

ABB RobotStudio Gyakorlat

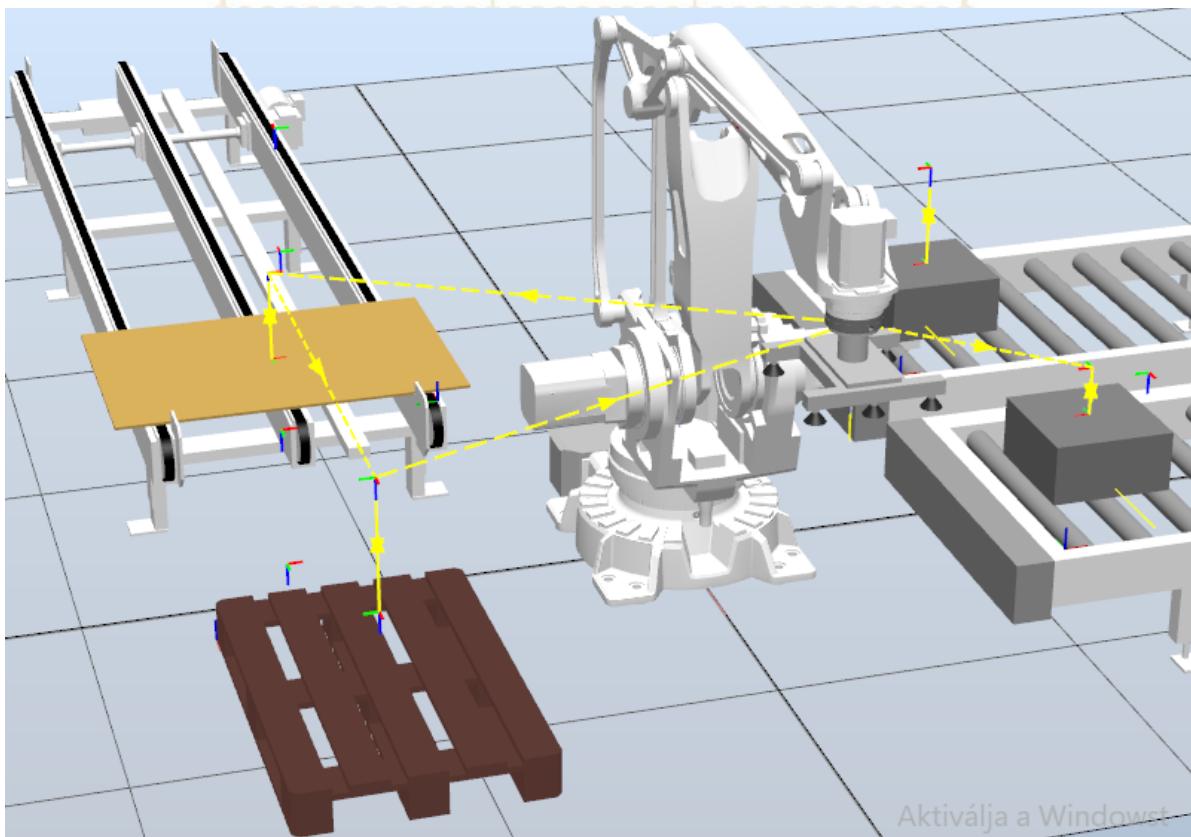
IV. Mérési Feladat

Palettázási feladat

Mérés Célja:

A mérés során a hallgatónak egy egyszerű palettázási feladatot kell elkészítenie RobotStudio segítségével. A robot feladata, hogy az alábbi ábrán látható dobozokat pakolja rá egy szabványos EURO raklapra.

A mérés célja, hogy a hallgatók megismerkedjenek, a struktúrált programozás két alapvető elemének, az elágazások és ciklusok használatával, Rapid szintaktikájával. Ezt követően a hallgató Önnállóan is képes legyen hasonló feladatok megoldására, komplexebb programok készítésére.



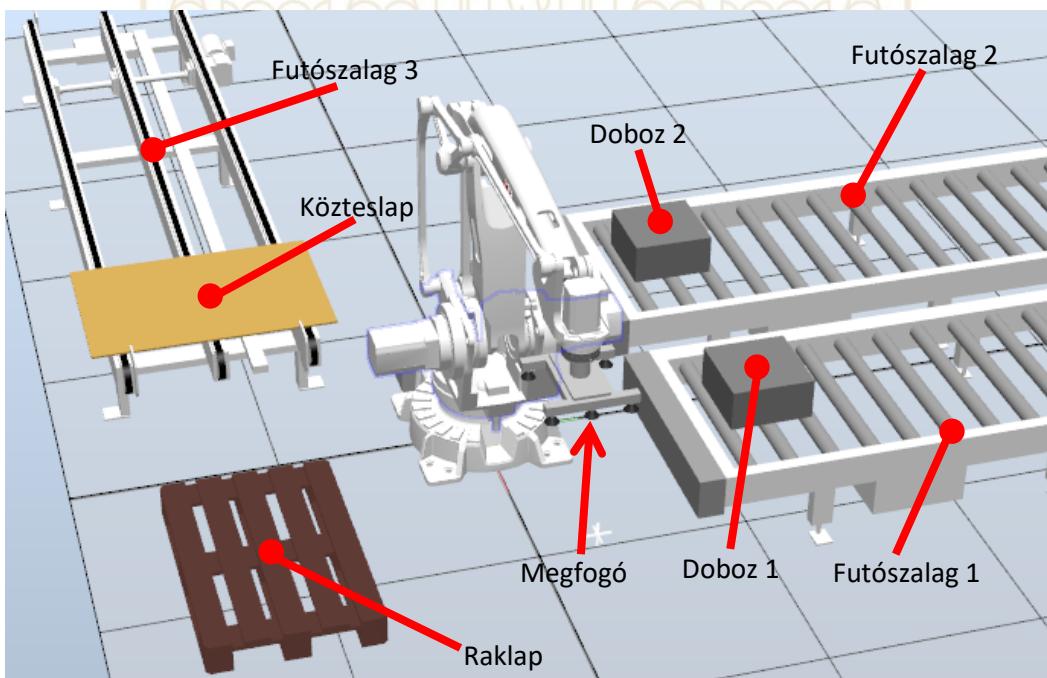
1. ábra Megvalósítani kívánt feladat

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

1. feladat – Robotcella felépítése és alaprogram megírása

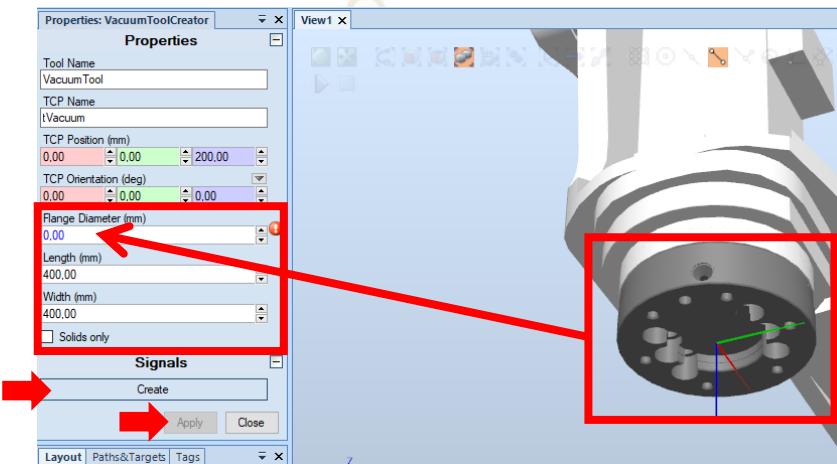
A szimulációt egy új virtuális cella segítségével fogjuk elkészíteni. Ehhez építse meg a 2. ábrán látható elrendezést, melyhez a következő elemeket használja:

- Robot: ABB IRB_460_110_240
- Megfogó: **Vacuum Tool Creator (Add-Ins)**
- Futószalag 1-2: **Conveyor**, **Width:950mm**, **Length:4m** (**ABB Library – Equipment**)
- Futószalag 3: **Conveyor 3str**, **Length: 4m** (**ABB Library – Equipment**)
- Raklap: **Euro Palett (ABB Library – Equipment)**
- Doboz 1-2: **Modeling>Box** (396x396x200mm)
- Közteslap: **Modeling>Box** (1200x800x10mm)



2. ábra Cella felépítése

A cellában használt **Vacuum Tool Creator** egy **Smart** komponens, amely segítségével egyedi vákuumos megfogót konfigurálhatunk a saját igényeinknek megfelelően. Ehhez kattintsunk a baloldali menüben a megfogó nevére, majd bal felső sarokban megjelenő ablakban adhatjuk meg a szükséges paramétereket. A megfogó szélessége és hosszúsága legyen 400 mm, **Flange Diameter** paraméter beállításához, pedig határozza meg a robot utolsó szegmensének végén található adapter (**Mounting Plate**) átmérőjét (3. ábra), ehhez használja a **Measuring Tool** a RobotStudioban vagy tekintse meg az IRB 460-as manipulátor kézikönyvét. Amennyiben elvégezte a szükséges beállításokat kattintson az **Apply** gombra, majd a **Create** gomb segítségével, hozza létre a megfogót. Ezt követően pedig szerelje fel a vákuumos megfogót a manipulátorunkra.



3. ábra Vákuumos megfogó konfigurációja

Amennyibe összeállítottuk a virtuális robotcellánkat elkezdhetjük a robotprogramunk elkészítését. Ehhez először a **workobjectek** létrehozása szükséges. A feladat megoldása során 4 **workobject** használata javasolt, minden futószalaghoz és a raklaphoz is rendeljünk hozzá 1-1 koordinátarendszert. Ne feledkezzen meg a koordináta rendszerek kényszerezéséről sem!

A következő lépés a pályapontok felvétele. Jelen esetben csak egy egyszerű „**Pick and Place**” típusú feladatot kell megoldanunk. A robot vegye fel a dobozt az 1-es futószalagról, majd helyezze azt a rakkalra.

Figyelem!

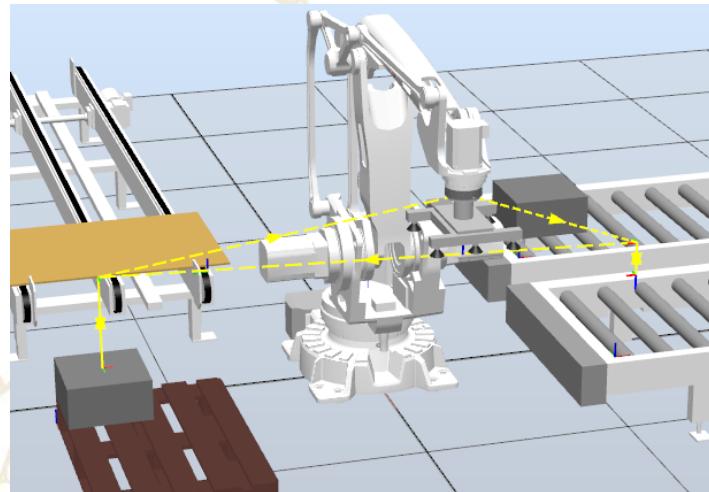
A pályapontok felvétele során mindig ügyeljen rá, hogy a megfelelő workobject legyen kiválasztva.

A pályapontok meghatározása során ügyeljen azok helyes felvételére, kerülje a felesleges robotmozgásokat (orientációk helyes beállításával). Ezt követően konfigurálja a robotmozgásokat, törekedjen legoptimálisabb robotprogram elkészítésére (robotmozgások típusának és a közelítési adatok (zonedata) helyes beállításával)

A feladat megoldáshoz konfigurálja szükséges I/O-kat és készítse el a Smart komponenst is!

Figyelem!

A szimuláció futtatása előtt, ne felejtse el elmenteni a cella jelenlegi állapotát!
(Simulation>Reset>Save Current Set)



4. ábra Első részfeladat

Kérdések:

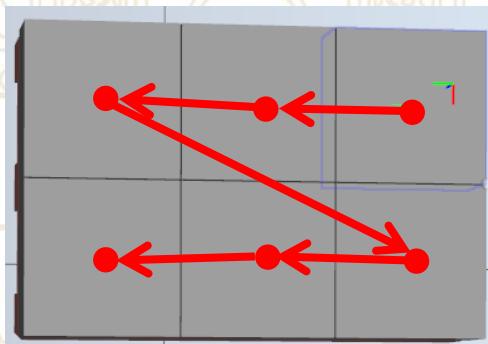
1. Milyen nehézséget tapasztalt a workobjectek felévttele során? Mi okozhatja a problémát? Segítségül használja a RobotStudio Operating Manual - Libraries, geometries and CAD files fejezetét.
2. Milyen I/O-kat konfigurált a feladat megoldásához? Mutassa be menetét?
3. Másolja a jegyzőkönyvbe az elkészített Robotprogramot!
4. Mutassa be a feladat elvégzéséhez szükséges Smart komponens felépítését!
Magyarázza el működését!

2. feladat - Dobozok palettázása

Módosítsuk a robotprogramot, úgy hogy a robotunk 6 db dobozt pakoljon a raklapra és a dobozok mindegyikét az 1. futószalagról vege fel. Ehhez mind **Smart** komponensünk, mind a robotprogramunk módosítanunk kell.

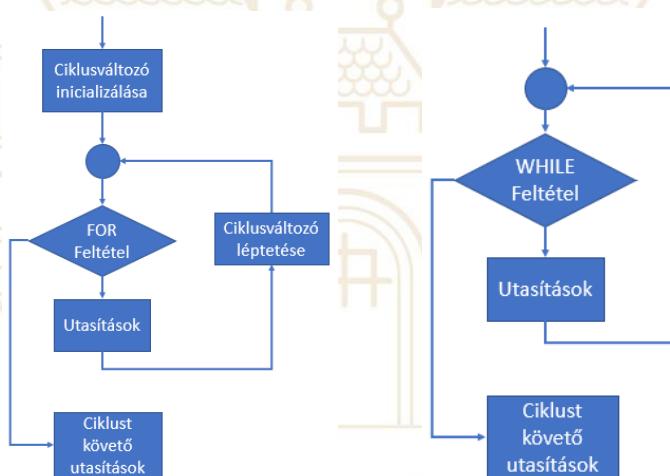
Először végezzük el a robotprogram módosítását. A feladattal kapcsolatban a következő megfontolásokat tehetjük:

- A robot minden ugyanabból a pozíóból veszi fel a munkarabokat (1. futószalag).
- A lerakási pozíció a program futása során folyamatosan változik a rakatképnek megfelelően (5. ábra).
- A feladat elvégzéséhez 6db „**Pick and Place**” ciklust kell megvalósítanunk.
- minden ciklus végrehajtása során a lerakási pozíció koordinátáit azonos mértékben kell offszetelnünk x vagy y irányban.



5. ábra Rakatkép kialakítása

Tehát a fentiekben leírtaknak megfelelően a programunk elkészítéshez, valamilyen ciklust kell alkalmaznunk. Mint a legtöbb magasszintű programozási nyelv esetében, a Rapidnál is használatunk **FOR**, illetve **WHILE** ciklusokat a programunk elkészítéséhez. A kétféle ciklus működését az alábbiakban bemutatott folyamatábrák szemléltetik.



6. ábra FOR és WHILE ciklusok folyamatárája

Jelen esetben, mivel tudjuk, hogy pontosan 6-szor kell végrehajtanunk a „Pick and Place” ciklust, így **FOR** ciklust fogunk alkalmazni.

Az előzőekben meghatározott kritériumok szerint, a másik fontos szempont az volt, hogy minden ciklus végrehajtása során azonos mértékben kell offszetelnünk a lerakási pozíció koordinátait. Ehhez az **offs** utasítást használhatjuk a Rapidban. Az utasítás pontos szintaktikájának megértéséhez, nézze meg az **ABB Technical Reference Manual – Rapid Instructions, Functions and Data Types** kézikönyvet.

Módosítsa a programot az alábbi ábrának megfelelően!

```
28  □      PROC main()
29      Path_10; !Add your code here
30
31  □      ENDPROC
32  □      PROC Path_10()
33          SetDo Hand,0; !Opening the gripper
34          WaitTime 1; !Wait for the gripper
35          MoveAbsJ Home,v1000,z0,tVacuum\WObj:=Palett; !Move to Home position
36  □          FOR cols FROM 0 TO 1 DO !To make 2 rows
37  □              FOR rows FROM 0 TO 2 DO !To put 3 boxes in a row
38                  !Picking up the box
39                  MoveJ Target_20,v1000,z100,tVacuum\WObj:=CNV1;
40                  MoveL Target_10,v1000,fine,tVacuum\WObj:=CNV1;
41                  SetDo Hand,1; !Closing the gripper
42                  WaitTime 1; !Wait for the gripper
43                  MoveL Target_20,v1000,z100,tVacuum\WObj:=CNV1;
44
45                  !Put down the box and offsetting the positions each time
46                  MoveJ offs(Target_40, rows*400, cols*400, 0),v1000,z200,tVacuum\WObj:=Palett;
47                  MoveL offs(Target_30, rows*400, cols*400, 0),v1000,fine,tVacuum\WObj:=Palett;
48                  SetDo Hand,0; !Opening the gripper
49                  WaitTime 1; !Wait for the gripper
50                  MoveL offs(Target_40, rows*400, cols*400, 0),v1000,z200,tVacuum\WObj:=Palett;
51          ENDFOR !FOR rows FROM 0 TO 2 DO
52          ENDFOR !FOR cols FROM 0 TO 1 DO
53          MoveAbsJ Home,v1000,z0,tVacuum\WObj:=Palett; !Move back to Home position
54      ENDPROC
55  ENDMODULE
```

7. ábra Rapid program palettázásához

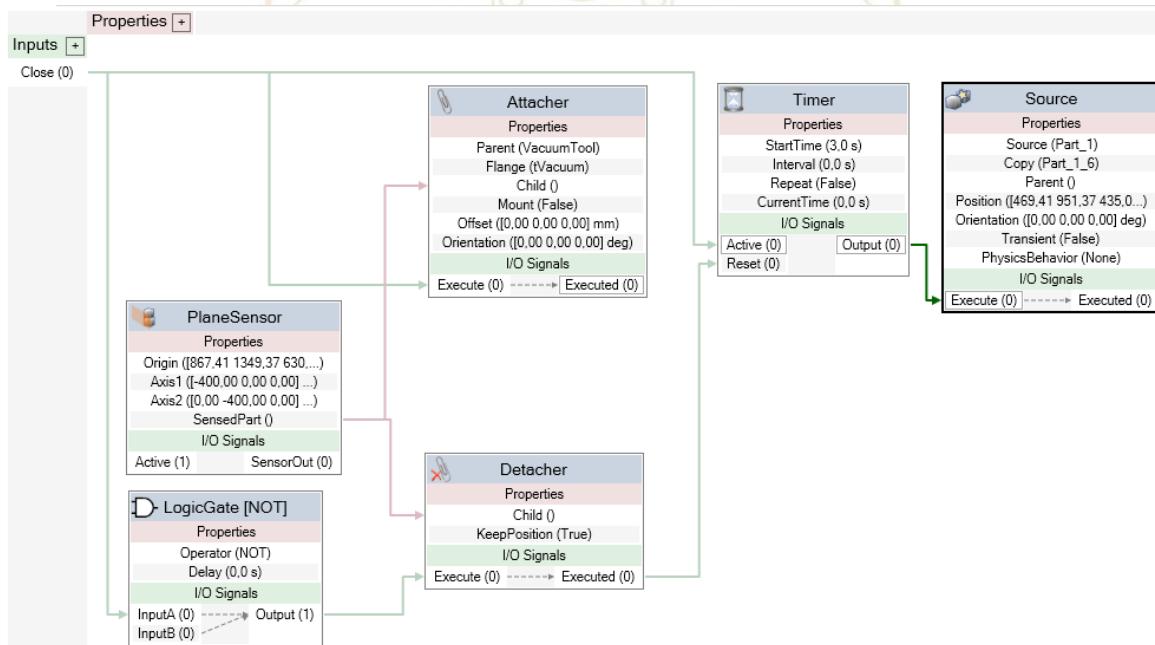
Amennyiben elkészítettük a programunkat elkezdhetjük a **Smart** komponensük módosítását. A jelenlegi szimulációban a futószalagot nem fogjuk működtetni, így a **Smart** komponensünk elkészítéséhez a következő megfontolásokat kell tennünk:

- minden ciklus lefutása előtt új munkadarabot kell helyeznünk a futószalagra.
- az eddigiekben az **Attacher** komponens **Child** paramétereként a 1-es dobozt állítottuk be, ez azonban most nem lesz megfelelő, ugyanis most minden ciklus során más-más munkadarabot kell felemelnünk.

Ezen kritériumok teljesítéséhez a következő blokkokat kell a **Smart** komponensünkhez adnunk:

- **Actions>Source** – A munkadarab másolásához
- **Signals and Properties> Timer** – A másolás időzítéséhez
- **Sensors>Plense Sensor** – Felemelni kívánt munkadarab kijelöléséhez

Módosítsa **Smart** komponensét az alábbi ábra szerint!



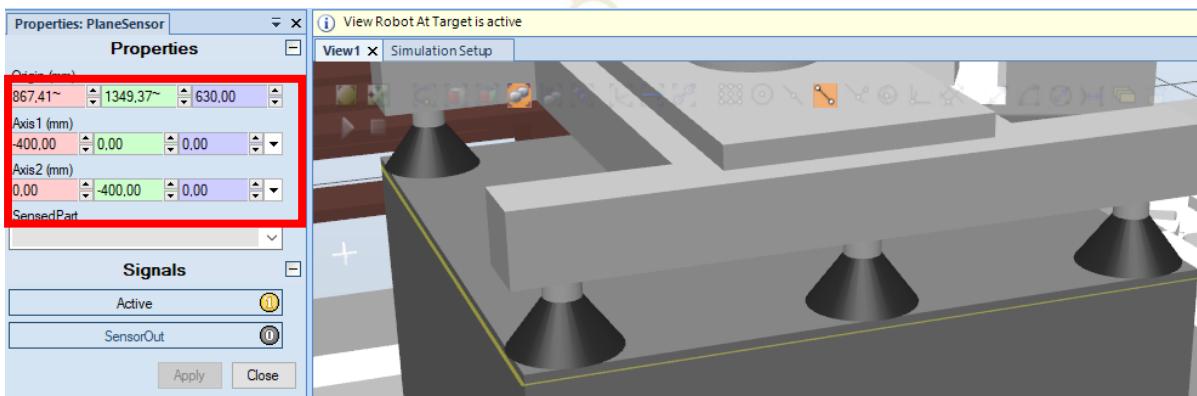
8. ábra Smart komponens palettázásához

A **PlaneSensor** beállítását a következők szerint végezze (9. ábra):

- Mozgassa a robotot a felvételi pozícióba (**View Robot at Target**).
- Az **PlaneSensor Origin** paraméterének megadásához válassza ki a doboz valamely sarokpontját, majd módosítsa úgy a koordinátákat, hogy a pont a doboz minden oldalától mért merőleges távolsága 2-2 mm legyen (orientációtól függően vonjon ki vagy adjon hozzá 2 mm-t az **Origin** paraméter x és y koordinátáihoz).
- Az **Origin** paraméter z koordinátájából vonjon ki 5-t. (Így nem fog belelógni a megfogó a **PlaneSensor**re.)
- Orientációtól függően az **Axis1 x**- és az **Axis2 y** koordinátáját állítsa ±400mm-re.
- A beállított paramétereknek köszönhetően egy olyan négyzetet kell kapni, amely metszi a dobozunk geometriáját, nincs közös pontja a megfogóval és túl lóg a doboz oldalain minden irányba 2-2mm-rel.

Amennyiben elvégeztük a szükséges beállításokat, kényszerízzük hozzá a **PlaneSensor**unkat a megfogónkhoz. Ehhez a baloldali menüben nyissuk le a **SmartComponent_1** menüpontot

és kattintsunk jobb egérgombbal a **PlaneSensor** nevére. Ezt követően válasszuk az **Attach To...** utasítást és a listából válasszuk ki a megfogónk nevét.



9. ábra PlaneSensor konfigurációja

Ha elvégeztük a szükséges beállításokat futtassuk a szimulációt!

Kérdések:

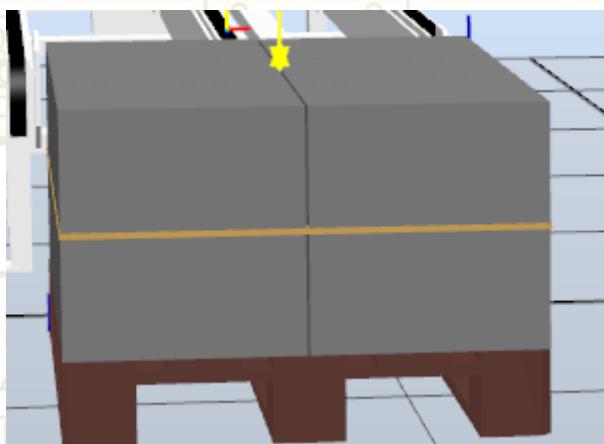
5. Magyarázza el a megírt program működését!
6. Mutassa be az elkészített Smart komponens működését.

3. feladat – Egyéni feladat

A két futószalagon véletlen időközönként jelenjenek meg (3-15s között) a dobozok és a robot mindenről a futószalagról vegye le azt, amelyiken éppen van munkadarab. Ha az egyik futószalagon sincs munkadarab a robot álljon Home pozícióba. A dobozok véletlenszerű megjelenítéséhez használjon egy **Other>Random** és **Signals and Properties>Timer** komponenseket, a munkadarabok detektáláshoz, pedig egy-egy **Sensors>LineSensor** blokkot.

A robotnak a következő rakatképet kell kialakítania (10. ábra):

- Először tegyen le 6db dobozt a raklapra
- Majd vegye fel a közteslapot a 3. futószalagról és helyezze a dobozok tetejére.
- Ezt követően tegyen még 6db dobozt a közteslapra.



10. ábra 2 rétegű rakat, közteslap

A robotprogram megvalósításához használjon **WHILE** ciklusokat és elágazásokat (**IF**, **ELSEIF**).

Kérdések:

7. Másolja be jegyzőkönyvébe a megvalósított Smart komponenst és magyarázza el működését!
8. Másolja be jegyzőkönyvébe az elkészített programot.