



Gyakorlófeladatok

Tartalomjegyzék

- 1) Bevezető feladatok
- 2) Egyszerű TCP Betanítás
- 3) Egyszerű Robot Programozás
- 4) User Frame Betanítás
- 5) Offsets és Position Register
- 6) Haladó programozás
- 7) Bemenetek / Kimenetek
- 8) Különböző megállási típusok
- 9) DCS Biztonsági Zóna
- 10) Testreszabott iPendant Kijelző
- 11) Macro
- 12) Menu segédprogram



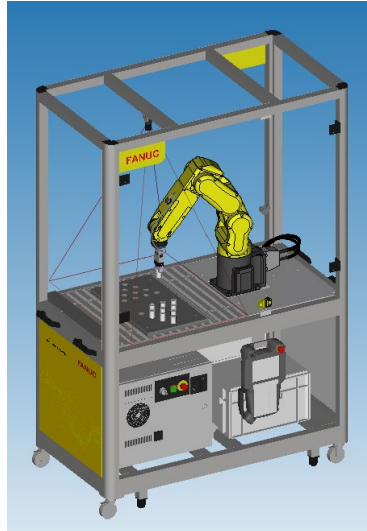
Gyakorlat 1

Bevezetés



Összefoglalás

Ebben a gyakorlatban gyors áttekintést adunk a robotrendszer felépítéséről, a szükséges alapvető ellenőrzésekről és ismeretekről. E gyakorlat után képesnek kell lennie a robot be- és kikapcsolására, a különböző üzemmódok módok közötti váltásra és az alapvető robot mozgások kivitelezésére.



VEGYE FIGYELEMBE:

Minden gyakorlat az alábbi útmutatón alapul:
R-30iB basic operator manual [B-83284EN_05]

A további gyakorlatokban a Kezelői Kézikönyvre hivatkozunk.

Minden gyakorlat az előző gyakorlatok során megszerzett ismereteken alapul. Ez azt jelenti, hogy a folytatás előtt meg kell győződnie arról, hogy a gyakorlatban minden tudás ismert-e. Ha az összegzésben valami nem ismert, vagy nem túl világos, akkor ezt a gyakorlatot kell újra elvégeznie. Minden gyakorlat elején adunk egy listát a szükséges felszerelésekről. Győződjön meg arról, hogy rendelkezik-e minden szükséges felszereléssel, mert különben nem fogja tudni elvégezni a gyakorlat minden részét.

3. lap



Gyakorlat 1

Bevezetés

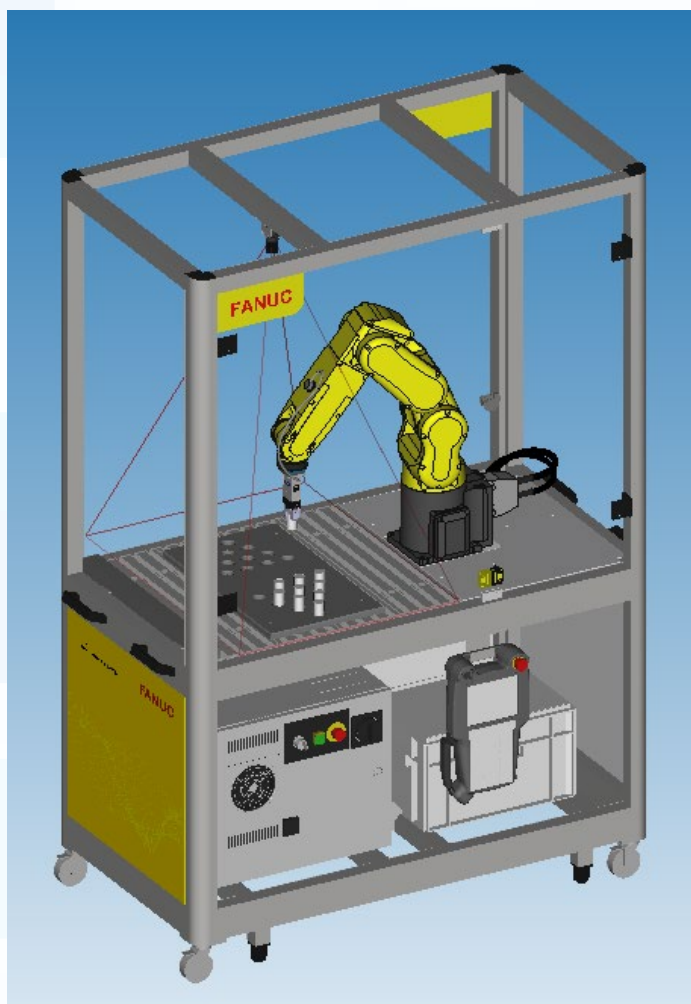
Tartalom:

Összefoglalás	-	3
Felszerelés	-	5
Vezérlőszekrény	-	6
Teach Pendant	-	8
Robot	-	10
Mozgatás	-	11
Program kiválasztás	-	13
Program indítás	-	14
Összegzés	-	15



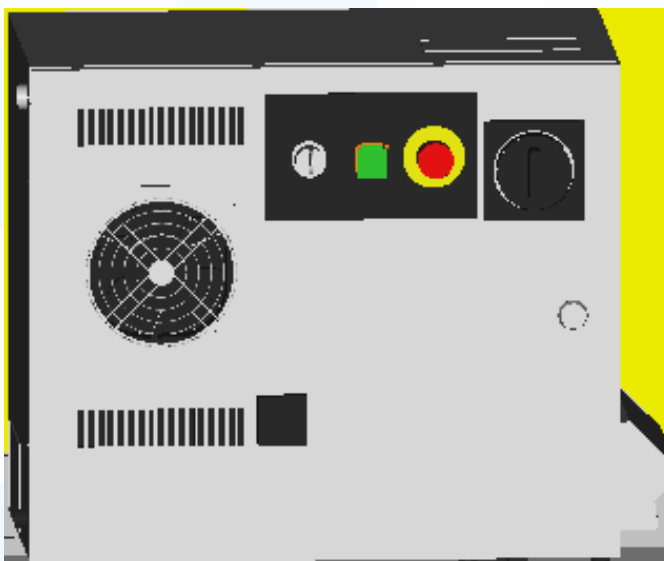
Felszerelés

A szükséges felszerelés:
Oktatócella



Vezérlőszekrény

Az első szükséges eszköz a vezérlőszekrény.



A vezérlőszekrény irányítja a robotot, tartalmazza a hardvert és a szoftvert, és bizonyos értelemben a robot „agya”. A következő oldalon áttekintjük a különböző gombokat és bekapcsoljuk a vezérlőt.



Robot

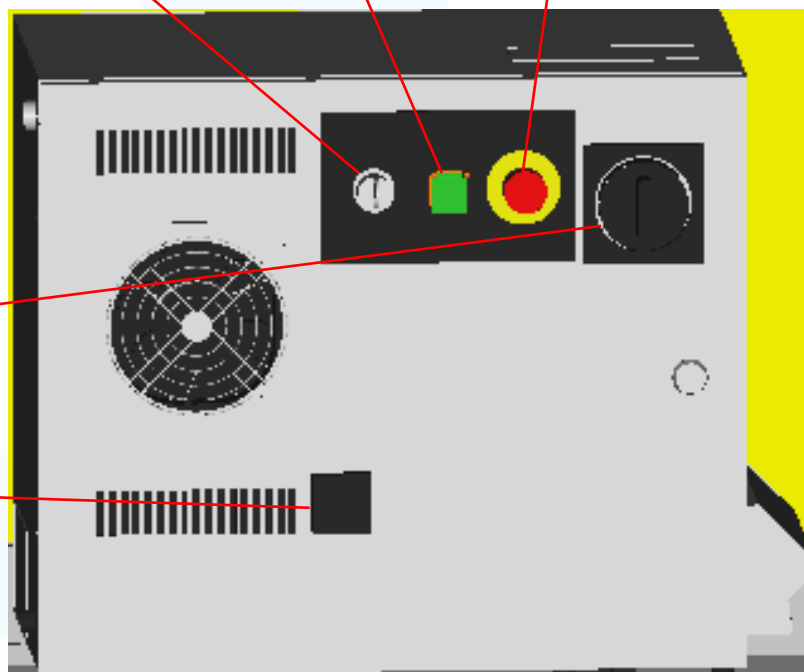
Ciklusstart nyomógomb.
Elindítja a programot.

Vészgomb

Háromállású üzemmódválasztó kapcsoló
(Több részletért olvassa el az Operator Manual 5.2.2 fejezetét.)

Főkapcsoló

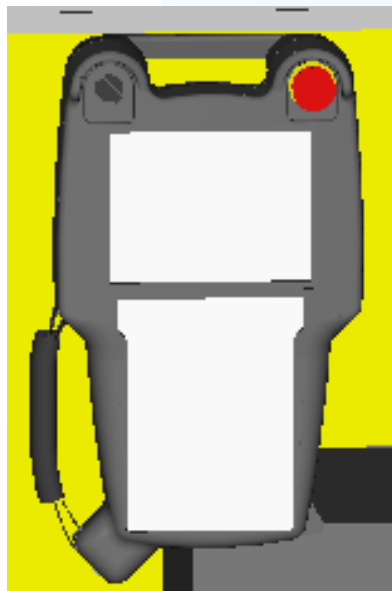
USB csatlakozás



Teach Pendant – Kézi kezelő

A Teach Pendant segítségével a robot működtethető és programozható. Biztosítja a kapcsolatot Ön és a vezérlő között. Minden programozás és tesztelés a Teach Pendant-tal történik.

Több részletért olvassa el az Operator Manual 2.3.1 fejezetét.

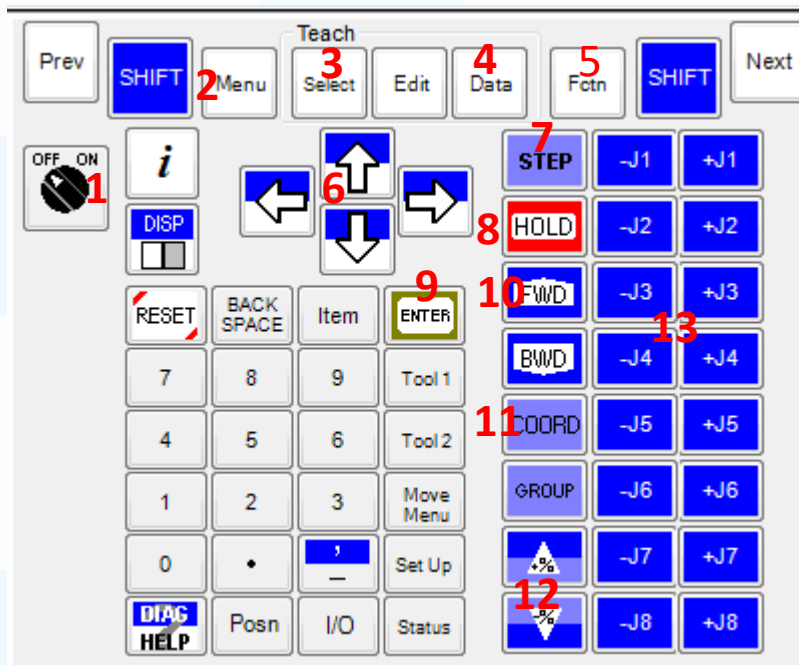


A Teach Pendant még részletesebb használatához a következő lapon talál segítséget.



Teach Pendant – Kézi kezelő

A Teach Pendant legfontosabb funkciói:



1. Teach Pendant engedélyező kapcsoló. Engedélyezi a működést.
2. **Menu** gomb megjeleníti a menürendszert.
3. **Select** gomb belép a programok listájába.
4. **Data** gomb belép a regiszterek (adatok) listájába.
5. **Function** gomb megjeleníti a funkciók listáját.
6. Kurzor gombok mozgatják a kurzort.
7. **Step** gomb engedélyezi/tiltja a lépésenkénti üzemódot. (Operator Manual 6.3.2 fejezet. "Step Test")
8. **Hold** gomb megállítja a program futását.
9. **Enter** gomb jóváhagyja a bevitt adatokat.
10. **Forward** gomb futtassa a programot a következő programsorra.
11. **Coordinate** gomb kiválasztja az aktuális koordináta rendszert.
12. **Override** gomb megváltoztatja a robot mozgási sebességét.
13. **Jog** gombok engedélyezik a robot mozgását kézi üzemmódban.

Robot

A robot mechanikai egysége több önálló karból áll, amelyek hajtóművekkel, vannak összekapcsolva és tengelyeket képeznek. A hajtóművek szervó motorokkal vannak hajtva. Ezek a szervomotorok a robot tengelyeit egymástól függetlenül mozgathatják, a vezérlő utasításainak megfelelően. Az oktatócella robotjának 6 tengelye van.

További részletek az Operator Manual 2.2 fejezetben. "Robot".



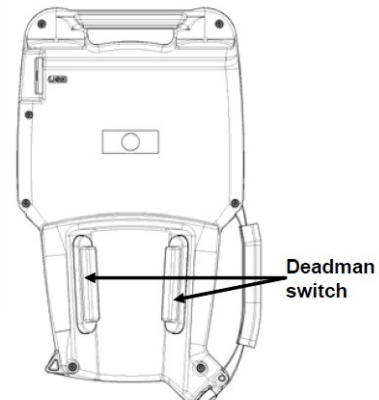
Mozgatás

Kezdeként nézzük a robot alapvető mozgását!

A robot mozgatásához az üzemmódválasztó kapcsolót a vezérlőszekrényen állítsa T1 vagy T2 állásba. (T1 a legbiztonságosabb módja a robottal történő munkának, ezért javasoljuk, hogy mindig használja a T1 üzemmódot).

Ezután szükséges engedélyezni a Teach Pendant-ot az engedélyező kapcsolójával.

A Teach Pendant hátulján a Deadman kapcsolót középállásban szükséges tartani a mozgás idejére. Ennek megnyomását a RESET gomb megnyomásával kell nyugtázni! Végezetül nyomja le és tartsa a SHIFT gombot majd nyomja le a szükséges mozgás irányához tartozó mozgató gombot.



Bővebb információk az Operator Manual 5.2.3 fejezetben. "Moving the robot by Jog Feed"

Bővebb információk az Operator Manual 3.9 fejezetben a mozgási koordinátarendszerekről "Setting Coordinate Systems".

Mozgatáshoz Ön a **COORD** gombbal tud választani a lehetséges koordinátarendszerek között.



Mozgatás

TEST LINE 0 **T2** ABORTED **JOINT**

10
%

JOINT üzemmódban a robot tengelyenként vezérelhető a különböző mozgató gombokkal. (Az egyes tengelyek és az irányok a gombokon feltüntetésre kerültek). Az egyéb más koordináta-rendszerekben a felső három gombpár szolgál az X, Y és Z tengelyek szerinti irányokban mozgatásra, valamint az alsó három gombpár szolgál a fenti tengelyek szerinti forgatás megvalósításához.

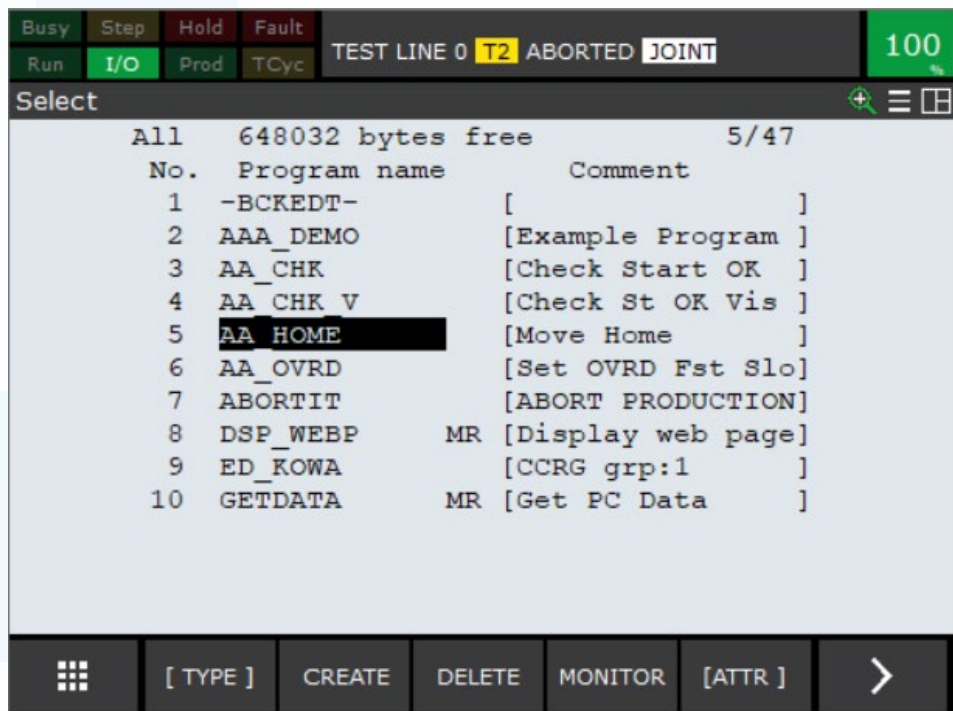
Próbálja mozgatni a robotot a különböző koordináta-rendszerek szerint és irányokba, amíg megszokja a robot mozgatását.

Ügyeljen a különböző koordináta-rendszerek különböző irányultságaira!



Program kiválasztás

Program kiválasztásához nyomja meg a “SELECT” gombot.
Most ki tudja választani a kívánt programot.



Hogy belépjen a kiválasztott programba, használja az **ENTER** gombot. Ezzel beléphet a kiválasztott programba, amely megjelenik a Teach Pendant kijelzőjén.

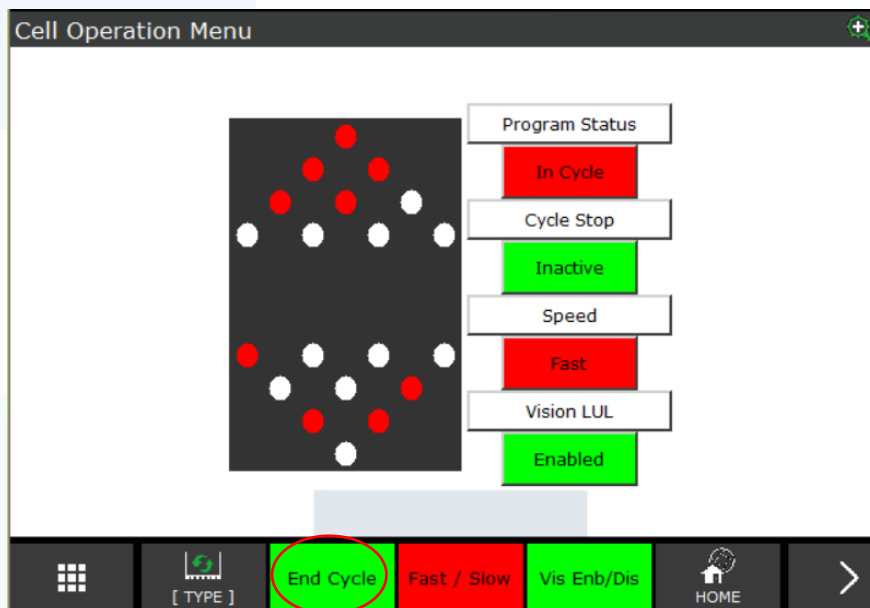


Programindítás

Program elindításakor először győződjön meg arról, hogy semmi sincs a robot útjában. Ezután zárja be a védőkerítést, cellaajtót. Kapcsolja a háromállású üzemmódválasztó kapcsolót AUTO üzemmódba, és kapcsolja ki a Teach Pendant-ot. A RESET megnyomásával szüntesse meg a korábbi hibaállapotot.

A robot most automata üzemmódban van.

Ha megnyomja a CiklusStart gombot, az aktuális program elindul. Válassza ki az AAA_Demo programot (az előző fejezetben bemutatottak szerint) majd nyomja meg a CiklusStart gombot. A program kérdést fog feltenni, hogy minden alkatrész a megfelelő pozícióban van-e. Ellenőrizze, majd nyomja meg az F2 gombot. A program leállításához nyomja meg ismét az F2 gombot.



Olvassa el az Oktatócella kezelési útmutatóját a bővebb információkért.



Összegzés

A gyakorlat végére Ön megszerezte a robottal kapcsolatos alapismereteket és a robot vezérlésével kapcsolatban is az alapvető információkat. Képes a robotot különböző koordináta rendszerek szerint mozgatni, valamint tengelyenként is.

Képes program kiválasztására és futtatására is.





Gyakorlat 2

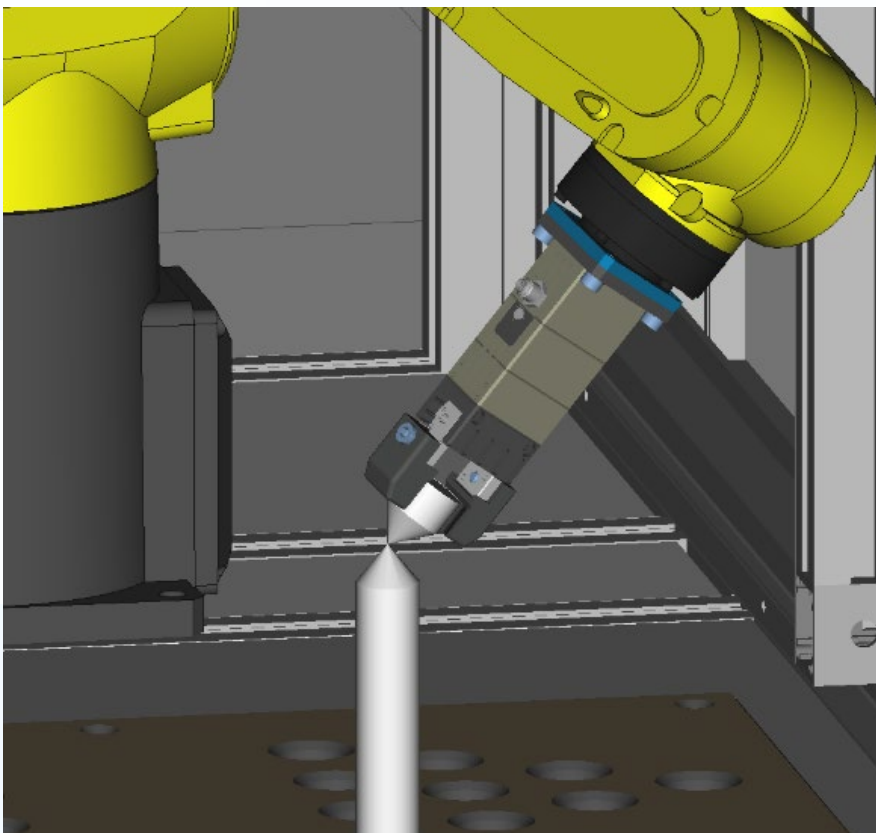
Egyszerű TCP betanítás



Összefoglalás

Ebben a gyakorlatban bemutatjuk a TCP bemérésének koncepcióját és lépéseit.

A TCP a Tool Center Point rövidítést jelenti, és különösen fontos a robot egyszerűbb irányításához a szerszámkoordináta-rendszerben. De nemcsak a mozgásban lehet különbség. Ha a mozgás közben megváltoztatja a szerszám felfogató lap (és a szerszám) helyzetét, akkor a szerszám nem mozog lineárisan, és a sebesség eltér a felfogató laptól. A gyakorlat végén meg kell értenie a TCP fogalmát és ennek a beállításnak a használatát, és képesnek kell lennie a TCP beállítására az összes rendelkezésre álló, de különböző módszerrel.



Gyakorlat 2

Egyszerű TCP betanítás

Tartalom

Összefoglalás	-	3
Háttér	-	5
Eszköz	-	8
TCP Betanítási módok	-	9
Hárompontos módszer	-	10
Hatpontos módszer	-	12
Direkt megadás	-	16
Két pont + Z módszer	-	17
Bővebb információk	-	18
Összegzés	-	19
Függelék	-	20



Háttér

A TCP rövidítése a “Tool Center Point” szavaknak. Ez egy meghatározott pozíció a szerszámfelfogató lap végétől mérve. Ezt általában “Munkapont”-ként definiálják a megfogó közepén, vagy az ívhegesztő pisztoly végén. Fontos, hogy ezt a pozíciót helyesen állítsuk be a tanításhoz, mivel a robot mozgató gombbal lehet mozgatni a robotot a TCP körül, és lineárisan mozgatni is, mivel a robot szoftver a TCP alapján vezérli a robot mozgását.

A helyes TCP beállítás biztosítja, hogy az eszköz koordináta-rendszere a szerszám tengelye mentén legyen. Ez lehetővé teszi az eszköz sokkal intuitívabb mozgását. A rosszul tanított TCP azt jelentheti, hogy működés közben a szerszám csúcs nem azon a helyen van, ahol a kezelő és a robot szeretné, másrészt a szerszám csúcsának forgatása a szerszám csúcsának nem kívánt mozgását eredményezi, miközben az hozzáérhet tárgyakhoz, vagy a robottesthez. (általában a TCP nem mozoghat, miközben egy vagy több tengely körül forog a megfogó vagy szerszám)

Ha a TCP-t helyesen tanítják, a szerszám orientációjának váltása nem eredményezi a sebesség változását vagy a szerszám középpontjának nem kívánt mozgását, és a robot úgy állítja be a mozgását, hogy a szerszám sebessége és mozgása helyes legyen.

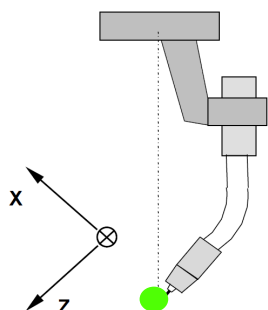
Néhány példáért nézze meg a következő oldalt.

Különböző TCP beállításokból adódó robot mozgásokra példát talál a 7. lapon.

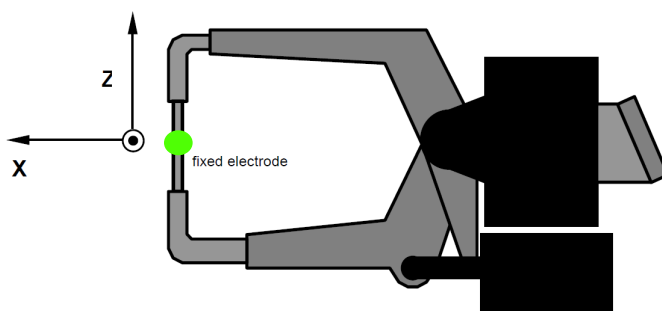


Háttér

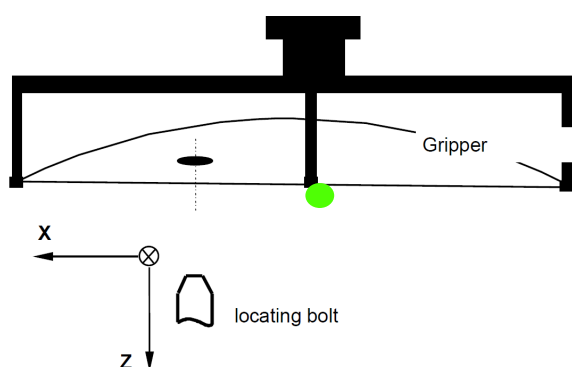
Hegesztő pisztoly



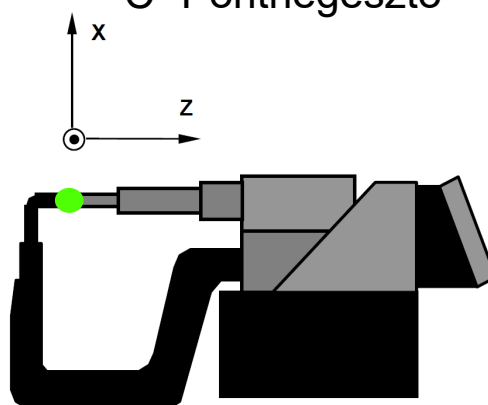
“X” Ponthegesztő



Megfogó

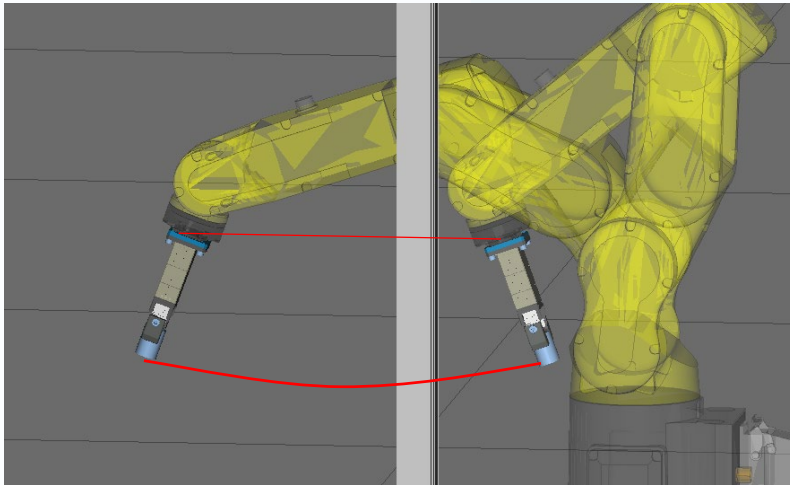


“C” Ponthegesztő

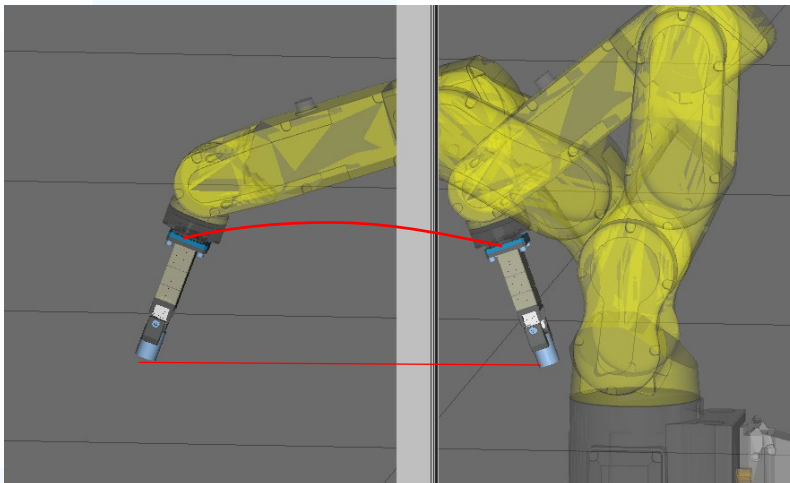


Ezekon a képeken a koordináta-rendszerek a TCP koordinátarendszert, a zöld pont pedig a TCP-t jelölik.

Háttér



Abban az esetben, ha a TCP nincs beállítva, az alapértelmezett TCP a robot szerszámfelfogató lapja, így a robot mozgása közben a szerszám nem fog lineáris mozgást végezni, ezért ezt el kell kerülni.



Amikor a TCP-t az eszköz szerint állítják be, a vezérlő úgy állítja be a robot mozgását, hogy a mozgás a szerszámnak megfelelően történjen.



Felszerelés

Szükséges felszerelések:

Oktatócella

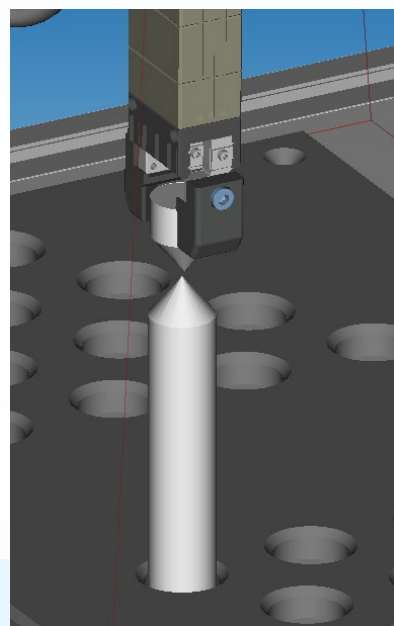
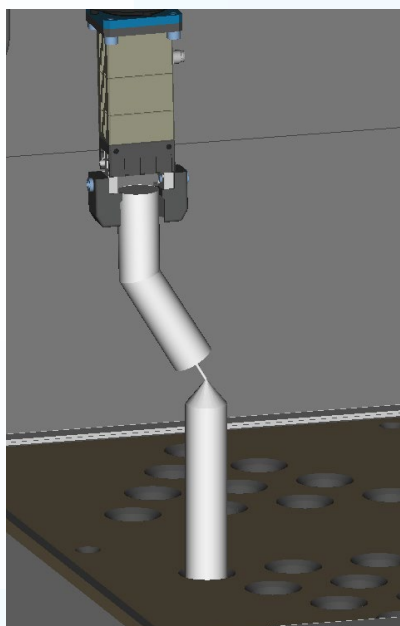
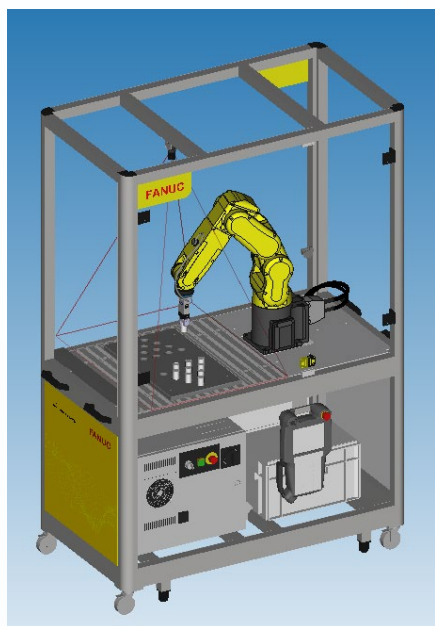
Bemérő tűske – az asztalra rögzítve.

TCP tűske a megfogóban:

Egyszerű TCP tűske a megfogóban

Hajlított TCP tűske a megfogóban (Hegesztő szerszám)

A tűskék leírása a Függelékben található.



TCP betanítási módszerek

A TCP az alábbi módokon tanítható:

- Hárompontos módszer
- Hatpontos módszer
- Közvetlen megadás
- Két pont+ Z módszer

Bővebb részletekért olvassa az Operator Manual 3.9.1 fejezetét. "Setting a Tool Coordinate System"



Hárompontos módszer (TCP Automatikus beállítás)

Ehhez a módszerhez használja az egyszerű TCP tuskét.

Ezzel a módszerrel a robot kiszámolja a TCP pontos helyét az által, hogy három különböző megközelítéssel ugyanahhoz a bemérési ponthoz. Ehhez három bemérési pont felvétele szükséges különböző irányokból és szögekben.

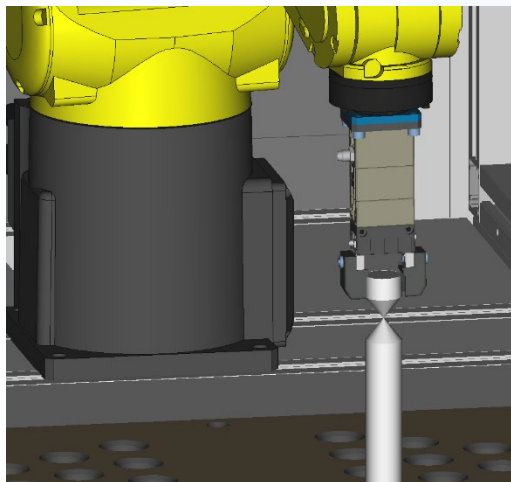
Miután beállította a TCP-t a hárompontos módszerrel, próbálja meg forgatni a TCP-t a szerszám mozgásával, és figyelje meg, hogy látja-e helyben forogni a TCP-t. Sikertelenség esetén alkalmazza ismét a hárompontos módszert, amíg az eredménnyel elégedett lesz.

Bőveb információk az Operator Manual 3-14 fejezetben. "TCP auto set (Three Point Method)"

Követve a példa pozíciókat, saját pozíciókat is alkalmazhat. Ne feledje, hogy szélsőségesebb pozíciók betanításával a TCP számításai pontosabbak lesznek.

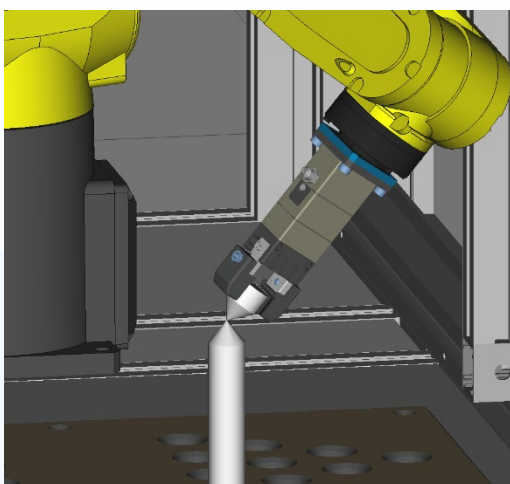
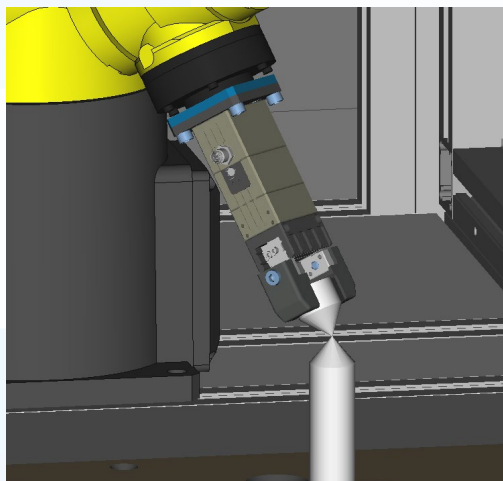


Hárompontos módszer (TCP Automatikus beállítás)



Első betanítási pont felülről.

Második megközelítési pont:
jobbról, döntött szerszámmal



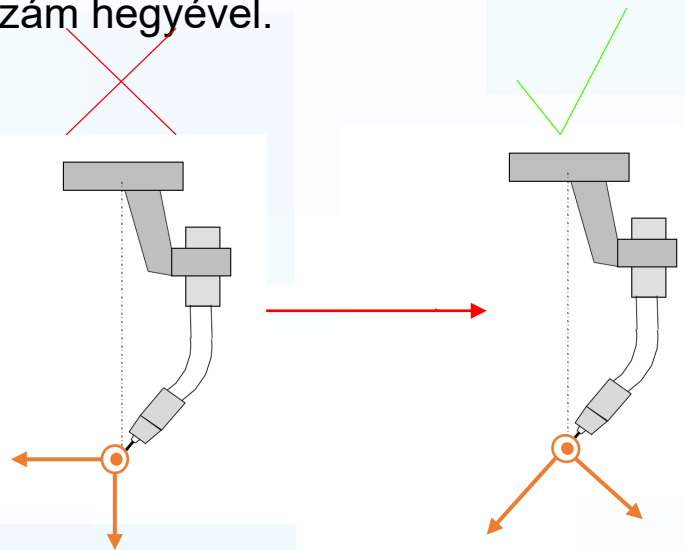
Harmadik megközelítési pont:
balról, a szerszám ellentétes
irányba dől

Hatpontos módszer

Miért szükséges a hatpontos módszer a hárompontos helyett?

Hárompontos módszert használnak olyan szerszámokhoz, amelyek egyenesek és összhangban vannak a szerszám felfogató lappal. Ha nem ez a helyzet, akkor a hárompontos módszer alkalmazásával az eszköz koordináta-rendszere nem megfelelő, mert a tengelyek nincsenek egy vonalban a szerszám hegyével.

A hárompontos módszer alkalmazásával a bal oldali kép TCP-jére kerülünk. A hatpontos módszer alkalmazásakor a TCP-t a megfelelő irányba tudjuk majd állítani.



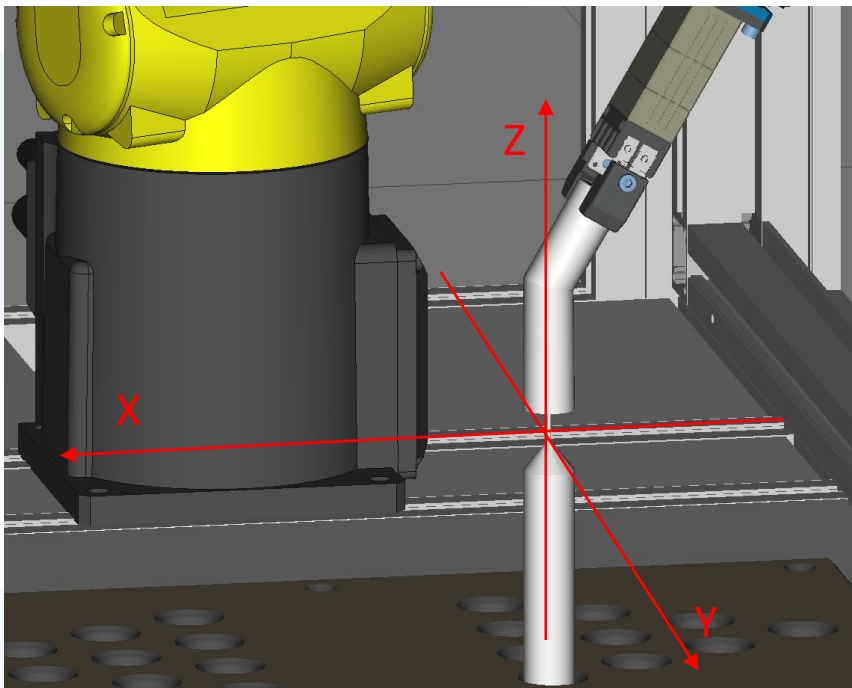
A hatpontos módszerrel képes a koordináta-rendszert úgy megcsavarni, hogy az a szerszám csúcsához igazodjon, így az eszköz sokkal intuitívabb módon mozog (ahogy fentebb látható a jobb oldali képen).

Hatpontos módszer

Ehhez a módszerhez használja a hajlított TCP tuskét.

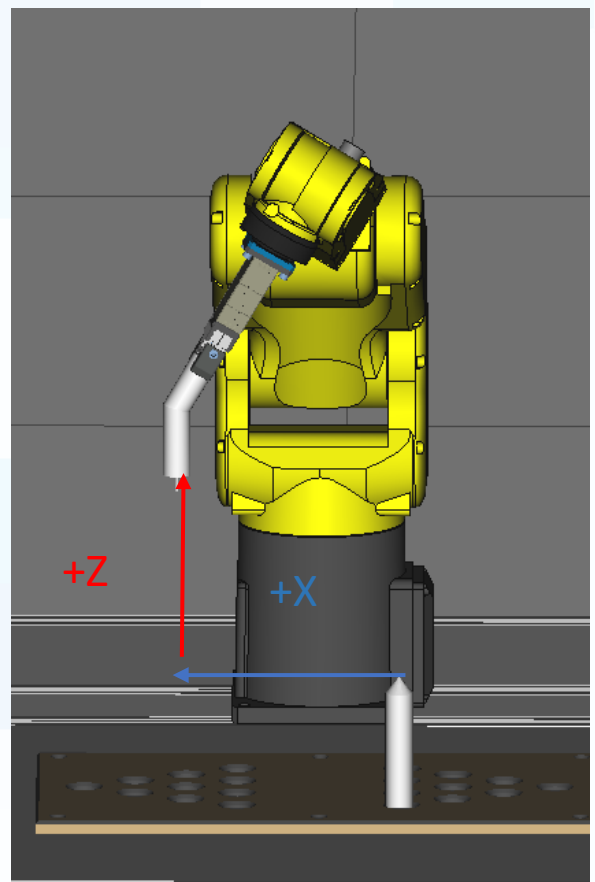
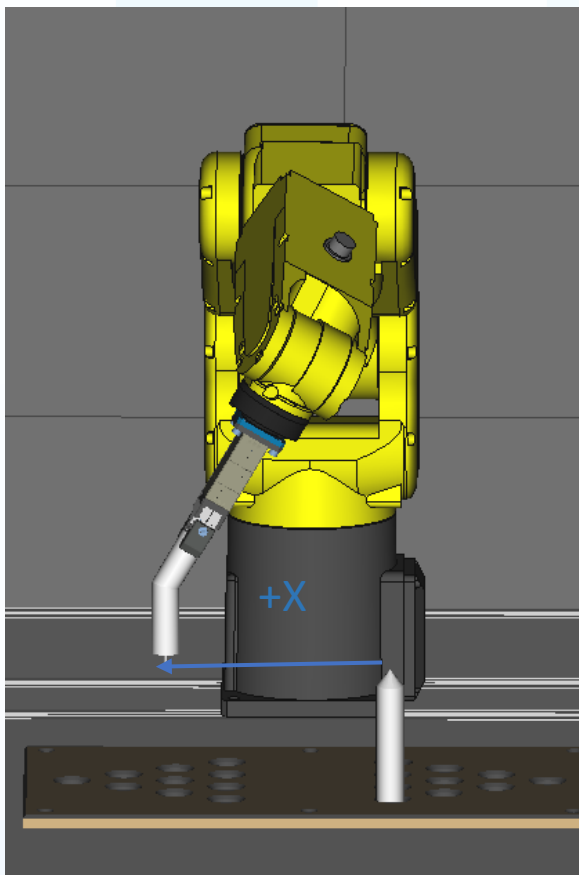
Első lépésben vegye fel a három érintőpontot, ahogy a korábbi (hárompontos) módszernél. (Ezzel meghatározza a TCP pontot)

Ezután tegye a hajlított tuskét függőlegesen a rögzített csap hegyére, és a szerszám hajlításának 90 fokos szögben kell lennie a WORLD koordináta-rendszer X tengelyéhez képest, majd állítsa be az Orient Origin Point pontot. (Végezze el ezt a műveletet a lehető leggondosabban és a lehető legpontosabban, mivel ez utólag kulcsfontosságú a pontos koordináta-rendszerhez)



Hatpontos módszer

A koordinátarendszer beállításához a koordináta-rendszer kívánt irányába kell haladnia. Mivel szerszámot 90 fokos hajlítással orientálta a WORLD koordinátarendszerének X tengelyére, Y irányú mozgással (WORLD-ben) beállíthatja az X tengelyt egy pontra ezen a vonalon (ez az egyenes lesz az X -tengely). Végül mozgassa felfelé a szerszámot (Z tengely a WORLD-ben), és állítsa be a Z tengelyt. Így a koordináta rendszer a szerszámhoz beállításra került.



Hatpontos módszer

A TCP megfelelő beállításához ellenőrizze a következőket:

- Ellenőrizze, hogy forgás közben a hegy nem mozog-e.
- Ellenőrizze, hogy a különböző irányokban (X-, Y- és Z- a szerszám koordináta-rendszerben) mozog-e, hogy a szerszám valóban a várt módon mozog-e.

Végül végezze el ismét a hárompontos módszert a hajlított tuskéval a megfogóban. Figyelje meg a szerszám mozgásának különbségét 3 és 6 pontos módszer között.

A hajlított túske hajlításának nem kell 90 fokos szögben lennie, de erősen javasoljuk, hogy állítsa egy vonalba az egyik tengellyel (X- vagy Y- a WORLD-ben), mert így könnyebb a tengelyek beállítása.

Bővebb információk az Operator Manual 3-15 fejezetben. "Setting up Tool Frame Using the Point (XZ) Method"



Direkt megadás módszere

Ehhez a módszerhez használja az egyenes tuskét a megfogóban.

A Direkt megadás módszernél az összes koordinátát és tájolást manuálisan kell beírnia. Ez a módszer akkor használható, ha ismeri az eszköz pontos méreteit és tájolását. Ez sokkal könnyebb módja az információk gyorsabb beírásának, mint a Hárompontos vagy a Hatpontos módszerrel.

Ez a módszer csak akkor működik, ha tudjuk a megfogó és a munkadarab pontos méreteit.

Gyakorlásként lépjen a TCP beállításához a Hárompontos módszerrel. Módosítsa a koordinátákat a Direkt megadás módszerrel, és figyelje a viselkedésbeli különbségeket, miközben az eszközt mozgatja a Tool koordináta módban. (Például eltolja a szerszám Z-koordinátáit, és figyelje az Y tengely körüli forgást egy új középponttal az éppen beállított pontban)

Bővebb információk az Operator Manual 3-16 fejezetben. "Setting Up Tool Frame Using the Direct List Method"



2 pont + Z módszer

Ehhez a módszerhez használja az egyenes tuskét a megfogóban.

Ez a módszer a Direkt megadás és a Hárompontos módszer kombinációja.

Elsőként adja meg a Z, W, P és R értékeit. (méréssel határozza meg)

Z a J6 előlap és a szerszám hegye között mért távolság.

W, P és R értékei nem fontosak ebben az esetben, mivel az egyszerű egyenes tüske van használatban, ahol a koordináta irányok a szerszámmal esnek egybe.

Más esetben ezeket az értékeket lemérni és megadni szükséges.

Ezután a két bemérési pont betanítása a már korábban megismert módon történhet.

Bővebb információk az Operator Manual 3-17 fejezetben. "Setting Up Tool Frame Using the Two Point + Z"



További információk

Mielőtt megpróbálná használni a TCP-t, mindig győződjön meg arról, hogy a SETIND funkcióval beaktiválta a megfelelő TCP-beállítást, mert különben a régi TCP-beállítás továbbra is aktív.

BusyStepHoldFaultRunI/OProdTCycAAA_DEMO LINE 0T2ABORTEDJOINT100%

SETUP Frames

Tool	Frame	/ Direct Entry		2/10
	X	Y	Z	Comment
1	0.0	0.0	115.0	[EGP40]
2	0.0	38.6	228.5	[
3	0.0	0.0	0.0	[
4	0.0	0.0	0.0	[
5	0.0	0.0	0.0	[
6	0.0	0.0	0.0	[
7	0.0	0.0	0.0	[
8	0.0	0.0	0.0	[
9	15.2	.4	99.7	[egp40_IRV]
10	0.0	0.0	0.0	[

Active TOOL \$MNUTOOLNUM[1] = 1

[TYPE]

DETAIL

[OTHER]

CLEAR

SETIND



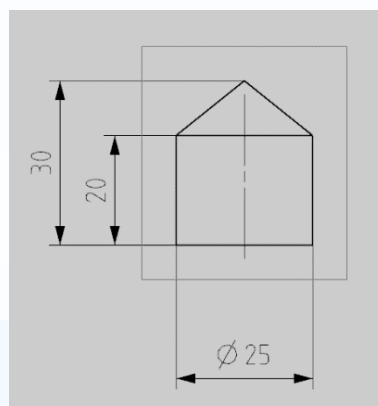
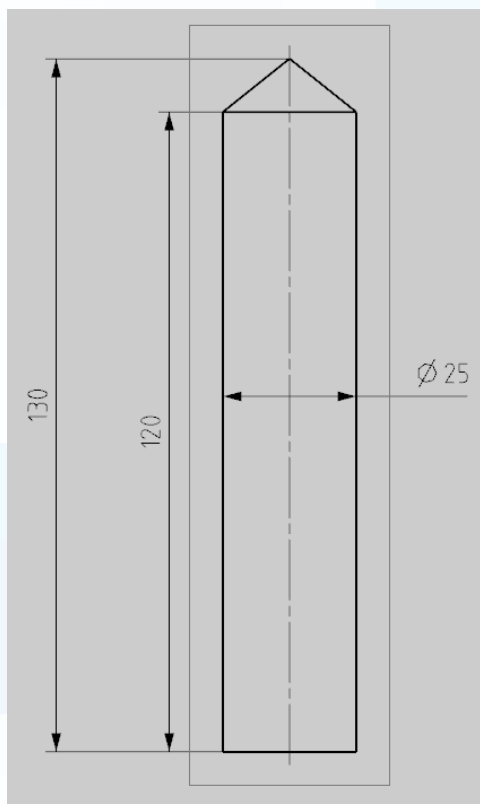
Összegzés

E gyakorlat után most már képesnek kell lennie a TCP fogalmának és használatának megértésére. Képesnek kell lennie a TCP beállítására a különböző típusú eszközök számára a különböző típusú TCP beállítási módszerekkel, és ismernie kell a módszerek közötti különbséget és a különbségek hatásait.

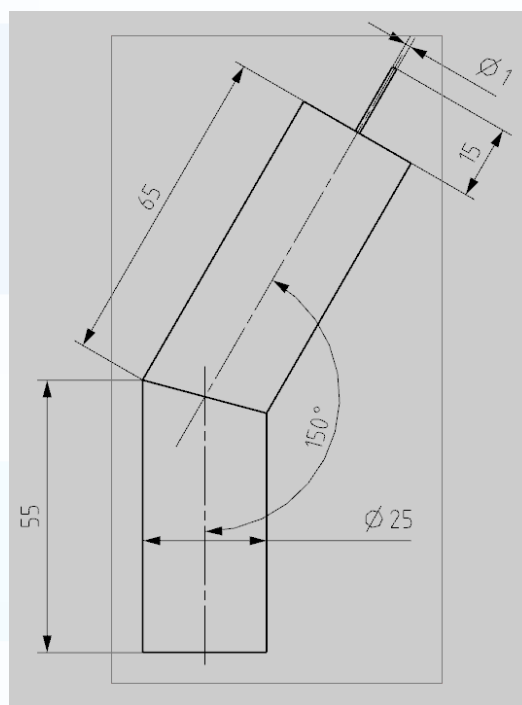


Függelék

Egyszerű TCP túske –
megfogóban történő
használathoz



Bemérő túske – az
asztalhoz legyen
rögzítve



Hajlított TCP túske –
megfogóban történő
használathoz





Gyakorlat 3

Egyszerű Robot Programozás

2. lap



Bevezetés

Ebben a gyakorlatban meg kell tanulnia a robot programozásának alapjait és a robot programozásához rendelkezésre álló különféle lehetőségeket.

E gyakorlat után képesnek kell lennie egy egyszerű mozgás programozására és megjegyzések beillesztésére a programba.



Gyakorlat 3

Egyszerű Robot Programozás**Tartalom**

Bevezetés	-	3
Háttér	-	5
Felszerelés	-	6
Programkészítés	-	7
Pozíciós adatok beillesztése	-	8
Első program futtatása	-	10
Mozgás megváltoztatása	-	13
Megjegyzések	-	14
Figyelmeztetések	-	16
Összefoglalás	-	17



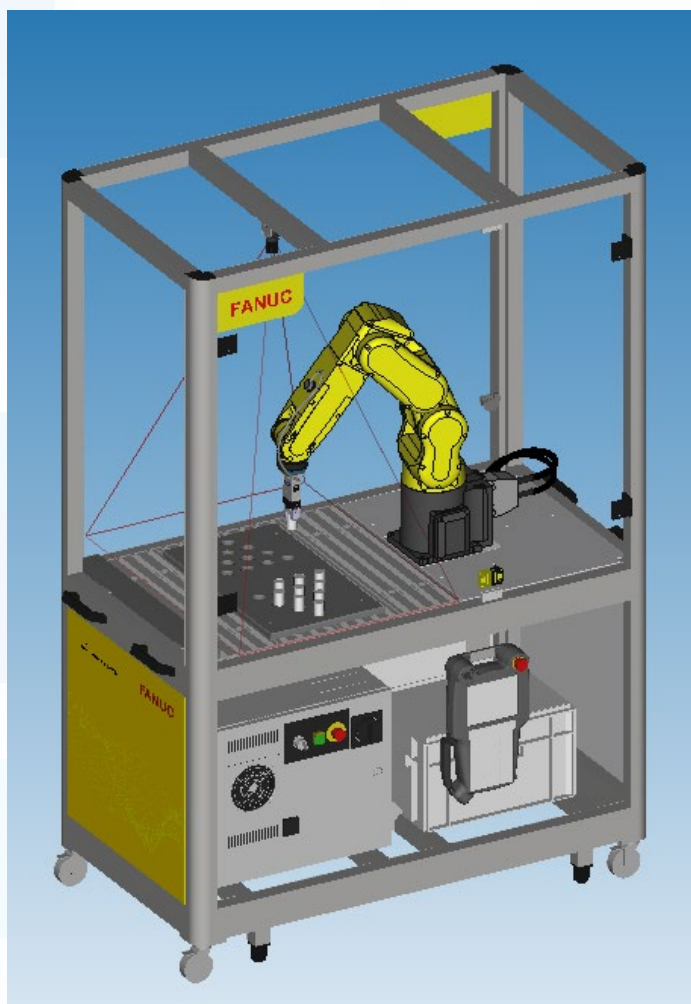
Háttér

Ipari környezetben azt akarjuk, hogy robotunk újra és újra elvégezzen egy bizonyos munkát. Ennek érdekében egyszer beprogramozzuk robotunkat, hogy tudja, mit kell tennie, és ezek után a robot csak ugyanazt a munkát végzi. Ez rendkívül kiszámíthatóvá és ezért rendkívül hatékonyá teszi a munkafolyamatot.



Felszerelés

Ehhez a gyakorlathoz szükséges:
Oktatócella



Új program készítése

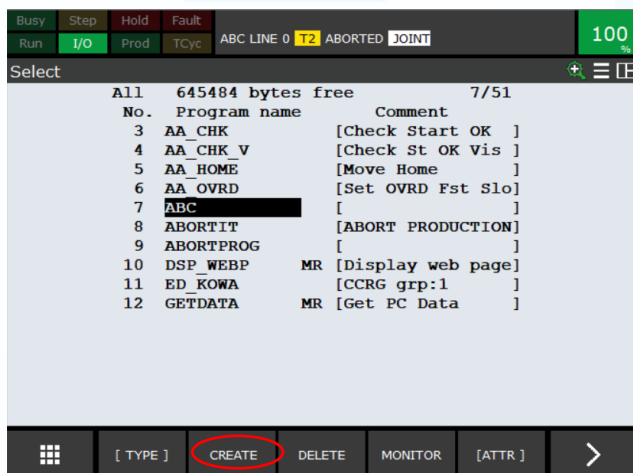
Az első dolog, amit meg kell tennünk, a három üzemmódú kapcsoló T1 állásba állítása.

Másodszor engedélyeznünk kell a Teach Pendant-ot.

Ezután egy új program létrehozásával kezdjük.

Ennek eléréséhez nyomja meg a „SELECT” gombot. Ezután nyomja meg az F2 „CREATE” gombot.

Adjon nevet a programnak, és nyomja meg az „ENTER” billentyűt. Most belépett a programba, és készen áll az első parancsok bevitelére.



Nyomja meg az F2[CREATE] program létrehozásához.

Miután létrehozta az új programot, egy üres program-oldal nyílik meg.

Bővebb információk a programok létrehozásáról az Operator Manual 5-2 fejezetében. “Registering a program”.



Pozíciós adatok beillesztése

Mozgassa a robotot tetszőleges pozícióba.

A robot pozíciójának programban történő rögzítésére két lehetősége van:

Nyomja meg a SHIFT+Point gombokat



Ebben az esetben az alapértelmezett Joint mozgás kerül kiválasztásra.

Nyomja meg a Point gombot.



Ebben az esetben a mozgás típusa kiválasztható egy megjelenő menüből.

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

Bővebb információk a mozgási opcióról az Operator Manual 4.3.1 fejezetben. "Motion Format", 4.3.3 "Feed Rate" and 4.3.4 "Positioning Path".

Bővebb információk a mozgási pontok betanításáról az Operator Manual 5-4 fejezetben. "Teaching a motion instruction".



Pozíciós adatok beillesztése

Mozgassa a robotot egy második pozícióba.

Végezze el a betanítási folyamatot ismét a SHIFT + Point megnyomásával.

Ez máris egy egyszerű és futtatható program.



Első program futtatása

A következő lépés az első program végrehajtása és annak megfigyelése, hogy a program azt csinálja-e, amit Ön szeretne. Ehhez állítsa a robot sebességét 10% -ra az OVERRIDE gombokkal. Ennek célja, hogy megbizonyosodjon arról, hogy a robot nem teljes sebességgel, hanem csak 10% -kal működik, tehát ha valami nem a várt módon történik, akkor van ideje megállítani a robotot, mielőtt komoly kár vagy sérülés történhetne.



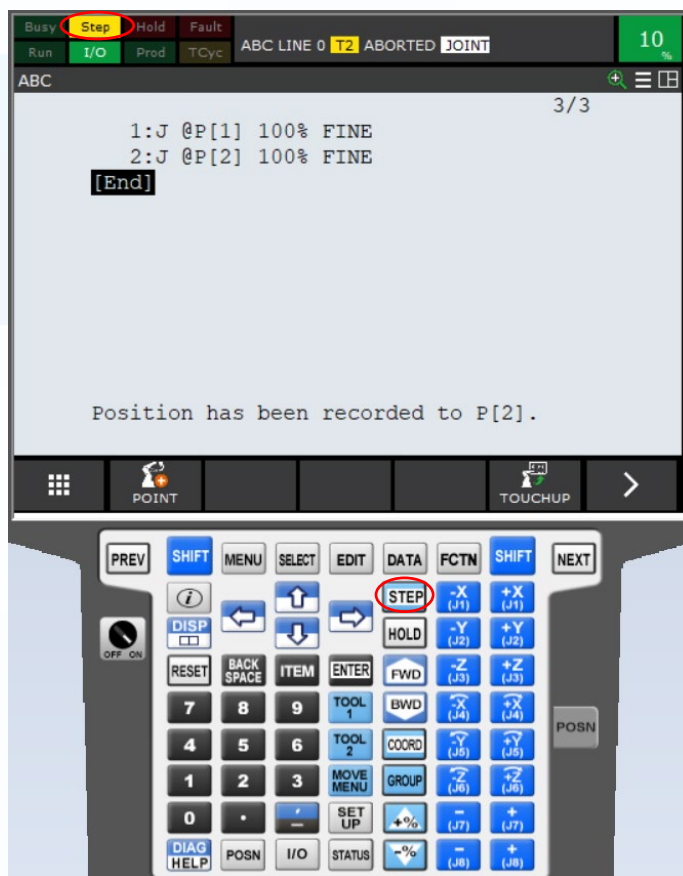
Első program futtatása

Állítsa át a programot lépésenkénti üzemmódba. Ehhez nyomja meg a "STEP" gombot.

Bővebb információk a lépésenkénti üzemmódról az Operator Manual 6.3.3 fejezetben. "Step Test".

Jelen esetben ez nem túl fontos, mivel ennek a programnak csak két lépése van, de a nagyobb programok esetében ez sokkal fontosabbá válik.

Ez a beállítás előírja, hogy a program csak lépésről lépésre hajtható végre, és csak akkor, ha utasítást ad a programnak, hogy folytassa.



Első program futtatása

Most a program végrehajtásához nyomva kell tartani a Deadman kapcsolót és a "SHIFT" gombot és nyomja meg az „FWD” (előre) gombot.

A program most végrehajtja az éppen kijelölt sort.

Miután sikeresen befejezte a programot, fokozatosan növelheti a sebességet, és ha úgy érzi, hogy minden működik, akkor újra megnyomhatja a „Step” gombot, hogy kikapcsolja a lépésenkénti üzemmódot, és a teljes program a végéig végrehajtásra kerül.

FONTOS

A Shift és a Deadman gombokat folyamatosan nyomva kell tartani a program futása alatt!



Mozgás megváltoztatása

Alapértelmezés szerint a mozgás JOINT mozgásként van beállítva. Ennek megváltoztatásához mozgassa a kurzort a módosítani kívánt mozgásra, és nyomja meg a „Choice” gombot.

Ezután válthat a 4 különböző mozgástípus között.

Állítsa a két meglévő mozgás típusát L (lineáris) értékre, és figyelje meg, hogy lát-e különbséget a program végrehajtásában.

A programban módosíthatja a mozgás sebességét is. Ehhez vigye a kurzort a sebesség értékre. Ha megnyomja a „Choice” gombot, megadhatja a sebesség típusát.

Változtassa a sebesség típusát most mm/s-re, a sebességet pedig 100 mm/s-ra.

Futtassa újra a programot, és figyelje meg, lát-e valamilyen különbséget.

```
ABC
1/3
1:L @P[1] 100mm/sec FINE
2:L @P[2] 100mm/sec FINE
[End]
```

Bővebb információk a sebesség értékekről és előtolási mértékegységekről az Operator Manual 4.3.3 fejezetben. “Feed Rate”.

Bővebb információk a mozgási opciókról és sebességekről az Operator Manual 5-3 fejezetben. “Changing a standard motion instruction”.



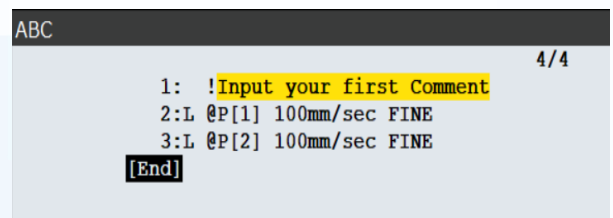
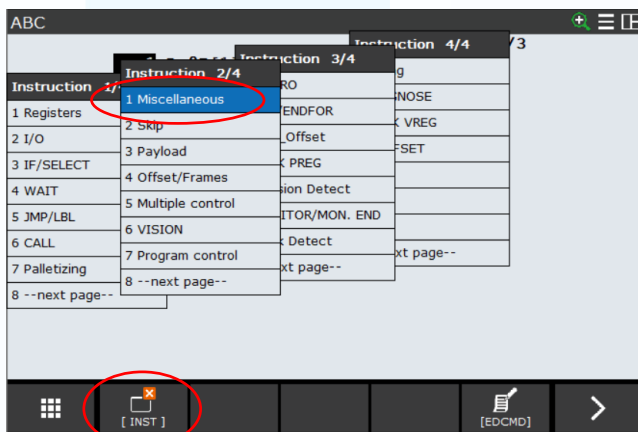
Megjegyzések

Sikeresen létrehozta és végrehajtotta az első programját.

Annak érdekében, hogy más is emberek megértsék a programot, megjegyzés fűzhető a különböző műveletekhez és programrészekhez. Ez kevésbé fontos a jelenlegi rövid program számára, mert csak két soros, de később, akár a 100 sornál is nagyobb programoknál nagyon megkönnyíti a program megértését.

Ehhez először egy új sort kell bevinni a parancssor elé. Vigye a kurzort az első sorra, nyomja meg az NEXT, majd az F5 (EDCMD) gombot, majd válassza az „Insert” menüpontot. Kérdést fog kapni, hogy hány sort kell hozzáadni. Adjon hozzá 1 sort.

Ezután nyomja meg az F1 (INST) gombot, a „MISCELLANEOUS” gombot a második oldalon, majd a „REMARK” gombot. Az „ENTER” megnyomása után megadhat egy megjegyzést, majd utána az „ENTER” újbóli megnyomásával bezárhatja a megjegyzést.



Megjegyzések

Helyezzen el egy második megjegyzést a program két mozgási utasítása közé.

Megjegyzéseket közvetlenül a pozícióba is tehet. Ehhez vigye a kurzort az adott pozíció sorszámára, és nyomja meg az "ENTER" billentyűt. Ezután beírhatja megjegyzését, és újra megnyomhatja az "ENTER" billentyűt.

A programnak hasonlóan kellene kinéznie:

```
ABC 5/5
1: !Input your first Comment
2:L @P[1] 100mm/sec FINE
3: !Input your secon Comment
4:L @P[2] 100mm/sec FINE
[End]
```

Most megtanulta a robot programozásának alapjait.

Képes több mozgási pontot, különböző mozgás típusokkal létrehozni, valamint megjegyzéseket fűzhet a programhoz.



Figyelmeztetések

Mindig mozgassa a robotot, és hajtsa végre a programokat alacsony sebességi beállításokkal. Előfordulhat, hogy rossz pontot programozott, vagy nincs teljesen tisztában a robot mozgásával, ezért az alacsony sebességi beállítás azt jelenti, hogy elegendő idő áll rendelkezésre a robot leállításához, mielőtt bármi rossz történne. Amíg nem teljesen biztos abban, hogy program problémamentesen működik, csak a STEP módban hajtsa végre a programot. Ez megakadályozza, hogy meglepődjön és ne tudjon közbelépni nem kívánt mozgás esetén.

Amennyiben nem tesztelte a programját 100% -os sebességen és a STEP funkcióval, ne működtesse a programot automatikus üzemmódban, mert nem lehet biztos abban, hogy a program a kívánt módon fog működni. Ha a programot automatikus módban futtatja, futtassa a programot többször alacsonyabb sebességen, mielőtt 100% -ra emelné a sebességet.



Összefoglalás

Ebben a gyakorlatban megtanulta a robot programozásának alapjait.

Képesnek kell lennie egy egyszerű mozgásparancs programozására, valamint a mozgás típusának és sebességének megváltoztatására.

Képes sorokat és megjegyzéseket adni a programjához, hogy megkönnyítse a program megértését.

Képesnek kell lennie megjegyzések hozzáadására a pozíciókhoz is.





Gyakorlat 4

User Frame betanítás

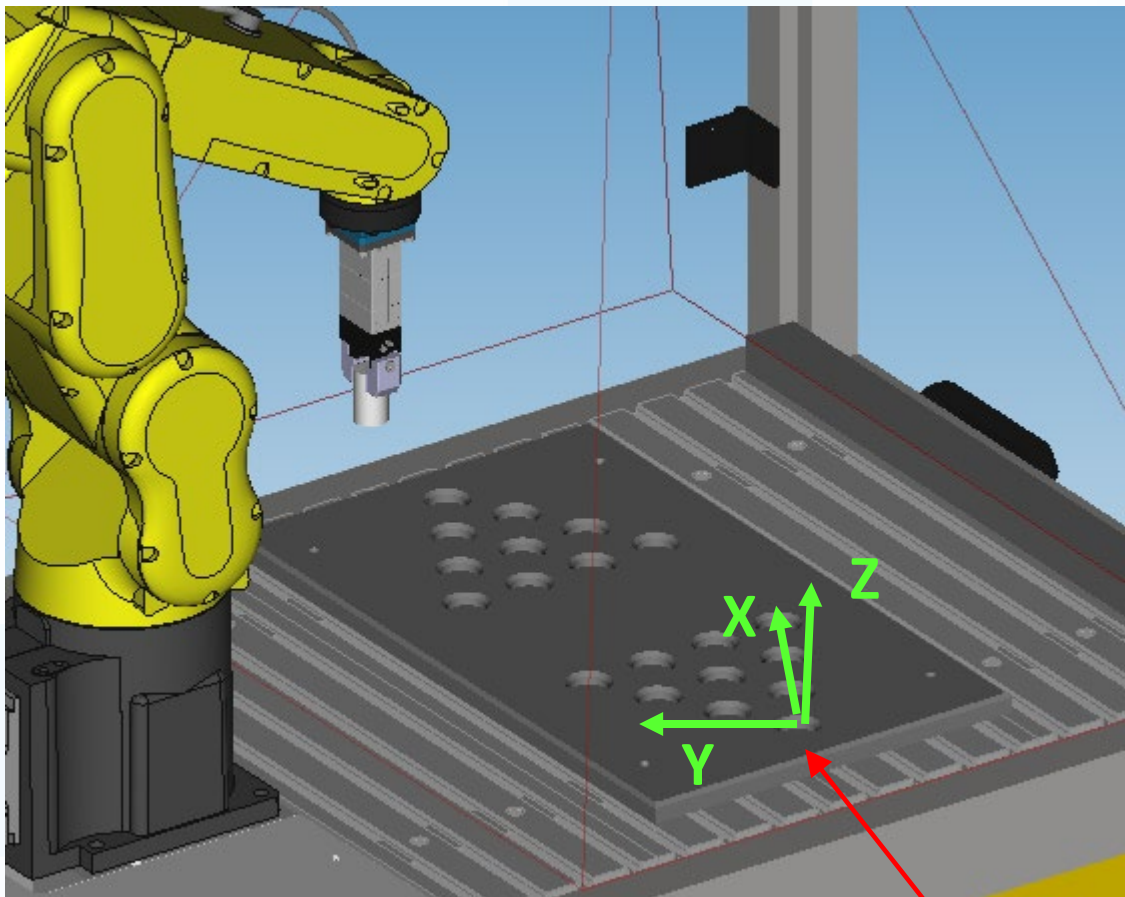


Bevezetés

Ebben a gyakorlatban megtanulja a User Frame tanításának és használatának alapjait. A User Frame (Felhasználói Koordináta-rendszer) egy derékszögű koordináta-rendszer, amelyet a robot betanításának és kezelésének egyszerűbb módjára vehet fel.

Miután elvégezte ezt a gyakorlatot, képesnek kell lennie egy tetszőleges UFrame-et betanítani és programokat készíteni ezzel a UFrame-mel.

Képes lesz különböző UFrame-eken használni, és tudni, hogyan kell programozni, hogy képes legyen gyorsan és sok munka nélkül változtatni több UFrame között.



UFrame

3. lap

Gyakorlat 4

User Frame betanítás

Tartalom

Bevezetés	-	3
Háttér	-	5
Felszerelés	-	6
User Frame meghatározása	-	7
Első program User Frame-el	-	10
2. User Frame meghatározása	-	12
2. User Frame betanítása	-	14
Egy jobb programozási technika	-	15
Figyelmeztetések	-	17
Összefoglalás	-	18
Függelék	-	19

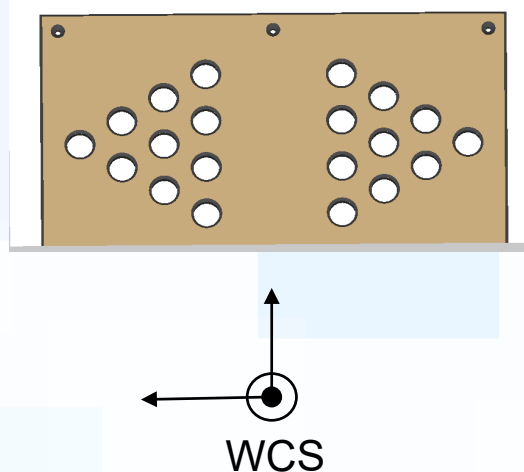


Háttér

A felhasználói koordinátarendszer (User Frame) meghatározza a munkaterület saját koordinátarendszerét, amelyben a robot dolgozik. Esetünkben ez nem olyan fontos, mint más esetekben, mert a pasziánsz tábla ugyanabban a síkban van, mint a WORLD, és merőleges a robotra. Mindazonáltal a robot mozgatása a pasziánsz táblán a WORLD rendszerben nem túl praktikus, mivel a lyukak nincsenek tengelyirányban, ezért az egyik lyukból a másikba való mozgás nem olyan egyszerű WORLD-ben. Ezért definiálunk egy olyan felhasználói koordinátarendszert, amely eltér a WORLD-től, és igazodik a lyukakhoz.

Ez a funkció még fontosabb, ha a robot ferde síkra van felszerelve, vagy ha a munkaterület ugyanabban a síkban van, de 0 vagy 90 foktól eltérő szöget zár be. A User Frame meghatározása ebben az esetben azt jelenti, hogy a robot mozgatása sokkal könnyebbé és intuitívabbá válik, a tanítás és a mozgatás pedig sokkal gyorsabbá válik.

Nem túl praktikus
WORLD-ben mozogni a
táblán.

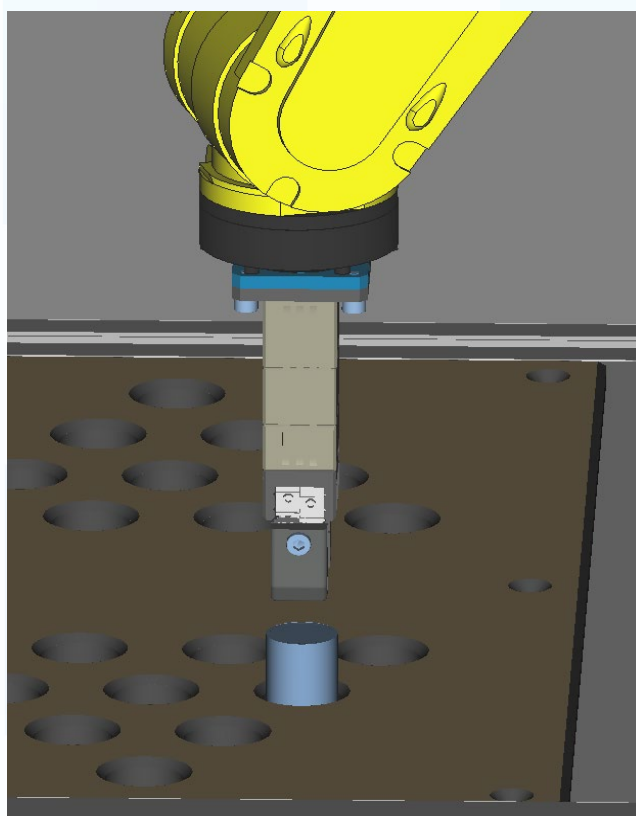
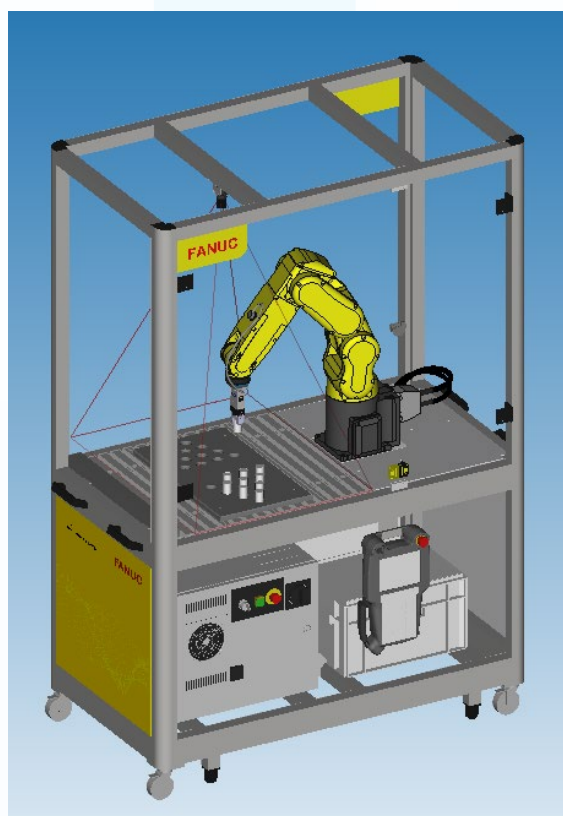


Felszerelés

Ehhez a gyakorlathoz szükséges lesz:

Oktatócella

Egyszerű hengeres munkadarab



User Frame meghatározása

Először egy User Frame-et veszünk fel a pasziánsz bal oldalán (a robotról nézve). A hárompontos módszert alkalmazzuk. Használja a 6.User Frame-et.

Az Origo beállításával kezdjük. Esetünkben az első lyukat vesszük Origo-nak. Ez a pont fontos, ezért gondosan és pontosan végezze el. Miközben a munkadarab benne van a megfogóban, állítsa be a robotot úgy, hogy a munkadarab a lehető legpontosabban legyen a lyukban. Ezután állítsa be a magasságot úgy, hogy egy papírdarab épp csak becsúszhasson a tábla és a munkadarab közé.

FONTOS

A Z-magasság így a tábla felületére van beállítva, mivel nehéz lenne egy furat legmélyebb pontját pontosan meghatározni.

Egy második pontra van szükség az X tengely beállításához. Mozgassa a munkadarabot a tábla robothoz legközelebbi lyukig. Tegye ugyanezt az eljárást, mint az első lyuknál. Végül vigye az alkatrészt a tábla harmadik pontjára. Ennek a pontnak a helye (X-, Y-) nem olyan fontos, de ne feledje, hogy nagyobb távolság esetén nagyobb pontossággal jár, ezért ne helyezze közvetlenül a másik két pont mellé. Ennek a pontnak az a fontos része, hogy a munkadarab azonos magasságban legyen, mint a másik két pont, ezért egy papírdarabnak képesnek kell lennie éppen alá csúszni.

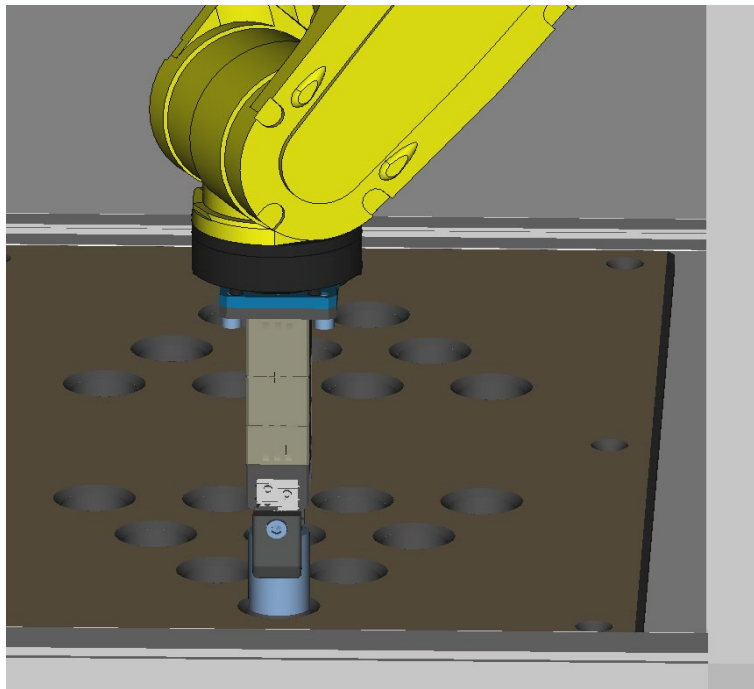
Nézze meg a képeket a következő lapon.

A UFrame beállítása után nyomja meg az F5 SetInd gombot és állítsa aktívra a 6. User Frame-et.

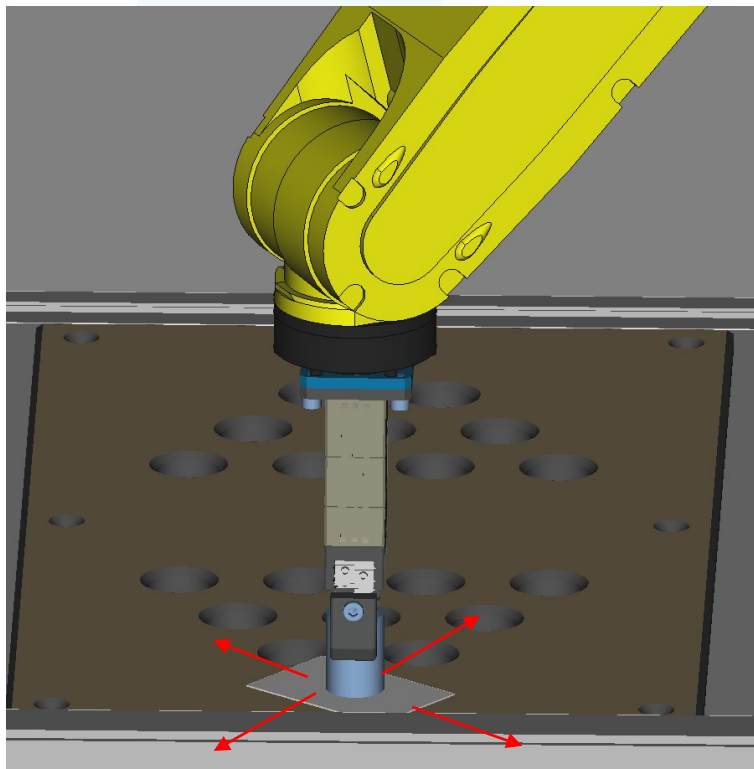
7. lap



User Frame betanítása

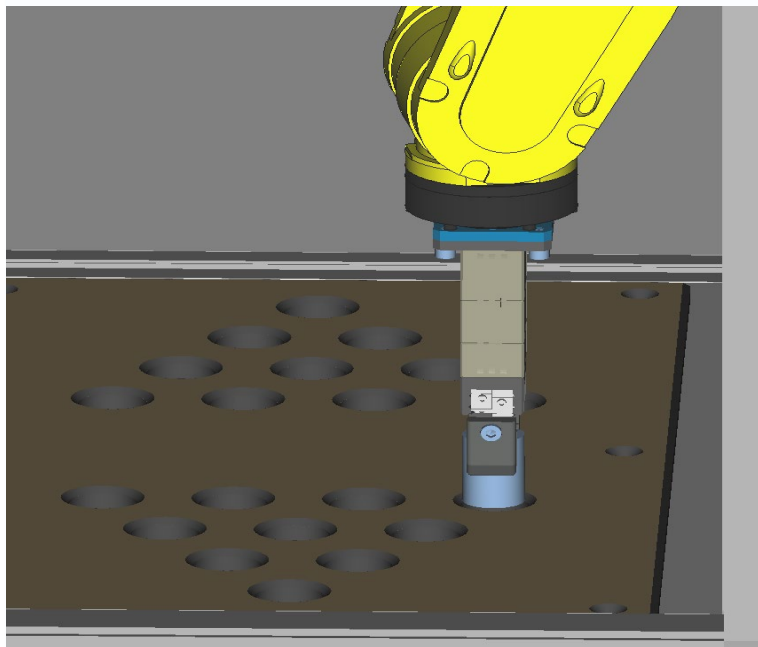


Origo beállítása

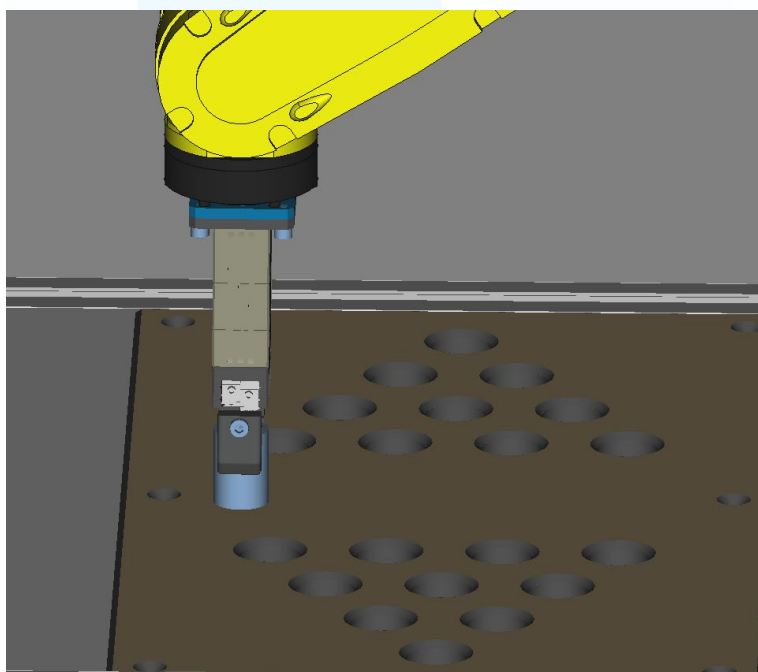


Egy papírlap épp csak aláférjen

User Frame betanítása



X-tengely beállítása



Y-tengely beállítása

Bővebb információk az Operator Manual 3-18 fejezetben. “Setting Up User Frame Using Three Point Method”



Az első program User Frame használatával

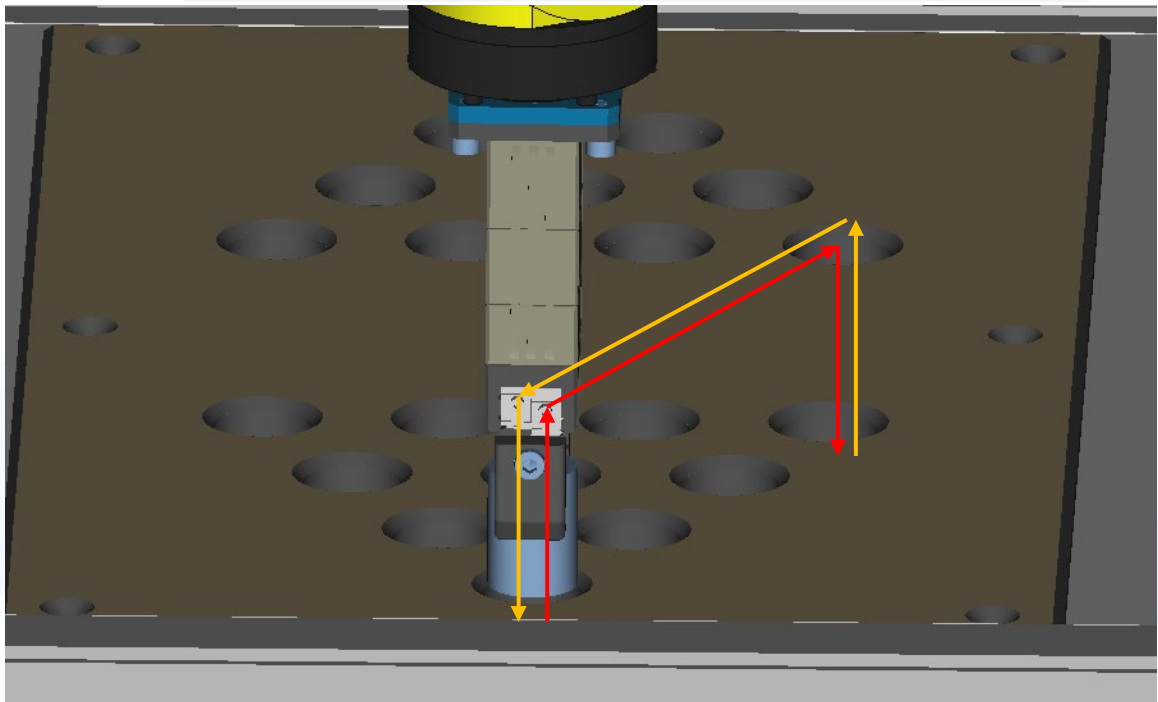
Most készítsen egy egyszerű programot, miközben a korábban meghatározott User Frame-et használja, hogy a munkadarabot az egyik lyukból a másikba vigye.

Ügyeljen arra, hogy a megfogót a munkadarabbal és anélkül is pontosan a furat fölé helyezze, egy kicsit magasabbra (néhány centiméterre), mielőtt a megfogó leereszkedik a furatba.

A Függelékben feltettünk egy példa programot, így megnézheti, hogyan kell elkészíteni a programot, ha nem sikerül a sajátját. Ügyeljen arra, hogy ugyanazt az UFrame-et használja a pontok tanítása közben, mint amikor ezeket a pontokat egy programban használja.

FONTOS

Ha nem sikerül a programot elkészítenie, javasoljuk, hogy ismételje át a 3. gyakorlatot.



10. lap

Az első program User Frame használatával

Legyen biztos benne, hogy a programjában a megfelelő User Frame-et állította be.

Az utasítás programban történő elhelyezéséhez nyomja meg az F1 [INST] gombot, válassza az “Offset/Frames”, azon belül a “UFRAME_NUM” lehetőséget, majd adja meg a használni kívánt UFrame számát. (Esetünkben az a 6-os UFrame)

```
1/18
1: !Example
2: !Defining UFrame
3: UFRAME_NUM=6
4:L @P[1:Above Pick 1] 250mm/sec
   : FINE
5:L @P[2:Pick Pos 1] 250mm/sec FINE
   :
6: !Gripping the pin
7: RO[7:Open Gripper]=OFF
8: !Making sure the gripper engages
9: WAIT .30(sec)
```

[INST] [EDCMD]

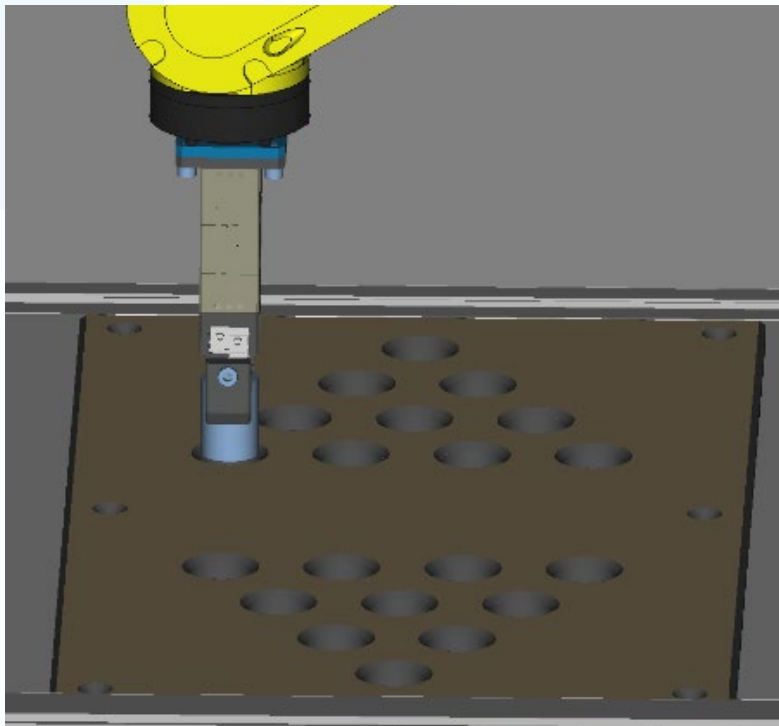


2. User Frame betanítása

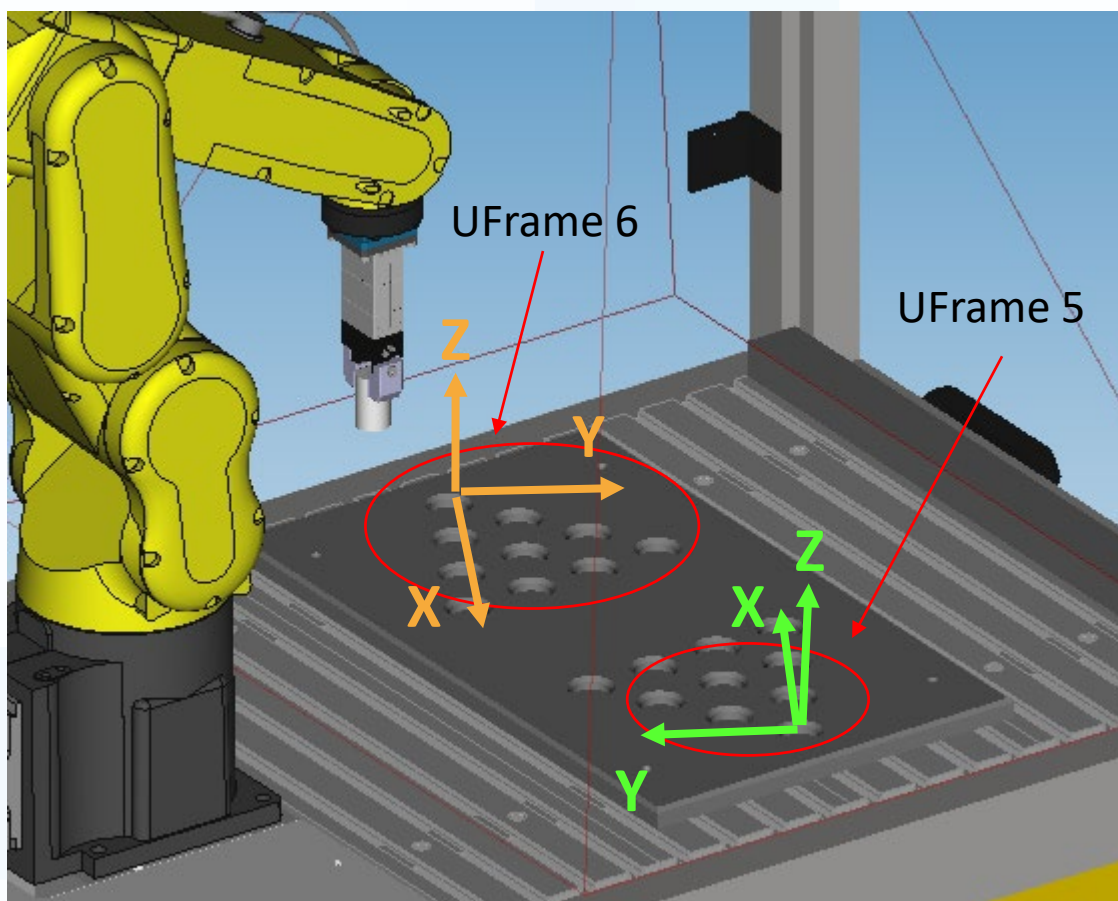
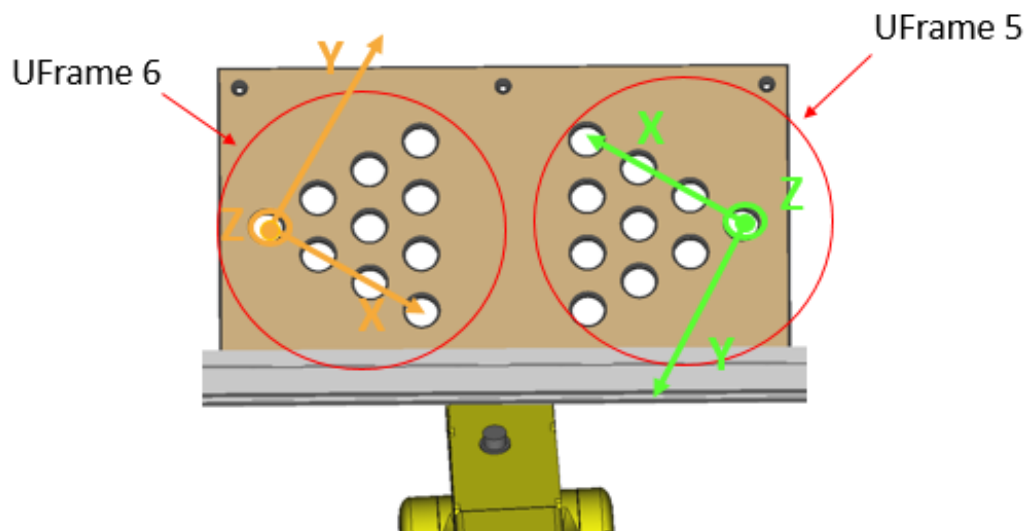
Ha a program sikeresen be lett tanítva, a User Frame felhasználható a tábla bal oldalán. (UFrame 6)

Hasonló programot szeretnénk a tábla jobb oldalára.

Ennek érdekében egy új felhasználói koordináta rendszert állítunk be a tábla másik oldalára, de ezúttal a robottól legtávolabbi lyukat használjuk X tengelypontként. Használja az 5. User Frame-et ehhez.



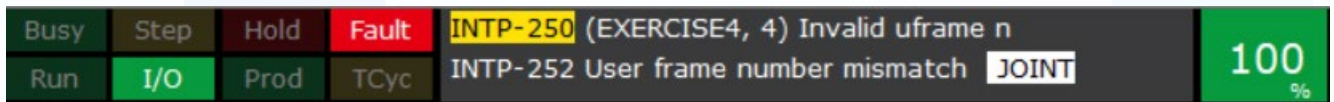
2. User Frame betanítása



2. User Frame betanítása

Most próbálja újra lefuttatni ugyanazt a programot. Figyeljen a User Frame megváltoztatására a programban, mert különben a program pontosan ugyanúgy fog eljárni, mint legutóbb. (UFrame 6-ról UFrame 5-re)

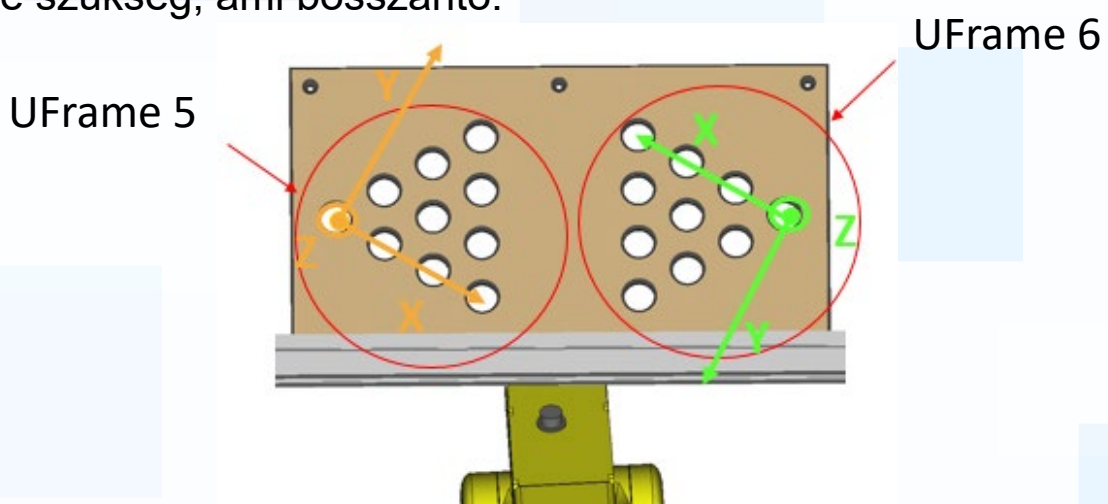
Észre fogja venni, hogy a program nem indul el, és a Teach Pendanton hiba jelenik meg.



Hogy a program megfelelően működjön, vegye fel ismét a User Frame 6-ot, most a tábla jobb oldalán.

A programban állítsa vissza a UFrame 6 használatát. A programnak így tökéletesen kell működnie a tábla jobb oldalán.

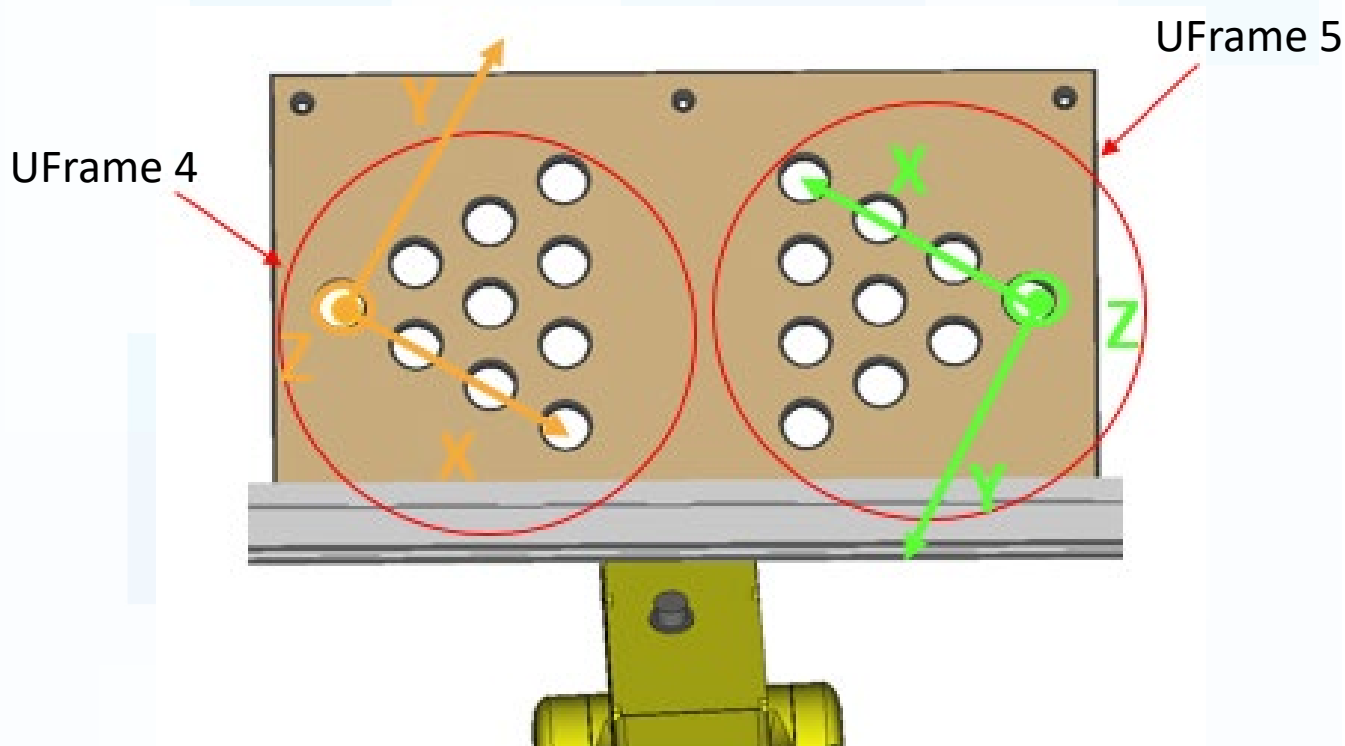
Most azt látjuk, hogy ezzel a módszerrel a programot a tábla egyik oldaláról a másikra át tudjuk vinni, csak a User Frame megváltoztatásával. De még mindig nem praktikus, mert most elvesztettük a bal oldali User Frame-et, és a program ismét bal oldalra történő áthelyezéséhez ismét ugyanarra a visszatanításra lenne szükség, ami bosszantó.



Egy jobb programozási módszer

A megoldás; olyan program készítése, amely mindkét oldalon képes dolgozni a User Frame-ek problémája nélkül.

Készítsen új programot. Tanítsa be a tábla bal oldalát mint UFrame 4, a jobb oldalit pedig UFrame 5-ként. Törölje a UFrame 6-ot.



Egy jobb programozási módszer

A Pozíció-regiszter bármely User Frame adatainak tárolására alkalmas. A User Frame 6 viszont bármely Pozíció-regiszterből feltölthető adatokkal, így a mozgás kivitelezése előtt tetszőleges User Frame adatok tölthetők bele. Így a User Frame váltások egyszerű adatmozgatással helyettesíthetők.

Bővebb információ a Regiszterek, Pozícióregiszterek használatáról az Operator MANUAL 4.5 FEJEZETBEN. "Register Instructions".

```
1: !Example
2: !Defining UFrame
3: UFRAME_NUM=6
4:L @P[1:Above Pick 1] 250mm/sec
: FINE
5:L @P[2:Pick Pos 1] 250mm/sec FINE
:
6: !Gripping the pin
7: RO[7:Open Gripper]=OFF
8: !Making sure the gripper engages
9: WAIT .30(sec)
```

```
1: !Example
2: !Save wanted UFrame in Pos Reg 10
3: PR[1]=UFRAME[4]
4: !Set UFrame 6 as wanted UFrame
5: UFRAME[6]=PR[10]
6: !Defining UFrame
7: UFRAME_NUM=6
8:L @P[1:Above Pick 1] 250mm/sec
: FINE
9:L @P[2:Pick Pos 1] 250mm/sec FINE
:
```

Figyelmeztetések

Figyelem!

A UFrame átfordításával az ellenkező oldalra lehetséges, hogy a robot elmozdul, vagy nem úgy viselkedik, mint amire Ön számított. Ez a User Frame változtatásának köszönhető, és könnyen hibákat okozhatunk, miközben megpróbáljuk megérteni a mozgás különbségét. Ez azt jelenti, hogy lassú sebesség beállítást kell használnia, és mindig kövesse a robot pontos útvonalát, és ellenőrizze, hogy nem lesz-e probléma a megfogó kábelével. Az első alkalommal a robotnak lépési módban kell lennie, hogy bármilyen probléma esetén könnyen megszakítsa a programot. Amikor a robot minden gond nélkül sikeresen végrehajtja a programot, fokozatosan növelheti sebességet, és később automata üzemre is válthat.

Ha nem figyel, a robot ütközhet a robotcellával, vagy a kábel szétszakadhat, ami a robot és a cella potenciális károsodását eredményezheti.



Összefoglalás

A gyakorlat végére megismerte a User Frame előnyeit a WORLD rögzített koordinátarendszeréhez képest.

Képesnek kell lennie a User Frame beállítására a hárompontos módszerrel.

Képesnek kell lennie egy program átvitelére az egyik UFrame-ről a másikkra.

Képesnek kell lennie arra, hogy megváltoztassa a programot úgy, hogy több, mint egy meghatározott User Frame-mel működjön, és így rugalmasabbá teszi a programját.



Függelék

A példaprogram áthelyez egy csapot egyik lyukból a másikba.

Figyelem!

Szükséges megtanítania a pozíciókat, nem elég ezt a programot csak átmásolni, mivel P [1] - P [6] előzőleg megtanították.

Megjegyzés:

A megfogási és elengedési pozíciók nem azonosak. Ez annak a ténynek köszönhető, hogy a csap nem biztos, hogy teljesen benne van a megfogóban, ezért a csap leütközhet az asztallal, ezért jobb, ha az elengedési magasságot valamivel magasabbra állítja, mint a tényleges felvételi magasságot.

```

PAUSED 1/29
1: !Example
2: !Defining UFrame
3: UFRAME_NUM=6
4:L @P[1:Above Pick 1] 250mm/sec
: FINE
5:L @P[2:Pick Pos 1] 250mm/sec FINE
:
6: !Gripping the pin
7: RO[7:Open Gripper]=OFF
8: !Making sure the gripper engages
9: WAIT .30(sec)
    
```

```

PAUSED 23/29
17: CALL AA_HOME
18: WAIT 2.00(sec)
19:L @P[3:Above Pick 2] 250mm/sec
: FINE
20:L @P[5:Pick Pos 2] 250mm/sec FINE
:
21: RO[7:Open Gripper]=OFF
22: WAIT R[...]<>...
23:L @P[3:Above Pick 2] 250mm/sec
: FINE
24:L @P[1:Above Pick 1] 250mm/sec
    
```

```

PAUSED 16/29
10:L @P[1:Above Pick 1] 250mm/sec
: FINE
11:L @P[3:Above Pick 2] 250mm/sec
: FINE
12:L @P[4:Release Pos 2] 250mm/sec
: FINE
13: !Releasing the Pin
14: RO[7:Open Gripper]=ON
15:L @P[3:Above Pick 2] 250mm/sec
: FINE
16: !Calling predefined program
    
```

```

PAUSED 29/29
23:L @P[3:Above Pick 2] 250mm/sec
: FINE
24:L @P[1:Above Pick 1] 250mm/sec
: FINE
25:L @P[6:Release Pos 1] 250mm/sec
: FINE
26: RO[7:Open Gripper]=ON
27:L @P[1:Above Pick 1] 250mm/sec
: FINE
28: CALL AA_HOME
[End]
    
```





Gyakorlat 5

Offset és Pozícióregiszter

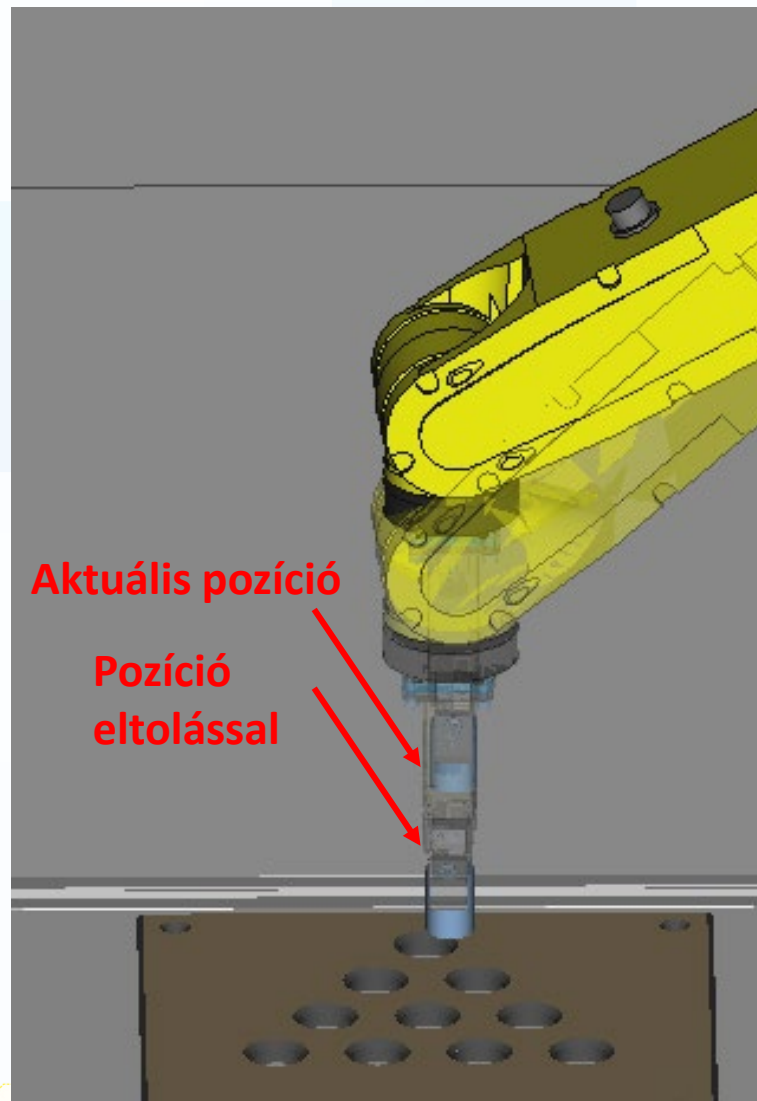
2. lap



Bevezetés

Ebben a gyakorlatban megismerheti az Offset fogalmát és használatát. Megtanulhatja, hogyan kell programozni egy eltolást a robot Pozícióregiszterében (PR), és ezt az eltolást később használni a programban.

Képes lesz a programozásra közvetlen helyezettanítás használata nélkül, miközben a PR-t használja a szerszám pozicionálásához, amely a programozást hatékonyabbá, gyorsabbá és könnyebbé teszi.



3. lap

Gyakorlat 5

Offset és Pozícióregiszter

Tartalom

Bevezetés	-	3
Háttér	-	5
Felszerelés	-	5
Pozíció tárolása a Pozícióregiszterben	-	6
Offset tárolása a Pozícióregiszterben	-	8
Robot	-	10
Példaprogram	-	11
Összefoglalás	-	14
Függelék	-	15



Háttér

Ez a gyakorlat hasonló lesz az előzőhöz, és ami a legfontosabb: a program hatása / végrehajtása ugyanaz lesz (ha helyesen hajtják végre).

A Tool Offset segítségével, amint a neve is sugallja, elmozdíthatja a megfogót/szerszámot egy bizonyos értékkel. Ez különösen praktikus lehet, ha ugyanazt a pozíciót használja, például különböző magasságokban, - ahogy a példaprogramban, - egy lyuk felett, vegyük fel a csapot és engedjük el ugyanezt a csapot.

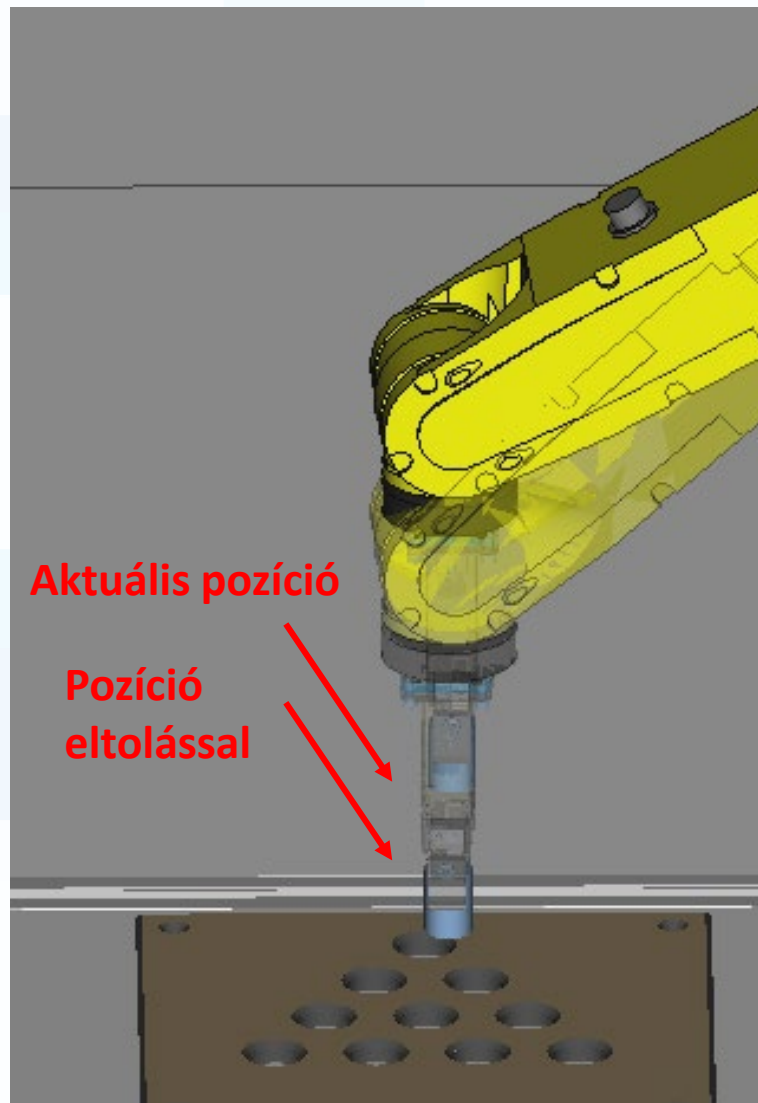
A Pozícióregiszter egy pozíció vagy egy eltolás mentésére szolgál, mint egy változó a robotban. A Pozícióregiszter beállítható a program végrehajtása előtt vagy a program végrehajtása közben. Az Offset és a Pozícióregiszter használatával sokkal egyszerűbbé tehetjük programjainkat és kevesebb pozíció betanításával, mint korábban.

Bővebb információk az Operator Manual 4.5 fejezetben. "Register Instructions".



Háttér

Az alábbi képen példát lát a Tool Offset használatáról:

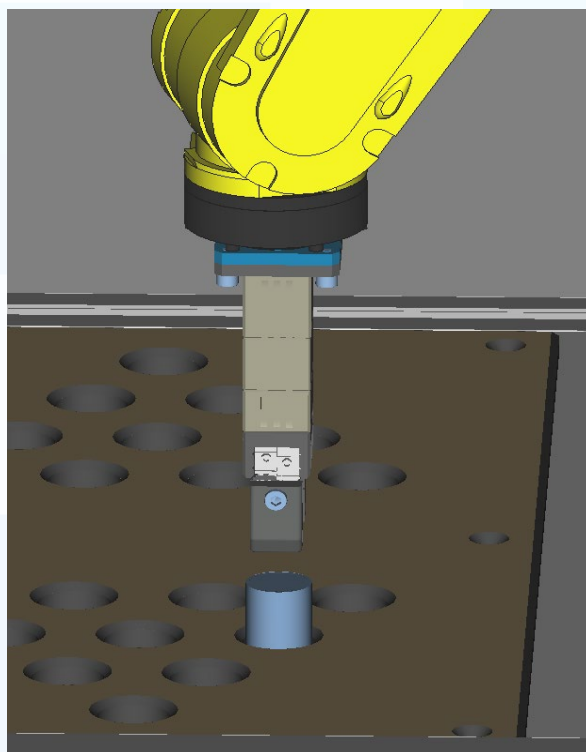
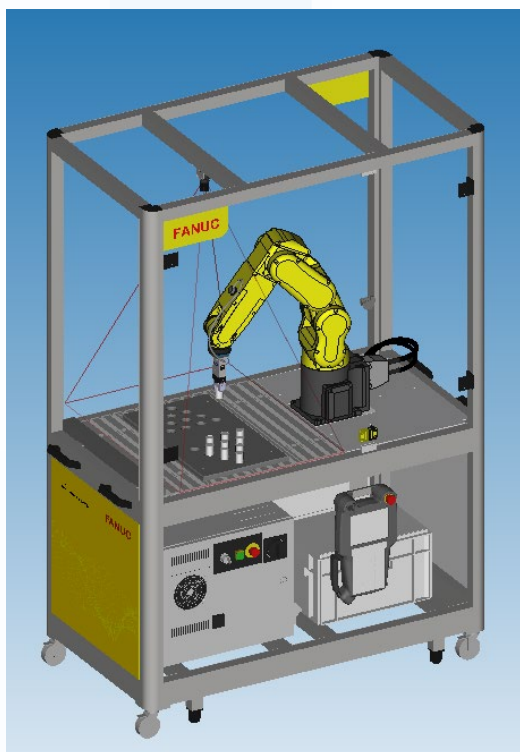


Ebben az esetben Z-irányú eltolást használtunk. Ha ezt az eltolást hozzáadja a pozícióhoz, akkor megváltozik a helyzet Z-koordinátája és ennél fogva a magassága is a koordináta-rendszerben.



Felszerelés

A gyakorlathoz szükséges:
Oktatócella
Hengeres munkadarab

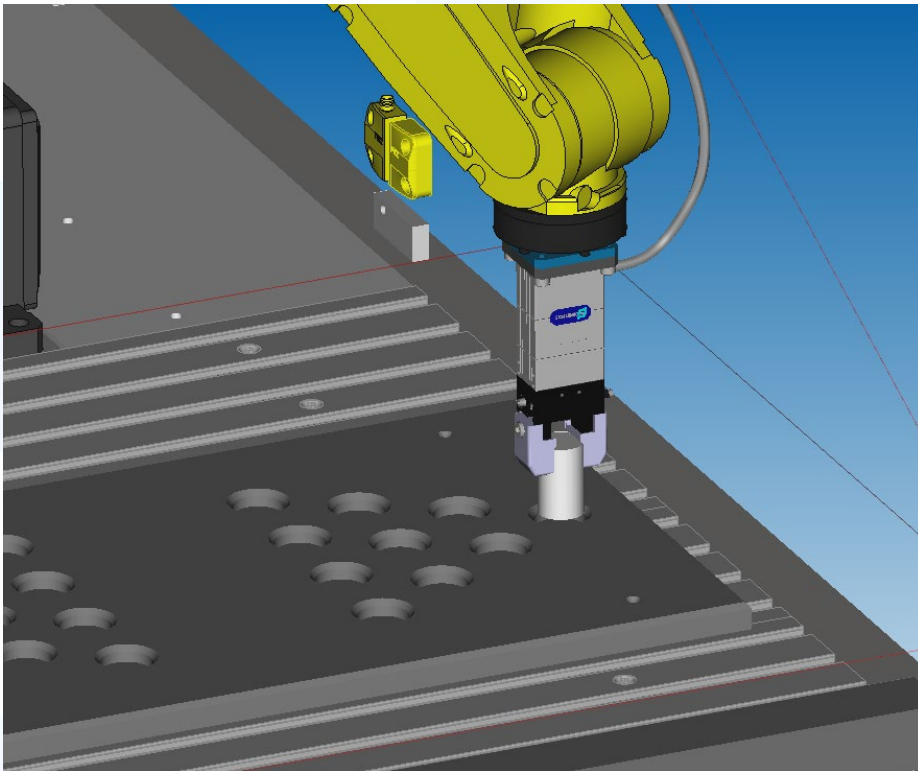


Pozíció tárolása Pozícióregiszterben

Mint már említettük, a Pozícióregiszter egyfajta változó; robot pozíciókhoz és az eltolásokhoz.

Pozíció beállításához vagy közvetlenül beírhatja az értékeket (a „POSITION” gombbal), vagy rögzítheti a pozíciókat a robotból. (A „RECORD” gombbal)

Kezdje azzal, hogy beállít a munkadarabok mozgásához két pozíciót (nemcsak két pozíciót rögzíthet, így a munkadarabokat nem csak két lyukba teheti, de csak a két első helyzetet tárgyaljuk, a másik pozíciók pontosan ugyanúgy működnek).



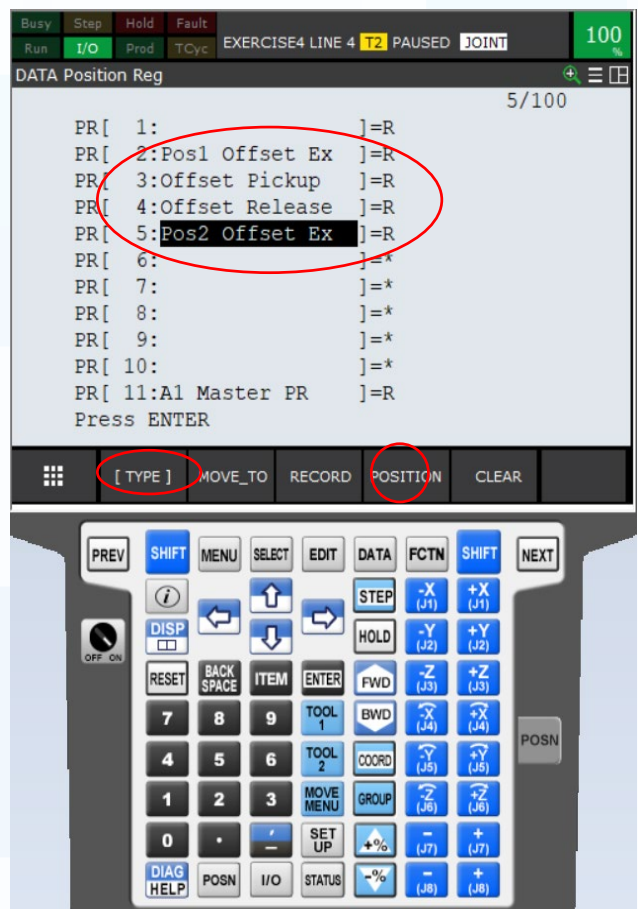
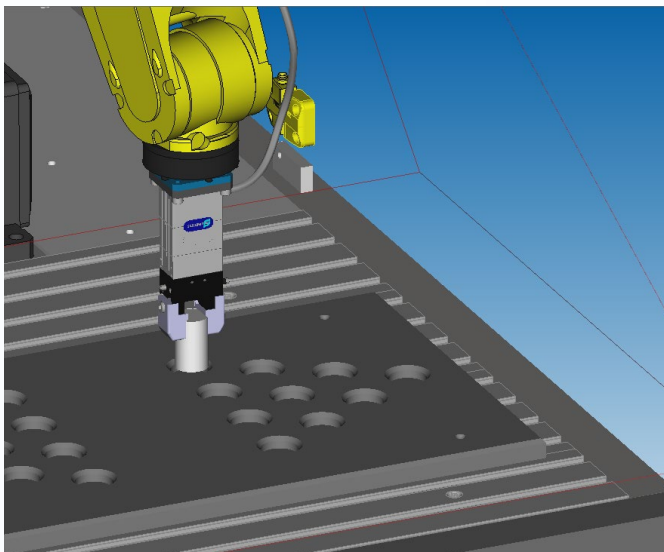
Tegye a csapot pontosan a furatba, majd menjen felfelé a Z tengelyen, hogy a csap tökéletesen illeszkedjen a furathoz és a megfelelő magasságban legyen.



Pozíció tárolása Pozícióregiszterben

Ha a pozíciót rögzíteni szeretné Pozícióregiszterben, nyomja meg a DATA gombot, az F1 [Type] menüben válassza ki a Pozíció regisztereket. A kurzorral válassza ki azt a PR-t, amelybe rögzíteni szeretné a pozíciót.

```
PR[ 1: ]=R
PR[ 2:Pos1 Offset Ex ]=R
PR[ 3:Offset Pickup ]=R
PR[ 4:Offset Release ]=R
PR[ 5:Pos2 Offset Ex ]=R
PR[ 6: ]=*
PR[ 7: ]=*
PR[ 8: ]=*
```



Ezután hajtsa végre ugyanezt az eljárást a beállítani kívánt második furaton. Miután beállította ezt a pozíciót, lépjen az 1. pozícióba, írja le az adott pozíció Z-magasságát, és írja be ugyanezt az értéket (Pozíció) manuálisan a 2. pozícióba, hogy mindkét pozíciónak ugyanaz a Z-értéke legyen, és ugyanazokat az eltolásokat tudja használni.

Offset tárolása Pozícióregiszterben

Most állítson be egy eltolást a PR-ben. Az eltolásnak csak a Z tengelyben kell lennie, és az első pozícióként választott pozíciótól függ (az első furat felett).

Ez a beállítás próbán és hibán alapul. Először alkalmazzon egy kis eltolást, majd fokozatosan növelje az eltolást, amíg meg nem kapja a kívánt eltolást. Ügyeljen arra, hogy a robot sebessége kis értéken legyen, így megállíthatja a robotot, ha az messzire vagy rossz irányba mozog.

Ennek legjobb módja egy hasonló program megírása, mint a 4. gyakorlatban, miközben a pozícióként beállított PR értékeket, majd az eltolást használja. Ezután hagyja, hogy a robot végrehajtsa az eltolást, és ellenőrizze, hogy az eltolás helyes-e vagy sem. Ha nem, akkor térjen vissza a PR-hez, változtassa meg kissé az eltolás értékét, és próbálja újra.

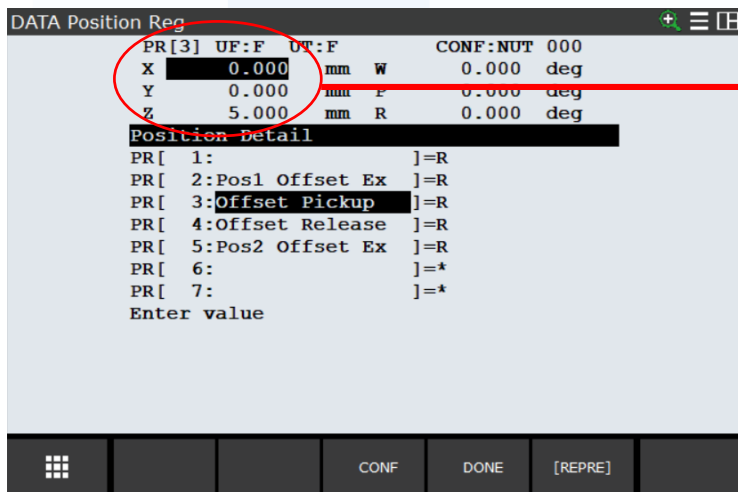
Azt is javasoljuk, hogy adjon meg két eltolást: az egyik eltolást a munkadarab felvételéhez, amelynek pontosan abban a magasságban kell lennie, amelyen a megfogó meg tudja fogni a munkadarabot, és egy másodikat egy kicsit magasabbnak, a munkadarab elengedéséhez, arra az esetre, ha a munkadarab nincs teljesen a markolóban.

Bővebb információk az Operator Manual 4.11 fejezetben "Offset Condition Instruction" és a 4.12 fejezetben. "Tool Offset Condition Instruction"



Offset tárolása Pozícióregiszterben

A PR-t eltoláshoz ugyanúgy kell használnia, mint a pozícióknál, csak a felvétel helyett a „POSITION” pontra kell lépnie, és manuálisan be kell írnia az értéket. Tegyen minden értéket, (kivéve Z-t) 0-ra, majd kezdje kis Z értékekkel (például 5).



PR[3]	UF:F	UT:F	CONF:NUT	000
X	0.000	mm W	0.000	deg
Y	0.000	mm P	0.000	deg
Z	5.000	mm R	0.000	deg



Példaprogram

Most, hogy mind a Pozíciót, mind az Eltolást beállította a PR-ben, létrehozhat egy új programot, csak ezúttal a pozíciók használata helyett közvetlenül használja a PR-ben beállított pozíciókat, majd a szerszámeltolás funkcióval mozogjon lefelé, hogy megragadja vagy elengedje a munkadarabot.

FONTOS

Feltételezzük, hogy tudja, hogyan kell elkészíteni ezt a programot. Ha nem, kérjük, térjen vissza, és ismételje meg a 4. gyakorlatot. A függelékbe egy referencia programot tettünk.

Ha egy PR-pozíciót szeretne használni, egyszerűen nyomja meg a rögzítés gombot. Mozgassa a kurzort a pozícióba, és változtassa meg a P[] értéket PR[] értékre, majd állítsa be a használni kívánt PR számát.

Amint elkészítette a programot, valószínűleg észreveszi, hogy kevesebb időbe telik az ilyen programozás, mint a másik (egyszerűbb) módszerrel.

Figyelnie kell azonban arra, hogy ez a módszer attól függ, hogy melyik Felhasználói és Szerszám koordináta-rendszert használja. Helytelen koordináta-rendszer használata a robot nem kívánt mozgását eredményezi, ami emberi sérüléseket és / vagy a robot károsodását eredményezheti.

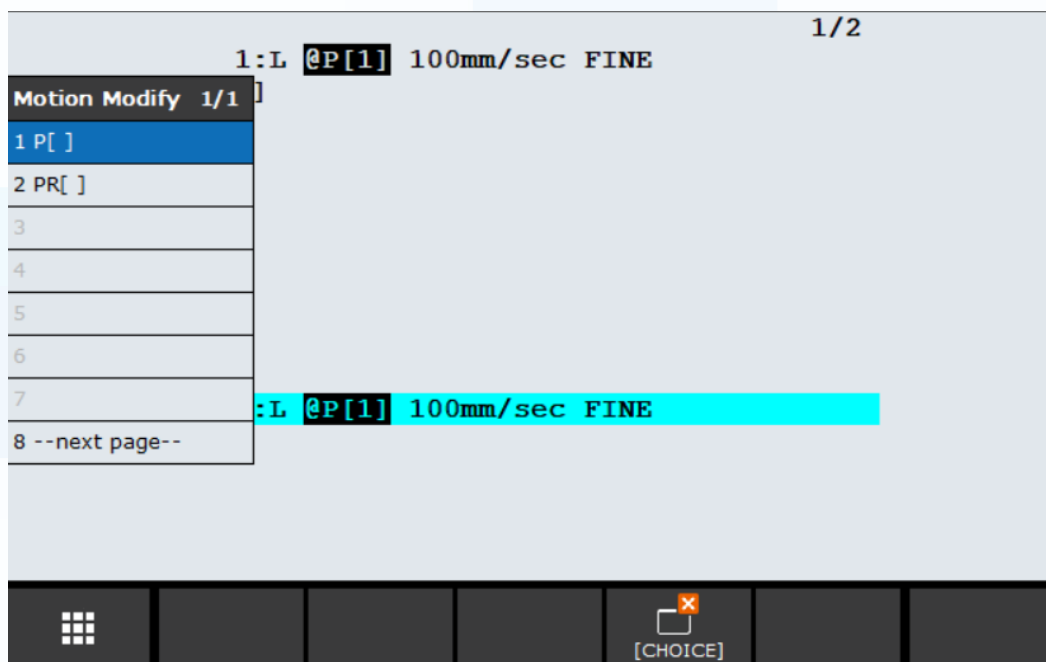


Példaprogram

Elsőként rögzítsen egy tetszőleges pozíciót, majd a kurzorral álljon a pozíció sorszámára.

```
1:L @P[1] 100mm/sec FINE
[End]
```

Nyomja meg a CHOICE gombot, majd válassza ki a PR[] lehetőséget a P[] lecserélésére a mozgásban.



Összefoglalás

Ennek a gyakorlatnak a során megismerte a pozíciók és eltolások bevitelét a pozícióregiszterekbe.

Képes értékek rögzítésére vagy pozíciók beállítására a PR-ben. Képes használni ezeket az értékeket a programjában önálló pozícióként vagy eltolásként.

Képes arra, hogy teljes egészében PR-alapú programot készítsen, anélkül, hogy minden egyes pontot rögzítenie kellene.



Függelék

Itt van a program, amelyre ebben a gyakorlatban hivatkoztunk. Felhívjuk figyelmét, hogy a PR-ben saját mentett értékek vannak, így ezek az értékek a saját beállításai szerint változhatnak.

1/34

```

1: !Move to Pos1
2:L @PR[2:Pos1 Offset Ex] 100mm/sec
: FINE
3: !Make the pickup offset active
4: TOOL_OFFSET CONDITION
: PR[3:Offset Pickup]
5: !Move again to Pos1 with offset
6:L @PR[2:Pos1 Offset Ex] 100mm/sec
: FINE Tool_Offset
7: !Grab the pin
8: RO[7:Open Gripper]=OFF
    
```

29/34

```

24: TOOL_OFFSET CONDITION
: PR[3:Offset Pickup]
25:L @PR[5:Pos2 Offset Ex] 100mm/sec
: FINE Tool_Offset
26: RO[7:Open Gripper]=OFF
27:L @PR[5:Pos2 Offset Ex] 100mm/sec
: FINE
28:L @PR[2:Pos1 Offset Ex] 100mm/sec
: FINE
29: TOOL_OFFSET CONDITION
: PR[4:Offset Release]
    
```

15/34

```

9: !Move to Pos1 without offset
10:L @PR[2:Pos1 Offset Ex] 100mm/sec
: FINE
11: !Move to Pos2
12:L @PR[2:Pos2 Offset Ex] 100mm/sec
: FINE
13: !Make the release offset active
14: TOOL_OFFSET CONDITION
: PR[4:Offset Release]
15: !Move to Pos2 with offset
16:L @PR[5:Pos2 Offset Ex] 100mm/sec
    
```

34/34

```

28:L @PR[2:Pos1 Offset Ex] 100mm/sec
: FINE
29: TOOL_OFFSET CONDITION
: PR[4:Offset Release]
30:L @PR[2:Pos1 Offset Ex] 100mm/sec
: FINE Tool_Offset
31: RO[7:Open Gripper]=ON
32:L @PR[2:Pos1 Offset Ex] 100mm/sec
: FINE
33: CALL AA_HOME
[End]
    
```

22/34

```

16:L @PR[5:Pos2 Offset Ex] 100mm/sec
: FINE Tool_Offset
17: !Release the pin
18: RO[7:Open Gripper]=ON
19: !Move to Pos2 without offset
20:L @PR[5:Pos2 Offset Ex] 100mm/sec
: FINE
21: !Call another Program
22: CALL AA_HOME
23:L @PR[5:Pos2 Offset Ex] 100mm/sec
: FINE
    
```





Gyakorlat 6

Haladó programozás

2. lap



Bevezető

Ebben a gyakorlatban alaposabban áttekintjük a robot programozásához felhasználható különféle lehetőségeket.

Egy egyszerű program segítségével, amelyet a gyakorlat során megváltoztatunk, meg szeretnénk mutatni az összes különböző lehetőséget, és ezzel arra készítjük Önt, hogy gondolkodjon saját „bonyolultabb” programjainak létrehozásáról.

Nagyon fontos, hogy sikeresen elvégezze a 3. gyakorlatot, mert ez a jelenlegi gyakorlat a 3. gyakorlatban megszerzett ismeretekre épít. Ebben a gyakorlatban az összes „megoldást” közvetlenül az feladatok után helyezzük el. Mindazonáltal meg kellene próbálnia a feladatokat megoldani anélkül, hogy előre megvizsgálná ezeket a „megoldásokat”.

Próbáljon meg mindenhova megjegyzést fűzni ahhoz, amit leprogramozott, hogy megkönnyítse a program elolvasását más emberek számára. Ez nem nagyon fontos ennél a gyakorlatnál, de mivel gyakorlatról van szó, ezt is gyakorolnia kell.

```
ADVANCED_PROG_EX_2 1/12
1: !UFrame Set
2: UFRAME_NUM=1
3: !Loop Instruction
4: FOR R[32:ForTo Value Ex]=1 TO 10
5: !Setting R33
6: R[33:If Value Ex]=
  : R[32:ForTo Value Ex] MOD 2
7: !If Commands
8: IF R[33:If Value Ex]=0,
  : CALL ADVANCED_PROGRAM_PART1
9: IF R[33:If Value Ex]=1,

ADVANCED_PROG_EX_2 12/12
5: !Setting R33
6: R[33:If Value Ex]=
  : R[32:ForTo Value Ex] MOD 2
7: !If Commands
8: IF R[33:If Value Ex]=0,
  : CALL ADVANCED_PROGRAM_PART1
9: IF R[33:If Value Ex]=1,
  : CALL ADVANCED_PROGRAM_PART2
10: !End of For loop
11: ENDFOR
[End]
```



Gyakorlat 6

Haladó Programozás

Tartalom

Bevezetés	-	3
Háttér	-	5
Felszerelés	-	6
Első program	-	7
Törlés	-	9
“For To” funkció	-	10
Pozícióregiszter	-	13
“If” funkció	-	14
Másodlagos programok	-	15
Ugrás és címke	-	16
CNT	-	17
“Wait” funkció	-	19
Összefoglalás	-	20



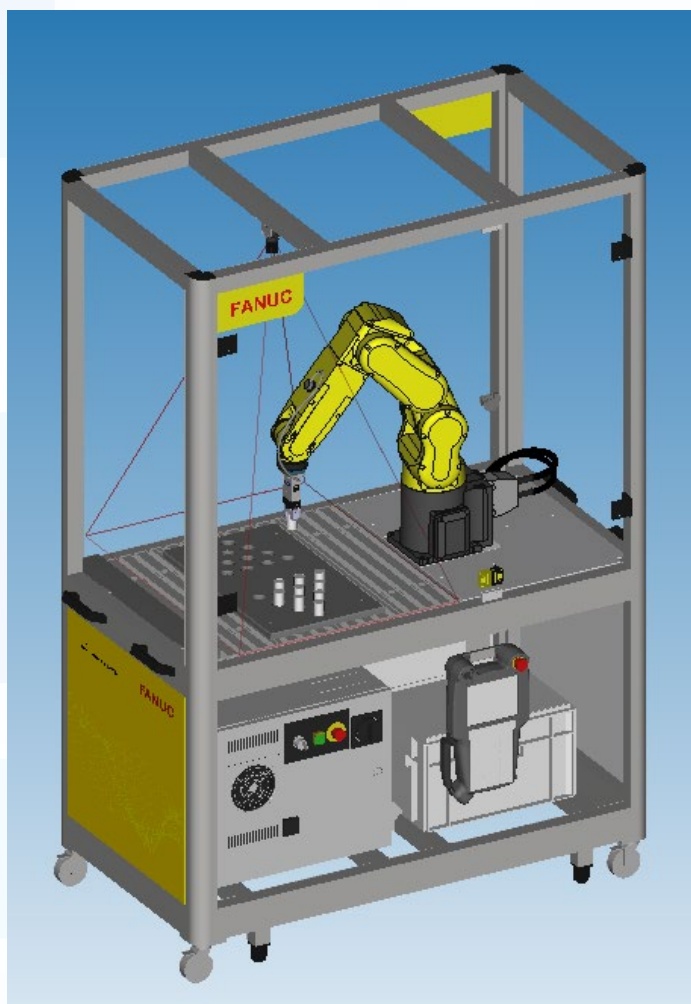
Háttér

Mostanáig már megismerhette a programozás alapjait. A 4. és 5. gyakorlatban már látott néhány fejlettebb technikát, de most egy kicsit részletesebben szeretnénk elemezni, hogy mit tett ott, és még mélyebben elmélyülni abban, hogy mit lehet tenni még a robottal. Ez fontos, mivel a legtöbb alkalmazás utólag nem lesz olyan egyszerű, mint egy egyszerű lépés két vagy több pont között. Például a robot vizsgálhatja a különböző szituációkat, majd eldöntheti, melyik alprogramot hívja meg.



Felszerelés

Szükséges eszköz:
Oktatócella



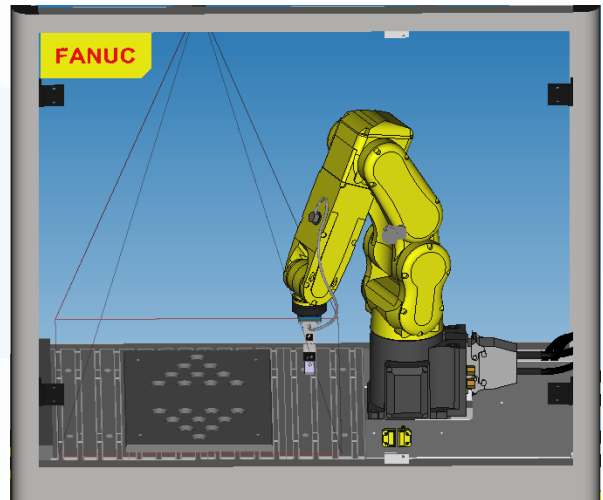
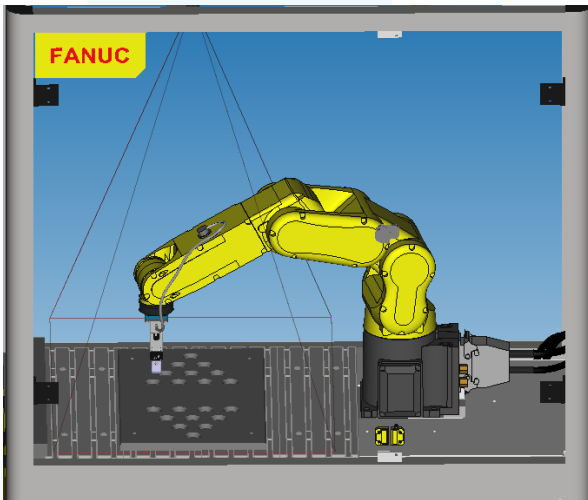
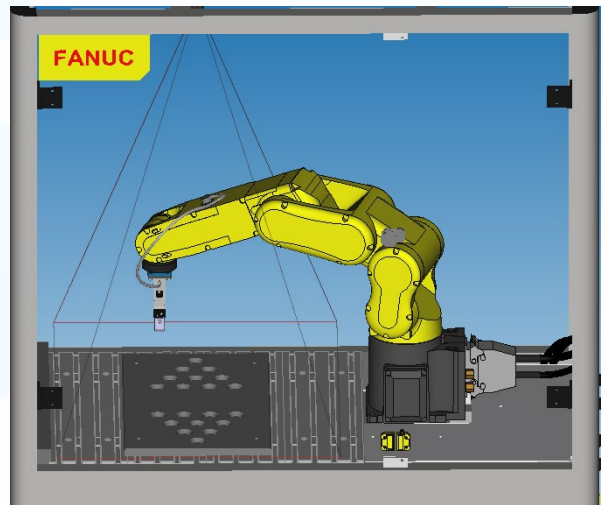
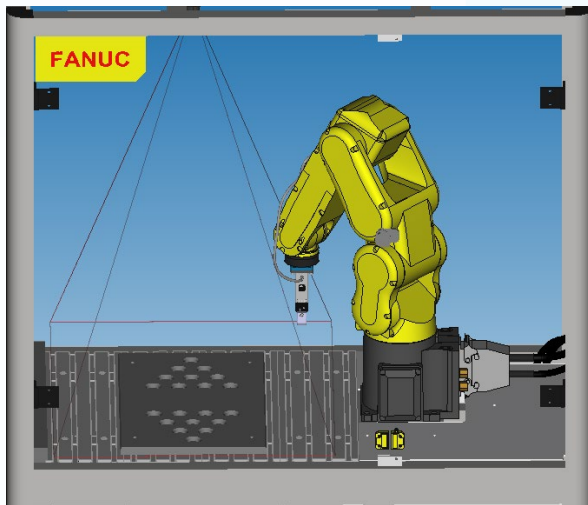
Első program

Első lépésben állítson be egy User Frame-et. Legyen ez az 1.-es. Nyomja meg az F1 [Inst] gombot, majd válassza ki a menüből az Offset/Frames lehetőséget, azon belül pedig a UFRAME_NUM=Constant lehetőséget. Állítsa ezt 1-re.



Első Program

Kezdje a programozást 4 különböző pont felvételével, amelyeket lineárisan köt össze 100mm/s sebességgel. Legcélszerűbb a pontokat egy négyzetet formálva felvenni. Lehetőség szerint használjon azonos magasságokat. Futtassa le a programot!

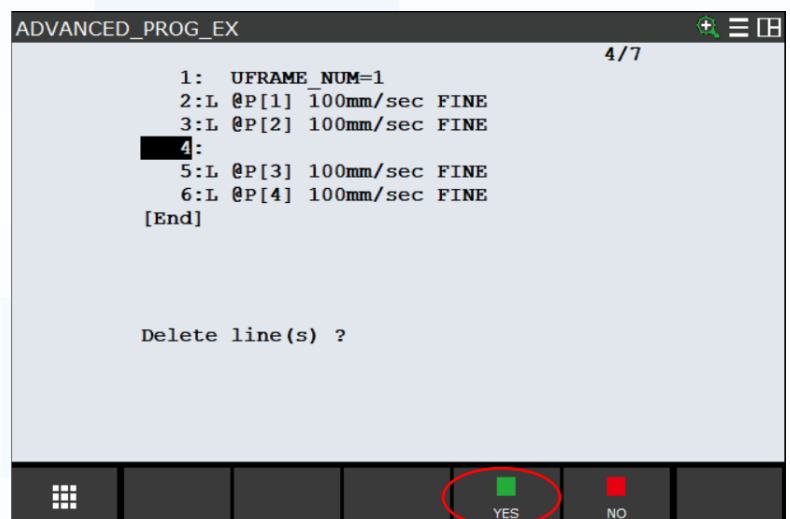
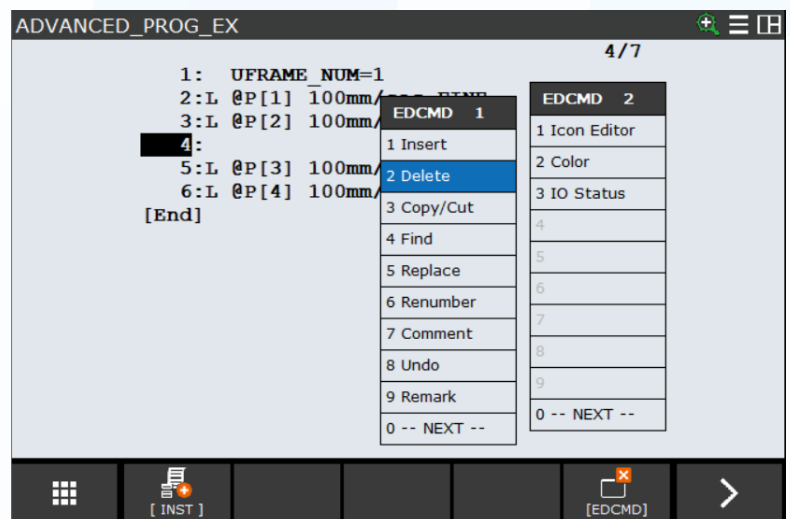


Törlés

Illesszen be egy üres sort a 2. és a 3. mozgás közé. Ha ez a sor nem tartalmaz utasítást, akkor nincs kihatással a program működésére. Ennek a sornak a törléséhez (ha üres vagy ha parancs található benne), nyomja meg az F5 [EDCMD] gombot, majd válassza a DELETE lehetőséget. A rendszer rákérdez a sor törlésére. Nyomja meg az F4-et.

Figyelem!

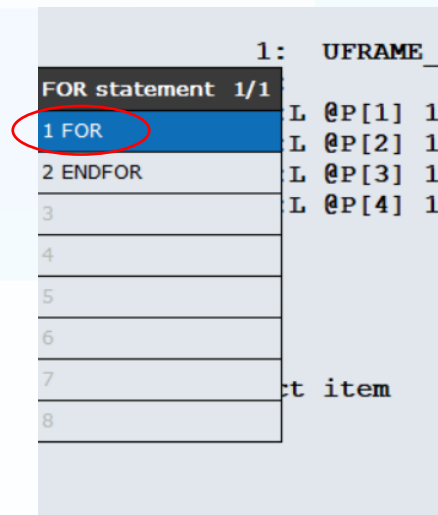
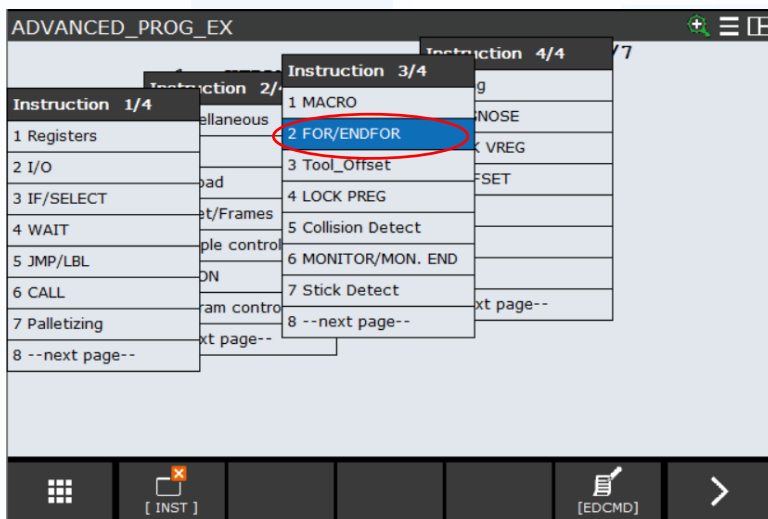
Egy sor törlése komoly károkat okozhat a programban, ezért a sor törlése előtt legyen teljesen tisztában azzal, hogy ez mit fog változtatni a programjában.



“For To” funkció

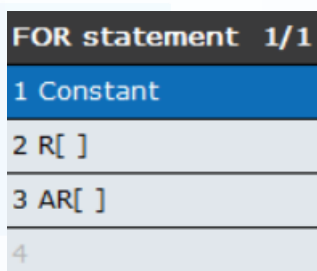
Egy mozgási sorozat, vagy egyéb utasítások többször ismételt végrehajtására alkalmas a “For to” funkció.

Illesszen be egy üres sort a mozgási utasítások elé, majd nyomja meg az F1[INST] gombot. Válassz a FOR/ENDFOR, azon belül pedig a FOR menüpontot.



Válasszon egy regisztert. Legyen biztos abban, hogy a kiválasztott regiszter nincs használva más programban. (Használhatja például a R[32]-t, ez általában nincs máshol használva)

Válassza a “Constant” típust kétszer, majd adja meg az 1 és 10 értékeket.



2: FOR R[32]=1 TO 10

“For To” Funkció

Ez azt jelenti, hogy a program megismétli az eljárást, és minden ismétlésnél a regiszter értéke eggyel nő. Tehát 1-től 10-ig fog jelen esetben számolni.

Lépjen a Regiszterek lapjára (nyomja meg a DATA gombot, és az F1[Type] gombbal válassza a Register lapot) nevezze át a R[32]-t a már megismert módon. Ez a név a programban is megjelenik majd.

```
R[ 30: ]=1  
R[ 31: ]=0  
R[ 32: ForTo Value Ex ]=30  
R[ 33: ]=5  
R[ 34: ]=0
```

```
ADVANCED_PROG_EX 3/7  
1: UFRAME_NUM=1  
2: FOR R[32:ForTo Value Ex]=1 TO 10  
3:L @P[1] 100mm/sec FINE  
4:L @P[2] 100mm/sec FINE  
5:L @P[3] 100mm/sec FINE  
6:L @P[4] 100mm/sec FINE  
[End]
```

[INST] [EDCMD] >

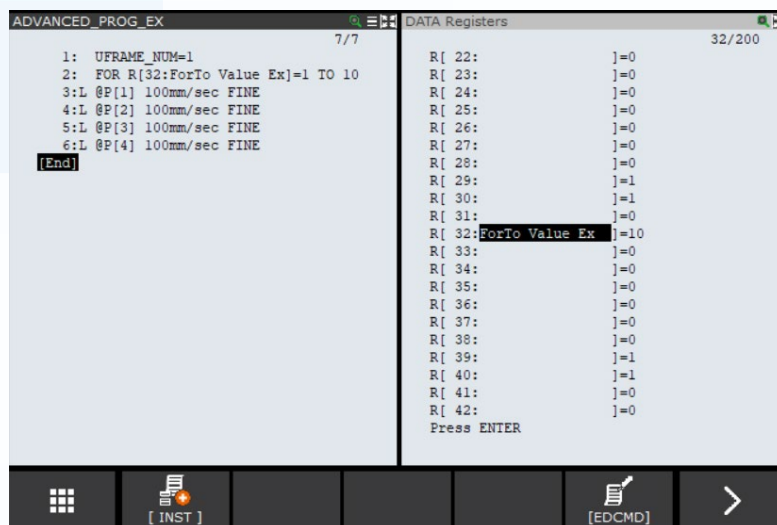


“For To” Funkció

Indítsa el a programját. Látni fogja, hogy a program nem indul el. Ennek oka, hogy a robot nem tudja, hogy a FOR utasítás mekkora tartományra érvényes. Ennek megadásához helyezzen el a program végére egy ENDFOR utasítást. (Ugyanúgy, ahogyan a FOR utasítást helyezte el)

Indítsa el újra a programot. Most a programnak a teljes mozgást 10-szer kell végrehajtania.

Hogy megtekinthesse, mi történik mindeközben a regiszterrel, a képernyőt kettéosztjuk. Nyomja meg a SHIFT + DISP gombokat, majd válassza a DOUBLE lehetőséget. A képernyő két részre oszódik. A képernyők közti váltáshoz használja a DISP gombot SHIFT nélkül. Lépjen a jobb oldali képernyőre, majd nyomja meg a DATA gombot. Állítsa be a Register képernyőt. A regiszter állapota így megtekinthető. Indítsa el a programot, majd figyelje meg, mi történik a regiszter értékével.

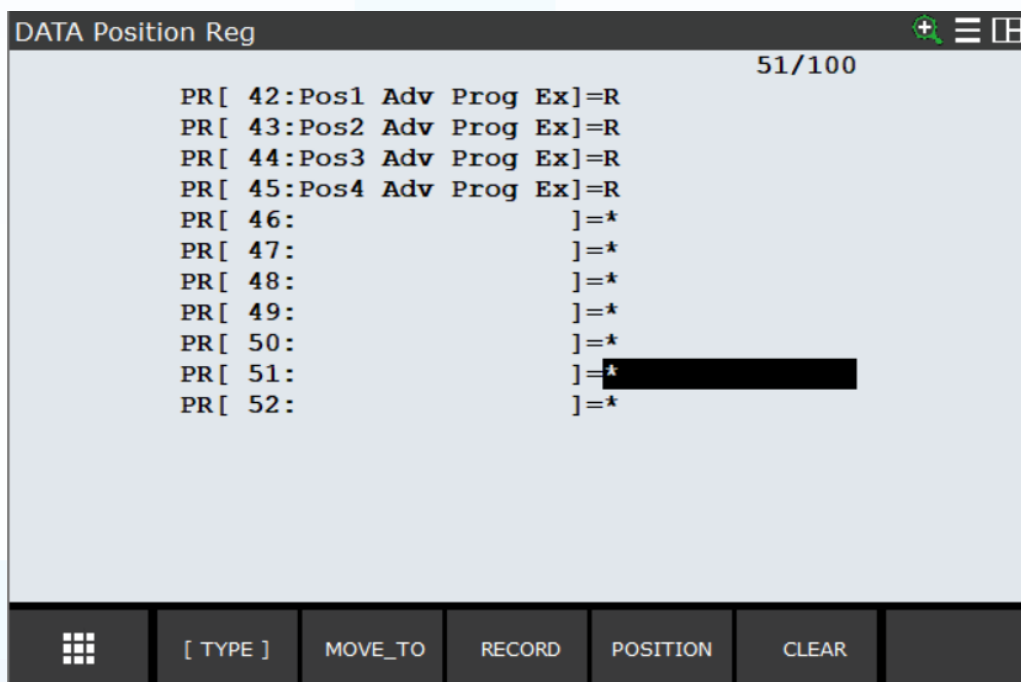


Bővebb információk a FOR-TO-ENDFOR használatáról az Operator Manual 4.18 fejezetben. “For/Endfor Instruction”.



Position Register – Pozíció regiszter

Tároljon le néhány különböző pontot pozíció regiszterekben a következő programhoz.



Készítsen két új programot a következő nevekkel :
Advanced_Program_part1 és Advanced_Program_part2.



“If” Funkció

Szeretnénk elérni, hogy programunk egyszer elvégezze a mozgást az egyik irányba, és egyszer az ellenkező irányba. Ezért használunk „If” függvényt. Arra használják, hogy megvizsgálják, az IF után írt vizsgálat/állapot igaz-e vagy sem. Ha igaz, akkor végrehajt egy adott műveletet, ha nem, akkor csak a következő parancssorra lép.

Célunk elérése érdekében helyezzünk el egy regiszter utasítást, ami az “R[32] MOD 2” értéket kapja. (minden második alkalommal 0 értéket kapunk). Írja be ezt az értéket R [33]-ba (vagy egy másik szabad regiszterbe).

Most szükséges két IF parancs beillesztése. Az első meghívja az Advanced_Program_part1 programot, R[33]=0, a másik meghívja az Advanced_Program_part2 programot, ha R[33]=1.

```

ADVANCED_PROG_EX 1/12
1: !UFrame Set
2: UFRAME NUM=1
3: !Loop Instruction
4: FOR R[32:ForTo Value Ex]=1 TO 10
5: !Setting R33
6: R[33:If Value Ex]=
  : R[32:ForTo Value Ex] MOD 2
7: !If Commands
8: IF R[33:If Value Ex]=0,
  : CALL ADVANCED_PROGRAM_PART1
9: IF R[33:If Value Ex]=1,

```

```

ADVANCED_PROG_EX 12/12
5: !Setting R33
6: R[33:If Value Ex]=
  : R[32:ForTo Value Ex] MOD 2
7: !If Commands
8: IF R[33:If Value Ex]=0,
  : CALL ADVANCED_PROGRAM_PART1
9: IF R[33:If Value Ex]=1,
  : CALL ADVANCED_PROGRAM_PART2
10: !End of For loop
11: ENDFOR
[End]

```

Bővebb információk az IF funkcióról az Operator Manual 4.7.4 fejezetben. “Conditional Branch Instructions” és a 4.7.5 fejezetben “If then/Else/Endif Statement”.

FONTOS

Az ELSE funkciót nem tárgyaljuk ebben a dokumentumban.



Másodlagos programok

Az első programnak a mozgást az “egyik irányban körbe” míg a második programnak a “másik irányba körbe”.

ADVANCED_PROGRAM_PART1

5/5

```
1:L @PR[44:Pos3 Adv Prog Ex]
: 100mm/sec FINE
2:L @PR[43:Pos2 Adv Prog Ex]
: 100mm/sec FINE
3:L @PR[42:Pos1 Adv Prog Ex]
: 100mm/sec FINE
4:L @PR[45:Pos4 Adv Prog Ex]
: 100mm/sec FINE
[End]
```

ADVANCED_PROGRAM_PART2

5/5

```
1:L @PR[42:Pos1 Adv Prog Ex]
: 100mm/sec FINE
2:L @PR[43:Pos2 Adv Prog Ex]
: 100mm/sec FINE
3:L @PR[44:Pos3 Adv Prog Ex]
: 100mm/sec FINE
4:L @PR[45:Pos4 Adv Prog Ex]
: 100mm/sec FINE
[End]
```



Ugrás és Cimke

Lehetőség van ugrási parancsok használatára. A JUMP LBL segítségével egy adott címkére ugorhat (amelyet a programban határozhat meg).

A JUMP LBL használatához nyomja meg az F1[Inst,] majd a (JMP / LBL) gombot.

Az Advanced_Program_Part1 programot megváltoztattuk az Ugrás/JMP parancsokkal való használat bemutatására. Amint láthatja, nagyon rendetlen lett, és ebben az esetben teljesen haszontalan, viszont láthatja, hogyan működik. A két program egyformán működik.

```
ADVANCED_PROGRAM_PART1 5/5
1:L @PR[44:Pos3 Adv Prog Ex]
: 100mm/sec FINE
2:L @PR[43:Pos2 Adv Prog Ex]
: 100mm/sec FINE
3:L @PR[42:Pos1 Adv Prog Ex]
: 100mm/sec FINE
4:L @PR[45:Pos4 Adv Prog Ex]
: 100mm/sec FINE
[End]
```

```
ADVANCED_PROGRAM_PART1 1/11
1: JMP LBL[1]
2: LBL[2]
3:L @PR[45:Pos4 Adv Prog Ex]
: 100mm/sec FINE
4: JMP LBL[3]
5: LBL[1]
6:L @PR[44:Pos3 Adv Prog Ex]
: 100mm/sec FINE
7:L @PR[43:Pos2 Adv Prog Ex]
: 100mm/sec FINE
8:L @PR[42:Pos1 Adv Prog Ex]
: 100mm/sec FINE
9: JMP LBL[2]
10: LBL[3]
[End]
```

Bővebb információk a JUMP használatáról az Operator Manual 4.7.1 fejezetben. "Label Instruction" és a 4.7.3 fejezetben. "Unconditional Branch Instructions".



CNT

Most hatékonyabbá akarjuk tenni programunkat, hogy gyorsabb legyen. Ehhez a CNT függvényt használhatjuk a FINE függvény helyett. Ez azt jelenti, hogy a robot nem mozog pontosan egy pontra, hanem csak egy pont környezetében mozog, majd megállás nélkül folytatja.

A CNT mozgásáról további információkat az Operator Manual 4.3.4 „Positioning Path” fejezetben talál.

A különbség megtekintéséhez változtassa meg az Advanced_Program_Part1 programot, módosítsa a mozgások beállítását FINE-ről CNT-re.

ADVANCED_PROGRAM_PART1

5/5

```
1:L @PR[42:Pos1 Adv Prog Ex]
: 100mm/sec CNT100
2:L @PR[43:Pos2 Adv Prog Ex]
: 100mm/sec CNT100
3:L @PR[44:Pos3 Adv Prog Ex]
: 100mm/sec CNT100
4:L @PR[45:Pos4 Adv Prog Ex]
: 100mm/sec CNT100
```

[End]



CNT

Amikor a program hibamentesen fut, változtassa meg a mozgások sebességbeállításait 1000mm/s-ra.

Figyelem: A robot sebességét állítsa 10% vagy kevesebbre! Most nem cél a gyors végrehajtás!

A főprogramban tegyen egy extra sort a program elejére, és tegyen be egy OVERRIDE beállítási funkciót (INST -> Miscellaneous -> OVERRIDE), és állítsa 10-re (%). A program végrehajtása közben ne változtassa meg a sebességet.

```
1: !UFrame Set
2: UFRAME_NUM=1
3: !Set Override at 10%
4: OVERRIDE=10%
5: !Loop Instruction
```

Most futtassa újra a programot, és észreveheti, hogy az útvonal sokkal kerekesebb, mint korábban, és gyorsabb is, mint a FINE beállításnál.

Ez annak a ténynek köszönhető, hogy a robot pontosan végrehajtja az utat, amelyet 100% -os OVERRIDE-nál (1000 mm/s) végezne. Minél gyorsabb egy adott mozgás sebessége, a CNT annál inkább befolyásolja a mozgást.



“Wait” Funkció - Várakozás

További lehetőség a várakozási funkció hozzáadása. Ez a függvény vár egy adott időig (regiszter vagy konstans szerint), majd folytatja a következő sorban. Most adjon hozzá egy várakozási funkciót minden alprogramhoz, amelyet hívott.

ADVANCED_PROGRAM_PART1

6/6

```
1:L @PR[42:Pos1 Adv Prog Ex]
: 1000mm/sec CNT100
2:L @PR[43:Pos2 Adv Prog Ex]
: 1000mm/sec CNT100
3:L @PR[44:Pos3 Adv Prog Ex]
: 1000mm/sec CNT100
4:L @PR[45:Pos4 Adv Prog Ex]
: 1000mm/sec CNT100
5: WAIT 2.00(sec)
```

[End]

ADVANCED_PROGRAM_PART2

6/6

```
1:L @PR[42:Pos1 Adv Prog Ex]
: 1000mm/sec FINE
2:L @PR[43:Pos2 Adv Prog Ex]
: 1000mm/sec FINE
3:L @PR[44:Pos3 Adv Prog Ex]
: 1000mm/sec FINE
4:L @PR[45:Pos4 Adv Prog Ex]
: 1000mm/sec FINE
5: WAIT 2.00(sec)
```

[End]

Bővebb információk a WAIT használatáról az Operator Manual 4.8.1 fejezetben “Time-specified Wait Instruction” és a 4.8.2 fejezetben “Conditional Wait Instructions”.



Összefoglalás

Ebben a gyakorlatban megtanulhatta a programozás fejlettebb módszereit, amelyek összetettebb programokat hoznak létre, amelyek elősegítik a kifinomultabb programok és végrehajtások fejlesztését. Használni tudja a "For to" és az "If" parancsokat, és különböző alprogramokat hívhat meg a fő programjában. Tudhatja a különbséget a szerszám útvonalában, ha a FINE helyett CNT funkciót használ. Tudhatja a sebesség hatását a CNT funkcióra. Képes várakozás funkció használatára.





Gyakorlat 7

Bemenetek / Kimenetek

2. lap



Bevezetés

Ebben a gyakorlatban megismerheti a robot bemenetét és kimenetét (I/O).

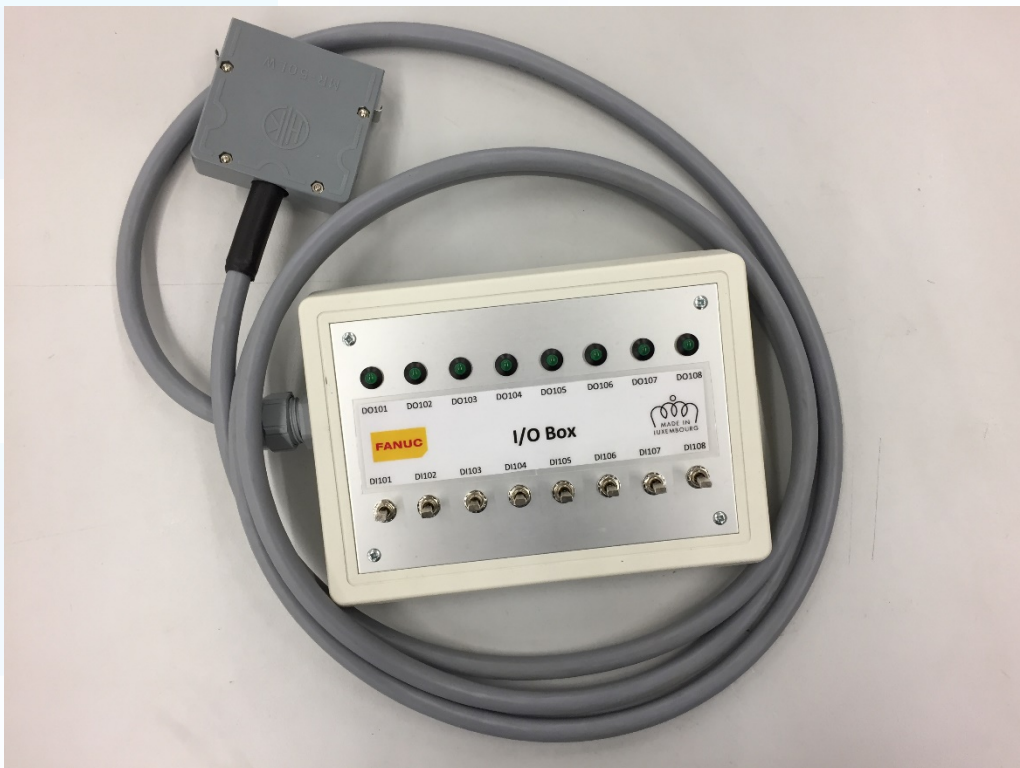
Az I/O nem más, mint egyszerű eszköz a jelekkel történő kommunikációra más eszközökkel.

Az I/O nagyon fontos, ha a robot nem önállóan működik, hanem perifériás eszközökkel, például detektorokkal, kamerákkal, más robotokkal stb.

Más eszközök szimulálására I/O dobozt terveztünk.

Ez a gyakorlat nem hajtható végre ezek nélkül a dobozok nélkül, ezért javasoljuk, hogy először építse fel ezeket a dobozokat, majd hajtsa végre ezt a gyakorlatot. A szükséges utasításokat és alkatrészlistákat a gyakorlat függeléke tartalmazza.

Ezeknek a bemeneteknek és kimeneteknek a szimulálásához a kapcsolókat bemenetként, a LED-eket pedig kimenetként használjuk a vizualizáláshoz.



Gyakorlat 7

Bemenetek / Kimenetek

Tartalom

Bevezetés	-	3
Háttér	-	5
Felszerelés	-	6
Digitális bemenetek és kimenetek	-	7
Interconnect	-	13
Csoportosított I/O	-	15
Felhasználói bemenetek és kimenetek	-	19
Párhuzamosan futó programok	-	23
Összefoglalás	-	28
Függelék	--	29



Háttér

Mint már említettük, az I / O-t arra használják, hogy a robot kommunikáljon más eszközökkel.

A robot információkat kaphat az érzékelőktől, kameráktól vagy más eszközöktől az Input segítségével, és az Output segítségével utasítást adhat motoroknak, más robotoknak stb.

Például az egyik robot kimenetet ad egy második robotnak, amint befejezte a munkáját egy adott munkadarabon, így a második robot elindíthatja saját programját ezen a részen, és tudja, hogy az első robot befejezte ez a rész, és nincs útjában a további műveleteknek.

Egy másik példa az, hogy egy közelségi kapcsoló észlelheti, hogy egy munkadarab a szállítószalag végén van, amely információ a robotnak, hogy vegye fel.

Bővebb információk az Operator Manual 3.1 fejezetben. "I/O".

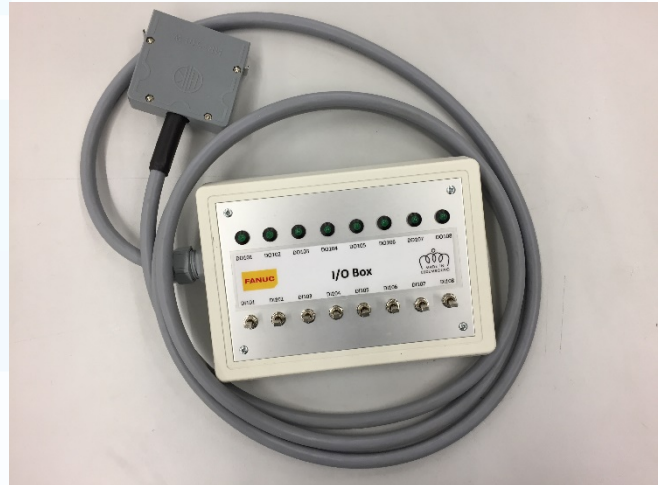
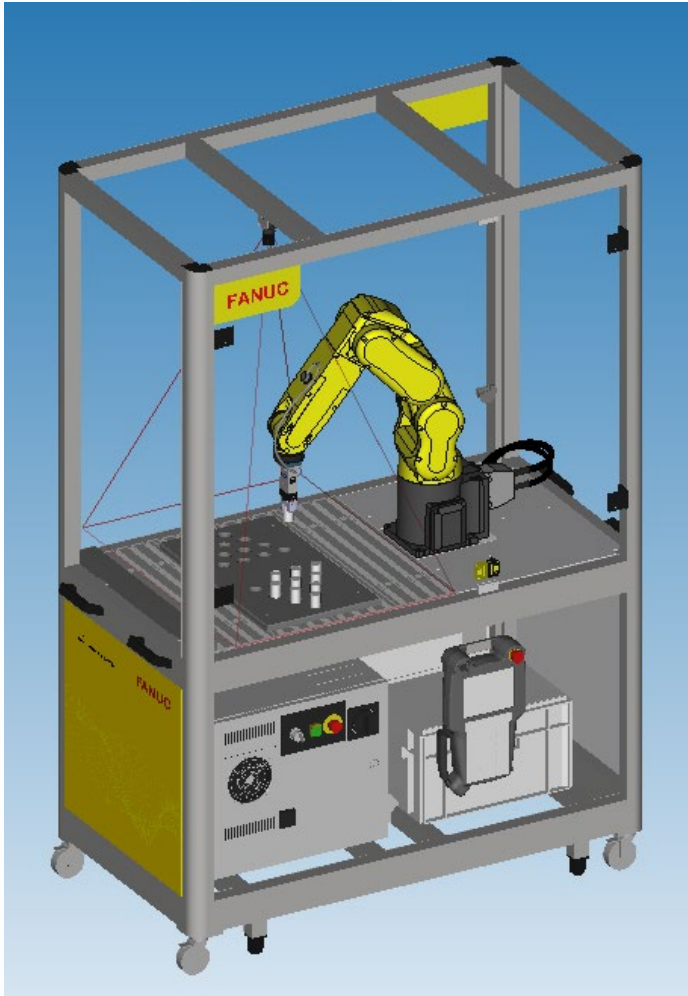


Felszerelés

A gyakorlathoz szükséges eszközök:

Oktatócella

Bemeneti/Kimeneti doboz (leírás a Függelékben)



Digitális bemenetek és kimenetek.

Először a bemeneti és kimeneti ablakra lépünk.

Nyomja meg a MENU gombot, majd válassza az I/O lehetőséget. Most az I/O menüben van. Változtassa meg a típust Digital-r az F1[Type] megnyomásával.

A bemenet és a kimenet közötti váltáshoz nyomja meg az F3 (IN/OUT) gombot.

Mivel csak 101-től vannak kimenetek, nyomja meg az ITEM gombot, és írja be a 101-et, majd ENTER, így a DO [101] –re ugrik a kurzor, ami a doboz első LED-je. Próbálja meg megváltoztatni ezt a kimenetet. (ON/OFF) Látni fogja, hogy ez nem lehetséges, mivel összefügg azzal, hogy a robot HOME pozícióban van, vagy sem.



Megváltoztathatja azonban a többi kimenetet, és figyelheti a LED-ek állapotának megváltozását.

Nézzük a bemeneteket. Ugorjon közvetlenül a DI[101] bemenethez. Mindegyik bemenetnek OFF állapotúnak kell lennie. Most kapcsoljon egy kapcsolót, és nézze meg, hogy a bemenet ON állásba kapcsol-e. Nézze meg, hogy minden kapcsoló működik-e kivétel nélkül.

Lehetőség van a bemenet „szimulálására”. Vigye a kurzort a Sim oszlopba, majd nyomja meg az F4 Simulate gombot. Most kapcsolhatja kézzel a bemenet állapotát. Ha kikapcsolja a szimulációt, akkor ismét kikapcsol (ha nincs bemenet az adott vonalon).

Bővebb információk az Operator Manual 6.4.2 fejezetben. “Simulated I/O” és 6-11 fejezet. “Simulated input/output”.



Digitális bemenetek és kimenetek

Készítsen egy új programot az új I/O használatához.

Különböző programokat hívhat meg a digitális I/O Box különböző bemeneteivel. Az alábbiakban bemutatunk egy példát. Figyeljen arra, hogy melyik programot hívja meg, mert előfordulhat, hogy az egyik program megváltoztatja a sebesség beállításokat vagy a koordináta rendszert, és ha nem állítja vissza; komoly károkat okozhat.

I_O_EX

1/16

```
1: !Set UFrame
2: UFRAME_NUM=1
3: LBL[1]
4: OVERRIDE=10%
5: IF DI[101:Switch 1]=ON,
: CALL ADVANCED_PROGRAM PART1
6: IF DI[102:Switch 2]=ON,
: CALL ADVANCED_PROGRAM PART1
7: IF DI[103:Switch 3]=ON,
: CALL ADVANCED_PROGRAM PART2
8: IF DI[104:Switch 4]=ON,
: CALL ADVANCED_PROGRAM PART2
9: IF DI[105:Switch 5]=ON,
: CALL AA_HOME
10: IF DI[106:Switch 6]=ON,
: CALL AA_HOME
11: IF DI[107:Switch 7]=ON,
: JMP LBL[2]
12: IF DI[108:Switch 8]=ON,
: JMP LBL[2]
13: WAIT .10(sec)
14: JMP LBL[1]
15: LBL[2]
[End]
```



Digitális bemenetek és kimenetek

Második gyakorlatként egy kicsit bonyolultabb programot hajtunk végre. Stoppert akarunk „építeni”.

A LED-eknek bináris formában kell megjelölniük az időt. (Ügyeljen arra, hogy a DO[101] jelzi, hogy a robot HOME pozícióban van-e, ezért ez a LED nem használható programunkhoz)[

3 kapcsolóra van szükségünk az időzítő elindításához, leállításához és visszaállításához.

Tipp:

Használjon 3 különböző programot:

- Állítsa az összes DO-t OFF-ra
- A tizedes számot bináris formává alakítja, és beállítja a kimeneteket.
- Főprogram, amely kezeli a különböző bemeneteket és beállítja az időzítőt (a Register-el).

A megoldás a következő oldalakon található.



Digitális bemenetek és kimenetek

Minden kimenet OFF-ra állítása.

```
DO_SET_OFF 9/9
1: !Setting Outputs OFF
2: DO[102:LED 2]=OFF
3: DO[103:LED 3]=OFF
4: DO[104:LED 4]=OFF
5: DO[105:LED 5]=OFF
6: DO[106:LED 6]=OFF
7: DO[107:LED 7]=OFF
8: DO[108:LED 8]=OFF
[End]
```

[INST] [EDCMD] >



Digitális bemenetek és kimenetek

Decimális szám binárisra alakítása és a kimenetek beállítása

```
DEC_TO_BIN 1/61
1: !Set Register
2: R[35:Bin Conv Reg]=R[34:Timer]
3:
4: !Reset Outputs
5: CALL DO_SET_OFF
6: !Start of Prog
7: LBL[9]
8: IF R[35:Bin Conv Reg]>=120,
9: : JMP LBL[1]
10: IF R[35:Bin Conv Reg]>=64,
11: : JMP LBL[2]
12: LBL[10]
13: IF R[35:Bin Conv Reg]>=32,
14: : JMP LBL[3]
15: LBL[11]
16: IF R[35:Bin Conv Reg]>=16,
17: : JMP LBL[4]
18: LBL[12]
19: IF R[35:Bin Conv Reg]>=8,
20: : JMP LBL[5]
21: LBL[13]
22: IF R[35:Bin Conv Reg]>=4,
23: : JMP LBL[6]
24: LBL[14]
25: IF R[35:Bin Conv Reg]>=2,
26: : JMP LBL[7]
27: LBL[15]
28: IF R[35:Bin Conv Reg]>=1,
29: : JMP LBL[8]
```

```
DEC_TO_BIN 22/61
21: JMP LBL[16]
22: !Counter max is 120 after substr
23: LBL[1]
24: R[35:Bin Conv Reg]=
25: : R[35:Bin Conv Reg]-120
26: JMP LBL[9]
27: !Substr 64 turn on LED64
28: LBL[2]
29: R[35:Bin Conv Reg]=
30: : R[35:Bin Conv Reg]-64
31: DO[108:LED 8]=ON
32: JMP LBL[10]
33: !Substr 32 turn on LED 32
34: LBL[3]
35: R[35:Bin Conv Reg]=
36: : R[35:Bin Conv Reg]-32
37: DO[107:LED 7]=ON
38: JMP LBL[11]
39: !Substr 16 turn on LED 16
40: LBL[4]
41: R[35:Bin Conv Reg]=
42: : R[35:Bin Conv Reg]-16
43: DO[106:LED 6]=ON
44: JMP LBL[12]
45: !Substr 8 turn on LED8
46: LBL[5]
47: R[35:Bin Conv Reg]=
48: : R[35:Bin Conv Reg]-8
49: DO[105:LED 5]=ON
50: JMP LBL[13]
```

```
DEC_TO_BIN 47/61
46: !Substr 4 turn on LED4
47: LBL[6]
48: R[35:Bin Conv Reg]=
49: : R[35:Bin Conv Reg]-4
50: DO[104:LED 4]=ON
51: JMP LBL[14]
52: !Substr 2 turn on LED2
53: LBL[7]
54: R[35:Bin Conv Reg]=
55: : R[35:Bin Conv Reg]-2
56: DO[103:LED 3]=ON
```

```
DEC_TO_BIN 61/61
53: R[35:Bin Conv Reg]=
54: : R[35:Bin Conv Reg]-2
55: DO[103:LED 3]=ON
56: JMP LBL[16]
57: !Substr 1 turn on LED1
58: LBL[8]
59: R[35:Bin Conv Reg]=
60: : R[35:Bin Conv Reg]-1
61: DO[102:LED 2]=ON
62: LBL[16]
[End]
```



Digitális bemenetek és kimenetek

I_O_TIMER

1/29

```

1:  !Reset the Timer
2:  R[34:Timer]=0
3:  LBL[2]
4:  !DI101 starts timer
5:  IF DI[101:Switch 1]=ON,
   :  JMP LBL[1]
6:  !DI102 stops program
7:  IF DI[102:Switch 2]=ON,
   :  JMP LBL[5]
8:  !DI103 resets timer
9:  IF DI[103:Switch 3]=ON,
   :  JMP LBL[4]
10: !Wait for not overloading Processor
11: WAIT    .10(sec)
12: !Jump to beginning
13: JMP LBL[2]
14: !Add 1sec to Reg and Output LED
15: LBL[1]
16: WAIT    1.00(sec)
    
```

I_O_TIMER

24/29

```

17: R[34:Timer]=R[34:Timer]+1
18: CALL DEC_TO_BIN
19: IF DI[102:Switch 2]=ON,
   :  JMP LBL[5]
20: IF DI[101:Switch 1]=OFF,
   :  JMP LBL[2]
21: JMP LBL[1]
22: !Reset Timer
23: LBL[4]
24: R[34:Timer]=0
25: CALL DO_SET_OFF
26: JMP LBL[2]
27: !End of Program
28: LBL[5]
[End]
    
```



Interconnect

Az "Interconnect"-el meghatározhat egy bemenetet, amely közvetlenül kimenetet kapcsol. (legyen körültekintő, ha az összekapcsolást használja).

Ebben a példában a DI[101]-et fogjuk használni a megfogó kinyitásához. Ha a DI[101] kapcsolva van, a megfogó kinyílik, és ha a DI[101] ki van kapcsolva, a megfogó bezáródik.

Az összeköttetés beállításához nyomja meg a „MENU” gombot, az „I/O” menüponton belül válassza az „Interconnect” lehetőséget. Nyomja meg az F3 „Select” gombot, majd válassza a „DI-> RO” menüpontot.

Lépjen a 7-es sorba, és a DI-hez írja be a 101. számot. Az összekapcsolás le van tiltva, ezért lépjen az Enb/Disabl oszlopra, és nyomja meg az F4 'Engedélyezés' gombot az engedélyezéshez (az F5-tel ismét kikapcsolhatja az összekapcsolást).

Busy	Step	Hold	Fault	TEST LINE 0 T2 ABORTED JOINT			100%	
Run	I/O	Prod	TCyc					
INTERCONNECT								
								7/8
No.	Enb/Disabl	INPUT			OUTPUT			
1	DISABLE	DI[	0]	->	RO[	1]		
2	DISABLE	DI[	0]	->	RO[	2]		
3	DISABLE	DI[	0]	->	RO[	3]		
4	DISABLE	DI[	0]	->	RO[	4]		
5	DISABLE	DI[	0]	->	RO[	5]		
6	DISABLE	DI[	0]	->	RO[	6]		
7	ENABLE	DI[	101]	->	RO[	7]		
8	DISABLE	DI[	0]	->	RO[	8]		



Interconnect

A DI[101] bekapcsolásakor a megfogó kinyílik, és ha a DI[101] ki van kapcsolva, a megfogó bezáródik.

Ez azonban előre nem látható eredményeket hozhat.

Ennek bemutatásához vegye ki az összes hengert az oktatási cellából, és indítsa el az AAA_DEMO programot 'AUTO' módban. Amikor a program végrehajtásra kerül, kapcsolja be és ki a DI jelet, hogy lássa, mi történik, különösen akkor, ha a robot éppen meg akar ragadni vagy elengedni egy hengert.



Csoportosított I/O (Group I/O)

Csoportosított I/O esetén két (vagy több) bemenet vagy kimenet egyidőben van használva. Az csoportosított I/O értéke egy bináris szám. (1, 2, 4, 8, 16, 32, ...).

Ebben a gyakorlatban a DO[102]-től DO[108]-ig kimeneteket használjuk.

Először is egy megosztott képernyőt szeretnénk, hogy könnyebben beállítható legyen. Nyomja meg a „SHIFT” és a „DISPLAY” gombot, majd válassza a „Double” lehetőséget. A második képernyőre állítsa be a DO-kat. Nyomja meg az F2 ‘Config’ gombot. Az elsődleges képernyőn nyomja meg a “MENU”, majd válassza az ‘I/O’ azon belül pedig válassza a ‘Group’ lehetőséget. Váltson a Group Output-ra. (GO). Nyomja meg az F2 ‘Config’ gombot.

A másodlagos képernyőn láthatja a DO[101]-től DO[120]-ig. Ez a sor információkat tartalmaz a beállításhoz.

A GO[1] beállításához, adja meg a ‘Rack’ oszlopban ‘48’, a ‘Slot’ oszlopban ‘1’ értékeket. A kezdőpont DO[102] ezért állítsa ‘START PT’ számot ‘2’-re és a ‘NUM PTS’ szám legyen ‘7’ mivel 7 kimenet lesz használva.

I/O Group Out						I/O Digital Out					
GO #	RACK	SLOT	START PT	NUM PTS		#	RANGE	RACK	SLOT	START	STAT.
1	48	1	2	7		1	DO[1- 80]	0	0	1	ACTIV
2	0	0	0	0		2	DO[81- 84]	48	1	21	ACTIV
3	0	0	0	0		3	DO[85- 100]	0	0	0	UNASG
4	0	0	0	0		4	DO[101- 120]	48	1	1	ACTIV
5	0	0	0	0		5	DO[121- 512]	0	0	0	UNASG
6	0	0	0	0							
7	0	0	0	0							
8	0	0	0	0							
9	0	0	0	0							
10	0	0	0	0							
11	0	0	0	0							
12	0	0	0	0							
13	0	0	0	0							
14	0	0	0	0							
15	0	0	0	0							
16	0	0	0	0							
17	0	0	0	0							
18	0	0	0	0							
19	0	0	0	0							
20	0	0	0	0							

Device Name : CRMA15/16

Csoportosított I/O (Group I/O)

A beállítások után a rendszer újraindítása szükséges. Használja ismét a megosztott képernyőt, hogy lássa a GO[1] értéke mely kimeneteket kapcsolja be.

I/O Group Out				I/O Digital Out			
#	SIM	VALUE	1/300	#	SIM	STATUS	101/512
GO[1]	U	42	[Group Exercise]	DO[97]	*	*	[]
GO[2]	*	*	[]	DO[98]	*	*	[]
GO[3]	*	*	[]	DO[99]	*	*	[]
GO[4]	*	*	[]	DO[100]	*	*	[]
GO[5]	*	*	[]	DO[101]	U	OFF	[HOME Signal]
GO[6]	*	*	[]	DO[102]	U	OFF	[LED 2]
GO[7]	*	*	[]	DO[103]	U	ON	[LED 3]
GO[8]	*	*	[]	DO[104]	U	OFF	[LED 4]
GO[9]	*	*	[]	DO[105]	U	ON	[LED 5]
GO[10]	*	*	[]	DO[106]	U	OFF	[LED 6]
GO[11]	*	*	[]	DO[107]	U	ON	[LED 7]
GO[12]	*	*	[]	DO[108]	U	OFF	[LED 8]
GO[13]	*	*	[]	DO[109]	U	OFF	[]
GO[14]	*	*	[]	DO[110]	U	OFF	[]
GO[15]	*	*	[]	DO[111]	U	OFF	[]
GO[16]	*	*	[]	DO[112]	U	OFF	[]
GO[17]	*	*	[]	DO[113]	U	OFF	[]
GO[18]	*	*	[]	DO[114]	U	OFF	[]
GO[19]	*	*	[]	DO[115]	U	OFF	[]
GO[20]	*	*	[]	DO[116]	U	OFF	[]
GO[21]	*	*	[]	DO[117]	U	OFF	[]

Mot szükséges egy program, amely a LED-eket működteti. Ebben a programban GO[1] lesz használva a LED-ek vezérlésére.

Bővebb információk az Operator Manual 3.1.2 fejezetben. "Group I/O" valamint a 3-2 fejezet részben. "Configuring Group I/O".

Csoportosított I/O (Group I/O)

Az alábbi program futófényt mutat be a DO[102]-től DO[108]-ig kimeneteken.

```
GROUP_I_O 7/7
1: GO[1:Group Exercise]=1
2: WAIT 1.00(sec)
3: FOR R[20:First W PR]=1 TO 6
4: GO[1:Group Exercise]=(
   : GO[1:Group Exercise]*2)
5: WAIT 1.00(sec)
6: ENDFOR
[End]
```

A második programban DO[105] lesz a központi elem, és a többi LED bekapcsolódik.

```
GROUP_I_O_2 1/11
1: GO[1:Group Exercise]=8
2: WAIT 1.00(sec)
3: GO[1:Group Exercise]=20
4: WAIT 1.00(sec)
5: GO[1:Group Exercise]=34
6: WAIT 1.00(sec)
7: GO[1:Group Exercise]=65
8: WAIT 1.00(sec)
9: GO[1:Group Exercise]=0
10: WAIT 1.00(sec)
[End]
```



Csoportosított I/O (Group I/O)

Ahogy láthatja, ez sokkal gyorsabb programozást tesz lehetővé, és a LED-ek egyszerre kapcsolódnak.

Ha minden esetben a kimeneteket külön-külön, DO[...]=ON formában programozza, a LED-ek kapcsolásában lesz egy minimális késleltetés, és kevésbé néz ki jól.

A Csoportosított Bemeneteket hasonlóképpen használhatja a bemeneti kapcsolókhoz. A GI[...] a GO[...] -hoz hasonlóan működik, fordított irányban.



Felhasználói Bemenetek és Kimenetek

Most elkezdjük az UOP doboz (felhasználói kezelőpanel) használatát. Ennek egy olyan panelt kell utánoznia, mint amit az iparban használnak, az alapvető parancsok, például a start és a program kiválasztása bevitelére.

A program kiválasztása a PNS kapcsolókon keresztül történik. A meghívni kívánt programot PNS-nek kell nevezni, amelyet egy négyjegyű szám követ (pl. PNS0011). A PNS bináris számként olvasható, ha egyetlen program sincs várakozásban, és megnyomja a start gombot.

Bővebb információk az Operator Manual 3.3 fejezetben. "Peripheral I/O" valamint a 3-5 fejezetben. "Assigning Peripheral I/O".

Emlékezzen rá, hogy ebben az esetben az UOP jelek egyszerű hozzárendelésben kell legyenek. (Simple CRMA16 kiosztás).



Felhasználói Bemenetek és Kimenetek

Elsőként engedélyezni szükséges az UOP jeleket, hogy a robot fogadja ezeket a parancsokat. Nyomja meg a MENU gombot, és a menu második lapján válassza a System menüpontot, majd azon belül a Config elemet. Engedélyezze az UI jeleket TRUE-val, és a “Start for continue only” legyen False.

```
6 Restore selected program: TRUE
7 Enable UI signals: TRUE
8 START for CONTINUE only: FALSE
9 CSTOPI for ABORT: FALSE
```

Remote/Local Setup lehetőséget állítsa Remote-ra.

```
41 Hand broken : <*GROUPS*>
42 Remote/Local setup: Remote
43 External I/O(ON:Remote):DI [ 0]
```

Az “UOP auto assignment” lehetőséget állítsa; Simple(CRMA16)

```
43 External I/O(ON:Remote):DI [ 0]
44 UOP auto assignment: Simple(CRMA16)
45 Multi Program Selection: FALSE
```

A rendszer újraindítása szükséges a beállítások után.

Bővebb információk az Operator Manual 3.15 fejezetben. “System Config Menu” valamint a 3-33 fejezetrészben. “Setting the System”.



Felhasználói Bemenetek és Kimenetek

Egy egyszerű példával kezdjük a program indítását az UOP-ból.

Válasszon ki egy programot a Teach Pendant segítségével. Kapcsolja a vezérlőt Auto módba, és kapcsolja ki a Teach Pendant-ot. Most engedélyezze az UOP-t (ENBL), és nyomja meg az UOP Hold gombját (a Hold jel alacsonyan aktív, ezért általában magasnak kell lennie). Nyomja meg a Start gombot. Figyelje meg, hogy a program csak a Start gomb megnyomásakor indul. Most elindította első programját egy UOP-ból.

Egy lépéssel tovább menve; képesek vagyunk kiválasztani egy programot az UOP-ból. Ehhez szükségünk van arra, hogy a programokat PNS-nek és egy 4 számjegyű számnak hívjuk. Másoljon le egy programot, és nevezze el a másolatot PNS0001-nek. Most a START kapcsoló kapcsolása előtt kapcsolja be a PNS választókapcsolót (esetünkben PNS1). A rendszernek a PNS0001 programot el kell indítani.

Bővebb információk az Operator Manual 3.8.2 fejezetben. "Program Number Selection (PNS)", valamint a 3-9 fejezetben. "Setting the PNS function" és a 6-15 fejezetben. "Automatic operation by program number selection".



Felhasználói Bemenetek és Kimenetek

Amíg a robot működik, figyelje meg a világító LED-eket.
Első LED, az ENBL kigyullad, ha a robot rendelkezésre áll.
Az utolsó LED, BUSY világít, ha egy program fut.
Aztán ott van a két LED középen. A FAULT LED, és a BATALM.

A BATALM nem fordulhat elő a robot működtetése közben, ezért nem fogunk sokat részletezni, további információkért olvassa el a kézikönyvet.

A FAULT LED azonban előfordulhat, és szeretnénk ezt szimulálni. Ennek érdekében indítson el egy programot. Amíg a program fut, nyissa ki a cella ajtaját. A robot megáll és a FAULT LED bekapcsol. Ennek oka a védőkerítés áramkörének megszakadása. Csukja be ismét az ajtót, nyomja meg a RESET gombot (az UOP-on), és folytassa a programot (az UOP START gombjával).



Párhuzamosan futó programok

Végül szeretnénk megmutatni, hogy több programot is futtathat párhuzamosan, nem csak egy-egy programot. Ez különösen praktikus, ha mozogni akar a robottal, és külön, a külső berendezéseket is irányítani.

Ennek bemutatásához egy, a robotot mozgó programot (négyzet rajzolása 10-szer) egyszerre fogjuk futtani az időzítő programmal.

Ez megint nem nagyon fontos és nem is praktikus az esetünkben, de vannak olyan helyzetek, amikor ez nagyon praktikus lehet.

Mivel később később újra fel akarjuk használni a programot a kiegészítések nélkül, készítsen egy másolatot arról a programról, amelyet `Advanced_Prog_Ex_IO`-nak hív. Folytassa a munkát ezen a programon.

Bővebb információk az `Operator Manual 4.16.1` fejezetben. "Program Execution Instruction".



Párhuzamosan futó programok

Nyissa meg az Advanced_Prog_Ex program-ot.

Adjon hozzá egy új sort az Override/sebességbeállítás után.

Adjon egy Run utasítást (Instr menu, második lap, Multiple Control menüpont) ami legyen: Run I_O_Timer.

Próbálja futtatni a programot.

A program nem fog elindulni, hibaüzenetet kap:

“PROG-040 Already locked by other task”.

Ennek feloldásához meg kell változtatnia a Group Mask beállításait a programnak, amelyet a RUN utasítás után használt. (I_O_Timer, DO_Set_OFF, Register_To_Binary) from 1 to *.

Ezt megteheti a programon állva az F2 ‘Detail’ megnyomásával.

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

Bővebb információk az Operator Manual 4.1.4 fejezetben.
“Group Mask”.



Párhuzamosan futó programok

Most a programnak már el kell indulnia, és képesnek kell lennie mindkét program párhuzamos futtatására, vagyis a robot mozog, miközben Ön „játszhat” a stopperrel.

Amint említettük, Ön képes a stopperrel „játszani”, ami azt jelenti, hogy semmi haszna nincs a programunkban. Mivel a robot nem játékra épült, most úgy fogjuk megváltoztatni a programot, hogy az egyik program kölcsönhatásba lépjen a másikkal, így a programok függetlenül futnak, de végül is valamilyen módon kapcsolódnak egymáshoz.

Már bemutattuk a várakozási utasítást, ami egy adott ideig várakozott. Most azt a várakozási jelet szeretnénk megváltoztatni, hogy ne pontosan 2 másodpercet várjon, hanem hogy addig várjon, amíg el nem éri egy, a második programban használt speciális regiszter a várt értéket. Ennek során a mozgási program mindig arra vár, hogy a nem mozgási programtól adatot kapjon.



Párhuzamosan futó programok

Az alábbiakban beírjuk a változásokat a programba.

```
11/20
10: !If Commands
11: IF R[33:If Value Ex]=0,
: JMP LBL[1]
12: !First part of prog
13: CALL ADVANCED PROGRAM PART1
14: !Wait on timer
15: R[36:Con Tm + Mtn]=
: R[32:ForTo Value Ex]*15
16: WAIT R[34:Timer]>=
: R[36:Con Tm + Mtn]
17: LBL[2]
```

```
21/29
20: JMP LBL[3]
21: !Second part of prog
22: LBL[1]
23: CALL ADVANCED PROGRAM PART2
24: !Wait on Timer
25: R[36:Con Tm + Mtn]=
: R[32:ForTo Value Ex]*15
26: WAIT R[34:Timer]>=
: R[36:Con Tm + Mtn]
27: !Jump back into loop
28: JMP LBL[2]
```



Párhuzamosan futó programok

Futtassa a programot, és figyelje meg, hogy a program leáll-e, ha a számláló nem a 15-ös többszöröse. Megpróbálhatja megállítani a számlálót, vagy újraindíthatja azt. A programnak várnia kell a megfelelő időpontra, hogy folytassa a következő program lépéssel. A mozgás befejezése után a program tovább fog futni, mert az Időzítő programnak nincs megállása. Ennek megváltoztatásához használjon egy regisztert, amelyet az időzítő program ellenőriz, és a mozgásprogram befejeztével fejezze be azt.



Összefoglalás

Ebben a gyakorlatban bemutattuk az Input és Output alapelveit.

Megismerte a különböző opciókat a bemenetekkel és a kimenetekkel, és képesnek kell lennie egy olyan program létrehozására, amely egyszerre két különböző programot hajt végre (csak az egyik egy mozgásprogram (!)).

Be kell tudnia állítani az UOP kezelői felületet, és ezen keresztül futtatni a robot programjait.

Képesnek kell lennie perifériás eszközök csatlakoztatására a robothoz, és be kell állítania a bemeneteket és kimeneteket arra, hogy ezeket az eszközöket robot működése közben használják.



Függelék

Ahhoz, hogy ezeket a gyakorlatokat elvégezhesse, szüksége lesz 2 be-, kimeneti dobozra.

Minden alkatrész az RS-Components weboldáról beszerezhető.
(www.rs-components.com)

Az alábbi alkatrészekre lesz szüksége:

- 8x 8mm zöld LED 24V (ref. RS 206867)
- 2x Aluminium ferde műszerdoboz 190x138x47 (ref. RS 505820)
- 2x 8mm piros LED 24V (ref. RS 206845)
- 2x 8mm sárga LED 24V (ref. RS 206463)
- 16x On/Off (On) kapcsoló 4A 30V (ref. RS 1035502)
- 2x Honda csatlakozó MR-50M (Fanuc ref. 816052)
- 2x Honda csatlakozó ház MR-50M (Fanuc ref. 836896)
- 6m hajlékony vezeték (min. 18 ér, 3m dobozonként)



Függelék

Elsőként fúrja ki a furatokat a műszerdobozon a LED-ek és a kapcsolók számára.

Ha a javasolt műszerdobozt használja, a következő lapon megtalálja a fúrási tervet.

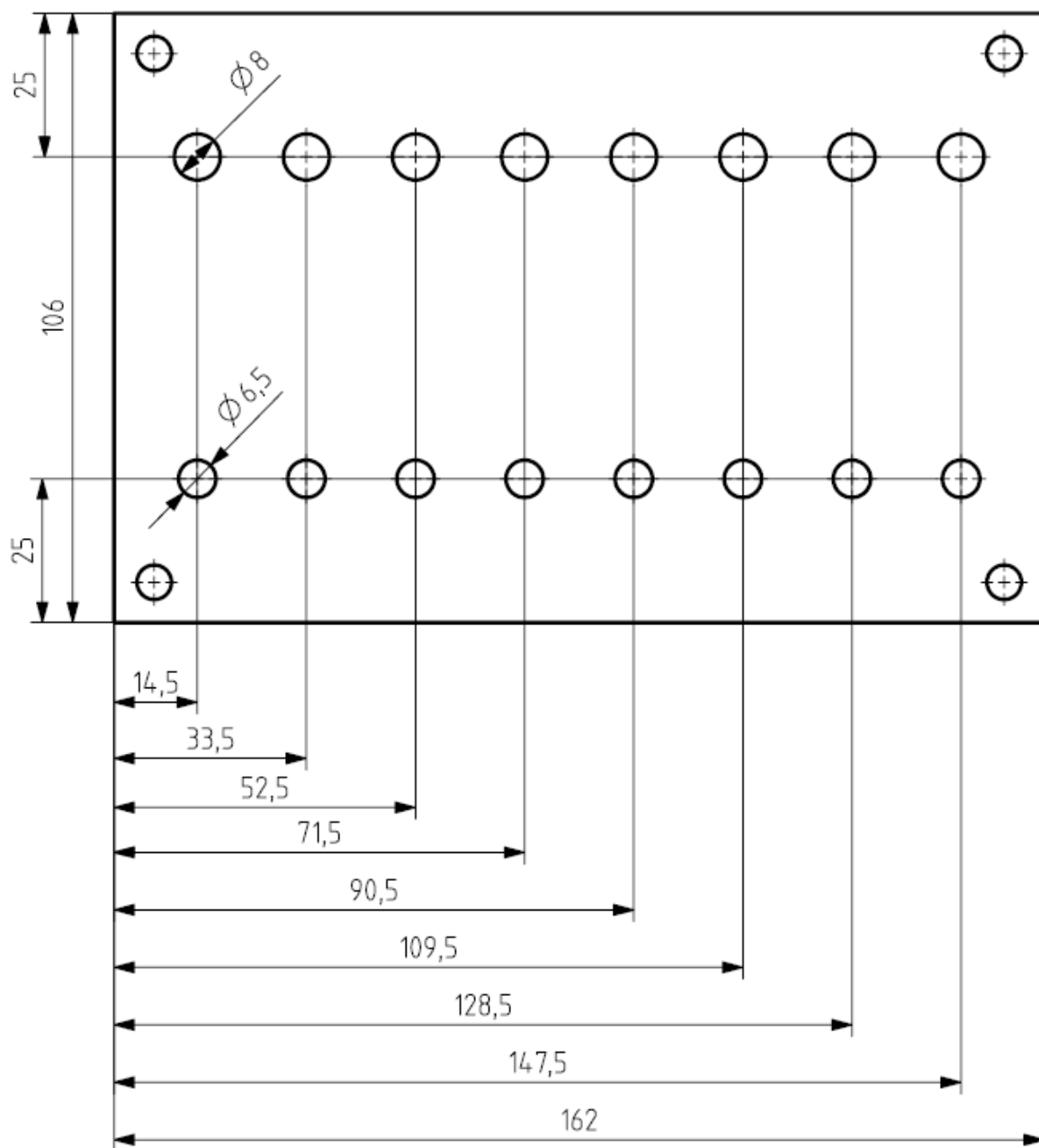
Az első műszerdobozba építsen be 8 kapcsolót és 8 zöld LED-et. (A LED-ek színe bármilyen lehet)

A második műszerdobozba építsen 8 kapcsolót, és mindössze 4 LED-et. Javasoljuk, hogy 2 piros és 2 sárga LED-et építsen be.

Ha a javasolt műszerdobozt használja, a következő lapon megtalálja a fúrási tervet.



Függelék

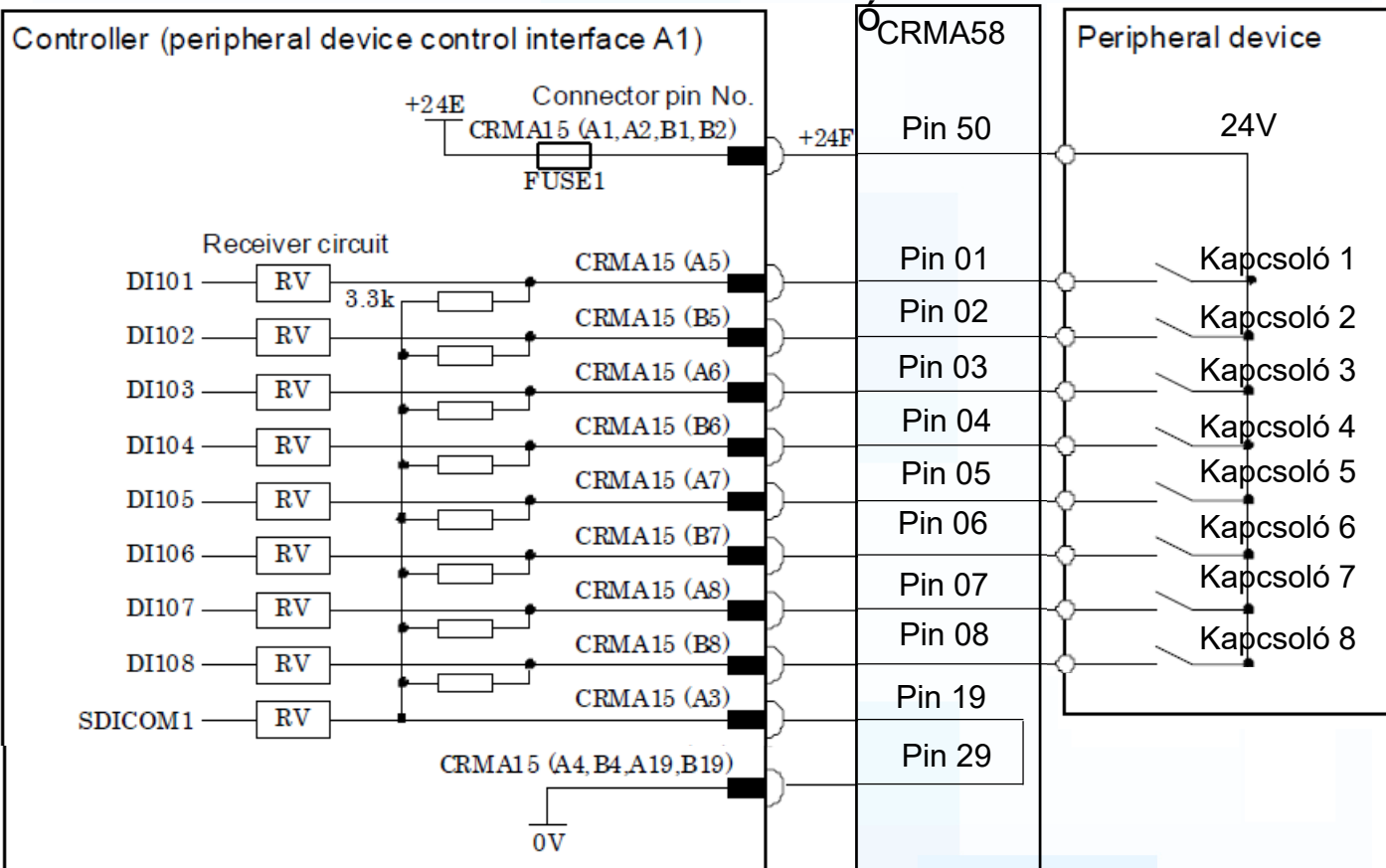


Függelék

Csatlakozás a vezérlőszekrényben

Honda csatlakoz

I/O doboz



A 0V-ot az SDICOM1 számára csatlakoztassa direktben a Honda csatlakozón belül. (PIN19 – PIN29)

Csatlakoztassa a 24V-ot az ON pontjára a kapcsolóknak.

Csatlakoztassa az OFF pontjait a kapcsolóknak a kábelhez. Legyen benne biztos, hogy minden kábel elér a kapcsolóig.

A teljes tervet az R-30iB Mate Controller MAINTENANCE MANUAL 4.3 fejezetben találja. "INTERFACE FOR PERIPHERAL DEVICES".

Csatlakozás a vezérlőszekrényben

ÖCRMA58
Pin 49
Pin 31
Pin 33
Pin 34
Pin 35
Pin 36
Pin 37
Pin 38
Pin 39
Pin 40
Pin 18

Peripheral device

LED1

LED2

LED3

LED4

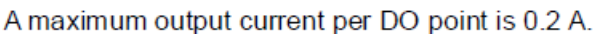
LED5

LED6

LED7

LED8

0V



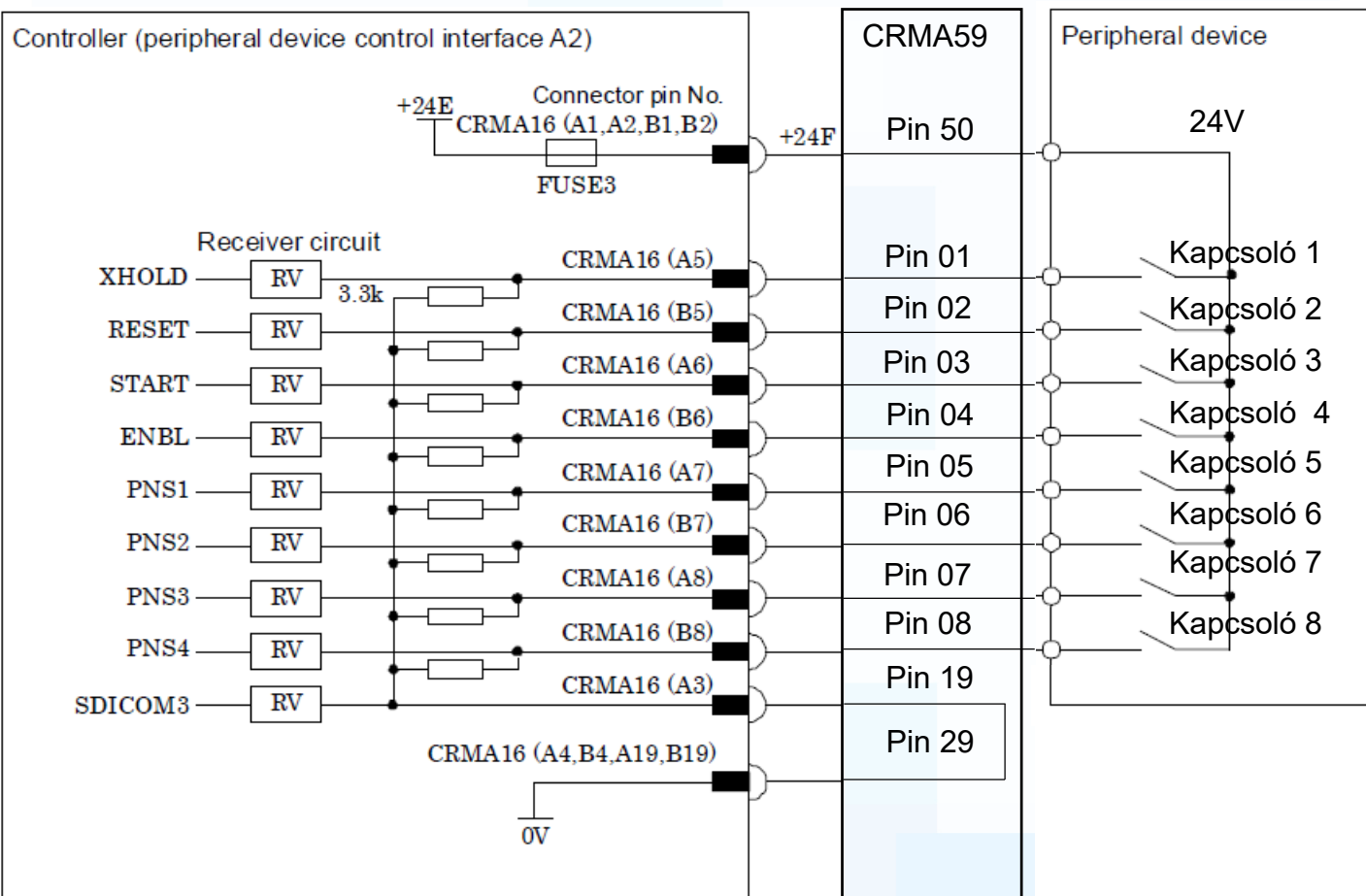
Page 33

Függelék

Csatlakozás a vezérlőszekrényben

Honda
csatlakozó

UOP doboz



A második doboz bekötése teljesen megegyezik az első dobozával. Javasoljuk, hogy használjon ugyanolyan bekötést, hogy elkerülje a későbbi félreérthetőséget.

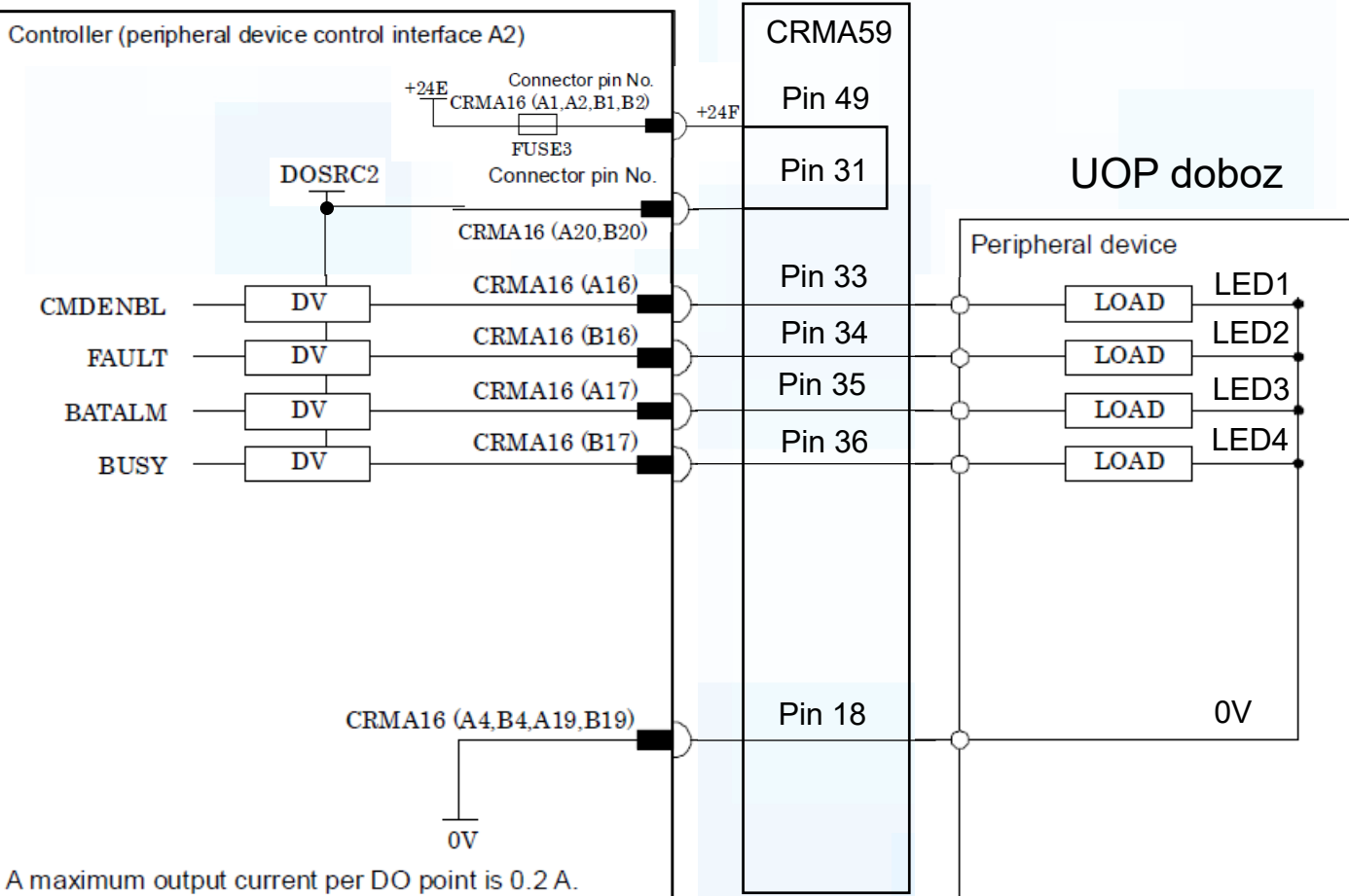
Ez a későbbi esetleges csatlakozócserét is megkönnyíti.

Függelék

Csatlakozás a vezérlőszekrényben

Honda csatlakozó

UOP doboz



A második dobozban is csatlakoztassa a LED-eket. Ebben az esetben itt ezt csak 4 LED-el kell megtennie. Ne felejtse el csatlakoztatni a 24V-ot a DOSRC2 lábra.

Felhasználó Bemenetek és Kimenetek

R30iB Mate Controller CRMA15/CRMA16									
CRMA 58					CRMA 59				
DI Bemenet	Jel	Ér színe	Pin #	Bemenet	UOP In	Jel	Ér színe	Pin #	Bemenet
	S1	Kék	1	DI 101		S1	Kék	1	XHOLD
	S2	Piros	2	DI 102		S2	Piros	2	RESET
	S3	Lila	3	DI 103		S3	Lila	3	START
	S4	Szürke	4	DI 104		S4	Szürke	4	ENBL
	S5	Rózsaszín	5	DI 105		S5	Rózsaszín	5	PNS1
	S6	Zöld	6	DI 106		S6	Zöld	6	PNS2
	S7	Fehér	7	DI 107		S7	Fehér	7	PNS3
	S8	Sárga	8	DI 108		S8	Sárga	8	PNS4
	/	Barna	50	24V		/	Barna	50	24V
DO Kimenet	Jel	Ér színe	Pin #	Kimenet	UOP Out	Jel	Ér színe	Pin #	Kimenet
	LED1	Szürke/Rózsaszín	33	DO 101		LED1	Szürke/Rózsaszín	33	CMDENBL
	LED2	Piros/Kék	34	DO 102		LED2	Piros/Kék	34	FAULT
	LED3	Barna/Zöld	35	DO 103		LED3	Barna/Zöld	35	BATALM
	LED4	Fehér/Sárga	36	DO 104		LED4	Fehér/Sárga	36	BUSY
	LED5	Sárga/Barna	37	DO 105		/	Fehér/Szürke	18	0V
	LED6	Fehér/Zöld	38	DO 106					
	LED7	Fehér/Rózsaszín	39	DO 107					
	LED8	Rózsaszín/Barna	40	DO 108					
	/	Fehér/Szürke	18	0V					
Belső csatlakozás	Jel	Pin #	Jel	Pin #	Belső csatlakozás	Jel	Pin #	Jel	Pin #
	SDICOM1	19	0V	29		SDICOM3	19	0V	29
	Jel	Pin #	Jel	Pin #		Jel	Pin #	Jel	Pin #
	DOSRC1	31	24V	49		DOSRC2	31	24V	49



Felhasználó Bemenetek és Kimenetek

Az I/O dobozok robotvezérlőhöz való csatlakoztatásához először meg kell tennie, hogy lekapcsolja a vezérlő tápellátását és kihúzza a cellát.

A cellának addig kell így maradnia, amíg a vezérlő ajtaja ismét be nem záródik.

Nyissa ki a vezérlő ajtaját. Fűzze be a kábeleket a vezérlő jobb oldalán. A Honda csatlakozók a vezérlőben a jobb oldalon vannak. Győződjön meg arról, hogy a megfelelő csatlakozót a megfelelő csatlakozóhoz csatlakoztatta (ott szerepel a CRMA58 és CRMA59 jelölés).

Győződjön meg arról, hogy a CRMA16 és 15 is csatlakozik. Ha megtalálja őket, kövesse a csatlakozók kábeleit, vagy egyszerűen keresse meg az ajtón a CRMA16 és 15 elemeket. Az ajtó bal alsó oldalán vannak.

Bekapcsolás előtt zárja vissza a vezérlőszekrényt!





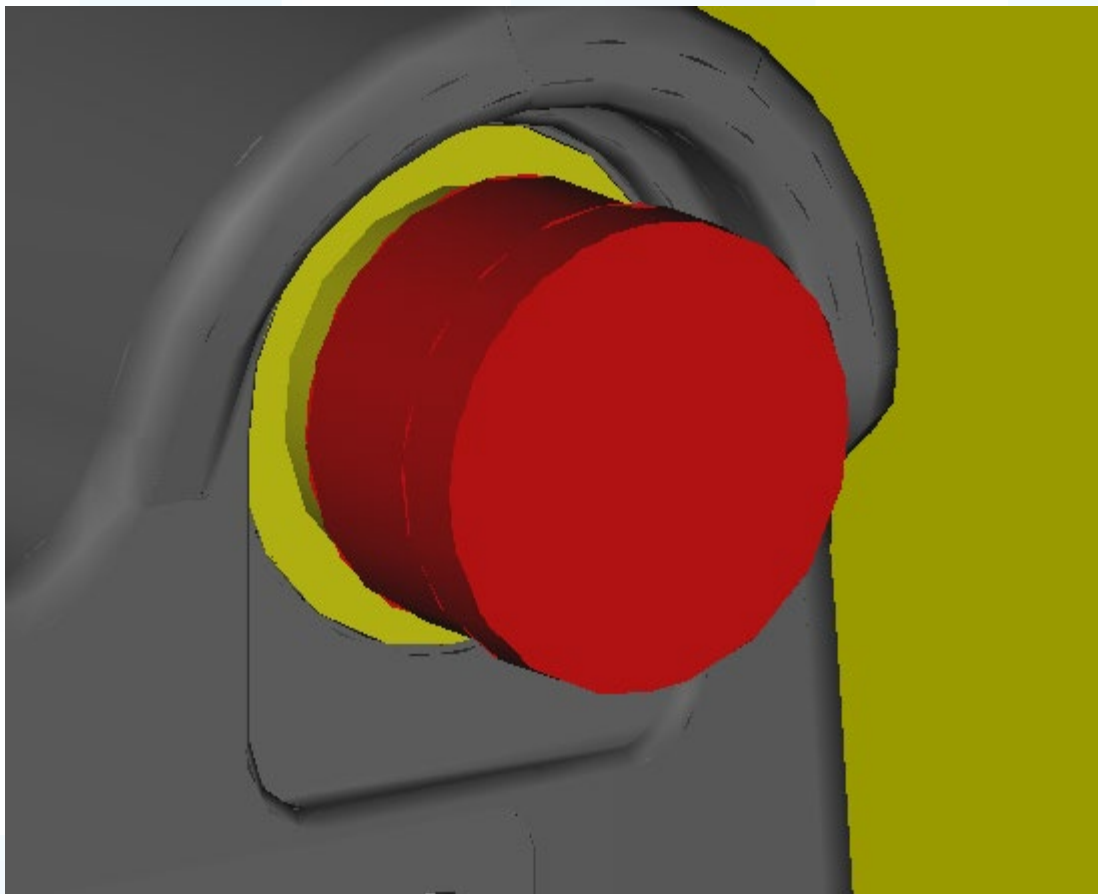
Gyakorlat 8

Megállás típusok



Bevezető

Ez a gyakorlat egyfajta áttekintést nyújt a FANUC robotjain elérhető különböző megállásokról, azokról az okokról, amelyek miatt a különböző megállásokat különböző esetekben alkalmazzuk, és a megállások mögött lévő normákat. Ennek a gyakorlatnak az ideje alatt meg kell tanulnia a megállások különbségeit és azt, hogy melyik megállás mikor következik be, és meg kell értenie, miért használnak egy bizonyos megállást.



Gyakorlat 8

Megállás típusok

Tartalom

Bevezető	-	3
Háttér	-	5
Eszköz	-	6
Különböző Stop kategóriák	-	7
Vész Stop (E-Stop)	-	8
Deadman kapcsoló elengedése	-	9
Ajtónyitás	-	10
Hold jel	-	11
Összegzés	-	12



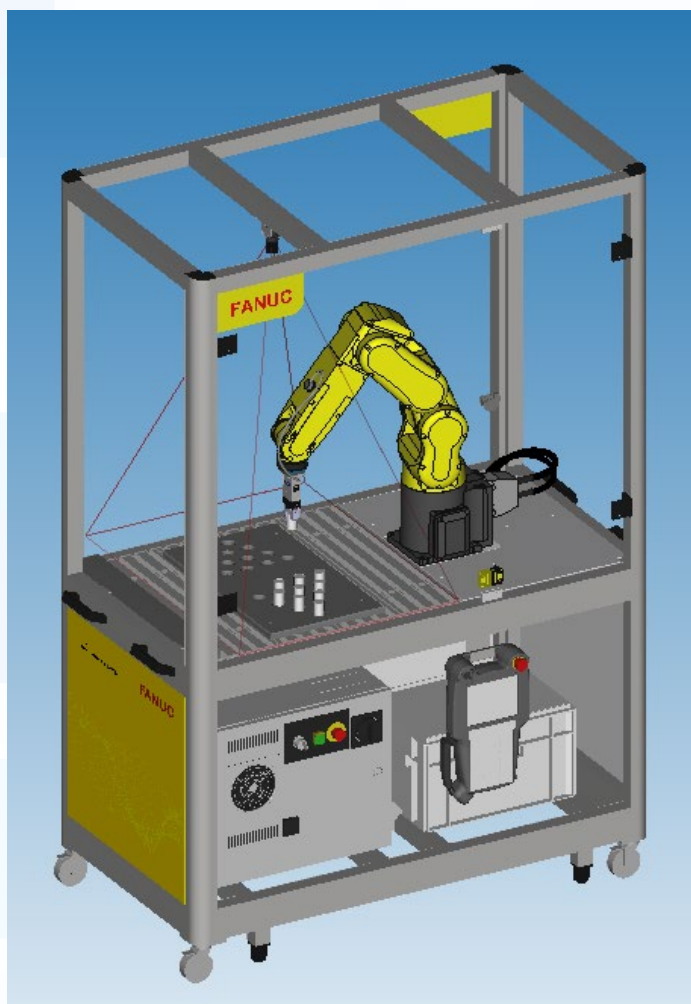
Háttér

A megállások az egyik legkézenfekvőbb funkciók, amelyet egy robotnak vagy bármilyen gépnek el kell tudnia végeznie. De a különböző megállások megértése és hogy miért használunk különböző megállásokat, már nem annyira nyilvánvaló. A munkáját befejező robotnak le kell állnia. Meg kell állítani azt a robotot, amely olyan mozgást végez, amelyet nem kéne megtennie. Sok különböző eset van, ahol megállásokra van szükségünk. Áttekintést adunk a különböző megállásokról.



Eszköz

A gyakorlathoz szükséges:
Oktatócella



Stop kategóriák

3 különböző megállás típus létezik (IEC 60204-1 szabvány):

- Kategória 2

Kategória 2 megállás olyan vezérelt megállás, ahol a megállás után feszültség marad a motorokon.

- Kategória 1

Kategória 1 megállás olyan vezérelt megállás, ahol a megállás után nem marad feszültség a motorokon.

- Kategória 0

Azonnali megállás, ahol a feszültség eltávolításra kerül a motorokról. A megállás útvonala nem kontrolált.

Látja, hogy a különböző kategóriák kevésbé szigorúak. Vegye figyelembe, hogy minél súlyosabb a megállási kategória, annál károsabb a robot számára, és annál biztonságosabb az emberi vagy a külső felszerelés számára. Az 1. vagy 0. kategóriás megállásból való kilépéshez távolítsa el a leállítási okát (engedje fel az E-stop gombot, zárja le a kerítést, stb ...), nyomja meg a reset gombot. Ezután folytathatja vagy újraindíthatja a programot (ez a 0. kategóriás leállítási után is ajánlott).

Bővebb információk a Stop kategóriákról az Operator Manual 7 fejezetben: "Stop Type of Robot".



Vészleállítás (E-Stop)

A vészleállást vészhelyzetben használják. Ezt a megállást akkor használják, ha emberek vannak veszélyben, vagy ha a robot ütközni készül más berendezésekkel, tárgyakkal vagy más robotokkal. Elsősorban a sérülések vagy sérülések elkerülésére szolgál, mint utolsó lehetőség, ezért 0 kategóriás megállás. Ez a legbrutálisabb megállás a robot számára, de a legbiztonságosabb is, mivel azonnal leállítja a robotot. Most próbálja meg használni a vészleállító gombot, amíg a robot programot hajt végre. Az Advanced_Prog_Ex programnak alkalmas erre a célra. Indítsa el a programot, majd mozgás közben nyomja meg a Vészleállítás gombot. Figyelje meg, hogyan áll meg a robot.

FONTOS

Vészleálláskor a robot elhagyhatja eredeti pályát. A visszaállítás és az indítás után a robot először visszalép a régi pályára, majd folytatja.

Bővebb információk a vészmegállásról az Operator Manual 6-1 fejezetben. "Emergency Stop and recovery".



Deadman elengedése

A robot mozgatásához vagy egy program végrehajtásához (tanítási módban) a Deadman kapcsolót közbenső helyzetben kell tartani. Ez azért történik, hogy megakadályozza a kezelő robot általi sérülését. Abban az esetben, ha a robot gyorsan megközelíti a kezelőt, a kezelő elengedheti a Deadman kapcsolót, és a robot leáll, vagy a kezelő pánikba eshet, és erősebben megragadhatja a Deadman kapcsolót, így a robot szintén megáll. A fenti leírásból megértheti, hogy ennek is 0 kategóriás megállásnak kell lennie, mivel a kezelő biztonsága az első helyen áll. Próbaképpen futtassa és állítsa le az Advanced_Prog_Ex programot T1 tanítási módban, miközben félállásban tartja a Deadman kapcsolót, majd engedje el a kapcsolót, és figyelje meg, hogy ugyanazt a viselkedést látja-e, mint az E-Stop esetén.

Ismét figyeljünk meg a program folytatásakor, (mivel a robot valószínűleg kissé a pálya mellett van), hogy indulás után a robot visszatér a korábbi pályára, és lehet, hogy váratlan mozgást fog végrehajtani.



Ajtónyitás

Normális esetben a robotot valamilyen védőkerítés veszi körül. Ez lehet fizikai kerítés, de fénysorompó is, ahol a robot észleli, ha valaki belép a munketerületére. Esetünkben a cella ajtaja. Akárhogy is, amikor a kaput (vagy ajtót) kinyitják, a robotot le kell állítani. De ebben az esetben a robotnak van egy kis ideje megállni. A szabvány szerint a robotnak 1 másodperc áll rendelkezésére, hogy megálljon, mivel a kezelőnek szüksége van egy kis időre, hogy a robothoz érjen. Ez az oka annak, hogy a cella kinyitása az 1. kategóriába tartozó megállásokat használja, mivel a működési zónába való belépés nem jelent közvetlen veszélyt az illető számára. Próbálkozzon újra a program futtatásával, és ezúttal működés közben nyissa ki az ajtót, és próbálja meg látni a különbséget a két 0. kategóriás megállás és az 1. kategóriás megállás között. Mivel ez egy vezérelt megállás, a program könnyen tovább futtatható, mivel az még mindig a mozgási pályán van.



Hold

A Hold segítségével a robot ellenőrzött módon állítható le annak érdekében, hogy valamit megváltoztassunk a robot beállításában vagy a robot konfigurációjában. Ha egy kezelő le akarja állítani a robotot, - és nincs veszélyhelyzet, - akkor mindig a Hold funkciót kell használnia. Ha megnyomja a Hold gombot, akkor a robot vezérelt megállást végez, hibaüzenet nem jelenik meg, és a robot újraindítható abból a helyzetből, amelyben megállt. A Hold egy 2. kategóriás megállás. Próbálja ki ezt a megállást a mintaprogrammal, és folytassa (indítsa el) újra a programot.

Bővebb információk a Hold használatáról az Operator Manual 6-2 fejezetben. "Hold and recovery".



Összefoglalás

Ebben a gyakorlatban megtanulhatta a rendelkezésre álló különböző megállási kategóriákat, és a különböző eseteket, hogy melyik típusú megállást mikor alkalmazzák. Meg kell tudnia szüntetni a leállítást okát, visszaállítani a robotot és újraindítani a programot.





Gyakorlat 9

DCS (Dual Check Safety) Biztonsági Zóna

2. lap

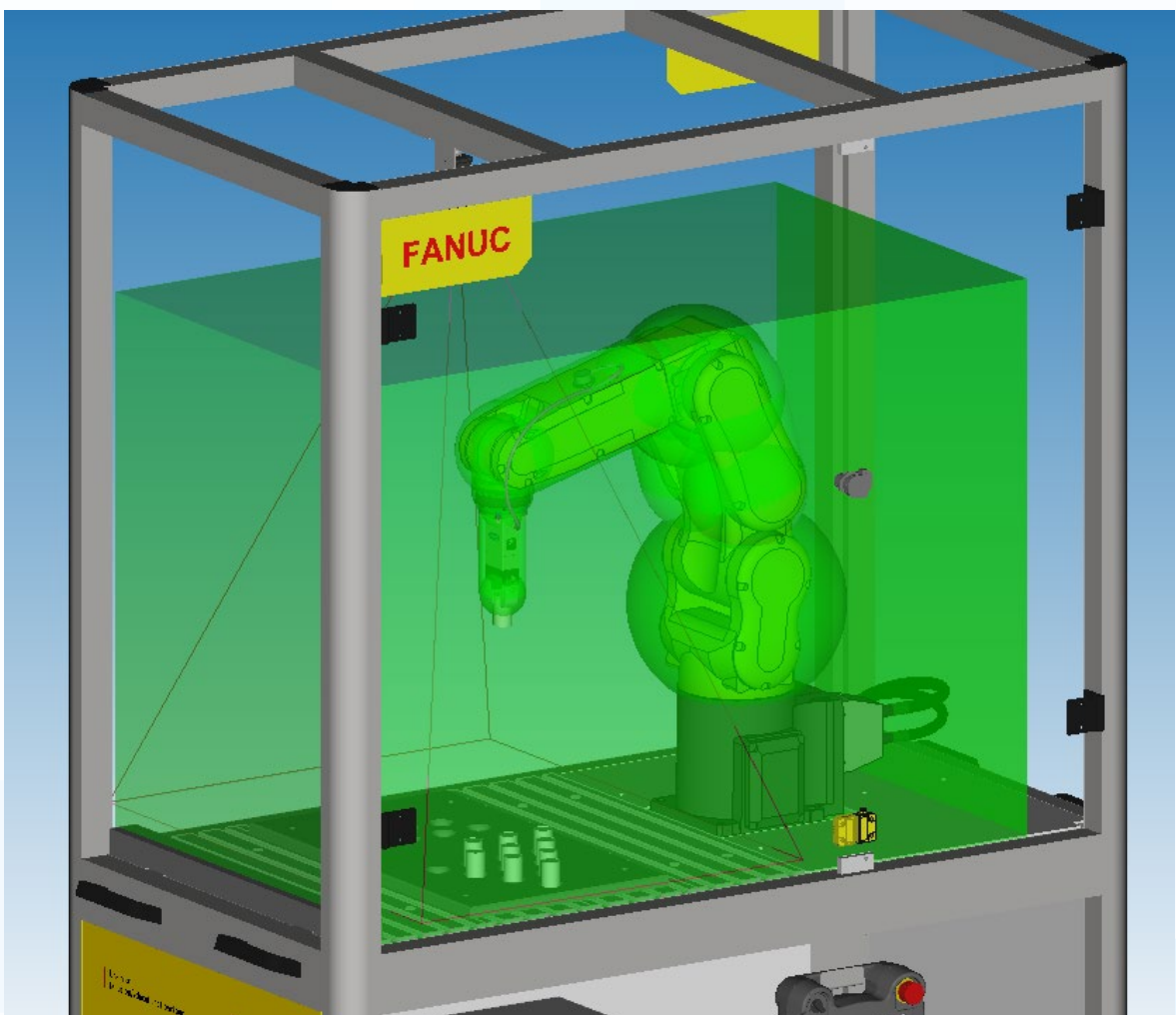


Bevezető

Ebben a gyakorlatban megtanulhatja a DCS biztonsági zónák alapjait, valamint azt, hogyan tudja a saját biztonsági zónáját beállítani.

A DCS biztonsági zóna tulajdonképpen egy virtuális doboz. Ön engedélyezheti a robot számára a zónában történő mozgást, vagy megtilthatja a zónába belépést.

Röviden, a DCS zónák behatárolhatják a robot mozgását, és alakjuk nem csak négyzetes lehet.



Gyakorlat 9

DCS Biztonsági Zóna

Tartalom

Bevezető	-	3
Háttér	-	5
Eszköz	-	6
DCS megjelenítése	-	7
DCS beállítása	-	13
Biztonsági zóna méretezése	-	18
User Model beállítása	-	21
Összefoglalás	-	26



Háttér

A DCS biztonságos zónákat leginkább akkor használják az iparban, amikor egy robot olyan területen mozog, ahol más robotok is működnek, vagy akár emberek is bekerülhetnek a robot hatósugarába.

Esetünkben a DCS biztonságos zóna gondoskodik arról, hogy a robot ne tudjon átjutni a cella falain. Ahogy a neve is mutatja, a DCS Biztonsági Zóna megmondja a robotnak, hogy melyik zónában biztonságos a működtetése, és leállítja, ha át akar jutni a zónán. Egy zóna sok különböző, egyszerű zónából állítható össze, így összetettebb zónákat hozhat létre.

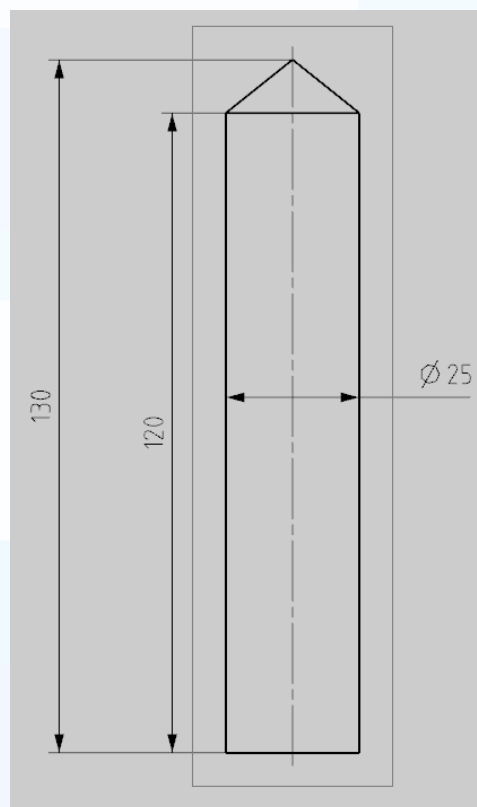
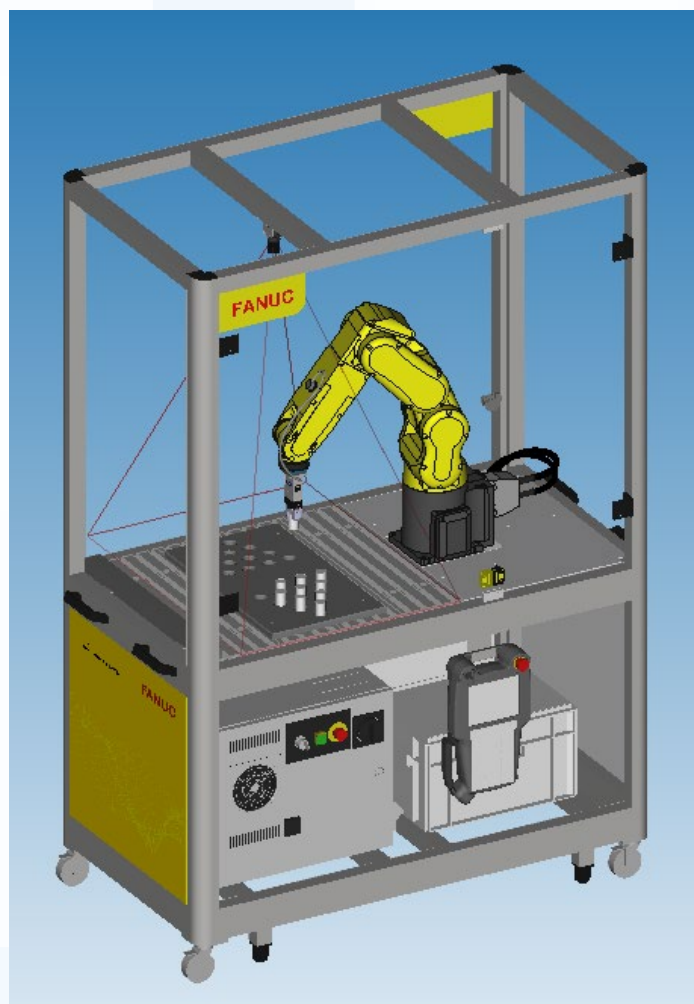


Eszköz

Ehhez a gyakorlathoz szükséges:

Oktatócella

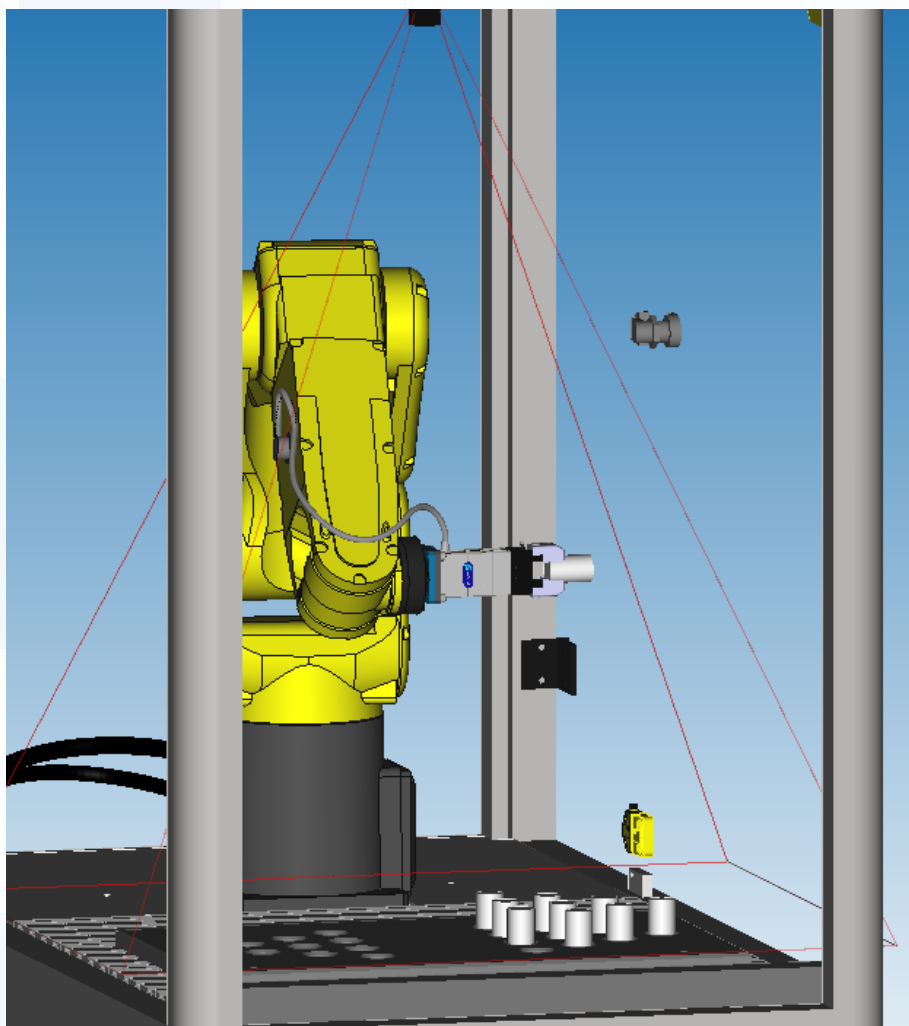
Az ábra szerinti csap – ez esetben a megfogóban



DCS megjelenítése

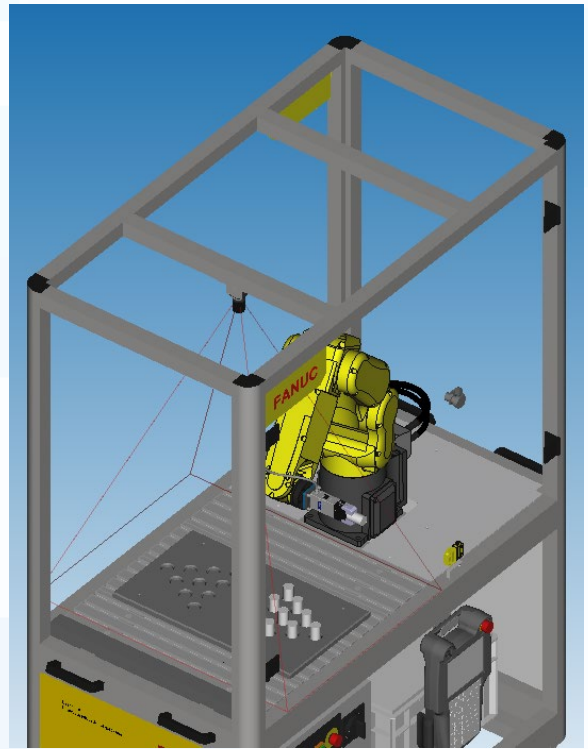
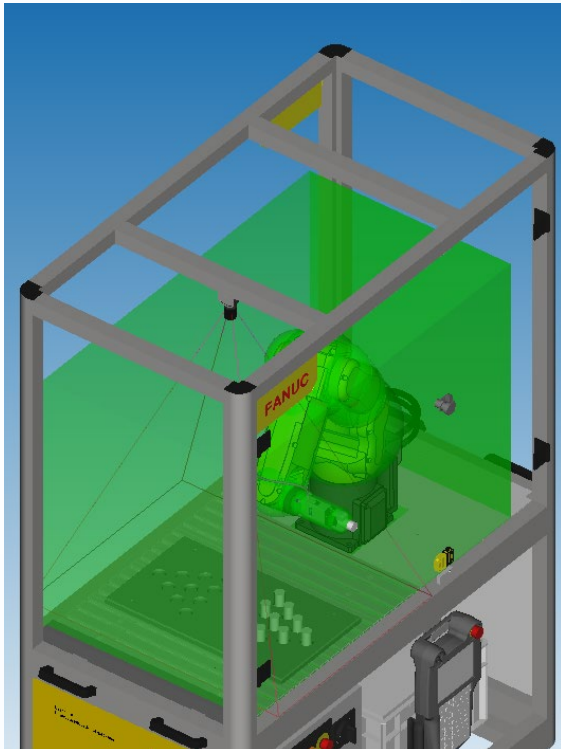
Hogy láthassa a DCS Biztonsági Zóna hatásait, próbálja meg a robottal elhagyni a zónát.

Fordítsa el a megfogót 90 fokkal úgy, hogy a megfogó az ajtó felé mutasson.



DCS megjelenítése

Mozgassa a robotot az ajtó felé, és figyelje meg, mennyit tud elmozdulni. Egy adott pillanatban a robot leáll, és már nem lehet a robotot bármely irányba mozgatni, kivéve a DCS zóna határától visszafelé.



A TP az alábbi hibaüzenetet mutatja ebben az esetben:

SRVO-402 DCS Cart. pos. limit(No.1:Ed Ce

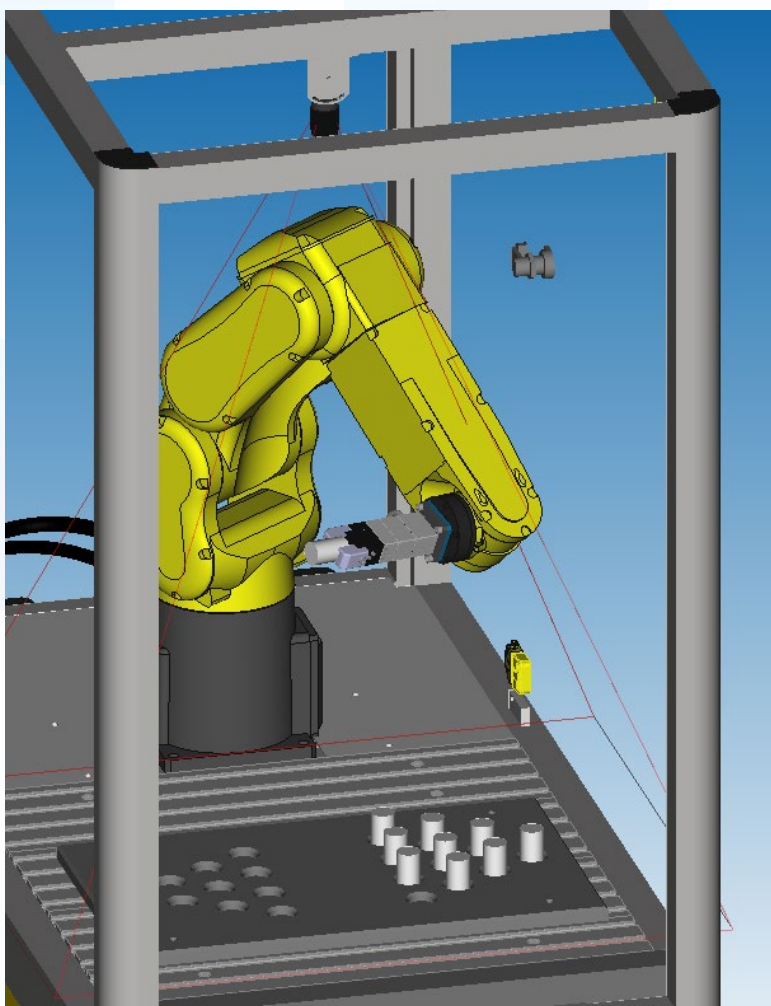
A robot hibahelyzetéből történő kiszabadításához engedje fel a shift gombot, majd nyomja meg újra a shift gombot, majd nyomja meg a reset gombot. Most mozgassa el a robotot a zóna határától.



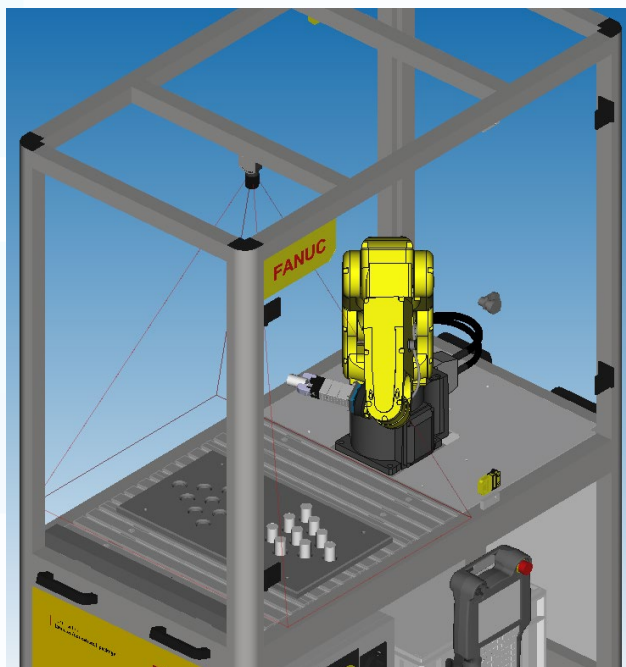
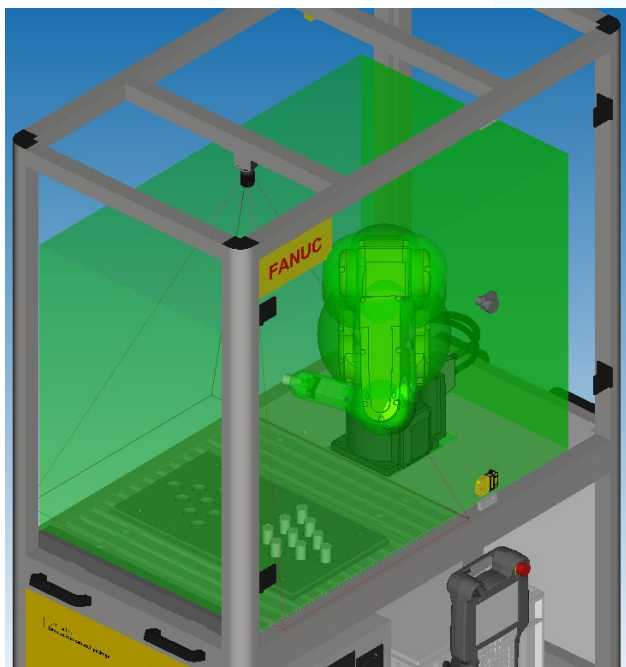
DCS megjelenítése

Mozgassa el a robotot az ajtótól, és fordítsa meg a szerszámot úgy, hogy a robotkar az ajtó közelében legyen.

Mozgassa újra a robotot az ajtó irányába. A robot ismét megáll az ajtó előtt.



DCS megjelenítése

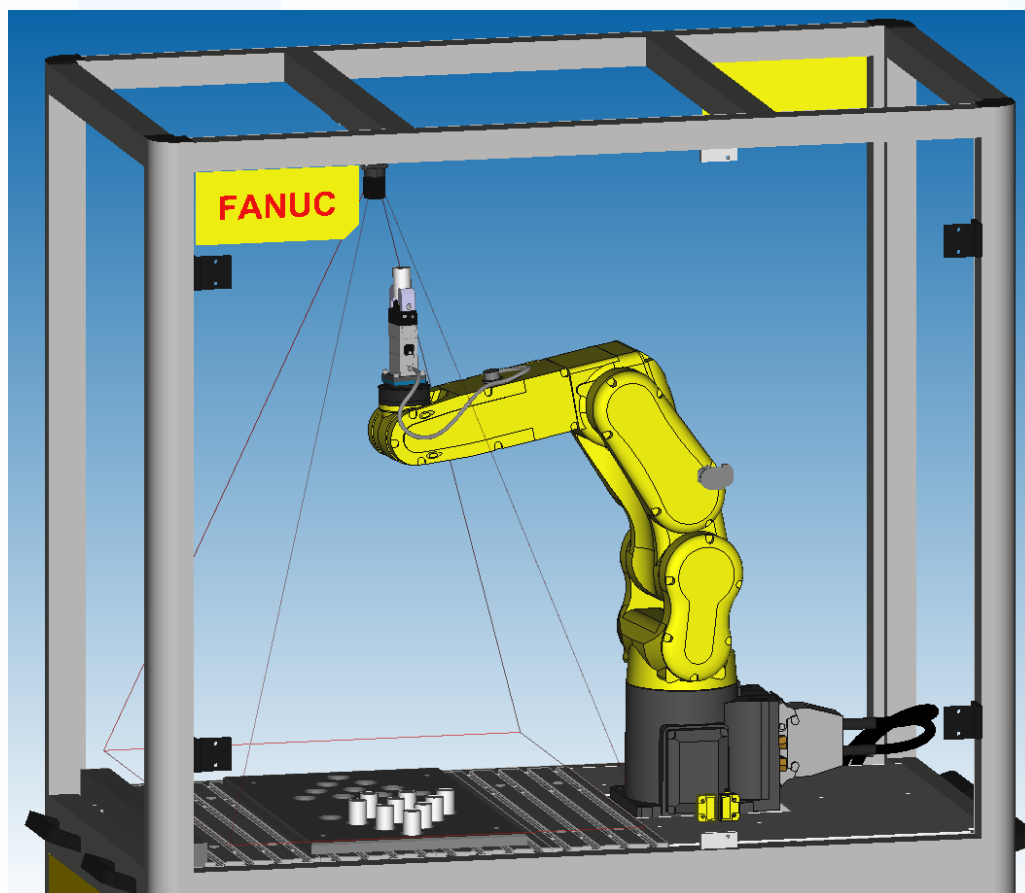


Ez azért lehetséges, mert a vezérlő nemcsak azt ellenőrzi, hogy az szerszám kívül van-e a zónán, hanem magát a robotot is, így a robot egyetlen része sem kerülhet veszélyes helyzetbe.



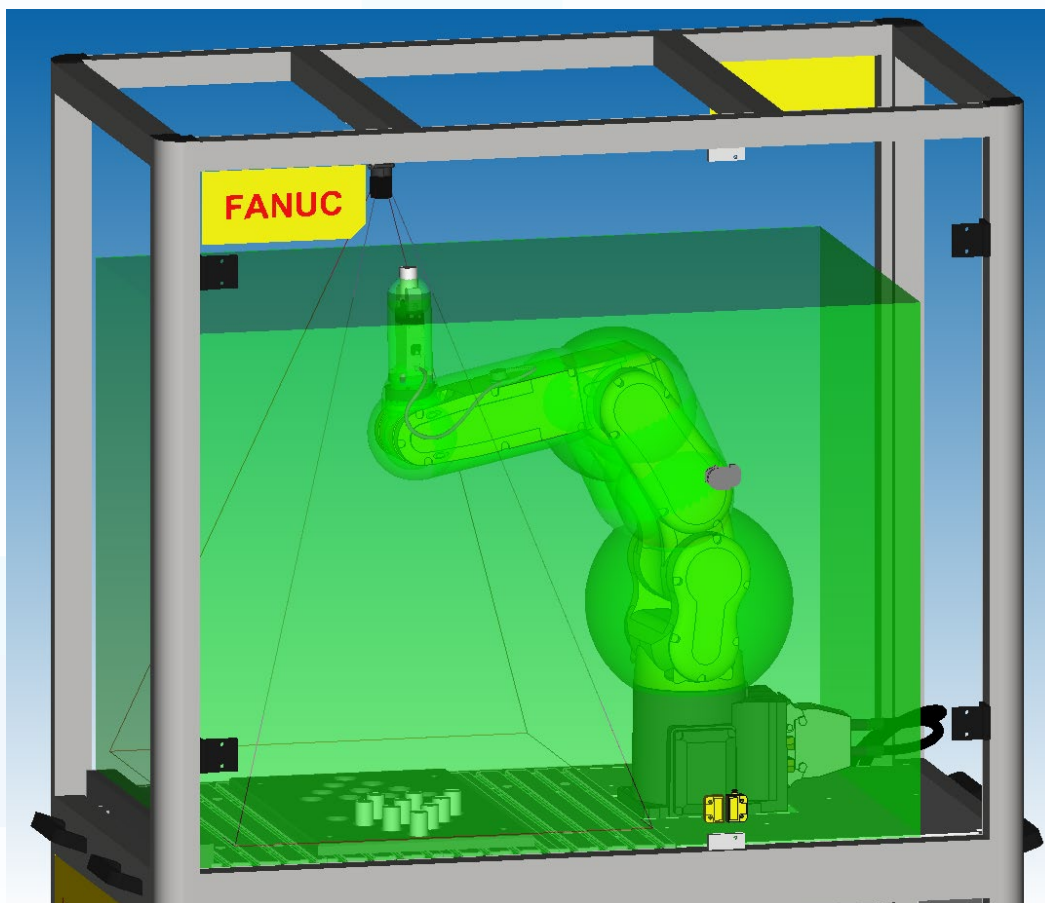
DCS Megjelenítése

Most meg akarjuk mutatni, hogy nemcsak a cella oldalfalain van biztonságos zóna, hanem a tetején is. Ennek érdekében mozgassa a robotját úgy, hogy a szerszám felfelé nézzen.



DCS Megjelenítése

Mozgassa a robotot felfelé, amíg az meg nem áll, és a hibaüzenet ismét megjelenik.



Következő lépésként megváltoztatjuk a DCS biztonsági zónát, és látható lesz a különbség a robot mozgásának korlátozásában.

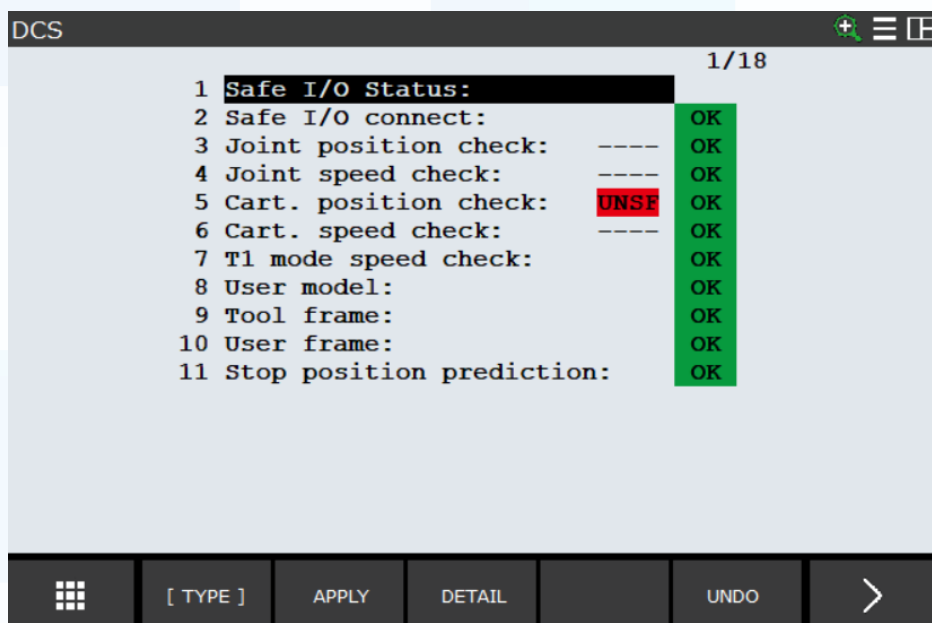
DCS beállítása

Először el kell jutnunk a DCS beállításaihoz. Ezért nyomja meg a menü gombot, a második oldalon lépjen a SYSTEM (6), majd a DCS (8) menübe.

FONTOS

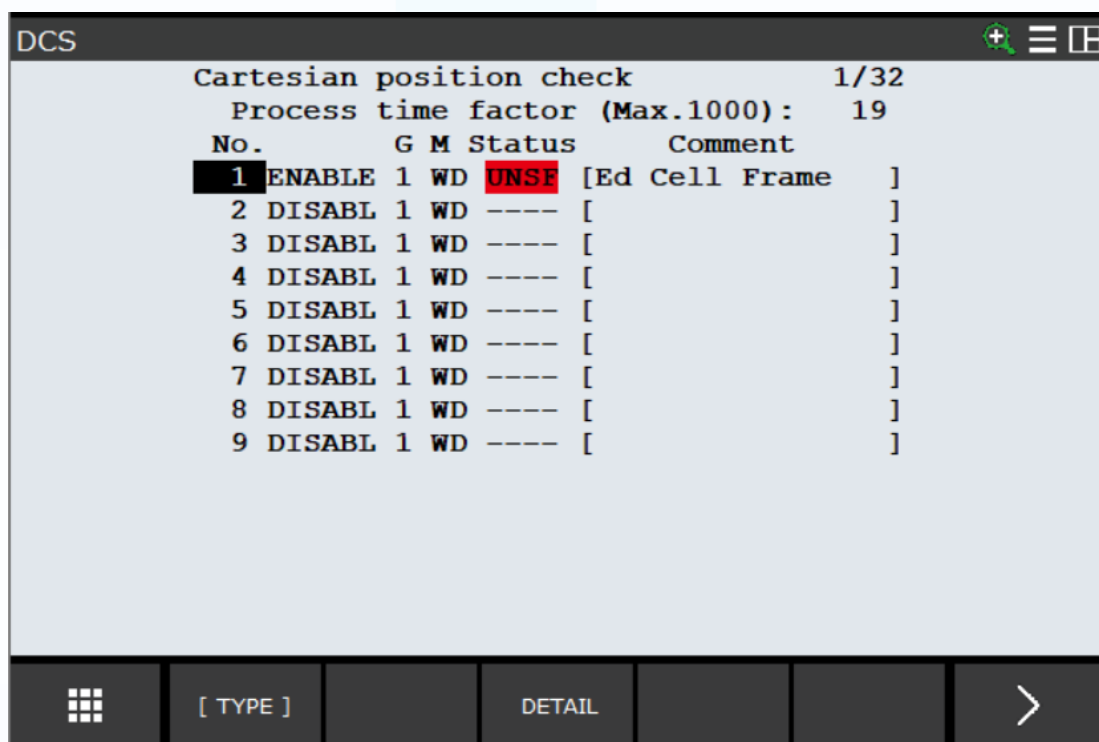
Módosítani készül a rendszerfájlokat és a beállításokat. Ügyeljen arra, hogy mindent pontosan úgy végezzen, ahogy mi tanácsoljuk, mert ellenkező esetben súlyos károk és sérülések léphetnek fel!

Különböző módszerek léteznek a DCS biztonságos zóna beállítására. A Cart. Position Check-et alkalmazzuk. Ha még nem mozdította el a robotot, akkor az UNSF-nek (UnSafe) pirossal jelöltnek kell lennie, mivel a robot a Biztonsági Zóna határán van, ezért nem biztonságos a jelenlegi állapot.



DCS beállítása

Lépjen be a **Cartesian position check** menübe.



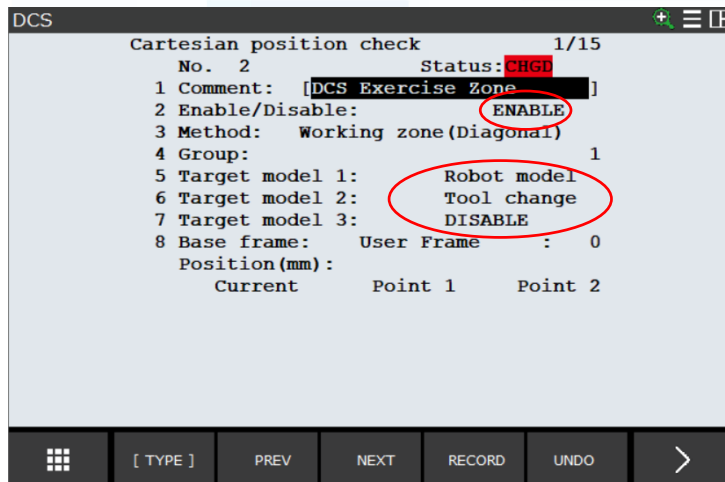
No.	G	M	Status	Comment
1	ENABLE	1	WD UNSF	[Ed Cell Frame]
2	DISABL	1	WD ----	[]
3	DISABL	1	WD ----	[]
4	DISABL	1	WD ----	[]
5	DISABL	1	WD ----	[]
6	DISABL	1	WD ----	[]
7	DISABL	1	WD ----	[]
8	DISABL	1	WD ----	[]
9	DISABL	1	WD ----	[]

Jelenleg csak egy biztonsági zóna engedélyezett. De lehetséges egynél több Biztonsági Zóna engedélyezése is, és abban az esetben a vezérlő folyamatosan ellenőrzi, hogy az összes robotrész az engedélyezett zónákon belül van (vagy kívül van).



DCS beállítása

Most szeretnénk létrehozni saját zónát. Menjen a második (üres) pozícióba és lépjen be a zóna beállításaihoz. Először jóvá kell hagynia, hogy változtatni szeretne a beállítóakon.



8 Base frame:	User Frame	: 0
Position (mm):		
Current	Point 1	Point 2
9 X	501.9	-250.0
10 Y	267.6	-290.0
11 Z	-167.5	400.0
12 Stop type:	Stop Category 0	
13 Speed check:	<DETAIL>	

Ezután adjon nevet a zónának, állítsa a (3) módszert átlóra (IN – azaz WorkZone), az 1. célmodellt (5) a robotmodellre, a 2. célmodellt (6) a szerszámcsere, és másolja az 1. és 2. pont helyzetadatait a másik (jelenleg aktív) beállításból. Végül engedélyezze ezt a zónát (2).

DCS beállítása

Most az új zóna engedélyezve van, de a régi is engedélyezve van. Lépjen a régi „Ed Cell Frame” beállítás oldalra, és tiltsa le azt a zónát. Most látnia kell, hogy a régi zóna le van tiltva, és az új engedélyezve van.

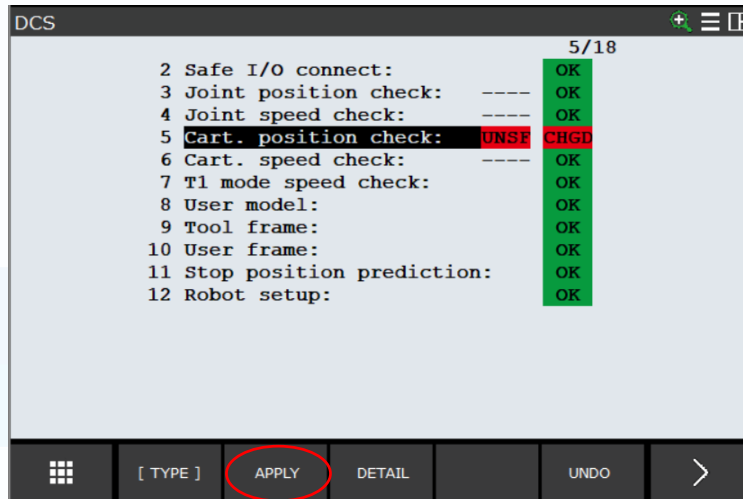
DCS				
Cartesian position check				1/32
Process time factor (Max.1000):				16
No.		G M	Status	Comment
1	DISABL	1 WD	CHGD	[Ed Cell Frame]
2	ENABLE	1 WD	CHGD	[DCS Exercise Zo>]
3	DISABL	1 WD	----	[]
4	DISABL	1 WD	----	[]

A CHGD jelölés azt jelenti, hogy az aktuális beállítások megváltoztak, aktiválásához jóvá kell hagynia a változtatásokat és a rendszert újraindítani.

A jóváhagyás előtt/nélkül a beállítások nem aktívak!

A változtatások jóváhagyásához, nyomja meg a PREV gombot, hogy visszalépjen a DCS főmenübe. Itt nyomja meg az F2 (Apply) gombot.

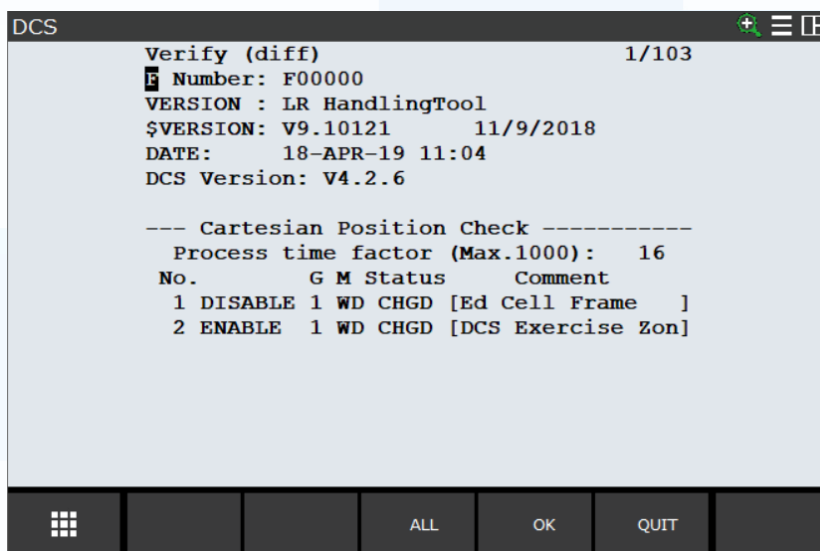
DCS beállítása



A rendszer megkérdezi a „Kódszám”-ot, írja be az 1111-et, majd megjelenik egy második ablak, ahol nyomja meg az F4 (OK) gombot. Most újra kell indítania a vezérlőt. Ezen eljárás után az új DCS biztonságos zónája aktív.

```

10 User frame:
11 Stop position prediction:
12 Robot setup:
Code number(master): ----
    
```



Biztonsági zóna méretezése

A beállításoknak az alábbi szerint kell kinézniük. Ha nem, lépjen vissza a 13. oldalra, és ismételje meg a lépéseket.

DCS				
Cartesian position check				1/32
Process time factor (Max.1000):				16
No.	G	M	Status	Comment
1	DISABL	1	WD ----	[Ed Cell Frame]
2	ENABLE	1	WD SAFE	[DCS Exercise Zo>]
3	DISABL	1	WD ----	[]
4	DISABL	1	WD ----	[]

Törölje a hibát, és kissé mozgassa a robotot lefelé (kb. 100 mm-t). Meg akarjuk változtatni a zóna magasságát, ezért térjünk vissza a beállításokhoz.

Módosítsa a magasságot az 1. pontban (Z koordináta) 340-re. Alkalmazza a beállításokat és indítsa újra a rendszert.

DCS				
Cartesian position check				11/15
3	Method:	Working zone (Diagonal)		
4	Group:	1		
5	Target model 1:	Robot model		
6	Target model 2:	Tool change		
7	Target model 3:	DISABLE		
8	Base frame:	User Frame	:	0
Position (mm) :				
	Current	Point 1	Point 2	
9	X	489.7	750.0	-250.0
10	Y	261.1	-290.0	290.0
11	Z	92.5	340.0	-330.0



[TYPE]

PREV

NEXT

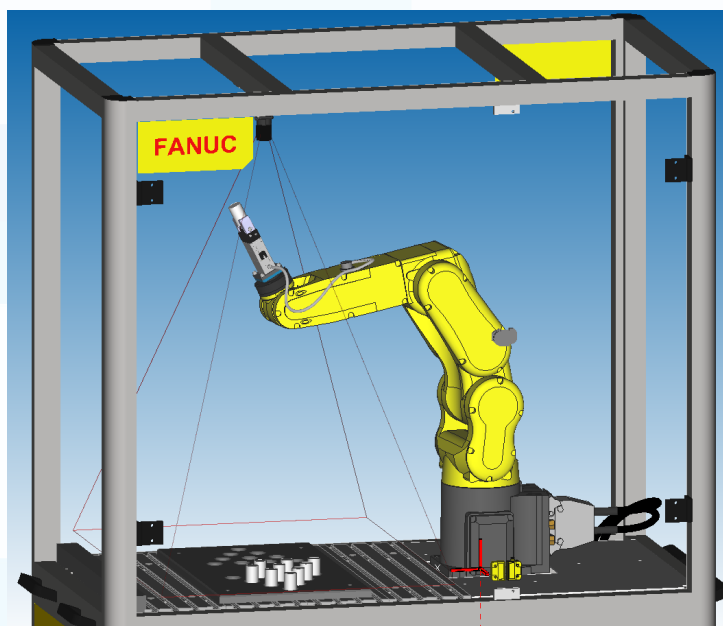
RECORD

UNDO

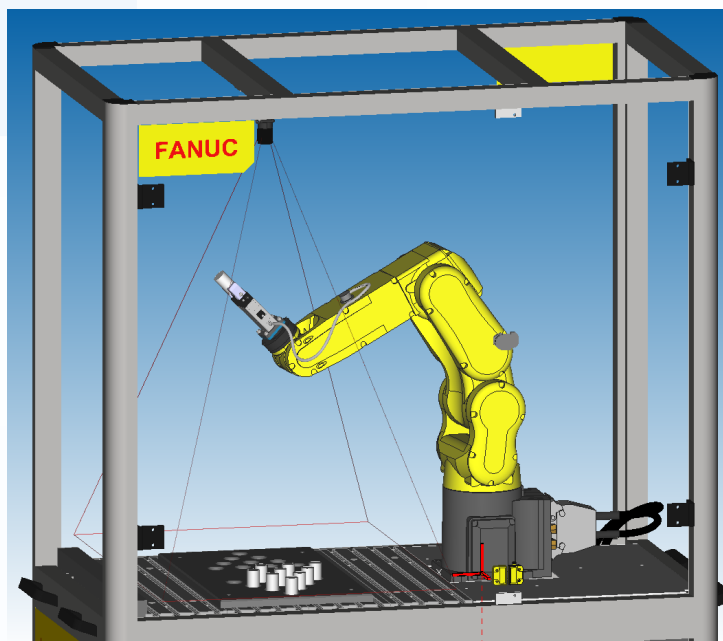


Biztonsági zóna méretezése

Most próbálja ismét felfelé mozgatni a robotot. Láthatja, hogy a robot valamivel korábban fog megállni.



Előtte

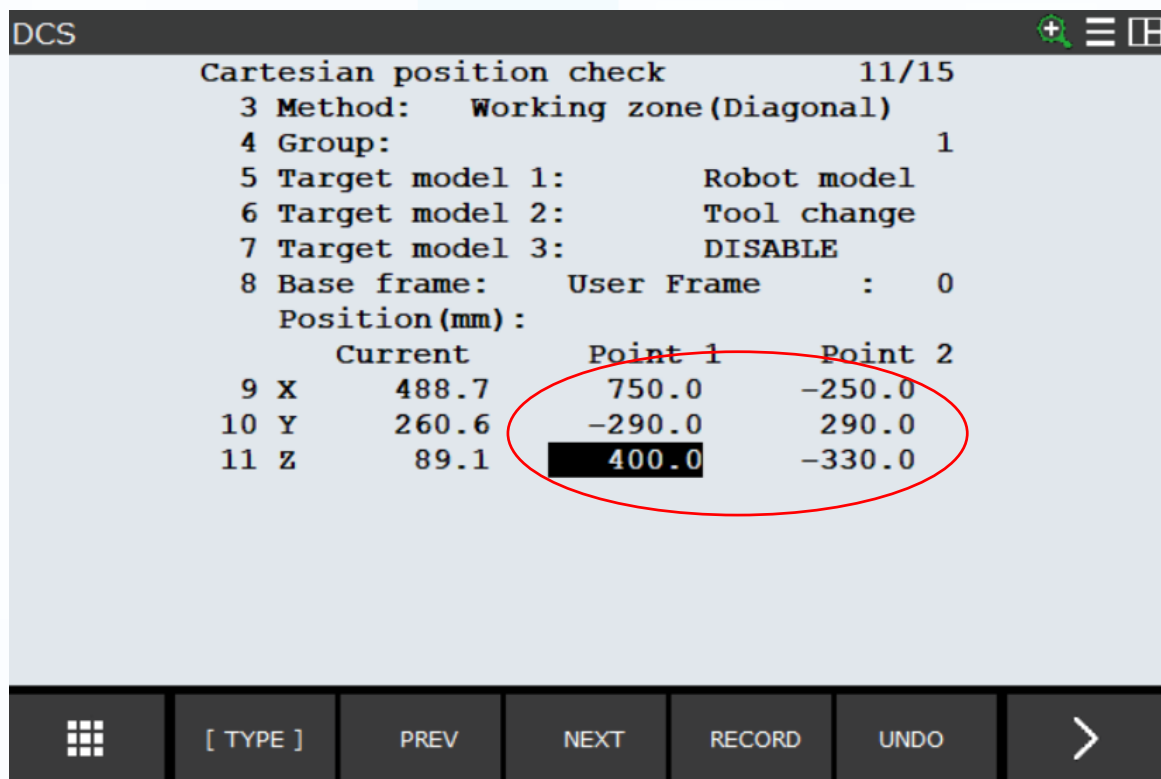


Utána



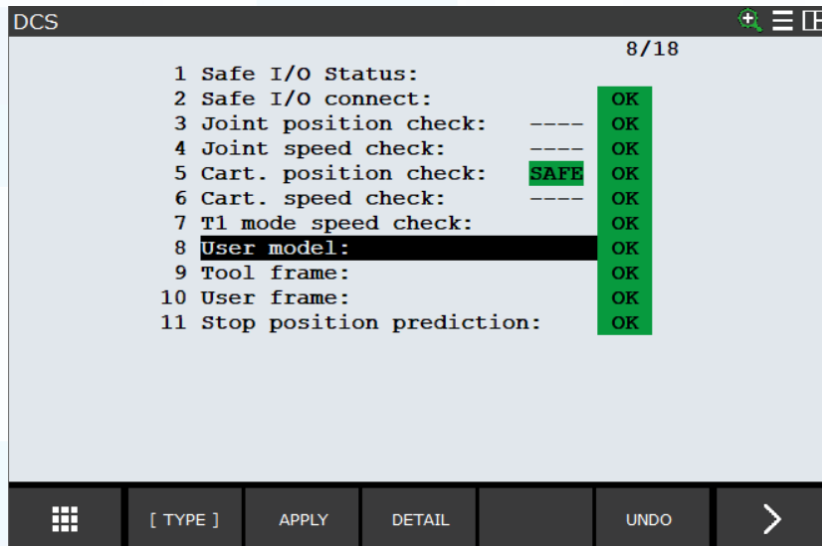
Biztonsági zóna méretezése

Állítsa vissza a zóna magasságát az eredeti értékre (400). A biztonság kedvéért nézze meg az alábbi kép tartalmát.



User Model beállítása

A másik lehetőség az eszköz beállításának megváltoztatása a zóna beállítás helyett. Ehhez először be kell állítanunk a User Model-t. (Van még más lehetőség is a Tool Frame használatára, de ebben a gyakorlatban csak a User Model beállítását tárgyaljuk).

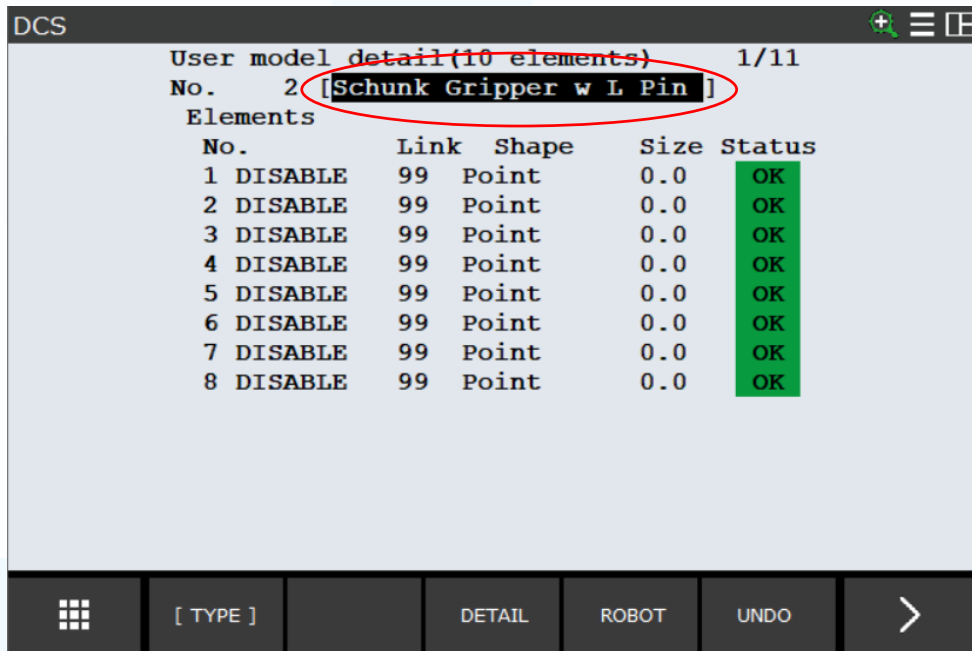


A User Model már előre be lett állítva a Schunk megfogóhoz. Mindazonáltal létrehozunk egy újat, amely megmutatja, hogyan kell beállítani.

DCS					2/16
User model list					
No.	Elem	Status	Comment		
1	1	OK	[Schunk Gripper		]
2	0	OK	[		]
3	0	OK	[		]
4	0	OK	[		]

User Model beállítása

Lépjen be a második beállításba.
Elsőként adja meg a model nevét.



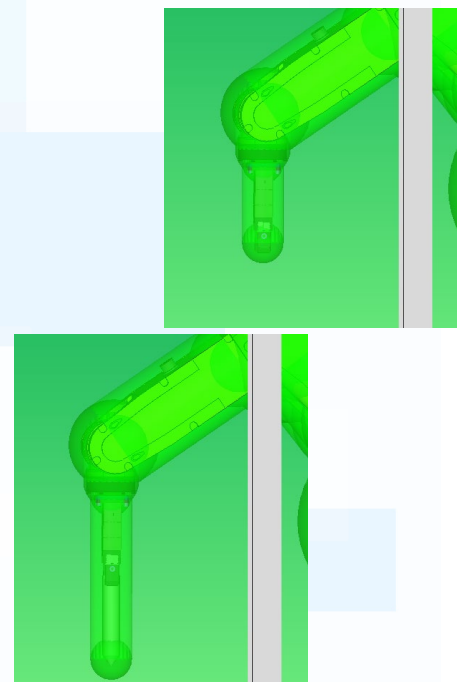
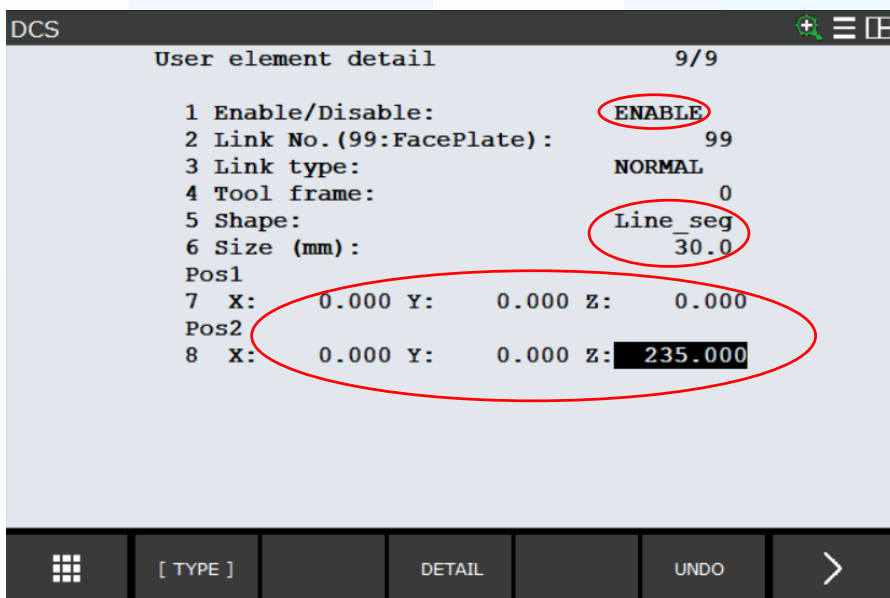
Ezután lehetőségünk van 10 különböző modell beállítására, amelyeket kombináltan használunk végső modellként. Ehhez az egyszerű eszközhöz azonban csak egy modellre van szükségünk. Adja meg az első modellt.

User Model beállítása

Be akarjuk állítani a megfogót a TCP tanításhoz használt hosszú TCP csap segítségével. Mivel ez a csap lényegesen hosszabb, mint a megfogó, új modellt kell megtanítanunk a vezérlőknek, hogy elkerüljük a csap csúcsának a biztonságos zónán kívülre jutását. Először engedélyeznünk kell az aktuális modellt, majd a Line_seg alakot választjuk, 30 mm-t beillesztünk méretként, majd a 2. pont Z helyzetét 235-re változtatjuk (távolság a mm-ben a szerszámrögzítő karimától a csap hegyéig)

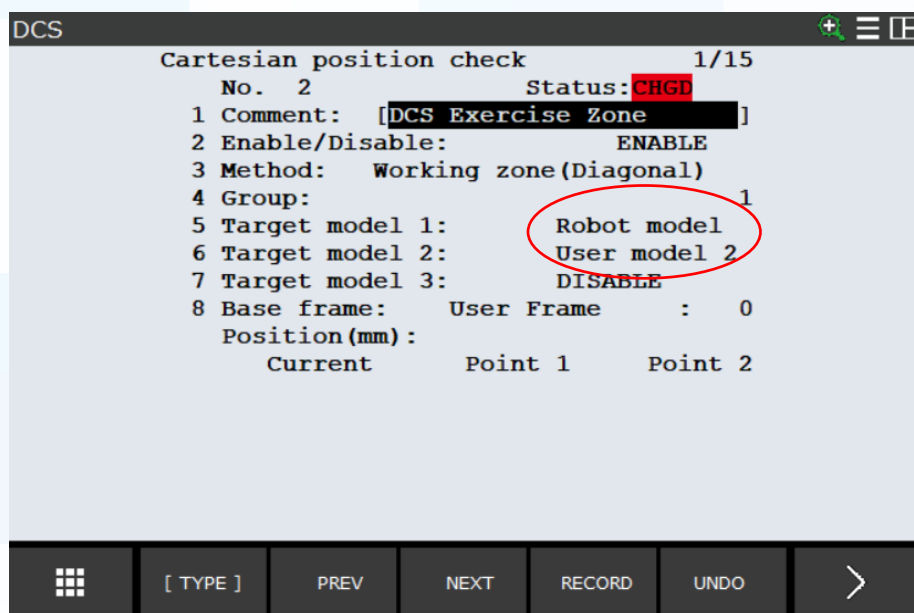
Bővebb információkat a User Model lehetőségeiről az R-30iB_Plus_DCS_Manual section 3.4. fejezetben "Setup of User Model" és 4.2 fejezetben "Setup of User Model" talál.

Ezt a távolságot a TCP csap hosszának megfelelően állíthatja be, ha rövidebb vagy hosszabb. A karima és a megfogó csúcsa közötti hossz 115 mm.



User Model beállítása

Most, hogy beállítottuk a megfogó modellt, ezt a modellt be kell állítanunk a Cartesian Position Check képernyőn. Ezekben a beállításokban lehetőségünk van 3 célmodell beállítására, amelyeket a vezérlő működése közben ellenőriz. Az első célmodell (robotmodell) a robot. Ez azt jelenti, hogy maga a robot nem léphet ki a biztonságos zónából. A 2. célmodellként beállítjuk a 2. felhasználói modellt, az imént létrehozott eszközt.



Most újra alkalmaznia kell a beállításokat és a rendszert újraindítani. Az újraindítás előtt mozgassa lefelé a robotot, hogy a megfogó ne érjen túl a zóna határán.

Az új szerszámot (TCP-t) be kell állítani.



User Model beállítása

Az ellenőrzéshez addig mozgassa a robotot, amíg le nem áll. Ha mindent jól csinált, akkor a TCP tű hegyének ott kell megállnia, ahol a megfogó hegye először megállt. Ugyanezt kipróbálhatja az oldalfalon is. Ha akarja, válthat az „Ed Cell Frame” zóna és a saját zóna között, hogy lássa a különbséget a két beállítás között.

FONTOS

Legyen biztos benne, hogy a csap nincs a megfogóban, ellenkező esetben az megkarcolhatja a falakat!

Ehhez győződjön meg arról, hogy engedélyezte az első zónát, majd letiltotta a második zónát, alkalmazza a beállításokat és indítsa újra a rendszert. Miután végzett, győződjön meg arról, hogy az első zónát újra aktívvá tette. Engedélyezze az első, és tiltsa le a második, hogy a rossz (keresztbe) biztonsági zóna beállítások miatt ne fordulhasson elő működési zavar. Alkalmazza a beállításokat és indítsa újra a rendszert. Győződjön meg róla, hogy a beállítások elfogadásra kerülnek a gyakorlat befejezéséig.

DCS				
Cartesian position check				1/32
Process time factor (Max.1000):				19
No.	G	M	Status	Comment
1	ENABLE	1	WD	SAFE [Ed Cell Frame]
2	DISABL	1	WD	---- [DCS Exercise Zo>]
3	DISABL	1	WD	---- [
4	DISABL	1	WD	----



Összefoglalás

Ebben a gyakorlatban megtanulhatta, hogy mi a DCS biztonsági zóna, és hogyan működik.

Önnek képesnek kell lennie arra, hogy önállóan létrehozzon egy biztonsági zónát, és beállítson egy felhasználói modellt is a megfogóhoz rögzített eszközének megfelelően.

Tudnia kell, hogyan kell alkalmazni a beállításokat a DCS menüben.





Gyakorlat 10

Testreszabott *i*Pendant Képernyő

2. lap

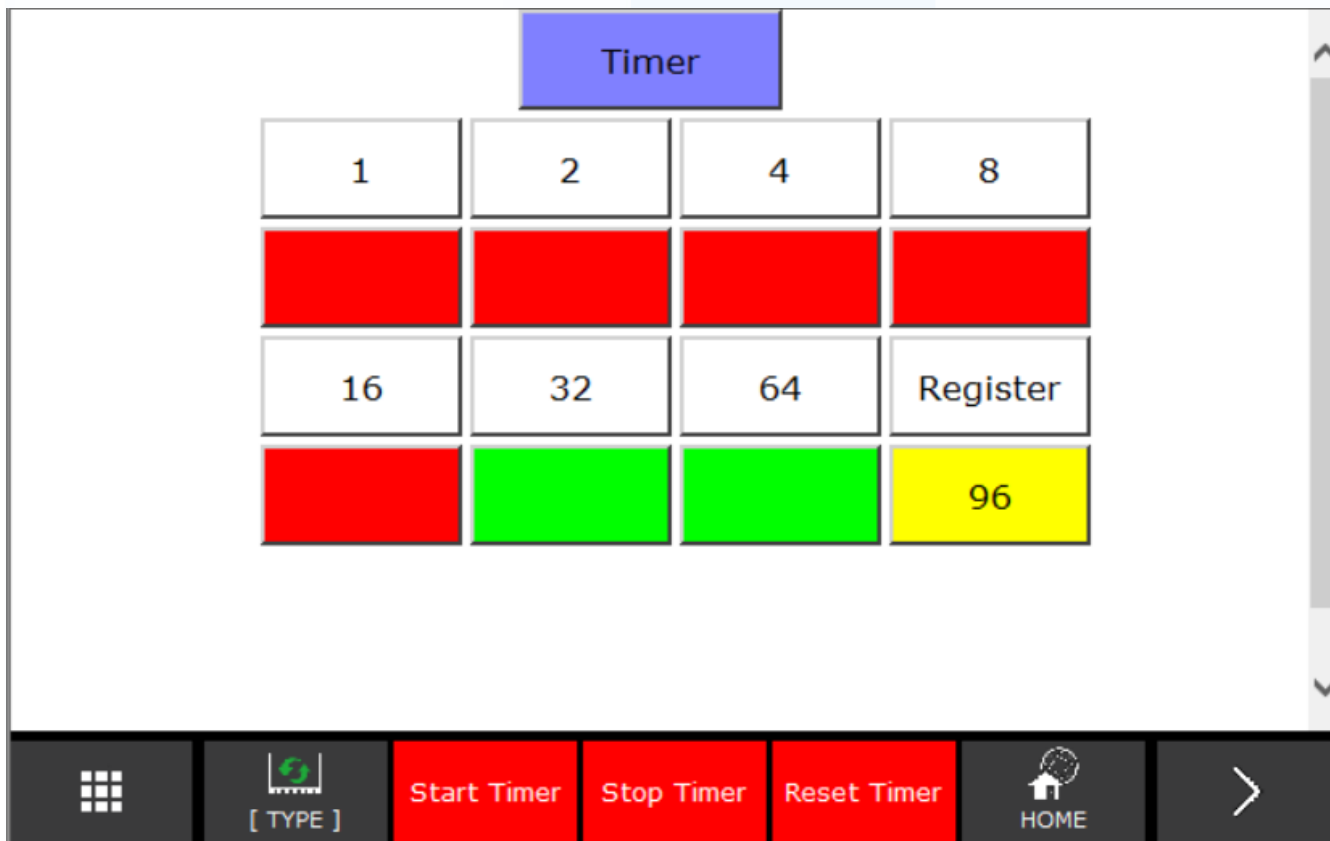


Bevezető

Ebben a gyakorlatban megtanulhatja, hogyan tudja testreszabni az iPendant (Teach Pendant) képernyőjét, és hogyan hívhatja meg ezeket a képernyőket a programokban.

Azt is megtanulhatja, hogy a testreszabáshoz milyen különféle vezérlők használhatók.

A gyakorlat után képesnek kell lennie arra, hogy létrehozzon, feltöltsön és használjon egy html weblapot a TP képernyőn.



Gyakorlat 10

Testreszabott iPendant képernyő

Tartalom

Bevezető	-	3
Háttér	-	5
Eszköz	-	6
MS SharePoint telepítés	-	7
Első lépések	-	8
Cimke	-	10
Feltöltés a robotra	-	12
Lap megjelenítése	-	15
Gombok és lámpák	-	16
Képek	-	23
Lap meghívása programból	-	25
Multi Control	-	26
Összefoglalás	-	29



Háttér

A felhasználó által definiált HTML oldalak arra szolgálnak, hogy az üzemeltető vizuális visszajelzést kapjon arról, hogy a robot mit csinál.

Interfész lehet a kezelő számára is, hogy további hardware bemenetek nélkül vezérelhesse a robotot (lásd I/O Box).

Már használt néhány HTML oldalt a robot működtetése közben, anélkül, hogy kifejezetten tudta volna. Az AAA_Demo program során megjelenített oldalak testreszabott HTML oldalak voltak, amelyeket a program hívott meg.

Ezek az oldalak segítettek lehetőséget adni a felhasználónak, hogy megadjon néhány parancsot, és hogy valamilyen információt kapjon arról, hogy mi történik a robottal.

FONTOS

Ebben a gyakorlatban minden példa az iPendant_customization_operator_manual_[B-83594EN_01] (iCOM)-n alapszik, kivéve a bejelentett részek.



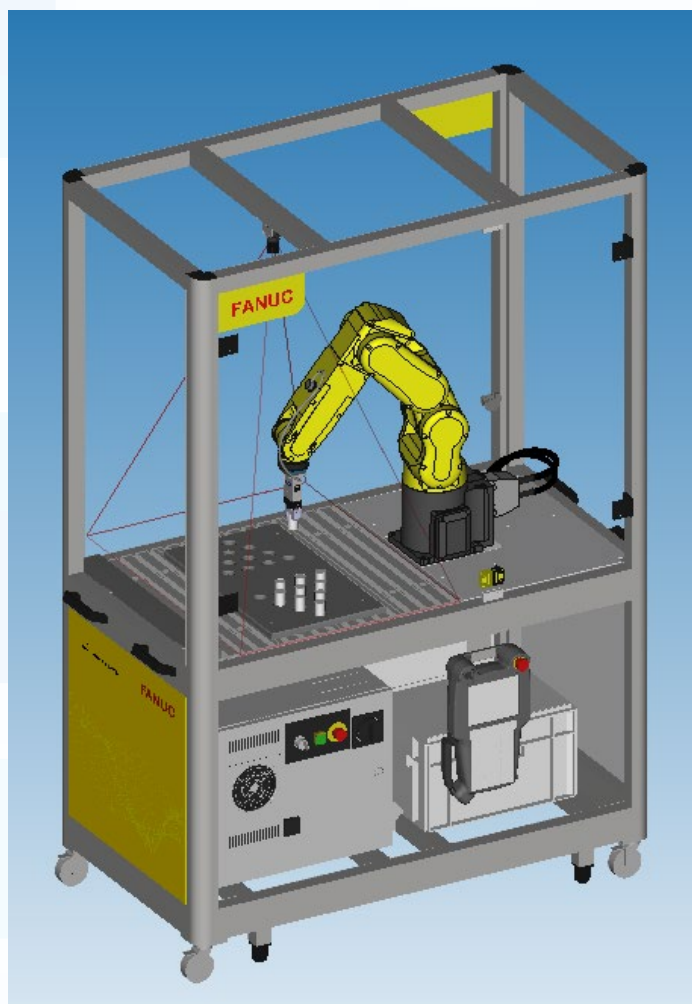
Eszköz

Ehhez a gyakorlathoz szükséges:

Oktatócella

Számítógép feltelepített Roboguide-al

Pendrive.



MS SharePoint Designer telepítés

Javasoljuk, hogy az .stm fájlokat a Microsoft Office SharePoint Designer 2007 alkalmazásban készítse el. Ez egy ingyenesen letölthető szoftver.

A SharePoint számítógépre történő telepítésének részletes leírása az iCOM 3.3. fejezetben található „Telepítés”. Győződjön meg róla, hogy hozzáadta az ActiveX vezérlőket az eszköztárához, ezt később használhatja az oldalhoz hozzáadott minden vezérlőhöz. (Lásd az iCOM 3.3.1. fejezet: „ActiveX vezérlő parancsikon”)



Első lépések

Ebben a gyakorlatban nem fogjuk lefedni az összes elérhető vezérlőt. E gyakorlat után azonban képesnek kell lennie arra, hogy a kézikönyv segítségével használja azokat a kezelőszerveket, amelyekre ez a gyakorlat nem terjed ki.

Az összes rendelkezésre álló vezérlés gyors áttekintéséhez olvassa el az iCOM 3.4 fejezet „Vezérlési funkciók összefoglalása” szakaszát.

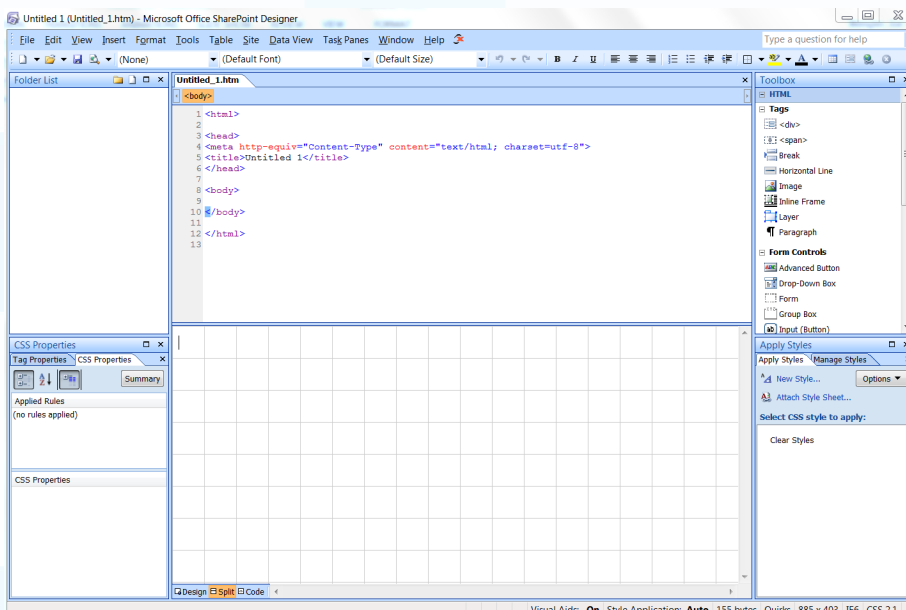
Ez a gyakorlat a következő ellenőrzésekre terjed ki:

- Címke
- Lámpa
- Váltó gomb
- Kép
- Többszörös kép



Első lépések

Most egy új dokumentum létrehozásával kezdjük a feladatot a SharePointban.



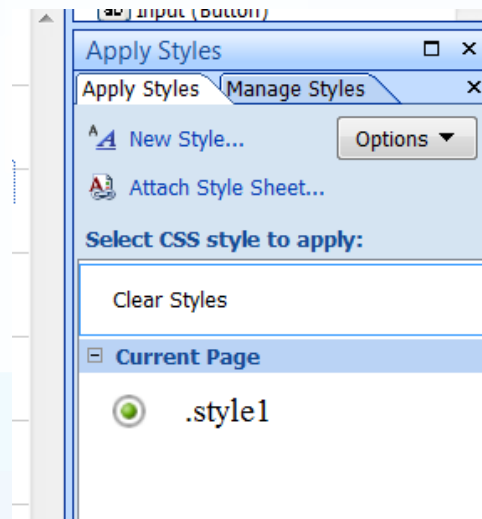
Elsőként változtassa meg a fejléctet “Exercise10”-re.

```
1 <html>
2
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
5 <title>Exercise10 first page</title>
6 </head>
7
8 <body>
9
10 </body>
11
12 </html>
13
```



Cimke

Kezdjük egy egyszerű címkevezérlő hozzáadásával. Kattintson az eszköztár ActiveX vezérlő ikonjára, válassza ki a listából a **FANUC iPendant Label Control** elemet, majd nyomja meg az OK gombot. Most van egy címke az oldalon. Igazíthatja a címkét szóközzel, tabulátorral és Enter-el. Alternatív megoldásként megadhat egy új stílust a jobb alsó sarokban.



Beállíthatja a címke helyét, a betűtípus beállításait, stb. Ha ezeket a beállításokat egy címkére (vagy más vezérlőre) kívánja alkalmazni, módosítsa a `<p>` értéket `<p class = "nameofyourstyle">` értékre.

```
20 <p>  
21 &nbsp;</p>  
22 <p class="style1">  
23 <object classid="clsid:710  
24 <param name="Caption"
```

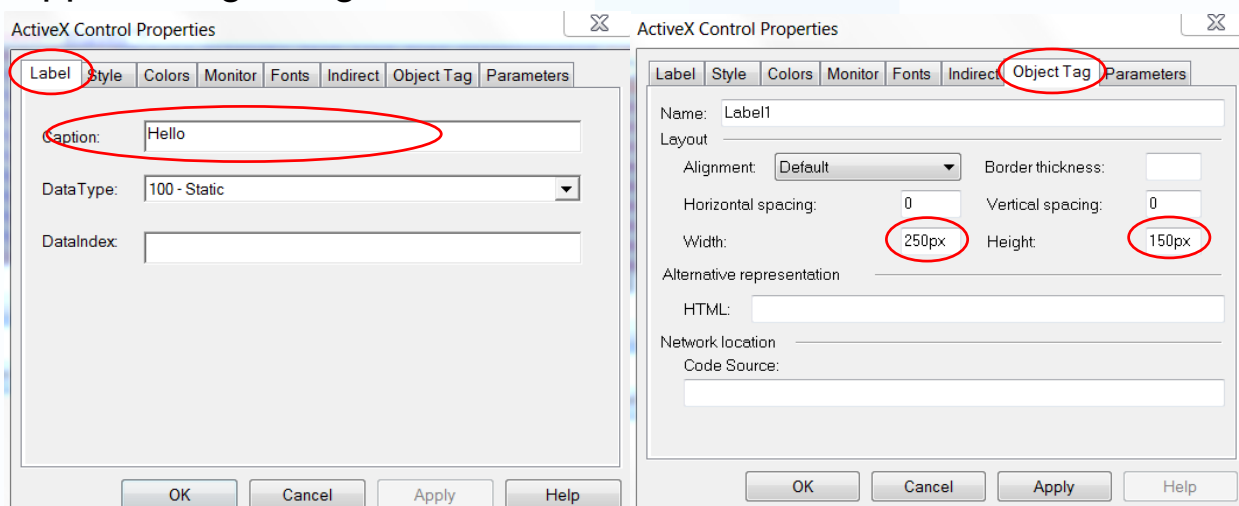


Címke

A vezérlő beállításainak módosításához kattintson duplán a vezérlőre.

Bővebb információk az iCOM 5.4.1 fejezet "Label Control" részben.

Egy kicsit elmélyülünk több beállításban. Egyelőre adjon hozzá egy Feliratot a Címke beállításaihoz, és módosítsa az Object Tag beállítások mezőjének méretét 250 képpontos szélességre és 150 képpont magasságra.



Most mentse a dokumentumot .stm néven (egyszerűen adja hozzá a .stm nevet a dokumentum neve mögé).

Feltöltés a robotra

Hogy a file-t feltöltse Roboguide-ra, helyezze el a dokumentumot az UD1 könyvtárban. Ez a könyvtár itt található: *Documents\My Workcells\FEC_Education_Cell_V2_4\Robot_1\UD1*.

Valós robotra való feltöltéshez másolja a fájlt egy USB pendrive-ra, majd dugja be azt a vezérlőszekrénybe.

A következő eljárás megegyezik valós roboton és Roboguide-on. Nyomja meg a menü gombot, lépjen a File (7) menübe. Nyomja meg az F5 (Util) gombot, majd Set Device. Válassza ki az USB Disk (6)-et.

FILE	
MENU 1	FILE 1
1 UTILITIES	1 File
2 TEST CYCLE	2 File Memory
3 MANUAL FCTNS	3 Auto Backup
4 ALARM	TP programs)
5 I/O	MN programs)
6 SETUP	variable files)
7 FILE	system files)
8 iRvision	config data)
9 USER	DEFAULT files)
0 -- NEXT --	part model files)
	bit-map images)
	PMC files)
	Variable Listings)
	Diagnostic files)
	Vision VD files)

UTIL 1	
1 Set Device	
2 Format	
3 Format FAT32	
4 Make DIR	
[BACKUP]	[UTIL

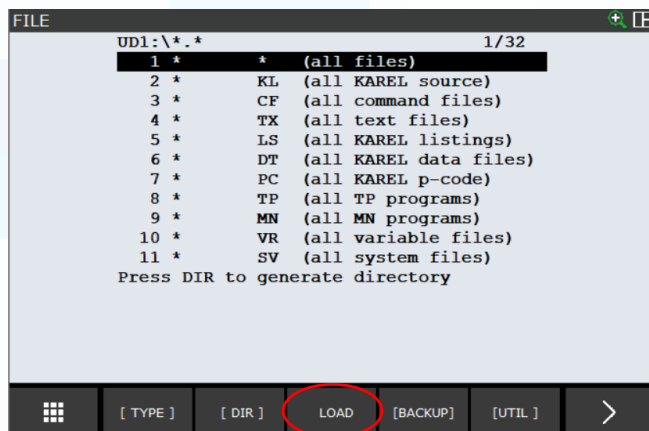
1
1 FROM Disk (FR:)
2 Backup (FRA:)
3 RAM Disk (RD:)
4 Mem Device (MD:)
5 Console (CONS:)
6 USB Disk (UD1:)
7 USB on TP (UT1:)
8



Feltöltés a robotra

Ha csak egyetlen .stm file található a meghajtón, nyomja meg az F3 (Load) és minden file feltöltődik.

A file már elérhető a vezérlőn.

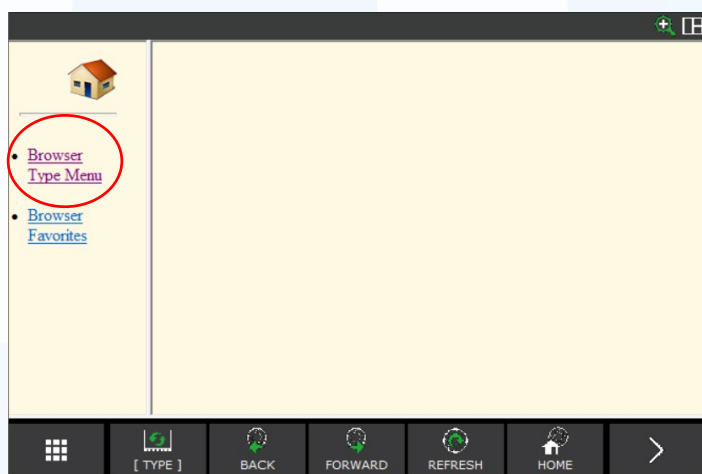
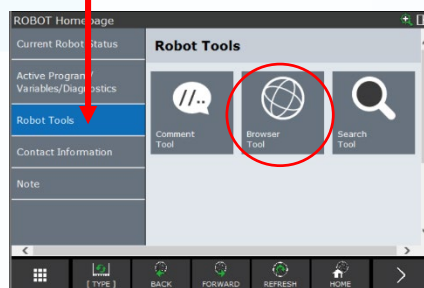
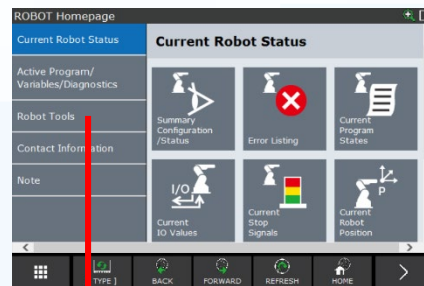
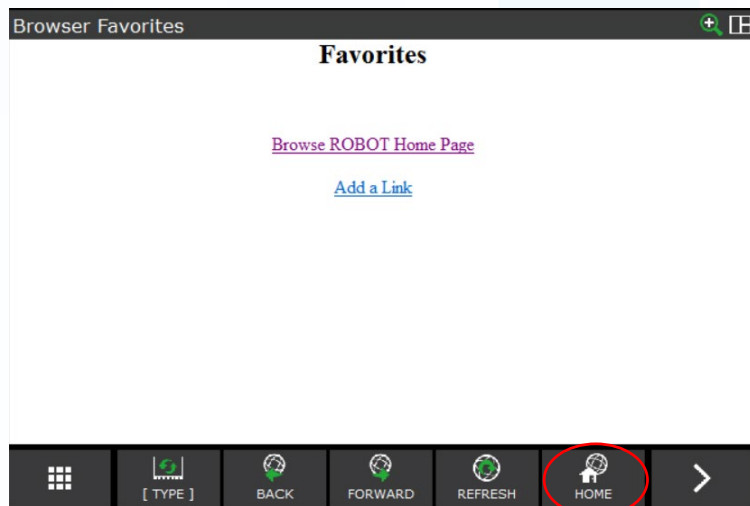


Már csak a lap elhelyezése van hátra a böngészőben. Ehhez nyomja meg ismét a menu gombot, majd a második lapon válassza ki a browser (8)-t.

MENU 2	BROWSER 1
1 SELECT	1 Browser
2 EDIT	2 Panel setup
3 DATA	3 Check
4 STATUS	4 Operate
5 4D GRAPHICS	5 FKey
6 SYSTEM	6 Test Page
7 USER2	VR (a
8 BROWSER	SV (a
9	to genera
0 -- NEXT --	

Feltöltés a robotra

Nyomja meg az F5 (Home) gombot, lépjen a Robot Tools majd Browser Tool menübe. Ezután válassza a Browser Type Menu-t adja hozzá a *.stm fájlt a listához. A címet ehhez hasonlóan adja meg: */fr/nameofyourdocument.stm*.

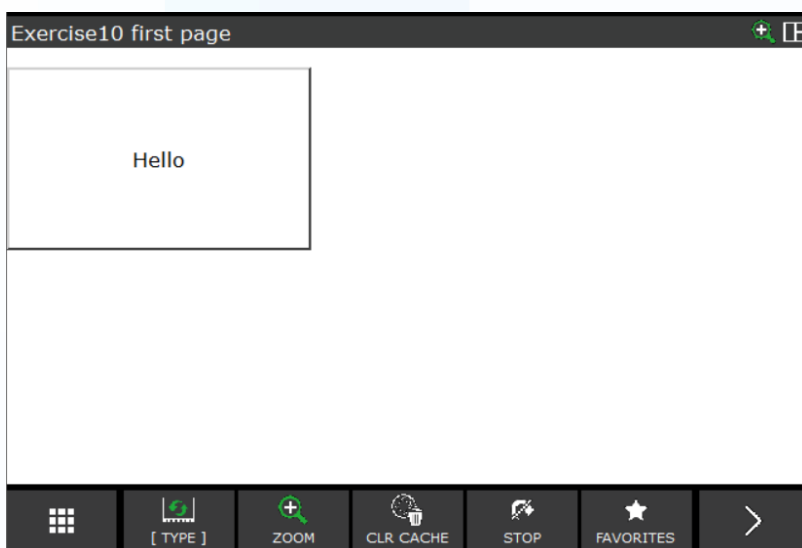
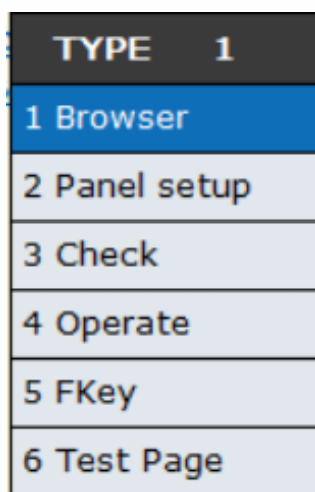


Browser Menu	Name	Add
[1]	Panel setup	ipnl/pnlgen.htm
[2]	Check	/fr/EdCellChk.stn
[3]	Operate	/fr/EdCellOp.stm
[4]	Fkey	/fr/EdCellFkey.st
[5]	Test Page	/fr/Ex10.stm
[6]		

Lap megjelenítése

Ha mindent megtett az utasításoknak megfelelően, akkor az oldalát már lehetséges megjeleníteni a robotvezérlőben.

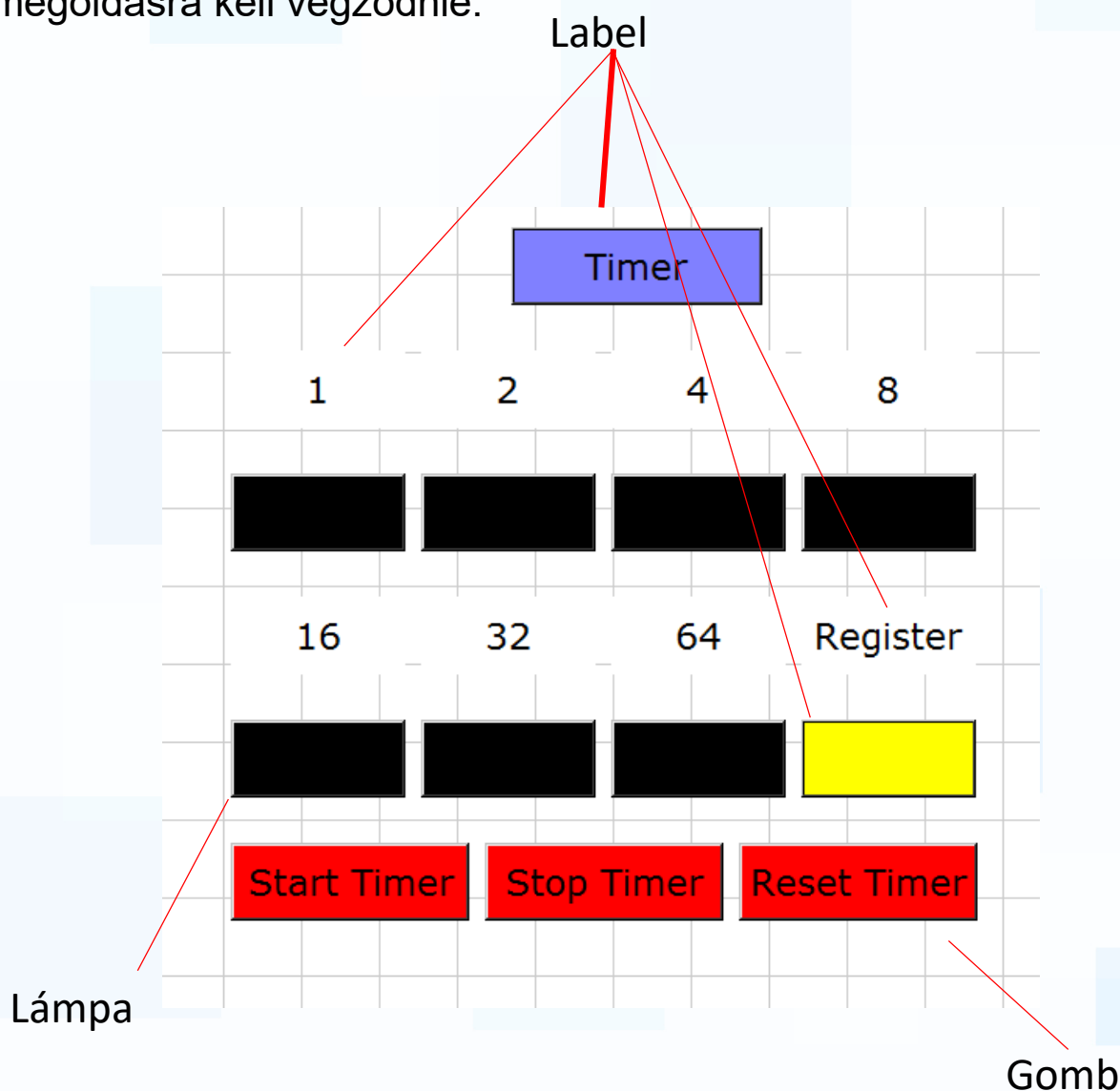
Az oldal megjelenítéséhez nyomja meg az F1 (Type) gombot, és válassza ki az oldalhoz rendelt nevet. Az oldal megjelenik. Legyen biztos abban, hogy az oldal minden módosítása vagy új feltöltés után megnyomta meg az F3 billentyűt (CLR gyorsítótár), mert különben előfordulhat, hogy a tényleges új oldal nem jelenik meg (helyesen).



Gombok és lámpák

Most, hogy ismerjük az oldal programozásának, feltöltésének és megjelenítésének alapjait, egy felhasználói felületet szeretnénk létrehozni a Stopper programhoz. (lásd a 7. gyakorlatot) 7 lámpával kell rendelkezünk a kimenetek számára és 3 gomb a bemenetek számára. És végül egy címke mező a regiszter értékének megjelenítéséhez.

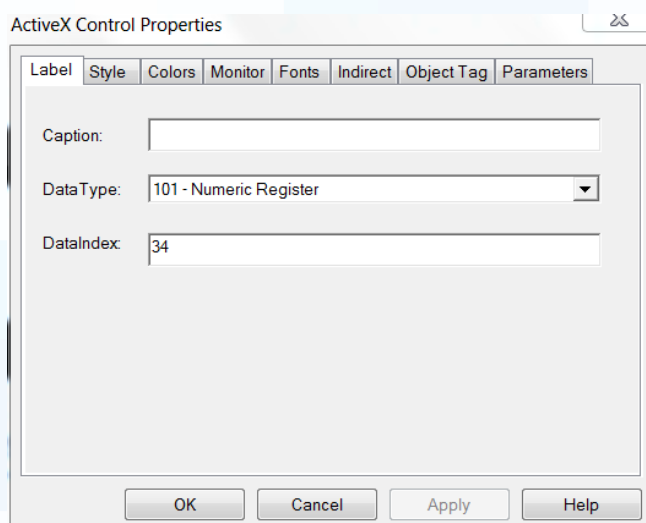
Az oldal kialakítása rajtad múlik. A következőkhöz hasonló megoldásra kell végződnie:



Gombok és lámpák

A sárga kivételével minden címke már beállítható az előző oldalon leírtak szerint.

Ami a sárga címkét illeti, meg kell változtatnia az adattípust 101-numerikus regiszterre, majd meg kell adnia a 34. adatindexet. Ez legyen az Időzítő regiszter, ha még nem változtatott semmit. Ellenőrizze, hogy a regiszter továbbra is ez az érték-e, és ha más, változtassa meg a regisztert a címkén.

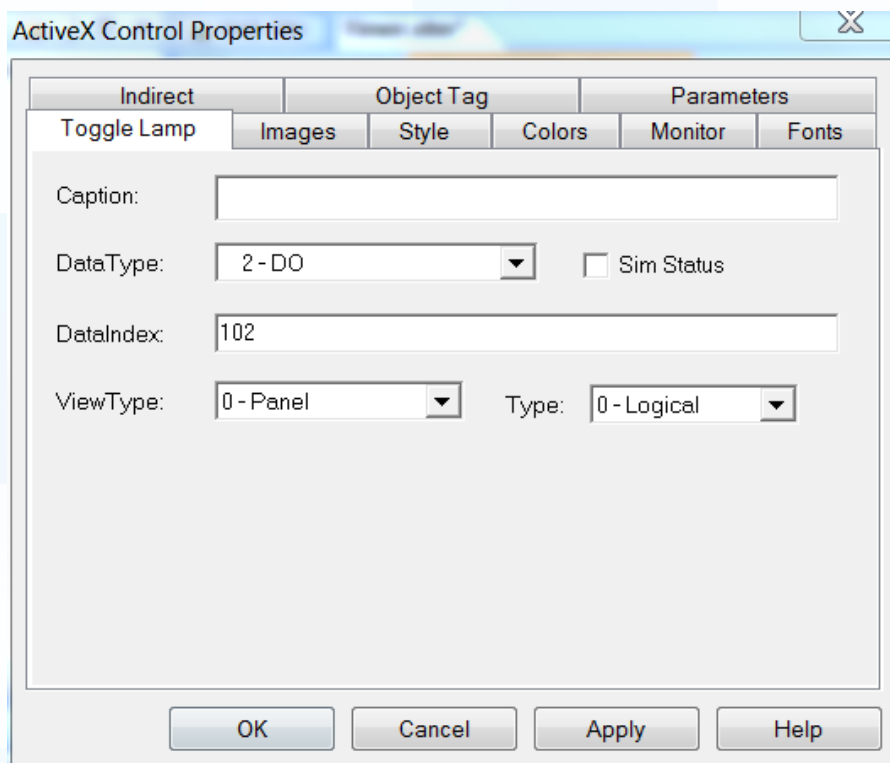


Gombok és lámpák

A lámpák és a gombok egészen hasonlóan működnek, mint az imént felállított címke.

A lámpák esetében a ToggleLamp alatt az Adattípust 2 - DO értékre kell kapcsolnia, és be kell adnia a lámpának megfelelő DataIndex értéket. (102 az első lámpához, 103 a másodikhoz ... magyarázatért lásd a 7. gyakorlatot)

A Colors alatt megváltoztathatja az igaz és hamis színeket, valamint a színt, ha feliratot szeretne kapni.

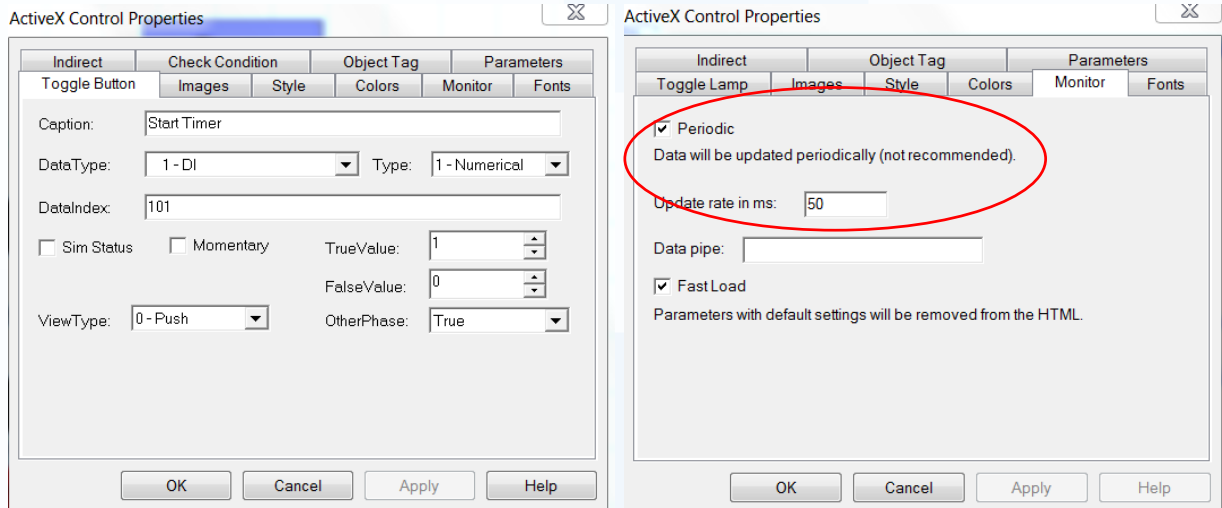


Gombok és lámpák

A gombokhoz az Adattípust 1 - DI-re kell kapcsolnia a ToggleButton alatt, és az Adatmutatót (DataIndex) a gomboknak megfelelően. (101 az első, 103 a harmadik)

A gombokhoz feliratot is kell tennie. (Indítási időzítő, Stop időzítő, Visszaállítás időzítő)

Meg kell változtatnia a monitor beállításait is. Jelölje be az Időszakos (Periodic) jelölőnégyzetet, és változtassa meg a frissítési sebességet 50 ms-ra. Ezzel elkerülhető, hogy a fények úgy tűnjenek, mintha valójában nem órák lennének.

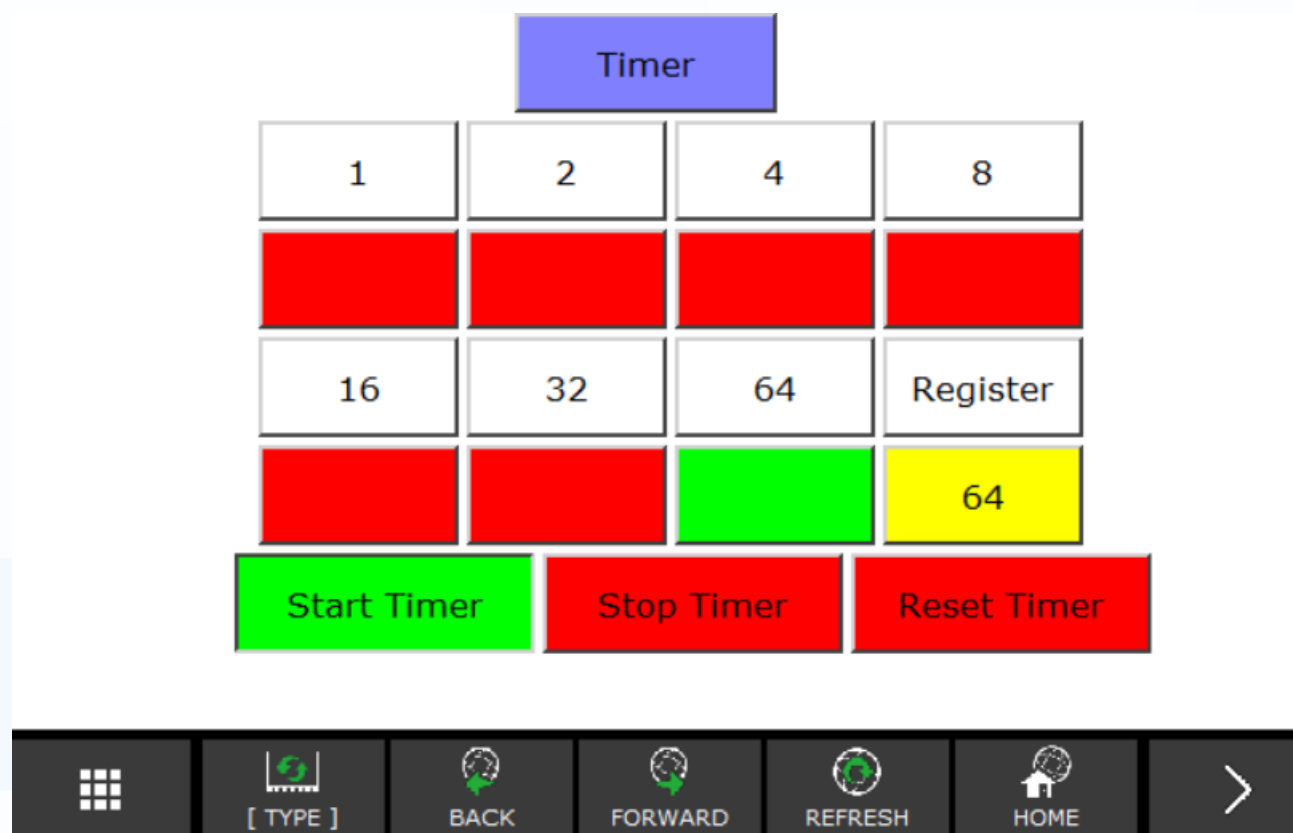


Gombok és lámpák

Mentse el az oldalt, töltsse fel a robotra, és állítsa be a böngészőbe. Indítsa el az I_O_Timer programot, majd váltson át a weboldalra, és ellenőrizze, hogy a be- és kimenetek működnek-e. (kapcsoló lámpa, kapcsoló gomb)

Ahhoz, hogy ez működjön, a DI101-et 103-ra kell állítania Sim-re, mert különben a robot nem változtathatja meg a bemenetet tényleges bemenet nélkül.

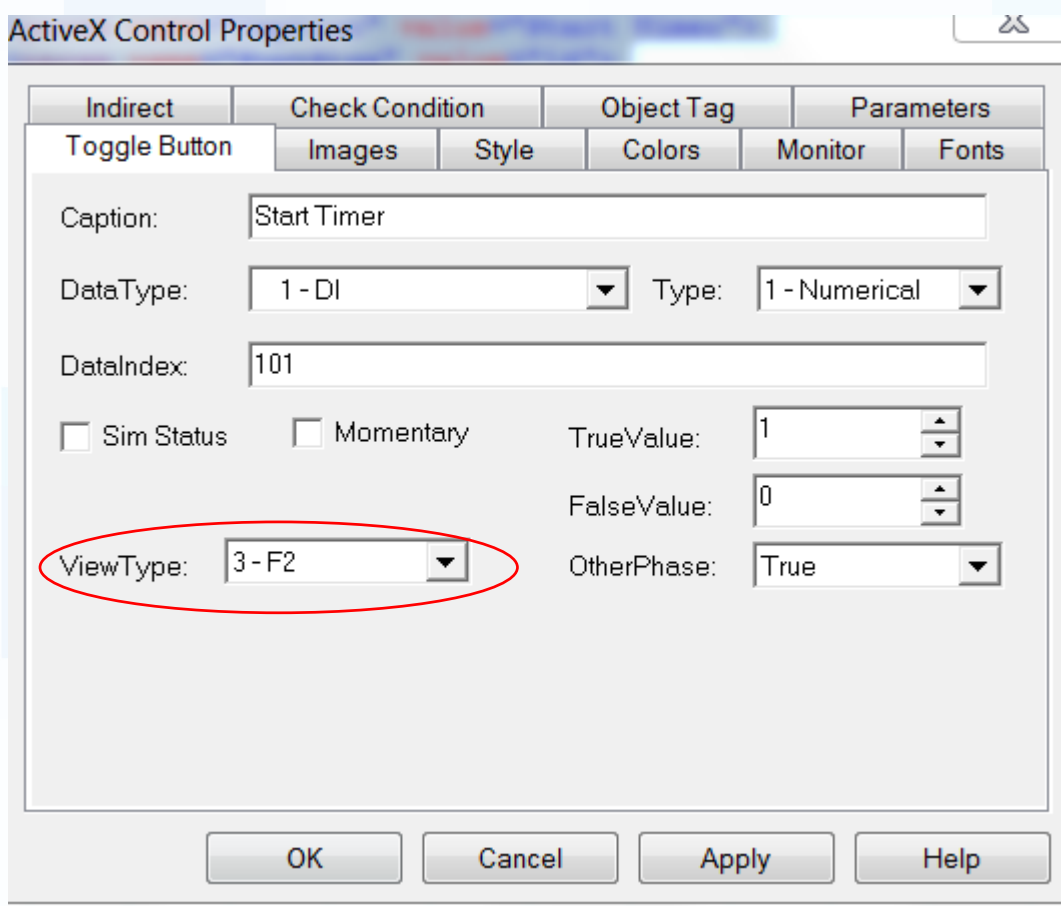
Bővebb információk a bemenetek szimulációjáról az R-30iB_Basic_Operator_Manual section 6.4.2 fejezetben. "Simulated I/O" és 6-11 fejezet "Simulated input/output".



Gombok és lámpák

A művelet megkönnyítése érdekében lehetőség van egyes gombok összekapcsolására az F2-F5 gombokkal.

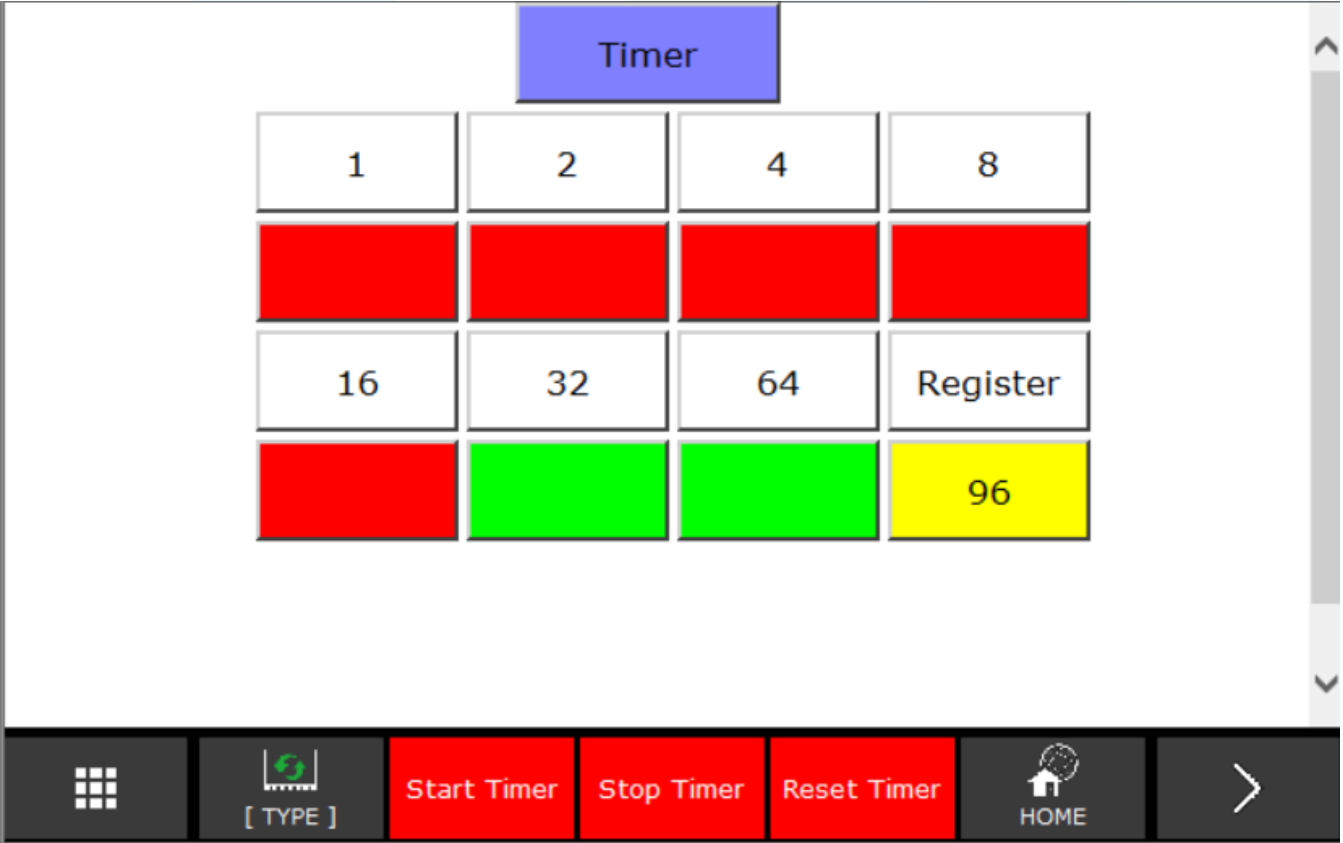
Ehhez el kell mennie a gombok beállításaihoz, és a Nézet típust 0-Push-ról F-re kell váltania (F2 például a Start Timer-hez stb.)



Gombok és lámpák

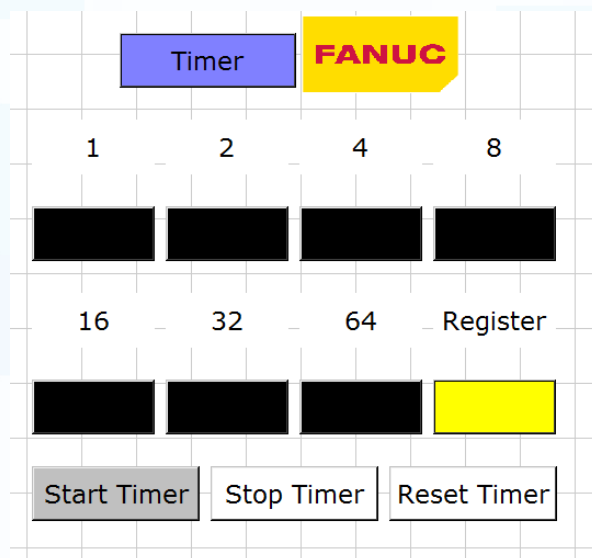
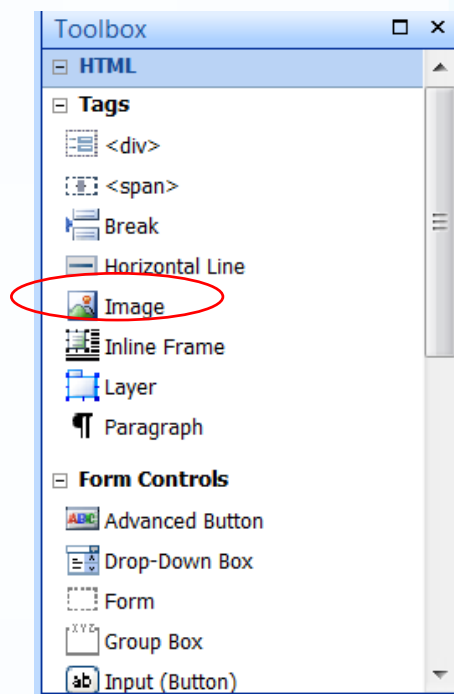
Mentse a munkáját, töltsse fel a robotra, majd próbálja újra.

A beállított gombok mostmár a Toolbar-on is elérhetőek. .



Képek

Lehetősége van képek hozzáadására az oldalára.
Az egyszerű Image parancs NEM szerepel az ActiveX Command listában, hanem az eszköztár jobb oldalán található meg.



A képbeállításokban böngészhet a képek között, és beillesztheti őket.

FONTOS

Ügyeljen arra, hogy csak .img, .gif és .jpg képeket adjon meg, mert a vezérlő nem tudja olvasni a többi formátumot.

Képek

Az oldal feltöltéséhez az eljárás közel azonos a korábbiakkal, csak annyit kell tennie, hogy a képet az UD1-re is fel kell töltenie, és fel kell töltenie az oldalt is. (az * összes fájl feltöltésével nem kell aggódnia, hogy valami nem kerül feltöltésre)

Állítsa be újra az oldalt, törölje a gyorsítótárat, és a képnek most az oldalán meg kell jelennie. (Ha nem változtatta meg az oldal nevét, akkor nem kell újra beállítania az oldalt, csak törölje a gyorsítótárat, lépjen vissza, és az új oldalt kell látnia.



Lap meghívása programból

Most, hogy van egy szép felületünk, azt akarjuk, hogy programunk az indítás után azonnal jelenítse meg ezt a felületet. A vezérlőben már van egy DSP_WEBP () nevű program, amely megjeleníti a kiválasztott weboldalt. Az oldal száma az a szám, amely az oldal előtt van a Type menüben.

A jelenleginek elméletileg a 6-os számmal kell rendelkeznie. Vegyen fel egy sort a program elejére, és hívja a DSP_WEBP (6) címet.

```
I_O_TIMER 1/30
1: CALL DSP_WEBP(6)
2: !Reset the Timer
3: R[34:Timer]=0
4: LBL[2]
5: !DI101 starts timer
6: IF DI[101:Switch 1]=ON,
: JMP LBL[1]
7: !DI102 stops program
8: IF DI[102:Switch 2]=ON,
: JMP LBL[5]
9: !DI103 resets timer
```



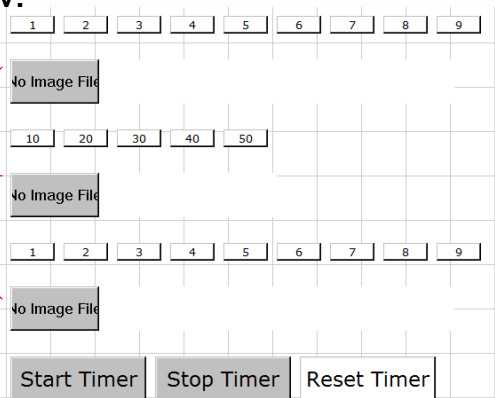
Multi Control

Ez a gyakorlatban leírt utolsó funkció. A Multi Control képnek tekinthető, de változhat a regiszterek, bemenetek vagy kimenetek különböző értékeinél. Egy Multi Control segítségével maximum 10 különböző kép jeleníthető meg.

Ezt a Multi Control funkciót most egy folyamatjelző sáv elkészítéséhez akarjuk felhasználni, amely megmutatja a Stopper előrehaladását.

De első dologra a különböző hosszúságú folyamatjelzőkre van szükségünk. Javasoljuk, hogy ezeket MS Paint alkalmazásban készítse el, kezdve egy 50x50px-es dobozzal, amely akár 500x50px-ig terjed, 50px lépésenként. Ne felejtse el olyan fájlokat létrehozni, amelyek olvashatók a vezérlő számára. Ezután helyezzen el 3 többszörös vezérlőt az oldalra, és tegye a címkéket a többszörös vezérlők fölé, hogy a felhasználó megértse, mit jelent a folyamatjelző sáv.

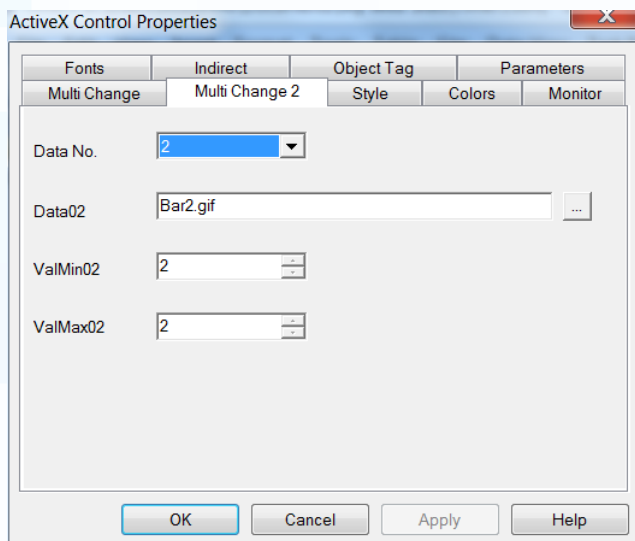
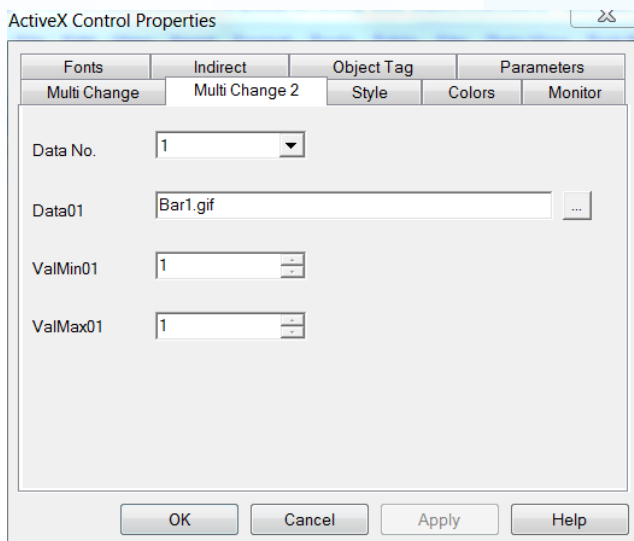
Multi Controls



Multi Control

Állítsa be a adat típusát (DataType) Numeric Register-re és a DataIndex-et 50, 51 és 52-re. (Később ezeket használjuk a programokban)

Ezután a Multi Change-en, állítsa be a Data Nbrs-t 1-9-ig.



Ne felejtse el ezeket beállítani minden Multi Control esetén. Az 1. számú adatnál az 50x50px sávot, a 2. számhoz az 50x100px stb. kell beállítani.

Ami a ToggleLamp vezérlőket illeti, a monitor beállításait ugyanúgy módosítsa, mint a lámpáknál.

És győződjön meg arról, hogy a Multi méretei valóban elég nagyok-e ahhoz, hogy elférjenek rajta a folyamatjelző sávok (50x500px)



Multi Control

Ne felejtse el összekapcsolni a gombjait a megfelelő DI-kkel. Most már csak el kell mentenie és feltöltenie az oldalt. Ne feledje, hogy fel kell töltenie a Paint-be létrehozott folyamatjelző sávokat (képeket) is.

Ez a képernyő egyelőre nem fog működni az aktuális programmal. Indíthatja, leállíthatja és visszaállíthatja az időzítőt, de mivel a Multi parancsokhoz beállított regiszterekben nincs változás, ezért nem történik semmi.

Készítsen másolatot az I_O_TIMER-ről az I_O_TIMER_PROG_BAR névvel. Próbálja megváltoztatni a programot úgy, hogy az R[50] -en 1-től 9-ig, az R[51]-en (10-50) minden másodperc tizedik és R[52]-nél percenként adja ki a másodperceket.

Tipp: Meg kell változtatnia a REGISTER_TO_BINARY programot olyan programra, amely percekre, tíz másodpercekre és másodpercekre bontja a regisztert, majd a megfelelő regiszterbe helyezi őket.

Ha nem tudja, hogyan kell elkészíteni ezt a programot, akkor menjen a függelékbe, ahol egy magyarázattal ellátott mintaprogramot készítettünk.

Most, amikor létrehozta a programot, hívja meg az oldalát a program elején (általában a 7. számnak kell lennie).

Próbálja ki a programot, és változtassa meg, ha úgy tetszik.



Összefoglalás

Ebben a gyakorlatban láthatta az alapvezérlőket, amelyeket testreszabott weboldalak létrehozásához használhat az iPendanton. Önnek képes lehet saját weboldalak létrehozására, mentésére, feltöltésére és megjelenítésére.

Tudhatja, hogyan hívhat meg egy adott weboldalt egy program során.



Appendix

Main Program

```
I_O_TIMER_PROG_BAR
1: 'Call user defined webpage
2: CALL DSP_WEBP(7)
3: 'Reset the Timer
4: R[34:Timer]=0
5: LBL[2]
6: 'DI101 starts timer
7: IF DI[101:Switch 1]=ON, JMP LBL[1]
8: 'DI102 stops program
9: IF DI[102:Switch 2]=ON, JMP LBL[5]
10: 'DI103 resets timer
11: IF DI[103:Switch 3]=ON, JMP LBL[4]
12: 'Wait for not overloading Processor
13: WAIT .10(sec)
14: 'Jump to beginning
15: JMP LBL[2]
16: 'Start the Timer
17: LBL[1]
18: WAIT 1.00(sec)
19: 'Timer Value increased
20: R[34:Timer]=R[34:Timer]+1
```

```
I_O_TIMER_PROG_BAR
16: 'Start the Timer
17: LBL[1]
18: WAIT 1.00(sec)
19: 'Timer Value increased
20: R[34:Timer]=R[34:Timer]+1
21: 'Transform Reg to Prog Bar
22: CALL REG_TO_PROG_BAR
23: IF DI[102:Switch 2]=ON, JMP LBL[5]
24: IF DI[101:Switch 1]=ON, JMP LBL[2]
25: JMP LBL[1]
26: 'Reset Timer
27: LBL[4]
28: R[34:Timer]=0
29: R[50:1sec]=0
30: R[51:10sec]=0
31: R[52:1min]=0
32: JMP LBL[2]
33: 'End of Program
34: LBL[5]
[End]
```

Sub Program

```
REG_TO_PROG_BAR
1: 'Set Registers
2: R[55:1minP]=0
3: R[54:10secP]=0
4: R[53:1secP]=R[34:Timer]
5: LBL[1]
6: 'If Timer over 1 minute
7: IF R[53:1secP]>=60, JMP LBL[2]
8: JMP LBL[3]
9: 'Minute Reg+1, Second Reg-60
10: LBL[2]
11: R[55:1minP]=R[55:1minP]+1
12: R[53:1secP]=R[53:1secP]-60
13: 'Timer maybe still > 1 minute
14: JMP LBL[1]
15: LBL[3]
16: 'If Timer over 10 seconds
17: IF R[53:1secP]>=10, JMP LBL[4]
18: JMP LBL[5]
19: '10 sec Reg+1, Second Reg-10
20: LBL[4]
```

```
REG_TO_PROG_BAR
9: 'Minute Reg+1, Second Reg-60
10: LBL[2]
11: R[55:1minP]=R[55:1minP]+1
12: R[53:1secP]=R[53:1secP]-60
13: 'Timer maybe still > 1 minute
14: JMP LBL[1]
15: LBL[3]
16: 'If Timer over 10 seconds
17: IF R[53:1secP]>=10, JMP LBL[4]
18: JMP LBL[5]
19: '10 sec Reg+1, Second Reg-10
20: LBL[4]
21: R[54:10secP]=R[54:10secP]+1
22: R[53:1secP]=R[53:1secP]-10
23: JMP LBL[3]
24: 'Set final Registers
25: R[50:1sec]=R[53:1secP]
26: R[51:10sec]=R[54:10secP]
27: R[52:1min]=R[55:1minP]
[End]
```





Gyakorlatok 11

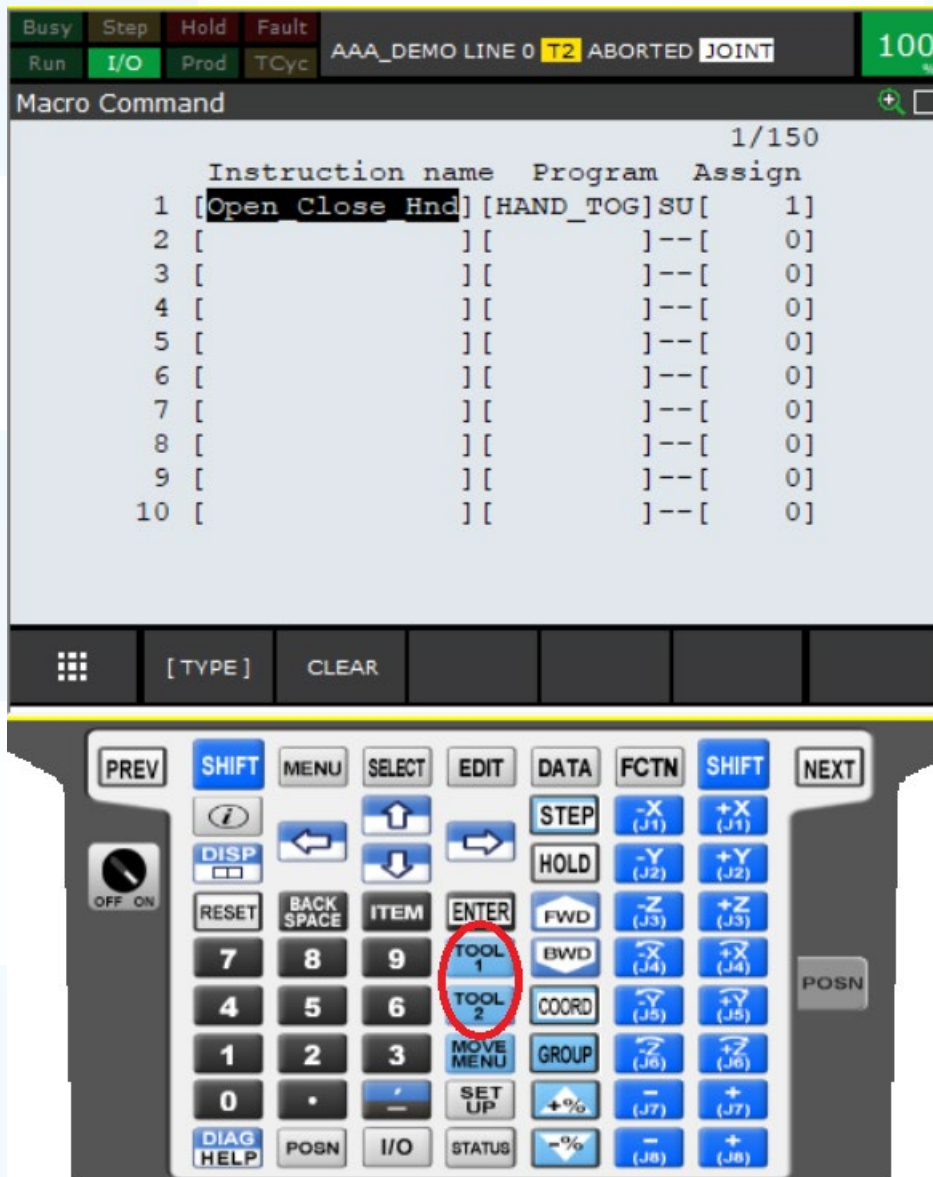
Macro



Bevezető

Ebben a gyakorlatban egy makró létrehozását és beállítását szeretnénk bemutatni.

A makró egy olyan program, amelyet a „Felhasználói” gombokkal (User Key) hívhat be, de a makró más programokban is meghívható.



Gyakorlat 11

Macro

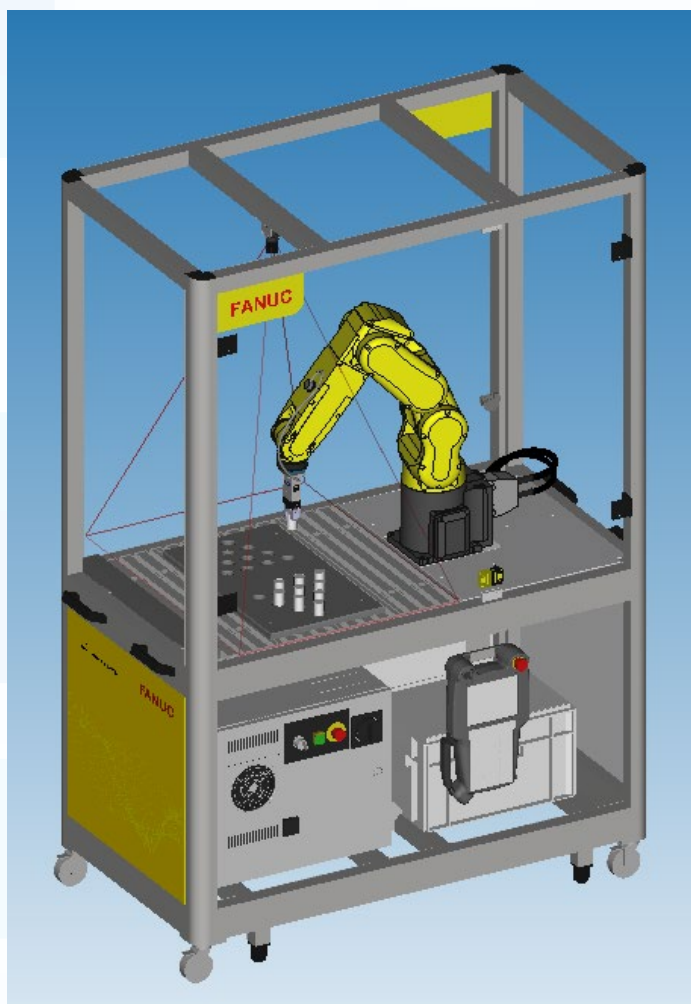
Tartalom

Bevezető	-	3
Eszköz	-	5
Macro példa	-	6
Macro programozása	-	7
Macro beállítása	-	8
Manual Function Macro	-	14
Macro meghívása	-	15



Eszköz

Ehhez a gyakorlathoz szükséges:
Oktatócella



Macro példa

Az Oktatócella már rendelkezik makróval. Nyomja meg a „SHIFT” és az „TOOL 1” gombot. Nem kell nyugtáznia a hibákat, és nem kell megnyomnia a biztonsági vészkapcsolót. Láthatja, hogy a megfogó becsukódik vagy kinyílik.



Ugyanazt a programot fogjuk programozni és felhasználjuk a „TOOL 2” billentyűhöz.

Később beprogramozhatja saját makróját.



Macro programozása

A makrót ugyanúgy programozzák, mint egy normál programot. Nyomja meg a „SELECT” gombot, és hozzon létre egy új programot. Nevezze el a programot „TOGGLE_HAND”-nek. A programnak ki kell nyitnia a megfogót, ha zárva van, és bezárni a megfogót, ha nyitva van.



The screenshot shows the FANUC macro editor interface. At the top, there are status buttons: Busy, Step, Hold, Fault, Run, I/O, Prod, and TCyc. The current program name is TOGGLE_HAND LINE 0, and it is in an ABORTED state. The speed is set to 100%. The macro code is displayed as follows:

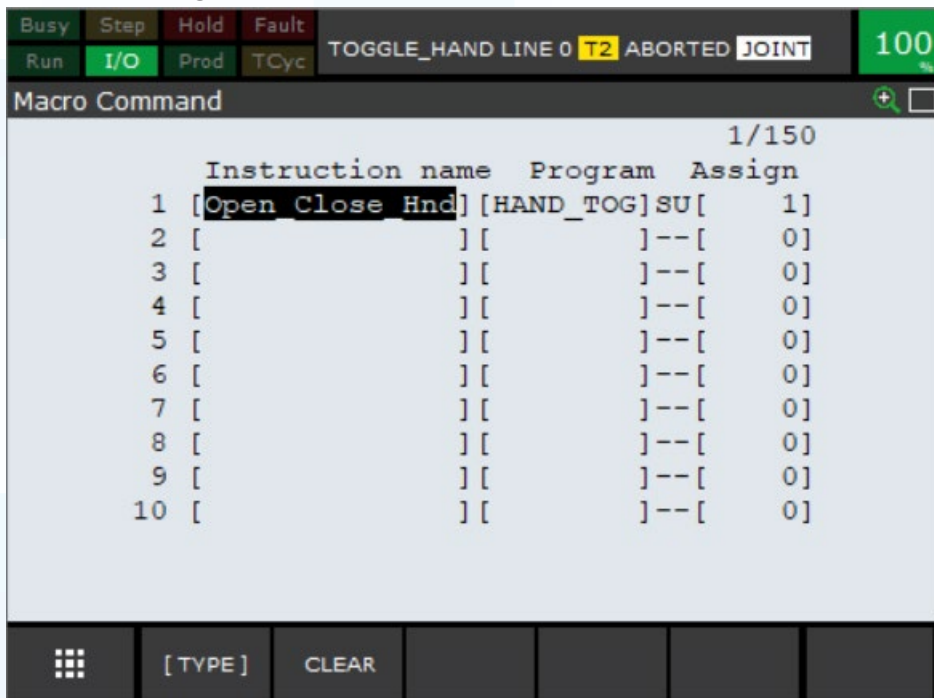
```
TOGGLE_HAND 6/6
1: IF (RO[7:Open Gripper]=ON) THEN
2: RO[7:Open Gripper]=OFF
3: ELSE
4: RO[7:Open Gripper]=ON
5: ENDIF
[End]
```

At the bottom, there are navigation buttons: a grid icon, POINT, TOUCHUP, and a right arrow.



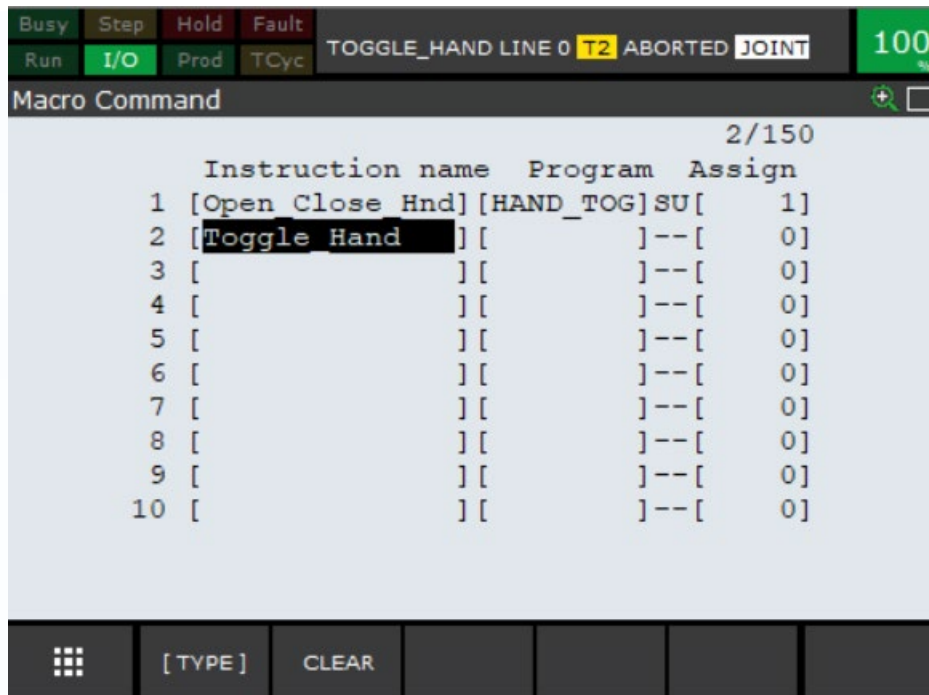
Macro beállítása

A makró beállításához el kell mennie a makró képernyőre. Nyomja meg a MENU, majd válassza a SETUP almenüt, majd válassza a „MACRO” lehetőséget.



Az első sorban látható a már beállított makró. Ugorjon a második sorra, és adja meg a makrónak az utasítás nevét, ebben az esetben Toggle_Hand nevet adjuk (az utasításnak nem feltétlenül azonos névvel kell rendelkeznie, mint az általunk használt program).

Macro beállítása

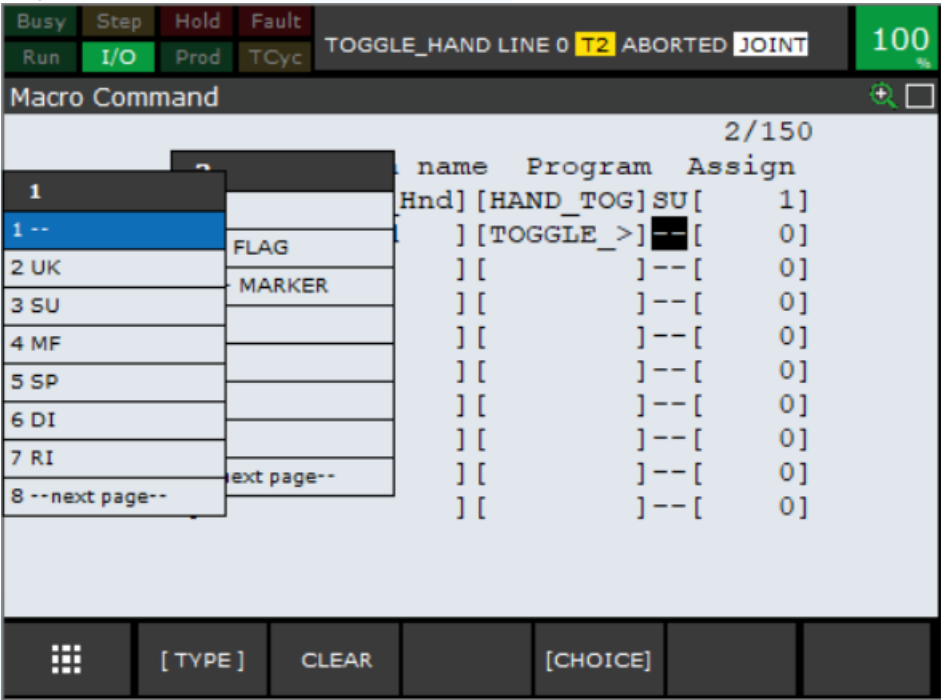


Most ki kell választania egy programot. Menjen a program oszlopba és nyomja meg az F4 „Choice” gombot. Válassza ki a használni kívánt programot. Keresse meg a „TOGGLE_HAND” programot és nyomja meg az „ENTER”-t.

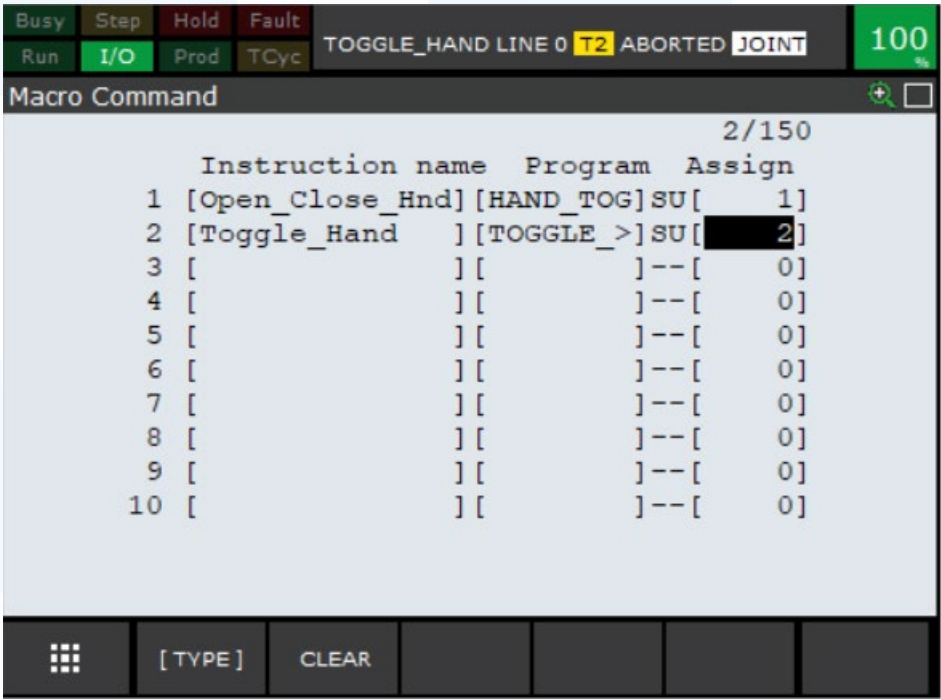


Macro beállítása

Most ki kell választania a makró típusát, menjen a „-” jelhez és nyomja meg az F4 „Choice” gombot



Válassza az „SU” lehetőséget, és a hozzárendelt számhoz írja be a 2. számot.



Macro beállítása

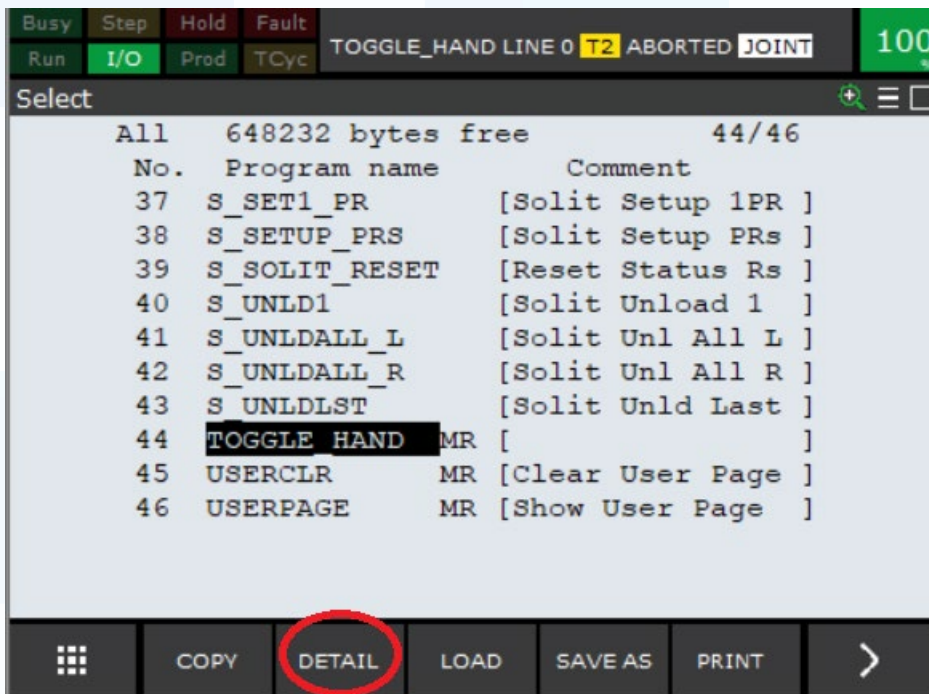
Bővebb részletek az Operaor Manual 9.1.1 fejezetben. "Setting Macro Instructions".

Most befejeztük a makró beállítását. Nyomja meg a „SHIFT” és a „TOOL 2” gombot.

Mint láthatja, semmi sem történik. Most meg kell nyomnia (és tartania) a DEADMAN kapcsolót, majd megnyomhatja a „SHIFT” és a „TOOL 2” gombokat.

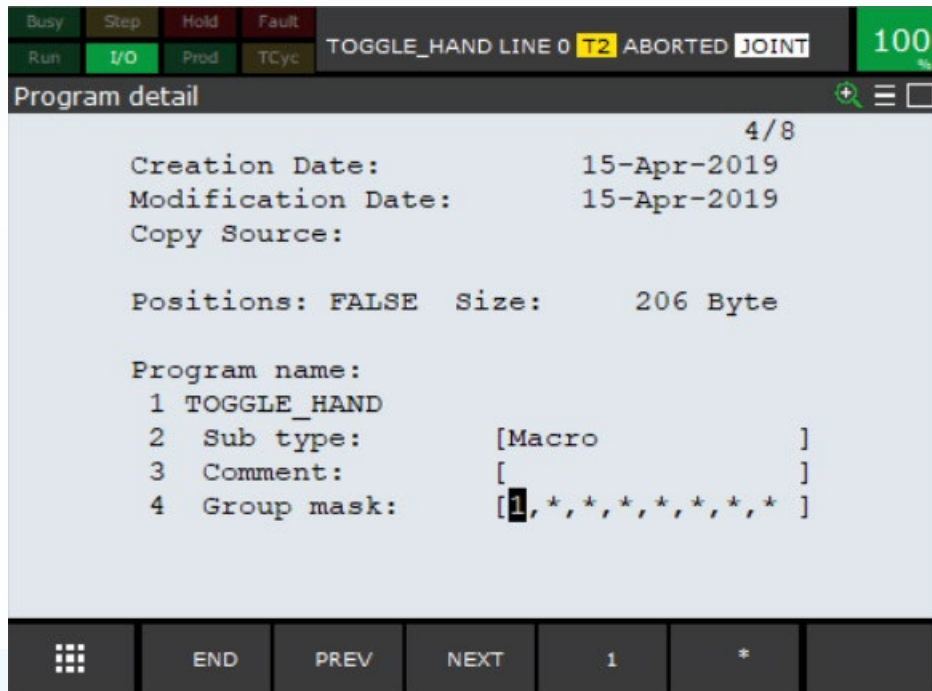
Ennek megváltoztatásához módosítanunk kell egy beállítást a „TOGGLE_HAND” programban. Az általunk módosított beállítás csak akkor változtatható meg, ha nincs mozgásutasítás a programban.

Lépjen a programhoz, nyomja meg a „NEXT” gombot és nyomja meg az F2 „Részlet” gombot.

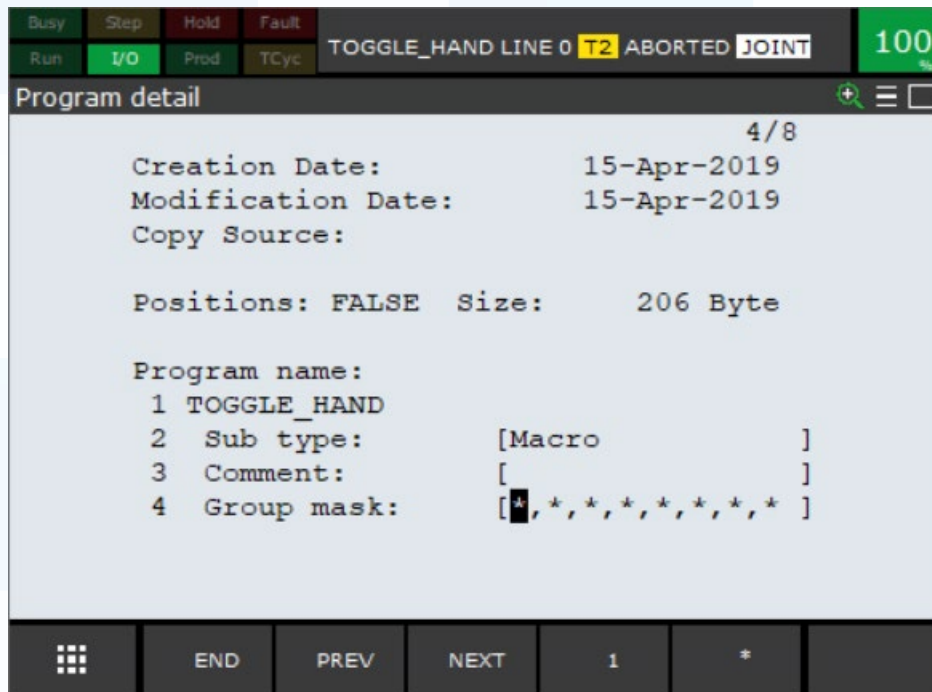


Macro beállítása

Lépjen a 'Group Mask' beállításhoz.



Most meg kell változtatnia az 1-et csillagra. Vigye a kurzort az 1-re, és nyomja meg az F5 gombot a * -ra változtatásához.



Macro beállítása

Nyomja meg az F1 „Vége” gombot, hogy visszatérjen a kiválasztási képernyőre. Most a makrónak a terv szerint kell működnie. Nyomja meg a „SHIFT” és a „TOOL 2” gombot, és nézze meg, hogyan záródik és nyílik a megfogó.

Mindegyik „Felhasználó” billentyűnél csak egy makró használható.

Bővebb információk a makrók futtatásáról az Operator Manual 9.1.2 fejezetben “Executing Macro Instructions”.



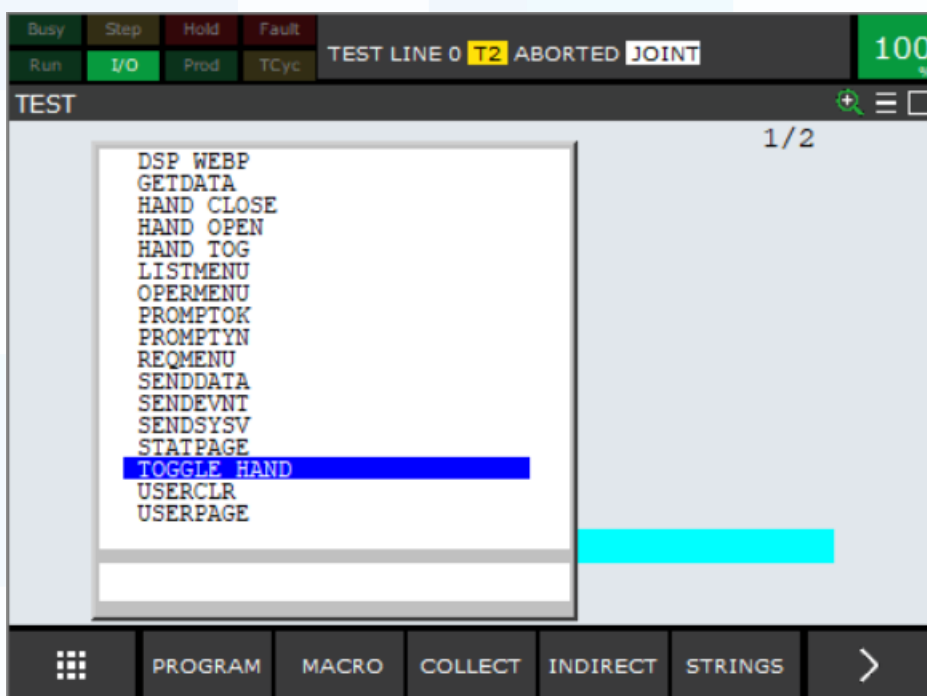
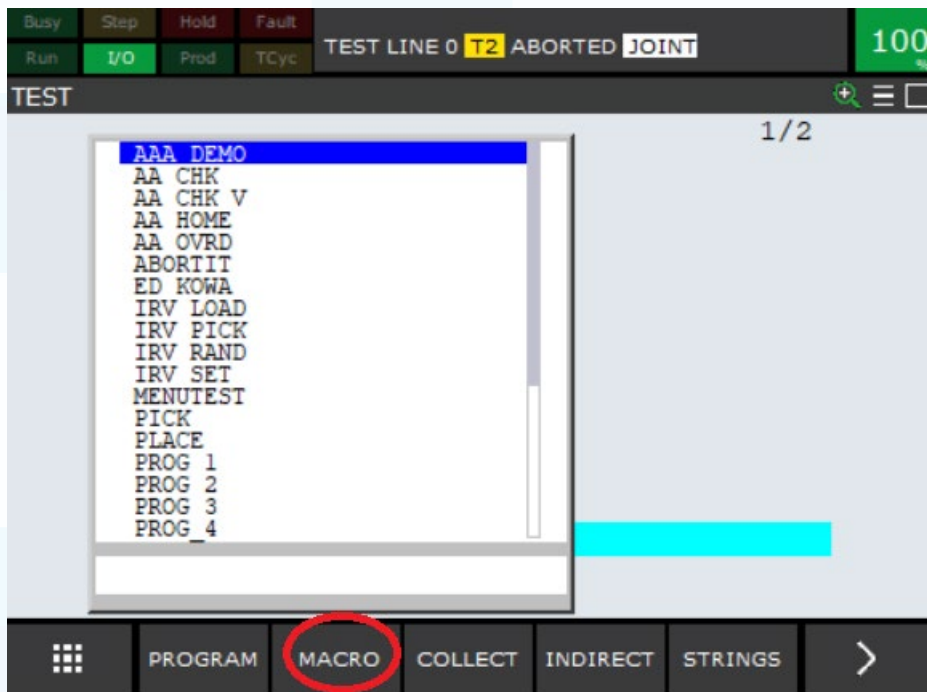
Manual Function Macro

Az „SU” makrótípus helyett használhatja az „MF” szót is. Az ‘MF’ makrók használatához csak meg kell nyomnia a MENU gombot, majd válassza a MANUAL FUNCTION menüpontot. Választania ki a kívánt makrót, majd nyomja meg a „SHIFT” és az F3 „Exec” gombot.



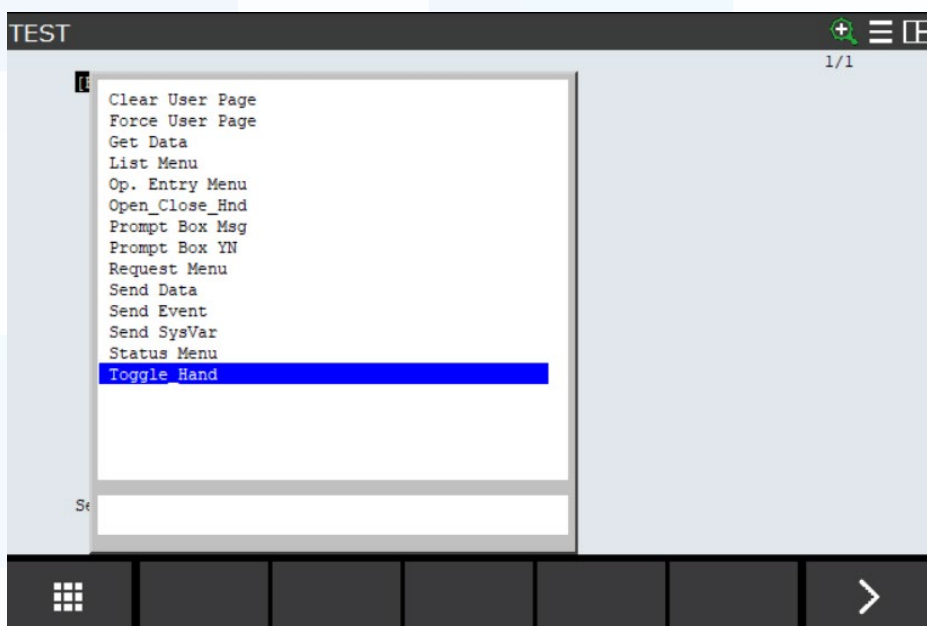
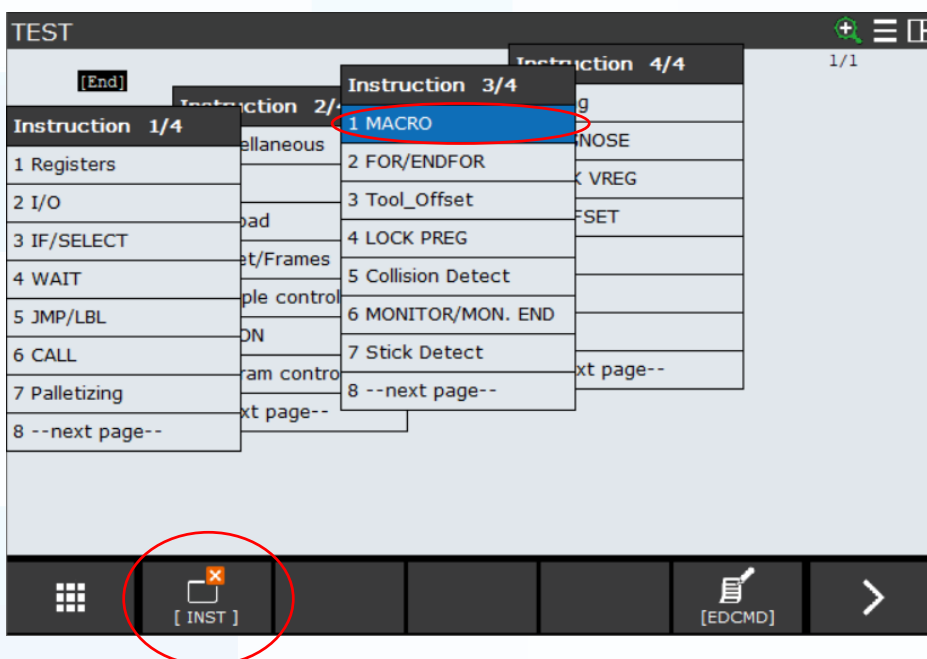
Macro meghívása

Pontosan úgy hívhat makrót egy másik programban, mint bármely más programot. A különbség az, hogy meg kell nyomnia az F2 billentyűt a „makró” megadásához a programválasztó menüben.



Macro meghívása

Vagy ha beállított egy makrót a makróképernyőn, akkor az F1 'Inst' megnyomásával hívhatja meg, lépjen a 3. oldalra, válassza a „MACRO” lehetőséget, és válassza ki a kívánt makrót.





Gyakorlat 12

Menü segédprogram

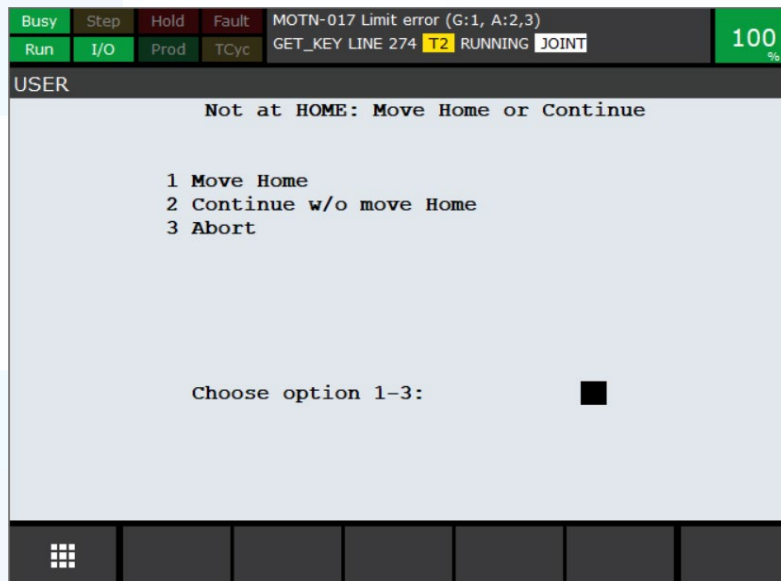
2. lap



Bevezető

A menü segédprogram olyan makrók gyűjteménye, amelyeket a programokban használhat. A makrók azért különlegesek, mert olyanok, mint a program előugró ablakai, ahol a folytatáshoz meg kell nyomni egy gombot.

Ebben a gyakorlatban 3 különböző típust fogunk látni, a PROMPTOK, a PROMPTYN és a LISTMENU.



Gyakorlat 12

Menü segédprogram

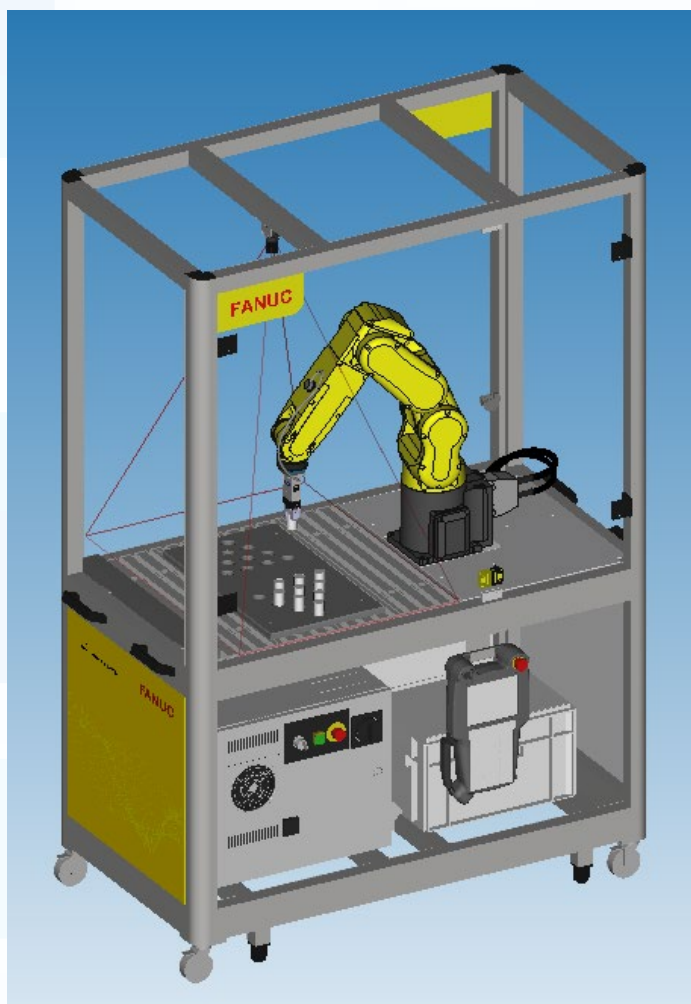
Tartalom

Bevezető	-	3
Eszköz	-	5
Menü segédprogram képernyő	-	6
PROMPTOK	-	7
PROMPTYN	-	12
LISTMENU	-	16



Eszköz

Ehhez a gyakorlathoz szükséges:
Oktatócella



Menü segédprogram képernyő

A Menü segédprogram képernyőhöz megnyitásához nyomja meg a „MENU” gombot, majd válassza a „6 Setup”, majd a „3 Menu Utility” menüpontot.

Itt létrehozhat és módosíthat néhány ilyen „segédprogramot”.



PROMPTOK

A PROMPTOK makró egy egyszerű előugró ablak megjelenítésére szolgál, ahol a felhasználónak a folytatáshoz meg kell nyomnia az "ENTER" gombot.

Lépjen a 'Prompt Box Msg' sora, majd nyomja meg az F2 'Részletek' gombot.

Új PROMPTOK létrehozásához nyomja meg az F3 'Create' gombot. A meglévő PROMPTOK módosításához nyomja meg az F2 „Részlet” gombot.



PROMPTOK

A PROMPTOK létrehozásához nyomja meg az F3 'Létrehoz' gombot. A rendszer megkérdezi, hogy mi legyen a száma (minden szám csak egyszer létezhet, ez az azonosító szám), írja be a 3-as számot (ha még nem vette fel).

ITEM	Menu no	Menu Name
1	1	NotAtPerch
2	2	VisFail

Enter menu number (1-99): 3

Most Menü nevet kell adnia neki, ebben a példában ezt „Hello world” -nek hívjuk.

Prompt msg menu no: 3 1/6

Menu name: Hello world

Line Text

1

2

3

4

5

PROMPTOK

A PROMPTOK-ban 5 sort használhat. Minden sor csak a sor első 28 karakterét mutatja!

Példaként ezt a mondatot tesszük az első sorba: „Helló világ. Hogy vagy?”.

Amint láthatja, ott van a „>” szimbólum, és ez azt jelenti, hogy ez a maximálisan megjelenített karakterek száma, a többi nem fog látszani.

Nem kell minden sort kitöltenie az 5-ből.

A második sorba írja be a “Jól”, a harmadikba pedig ezt "És te?"



PROMPTOK

A PROMPTOK teszteléséhez nyomja meg az F3 'Test' gombot.



A képernyő megjelenik, és a folytatáshoz nyomja meg az „ENTER” gombot.

Láthatja, hogy nem fért ki az első sor.

Most megváltoztathatja az első sort a „Hello Világ. Hogy vagy?”-ra, amely már kifér a megjelenő ablakra.

További részletekért lásd az Operator Manual 36.1.1. fejezetet

A PROMPTOK törléséhez menjen vissza a menübe, lépjen a kívánt PROMPTOK-hoz, és nyomja meg a „NEXT” gombot, majd az F3 „Clear” gombot.



PROMPTOK

A PROMPTOK meghívásához egy programban használja a „CALL” utasítást. A menu kiválasztásához nyomja meg az F2 „Makró” gombot, majd válassza a „PROMPTOK” lehetőséget.



Most, hogy meghatározza, melyik PROMPTOK-ot akarja, nyomja meg az F4 „Choice” gombot, válassza a „2 Constant” lehetőséget, és írja be a 3-as számot. A PROMPTOK létrehozásának elején kiválasztott szám a PROMPTOK azonosítója, és a PROMPTOK meghívásához szükséges.

Most elindíthatja a programot, a PROMPTOK felugrik a Teach Pendant képernyőjén, és a felhasználónak a folytatáshoz meg kell nyomnia az „ENTER” gombot.

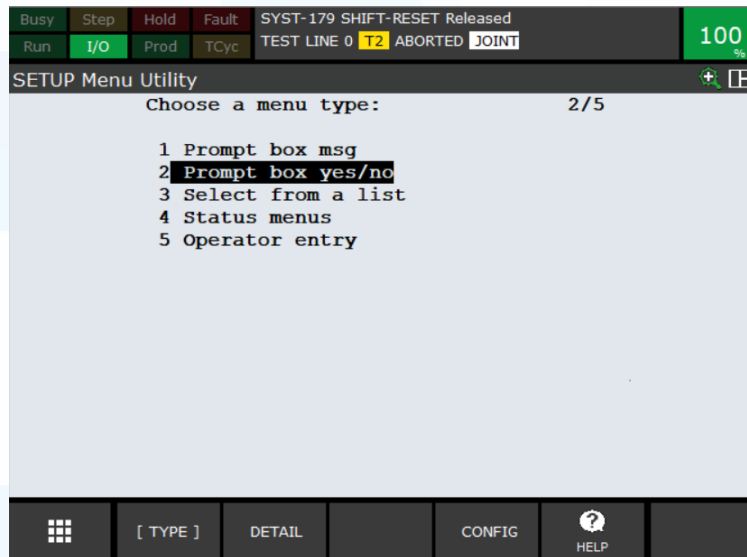
Meghívhatja a PROMPTOK-t úgy is, ha a „CALL” helyett a „MACRO” lehetőséget választja, majd a „Prompt Box Msg” lehetőséget választja.



PROMPTYN

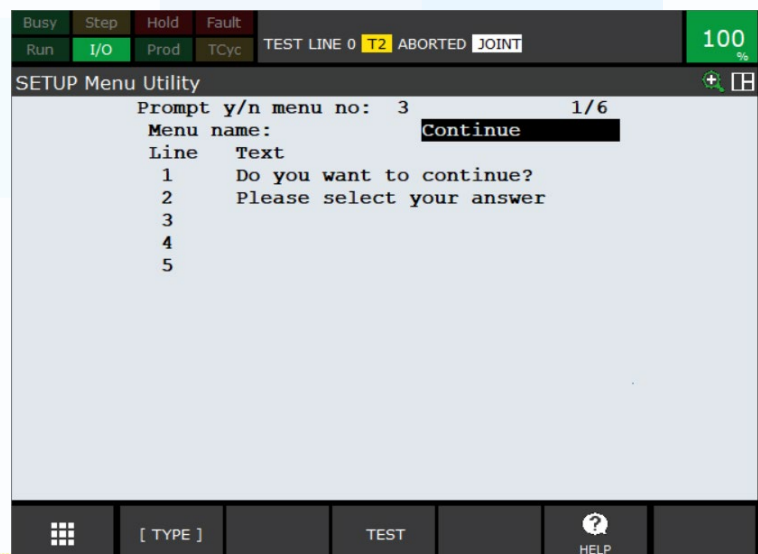
A PROMPTYN egy igen/nem kérdés, amelyet egy programban feltesznek.

Nyomja meg az F2 „Részlet” gombot a „Prompt box yes/no” elemen.



PROMPTYN létrehozásához nyomja meg az F3 „Létrehozás” gombot, és írja be a 3-as számot (ha még nincs ilyen).

Adjon neki menu nevet, és 2 sornyi szöveget írhat be. A sorok pontosan úgy működnek, mint a PROMPTOK-ban, csak 28 karaktert mutatnak.



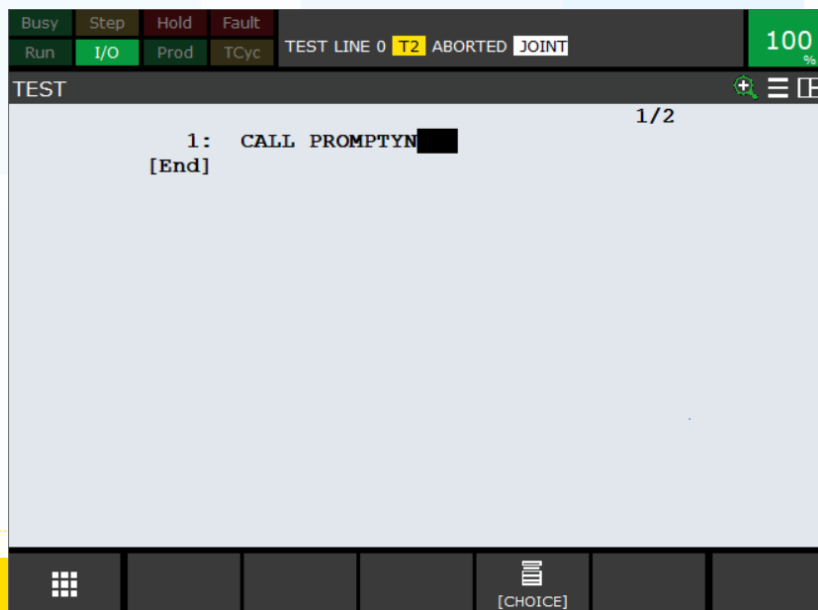
PROMPTYN

Nyomja meg az F3 'Teszt' gombot az ablak megjelenítéséhez. A billentyűkkel navigáljon a „NEM” vagy „IGEN” pontra, majd nyomja meg az „ENTER” gombot.



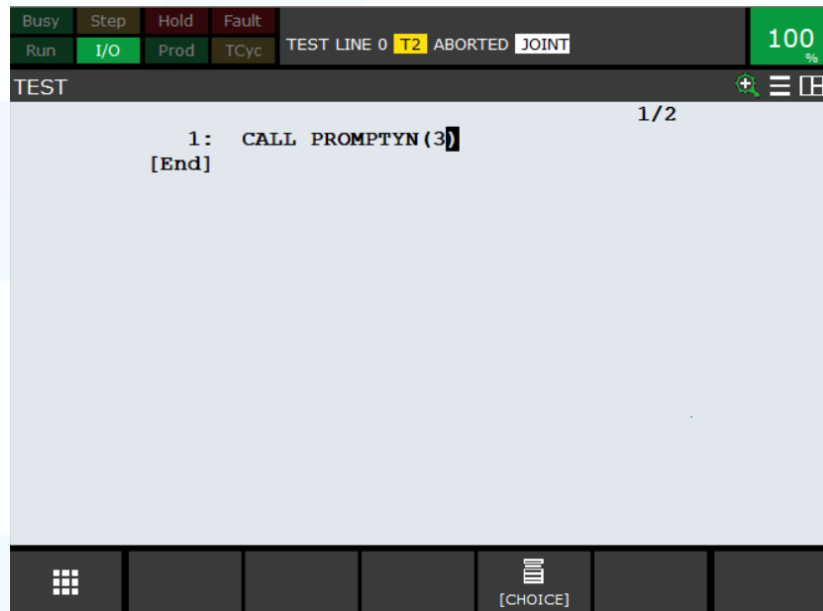
Bővebb részletek az Operator Manual 36.1.2 fejezetben. "Prompt Box YES/NO Menu".

Ha PROMPTYN-t szeretne meghívni egy programban, használja a „CALL” utasítást, és nyomja meg az F2 „Makró” gombot, és válassza a „PROMPTYN” lehetőséget.



PROMPTYN

Nyomja meg az F4 „Choice” gombot, válassza a „Constant” lehetőséget, és írja be a PROMPTYN azonosító számát, ebben az esetben 3.



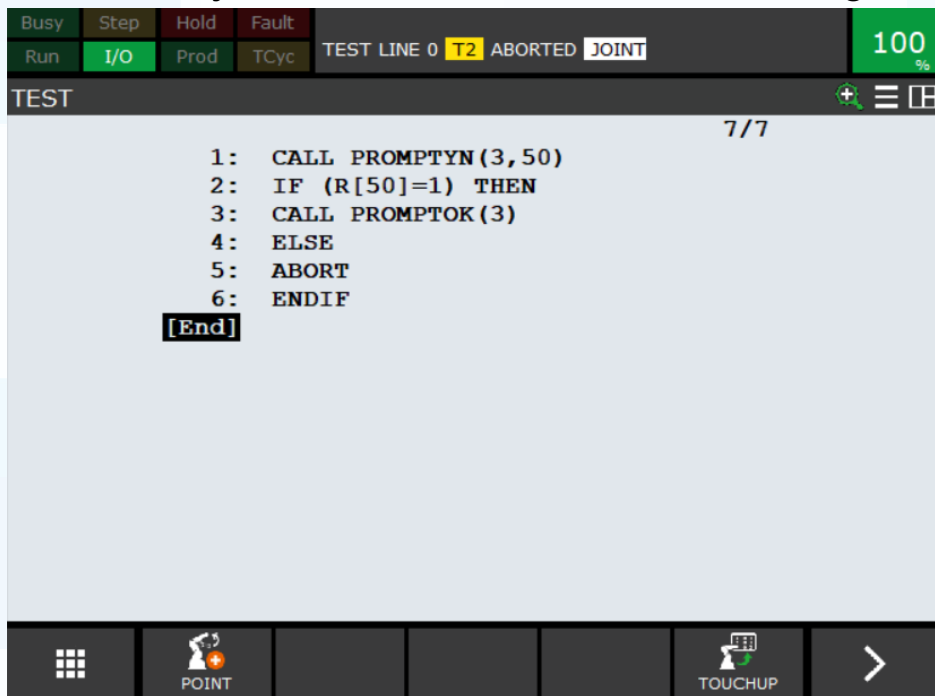
Mivel a PROMPTYN rendelkezik kimenettel, ezt a kimenetet egy regiszterbe kell mentenünk, ezért nyomja meg ismét az F4 „Choice” gombot, válassza a „Constant” lehetőséget, és írja be az 50 értékét [50].

Ha a válasz 'IGEN', akkor az R [50] = 1, különben 0 lesz.



PROMPTYN

Most az 'IF' funkcióval programozzuk, mi fog történni, amikor válaszolunk a PROMPTYN-re. Ha a válasz „IGEN”, akkor felhívjuk a korábban létrehozott PROMPTOK-ot, ha „NEM” megszakítjuk a programot. Az „ABORT” utasításhoz nyomja meg az F1 „INST” gombot, a következő oldalon válassza a "Program Control" elemet, majd válassza az „ABORT” lehetőséget.



Bővebb információk az ABORT-ról az Operator Manual 4.14.2 fejezetben. "Abort Instruction"

Most tesztelheti a programot. Mentse el ezt a programot, mert a következő példában fogjuk használni.



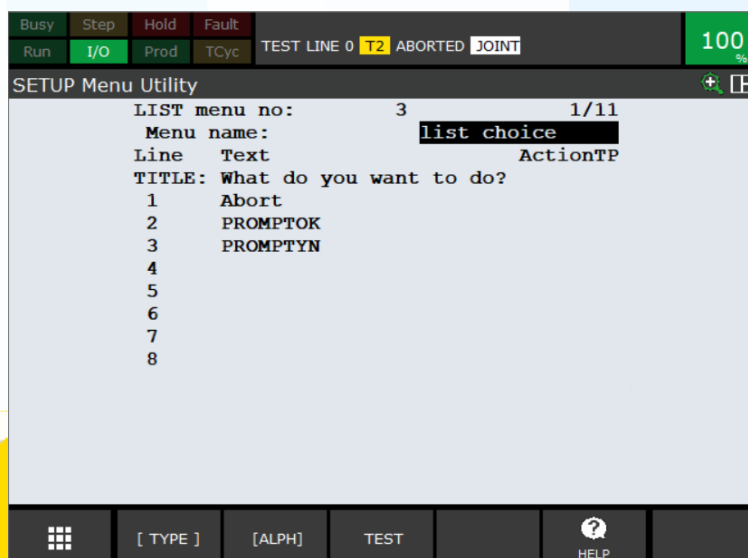
LISTMENU

A LISTMENU felsorol egy listát, ahol választhat egy maximum 8 elemű listából.

Menjen a menü segédprogram képernyőjére, és nyomja meg az F2 „Részlet” gombot a „Kiválasztás listából” (Select from a list) soron állva.



LISTMENU létrehozásához nyomja meg az F3 „Létrehozás” gombot, és adja meg a 3-as számot (ha még nincs ilyen). Adja a LISTMENU-nek a „lista választás” nevet, a cím pedig „Mit szeretnél csinálni?” és az első sor lesz az „ABORT” második „PROMPTOK” és a harmadik „PROMPTYN”. Minden sorban 28 karakter jeleníthető meg.

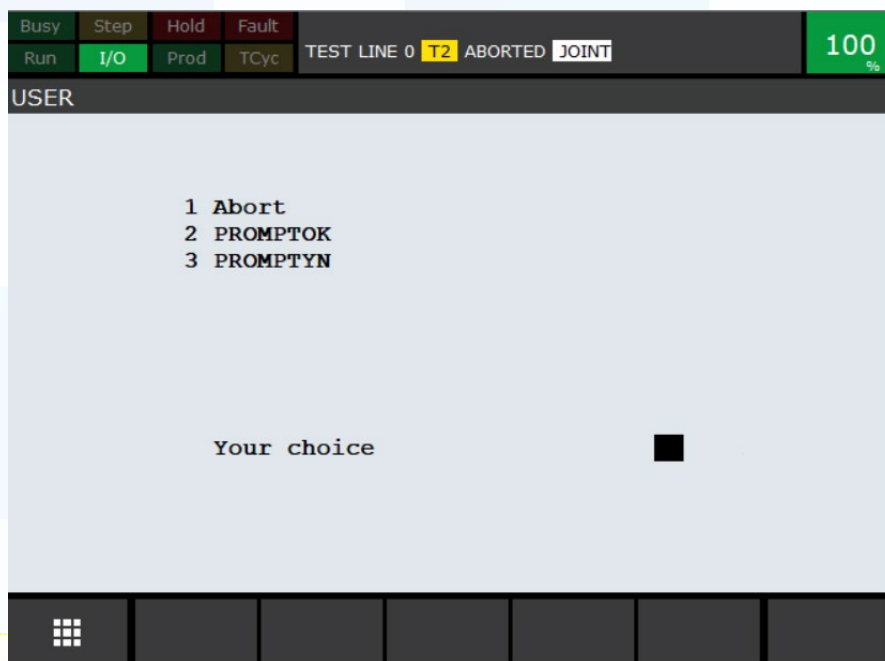


LISTMENU

Most görgessen le a „PROMPT” menüpontra, és írja be az „Ön választása” kifejezést. Ez ott fog megjelenni, ahol a felhasználó beírja a választást.



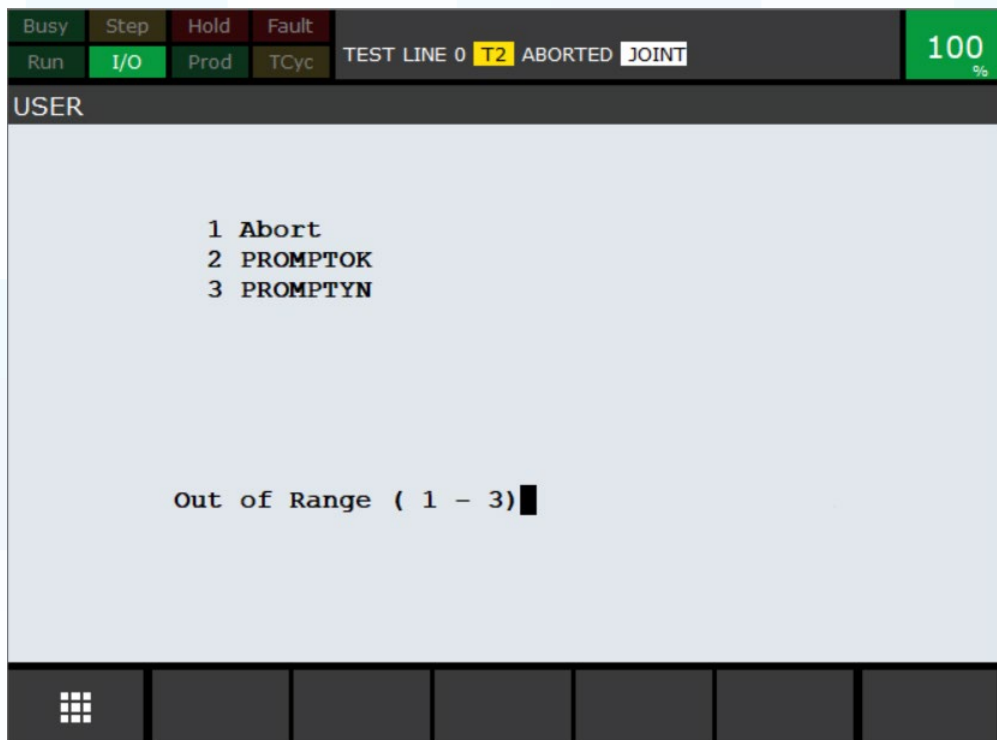
Most tesztelje a programot az F3 'Teszt' segítségével, és írja be a választott számot.



LISTMENU

Ha 3-nál nagyobb számot próbál megadni, a program jelzi, hogy „Out of Range (1 - 3)”. Ez annak biztosítása, hogy a felhasználó kövesse a listát, és a program programozottan futtatható legyen. Legfeljebb 8 különböző sort használhat arra, hogy választ kapjon.

Amint láthatja, a „Cím” nem jelenik meg, de később, amikor egy programban hívjuk, akkor megjelenik.



Bővebb részletek az Operator Manual 36.1.3 fejezetben. “List Menu”.



LISTMENU

A LISTMENU hívásához használja a „CALL” utasítást. Csakúgy, mint korábban, nyomja meg az F2 „Makró” gombot, amikor programot választ, és válassza a „LISTMENU” lehetőséget.

Most nyomja meg az F4 „Choice” gombot, válassza a „Constant” lehetőséget, és írja be a 3-as azonosítószámot. Ezután meg kell adnia egy regiszter számát, ahová a választás eredménye érkezik. R[51]. Nyomja meg ismét az F4 ‘Choice’, ‘Constant’ gombot, és írja be az 51-es számot.



LISTMENU

A programozáshoz használhatjuk az 'IF' vizsgálatot, de amikor sok választ kell összehasonlítanunk, a 'SELECT' vizsgálat gyorsabb (Minden választ egy lépésben hasonlít össze, és nem egy lépést a másik után, mint az 'IF' vizsgálat.)

Bővebb információk a SELECT használatáról az Operator Manual 4.7.4 fejezetben "Conditional Branch Instruction"

A „Select” vizsgálat beszúrásához nyomja meg az F1 „INST”, „IF / SELECT”, „ next page”, majd a „Select R [] =...” gombot. Írja be az R [] értékének az 51-es számot. Az első eredménynek a programot meg kell szakítania, így először hozzon létre egy másik programot a megszakítási utasítással, amelyet meghívhatunk. Vagy használja a JMP LBL [] utasítást.



LISTMENU

A második eredményhez lépjen a következő sorra, nyomja meg az F1 'INST', 'IF / SELECT' gombot. '8 következő oldal', majd '<választ> =....

Mivel továbbra is használjuk az R[51] -et, ezt nem kell újra tisztáznunk, és beírhatjuk a következő vizsgálatot, amely 2 a „...” -ba. Az általunk hívott programnak szüksége van egy „Argumentumra”, ezért válassza a „CALL program ()” lehetőséget, különben nem tudjuk megadni a PROMPTOK (3) számát.

Bővebb információk az Argumentumokról az Operator Manual 4.7.6 fejezetben “Arguments”



Most ugyanezt tesszük a harmadik vizsgálattal is, de a PROMPTYN-nel létrehozott programot fogjuk használni, ahol nincs szükségünk argumentumra.



LISTMENU



A program elkészült, és kipróbálhatja.

Most megpróbálhatja leprogramozni saját programjait a PROMPTOK, a PROMPTYN és a LISTMENU segítségével.

