


Robotkezelés, robotprogramozás (Fanuc)

Maczik Mihály András

2022

Robotkezelés, robotprogramozás (Fanuc).....	1
Maczik Mihály András.....	1
2022.....	1
Robotika alapfogalmak	1
Főbb ipari robottípusok	2
Univerzális csuklós robotok.....	2
Festőrobotok.....	2
Kollaboratív (együttműködő) robotok (KOBOT).....	3
Delta robotok.....	3
SCARA robotok.....	4
Ipari robotok osztályozása tengelyszám szerint.....	5
A legismertebb robotgyártó cégek	7
Néhány statisztikai adat 2020-ból	7
Általános robotjellemzők	8
Szabadságfokok.....	8
Mozgástér, koordináta rendszerek.....	9
Terhelhetőség	9
Sebesség	9
Ciklusidő	10
Lecsengési idő.....	10
Pontosság.....	10
Felbontóképesség	10
Ismétlési pontosság	10
Alkalmazkodóképesség.....	10
A robotok fő egységei	11
Az alapzat.....	11
A karrendszer	11
A munkavégző eszköz.....	12
Az érzékelők.....	13
A mozgatószerkezet	14
A vezérlőegység	15
A robot vezérlési módjai	15
Pontvezérlésű robotok	15
Pályavezérlésű robotok	16
Robotprogramozás	16
Az ON-LINE programozás	16
Az OFF-LINE programozás.....	17
ER-4iA robotkar.....	18
A megfogó.....	21
R-30iB Mate Plus vezérlő	22
Kulcsos üzemmód-váltó kapcsoló, start gomb.....	22
AUTO Folyamatos programfuttatás üzemmód	22
T1 Tanítási üzemmód 1	23
T2 Tanítási üzemmód 2	23
Vészstop nyomógomb	23
Főkapcsoló	24
USB 2.0 csatlakozó	24
A kezelő konzol (Teach Pendant)	25
Vészstop	25
Deadman (halott ember) nyomógombok.....	26
Tanítási mód be-, kikapcsoló	26
USB csatlakozó	26

Fontosabb kezelőgombok a kezelőn	27
Ablakok beállítása a kezelőn	31
Kezdeti beállítás (Initial Setup).....	33
Állapot ablak a kezelőn	34
FCTN Menü	35
 +nyomógomb kombinációk.....	36
Rendszerüzenetek, naplók, riasztások.....	38
A leggyakoribb hibakódok és jelentésük	39
A Roboguide szoftver.....	40
Új cella létrehozása	40
A robot mozgatása.....	42
Nézetek.....	42
Szingularitás	43
Program csomópont térkép beállításai (Program Node Map Settings).....	44
A robotkar mozgatása a TCP pont vonzolásával.....	44
TCP mozgás pontos értékek bevitelével	45
Előzőleg felvett pontok vonzolása.....	46
Roboguide RUN Panel	47
Roboguide 3D Player	49
Egérfunkciók	50
Robot-mozgástér megjelenítése	50
Új szerszám felvétele	51
TCP beállítása	52
Munkasztal létrehozása	54
Munkadarab létrehozása.....	55
A munkadarab hozzárendelése az asztalhoz	56
Főlsleges tároló lap eltüntetése	56
Több azonos munkadarab elhelyezése az asztalon (paletta)	57
Uframe definiálása	57
Move to	58
A Teach Pendant a Roboguide-ban.....	60
Fájlkezelés	62
Fájlok listája	62
Fájl megnyitása	62
Képernyő lapozás több lehetőség esetén.....	62
Szűrés fájl típusokra (TYPE)	62
Új program létrehozása (CREATE).....	62
Program törlése (DELETE).....	64
Jellemzők megjelenítése (ATTR).....	64
Másolás (COPY)	65
A fájl részletes paramétereinek beállítása (DETAIL)	65
Betöltés (LOAD).....	65
Mentés máshova, más néven (SAVE AS).....	65
Nyomtatás (PRINT)	66
Meghajtó váltása	67
Könyvtárműveletek	67
A valódi roboton lévő .TP program biztonsági mentése USB adathordozóra	68
Valódi roboton megírt *.TP program áttöltése Roboguid-ba.....	69
Valódi roboton megírt pozícióregiszterek áttöltése Roboguid-ba.....	70
Felhasználói keretek átvitele a valóságos robot és a Roboguide között	71
IO beállítások megjegyzések (Comment) átvitele a valóságos robot és a Roboguide között.....	72

Roboguide-ben írt program betöltése a robotba USB-ről	73
A valódi roboton lévő minden fájl biztonsági mentése USB adathordozóra	74
Parancsszerkesztő (Edit Command).....	75
Üres sor beszúrása (Insert)	75
Sor törlése (Delete).....	75
Másolás / kivágás (Copy / Cut).....	75
Keresés (Find)	76
Csere (Replace)	76
Újra sorszámozás (Renumber)	77
Magyarázat (Comment).....	77
Mégsem (Undo).....	77
Megjegyzés (Remark)	78
Ikon alapú szerkesztő mód (Icon Editor)	78
Szín megjelenítése (Color)	78
IO állapot (IO Status)	78
Koordinátarendszerek.....	79
JOINT.....	79
WORLD	79
TOOL (szerszám koordináta rendszer)	80
TCP eltolás, UTOOL.....	81
UFRAME (felhasználói koordinátarendszer).....	83
Robot I/O.....	87
Általános célú digitális bemenetek és kimenetek DI/DO.....	89
Range.....	89
Rack.....	89
Slot	89
Beállítás, kezelés	89
A felszerelt perifériás eszközök:	90
Bemenetek és kimenetek megjelenítése a Roboguide-ban	90
DCS (Dual Check Safety) biztonsági zóna	93
Ütközés.....	93
Kioldás	93
Biztonsági rendszer ellenőrzése	93
Robotkar mozgatása	96
Új pont felvétele	96
A mozgatási parancssor.....	97
Regiszterek (változók).....	101
Értékkadás	101
Műveletek regiszterekkel	101
Regiszterek elnevezése, értékeinek megtekintése.....	101
Regiszter programsor bevitele.....	102
Pozícióregiszterek	104
Pozícióregiszter létrehozása	104
Pozícióregiszter bevitele a programsorba	105
Megfogási helyzet programozása Tool_Offset segítségével.....	106
Egyszerű robotparancsok	107
Címkeutasítás (LBL[i])	107
Ugrás címkeutasításra (programon belül) (JMP LBL[i]).....	107
Programhívási utasítás (ugrás másik programra) (CALL).....	107
Program vége utasítás ([END])	108
Feltételes elágazás ugrással (IF).....	108
IF_THEN / ELSE / ENDIF utasítás csoport	109

Várakozás utasítás (WAIT).....	111
Mozgatott teher beállítása (PAYLOAD[i]).....	112
Sebesség felülbírálás megváltoztatása (OVERRIDE).....	112
Megjegyzés (REMARK).....	112
FOR / ENDFOR ciklusparancsok	113
Makró	115
Szín szerinti válogatás feladat.....	118
Fém-nem fém szerinti válogatás feladat.....	122
Palettázási feladat FOR / ENDFOR segítségével	124
Felhasznált irodalom	129

Robotika alapfogalmak

Robot

Többtengelyes, szoftverrel irányított mozgatómechanizmus, ami magában foglalja a mechanikus mozgatórendszert, a vezérlő elektronikát, a programozó eszközt, és a szoftvert.

Ipari robot

Robot ipari termeléshez.

Kiszolgáló robot

Robot, amely hasznos feladatokat lát el emberek vagy berendezések számára, kivéve az ipari termelést. Míg a gyártósorokon használt csuklós robotok ipari robotok, addig az ételek felszolgálására használt hasonló csuklós robotok a kiszolgáló robotok.

Személyi kiszolgáló robot

Kiszolgáló robot személyes használatra, amelyet általában laikusok használnak nem kereskedelmi célokra, pl.: háztartási fűnyíró robot, automatizált kerekesszék, személyi mozgást segítő robot.

Professzionális kiszolgáló robot

Kiszolgáló robot, amelyet kereskedelmi feladatra használnak, általában megfelelően képzett kezelő üzemeltet, pl.: takarítórobot nyilvános helyekre, kézbesítő robot irodákban vagy kórházakban, tűzoltó robot, rehabilitációs robot és sebészeti robot a kórházakban.

Robotrendszer

Robot kiegészítve robotszerszámmal, szerszámváltóval, munkadarab-tárolókkal, befogó készülékekkel.

Ipari robotrendszer

Robotrendszer ipari termeléshez.

Ipari robotcella

Ipari robotrendszer, kiegészítve kapcsolódó védett térrel, kerítéssel és védelmi intézkedésekkel.

Ipari robotsor

Egynél több ipari robotcella, amik egyetlen vagy egy összekapcsolt védett térben vannak

Főbb ipari robottípusok

Univerzális csuklós robotok



- Univerzális, hattengelyes humanoid kivitel.
- Különböféle szerszámozási lehetőségek: megfogó, vákuum-tappancs, AFI hegesztő, ponthegeztő, stb.
- Palettázásnál, rakodásnál akár több száz kilogramm tömeget is tud mozgatni.

Festőrobotok



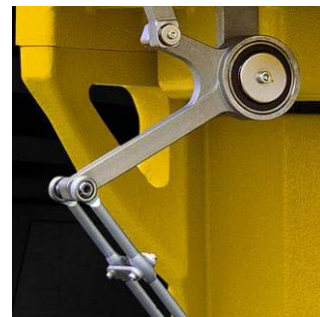
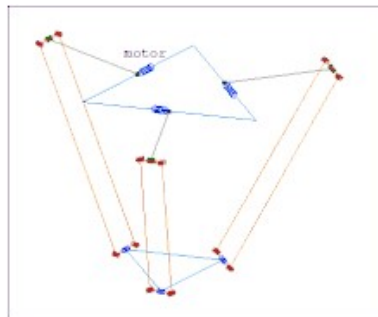
- Speciális könnyűsúlyú alumínium karok, kisebb teherbírás.
- Sima külső felület, könnyű tisztíthatóság.
- Üreges csuklókialakítás vezetékek, csövek számára.
- Felszerelhető padlóra, falra, ferdén vagy a mennyezetre.
- Robbanás-biztos kivitel.
- Speciális, festést segítő szoftvertámogatás.
- Speciális forgó mozgás az ötödik és a hatodik tengely mentén
- Hattengelyes humanoid kialakítás.

Kollaboratív (együttműködő) robotok (KOBOT)



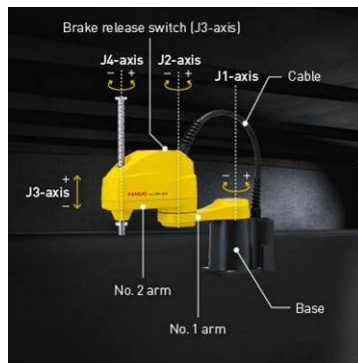
- Nem kell biztonsági kerítés a működtetéséhez.
- Kontaktstop funkció: ütközéskor (emberi test érintésekor) azonnali biztonsági leállítás.
- Kézi irányítású tanítás: elvezethetjük a robotkart kézzel a kívánt helyzetbe, majd elmenthetjük a pozíciót.
- A **Teach Pendant Tablet** alkalmazásával egyszerű ikon-vonzolásos programozás.
- Hattengelyes humanoid kialakítás.

Delta robotok



- Kisméretű és kistömegű alkatrészek nagy sebességű, pontos mozgatása (pl.: apró alkatrészek dobozolása).
- Paralelogramma elven működő robot 3 db csuklós karral.
- 3-, 4-, vagy 6-tengelyes kialakításban gyártják.
- A háromtengelyes kialakításnál a munkadarab csak síkban mozog,
- A négytengelyes kialakításnál a munkadarab dönthető is.
- A hattengelyes kialakításnál a munkadarab forgatható is.

SCARA robotok



- **Selective Compliance Articulated Robot Arm** (szelektív feladatok elvégzésére kialakított robotkar).
- Kisméretű, gyors és olcsóbb a többi típusú robotnál.
- Kiváló az egy síkban lévő dolgok pakolásához, csomagolásához, továbbá összehúzóhoz is.
- Négytengelyes kivételben a leggyakoribb.

Ipari robotok osztályozása tengelyszám szerint

Háromtengelyes robotok



Négytengelyes robotok



Öttengelyes robotok



Hattengelyes robotok



Héttengelyes robotok



A legismertebb robotgyártó cégek

Néhány statisztikai adat 2020-ból

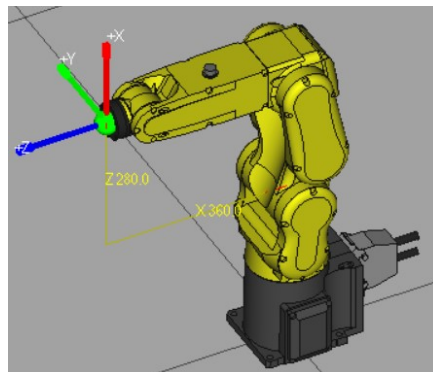
- 383 500 db új ipari robotot üzemelnek be a világon.
- 3 millió ipari robot dolgozik a világon (2010-ben egymillió volt).
- Fő területek: elektronika 35%, autógyártás 25%, gépipar 15%, műanyagipar 10%, élelmiszeripar 5%, egyéb 10%.
- A kollaboratív robotok aránya csupán 8%.
- Országokénti felhasználás: Kína 50%, Japán 15%, USA 10%, Dél Korea 10%, Németország 10%, Egyéb 5%.

Általános robotjellemzők

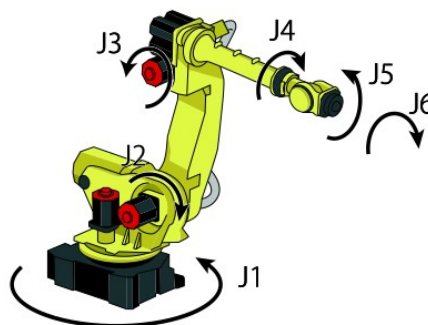
Szabadságfokok

A robot feladata a végrehajtóelem elmozgatása a tér előírt pontjába. Ehhez a robotkar minden szegmensének megfelelő mozgatása szükséges. A végrehajtóelem térbeli elhelyezkedését és irányítottságát **6 szabadsági fokkal** tudjuk leírni, melyek a következők:

1. $\pm X$: x irányú elmozdulás,
2. $\pm Y$: y irányú elmozdulás,
3. $\pm Z$: z irányú elmozdulás,
4. $\pm W$: x tengely körüli elfordulás,
5. $\pm P$: y tengely körüli elfordulás,
6. $\pm R$: z tengely körüli elfordulás.



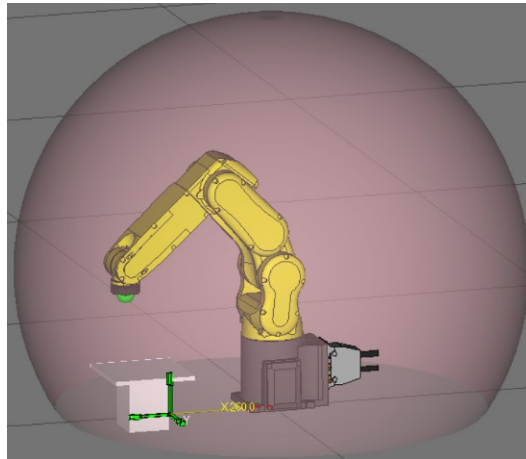
A robotok tengelyszáma összefüggésben van a szabadsági fokkal. Ha a tengelyszám hatnál kevesebb, a robot elveszíti valamelyik irányú mozgási vagy forgási lehetőségét. Hét tengely esetén is hat a szabadsági fokok száma, de a robot sokkal eldugottabb helyekre is be tud férni.



Mozgástér, koordináta rendszerek

A mozgástér vagy munkaterület a háromdimenziós tér azon részét jelöli, melyben a robot a maximális terheléssel a megadott pontossági értéken belül tud működni. A terület alakja a szabadsági fokok számától, a kar szegmenseinek hosszától és az egyes tengelyek mozgástípusától függ.

A legelterjedtebb az iparban **humanoid** rendszer (munkaterülete hiányos gömb alakú).



A munkahelyeken, a robot működési területén szükség van rendszerint tiltott zónákra is. Ezek olyan területek, ahol a robot biztonsági okokból nem manőverezhet.

Terhelhetőség

A terhelhetőség fogalmán azt a maximális tömeget értjük, melyet a robot kinyújtott karral a működési terület vízszintes és függőleges síkjában egyaránt hordozni tud. A terhelhetőséget három fő tényező határozza meg:

1. Az első a robot működtető **motorjai** által szolgáltatott **teljesítmény**.
2. A második tényező a robot **szerkezeti** felépítése. Ha a robotkart vízszintes irányban teljesen kinyújtjuk, már viszonylag kis terhelés mellett is óriási nyomaték nehezedik a vállra.
3. A harmadik, hogy a terhelés **mozgatásának sebessége** is hatással van a terhelhetőségre. Ennek oka, hogy **gyorsuláskor** illetve **lassuláskor** nyomaték hat a szerkezetre.

Sebesség

A sebesség a végrehajtó mozgásának gyorsaságára utaló jellemző. A kar mozgásának sokszínűsége miatt ezt a jellemzőt igen nehéz pontosan meghatározni. A robotgyártók vagy a **végrehajtóelem maximális sebességét** vagy pedig az **ízületek maximális forgási sebességét** adják meg. Általában az értéke 0...2500 mm/s között változik.

Ciklusidő

A ciklusidő, egy adott feladat elvégzésének ideje. Ez, a **sebességen** kívül függ a robot egyéb jellemzőitől és az **elvégzendő feladattól** is. Például az ívhegesztés varratának elkészítése sokkal több időt vesz igénybe, mint egy ponthegesztés.

A **terheléstől** két okból függ a ciklusidő. Először a robotkar nagyobb terhelés esetén valamivel lassabban gyorsul illetve lassul. Másodszor a nagyobb terhelések tehetetlensége túllendülést okoz a megállásoknál, ami rontja a mozgás pontosságát.

Lecsengési idő

Egy-egy mozzanat befejezése után a robot végrehajtóeleme még rövid ideig tovább mozog, vagy leng. A lecsengési idő tehát az az idő, mely alatt a végrehajtó nyugalomba jut.

Pontosság

A pontosság megmutatja, hogy a robot a végrehajtó elemet a tér egy kijelölt pontjába milyen pontosan tudja elhelyezni. Minden egyes ízületnél keletkezik a mechanikai kialakítások miatt egy kis hiba. Bár ezek a hibák az egyes pontokban igen kicsik, az összehatásuk már számottevő lehet.

Felbontóképesség

A felbontóképesség alatt azt a legkisebb elmozdulást értjük, melyet a robot még érzékelni tud.

Ismétlési pontosság

Az ismétlési pontosság azt mutatja meg, hogy a robot egy korábban már megtalált pontba hány ezred mm eltéréssel tud visszamenni.

Alkalmazkodóképesség

A robotok minden mozdulatában van egy kis hiba. A feladatok elvégzése során ezért valamilyen módon a hibákat kompenzálni kell. Az alkalmazkodó képesség erre a kompenzáló mozgásra utal.

A kompenzálásnak két módja használatos:

1. passzív kompenzálás
2. aktív kompenzálás

A passzív kompenzálás során a pontatlanságból keletkező feszültséget a csukló rugalmassága nyeli el.

Az aktív kompenzálás folyamán érzékelők, nyúlásmérő bélyegek jelzik a vezérlőnek a csuklóra ható feszültségeket. Ezt a motorok kompenzálják.

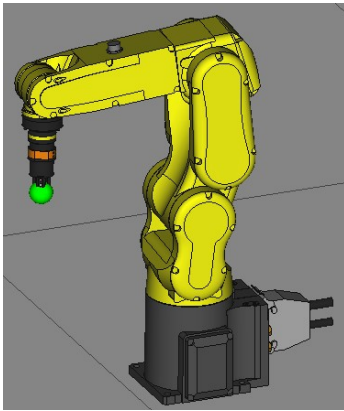
A robotok fő egységei

1. alapzat,
2. karrendszer,
3. munkavégző eszköz,
4. érzékelők,
5. mozgatószerkezet,
6. vezérlőegység.

Az alapzat

Az alapzat a robot szerkezetének bázisa, melynek két fő típusa:

1. rögzített (padlóra, falra, mennyezetre, másik gépre),
2. mozgó (sínrendszeren, futópályán).



A karrendszer

A robotkar ízületekkel (csuklókkal, csúszkával) összekapcsolt szegmensekből áll, mely az alapzathoz kapcsolódik. A feladata a munkavégző eszköz mozgatása az adott térrészben. A szegmensek merev részek, melyek ízületekkel kapcsolódnak egymáshoz. Ezek az ízületek teszik lehetővé a szegmensek adott irányú, vagy adott tengely körüli mozgását.









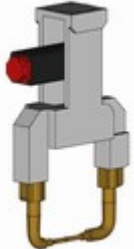

Az alapzathoz a derékon keresztül kapcsolódó szegmenseket kezdőszegmensnek nevezük. Azt a karrészt viszont, amelyhez a csukló által a munkavégző eszköz kapcsolódik, végső szegmensnek hívjuk.

A munkavégző eszköz

A robotkar végső szegmenséhez egy csuklón keresztül kapcsolódik. Típusai:

- megfogó, szorító, (mely az emberi kezet utánozza)
- vákuumos tapadókorong (sima, tiszta felületek pl.: üveglap megfogásához),
- elektromágnes (mágnesezhető anyagok (vas) megfogásához),
- szerelőszerszám (pl.: csavarhúzó, dugókulcs),
- ragasztótubus,
- festékszóró,
- AFI hegesztőpisztoly,
- ponthegesztő szorítópofa.

Amennyiben növelni szeretnénk a robot sokoldalúságát, akkor a szerszámot úgy kell kialakítani, hogy könnyen lepattintható legyen a csuklóról és a helyébe egy másikat lehessen rögzíteni.

				
Lefújó	Sorjázó	Ultrahangos kés	Kétujjas megfogó	Háromujjas megfogó
				
Többszörös megfogó	Vákuum tappancs	Lézerszkener	Ponthegesztő	AFI hegesztő

Az érzékelők

Az érzékelő-rendszer lehet a robot szerves alkotórésze, de gyakori az is, hogy elkülönül tőle, és csak információt küld a robotnak. Az információ továbbítása elektronikus vagy elektromechanikus úton történhet. Főbb típusai:

- végállást érzékelő elektromos mikrokapcsoló,
- a karrendszer helyzetét visszacsatoló kódolók
- induktív közelítésérzékelő,
- mágneses, REED- relés közelítésérzékelő,
- kapacitív közelítésérzékelő,
- különféle optikai közelítésérzékelők,
- ipari kamerás látórendszer,
- tapintó érzékelők,
- távolságmérők.

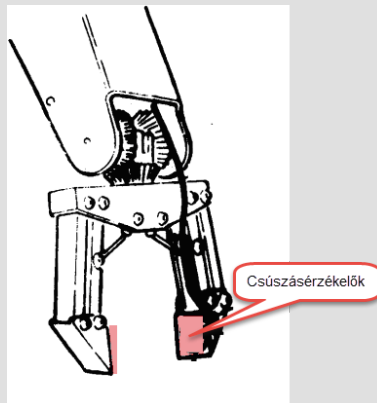
Igen gyakori, hogy a robotnak éreznie kell a tárgyakat. Ez a feladat a különböző **tapintóérzékelők, erőérzékelők, közelítésérzékelők** segítségével valósítható meg. A főbb típusok:

1. elektromechanikus mikrokapcsoló,
2. elemi, miniatűr érzékelők mátrixba rendezve.

Van olyan eset, amikor nem elég azt tudni, hogy a megfogó megragadta a tárgyat, hanem információra van szükségünk arról is, hogy az adott tárgy a megfogóban mozdulatlan.

Erre használhatók a **csúszásérzékelők**. Főbb típusai:

1. mágneses szegmensekkel ellátott gumigörgő és vele szemben egy mágneses érzékelő,
2. fényvisszaverő szegmensekkel ellátott gumigörgő és vele szemben egy optikai érzékelő,



A távolságmérő rendszerek

Segítségükkel a robot a tárgy pontos távolságát tudja meghatározni. Főbb típusai:

1. ultrahangos távmérő,
2. lézersugaras távmérő.

Az ultrahangos távmérő kiküld egy impulzust, ami az érzékelendő tárgyról visszaverődik. Az impulzus terjedési sebességének valamint a kibocsátás és a visszaverődés közötti idő ismeretében a mikroprocesszor vagy mikrokontroller ki tudja számítani a tárgy távolságát.

A lézeres távolságmérő sokkal pontosabb eszköz. Segítségével nanométernél kisebb távolságok is mérhetők. Ez az érzékelő kibocsátott lézersugár visszaverődési idejével méri a távolságot. Sajnos a környezeti hatások például füst, köd, por és eső - kedvezőtlenül befolyásolják a működését, szétszórják, elnyelik a sugarat.

A mozgatószerkezet

A mozgórendszer minden robotban három fő egységből tevődik össze:

1. energiaforrás,
2. a forrásenergiát mechanikaivá alakító egység,
3. a mechanikai energiát továbbító egység.

Az energiaforrások:

1. pneumatikus (hengerek, vagy motorok mozgása sűrített levegővel),
2. hidraulikus (hengerek, vagy motorok mozgása nagynyomású olajjal),
3. elektromos (elektromos szervomotorok vagy léptetőmotorok).

Az energia eljuttatása az ízületekhez két módon lehetséges:

1. A motorokat a megfelelő ízületekben helyezik el.
 - A módszer előnye, hogy nincs szükség mozgástovábbító fogaskerekekre, csigákra, láncokra. Emiatt a robot felépítése egyszerűbb, kevesebb a teljesítményvesztés és nagyobb pontosság érhető el.
 - A módszer hátránya, hogy amennyiben a motor nagy tömegű, akkor a robotkar robosztus lesz, s így a manőverező képessége romlik. A másik hátrány, hogy a működtetők viszonylag jelentős tömege komoly mechanikai feszültségeket ébreszt a robot szerkezetében.
2. A motorokat az alapsztatban helyezik el. A mozgást ekkor csigák, fogaskerekek valamint láncok, fogazott szíjak segítségével továbbítjuk az ízületekhez.
 - A módszer előnye, hogy el lehet kerülni az ízületek nehézkes mozgását.

- A módszer hátránya, hogy összetett feladat a mozgás továbbítása, ami növeli az alkatrészek számát és a robot mechanikai bonyolultságát ugyanakkor a robot pontossága erősen lecsökken. Az egyes alkatrészek között felépő súrlódás növeli a teljesítményvesztésüket.

Napjaink robotjainak leggyakrabban a két módszer kombinációját használják. A kisméretű működtetőket az ízületekbe szerelik, növelve a pontosságot és csökkentve az energiavesztést. A nagyméretű működtetőket pedig az alapszintre helyezik el, megkímélve ezzel a robot szerkezetét a túlzott igénybevételtől és javítva a kar illetve az adott szegmensek mozgékonyosságát.

A vezérlőegység

A vezérlőegység irányítja a berendezés összes mozgását. Főbb típusai:

- mechanikus vezérlők (végállás-ütközők, vezértárcsák, vezérdobok)
- pneumatikus vezérlés (pneumatikus szelepek, léptető),
- hidraulikus vezérlés (hidraulikus szelepek, léptető),
- elektromechanikus, relés vezérlés,
- **mikroprocesszoros** vagy számítógépes vezérlés.

Ma korszerűnek csak a mikroprocesszorral vagy számítógéppel vezérelt robotok tekinthetők, melyek utasításkészlettel programozhatók. Ezek főbb előnyei:

- kisebb, mint bármely más típus,
- ára viszonylag alacsony,
- könnyen programozható,

A robot vezérlési módjai

A robotok a szerint is csoportosíthatók, hogy a vezérlőegység milyen módon végzi a mozgások irányítását. Eszerint két fő típust különböztetünk meg:

1. **pontvezérlésű** robotokat,
2. **pályavezérlésű** robotokat.

Pontvezérlésű robotok

A pontvezérlés során az út egymást követő lényeges irányváltó pontjainak koordinátáit elraktározzák a robot memóriájában. A robot pontról-pontra halad, mozgása két pont között ív/egyenes vonal menti, (esetleg három pontra illeszthető körvonal menti). A robot minden egyes szegmense mozog egészen addig, amíg a végrehajtóelem el nem éri a kívánt pozíciót.

A robotkar kapcsolatban van egy visszajelző rendszerrel. Ez a visszajelző rendszer jelet küld a vezérlő bemenetére, mely összehasonlításra kerül a kívánatos értékkel. Amikor a referencia jel megegyezik a visszacsatoló jelével, a szabályzó kimenetén megszűnik a hibajel, így a működtető leáll. A robot tehát elérte a kívánt pontot.

Ezt a rendszert zárt hurkú szervomechanizmusnak nevezzük. Szervomechanizmus azért, mert kis energiával (jeladó kimeneti feszültsége) irányítunk egy nagy energiájú rendszert (a hidraulikus működtető). A zárt hurkú kifejezés pedig arra utal, hogy visszacsatolás van a kimenet (a manipulátor mozgása) és a bemenet (a jeladó feszültsége) között.

Pályavezérlésű robotok

A pályavezérlésű robotok esetében a végrehajtóelem mozgásának útvonalát nagy adatsűrűséggel mentjük el, tanítjuk be. Általában tanítás (felvétel) üzemmódban kézzel mozgatjuk a robotkart, a felvétel végét követően ezt visszajátsszuk, ha kell, korrigáljuk.

A pályavezérlést akkor használják, amikor speciális úton kell mozgatni a végrehajtót. Igen jó ipari példa a festő- és az ívhegesztő robot vezérlése, a pályavezérlés segítségével ugyanis bármilyen alak követhető. A rendszer egyetlen hátránya, hogy nagyobb memóriakapacitást igényel.

Robotprogramozás

A robotprogramozás két nagy csoportra osztható:

- ON-LINE,
- OFF-LINE.

Az ON-LINE programozás

Az ON-LINE programozás során magát a robotot programozzuk valamilyen tanító eszköz segítségével. A rendszer előnye, hogy a programozó számításba tudja venni a munkaterületen elhelyezkedő tárgyakat és azonnal ellenőrizni tudja a működést. A hátránya különösen ipari szempontból jelentős, a programozás idejére a robotot le kell állítani, tehát ez alatt nem termel. Az ON-LINE programozás módszerei:

Pontvezérlésű robotok programozása

Ennek lényege, hogy a robotot a kezelő számítógép, vezérlőkonzol vagy kézi vezérlőkészülék segítségével tanítja meg a szükséges mozgásokra. A tanítás során a kezelő pontonként halad, és az egyes pontokat külön-külön beírja a robot memóriájába. A vezérlőkonzol a robotra vagy mellé rögzített kezelőpult, míg a kézi vezérlőkészülék egy kisméretű, hordozható

billentyűzet. A mozgáspontok rögzítése után valamilyen programnyelven meg kell írni azt a programot, amelyik közli a robottal, hogy mit kezdjen az előzetesen letárolt pontokkal.

Pályavezérlésű robotok programozása

A programozó kézzel végigvezeti a robotkارت az elvégzendő feladat útvonalán. A mozgás paramétereit ennek során a robot vezérlőegysége önállóan rögzíti. Ehhez természetesen szükséges, hogy a programozó az adott feladat elvégzésében járatos legyen (pl. hegesztésnél tudja a hegesztés mozzanatait stb.). Ezzel a programozási móddal igen bonyolult működések is elérhetők, de viszonylag nagy memóriakapacitás kell az útvonal paramétereinek rögzítéséhez.

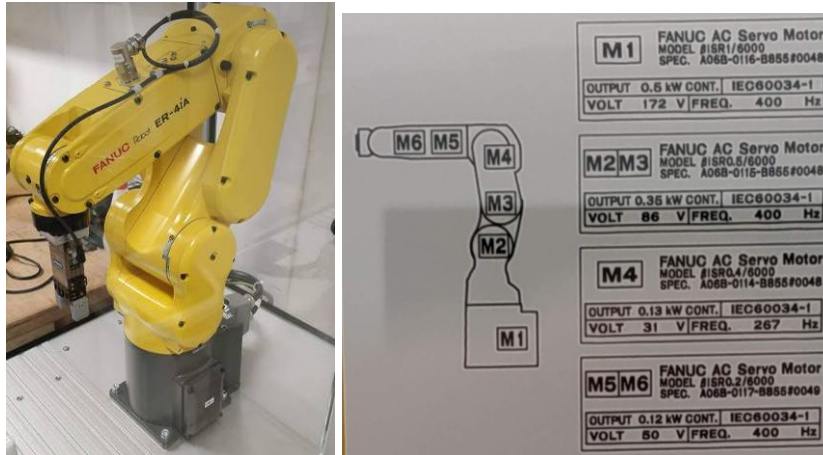
Az OFF-LINE programozás

Ennek során a programozó egy számítógép előtt, szimulációs programmal, a robottól függetlenül fejleszti ki a működést irányító programot. Ezután ezt a programot valamilyen módon átviszi a robot memóriájába. A koordinátarendszereket, eltolásokat, pontok koordinátáit viszont ilyenkor a valós környezetben újra fel kell venni, illetve ellenőrizni kell.

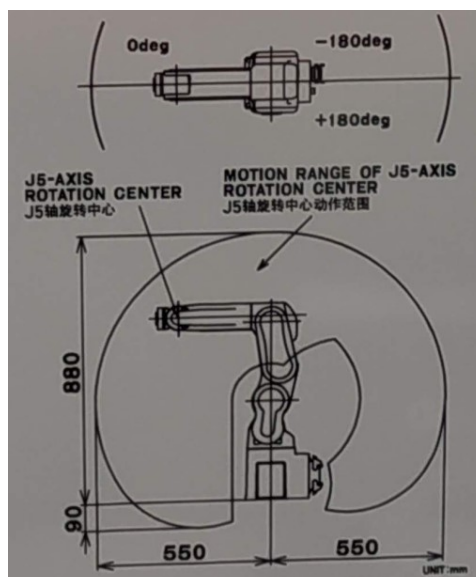
A legnagyobb előnye ennek az eljárásnak, hogy a programozás illetve a programfejlesztés alatt a robotot nem kell kikapcsolni.

ER-4iA robotkar

Gyártó: FANUC (Japán), a robotkar típusa: ER-4iA



- Tengelyek és motorok száma: 6
- Motor típusok: különböző teljesítményű, feszültségű, frekvenciájú szervomotorok
- Teherbírás maximum: 4 kg
- Elfordulás maximum: $\pm 180^\circ$
- Vízszintes kinyúlás maximum: 550 mm
- Függőleges kinyúlás maximum: -90 mm + 880 mm



Vezérlő típusa: **R-30iB Mate Plus**



Kezelés: **iPendant** érintőképernyős kézi konzollal



Látás: **iRVision SC130EF2** fekete-fehér kamera



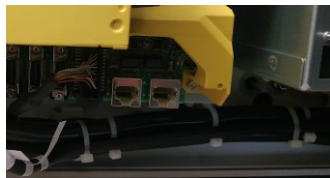
Bemenet, kimenet a robotkaron: **8 db** digitális bemenet és **8 db** digitális kimenet



Bemenet, kimenet a vezérlőn: **28 db** digitális bemenet és **24 db** digitális kimenet található a vezérlő belsejében, ha kinyitott ajtóval szemben állunk, akkor a jobb kéz felőli oldalon 2 db kék színű csatlakozóra kivezetve. Jelölésük: **CRMA58** és **CRMA59**.



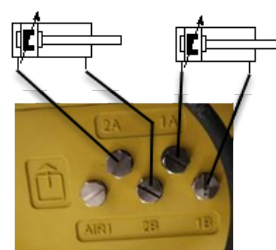
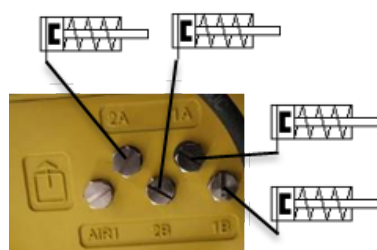
Számítógépes hálózat a vezérlőn: **2 db Ethernet** port a dobozon belül. A két port nem lehet egy hálózaton. Egy robothoz egy **PC** csatlakoztatható egyszerre (azonos hálózat, **fix IP** címek). Magán a roboton fut egy web szerver, a **PC**-ről böngészővel (**Edge** ajánlott) be tudunk lépni. Egyenes Ethernet kábelt kell használni.



Pneumatika: 2 db sűrített levegő betáplálási lehetőség a robot talapzatán, továbbá 1+4 db kivezetés a robotkaron.

Az **AIR1** talapzati csatlakozó csak egy sima cső, átvezeti a sűrített levegőt a robotkaron lévő **AIR1** kimenetre, ami elektropneumatikus megfogónak biztosít táplevegőt.

Az **AIR2** talapzati csatlakozó 4 db mágnes-szelephez vezeti a sűrített levegőt, amik nem komplementerbe állítva 4 db egyszeres működésű elemet, komplementerbe állítva 2 db kettős működésű elemet vezérelhetnek.



A megfogó

Schunk EGP 40-NNB

- Lökethossz állkapcsenként:..... 6 mm, (összesen 12 mm)
- Fogóerő: 35 N...140 N (állítható)
- Tápfeszültség és a kapcsolási jelek feszültségei: 24 V
- TCP, vagyis a szerszámközep a felfogatástól:..... 115 mm



- 1 2 db megfogó ujj belül fogasléccel
- 2 Holtjátéktól mentes keresztgörgős csapágyazás
- 3 Hajtás fogaskerék áttétele
- 4 Szénkefe nélküli DC szervomotor
- 5 Vezérlő elektronika csatlakozóval

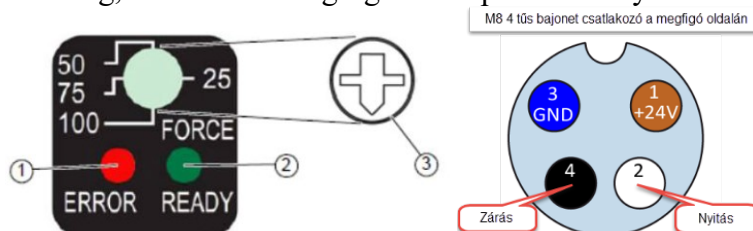
A megfogó oldalán egy kis ledugózott forgatható potenciométerrel (3) állítható be a **szorítóerő** nagysága %-ban.

A **READY LED** megfelelő tápellátást, az **ERROR LED** pedig figyelmeztetést vagy hibát jelez.

A zárási és a nyitási jel is **+24V**, elméletileg a nyitás / zárás ideig tartó impulzus is elég lenne a működtetéshez (ennél rövidebb impulzus esetén részben mozdul a megfogó), valójában folyamatos kapcsolójelet ad ki a robot vezérlője. A **RO[7]** és a **RO [8]** robot-kimenet vezérli a megfogót komplementerbe kötve, hogy ne lehessen egyidejűleg kétirányú mozgató parancs.

Az **ERROR LED**

- Folyamatosan világít „túl magas hőmérséklet” figyelmeztetésnél (magától kialszik, ha a probléma megszűnik)
- Lassan villog „túl magas hőmérséklet hibánál”, (a hibát külön nyugtázni kell)
- Gyorsan villog, ha elakad a megfogó két kapcsolási helyzet között.



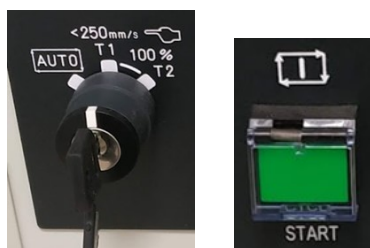
Hiba nyugtázása:

1. Kapcsoljuk ki a robotot!
2. Várjuk meg, amíg a megfogó lehűl!
3. Kapcsoljuk be a robotot!
4. Nyissuk ki, majd csukjuk be a megfogót!

R-30iB Mate Plus vezérlő



Kulcsos üzemmód-váltó kapcsoló, start gomb



AUTO Folyamatos programfuttatás üzemmód

- Zárjuk be a robotcella ajtaját, csak így lehetséges az **AUTO** mód!
- A vezérlőn a kulcsos kapcsolót fordítsuk el **AUTO** módba!
- A **Teach Pendant Teach** kapcsolóját fordítsuk **OFF** állapotba!
- Nyomjuk le a **RESET** gombot a **Teach Pendanton**, erre kialszik a **FAULT LED**!
- Nyomjuk be a vezérlőn a **zöld** folyamatos üzemmód indító **nyomógombot**, ekkor a gomb zölden világít. A gombot egy átlátszó, felhajtható műanyag fedél védi a véletlen benyomódástól.
- Ha programoztuk, és ki van építve, nyomjuk le a **START** digitális input nyomógombot! Ha nincs, azonnal indul a ciklus!
- **FIGYELEM!** Ha nem programoztunk **VERRIDE** értéket, akár **100%** sebességgel indulhat a robotkar, ami veszélyesen gyors lehet! (Ajánlott a 30%).

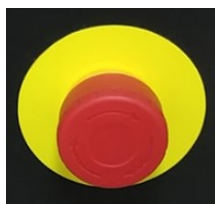
T1 Tanítási üzemmód 1

- Nyitott ajtó mellett is működik.
- **100%** felülírási érték mellett csak **250 mm/s** sebesség érhető el.
- A vezérlőn a kulcsos kapcsolót fordítsuk el **T1** módba!
- A **Teach Pendant Teach** kapcsolóját fordítsuk **ON** állapotba!
- A **DEADMAN** kapcsolót tartsuk középső állásban!
- Nyomjuk le a **RESET** gombot a **Teach Pendanton**, erre kialszik a **FAULT LED**!


T2 Tanítási üzemmód 2

- A program tesztfutása során **100%** felülírás mellett a program teljes sebessége engedélyezett. A többi megegyezik **T1**-el.


Vészstop nyomógomb





Ha lenyomjuk, a robot megáll ott, ahol éppen tart.

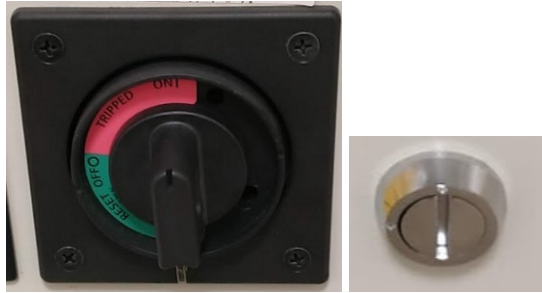
Feloldás AUTO üzemmódban: a veszély elhárítása után a vészstop gomb óra mutató járásával megegyező irányú elfordítása (erre kipattan a gomb), majd a  gomb egyszeri

lenyomása a **Teach Pendanton** (erre a **Fault LED** kialszik), majd a  gomb egyszeri lenyomása a vezérlőn. *A robot ott folytatja a mozgását, ahol abbahagyta.*

Feloldás T1 vagy T2 üzemmódban: a veszély elhárítása után a vészstop gomb óra mutató járásával megegyező irányú elfordítása (erre kipattan a gomb), majd a **DEADMAN** gomb folyamatos középállásban tartása mellett a  gomb egyszeri lenyomása a **Teach**

Pendanton (erre a **Fault LED** kialszik), majd a  gomb folyamatos lenyomása mellett a  gomb egyszeri lenyomása a **Teach Pendanton**. *A robot ott folytatja a mozgását, ahol abbahagyta.*

Főkapcsoló



Piros állásban be van kapcsolva a vezérlő, zöld állásban pedig ki.

Ha a vezérlő ajtaját ki szeretnénk nyitni az ott elhelyezett zárat egy széles, lapos csavarhúzó segítségével vízszintes állásba kell fordítanunk, de az ajtó csak akkor nyitható, ha a főkapcsolót kikapcsoljuk, és tovább forgatjuk a „Reset” irányban, (vagy egy rejtett nyíláson benyúlunk).

USB 2.0 csatlakozó



Kis műanyag ajtó védi az **USB 2.0** csatlakozót. **PC**-n megírt programunkat vihetjük segítségével át a vezérlőre, vagy a vezérlőről a **PC**-re.

USB memóriakártya meghajtó jele a vezérlőn (**UD1:**)

A vezérlő képes olvasni és írni a **FAT** vagy **FAT32** típusú formázással formázott **pendrive**-kat. Ha a **pendrive**-t a vezérlőn formázzuk, az **FAT** vagy **FAT32** típusúra formázható.

A kezelő konzol (Teach Pendant)

A **Teach Pendant** a robot program futtatására, programozására, a program tesztelésére, a beállítások megváltoztatására, állapotok ellenőrzésére szolgál. Ez a kapcsolat a kezelő személy és a vezérlő berendezés között. Minden programozás és tesztelés a **Teach Pendant** segítségével történik. Érintőképernyője **1024x768** pixel felbontású.



Vészstop

A vészstop gomb működése megegyezik a vezérlő egység vészstop gombjával.


Deadman (halott ember) nyomógombok

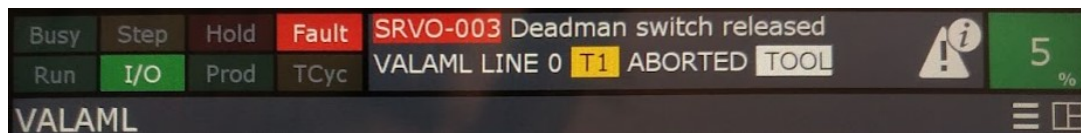
Háromállású nyomógombok, két teljesen azonos funkcióval, bal-, és jobbkezesek számára. Egyszerre elegendő az egyiket használni. Felengedett, és teljesen lenyomott állásban veszleállítás történik.

A robotkart tanítási üzemben mozgatni, futtatni csak úgy lehet, ha folyamatosan az egyik Deadman gombot középállásban tartjuk.

Programozni, szerkeszteni, fájl műveleteket végezni lehet nyomva tartás nélkül is.

Ha elengedjük a **Deadman** gombot, a felső sorban világít a **Fault LED**, kíródik a **SRVO-003 Deadman swich released** hibaüzenet, és villog a veszélyt jelző felkiáltó jeles háromszög.

Feloldás: folyamatosan nyomjuk középállásba az egyik **Deadman** gombot, és lenyomjuk egy pillanatra a  gombot.



Tanítási mód be-, kikapcsoló



A robot tanításához, kézi mozgatásához, teszteléshez ennek a kapcsolónak bekapcsolt állapotban kell lennie. **AUTO** üzemmódba kapcsoláskor **OFF** állásba kell fordítani!

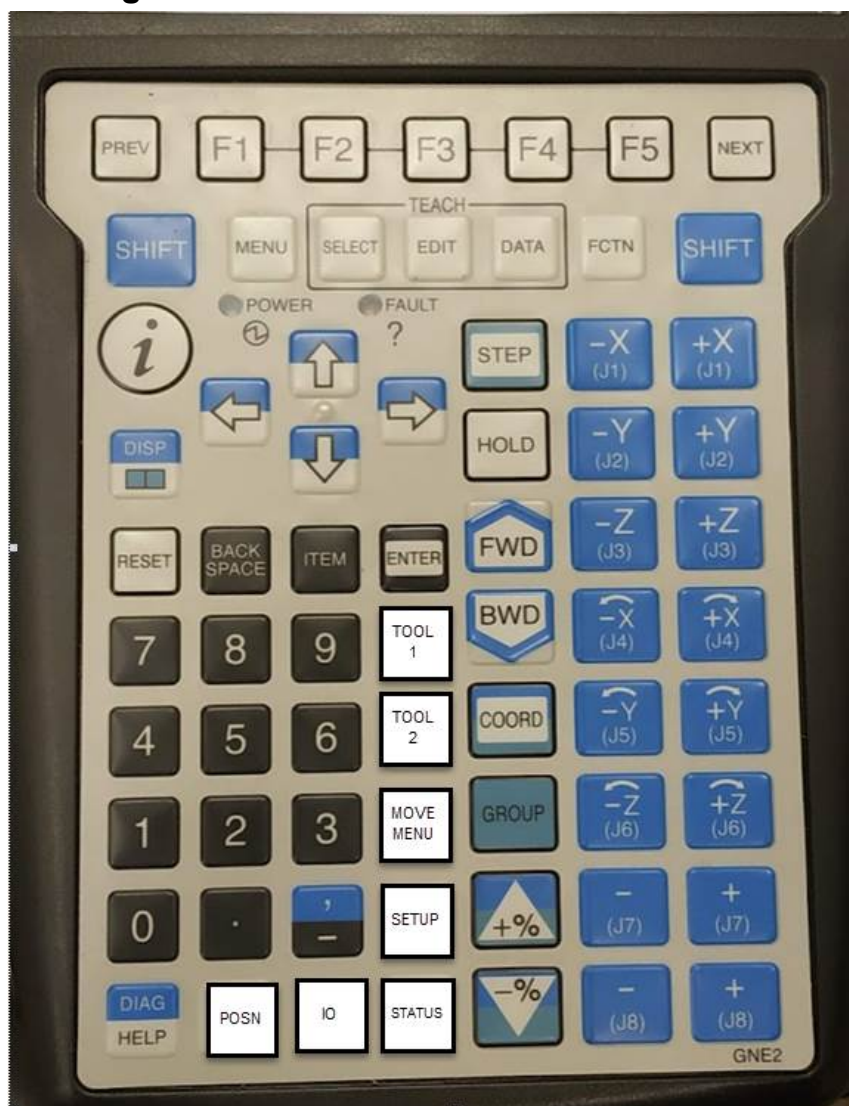
USB csatlakozó






Kis műanyag ajtó védi az **USB 3** szabványú csatlakozót. Az **USB 3** csatlakozóhoz billentyűzetet, egeret és **USB pendrive**-t (ami mérete lehet **32 MB-tól 1 GB-ig**) csatlakoztathatunk.




















A **Teach Pendant** képes olvasni és írni a **FAT** vagy **FAT32** -re formázott **pendrive**-kat, de itt formázni nem lehet.








Az **USB** meghajtó jele a **Teach Pendant-en (UT1:)**











Fontosabb kezelőgombok a kezelőn



	Egy szinttel feljebb lép a menürendszerben
F1 F2 F3 F4 F5	Az érintőképernyőn fölöttük megjelenő feliratoknak megfelelően más-más funkciók.
	Ha a képernyő alján jobbról megjelenik a  ikon, a Next lenyomásával újabb funkciók hozhatók elő. Ismét lenyomva visszatér az eredeti képernyő.
	Lenyomva a kék színű gombok aktívak, felengedve a fehér színű gombok aktívak. A két SHIFT gomb egyenértékű, elég az egyiket nyomni.
	Előhoz egy többablakos menüválasztékot.

	Előhozza a letárolt programokat, vagyis megnyitja a programok listáját.
	Bárhol is járunk, bármely képernyőt látjuk, visszavisz a legutóbbi programunk szerkesztésének képernyőjéhez.
	Különbféle regisztereket beállító ablakot nyit meg.
	Megjeleníti a funkciómenüt, ahol elsősorban különféle robotkezelési parancsok adhatók ki (pl.: ABORT (ALL)), vagyis minden futó program leállítása).
POWER LED	A robot bekapcsolt állapotát jelzi.
FAULT LED	Leállást jelez, pl.: ha nem nyomjuk le félig a DEADMAN gombot.
	Lenyomva megjelenítheti az „ értesítések képernyő ”, amikor a  villogó ikon megjelenik az állapotsoron.
	Több képernyő között tudunk váltani (ha van), SHIFT -el együtt megjeleníti a képernyő felosztás menüt, ahol beállíthatjuk, hány darab és milyen felosztású képernyőt szeretnénk.
	Kurzormozgató gombok a programban történő navigáláshoz.  -tel együtt nyomva balra a sor elejére, jobbra a sor végére, fölfelé és lefele pedig 10 sort ugrik.
	Lépésenkénti üzemmód be-, kikapcsolása programfuttatáskor (beragadós kapcsoló). A programfutást léptetni AUTO módban a vezérlőn elhelyezett  gomb lenyomásával, T1 és T2 módban  nyomva tartásával+  lenyomásával lehet.
	Programfutás pillanat- állj. Egyszeri lenyomására a robotmozgás megáll ott, ahol éppen tart. A programfutást folytatni AUTO módban a vezérlőn elhelyezett  gomb lenyomásával, T1 és T2 módban  nyomva tartásával +  lenyomásával lehet.
	Előre felé lefut a program, Step üzemmódban 1 lépést előre lép a program
	Hátra felé lefut a program, Step üzemmódban 1 lépést hátra lép a prog-

	ram
	<p>A  gomb minden egyes megnyomásakor kiválaszt egy koordináta rendszert a következő sorrendben: JOINT, WORLD, TOOL, USER.</p> <p>Ha ezt a gombot lenyomjuk, miközben  be van kapcsolva, megjelenik egy sárga háttérű kis menü a koordinátarendszer megváltoztatásához. Itt nem csak választani, de értéket bevinni is lehet!</p>
	Mozgáscsoportokban lehet váltani. Jelenleg nem használjuk.
	<p>Robot mozgási sebességének növelése a következő sorrendben: "VFINE", "FINE", "1%, ... 100%".</p> <p>5% alatt egy százalékával, 5% fölött öt százalékával.</p> <p>A "VFINE", "FINE" csak lépésenkénti módot biztosít!</p>
	<p>Robot mozgási sebességének csökkentése a következő sorrendben: "VFINE", "FINE", "1%, ... 100%".</p> <p>5% alatt egy százalékával, 5% fölött öt százalékával.</p> <p>A "VFINE", "FINE" csak lépésenkénti módot biztosít!</p>
	Bevitt adatok, parancsok elfogadása.
	Hibák és hibaüzenetek törlése.
	Visszatörlés a szövegben.
	A kurzort arra a sorra mozgatja, amely sorszámát beírjuk.
	JOINT koordináta rendszerben a hat ízület ($\pm J1 \dots J6$) forgatása.
	WORLD, TOOL, USER koordináta rendszerben X, Y, Z tengelyek menti mozgatás.

	WORLD, TOOL, USER koordináta rendszerben X, Y, Z tengelyek körüli forgatás.
	Egyszerű segítség jeleníthető meg,  -el riasztási sorokat bemutató képernyő jelenik meg.
	Megjeleníti a robotkar aktuális koordinátáit. Választható különféle koordináta-rendszer, sőt 4D is.
	Megjeleníti, (sőt beállítható) a robot I/O állapotait.
	Rengeteg féle robot paraméter kijelezhető (pl.: memória állapotok).
	Rengeteg féle beállítási állapot jeleníthető meg vele.
	Nem használatos.
	Meghívhat egy előzőleg általunk írt makro programot (pl.: megfogó nyitása).
	Meghívhat egy előzőleg általunk írt makro programot (pl.: megfogó zárása).

Ablakok beállítása a kezelőn

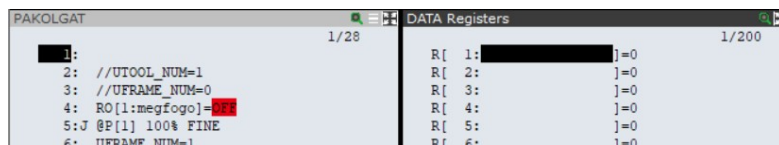
Ablakok számának beállítása



gombok együttes lenyomásával sok lehetőség közül választhatunk. Általában az egy (**Single**) és a két (**Double**) a legelterjedtebb.

DISPLAY 1	
1	Single
2	Double
3	Triple
4	TreeView
5	Single Wide
6	Double Horizontal
7	Triple Horizontal
8	Help/Diagnostics
9	EDIT<-->EDIT
0	-- NEXT --

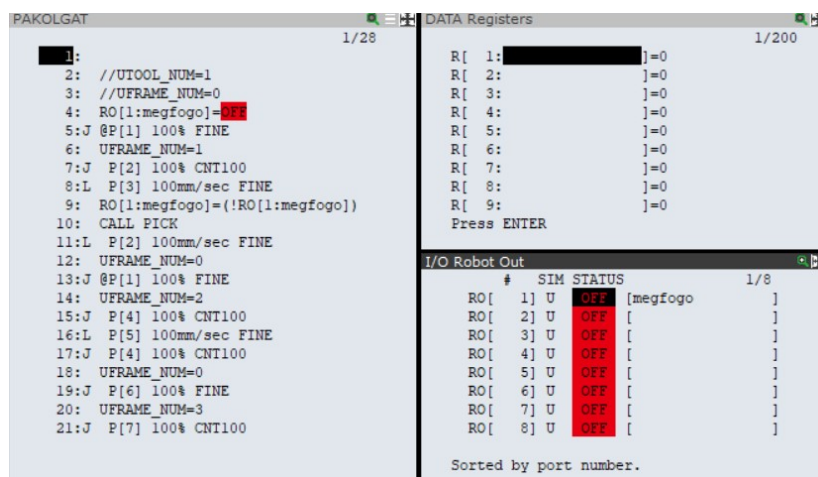
Két ablak esetén a baloldali ablakban futtathatjuk a programunkat, míg a másikban pl. ellenőrizhetjük a változók, regiszterek pillanatnyi értékeit.



Váltás az ablakok között, vagyis melyik legyen aktív



gomb többszöri lenyomásával. Az aktív ablak kerete sötétebb.

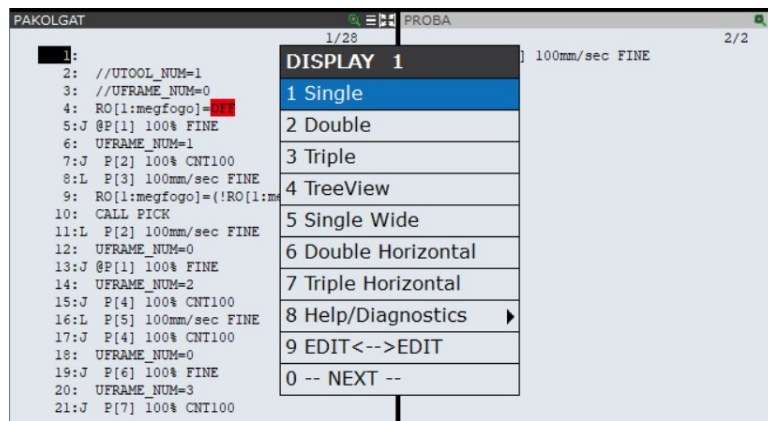


Csak a bal oldali ablak tartalma futtatható!



Váltás futtatható programok között


Ha mindkét ablakban futtatható program van, és mivel csak a bal oldalsó futtatható, de én a jobboldalit szeretném futtatni, meg kell őket cserélni!

Erre való a   hatására megjelenő ablak 9. pontja .

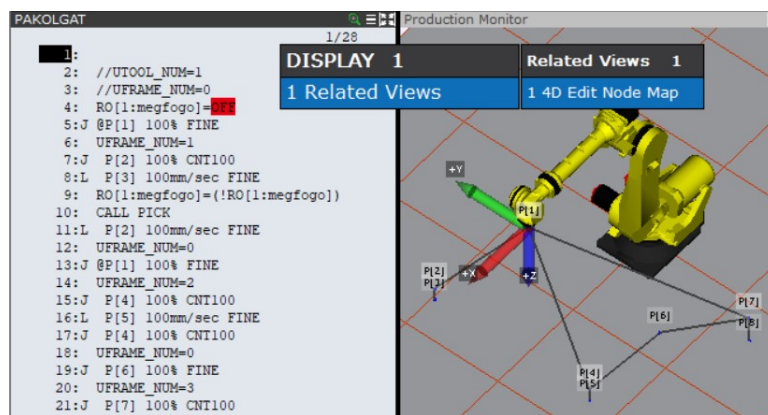




Egyéb megjelenítési lehetőségek

A kijelző jobb felső részén felváltva megjelenő ikonokkal   van lehetőség a betűméret növelésére, majd visszaállítására.

Az ikoncsoport  középső elemével előhozható egy helyi menü, ahol a „**4D Edit Node Map**”-ot választva, megjeleníthető a robot **4D** képe. A kép

 nagyítható-kicsinyíthető, mozgatható, forgatható.



A  ikon az aktív ablakot teljes képernyősre teszi, míg a helyette megjelenő  ikon újra felosztja a képernyőt.

Kezdeti beállítás (Initial Setup)

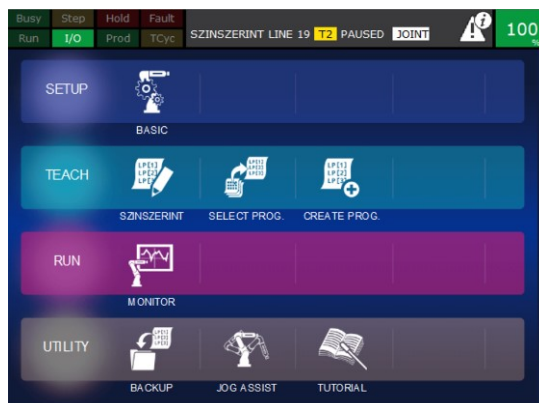
A kijelző bal alsó részén lévő ikonra  kattintva egy soklépéses varázsló indul el. Ezt új cella létrehozásakor célszerű használni.

Az üdvözlő képernyő és néhány figyelmeztetés után itt beállítható a régió (Amerika, Európa, Ázsia), az időzóna, a dátum, az idő, a hálózati beállítások, az alapértelmezett szerkesztő felület (egyszerű, vagy ikonos), továbbá olvasható egy egyszerű leírás az alapvető használatról.

Ezen túllépve az **End of Arm Tooling (EOAT)** beállításába léphetünk. Ebben olyan elemeket állít be, mint a szerszám típusa (megfogó vagy vákuum-tappancs) a szerszám-középpont pozíciója (TCP), a szerszám tömegközéppontja és az I/O jelek a robothoz csatlakoztatott szerszámhoz.



Befejezve a beállítást egy ikonos kezelőfelülethez jutunk, ahol a **SETUP** ikonnal visszakerülhetünk az előző felületre.



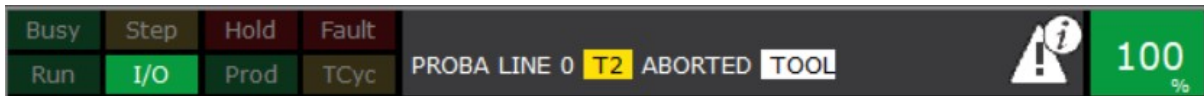
Legegyszerűbben kilépni belőle a **TEACH** sor adott programjára mutatóra, vagy a **SELECT PROG.** ikonra mutatóra lehet.



A Teach Pendant-on az  vagy a  gomb lenyomása ennél is jobb megoldás.

Állapot ablak a kezelőn

LED ikonok

A kijelző tetején lévő két sort állapotablaknak nevezzük. Ebben az ablakban nyolc visszajelző lámpa, riasztás jelzés, továbbá mozgási sebesség % jelenik meg.



- **Busy:** zölden világít, ha a robot működik.
- **Run:** zölden világít, ha a robotprogram fut.
- **Step:** sárgán világít, ha a lépésenkénti programfutás  kapcsolót bekapcsoltuk.
- **I/O:** zölden világít, alkalmazás-specifikus
- **Hold:** pirosan világít a programmegszakítás  gomb lenyomásakor.
- **Prod:** zölden világít, alkalmazás-specifikus
- **Fault:** pirosan világít hiba, riasztás esetén.
- **TCyc:** sárgán világít, alkalmazás-specifikus

Riasztás sor

A LED-ek utáni felső sor az aktuális riasztást mutatja. Példánkban a **DEADMAN** kapcsoló nincs középállásban.

SRVO-003 Deadman switch released


Státusz sor

PROBA LINE 0 T2 ABORTED TOOL

- Első helyen a betöltött program nevét látjuk (**PROBA**)
- Ezt követi, hogy hányadik programsorban áll a kurzor (**LINE 0**)
- Ezt követi a kiválasztott üzemmód (**T2**)
- Ezt követi a program jelenlegi állapota (**ABORTED**, vagyis megszakított)
- Ezt követi az aktuális koordinátarendszer (**TOOL**)

Villogó háromszög






Valamilyen hibára, hiányosságra figyelmeztet. A billentyűzet  gombját lenyomva tudhatunk meg részleteket.

Sebesség felülírás értéke

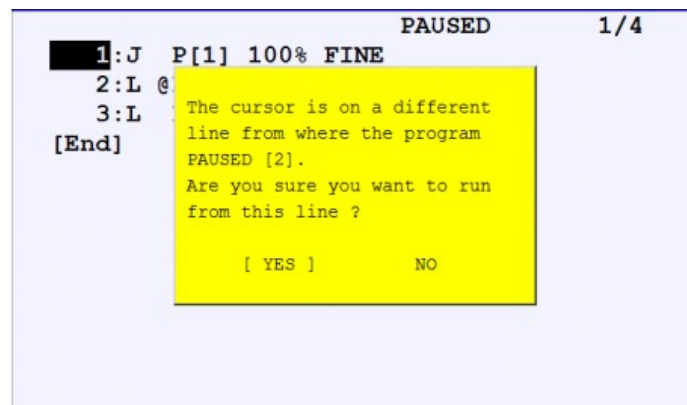
100%

A robotkar sebessége hány százalékra van lecsökkentve.






Figyelmeztető ablak

A sárga színű figyelmeztető ablakoknál nem lehet érintéssel választani, nyugtázni, helyette a kurzormozgató nyilakat   és az  kell használnunk.

Ez legtöbbször akkor látható, ha nem attól a sortól szeretnénk futtatni a programunkat, ahol legutóbb megszakítottuk.


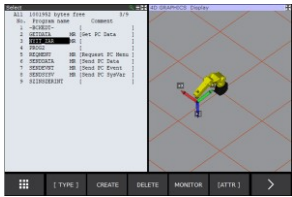
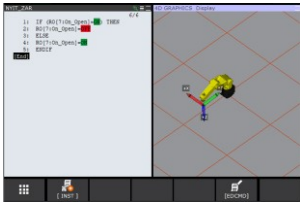


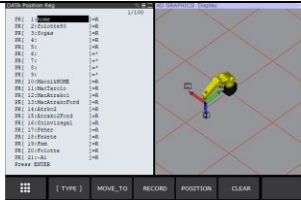
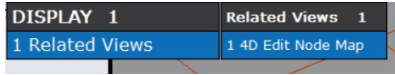
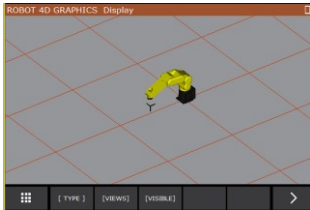
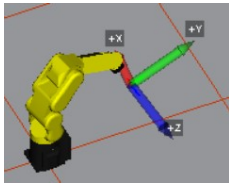


FCTN Menü

Jel	Funkció
ABORT ALL	Megszakítja az összes futó vagy szüneteltetett programot. Ha lefagyna a program, ez fontos!
Disable FWD/BWD	Letiltja a  +  vagy a  +  robotkarmozgatási lehetőségeket. Újra engedélyezéshez meg kell, nyomja az FCTN gombot, és ismét kiválasztani a Disable FWD/BWD lehetőséget.
RELEASE WAIT	Programfutáskor a programban lévő várakozási feltételek felülbírálása (amikor pl. a robot az I/O feltételek teljesülésére vár).
QUICK/FULL MENU	Vált a  lenyomásakor megjelenő teljes vagy rövidebb menülista között.
SAVE	Menti az aktuális programot, (amit egyébként automatikusan úgy is mentesz).
PRINT SCREEN	A Teach Pendant képernyőjét menti TPSCRN.LS néven az UD1:\ meghajtóba helyezett pendrive gyökér könyvtárába. Minden képernyőmentést ebbe a fájlba ír hozzáfűzéssel. Ezt megnyithatjuk pl.: Jegyzetömbbel. Ezzel egy időben ugyanide TPSCRN_DÁTUM_IDŐPONT.PNG néven képet is készít.

PRINT	Az aktuális teljes programot az alapértelmezett eszközre menti <i>program_name.ls</i> néven. Ezt megnyithatjuk valamilyen szövegszerkesztővel pl.: Jegyzetömbbel. A robot alapértelmezetten a UD1-be helyezett pendrive gyökér könyvtárába menti a *.ls fájlt.
UNSIM ALL I/O	Letiltja az összes szimulált I/O jelet.
CYCLE POWER	Újraindítja a vezérlőt. Ha a TP engedélyezve van, megjelenik egy panel, amely megkérdezi, hogy biztosan akarjuk-e.
ENABLE/DISABLE HMI MENUS	Engedélyezi vagy letiltja az opcionális HMI menüket TELJES és GYORS menü módban.
REFRESH PANE	Frissíti az oldalt az aktív ablakban.

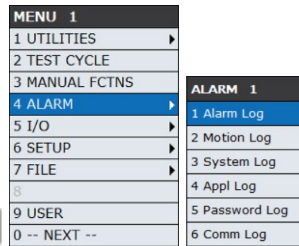
i +nyomógomb kombinációk

Kombináció	Funkció
i + MENU	Megjelenít egy ikonos rendszerű több ablakos menürendszert 
i + SELECT	A bal oldali ablaktáblában a SELECT menüt, a jobb oldali ablaktáblában pedig a 4D képet jeleníti meg. 
i + EDIT	A bal oldali ablaktáblában az EDIT képet, a jobb oldali ablaktáblában pedig a 4D képet jeleníti meg. 
i + DATA	A bal oldali ablaktáblában a DATA képernyőt, a jobb oldali ablaktáblában pedig a 4D képet jeleníti meg.

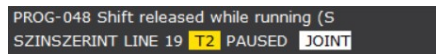
	
<p>① + FCTN</p>	<p>Megjeleníti a kapcsolódó nézet menüt, ha elérhető.</p> 
<p>① + POSN</p>	<p>A 4D csomóponttérképen jeleníti meg a robot helyzetét.</p> 
<p>① + Mozgatógombok</p>	<p>4D képernyőn láthatók a robotkar mozgásai.</p> 
<p>①</p>	<p>Ha az állapotsorban villog a  jelző, akkor a gomb megnyomásával és elengedésével megjelenik az ÁLLAPOT - értesítések képernyő.</p> 

Rendszerüzenetek, naplók, riasztások

Többféleképpen tudjuk megnézni a különféle naplózott eseményeket.



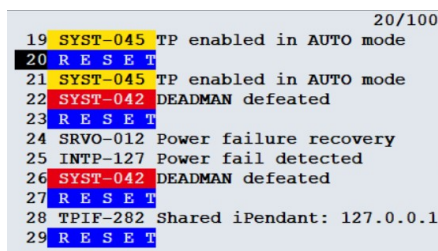
Az egyik módszer a **MENU** gomb útvonala, a másik a Teach Pendant értesítési területére koppintás (Roboguid-ban bal kattintás).



Ha éppen nincs aktív riasztás a **There are no active alarms.** üzenetet kapjuk.

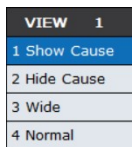
Az **[TYPE]**, **[VIEW]**, **HIST**, **RES_1CH** menüből választhatunk. A **HIST** időrendben mutatja a bejegyzéseket. A **ACTIVE** az aktív riasztásokat mutatja, a **CLEAR** törli azt a sort, amin állunk.

Az időrendi listában láthatjuk azt is, mikor nyomtuk le a **RESET** gombot.



A **[TYPE]** ugyanazt a választékot hozza, mint ha a **MENU** módszert választottuk volna.

- 1 **Alarm Log**: az éppen aktív riasztásokat mutatja
- 2 **Motion Log**: a mozgással kapcsolatos riasztásokat mutatja
- 3 **System Log**: a rendszerszintű riasztásokat mutatja
- 4 **Appl Log**: az applikációk riasztásait mutatja
- 5 **Password Log**: a jelszó kezelés riasztásait mutatja
- 6 **Comm Log**: a kommunikációs csatornák riasztásait mutatja



A **[VIEW]** ikon lenyomásával választhatunk lássuk-e az esemény dátumát vagy nem, széles, vagy normál megjelenítés legyen-e.

A **DETAIL** ikonnal megnézhetjük a részleteket, még egyszer lenyomva visszatérhetünk a listához.

A leggyakoribb hibakódok és jelentésük

INTP-105 (%s,%d) Run request failed	A program nem indítható el
INTP-127 power failure detected	Áramszünet érzékelve
INTP-248 (%s, %d) MACRO failed	A makró parancs nem hajtható végre
INTP-250 (%s, %d) Invalid uframe number	Az UFRAME száma nem érvényes
INTP-251 (%s, %d) Invalid utool number	A munkaeszköz száma érvénytelen
INTP-252 User frame number mismatch	Felhasználói keretszám nem egyezik
INTP-253 Tool frame number mismatch	Szerszám keretszám eltérés
INTP-447(%s,%d) Parentheses mismatch	Rossz zárójel kombináció
JOG-007 Press shift key to jog	A léptetéshez nyomja meg a shift gombot!
JOG-010 Jog pressed before shift	A robot egyik kézi mozgatósi billentyűjét a SHIFT billentyű lenyomása előtt lenyomták
MACR-013 MACRO execution failed	A makró nem hajtható végre
MACR-016 The macro is not completed	A makró végrehajtása megszakadt
MOTN-017 Limit error (G:1, A:1)	A megadott pozíció kívül esik a közös mozgatható tartományon
MOTN-018 Position not reachable	A pozíció nem érhető el
MOTN-019 In Singularity	A kért pozíció közel van a szingularitáshoz (J4 és J6 párhuzamos)
MOTN-063 Position config change (G :)	A robot nem tud elmenni a kívánt pozícióba.
PROG-048 Shift released while running (prog)	A SHIFT billentyűt felengedték, miközben a program T1 vagy T2 módban futott
SRVO-003 or SRVO-234 Deadman switch release	Deadman kapcsoló elengedve
SRVO-062 BZAL alarm (G :x A :x) or SRVO-084 BZAL alarm	A jeladó akkumulátorai alacsony feszültségűek, vagy a kódoló kábel le van választva
SRVO-012 power failure recovery	Áramszünet helyreállítása
SRVO-080 OVC alarm	A szervó hajtás által számított áram meghaladja a megengedett értéket
SRVO-160 Panel/external E-stop	A vezérlő vészleállító gombját vagy a külső vészleállító gombot megnyomták
SRVO-161 Fence open or Deadman switch released	A Deadman gombot elengedték, vagy a robot kerítése nyitva van
SRVO-162 Deadman/Fence or Panel/External E-stop	A Deadman gombot elengedték, vagy a robot védőburkolata (kerítés) nyitva van, vagy a vészleállító gombok valamelyikét megnyomják
SRVO-230 or SRVO-231 Chain (1 or 2) abnormal	Megszakadt a vészleállító áramkör 1. csatornája (24 V) vagy 2. csatornája (0 V)
SRVO-324 Disturbance excess	Elérte egy adott csukló nyomatékhatárát
SYST-003 TP is enabled	A TP engedélyezve van
SYST-009 Safety Fence is open	A biztonsági kerítés nyitva van
SYST-017 Single-step operation effective	A STEP mód be van kapcsolva
SYST-042 Deadman defeated	Deadman gomb benyomva
SYST-043 TP disabled in T1/T2 mode	A TP le van tiltva van T1/T2 módban
SYST-045 TP enabled in AUTO mode	A TP engedélyezve van AUTO módban

A Roboguide szoftver



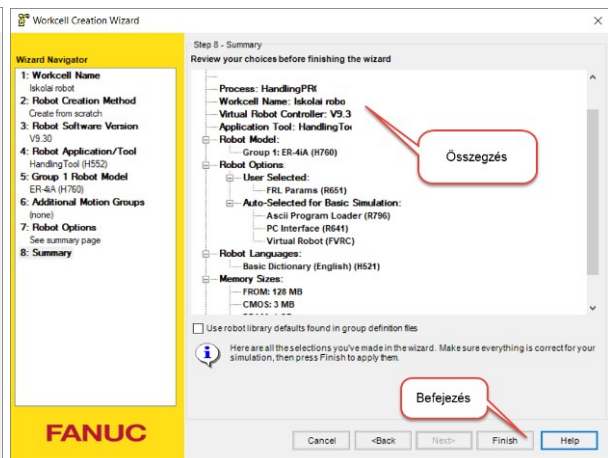
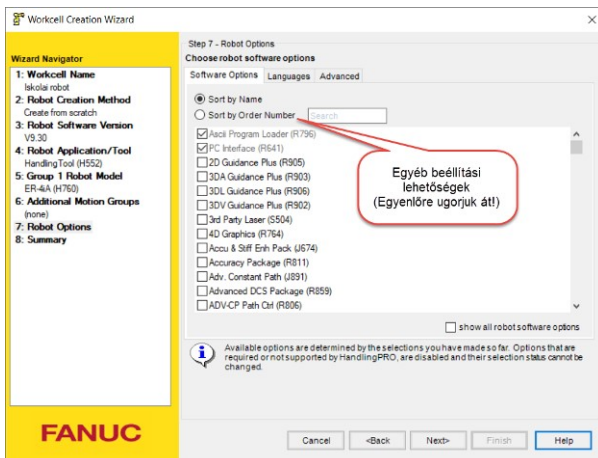
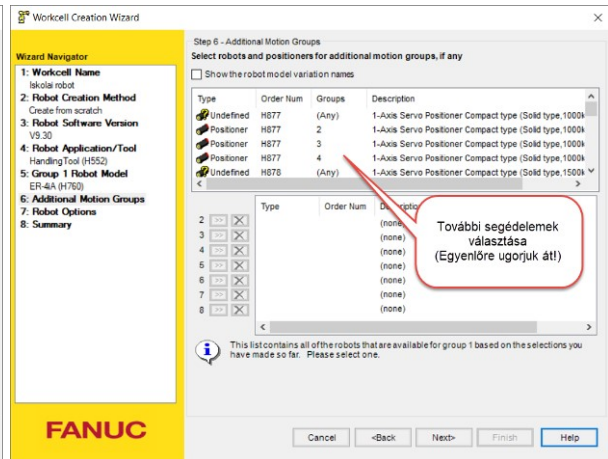
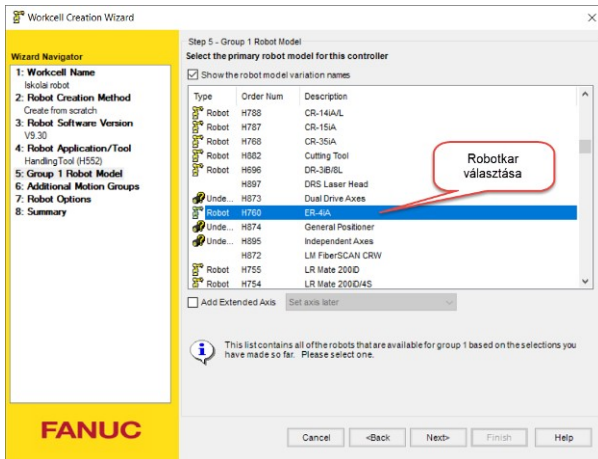
A Roboguide szoftver egy olyan szimulációs program, ami segít a robotprogramozás tanulásában. Megírhatjuk benne a programjainkat virtuális **Teach Pendant** segítségével, szimulálhatjuk a robotcellát, a mozgásokat, a programunkat, illetve le is menthetjük művünket pl.: USB tárolóra. Ezt az USB tárolót áttéve a valós robot UD1: meghajtójába, betölthetjük programunkat, vagyis **offline** programozhatunk. Azonban **minden esetben újra kell tanítanunk a kereteket (userframe, toolframe) a pozíciókat, pozícióregisztereket!**

Új cella létrehozása

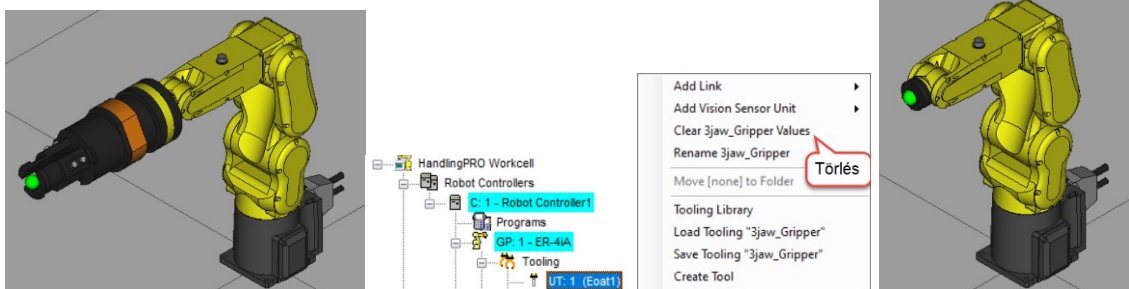
Új robotcella készítése a **File > New Cell** útvonalon, vagy a **New Cell** ikon segítségével lehetséges, ami elindít egy létrehozási varázslót.

The image displays the FANUC Workcell Creation Wizard software interface, showing four sequential steps in the wizard:

- Step 1 - Workcell Name:** The user enters a name for the new workcell (e.g., "HandlingPRO3"). The interface shows a list of existing workcells and a "Delete Item" button. Callouts indicate: "A varázsló lépései" (Wizard steps), "A felajánlott új programnév, ami átírható" (Suggested new program name, which can be changed), "Az eddig létrehozott robotcellák" (Previously created robot cells), "A kijelölt cella törlése" (Deletion of the selected cell), and "A következő lépés" (Next step).
- Step 2 - Robot Creation Method:** The user selects the method to create a robot. Options include creating a new robot with default, last used, or backup configurations, or creating an exact copy of an existing robot. Callouts indicate: "Az új név" (New name), "Új cella alapbeállításokkal" (New cell with default settings), "Új cella a legutóbbi beállításokkal" (New cell with last used settings), "Új cella mentés fájlból" (New cell from backup file), and "Másolat egy meglévő robotból" (Copy of an existing robot).
- Step 3 - Robot Software Version:** The user selects the software version to be loaded on the robot. A list of versions is shown, with "V8.30- R-30B Plus" selected. Callouts indicate: "Vezérlőszoftver választás" (Control software selection).
- Step 4 - Robot Application/Tool:** The user selects the Application/Tool package to be loaded. Options include Default Eoat, CAD Library, and various tool packages. Callouts indicate: "Szerszámválasztás" (Tool selection).




Alapbeállításokkal egy eléggé túlméretezett háromujjas megfogót kapunk. A baloldali fa-struktúrában (**UT:1(Eoat1)**) ezt meg is találhatjuk. Jobb egérgomb helyi menüjében törölhetjük

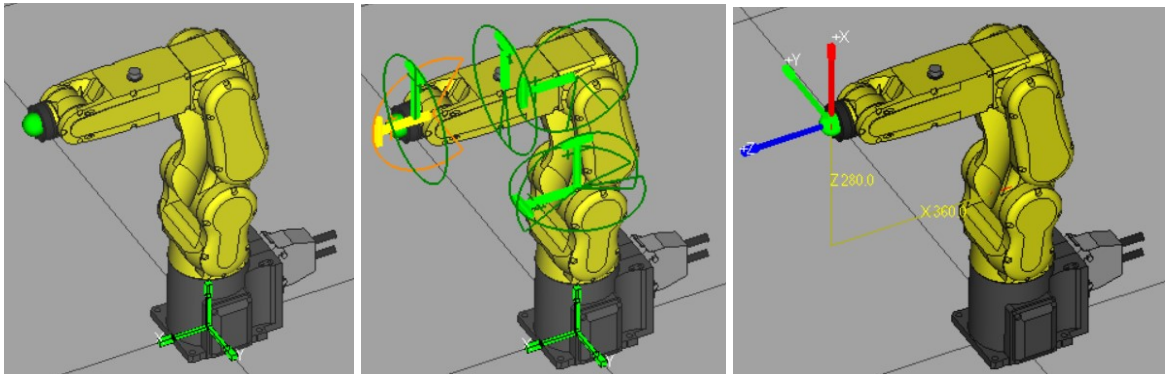


A varázsló negyedik lépésében a **Set Eoat later** választásával szerszám nélküli robot lesz a kiindulás. A kis zöld gömb a robotkar végén a **TCP-t** jelöli.



A robot mozgatása

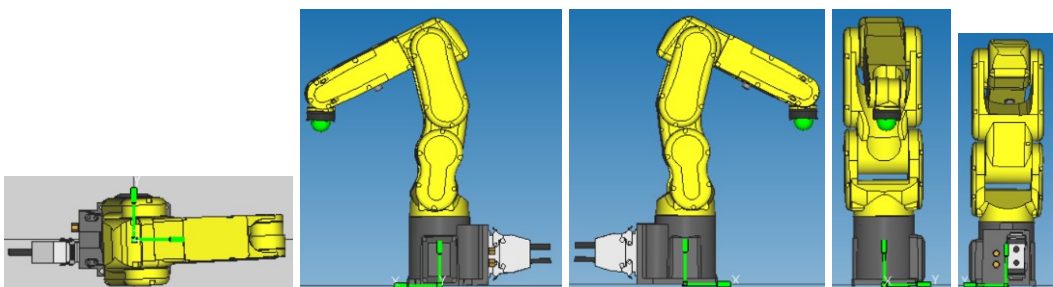
Joint Jog Tool ikont  bekapcsolva, és a robotra kattintva a robotkar forgástengelyei körül mozgatható a kis zöld kalapács-szerű íveket egérrel megragadva. A sárga színű ív és kalapács szingularitási hibát jelez.

A **Joint Jog Tool** kikapcsolása után a **TCP-t** jelképező kis zöld gömbre kattintva megjelennek a derékszögű koordináta tengelyek, amelyeknél fogva mozgatható, forgatható a robotkar.



Nézetek

 ikonok a legfontosabb nézeteket adják. Az első az éppen kijelölt objektumot a képernyő közepére hozza. A harmadik fölülnézet +**Z**, a negyedik jobbnézet +**Y**, az ötödik balnézet -**Y**, hatodik előlnézet +**X**, a hetedik hátulnézet -**X** képet ad. Menüből még egy izometrikus térbeli nézet is választható  **Isometric** Pontos beállításokhoz a nézeti kép jobb, mint a 3D.



Szingularitás

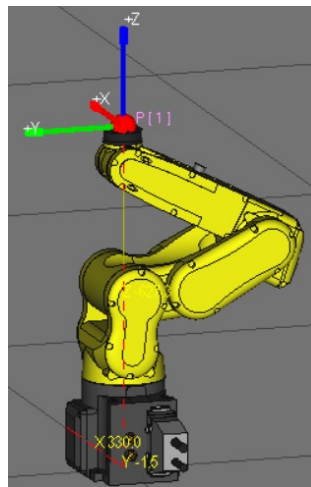
Egy megadott pontban a robot két vagy több tengelyének tengelyvonala egybeesik, ami kiszámíthatatlan robotmozgást és sebességeket eredményez. **MOTN-023** In singularity

Ebből a kezelő közreműködése nélkül nem tud kilépni, ezért a robotkar megáll.

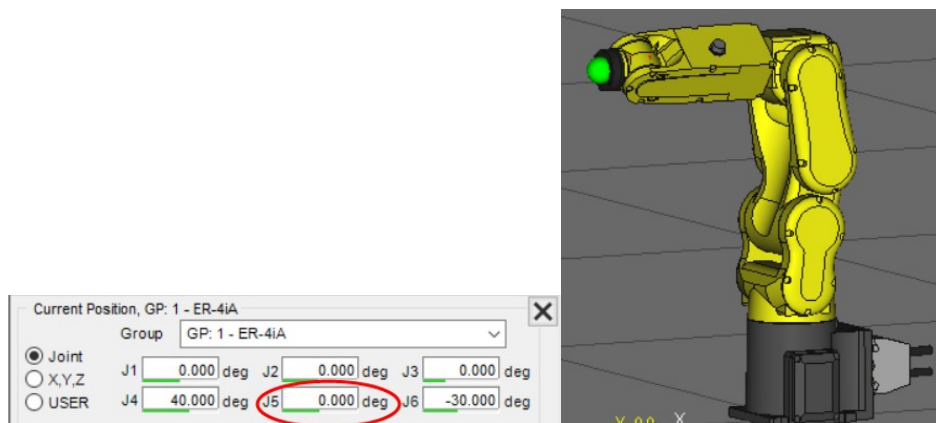
A robotnak kétféle szingularitása van:

- Kar szingularitás
- Csukló szingularitás

A kar szingularitás akkor következik be, ha a hatodik tengely középpontja egy vonalba esik az első tengely vonalával.



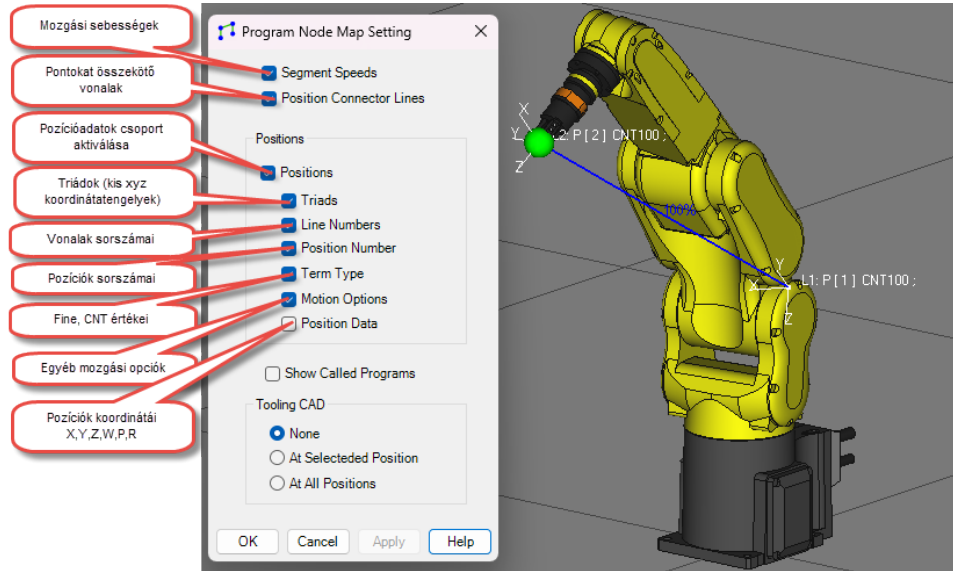
A csukló szingularitás abban az esetben fordul elő, ha a 4. és a 6. tengelyek vonala esik egy egyenesre, **vagyis az J5 0°-ban van.**



Megoldás: J5 értékét állítsuk el (pl.: 6°-al), és oda rögzítsünk pontot!

Program csomópont térkép beállításai (Program Node Map Settings)

A Program Node Map panelon beállíthatjuk, milyen adatok, segédvonalak jelenjenek meg a robotkar mellett. Elérés: **View > Program Node Map**



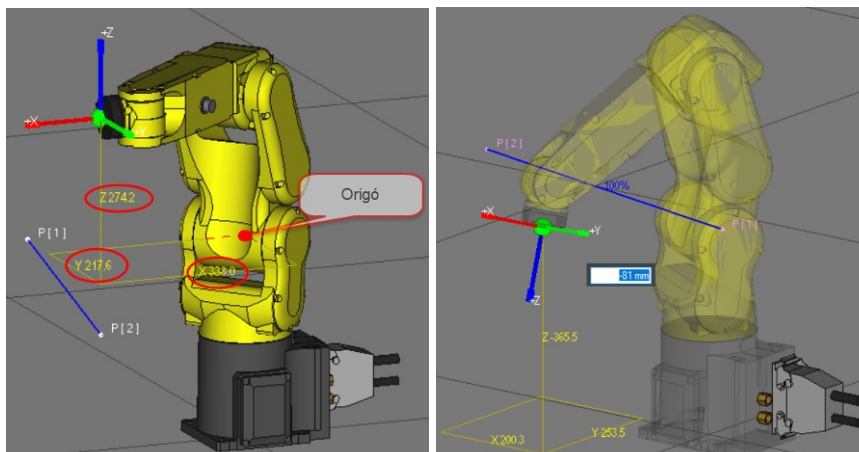
A robotkar mozgása a TCP pont vonszolásával

Úgy kell rákattintani a TCP-t jelölő zöld kis gömbre, hogy a színes koordinátatengely-hármas jelenjen meg.

Vékony sárga kerettel és a koordináták megjelenítésével láthatjuk, hol van a TCP a választott origóhoz képest.

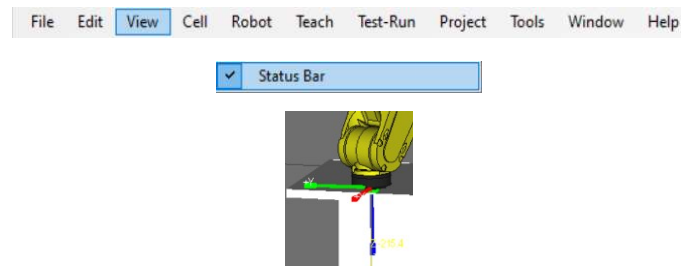
A **bal** egérgombbal megfogva valamelyik tengely közepét, a jelölt **X, Y, Z** az irányba mozgatható, a tengelyek végét megfogva a jelölt **X, Y, Z** tengelyek körül forgatható a TCP. Vonszolás közben az elmozdulás **mm-ben**, az elfordulás **°-ban** egy ablakban olvasható, a **bal** egérgombot nyomva és egy értéket beírva, **Entert** ütve pontosan állítható.

A **CTRL + bal** egérgomb segítségével mindhárom irányba egyszerre elmozdítható a TCP.

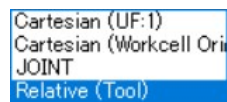


TCP mozgatás pontos értékek bevitelével

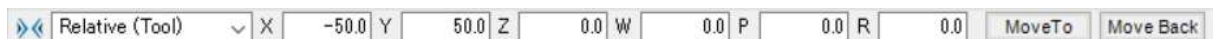
Ha be van kapcsolva a **View** menüpontban a **Status Bar** és a **TCP** van kijelölve, alul az értesítési területen közvetlenül megadhatjuk, hová mozduljon a **TCP**.



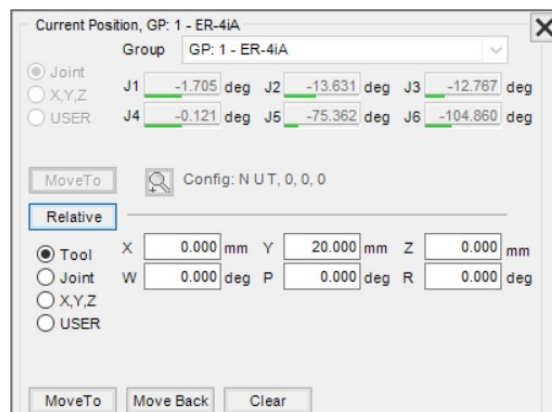
Választhatunk különböző koordináta-rendszerek között.



Ha az elmozdulások növekményeit ismerjük vagy kiszámítjuk, válasszuk a „**Relative (Tool)**” lehetőséget! Beírva az értéket a „**MoveTo**” gombbal oda tudjuk mozgatni a **TCP**-t, ha hibáztunk, a „**Move Back**” gombbal visszafelé is tudunk mozogni. Ha beállítottuk a pont megfelelő helyzetét, a **Teach Pendant-on** rögzíthetjük azt.



Ugyanezek a beállítások elérhetőek a **Teach Pendant**  gombjának lenyomásával is. Így esetleg könnyebb rögzíteni a kiválasztott pontot.



Előzőleg felvett pontok vonszolása

Amennyiben a **Program Node Map** felületen bekapcsoltuk a **Position Connector Lines**, a **Positions**, és a **Position Number** visszajelzését, ki tudjuk jelölni a pontokat jelölő kis pöttyöket.

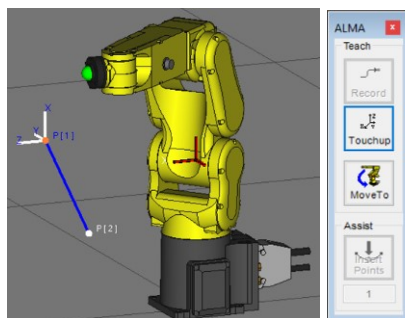
Megragadva a kis tengely-jeleket (triádokat) **x,y,z** irányban, míg **CTRL + bal** egérgombbal tetszőleges helyre vonszolhatjuk a pontokat. Ezzel a programban lévő elmozgatott pontok koordinátái is változnak!

Ha a tengelyhármas vonszolás közben piros színű lesz, oda már nem ér el a robotkar!

Figyelem! A pontok vonszolásával csak a helyzetüket változtatjuk, az **orientációjukat nem**, a **SHIFT** billentyű lenyomásával tudjuk a pontot **xyz** tengelyek körül forgatni.

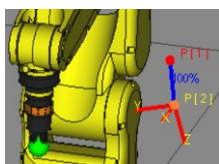
A **View > Quick Bars > Teach** panelt bekapcsolva, ha kijelölünk egy pontot, a **MoveTo** gomb lenyomásával odamozgathatjuk a robotkart.

A **Tachup** gombbal a kijelölt pontot áthelyezhetjük az éppen aktuálisan álló robotkar **TCP** pontjába. **Teach Pendant** alapú programozásnál a **Record** és az **Assist** nem aktív.



Pont orientációjának beállítása


Ha kijelölünk egy már felvett pontot (a példában **P[2]**) akár a robot terében



, akár a **Teach Pendant** képernyőjén

```
1:J P[1] 100% FINE
2:J P[2] 100% FINE
[End]
```

a sorszámra

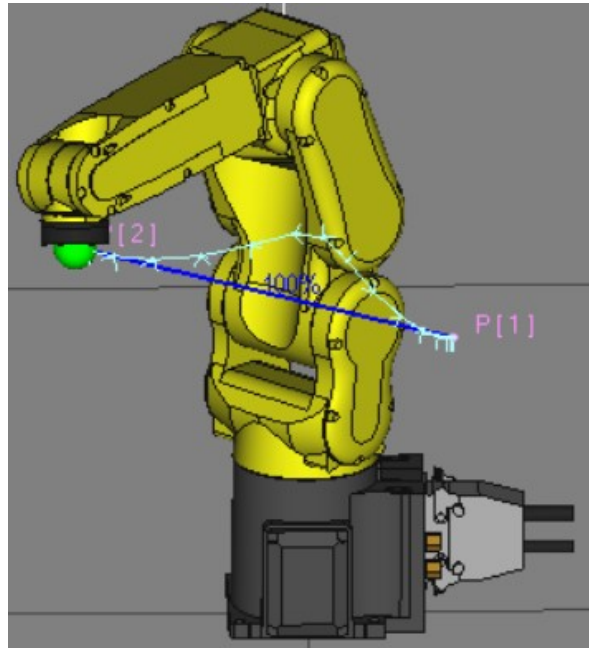
kattintva, megjelenik a **Teach Pendant** alján a **Position**  ikon. Erre kattintva megjelennek a kijelölt pont koordinátái, és orientációi, és **át is írhatók**.

Koordináták		Orientációk	
P[2]	UT:1	CON	FUT 000
X	71.506 mm W	1	.140 deg
Y	239.283 mm P		.000 deg
Z	-466.644 mm R		-180.000 deg
Position Detail			
1:J	P[1]	100%	FINE
2:J	P[2]	100%	FINE
[End]			

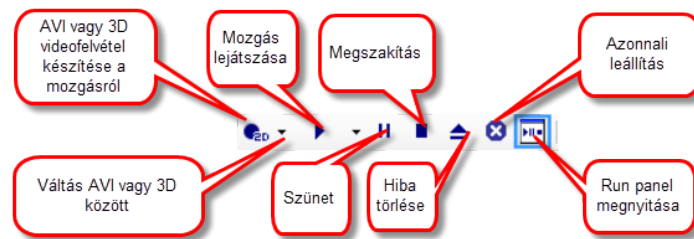
Roboguide RUN Panel

Segítségével lejátszhatunk egy robotmozgást, illetve 2D *.avi, vagy 3D *.rg3d felvételt készíthetünk róla. A 2D felvétel a legtöbb ismert video lejátszóval, míg a 3D felvétel a **Roboguide 3D Player** lejátszóval utólag visszanezhető.

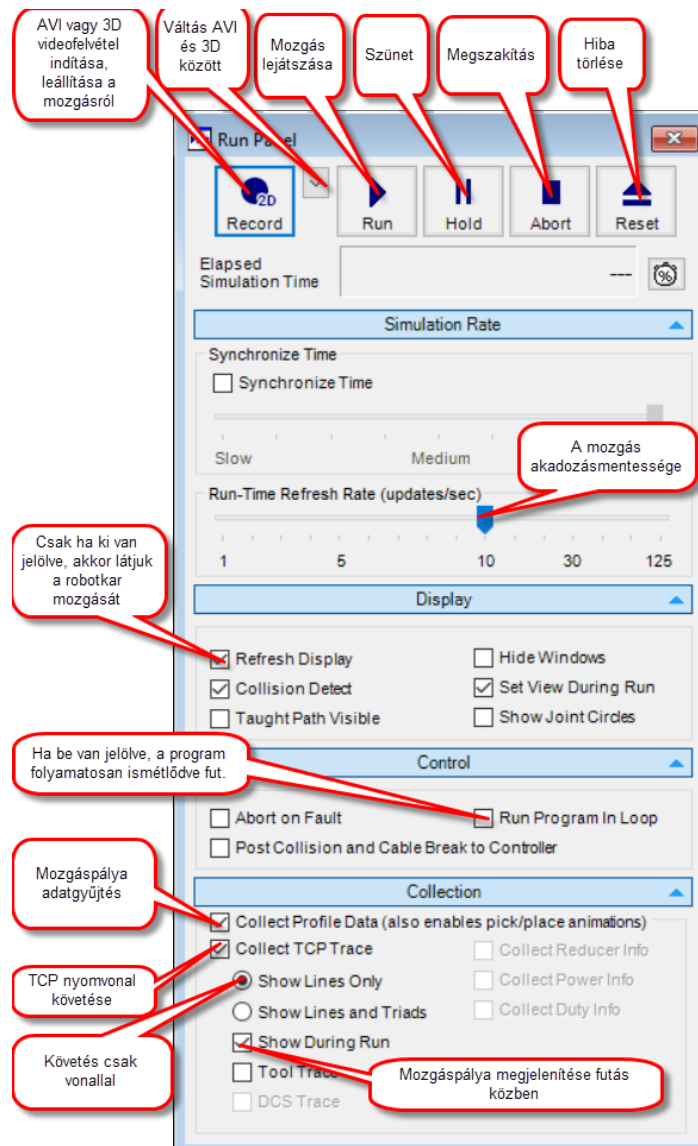
Az alábbi ábrán a **Roboguide RUN Panel** segítségével láthatunk egy lejátszási képet. Megfelelő beállítások mellett láthatóak a **pontok**, az őket **összekötő egyenesek**, és „kiszűrött” vonallal a **mozgáspálya** képe.



A **Roboguide** program ikonsorában találhatóak a **RUN Panel** legfontosabb kezelőgombjai az alábbiak szerint:



A „**RUN panel megnyitása**” ikonnal egy részletesebb, beállítási lehetőségeket is tartalmazó panel nyitható meg az alábbiak szerint. Itt magyarázatokkal láthatjuk el a legfontosabb beállítási lehetőségeket.



Roboguide 3D Player

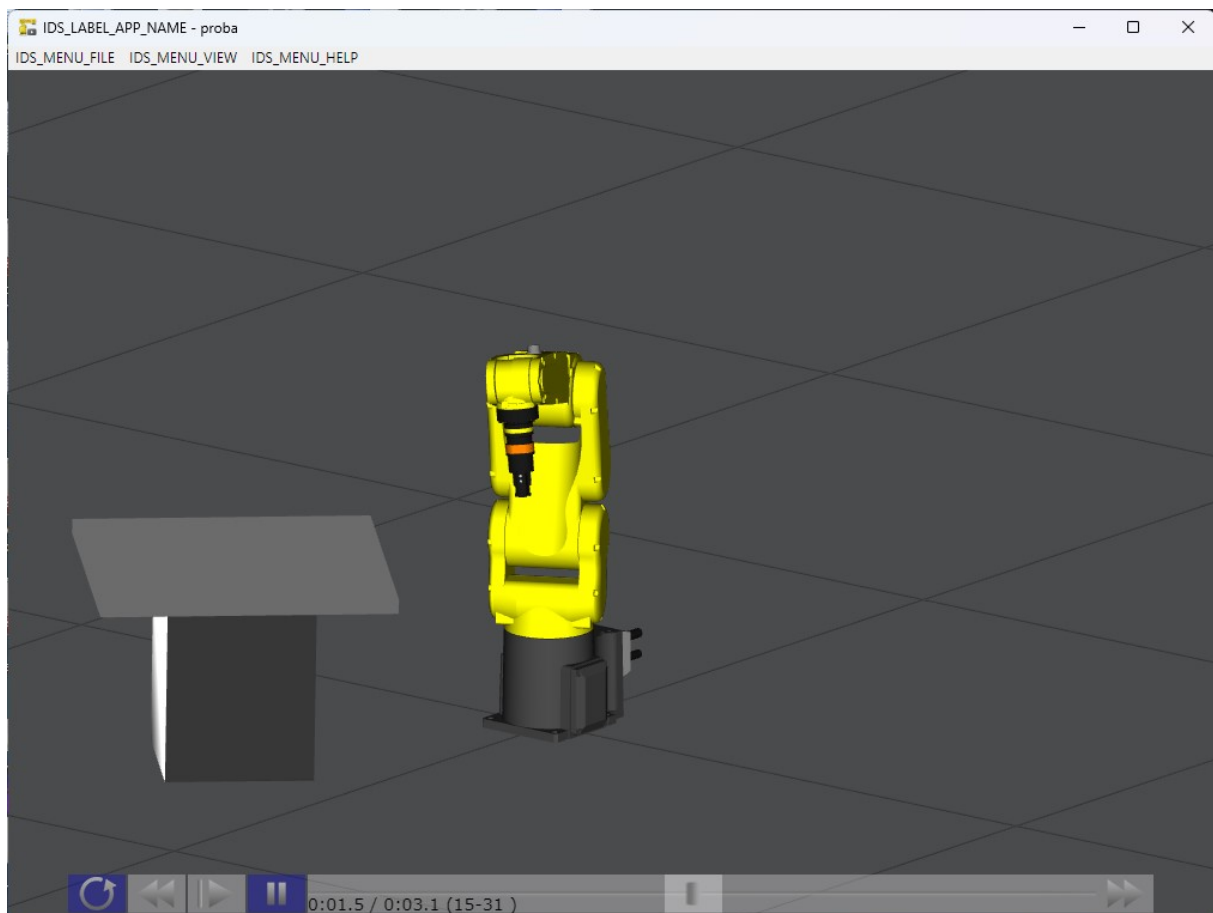


A szoftvercsomag része, de külön program a **Roboguide 3D Player** alkalmazás, amit elindítva kitallózhatjuk, és lejátszhatjuk 3D robotmozgás-felvételeinket. Csak a mozgást láthatjuk, segédinformációkat nem, de térben tudjuk forgatni, nagyítani, kicsinyíteni a cellát.

A fájlok helye, kiterjesztése:

C:\Users\Felhasználónév\Documents\MyWorkcells\Robotprogram_fájlnév\RG3DPlayer\Felvétel_egyediazonosító*.rg3d

Az alábbi ábrán a **Roboguide 3D Player** segítségével láthatunk egy lejátszási képet.



Egérfunkciók

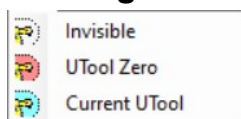


íkonnal bekapcsolható és kikapcsolható úszó panel, mint segítség.

3D World Mouse Commands		
View Functions	Object Functions	MoveTo Functions
Rotate view: RIGHT Drag	Move object, one axis: LEFT Drag triad axis	Move robot to surface: [CTRL] + [SHIFT] + LEFT-Click
Pan view: [CTRL] + RIGHT Drag or MIDDLE Drag	Move object, multiple axes: [CTRL] + LEFT Drag triad	Move robot to edge: [CTRL] + [ALT] + LEFT-Click
Zoom in/out: BOTH Drag (mouse Y axis) or Rotate the wheel	Rotate object: [SHIFT] + LEFT Drag triad axis	Move robot to vertex: [CTRL] + [ALT] + [SHIFT] + LEFT-Click
Select object: LEFT-Click	Object property page: DOUBLE-LEFT Click	Move robot to center: [SHIFT] + [ALT] + LEFT-Click

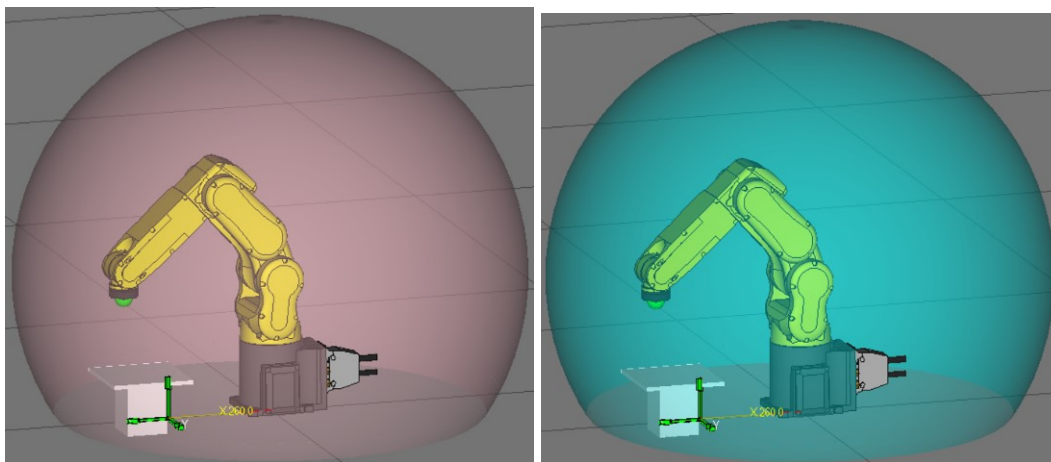
Nézeti kép funkciók		Objektum (pont tiád) funkciók		Mozgatás funkciók	
3D forgatás	JOBB vonszolás	Mozgatni, forgatni objektumot egy tengely mentén	BAL vonszolás a tengelyt (mozgat) vagy tengelyvéget (forgat) megragadva	Objektum felületére mozgatni a TCP-t	CTRL+SHIFT+BAL klikk az objektumon
Föl, le, balra, jobbra	CTRL+JOBB vonszolás vagy GÖRGŐ lenyomás és vonszolás	Mozgatni objektumot több tengely mentén	CTRL+BAL vonszolás a tengelyt megragadva	Objektum élére vagy átlójára mozgatni a TCP-t	CTRL+ALT+BAL klikk az objektumon
Nagyít, kicsinyít	GÖRGŐ forgatás	Forgatni objektumot egy tengely körül	SHIFT+BAL vonszolás a tengelyt megragadva	Objektum csúcsára mozgatni a TCP-t	CTRL+ALT+SHIFT+BAL klikk az objektumon
Kijelölés	BAL klikk	Objektum tulajdonságai	Dupla BAL klikk	Objektum közepére mozgatni a TCP-t	CTRL+ALT+BAL klikk az objektumon

Robot-mozgáster megjelenítése



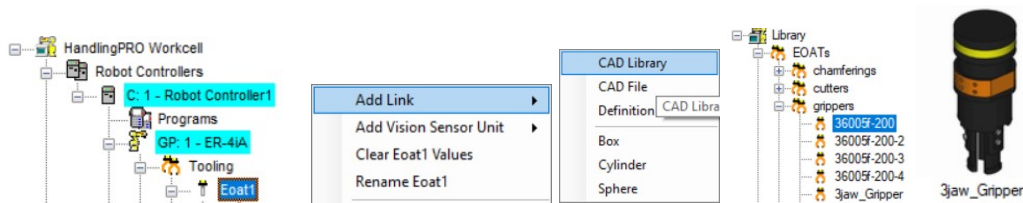
íkonok segítségével.

A fehér jel nem mutat mozgásteret, a rózsaszín a szerszám nélküli, a türkiz a szerszámmal együtt kialakuló mozgásteret mutatja.



Új szerszám felvétele

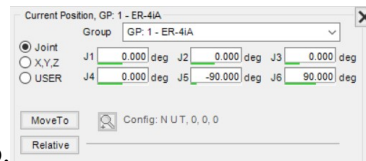
Új szerszám felvételéhez jobb egérgombbal kattintás az **Eoat1**-en, majd az alábbi kiválasztás.



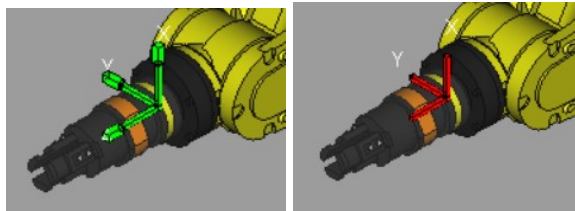
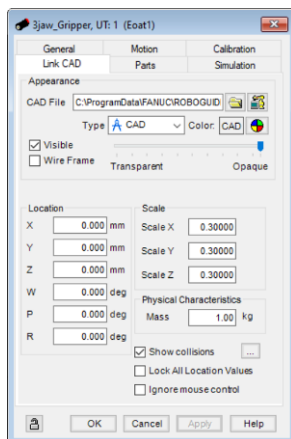
Ha elmozdítottuk volna a virtuális **Teach Pendant**on állítsuk a robotot alapállásba!

POSN

gomb, majd beállítás, majd **MoveTo** gomb.

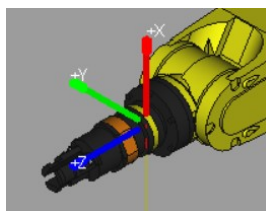


Elsőre általában sem a méret sem, a pozíció nem megfelelő. Kattintunk kettőt a megfogón, a felbukkanó panelon a **Link CAD** fülön állítsuk a méretet minden irányban 0.3-ra, a tömeget 1 kg-ra, a pozíciót pedig úgy, hogy középen, egyenesen álljon a szerszám! A háromujjas megfogón a középső rész és a 3 db megfogó elemet külön-külön kell lekicsinyíteni!

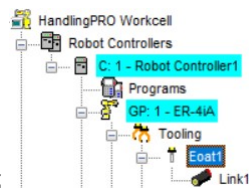


Ha pontosan beállítottuk a szerszámot, pipáljuk ki a „**Lock All Location Values**” négyzetet, nehogy később véletlenül elmozdítsuk! Ekkor a zöld tengelyek pirossá válnak. Szintén nem csak a középső részen, hanem az ujjakon is meg kell ezt tenni!

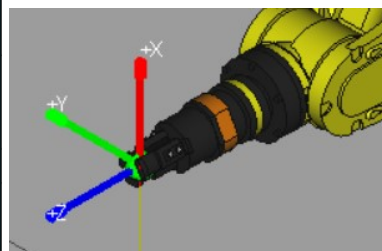
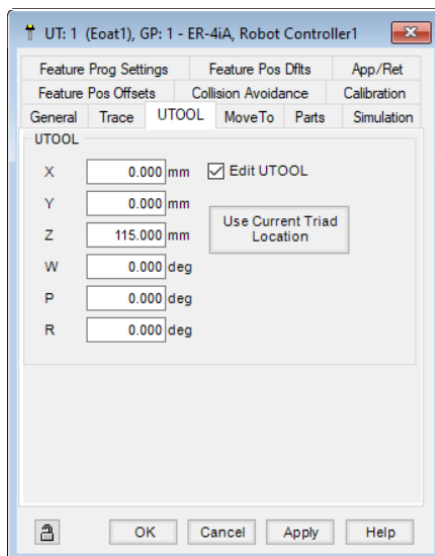
A szerszámon egyet kattintva a mozgatás a tengelyek húzásával történik **XYZ** irányokban, és a tengelyek végét megfogva forgathatjuk **XYZ** tengelyek körül.



TCP beállítása

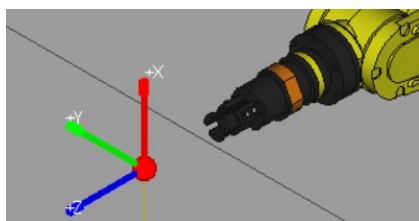


Kettőt kattintva az **Eoat1**-en, megnyílik egy beállító panel. Az **UTOOL** fülön ki kell pipálni az **Edit UTOOL** jelölőnégyzetet, **Z** mögé beírni a kívánatos értéket (**115 mm**), majd **Apply**!



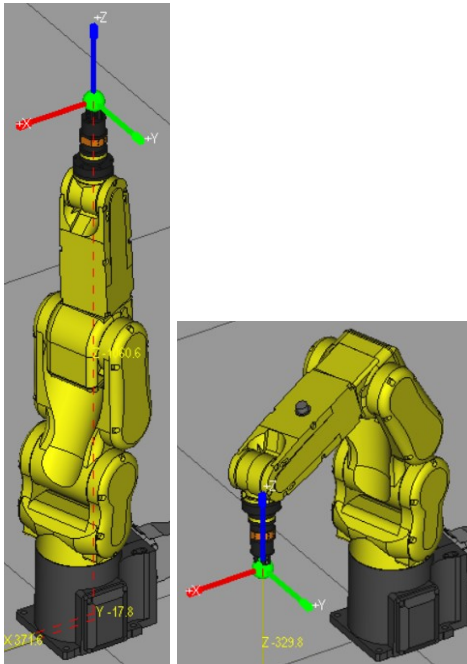
Megjelenik a **TCP**-t jelölő zöld gömb, továbbá a koordináta tengelyek, amelyeket megfogva tudom mozgatni a robotkart!

Ha olyan helyre vonszolom a robotkart, amit már nem ér el, a **TCP** piros lesz! Ilyenkor toljuk vissza, amíg ismét zöld színű nem lesz!



WORLD koordináta-rendszer

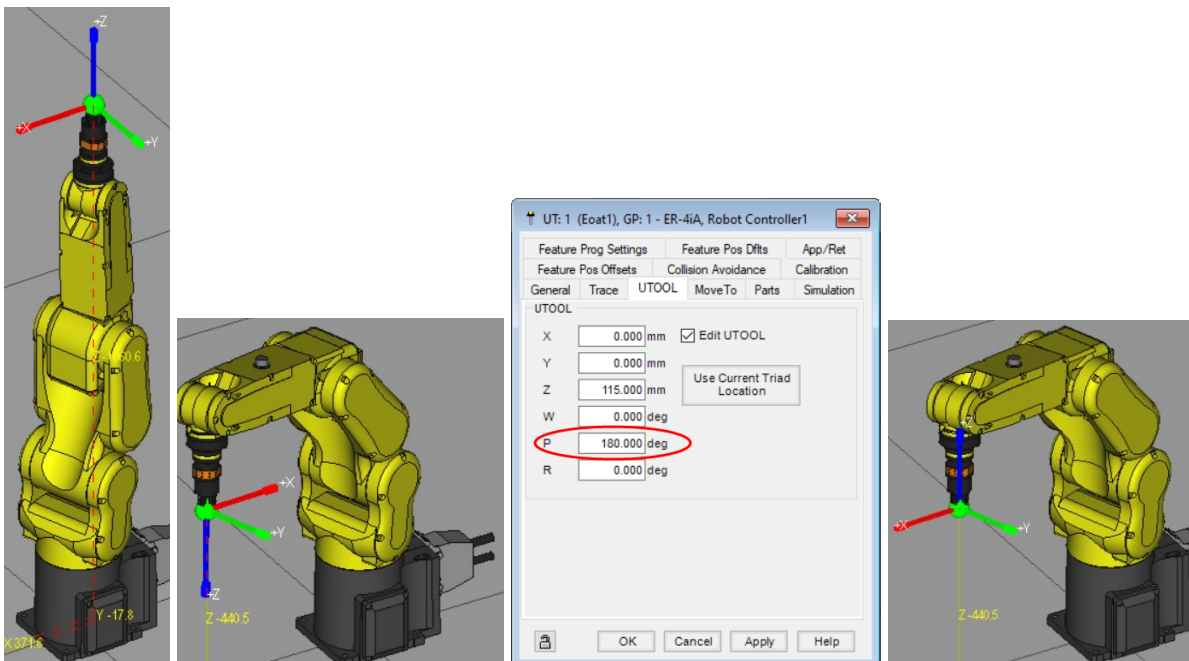
Bárhogy áll a robotkar, a koordináta tengelyek mindig ugyanarra mutatnak



TOOL koordináta-rendszer

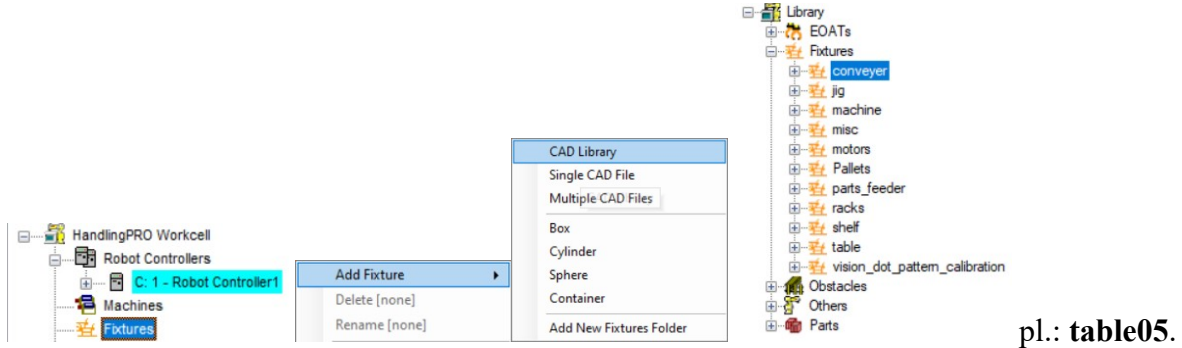
Felfelé mutatva a robotkar, a koordináta tengelyek ugyanúgy állnak, mint **WORLD** esetén. Lefelé mutatva viszont a **Z** és az **X** orientációja a robotkarral együtt megfordul.

Ha viszont **P** értékét **180°**-ra állítjuk, lefelé mutatva ugyanazok az irányok maradnak, mint a **WORLD** esetén.

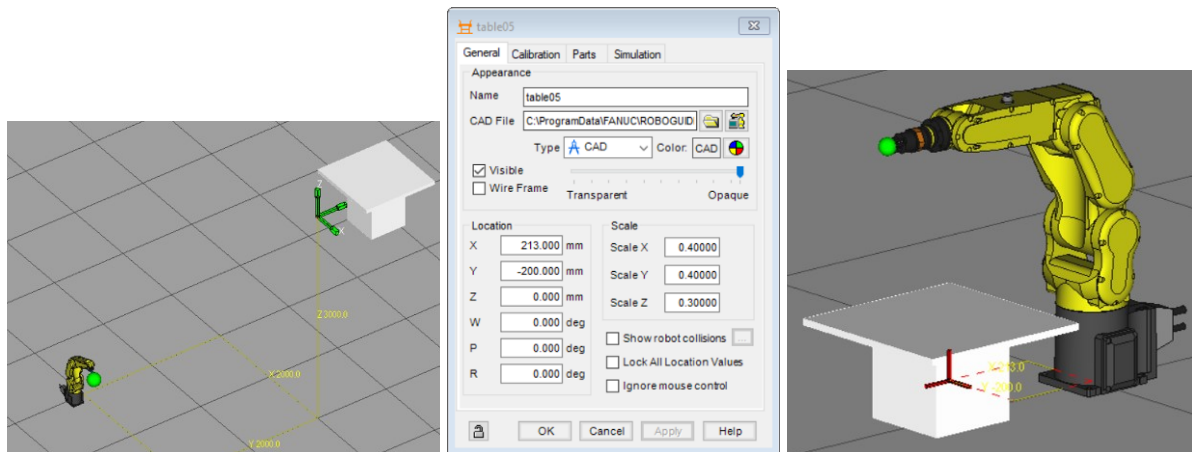


Munkaasztal létrehozása

Jobb kattintás a **Fixtures** (berendezési tárgyak) feliratra, majd **Add Fixture**,
CAD Library, majd a listából pl.: a **table**.



Ez általában túl nagy, és nincs jó helyen, állítsuk be!

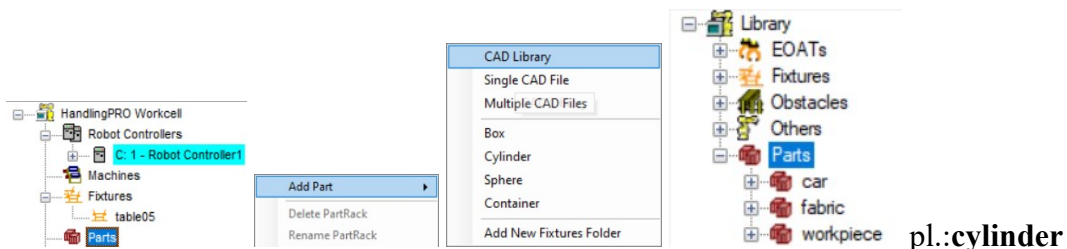


Ha végeztünk, zárjuk le az asztal mozgatásának lehetőségét is (**Lock All Location Values**)!

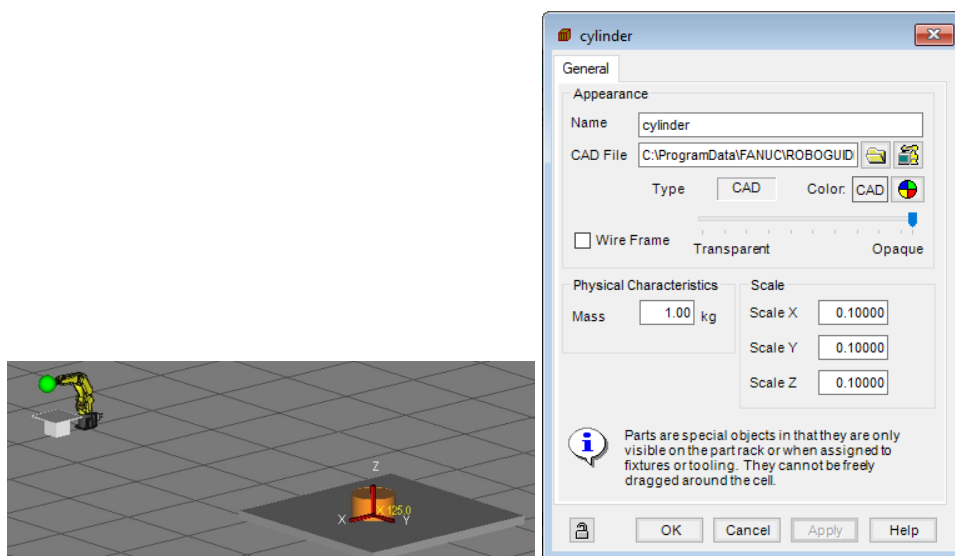
A koordináta-tengelyek pirosra váltanak.

Munkadarab létrehozása

Jobb kattintás a **Parts** (alkatrészek) felíratra, majd **Add Part, CAD Library**, majd a listából a **workpiece**.

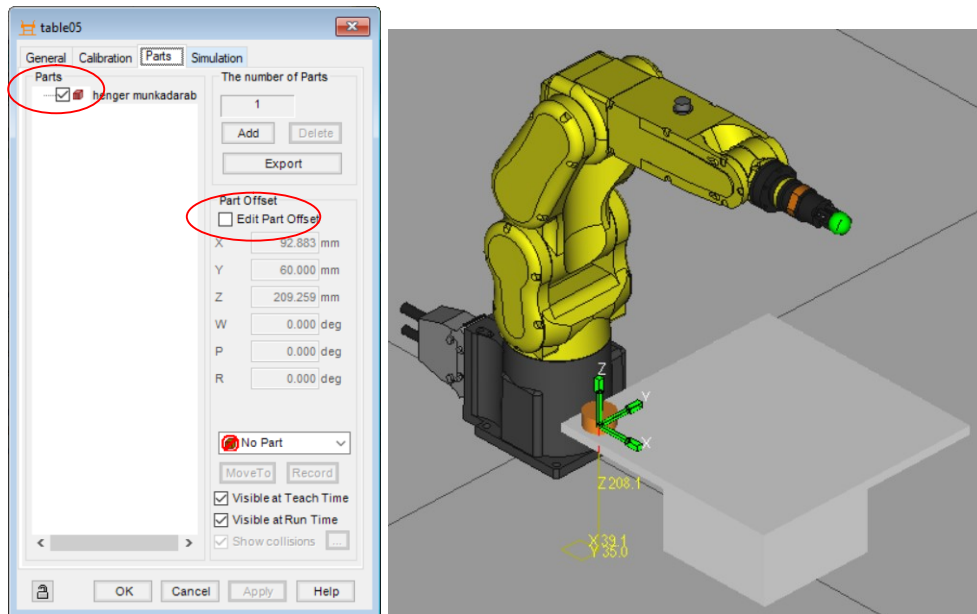


Ez általában túl nagy, és nincs jó helyen, állítsuk be a megfelelő méretet, majd **Apply!**



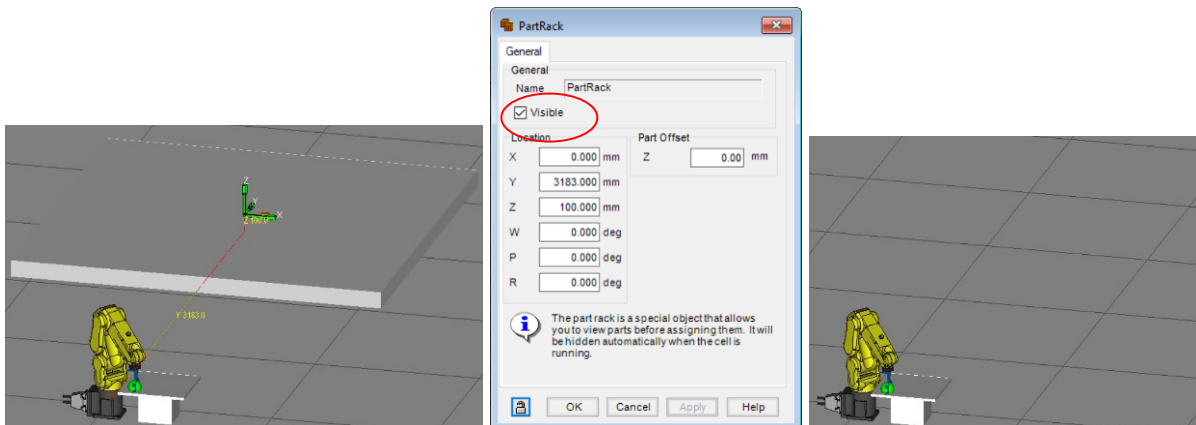
A munkadarab hozzárendelése az asztalhoz

Kettőt kattintunk az asztalon, a megjelenő panel **Parts** fülén bejelöljük a munkadarabunk jelölőnégyzetét, majd **Apply!** Ezután kipipáljuk az **Edit Part Offset** jelölőnégyzetet, majd vonszolással beállítjuk a munkadarab kívánatos helyzetét, majd **Apply**.



Fölösleges tároló lap eltüntetése

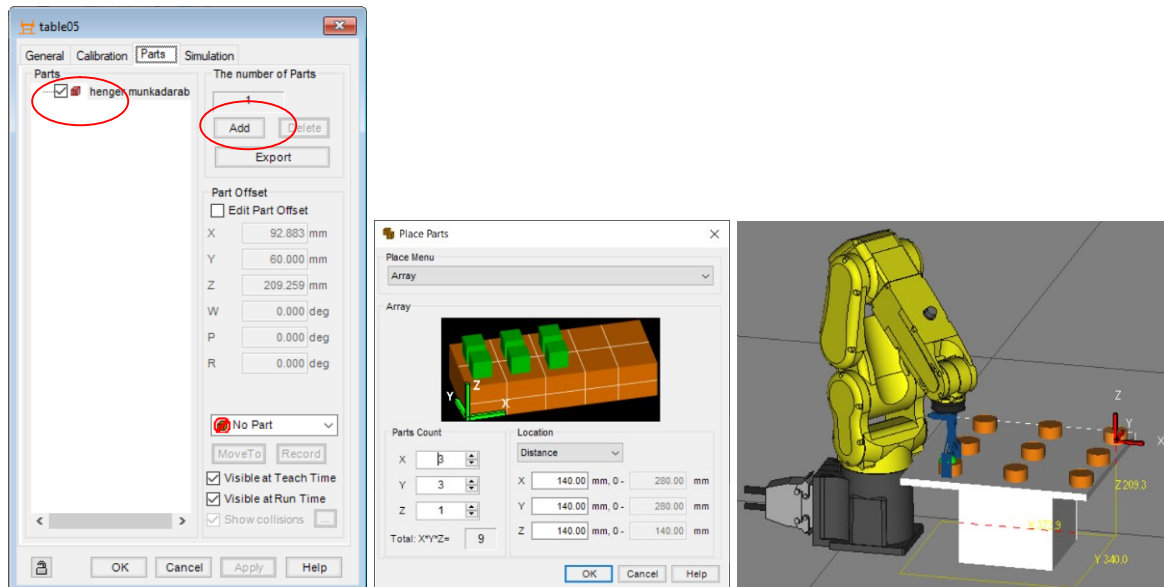
Ha lekicsinyítjük a cellánkat, látszik az eredeti munkadarab tároló lap.



Kattintsunk kettőt a tároló lapon, a megjelenő panelon vegyük ki a pipát a **Visible** elől, majd **Apply!**

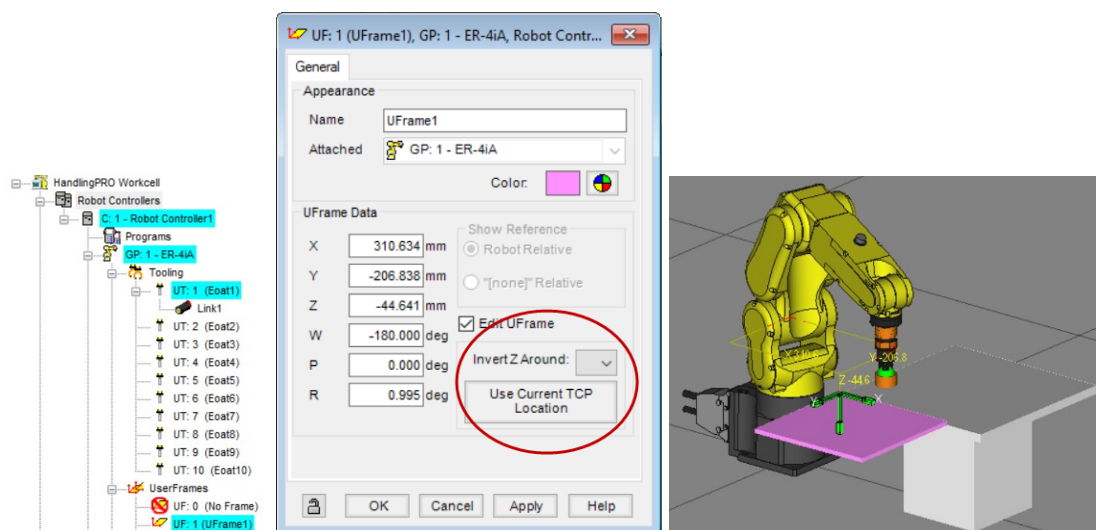
Több azonos munkadarab elhelyezése az asztalon (paletta)

Az asztalra kettőt kattintva a **Parts** fülön nyomjuk le az **Add** ikont! Állítsuk be hány oszlop, hány sor, hány réteg legyen, továbbá állítsuk be az osztásokat!



Uframe definiálása

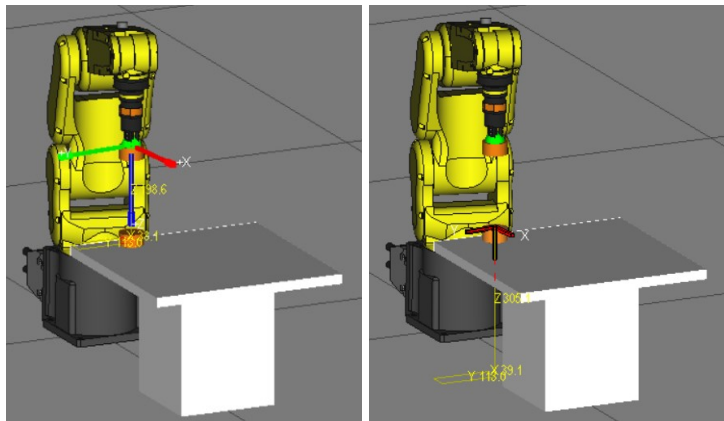
Kattintsunk kettőt az **(UF:1)** elemen (az **UF:0**-ra nem lehet, mert az a **WORLD** koordináta-rendszert jelenti). A megjelenő panelen elnevezhetjük, átszínezhetjük, az **Edit Uframe** bekapcsolásával vonszolással elhelyezhetjük a kívánt helyre és dőlésszöggel, majd **Apply**! A megfelelő irányok miatt használjuk a **USE Current TCP Location** lehetőséget, majd **Apply**!



Move to

Roboguide-ban van arra lehetőség, hogy a robotkar bárhol is áll, ráálljon a tálcán lévő munkadarabra. Ennek feltételei:

A **TCP** tengelyei ugyanúgy álljanak, mint a munkadarab tengelyei. Ha nem így lenne, a munkadarabot addig kell forgatni, míg ez nem teljesül. Pl.: ha a **Z** tengely fordítva áll a munkadarabon, alulról próbálja megfogni, ami nem lehetséges, ezért a **TCP** kis piros gömbbé változik, és elválik a szerszámtól.

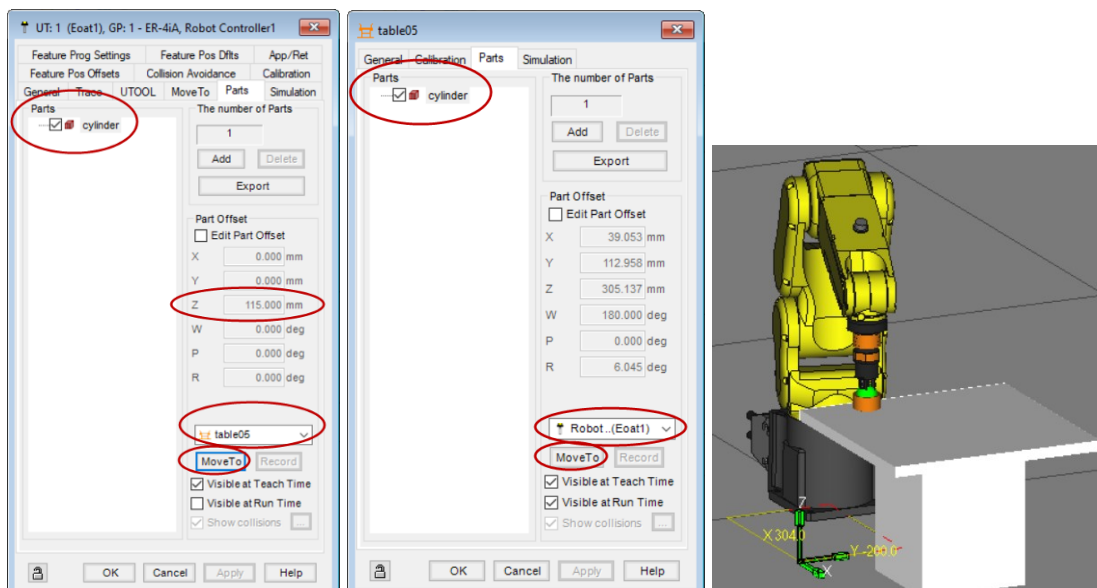



A megfogóhoz hozzá legyen rendelve a munkadarab, vagyis a **Parts** fülön ki van pipálva a munkadarab jelölőnégyzete. Ezen kívül **Z Offszet** ugyanannyi kell, hogy legyen, mint a **TCP**.

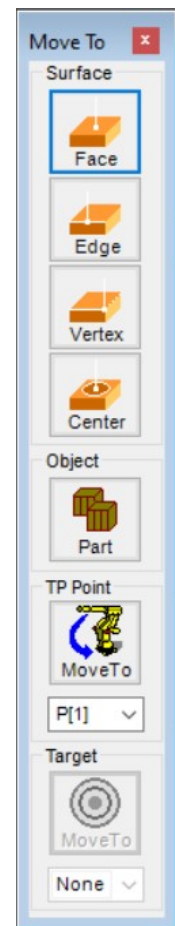
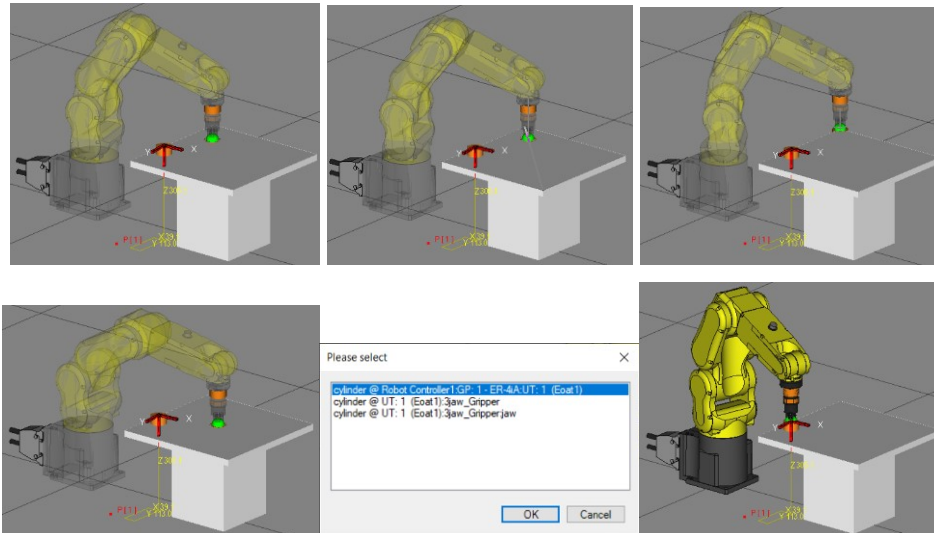
Az asztalhoz is hozzá legyen rendelve a munkadarab, vagyis a **Parts** fülön ki van pipálva a munkadarab jelölőnégyzete.

Egyébként a panelokon látszik a másik hozzárendelés is a **MoveTo** fölött.

Ha minden rendben, nyomjuk meg a **MoveTo** gombot bármelyik panelon, és a robot az asztal megfelelő helyére megy!



A menük között van a  **Move To** ikon, ami előhoz egy eszköztárat. Ha kiválasztjuk a **Face** ikont, majd rákattintunk egy felületre (pl.: asztallap) a robotkar oda mozog. Ugyanígy az **Edge** egy kiválasztott élre vagy átlóra, a **Vertex** egy kiválasztott sarokra, a **Center** a középpontba mozgatja a robotkارت. A **Part** a kiválasztott alkatrészre mozgat, ha több lehetőség van, felajánlja a választást.





Az első négy lehetőség egyébként elérhető billentyűzet-egér kombinációval is (lásd előbb), a **Part** pedig csak a fentebb leírt beállítások után működik.

A Teach Pendant a Roboguide-ban

A virtuális **Teach Pendant** be-, és kikapcsolása a  ikonnal történik.

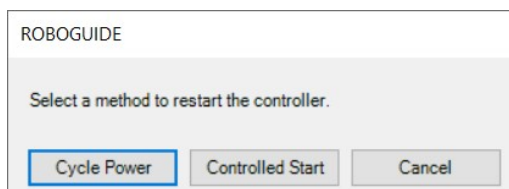
Néhány apró eltéréssel teljesen megegyezik a valós tanító eszközzel, az eltérések:





1. A robotvezérlő nevét mutatja, több vezérlő esetén lenyíló listából választhatunk.
2. A betöltött program nevét mutatja, több program esetén választhatunk.
3.  Be-, és kikapcsolja a **TP** nyomógomb részét.
4.  Be-, és kikapcsolja a számítógépünk bizonyos billentyű-kombinációit az adat-bevitelhez (pl.: **Q** gomb = **Shift-X**). A **TP** képernyőjének sárga kerete az összerendelés kikapcsolt állapotát mutatja.





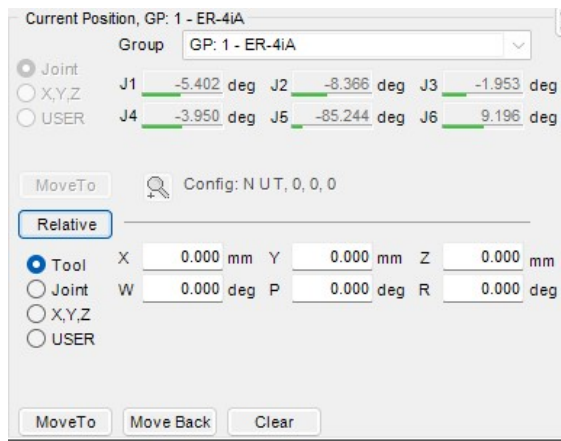
5.  Kérdésre válaszolás után többféle módon újra indítható a vezérlő




6.  Beállítható, hogy a **TP** minden megnyitott ablak felett legyen a **PC** képernyőn, vagy nem.


7. A  ikonnal válthatunk **AUTO** és **T2** üzemmód között, szimulációban nincs **T1** üzemmód.

8.  ikonnal, (ami nem ugyanaz, mint a  nyomógomb) bekapcsolhatunk egy panelt, ahol különféle koordináta-rendszerekben adhatunk meg értékeket, a **MoveTo** ikon lenyomásával a robotkar odaáll. A „**Relative**” gombbal növekményes mozgás-pálya-megadás lehetséges, pl.: egyenesekből és körívekből pálya kirajzolása.



9. A  gombot nem kell folyamatosan nyomni, **kapcsolóként** működik.



10.  gombokat nem kell folyamatosan nyomni, **kapcsolóként** működnék.
11. Mivel a **SHIFT** kapcsoló, gyakran elfeledkezhetünk, hogy be van kapcsolva, a **TP** több menüpontja nem működik ilyenkor, hiába kattintunk rá! Ilyenkor kapcsoljuk ki a **SHIFT**-et!




Fájlkezelés

Fájlok listája


A meglévő programok listája a  gomb lenyomásával nyílik meg.

No.	Program name	Size
1	-BCKEDT-	0/ 132]
2	DROP	6/ 348]
3	GETDATA MR	2/ 286]
4	PAKOLGAT	27/ 1160]
5	PICK	6/ 348]
6	REQMENU MR	2/ 262]
7	SENDDATA MR	2/ 288]
8	SENDEVNT MR	2/ 224]
9	SENDSYSV MR	2/ 274]

Fájl megnyitása

Fel-le kurzor gombokkal   tudjuk kiválasztani a megnyitandó fájlt, majd az  gombot kell lenyomnunk.

Képernyő lapozás több lehetőség esetén

 ikon segítségével lapozhatunk képernyők között, (még egyszer lenyomva visszalép).

Szűrés fájl típusokra (TYPE)

 gomb segítségével tudunk szűrni.

TYPE	1
1	Recent
2	All
3	Collections
4	TP Programs
5	Macro
6	Cond

A „Recent” a legfrissebbeket mutatja, az „All” az összeset, a „TP Programs” az általunk Teach Pendant segítségével írt, a „Macro” a makró programokat szűri, mutatja.

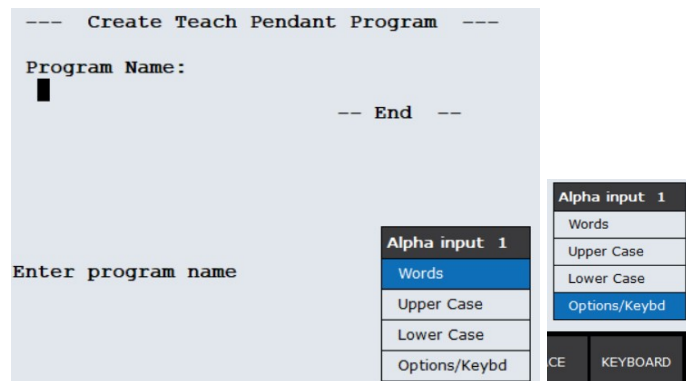
Új program létrehozása (CREATE)

Lenyomva a Create gombot, megkérdezi, mi legyen az új program neve. Válasszuk az Options/Keyboard lehetőséget, majd alatta a KEYBOARD lehetőséget!

A programnévnek **1-36** karakterből kell állnia. Minden programhoz egyedi nevet kell rendelni.

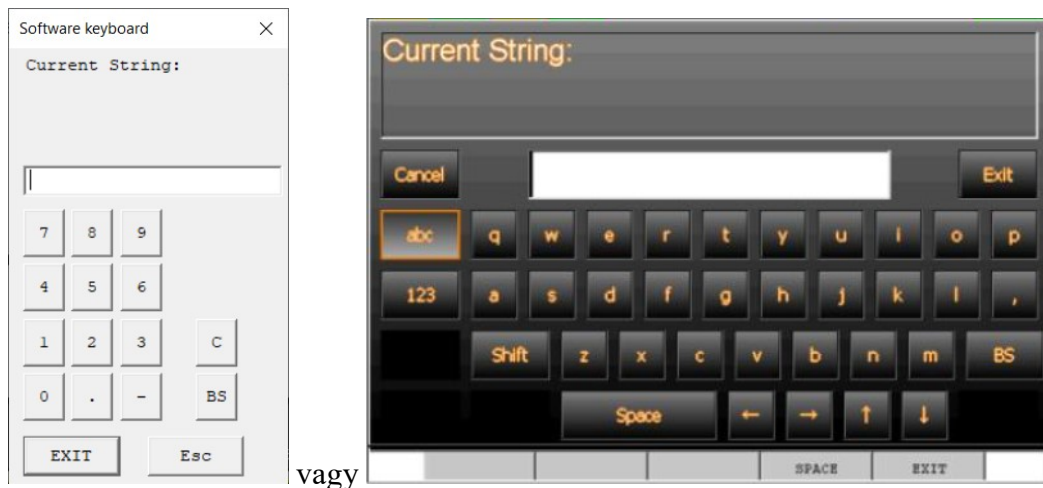
Ha **36** karakter nem jeleníthető meg, akkor a programnév utolsó karaktere „>” - például az „ABCDEF>” - ebben a helyzetben, a teljes programnév a prompt sorban jelenik meg.

A karakterek alfabetikus karakterek, számok **0-tól 9-ig**. **A program neve nem kezdődhet számmal!** Szimbólum csak aláhúzás **_** lehet! A at jel **@** és a csillag ***** nem használható!



Megjelenik egy érintőképernyős billentyűzet, ahol bevihetjük a nevet. **Magyar ékezetes karaktereket ne használjunk!**

EXIT az elfogadás, **Esc** az elvetés, **C** a teljes eddig bevitt karakterek törlése, **BS** az utolsó karakter törlése.



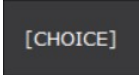
Ha elfogadjuk és **ENTER** gombot ütünk, vagy az **EDIT** ikonra koppintunk, az új program készítéséhez jutunk.

Ha **DETAIL** ikont választjuk, egy beállítási képernyőhöz jutunk.


```

Program name:
1 PROBA
2 Sub type:      [None      ]
3 Comment:      [          ]
4 Group mask:   [1,*,*,*,*,*,* ]
5 Write protect: [OFF      ]
6 Ignore pause: [OFF      ]
7 Stack size:   [    500  ]
8 Collection:   [          ]

```

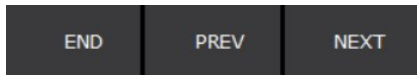
Itt a „**Sub type: (None)**” az egyszerű TP programot jelenti, de a  ikont lenyomva, egyéb programtípust is választhatunk.

- 1 None
- 2 Collection
- 3 Macro
- 4 Cond

A „**Comment**” mezőbe magyarázatot írhatunk. A magyarázat **1-16** karakterből állhat, a karakterekre ugyanazok vonatkoznak, mint a névnél.

A „**Group mask**” TP program esetén maradjon a fenti ábra szerinti!

Ha a „**Write protect**” **ON**, a fájlunk írásvédett lesz, de ekkor nem tudjuk majd szerkeszteni, ezért ezt itt nem célszerű átállítani! (A többit lásd később...)




Az **END** ikonnal befejezhetjük a paraméterezést, **PREV**, **NEXT** ikonokkal lépegethetünk előre, hátra, ha nem férünk ki egy képernyőre.

Program törlése (DELETE)



Fel-le kurzor gombokkal tudjuk kiválasztani a törlendő fájlt. A program megkérdezi,

hogy törölje-e **Delete OK ?**. Választhatunk: . **FIGYELEM:** törléssel a programunk végleg elvész!

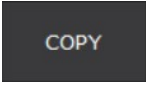
Jellemzők megjelenítése (ATTR)

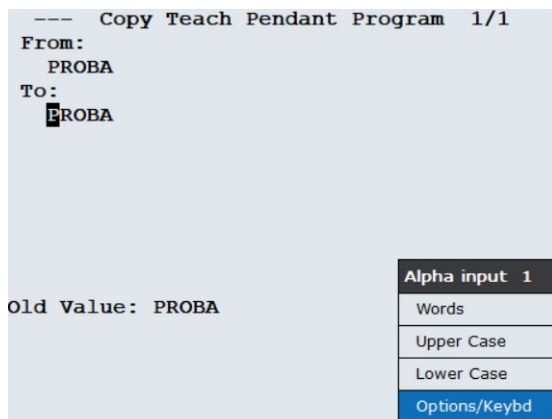
Kiválaszthatjuk, hogy a fájl neve mellett még mi jelenjen meg a listán: magyarázat, védelem, utolsó módosítás, méret, másolási forrás, vagy csak a név.

- | ATTR | 1 |
|------|---------------|
| 1 | Comment |
| 2 | Protection |
| 3 | Last Modified |
| 4 | Size |
| 5 | Copy Source |
| 6 | Name Only |

Másolás (COPY)

Ez tényleg egy másolatot készít más néven (nem vágólapra másol)! Ha egy programot tovább akarunk fejleszteni, de a régi változatra is szükségünk van, ezt használjuk! A parancs

kiadása előtt rá kell állni a kérdéses fájl nevére, majd lenyomni a  ikont!

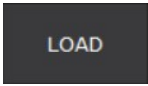


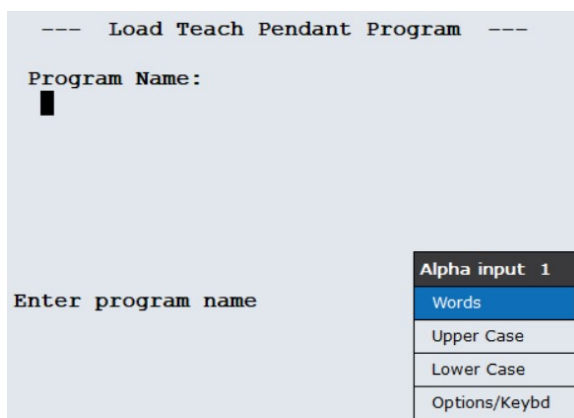
Mutatja a régi nevet, és kéri az új nevet. Ugyanúgy virtuális billentyűzettel vihetjük be.

A fájl részletes paramétereinek beállítása (DETAIL)


Ugyanoda jutunk, és ugyanúgy járhatunk el, mint amit az új fájl létrehozásánál láttunk, csak egy meglévő fájlra alkalmazzuk. Itt már van értelme az írásvédelem beállításának.

Betöltés (LOAD)

 ikon lenyomására behívhatunk egy programot úgy is, hogy begépeljük a nevét.



Mentés máshova, más néven (SAVE AS)

 ikon lenyomására nem csak a fájl nevét, hanem a mentés helyét is beállíthatjuk.

```

From Path: 1/3
MD:\
From Filename:
PROBA.TP

To Device: [UD1:]
To Directory:
\
To Filename:
PROBA.TP

```

A példában **MD:** a robot központi tárhelye, **PROBA.TP** a fájl neve, **UD1:** a vezérlő oldalán lévő **USB** meghajtó, a fájlnev marad **PROBA.TP**

Az **UD1:** az alapértelmezett, de előtte tegyünk megfelelően formázott **pendrive**-t a nyílásba, különben hibaüzenetet kapunk!

Ha a cél van kijelölve, a **[CHOICE]** ikonnal választhatunk más célt egy felugró listáról:

- | |
|--------------------|
| 1 FROM Disk (FR:) |
| 2 Backup (FRA:) |
| 3 RAM Disk (RD:) |
| 4 Mem Card (MC:) |
| 5 Mem Device (MD:) |
| 6 USB Disk (UD1:) |
| 7 USB on TP (UT1:) |

Ha az új fájlnev van kijelölve, a **CHANGE** ikonnal átnevezhetjük a már ismertetett módon.

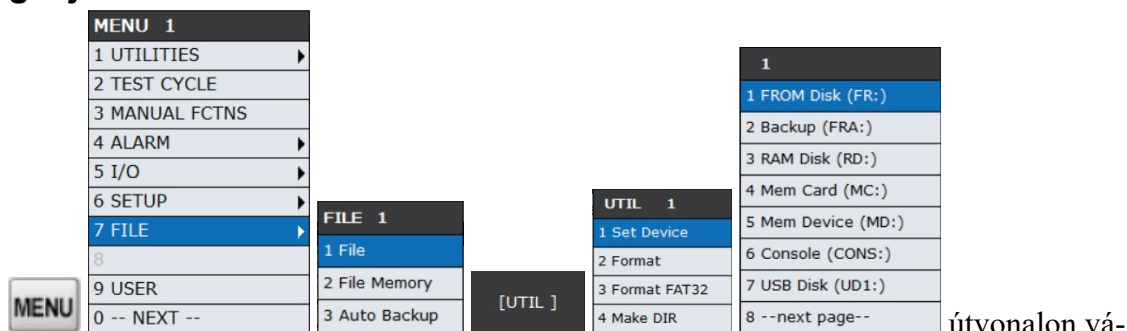
A **DO_SAVE** ikonnal tudjuk végrehajtani a mentést.

Nyomtatás (PRINT)

A **PRINT** ikonnal tudjuk a programot **.LS** kiterjesztéssel elmenteni, ami egy jegyzet-tömbi társítással később olvasható. A nyomtatás előtt még van lehetőségünk átnevezni a fájlt.

A robot a UD1-be helyezett pendrive gyökér könyvtárába menti az .LS fájlt.

Meghajtó váltása



lasszuk vagy a robot **MD:** meghajtóját, vagy a vezérlőn lévő **UD1:** USB csatlakozót!

Miután váltott a program, felszólítást kapunk, hogy a [DIR] ikon megérintésével generáljunk tartalomjegyzéket. Sajnos e nélkül nem látjuk a meghajtó tartalmát!

Press DIR to generate directory

Könyvtárműveletek

Legyünk abban a meghajtóban, ahová a könyvtárat létre akarjuk hozni! pl.: **UD1:**

```
UD1:\*.*                               1/37
 1 PAKOLGAT                             LS      2675
 2 PAKOLGAT                             TP       436
 3 TPSCRN                                LS      1394
 4 VALAMI                                 <DIR>
```

A [DIR] tartalomjegyzék ikonnal válasszuk a *.* lehetőséget, mert így minden fájl, és minden könyvtár látszik!

```
Directory Subset 1/5
 1 *.*
 2 *.KL
 3 *.CF
 4 *.TX
 5 *.LS
 6 *.DT
 7 *.PC
 8 --next page--
```

A könyvtárat a <DIR> jelzi. Belemenni a könyvtárba a kurzormozgatók segítségével, kijelöléssel, majd **ENTER**-el lehet.

```
4 VALAMI                                <DIR>
```

Azt, hogy a könyvtárban vagyunk, a felső sorban az útvonal jelzi: **UD1:\VALAMI*.***

```
UD1:\VALAMI\*.*                          1/33
 1 .. (Up one level)                       <DIR>
```

Egy szinttel visszalépni az (UP one level) <DIR> sorra navigálással, majd az **ENTER**-el lehet.

Ne tévesszen meg minket, hogy **32** sorban felsorolja a kezelőprogram a különféle fájlformátumokat segítségképpen.

Könyvtárat törölni a sor kijelölésével **5 VALAMI2 <DIR>**, majd a

DELETE

ikkonnal lehet.

A program megkérdezi, hogy biztosan törölni szeretnénk-e, válaszolhatunk rá **YES** vagy **NO**.

Delete UD1:\VALAMI2? YES NO

Könyvtárat másolni nem lehet, a **COPY** lenyomásakor erre kapunk is figyelmeztetést.

Directory is not processed

A **[VIEW]** ikonnal háromféle képernyőnézet közül választhatunk: normál, csak a név, széles.

VIEW	1
1	Normal
2	Name Only
3	Wide

A valódi roboton lévő .TP program biztonsági mentése USB adathordozóra

Helyezzük az **USB** adathordozót a vezérlőn lévő **UD1:** nyílásba!

SELECT, majd **>** gomb lenyomása után   gombokkal válasszuk ki a menteni kívánt programot, pl.: **7 PAKOLGAT** !

SAVE AS

lenyomása után láthatjuk, hová szeretne menteni a program.

```
From Path: 1/3
MD:\
From Filename:
PAKOLGAT.TP

To Device: [UD1:]
To Directory:
\
To Filename:
PAKOLGAT.TP
```

Ha kell, a **[CHOICE]** lenyomásával válthatunk meghajtót.



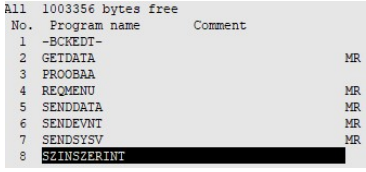
1
1 FROM Disk (FR:)
2 Backup (FRA:)
3 RAM Disk (RD:)
4 Mem Card (MC:)
5 Mem Device (MD:)
6 USB Disk (UD1:)
7 USB on TP (UT1:)
8

Végül nyomjuk meg a **DO_SAVE** ikont! Ha már létezik a fájl a meghajtón, megkérdezi, hogy fölülrja-e. **Overwrite?** . Eldönthetjük, hogy mit akarunk **YES** **NO** .

Valódi roboton megírt *.TP program áttöltése Roboguid-ba

1. A .TP program mentése a robotról a Pendrive-re

- Helyezzünk egy megfelelően formázott (FAT32) pendrive-t a robot UD1:\ nyílásába!

- **SELECT**, majd   gombokkal válasszuk ki a fájlt:  !
- **SAVE AS** ikonnal lépünk be a mentés másként almenübe!
- Ha kell, a **[CHOICE]** ikonnal válasszunk másik meghajtót, de alapértelmezetten az

```
From Path:
MD:\
From Filename:
SZINSZERINT.TP

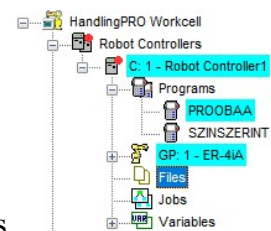
To Device:
To Directory:
\
To Filename:
SZINSZERINT.TP
```

UD1: van felajánlva (esetleg át is nevezhetjük a fájlt)!

- **DO_SAVE** ikonnal mentjük el a .TP fájlt a pendrive-re!





2. A .TP program betöltése pendrive-ról a Roboguid-ba

- Helyezzük a .TP programot tartalmazó pendrive-t a számítógép valamelyik USB nyílásába!



- A bal oldali struktúrában keressük meg a **Files** ikont, és válasszuk az **Add** lehetőséget!
- Tallózzuk ki a .TP fájlt a pendrive-n, majd nyomjuk meg a **Megnyitás** gombot!

- A megjelenő új fájlneven  SZINSZERINT.TP kattintsunk rá jobb egérgombbal, majd válasszuk a **Load** lehetőséget!

- Most már a virtuális **Teach Pendant**-on a  gombot lenyomva a   gombokkal kitallózható az új .TP fájl,  gombbal betölthető a programozó konzolba!

FIGYELEM! MOST CSAK A PROGAMOT VITTÜK ÁT, A KERETEKET, A PONTOKAT, REGISZTEREKET, POZÍCIÓREGISZTEREKET NEM!

Valódi roboton megírt pozícióregiszterek áttöltése Roboguid-ba


3. A pozícióregiszterek mentése a robotról a Pendrive-re

- Helyezzünk egy megfelelően formázott (FAT32) pendrive-t a robot **UD1:**\ nyílásába!


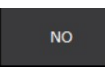
MENU 1	FILE 1
1 UTILITIES	1 File
2 TEST CYCLE	2 File Memory
3 MANUAL FCTNS	3 Auto Backup
4 ALARM	4 TATE
5 I/O	5 2
6 SETUP	6 1
7 FILE	7 1
8	8 WLD
9 USER	9 FG
0 -- NEXT --	

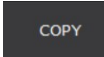
-  útvonalon tallózzuk ki a **POSREG** fájlokat!

111 POSREG	VA	0
112 POSREG	VR	0

- A **.VR** a pozíció regisztereket tartalmazza, míg a **.VA** ezek valamilyen szövegszerkesztőben megjeleníthető formátumát. A **.VA** soron állva a  ikont lenyomva ellenőrizhetjük, hogy jó pozíció regisztereket akarunk-e másolni.

```
[*POSREG*]$POSREG Storage: SHADOW Acce
[1,1] = 'alul'
Group: 1 Config: N U T, 0, 0, 0
X: 393.908 Y: 149.373 Z: 169
W: -82.744 P: 80.222 R: 115
[1,2] = 'fent'
Group: 1 Config: N U T, 0, 0, 0
X: 409.150 Y: 117.696 Z: 373
W: 77.040 P: 84.513 R: -84
[1,3] = '' Uninitialized
[1,4] = '' Uninitialized
Continue displaying?
```

-   A **YES** ikonnal folytathatjuk az ellenőrzést, a **NO** ikonnal visszaléphetünk az előző képernyőre.

- **.VR** soron állva a  ikonnal kezdeményezhetjük a másolást!

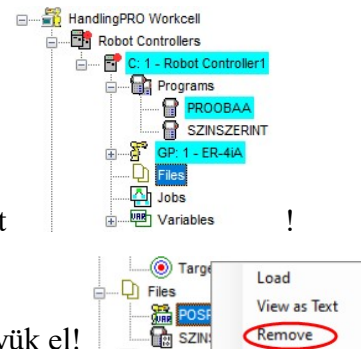
```
From Path: 1/3
MD:\
From Filename:
POSREG.VR

To Device: [UD1: ]
To Directory:
\
To Filename:
POSREG.VR
```

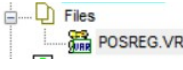
- A [CHOICE] ikonnal állítsuk a mentés helyét **UD1:-re**, majd a DO_COPY ikonnal mentjük is el a fájlt!

4. A POSREG.VR program betöltése pendrive-ről a Roboguid-ba

- Helyezzük a **POSREG.VR** programot tartalmazó pendrive-t a számítógép valamelyik **USB** nyílásába!



- A bal oldali struktúrában keressük meg a **Files** ikont
- Ha már van ott egy régebbi **POSREG.VR** fájl, vegyük el!
- Jobb egérgombbal kattintva a „Files”-en válasszuk az **Add** lehetőséget!
- Tallózzuk ki a **POSREG.VR** fájlt a pendrive-n, majd nyomjuk meg a **Megnyitás** gombot!

- A megjelenő új fájlneven  kattintsunk rá jobb egérgombbal, majd válasszuk a **Load** lehetőséget!
- Ellenőrzésképpen a Roboguide virtuális Teach Pendantján nyomjuk le a **DATA**

TYPE	1
1	Registers
2	Position Reg
3	String Reg
4	KAREL Vars
5	KAREL Posns

gombot, a [TYPE] ikonnal válasszuk a pozícióregisztert, és nézzük meg, hogy a megfelelő értékek jöttek-e át!

```
PR[ 1:alul ]=R
PR[ 2:fent ]=R
```

Felhasználói keretek átvitele a valóságos robot és a Roboguide között

Az előbb leírt módszerrel az alábbi két fájlt kell átmozgatni a valóságos robot és a **Roboguide** között:

FRAMEVAR.VR a felhasználói keretek (egy vagy több) elnevezéseit viszi át

SYSTRAME.SV a felhasználói keretek koordinátáit viszi át. A **sysframe.sv** betöltéskor (**load**) a **Roboguide** és a robot is újraindítást kér.

SYSTRAME.VA a felhasználói keret koordinátáinak olvasható formátuma (nem kell átmásolni, de ebben meg lehet nézni, ellenőrizni lehet).

Mentéskor, ha már létezik a fájl a kiválasztott meghajtón (UD1:), megkérdezi, hogy felülírja-e.

IO beállítások megjegyzések (Comment) átvitele a valóságos robot és a Roboguide között

Az előbb leírt módszerrel az alábbi fájlt kell átmozgatni a valóságos robot és a **Roboguide** között:

DIOCFGSV.IO az IO beállításokat és megjegyzéseket viszi át. Betöltéskor (**load**) a **Roboguide** és a valós robot is újraindítást kér.

Mentéskor, ha már létezik a fájl a kiválasztott meghajtón (UD1:), megkérdezi, hogy felülírja-e.

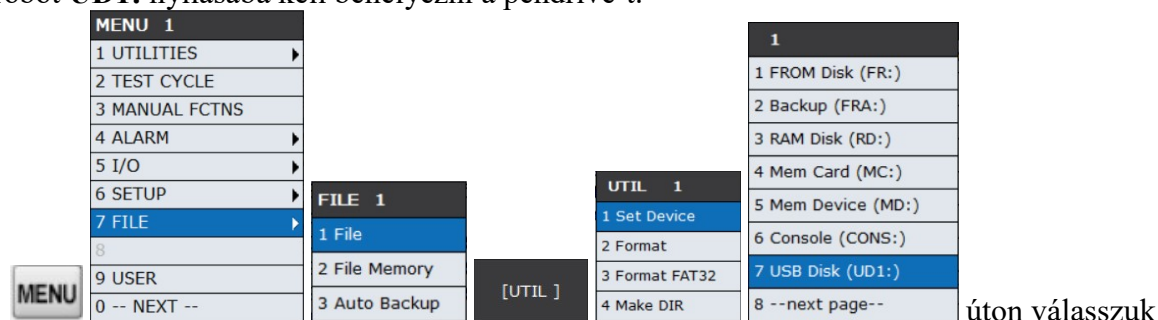
Roboguide-ben írt program betöltése a robotba USB-ről

A Roboguide **mentési helye** általában: Dokumentumok > My Workcells > *a cella neve*,
amin dolgoztam > SavePoints > *a szerkesztés dátuma* > Robot_1

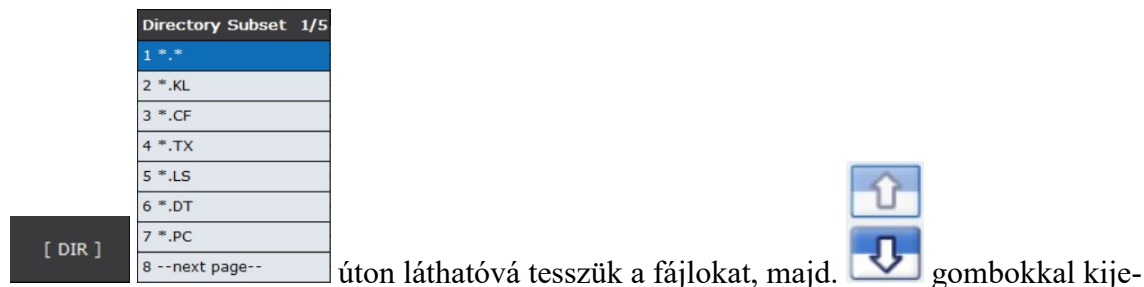
Itt kell kiválasztani a *.tp fájlokat és a pendrive gyökér könyvtárába másolni.

A **Robot_1** mappában is van egy virtuális **UD1** mappa, de ez természetesen nem alkalmas a valódi eszközök közötti fájlok mozgatására, csupán a virtuális **Teach Pendant** a virtuális meghajtó váltásra való!

A robot **UD1**: nyílásába kell behelyezni a pendrive-t.

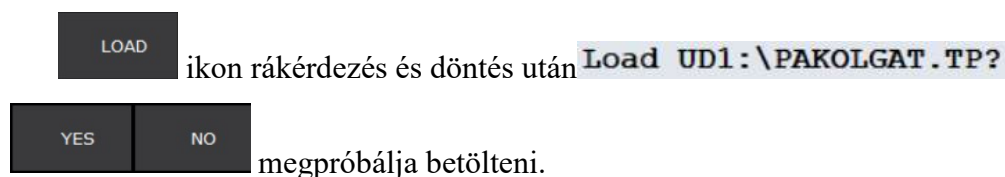


ki a meghajtót, amiről be szeretnénk olvasni!



löljük a betöltendő fájlt.

Pl.: **2 PAKOLGAT TP 436**



Ha már benn van a memóriában, **UD1:\PAKOLGAT.TP already exists**,

választhatunk, hogy felülírja, kihagyja, vagy mégsem akarjuk. **OVERWRITE SKIP CANCEL**

FIGYELEM! MOST CSAK A PROGAMOT VITTÜK ÁT, A KERETEKET, A PONTOKAT, REGISZTEREKET, POZÍCIÓREGISZTEREKET NEM!

Ezzel a módszerrel a megfelelő fájlokat kiválasztva átvihetők keretek, pontok, regiszterek, pozícióregiszterek, IO beállítások is. Keretknél, IO beállításoknál a robot újraindítást kér.

A valódi roboton lévő minden fájl biztonsági mentése USB adathordozóra

Helyezzük az **USB** adathordozót a vezérlőn lévő **UD1**: nyílásba.

The screenshot shows a sequence of menu screens. On the left, a 'MENU' button is shown. The main menu 'MENU 1' lists options 1-9, with '7 FILE' selected. The 'FILE 1' sub-menu shows '1 File', '2 File Memory', and '3 Auto Backup'. A '[UTIL]' button is shown. The 'UTIL 1' sub-menu lists '1 Set Device', '2 Format', '3 Format FAT32', and '4 Make DIR'. On the right, a menu '1' lists storage options: '1 FROM Disk (FR:)', '2 Backup (FRA:)', '3 RAM Disk (RD:)', '4 Mem Card (MC:)', '5 Mem Device (MD:)', '6 Console (CONS:)', '7 USB Disk (UD1:)', and '8 --next page--'. The '7 USB Disk (UD1:)' option is highlighted. To the right of this menu, the text 'úton válasszuk ki az' is visible.

UD1: meghajtót!

The screenshot shows a 'BACKUP 1' menu with options: '1 System files', '2 TP programs', '3 Application', '4 Applic.-TP', '5 Error log', '6 Diagnostic', '7 Vision data', '8 All of above', '9 Maintenance data', and '0 -- NEXT --'. The '8 All of above' option is highlighted. To the left of this menu, a '[BACKUP]' button is shown. Below the button, the text 'ikon lenyomása után válasszuk az „All of above” lehetőséget' is visible.

A külső tároló mentés előtti törlésre kérdez rá a program, az esetleges fájl keveredések elkerülése végett célszerű lehet a háttértárolót törölni.

Delete UD1:\ before backup files?

FIGYELEM! Ha **YES**-t választunk, minden a Pendrive-n lévő adatunk törlődik!

Parancsszerkesztő (Edit Command)

Az **EDCMD** ikon érintésével jutunk az üzemmódba, ekkor ikonja megváltozik, jobb felül egy piros **X** jelenik meg, ezért, ha még egyszer megérintjük, kilépünk az üzemmódból. Alább láthatók a menüpontok.

EDCMD 1	EDCMD 2
1 Insert	1 Icon Editor
2 Delete	2 Color
3 Copy/Cut	3 IO Status
4 Find	4
5 Replace	5
6 Renumber	6
7 Comment	7
8 Undo	8
9 Remark	9
0 -- NEXT --	0 -- NEXT --

Almenüből feljebb lépni a baloldalon lévő **PREV** gombbal lehet.

Üres sor beszúrása (Insert)

A parancs kiadása előtt rá kell állni a sor elejére! A sor **FÖLÉ** szűr be! Ha kiválasztjuk, megkérdezi, hogy hány sort. Billentyűzetről beírhatjuk, majd **ENTER**, de ha csak egy sort akarunk, elég az **ENTER**.

```
How many line to insert ? : █ ENTER
```

Sor törlése (Delete)

A parancs kiadása előtt rá kell állni a sor elejére! Egyszerre **CSAK EGY SORT** törölhetünk! Megkérdezi, hogy töröljük-e. Igenrel válaszolva a sor törlődik.

```
Delete line(s) ? YES NO
```

Másolás / kivágás (Copy / Cut)

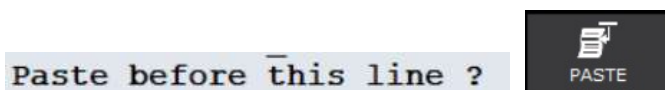
A parancs kiadása előtt rá kell állni a sor elejére! Ha kiválasztjuk a **Copy/Cut** menüt, megjelenik a kiválasztás (**SELECT**) lehetőség. Több egymást követő sort a kurzormozgató nyilakkal tudunk kijelölni.

```
SELECT Move cursor to select range 1: //UTOOL_NUM=1 2: //UFRAME_NUM=0
```

Ezután kell választanunk, hogy másolás (**COPY**) vagy kivágás (**CUT**).



Annak a sornak az elejére kell állni, ami fölé akarunk beilleszteni. Ezután tudjuk kiválasztani a beillesztés (**Paste**) lehetőséget. Erre figyelmeztet is.



Lenyomva a beillesztés gombot, három féle mód, és a „mégsem” közül választhatunk.



4:J P[...] 100% FINE Csak a paramétereket szűrja be, a pozíciót nem.



5:J @P[1] 100% FINE Pozíciót és paramétereket is beszúr.



6:J @P[9] 100% FINE Új pozícióként veszi fel a pozíciót.



Mégsem.

Amikor olyan sort szűrünk be, amiben nincs pozíció, a három féle beillesztés egyenértékű.

Keresés (Find)

Megnyílik egy menü, abból választva egy almenü, végül megkérdezi a sorszámát (index)

Select Find menu 1/2	Select Find item 1/2
1 Registers	1 DO[]
2 I/O	2 DI[]
3 IF/SELECT	3 RO[]
4 WAIT	4 RI[]
5 JMP/LBL	5 GO[]
6 Miscellaneous	6 GI[]
7 CALL	7 SO[]
8 --next page--	8 --next page--

Enter index value

Beírva, majd **ENTER**-t ütve megkeresi a keresett értéket. **22: RO[1]=(!RO[1])**

A **NEXT** gombbal a következő találatra ugrik, az **EXIT** befejezi a keresést.



Csere (Replace)

Először azt kell megmondanunk, hogy mit cseréljen. Megnyílik egy menü, abból választva egy almenü, végül megkérdezi a sorszámát (index), majd **ENTER**.

Select Replac menu 1/1	Select Replac item 1/2
1 Registers	1 DO[]
2 Motion modify	2 DI[]
3 I/O	3 RO[]
4 JMP/LBL	4 RI[]
5 CALL	5 GO[]
6 TIME BEFORE/AFTE	6 GI[]
7	7 SO[]
8	8 --next page--

Most meg kell mondanunk, hogy mivé cserélje. Megnyílik egy menü, abból választva egy almenü, végül megkérdezi a sorszámát (index), majd **ENTER**.

Select Replac menu 1/1	Select Replac item 1/2
1 Registers	1 DO[]
2 Motion modify	2 DI[]
3 I/O	3 RO[]
4 JMP/LBL	4 RI[]
5 CALL	5 GO[]
6 TIME BEFORE/AFTE	6 GI[]
7	7 SO[]
8	8 --next page--

Ezután választanunk kell, hogy minden találatot cserélünk (**ALL**), vagy csak azt, ahol a kurzor áll (**YES**). A **NEXT** gombbal léphetünk a következő csere helyére. **EXIT** kiléptet az üzemmódból.



Újra sorszámozás (Renumber)

Ha esetleg szükség lenne rá, újrásorszámozza a programot. Megkérdezi, hogy újra sorszámozza-e, **YES** gombra meg is teszi.



Magyarázat (Comment)

Minden változóhoz rendelhető egy **Comment**, ami ezzel a paranccsal megjeleníthető, vagy még egyszer kiadva eltüntethető a programban.

Comment nélkül: `4: RO[1]=OFF` Commenttel: `4: RO[1:megfogo]=OFF`

Mégsem (Undo)

Az utolsó egy darab művelet visszavonható. Ki is írja, mi volt az. **Undo? (Paste)** Választhatunk, igen, vagy nem.



Még egyszer kiadva a parancsot, a visszavonást vonja vissza, vagyis **Redo**. Redo?

Megjegyzés (Remark)

A // jel utáni sor nem hajtódik végre.

A parancs kiadása előtt rá kell állni a sor elejére!

```
1: //UTOOL_NUM=1
2: //UFRAME_NUM=0
```

A parancs kiadása után felszólít a sorok kijelölésére.

Select lines to remark or unremark

Kurzormozgató nyilakkal tudunk több sort kijelölni.

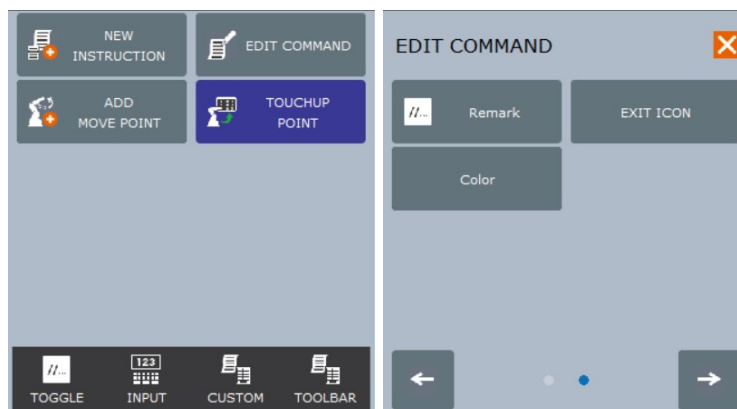
```
1: //UTOOL_NUM=1
2: //UFRAME_NUM=0
```

REMARK megjegyzéssé alakít, **UNREMARK** újra végrehajtandó sorrá tesz.



Ikon alapú szerkesztő mód (Icon Editor)

Egy másik kinézetű programszerkesztőt hoz elő, amiből az **EDIT COMMAND** parancs második képernyőjén lévő **EXIT ICON** parancssal léphetünk ki.



Szín megjelenítése (Color)

Az I/O célértékek háttérét színezi. **ON** zöld, **OFF** piros. Még egyszer kiadva a parancs elveszi a háttérszíneket.

```
3: RO[1]=OFF
```

```
3: RO[1]=OFF
```

IO állapot (IO Status)

A programsorban megjeleníti, hogy az aktuális IO milyen állapotban van éppen. Még egyszer kiadva a parancs elveszi az állapot kijelzését.

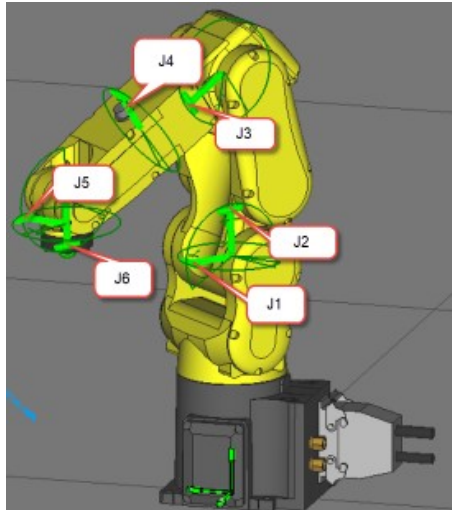
```
3: RO[1:OFF]=OFF
```

Koordinátarendszerek

Tanítás közben a kiválasztott koordinátarendszer határozza meg, hogyan fog mozogni a robotkar. A koordinátarendszerek a következők:

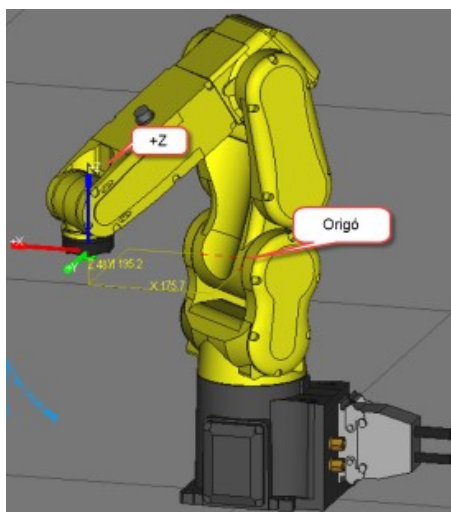
JOINT

Külön-külön mozgatja a robot egyes tengelyeit (**J1...J6**).



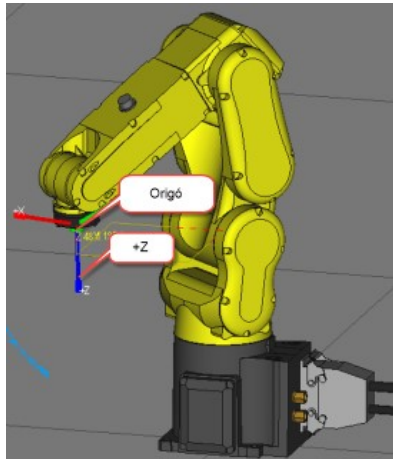
WORLD

Derékszögű rendszer, mozgatja a robotot **x**, **y** vagy **z** irányba, és elforgatja **x(w)**, **y(p)** és **z(r)** tengelyek körül. Az **origó** a robot **J2** tengelyének a középpontja, **+Z** mindig felfelé mutat.

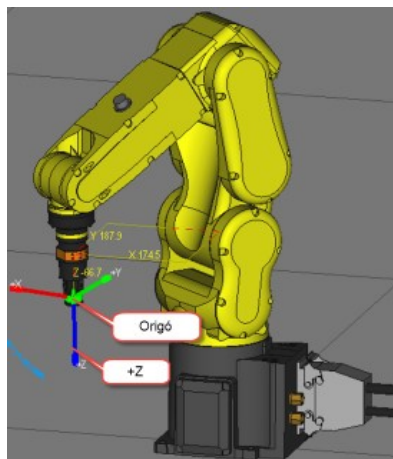


TOOL (szerszám koordináta rendszer)

Derékszögű rendszer, mozgatja a robotot x , y vagy z irányba, és elforgatja $x(w)$, $y(p)$ és $z(r)$ tengelyek körül. **+Z fordul a robotkarral**, az ábrán ezért lefelé mutat. **Az origó a robotkar pereme**, amire a szerszámokat fel kell szerelni.

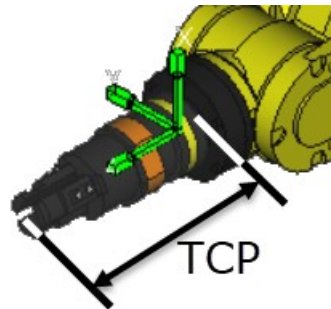


Az alapértelmezés nem megfelelő a későbbiekben, mert a peremre még szerszámot fogunk szerelni, ezért ezt az értéket el kell majd tolni a szerszám hosszával (TCP)!



TCP eltolás, UTOOL

Tool Center Point, vagyis a szerszám középpontja. Ez bizonyos távolságban van a peremtől, és ahhoz, hogy helyesen mozogjon a robotkar, ezt be kell állítani. A Schunk EGP 40-NNB megfogónál ez **Z= 115**.



Beállítása:

A **MENU** gombot lenyomva válasszuk a **SETUP>Frames** lehetőséget!



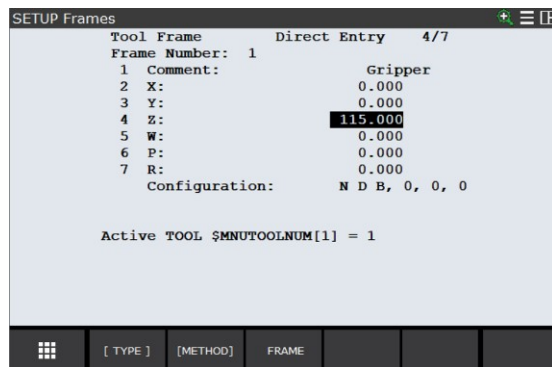
A **|OTHER|** gombot lenyomva van lehetőség a **Tool Frame** kiválasztására. 10 db beállítást van lehetőség definiálni. A szükségtelenek kijelölés után törölhetők a **CLEAR** gombbal.

Tool Frame	X	Y	Z	Direct Entry	Comment	1/10
1	0.0	0.0	115.0	[Gripper]
2	0.0	0.0	0.0	[Eoat2]
3	0.0	0.0	0.0	[Eoat3]
4	0.0	0.0	0.0	[Eoat4]
5	0.0	0.0	0.0	[Eoat5]
6	0.0	0.0	0.0	[Eoat6]
7	0.0	0.0	0.0	[Eoat7]
8	0.0	0.0	0.0	[Eoat8]
9	0.0	0.0	0.0	[Eoat9]
10	0.0	0.0	0.0	[Eoat10]

OTHER 1

- 1 Tool Frame
- 2 Jog Frame
- 3 User Frame
- 4 Cell Frame
- 5 Cell Floor

A **DETAIL** választva a kijelölt sor szerkeszthető. **Schunk EGP 40-NNB** megfogónál **Z** oszlopába írjunk be **115**-öt, majd üssünk **ENTER**-t!, (Esetleg írjunk magyarázatot pl.: **Gripper**).



Ezt a beállítást a mozgatáshoz, tanításhoz külön aktiválni kell!

A használni kívánt **Tool Frame** kiválasztásához nyomjuk meg a **SETIND** billentyűt, írjuk be a kívánt felhasználói keret számát, és nyomjuk meg az **ENTER** billentyűt. Ez beállítja az aktív szerszámkeretet a kívánt keretszámra.

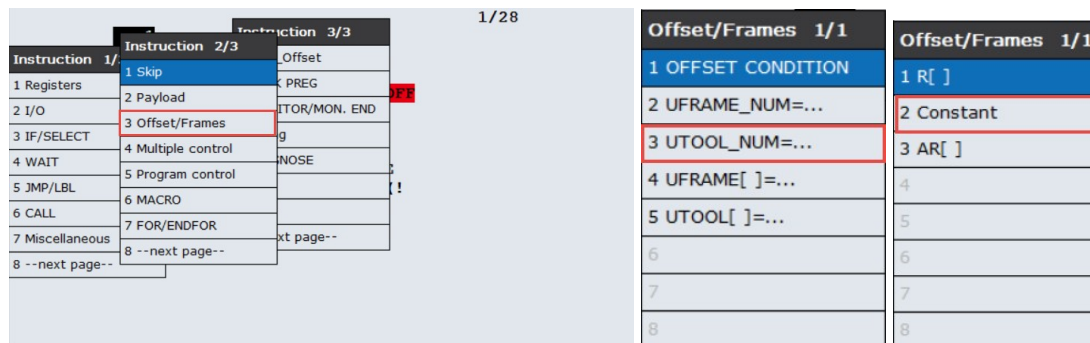
Ha nem cserélünk szerszámot a roboton, ezt csak egyszer kell a felszerszámozáskor beállítanunk!

Nyomjuk meg **CLRIND** gombot, ha törölni szeretnénk a jelenleg használatra kiválasztott felhasználói keret számát. Ekkor nem lesz aktív keretünk, újra fel kell venni valamelyiket aktívnek.

Programsor Utool váltásra

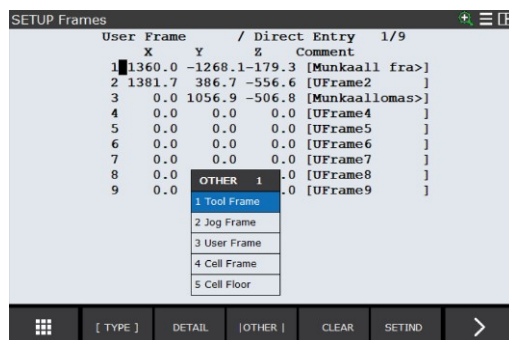


Programunk egy üres során állva válasszuk a **[INST]** lehetőséget, majd a második képernyőn az **Offset Frames** lehetőséget, ezután a **UTOOL_NUM=**, majd a **Constant** lehetőséget! Végül írjuk be a billentyűzetről a kívánt keret számát és **ENTER**!



Eredmény: **1: UTOOL NUM=1**

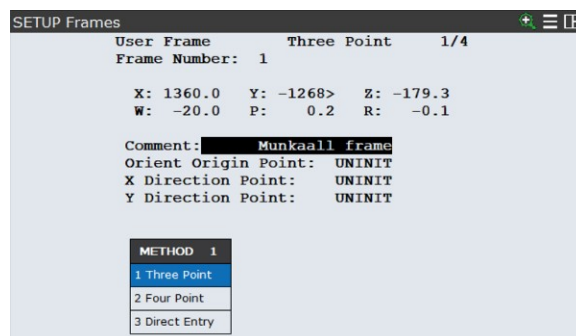
Ha a robotunk a programfutás során több szerszámot használ, ezeket a váltásokat a programunk megfelelő soraiban kell elhelyezni.



9 db beállítást van lehetőség definiálni. A sürgősszerűen kijelölés után törölhető a **CLEAR** gombbal (**YES** megerősítést igényel).

Ha új keretet akarunk létrehozni, álljunk rá egy üres sorra!

A **[METHOD]** gombbal ki kell választanunk, milyen módszert akarunk használni (három pont felvételével, négy pont felvételével, vagy közvetlen koordináták bevitele). **Válasszuk a három pont lehetőséget!**

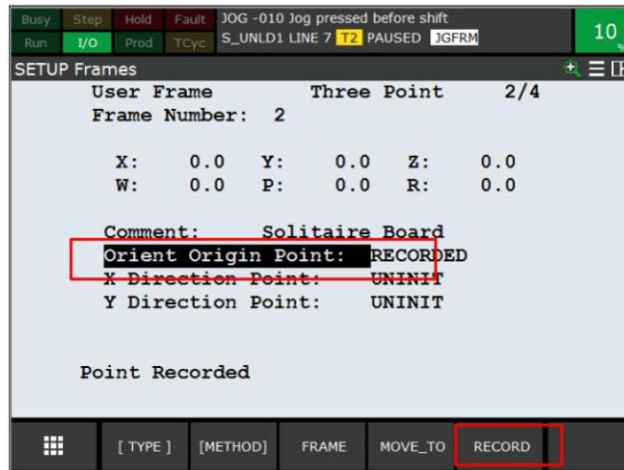


A **DETAIL** gombbal láthatjuk a részleteket, tudunk belépni a szerkesztőbe.

- A kurzorral válasszuk ki az **Orient Origin Point** sort!
- Mozgassuk a robotot az új origóba, majd nyomjunk **SHIFT+ RECORD**-ot!
- Válasszuk ki az **X Direction Point** sort!
- Mozgassuk a robotot abba az irányba, amit **+X** -nek szeretnénk, majd nyomjunk **SHIFT+ RECORD**-ot! (A könnyebb kezelhetőség érdekében **+X** irányba minél inkább hasonlítson a **WORLD +X** irányhoz!)
- Válasszuk ki az **Y Direction Point** sort!
- Mozgassuk a robotot **az előzőre merőlegesen** abba az irányba, amit **+Y** -nek szeretnénk, majd nyomjunk **SHIFT+RECORD**-ot! (Szintén **WORLD +Y** közeli!)

MOVE_TO Nyomjuk meg ezt a gombot a rögzített pozícióra lépéshez. Vigyük a kurzort a kívánt pontra, és nyomjuk meg a **MOVE_TO** gombot a **SHIFT** billentyűvel együtt. A robot rááll az előzőleg rögzített pontra.

FRAME Ha lenyomjuk megkérdezi, milyen sorszámú keretet akarjuk látni. (Használatához tudni kell fejből a keretek sorszámát.)




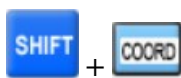
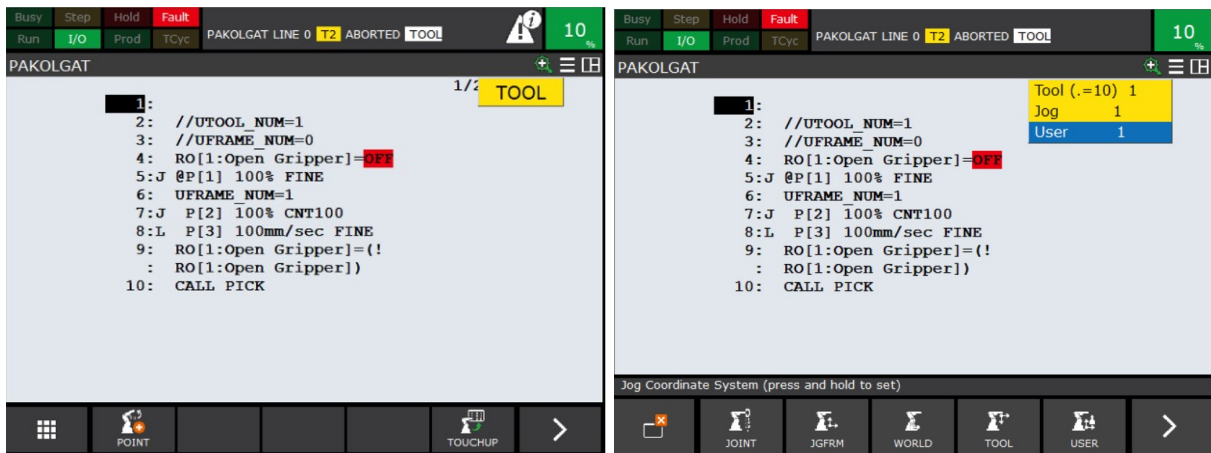
Ezt a beállítást a mozgatáshoz, tanításhoz külön aktiválni kell!

A használni kívánt felhasználói keret kiválasztásához nyomjuk meg a **SETIND** billentyűt, írjuk be a kívánt felhasználói keret számát, és nyomjuk meg az **ENTER** billentyűt. Ez beállítja az aktív felhasználói keretet a kívánt keretszámra.

Nyomjuk meg **CLRIND** gombot, ha törölni szeretnénk a jelenleg használatra kiválasztott felhasználói keret számát. Ekkor nem lesz aktív keretünk, újra fel kell venni valamelyiket aktívnak.

Egyszerű koordináta-rendszer váltás a kezelőn a tanításhoz



A koordináta-rendszer megváltoztatásához nyomjuk meg egymás után többször a  gombot a konzolon. A választott koordináta-rendszer a képernyő jobb felső sarkában jelenik meg.




esetén, választéklista jelenik meg, amik között kurzormozgatókkal tudunk lépkedni. Itt, ha kell, **átírható** az **UFRAME** sorszám.

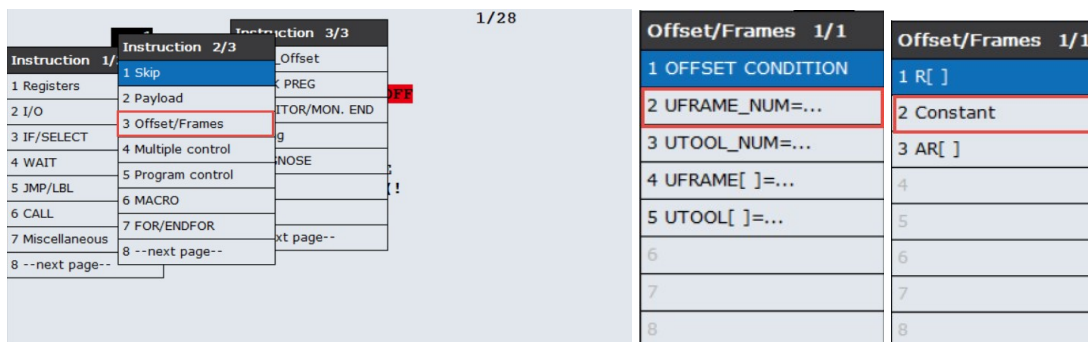
A **(.=10)** azt jelenti, hogy 10 lehetőségünk van.

Ettől még nem vált át a robot az új koordináta-rendszerre futtatáskor. Vagy a fentebb említett SETIND módszert kell alkalmaznunk, vagy írunk kell erre egy programsort, és azt le is kell futtatnunk (a `6: UFRAME_NUM=1` sor elejére állunk, majd nyom-

juk le a DEADMAN +  +  gombokat!) !

Programsor Uframe váltásra

Programunk egy üres során állva válasszuk a  lehetőséget, majd a második képernyőn az **Offset Frames** lehetőséget, ezután a `UFRAME_NUM=`, majd a **Constant** lehetőséget! Végül írjuk be a billentyűzetről a kívánt keret számát és **ENTER**!





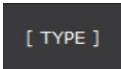
Eredmény: `6: UFRAME_NUM=1`

Ha a robotunk a programfutas során több munkasikot használ, ezeket a váltásokat a programunk megfelelő soraiban kell elhelyezni.

Robot I/O

Robot kimenetek: **RO[1]... RO[8]** a robot karján fõnt lévõ kimenetek

- **RO[1]... RO[4]** pneumatika szelepek.
- Ha nincsenek komplementerbe állítva, 4 db egyszeres mûködésû munkahenger köthetõ rájuk, egyenként **ON**-ba kapcsolva õket, a munkahengerek pozitív mozgást végeznek, **OFF**-ba kapcsolva negatív mozgást végeznek.
- Ha komplementerbe vannak állítva, 2 db kettõs mûködésû munkahenger köthetõ rájuk, a páratlanokat **ON**-ba kapcsolva, a munkahengerek pozitív mozgást végeznek, **OFF**-ba kapcsolva negatív mozgást végeznek.
- **RO[5]... RO[8]** 24V elektromos kimenetek (**RO[7]... RO[8]** van jelenleg bekõttve)

Megjelenítés:  gomb megnyomása a kezelõn, vagy  > I/O. A  ikonnal sokféle I/O közül választhatok, jelen esetben a **Robot**-ot kell választani!

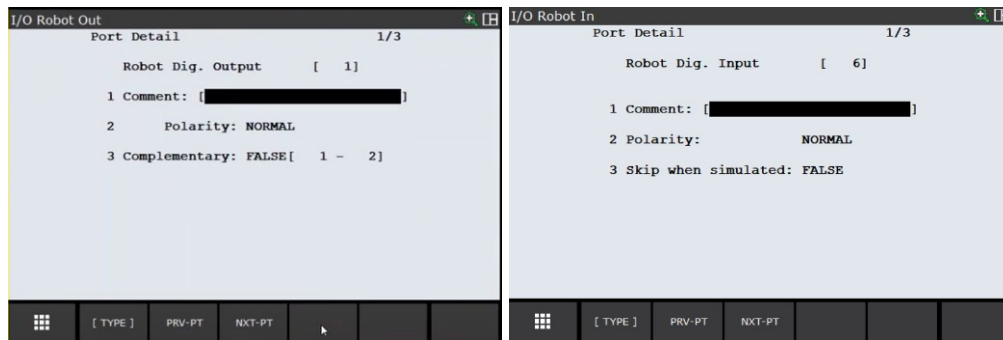
TYPE 1	TYPE 2
1 Cell Interface	1 Link Device
2 Custom	2 Flag
3 Digital	3
4 Analog	4
5 Group	5
6 Robot	6
7 UOP	7
8 SOP	8
9 Interconnect	9
0 -- NEXT --	0 -- NEXT --

A bemenet és a kimenet megjelenítése között a  gomb vált. Kijelõlve egy sorban egy státuszt  gomb bekapcsolja,  gomb kikapcsolja õt. **Alapesetben csak a kimenetek kapcsolhatók, a bemenetek kapcsolgatásához be kell kapcsolni a szimulációt!**

I/O Robot Out				I/O Robot Out			
#	SIM	STATUS		#	SIM	STATUS	
RO[1]	U	OFF	[]	RO[1]	U	OFF	[Open Gripper]
RO[2]	U	OFF	[]	RO[2]	U	OFF	[Close Gripper]
RO[3]	U	OFF	[]	RO[3]	U	OFF	[]
RO[4]	U	OFF	[]	RO[4]	U	OFF	[]
RO[5]	U	OFF	[]	RO[5]	U	OFF	[]
RO[6]	U	OFF	[]	RO[6]	U	OFF	[]
RO[7]	U	OFF	[]	RO[7]	U	OFF	[]
RO[8]	U	OFF	[]	RO[8]	U	OFF	[]

Ha az „U” betûn állunk, megjelenik a **SIMULATE (szimulálom)** és az **UNSIM (nem szimulálom)** lehetőség, vagyis szimulálhatom bármely kimenet, vagy bemenet **ON / OFF** értékét. Ez jól jõhet programteszteléskor, amikor még nem állnak készen a hardver elemek.

A kiválasztott soron állva **DETAIL** hatására megjelennek a beállítható részletek.



A **Comment**-ben elnevezhetem pl.: **[ON_Open]**. Az **RO[7]**-t nevezzük el **[ON_Open]**-nek, Az **RO[8]**-t nevezzük el **[ON_Close]**-nek!

A **NXT-PT** ikonnal a következő **RO**-ra ugorhatok, azt is elnevezhetem. **PRV-PT**-vel vissza léphetek.

2 Polarity: NORMAL. Ez azt jelenti, ha **ON**, akkor van kimeneti **24V**, ha **OFF**, akkor **0V**. Alul a **INVERSE** **NORMAL** ikonokkal átállítható inverzzé, ekkor ha **OFF**, akkor van kimeneti **24V**, ha **ON**, akkor **0V**.

Az egymás utáni **RO**-k (pl. az 7 és a 8) beállíthatók komplementernek, vagyis amikor az egyik **ON**, akkor a másik **OFF** és fordítva. A komplementer beállítás csak a páratlan sorszámú kimeneteken lehetséges. Ha kurzorral rajta állok a **FALSE** (hamis) feliraton,

3 Complementary: FALSE [7 - 8] megjelenik alul a **TRUE** **FALSE** lehetőség, átállíthatom **TRUE** (igaz) állapotra.

A polaritás és a komplementer átállítása után a változtatás érvénybe lépéséhez újra kell indítani a robotot!

Általános célú digitális bemenetek és kimenetek DI/DO

A bemenetekre 24 V-os különféle érzékelőket, szenzorokat, nyomógombokat köthetünk.

A kimenetekre 24 V-os fogyasztókat, lámpákat, motorokat köthetünk (figyelve a terhelésre). Elindíthat, megállíthat például egy szállítószalagot, vagy egy CNC megmunkáló gépet, esetleg egy másik robotot.

Az **R-30iB Mate** alaplapi vezérlő összesen **28 db** bemeneti és **24 db** kimeneti ponttal rendelkezik az alábbi címtartományokban.

Range

- **DI[101-120]**, általános célú digitális bemenet
- **DO[101-120]** általános célú digitális kimenet
- **DI[81-88]** választható célú digitális bemenet
- **DO[81-84]** választható célú digitális kimenet

Rack

A rack jelzi az I/O modul típusát. Pl.:

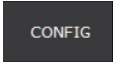
48 = R-30iB Mate alaplap (CRMA15, CRMA16)


Slot

R-30iB Mate alaplap (CRMA15, CRMA16) esetén a szám mindig **1**.

Beállítás, kezelés

Teljesen megegyezik a **RI/RO**-val az alábbi eltéréssel.

Alul megjelenik a  ikon, amit lenyomva megnézhetjük, melyik **RACK**, **SLOT**,

RANGE aktív. (Esetleg törölhetjük, bár a robot újraindulásakor újra felismeri.) A  ikon lenyomásával juthatunk vissza a státusz értékekhez.

#	SIM	STATUS	1/512
DO[1]	U	OFF	[]
DO[2]	U	OFF	[]
DO[3]	U	OFF	[]
DO[4]	U	OFF	[]
DO[5]	U	OFF	[]
DO[6]	U	OFF	[]
DO[7]	U	OFF	[]
DO[8]	U	OFF	[]
DO[9]	U	OFF	[]
DO[10]	U	OFF	[]
DO[11]	U	OFF	[]

Sorted by port number.

#	RANGE	RACK	SLOT	START	STAT.	1/8
1	DO[1- 8]	0	1	21	ACTIV	
2	DO[9- 16]	0	1	29	ACTIV	
3	DO[17- 20]	0	1	37	ACTIV	
4	DO[21- 24]	0	0	0	UNASG	
5	DO[25- 64]	0	2	1	ACTIV	
6	DO[65- 104]	0	3	1	ACTIV	
7	DO[105- 144]	0	4	1	ACTIV	
8	DO[145- 512]	0	0	0	UNASG	

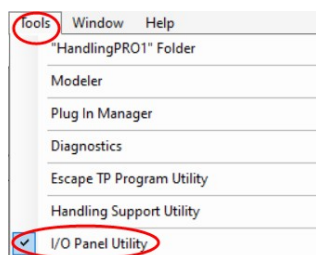
Device Name : Unknown

A felszerelt perifériás eszközök:

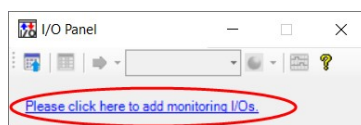
Tároló érzékelő (ON, ha van a tárolóban)	DI[101]
Fémérzékelő (ON, ha fém)	DI[102]
Fehér-fekete érzékelő (ON, ha fehér)	DI[103]
Piros nyomógomb (OFF, ha lenyomjuk)	DI[104]
Zöld nyomógomb (ON, ha lenyomjuk)	DI[106]
Zöld lámpa (ON esetén világít)	DO[103]
Piros lámpa (ON esetén világít)	DO[104]

Bemenetek és kimenetek megjelenítése a Roboguide-ban

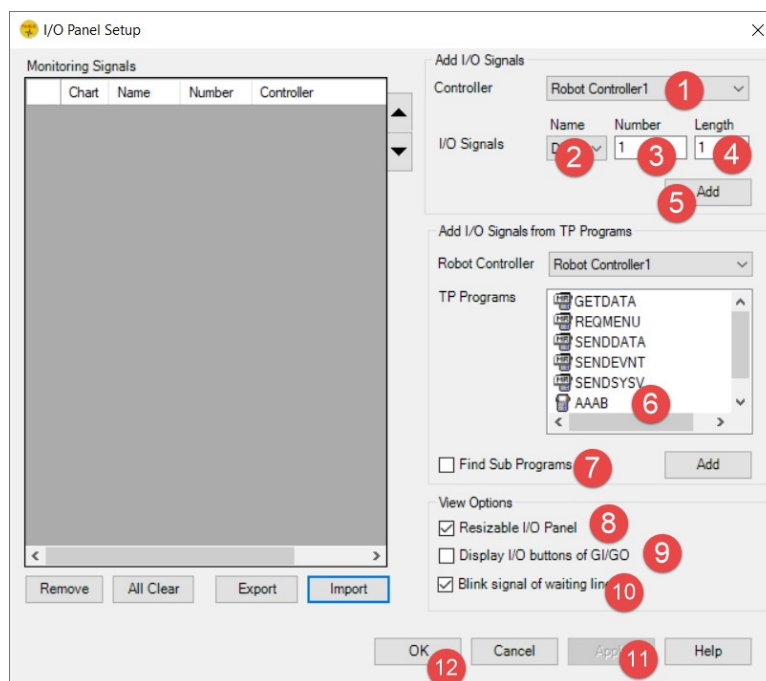
A Tools menü I/O Panel Utility pontjában be- és kikapcsolható a panel.



Kiválasztásakor felugrik egy panel, ami arra kér, kattintsunk a feliratára!

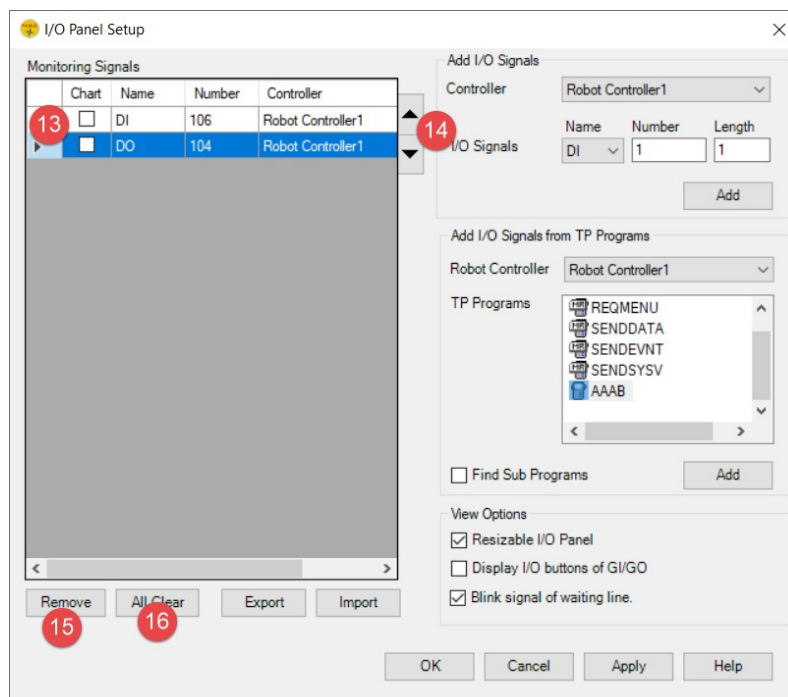


Rákattintva megjelenik egy beállító felület



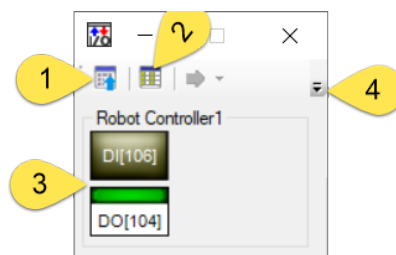
1. Melyik vezérlőre vonatkozzon (alap esetben az éppen használt)
2. I/O kiválasztása (a legjellemzőbbek: RI, RO, DI, DO)
3. Az I/O sorszámának beírása (pl.. 103)

4. Az előző számtól kezdve hány darab **I/O** legyen (3-at választva: **103, 104, 105**)
5. Hozzáadja a baloldali sötétszürke területhez a kiválasztott **I/O**-kat
6. Kiválaszthatunk egy **TP** programot, ha abban van egy vagy több **I/O**, automatikusan a baloldali sötétszürke területre teszi.
7. *Alprogramokhoz is hozzárendelhetünk*
8. Az elkészített **I/O** panelünk egérrel átméretezhető lesz
9. A **GI/GO** csoport *be- és kimenetek is választhatók lesznek*
10. Villogó jel a várakozáskor
11. Beállításaink alkalmazása
12. **OK**, beállítás befejezése



13. A hozzáadott **I/O**-k, sorok kijelölhetők
14. A kijelölt sorok föl-le mozgathatók
15. A kijelölt sor törölhető
16. Minden sor törlése

A beállítások után a következő panel jelenik meg:







1. Visszatérés a beállító panelhez
2. Részletek megjelenítése

3. A kiválasztott **I/O**-k megjelenítése
4. További kezelógombok megjelenítése

A részletes nézeten a megjegyzések és a további kezelógombok is láthatóak, illetve az értelmezéshez írt kis program mutatja az összefüggéseket.

Megjegyzések:

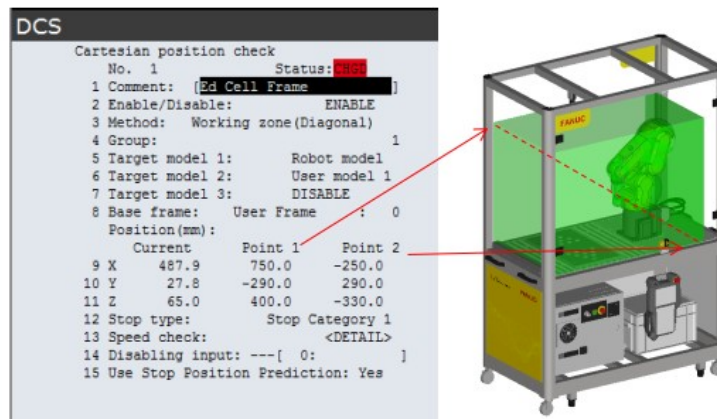
- A kikapcsolt **Input** kapcsoló szürke színű , a bekapcsolt zöld színű , színt változtatni jelenleg nem lehet.
- A kikapcsolt **Output** lámpa szürke színű , a bekapcsolt zöld színű. , színt változtatni jelenleg nem lehet.
- Az **Input** kapcsoló, nem nyomógomb, ezen jelenleg változtatni nem lehet.
- Az **Output** lámpa is be-, kikapcsolható egérrel, de ezt a futó program felülírja.
- Ha ezeket az **I/O**-kat használjuk, nem kell a **Teach Pendant**-on az **I/O**-kat szimulálni.
- A panel rögzíteni is tud **I/O** állapotokat, sőt idődiagramot is tud rajzolni belőle.

DCS (Dual Check Safety) biztonsági zóna

A DCS biztonsági zóna gondoskodik arról, hogy a robot ne tudjon átjutni a cella falán.

Egy DCS derékszögű pozícióellenőrző zónát egy **téglatest átlója** segítségével állíthatunk be, amely a zóna bal felső pontjától (**Point 1**) a jobb alsó pontig (**Point 2**) fut.

Egy zóna sok különböző, egyszerű zónából is összeállítható, így összetettebb zónákat is létrehozhatunk.






Ütközés

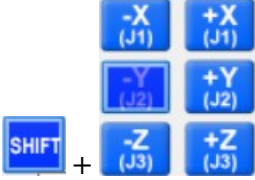
Mozgatva a robotkart a zóna felé egy adott pillanatban a robot megáll, és többé nem lehet mozgatni semmilyen irányba, kivéve elfelé a DCS biztonságos zóna határától.

A TP a következő hibaüzenetet adja: **SRVO-402 DCS Cart. pos. limit(No.1:Ed Ce**

Kioldás

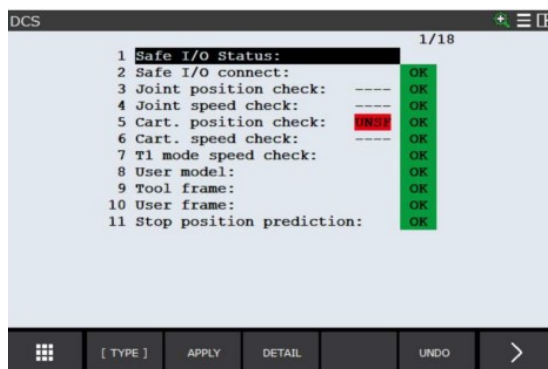
A robot hibaállapotból való kioldásához nyomjuk le **EGYSZERRE** a

DEADMAN+  +  gombokat mindaddig, míg a  **LED** ki nem alszik, és a motorok hangja nem hallható. Ezután léptessük el a robotot a zóna határától a

 gombokkal!

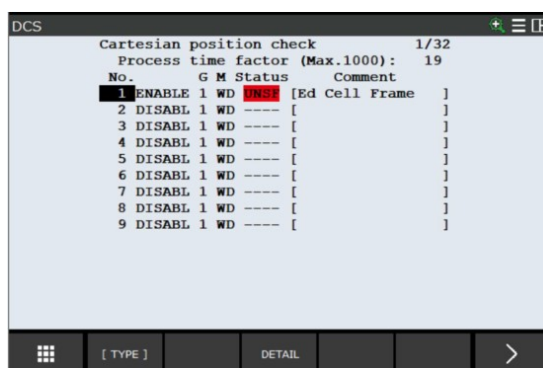
Biztonsági rendszer ellenőrzése

MENU>Next az első oldal alján>**System >DCS**



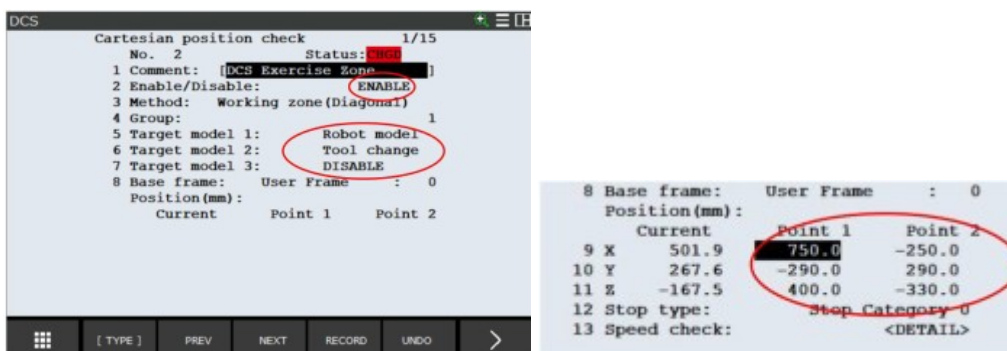
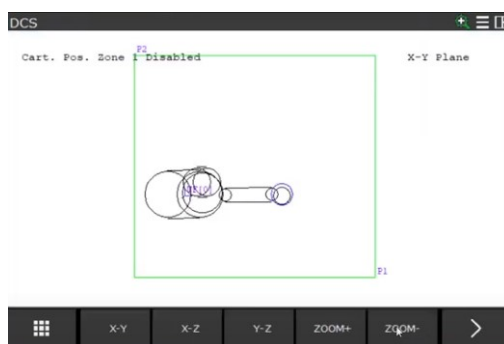
18 féle biztonsági dolgot ellenőriz a vezérlő. Ahol piros háttérű **UNSAFE** jelölés van, ott éppen le tiltási állapot van.

Menjünk a kurzorral a derékszögű koordinátarendszer pozíció ellenőrzésre (**Cart. Position check:**), majd az **DETAIL**-al lépünk be!



Az ábrán 32 féle tiltott terület lehetne, de csak az első van engedélyezve (**ENABLE**). **DETAIL** lenyomásával beléphetünk a részletekhez.

NEXT> Wiew ikonokkal megnézhetjük a biztonsági területet X-Y, X-Z, Y-Z vetületekben:







Itt módosíthatjuk az értékeket. Ha az előző ablakon egy üres sort választottunk, új biztonsági cellát vehetünk fel. A **PREV** gombbal kettőt visszalépve, az **APPLY** gombbal fogadjuk el a változtatásokat.

A rendszer megkérdezi, hogy „**Code number (master): ----**”, ahová be kell írni a mesterkódot, ezt beírva, megjelenik egy második ablak, ahol nyomjuk meg az **(OK)** billentyűt!



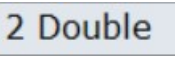


A **CHGD** azt jelenti, hogy a változtatások érvénybe léptetéséhez **újra kell indítanunk a robotot!**


A BIZTONSÁGI ZÓNÁT NE KAPCSOLJUK KI!

Robotkar mozgatása

Először a  nyomógomb többszöri lenyomásával váltsunk a megfelelő koordináta-rendszerbe! Kezdetnek megfelelő a , a , vagy a . Mozgatáskor figyeljük meg a mozgások közti lényeges különbségeket!

A **DEADMAN + SHIFT + MOZGATÓGOMBOK** segítségével tudjuk a robotkart mozgatni.


Kapcsoljunk kéttablakos üzemmódra ( + , majd , tegyük aktívvá a jobboldalit (, majd nyomjuk le a  gombot! Ezzel a jobb oldalon láthatóvá válnak a robotkar aktuális pozíciói.

Alul váltsunk át  kijelzésre, majd próbáljuk mind a hat koordinátát nullára állítani, kivéve a J5-öt, amit állítsunk 6°-ra (szingularitás elkerülése)!

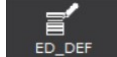
POSITION					
Joint			Tool: 1		
J1:	.000	J2:	.000	J3:	.000
J4:	.000	J5:	6.000	J6:	.000

Új pont felvétele

Álljunk **Tool** koordináta-rendszerbe!

A  lenyomásával mentsük el azt a pontot, ahol áll a robotkar! Ekkor felugrik egy menü, ahol a legutoljára használt négy beállítás közül választhatunk.

Select Motion 1/1	
1 J P[]	100% FINE
2 J P[]	100% CNT100
3 L P[]	100mm/sec FINE
4 L P[]	100mm/sec CNT100


Ha most nem választunk, hanem lenyomjuk az  ikont, szabadon szerkeszthetjük a választékot.



Edit Default Motion	
1:J P[]	100% FINE
2:J P[]	100% CNT100
3:L P[]	100mm/sec FINE
4:L P[]	100mm/sec CNT100

A mozgási parancssor

Felülírás

```
28:J @P[9] 100% CNT100
```

A sor elején az automatikusan kapott sorszám van. Ha közben nem mozgattam el a robotkart  ikonnal újra felvehetem ugyanezt a P9 pontot, de újra választhatok a négy féle alapbeállításból.

 +  egyszerűen felülírja a P9 koordinátáit választéklista nélkül (ha közben nem mozdultam el, nem történik változás).

A @ jel azt jelzi, hogy éppen ott áll a robotkar, vagyis nem mozdítottam el. Ha eltűnik a @ jel, jelzi, hogy elmozdítottam a robotkart

```
28:J P[9] 100% CNT100
```

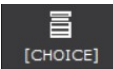
Új sorszámú pont akkor keletkezik, ha bármely nem pontot meghatározó soron állok (pl.: üres sor, [END], egyéb parancssor). **VIGYÁZAT, FELÜLÍRJA a sort!** Ilyen soroknál a

 +  nem működik.

A mozgás típusa

Amennyiben a parancssor mozgástípusán áll a kurzor

```
28:J P[9] 100% CNT100
```

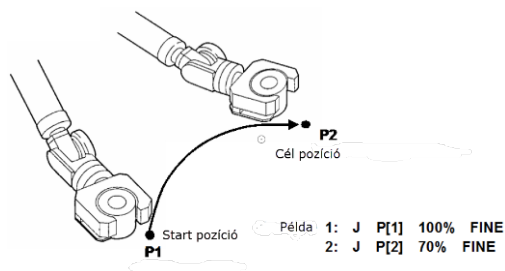
 megjelenik a  ikon, lenyomva megjelenik egy választéklista és módosíthatjuk mozgásunkat.

Motion Modify 1/1	
1	Joint
2	Linear
3	Circular
4	Circle Arc

Joint

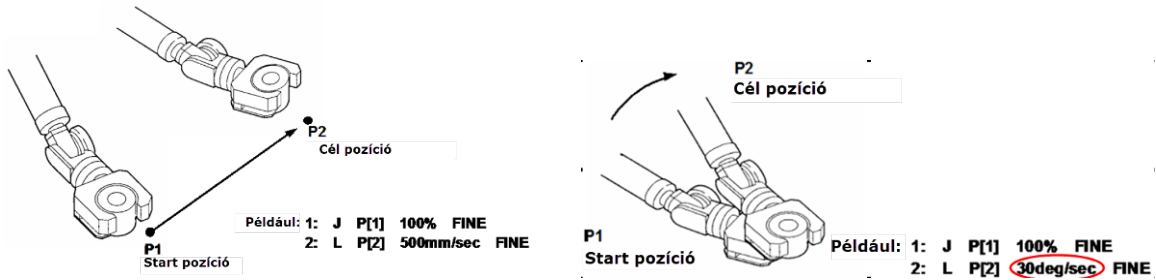
```
28:J P[9] 100% CNT100
```

A robotkar az általa optimálisnak (legkönnyebben kiszámolható) tartott úton jut el a pontba (ez általában valamilyen ív). **Bár a neve ugyanaz, ez más, mint a JOINT koordináta-rendszer!**



Linear 28: P[9] 3000mm/sec CNT100

A robotkar egyenes mentén jut el a pontba (vagy egy tengely körül fordul a szerszám).



Circular

A robotkar mozgása körívet ír le a kezdő pont, az átmeneti pont és a végpont között.

Programozáskor a három pontot vegyük fel J vagy L típusúra, majd a középsőt állítsuk át C típusúra.

```
1:J P[1] 100% CNT100  
2:J P[2] 100% CNT100  
3:J @P[3] 100% CNT100
```

```
1:J P[1] 100% CNT100  
2:C P[2]  
: P[...] 3000mm/sec CNT100  
3:J @P[3] 100% CNT100
```

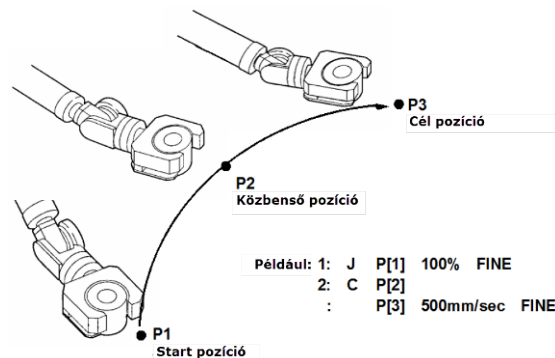
```
1:L P[1] 3000mm/sec FINE  
2:C P[2]  
: @P[3] 3000mm/sec FINE
```

A példában az 1. sorban P1 a kezdőpont. A 2. sorban lévő P2 mozgástípusát kell megváltoztatni C típusra, majd megmondani a végpont koordinátáját (a példában P3).

Ezután törölhetjük az eredeti 3. sort (akár maradhat is, úgy sem mozdul, bár a programfutást lassítja).

A végpont sorszámozása ekkor eltűnik, P3 mögötti paraméterek a teljes körívre fognak vonatkozni. Körmozgásnál a pontközelítés legyen FINE típusú!

Teljes kört két félkörből rakjunk össze!



A mozgás sebessége

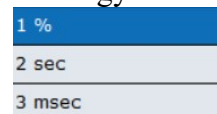
J típusnál

1%...100%-ban kell megadni (a szám csak egész lehet). Lehet egyebet is választani, ha a


mértékegységen áll a kurzor, és [CHOICE], (de nem célszerű).

Változtatni úgy kell, hogy a kurzorral ráállok az értékre, beírom a számértéket, majd

ENTER gombot nyomok.



L és C típusnál

1mm/sec...2500mm/sec-ban kell megadni (a szám csak egész lehet) .Lehet egyebet is választani, ha a mértékegységen áll a kurzor, és  , (de nem célszerű, kivéve a forgást).

1 mm/sec
2 cm/min
3 inch/min
4 deg/sec
5 sec
6 msec

Változtatni úgy kell, hogy a kurzorral ráállok az értékre, beírom a számértéket, majd **ENTER** gombot nyomok.

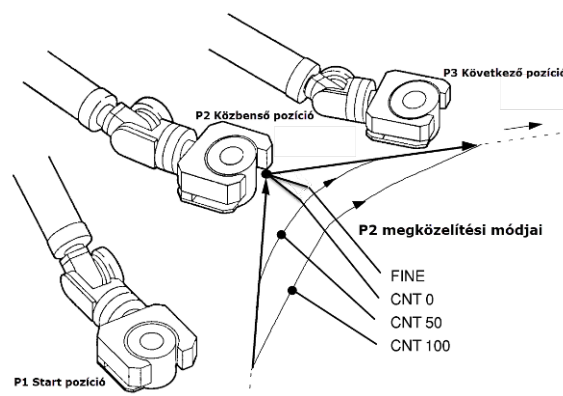
A mozgás pontközelítése

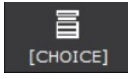
FINE

A robotkar eléri a célpontot és egy pillanatra megáll. A programfutás csak akkor lép a következő sorra, ha ez a pillanatnyi megállás megtörtént.

CNT 0...100

A robotkar általában nem éri el a célpontot, csak megközelíti és nem áll meg. A programfutás már a megközelítéskor továbblép a következő sorra. **CNT0** esetén el is éri a pontot, **CNT100** esetén elég nagy ívben elkerüli.




Változtatni a típuson úgy kell, hogy a kurzorral ráállok az értékre,  , majd választéklista.

1 Fine
2 Cnt

Változtatni az értéken úgy kell, hogy a kurzorral ráállok az értékre, beírom a számértéket, majd **ENTER** gombot nyomok.

Egyéb mozgás-módosítások

Amikor a programsor végén áll a kurzor `1:J P[1] 1$ FINE` a  hatására sok hasznos módosítási lehetőség nyílik meg. Talán kiemelném a később tárgyalandó

Tool_Offset,PR[] lehetőséget, illetve a **No option** a törléshez.

Motion Modify 1/3	Motion Modify 2/3	Motion Modify 3/3
1 No option	1 Tool_Offset	1 RO[]=...
2 ACC	2 Tool_Offset,PR[]	2 GO[]=...
3 Skip,LBL[]	3 TIME BEFORE	3 AO[]=...
4 BREAK	4 Skip,LBL,PR	4 PTH
5 Offset/Frames	5 TIME AFTER	5
6 Offset,PR[]	6 DISTANCE BEFORE	6
7 Incremental	7 DO[]=...	7
8 --next page--	8 --next page--	8 --next page--

Regiszterek (változók)

A regiszter egy **egész szám** vagy **tizedes tört** tárolására szolgáló változó. **200 db** regisztert használhatunk.

Értékadás

$R[i]$ = (érték) Ezzel az utasítással adhatunk értéket a regiszternek. Az $[i]$ a regiszter sorszáma, az érték lehet **Constans**, $R[i]$, $PR[i, j]$, $DI[i]$, $DO[i]$, $RI[i]$ $RO[i]$, és még többféle jel.

Műveletek regiszterekkel

A regiszterek összeadhatók, kivonhatók, szorozhatók, oszthatók egymással. Például:

$$1: R[2]=R[3]-R[4]+R[5]-R[6]$$

$$2: R[10]=R[2]*100/R[6]$$

Egy sorba legfeljebb 5 művelet írható. A „+” és „-” műveletek egy sorban keverhetők, ugyanígy a „*” és „/” műveletek is. A „+” és „-” azonban nem keverhető „*” és „/” jelekkel.

Az $R[i]$ = (érték) + (érték) utasítás két érték összegét tölti be egy megadott regiszterbe.

Az $R[i]$ = (érték) - (érték) utasítás két érték különbségét betölti egy megadott regiszterbe.

Az $R[i]$ = (érték) * (érték) utasítás két érték szorzatát tölti be egy megadott regiszterbe.

Az $R[i]$ = (érték) / (érték) utasítás két érték hányadosát tölti be egy megadott regiszterbe.

Az $R[i]$ = (érték) **MOD** (érték) utasítás a hányados maradékát (tizedespont utáni értéket) tölti be egy meghatározott regiszterbe.



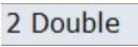


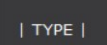
Az $R[i]$ = (érték) **DIV** (érték) utasítás két érték hányadosának egész részét tölti be egy megadott regiszterbe.

Regisztereket használhatunk például számolásra.

$R[1]=0$ Ezzel a program elején lenullázzuk a számlálót

$R[1]=R[1]+1$ Így tudunk egyesével számolni

Regiszterek elnevezése, értékeinek megtekintése

Kapcsoljunk kétablakos üzemmódra ( +  , majd ), tegyük aktívvá a jobboldalit (), majd nyomjuk le a  gombot! Ezzel a jobb oldalon láthatóvá válnak a regiszterek. Ha nem azokat látjuk,  ikonnal megnyithatunk egy menüt, ahonnan kiválaszthatjuk.

TYPE	1
1 Registers	
2 Position Reg	
3 String Reg	
4 KAREL Vars	
5 KAREL Posns	

Egymás alatt láthatjuk a **200 db** regisztert, ha a magyarázat helyére visszük a kurzort, és




ütünk, nevet adhatunk neki (nem kötelező, de bonyolultabb programnál jól jön).

DATA Registers		1/200
R[1:	counter]=5
R[2:]=0

A sor jobb oldalán leolvashatjuk a regiszter pillanatnyi értékét, a példában ez 5.

Regiszter programsor bevitele

Álljunk oda a kurzorral, ahová a programsort be akarjuk vinni, majd  [INST], lenyomása.

Válasszuk utasítást! Pl.: **Registers**

Instruction	1/3
1 Registers	
2 I/O	
3 IF/SELECT	
4 WAIT	
5 JMP/LBL	
6 CALL	
7 Miscellaneous	
8 --next page--	

Válasszuk műveletet! Pl. ha összeadást szeretnénk: $\dots = \dots + \dots$

REGISTER statement	1/1
1 $\dots = \dots$	
2 $\dots = \dots + \dots$	
3 $\dots = \dots - \dots$	
4 $\dots = \dots * \dots$	
5 $\dots = \dots / \dots$	
6 $\dots = \dots \text{DIV} \dots$	
7 $\dots = \dots \text{MOD} \dots$	
8 $\dots = (\dots)$	

Válasszuk, mi kerüljön az első... helyére! Pl.: **R[]**

REGISTER statement	1/1
1 R[]	
2 PR[]	
3 PR[i,j]	
4 SR[]	
5	
6	
7	
8 --next page--	

A megjelenő programsorba **3: R[...]=...+...** írjuk be az értéket,

3: R[8]=R[8]]+... majd **ENTER**

Válasszunk a második...helyre regisztertípust! Pl.: **R[]**

REGISTER statement 1/3	
1	R[]
2	Constant
3	DO[]
4	DI[]
5	RO[]
6	RI[]
7	GO[]
8	--next page--

A megjelenő programsorba 3: R[8]=R[]+... írjuk be az értéket,

3: R[8]=R[7]]+... majd **ENTER**

Válasszunk a harmadik...helyre konstanst! Pl.: **Constant**

REGISTER statement 1/3	
1	R[]
2	Constant
3	DO[]
4	DI[]
5	RO[]
6	RI[]
7	GO[]
8	--next page--

A megjelenő programsorba 3: R[8]=R[7]+Constant írjuk be az értéket,





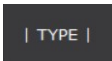
3: R[8]=R[7]+2] majd **ENTER**

A végeredmény: 3: R[8]=R[7]+2

Pozícióregiszterek

Amíg a **P[i]** jelű egyszerű pozíció csak egy, az aktuális program számára elérhető, addig a **PR[i]** jelű pozícióregiszter minden robotprogram számára elérhető (és természetesen a felelősen felülírásával minden ezt használó programot el is ront).


Pozícióregiszter létrehozása

Kapcsoljunk kéttablakos üzemmódra ( + , majd **2 Double**), tegyük aktívvá a jobboldalit (), majd nyomjuk le a  gombot! Ezzel a jobb oldalon láthatóvá válnak a pozícióregiszterek. Ha nem azokat látjuk,  ikonnal megnyithatunk egy menüt, ahonnan kiválaszthatjuk. 100 db pozícióregisztert használhatunk.

TYPE	1
1 Registers	
2 Position Reg	
3 String Reg	
4 KAREL Vars	
5 KAREL Posns	

A sorszám utáni mezőben el is nevezhetjük a pozícióregisztert, ez programozásnál hasznos lehet. A jobb oldalon a „**R**” azt jelenti, hogy ez a pozícióregiszter már rögzítve lett, míg a „*****” azt jelenti, hogy még nem létezik.

DATA Position Reg	
PR[1:Alaphelyzet]=R
PR[2:]=*

Ha már létezik a pozícióregiszter (az „**R**” jelzi), ráállunk a kurzorral a „**R**” jelre, és a 



billentyűkombinációval a robotot oda tudjuk mozgatni.

A




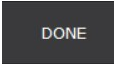

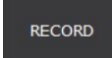
billentyűkombinációval felül tudjuk írni arra a pontra, ahol a robotkar éppen áll, ha már létezik „**R**”, és létre tudjuk hozni, ha még nem létezik „*****”.

A

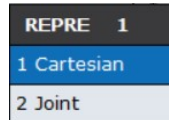


gombbal billentyűzetről is bevihetjük a koordinátákat, de ehhez persze tudnunk kell, a térbeli elhelyezkedést. Elsősorban **Offset** beállításánál hasznos.

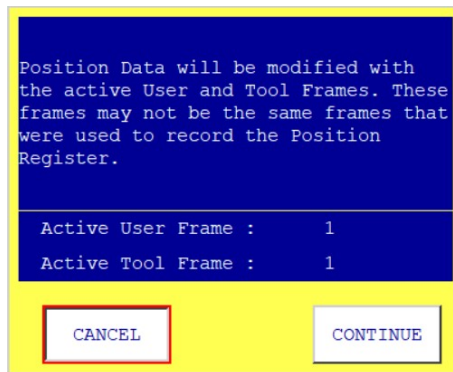
PR[1] UF:F UT:F					
J1	16.331	deg	J4	6.346	deg
J2	22.444	deg	J5	-52.983	deg
J3	-34.233	deg	J6	-182.991	deg
Position Detail					
PR[1:Alaphelyzet]=R
PR[2:]=*

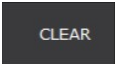
A számértékre ráállva a billentyűzetről bevihető az új szám, majd  gombbal továbbléphetünk a következőre. A  ikonnal tudjuk a változtatásokat véglegesíteni. Ilyenkor természetesen már **nem kell** a  .

A  ikonnal tudunk koordinátarendszert váltani.



Ha váltunk, megjelenik egy figyelmeztetés: „A pozícióadatok az aktív felhasználói és eszközköretekkel módosulnak. Előfordulhat, hogy ezek a keretek nem ugyanazok a keretek, amelyeket a pozícióregiszter rögzítésére használtak.”

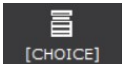


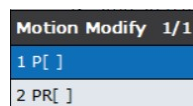
A  ikon lenyomása törli a kijelölt pozícióregisztert.

Pozícióregiszter bevitele a programsorba

A programban ráállok arra a pozícióra, amit át szeretnék alakítani pozícióregiszterré.

```
2:J @P[2] 100% CNT100
```

 lenyomására megjelenik egy menü, ahol kiválaszthatom a **PR[]** pontot.



Ha a **PR[]**-t választjuk, megjelenik a beviteli rész, és alul egy újabb menüsor.

```
2:J PR[... ] 100% CNT100 2:J PR[R[... ]] 100% CNT100
```



Alapértelmezetten a **DIRECT** az aktív, ami egy számérték bevitelét várja. Az **INDIRECT** egy regiszter értéket vár, a **CHOICE** visszavisz a **P[]** és a **PR[]** közötti váltáshoz.

Ha még nem létező pozícióregiszter számot adunk meg, a szám piros háttérrel jelenik meg.

```
2:J PR[3] 100% CNT100
```



A ikonnal ekkor billentyűzetről, koordináták megadásával felvehető az új pozícióregiszter.

```
PR[3] UF:F UI:F CONF:NDB 000
X *****.*** mm W *****.*** deg
Y *****.*** mm P *****.*** deg
Z *****.*** mm R *****.*** deg
Position Detail
1: R[1:counter]=R[1:counter]+1
2:J PR[3] 100% CNT100
3: R[8]=R[7]+Constant
[End]
```

Megfogási helyzet programozása Tool_Offset segítségével

Általában a munkadarab megfogása, vagy elengedése előtt (kb. 50 mm-el) célszerű egy pontot felvenni a célpozíció fölött, ahonnan a megfogó lassan tudja megközelíteni a célt. Ezt a következőképpen a legegyszerűbb:

Állítsunk be egy pozícióregisztert **Z=-50** értékre (a példában **PR[3]**)!

```
PR[3] UF:F UI:F CONF:NDB 000
X 0.000 mm W 0.000 deg
Y 0.000 mm P 0.000 deg
Z -50.000 mm R 0.000 deg
Position Detail
PR[ 1: ]=R
PR[ 2: ]=R
PR[ 3:Fogás felett ]=R
```

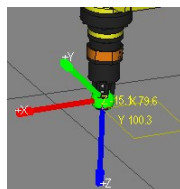
Vigyünk a robotot óvatosan fogásba, jegyeztessük meg ezt a pontot!

```
3:L @PR[2] 100mm/sec FINE
```

Másoljuk be ezt a sort még egyszer az eredeti fölé! Írjuk át a paramétereit gyorsabb mozgásra, továbbá egészítsük ki a **Tool_Offset, PR[3]** lehetőséggel!

```
2:L @PR[2] 1000mm/sec FINE
: Tool_Offset,PR[3:Fogás felett]
```

Z = -50 csak akkor működik helyesen, ha az aktív koordináta-rendszerünkben **+Z** lefelé mutat (**TOOL** rendszer). (Ha **WORLD** rendszert használunk, a **+Z** felfelé mutat, ezért itt **+50**-et kell programoznunk.)

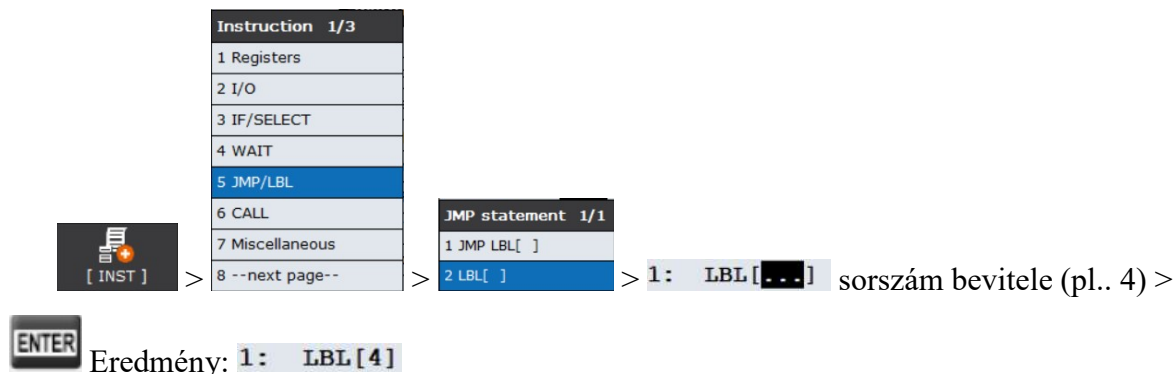


Egyszerű robotparancsok

Címkeutasítás (LBL[i])

LBL[i], ahol i sorszám, Pl.: LBL[1], (i értéke 1-től 32766-ig terjedhet).

A programban egy hely megjelölésére szolgál, ahová majd lehet ugrani.

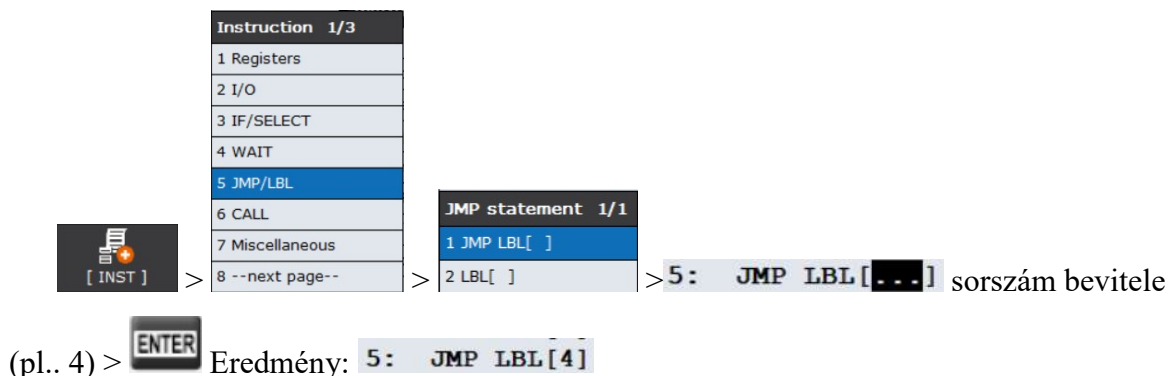


The screenshot shows a programming interface with two main panels. The left panel, titled 'Instruction 1/3', lists various instruction types: 1 Registers, 2 I/O, 3 IF/SELECT, 4 WAIT, 5 JMP/LBL (highlighted in blue), 6 CALL, 7 Miscellaneous, and 8 --next page--. To the left of this panel is a button labeled '[INST]'. The right panel, titled 'JMP statement 1/1', shows two options: 1 JMP LBL[] and 2 LBL[] (highlighted in blue). Below the right panel, the text '1: LBL[...]' is followed by 'sorszám bevitele (pl.. 4) >'. Below the entire interface, a button labeled 'ENTER' is shown, followed by the text 'Eredmény: 1: LBL[4]'.

Ugrás címkeutasításra (programon belül) (JMP LBL[i])

JMP LBL[i], ahol i sorszám, Pl.: JMP LBL[1], (i értéke 1-től 32766-ig terjedhet).

Utasít, hogy hányas címkére ugorjon a program.



The screenshot shows a programming interface similar to the one above. The left panel, titled 'Instruction 1/3', lists various instruction types: 1 Registers, 2 I/O, 3 IF/SELECT, 4 WAIT, 5 JMP/LBL (highlighted in blue), 6 CALL, 7 Miscellaneous, and 8 --next page--. To the left of this panel is a button labeled '[INST]'. The right panel, titled 'JMP statement 1/1', shows two options: 1 JMP LBL[] (highlighted in blue) and 2 LBL[]. Below the right panel, the text '5: JMP LBL[...]' is followed by 'sorszám bevitele (pl.. 4) >'. Below the entire interface, a button labeled 'ENTER' is shown, followed by the text 'Eredmény: 5: JMP LBL[4]'.

A példában az ötödik programsorhoz érve a program átugrik az első sorra, így létrehozható egy végtelen ciklus.

Programhívási utasítás (ugrás másik programra) (CALL)

A CALL PROGRAMNÉV utasítás a vezérlést egy másik, a programnévben megnevezett program vagy makró első sorába adja át. Amikor a meghívott program elér az END utasításhoz, a vezérlés visszakerül a hívó program CALL utáni sorára.

[INST] > Instruction 1/3

1 Registers
2 I/O
3 IF/SELECT
4 WAIT
5 JMP/LBL
6 CALL
7 Miscellaneous
8 --next page--

> CALL statement 1/1

1 CALL program
2 END

lista

- DROP
- KORIV
- PAKOLGAT
- PICK
- PROBA
- REGISTER

, ami után megjelenik egy már létező program-

, amiből tudunk választani.

Ha még nem létezik a meghívandó program, a **STRINGS** ikonnal a szövegbeviteli részhez jutunk, ahol begépelhetjük a programnevet. Ne felejtjük el később létrehozni a most megadott nevű programot! Az eredmény pl.: **6: CALL PAKOLGAT**

Program vége utasítás ([END])

A program végrehajtását ez az utasítás leállítja, majd a program az első sorra ugrik. Ha egy programot egy másik programból hívunk meg **CALL** utasítással, **END** hatására a vezérlés visszakerül a másik program **CALL** utasítását követő sorra.

[INST] > Instruction 1/3

1 Registers
2 I/O
3 IF/SELECT
4 WAIT
5 JMP/LBL
6 CALL
7 Miscellaneous
8 --next page--

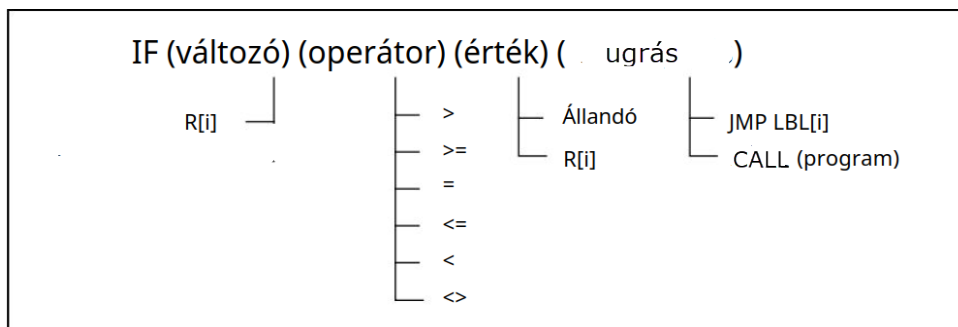
> CALL statement 1/1

1 CALL program
2 END

Az eredmény: **9: END**

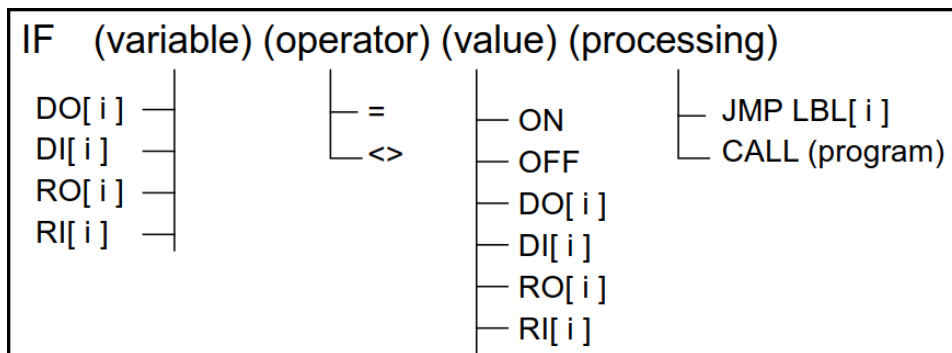
Feltételes elágazás ugrással (IF)

Egy feltételes elágazási utasítás elágazást hoz létre a program egyik helyéről a másikra, ha valamilyen feltétel teljesül. Összehasonlítja a regiszterben tárolt értéket egy másik értékkel, és ha az összehasonlítási feltétel teljesül, végrehajtja a feldolgozást, ugrik valahová.



Példa: **7: IF R[1] = R[2], JMP LBL[1]** vagyis ha az **R[1]** regiszter értéke egyenlő az **R[2]** regiszter értékével, az **LBL[1]** címkéjű sorra ugrik.

Az eljárás nem csak regiszterrel, hanem bemenetek és kimenetek értékeivel is működik, bár ott a bináris mivolta okán kevesebb reláció lehetséges!



Példa: **11: IF DI[103] = ON, CALL PROGRAM2** vagyis ha az **DI[103]** bemenet be van kapcsolva, meghívja a **PROGRAM2** nevű programot.

IF utasításban egyetlen sorában több feltétel is megadható a logikai operátorok **AND** és **OR** használatával. Tiltva van az **AND** és **OR** logikai operátorok együttes használata! Legfeljebb **5** feltétel adható meg egy parancssorban.

MENETE:

> 6: IF DI[...]=... .. >

> 6: IF DI[103]=ON ... >

> 6: IF DI[103]=ON, CALL PROGRAM2 >

> 6: IF DI[103]=ON, CALL PROGRAM2

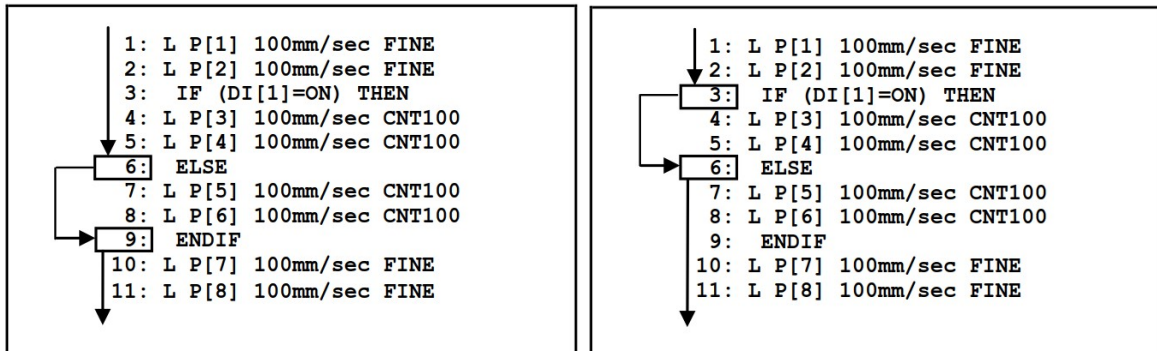
IF_THEN / ELSE / ENDIF utasításcsoport

Az **IF_THEN** utasítást az **ELSE** és az **ENDIF** utasítással hármasával használjuk. Ha a feltétel teljesül, az **IF_THEN** utasítás alatti sorok végrehajtásra kerülnek az **ENDIF**-ig, úgy

hogy az **ELSE** utasítás alatti sorok nem kerülnek végrehajtásra. Ellenkező esetben, vagyis ha a feltétel nem teljesül, az **ELSE** és az **ENDIF** közötti sorok kerülnek végrehajtásra.

Példa: Ha az **IF_THEN** utasítás feltétele a 3. sorban teljesül (**DI[1]=ON**), a 4. és 5. sor végrehajtásra kerül. Az **ELSE** sornál a végrehajtás az **ENDIF** sorra ugrik, a 7. és 8. sor nem kerül végrehajtásra.

Ha az **IF_THEN** utasítás feltétele a 3. sorban nem teljesül (**DI[1]=OFF**), a végrehajtás az **ELSE** sorra ugrik, a 4. és 5. sor nem kerül végrehajtásra. Ezt követően az **ELSE** utasítások alatti sorok végrehajtásra kerülnek.



MENETE: [INST] >

Instruction 1/3
1 Registers
2 I/O
3 IF/SELECT
4 WAIT
5 JMP/LBL
6 CALL
7 Miscellaneous
8 --next page--

 >

IF statement 2/2
1 IF (...) THEN
2 ELSE
3 ENDIF
4 SELECT R[]=...
5 <select> =...
6 <select>ELSE
7
8 --next page--

 >

Mixed Logic 1/4
1 (
2 DI[]
3 DO[]
4 R[]
5 F[]
6 On
7 Off
8 --next page--

 > 6: IF (DI[...]) THEN >

6: IF (DI[103]) THEN > [ENTER] > 6: IF (DI[103]) THEN > [INSERT] >

Mixed Logic 2/3
1 DIV
2 MOD
3 =
4 <>
5 <
6 <=
7 >
8 --next page--

 >

Mixed Logic 1/4
1 (
2 DI[]
3 DO[]
4 R[]
5 F[]
6 On
7 Off
8 --next page--

 > 6: IF (DI[103]=ON) THEN

ILL.: [INST] >

Instruction 1/3
1 Registers
2 I/O
3 IF/SELECT
4 WAIT
5 JMP/LBL
6 CALL
7 Miscellaneous
8 --next page--

 >

IF statement 2/2
1 IF (...) THEN
2 ELSE
3 ENDIF
4 SELECT R[]=...
5 <select> =...
6 <select>ELSE
7
8 --next page--

 > 5: ELSE VAGY >

IF statement 2/2
1 IF (...) THEN
2 ELSE
3 ENDIF
4 SELECT R[]=...
5 <select> =...
6 <select>ELSE
7
8 --next page--

 > 5: ENDIF

Várakozás utasítás (WAIT)

A várakozási utasítás arra szolgál, hogy leállítsa a program végrehajtását egy meghatározott ideig, vagy amíg egy feltétel, feltételcsoport nem teljesül.

Idővel megadott várakozás

Általában a munkadarabok elengedése vagy megfogása után használjuk, hogy biztosan a tárolóba essen a munkadarab, mielőtt a robotkar elindul visszafelé, illetve biztos legyen a megfogás.

Instruction	1/3	Wait statements	1/1
1	Registers	1	WAIT ... (sec)
2	I/O	2	WAIT ...=>...
3	IF/SELECT	3	WAIT ...<>...
4	WAIT	4	WAIT ...<=...
5	JMP/LBL	5	WAIT ...>=...
6	CALL	6	WAIT ...>...
7	Miscellaneous	7	WAIT ...>=...
8	--next page--	8	WAIT (...)

[INST] > 6: WAIT 0.00 (sec) >

6: WAIT 0.2 (sec) (érték megadása tizedespont használatával) > ENTER >

5: WAIT .20 (sec)

Feltételes várakozás

Például akkor használjuk, ha a tárolóból kifogy a munkadarab, a robotkar leáll, feltöltjük a tárolót, de addig várakozik a robot, amíg a **Start** nyomógombot le nem nyomjuk.

Instruction	1/3	Wait statements	1/1	Wait statements	1/3
1	Registers	1	WAIT ... (sec)	1	R[]
2	I/O	2	WAIT ...=>...	2	DO[]
3	IF/SELECT	3	WAIT ...<>...	3	DI[]
4	WAIT	4	WAIT ...<=...	4	RO[]
5	JMP/LBL	5	WAIT ...>=...	5	RI[]
6	CALL	6	WAIT ...>...	6	GO[]
7	Miscellaneous	7	WAIT ...>=...	7	GI[]
8	--next page--	8	WAIT (...)	8	--next page--

[INST] > 6: WAIT DI[...]=... >

6: WAIT DI[106]=... > ENTER > 2 On > 6: WAIT DI[106]=ON

Az **On+** felfutó élre ad tüjelet, az **Off-** lefutó élre ad tüjelet!

Feladat: Egy átrakás során, ha kifogy a tárolóból a munkadarab, a robot álljon le, és csak, ha újból érzékel munkadarabot a tárolóban, és ha lenyomjuk a **Start** gombot, kezdjen újra pakolni!

A **DI[101]** sorszámú kapus optikai érzékelő akkor ad jelet, ha van munkadarab a tárolóban, a start gomb **DI[106]** sorszámú és **NO** típusú, akkor ad jelet, ha lenyomjuk.

Megoldás (mozgások nélkül):

```
1: WAIT DI[101]=ON AND DI[106]=ON
:
2: LBL[1]
[End]
```


Mozgatott teher beállítása (PAYLOAD[i])

Nagyobb terhek mozgatásakor fontos azok tömege, tömegközéppontja, inercianyomaté-
kai. Ezeket egy külön felületen, adott sorszám alatt beállíthatjuk, majd programból hivatkoz-
hatunk rá, pl.:

The screenshot shows the 'Instruction 2/3' menu with '2 Payload' selected. Below the menu, a sequence of commands is shown: '[INST] > 1 PAYLOAD[...] > 4: PAYLOAD[...] > ENTER > 4: PAYLOAD[7]

Sebesség felülbírlás megváltoztatása (OVERRIDE)



A gombokkal akármekkora állítottuk be a sebesség felülbírlását (pl.: 100%), ha a programban kiadjuk ezt az utasítást, felülírja. Ha az első sorok valamelyikében kiadjuk a **4: OVERRIDE=25%** parancsot, a véletlenül nagy sebességen felejtett robotkart induláskor lelassítja az adott értékre.

The screenshot shows the 'Instruction 1/3' menu with '7 Miscellaneous' selected. Below it, the 'Miscellaneous statements 1/2' menu is shown with '4 OVERRIDE' selected. A sequence of commands is shown: '[INST] > 4: OVERRIDE=Constant > 4: OVERRIDE=25% > ENTER > 4: OVERRIDE=25%

Megjegyzés (REMARK)

A programok jobb megértése végett magyarázó sorokat szúrhatunk be. A sor elején egy felkiáltó jel áll, ha be van kapcsolva a „Colour” funkció, a háttérszín sárga. 32 db magyar ékezet nélküli karakter lehet (* _ @ is lehet).

The screenshot shows the 'Instruction 1/3' menu with '7 Miscellaneous' selected. Below it, the 'Miscellaneous statements 1/2' menu is shown with '5 Remark' selected. A sequence of commands is shown: '[INST] > 4: ! > 4: !Kettes bemenet

Létezik nyelvfüggő megjegyzés is, de mivel a robot nem tud magyarul, ez egyenértékű a megjegyzéssel **2 Multi-Ing Remark**.

Felkiáltó jel helyett -- jel látható a sor elején **4: --Kettes bemenet**

FOR / ENDFOR ciklusparancsok

A **FOR / ENDFOR** egy olyan utasításpár, amely megismétli a ciklust a **FOR** és **ENDFOR** utasításokon belül meghatározott számban.

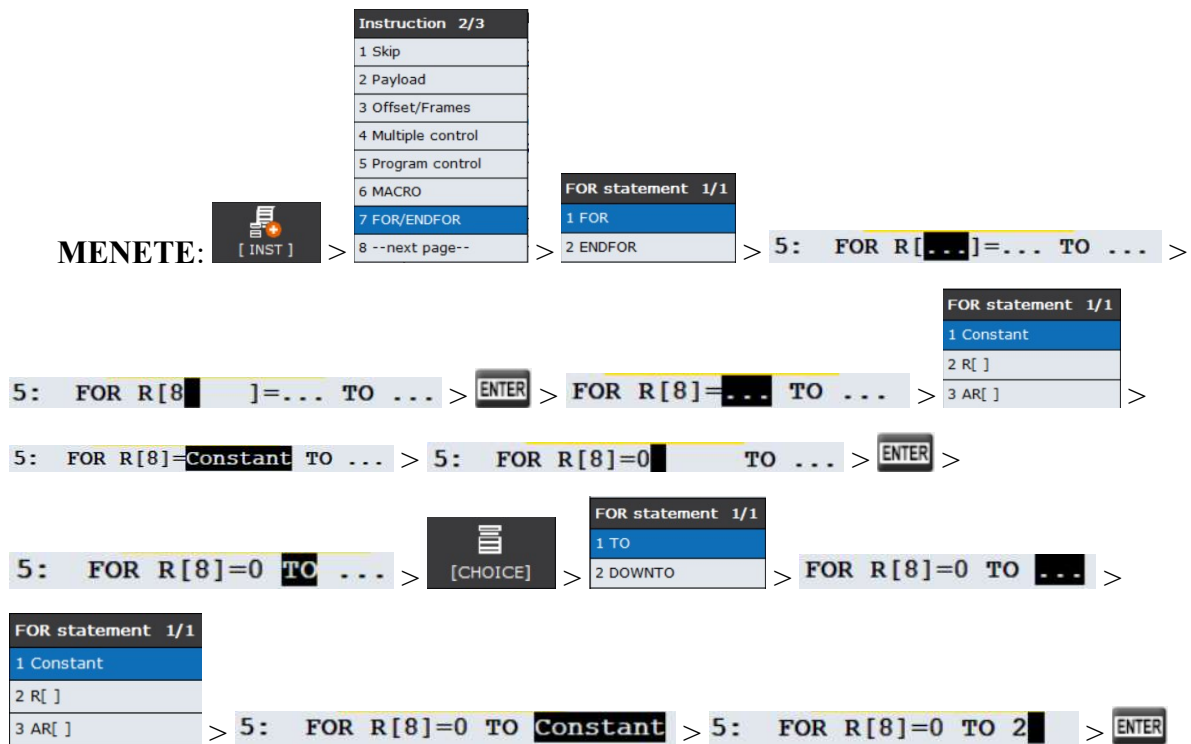


Ha **TO** helyett **DOWNTO** utasítást használunk, visszafelé számol. Az állandók értéke -32767...32766 között lehet. A ciklust az **ENDFOR** zárja, ha kifelejtyük, hibát jelez a program. Maximum 10 db **FOR / ENDFOR** egymásba ágyazható. A következő példában a **FOR / ENDFOR** közötti rész ötször végrehajtódik, csak utána lép a 7. sorra.

```
1:  FOR R[1]=1 TO 5
2:  L  P[1] 100mm/sec CNT100
3:
4:  L  P[2] 100mm/sec FINE
5:  L  P[3] 100mm/sec CNT100
6:  ENDFOR
7:  L  P[4] 100mm/sec CNT100
[End]
```

A következő példa egymásba ágyazott ciklusokat mutat. Ügyeljünk, hogy más legyen a két ciklusváltozó (**R[1]**, **R[2]**)!

```
1:  FOR R[1]=1 TO 5
2:  L  P[1] 100mm/sec CNT100
3:  FOR R[2]=5 DOWNTO 1
4:  L  P[2] 100mm/sec FINE
5:  L  P[3] 100mm/sec CNT100
6:  ENDFOR
7:  L  P[4] 100mm/sec CNT100
8:  ENDFOR
[End]
```



EREDMÉNY: 5: FOR R[8]=0 TO 2

A FOR az R[i] értékét kezdetire állítja, 0 kezdeti érték esetén nullázza, tehát nem szükséges külön nullázás sor!


Makró

A **Makró** általában egy rövid program, amelyet gyakran ismétlődő funkciók végrehajtására használnak, pl.: bizonyos **Teach Pendant** gombok lenyomására akkor is végrehajtódik, ha egyébként a roboton egy mozgásprogram sem fut éppen. A **Makró** program bármely másik robotprogramból is meghívható.

Léteznek rendszerszintű **Makrók**, amik előre meghatározottak, és a felhasználó nem módosíthatja őket. Ezeket "S" betű jelzi a **Menu>>Setup>>Macro** útvonalon elérhető **Macro Setting** képernyő jobb szélén.

150 db különféle makrót használhatunk, de ebből vannak rendszerszintű makrók is.

		115/150	
Instruction name	Program	Assign	
106 []	--	[0]
107 []	--	[0]
108 []	--	[0]
109 [Send Event]	[SENDEVNT]	--[0]S
110 [Send Data]	[SENDDATA]	--[0]S
111 [Send SysVar]	[SENDSYSV]	--[0]S
112 [Get Data]	[GETDATA]	--[0]S
113 [Request Menu]	[REQMENU]	--[0]S
114 []	--	[0]

Feladat: Írjunk egy olyan programot, ami  nyomógomb egyszeri lenyomására kinyitja a megfogót, másodszori lenyomására becsukja!


Ugyanúgy kell a programot megírni, mint bármely más programot. , majd , majd nevezzük el, pl.: **Nyit_Zar** névre, majd !

A program:

```

ROBOT NYIT_ZAR
1: IF (RO[7:On_Open]=ON) THEN
2: RO[7:On_Open]=OFF
3: ELSE
4: RO[7:On_Open]=ON
5: ENDIF
[End]
    
```

MENU 1	SETUP 1
1 UTILITIES	1 Prog Select
2 TEST CYCLE	2 ZDT Client
3 MANUAL FCTNS	3 General
4 ALARM	4 Frames
5 I/O	5 Macro
6 SETUP	6 Ref Position
7 FILE	7 Port Init
8	8 Ovrld Select
9 USER	9 User Alarm
0 -- NEXT --	0 -- NEXT --

Alakítsuk programunkat **Makróvá!**  útvonalon a beállítási képernyőhöz „**Macro Setting**” jutunk.

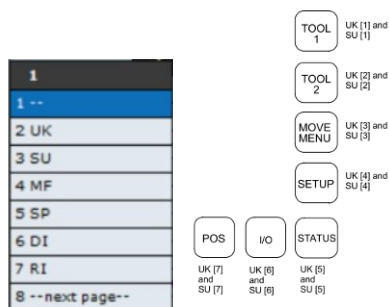
Az „**Instruction name**” oszlopban adnunk kell nevet a **Makrónak**, ami nem feltétlenül kell, hogy ugyanaz legyen, mint az előbb megírt program neve, most legyen: **Nyit_Zar_Mac**

Instruction name	Program	Assign
1 [Nyit_Zar_Mac]	--[0]

Lépünk a „**Program**” oszlopba, a [CHOICE] ikonnal nyissunk meg egy listát, és válasszuk ki az előbb megírt programunkat, majd **ENTER**!



Az „**Assign**” oszlopban válasszuk ki a **Makró** típusát! Ehhez álljunk az oszlopra, majd nyomjuk meg a [CHOICE] ikont, és válasszuk a listából!



A **SU** jelentése, hogy **SHIFT** gombot is le kell nyomni, az **UK** pedig, hogy csak magában a választott gombot. A sorszám jelöli a gombot a fenti ábra alapján, ami 1...7 lehet. Egy gombhoz csak egy makró rendelhető, ha már foglalt, a meglévőt tegyük más sorszámra, vagy töröljük, ha már nem kell.

```
Instruction name Program Assign
1 [Nyt_Zar_Mac ] [NYIT_ZAR]UK[ 1]
```

Megjegyzés:

- Alapesetben az **UK** nem választható, csak a **SU**, az **UK**-hoz meg kell változtatnunk a „**Gropm mask**” értékét a későbbiekben leírt módon.
- Az **MF** típus beállítása esetén **TOOL 1** lenyomására az 1...10 sorszámú, **TOOL 2** lenyomására 11...20 sorszámú makrók közül választhatunk egy menüből, a kiválasztottat kijelölve **SHIFT +EXEC** lenyomásával futtathatunk.

```
Instruction
1 Nyit_Zar_Mac
Press SHIFT-EXEC(F3) to run program
```

A programok listájában a **Makró** típusú programok kiterjesztése **MR**. A makrók beállításait (**Macro Setting**) viszont a **SYSMACRO.SV** tartalmazza.

```
3 NYIT_ZAR MR [ ]
```

Ebben a beállításban még a **DEADMAN** nyomógomb lenyomása is szükséges a működtetéshez. Ha a **Makróban** nincs mozgás-utasítás, ezt ki is kapcsolhatjuk. Ehhez a **SELECT** lenyomásával válasszuk ki a **Makró** programunkat, majd a **DETAIL** ikonnal menjünk a részletekhez!

A

```

2 Sub type:      [Macro      ]
3 Comment:      [           ]
4 Group mask:   [1,*,*,*,*,*,*]

```

felületen az **1** karaktert ki kell cserélni* ka-

rakterre a lenti lehetőségekkel , majd **END!**

A **Roboguid**-ban leellenőrizhetjük a helyes működést, ha bekapcsoljuk a megosztott panel jobb oldalán a robot kimenetek megjelenítését, és a **Teach Pendanton** nyomogatjuk a gombot.

ROBOT NYIT_ZAR		ROBOT I/O Robot Out	
	1/6	#	SIM STATUS
1:	IF (RO[7:On_Open]=ON) THEN	RO[1] U	OFF []
2:	RO[7:On_Open]=OFF	RO[2] U	OFF []
3:	ELSE	RO[3] U	OFF []
4:	RO[7:On_Open]=ON	RO[4] U	OFF []
5:	ENDIF	RO[5] U	OFF []
	[End]	RO[6] U	OFF []
		RO[7] U	ON [On_Open]
		RO[8] U	OFF []

Makró meghívása másik programból

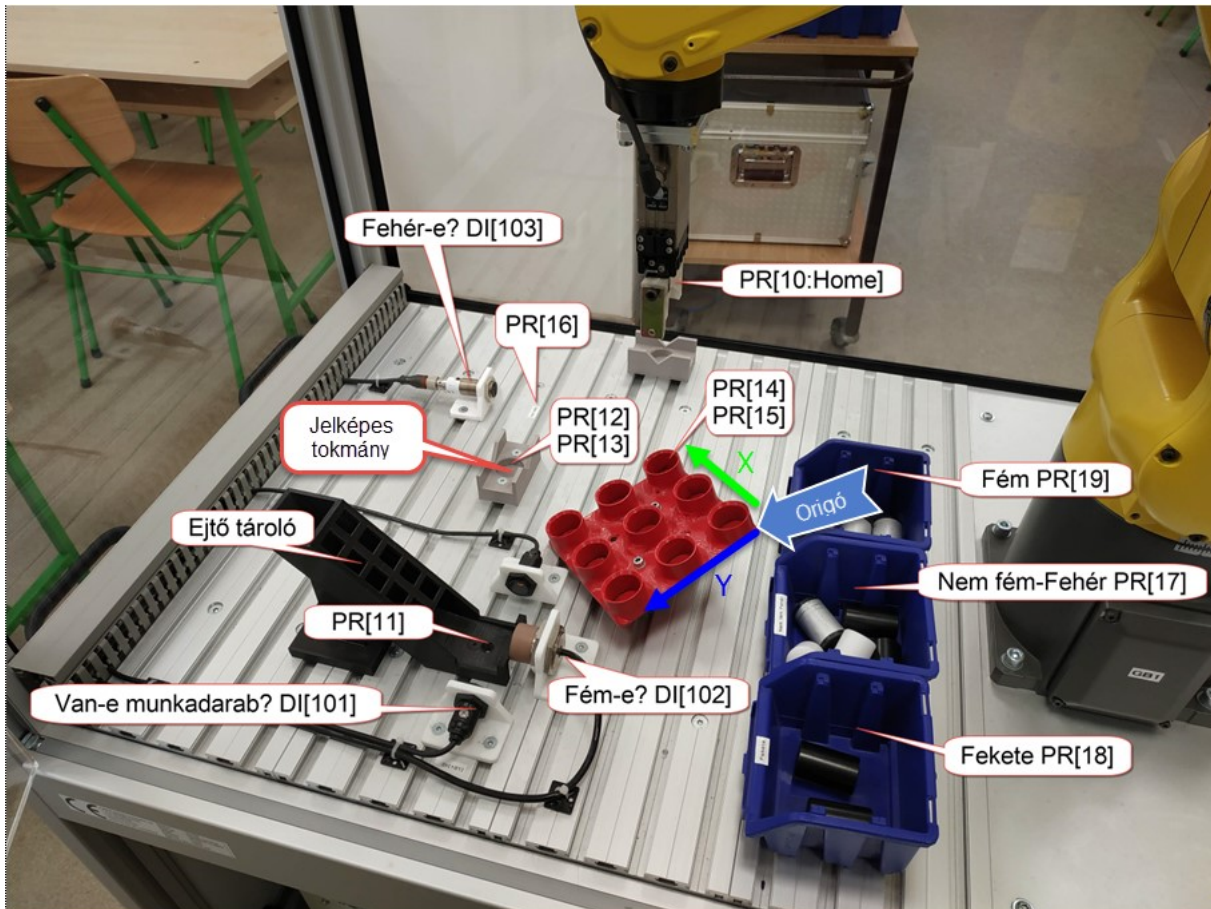
A megfelelő soron állva , majd , majd .

Ha nem egy **TP** programot akarunk meghívni, hanem egy **Makrót**, az ikonok közül azt kell lenyomnunk , majd a megjelenő lenyíló listából kitallóznunk, majd **ENTER**.

Az eredmény: **6: CALL NYIT_ZAR** .

Szín szerinti válogatás feladat

Az alábbi ábrán a robot felszerszámozása, beállított pozícióregiszterei, IO-i, láthatóak:



A pozícióregisztereket a valós roboton, **ha még nincsenek felvéve**, vegye fel, a **Roboguide**-ban töltsse be külső forrásból!

A felhasználói kereteket, **ha még nincsenek felvéve**, vegye fel, a **Roboguide**-ban töltsse be külső forrásból!

Készítse el az alábbi mozgás programját!

Kezdeti beállítások:

- A szerszámkeretet állítsuk be az 1. sz. szerszámmra (Z=115)!
- A koordináta-rendszert állítsuk be World típusra!
- A sebesség felülírást állítsuk be 30%-ra!
- A robot nyissa ki a megfogót!
- A piros visszajelző lámpa világítson, a zöld nem!
- A robot vegye fel a HOME pozíciót (J, CNT100)!

Mozgás:

- A robot várakozzon addig, amíg teszünk a tárolóba munkadarabot, és amíg lenyomjuk a start gombot!
- Ha elindul a robot, a piros lámpa kialszik, a zöld lámpa bekapcsol!
- Menjen a robot a tároló fölé 60 mm-el (J, CNT100)
- Lassan közelítsen a tárolóra (L, FINE)
- Fogja meg a munkadarabot!
- Emelje ki a munkadarabot a tároló fölé 60 mm-el (J, CNT100)
- Menjen a robot az 1. átrakó fölé 60 mm-el (J, CNT100)
- Lassan közelítsen az 1. átrakóra (L, FINE)
- Engedje le a munkadarabot!
- Menjen a robot az 1. átrakó fölé 60 mm-el (J, CNT100)
- Forduljon másik fogásra a robot az 1. átrakó fölött 60 mm-el (J, CNT100)
- Lassan közelítsen az 1. átrakóra (L, FINE)
- Fogja meg a munkadarabot!
- Emelje ki a munkadarabot a 1. átrakó fölé 60 mm-el (J, CNT100)
- Menjen a robot a 2. átrakó fölé 60 mm-el (J, CNT100)
- Lassan közelítsen a 2. átrakóra (L, FINE)
- Engedje le a munkadarabot!
- Menjen a robot a 2. átrakó fölé 60 mm-el (J, CNT100)
- Forduljon másik fogásra a robot a 2. átrakó fölött 60 mm-el (J, CNT100)
- Lassan közelítsen a 2. átrakóra (L, FINE)
- Fogja meg a munkadarabot!
- Emelje ki a munkadarabot a 2. átrakó fölé 60 mm-el (J, CNT100)
- Vigye el a robot a munkadarabot a színvizsgálati ponthoz!
- Ha fehér színű, az alaphelyzeten keresztül vigye a fehér tároló fölé, ott engedje el, majd ismét menjen az alaphelyzetbe!
- Ha fekete színű, az alaphelyzeten keresztül vigye a fekete tároló fölé, ott engedje el, majd ismét menjen az alaphelyzetbe!
- Ha még van munkadarab a tárolóban, vegye ki a következőt, ha nincs, kapcsolja ki a zöld lámpát, kapcsolja be a piros lámpát, és ugorjon a program elejére!

Tesztelje, finomítsa a programját!

A használt pozícióregiszterek beállításai (a valóságos roboton felvéve):

[1,10] = **'MaczikHOME'**

Group: 1 Config: N U T, 0, 0, 0

X: 360.816 Y: .782 Z: -69.480

W: -179.014 P: 2.898 R: 1.185

[1,11] = **'MacTarolo'**

Group: 1 Config: N U T, 0, 0, 0

X: 452.315 Y: 153.952 Z: -275.832

W: -179.522 P: 2.296 R: 6.166

[1,12] = **'MacAtrako1'**

Group: 1 Config: N U T, 0, 0, 0

X: 507.274 Y: -35.284 Z: -262.284

W: 177.517 P: .254 R: -89.288

[1,13] = **'MacAtrakoFord'**

Group: 1 Config: N U T, 0, 0, 0

X: 506.471 Y: -35.237 Z: -280.519

W: -179.781 P: 2.490 R: -.060

[1,14] = **'Atrko2'**

Group: 1 Config: F U T, 0, 0, 0

X: 359.296 Y: -42.605 Z: -257.014

W: -167.058 P: -73.170 R: 18.855

[1,15] = **'Atrako2Ford'**

Group: 1 Config: N U T, 0, 0, 0

X: 370.840 Y: -38.337 Z: -244.265

W: 175.863 P: 16.219 R: -21.480

[1,16] = **'Szinvizsgal'**

Group: 1 Config: N U T, 0, 0, 0

X: 446.822 Y: -142.687 Z: -263.314

W: -179.216 P: 3.593 R: -20.632

[1,17] = **'Feher'**

Group: 1 Config: N U T, 0, 0, 0

X: 219.212 Y: 87.965 Z: -185.883

W: -179.014 P: 2.899 R: 1.186

[1,18] = **'Fekete'**

Group: 1 Config: N U T, 0, 0, 0

X: 219.211 Y: 205.303 Z: -185.891

W: -179.020 P: 2.898 R: 1.196

[1,19] = **'Fem'**

Group: 1 Config: N U T, 0, 0, 0

X: 219.171 Y: -20.205 Z: -185.913

W: -179.016 P: 2.907 R: 1.187

[1,20] = **'Folotte'**

Group: 1 Config: N D B, 0, 0, 0

X: 0.000 Y: 0.000 Z: 60.000

W: 0.000 P: 0.000 R: 0.000

Egy lehetséges megoldás TP programja:

```
1: UTOOL_NUM=1 ;
2: UFRAME_NUM=0 ;
3: OVERRIDE=30% ;
4: RO[7:ON :Megfogó On Open]=ON ;
5: DO[104:ON :Piros lámpa]=ON ;
6: DO[103:OFF:Zöld lámpa]=OFF ;
7: J PR[10:MaczikHOME] 100% CNT100 ;
8: WAIT DI[101:OFF:Mdb.érzékelő]=ON AND DI[106:OFF:Start gomb]=ON ;
9: LBL[1] ;
10: DO[104:ON :Piros lámpa]=OFF ;
11: DO[103:OFF:Zöld lámpa]=ON ;
12: J PR[11:MacTarolo] 100% CNT100 Offset,PR[20:Folotte] ;
13: L PR[11:MacTarolo] 800mm/sec FINE ;
14: RO[7:ON :Megfogó On Open]=OFF ;
15: J PR[11:MacTarolo] 100% CNT100 Offset,PR[20:Folotte] ;
16: J PR[12:MacAtrako1] 100% CNT100 Offset,PR[20:Folotte] ;
17: J PR[12:MacAtrako1] 20% FINE ;
18: RO[7:ON :Megfogó On Open]=ON ;
19: J PR[12:MacAtrako1] 100% CNT100 Offset,PR[20:Folotte] ;
20: J PR[13:MacAtrakoFord] 100% CNT100 Offset,PR[20:Folotte] ;
21: L PR[13:MacAtrakoFord] 800mm/sec FINE ;
22: RO[7:ON :Megfogó On Open]=OFF ;
23: J PR[13:MacAtrakoFord] 100% CNT100 Offset,PR[20:Folotte] ;
24: J PR[14:Atrko2] 100% CNT100 Offset,PR[20:Folotte] ;
25: J PR[14:Atrko2] 20% FINE ;
26: RO[7:ON :Megfogó On Open]=ON ;
27: J PR[14:Atrko2] 100% FINE Offset,PR[20:Folotte] ;
28: J PR[15:Atrako2Ford] 100% CNT100 Offset,PR[20:Folotte] ;
29: L PR[15:Atrako2Ford] 800mm/sec FINE ;
30: RO[7:ON :Megfogó On Open]=OFF ;
31: WAIT .50(sec) ;
32: J PR[15:Atrako2Ford] 100% FINE Offset,PR[20:Folotte] ;
33: J PR[16:Szinvizsgal] 100% FINE ;
34: IF (DI[103:OFF:Fehér érzékelő]) THEN ;
35: J PR[10:MaczikHOME] 100% CNT100 ;
36: J PR[17:Fehér] 100% FINE ;
37: RO[7:ON :Megfogó On Open]=ON ;
38: J PR[10:MaczikHOME] 100% CNT100 ;
39: ELSE ;
40: J PR[10:MaczikHOME] 100% CNT100 ;
41: J PR[18:Fekete] 100% FINE ;
42: RO[7:ON :Megfogó On Open]=ON ;
43: ENDIF ;
44: J PR[10:MaczikHOME] 100% CNT100 ;
45: IF DI[101:OFF:Mdb.érzékelő]=ON,JMP LBL[1] ;
46: DO[103:OFF:Zöld lámpa]=OFF ;
47: DO[104:ON :Piros lámpa]=ON ;
/END
```

Fém-nem fém szerinti válogatás feladat

Készítse el az alábbi mozgás programját!

Kezdeti beállítások:

- A szerszámkeretet állítsuk be az 1. sz. szerszámmra (Z=115)!
- A koordináta-rendszert állítsuk be World típusra!
- A sebesség felülírást állítsuk be 30%-ra!
- A robot nyissa ki a megfogót!
- A piros visszajelző lámpa világítson, a zöld nem!
- A robot vegye fel a HOME pozíciót (J, CNT100)!

Mozgás:

- A robot várakozzon addig, amíg teszünk a tárolóba munkadarabot, és amíg lenyomjuk a start gombot!
- Ha elindul a robot, a piros lámpa kialszik, a zöld lámpa bekapcsol!
- Menjen a robot a tároló fölé 60 mm-el (J, CNT100)
- Lassan közelítsen a tárolóra (L, FINE)
- Fogja meg a munkadarabot!
- Érzékelő segítségével döntse el a robot, hogy fém, vagy nem fém alkatrész van a tárolóban!
- Emelje ki a munkadarabot a tároló fölé 60 mm-el (J, CNT100)
- Ha fém, az alaphelyzeten keresztül vigye a fém tároló fölé, ott engedje el, majd ismét menjen az alaphelyzetbe!
- Ha nem fém, az alaphelyzeten keresztül vigye a nemfém tároló fölé, ott engedje el, majd ismét menjen az alaphelyzetbe!
- Ha még van munkadarab a tárolóban, vegye ki a következőt, ha nincs, kapcsolja ki a zöld lámpát, kapcsolja be a piros lámpát, és ugorjon a program elejére!

Tesztelje, finomítsa a programját!

Egy lehetséges megoldás TP programja:

```
1: UTOOL_NUM=1 ;
2: UFRAME_NUM=0 ;
3: OVERRIDE=30% ;
4: RO[7:ON :Megfogó On Open]=ON ;
5: DO[104:ON :Piros lámpa]=ON ;
6: DO[103:OFF:Zöld lámpa]=OFF ;
7: J PR[10:MaczikHOME] 100% CNT100 ;
8: WAIT DI[101:OFF:Mdb.érzékelő]=ON AND DI[106:OFF:Start gomb]=ON ;
9: LBL[1] ;
10: DO[104:ON :Piros lámpa]=OFF ;
11: DO[103:OFF:Zöld lámpa]=ON ;
12: J PR[11:MacTarolo] 100% CNT100 Offset,PR[20:Folotte] ;
13: L PR[11:MacTarolo] 800mm/sec FINE ;
14: RO[7:ON :Megfogó On Open]=OFF ;
15: IF (DI[102:OFF:Fém érzékelő]) THEN ;
16: J PR[11:MacTarolo] 100% CNT100 Offset,PR[20:Folotte] ;
17: J PR[10:MaczikHOME] 100% CNT100 ;
18: J PR[19:Fem] 100% FINE ;
19: RO[7:ON :Megfogó On Open]=ON ;
20: J PR[10:MaczikHOME] 100% CNT100 ;
21: ELSE ;
22: J PR[11:MacTarolo] 100% CNT100 Offset,PR[20:Folotte] ;
23: J PR[10:MaczikHOME] 100% CNT100 ;
24: J PR[17:Feher] 100% FINE ;
25: RO[7:ON :Megfogó On Open]=ON ;
26: ENDIF ;
27: J PR[10:MaczikHOME] 100% CNT100 ;
28: IF DI[101:OFF:Mdb.érzékelő]=ON,JMP LBL[1] ;
29: DO[103:OFF:Zöld lámpa]=OFF ;
30: DO[104:ON :Piros lámpa]=ON ;
/END
```

Palettázási feladat FOR / ENDFOR segítségével

Feladat: A 3x3-as tárolóból egyenként szedje ki a robot a munkadarabokat, tegye bele egy jelképes tokmányba, várakozzon 3 másodpercet (jelképes megmunkálási idő), majd vegye ki a tokmányból és tegye vissza a tárolóba ugyanoda, ahonnan kivette!

A használt regiszterek elnevezése, ellenőrzése

Lépünk be a regiszterek közé a **DATA** gombbal, ha kell, a **[TYPE]** gombbal váltsunk a regiszterek oldalára, és a későbbi könnyebb érthetőség kedvéért nevezzük el **R[21]** és **R[22]**

regisztereket, továbbá ellenőrizzük, hogy 0 az értékük!

```
R[ 21:X_valtozo ]=0
R[ 22:Y_valtozo ]=0
```

Új pozícióregiszterek felvétele

Ezt követően szintén a **[TYPE]** gombbal váltsunk a pozícióregiszterekhez! majd vegyük fel a még eddig fel nem vett hiányzó pozícióregisztereket!

A **PR[10:MaczikHOME]** és a **PR[20:Folotte]** már az előző programjainkból meglévőek, maradjanak!

Az újonnan felveendő pozícióregiszterek: **PR[21:Talcaeltolas]** **PR[22:Talca]**

PR[23:Tokmany]

A **PR[21]** **Tálcaeltolás** minden értékét kézi beírással nullára állítjuk.

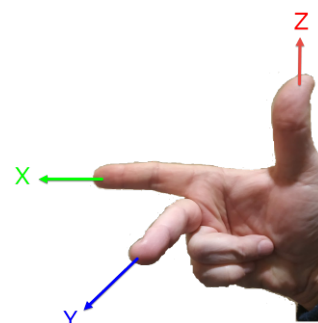
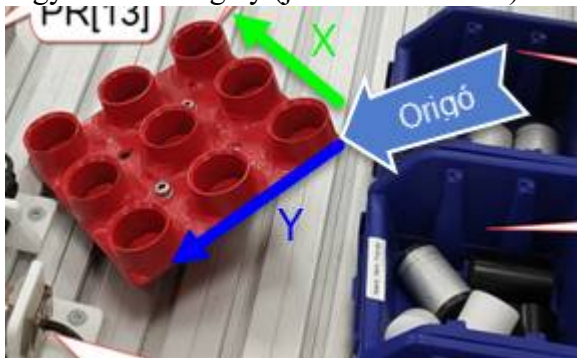
PR[6]	UF:F	UT:F		CONF:NUT	00-1
X	0.000	mm	W	0.000	deg
Y	0.000	mm	P	0.000	deg
Z	0.000	mm	R	0.000	deg

A **PR[22]** **Tálca** a 3x3 tároló robothoz legközelebb eső csúcsánál, behelyezett munkadarab helyzetében legyen! Érintőfogással vegyük fel!

A **PR[23]** **Tokmány** a jelképes tokmányba behelyezett munkadarab helyzetében legyen! Érintőfogással vegyük fel!

UFRAME 4 létrehozása

A robotot állítsuk **WORLD** koordináta-rendszerbe! Mivel a tároló ferde, ezért a hárompontos módszerrel hozzunk létre egy **UFRAME 4**-et, olyan módon, hogy az origó a tároló felső csúcsa legyen, a robottól nézve balra lefelé legyen az **Y** tengely, rá merőlegesen jobbra lefelé nézve legyen az **X** tengely (jobb kéz rendszer)!



Beállítás után váltsunk **Uframe 4** koordináta-rendszerre, és ellenőrizzük is le, hogy a robot a most létrehozott koordinátarendszer szerint mozog-e!

Ha **X**-et és **Y**-t véletlenül felcseréljük, a **PR[20]** értékénél a **Z=-60**, vagyis előjelet kell váltanunk!

TALCATOTOKMANY

Hozzuk létre a **TALCATOTOKMANY** nevű programot, amiben a robot felszedi a munkadarabokat a palettáról és behelyezi a tokmányba!

```
1: !Felszedés a Tálcáról
2: UFRAME_NUM=4
3:J PR[22:Talca] 100% CNT100 Tool_Offset,PR[20:Folotte]
   : Offset,PR[21:Talcaeltolas]
4:L PR[22:Talca] 100mm/sec FINE Offset,PR[21:Talcaeltolas]
5: RO[7:ON]:Megfogó On Open]=OFF
6: WAIT .20(sec)
7:J PR[22:Talca] 100% CNT100 Tool_Offset,PR[20:Folotte]
   : Offset,PR[21:Talcaeltolas]
8: UFRAME_NUM=0
9:J @PR[10:MaczikHOME] 100% CNT100
10: !Berakás a tokmányba
11:J PR[23:Tokmany] 100% CNT100 Tool_Offset,PR[20:Folotte]
12:L PR[23:Tokmany] 100mm/sec FINE
13: RO[7:ON]:Megfogó On Open]=ON
14: WAIT .20(sec)
15:J PR[23:Tokmany] 100% CNT100 Tool_Offset,PR[20:Folotte]
[End]
```

A felkiáltó jellel kezdődő, sárga háttérű sorok csak megjegyzések, segítik a megértést, el is hagyhatóak.

A második sorban mivel a tálca a ferde síkon van, **UFRAME_NUM=4** beállítása szükséges!

A harmadik sorban a **PR[22]** tálca pozíciót közelítem meg. A **Tool_Offset** kiegészítés arra való, hogy a felszedési pozíció fölött 60 mm-ig gyorsjáráttal, CNT-vel menjen a robot. Az **Offset** arra való, hogy a robot maga számolja a ciklusok alapján a helyzeteket.

A negyedik sor majdnem ugyanaz, mint a harmadik, de lassabb a mozgás, FINE típusú, és nincs **Tool_Offset** kiegészítés, amittől a robot lassan ráközelít a fogásra. Célszerű a harmadik sort másolni, majd átalakítani.

Az ötödik sorban a megfogó megfogja a munkadarabot, a hatodik sorban vár 0,2s-ot a biztos fogás miatt.

A hetedik sor teljesen megegyezik a harmadik sorral, másoljuk! Hatására újra a tároló fölé 60 mm-el kerül a robot, de már fogja a munkadarabot.

A nyolcadik sorban álljunk vissza WORLD koordináta rendszerre, majd a kilencedik sorban a biztonság kedvéért küldjük a robotot a HOME pozícióba!

A program második felét célszerű szintén másolással létrehozni, majd amit kell, kicserélni: a megjegyzés szövegét, A **PR[22]**-t mindenhol **PR[23]**-ra, az **RO[7]**-et **ON**-ra. Mivel a tokmány mindig ugyanott van, az **Offset**-re nincs szükség, töröljük!

TOKMANYTOTALCA

Hozzuk létre a **TOKMANYTOTALCA** nevű programot, amiben a robot felszedi a munkadarabokat a tokmányból és behelyezi a paletta megfelelő részébe!

```
1: !Kiszedés tokmányból
2: UFRAME_NUM=0
3: WAIT 3.00(sec)
4:L PR[23:Tokmany] 100mm/sec FINE
5: RO[7:DN:Megfogó On Open]=OFF
6: WAIT .20(sec)
7:J PR[23:Tokmany] 100% CNT100 Tool_Offset,PR[20:Folotte]
8:J @PR[10:MaczikHOME] 100% CNT100
9: !Lerakás tálcára
10: UFRAME_NUM=4
11:J PR[22:Talca] 100% CNT100 Tool_Offset,PR[20:Folotte]
: Offset,PR[21:Talcaeltolas]
12:L PR[22:Talca] 100mm/sec FINE Offset,PR[21:Talcaeltolas]
13: RO[7:DN:Megfogó On Open]=DN
14: WAIT .20(sec)
15:J PR[22:Talca] 100% CNT100 Tool_Offset,PR[20:Folotte]
: Offset,PR[21:Talcaeltolas]
[End]
```

Mivel a mozgás az előző tükörképe, a legegyszerűbben akkor járunk el, ha a **TALCATOTOKMANY** programot két részletben átmásoljuk, elejére a második részét, végére az első részét, majd átírjuk a megjegyzéseket, kicseréljük **RO[7]** értékeit.

PALETTAZ

Hozzunk létre egy főprogramot, a neve legyen „**PALETTAZ**”! **SELECT > CREATE > PALETTAZ** az alábbi módon!

```
1: UTOOL_NUM=1
2: OVERRIDE=30%
3: UFRAME_NUM=0
4:J @PR[10:MaczikHOME] 100% CNT100
5: RO[7:DN:Megfogó On Open]=DN
6: PR[21,1:Talcaeltolas]=(0)
7: PR[21,2:Talcaeltolas]=(0)
8: !Oszlopciklus
9: FOR R[22:Y_valtozo]=1 TO 3
10: !Sorciklus
11: FOR R[21:X_valtozo]=1 TO 3
12: !Megfogások meghívása
13: CALL TALCATOTOKMANY
14: CALL TOKMANYTOTALCA
15: !X eltolás
16: PR[21,1:Talcaeltolas]=PR[21,1:Talcaeltolas]+51
17: ENDFOR
18: !Reset x eltolás
19: PR[21,1:Talcaeltolas]=(0)
20: !Y eltolás
21: PR[21,2:Talcaeltolas]=PR[21,2:Talcaeltolas]+51
22: ENDFOR
23: UFRAME_NUM=0
24:J @PR[10:MaczikHOME] 100% CNT100
[End]
```


A tálcáeltolás alaphelyzetbe hozásának módja:

Instruction 1/3	REGISTER statement 1/1	Mixed Logic 1/2	Mixed Logic 2/2
1 Registers	1 ...=...	1 DO[]	1 PR[i,j]
2 I/O	2 ...=...+...	2 R[]	2 UO[]
3 IF/SELECT	3 ...=...-...	3 F[]	3 RO[]
4 WAIT	4 ...=...*...	4 GO[]	4 TIMER[]
5 JMP/LBL	5 ...=.../...	5 SO[]	5
6 CALL	6 ...=...DIV...	6 AO[]	6
7 Miscellaneous	7 ...=...MOD...	7 Parameter name	7
8 --next page--	8 ...=(...)	8 --next page--	8 --next page--

Eredmények:
6: PR[21,1:Talcaeltolas]=(0)
7: PR[21,2:Talcaeltolas]=(0)

Az első szám a pozícióregiszter száma, a második szám szabadságfokot jelöli.

Szám	Jelentés
1	X
2	Y
3	Z
4	x(w)
5	y(p)
6	z(r)

A feladatban a **PR[21]** pozícióregiszter X és Y értékét nullázzuk, hozzuk alaphelyzetbe. Ezután létrehozuk az **Y** irányú oszlopciklust egytől háromig, azon belül egy **X** irányú sorciklust szintén egytől háromig, hiszen 3x3-as tárolónk van.

```

8: !Oszlopciklus
9: FOR R[22:Y_valtozo]=1 TO 3
10: !Sorciklus
11: FOR R[21:X_valtozo]=1 TO 3

```

Hozzuk létre két programhívást, az első tálcáról a tokmányba pakolást, míg a második a tokmányból a tálcára pakolást fogja leírni. Ahhoz, hogy ezeket a programokat be tudjam hívni, előzőleg létre kellett hoznom!

```

12: !Megfogások meghívása
13: CALL TALCATOTOKMANY
14: CALL TOKMANYTOTALCA

```

Az **X** irányú eltolást a tárolónak megfelelő 51mm-re állítom, majd **ENDFOR** utasítással zárom a belső ciklust.

Ezt követően nullázni kell a belső eltolást, különben a második sor nem az első pozícióból indulna.

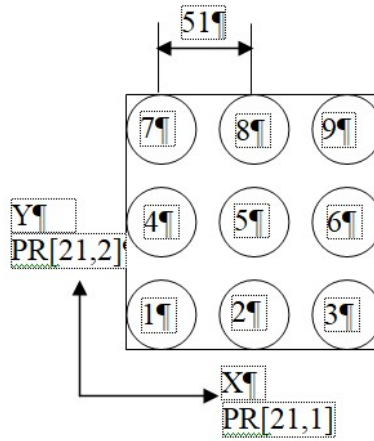
Az **Y** irányú eltolást a tárolónak megfelelően szintén 51mm-re állítom, majd **ENDFOR** utasítással zárom a külső ciklust.

A végén **WORLD** koordináta-rendszerre állok, és **HOME** helyzetbe küldöm a robotot!

Megjegyzések

- Ha az oszlopokat és sorokat fel akarjuk cserélni, akkor a főprogramban kell a belső és a külső ciklus változóit (**PR[21,1]** és **PR[21,2]**) felcserélni, és természetesen **PR[21,2]** kell nullázni.
- Ha a paletta osztásai különbözőek, a programban a növekményeken kell változtatni **X** vagy **Y** irányban, vagy mindkét irányban.

- Az egész programban csak 5 db pozícióregiszter van, ha a programot átesszük a robotra, csak ezeket, továbbá az **UFRAME** koordinátarendszert kell beállítani, és már futhat is a program.
- Ha több tároló rekesz van a palettán, a ciklusok **TO** értékét kell megváltoztatni **X** vagy **Y** irányban, vagy mindkét irányban.



Felhasznált irodalom

- FANUC Robot series R-30+B/R-30+B Mate/R-30+B Plus/R-30+B Mate Plus/ R-30+B Compact Plus/R-30+B Mini Plus CONTROLLER OPERATOR'S MANUAL (Basic Function)