

# ROBOT

A mérésnek több célja van, egyebek között nagy vonalakban megismerkedni a robotok kialakulásával, elvi működésével és néhány egyszerűbb célfeladaton keresztül a gyakorlati programozásukkal.

A probléma már ott kezdődik, hogy tulajdonképpen mit is kell érteni a robot kifejezésen. Vannak, akik nem teljesen helyesen - mint látni fogjuk a későbbiekben - automatáknak nevezik ezeket. Természetesen az embernek nem kell feltétlenül ragaszkodnia a kifogástalanul pontos meghatározáshoz, hiszen e nélkül is legtöbbször el tudják képzelni a robotok lényegét, még ha esetleg nem is képesek azt egzakt módon szavakba foglalni.

Két különböző módon közelíthetünk a kérdéshez, egyrészt a működésének bonyolultsági foka, másrészt, hogy külsőségeiben mennyire hasonló (antropomorf) az emberhez, amelyhez persze az emberi típusú (humanoid) mozgásforma és szolgáltatás is tartozhat. Ez utóbbiak egyelőre inkább csak - bár nem elhanyagolható tudományos újdonságokkal rendelkeznek - a látványosságok kategóriájába tartoznak. Ilyenek például a zenélő, a felszolgáló stb. robotok.

Az első típusúak sokkal hasznosabban „tevékenykednek”, ezekből ezért is van ma már több millió működő darab a világban, különösen a műszaki színvonalban élenjáró országokban például ilyen robotok végzik a gépkocsik (és egyéb gépek) gyártásában a hegesztési és szerelési munkák egy jelentős részét. Jellemző rájuk, hogy ezeket célfeladatokra fejlesztették ki, így nem meglepő, hogy általában nem ember formájúak, legfeljebb bizonyos részeik a kézre hasonlítanak.

A működési elvet tekintve természetesen igen alapvető különbségek vannak az egy-két előre rögzített célmozgást kontroll nélkül végzők és az - általunk valóban automata robotnak tekintett - önmaga által érzékelt környezeti változásokra azonnal reagáló ún. intelligens szerkezetek között. Valójában ezt még fokozhatjuk is, például szövegfelismerő és szintetikus beszédhang előállító stb. stb. egységek hozzáadásával.

Átgondolva a ma és a közeljövő lehetőségeit is, az nem lehet a fő célunk, hogy mesterséges embert (gólem, humanoid, kiborg, android stb.) állítsunk elő, hiszen az ember „előállítására” már van egy elég régen sikeresen működő módszerünk, viszont a nehéz, a veszélyes, valamint az egyszerű de monoton munkákra hasznos és érdemes kifejlesztenünk robotokat, ahol az emberi intelligenciára, találékonyságra nincs feltétlenül szükség. Ezt bizonyítja az ipari robotok nagyarányú elterjedése is. Részletesebb leírást a függelékben találhatjuk meg. E fejezetet elolvasván megismerhetjük, pl. az ún. ipari robotok definícióját, a vezérlési módokat, a programozási nyelveket stb.

## A ROB3 RENDSZER

A ROB3 egy ipari robot modellje (ún. laboratóriumi tanrobot), hat szabadsági fokkal (5 forgatható tengellyel és egy fogó karral). Az egész szerkezet egy 80 x 60 cm-es lap felett van rögzítve.

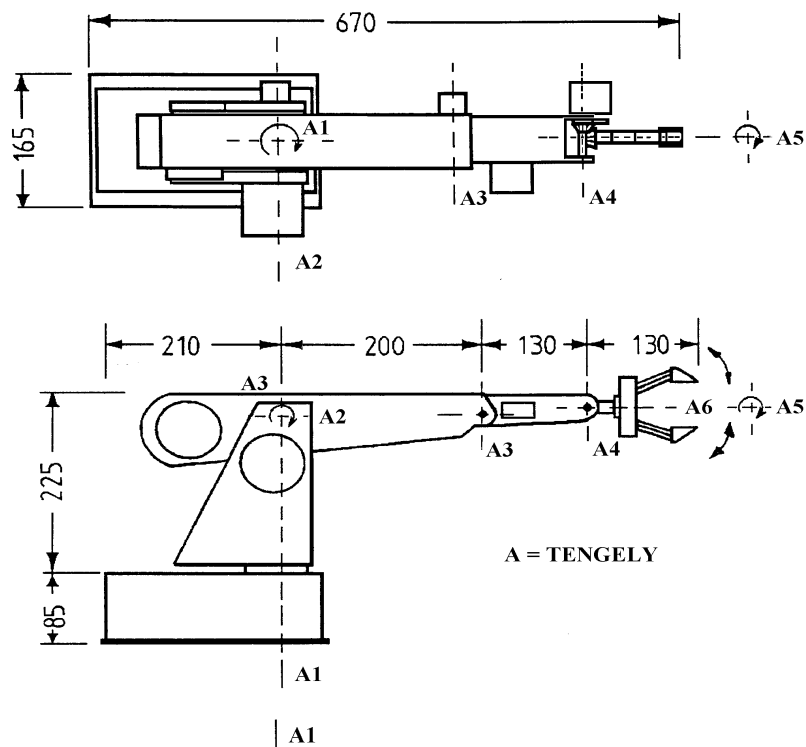
A robot három fő részből áll:

- Manipulátor (karok, motorok, áttételek, szenzorok, helyzetérzékelők)
- Tápegység (a motorok, és a vezérlő logika tápellátása)
- Vezérlés (a speciális interface, a számítógép, vagy az ún. "teach-box",)

## 1. A manipulátor

Ez a robot tulajdonképpen mechanikusan mozgó, végrehajtó része, amelyet fogasáttétellel és fogas szíjjal hat egyenáramú szervomotor hajt meg. Az első, a robottestet (törzset) forgatja (body rotation), a második az un. vállizületet (shoulder joint), a harmadik a könyökizületet (elbow joint), a negyedik a csuklóizületet mozgatja (wrist raising & lowering), az ötödik a fogókart - kezét - forgatja (wrist rotation), végül a hatodik motor bowden-huzalok és rugók segítségével a nem cserélhető kétujjas fogót nyitja-zárja rugalmasan (gripper opening & closing). A tengelyek szimultán mozgathatók.

A robot fontosabb méretei és mozgástengelyei a 1. ábrán láthatók.



1. ábra.

A robot munkatere (working space) nagyjából egy 450 mm sugarú negyedgömb. A berendezés csak e térben tud műveleteket végezni. Ebben a munkatérben - elvileg, ha nem lenne hiba -  $256^5$  különböző térbeli pont jelölhető ki és érhető el! (Azért nem  $256^6$ , mert a hatodik szabadsági fok nem ad újabb térbeli helyzetet, csak a fogó csukását, nyitását engedélyezi). Mivel az ismétlési pontosság terheletlen állapotban kb.  $\pm 2$  mm, viszont a pozícionáló pontosság a terheléstől és az ízületek (tengelyek) helyzetétől függően  $\pm 1$  mm és  $\pm 4$  mm között van, ezért a térbeli felbontás jóval kisebb az előbb leírt  $256^5$  értéknél is. A pontatlanság értelemszerűen a szögekre is érvényes. Átszámolva a hibát, ez maximálisan  $\pm 3^\circ$  közötti értéket vehet fel.

A pozíciómérő-rendszer az egyes tengelyek abszolút helyzetét méri  $256 (2^8)$  osztású precíziós potenciométeres elfordulásjelzők segítségével. A ROB3 vezérlő elektronikája így még áramkimaradás után is képes felismerni az összes tengely aktuális helyzetét. A  $2^8$  felosztásból következik, hogy a legfinomabb vezérlési és helyzetbeállási lépésköz a teljes mozgási tartomány 256-od része, (az un. STEP NUMBER  $\Rightarrow S_i$  maximális értéke = 256).

A hat szabadsági fok  $S_1, S_2, S_3, S_4, S_5,$  és  $S_6$  felveheti a 0 és 255 közötti értéket ( $0 \leq S_i \leq 255$ ), és a hozzájuk tartozó szögbeosztások az alábbiak:

$$1.- 80^\circ \leq \alpha_1 \leq +80^\circ, \quad \text{a szögtartomány} \approx 160^\circ \\ 255 \geq S_1 \geq 0, \quad \text{egy lépés} \approx 0.63^\circ$$

$$2.- 45^\circ \leq \alpha_2 \leq +56^\circ, \quad \text{a szögtartomány} \approx 101^\circ \\ 255 \geq S_2 \geq 0, \quad \text{egy lépés} \approx 0.39^\circ$$

$$3. +75^\circ \leq \alpha_3 \leq +172^\circ, \quad \text{a szögtartomány} \approx 97^\circ \\ 255 \geq S_3 \geq 0, \quad \text{egy lépés} \approx 0.38^\circ$$

$$4. -92^\circ \leq \alpha_4 \leq +92^\circ, \quad \text{a szögtartomány} \approx 184^\circ \\ 255 \geq S_4 \geq 0, \quad \text{egy lépés} \approx 0.72^\circ$$

$$5. -92^\circ \leq \alpha_5 \leq +92^\circ, \quad \text{a szögtartomány} \approx 184^\circ \\ 255 \geq S_5 \geq 0, \quad \text{egy lépés} \approx 0.72^\circ$$

6.-ik a kétujjas fogó (kéz), amelynél, ha a teljesen nyitott ( $S_6 = 0$ ), akkor a pofák 60 mm-re vannak egymástól, amikor teljesen zárt ( $S_6 = 255$ ), akkor viszont természetesen 0 mm-re csökken a távolság. Előző öt esethez képest, ahol a szögtartományt osztottuk, itt ezt a 60 mm-t kellene elosztanunk 256-tal, hogy megkapjuk az egy lépésköz értékét, de a pofa már előbb becsukódik (nevezetesen  $S_6 \approx 240$ -nél), így itt egy lépésköz kb. 0.25 mm-nek felel meg. Mivel a fogókar rugalmas ezért, amikor a nyitott állásból elkezdünk zárt állás felé vezérelni ( $S_6 = 0 \rightarrow 255$ ), akkor - ha tárgy van a pofák között - a " kéz " szorító ereje fog növekedni.

Az alább következő fejezetekben - megelőzve a részletesebb magyarázatot - többször előfordulnak, olyan nagybetűs robot-programozási kifejezések, amelyek a ROB3-at vezérlő ún. " ROLAN " (Robot Language) nyelv utasítás készletéhez tartoznak. Ezeket a vezérlő és egyéb kifejezéseket a programozással foglalkozó fejezetekben magyarázzuk meg részletesen.

Más robotoktól eltérően, amelyeknél a kiinduló helyzet általában mechanikusan definiált, a ROB3 startpozíciója a programban meghatározható, így a robot akkor is használható, ha a kiinduló helyzetben akadály lenne. Esetünkben a bekapcsolási alapállás az ún. FIX 0-ban leírt paramétereknek felel meg. Ez a már előbbieken többször is említett beállási és visszaállási pontatlanságok miatt, - néhány mm illetve néhány fok eltéréssel (**jelenleg**, amíg valaki át nem programozza) - az alább értékeket jelenti:

$$S_1 = 128 \quad (\alpha_1 \approx 0^\circ)$$

$$S_2 = 141 \quad (\alpha_2 \approx 0^\circ)$$

$$S_3 = 222 \quad (\alpha_3 \approx 90^\circ)$$

$$S_4 = 126 \quad (\alpha_4 \approx 0^\circ)$$

$$S_5 = 129 \quad (\alpha_5 \approx 0^\circ)$$

$$S_6 = 240$$

A programozásnál láthatjuk majd a későbbiekben, hogy mi magunk tudunk definiálni bármilyen (természetesen az értelmezési tartományon belül) FIX X függvényt a SETF X utasítást segítségével.

Van két olyan angol rövidítés, amely sokszor előfordul majd a leírásban, nevezetesen, a:

**TCP** (Tool Center Point), amely általában a befogó szerkezet (kéz) geometriai közepének koordinátája, és a

**PTP** (Point To Point), amely olyan vezérlési módot jelent, amikor a TCP mozgását a kiinduló pontból a végső pontba egyetlen utasítással hajtja végre.

A PTP típusú mozgatsnál a maximális terhelés kb. 250 g lehet, míg az un. sok-pontos vezérlés (**MPC**) esetén, tehát amikor tengelyenként csak egy pozicionáló lépésköz állítható be, akkor maximálisan 50 g. Nagyobb terhek esetén előfordulhat, hogy a vezérlő elektronika - különösen az alsó és felső kar emelő irányú mozgatsakor - nem tudja a kívánt pozíciót beállítani.

## 2. A tápegység

A robotban - amint már tudjuk - hat darab egyenáramú szervomotor, valamint a hozzájuk tartozó vezérlő és meghajtó elektronika van. Bár a motorok nem nagy teljesítményűek, de a szimultán működési lehetőség miatt ezek fogyasztása összeadódhat, így a tápegység - amely külön önálló dobozban van, és saját kapcsolóval rendelkezik - 9 Voltos stabilizált feszültség mellett maximum 3 Amper áram leadására képes.

## 3. A vezérlés

A ROB3 vezérlő programja egy Pascal (Turbo Pascal 5.0) nyelven írt interpretatív program. Ennek segítségével egy egyszerű programnyelv - a már említett ROLAN-alapján (on-line, teach-in vagy off-line módban) vezérlőprogramok írhatók. Csak PTP (startpont → célpont) és MPC (sok-pontos) vezérlés lehetséges.

A ROB3 törzsének aljában van az elektronika, és itt található a csatlakozók is a külvilág felé.

A kivezetések a következők:

- 1./ A piros (+) és fekete (-) vezetékek, melyeket az előző fejezetben említett tápegységhez kell csatlakoztatni.
- 2./ Egy 9-tűs csatlakozó anya (DB-9 female), amelyet a vezérlő computer soros portjára (RS-232, COM1) kell kötnünk. Ezen keresztül kommunikál a számítógép a robottal.
- 3./ Egy 25-tűs csatlakozó anya (DB-25 female), amelyen keresztül a robot kontroll illesztő egysége (control interface) tartja a kapcsolatot a külvilággal. Ilyen egység például a függelékben részletezett un. " Teachbox " kézi vezérlő rendszer, melynek segítségével kontroll computer nélkül is tudjuk működtetni a robotot.

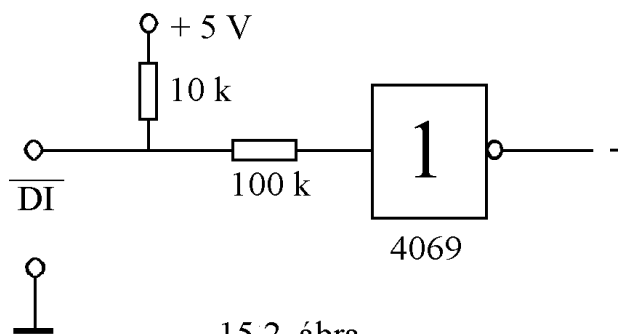
A csatlakozón az alábbi pontok találhatóak:

a./ 1 bites digitális un." EMERGENCY OFF " bemenet aktív LOW szinttel, amely hibás működés vagy váratlan akadály esetében biztonsági gyors-kikapcsolást végez a roboton.

b./ Két analóg bemenet 0 - 10 Voltos működési tartománnyal, 8 bites felbontású analóg-digital átalakítóval.

c./ 8-bites digitális TTL bemenet aktív LOW jelszinttel, az alábbi lábkiosztással és az egyik bemenet kapcsolási rajzával:

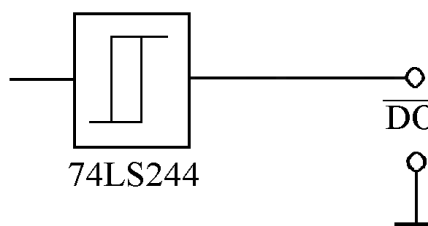
| Bemenet         | lábszám |
|-----------------|---------|
| DI <sub>0</sub> | 8       |
| DI <sub>1</sub> | 20      |
| DI <sub>2</sub> | 7       |
| DI <sub>3</sub> | 19      |
| DI <sub>4</sub> | 6       |
| DI <sub>5</sub> | 18      |
| DI <sub>6</sub> | 5       |
| DI <sub>7</sub> | 17      |



15.2. ábra.

d./ 8-bites digitális TTL kimenet aktív LOW jelszinttel, az alábbi lábkiosztással és az egyik kimenet kapcsolási rajzával:

| kimenet         | lábszám |
|-----------------|---------|
| DO <sub>0</sub> | 25      |
| DO <sub>1</sub> | 12      |
| DO <sub>2</sub> | 24      |
| DO <sub>3</sub> | 11      |
| DO <sub>4</sub> | 23      |
| DO <sub>5</sub> | 10      |
| DO <sub>6</sub> | 22      |
| DO <sub>7</sub> | 9       |



15.3. ábra.

## A RENDSZER INDÍTÁSA

A képernyőn megjelenő „MÉRÉS”-re kattintunk, vagy a számítógépben keressük meg azt az alkönyvtárat, ahonnan el tudjuk indítani a "robstart.bat"-ot. Ezután egy "ROB3 function test" üzenet jelenik meg a képernyőn. Az ENTER gomb megnyomásával megindul a motorok ellenőrzése, ha viszont az ESCAPE -et nyomjuk le akkor a motor ellenőrzés, elmarad. A robot tápfeszültségének hiánya esetén a program külön figyelmeztető szöveget jelenít meg a képernyőn.

## A ROB3 PROGRAMJA

Az előző részben leírt rendszer-ellenőrzés és indítás után mindegyik motor kiinduló helyzetbe, majd az előre beállított FIXPOS állapotba kerül. Bármelyik motor hibás működése esetén hibáüzenet jelenik meg. Ez történik akkor is, ha a robot nem tud az előre beállított pozícióba beállni. Normális esetben ezután a főmenü jelenik meg.

A főmenü két részből áll: a baloldali keskenyebb mező az egyes menüpontokat tartalmazza, a jobboldali - szélesebb - információs mező alsó része általános utasításokat tartalmaz a menüpontok kiválasztására és aktivizálására nézve, felső része pedig a kiválasztott menüpontot magyarázza. A következő két alfejezetben a program működését, majd a ROLAN utasításkészletét tárgyaljuk. A sorrend önkényes, de a részek sokat hivatkoznak egymásra.

### 1. A program működése, felépítése

#### a./ Szerkesztés (EDITING)

Az **EDIT**-et aktiváljuk az **E**, vagy **e** betűvel, és az ENTER-el, majd beírjuk a szerkesztendő program címét, ill. ennek híján az előzőleg szerkesztett ROLAN program kerül be a szerkesztőterületre. A képernyő baloldala a szerkesztőterület.

ROLAN programok szerkesztése: minden sor egy és csak egy ROLAN utasítást tartalmazhat. Futás közben az egyes utasítások sorrendben hajtódnak végre. Az **F1**, **F2**, **F3**, **Pgup**, **Pgdown**, **END** gombok funkciója: mindegyik kétállású kapcsolóként működik és csak az egyes utasítás sorok elején állíthatók!

**F1**: megállási pontot állít be ill., töröl. A megállási pontok csak akkor hatásosak, ha a programot a RUN utasítással indítjuk. Lépésenkénti futtatás esetén a megállási pontoknak nincs jelentőségük.

**F2**: on-line/off-line kapcsoló. Az aktuális beállítás a felső sorban látszik. On-line állapotban minden utasítás, azonnal végrehajtódik (ld. még 4.)

**F3**: a kapcsoló állásától függ, hogy a **PTP**, **LIN** és **CIR** (on-line vagy off-line) mozgató utasítások paramétereit robot-koordináták vagy Descartes-koordináták alakjában jelzi ki.

**Pgup** ill. **Pgdown**: a programlistát 10 sorral feljebb, vagy lejjebb állítja.

**END**: kilépés szerkesztési üzemmódból. Visszatérés a főmenühez.

## Az első ROLAN-program kipróbálása

A főmenüből válasszuk ki az **EDIT** opciót és írjuk be a betöltendő program nevét:

TEST.RLN.

A képernyő baloldali felén (listamező) a következő rövid ROLAN program látható:

FIX 1

PTP 128 10 11 127 129 0

FIX 0

WAIT Z 30

RPT

A kurzor a "FIX 1" F betűjén áll.

A képernyő jobboldala az információs terület. Ez tartalmazza a ROLAN programok könyvtárát (max. 48), és a ROB2R vezérlő utasításainak listáját rövid leírásokkal együtt.

### Szerkesztőfunkciók

Először nyomjuk meg az F1 gombot. A "FIX 1" most inverz színben látszik. F1 ismételt lenyomása visszaállítja az eredeti színeket. Ellenőrizzük most a megállási pont hatását: nyomjuk meg még egyszer az F1 gombot. Térjünk vissza a szerkesztési üzemmódból a főmenübe az END gomb lenyomásával. Az R gomb elindítja a programot. A robot a FIX 1 helyzetbe kerül (kinyújtott felfelé mutató kar), és ott megáll. Bármely gomb lenyomásával a program futása folytatódik (kivéve ESC vagy END): először a PTP utasítás hatására a fogó kinyílik, majd a robot a FIX 0 helyzetbe kerül, ahol 30 tizedmásodpercig marad (WAIT Z 30), végül pedig az RPT utasítás hatására újra a FIX 1 helyzetbe kerül, és újra megáll. Tetszőleges ismétlés után a programot az END gombbal állíthatjuk le.

Olvassuk be újra az eredeti programot és nyomjuk meg az F2 gombot. Ennek hatására on-line üzemmódba kerültünk. Az ENTER gomb ismételt lenyomásával az utasításokat egyenként hajthatjuk végre (SINGLE -STEP PROCEDURE). Az utolsó utasítás, RPT most nem határos, az ENTER lenyomására figyelmeztető hangjelet kapunk. A kurzor az utolsó, üres sorban villog.

Próbáljuk ki a Pgup, Pgdwn, '↑', '↓' gombok hatását: a kurzort a kívánt sorra állítva az ENTER továbbra is az adott sorban lévő utasítás végrehajtását eredményezi.

### Programsorok beszúrása

Álljunk rá a kurzorral az ötödik sor elejére és nyomjuk meg az INSERT gombot. Ezáltal az RPT-vel kezdődő sor eggyel lejjebb tolódik. A keletkező üres sorba új utasítást írhatunk be: FIX + =>. A sorban FIX 0 jelenik meg, a kurzor a 0 fölött villog. Írjuk be a 2-t, zárjuk le a sort ENTER-rel. Ezáltal a robotot a FIX 2 pozíció felvételére utasítottuk 3 s késleltetési időt követően. Hasonló módon (a LIN és CIR utasítások kivételével) bárhol új utasításokat szúrhatunk be a programba. Lépjünk ki a szerkesztésből (END), tároljuk a módosított programot (S) és ellenőrizzük a működését (RUN).

### Programsorok törlése

A LIN és CIR utasítássorok kivételével bármely sort törölhetjük a DELETE gomb lenyomásával.

## Programsorok felülírása

Tegyük fel, hogy a programunkat nem akarjuk ciklikusan futtatni, hanem a végére további utasításokat kívánunk hozzáadni. A kurzor által megjelölt sort (RPT) egyszerűen felülírhatjuk, pl. a FIX 3 utasítással. Tároljuk ezt a változatot is. Bármely ROLAN-utasítás lezárása a ' ⇒ ' gombbal történik. Ha a lezárás előtt a felülírt sort ESC gombbal zárjuk, a felülírás érvénytelen lesz és visszkapjuk az eredeti sort.

### További utasítások hozzáadása

Olvassuk be újra programunkat:

```
FIX 1
PTP 128 10 11 127 129 0
FIX 0
WAIT Z 30
FIX 3
```

Állítsuk a kurzort az utolsó programsorra és nyomjuk meg az ENTER gombot. Írjuk be az új WAIT Z 20 utasítást (ENTER). További sorokat is lehet hasonlóan beírni (FIX 0).

### Megjelenítés Descartes-koordinátákban

Nyomjuk meg az F3 gombot, ezáltal a robot koordináták helyett az összes PTP, LIN és CIR utasítások paraméterei Descartes-koordinátákban jelennek meg:

```
FIX 1
PTP      X 328.6      Y 0.0      Z 309.5
          0.0°        -36.6°      0.0°
WAIT Z 30
FIX 3
WAIT Z 20
FIX 0
```

Descartes-koordináták esetén a PTP, LIN és CIR utasítások paramétereiként a TCP beállítandó helyzetét az alaprendszerhez képest mm-ben adjuk meg - ellentétben a robotkoordinátákkal, ahol az adatok a 6 motor helyzetét lépésekben kell megadni. Az egyes tengelyek jele alatti szög-értékek az aktuális rendszer megfelelő tengelyeinek irányát adják meg az alaprendszerhez képest (ld. Robot kiegészítés, függelék).

### b./ Mentés (SAVE)

Ennek a menüpontnak a kiválasztása az aktuális ROLAN program lemezre mentését eredményezi változatlan név alatt. A megállási pontok a tárolt program részét képezik. A tárolt program továbbra is a képernyőn marad. Ha a memóriában nincs ROLAN program, a mentés hatástalan.

### **c./ Futtatás (RUN)**

A főmenüben ennek a menüpontnak a kiválasztására az aktuális ROLAN program futása indul meg az első sortól kezdve a főmenü elhagyása nélkül. A következőkre ügyeljünk:

- ha a programon belül RPT utasítás van, akkor a program futása ciklikusan újra kezdődik, ameddig meg nem szakítjuk. Az RPT várhatóan csak a program végén hasznos;
- ha megállási pontok szerepelnek a program egy vagy több során: a program futása megáll a megállási pontot tartalmazó utasítássor végrehajtása után. Egy gomb megnyomása hatására a futás folytatódik. Megállási pontokat F1 gombbal definiálhatunk, ill. törölhetünk a szerkesztési módban;
- ha a programot ESC gomb lenyomásával megállítjuk, a programciklus az RPT utasítással bezárólag hajtódik végre, majd megáll;
- ha a programot END gombbal állítjuk meg, a program az éppen végrehajtásra kerülő utasítást követően végleg leáll. Az EMERGENCY OFF funkció nem kerül végrehajtásra; ezt a vezérlő programtól függetlenül kell megvalósítanunk (ld. a manipulátor leírásánál).

Egyéb esetekben a program pontosan egyszer fut le az első programsortól az utolsóig, megállás nélkül (kivételek a programban szereplő WAIT megszakító utasítás esete).

### **d./ Modell-világ (Model World)**

A képernyőn megjelenő információk egy része csak megfelelő beállítás esetén érvényes, ami azt jelenti, hogy a megjelenítő program inicializálja a változókat, a tényleges futtatáskor minden a pillanatnyi állástól függ, ha nem állítottunk be mindent alapállapotba.

Az XY sík az alaplappal párhuzamos, a vállízületet tartalmazó sík, az alaplaptól való távolsága ez által a ROB3 mechanikai beállításának függvénye. A határoló szögek modellfüggőek, és a beállítás során a 2-es és 3-as tengely állításával változnak. Megfelelő beállítás hiányában a Descartes-koordináták és a szögek eltérhetnek a valóságtól.

A TCP-elmozdulásokat, azaz az aktuális rendszerbeli TCP koordinátákat a csuklóízülethez viszonyítjuk, ezek a fogóhoz képest változatlanok és egy új ROLAN program beolvasása előtt definiálatlanul maradhatnak. A TCP-elmozdulás újradefiniálása előtt írt programok futására az újradefiniálás nincs hatással! A TCP a fogó hegyére van beállítva.

### **e./ Nyomtatás (PRINT)**

Az aktuális ROLAN program nyomtatása; a képernyőn látható formától annyiban tér el, hogy Descartes-koordináták esetén minden paraméter külön sorba kerül. Hibaüzenet kikapcsolt vagy nem bekötött nyomtató, vagy hibás státuszjel esetén. Ha a programtár üres, a PRINT utasításnak nincs hatása.

### **f./ Felhasználói eljárás (USER)**

Ez az egység egyetlen eljárást tartalmaz (UserWork), amely a felhasználó által írt eljárások és függvények használatához ad keretet. A "Uses" alatt felsorolt unit-ok deklarálásával a ROB3 minden alapvető eljárása és függvénye elérhető.

### **g./ Beállítás (Electrical & Mechanical Adjustment)**

Az elektromos és mechanikus beállítások részletes leírása szintén a függelékben található.



## h./ Befejezés (TERMINATE)

A következő választásokat kínálja:

- Program mentése mellett kilép a ROLAN programból (S gomb)
- **End** gomb: kilépés, mentés nélkül; a program elvész.
- Megszakítás: Az **Esc** gomb megnyomásával visszatérünk a főprogramba a program státusának megváltoztatása nélkül.

Ha a programból kilépünk, a 10 fix pozíció aktuális értéke a FIXPOS.DEF *file*-ba kerül, és a következő programkezdésnél újratöltődnek. A fix pozíciókat a SETF X, X $\in$ {0,...,9} utasítással változtathatjuk meg. A kilépés után a DOS rendszerbe kerülünk vissza.

## 2. A ROLAN-utasítások működése és használata

Az utasításkészlet 10 utasításból áll: PTP, LIN, CIR, FIX, GRIP, SETF, WAIT, DIN, DOUT, RPT. Minden utasítás általában külön sorba kerül, kivéve a PTP, LIN és CIR utasításokat, melyek Descartes-ábrázolás esetén a paraméterekkel együtt két sort foglalnak el. Nyomtatás esetén minden utasítás pontosan egy sor. ROLAN utasítások programozása során a következő gombok rendelkeznek különleges funkcióval:

**Nagybetűk:** sor elején az utasítás szó bevitelére.

'  $\Rightarrow$  ': két funkciója van. Utasítás vége és szintakszis ellenőrzés. Ha az utasítás szó nem értelmezhető, hangjelzés kíséretében törlődik. Másodsor: ezt a gombot használjuk egyik paraméterről a következőre való áttérés alkalmával (zárt ciklus). Mindegyik paraméter, melyet a '  $\Rightarrow$  ' gombbal "átlépünk", a pillanatnyi értékére állítódik be, függetlenül attól, hogy megváltozott-e vagy sem.

**Számok:** új paraméter értékek bevitelére.

' + ' és ' - ': A PTP, LIN és CIR utasítások esetén a precíziót állítja.

**Backspace:** paraméter értékek korrekciójára.

**Enter:** utasítás lezárására. Egyben új utasítás-sor nyitására.

### a./ PTP

Kijelzés robot-koordinátákban: Az aktuális paraméter a ' + ' és ' - ' gombokkal egyenként változtatható, vagy egész számokként (0 és 255 között) beírható.

Kijelzés Descartes-koordinátákban: A paraméter csak a ' + ', ' - ' gombokkal változtatható. Tekintve, hogy a ROB3 csuklójának csak két tengelye van (x és y), a z-tengely szögértéke rendre az 1-es tengely szögértékének ellentettje, és nem változtatható szabadon. A ROB3 távolságmérő rendszerének alacsony felbontása miatt egy paraméter megváltoztatása a többi paraméter értékének kismértékű változását okozza.

Ha on-line állapotban vagyunk, a paraméter változtatások rögtön életbe lépnek. Hogyha új paraméter értéket írunk be, akkor a '  $\Rightarrow$  ' lenyomása után rögtön az új pozícióba áll be. A '  $\Rightarrow$  ' ismételt lenyomása a következő paramétert jelenti. Az Enter lenyomás hatására a PTP utasítása az új paraméter értékekkel együtt tárolódik.

Vigyázat: off-line állapotban az új paraméter értékeket az Enter lenyomása előtt a '  $\Rightarrow$  ' gombbal kell érvényesíteni.

## b./ LIN

Az utasítást lineáris interpoláció programozására használjuk; előzőleg a kiindulási **A** pontot a PTP, LIN, CIR vagy FIX utasítással kell definiálni. A paraméterek beállítása megegyezik a PTP utasításnál leírtakkal. Az interpolációs út **B** végpontja a LIN utasítás paramétere. Az 'Enter' gomb megnyomása után a számítógép kiszámítja a közbenső pontokat, és a programban tárolja. Futás esetén a TCP a sok-pontos vezérlés módszerrel mozog **A** pont és **B** pont között.

Az **A** és **B** pontokat definiáló utasítások egy blokkot képeznek; egyéb utasításokat nem írhatunk a blokk belsejébe.

## c./ CIR

Az utasítást cirkuláris interpoláció programozására használjuk; előzőleg egymás után három (**A B C**) pontot kell a robot munkaterületére beprogramozni. Az **A** pont a PTP, LIN, CIR vagy FIX, a **B** pont a PTP vagy FIX utasítások egyikével állítható be, végül a **C** pont a CIR utasítás paramétere. Az Enter gomb megnyomása után a számítógép kiszámítja az **ABC** körív közbenső pontjait, és a programban tárolja. Az alapelvek hasonlatosak a LIN utasítás leírásában foglaltakkal.

## d./ SETF

Az utasítás paraméterként a 0-át kínálja fel, ezt az Enter gombbal elfogadhatjuk, vagy a számbillentyűkkel megváltoztathatjuk. Mindkét esetben a legutolsó PTP, LIN, CIR, vagy FIX utasítás paramétereit FIXPOSITION-ként veszi át. Ha a SETF nem a program első utasítása, akkor a PTP, LIN, CIR vagy FIX közvetlenül kell, hogy megelőzze. Ha a SETF a program első utasítása, akkor a ROB3 aktuális pozíciója a referencia. Ilyen módon 10 különböző FIX 0, ... , FIX 9 pozíció állítható, változtatható és a FIX utasítással (4.5.) bevihető.

A tíz FIXPOSITION érték a lemezen "FIXPOS.DEF" nevű file-ban tárolódik a vezérlőprogram lefutása után. A program újraindítása esetén ezek betöltődnek és a FIX utasítással használhatók. A FIXPOSITION újradefiniálása nincs hatással a korábban programozott, megfelelő FIX utasításokra, csakis az ezt követőekre.

Célszerű, hogy egy jellemző pozíciót a SETF 0 utasítással, mint kiindulási pozíciót definiáljunk (pl. 1, 2, és 5 tengelyek 0 fokra, a 3-as tengely 90 fokra, fogó zárva).

A ROB2R kezdeti beállításánál ez a pozíció tárolódik, mint FIX 0, amennyiben a "Save adjustment data" opciót választottuk.

## e./ FIX

Ha a FIX X (X=0, ..., 9) utasítás kerül végrehajtásra, a robot - on-line állapot esetén - abba a pozícióba kerül, amelyet a SETF X utasításban állítottunk be.

## f./ WAIT

Ennek az utasításnak hatására a program futása megszakad, amíg vagy az előre beállított tizedmásodpercekben mért idő letelik, vagy egy megfelelő bitminta érkezik a 8 bit-es digitális bemenetre (ld. 6. oldal). Ez a bitminta 0 és 255 közötti decimálisan megadott paraméter. Ha a "WAIT" van a bemeneten, és a ' ⇒ ' billentyűvel van lezárva, akkor a következő sor jelenik meg: WAIT D. Ilyenkor megnyomva a jobb kurzort ( ⇒ ) a ' D ' változó értékének megfelelően ( ha D > 255, akkor " örökké ") a program várni fog, hacsak az ESCAPE gombbal nem érvénytelenítjük az utasítást .

A 'D'-t felülírva 'Z'-vel és megnyomva a '⇒' gombot a várakozási idő 0 és 999 tizedmásodperc közötti tartományra állítható be. Ellentétben az előzőekkel, ilyenkor a várakozást nem lehet az ESCAPE gombbal lerövidíteni.

### **g./ GRIP**

Az utasítást paraméterek nélkül, a '⇒' és az ENTER gombok lenyomásával zárjuk. Az utasítás a fogó nyitott és csukott állapotai között vált. Csukott helyzetből kiindulva az első GRIP utasítás teljesen nyit (a Gripper motor 0 pozíciója), a következő pedig újra visszacsuk (255-ös pozíció). A közbenső pozíciók a PTP, LIN vagy CIR utasítások hatodik paramétereit állítva érhetők el.

### **h./ DOUT**

Ez az utasítás egy 8 bites digitális kimenetet eredményez (lásd 4-5 oldal).

A '⇒' gomb megnyomása után a vezérlő program egy 0 és 255 közötti számra vár.  $\geq 300$  számok tiltottak, 256 és 299 közötti számoknak nincs hatásuk. Ha a robot on-line üzemmódban működik, az ENTER gomb lenyomása után a csatlakozó kimenetén rögtön megjelenik megfelelő digitális érték.

### **i./ DIN**

Az utasítás a digitális bemenet aktuális értékét jeleníti meg a képernyőn. Üres bemenet esetén ez az érték 0. On-line üzemmódban az ENTER gomb megnyomása hatására, ez az érték a kimeneten jelenik meg. A programozás a GRIP utasításának megfelelő.

### **j./ RPT**

Ez az utasítás mindig az utolsó egy ROLAN programban. Hatására az egész program ciklikusan ismétlődik, amennyiben a program "RUN"-nal indult. Szerkesztési üzemmódban semmilyen hatása sincs. Amennyiben az "RPT" nem a program végén helyezkedik el, akkor a programnak az azt megelőző sorig terjedő része ismétlődik, a további sorok hatástalanok. Az RPT-utasítást a GRIP vagy DIN utasítással azonos módon programozzuk (RPT + '⇒' + ENTER) További információkat még a RUN-utasítás leírásánál találhatunk.

**IRODALOM SEGÍTSÉG:** A „ROBKIEG” nevű kiegészítést (függelék) hasznos elolvasni, mert nagy segítséget ad a mérések elvégzéséhez. Aki szükségét érzi, az kérhet bővebb angol nyelvű irodalmat is.