



M Ű E G Y E T E M 1 7 8 2

**BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM
GÉPÉSZMÉRNÖKI KAR
GYÁRTÁSTUDOMÁNY ÉS –TECHNOLÓGIA TANSZÉK**

AUTOMATIZÁLT GYÁRTÓRENDSZER TERVEZÉSE ÉS SZIMULÁCIÓJA

TDK MUNKA

**ÚJVÁRI ZOLTÁN
GÉPÉSZMÉRNÖKI SZAK, IV. ÉVFOLYAM**

KONZULENSEK:

DR. NÉMETH ISTVÁN	EGYETEMI DOCENS
HARASZKÓ CSABA	DOKTORANDUSZ
JUHÁSZ ISTVÁN	IRODAVEZETŐ

**A dolgozat bemutatását a TÁMOP - 4.2.2.B-10/1--2010-0009 "Új
tehetséggondozó programok és kutatások a Műegyetem tudományos műhelyeiben"
című program támogatta.**

Tartalomjegyzék

1. Bevezetés, célkitűzés ismertetése	3
2. A gyártási folyamat	4
2.1 A gyártandó munkadarab bemutatása	4
2.2 A gyártási technológia és a sorozatgyártási követelmények meghatározása	5
3. A munkadarab megfogása, készülékbe helyezése.....	10
3.1 OP10-ben történő rögzítés	10
3.2 OP20-ban történő rögzítés	11
3.3 OP30-ban történő rögzítés	12
3.4 Robotos manipulálás	13
4. Kidolgozott javaslatok a gyártórendszer telepítésére.....	21
4.1 A gyártórendszer elemeinek ismertetése	21
4.2 A gyártórendszer elrendezési koncepciójának ismertetése.....	22
4.3 A gyártórendszer működésének folyamata	23
5. Plant Simulation szimuláló program ismertetése	25
5.1 A program áttekintése	25
5.2 A program funkciói	26
5.3 A program előnyei	27
5.4 Felhasználása az iparban	27
6. Az elkészített elrendezési javaslat szimulációja.....	27
6.1 Szimulációs követelmények.....	27
6.2 A szimuláció alapegységeinek beillesztése.....	28
6.3 A szimulációs folyamat ismertetése	31
6.3.1 Robotrekesz nélküli stratégia	31
6.3.2 Robotrekeszes stratégia.....	33
7. Szimulációs eredmények összehasonlítása	34
7.1 Megmunkáló gépek összehasonlítása	34
7.2 Robotok működésének összehasonlítása	35
7.3 Előállított darabszám összehasonlítása	37
8. Összegzés.....	38
9. Irodalomjegyzék.....	39
10. Melléklet.....	40

1. Bevezetés, célkitűzés ismertetése

Napjainkban rendkívül gyakori, hogy egy adott alkatrész gyártását automatizálják, ugyanis a folyamatosan növekvő vásárlói igények kielégítése végett elkerülhetetlen ezen gyártási forma alkalmazása. Jelen TDK munkámban egy autókba épített turbófeltöltő automatizált gyártásához keresem a legmegfelelőbb elrendezési koncepciót, melynek segítségével elérhető a kitűzött gyártási darabszám.

TDK munkám felépítése az alábbi gondolatmenetet követi: Kezdetben szeretném ismertetni a szóban forgó munkadarabot, annak főbb jellegzetességeit, paramétereit. A gyártási technológia, illetve a sorozatgyártási követelmény szintén említésre kerül. Ezt követően bemutatom a turbófeltöltő robotos manipulálását, a megfogás módját, amellyel kiváltható a korábban emberek által végzett munka. Végezetül a gyártórendszer elrendezési vizsgálata, szimulációja következik, ami a leginkább foglalkoztathat egy céget, amely a gyártási darabszám maximalizálására törekszik.

Tanulmányom során egy gyártórendszer telepítésére és folyamatirányítására dolgoztam ki javaslatokat. A gyártórendszer az alábbi berendezésekből áll: 6-7 db megmunkáló központ, automatikus munkadarabmosó berendezés, szivárgástereszt berendezés, valamint az automatizálás során megvalósítandó sorjázó, nyersdarab beléptető és készdarab lerakó berendezések. Az egyes gépek és a technológiai berendezések helyszükségletei rendelkezésemre álltak. A technológiai berendezések kiszolgálása önálló portál rendszerű manipulátorral történik. A gépeknél a munkadarab berakását és kivételét, valamint a készüléken belüli munkadarab átrakást is a robotnak kell végeznie. Miután elkészültek a 3D-s elrendezési változatok, elkészítettem a Siemens Tecnomatix Plant Simulation nevű programmal azok diszkrét eseményvezérelt szimulációit is. A Plant Simulation használatát nem oktatják BSc-s képzéseken, így ennek elsajátítása szintén kihívást jelentett számomra. A szimulációs modellek segítségével meghatároztam, hogy hány megmunkáló központot képes egy robot optimálisan kiszolgálni. Megvizsgáltam, hogy szükségesek-e köztes munkadarab tároló helyek, ahol a robottal együtt mozgatott helyek is számításba jöhettek. Kiválasztottam egy elrendezési koncepciót, amelyen belül különböző anyagmozgatási stratégia lehetőségeket vizsgáltam, amiben segítségemre volt az újonnan megismert szimulációs program.

Mindent összevetve elmondhatjuk, hogy a gyártás szempontjából optimális elrendezéshez és gyártási folyamat irányításhoz nélkülözhetetlen egy szimulációs modell kidolgozása, amit ezen munkámmal be is bizonyítottam.

2. A gyártási folyamat

2.1 A gyártandó munkadarab bemutatása

A szóban forgó alkatrész gépjárművek kipufogórendszerének főbb alkotóeleme. Öntvény, súlya kb. 6,5kg. Többnyire egyenetlen felületek jellemzik, sok furattal és mart felület együttesel. A munkadarab 3D-s modellje az alábbiakban látható:



1. ábra: A turbófeltöltő 3D-s modellje [1]



2. ábra: A turbófeltöltő 3D-s modellje [1]

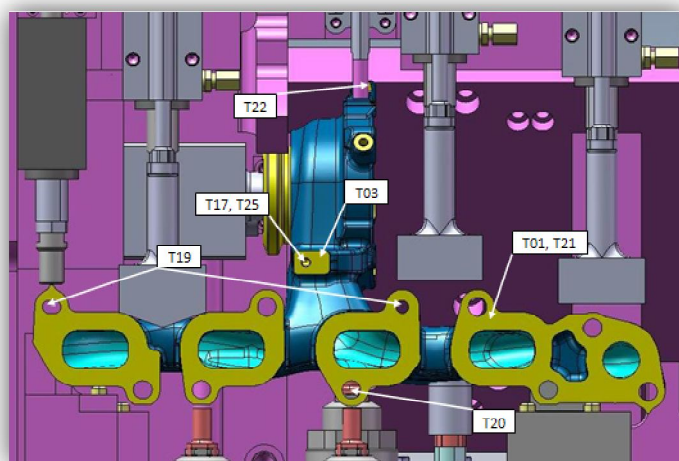
A 3D-s modellek Catia V5 tervezőprogram segítségével lettek megjelenítve. A munkadarab műszaki paramétereit a mellékletben található 2D-s műszaki rajzokról vehetjük le.

2.2 A gyártási technológia és a sorozatgyártási követelmények meghatározása

A munkadarabot egy készüléken belül 3 formában fogatjuk fel, amelyek a megmunkálás közben biztosítják annak mozdulatlanságát, helyzetének meghatározását. Ezeket rendre OP10, OP20, OP30 néven keresztelték el. Vegyük sorra a különböző felfogatásokban történő folyamatokat:

0°-os asztalpozíció esetén OP10-ben a következő megmunkálások történnek:

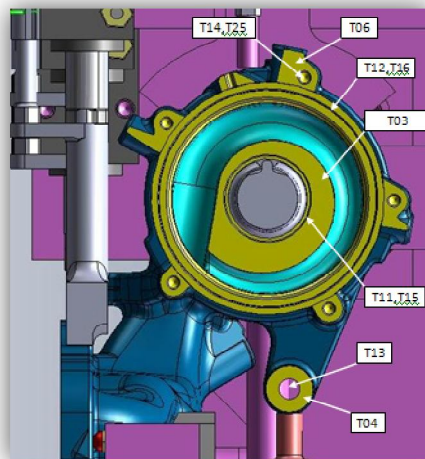
- Ø100 mm szerszámmal történő nagyoló síkmarás (T01)
- Ø32 mm szerszámmal végzett nagyoló lyukmarás (T03)
- Először Ø5.2 mm, majd Ø8 mm furat elkészítése (T17)
- Ø8.1 mm fúrómaróval történő megmunkálás (T19)
- Ø12 mm él letörős fúróval végzett megmunkálás (T20)
- Ø100 mm szerszámmal végzett simító síkmarás (T21)
- Ø22 mm szármaróval történő marás (T22)
- M6-os menet készítése (T25)



3. ábra: OP10-ben történő megmunkálás [1]

90°-os asztalpozíció esetén OP10-ben a következő megmunkálások történnek:

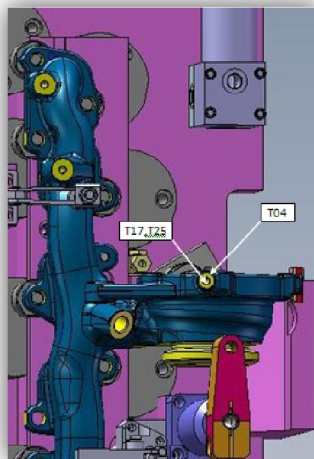
- Ø32 mm szerszámmal végzett lyukmarás (T03)
- Ø28 mm süllyesztővel történő megmunkálás (T04)
- Síkmarás Ø84 mm szerszámmal (T06)
- Fúrás, Ø40 mm, él letörővel ellátott fúróval (T11)
- Nagyoló kiesztergálás (T12)
- Ø10 mm fúróval történő megmunkálás (T13)
- Ø5.2-6.3-7 mm szerszámokkal végzett fúrás (T14)
- furatsimítás Ø41.03 mm-re (T15)
- furatsimítás Ø111.8 mm és Ø104.65 mm-re (T16)
- M6-os menet fúrása (T25)



4. ábra: OP10-ben történő megmunkálás [1]

170°-os asztalpozíció esetén OP20-ban a következő folyamatok zajlanak:

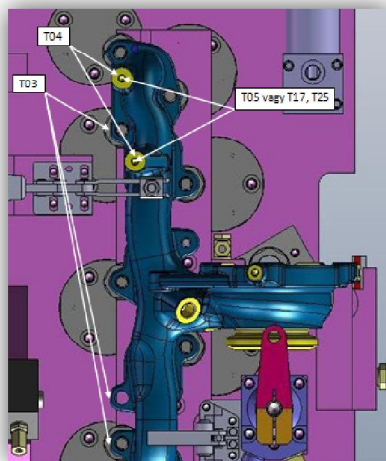
- Ø28 mm süllyesztővel történő megmunkálás (T04)
- Ø5.2 és Ø8 mm fúrókkal történő munkavégzés (T17)
- M6-os menet fúrása (T25)



5. ábra: OP20-ban történő megmunkálás [1]

180°-os asztalpozíció esetén OP20-ban a következő folyamatok zajlanak:

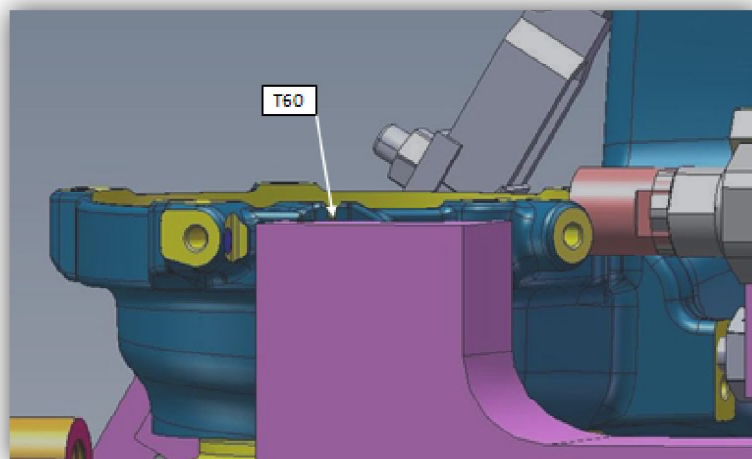
- Ø32 mm nagyolóval történő lyukmarás (T03)
- Ø28 süllyesztővel történő megmunkálás (T04)
- fúrás (T05) és (T17)
- M6-os menet fúrása (T25)



6. ábra: OP20-ban történő rögzítés [1]

197.5°-os asztal pozíció OP20-ban történő megmunkálás során:

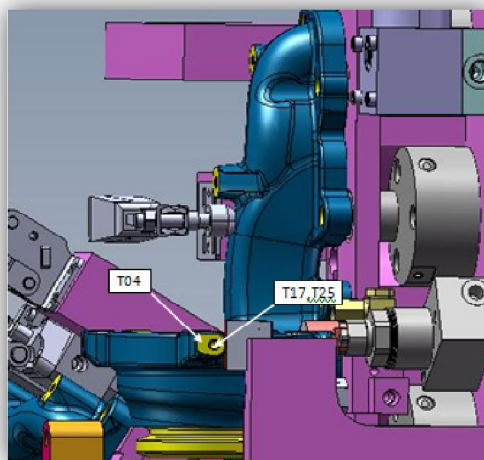
- Ø44 mm profilos tárcsamaróval végzett marás (T60)



7. ábra: OP20-ban történő folyamat [1]

258°-os asztal pozíció OP20-ban történő megmunkálás során:

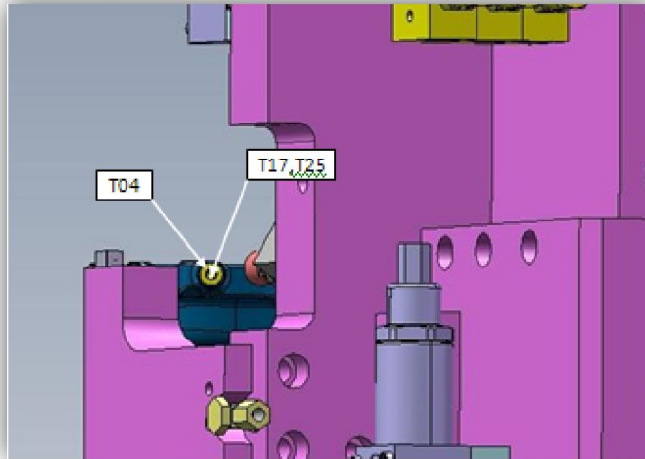
- Ø28 mm süllyesztővel történő megmunkálás (T04)
- M6-os menetes furat készítése (T25)
- Először Ø5.2, majd Ø8 mm furat készítése (T17)



8. ábra: OP20-ban történő megmunkálás [1]

333°-os asztal pozíció OP20-ban történő megmunkálás esetén:

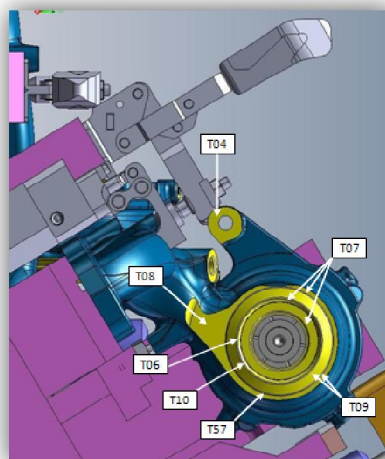
- Ø28 mm süllyesztővel végzett eljárás (T04)
- Ø5.2 majd Ø8 mm fúróval végzett forgácsolás (T17)
- M6-os menetfúrás (T25)



9. ábra: OP20-ban történő megmunkálás [1]

90°-os asztal pozíció OP30-ban történő megmunkálás esetén:

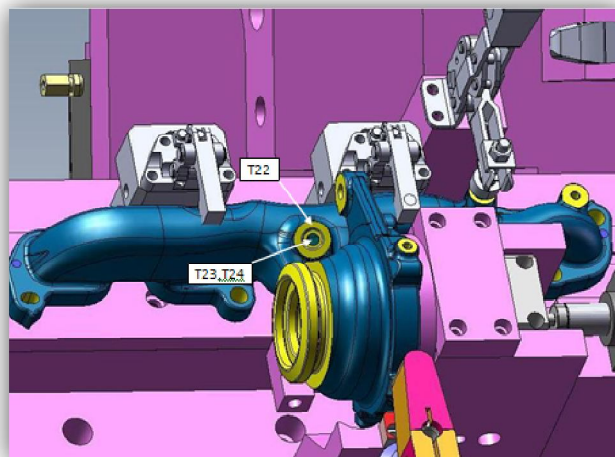
- Ø28 mm süllyesztővel végzett anyagleválasztás (T04)
- Ø84 mm síkmaróval történő marás (T06)
- Beszúró fúrás Ø78, Ø54.5, Ø40.7 mm (T07)
- Ø32 mm szármaróval történő megmunkálás (T08)
- Speciális külső simítás Ø62, majd Ø77 mm-essel (T09)
- Ø55.6 mm furatsimítóval végzett megmunkálás (T10)
- Ø78 mm harangmaróval végzett eljárás (T57)



10. ábra: OP30-ban történő megmunkálás [1]

161.78°-os asztal pozíció OP30-ban történő megmunkálás esetén:

- Ø22 mm szármaróval történő megmunkálás (T22)
- Fúrás: Ø16, Ø12.5, Ø8 mm (T23)
- M14x1.5-es menetes furat készítése (T24)



11. ábra: OP30-ban történő megmunkálás [1]

3. A munkadarab megfogása, készülékbe helyezése

A turbófeltöltőt korábban emberi közreműködés segítségével helyezték be az OP10, OP20, OP30 felfogatásokba. Azonban igény mutatkozott a gyártási darabszám növelésére, így ennek megvalósításához robotos kiszolgálást kíván alkalmazni a cég. Ahhoz, hogy elkészítsük a robotos manipulálással történő felrakási tervet, először ismernünk kell a kézi eljárás keretében véghezvitt folyamatokat. Vegyük sorra a 3 felfogatás lépéseit!

3.1 OP10-ben történő rögzítés

A készülékbe helyezés előtt vizuálisan ellenőrizni kell a darabot. Az eljárást minden esetben az érvényben lévő hibakatalógus és hibaértesítő alapján kell elvégezni úgy, hogy a darabot teljesen körülfordítsuk, az öntvényt átvizsgáljuk. Így csak minőségileg megfelelő darab kerülhet fel a gépre. Ha végeztünk a hibakereséssel következhet a készülék levegőpisztollyal történő megtisztítása. Oly módon kell a forgácsot eltávolítani, hogy az a gép belseje felé, a gyűjtő tálca irányában haladjon. Csak úgy tudom biztosítani a megfelelő felhelyezést, ha a készülék felfekvő felületein semmilyen szennyeződés nincs. Miután a kezdeti vizsgálatok megtörténtek, felhelyezhetjük a munkadarabot a készülékbe. A darabot két kézzel megbillentjük, így lehet a turbó részt felfűzni a támasztó elemre. Majd befelé és jobbra kell nyomni, hogy felfeküdjön a tájoló elemeken. Csak így biztosítható a jó megmunkálás, helytelen pozíció furatpozícionálási, talpvastagsági hibát okozhat, rosszabb esetben eltörhet a szerszám. Miután megtörtént a munkadarab berakása, következhet a levegős előszorítás bekapcsolása. Bal kezünkkel a talprész

beömlőnyílásainál megfogva a darabot a tájolóelemeken tartjuk, míg jobb kezünkkel az "OP10 Levegő" feliratú gombbal végrehajtjuk az előrögzítést. A kezünket a turbórész felső részére tilos helyezni, ugyanis a szorítóelem sérülést okozhat. Ezen eljárással biztosítható, hogy a munkadarab a megfelelő pozícióban maradjon a hidraulikus szorításkor és ezzel együtt a teljes megmunkálás alatt. A hidraulikus leszorítást az "OP10 hidraulika" és a biztonsági gomb együttes lenyomásával tudjuk bekapcsolni. A leszorítók működése idején a kéz nem lehet a gépben az ajtó síkján belül, így a balesetek elkerülhetőek. Ezért a hidraulikus leszorítás csak a két gomb együttes lenyomásával hozható működésbe. Továbbá fontos meggyőződni arról, hogy a nyomás eléri a 200 bar-t, ugyanis, ha ez nem valósul meg, akkor nem lesz tökéletes a munkadarab megfogása, ami súlyos balesetekhez, gépi sérülésekhez vezethet. Majd következik a paletta elforgatása az OP20- OP30-as művelethez. A forgatás előtt ellenőrizni kell, hogy az ajtó mögötti területen nincsen semmilyen zavaró elem, amely a forgatást akadályozná. Bal kézzel nyomva tartjuk a biztonsági gombot, míg jobb kezünkkel megnyomjuk az "OP20-OP30 pozíció" gombot. Végezetül megvárjuk a mozgás végét és csak azután folytathatjuk a munkavégzést.

3.2 OP20-ban történő rögzítés

Az OP20-as felfogatás során az OP10-ben végrehajtott kezdeti munkálatokat szintén végrehajtjuk. Miután megbizonyosodtunk mind a munkadarab hibátlanságáról, mind a készülék tisztaságáról következhet a munkadarab felhelyezése. Két kézzel a darabot a két tájoló csapra, illetve furatleszorító elemekre rakjuk és a kézi rögzítővel rögzítjük. Így biztosítható, hogy a turbófeltöltő a megfelelő pozícióban maradjon a hidraulikus szorításkor és ezzel együtt a teljes megmunkálás alatt. A következő lépés a munkadarab ellenőrzése, hogy a készüléklapon kotyogásmentesen felfeküdjön. Kézzel finoman megmozgatjuk a darabot, ha stabilan a helyén marad, akkor megfelelő, ellenkező esetben selejt keletkezhet. A mechanikus szorító alkalmazásával rögzítés következik. Miközben ráhúzzuk a szorítást a gyártmányra, a másik kezünkkel rá kell nyomni a tájolóelemeknek, hogy ne történjen elmozdulás, ugyanis az szintén selejtet eredményezhet. Végül a hidraulikus leszorítást az "OP20 hidraulika" és a biztonsági gomb együttes lenyomásával tudjuk bekapcsolni. A leszorítók működése idején a kéz nem lehet a gépben az ajtó síkján belül, így a balesetek elkerülhetőek. Ezért a hidraulikus leszorítás csak a két gomb együttes lenyomásával hozható működésbe. Továbbá fontos meggyőződni arról, hogy a nyomás

eléri a 200 bar-t, ugyanis, ha ez nem valósul meg, akkor nem lesz tökéletes a munkadarab megfogása, ami súlyos balesetekhez, gépi sérülésekhez vezethet.

3.3 OP30-ban történő rögzítés

Az OP30-as felfogatás során az OP10-ben végrehajtott kezdeti munkálatokat szintén végrehajjtjuk. Miután megbizonyosodtunk mind a munkadarab hibátlanságáról, mind a készülék tisztaságáról következhet a munkadarab felhelyezése. A turbófeltöltőt a csapágyház felőli oldalánál fogva a tájoló elemre húzzuk úgy, hogy közben a talprészt a készüléklapra illesztjük. Majd kézi leszorítóval rögzítjük. Így biztosítható, hogy a darab a megfelelő pozícióban maradjon a hidraulikus szorításkor és ezzel együtt a teljes megmunkálás alatt. Ezt követően ellenőrizzük, hogy a munkadarab a készüléken tökéletesen felfekszik-e, amit a segédcsapnál lévő hézag vizsgálatával valósíthatunk meg. A nem megfelelően felrakott darab selejtet okozhat. Következő lépésként a mechanikus szorító alkalmazásával rögzítjük a darabot a készüléken. Miközben ráhúzzuk a szorítást a gyártmányra, a másik kezünkkel rá kell nyomni a tájolóelemeknek, hogy ne történjen elmozdulás, ugyanis az szintén selejtet eredményezhet. A hidraulikus leszorítást az "OP30 hidraulika" és a biztonsági gomb együttes lenyomásával tudjuk bekapcsolni. A leszorítók működése idején a kéz nem lehet a gépben az ajtó síkján belül, így a balesetek elkerülhetőek. Ezért a hidraulikus leszorítás csak a két gomb együttes lenyomásával hozható működésbe. Továbbá fontos meggyőződni arról, hogy a nyomás eléri a 200 bar-t, ugyanis, ha ez nem valósul meg, akkor nem lesz tökéletes a munkadarab megfogása, ami súlyos balesetekhez, gépi sérülésekhez vezethet. Végül következik a paletta elfordítása az OP10-es művelethez, ez a palettacseréhez szükséges pozíció. A forgatás előtt ellenőrizni kell, hogy az ajtó mögötti területen nincsen semmilyen zavaró elem, amely a forgatást akadályozná. Bal kézzel nyomva tartjuk a biztonsági gombot, míg jobb kezünkkel megnyomjuk az "OP10 pozíció" gombot. Végezetül megvárjuk a mozgás végét és csak azután folytathatjuk a munkavégzést. Ebben a palettaállásban lehet a palettát cserélni.

3.4 Robotos manipulálás

A kézi megvalósítás ismeretében hozzáláthatunk a robotos kiszolgálás megtervezéséhez. Elsőként a robot kiválasztása történik, amit a FANUC Robotics gyártótól szerzünk be. Jelen esetben egy FANUC Robotics M-20iA típusú robotról beszélünk, amely főbb funkciói a következők:

A robot főbb sajátosságai: M-20iA

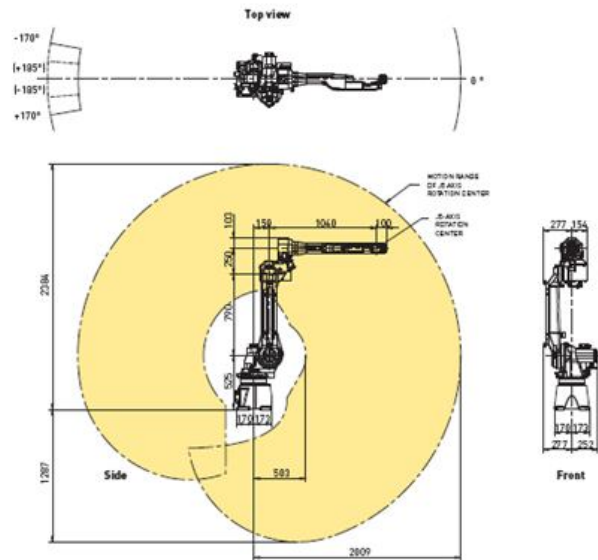
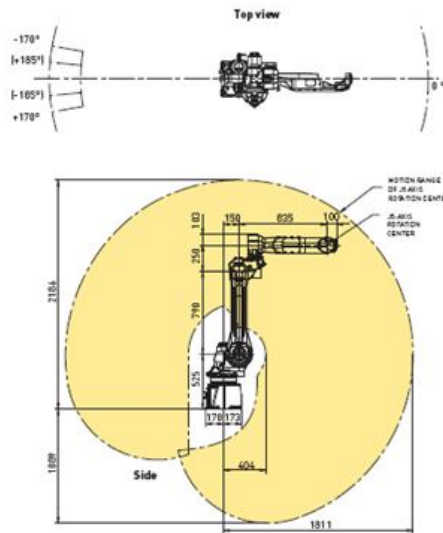
- Kinyúlás: 1811mm
- Ismétlési pontosság: +/- 0.08mm
- Terhelhetőség: 20kg

Az M-20iA már jóval nagyobb értéket képvisel elődjénél a megnövelt teljesítményének, megbízhatóságának és karbantarthatóságának köszönhetően. Főbb felhasználási területei:

- Pick and Place
- Összeszerelő alkalmazások
- Nagyobb munkadarabok mozgatása
- Anyag eltávolítás, mint sorjázás, vágás és polírozás
- Gépkiszolgálás: öntvény kisedés, fröccsöntő gép kirakodás
- Ragasztó adagolása

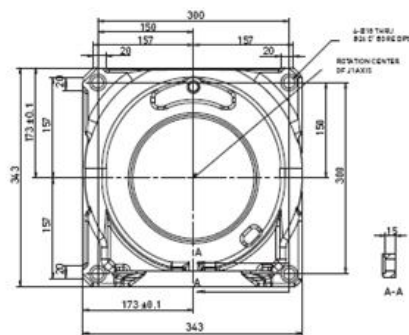
Az R-30iA vezérlővel az M-20iA robot intelligens funkciókra és mozgásteljesítményre képes. Az M-20iA széria a 20kg-os alapmodellt és a megnövelt kinyúlású 10L (karkinyúlás 2009mm) modellt kínálja. A karcsú kar és csukló kialakítás minimalizálja a rendszer egyéb elemeivel az ütközés esélyét és lehetővé teszi a robot működését szűk munkatérben is. [2]

M-20iA/10L, Arc Mate 120iC/10L

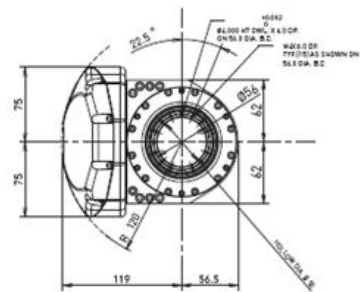


15

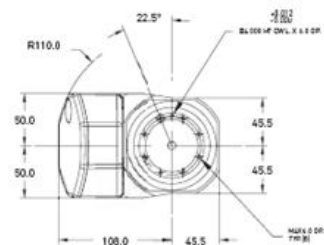
Footprint M-20:A, M-20:A/10L, Arc Mate 120:C, Arc Mate 120:C/10L



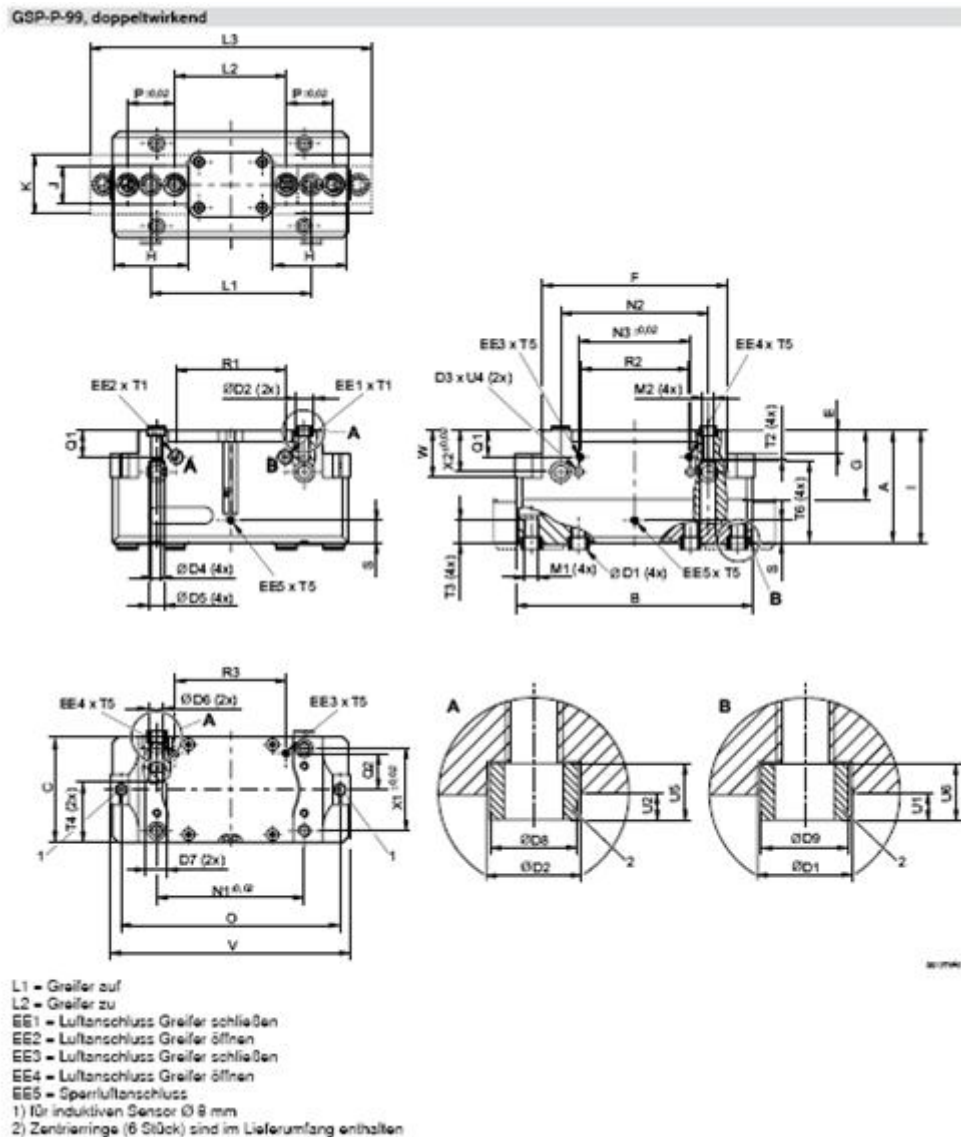
Wrist M-20iA, Arc Mate 120iC



Wrist M-20iA/10L
(M-10iA, M-10iA/6L, Arc Mate 100iC,
Arc Mate 100iC/6L, Arc Mate 120iC/10L)



A robot kiválasztását követő folyamat az optimális megfogó szerkezet meghatározása. Ez szintén összetett feladat, ugyanis több dologra figyelniünk kell. Figyelembe kell vennünk az adott munkadarab tömegét, a kialakítását, az alkalmazni kívánt működtetési módot, illetve a helyigényt is egyaránt. Mindezeket figyelembe véve 2 gyártó internetes oldalán bukkantam az elvárásoknak megfelelő szerkezetre. Elsőként a SCHUNK PGN-plus 100-2 0371152 típusjelzésű megfogóját véltem optimálisnak, majd a Bosch Rexroth-tól egy GSP-P-99 típusjelzésűt. Végül az utóbbira esett a választás, nemcsak azért, mert tulajdonságait tekintve jóval túlszárnyalta vetélytársát, hanem a cég is a Bosch-t szorgalmazta leginkább. Egy virtuális felületen megadhattuk a kívánt paramétereket, amik a munkadarab megfogási sajátosságaiból fakadnak, majd egy kalkulációt követően ezen eredményre jutottunk.



Greif- und Vakuumtechnik → Greifer

Serie GSP-P, Ausführung: hohe Werkstückmasse

► 2-Finger, Parallelgreifer ► mit Magnetkolben ► Hub/Finger: 16 - 25 mm ► Empf. Werkstückmasse: 8,2 - 13,5 kg



Typ	A	B	C	D1 H7	D2 H7	D3 H7	D4	D5	D6	D7	D8 h6	D9 h6
GSP-P-72	76	160	72	14	12	6	6,8	10,5	9	13,8	12	14
GSP-P-99	90	184	100	16	14	8	8,5	14	10,2	17	14	16

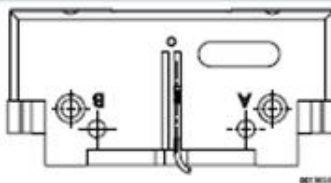
Typ	E	EE1	EE2	EE3	EE4	EE5	F	G	H	I	J	K
GSP-P-72	18	G1/8	G1/8	M5	M5	M5	125	48	50	77	26	40
GSP-P-99	22	G1/8	G1/8	M5	M5	M5	154	55	63	91	34	50

Typ	L1	L2	L3	M1	M2	N1	N2	N3	O	P	Q1	Q2
GSP-P-72	109	77	191	M10	M8	100	100	76	148,5	32	18	24,5
GSP-P-99	126	76	233	M12	M10	130	130	100	176	40	22	39,5

Typ	R1	R2	R3	S	T1	T2	T3	T4	T5	T6	U1	U2
GSP-P-72	74	74	75	15	7	20	17	41	6	55,5	4	3
GSP-P-99	82	82	82	22	7	20	17	23	6	70	4	4

Typ	U3	U4	U5	U6	V	W	X1	X2				
GSP-P-72	12	7	6,65	8,6	162,5	32	56	28				
GSP-P-99	-	12	8,6	8,6	190	38	70	34				

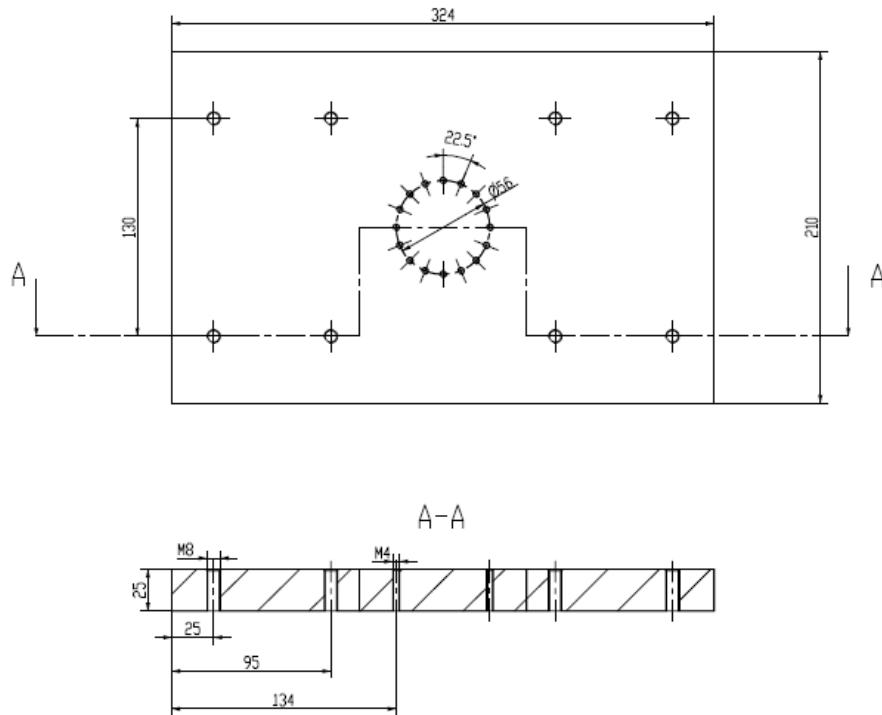
Klemmposition des Sensors



Nutbreite = 4 mm, geeigneter Sensor: ST4-2P

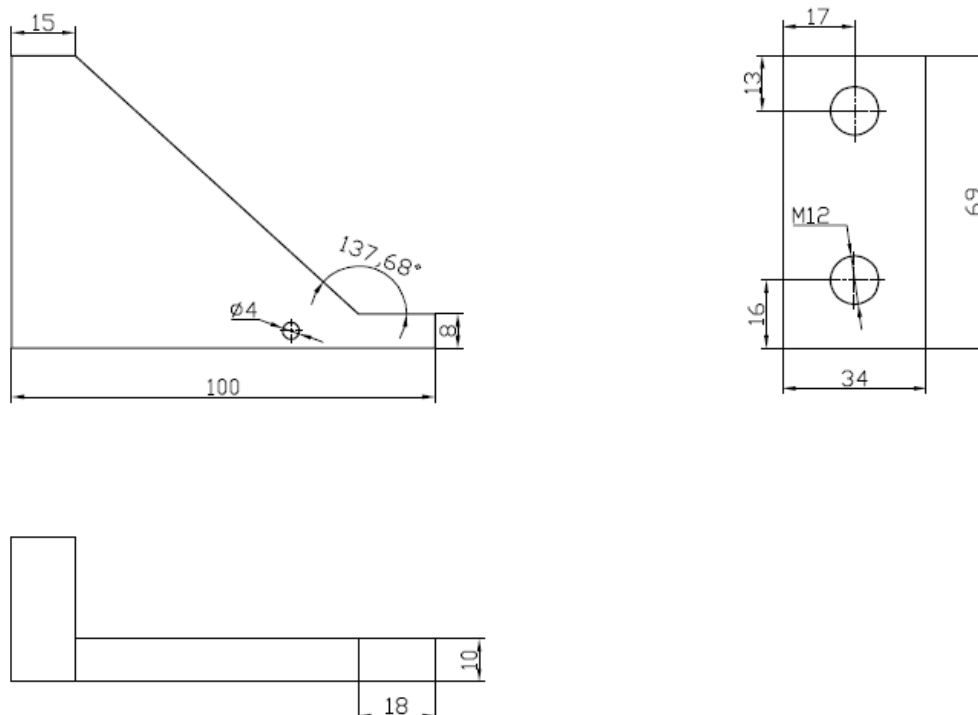
14-15. ábra: GSP-P-99 típusú megfogó [3]

Miután kiválasztottuk a megfelelő robotot, illetve megfogót, következhet a csatlakozó elemek megtervezése. Mindenekelőtt több videót, fényképet megnéztem, amin végigkövettem a gyártmány manuális úton történő készülékbe helyezését, majd a megmunkálást. A munkadarabot és a 3 felfogást biztosító készüléket testközelből analizáltam, méreteket vettem. Miután minden adat a rendelkezésemre állt nekiláttam megtervezni az adaptert. Ezen alkatrész rácsatlakozik a robot karjának végére, illetve erre helyezhetők fel a megfogók is. Egy alumínium lemezből kell kivágni a 324x210x25 mm-es paraméterekkel rendelkező téglalap alakú hasábot. Ezt követően egy, a robot csatlakozófelületével kompatibilis furatkiosztást kell rajta kialakítani. A másik oldalán 2 felfogató helyet a 2 megfogónak, ugyanis a stabilitás érdekét figyelembe véve 2 megfogót kell alkalmazni, amelyek párhuzamos működtetésűek lesznek, levegőhajtással.



16. ábra: Adapter műszaki rajza

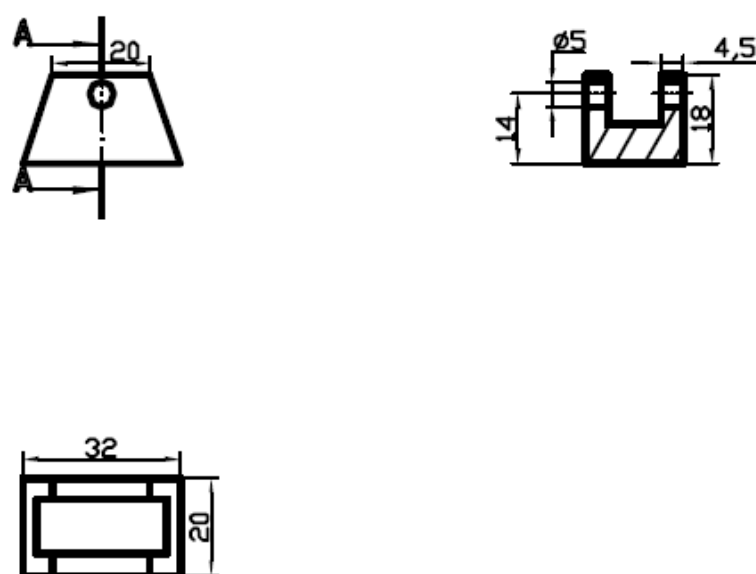
Az adapter megalkotását követően következett a megfogó ujjak megtervezése. Ezen ujjaknak csatlakozniuk kell a Boschtól rendelt megfogókhoz, így két darab M12-es menetes furatot kell készíteni rá a felfogás biztosítása érdekében. Továbbá a hosszas analízis után nyilvánvalóvá vált az ujjak méreteinek igénye. A 3 felfogás közti manipulálás során 1-szer simított felületeket, 2-szer pedig szabálytalan felületeket kell megfogni. Ezen követelménynek eleget téve úgy alakítottam ki az ujjakat, hogy azok képesek legyenek ezt a feladatot ellátni. Egyrészt recézett felületű ujjkialakítással, másrészt pedig egy beálló megfogó megoldással igyekeztem megvalósítani a tervezetet. Ezt a szabálytalan felülethez való alkalmazkodást egy csapra illesztett billenő résszel igyekeztem megoldani.



17. ábra: Megfogó ujj műszaki rajza

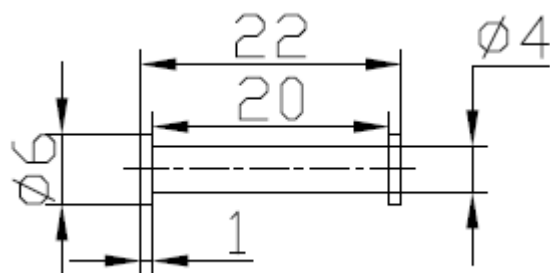
(2 fajta van, attól függően, hogy a furatok melyik oldalán helyezkedik el a fogó rész)

A beálló pofa kialakítása teljes mértékben olyan, hogy könnyedén alkalmazkodhasson a meglévő felülethez, továbbá biztosítsa annak stabil megfogási lehetőségét is.

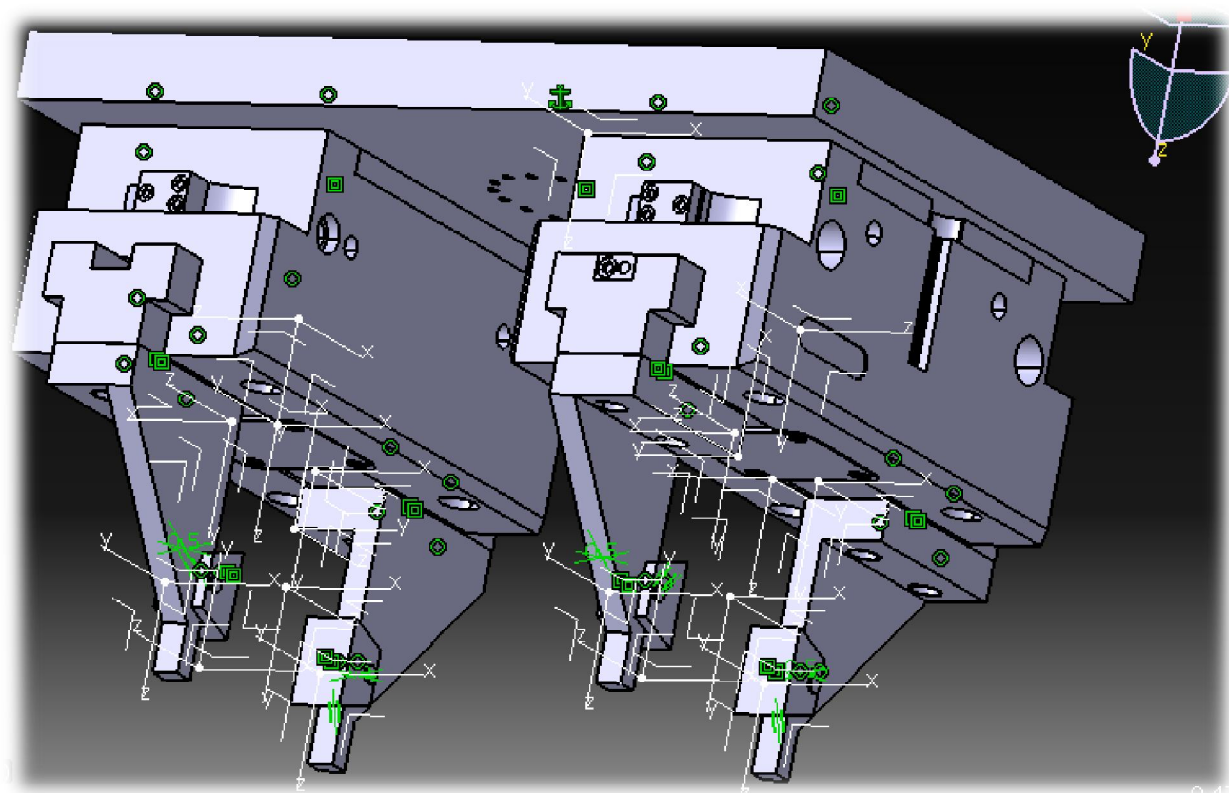


18. ábra: Billenő fogó műszaki rajza

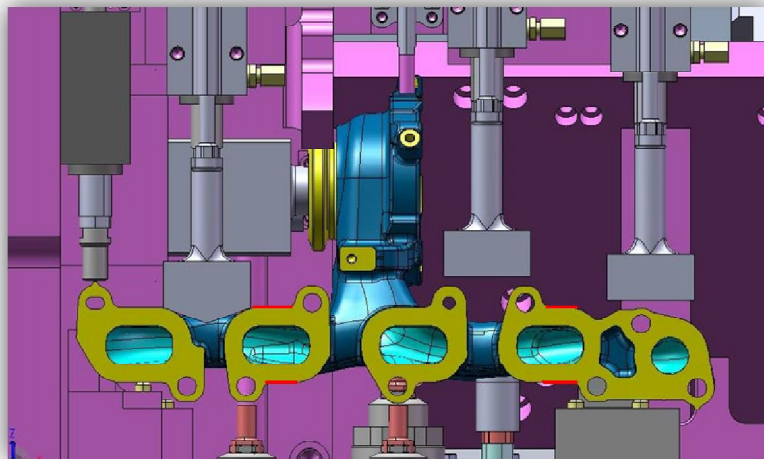
A csapot, amely a pofa mozgathatóságát biztosítja a következő ábrán láthatjuk:



19. ábra: Illesztő csap műszaki rajza

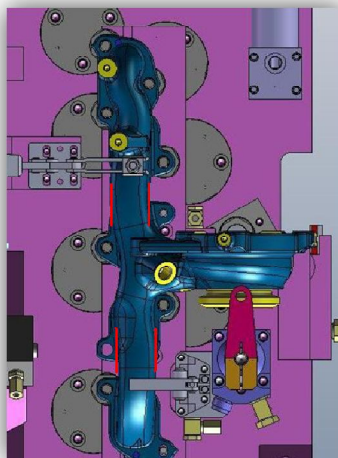


20. ábra: Az összeállítás modellje



21. ábra: OP10-ben történő megfogás [1]

A gyártmány megfogása az ábrán piros vonallal jelölt felületen történik, mikor az OP10-ből az OP20-ba, illetve egy tárolóból az OP10-be kerül a darab. Ezen esetekben a recézett felületű megfogó ujj végzi a munkafolyamatot.



22. ábra: OP20-ban történő megfogás [1]

A gyártmány megfogása az ábrán piros vonallal jelölt felületen történik, mikor az OP30-ból egy tároló rekeszbe, illetve az OP20-ból az OP30-ba történik az átrakás. Ezen esetekben a billenő megfogó végzi a manipulálást.

A feladatkiírásban gépsorra történő robotizálás volt feltüntetve, azonban a cég csupán a készülékben történő átrakást követelte meg. Az általam elkészített összeállításhoz szükséges még egy jelenlét érzékelő szenzor is, amely tájékoztat arról, hogy található-e munkadarab a megfogóban, avagy nem. Továbbá a stabilitás megőrzésének érdekében egy

támasztó csapot is ráhelyezhetünk az adapterre, amely segít abban, hogy ne forduljon ki a munkadarab a manipulálás során.

A megfogási folyamat rövid leírása:

Az OP30-as felfogatásból a robot kiveszi a gyártmányt, majd behelyezi egy tároló részbe a kész munkadarabot. Ezt követően kiveszi az OP20-as felfogatásban lévő gyártmányt és belerakja az OP30-asba. Továbbá az OP10-es részből az OP20-ba helyezzük a munkadarabot. Az első fázis pedig az OP10-be történő elhelyezés. Az OP30 és OP20 felfogatás közötti manipulálás a billenő pófák segítségével történik. Az OP10-es esetben vesszük igénybe a másik részét a megfogó ujjaknak. Így tudjuk biztosítani a hibamentes feladatvégzést.

A folyamat időigényét a gépek közti távolságok, illetve a robot mozgási sebességének ismeretében könnyen meghatározhatjuk.

4. Kidolgozott javaslatok a gyártórendszer telepítésére

4.1 A gyártórendszer elemeinek ismertetése

A turbófeltöltő automatizált megmunkálásához, illetve a rendszerben történő szállításához rengeteg gépre van szükség. Az általam elkészített elrendezési vázlatban hét darab megmunkáló központ, egy darab automatizált munkadarabmosó, egy darab szivárgástereszt berendezés, egy darab sorjázó gép, illetve nyersdarab beléptető és készdarab lerakó helyek kerültek elhelyezésre. A berendezések helyszükségletei rendelkezésemre álltak, ennek függvényében állíthattam össze egy elrendezési javaslatot. A gépek kiszolgálását 3 darab önálló portál rendszerű robot segítségével szeretnénk megvalósítani. A hét megmunkáló központot kiszolgáló két robotnak FANUC M20i-A típusú robotot, míg a többi folyamat kiszolgálását végző robotnak pedig FANUC M10i-A típusú robotot választottam.

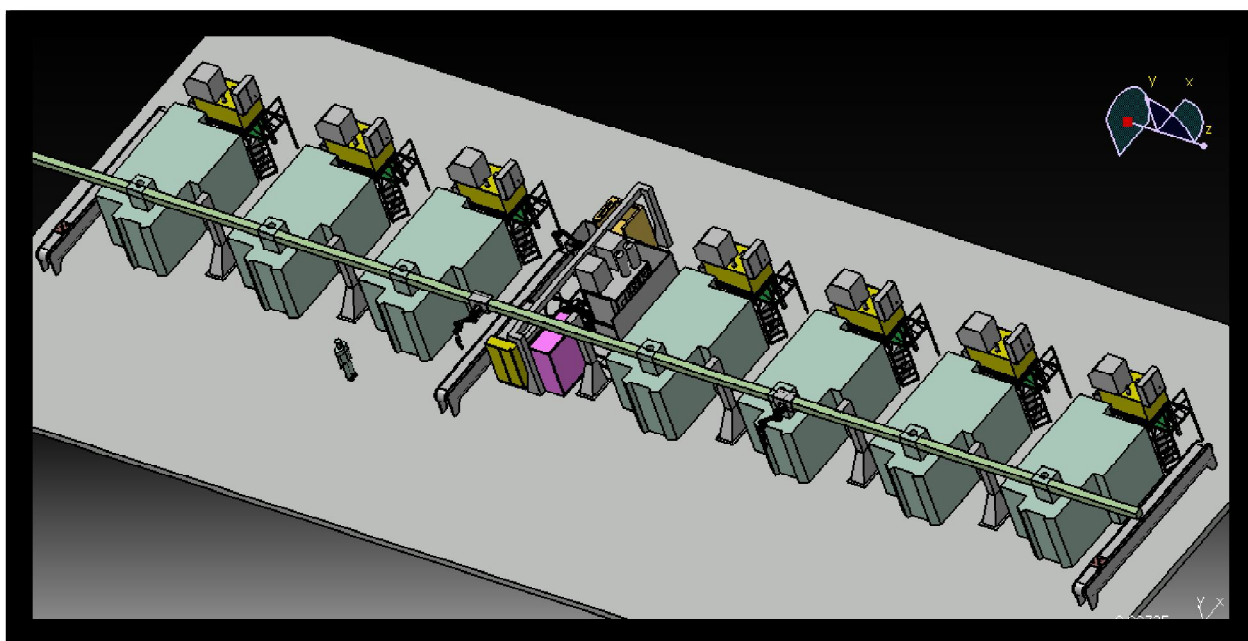
4.2 A gyártórendszer elrendezési koncepciójának ismertetése

A legkézenfekvőbb koncepció egy sorban letelepített elrendezés kialakítása. Ugyanakkor nem szabad figyelmen kívül hagyni annak a lehetőségét sem, hogy tömbösített formában történjen a gépek elhelyezése. Jelen esetben a soros változatot részesítettem előnyben. Miután meghatároztuk az elrendezés lehetséges módját, következhet a berendezések soron belüli helyének meghatározása. A nyersdarab beléptetése szállítószalagok segítségével, palettákon való elhelyezéssel történik. A szalagok elhelyezésére két lehetőséget véltem megvalósíthatónak. Egyrészt pozícionálhatóak lennének a sor közepére, illetve másik lehetőségként a sor két szélére. A biztonságot, illetve a hatékonyságot figyelembe véve arra a megállapításra jutottam, hogy a nyersdarab behozatalára hasznosított szalagokat a sor két szélére kell telepíteni. Ekkor ugyanis a gépeket kiszolgáló két robot nem egy helyről venné fel a munkadarabokat, így annak valószínűsége, hogy esetleges meghibásodás következtében a két kiszolgáló egység ütközne, minimálisra redukálódott. Továbbá nincs szükség arra, hogy várakozási pozícióba kerüljön a robot, míg a másik felveszi a beérkező nyersdarabokat. A megmunkáló központokat célszerűen a két robot maximális kihasználtságát figyelembe véve három-négy arányban, egymás mellett helyezzük el. A félkész darabokat szintén szállítószalag segítségével jutatjuk el a további gépekhez. Így a lehető legjobb elrendezés úgy alakul, hogy ezen szalag fog a sor közepén elhelyezkedni. A munkadarabokat ismételten egy portál rendszerű robot fogja mozgatni. A sorjázó gépet, az automatikus munkadarabmosót, illetve a szivárgásteszt berendezést célszerű szintén egy sorban egymás mellett elhelyezni, méghozzá úgy, hogy figyelembe vesszük a gyártási folyamat sorrendjét. Így az előbb említett formában történik azok elhelyezése. Miután elkészült a darab egy tárolóba történik a behelyezése, ahonnan majd a későbbiek során elvitelre kerül. Átmeneti tárolók kialakítására több helyen is igény mutatkozhat. Elsősorban a robot közvetlen környezetében, ugyanis a stratégia optimalizálása végett így egyszerre több nyersdarabot, majd félkész darabot képesek a robotok szállítani, ezáltal csökkenhet a gyártási idő. Továbbá átmeneti tároló kialakítására kerülhet sor a félkész darabok szalagra helyezése előtt, ugyanis, ha felhalmozódik a szalagon a termék, a robotok nem lesznek képesek az általuk szállított darab letételére. Ezáltal a folyamat megáll, nem tud folytatódni. Ahhoz, hogy ezen feltevést igazoljuk, kizárólag a rendszerről elkészített szimulációt hívhatjuk segítségül. Az eredmények ismeretében módosíthatjuk az elrendezési koncepciót, amelyről ismételt szimulációt tehetünk.

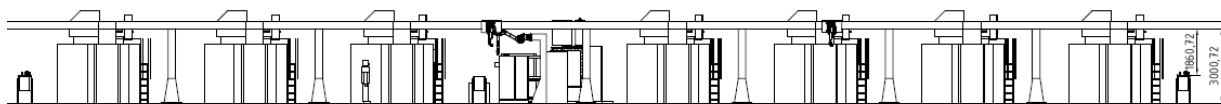
Így a gyártás szempontjából legoptimálisabb elrendezést valósíthatjuk meg.

4.3 A gyártórendszer működésének folyamata

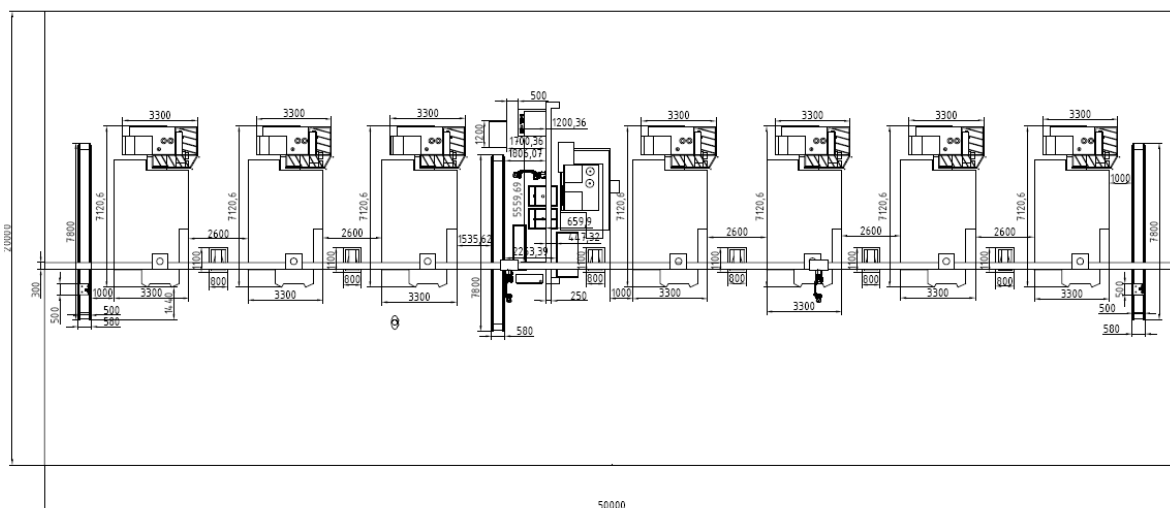
Az előző fejezetben részletesen ismertettem egy elképzelt elrendezés koncepcióját. A következőkben a folyamat leírása kerül bemutatásra egy adott stratégiával. Természetesen a szimuláció segítségével több stratégia kidolgozásra kerül, amelyek közül a legoptimálisabbat kiválasztjuk. A nyersdarabok behozatala a megmunkáló részlegbe szállítószalagok segítségével történik. A turbófeltöltők rögzítésére és tájolására palettákat alkalmazunk. Amikor a két szalagon elegendő darab halmozódott fel, illetve elérték a robot aktív munkaterébe a két FANUC M20i-A típusú robot három, illetve négy darabot vesznek fel egy a későbbiekben kidolgozásra kerülő átmeneti tároló rekeszbe. Ezt követően megkezdődik a turbófeltöltők megmunkáló központokban való elhelyezése. Mindkét esetben, sorban halad a robot, folyamatosan lépteti be és rögzítetteti a munkadarabot a kívánt felfogásban. Jelen esetben az OP10 névre hallgató felfogatásban kezdődik a megmunkálás. Miután kiürítették a tároló rekeszt, visszapoziционálnak a szélső géphez és amikor az végzett, a megmunkált darabot áthelyezik az OP20-as felfogatásba. Ismételten sorban történik a darabok beültetése, majd amikor a végére ér a sornak, visszapoziционálás történik. Ezt követi az OP30-ban történő rögzítés. Miután elkészültek a darabok a három felfogásban, a robot összegyűjti azokat és a közepén található szállítószalagra helyezi. Abban az esetben, amikor telítődik a rendszer, igénybe vehetjük az átmeneti tárolót, ahol az üressé válásig tárolódhatnak a félkész darabok. Eközben a gyártórendszer középső részén elhelyezkedő FANUC M10i-A típusú robot leveszi a félkész darabokat a szalagról és első lépésként a sorjázóba helyezi. Miután befejeződött a sorjázás a robot a mosóba helyezi egy keretre. Ezt követően megkezdődik a mosási folyamat, míg eközben a sorjázás párhuzamosan zajlik. A mosás befejeztével egyesével kerülnek a darabok a szivárgásteresztelő berendezésbe, ahonnan az elkészített termékek egy tárolóba kerülnek. Később innen a megfelelő raktározási környezetbe kerülnek. A folyamat összetettsége miatt, továbbá a párhuzamos megmunkálás következtében nem kizárt átmeneti tárolók beiktatása sem. Mindezt szintén az automatizált gyártórendszerrel elkészített szimuláció segítségével tudjuk megvizsgálni.



23. ábra: A gyártórendszer elrendezési koncepciója



24. ábra: A gyártórendszer 2D-s rajza



25. ábra: A gyártórendszer 2d-s rajza

5. Plant Simulation szimulációs program ismertetése

5.1 A program áttekintése

A Siemens PLM Software egyik, az iparban rendkívül elterjedt alkalmazása a Plant Simulation (korábban: eM-Plant). A program alapvetően az iparban előforduló termelési és logisztikai folyamatok optimalizációjára, illetve szimulációjára szolgál, vagyis anyagáramlás szimuláció végrehajtásra alkalmas rendszer. Klasszikus értelemben anyagáramlásnak nevezzük az üzemekben történő anyag, gyártmány áramlását (pl. szállítószalagon vagy akár járművön). A program használatával a gyártócellák működésétől kezdve a teljes gyárkomplexumok modellezéséig a termelés különböző szintjei könnyen modellezhetők. Ilyen esetekben a három legfontosabb kérdés, ami miatt szimulációt érdemes végrehajtani:

- Hány munkás, gép, tároló cella, stb. szükséges?
- Logisztikailag ki mit, mikor készít el?
- Elrendezés szempontjából hol van a legjobb helye a gépeknek és a dolgozóknak?

Mivel a Plant Simulation egy objektumorientált alkalmazás, ennek köszönhetően a fejlesztők által beépített, vagy a felhasználó által létrehozott objektumokból építhetők fel a program modelljei. A modell működését nagymértékben befolyásolják az úgynevezett metódusok, illetve programtöredékek. Az iparban egy felépített rendszer működésének vizsgálatát, illetve valamilyen szempont(ok) szerinti vizsgálatát, optimalizációját a Plant Simulation programmal hajthatjuk végre. A Plant Simulation nem más, mint egy diszkrét, esemény-vezérelt szimulációs rendszer, ami azt jelenti, hogy a folyamatosan előre haladó időben a rendszer csak a valamilyen szempontból lényeges események diszkrét pillanatait vizsgálja. Ilyen értelemben lényeges eseményről beszélhetünk például, amikor egy darab megérkezik egy szállítószalag elemre, vagy amikor elhagyja azt. A szimulációk segítségével a gyártási folyamatban kimutathatók az alacsony vagy túlzott raktárkészletek, az alul vagy túlterhelt szállítószalagok, és a rendszer szűk keresztmetszetei, amelyek a gyártás hatékonyságának visszaesését eredményezhetik. A rendszer szabad kezet ad sok mindenben, így készíthetünk saját elemeket is a 2D-s, illetve 3D-s szimulációk kész alapelemei mellé. A vállalatok számára komoly költségmegtakarítást tud előidézni egy adott szimuláció. Például 2002-ben a BMW müncheni gyárában hajtottak végre egy elemzést a kamionforgalom csökkentésére a gyár területén Plant Simulation alapon,

aminek az eredményeként 10%-kal csökkent a kamionok száma a gyár területén, 12,7%-kal csökkent a kamionok által a gyár területén töltött idő, és jelentősen csökkentek a belső közlekedés várakozási idejei.

A Plant Simulation segítségével a gyártás logisztikája modellezhető a rendszer viselkedésének elemzésére és teljesítményének optimalizálására. A számítógépes modell lehetőséget nyújt különböző változatok kipróbálására, vagy "mi lenne ha" jellegű vizsgálatokra a meglévő gyártósor megbontása, vagy a tervezett gyártósor megépítése előtt. A számos elemzőeszköz, statisztika és grafikon lehetővé teszi a különböző gyártási helyzetek elemzését, és gyors, megfelelően alátámasztott döntések meghozását a gyártástervezésnek már a korai szakaszában. Segít azoknak a problémáknak a felismerésében és elkerülésében, amelyek később drága és időigényes javítást igényelnének a gyártás során. Minimalizálja az új gyártósor költségét a termelékenységének csökkentése nélkül. Optimalizálja a meglévő rendszerek teljesítményét olyan paraméterekkel, amelyek eredménye a szimulációban jól ellenőrizhető a valós elkészítés előtt.

5.2 A program funkciói

- Segítségével képesek vagyunk komplex gyártórendszerek és vezérlési rendszereik szimulációjára
- Segítségével könnyedén elkészíthető objektum-orientált, hierarchikus gyármodellek, beleértve az üzleti logikát, a logisztikát és a gyártási folyamatokat is
- Rendelkezik objektum könyvtárakkal a legjellemzőbb modellezési feladatok gyors, hatékony elvégzéséhez
- Rendelkezésre állnak grafikonok és diagramok a kimenet, az erőforrások és a szűk keresztmetszetek elemzéséhez
- Rendelkezésre állnak átfogó analíziseszközök, beleértve az automatikus szűk keresztmetszet elemzést, a Sankey-diagramokat és a Gantt-grafikonokat is egyaránt
- 3D-s megjelenítő és animáló képesség
- Integrált tapasztalatkezelési lehetőségek
- Genetikus algoritmus a rendszerparaméterek optimalizálásához
- Nyitott architektúra, integrációs lehetőségek (ActiveX, CAD, Oracle, SQL, ODBC, XML, stb.)

5.3 A program előnyei

- Meglévő gyártósorok termelékenységének akár 15-20%-kal történő növelése
- Új gyártósorok tervezési költségeinek csökkentése akár 20%-kal
- A gyártási idő és a raktárkészlet csökkentése 20-60%-kal
- Rendszerméretek optimalizálása, beletartoznak a pufferméretek is
- Tervezési veszélyek elkerülése a koncepcióalkotás fázisában
- Gyártási erőforrások használatának maximalizálása
- Gyártósor tervezésének és elkészítésének gyorsítása

5.4 Felhasználása az iparban

A Plant Simulation-t az ipar számos területén alkalmazzák:

- Járműgyártás
 - Hajó gyártás
 - Gépjármű gyártás
 - Repülőgép gyártás
- Űripar
- Hadiipar
- Fogyasztási cikkek gyártása

[4] és [5]

6. Az elkészített elrendezési javaslat szimulációja

6.1 Szimulációs követelmények

A korábbiakban felvázolt automatizált gyártórendszer elrendezési javaslatát követően segítségül hívhatjuk a Plant Simulation nevű programot, annak vizsgálatára, illetve optimalizálására. A program segítségével könnyedén meghatározható, hogy képesek-e a robotok az adott elrendezési formában kiszolgálni a gépeket, továbbá szükség van-e átmeneti tároló helyek kialakítására. Diagramok segítségével látványosan kimutatható a vizsgálat eredményei, majd következtetéseket vonhatunk le. Így még a valódi telepítést megelőzően fény derülhet az esetleges pontatlanságokra, illetve javítható a gyártási idő, ennek függvényében növelhető a gyártási darabszám.

6.2 A szimuláció alapegységeinek beillesztése

A program elindítását követően létrehozunk egy új Frame-t, azaz egy új fájlt. Ezt követően elmentjük a létrehozott szimulációt egy tetszés szerinti, könnyen visszakereshető néven. Miután elkészítettük a szimulációs modellünk alapját, következhet az alapegységek beillesztése a modelltérbe.

Beillesztett elemek:

➤ EventController

Az EventController koordinálja és szinkronizálja a szimuláció futása közbeni eseményeket. Megadható a Settings fülön, hogy mikor kezdődött, illetve, hogy mikor ért véget az adott szimuláció. A Controls fülön lehet a sebességet manipulálni, azaz, hogy milyen gyorsan menjen végbe a szóban forgó folyamat.



25. ábra: EventController [6]

➤ ShiftCalendar

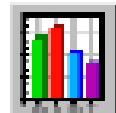
A ShiftCalendar az üzemben lévő különböző műszakok modellezésére szolgál. Egy műszakban ki lehet kapcsolni a gépeket, továbbá meghatározhatunk ünnepnapokat is, amikor nem történik munkavégzés.



26. ábra: ShiftCalendar [6]

➤ Chart

A Chart statisztikai adatok megjelenítésére szolgál, lehetővé teszi az aktuális értékek és szimulációs eredmények gyors elemzését.



27. ábra: Chart [6]

➤ Source

A Source hozza létre az MU-kat (mozgó egységek), a megadott paraméterek alapján.



28. ábra: Source [6]

➤ Drain

A Drain eltávolítja az MU-t az anyagáramból, amint a megadott set-up time (beállítási idő) és processing time (feldolgozási idő) eltelik.



29. ábra: Drain [6]

➤ Entity

Az Entity egy olyan objektum, amit a Source hoz létre, az anyagáramlás objektumok dolgoznak fel, és a Container és Transporter objektumok szállítanak. Az Entity nem szállíthat, nem tartalmazhat másik alkatrészt vagy elemet.



30. ábra: Entity [6]

➤ Container

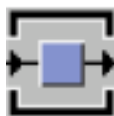
A Container egy MU amely elemeket (Entity) szállít és más Container objektumokat. Ez egy palettát reprezentál. A Container használható elemként is. A Containeren lévő Container szállíthat más elemeket vagy konténereket.



31. ábra: Container [6]

➤ SingleProc

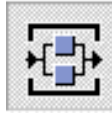
A SingleProc egy MU-t fogad, feldolgozza azt és továbbmozgatja a következő állomásra az anyagáramban. Ha az MU típusa különbözik a korábbi típustól, akkor a SingleProc objektumot fel kell készíteni a feldolgozásra.



32. ábra: SingleProc [6]

➤ ParallelProc

A ParallelProc több cellával rendelkezik, hogy az MU-kat párhuzamosan képes legyen feldolgozni. A feldolgozási időt bármikor meg lehet változtatni. Ez nem fogja érinteni a ParallelProc másik celláján lévő MU-kat. Ennek következtében az MU-k megelőzhetik egymást.



33. ábra: ParallelProc [6]

➤ Line

A Line egy szállítószalagot reprezentál. A hossz, illetve a sebesség határozza meg azt az időt, ami alatt az MU-k végighaladnak rajta. Az MU-k feltorlódását szintén lehet engedélyezni.



34. ábra: Line [6]

➤ PortalCrane

A PortalCrane segítségével ültethetünk be portál robotos megoldást a szimulációba. Egyszerre csak egy elem szállítására képes.



35. ábra: PortalCrane [6]

➤ PlaceBuffer

A PlaceBuffer több sorban lévő cellával rendelkezik. Tetszés szerinti számú MU-t tud feldolgozni, illetve tárolni. A feldolgozási időt a teljes PlaceBufferre lehet megadni, nem egyesével a cellákra.



36. ábra: PlaceBuffer [6]

➤ Store

A Store esetén az MU-k addig rajta maradnak, amíg onnan egy metódus el nem távolítja őket. A Plant Simulation a beérkező MU-kat tetszés szerinti szabad cellákra helyezi, ha megtelt a Store nem fogad több MU-t.



37. ábra: Store [6]

➤ Method

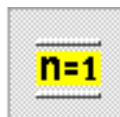
A Method egy kisebb program, amely hasonlít a Basic, Pascal, vagy C++ nyelvek függvényeihez. Képes a szimuláció közben történő eseményekre való reagálásra, különböző jellemzők lekérdezésére és beállítására, kifejezések futtatására, objektumok módosítására továbbá új felhasználók számára egyéni párbeszédablak készítéshez.



38. ábra: Method [6]

➤ Variable

A Variable objektum értékek tárolására szolgál a szimuláció hosszabb idejére. Ugyanúgy meghívható, mint bármely más objektum. Az értékek lehetséges tartománya az adattípustól függ. Meghívható más változók vagy Method-usok által.



39. ábra: Variable [6]

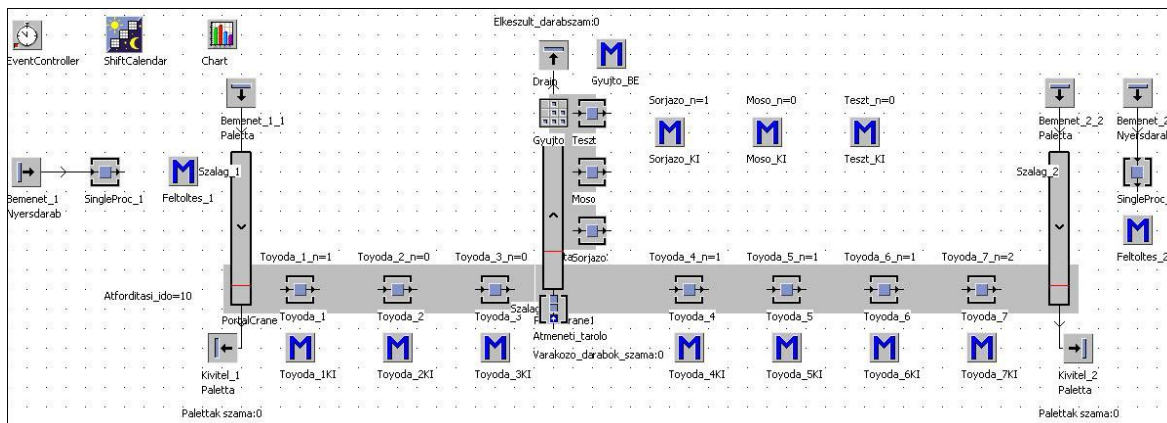
6.3 A szimulációs folyamat ismertetése

Miután elkészítettük a gyártórendszerünk 3D-s modelljét, következhet a szimulációs program használata. Az adott elrendezéshez kétféle stratégiát dolgoztam ki, amelyeknél a lényeges eltérés abban nyilvánul meg, hogy alkalmazunk-e a robot környezetében egy „rekesznek” nevezett átmeneti tárolót, amelybe a darabokat felhalmozzuk, így könnyebb megvalósítani a darabutánpótlást. Ennek következtében jobb gyártási eredményeket várunk.

6.3.1 Robotrekesz nélküli stratégia

A beáramló nyersdarabokat két külön elnevezéssel láttam el, attól függően, hogy a gyártósor melyik oldaláról érkezik. Ennek függvényében a bal oldali szalagon érkező darabokat Nyersdarab_1_1-nek, míg a jobb oldalról érkezőket Nyersdarab_2_1-nek neveztem el. Ezen nyersdarabok a szintén eltérő nevet viselő palettákon rögzítve érkeznek be a megmunkáló térbe. Ehhez hasonlóan a baloldaltól érkező palettákat Paletta_1-nek, a jobb oldalról érkezőket Paletta_2-nek neveztük el. A nyersdarabokat egy Feltoltes_1 (1. melléklet), illetve Feltoltes_2 (2. melléklet) nevű methodokkal helyezzük rá a szalagon áramló palettákra. A szállítószalag hossza 7,8m. 6.8m távolságban egy OnSensore (17. melléklet és 18. melléklet) nevű szenzort helyeztem el, amelynek segítségével történik a nyersdarab robotra való helyezése. Ugyanis amikor elérkeznek a szenzorig, az megállítja a sort, a robotot hívja, felveszi, majd a paletta elnyelődik, míg a robot megkezdí a munkáját. Először a Toyoda_1, majd a Toyoda_2 és végül a Toyoda_3 nevű gépekre történik munkadarab elhelyezés. Amíg nincsen nyersdarab a gépeken, addig a Toyoda_1_n, Toyoda_2_n, Toyoda_3_n nevű változók 0 értéket mutatnak. Miután ráhelyeztünk egy darabot, az automatikusan átvált 1-re. Ezt követően a Toyoda_1_KI (3. melléklet), Toyoda_2_KI (4. melléklet), Toyoda_3_KI (5. melléklet) methodok segítségével történik a 3 felfogás közti manipulálás vezérlése. A megmunkálási időt a Processing time segítségével határozhatjuk meg. Miután megtörtént a 3 felfogásban történő megmunkálás a változók 3-ra váltanak, a Nyersdarab_1_1 név pedig átalakul Felkeszdarab névre. Ezt követően a félkész darabokat egy átmeneti tárolóba viszi el a robot, ahonnan azok a szállítószalagra kerülnek. Ezek után lenullázódik a változók értéke és a folyamat kezdetét veszi előről. A jobb oldali munkavégzés hasonló az előbbieken említettekhez, csupán ott a 3 gép helyett 4-t kell kiszolgálni, de maga a gondolkodásmód teljes mértékben azonos. Az átmeneti tárolóban folyamatosan várakoznak a darabok, amiknek a számát egy változó segítségével szemléltethetjük. Ennek neve: Varakozo_darabok_szama. A középső szalag szintén 7.8m hosszúságú, 1.95m-es hosszúságban elhelyeztem egy OnSensore-t (19. melléklet) amely az ottani munkafolyamatokat végzi. Így amikor a Sorjazo gép nem tartalmaz darabot a robot odapozicionál és egyet beletesz. Majd a Sorjazo_KI (10. melléklet) method segítségével áthelyezi a Sorjazott_darab-ra keresztelt terméket a Mosóba. Ezt követően a Moso_KI (11. melléklet) parancs miatt át fog helyeződni a darab a Teszt gépre, amely akkor már Mosott_darab néven kerül behelyezésre. Végül a Teszt gépről egy

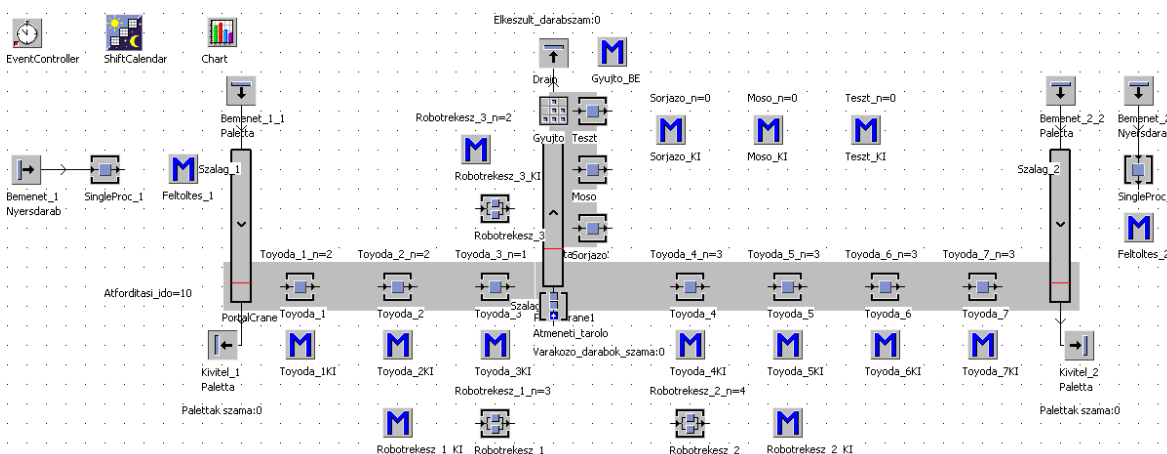
gyűjtőbe kerül a Keszdarab-ra keresztelt turbófeltöltő (12. melléklet). Majd az egész ciklus kezdődik előlről.



40. ábra: Robotrekesz nélküli szimulációs modell

6.3.2 Robotrekeszes stratégia

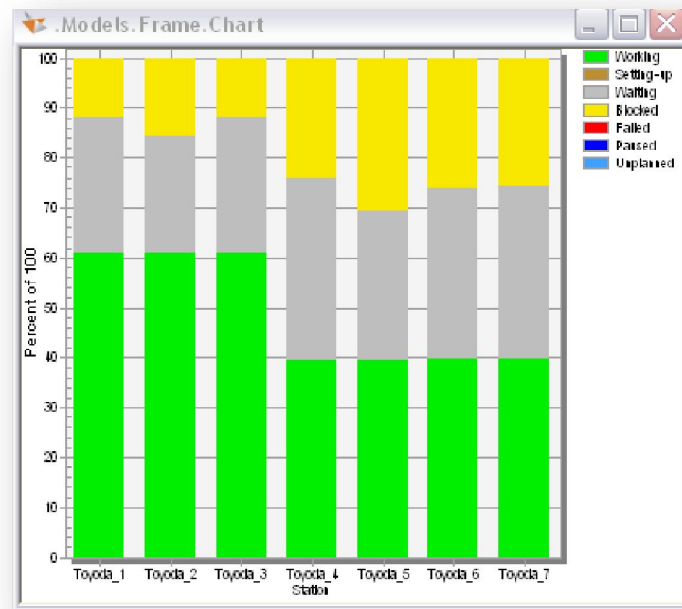
A robotrekeszes stratégia annyiban tér el az előbb említettől, hogy amikor a robot felveszi a darabot a szalagról, akkor nem egyesével szállítja el azokat a gépekbe, hanem egy tároló rekeszbe gyűjt annyi darabszámút, amennyit később majd egymás után szétoszt a megmunkáló központokban. Ezen robotrekesz neve Robotrekesz_1, Robotrekesz_2, illetve Robotrekesz_3. Mindhárom esetben bevezettünk egy változót, amely a rekeszben lévő nyersdarab számot képes számlálni. Miután elérte a megfelelőt megkezdődik a rekeszből való kipakolás és közben a rekesz változója nullázódik, addig a gépeké 1-re vált. A stratégiát szintén egy method segíti, amelynek nevei: Robotrekesz_1_KI (14. melléklet), Robotrekesz_2_KI (15. melléklet) és Robotrekesz_3_KI (16. melléklet). A módosított stratégia következtében változott a szalagra elhelyezett szenzorok methodjai is (20., 21. és 22. melléklet)



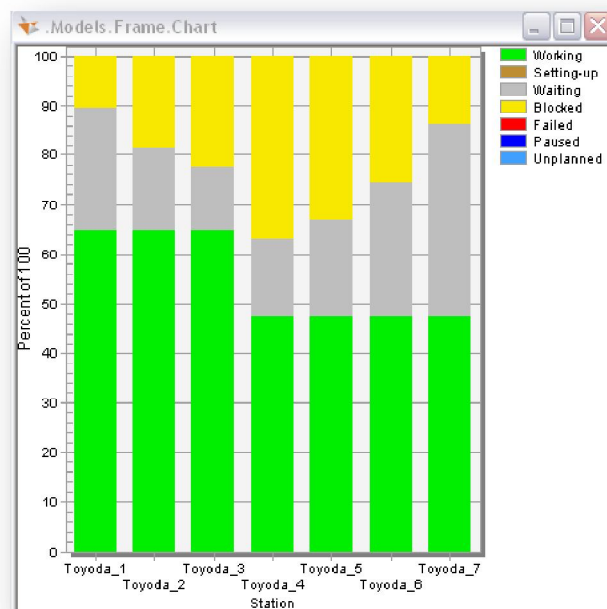
41. ábra: Robotrekeszes szimulációs modell

7. Szimulációs eredmények összehasonlítása

7.1 Megmunkáló gépek összehasonlítása



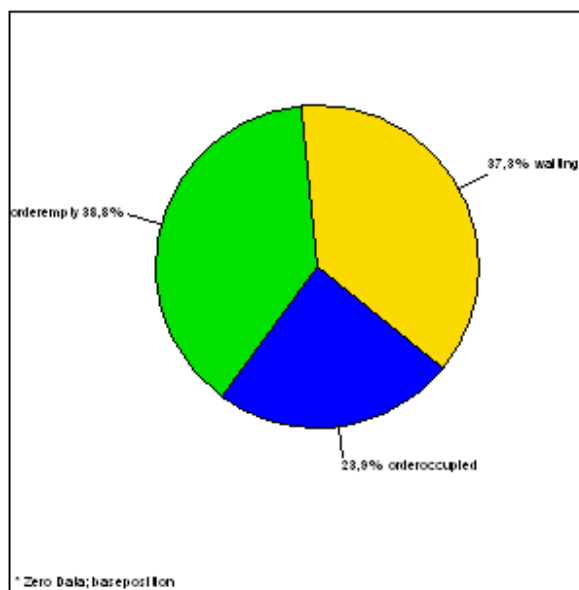
42. ábra: Megmunkáló gépek kihasználtsága rekesz nélkül



43. ábra: Megmunkáló gépek kihasználtsága rekesszel

Az elkészített oszlopdiaagramokból jól kivehető, hogy a második esetben mindegyik gép kevesebb ideig kell, hogy várjon, amíg a robot odapozícionál. Ugyanakkor lényegesen nagyobb a kihasználtságuk is, összevetve az első esettel. Viszont az is jól megfigyelhető, hogy a 3 gép nagyobb kihasználtságú, mint a másik 4.

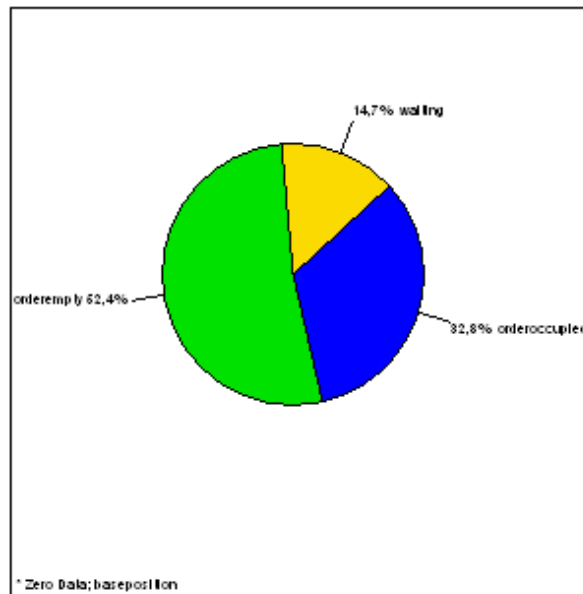
7.2 Robotok működésének összehasonlítása



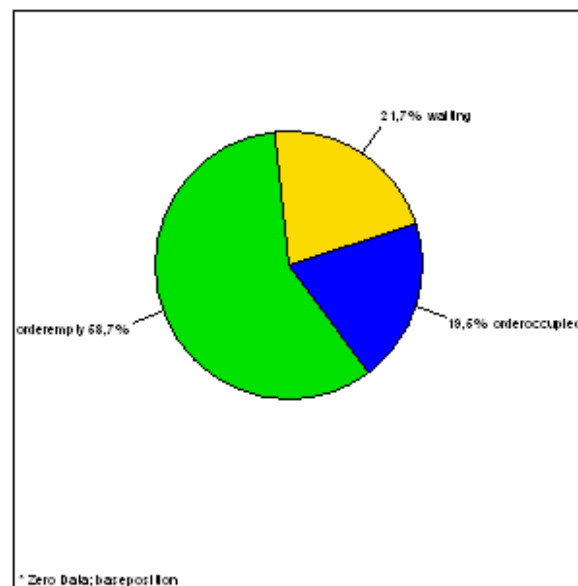
44. ábra: Rekesz nélkül, bal oldali robot



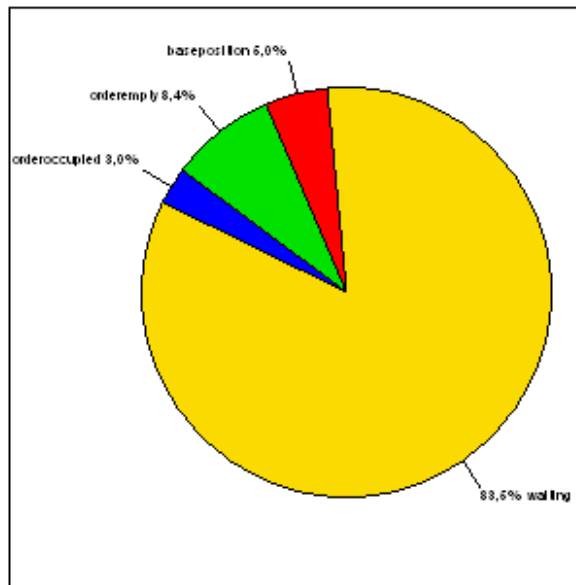
45. ábra: Rekessel, bal oldali robot



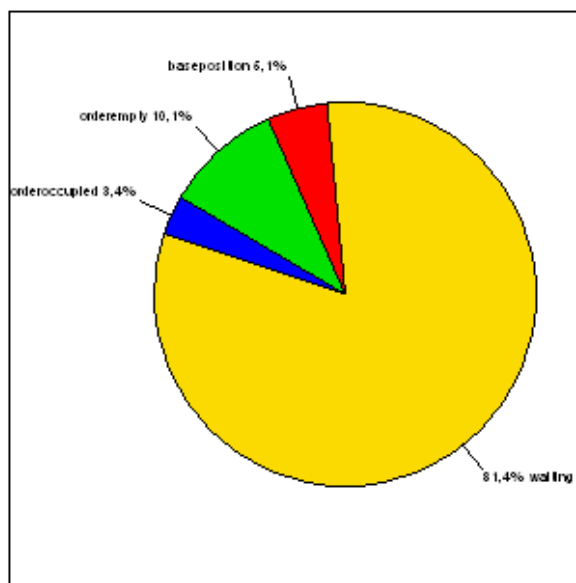
46. ábra: Rekesz nélküli, jobb oldali robot



47. ábra: Rekeszel, jobb oldali robot



48. ábra: Rekesz nélküli, középső robot



49. ábra: Rekessel, középső robot

Amint azt a kördiagramokból jól láthatjuk, a rekeszes megoldás esetében a robotok nagyobb hatékonysággal dolgoznak, mint a rekesz nélküli esetben. Ennek következtében megnő a robot munkaideje, ezáltal a termelés hatékonysága is egyaránt.

7.3 Előállított darabszám összehasonlítása

A rekesz nélküli esetben az egy nap alatt előállított darabszám: 648 db

A rekeszekkel felszerelt esetben az egy nap alatt előállított darabszám: 727db

Következtetés: A szimuláció fényt derített arra, hogy a rekeszes megoldás segítségével egy nap alatt többet tudunk gyártani, így az adott berendezési koncepcióhoz ezen stratégia javasolt a hatékony termelés elérése érdekében!

8. Összegzés

Dolgozatommal igazoltam azon állításom, miszerint egy automatizált gyártórendszer elkészítésével jelentős darabszám növekedést érhetünk el. Annak kiderítésére, hogy melyik elrendezés segítségével érhetjük el a maximális gyártási darabszámot a Plant Simulation program volt segítségemre. Így a gyártórendszer valódi letelepítése nélkül képesek voltunk meghatározni pl. a napi gyártási darabszámot, illetve a legjobb elrendezési formát, továbbá szükségessé váltak átmeneti tárolók is.

Jelen munkámat szeretném a későbbiekben továbbfejleszteni. Ugyanis más elrendezési koncepciók más eredményt adnak. Így minél több elrendezést készítünk el, többféle stratégiákkal, annál közelebb kerülünk a legjobb kialakításhoz. Továbbá a gépek kapacitásának változtatásával szintén növekvő eredményekhez jutunk.

Munkám elkészítésében nagy segítséget nyújtott és köszönetem fejezem ki Dr. Németh Istvánnak és Haraszkó Csabának, akik a program elsajátítását segítették. Illetve Juhász Istvánnak, aki a cég oldaláról nyújtott támogatást.

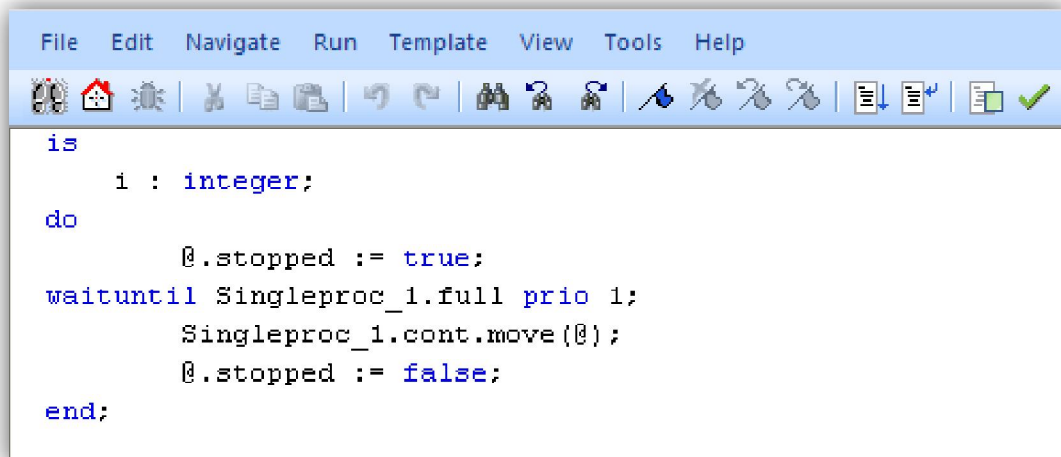
9. Irodalomjegyzék

- [1] Excel-Csepel Szerszámgépgyártó Kft. adatbázisa
- [2] www.fanucrobotics.hu
- [3] www.boschrexroth.com
- [4] www.graphit.hu
- [5] www.wikipedia.hu
- [6] Plant Simulation alaptanfolyam, oktató segédanyag, graphIT Kft. 2010

10. Melléklet

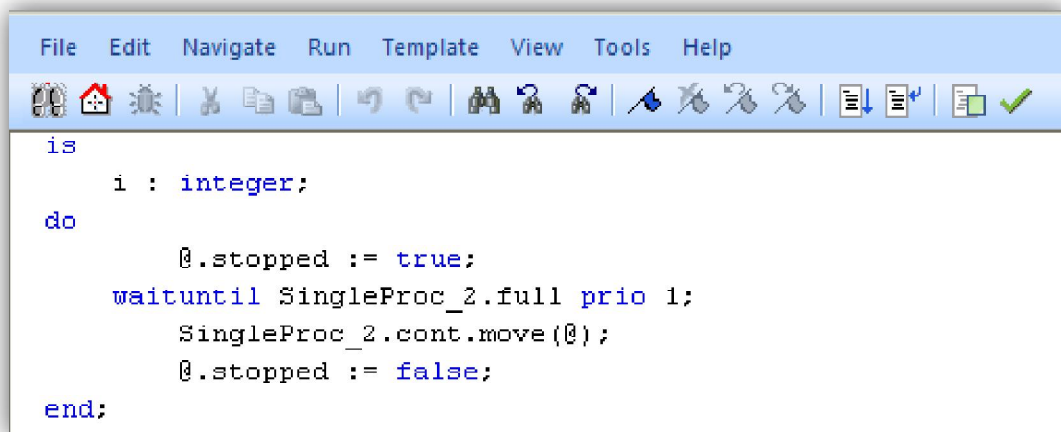
A Methodok programjai az alábbi képeken láthatóak:

1. melléklet (1)



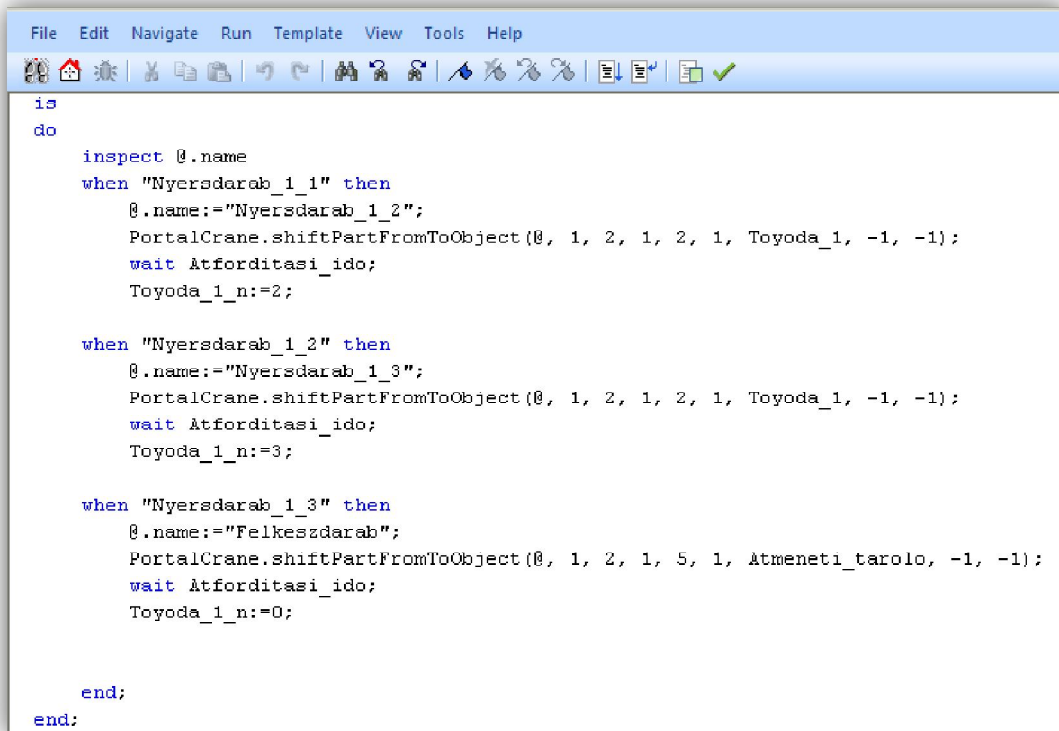
```
is
    i : integer;
do
    @.stopped := true;
waituntil Singleproc_1.full prio 1;
    Singleproc_1.cont.move(@);
    @.stopped := false;
end;
```

2. melléklet (2)



```
is
    i : integer;
do
    @.stopped := true;
waituntil SingleProc_2.full prio 1;
    SingleProc_2.cont.move(@);
    @.stopped := false;
end;
```

3. melléklet (3)



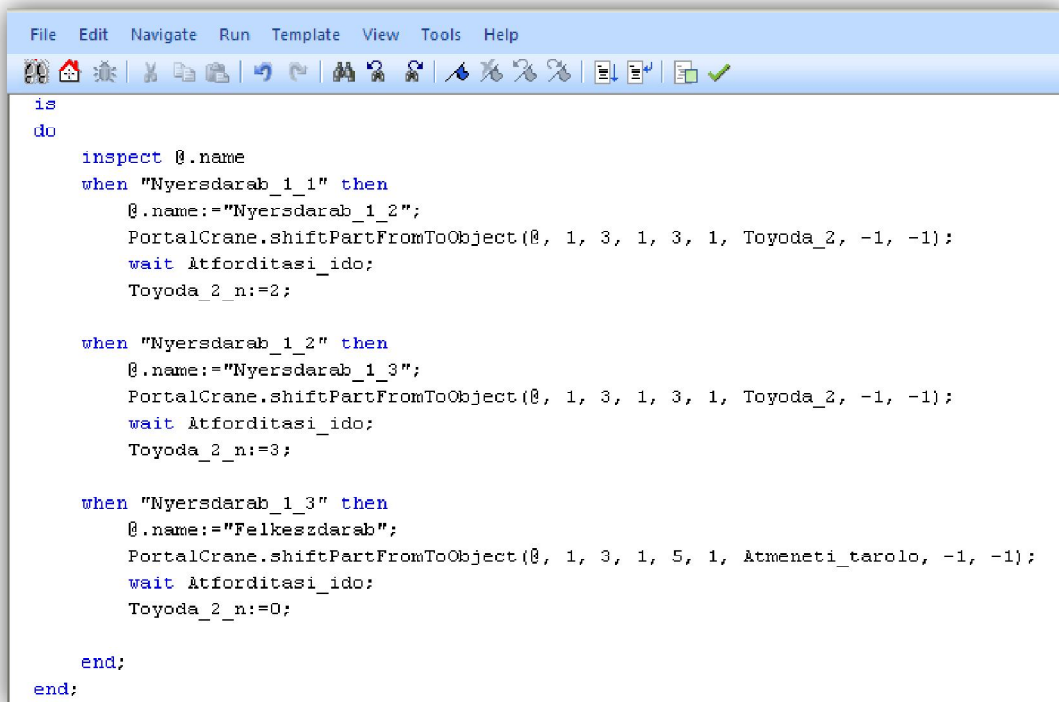
```
is
do
  inspect @.name
  when "Nyersdarab_1_1" then
    @.name:="Nyersdarab_1_2";
    PortalCrane.shiftPartFromToObject(@, 1, 2, 1, 2, 1, Toyoda_1, -1, -1);
    wait Atforditasi_ido;
    Toyoda_1_n:=2;

  when "Nyersdarab_1_2" then
    @.name:="Nyersdarab_1_3";
    PortalCrane.shiftPartFromToObject(@, 1, 2, 1, 2, 1, Toyoda_1, -1, -1);
    wait Atforditasi_ido;
    Toyoda_1_n:=3;

  when "Nyersdarab_1_3" then
    @.name:="Felkeszdarab";
    PortalCrane.shiftPartFromToObject(@, 1, 2, 1, 5, 1, Atmeneti_tarolo, -1, -1);
    wait Atforditasi_ido;
    Toyoda_1_n:=0;

  end;
end;
```

4. melléklet (4)



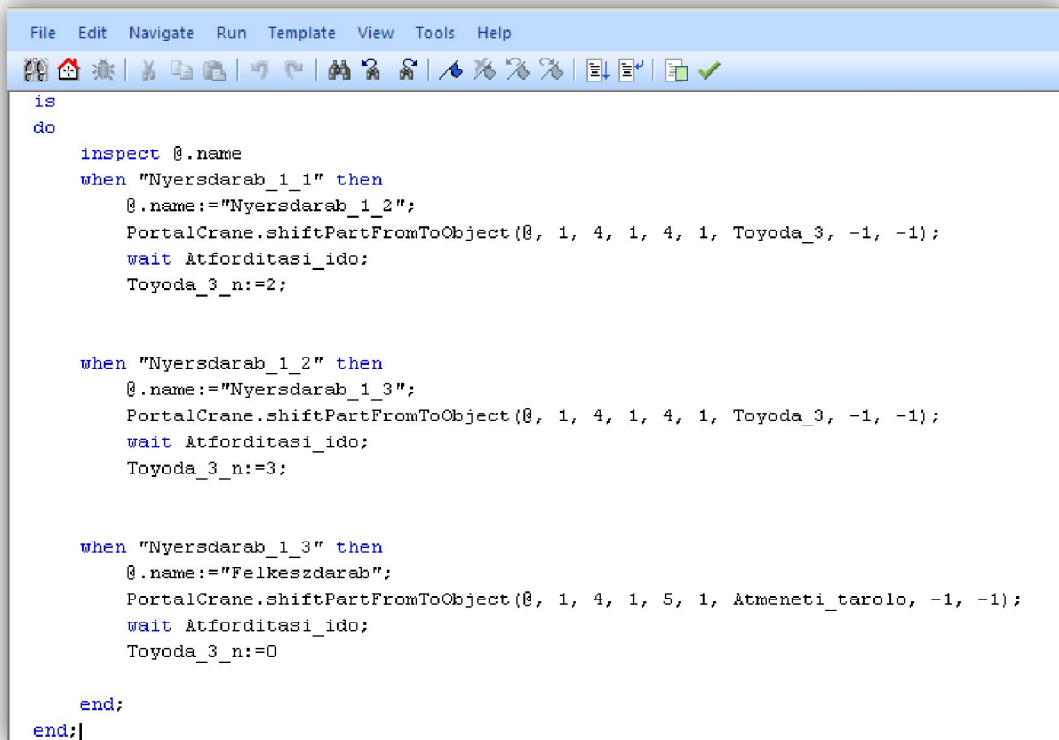
```
is
do
  inspect @.name
  when "Nyersdarab_1_1" then
    @.name:="Nyersdarab_1_2";
    PortalCrane.shiftPartFromToObject(@, 1, 3, 1, 3, 1, Toyoda_2, -1, -1);
    wait Atforditasi_ido;
    Toyoda_2_n:=2;

  when "Nyersdarab_1_2" then
    @.name:="Nyersdarab_1_3";
    PortalCrane.shiftPartFromToObject(@, 1, 3, 1, 3, 1, Toyoda_2, -1, -1);
    wait Atforditasi_ido;
    Toyoda_2_n:=3;

  when "Nyersdarab_1_3" then
    @.name:="Felkeszdarab";
    PortalCrane.shiftPartFromToObject(@, 1, 3, 1, 5, 1, Atmeneti_tarolo, -1, -1);
    wait Atforditasi_ido;
    Toyoda_2_n:=0;

  end;
end;
```

5. melléklet (5)



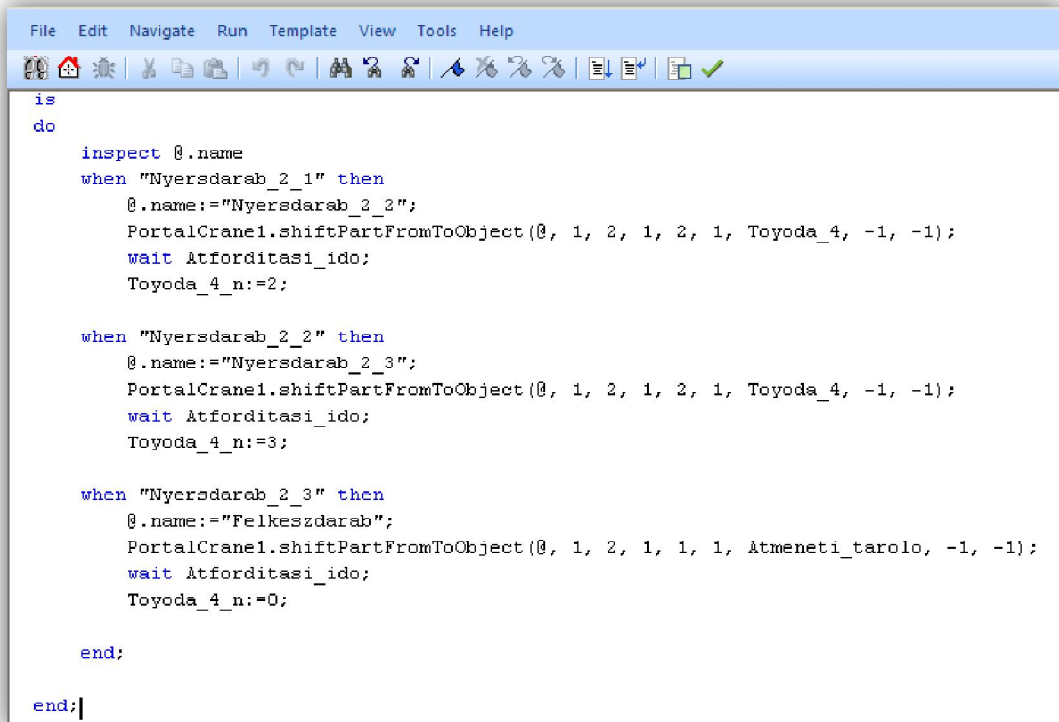
```
is
do
  inspect @.name
  when "Nyersdarab_1_1" then
    @.name:="Nyersdarab_1_2";
    PortalCrane.shiftPartFromToObject(@, 1, 4, 1, 4, 1, Toyoda_3, -1, -1);
    wait Atforditasi_ido;
    Toyoda_3_n:=2;

  when "Nyersdarab_1_2" then
    @.name:="Nyersdarab_1_3";
    PortalCrane.shiftPartFromToObject(@, 1, 4, 1, 4, 1, Toyoda_3, -1, -1);
    wait Atforditasi_ido;
    Toyoda_3_n:=3;

  when "Nyersdarab_1_3" then
    @.name:="Felkeszdarab";
    PortalCrane.shiftPartFromToObject(@, 1, 4, 1, 5, 1, Atmeneti_tarolo, -1, -1);
    wait Atforditasi_ido;
    Toyoda_3_n:=0

  end;
end;
```

6. melléklet (6)



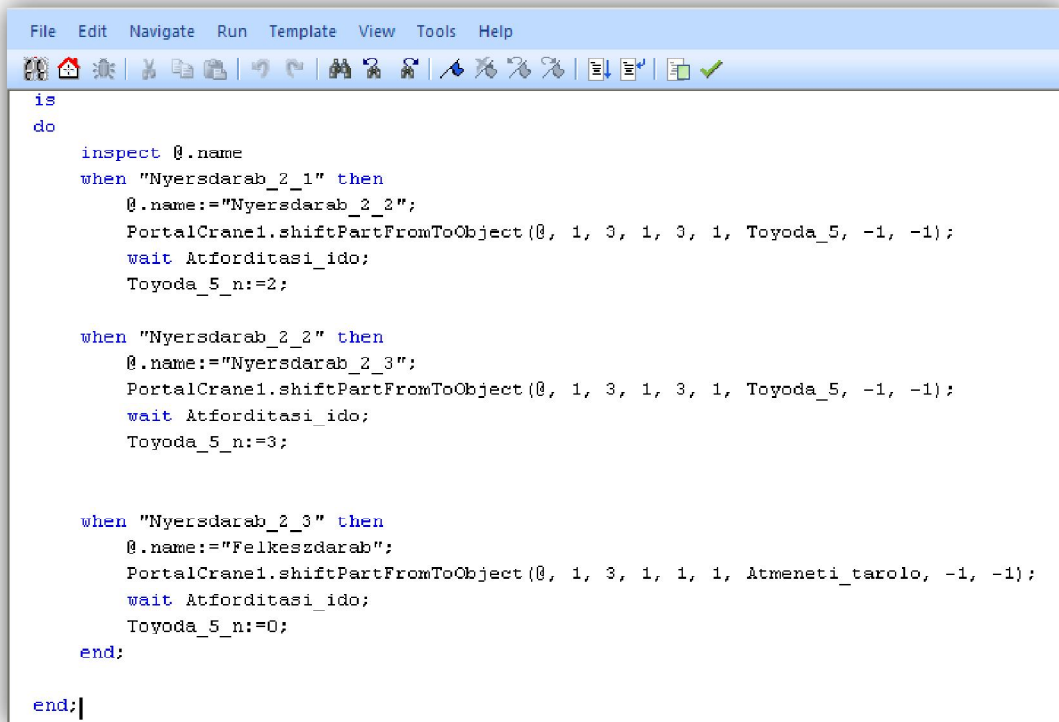
```
is
do
  inspect @.name
  when "Nyersdarab_2_1" then
    @.name:="Nyersdarab_2_2";
    PortalCrane1.shiftPartFromToObject(@, 1, 2, 1, 2, 1, Toyoda_4, -1, -1);
    wait Atforditasi_ido;
    Toyoda_4_n:=2;

  when "Nyersdarab_2_2" then
    @.name:="Nyersdarab_2_3";
    PortalCrane1.shiftPartFromToObject(@, 1, 2, 1, 2, 1, Toyoda_4, -1, -1);
    wait Atforditasi_ido;
    Toyoda_4_n:=3;

  when "Nyersdarab_2_3" then
    @.name:="Felkeszdarab";
    PortalCrane1.shiftPartFromToObject(@, 1, 2, 1, 1, 1, Atmeneti_tarolo, -1, -1);
    wait Atforditasi_ido;
    Toyoda_4_n:=0

  end;
end;
```

7. melléklet (7)

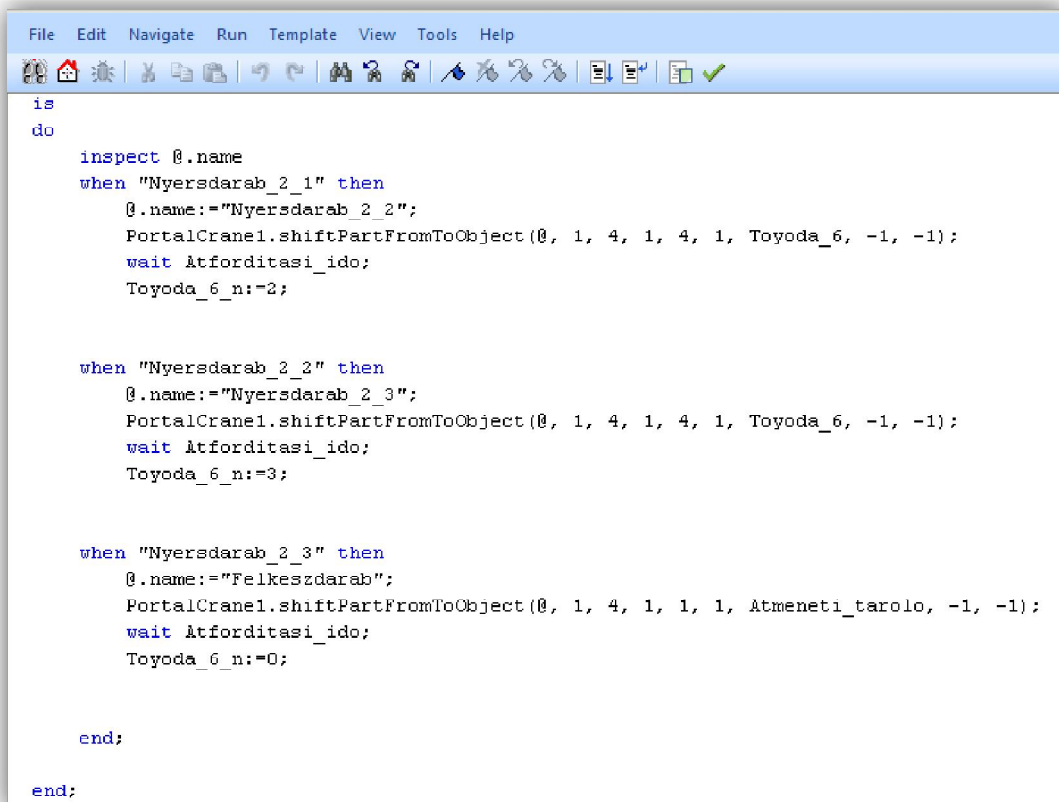


```
File Edit Navigate Run Template View Tools Help
is
do
  inspect @.name
  when "Nyersdarab_2_1" then
    @.name:="Nyersdarab_2_2";
    PortalCranel.shiftPartFromToObject(@, 1, 3, 1, 3, 1, Toyoda_5, -1, -1);
    wait Atforditasi_ido;
    Toyoda_5_n:=2;

  when "Nyersdarab_2_2" then
    @.name:="Nyersdarab_2_3";
    PortalCranel.shiftPartFromToObject(@, 1, 3, 1, 3, 1, Toyoda_5, -1, -1);
    wait Atforditasi_ido;
    Toyoda_5_n:=3;

  when "Nyersdarab_2_3" then
    @.name:="Felkeszdarab";
    PortalCranel.shiftPartFromToObject(@, 1, 3, 1, 1, 1, Atmeneti_tarolo, -1, -1);
    wait Atforditasi_ido;
    Toyoda_5_n:=0;
  end;
end;
```

8. melléklet (8)

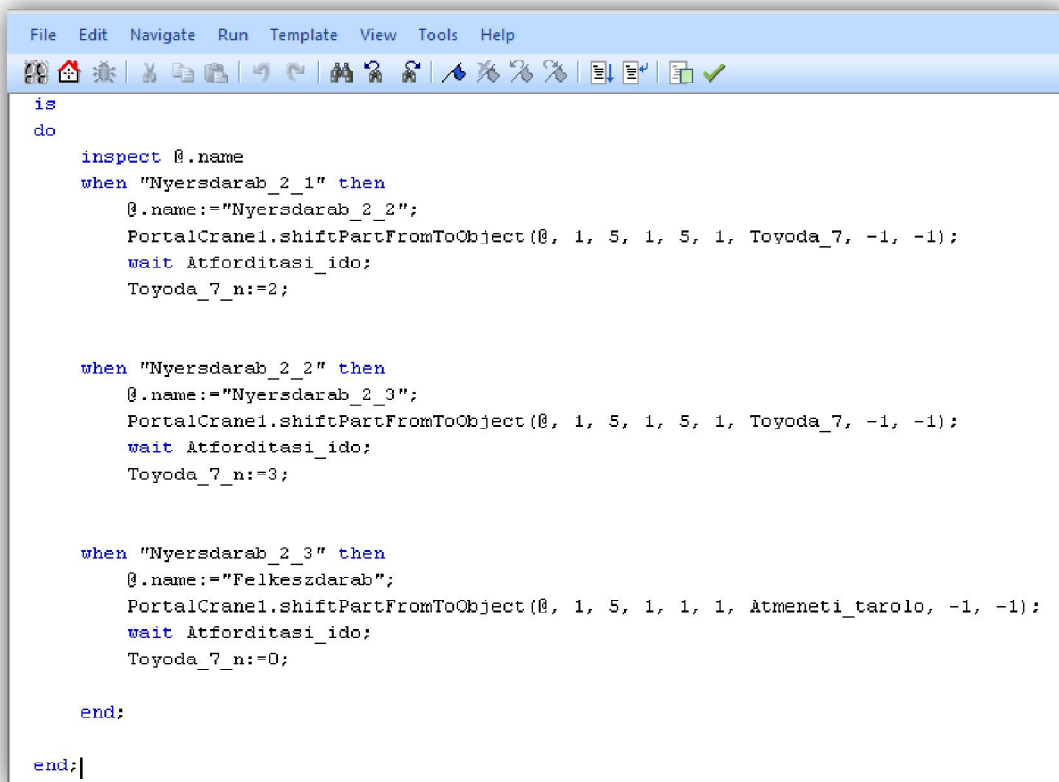


```
File Edit Navigate Run Template View Tools Help
is
do
  inspect @.name
  when "Nyersdarab_2_1" then
    @.name:="Nyersdarab_2_2";
    PortalCranel.shiftPartFromToObject(@, 1, 4, 1, 4, 1, Toyoda_6, -1, -1);
    wait Atforditasi_ido;
    Toyoda_6_n:=2;

  when "Nyersdarab_2_2" then
    @.name:="Nyersdarab_2_3";
    PortalCranel.shiftPartFromToObject(@, 1, 4, 1, 4, 1, Toyoda_6, -1, -1);
    wait Atforditasi_ido;
    Toyoda_6_n:=3;

  when "Nyersdarab_2_3" then
    @.name:="Felkeszdarab";
    PortalCranel.shiftPartFromToObject(@, 1, 4, 1, 1, 1, Atmeneti_tarolo, -1, -1);
    wait Atforditasi_ido;
    Toyoda_6_n:=0;
  end;
end;
```

9. melléklet (9)



```
is
do
    inspect @.name
    when "Nyersdarab_2_1" then
        @.name:="Nyersdarab_2_2";
        PortalCrane1.shiftPartFromToObject(@, 1, 5, 1, 5, 1, Toyoda_7, -1, -1);
        wait Atforditasi_ido;
        Toyoda_7_n:=2;

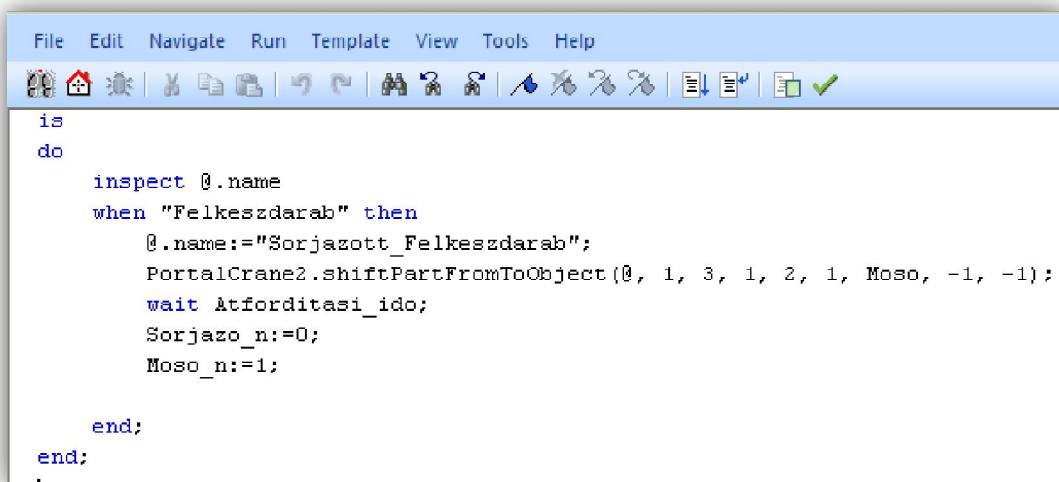
    when "Nyersdarab_2_2" then
        @.name:="Nyersdarab_2_3";
        PortalCrane1.shiftPartFromToObject(@, 1, 5, 1, 5, 1, Toyoda_7, -1, -1);
        wait Atforditasi_ido;
        Toyoda_7_n:=3;

    when "Nyersdarab_2_3" then
        @.name:="Felkeszdarab";
        PortalCrane1.shiftPartFromToObject(@, 1, 5, 1, 1, 1, Atmeneti_tarolo, -1, -1);
        wait Atforditasi_ido;
        Toyoda_7_n:=0;

    end;

end;|
```

10. melléklet (10)

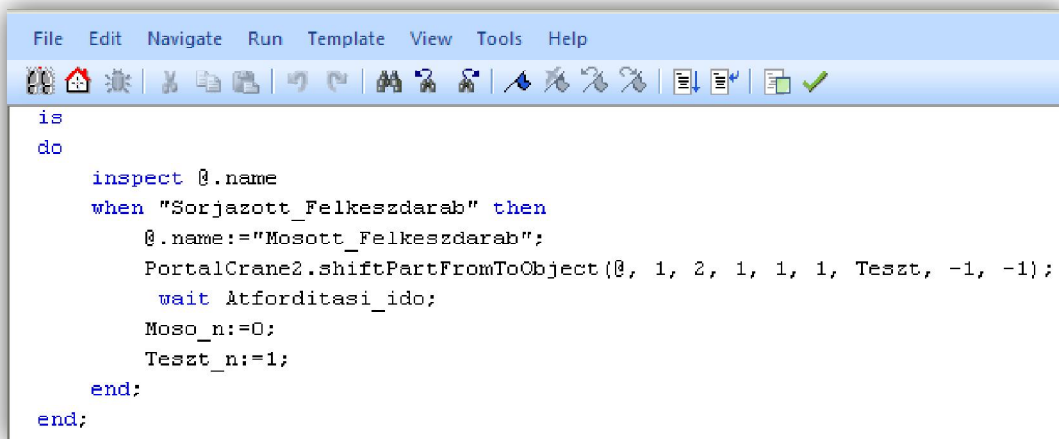


```
is
do
    inspect @.name
    when "Felkeszdarab" then
        @.name:="Sorjazott_Felkeszdarab";
        PortalCrane2.shiftPartFromToObject(@, 1, 3, 1, 2, 1, Moso, -1, -1);
        wait Atforditasi_ido;
        Sorjazo_n:=0;
        Moso_n:=1;

    end;

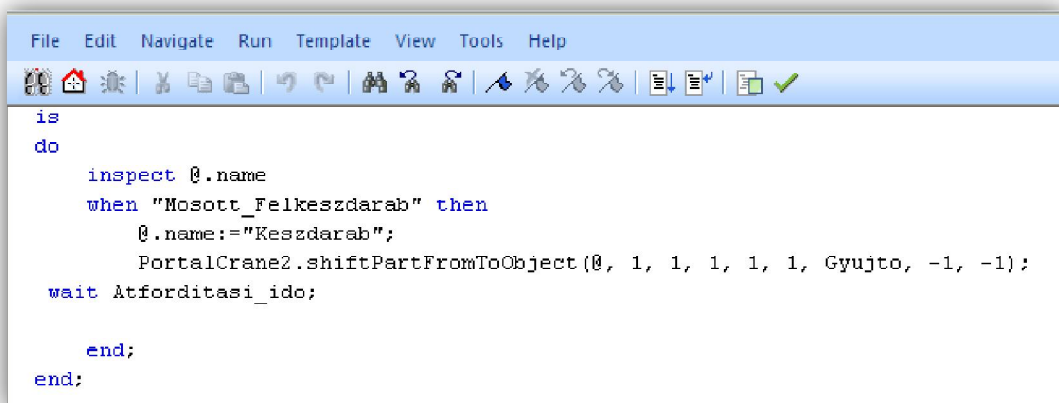
end;
```

11. melléklet (11)



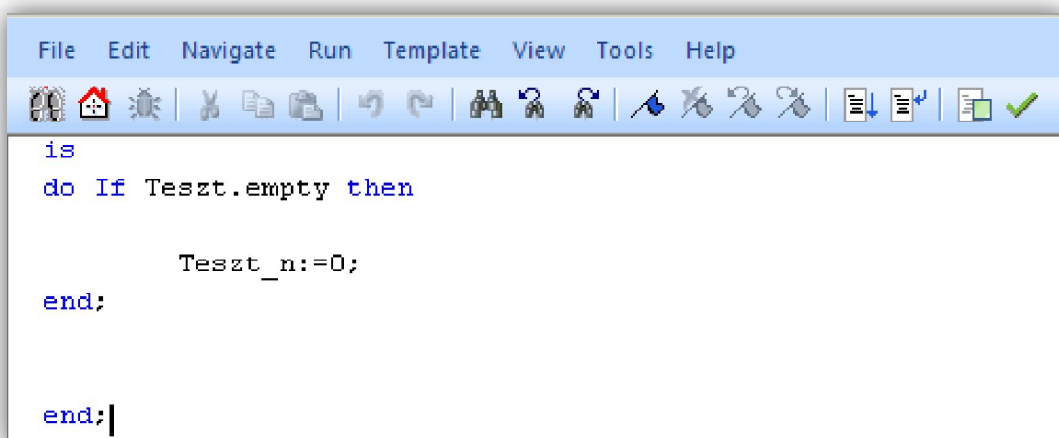
```
File Edit Navigate Run Template View Tools Help
is
do
  inspect @.name
  when "Sorjazott_Felkeszdarab" then
    @.name:="Mosott_Felkeszdarab";
    PortalCrane2.ShiftPartFromToObject(@, 1, 2, 1, 1, 1, Teszt, -1, -1);
    wait Atforditasi_ido;
    Moso_n:=0;
    Teszt_n:=1;
  end;
end;
```

12. melléklet (12)



```
File Edit Navigate Run Template View Tools Help
is
do
  inspect @.name
  when "Mosott_Felkeszdarab" then
    @.name:="Keszdarab";
    PortalCrane2.ShiftPartFromToObject(@, 1, 1, 1, 1, 1, Gyujto, -1, -1);
    wait Atforditasi_ido;
  end;
end;
```

13. melléklet (13)

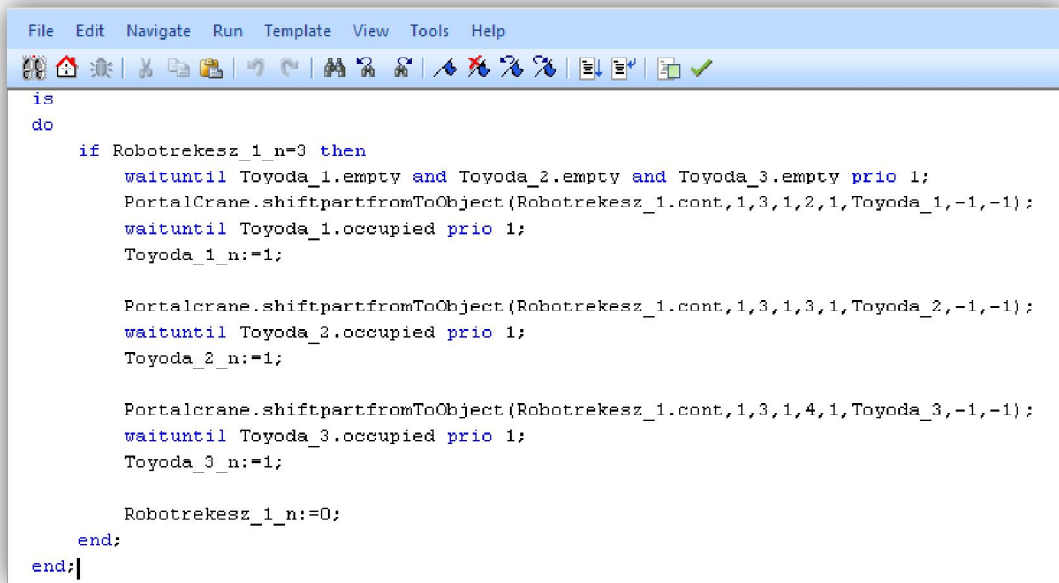


```
File Edit Navigate Run Template View Tools Help
is
do If Teszt.empty then

    Teszt_n:=0;
end;

end;
```

14. melléklet (14)



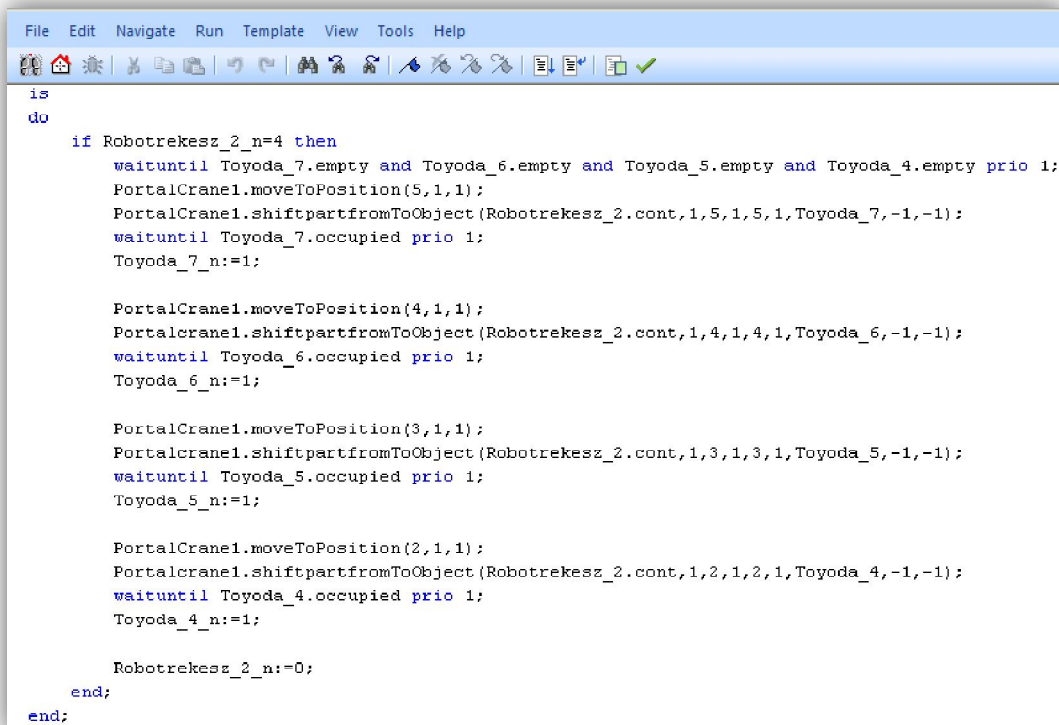
```
is
do
  if Robotrekesz_1.n=3 then
    waituntil Toyoda_1.empty and Toyoda_2.empty and Toyoda_3.empty prio 1;
    PortalCrane.shiftpartfromToObject(Robotrekesz_1.cont,1,3,1,2,1,Toyoda_1,-1,-1);
    waituntil Toyoda_1.occupied prio 1;
    Toyoda_1.n:=1;

    PortalCrane.shiftpartfromToObject(Robotrekesz_1.cont,1,3,1,3,1,Toyoda_2,-1,-1);
    waituntil Toyoda_2.occupied prio 1;
    Toyoda_2.n:=1;

    PortalCrane.shiftpartfromToObject(Robotrekesz_1.cont,1,3,1,4,1,Toyoda_3,-1,-1);
    waituntil Toyoda_3.occupied prio 1;
    Toyoda_3.n:=1;

    Robotrekesz_1.n:=0;
  end;
end;
```

15. melléklet (15)



```
is
do
  if Robotrekesz_2.n=4 then
    waituntil Toyoda_7.empty and Toyoda_6.empty and Toyoda_5.empty and Toyoda_4.empty prio 1;
    PortalCrane1.moveToPosition(5,1,1);
    PortalCrane1.shiftpartfromToObject(Robotrekesz_2.cont,1,5,1,5,1,Toyoda_7,-1,-1);
    waituntil Toyoda_7.occupied prio 1;
    Toyoda_7.n:=1;

    PortalCrane1.moveToPosition(4,1,1);
    PortalCrane1.shiftpartfromToObject(Robotrekesz_2.cont,1,4,1,4,1,Toyoda_6,-1,-1);
    waituntil Toyoda_6.occupied prio 1;
    Toyoda_6.n:=1;

    PortalCrane1.moveToPosition(3,1,1);
    PortalCrane1.shiftpartfromToObject(Robotrekesz_2.cont,1,3,1,3,1,Toyoda_5,-1,-1);
    waituntil Toyoda_5.occupied prio 1;
    Toyoda_5.n:=1;

    PortalCrane1.moveToPosition(2,1,1);
    PortalCrane1.shiftpartfromToObject(Robotrekesz_2.cont,1,2,1,2,1,Toyoda_4,-1,-1);
    waituntil Toyoda_4.occupied prio 1;
    Toyoda_4.n:=1;

    Robotrekesz_2.n:=0;
  end;
end;
```

16. melléklet (16)

```
File Edit Navigate Run Template View Tools Help
is
do
  if Robotrekesz_3_n=3 then
    waituntil Sorjazo.empty prio 1;
    PortalCrane2.shiftpartfromToObject(Robotrekesz_3.cont,1,4,1,3,1,Sorjazo,-1,-1);
    waituntil Sorjazo.occupied prio 1;
    Sorjazo_n:=1;

    Robotrekesz_3_n:=2;
  end;
end;
```

17. melléklet (17)

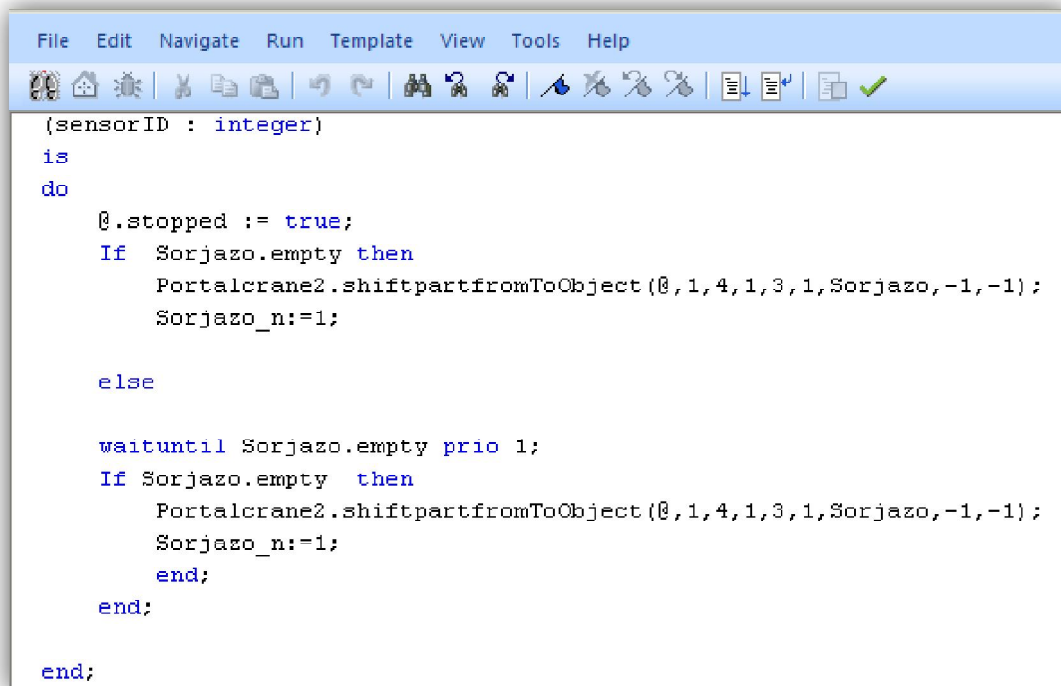
```
File Edit Navigate Run Template View Tools Help
(sensorID : integer)
is
do
  @.stopped := true;
  If Toyoda_1_n=0 then
    portalcrane.shiftpartfromToObject(@.cont,1,1,1,2,1,Toyoda_1,-1,-1);
    Toyoda_1_n:=1;
  elseif Toyoda_2_n=0 then
    portalcrane.shiftpartfromToObject(@.cont,1,1,1,3,1,Toyoda_2,-1,-1);
    Toyoda_2_n:=1;
  elseif Toyoda_3_n=0 then
    portalcrane.shiftpartfromToObject(@.cont,1,1,1,4,1,Toyoda_3,-1,-1);
    Toyoda_3_n:=1;
  else
    waituntil Toyoda_3_n=0 prio 1;
    If Toyoda_1_n=0 then
      portalcrane.shiftpartfromToObject(@.cont,1,1,1,2,1,Toyoda_1,-1,-1);
      Toyoda_1_n:=1;
    elseif Toyoda_2_n=0 then
      portalcrane.shiftpartfromToObject(@.cont,1,1,1,3,1,Toyoda_2,-1,-1);
      Toyoda_2_n:=1;
    elseif Toyoda_3_n=0 then
      portalcrane.shiftpartfromToObject(@.cont,1,1,1,4,1,Toyoda_3,-1,-1);
      Toyoda_3_n:=1;
    end;
  end;
  waituntil @.Empty prio 1;
  @.stopped := false;
end;
```

18. melléklet (18)

```
(sensorID : integer)
is
do
  @.stopped := true;
  If Toyoda_7_n=0 then
    portalcra1.shiftpartfromToObject (@.cont,1,6,1,5,1,Toyoda_7,-1,-1);
    Toyoda_7_n:=1;
  elseif Toyoda_6_n=0 then
    portalcra1.shiftpartfromToObject (@.cont,1,6,1,4,1,Toyoda_6,-1,-1);
    Toyoda_6_n:=1;
  elseif Toyoda_5_n=0 then
    portalcra1.shiftpartfromToObject (@.cont,1,6,1,3,1,Toyoda_5,-1,-1);
    Toyoda_5_n:=1;
  elseif Toyoda_4_n=0 then
    portalcra1.shiftpartfromToObject (@.cont,1,6,1,2,1,Toyoda_4,-1,-1);
    Toyoda_4_n:=1;
  else
    waituntil Toyoda_4_n=0 prio 1;
    If Toyoda_7_n=0 then
      portalcra1.shiftpartfromToObject (@.cont,1,6,1,5,1,Toyoda_7,-1,-1);
      Toyoda_7_n:=1;
    elseif Toyoda_6_n=0 then
      portalcra1.shiftpartfromToObject (@.cont,1,6,1,4,1,Toyoda_6,-1,-1);
      Toyoda_6_n:=1;
    elseif Toyoda_5_n=0 then
      portalcra1.shiftpartfromToObject (@.cont,1,6,1,3,1,Toyoda_5,-1,-1);
      Toyoda_5_n:=1;
    elseif Toyoda_4_n=0 then
      portalcra1.shiftpartfromToObject (@.cont,1,6,1,2,1,Toyoda_4,-1,-1);
      Toyoda_4_n:=1;
    end;
  end;
end;
waituntil @.Empty prio 1;
```

```
  @.stopped := false;
end;
|
```

19. melléklet (19)



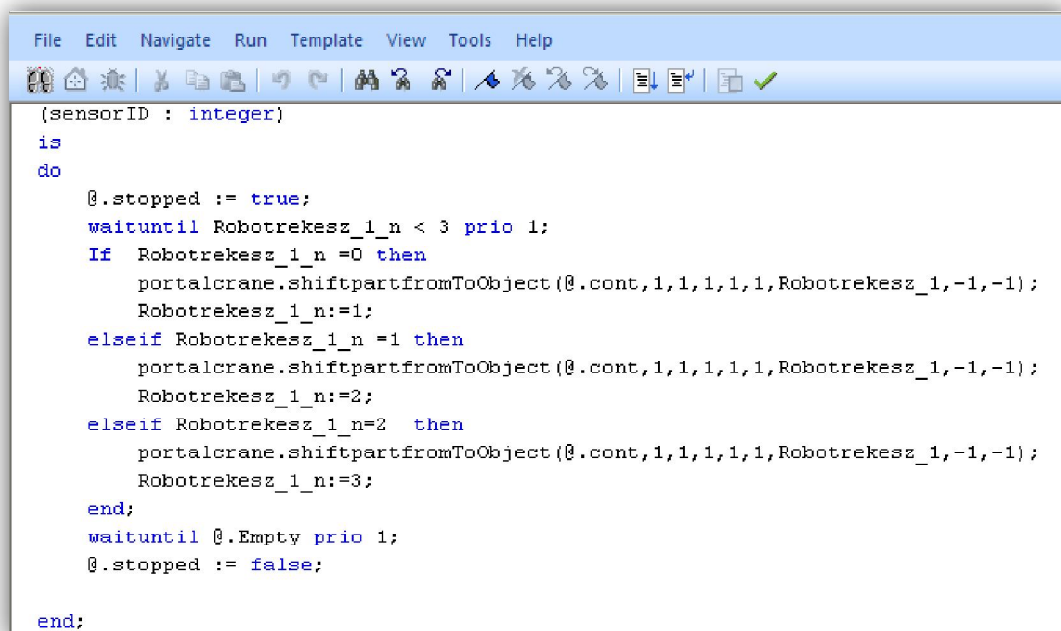
```
(sensorID : integer)
is
do
    @.stopped := true;
    If Sorjazo.empty then
        Portalcrane2.shiftpartfromToObject (@,1,4,1,3,1,Sorjazo,-1,-1);
        Sorjazo_n:=1;

    else

        waituntil Sorjazo.empty prio 1;
        If Sorjazo.empty then
            Portalcrane2.shiftpartfromToObject (@,1,4,1,3,1,Sorjazo,-1,-1);
            Sorjazo_n:=1;
            end;
        end;

end;
```

20. melléklet (20)



```
(sensorID : integer)
is
do
    @.stopped := true;
    waituntil Robotrekesz_1_n < 3 prio 1;
    If Robotrekesz_1_n =0 then
        portalcrane.shiftpartfromToObject (@.cont,1,1,1,1,1,Robotrekesz_1,-1,-1);
        Robotrekesz_1_n:=1;
    elseif Robotrekesz_1_n =1 then
        portalcrane.shiftpartfromToObject (@.cont,1,1,1,1,1,Robotrekesz_1,-1,-1);
        Robotrekesz_1_n:=2;
    elseif Robotrekesz_1_n =2 then
        portalcrane.shiftpartfromToObject (@.cont,1,1,1,1,1,Robotrekesz_1,-1,-1);
        Robotrekesz_1_n:=3;
    end;
    waituntil @.Empty prio 1;
    @.stopped := false;

end;
```

21. melléklet (21)

```
File Edit Navigate Run Template View Tools Help
[Icons]

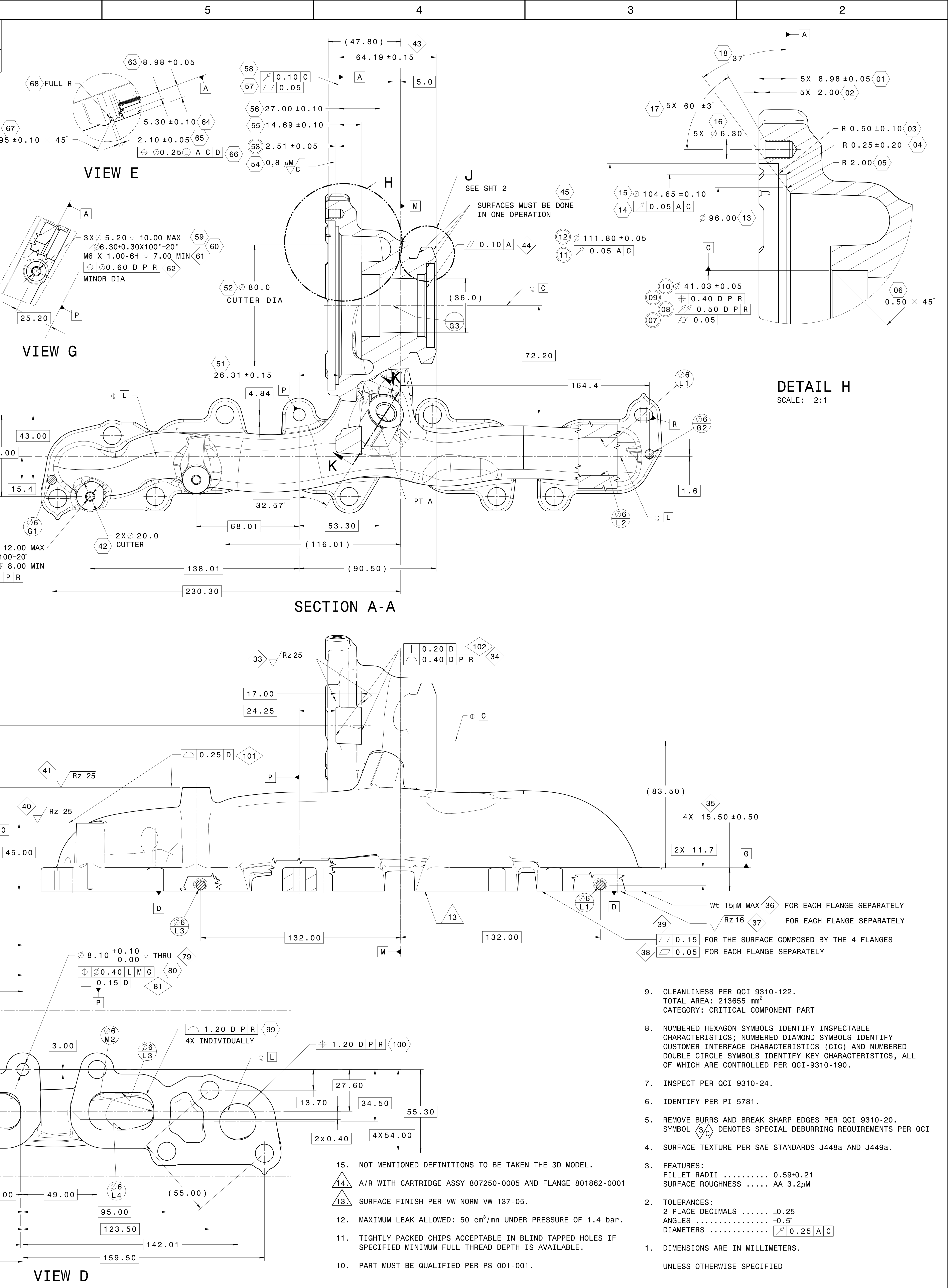
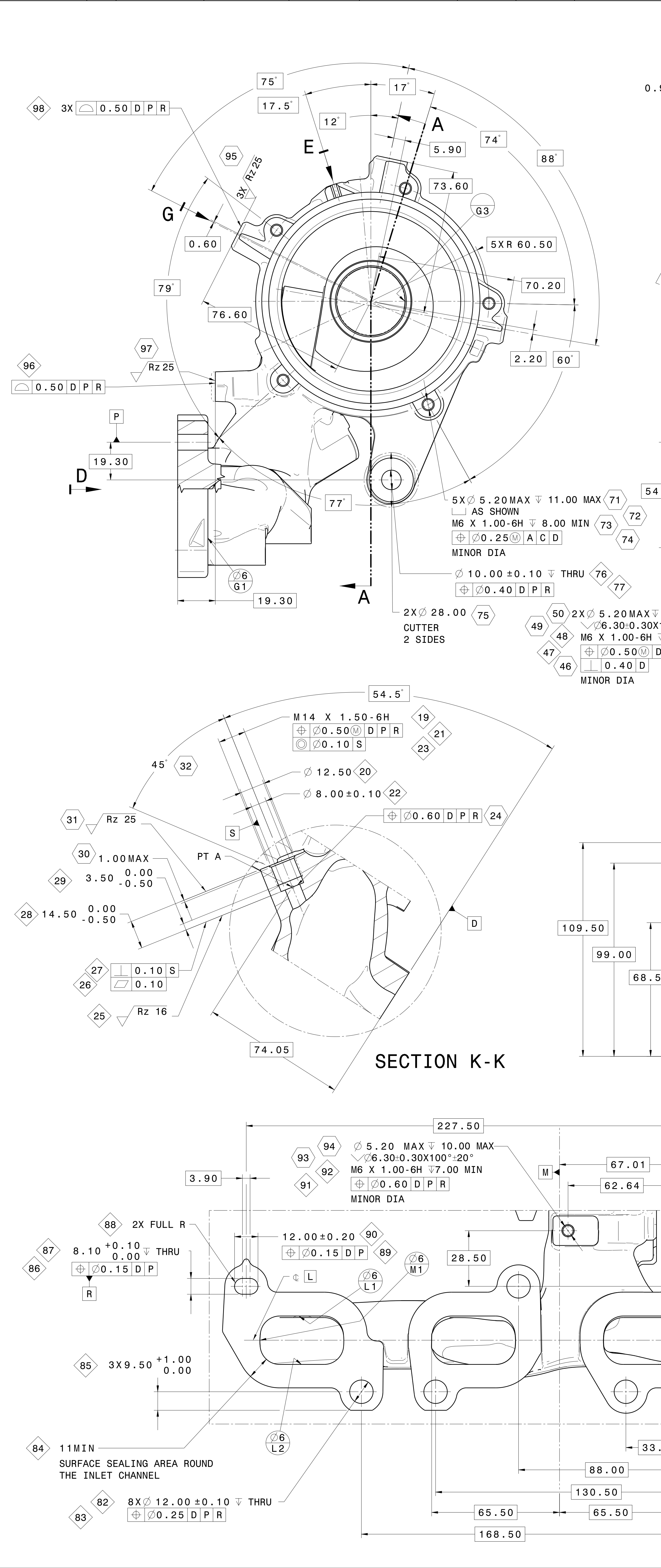
(sensorID : integer)
is
do
    @.stopped := true;
    waituntil Robotrekesz_2_n < 4 prio 1;
    If Robotrekesz_2_n =0 then
        portalcrane1.shiftpartfromToObject (@.cont,1,6,1,6,1,Robotrekesz_2,-1,-1);
        Robotrekesz_2_n:=1;
    elseif Robotrekesz_2_n =1 then
        portalcrane1.shiftpartfromToObject (@.cont,1,6,1,6,1,Robotrekesz_2,-1,-1);
        Robotrekesz_2_n:=2;
    elseif Robotrekesz_2_n=2 then
        portalcrane1.shiftpartfromToObject (@.cont,1,6,1,6,1,Robotrekesz_2,-1,-1);
        Robotrekesz_2_n:=3;
    elseif Robotrekesz_2_n=3 then
        portalcrane1.shiftpartfromToObject (@.cont,1,6,1,6,1,Robotrekesz_2,-1,-1);
        Robotrekesz_2_n:=4;
    end;
    waituntil @.Empty prio 1;
    @.stopped := false;
end;
```

22. melléklet (22)

```
File Edit Navigate Run Template View Tools Help
[Icons]

(sensorID :integer)
is
do
    @.stopped := true;
    waituntil Robotrekesz_3_n < 3 prio 1;
    If Robotrekesz_3_n =0 then
        portalcrane2.shiftpartfromToObject (@,1,4,1,4,1,Robotrekesz_3,-1,-1);
        Robotrekesz_3_n:=1;
    elseif Robotrekesz_3_n =1 then
        portalcrane2.shiftpartfromToObject (@,1,4,1,4,1,Robotrekesz_3,-1,-1);
        Robotrekesz_3_n:=2;
    elseif Robotrekesz_3_n=2 then
        portalcrane2.shiftpartfromToObject (@,1,4,1,4,1,Robotrekesz_3,-1,-1);
        Robotrekesz_3_n:=3;
    end;
end;
```

PART NUMBER	REV	MAKE FROM	A/R (INCH)	CALCULATED MASS (KG)	REMARKS	3D MODEL YES/NO	3D MODEL LINKED	RELEASE STATUS
812297-0001	E	812296-0001	0.53	5.473	VW 1.6L MDB PVS EU5/EU6	YES	YES	PRODUCTION



CAD DRAWING

CHANGES TO THIS DRAWING MUST BE INCORPORATED VIA THE CAD SYSTEM

REVISIONS

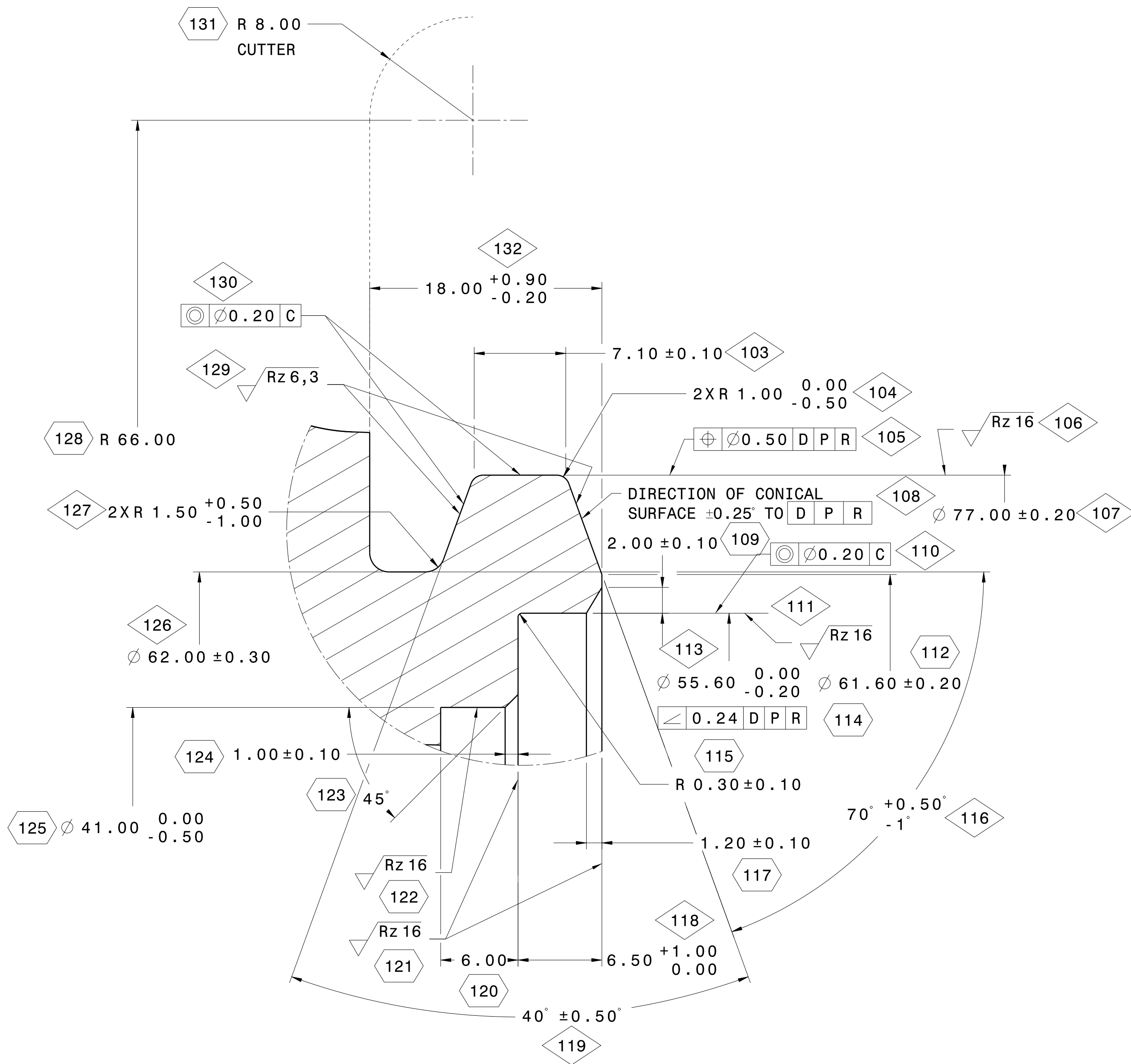
ZONE	REV	DATE	DESCRIPTION	MM/DD/YY	DES	CHK	APP
-	-	01102000	PRE-PRODUCTION RELEASED	04/29/11	-	-	-
F48	A	E1101002	0001 PRODUCTION RELEASED	06/17/11	ESH	KLU	JSK
S268	B	E1101290	HEX 115 WAS RD-400.10	08/07/11	LWA	KLU	JSK
S184	C1	E1101539	HEX 80 WAS C1C 80 0.3UNG				
S184	C2		HEX 100 WAS C1C 80 0.3UNG				
	C3		DATUMS L WERE REDEFINED	09/22/11	JDY	LZA	JSK
S144	D1	E1101968	ADDED NOTE 15				
S184	D2		SYM 99 WAS C1C 80 0.3UNG				
S184	D3		SYM 100 WAS C1C 80 0.3UNG				
-	D4		REDEFINED DATUM M1,M2	12/12/11	GSZ	KLU	JRO
S180	E1	E1200042	SYM 99 WAS C1C 80 0.3UNG				
S180	E2		SYM 99 WAS C1C 80 0.3UNG	01/21/12	DYU	KLU	JRO

PART DEFINITION CONTROLLED BY SOLID MODEL

SIMILAR TO 809831

ITEM	REQ	PART NUMBER	DESCRIPTION
---	---	---	GTD12V
NEXT LEVEL		TOP ASSY	MODEL
AERO ENGR	---	---	MFG ENGR MDR 04/29/11
APPL ENGR	JSK	04/29/11	PROD ENGR OES 04/29/11
MATL ENGR	---	---	QUAL ENGR CVA 04/29/11
MECH ENGR	DUR	06/13/11	SDA ENGR PSK 04/29/11
PROCESS CORROSION PROTECTION PER IFP-1			
MATERIAL SEE TAB			
2D DESN	OLD	04/29/11	LAST INSPECTION HEX USED 132
3D DESN	OLD	04/29/11	
CHECKER	PMU	04/29/11	3RD ANGLE PROJECTION
GEOMETRIC TOLERANCING PER ASME Y14.5M-1994			
THIS DOCUMENT CONTAINS PROPRIETARY INFORMATION AND SUCH INFORMATION MAY NOT BE DISCLOSED TO OTHERS FOR ANY PURPOSE OR USED FOR MANUFACTURING WITHOUT WRITTEN PERMISSION FROM THE HONEYWELL CORPORATION.			
Honeywell			
TITLE HOUSING, TURBINE			
SIZE E	DRAWING NUMBER 812297		REV. E
SCALE 1:1 & NOTED		DO NOT SCALE	SHEET 1 OF 2

SEE SHEET 1 FOR TABULATION



DETAIL J

SEE SHEET 1 FOR NOTES

		LAST INSPECTION HEX USED 132	3RD ANGLE PROJECTION 3RD ANGLE PROJECTION
GEOMETRIC TOLERANCING PER ASME Y14.5M-1994			
THIS DOCUMENT CONTAINS PROPRIETARY INFORMATION AND SUCH INFORMATION MAY NOT BE DISCLOSED TO OTHERS FOR ANY PURPOSE OR USED FOR MANUFACTURING WITHOUT WRITTEN PERMISSION FROM THE HONEYWELL CORPORATION.			
Honeywell			
TITLE HOUSING, TURBINE			
SIZE E	DRAWING NUMBER 812297		REV. E
SCALE 4:1	DO NOT SCALE	SHEET 2 OF 2	
CATIA V5	1	REV: 12/15/08	