota: *speeddata*

Mugimenduei aplikaturiko abiadura-datuak dira, eta TCParen abiadura eta erremintaren eta kanpo-ardatzen orientazioa definitzen dituzte.

[\V] Datu mota: *num*

TCParen abiadura (mm/s-tan) zuzenean aginduan zehazteko balio du argumentu honek. Ondoren, abiadura-datuetan zehazturik dagoen abiadurak ordezkatuko du.

[\T] Datu mota: *num*

Bai robota eta bai kanpo-ardatzak zenbat denboraz mugitu behar diren (segundotan) zehazteko balio du argumentu honek.

Zona Datu mota: *zonedata*

Mugimenduaren eremu-datuak dira, eta sorturiko ertzeko ibilbidearen tamaina deskribatzen dute.

[\Z] Datu mota: *num*

TCParen posizioaren doitasuna aginduan bertan zehazteko balio du argumentu honek. Ertzeko ibilbidearen luzera mm-tan ematen da, eta eremu-datuetan zehazturiko eremuak ordezkatuko du.

Herramienta Datu mota: *tooldata*

Robota mugitzean erabiltzen ari den erreminta da. TCPa zehazturiko helmuga-puntura joango da.

[\WObj]

Robotaren posizioa dela eta, aginduan adierazitako lan-objektua da (objektuaren koordenatu-sistema).

Argumentu hau bazter daiteke eta, hala izanez gero, posizioa koordenatu-sistema orokorrekoa izango da. Bestalde, TCP geldikorra edota kanpo-ardatz koordinatuak erabiliz gero, lan-objektu batekiko zirkulu bat exekutatu ahal izateko moduan zehaztu behar da argumentu hau.

Programaren exekuzioa

Robota eta kanpo-unitateak honela mugituko dira helmuga-puntura:

- Erremintaren TCPa biraka eta programatutako abiadura konstantean mugituko da.

- Erreminta abiadura konstantez berrorientatzen da, hasiera-puntuko orientaziotik amaiera-puntuko orientaziora.

- Berrorientazioa ibilbide zirkularrarekiko gauzatzen da. Hala, ibilbidearekiko orientazioa bera bada hasiera- eta amaiera-puntuetan, orientazioa konstante mantenduko da ibilbide osoan.

- Zirkulu-puntuan orientazioa ez da kritikoa; bi berrorientazio aukera bereizteko soilik balio du. Berrorientazioaren ibilbidearekiko doitasuna hasiera- eta amaiera-puntuetako orientazioaren menpe egongo da.

- Kanpo-ardatz koordinatu gabeak abiadura konstantez exekutatzen dira, amaiera-puntura robotaren ardatzekin batera iritsi daitezen. Zirkuluko posizioa ez da erabiltzen.

Berrorientazioak edota kanpo-ardatzek programaturiko abiadura ezin badute lortu, TCParen abiadura murriztu egingo da.

Mugimendua ibilbide baten hurrengo sekziora transferitzean, ertzeko ibilbidea sortzen da. Eremu-datuetan gelditze-punturen bat zehaztuz gero, programaren exekuzioak robota eta kanpo-ardatzak posizio egokira iritsitakoan soilik jarraituko du.

Adibideak

MoveC,*,v500\V:=550,z40\Z:=45,pinza3;*

pinza3 erremintaren TCPa aginduan gordetako posiziora joango da biraka. Mugimendua *v500* eta *z40* datuekin egingo da; TCParen eremu-neurria eta abiadura *40 mm* eta *550 mm/s*-koak dira, hurrenez hurren.

MoveC\Conc,,*,v500,z40,pinza3;*

pinza3 erremintaren TCPa aginduan gorderiko posiziora joango da biraka. Zirkulu-puntua ere aginduan gordea dago. Ondorengo agindu logikoak robota mugitzen den heinean beteko dira.

MoveC cir1,p15,v500,z40,pinza3\WObj:=fijación;

pinza3 erremintaren TCPa *p15* posiziora joango da biraka, *cir1* zirkulu-puntutik igarota. Posizio hauek objektuaren koordenatu-sisteman *fijación*-en zehaztuak daude.

Mugak

Robotaren ibilbide birakarian geldirik dagoen bitartean ez da onartuko exekuzio-moduaren aldaketarik, aurretik atzera nahiz alderantziz, eta, ondorioz, akats-mezua sortuko da.

MoveC agindua (edota mugimendu birakaria barneratzen duen beste edozein agindu) inoiz ez da hasieratik abiaraziko TCPa zirkulu-puntuaren eta amaierako puntuaren artean dela. Bestela, robotak ez dio programaturiko ibilbideari ekingo.

Robotak exekuzio-denboran zirkulu-puntua irits dezakeela ziurtatu behar da, eta zirkulua zatitu, hala behar izanez gero.

Sintaxia

MoveC
[´\´Conc´,´]
[PuntoCírculo´:=´]<expresión(IN) de robtarget>´,´
[AlPunto´:=´]<expresión (IN) de robtarget>´,´
[Velocidad´:=´]<expresión (IN) de speeddata>
[´\´V´:=´<expresión (IN) de num>]
[´\´T´:=´<expresión (IN) de num>]´,´
[Zona´:=´]<expresión (IN) de zonedata>
[´\´Z´:=´<expresión (IN) de num>]´,´
[Herramienta´:=´]<persistente (PERS) de tooldata>
[´\´WObj´:=´<persistente (PERS) de wobjdata>]´;´

2.3.2. ROBOTAREN ARDATZEZ ARDATZEKO MUGIMENDUA

MoveJ agindua

Mugimenduak lerro zuzeneko ibilbiderik egin behar ez duenean, robota puntu batetik bestera azkar mugitzeko balio du agindu honek.

Robota eta kanpo-ardatzak zuzena ez den ibilbideari jarraituz joango dira amaiera-posiziora. Ardatz guztiak aldi berean iritsiko dira helmuga-posiziora.

Adibideak

MoveJ p1,vmax,z30,herram2;

herram2 erremintaren erdiko puntua (TCP) zuzena ez den ibilbidea eginez mugituko da *p1* posiziora iritsi arte, *vmax* abiadura-datuarekin eta *z30* eremu-datuarekin.

*MoveJ *, vmax \T:=5,fine,pinza3;*

pinza3 erremintaren erdiko puntua zuzena ez den ibilbidean mugituko da, aginduan gordetako geldiune-puntura iritsi arte. Mugimendu osoa gauzatzeko, 5 segundo behar dira.

Argumentuak

[\Conc] Datu mota: *switch*

Argumentu honen ondorengo agindu logikoak robota mugitzen ari den bitartean gauzatuko dira. Argumentu honek zikloaren denbora laburtzeko balio du; adibidez, periferikoren batekin komunikatzen ari den eta inongo sinkronizaziorik behar ez den kasuetan.

\Conc argumentuaren arabera, gehienez, bost agindu erabil daitezke jarraian. *StorePath-RestorePath* erabiltzen den programa ataletan, ez dago *\Conc* argumentua erabiltzerik.

Argumentu hau baztertuz gero, hurrengo agindua ez da abiatuko robota eremu zehatzera iritsi arte.

Alpunto Datu mota: *robtarget*

Robotaren eta kanpo-ardatzen helmuga-puntua da. Posizio ezagun moduan definitu behar da, edota zuzenean aginduan gorde.

Velocidad Datu mota: *speeddata*

Mugimenduei aplikatzen zaizkien abiadura-datuak dira, eta TCParen abiadura eta kanpo-ardatzen nahiz erremintaren orientazioa definitzen dituzte.

[\V] Datu mota: *num*

TCParen abiadura aginduan bertan, mm/s-tan, zehazteko balio du argumentu honek.

[\T] Datu mota: *num*

Robota zenbat denboraz mugitzen den (segundotan) zehazteko balio du argumentu honek. Ondoren, dagozkion abiadura-datuek balioa egokituko dute.

Zona Datu mota: *zonedata*

Mugimendurako eremu-datuak dira, eta eginiko ertz-ibilbidearen neurria deskribatzen dute.

[\Z] Datu mota: *num*

TCParen posizio-doitasuna aginduan zuzenean zehazteko balio du argumentu honek. Ertz-ibilbidearen luzera mm-tan ematen da, eta eremu-datuetan zehazturiko datua ordezkatuko du.

Herramienta Datu mota: *tooldata*

Robotak mugitzen ari den bitartean zer erreminta duen adierazten du.

[\WObj] Datu mota: *wobjdata*

Robotaren posizioak aginduan zer lan-objektu erabiltzen duen adierazten du.

Argumentu hau baztertu daiteke eta, hala eginez gero, posizioa koordenatu-sistema orokorraren arabera zehaztua egongo da. Bestalde, TCP geldikorren bat edota kanpo-ardatz koordinatuak erabiliz gero, argumentu honek zehaztua egon behar du.

Programaren exekuzioa

Erremintaren TCPa ardatzen angeluen interpolazioz joango da helmuga-puntura. Horrek esan nahi du ardatz bakoitza abiadura konstantez mugituko dela, eta ardatz guztiak aldi berean iritsiko direla helmuga-puntura. Horrek ibilbide ez zuzena sortuko du.

Orokorrean, TCPa programaturiko abiaduraz mugituko da. Mugitzen den bitartean, erreminta orientatu eta kanpo-ardatzak mugituko dira. Kanpo-ardatzen orientazioak programaturiko abiadurara iritsi ezin duen kasuetan, TCParen abiadura murriztuko da.

Mugimendua ibilbide baten hurrengo zatira bidaltzen denean, ertz-ibilbidea edo ertzeko ibilbidea sortzen da. Eremu-datuetan gelditze-puntua zehazten bada, programaren exekuzioak robota eta kanpo-ardatzak posizio egokietara iritsitakoan soilik jarraituko du.

Adibideak

MoveJ,v2000\V:=2200,z40\Z:=45,pinza3;*

> *pinza3* erremintaren TCPak zuzena ez den ibilbidea egingo du aginduan jasotako posiziora iristeko. Mugimendu hori *v2000* eta *z40* datuekin emango da; TCP-eremuaren abiadura eta neurria 2.200 mm/s eta 45 mm dira, hurrenez hurren.

MoveJ\Conc,,v2000,z40,pinza3;*

> *pinza3* erremintaren TCPak zuzena ez den ibilbidea egingo du aginduan jasotako posiziora iristeko. Ondorengo agindu logikoak robota mugitzen den bitartean beteko dira.

MoveJ arranque,v2000,z40,pinza3\WObj:=fijación;

> *pinza3* erremintaren TCPak zuzena ez den ibilbidea egingo du, *arranque* posiziora iritsi arte. Posizio hori objektuaren koordenatu-sisteman zehaztua dago, *fijacion*-en.

Sintaxia

```
MoveJ
        [´\´Conc´,´]
        [Alpunto´:=´]<expresión (IN) de robtarget>´,´
        [Velocidad ´:=´]<expresión (IN) de speeddata>
                [´\´V´:=´<expresión de (IN) de num>]
                [´\´T´:=´<expresión de (IN) de num>];
        [Zona´:=´]<expresión (IN) de zonedata>
                [´\´Z´:=´<expresión (IN) de num>]´,´
        [Herramienta´:=´]<persistente (PERS) de tooldata >
        [´\´WObj´:=´<persistente (PERS) de wobjdata>];
```

2.3.3. ROBOTAREN MUGIMENDU LINEALA

MoveL agindua

TCPa ibilbide zuzenarekin helmuga-puntu zehatzera mugitzeko balio du. TCPak geldirik egon behar duenean ere agindu hau erabil daiteke erreminta orientatzeko.

Adibideak

MoveL p1,v1000,z30,herram2;

herram2 erremintaren TCPak ibilbide zuzena egingo du *p1* posizioraino *v1000* abiadurarekin eta *z30* eremu-datuarekin.

MoveL,v1000\T:=5,fine,pinza3;*

pinza3 erremintaren TCPak ibilbide zuzena egingo du aginduan jasotako punturaino. Mugimendua osatzeko, bost segundo beharko ditu.

Argumentuak

[\Conc] Datu mota: *switch*

Argumentu honen ondorengo agindu logikoak robota mugitzen ari den bitartean gauzatuko dira. Argumentu honek ziklo-denbora laburtzeko balio du, adibidez, periferikoren batekin komunikatzen ari den eta inongo sinkronizaziorik behar ez den kasuetan.

Conc argumentuaren bidez, gehienez, bost mugimendu-agindu erabil daitezke jarraian. *StorePath-RestorePath* erabiltzen den programa ataletan ez da onartzen *Conc* argumentua erabiltzea.

Argumentu hau baztertuz gero, hurrengo agindua ez da abiatuko robota posizio zehatzera iritsi arte.

Alpunto Datu mota: *robtarget*

Robotaren eta kanpo-ardatzen helmuga puntua da. Posizio ezagun moduan definitua egon behar du edota zuzenean aginduan gordea.

Velocidad Datu mota: *speeddata*

Mugimenduei aplikatzen zaizkien abiadura-datuak dira, eta TCParen abiadura eta kanpo-ardatzen nahiz erremintaren orientazioa definitzen dituzte.

[\V] Datu mota: *num*

TCParen abiadura aginduan bertan, mm/s-tan, zehazteko balio du argumentu honek.

[\T] Datu mota: *num*

Robota zenbat denboraz mugitzen den (segundotan) zehazteko balio du argumentu honek. Ondoren, dagozkion abiadura-datuetara itzuliko da.

Zona Datu mota: *zonedata*

Mugimendurako eremu-datuak dira, eta eginiko ertz-ibilbidearen neurria deskribatzen dute.

[\Z] Datu mota: *num*

TCParen posizioaren doitasuna aginduan zuzenean zehazteko balio du argumentu honek. Ertz-ibilbidearen luzera mm-tan ematen da, eta eremu-datuetan zehazturiko eremuak ordezkatuko du.

Herramienta Datu mota: *tooldata*

Robotak mugitzen ari den bitartean zer erreminta duen adierazten du.

[\WObj] Datu mota: *wobjdata*

Robotaren posizioak aginduan zer lan-objektu erabiltzen duen adierazten du.

Argumentu hau baztertu daiteke eta, hala eginez gero, posizioa koordenatu-sistema orokorrari zuzendurik egongo da. Bestalde, TCP geldikorren bat edota kanpo-ardatz koordinatuak erabiliz gero, argumentu honek zehaztua egon behar du.

Programaren exekuzioa

Bai robota eta bai unitate periferikoak helmuga-puntura honela mugituko dira:

- Erremintaren TCPa zuzen eta programatutako abiadura konstantez mugituko da.

- Erreminta, denbora-tarte jakin eta berdinetan, ibilbidearen arabera orientatzen da.

- Kanpo-ardatz koordinatu gabeak.

Kanpo-ardatzek edota orientazio-mugimenduak programatutako abiadura lortu ezin duten kasuetan, TCParen abiadura murriztu egingo da.

Mugimendua ibilbide baten hurrengo zatira bidaltzen denean, ertz-ibilbidea edo ertzeko ibilbidea sortzen da. Eremu-datuetan gelditze-puntua zehazten bada, programaren exekuzioak robota eta kanpo-ardatzak posizio egokietara iritsitakoan soilik jarraituko du.

Adibideak

*MoveL *,v2000\V:=2200,z40\Z:=45,pinza3;*

pinza3 erremintaren TCPak ibilbide zuzena egingo du aginduan jasotako posiziora iristeko. Mugimendu hori *v2000* eta *z40* datuekin emango da; TCP-eremuaren abiadura eta neurria 2.200 mm/s eta 45 mm dira, hurrenez hurren.

MoveL\Conc,,v2000,z40,pinza3;*

pinza3 erremintaren TCPak ibilbide zuzena egingo du aginduan jasotako posiziora iristeko. Ondorengo agindu logikoak robota mugitzen den bitartean beteko dira.

MoveL arranque,v2000,z40,pinza3\WObj:=fijación;

pinza3 erremintaren TCPak ibilbide zuzena egingo du *arranque* posiziora iritsi arte. Posizio hori objektuaren koordenatu-sisteman zehaztua dago, *fijacion*-en.

Sintaxia

MoveL
[´\´Conc´,´]
[Alpunto´:=´]<expresión (IN) de robtarget>´,´
[Velocidad ´:=´]<expresión (IN) de speeddata>
[´\´V´:=´<expresión de (IN) de num>]
[´\´T´:=´<expresión de (IN) de num>];
[Zona´:=´]<expresión (IN) de zonedata>
[´\´Z´:=´<expresión (IN) de num>]´,´
[Herramienta´:=´]<persistente (PERS) de tooldata >
[´\´WObj´:=´<persistente (PERS) de wobjdata>]

2.3.4. IRTEERA-SEINALE DIGITALA AKTIBATZEA

Set agindua

iIrteera-seinale digitalari bat balioa emateko balio duen agindua da.

Adibideak

Set do 15;

do15 seinaleak bat balioa hartuko du.

Set sold;

sold seinaleak bat balioa hartuko du.

Argumentuak

Set *Señal*

Señal Seinale mota: *signaldo*

Bat balioa eman nahi zaion seinalearen izena sartu behar da.

Programaren exekuzioa

Benetako balioa seinalearen konfigurazioaren menpe egongo da. Seinalearen balioa aldatu bada sistemaren parametroetan, aginduak kanal fisikoari zero balioa emango dio.

Sintaxia

Set

[\Señal´:=´]<variable (VAR) de signaldo>´;´

2.3.5. IRTEERA-SEINALE DIGITALA DESAKTIBATZEA

Reset agindua

iIrteera-seinale digitalari zero balioa emateko balio du.

Adibideak

Reset do 15;

do15 seinaleak zero balioa hartuko du.

Reset sold;

sold seinaleak zero balioa hartuko du.

Argumentuak

Reset Señal

Señal Seinale mota: *signaldo*

Zero balioa eman nahi zaion seinalearen izena da.

Programaren exekuzioa

Benetako balioa seinalearen konfigurazioaren menpe egongo da. Seinalearen balioa aldatu bada sistemaren parametroetan, aginduak kanal fisikoari bat balioa emango dio.

Sintaxia

Reset

[\Señal´:=´]<variable (VAR) de signaldo>´;´

2.3.6. SARRERA-SEINALE DIGITALARI ITXAROTEA

WaitDI agindua

Sarrera-seinale digitalen bat aktibatu arte itxaroteko balio du.

Adibideak

WaitDI di4, 1;

Programaren exekuzioak *di4* sarrera aktibatu ondoren bakarrik jarraituko du.

WaitDI grip_status,0;

Programaren exekuzioak *grip_status* sarrera abiatu ondoren bakarrik jarraituko du.

Argumentuak

Señal Datu mota: *signaldi*

Seinalearen izena da.

Valor Datu mota: *dionum*

Seinaleari eman nahi zaion balioa da.

[\Max Time] Datu mota: *num*

Gehienez onartzen den itxarote denbora da, segundotan adierazita. Denbora-tarte hori baldintza bete aurretik igaro bada eta *ERR_WAIT_MAXTIME* kodedun akatsik baldin badago, sistemak akats-kudeatzaileari dei egingo dio. Sistemak akats-kudeatzailerik ez badu, robotaren exekuzioa gelditu egingo da.

[\TimeFlag] Datu mota: *bool*

Baldintza bete baino lehen gehienez onartutako denbora-tartea igaro denean, *TRUE* balioa duen irteera-parametroa da. Parametro hori aginduan barneraturik badago, ez da akastzat hartuko gehienezko itxarote-denbora igaro den kasuetan. Argumentu hau baztertu egingo da aginduan *Maxtime* argumentua ageri ez bada.

Programaren exekuzioa

Agindua exekutatzeko garaian seinalearen balioa zuzena bada, programak hurrengo aginduarekin jarraituko du, besterik gabe.

Seinalearen balioa okerra bada, robota itxarote-egoeran sartuko da, eta seinaleak balio zuzena hartzean, aurrera jarraituko du. Erantzun azkarra eskaintzen duen etenagatik detektatzen da aldaketa.

Robota itxaroten ari denean denbora kudeatua dago, eta gehienezko denbora-tartea gainditzen den kasuetan, programak aurrera jarraituko du, baldin eta *TimeFlag* argumentua zehaztu bada.

Argumentua zehaztu ez bada, akats-seinalea igorriko du. *TimeFlag* zehazten bada, *TRUE* eran abiatuko da, denbora-tartea gainditzean; bestela, berriz, *FALSE* eran.

Sintaxia

WaitDI
[Señal´:=´]<variable (VAR) de signaldi>´;´
[Valor ´:=´]<expresión (IN) de dionun>´;´
[´\´MaxTime´;´<expresión (IN) de num>]
[´\´TimeFlag´:=´<variable (VAR) de bool>]´;´

2.3.7. IRTEERA-SEINALE DIGITALARI ITXAROTEA

WaitDO agindua

iIrteera-seinale digitalen bat aktibatu arte itxaroteko balio du.

Adibideak

WaitDO do4, 1;

Programaren exekuzioak *do4* irteera aktibatu ondoren bakarrik jarraituko du.

WaitDO grip_status,0;

Programaren exekuzioak *grip_status* irteera abiatu ondoren bakarrik jarraituko du.

Argumentuak

Señal Datu mota: *signaldi*

Seinalearen izena da.

Valor Datu mota: *dionum*

Seinaleari eman nahi zaion balioa da.

[\Max Time] Datu mota: *num*

Gehienez onartutako itxarote denbora da, segundotan adierazita. Denbora-tarte hori baldintza bete aurretik igaro bada eta *ERR_WAIT_MAXTIME* kodedun akatsik baldin badago, sistemak akats-kudeatzaileari dei egingo dio. Sistemak akats-kudeatzailerik ez badu, robotaren exekuzioa gelditu egingo da.

[\TimeFlag] Datu mota: *bool*

Baldintza bete baino lehen gehienez onartutako denbora-tartea igaro denean, TRUE balioa duen irteera-parametroa da. Parametro hori aginduan barneraturik badago, ez da akastzat hartuko gehienezko itxarote-denbora igaro den kasuetan. Argumentu hau baztertu egingo da aginduan *Maxtime* argumentua ageri ez bada.

Programaren exekuzioa

Agindua exekutatzeko garaian seinalearen balioa zuzena bada, programak hurrengo agin-uarekin jarraituko du, besterik gabe.

Seinalearen balioa okerra bada, robota itxarote-egoeran sartuko da, eta seinaleak balio zuzena hartzean, aurrera jarraituko du. Erantzun azkarra eskaintzen duen etenagatik detektatzen da aldaketa.

Robota itxaroten ari denean denbora kudeatua dago, eta gehienezko denbora-tartea gainditzen den kasuetan, programak aurrera jarraituko du, baldin eta *TimeFlag* argumentua zehaztu bada. Argumentua zehaztu ez bada, akats-seinalea igorriko du. *TimeFlag* zehazten bada, *TRUE* eran abiatuko da denbora-tartea gainditzean; bestela, berriz, *FALSE* eran.

Sintaxia

WaitDO
[Señal´:=´]<variable (VAR) de signaldi>´;´
[Valor ´:=´]<expresión (IN) de dionun>´;´
[´\´MaxTime´;´<expresión (IN) de num>]
[´\´TimeFlag´:=´<variable (VAR) de bool>]´;´

2.3.8. ERREPIKATU AGINDUA… BITARTEAN

WHILE agindua

Baldintza jakin bat betetzen den denbora-tartean hainbat agindu errepikatzea nahi den kasue-tarako erabiltzen da.

Aurrez errepikaldi kopurua ezagutzen den kasuetan *FOR* agindua erabiliko da.

Adibidea

WHILE reg1<reg2 DO
…
reg1:=reg1+1;
ENDWHILE

WHILE begiztako aginduak errepikatuko ditu *reg1<reg2* den bitartean.

Argumentuak

WHILE Baldintza *DO…* *ENDWHILE*

Baldintza Datu mota: *bool*

Baldintza bete egin behar da, *WHILE* begiztako aginduak bete ahal izateko.

Programaren exekuzioa

1. Baldintza kalkulatzen da. Baldintza ez bada bete, *WHILE* begizta amaitu egingo da, eta programaren exekuzioak *ENDWHILE*ren ondoren datorren aginduarekin jarraituko du.

2. *WHILE begiztako aginduak exekutatzen dira.*

3. WHILE begizta errepikatuko da 1. puntutik hasita.

Sintaxia

(EBNF)
WHILE<baldintzazko adierazpena> DO
ENDWHILE

2.3.9. BALDINTZA BETEZ GERO… AGINDUA

Compact IF agindua

Agindua baldintza zehatza betez gero bakarrik gauzatu nahi denerako balio du.

Baldintza zehatza betetzen den edo ez, agindu bat baino gehiago exekutatu nahi izanez gero, IF agindua erabiliko da.

Adibideak

IF reg1>5 GOTO sig;

reg1>5 bada, programaren exekuzioak *sig* etiketari jarraituko dio.

IF contador>10 Set do1;

contador>10 bada, *do1* seinalea aktibatuko da.

Argumentuak

IF baldintza…

Baldintza *Datu mota: bool*

Agindua gauzatu ahal izateko bete beharreko baldintza da.

Sintaxia

(EBNF)
IF <baldintzazko adierazpena>(<agindua>?<SMT>)´;´

2.3.10. BALDINTZA BETEZ GERO... ; BESTELA…

IF agindua

Baldintzaren araberako aginduak betetzea nahi denerako balio du.

Adibideak

```
IF reg1>5 then
        Set do1;
        Set do2;
ENDIF
```

do1 eta do2 seinaleak reg1 5 baino handiagoa bada soilik aktibatuko dira.

```
IF reg1>5 THEN
        Set do1;
        Set do2;
ELSE
        Reset do1;
```

```
            Reset do2;
    ENDIF
```

Do1 eta do2 seinaleak aktibatu ala desaktibatu egingo dira reg1 5 baino handiago ala txikiago izatearen arabera.

Argumentuak

iIF Baldintza THEN…
{*ELSEIF* Baldintza *THEN*…}
[ELSE…]
ENDIF

Baldintza Datu mota: *bool*

THEN eta *ELSE/ELSEIF* tartean dauden baldintzak exekutatzeko bete beharreko baldintza.

Adibidea

```
IF contador>100 THEN
        contador:=100;
ELSEIF contador<0 THEN
        contador:=0;
ELSE
        contador:=contador+1;
ENDIF
```

contador-i bat gehitu zaio. Hala ere, contador-en balioa *[0-100]* tartetik kanpo badago, *contador*-ek dagokion muga-balioa izendatuta izango du.

Programaren exekuzioa

Baldintzak agindu sekuentzial bati jarraituz egiaztatzen dira, haietako bat bete arte. Programaren exekuzioak aurrera jarraituko du baldintza horri lotutako aginduekin. Baldintzarik betetzen ez bada, programaren exekuzioak *ELSE*-ri jarraitzen dioten aginduekin segituko du. Baldintza bat baino gehiago betez gero, lehen baldintzari lotutako aginduak bakarrik gauzatuko dira.

Sintaxia

(EBNF)
IF<baldintzazko adierazpena> *THEN*
<agindu zerrezda>
{ELSEIF<baldintzazko adierazpena> *THEN* <agindu zerrenda> *? <EIF>}*

[ELSE

<agindu zerrenda>]
ENDIF

2.3.11. BAT GEHITZEA

Incr agindua

Agindu honek aldagai bati bat gehitzeko balio du.

Adibidea

Incr reg1;

reg1-i bat gehituko zaio, hau da, *reg1:=reg1+1.*

Argumentuak

Incr Nombre Datu-mota*: num*

Nombre

Aldatu nahi den aldagaiaren izena da.

Adibidea

```
WHILE DInput(stop_produc)=0 DO
        Product_part;
        Incr_parts;
        TPWrite "Ekoizturiko pieza kopurua = "\Num:=parts;
ENDWHILE
```

Ziklo bakoitzean ekoizturiko pieza kopurua programazio-unitatean eguneratuko da. Ekoizte-prozesuak jarraitu egingo du stop_production seinalea aktibatzen ez den bitartean.

Sintaxia

Incr
[Nombre´=´]<aldagaia (INOUT) num]´;´

2.3.12. HAINBAT ALDIZ ERREPIKATZEA

FOR agindua

Agindu bat edo gehiago guk esandako alditan errepikatzea nahi denean erabiltzen da FOR agindua.

Aginduak baldintza jakin baten arabera errepikatu behar diren kasuetarako, WHILE agindua erabiltzen da.

Adibidea

```
FOR i FROM 1 TO 10 DO
        Rutina 1;
ENDFOR
```

rutina1 prozedura 10 aldiz errepikatuko du.

Argumentuak

Contador bucle Identifikatzailea

Uneko begizta-zenbatzailearen balioa duen datuaren izena. Datua automatikoki ezarriko da, eta ezingo du izan aurrez beste daturen batek zuen izen bera.

Valor inicial *Datu mota: num*

Begizta-zenbatzaileari eman nahi zaion hasierako balioa da.

Valor final *Datu mota: num*

Begizta-zenbatzaileari eman nahi zaion amaierako balioa da.

Incremento *Datu mota: num*

Begizta-zenbatzaileari gehitu edo gutxituko zaion balioa da. Balioa zehaztu gabe badago, gehitzea edo gutxitzea batekoa izango da automatikoki.

Adibidea

```
FOR i FROM 10 TO 2 STEP -1 DO
        A{i}:=a{i-1};
```

ENDFOR

Matrizearen balioak hazkuntza moduan doitzen dira, honela: {10}:=a{9},:=a{8}, etab.

Programaren exekuzioa

1. Hasierako, amaierako eta gehikuntzako balioei dagozkien adierazpenak kalkulatuko dira.

2. Begizta-zenbatzaileak hasierako balioa izango du.

3. Begizta-zenbatzailearen balioa egiaztatuko da hasierako eta amaierako balioen artean dagoen edota horietako baten berdina den egiaztatzeko.

4. *FOR* begiztako aginduak exekutatuko dira.

5. Begizta-zenbatzailea gehikuntza balioaren arabera gehitu edo gutxituko da.

6. *FOR* begizta errepikatuko da 3. puntutik hasita.

Mugak

Begizta-zenbatzailera *FOR* begiztatik bakarrik irits daiteke, eta, horren ondorioz, izen berdina duten gainerako datu eta errutinak ezkutatu egingo dira.

Hasiera-, amaiera- eta gehikuntza-datuek ez dute zifra hamartarrik duen baliorik onartzen.

Sintaxia

(EBNF)

```
FOR<begizta-aldagaia>FROM<adierazpena>TO<adierazpena>
        [STEP<adierazpena>]DO
        <agindu-zerrenda>
ENDFOR
<begizta-aldagaia>:=<identifikatzailea>
```

Bibliografia

* ABB: Robot Based Automation (www.abb.com/robots)
* ABB robotaren eskuliburua