

ABB RobotStudio Gyakorlat

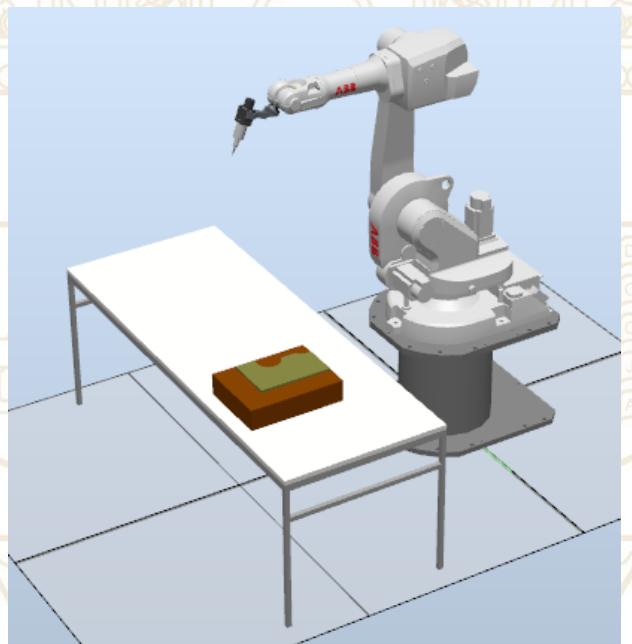
I. Mérési Feladat

Virtuális Robotcella Összeállítása

Mérés Célja:

A mérés célja, hogy a feladat megoldása során a hallgató megismerkedjen a RobotStudio menürendszerével, alapvető kezelő felületekkel, a mérés végén a hallgató képes legyen önállóan is új projektek létrehozására, megtanulja a szükséges gyártói dokumentációk kezelését és azok értelmezését.

A mérés során a hallgatónak az alábbi ábrán látható összeállítást kell elkészíteni a RobotStudio segítségével:



1. ábra Robot cella felépítése

A mérésről a hallgatónak jegyzőkönyvet kell késztenie, amelyben az egyes részfeladatok végén található kérdéseket kell megválaszolnia.

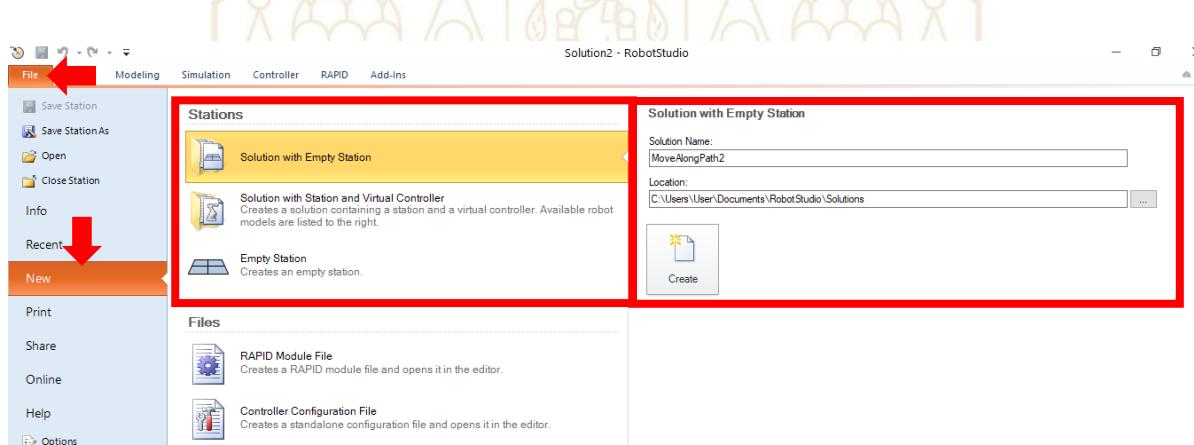
1. feladat – Új Projekt Létrehozása

Indítsa el a RobotStudiót! Ezt követően a szoftver automatikusan a **File>New** ablakot nyitja meg számunkra. Itt a **Stations** pont alatt tudunk új projektet (**Solution**) vagy önálló cellát (**Empty Station**) létrehozni.

Kattintson a **Solution with Empty Station** menüpontra, majd a képernyő jobb oldalán a **Solution Name** alatt nevezze el a projektet, illetve adja meg a mentés helyét is (**Location**)! (2. ábra)

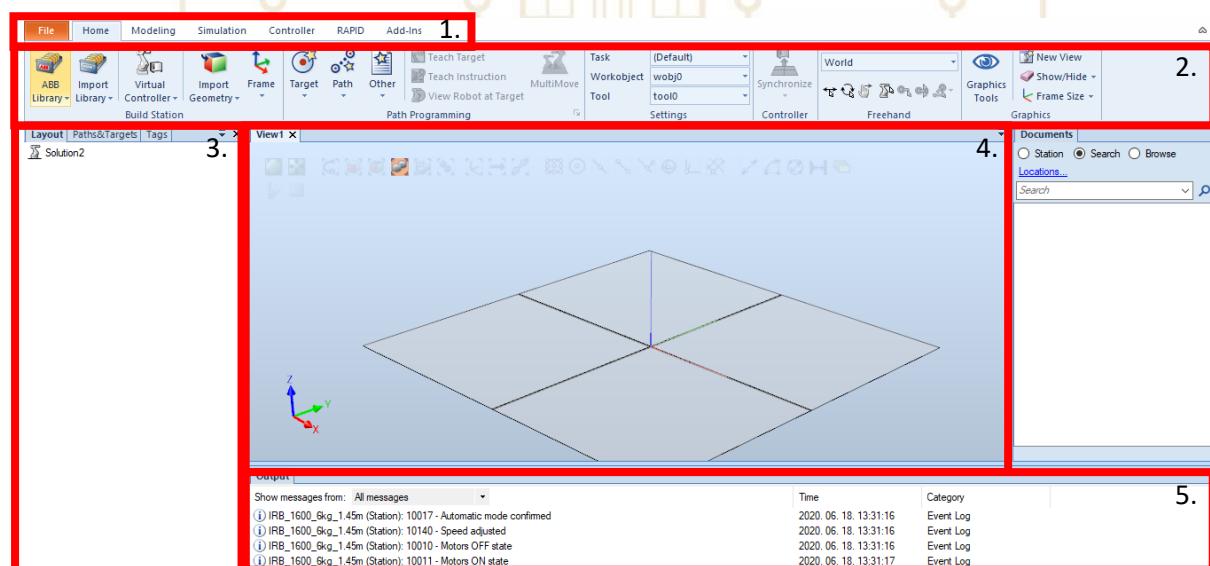
Figyelem!

Amennyiben az egyetemi labort használja (OE-BGK), ügyeljen rá, hogy a számítógép újraindítását követően a C meghajtó automatikusan törlődik, így a feladatát a D meghajtora mentse!



2. ábra Projekt létrehozása

Amennyiben a beállításokat elvégezte, kattintson a **Create** gombra. Az elvégzett feladatok eredményeként a 3. ábrán látható, üres cellát kapja. A cellát mozgatni, a **ctrl billentyű** és **bal egérgomb** folyamatos nyomva tartásával tudja, a kamera orientációját pedig a **shift** és **ctrl billentyűk**, illetve a **balegérgomb** folyamatos nyomva tartásával változtathatja.



3. ábra Menürendszer ismertetése

Menürendszer ismertetése:

1. A képernyő felső részén található fülek segítségével tudjuk változtatni a főablak (4.), illetve a szalagmenü (2.) tartalmát. (Pl.: A megírt programkód megttekintéséhez a **RAPID** fülre kell kattintanunk)
2. A felső szalagmenü tartalmazza a legfontosabb utasításokat, amelyeket a cellánk megépítése vagy éppen a programozás során használunk. Az egyes utasítások kategorizálva vannak, az egyes kategóriák között az előzőekben bemutatott fülek segítségével válthatunk.
3. A baloldali fa struktúrájú menüben további információkat találhatunk a cellánkról, virtuális vezérlőnkről vagy éppen a megírt programunkról. A menü tartalma a felső fülek (1.), illetve közvetlen a menü fölött található fülek segítségével változtatható
4. A főablak vagy szimulációs ablakban láthatjuk a cellánkat vagy éppen a megírt programunkat, illetve itt tudjuk lekövetni az elkészített szimulációkat.
5. Az **output window** segítségével követhetjük nyomon, a szimuláció vagy éppen a virtuális vezérlő állapot változásait. Kiemelt szerepe van a hibakeresés szempontjából, mivel itt tudjuk kiolvasni az esetleges hibaüzeneteket. **Az output windowban megjelenő hibaüzenetekről részletesebb információt kaphatunk az üzenetre kattintva!**

Figyelem!

A menürendszer, a bal felső sarokban található ➔ ikonra kattintva, majd a DEFAULT LAYOUT menüpontot kiválasztva, az eredeti állapotába állítható vissza.

Kérdések:

1. Röviden magyarázzá el a mit jelent a Station, Virtual Controller és Solution a RobotStudioban! Használja segítségül a RobotStudio Operating Manual – Understanding stations and solutions fejezetét.
2. Röviden ismertesse mi a különbség a **Solution with Station**, a **Solution with Station and Virtual Controller** és az **Empty Station** utasítások között (projekt/ cella létrehozás)!
3. A projekt létrehozásánál, milyen mappákat generált a RobotStudio? Ismertesse az egyes mappákat tartalmuk szerint.

2. feladat – Robot és Virtuális vezérlő beimportálása.

A robot beimportálásához, válassza ki a **Home** fület, a szalagmenüben kattintson az **ABB Library** menüpontra. A megjelenő listából válassza ki az IRB 1600-as manipulátort, azonban mivel ennek a robottípusnak több változata is létezik az ABB Kínálatában, a középen felugró ablakban további konfigurációkból választhatunk. Itt válassza a 6kg-os terhelhetőségű és 1,45m karkinyúlású verziót.

Ezt követően a robotunk megjelenik a cellában és már szabadon pozícionálhatjuk a munkatérben, illetve akár csuklónkként mozgatni is tudjuk. Mielőtt rátérnénk a virtuális vezérlő beimportálására, nézzünk meg 1-2 ehhez kapcsolódó utasítást.

Munkadarabok szabadkézi mozgatása a munkatérben:

1. Válassza ki a **Home** fület, majd a felső szalagmenüben kattintson a **Freehand**-nél található **Move** utasításra (4. ábra – 1.). Ezt követően a baloldali menüben kattintson robot (vagy a mozgatni kívánt munkadarab) nevére, ekkor egy koordináta rendszer jelenik meg a robot (vagy munkadarab) alapjánál, amely tengelyét megragadva a robotot szabad kézzel újra pozícionálhatja a munkatérben.
2. A robot szabadkézi forgatásához kövesse az előző pontban leírt utasításokat, csak most a **Rotate** utasításra kattintson a szalagmenüben (4. ábra – 2.). A forgatás referencia koordináta rendszerének kiválasztásához kattintson a 4. ábrán látható legördülő menüre, majd a listából válasszon egy koordináta rendszert. (A Robotstudioban használt koordináta rendszerek a későbbiekben kerülnek bemutatásra). Természetesen a referencia koordináta rendszer a szabadkézi mozgatás esetében is állítható!

Robot szabadkézi mozgatása:

1. A manipulátor csuklónkként történő szabadkézi mozgatásához a **Freehand**-nél kattintson a **Jog Joint** utasításra (4. ábra – 3.). Ezt követően a robot valamely szegmensére kattintva, végállástól-végállásig forgathatja az adott csukló tengely körül.
2. A TCP (**Tool Center Point**) egyenes mentén történő mozgatásához a szalagmenüben kattintson a **Jog Linear** utasításra (4. ábra – 4.), majd a baloldali menüben kattintson a mozgatni kívánt robot nevére. Ezt követően a robot TCP-jénél megjelenő koordináta rendszer segítségével mozgathatja a manipulátort. Ez esetben is, a referencia koordináta rendszer megváltoztatásával különböző irányokba mozgathatja a robotot.
A Jog Linear utasítás csak akkor elérhető ha a robot már rendelkezik virtuális vezérlővel. A vezérlő beimportálását követően térjen vissza ehhez a feladathoz!
3. A szerszám koordináta rendszer orientációjának megváltoztatásához válassza a **Jog Reorient** utasítást a szalagmenüben (4. ábra – 5.), majd kattintson a manipulátor

nevére a baloldali menüben. Ezt követően a megjelenő nyilak segítségével tudja változtatni a szerszám koordináta rendszer orientációját. **A vezérlő beimportálását követően térjen vissza ehhez a feladathoz!**



4. ábra Freehand menü

Figyelem!

A robot Home pozícióba állításához, kattintson jobb egérgombbal a manipulátor nevére a baloldali menüben, majd a Move To Pose>Jump Home utasítással, küldje Home pozícióba a robotot.

A virtuális vezérlő beimportálásához válassza ki a **Home** fület, majd a szalagmenüben kattintson a **Virtual Controller** menüpontra. A megjelenő listában három menüpont közül választhat:

1. From layout...

Ezesetben a Robotstudio a már összerakott robotcella alapján konfigurálja fel a virtuális vezérlőt.

2. New Controller...

Ezt a menüpontot választva egy teljesen új vezérlőt adhatunk hozzá a cellánkhöz, akár új robottal együtt, azonban a virtuális vezérlő paramétereit nekünk kell megadnunk.

3. Existing Controller...

Ezesetben egy már létező, valamely korábbi projekt során teljesen felkonfigurált virtuális vezérlőt adhatunk a cellánkhöz.

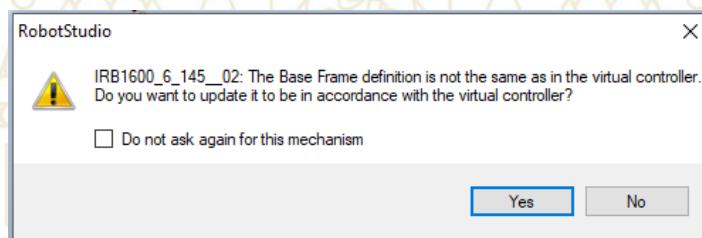
Kattintson a **New Controller** menüpontra! A felugró ablakban a **Robot model** menüpont alatt, válassza ki a legördülő listából a beimportált robot modellt. Illetve amennyiben valós hardverrel dolgozik adja meg a valós vezérlőn futó **RobotWare** típusát. A **Customize Option** opcionál egy későbbi projekt során fogunk foglalkozni! Amennyiben elvégezte a szükséges beállításokat a **Mechanisms** pont alatt válassza ki a „**Use existing mechanism**” opciót, ellenkező esetben a RobotStudio egy újabb robotot fog beimportálni a munkatérbe.

A virtuális vezérlő megnevezésének módosítása nem szükséges, azonban amennyiben több azonos típusú robotot használ megkönnyítheti az egyes manipulátorok beazonosítását.

Figyelem!

A robot típusának kiválasztásánál ügyeljen rá, hogy a megfelelő terhelhetőségű és karkinyúlású opciónt válassza!

Az adatok ellenőrzését követően kattintson az **OK** gombra. A virtuális vezérlő beimportálása során az 5. ábrán látható hibaüzenetet kaphatja, ez esetben kattintson a **YES** gombra.

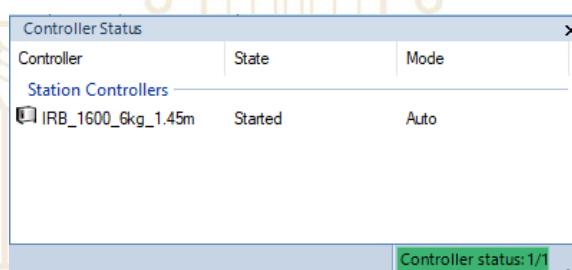


5. ábra Hibaüzenete a virtuális vezérlő beimportálása során

A virtuális vezérlő aktuális állapotát a képernyő jobb alsó sarkában található státusz mutatóról olvashatja le. Illetve a státusz mutatóra kattintva részletesebb információkat is kaphat a vezérlő állapotáról (6. ábra).

Az egyes állapotokhoz tartozó színkódok:

- **Zöld:** Automata mód
- **Sárga:** Manuális mód
- **Piros:** A vezérlő leállt



6. ábra Státusz mutató

Figyelem!

A feladat végrehajtása során folyamatosan figyeljen a virtuális vezérlő állapotára! A valósághoz hasonlóan a virtuális vezérlő nélkül a robot üzemképtelen.

Amint azt a korábbi tanulmányaik során láthatták, a robotikában kiemelkedő szerepe van a különbéle koordináta rendszereknek, illetve a köztük leírható transzformációknak. A RobotStudioban a szimulációk készítése során is kb. 7 különböző koordináta rendszert alkalmazunk, így mielőtt tovább haladnánk a feladat megoldásában elengedhetetlen, hogy áttekintsük ezeket. Például az 5. ábrán látható hibaüzenet sem értelmezhető számunkra a RobotStudioban használatos koordináta rendszerek ismerete nélkül.

A RobotStudioban használt koordináta rendszerek:

- **Világ Koordináta Rendszer/ World Coordinate System**

A szimuláció során használt koordináta rendszerek hierarchikus felépítésűek, mely tetején a világ koordináta rendszer áll. Ez adja meg a szimulációs tér abszolút nulla pontját. A világ koordináta rendszer nem más mint az üres cellánk (3. ábra) középpontjában található koordináta rendszer, amely orientációját folyamatosan nyomon követhetjük a szimulációs ablak bal alsó sarkában található koordináta rendszer segítségével.

Itt azonban nagyon fontos megjegyezni, hogy a megnevezés becsapós lehet. Ugyanis a robotok kinematikai leírása során használt világ koordináta rendszer nem egyezik meg a RobotStudioban használt **World Coordinate System** kifejezéssel. A kinematikai leírás során a világ koordináta rendszer, legtöbb esetben egy a robot alapjához rögzített koordináta rendszer, amelyben a robot (a szerszám koordináta rendszer) mozgását vizsgáljuk. Ezzel szemben a RobotStudioban a világ koordináta rendszer egy abszolút viszonyítási alapként szolgál, ehhez viszonyítva adhatjuk meg a cellánkat felépítő egyes elemek- vagy adott esetben szimulációban használt egyéb koordináta rendszerek pozícióját. Habár majd a későbbiek során, láthatjuk hogy az egyes elemek pozícionálása során, a világ koordináta rendszertől eltérő referencia rendszert is választhatunk, azonban ez elsősorban csak kényelmi funkció, a valós pozíció a világ koordináta rendszerre vezethető vissza.

Fontos megjegyezni, hogy a virtuális térbe újonnan beimportált elemeket, minden úgy helyezi el a RobotStudio, hogy az elem saját koordináta rendszere a világkoordináta rendszerrel essen egybe.

- **Base Frame/ Local Coordinate System**

A szimuláció során a cellánkat felépítő minden elemhez egy-egy koordináta rendszert rendelünk, amelyet **Local Coordinate Systemnek** (LSC) nevezünk. Amikor az egyes darabokat pozícionáljuk a munkatérben, akkor tulajdonképpen ezen koordináta rendszer pozícióját és orientációját adjuk meg a világ koordináta rendszerhez képest. A robotok esetében az LCS-t **Base Framenek** (Alap Koordináta Rendszer) nevezzük és minden esetben a robot alapjának középpontjában található.

- **Task Frame**

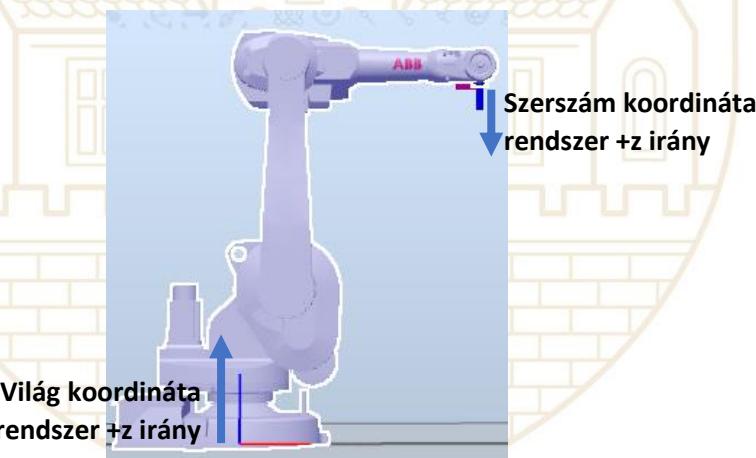
A **Task Frame** a virtuális vezérlőnkhoz rendelt koordináta rendszer. Funkciója egész egyszerűen kifejezhető: a **Task Frame** virtuális vezérlőnk világ koordináta rendszere.

Az előző mondat egyszerűen értelmezhető, ugyanis robotunk mozgását a virtuális vezérlő irányítja a szimulációs térben, tehát mozgást a virtuális térben a **Task Frame**ben vizsgájuk. Így kinematikai szempontból a **Task Frame** lesz a világ koordináta rendszerünk. Alapértelmezett esetben a **Task Frame** a robot alapkoordináta rendszerével esik egybe (kivételt képeznek például multi robot rendszerek, azonban ezekkel csak a későbbiekben foglalkozunk). **Nagyon fontos, hogy a robotcellánk konfigurációja során ügyeljünk rá, hogy a robot alap koordináta rendszere egybe essen a Task Frammel, ellenkező esetben a robot nem a kívánt pályán fog végig haladni.**

- **Szerszám Koordináta Rendszer**

A szerszám koordináta rendszer a robotunk végberendezésére ültetett, azzal együtt mozgó koordináta rendszer. Origója az úgy nevezett TCP (Tool Center Point), szerszámközéppont. A robot programozás során a célunk, hogy a szerszám koordináta rendszert adott pozíciókba- vagy egy jól definiált pálya mentén mozgassuk. Alap helyzetben a szerszám koordináta rendszer, a robotkar utolsó szegmensén található szerelőlap (ide kerül felszerelésre a technológiához kiválasztott végberendezés) középpontjában található és a koordináta rendszer z tengelye a felületből kifelé mutat. **Ezt fontos kiemelni, mivel pl. a 7. ábrán látható robot pozíció esetén, amennyiben a manipuláltot a z tengelymentén pozitív irányba mozgatjuk és referencia koordináta rendszernek a világ koordináta rendszert választjuk a robot „felfelé” fog mozogni, ellenben ha a referencia koordináta rendszer szerszám koordináta rendszer, a robot az ellenkező irányba fog elindulni, tehát „lefelé” mozdul el.**

A robot programozása során a szerszám koordináta rendszert kiválasztott végberendezésnek megfelelően ofszettelenünk kell. Egy robothoz több szerszám koordináta rendszer is hozzá rendelhető (multitool rendszerek), azonban a mozgások során mindenki ki kell jelölnünk az aktív szerszám koordináta rendszert.



7. ábra Szerszám- és világ koordináta rendszer

- **Target**

A **Targetek** a bejárni kívánt pályát leíró koordináta rendszerek. Fontos megjegyezni, hogy eddig tanulmányainkkal ellentétben a pályát nem pontokkal, hanem koordináta rendszerekkel jellemizzük, erre azért van szükség mert a pálya egy adott pontjában nem csak a szerszámunk pozícióját, hanem orientációját is meg kell határoznunk.

- **Workobject**

A pályát leíró **Targeteket** úgynevezett **Workobjectekben** értelmezzük. A programozás során minden egyes pálya pontot egy **Workobjecthez** kell hozzárendelnünk. A virtuális vezérlőnk beimportálása során a RobotStudio létrehoz számunkra egy alapértelemzett **Workobject** koordináta rendszert és amennyiben a felhasználó nem hoz létre újat a RobotStudio az alapértelemezett koordináta rendszerhez rendeli hozzá a felvett **Targeteket**. A **Workobjectek** a pályapontok egyszerű újra kalibrációját teszi lehetővé a munkatér módosítása esetén.

- **User Coordinate System**

A **User Coordinate System** (UCS) segítségével referencia pontokat hozhatunk létre a szimulációs térben, ezzel megkönnyítve a szimuláció elkészítését.

A RobotStudioban használt koordináta rendszerekről a RobotStudio Operating Manual – Programming Robots in the 3D Environment fejezetében olvashat részletesebben.

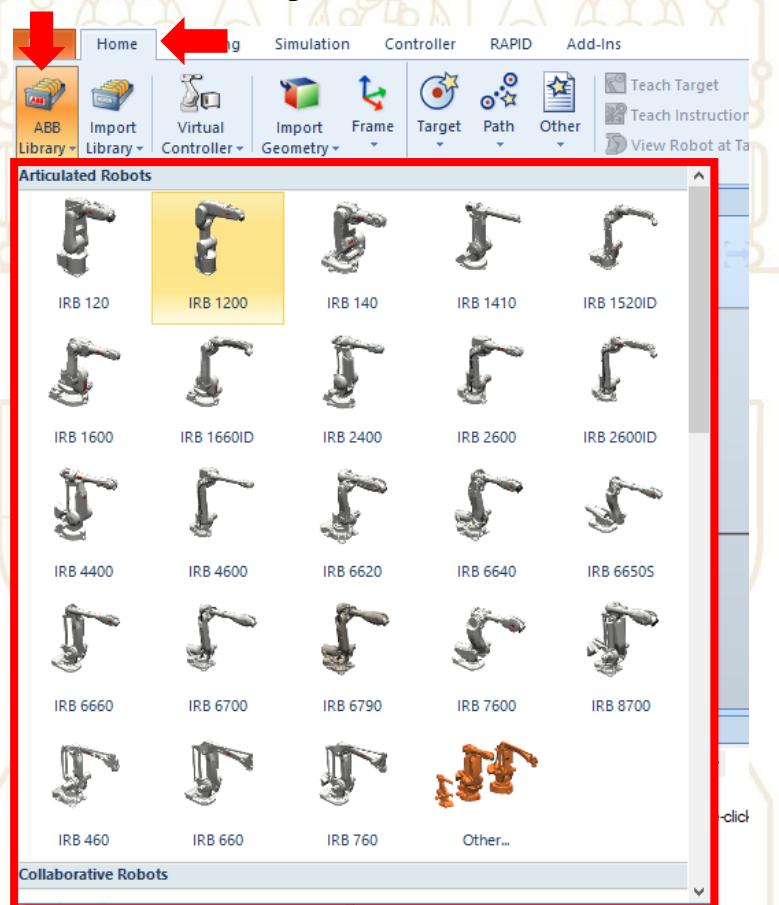
Kérdések:

4. Hány csuklóval rendelkezik az IRB1600-as manipulátor? Miért előnyös ez a konfiguráció?
5. Melyik alapkonfigurációnak feleltethető meg az IRB1600-as manipulátor? Röviden jellemzze a SCARA és Humanoid alapkonfigurációt!
6. Milyen elnevezést ad az alapértelemzett szerszám koordináta rendszernek, illetve workobjectnek a RobotStudio? Hol helyezkedik el a szimulációs térben az alapértelemezett Workobject? (A válaszhoz vizsgálja meg a RobotStudio baloldali menüpöt, illetve fenti szalag menüt a Home fül kiválasztása esetében)
7. Röviden próbálja megmagyarázni, mi okozhatja az 5. ábrán látható hibaüzenetet!

3. feladat – A robotcella építőelemeinek beimportálása a virtuális térbe

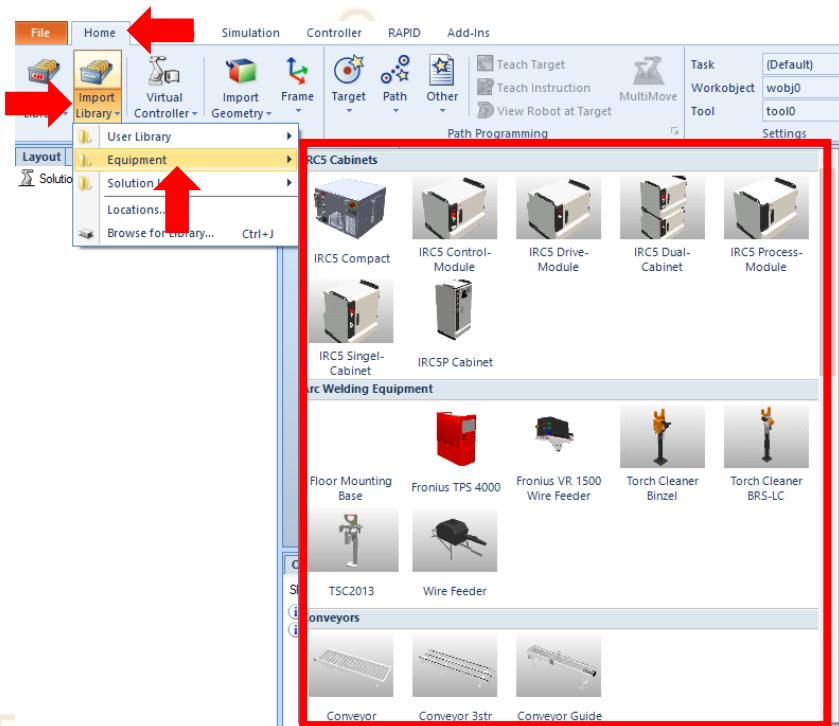
A robotunk virtuális programozása előtt, a cellánk valósággal megegyező modelljét kell elkészítenünk a szimulációs térben. Ehhez a különböző modelleket kell beimportálnunk, pl. végberendezések, szerszámok, pozícionáló eszközök stb. modelljeit.

A RobotStudio telepítésével automatikusan elérhetővé vált számunkra néhány egyszerűbb modell, amelyek az ...>**ABB>RobotStudio>ABB Library** mappában találhatunk meg. Itt találhatóak például a különböző ABB robotok vagy pozícionáló berendezések modelljei, amelyeket, a **Home** fület kiválasztva, majd a felső szalagmenüben az **ABB Library** menüpontra kattintva tekinthetünk meg (8. ábra).



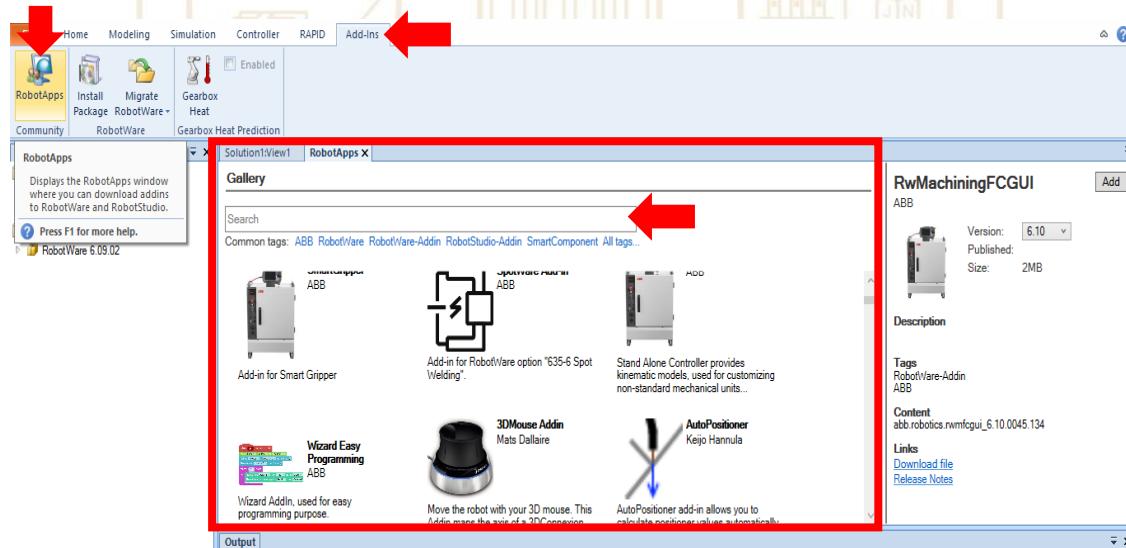
8. ábra ABB Library

A különböző robotokon, mechanizmusokon kívül a könyvtárban néhány egyszerűbb építő elem is megtalálható, mint például végberendezések, megfogók, vezérlő szekrények, futószalagok vagy egyszerűbb kerítések modelljei. Ezeket elsősorban a példafeladatok során fogjuk alkalmazni. Ezen modellek a **Home** fület kiválasztva, majd a felső szalagmenüben az **Import Library** menüpontot lenyitva (az ikon alatt található kis nyilacsára kell kattintani) és a megjelenő menüben az **Equipment** menüpontot kiválasztva érhetjük el (9. ábra).



9. ábra Import Library

A modell állományunkat tovább bővíthetjük, amennyiben az **Add-Ins** fülre kattintunk és felső szalagmenüben a **RobotApps** menüpontot választjuk. A középső ablakban található keresőmező segítségével további modelleket, illetve különböző modulokat tölthetünk le az ABB szerveréről (10. ábra).



10. ábra ABB letöltési központ

Az újonnan letöltött fájlok beimportálásához az előzőekhez hasonlóan a fenti szalagmenüben az **Import Library** menüpontot kell lenyitnunk, majd a lenyíló menüben a **Browse for Library** opciót kell kiválasztanunk (9. ábra).

Természetesen egy valós feladat esetében legtöbbször egyedi modellekkel dolgozunk. A RobotStudioban lehetőség van rá, hogy különböző formátumú CAD fájlokat importálunk be virtuális cellánkba. A CAD fájlokat a szoftver automatikusan konvertálja, ami mérettől függően akár néhány percet is igénybe vehet és csak ezután látjuk a modellt a munkaterben.

Figyelem!

A School Edition licensz nem tartalmaz CAD konverter modulokat (pl.: Inventor Convert, STEP converter, SolidEdge Converter), így amennyiben ezt a licensz használja ügyeljen rá, hogy a beimportálni kívánt modell, egy a RobotStudio által támogatott formátumba legyen mentve.

Amennyiben az **ABB Library**ból vagy a RobotStudio letöltési központjából importál be modelleket nincs szükség konvertálásra, mivel ezek a fájlok **RobotStudio Library File** (kiterjesztés: .rslib) formátumba vannak elmentve. A .rslib következő előnyökkel jár a sima CAD fájlokkal szemben:

- A **Library** fájlok a modellre vonatkozó geometriai információkon kívül, további RobotStudio specifikus információkat is hordoz magában. Ilyen lehet például egy felkonfigurált megfogó berendezés szerszám koordináta rendszere.
- A CAD fájlok konvertálása során a RobotStudio a projekt mappába másolja a modellt, így egy nagyobb modell esetében jelentősen megnőhet a mappa mérete. Ezzel szemben **Library** fájlok esetében csak az adott modell elérési útvonalát rendeli a projekthez a RobotStudio. Így amennyiben egy nagyobb méretű modellt többszer szeretne felhasználni a robotcella felépítése során, érdemes azt **Library** fájlként menteni először. Ehhez importálja a modellt a RobotStudioba, majd kattintson a **Home** fülre, ezt követően pedig a baloldali menüben a **Layout** fülre, a baloldali menüben a modell nevére jobb gombbal kattintva megjelenik egy legördülő menü és itt válassza a **Save As Library...** menüpontot.

A mérés során a következő modellekre lesz szüksége:

- Szerszám: MyTool – ABB Library
- Munkadarab: Curve Thing – ABB Library
- Asztal: Table – Add-Ins>RobotApps
- Robot alap: IRB1600 Base 500mm - Add-Ins>RobotApps

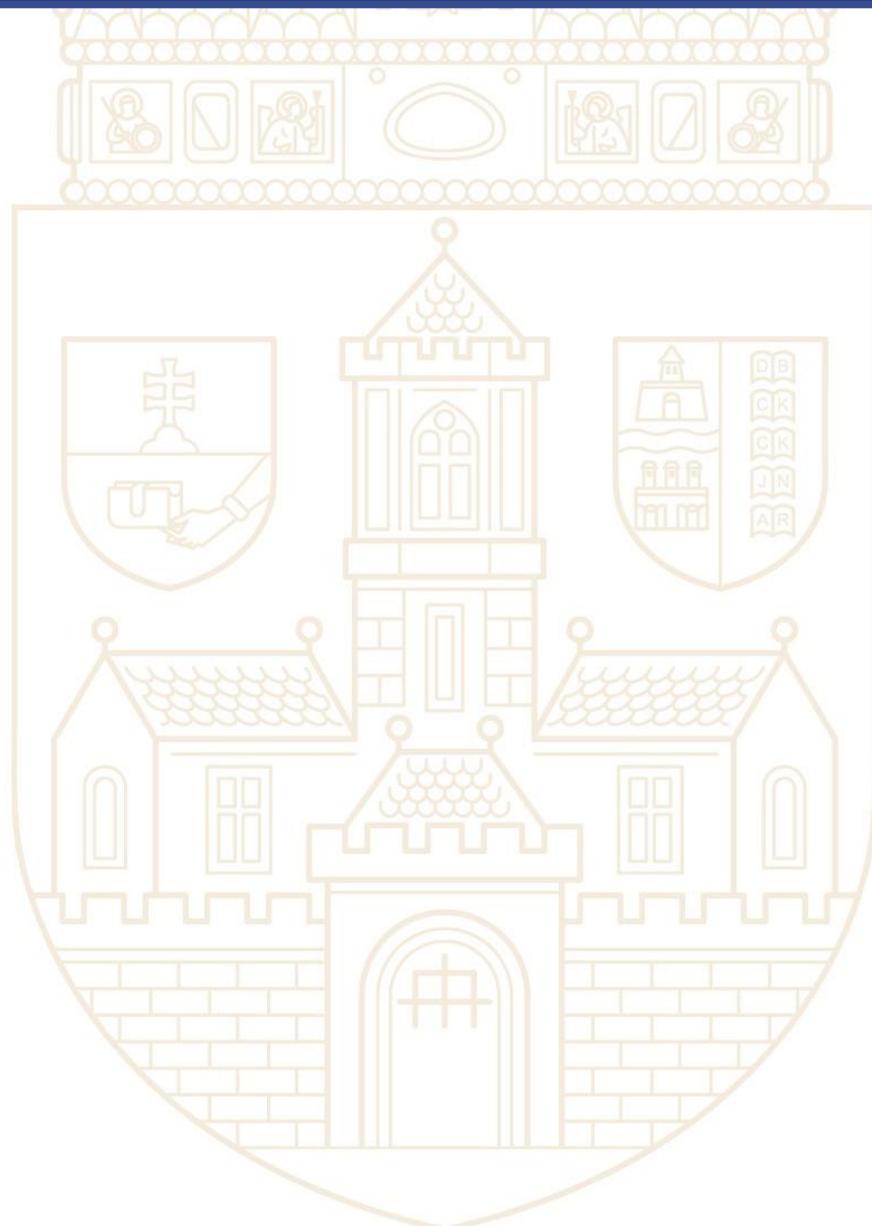
Tölts le is importálja a megfelelő modelleket a robotcellába!

Figyelem!

A modellek importálása során ügyeljen rá, hogy minden fájl típusának megfelelő utasítást használja (Import Library/ Import Geometry).

Kérdések:

8. Soroljon fel 5 olyan CAD formátumot, amelyet kiegészítő modul nélkül is támogat a RobotStudio! (Használja segítségül a RobotStudio – Operating Manual 1. fejzetét.)

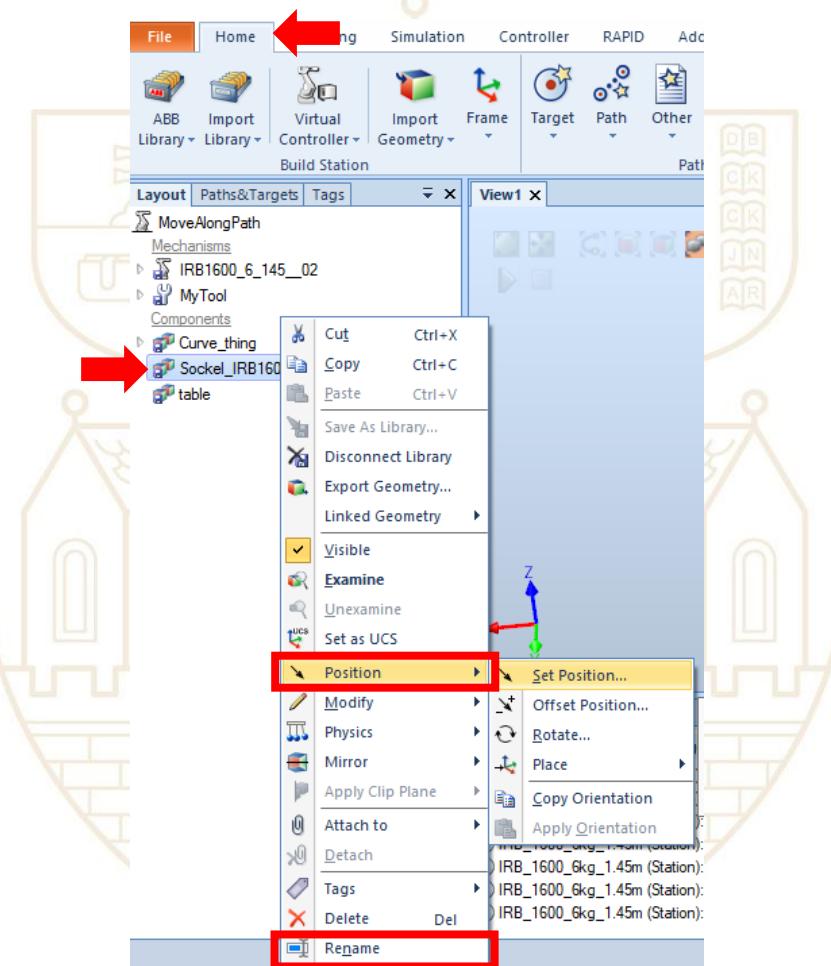


4. feladat – A robotcella építőelemeinek pozícionálása a virtuális térben

A cellát felépítő különböző elemek pozícionálása során az adott geometriához rendelt koordináta rendszer (**Local Coordinate System**) pozícióját adjuk meg a világ koordináta rendszerhez képest. A RobtosStudio több utasítást kínál számunkra, a munkadarabok pozícionálásának megkönnyítése érdekében. Ebben a fejezetben minden összes néhány utasítást fogunk csak áttekinteni.

A 2. fejezetben már megismertedtünk a **Move** utasítással, amely az egyes elemek kézi mozgatását teszi lehetővé. Az utasítás segítségével, a robot kivételével, mozgassa át elemeket úgy a szimulációs térben, hogy azok könnyedén elérhető, elkülöníthetők legyenek!

Ezt követően kattintson a **Home** fülre felül, majd a baloldali menüben válassza ki a **Layout** fület. Így a baloldali menüben a munkatérbe importált modellek megnevezései láthatók, amennyibe szükséges az egyes elemekre jobb egérgombbal rákattintva, majd a legörökítő menüből a **Rename** utasítást kiválasztva az adott elem átnevezhető (11. ábra). A munkadarabok pozícionálásához pedig, ugyanebben a menüben válassza a **Position** menüpontot (11. ábra).



11. ábra Modellek átnevezése és pozícionálása

A jelenlegi feladatban a virtuális munkatér origójában a robot alap helyezkedik el. A RobotStudio a beimportált elemeket a legtöbb esetben az origóba pozícionálja, így amennyibe az alapot nem mozgatta el az megfelelő helyen van. Amennyiben elmozgatta kattintson jobb egérgombbal az elem nevére a bal oldali menüben majd, a **Position** menüpont alatt válassza a **Set Postion** utasítást (11. ábra).

A **Set Postion** utasítás segítségével abszolút módon adhatjuk meg az adott elem pozícióját a referencia koordináta rendszerhez képest. A **Set Postion** utasításra kattintva a balfelső sarokban a 12. ábrán látható ablak jelenik meg.



12. ábra Set Postion utasítás

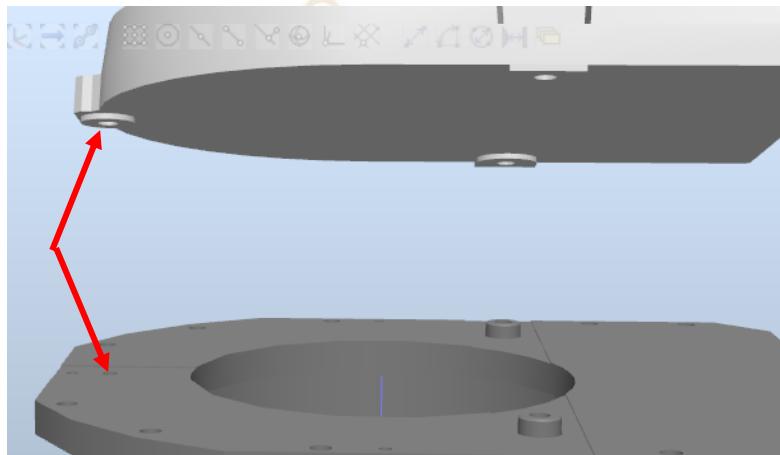
Az ablakban a következő információkat láthatjuk:

1. A pozícionálni kívánt munkadarab neve
2. Referencia koordináta rendszer
3. X, Y, Z pozíció adatok
4. RotX, RotY, RotZ orientációs adatok

Mivel a robot alapot a munkatér középpontjába, tehát a **World Coordinate System** origójába szeretnénk mozgatni, így a referencia koordináta rendszer esetünkben a **World** lesz és a pozíció és orientációs adatok helyére pedig 0-át írunk mindenhol. Az ezután az **Apply** gombra kattintva robotalap újra az origóba kerül.

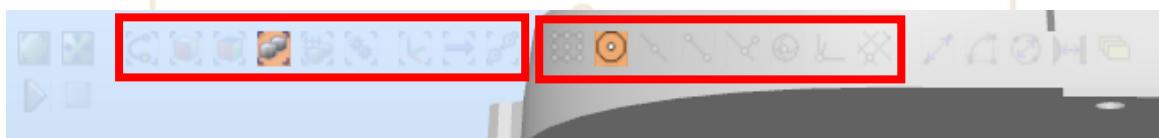
Ezt követően helyezzük a robotot az alapra. Mivel pontosan nem ismerjük a robotalap magasságát így abszolút módon nem tudjuk megadni a robot koordinátáit. Jelen esetben a **Postion** menüpont alatt található **Place** opciót választjuk, majd ott a **One Point** utasításra kattintunk.

Ezen utasítás segítségével a munkadarab egy adott pontját a tér egy tetszőleges pontjába mozgathatjuk. Jelen esetben a robot alapján található egy furatot tudjuk a robotalapon található egyik furathoz illeszteni és így a robotot az alapra helyezhetjük (13. ábra).



13. ábra Furatok illesztése

Azonban ahhoz hogy az utasítást használni tudjuk ismernünk kell a két furatközéppont pontos koordinátáit. Ebben lesz segítségünkre a szimulációs ablak felső részén található ikonok, az úgynevezett **Selection** és **Snap Tools** (14. ábra).



14. ábra Selection és Snap Tools

A **Snap Tools** segítségével az adott elem nevezetes pontjait tudjuk kijelölni. Például A **Snap Center** utasítás segítségével, amennyiben egy felület középpontja felé közelítjük kurzorunkat a RobotStudio automatikusan kijelöli azt számunkra. A kijelölt pontot egy szürke gömbbel jelzi számunkra a szoftver. A **Selection Tools** segítségével pedig a választás „szintjét” határozhattuk meg. Például a **Surface Selection** segítségével egy adott felületen tudunk pontokat kijelölni, míg a **Part Selection** esetében az egész munkadarabról választhatunk pontokat, de a **Target/ Frame Selection** esetében csak koordináta rendszerek pontjait jelölhetjük ki.

Így már meg tudjuk adni a megfelelő bemenő paramétereket a **Place>One Point** utasítás számára. A **Primary Point - Fom (mm)** paraméter jelöli azt a pontot amit szeretnénk elmozdítani, a **Primary Point - To (mm)** pedig ahova szeretnénk mozgatni. Bármelyik pont koordinátáit megadhatjuk úgy, hogy belekattintunk valamely koordinátájába majd a szimulációs térben kiválasztjuk azt a pontot amely koordinátáit szeretnénk beleírni, ezt követően a RobotStudio automatikusan kitölti az adott paraméter összes koordinátáját.

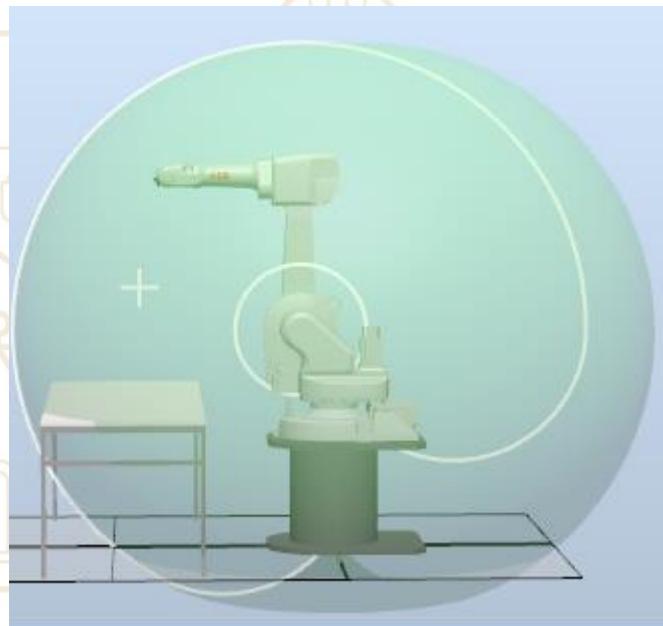
Válassza ki a feladathoz legmegfelelőbb **Selection** és **Snap Tool**-t, emelje a robotot a robotalap fölé (csak z irányba mozgassa a manipulátort), majd helyezze a robotot az alapra. **A robot mozgatása során ügyeljen rá, hogy a Task Framet mindig mozgassa a robot alap koordináta rendszerével együtt!**

Ezt követően pozícionáljuk az asztalt a munkatérben. Az asztalt a **World Coordinate System** x tengelyére szimmetrikusan helyezzük el, hogy az x tengely az asztal hosszabbik oldalának felező pontja alatt haladjon. Ehhez azonban ismernünk kell az asztal pontos méreteit.

A méretek meghatározáshoz kattintson a **Modelling** fülre felül, majd válassza a **Measure Point to Point** utasítást. A **Selection** és **Snap Tool** helyes beállításai mellett határozza meg az asztal fő méreteit, majd ezt követően **Set Position** utasítás segítségével helyezze el az asztalt a fenti leírásnak megfelelően (jelenleg csak az y koordináta meghatározása szükséges)!

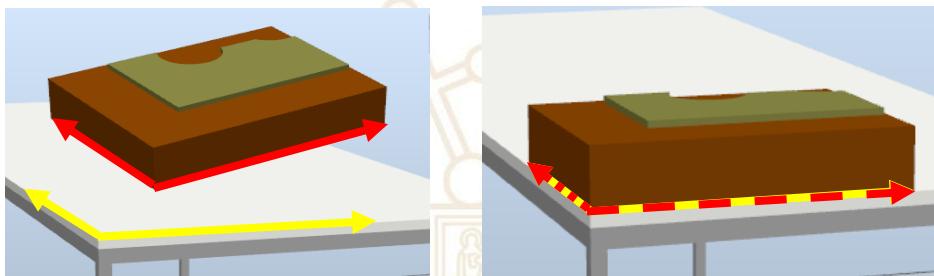
Mivel a munkadarabot az asztalon fogjuk elhelyezni így kiemelten fontos, hogy az asztal a robot munkaterén belül helyezkedjen el. Kattintson felül a **Home** fülre, majd a bal oldali menüben a válassza a **Layout** fület. Jobb egér gombbal kattintson a robot nevére és a legördülő menüből válassza a **Show Work Envelope** utasítást! Így 2 és 3 dimenzióban is megjeleníthetjük a robot munkaterét a szoftver segítségével, illetve akár a jelenlegi szerszám koordinátára rendszernek megfelelő korrekciót is állíthatunk be.

A **Move** utasítás segítségével módosítsa az asztalt x koordinátáját, úgy hogy az a munkatéren belül legyen (15. ábra)!



15. ábra A robot munkatere

A következő lépés, hogy a munkadarabunkat az asztalra helyezzük. Ehhez megint nehézkes lenne a **Set Position** utasítást alkalmazni, mivel bonyolult lenne meghatározni az asztal valamely pontjának a koordinátáit. Továbbá a **Place>One Point** utasítást sem használhatjuk mivel a munkadarab pozíóján kívül orientációja sem megfelelő. Ez esetben válasszuk a Place menüpont alatt **Three Points** utasítást.



16. ábra Place Three Points utasítás

Az utasítás működésének megértéshez tekintsük meg a 16. ábrát. Az utasítást működése nagyban megegyezik a **Place>One Point** utasításával, csak ez esetben a mozgatni kívánt munkadarabon is 3 pontot és térben is 3 pontot jelölünk ki. A mozgatás során a RobotStudio a 3-3 pontból egy-egy koordináta rendszert képez, majd ezeket a koordináta rendszereket igazítja egymáshoz a munkadarabunk mozgatásával.

Helyezze a munkadarabot a 15. ábrán látható pozícióba az utasítás segítségével. A koordináta rendszerek pontjainak kiválasztásánál a munkadarab és az asztal sarokpontjait használja segítségül!

Figyelem!

A paraméterek véglegesítése előtt mindenkor ellenőrizze, hogy a megfelelő név szerepel-e a mozgatni kívánt munkadarab helyén, illetve hogy az adott pontok megfelelően helyezkednek-e el a szimulációs térben!

Az utolsó feladat, hogy 'felszereljük' a szerszámunkat a robotunkra. Ehhez kattintsunk a baloldali menüben a szerszám nevére, majd húzzuk rá a robot nevére. Ilyenkor a RobotStudio hozzá kényszeríti a szerszámunk alapkoordináta rendszerét a robotunk alap szerszám koordináta rendszeréhez, így innentől kezdve a szerszám koordináta rendszer és a megfogó alapkoordináta rendszeréhez megőrzi relatív távolságát a robot mozgása során. Mivel ez esetben megfogót modelljét szeretnénk a szerszámkoordináta rendszerhez mozgatni, így mikor a RobotStudio rákérdezi, hogy szeretnénk frissíteni a szerszám pozíóját kattintsunk az igen gombra!

Kérdések:

9. A robot alapra helyezése során milyen Selection és Snap Tool beállításokat használt?
10. Adja meg az asztal fő méreteit!
11. Miért és hogyan változott az robotunk aktív szerszám koordináta rendszere a végberendezés felszerelését követően?