

## 22. GYAKORLAT – GYÁRTÓRENDSZEREK IRÁNYÍTÁSA ÉS PROGRAMOZÁSA

### 22.0 A GYAKORLAT CÉLJA

a rugalmas gyártórendszerek irányítási és programozási kérdéseinek bemutatása, a gyártórendszerek működésének tanulmányozása, alapvető rendszerirányítási feladatok megértése, rendszerprogramozási alapismeretek megszerzése.

Célja továbbá a gyártórendszer-irányítás szerteágazó, multidiszciplináris jellegének demonstrálása, olyan feladatok bemutatása, amelyek más tárgyakban ismertett eljárások, módszerek alkalmazását igénylik, szükségességüket alátámasztják és megoldásaik gyakorlati háttérül szolgálnak. Ilyenek: rendszertechnika, diszkrét folyamatok irányítása és szimulációja, gyártási folyamatok tervezése, matematika, operáció kutatás, informatika stb.

Nem célja a gyakorlatnak a gyártórendszerek irányításának és programozásának teljeskörű megismerése, a programozási utasítások memorizálása. Az ismeretek bővítésére a későbbiekben a további tanulmányok során nyílik lehetőség célirányos modulválasztás útján (robottechnika, gyártásautomatizálás, gépgyártástechnológia modulok) továbbá TDK-ba történő bekapcsolódás útján és a diplomatervezés keretein belül.

#### 22.0.1 Alapfogalmak

- Gyártó rendszer:

olyan gépekből, kiszolgáló berendezésekből és irányító rendszerből álló komplex technológiai objektum, amely anyagi, energetikai és informatikai erőforrások felhasználásával gyártási rendelések alapján félkész vagy kész terméket állít elő.

-Rugalmas gyártórendszer (Flexible Manufacturing System: FMS):

többféle termék ill. termékcsoport előállítására alkalmas gyártórendszer, amely egyszerűen átprogramozható a termelési célnak megfelelően.

-Gyártócella:

olyan technológiai erőforrások, munkahelyek, berendezések csoportja, amely a technológiailag hasonló feladatok operációinak zárt vagy közel zárt termelési ciklusban való végrehajtására képes.

-Rugalmas gyártórendszer irányítása:

olyan irányítási feladat, amely valós időben irányítja és felügyeli a gyártórendszerben lezajló komplex gyártási feladatokat. Az irányítás az előre megtervezett és a gyártási rendelésben megadott aktuális célokat igyekszik elérni. A gyártórendszerek irányítása *sokváltozós, komplex,koordinációs* típusú irányítás:

*sokváltozós*, mert a rugalmas gyártás körülményei között a napi gyártási események száma igen nagy is lehet,

*komplex*, mert a lehetséges vezérlések tartományaihoz többféle komplex célfüggvény adható meg, amelyek extremuma számos korlátozó feltétel mellett keresendő,

*koordinációs*, mert az adott erőforrásokhoz az operációk sokféle soros-párhuzamos kombinációja rendelhető.

-Rugalmas gyártórendszer programozása:

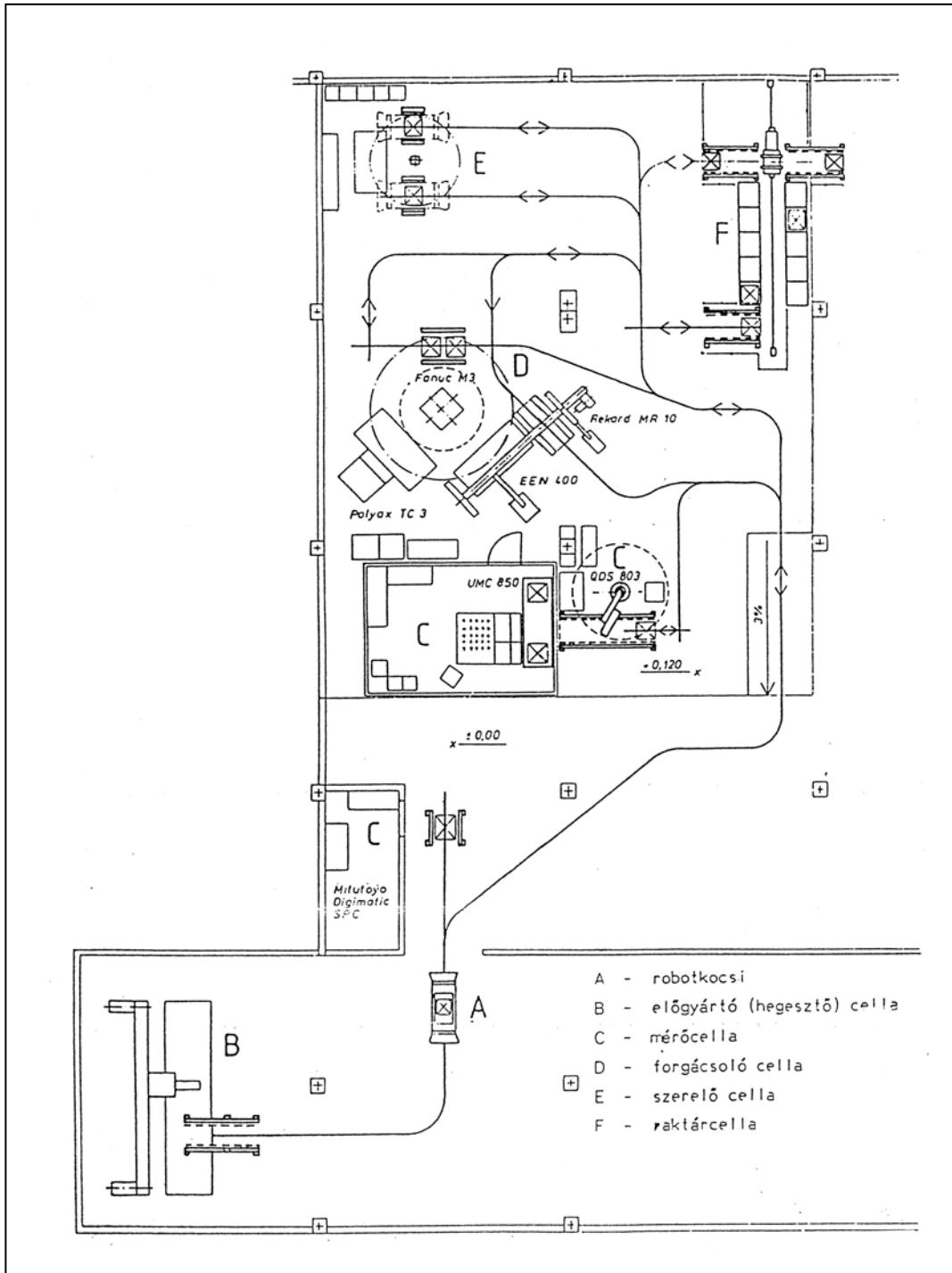
gyártási utak, gyártási szekvenciák kijelölése az optimális gyártás feltételeinek megfelelően. Hierarchia szintben az NC programozás és robot programozás felett áll, de

támaszkodik ezekre (a gyártórendszer program indítja és leállítja az NC, robot stb. programokat). Eszköze a programnyelv (CSL, WALLI stb.)

-GANTT diagram (ütemdiagram):

az események időbeni sorrendjének ábrázolására alkalmas speciális idődiagram, amely ütemenként (lépésenként) szemlélteti a változásokat. Az ütemdiagramon ábrázoljuk pl. a gyártóeszközök foglaltsági állapotát. Az ütemek hossza nem feltétlenül időarányos, de sorrendjük a valóságnak megfelelő.

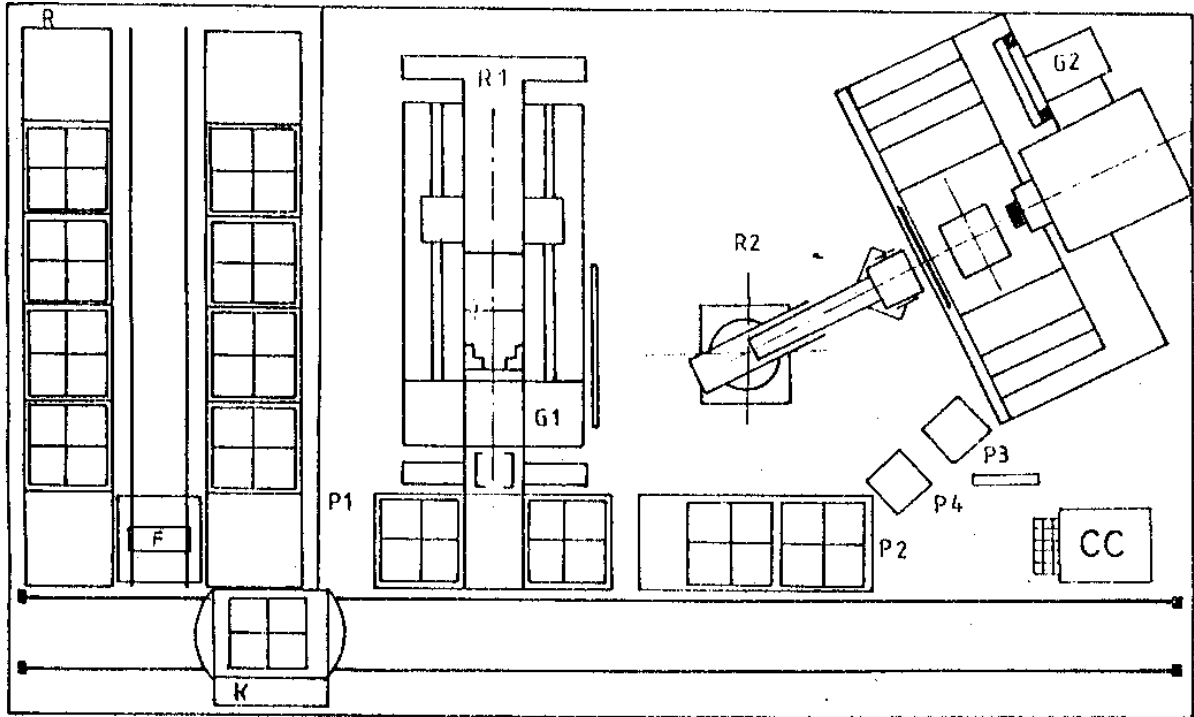
### 22.0.2 A BME GTT mintarendszerének (FMS) telepítési vázlatja



## 22.1 A SICELL 2.1 (BME) GYÁRTÓCELLA SZIMULÁTOR

### 22.1.1 A szimulációs program általános leírása

A SICELL 2.1 (BME) szimulációs program a BME GTT mintarendszerének strukturáját modellezi. A szimulátor a mintarendszer programozásának és időbeli eseményfolyamatának tanulmányozására alkalmas.



22.1. ábra – A SICELL gyártócella vázlatja

A cellaszintű programozás a CSL programozási nyelv szűkített és interaktív változatával történik. A rendszerszintű cellavezérlő program három egymással párhuzamosan futó megmunkáló és szállító rész-programból (taszk-ból) áll. A szimulált rendszer paramétertáblázata lehetővé teszi az egyes funkciók tulajdonságainak bizonyos határok közötti változtathatóságát. Ezáltal analizálható e paraméterek hatása a cella működését értékelő mutatókra.

A program által szimulált cella vázlatos telepítési elrendezése (lay-out) a 22-1 ábrán látható. A cella a következő részegységekből áll:

- CNC vezérlésű univerzális eszterga (G1)
- CNC vezérlésű fúró-maró központ (G2)
- Felsőpályás kiszolgáló robot (R1),
- Henger-koordináta-rendszerben működő 4 tengelyes kiszolgáló robot (R2),
- Palettatároló aktív puffer a G1-hez (P1),
- Fordító vagy segédművelet végző állomás (P3),
- Ideiglenes tárolóhely (puffer)(P4),
- Palettaszállító robotkocsi (vezető nélküli targonca, AGV) (K),
- Palettatároló magasraktár (R),
- Magasraktári felrakógép (F),
- Cellavezérlő számítógép (CC).

A szimulátor (ebben a verzióban) a következő jellegzetes cellaműködést tételezi fel:

A magasraktárban a nyers munkadarabok palettán helyezkednek el. A cellavezérlő számítógép cellaszintű termelési programja alapján a felrakógép egy palettát a robotkocsira helyez. A robotkocsi a palettát a program szerint a P1 vagy P2 pufferhez viszi, és ott betárolja. A pufferek aktív tárolók, melyek léptetésével viszik a palettát az R1, illetve az R2 robotok által elérhető felszedési „pick up”, illetve lerakási. „put down” helyzetbe.

A munkadarabok megmunkálása a művelettervek, illetve a hozzájuk tartozó NC programok szerint zajlik le. A készre munkált munkadarabok ugyanarra a palettára kerülnek vissza, amelyen érkeztek. A palettákat a robotkocsi szállítja vissza a cellából a raktárba, ahol a felrakógép helyezi azokat eredeti helyükre.

A szimulátor a következő korlátozó megszorításokkal működik:

- A raktárban és a cellában egyidejűleg csak maximum kétféle (A és B) típusú munkadarab tartózkodhat (zöld és piros jelű).
- A nyers és kész munkadarabok palettán helyezkednek el. Egy palettán csak egyféle és maximum 4 munkadarab lehet.
- A magasraktár kapacitása maximum 10 paletta.
- A P1 és P2 aktív pufferen egyidőben maximum 2 paletta tartózkodhat.
- A P3 állomáson és a P4 pufferen egyidejűleg csak egy munkadarab tartózkodhat.
- Mindkét robot „egyes” markolóval van ellátva, fordítási művelet mindkét robottal lehetséges.
- A cellában a robotok csak az előzetesen betanított programokat tudják végrehajtani. A programok cellaszintű parancsokkal aktiválhatók.
- Az R1 roboton betanított különféle programok (MOVE utasítások) száma 3.(lásd a 2. taszk leírását)
- Az R2 roboton betanított (mozgató (MOVE) utasítások) száma 14. (lásd a 2. taszk leírását)
- A cella szekvenciális működését a cellavezérlő irányítja, maximum három parallel task-ból álló cellaprogram alapján. (1. taszk: felrakógép+robotkocsi, 2. taszk: R1 robot+G1 gép, 3. taszk: R2 robot+G1 gép+G2 gép).
- Ügyelni kell a taszkok írásakor, hogy a paletták illetve a munkadarabok taszkon belüli belépési és kilépési pontja azonos legyen.(szinkronizációs pont)
- Az aktuálisan indítható megmunkálási vagy anyagkezelési műveleteket a cellaprogram és a cella státusa együttesen határozzák meg.

*A lehetséges taszkok és az azokban kiadható műveletek az alábbiak:*

A **taszk 1** a raktár és a robotkocsi rendszerszintű programozására szolgál. (választható F1-gyel)

A **taszk 1**-ben kiadható utasítások:

- FRKA MOVE 4 (Felrakógép, Raktár, Kocsi, A tip. mdb)
- FRKB MOVE (Felrakógép, Raktár, Kocsi, B tip. mdb)
- FKR MOVE (Felrakógép, Kocsi, Raktár, paletta)
- KRP1 MOVE (Kocsi, Raktár, P1 puffer, paletta)
- KRP2 MOVE (Kocsi, Raktár, P2 puffer, paletta)
- KP1R MOVE (Kocsi, P1 puffer, Raktár, paletta)
- KP2R MOVE (Kocsi, P2 puffer, Raktár, paletta)
- KP1P2 MOVE (Kocsi, P1 puffer, P2 puffer, paletta)

KP2P1 MOVE (Kocsi, P2 puffer, P1 puffer, paletta)

WP1 WAIT (Kocsi, P1 puffer)

WP2 WAIT (Kocsi, P2 puffer)

A **taszk 2** a P1 puffer, a felsőpályás R1 robot és a G1 eszterga rendszerszintű programozására szolgál. (választható F2-vel)

A *taszk 2-ben kiadható utasítások:*

R1 P1 G1 MOVE (R1 robot, P1 puffer, G1 gép, munkadarab)

R1 G1 P1 MOVE (R1 robot, G1 gép, P1 puffer, munkadarab)

R1 G1 F MOVE (R1 robot, G1 gép, fordít, munkadarab)

ST G1 START (G1 gép, NC program indul)

W G1 WAIT (G1 gép, NC program kész)

W R2 WAIT (R2 robot, ROC flag)

A **taszk 3** a hegerkoordinátás R2 robot, a P2 puffer, a G2 fúró-maró gép, a G1 eszterga, és a P3 kiegészítő állomás rendszerszintű programozására szolgál. (választható F3-mal)

A *taszk 3-ban kiadható utasítások:*

R2 P2 G1 MOVE (R2 robot, P2 puffer, G1 gép, mdb)

R2 G1 P2 MOVE (R2 robot, G1 gép, P1 puffer, mdb)

R, P2 G2 MOVE (R2 robot, P2 puffer, G2 gép, mdb)

R2 G2 P2 MOVE (R2 robot, G2 gép, P2 puffer, mdb)

R2 G1 G2 MOVE (R2 robot, G1 gép, G2 gép, mdb)

R2 G2 G1 MOVE (R2 robot, G2 gép, G1 gép, mdb)

R2 P2 P3 MOVE (R2 robot, P2 puffer, P3 állomás, mdb)

R2 P3 G2 MOVE (R2 robot, P3 állomás, 'G2 gép, mdb)

R2 G2 P3 MOVE (R2 robot, G2 gép, P3 állomás, mdb)

R2 P3 P2 MOVE (R2 robot, P3 állomás, P2 puffer, mdb)

R2 G1 P3 MOVE (R2 robot, G1 gép, P3 állomás, mdb)

R2 P3 G1 MOVE (R2 robot, P3 állomás, G1 gép, mdb)

R2 G1 P MOVE (R2 robot, G1 gép, fordít~, mdb)

R2 G2 F MOVE (R2 robot, G2 gép, fordít, mdb)

ST G1 START (G1 gép, NC program)

ST G2 START (G2 gép, NC program)

ST P3 START (P3 állomás, PLC program)

W.F1 WAIT' (G1 gép, NC program kész)

W G2 WAIT (G2 gép, NC program kész)

W R1 WAIT (R1 robot, ROC flag)

W P3 WAIT (P3 állomás, PLC bit)

### 22.1.2 A szimulátor programozása és kezelése

A szimulátor működése négy szakaszra bontható:

1. A rendelés programozása. (kézi adatbevitel dialógus alapján)
2. A gyártórendszer szekvenciális működésének programozása.(taszkok megírása )
3. Eseményszimuláció grafikus animációval. (szimulátor végzi automatikusan)

4. A cellaműködés értékelése, dokumentálása. (szimulátor végzi automatikusan)

#### **22.1.2.1 A rendelés programozása (felvétele)**

A cellaprogramozás és szimuláció a SICELL parancs kiadásával indul. A megjelenő menü alapján választhatjuk a "demo" módot, vagy betöltünk egy már korábban elkészített programot, vagy a szekvenciabeviteli üzemmódot választjuk saját program készítése céljából

A rendelés felvétele a szimulált anyagkezelési műveletek adatainak (paramétereinek) megadásával kezdődik. A paraméterek a szimulátor dialógusában a megadott tartományhatárok között beállíthatók. A rendelésvétel következő lépése a pillanatnyilag rendelkezésreálló kapacitás kijelölése. Megadhatjuk, a gépek technológiai kapacitásának idejét (sec/műszak)-ban. A default érték 28800 (sec/műszak) gépenként. Ugyanebben a táblázatban adjuk meg a raktári. nyersdarab készletet az A és B típusú alkatrészekre: A maximális raktári kapacitás  $10 \times 4 = 40$  munkadarab. A default érték 20 db A és 20 db B alkatrész. Az animátor a palettázást automatikusan végzi el. A raktárban maximálisan 10 paletta fér el, egy palettán maximálisan 4 munkadarab lehet. A szimulátor ezen verziója a palettázás nagyobb rugalmasságát - pl. a palettákon csak egy-egy darab - nem teszi lehetővé:

#### **22.1.2.2 A rendszer szekvenciális működésének programozása**

A következő lépés a munkadarabok gyártási-szállítási szekvenciáinak megadása egyszerűsített műveleti sorrendterv formájában - a műveleti sorrendtervben meg kell adni a műveletek kívánt sorrendjét és az A és B típusú alkatrészekre vonatkozóan a G1 és a G2 gépeken igényelt műveleti időket.

A műveleti sorrendterv meghatározza a munkadaraboknak a rendszeren belüli útvonalát, illetve az útvonalhoz kapcsolódó anyagkezelési eseménysorozatot.

A műveletterv három rovatból áll:

1. A művelet megvalósításának sorszáma ( Ez a műveletek sorszáma)
2. A technológiai erőforrás sorszáma ( Az adott rendszerben ez csak G1 vagy G2 NC gépek lehetnek)
3. A művelethez tartozó NC program futási ideje.( Ez a teljes várható NC műveleti idő)

A művelet megadásánál csak az erőforrás (NC-gép) azonosító számát és a műveleti időt kell megadni, ez utóbbit (sec) mértékegységben. A műveletterv végét a G betű után irt 0 azonosítóval jelezzük.(G0).

A műveletterv menü képen kiadható parancsok:

ALT X kilépés a programból

ESC visszalépés a "programozási mód" menüképre,

ENTER "A" típusú műveletterv táblázatból, átlépés a "B" típusra (G0 után), műveletterv megadás vége, továbblépés a programozási menüképre (G0 után)

A műveletterv szerkesztő a G azonosító számaként-csak 0, 1, 2 számokat fogad el.

Az idő rovatban a maximális érték 9999 lehet. Hibás leütés az időadatban karakterenként a szokásos PC szerkesztő billentyűkkel javítható.

## PROGRAMOZÁSI PÉLDA

### Feladat „M1” :

Az „A” típusú munkadarab megmunkálási igénye: a G1 gépen egy esztergálási művelet (200 sec). A palettán 3 db „A” típusú munkadarab helyezkedik el.

„B” típusú munkadarab megmunkálási igénye: a G1 gépen egy 100 sec idejű művelet és a G2 gépen egy 200 sec idejű művelet, végül mérés a P3-on lévő mérőkészülékben. A palettán 1 db „B” típusú munkadarab helyezkedik el.

### Programozás:

A feladatnak megfelelően írjuk le az alkatrészek útvonalát és a kezelési eseménysorozatot:

Az „A” típusú alkatrészekre : R-F-K-P1-R1-G1-R1-P1-K- F-R

A „B” típusú alkatrészekre: R-F-K-P2-R2-G1-R2-G2-R2-P3-R2-P2-K-F-R

A programozás következő lépésében a sorrendterveket három egy-időben, párhuzamosan futó feladatra (taszk) kell bontani: **szállítási** (R,F,K,P1,P2), **megmunkálási1** (P1,R1,G1) és **megmunkálási2** (P2,R2,G1,G2,P3) taszkokra.

### Taszk 1. (szállítási taszk):

0. INIC ( Inicializálás)

1. FRKA ( a felrakógép egy „A” típusú munkadarabokat tartalmazó palettát a kocsira tesz)

2. KRP1 ( A kocsi a palettát a cellába viszi és a P1 pufferre helyezi)

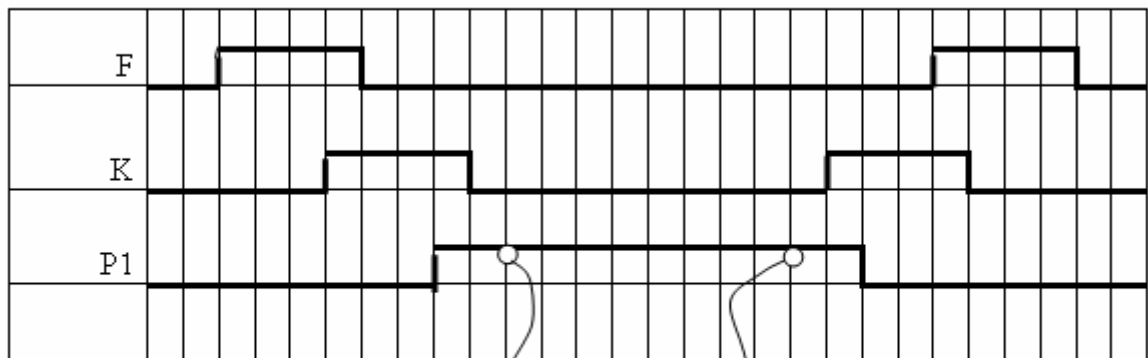
A következő 3. és 4. utasítások csak akkor aktivizálódnak,  
ha a megmunkálási1 taszk jelző bitet küld.

3. KP1R ( A.kocsi a kész palettát visszaviszi)

4. FKR (A felrakógép a palettát visszateszi a helyére)

5. REPEAT (Ismétlés)

#### Taszk 1. (szállítás raktárból P1-re és vissza)



Jelző bit (szemafor) a  
**megmunkálási1 taszkhoz:**  
„paletta érkezett nyers  
darabokkal a P1 pufferbe”

Jelző bit a **megmunkálási1**  
**taszktól:** „a palettán lévő  
minden munkadarab kész, a  
paletta a raktárba szállítható”

A fenti program az egyszerű érthetőség miatt csak a P1 puffer kiszolgálását mutatja be. Példánkban azonban a P2 puffert is ki kell szolgálni, ezért a programot értelemszerűen ki kell bővíteni:

0. INIC ( Inicializálás)

1. FRKA ( a felrakógép egy „A” típusú munkadarabokat tartalmazó palettát a kocsira tesz)

2. KRP1 ( A kocsi a palettát a cellába viszi és a P1 pufferre helyezi)

3. KP1R ( A.kocsi a kész palettát visszaviszi)

4. FKR ( A felrakógép a palettát visszateszi a raktárba)

-----

5. FRKB ( a felrakógép egy „B” típusú munkadarabokat tartalmazó palettát a kocsira tesz)

6. KRP2 ( A kocsi a palettát a cellába viszi és a P2 pufferre helyezi)

7. KP2R ( A.kocsi a kész palettát visszaviszi)

8. FKR (A felrakógép a palettát visszateszi a raktárba)

-----

9. REPEAT (Ismétlés)

Az így módosított **Taszk 1**-hez tartozó GANTT-diagram az egyszerű példában bemutatotthoz hasonló felépítésű, csak több elemből áll.

### Taszk 2 (megmunkálási1 taszk) :

Akkor indul a megmunkálási1 taszk, amikor a szállítási taszk munkadarabot tesz a P1 pufferre. A taszk egymarkolós robot esetében az alábbi hat utasításból áll:

0. INIC (Inicializálás)

1. R1 P1 G1 (Az R1 robot betölt egy nyers mdb-ot a P1 pufferból a G1 gépre)

2. ST G1 (Indul ez NC program)

3. W G1 (A robot a pr. végére vár)

4. R1 G1. P1 (A robot a kész darabot visszateszi a P1 pufferbe)

5. REPEAT (Ismétlés)

### Taszk 2. (megmunkálás1: P1, R1, G1)



A megmunkálás után a munkadarabok ismét a P1 pufferbe kerülnek vissza. (Programozási szabály, hogy a belépési és kilépési puffer legyen azonos)

### Taszk 3 (megmunkálási2 taszk):

A megmunkálási2 taszk akkor indul, amikor a szállítási taszk munkadarabot tesz a P2 pufferre. A taszk a G1, G2 gépen végzett megmunkálásokat és a P3 pufferre telepített PLC vezérelt mérési műveleteket végzi, majd a műveletek befejeztével jelzőbitet (flag) állít át a szállítási taszk részére. A taszk a következő utasításokból áll:

0. INIC (Inicializálás)

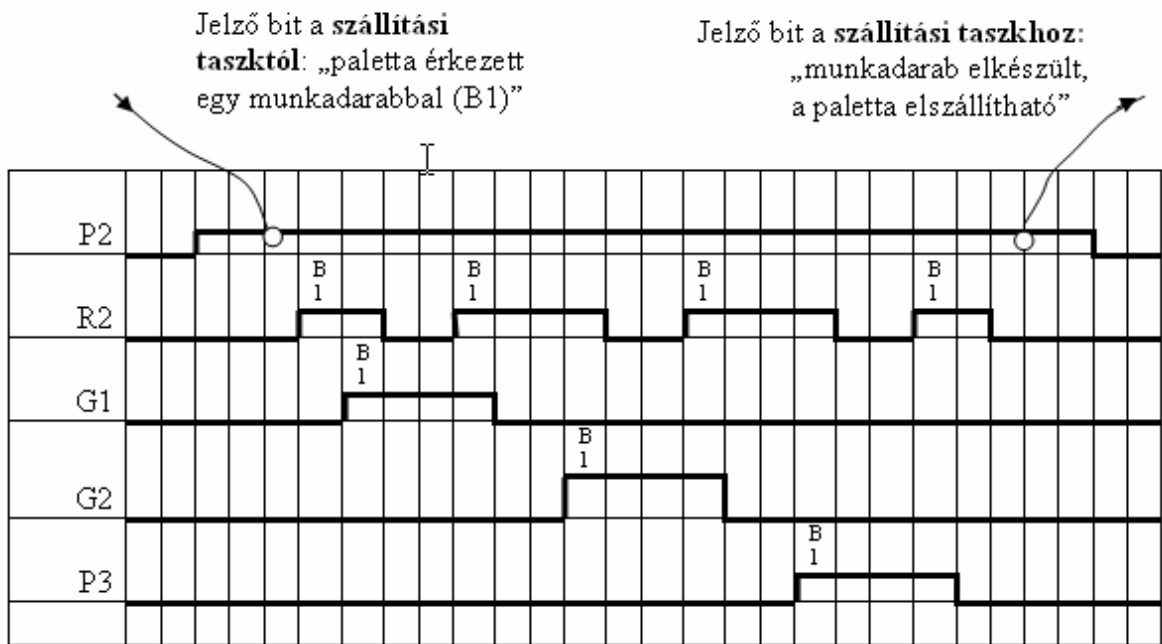
1. R2 P2 G1 ( a robot betölti a nyers mdb-ot a P2 pufferen lévő palettáról a G1 gépbe)

2. ST G1 (Indul az 1. NC program)



3. W G1 (a robot a program végére vár)
4. R2 G1 G2 (a darabot a robot áthelyezi)
5. ST G2 (indul a 2. NC program)
6. W G2 (a robot a megmunkálás végére vár)
7. R2 G2 P3 (a robot áthelyezi a kész darabot a P3-on lévő mérő berendezésbe)
8. STP3 (indul a PLC vezérlésű mérés)
9. WP3 (a robot a mérés végére vár)
10. R2 P3 P2 (a robot a kész darabot a P2 pufferben lévő palettára teszi)
11. REPEAT(ismétlés)

**Taszk 3. (megmunkálás2: P2, R2, G1, G2, P3)**



A cellavezérlő összehangolja („összefesüli”) a fenti taszkok működését, azaz a taszkok és a rendelési állomány ismeretében elvégzi az ütemezést, idő-optimális finomprogramot készít a közös erőforrások terhelésének célszerű beosztásával, végül grafikus animációt végez.

A programozót nem terheli a teljes gyártási folyamat egyben történő áttekintése, elegendő a különálló taszkok kidolgozása. A számítógép elvégzi az optimalizációt a programozó helyett.

Előnyös az is, hogy a taszkok megírásakor nem kell figyelembe venni a rendelésiállományt, bár a különféle típusú munkadarabok száma jelentősen befolyásolhatja az optimális finomprogramot, mert a közösen használt erőforrások (robotkocsi, szerszámgépek stb.) terhelése változhat. A számítógép ebben is segít.

Az optimalizációnál az alábbi konfigurálható szerviz időket veszi figyelembe a program:

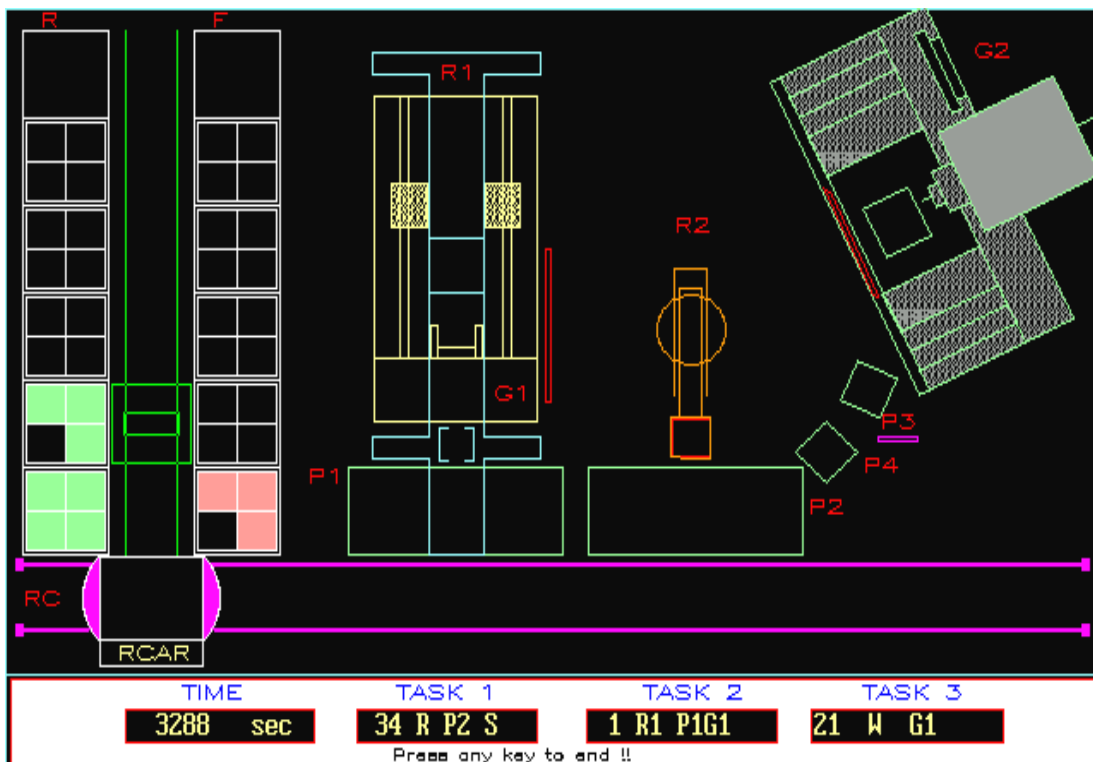
R2 robot ki-betárol gépbe (20-100)	40 [ sec ]
R2 ki-betárol pufferbe (10 - 50)	30 [ sec ]
R2 robot lin. sebessége (10-50)	30 [mm/sec]
R2 robot szögsebessége (10-25)	20 [°/sec ]
R1 robot ki-betárol gépbe (5-50)	5 [ sec ]
R1 robot ki-betárol pufferbe (1-10)	5 [ sec ]

R1 robot lin. sebessége	(50-500)	300 [mm/sec]
R1 robot fordítási idő	(5-30)	5 [ sec ]
Ajtó ki/be idő	(3-10)	5 [ sec ]
Paletta puffer lépési idő	(3-10)	5 [ sec ]
Felrakógép sebessége	(50-500)	100 [mm/sec]
RC sebessége	(200-500)	300 [mm/sec]
R2 robot fordítási idő	(5-30)	5 [ sec ]
Raktár-cella távolság	(7-28)	7 [ m ]
P3 műveleti idő	(5-50)	20 [ sec ]

### 22.1.2.3 Eseményszimuláció grafikus animációval

A SICELL 2.1 program elsősorban oktatási célra készült. Ennek megfelelően fontos szolgáltatása a programozott cellaműködés grafikus animációja. Az animáció elsősorban a munkadarabok illetve a paletták rendszeren belüli mozgásának szemléletes nyomkövetését szolgálja. A szimuláció és az animáció eseményvezérelt, és virtuális időben folyik. A szimulációs idő diszkrét inkrementekben múlik ( $dT = 1$  sec). A robotok mozgása, a munkadarabok és paletták helyváltoztatása az animációs alprogramok időosztásos futásával valósul meg. Az animációs layout-on követhetjük a felrakógépet, a robotkocsit, a robotokat, a palettákat és a munkadarabok mozgását és a cella minden fontos eseményét. Mivel a grafika futtatási időigénye változik, a szimuláció látszólagos "időmúlása" nem lineáris.

A szimuláció a grafikus animáció mellett egy; a háttérben futó időbeli eseménykövetésből áll. A képernyő alján megjelenik a szimulációs cellaidő valóságos (aktuális) értéke és a három tászk aktuális végrehajtás alatt álló cellaparancsa is. A szimuláció futását bármikor megszakíthatjuk tetszőleges billentyű leütésével. Az alábbi ábrán a szimulátor képernyőjét láthatjuk:



#### **22.1.2.4 A cellaműködés értékelése, dokumentálása**

A szimuláció lefutása után a SICELL program elvégzi a szimuláció értékelését a legfontosabb adatok összeállításával. Az adatok három táblázatban jelennek meg. Az első táblázat az összegzett idő adatokat tartalmazza (sec) mértékegységben. A második táblázatban a paletta ciklusidők egyedi és átlagos értéke található. Az első sorban az összes azonos típusú paletta ciklusidejének átlaga, a következő sorokban az egyedi ciklusidők. A harmadik táblázatban a cella működési hatékonyságának mutatói találhatók.

Ezek a mutatók mind relatív, azaz %-os értékűek. Két típusuk van. Az első típus vonatkozási alapja a teljes szimulációs idő. A második típus vonatkozási alapja valamely funkció összegzett működési ideje. Ez utóbbi mutatótípust a paletták kihasználtságának jellemzésénél használjuk.

A táblázatokat a program dokumentáció formájában printeren is kinyomtatja. Ha a szimuláció programját meg akarjuk őrizni, lehetőség van file kezelési művelettel a cellaprogramot és a művelettervi adatokat adatfile-ba menteni. Ez a file később újra betölthető, és a program újbóli felépítése megtakarítható. Az összesített időadatok és a kihasználtsági mutatók alapján minősíthetjük az adott "rendelés" megvalósításának gazdaságosságát

#### **22.1.2.5 Programozási hibák**

A programozásnál a következő jellegzetes hibákat követhetjük el.

A robotkocsi műveletek programozásánál:

- Nincs FRK vagy FKR utasítás. Nem indul a felrakógép, az idő fut.
- Nincs P1 vagy P2 célú utasítás. A kocsi nem indul el, a falrakógép eredménytelenül kísérletezik.
- Nincs P1 célú utasítás. A kocsi nem megy a palettákért, az idő fut.
- Van P1 vagy P2 célú utasítás, de a kész darabok a másik pufferben vannak. A kocsi nem indul el, az idő fut.

-

A cella taszkok programozásánál előforduló hibák:

- Nincs P1 vagy P2 forrású robotutasítás. A nyers darabok a palettán maradnak, az idő fut.
- Nincs elindítva a gépeken az NC program. A cellaciklus elakad, az idő fut.
- Nincs P1 vagy P2 célú robotutasítás. A kész darab nem kerül vissza a palettára, az idő fut.
- Hiányzik a fordító utasítás. A két egymást követő NC program lefut. A hiba nem észrevehető.
- Nincs elindítva a P3 állomáson a PLC program. A cellaciklus elakad, az idő fut.
- Egy munkadarabra a G1 és G2 gépen is van művelet tervezve és hiányzik a G1→G2, illetve a G2→G1 paraméterű robotutasítás. A cellaciklus elakad, az idő fut.

A WAIT utasítások kiadására a cellaszintű program nem túlságosan kényes. Legtöbb esetben az egyéb feltételek teljesülése helyes irányába tereli a cellaciklus eseményeinek szekvenciáját. Megadása azonban ajánlatos, mert ezek biztos szinkronizációs pontokat jelentenek, különösen bonyolultabb munkadarab útvonalak programozásánál.

A robotkocsi programjának futása a rugalmasabb. Itt a megadott szekvencia alárendeltebb jelentőségű a feltételekhez képest. (Ilyen feltételek például : van paletta, nincs paletta, milyen a paletta státusa, stb.).

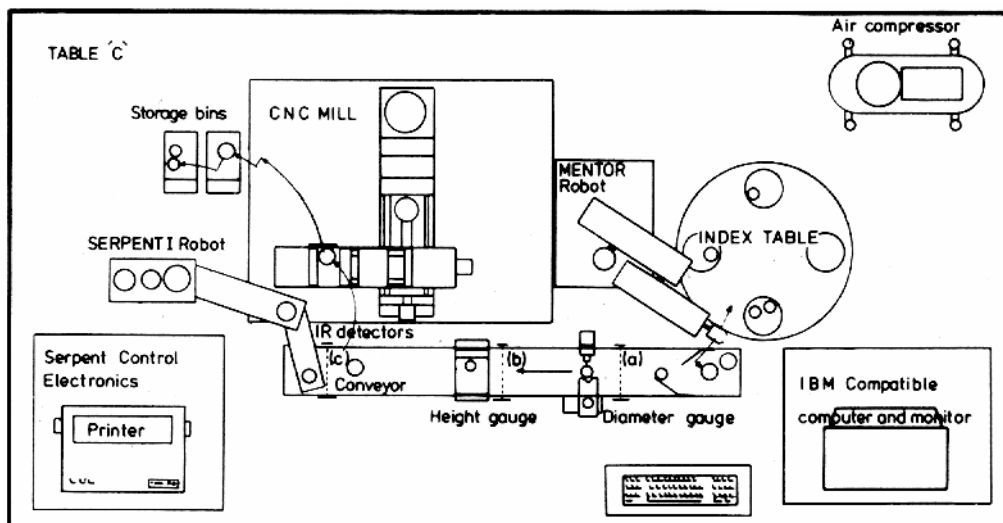
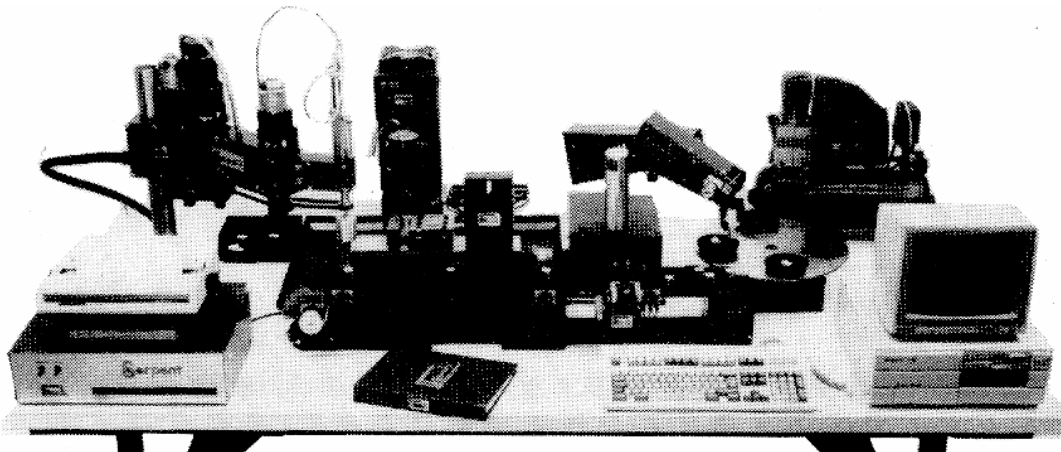
A kocsimozgás szekvenciáját a feltételek teljesülése vezérli.

A cellán belüli két taszk ( a 2. és 3. taszk) belső szekvenciáját viszont elsősorban a cellaprogram határozza meg. Ez szorosan összefügg a művelettervek szekvenciájával. Nem lehet ugyanis ellentmondás a technológiai sorrendi követelmények és az anyagkezelési műveletek sorrendje között. Ha tehát egy művelettervet megadtunk, a cellaprogram ezzel nem lehet ellentmondásos. A programok elkészítésében további segítséget nyújthat:

Erdélyi F.- Csáki T.- Urbán I.: Gyártócellák működésének időbeli szimulációja c. programozási leírás, amely a jelen gyakorlat alapjául szolgált.

## 22.2 A SPEC GYÁRTÓRENDSZER

egy grafikus szimulátor a gyártórendszer eseményeinek jó áttekintését biztosítja, de sokkal szemléletesebb képet ad egy valóságos működő modell, amelyen a gyártási folyamat nyomon követhető és irányítórendszerének működése hardveresen is tesztelhető. A 22.2. ábrán látható rendszer egy marógépből (CNC MIL), két robotból (Serpent , Mentor), egy körasztalból (Index table) és egy szállítószalagból (Conveyor) áll. A szállítószalagon egy átmérő mérő (Height gauge) és egy magasságmérő (Diameter gauge) szenzor van elhelyezve a nyers munkadarabok méretének ellenőrzése céljából. A fenti eszközök közös busz-rendszeren keresztül kommunikálnak egymással, de ezt megkerülve, alsóbb szinten: közvetlen input/output vonalakon is elérhetik egymást.



22.2. ábra A

### SPEC gyártórendszer

A rendszer magas szintű programozási nyelvvel (WALLI 2.0) programozható. A nyelv alkalmas a gyártási folyamat eseményorientált irányítására. Hasonlóan más strukturált nyelvekhez tartalmaz deklarációs utasításokat, szubrutin hívásokat stb. A nyelvnek

különleges tulajdonsága, hogy rendelkezik a párhuzamos folyamatok indításának és leállításának utasításaival, ami gyártórendszerek irányításához elengedhetetlen, hiszen a munkagépek időben párhuzamosan dolgoznak. A programozási nyelv jellegének érzékeltetése céljából az alábbiakban felsoroljuk utasításait kategóriájuk szerint csoportosítva és bemutatunk egy programozási példát.

#### **Eszközdeklaráló utasítások:**

- DR Define Robot <n> <Iv'1/N2/ME/SE/NA/KE> addressed on line A<n> sequence in <fn>
- DM Define Mill <n> addressed on line A<n> program <n> in <fn>
- DX Define eXpansion board <n> addressed on line A<n>
- \* DC Define Conveyor <n> increment from <R/X> <n> digital output <n> start from <R/X> <n> digital output <n> halt from <R/~> <n> digital output <n>
- \* DT Define indexing Table <n> increment from <R/X> <n> digital output <n> start from <R/X> <n> digital output <n> halt from <R/X> <n> digital output <n>
- \* DG Define Gauge <n> close from <R/X> <n> digital output <n> opera from <R/X> <n> digital output <n> potentiometer to robot <n> analogue input <n>
- \* DS Define Sensor <n> output to <R/M/X> <n> digital input <n>

#### **Eszköz aktivizáló utasítások:**

- MO MOve robot <n> to position <n/A/B/C/D>
- RG Robot <n> Gripper <0/C/P>
- SM Start Mill <n> program <n>
- CM Continue Mill <n> program
- \* JM Jog Mill <n> <n> microns and continue
- WA WAit for <n> second(s)
- TR TRigger from <R/X> <n> digital output <n>
- PU PUlse train from <R/X> <n> digital output <n> pulses <n> at speed <n> `
- SE SEt <R/X> <n> digital output <n> <H/L>
- \* SC Start Conveyor <n>
- \* HC Halt Conveyor <n>
- \* IC Inerement Conveyor <n> <n> mm at speed <n>
- \* ST Start indexing Table <n>
- \* HT Halt indexing Table <n>
- \* IT Inerement indexing Table <n> <n> degrees at speed <n> \* CG Close Gauge <n>
- \* OG Open Gauge <n>

#### **Program szervező utasítások:**

- FO FOr <A/B/C/D> = <n> to <n>
- NE Next <A/B/C/D>
- GO GOto line <n>
- GS Goto Subroutine at line <n>
- RE REturn from subroutine
- SP Start Parallel subroutine at line <n>
- EP End Parallel subroutine
- SF Set Flag <n> <H/L>
- \* CB Call Basic program <fn>
- CO COmment <text>

#### **Várakozó, szinkronizációs utasítások:**

- PF Pause until Flag <n> <H/L>
- PP Pause until robot <n> is in Position
- PD Pause until <R/M/~> <n> Digital input <n> <H/L>

- PM Pause until Mill Gn> flag <F/P> <H/L>
- \* PS Pause until Sensor <n> <H/L>

### Feltételes utasítások:

- IF If Flag <n> <H/C> then <statement>
- IP If robot <n> is in Position then <statement>
- ID If <R/M/X> Digital input <n> <H/I> then <statement>
- IA If robot <n> Axis <n> is between <n> and <n> then <statement>
- II If robot <n> analogue Input <n> is between <n> and <n> then <statement>
- IM If Mill <n> flag <F/P> <H/L> then <statement>
- \* IG If Gauge <n> is between <n> and <n> then <statement>
- \* IS If Sensor <n> <H/L> then <statement>
- \* IB If Basic flag <H/C> then <statement>

### FMS mintaprogram WALLI 2.0 nyelven:

A MINTAPROGRAM HÍVATKOZIK: A ROBOTOK KORÁBBAN BETANÍTOTT, KOORDINÁTÁKKAL ADOTT POZÍCIÓIRA (POSITION 1, 2, 3, ...N),  
ROBOT PROGRAMOKRA ( A SERP-1 ÉS SERP-2 FÁJLOKBAN)  
NC PROGRAMOKRA (A MILL1-C ÉS MILL2-C FÁJLOKBAN)

FONTOS, HOGY AZ ADATOK KORÁBBAN A RENDSZERBE BE LEGYENEK TÖLTVE, HOGY A KIVÁNT PILLANATBAN AKTIVIZÁLHATÓK LEGYENEK

```
0 COMMENT .....
10 COMMENT ..... Table C 10/12/88
20 COMMENT .....
30 DEFINE ROBOT SERPENT ADDRESSED ON LINE A5 SEQUENCE IN SERP-C
40 DEFINE ROBOT 2 MENTOR ADDRESSED ON LINE A4 SEQUENCE IN MENT-C
50 DEFINE MILL 1 ADDRESSED ON LINE A4 PROGRAM 1 IN MILL1-C
60 DEFINE MILL 1 ADDRESSED ON LINE A4 PROGRAM 2 IN MILL2-C
70 DEFINE SENSOR 0 OUTPUT TO ROBOT 2 DIGITAL INPUT 1
80 DEFINE SENSOR 1 OUTPUT TO ROBOT 1 DIGITAL INPUT 7
90 DEFINE SENSOR 2 OUTPUT TO ROBOT 1 DIGITAL INPUT 2
100 DEFINE CONVEYOR 0 INCREMENT FROM ROBOT 1 DIGITAL OUTPUT 3
    START FROM ROBOT 1 DIGITAL OUTPUT 1
    HALT FROM ROBOT 1 DIGITAL OUTPUT 2
110 DEFINE INDEX TABLE 0 INCREMENT FROM ROBOT 2 DIGITAL OUTPUT 2
    START FROM ROBOT 2 DIGITAL OUTPUT 1
    HALT FROM Not connected
120 DEFINE GAUGE 0 CLOSE FROM ROBOT 1 DIGITAL OUTPUT 5 OPEN FROM ROBOT 1
    DIGITAL OUTPUT 4 POTENTIOMETER TO ROBOT 2 ANALOGUE INPUT 1
130 COMMENT ..... Open Mentor gripper
140 ROBOT 2 CRIPPER OPEN
150 OPEN GAUGE 0
160 START INDEX TABLE 0
170 PAUSE UNTIL SENSOR 0 LOW
180 COMMENT ..... Move Mentor
190 MOVE ROBOT 2 TO POSITION 1
200 PAUSE UNTIL ROBOT 2 IS IN POSITION
210 INCREMENT INDEX TABLE 0 13 degrees SPEED 1
220 ROBOT 2 GRIPPER PROGRAMMED
230 SET FLAG 1 LOW
240 START PARALLEL SUBROUTINE AT LINE 410
250 START PARALLEL SUBROUTINE AT LINE 1150
260 COMMENT ..... *****
270 COMMENT ..... " Gauge control
280 COMMENT .....
290 PAUSE UNTIL SENSOR 1 LOW
300 SET FLAG 2 HIGH
310 COMMENT .. Step block into gauge
320 INCREMENT CONVEYOR 0 64 mm SPEED 1
330 CLOSE GAUGE 0
340 WAIT FOR 3 SECOND(S)
350 COMMENT ..... Gauge block width
360 IF GAUGE 0 IS BETWEEN 5 AND 90 THEN SET FLAG 2 LOW
370 OPEN GAUGE 0
380 WAIT FOR 3 SECOND(S)
390 START CONVEYOR 0
400 GOTO LINE 290
410 COMMENT .....
420 COMMENT ..... Serpent/Mill control
430 COMMENT .....
440 PAUSE UNTIL SENSOR 2 LOW
450 COMMENT ..... Decide which Mill program
460 SET FLAG 3 HIGH
470 IF FLAG 2 LOW THEN SET FLAG 3 LOW
480 IF FLAG 2 LOW THEN START MILL 1 PROGRAM 1
490 IF FLAG 2 HIGH THEN START MILL 1 PROGRAM 2
500 PAUSE UNTIL MILL 1 FLAG PAUSED HIGH
510 COMMENT ..... Place block in vice
520 MOVE ROBOT 1 TO POSITION 1
530 WAIT FOR 1 SECOND(S)
540 ROBOT 1 GRIPPER CLOSED
550 WAIT FOR 1 SECOND(S)
560 MOVE ROBOT 1 TO POSITION 2
570 WAIT FOR 1 SECOND(S)
580 SET FLAG 1 LOW
590 ROBOT 1 GRIPPER PROGRAMMED
600 MOVE ROBOT 1 TO POSITION 3
610 PAUSE UNTIL ROBOT 1 IS IN POSITION
620 MOVE ROBOT 1 TO POSITION 4
630 WAIT-FOR 1 SECOND(S)
640 ROBOT 1 CRIPPER OPEN
650 MOVE ROBOT 1 TO POSITION 5
660 WAIT FOR 1 SECOND(S)
670 COMMENT .. Mill block
680 CONTINUE MILL 1 PROGRAM
690 PAUSE UNTIL MILL 1 FLAG PAUSED HIGH
700 MOVE ROBOT 1 TO POSITION 6
710 CONTINUE MILL 1 PROGRAM
720 PAUSE UNTIL ROBOT i IS IN POSITION
730 COMMENT ..... Remove block from vice
740 PAUSE UNTIL MILL 1 FLAG FINISHED HIGH
```

```
750 IF FLAG 3 HIGH THEN GOTO LINE 950
760 COMMENT ..... Block is small
770 MOVE ROBOT 1 TO POSITION 7
780 PAUSE UNTIL ROBOT 1 IS IN POSITION
790 MOVE ROBOT 1 TO POSITION 8
800 WAIT FOR 1 SECOND(S)
810 ROBOT 1 GRIPPER CLOSED
820 WAIT FOR 1 SECOND(S)
830 MOVE ROBOT 1 TO POSITION 9
840 WAIT FOR 1 SECOND(S)
850 FOR A = 10 TO 11
860 MOVE ROBOT 1 TO POSITION A
870 PAUSE UNTIL ROBOT 1 IS IN POSITION
880 NEXT A
890 ROBOT 1 GRIPPER OPEN
900 WAIT FOR 1 SECOND(S)
910 MOVE ROBOT 1 TO POSITION 0
920 PAUSE UNTIL ROBOT 1 IS IN POSITION
930 GOTO LINE 410
940 COMMENT ..... Block is large
950 MOVE ROBOT 1 TO POSITION 7
960 PAUSE UNTIL ROBOT 1 IS IN POSITION
970 MOVE ROBOT 7 TO POSITION 8
980 WAIT FOR 1 SECOND(S)
990 ROBOT 1 GRIPPER CLOSED
1000 WAIT FOR 1 SECOND(S)
1010 MOVE ROBOT 1 TO POSITION 9
1020 WAIT FOR 1 SECOND(S)
1030 FOR A = 70 TO 12
1040 MOVE ROBOT 7 TO POSITION A
1050 PAUSE UNTIL ROBOT 1 IS IN POSITION 1060 NEXT A
1070 ROBOT 1 GRIPPER OPEN
1080 WAIT FOR 1 SECOND(S)
1090 MOVE ROBOT 1 TO POSITION 0
1100 PAUSE UNTIL ROBOT 1 IS IN POSITION
1110 GOTO LINE 410
1120 COMMENT .....
1130 COMMENT ..... Mentor control
1140 COMMENT .....
1150 FOR C = 0 TO 7
1160 PAUSE UNTIL FLAG 1 LOW
1170 SET FLAG 1 HIGH
1180 COMMENT ..... Pick up block
1190 WAIT FOR 1 SECOND(S)
1200 MOVE ROBOT 2 TO POSITION 1
1210 PAUSE UNTIL ROBOT 2 IS IN POSITION
1220 ROBOT 2 GRIPPER CLOSED
1230 WAIT FOR 1 SECOND(S)
1240 COMMENT ..... Check for block in jaws
1250 IF ROBOT 2 AXIS 5 IS BETWEEN 0 AND 35 THEN
    GOTO LINE 1310
1260 ROBOT 2 GRIPPER OPEN
1270 WAIT FOR 1 SECOND(S)
1280 SET FLAG 7 LOW
1290 GOTO LINE 1440
1300 COMMENT ..... Place block on conveyor
1310 FOR B = 2 TO 4
1320 MOVE ROBOT 2 TO POSITION B
1330 PAUSE UNTIL ROBOT 2 IS IN POSITION
1340 NEXT B
1350 ROBOT 2 GRIPPER OPEN
1360 WAIT FOR 1 SECOND(S)
1370 START CONVEYOR 0
1380 WAIT FOR 1 SECOND(S)
1390 COMMENT ..... Prepare for next block
1400 MOVE ROBOT 2 TO POSITION 5
1410 PAUSE UNTIL ROBOT 2 IS IN POSITION
1420 MOVE ROBOT 2 TO POSITION 1
1430 PAUSE UNTIL ROBOT 2 IS IN POSITION
1440 INCREMENT INDEX TABLE 0 45 degrees SPEED 1
1450 NEXT C
1460 COMMENT ..... Re-zero index table
1470 START INDEX TABLE 0
1480 PAUSE UNTIL SENSOR 0 LOW
1490 WAIT FOR 20 SECOND(S)
1500 INCREMENT INDEX TABLE 0 13 degrees SPEED 1
1510 GOTO LINE 1150
```

Értelmezzük a fenti program utasításait és írjuk az utasítások mellé blokkonkénti valós tartalmukat .

### 22.3 A GYAKORLAT VÉGREHAJTÁSA

- a. A gyakorlati útmutatóban leírtak elsajátításának ellenőrzése írásban.
- b. A BME-GTT mintarendszerének megtekintése, működésének elemzése.
- c. A gyártórendszer szimulátor (SICELL) mintaprogramjának elemzése és a mintaprogram animációjának megtekintése.
- d. Gyártórendszer modell (SPEC) megtekintése, működésének elemzése.
- e. Jegyzőkönyv készítése a gyakorlaton szerzett ismeretek alapján.

