

**Jancskárné Anweiler Ildikó**

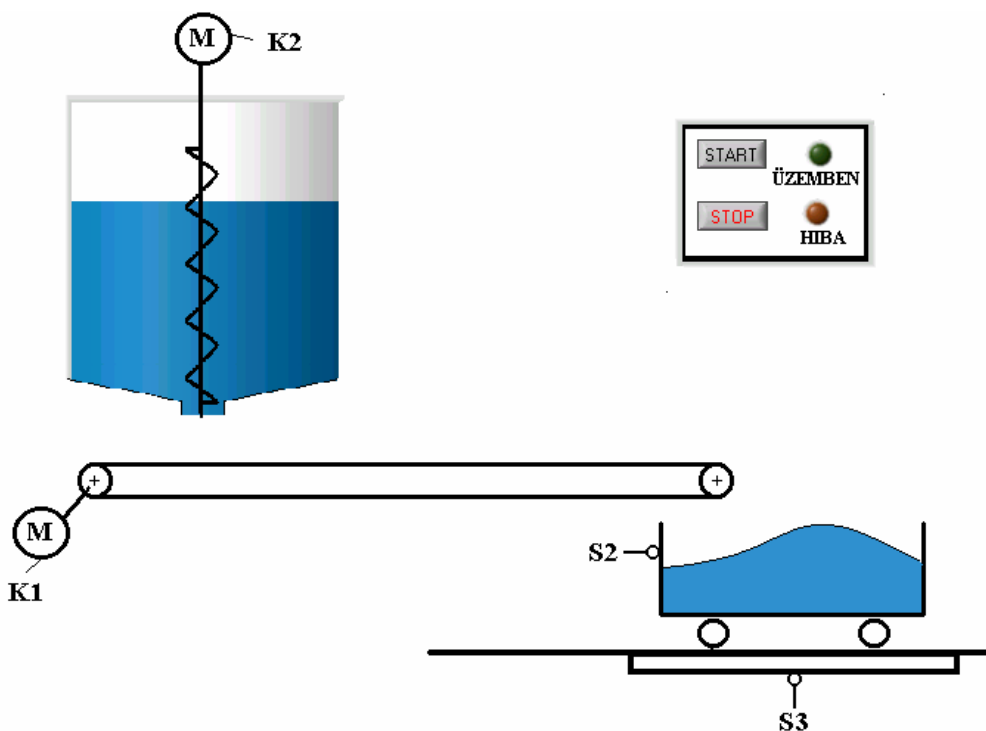
**PLC programozás II. rész**

**KÉZIRAT!**

## Követővezérlési feladatok megoldása állapotgráf segítségével

### Vagontöltő berendezés

A vagon a szilárdanyag tárolóból (siló), adagolócsiga és szállítószalag segítségével töltik fel. Az adagolást a **START** gomb megnyomásával engedélyezik. A **START** jel csak akkor hatásos, ha a vagon töltési helyzetben van (**S2** jelez). Ekkor az adagolandó anyag feltorlódásának elkerülése érdekében először a szállítószalagot kell elindítani, és **3s**-ig üresen jártni. Az idő letelte után bekapcsolható az adagolócsiga motorja is. Ha megtelt a vagon, vagy a vagon elmozdult a töltési pozíciójából, vagy megnyomták a **STOP** gombot, az adagolócsigát azonnal le kell állítani. Ekkor a szállítószalag még **5s**-ig bekapcsolva marad, hogy teljesen leürüljön. Újabb adagolást a **START** gomb ismételt benyomásával lehet elindítani.



1. ábra Vagontöltés

### Összerendelési táblázat

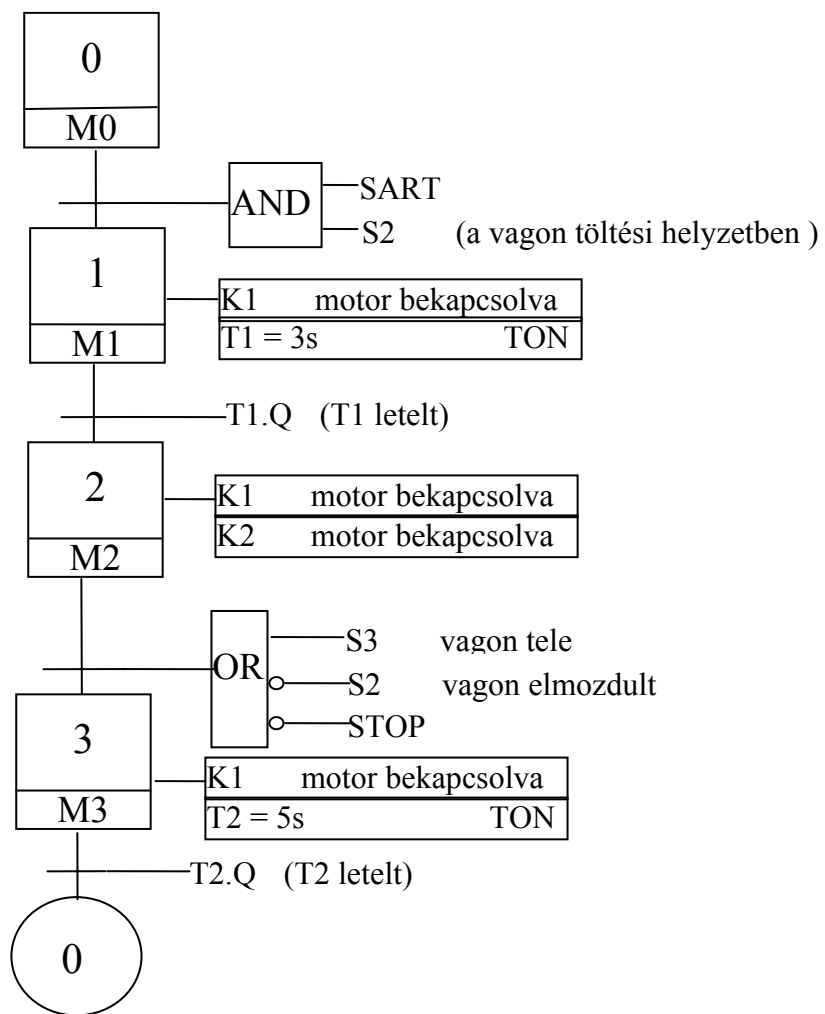
Bemenetek	Jel	Logikai összerendelés	Cím
START nyomógomb	START	benyomva: START=1	I0.0
STOP nyomógomb	STOP	benyomva: STOP=0	I0.1
Rámpaérzékelő	S2	a vagon töltési pozícióban: S2=1	I0.2
Súlyérzékelő	S3	a vagon tele: S3=1	I0.3
Kimenetek			
Szállítószalag motor	K1	bekapcsolva: K1=1	Q0.0
Adagolócsiga motor	K2	bekapcsolva: K2=1	Q0.1

## A vezérlés állapotai

- **Alapállapot (M0):** semmit sem működik, a vezérlés a **START** jelre vár
- **(M1):** Adott felfutási ideig csak a szállítószalag motorja van bekapcsolva
- **(M2):** Mindkét kimenetet (szállítószalag és adagolócsiga) működteti a vezérlés
- **(M3):** Adott leállítási ideig csak a szállítószalag működik

Az állapotok közötti kapcsolatokat, egyik állapotból a másikba történő átlépés feltételeit állapotgráf segítségével írjuk le.

## Állapotgráf

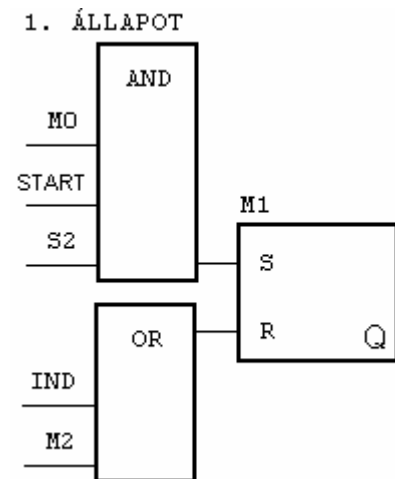
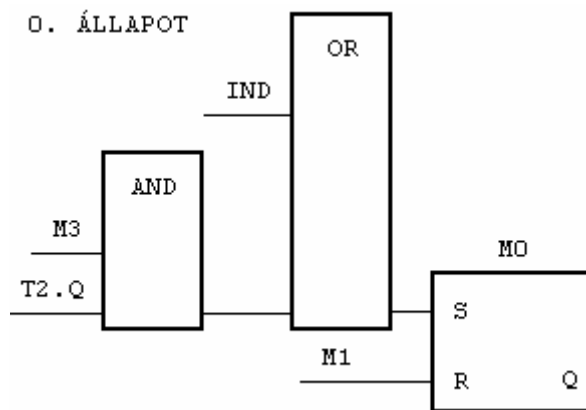


## Az állapotgráf funkciótervbe történő átírásának szabályai

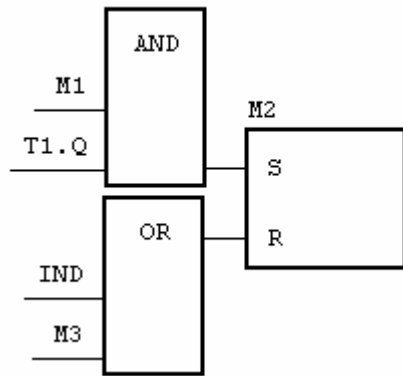
- Minden állapothoz hozzárendelünk egy RS-tárolót.
- Ha a tárolókat nem-remanens memóriaterületre címezzük, elegendő a 0. állapotot (**M0**) az ún. indító impulzussal beállítani, a többi tároló értéke az újraindításkor úgy is törlődik. Ha remanens memóriaterületen tároljuk az állapotokat, az indító impulzus segítségével az összes állapot tárolóját (kivéve a 0. állapot) törölni („resetelni”) kell!

- Gondoskodnunk kell arról, hogy a vezérlésben mindig csak egy állapot legyen aktív. Ezt úgy tudjuk biztosítani, hogy az állapotok tárolóit (set-oldal) az őt megelőző állapot és az átváltás feltétele állítja be és az őt követő állapot törli (reset-oldal).
- Elágazás előtti állapotot az összes őt követő állapot törölheti (vagy-kapcsolat). Ha az elágazásban az átváltások feltételei egyszerre, egy időpillanatban teljesülhetnek, az ágak között prioritási sorrendet kell megállapítanunk. A magasabb prioritású állapot reteszeli az elágazásban lévő nála alacsonyabb prioritású állapotokat.
- Ha két állapot hurokba kerül, a hurokban lévő állapotokat az őket követő állapot és az átváltás feltétele együttesen törlik.
- A kimenetek azon állapotok VAGY-kapcsolataként írhatók fel, amelyekhez hozzárendeltük őket. Az esetleges plusz feltételeket (retesz feltételeket) ÉS feltételként hozzáillesztjük.

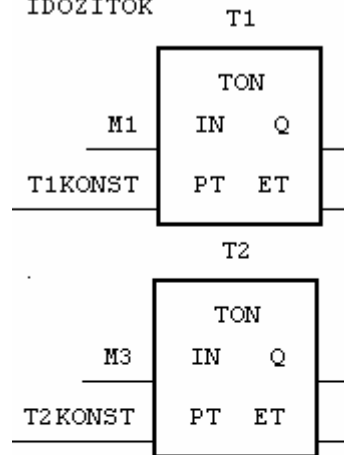
**Funkcióterv**



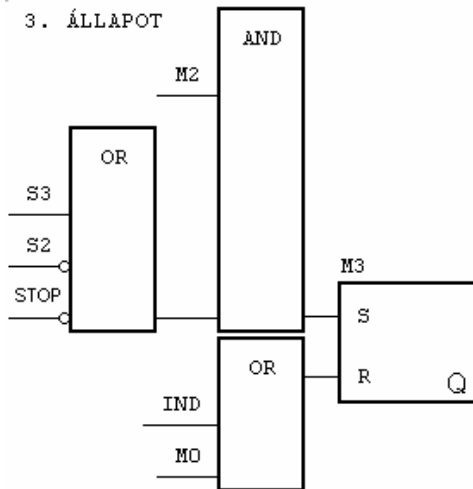
## 2. ÁLLAPOT



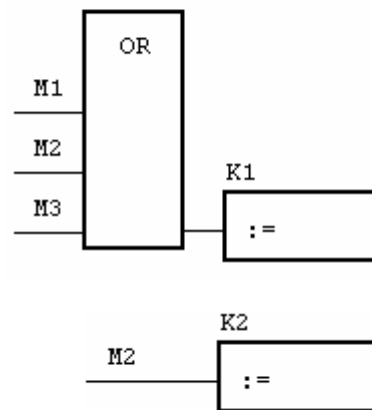
## IDŐZÍTŐK



## 3. ÁLLAPOT



## KIMENETEK



## Utasításlista

PROGRAM PRVAGTLT

VAR

```

START AT %I0.0.0.0.0:   BOOL;
STOP AT %I0.0.0.0.1:   BOOL;
RAMPA AT %I0.0.0.0.2:  BOOL;
SULY AT %I0.0.0.0.3:   BOOL;
MOTOR1 AT %Q0.0.0.0.0: BOOL;
MOTOR2 AT %Q0.0.0.0.1: BOOL;
FGVBL:   VAGTOLT;
IMPULZUS:  BOOL;

```

END\_VAR

(\*INDÍTÓ IMPULZUS\*)

```

LD  ISA
PLC_Message
ST  IMPULZUS

```

(\*FÜGGVÉNYBLOKK HÍVÁSA\*)

```
CAL FGVBL(IND:=IMPULZUS, START:=START, STOP:=STOP, S2:=RAMPA,
S3:=SULY)
```

```
(*KIMENETEK TÁROLÁSA*)
```

```
LD FGVBL.K1
ST MOTOR1
LD FGVBL.K2
ST MOTOR2
END_PROGRAM
```

Az állapotgráfot függvényblokkban írtuk meg.

```
FUNCTION_BLOCK VAGTOLT
```

```
VAR_INPUT
    START:BOOL;
    STOP: BOOL;
    S2:  BOOL;
    S3:  BOOL;
    IND:  BOOL;
END_VAR
```

```
VAR_OUTPUT
    K1:  BOOL;
    K2:  BOOL;
END_VAR
```

```
VAR
    M0:  BOOL;
    M1:  BOOL;
    M2:  BOOL;
    M3:  BOOL;
    T1:  TON;
    T2:  TON;
END_VAR
```

```
VAR CONSTANT
    T1KONST:  TIME := T#3s;
    T2KONST:  TIME := t#5s;
END_VAR
```

```
(*0. ÁLLAPOT*)
LD  IND
OR(  M3
AND T2.Q
)
S   M0
LD  M1
R   M0
```

```
(*1. ÁLLAPOT*)
LD  M0
AND START
AND S2
S   M1
LD  IND
OR  M2
R   M1
```

(\*2. ÁLLAPOT\*)

```
LD M1
AND T1.Q
S M2
LD IND
OR M3
R M2
```

(\*3. ÁLLAPOT\*)

```
LD M2
AND S3
ORN S2
ORN STOP
S M3
LD IND
OR M0
R M3
```

(\*IDŐZÍTŐK\*)

```
CAL T1(IN := M1,PT :=T1KONST)
CAL T2(IN := M3,PT :=T2KONST)
```

(\*KIMENETEK\*)

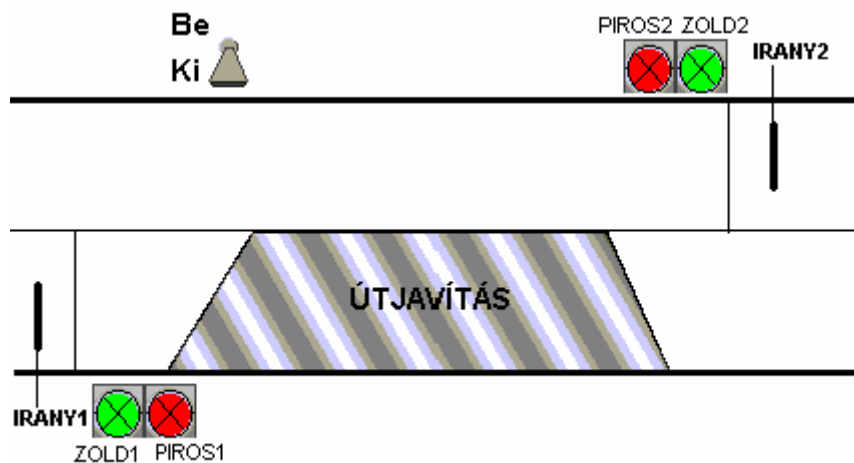
```
LD M1
OR M2
OR M3
ST K1
```

```
LD M2
ST K2
```

END\_FUNCTION\_BLOCK

## Útjavítást jelző lámpa

Útjavítás miatt egy bekötőutat adott útszakaszon egysávosra kell szűkíteni. Mivel napközben igen nagy a forgalom, jelzőlámpákat állítottak fel a szűkítés végpontjain. A vezérlőberendezés bekapcsolásakor mindkét jelzőlámpa pirosat mutat. Ha az egyik irányú érzékelő jelez, a megfelelő lámpát 10s múlva zöldre váltja. A zöld fázist kb. 20s-ig tartani kell, mielőtt a másik érzékelő jelzése mindkét lámpát pirosra váltja. 10s múlva a másik irány lesz zöld. Ha egyik érzékelő sem jelez, a lámpajelzés az előző állapotában marad. A berendezést csak valamely irány zöldfázisa után lehet kikapcsolni.



2. ábra Útlezárás

## Összerendelési táblázat

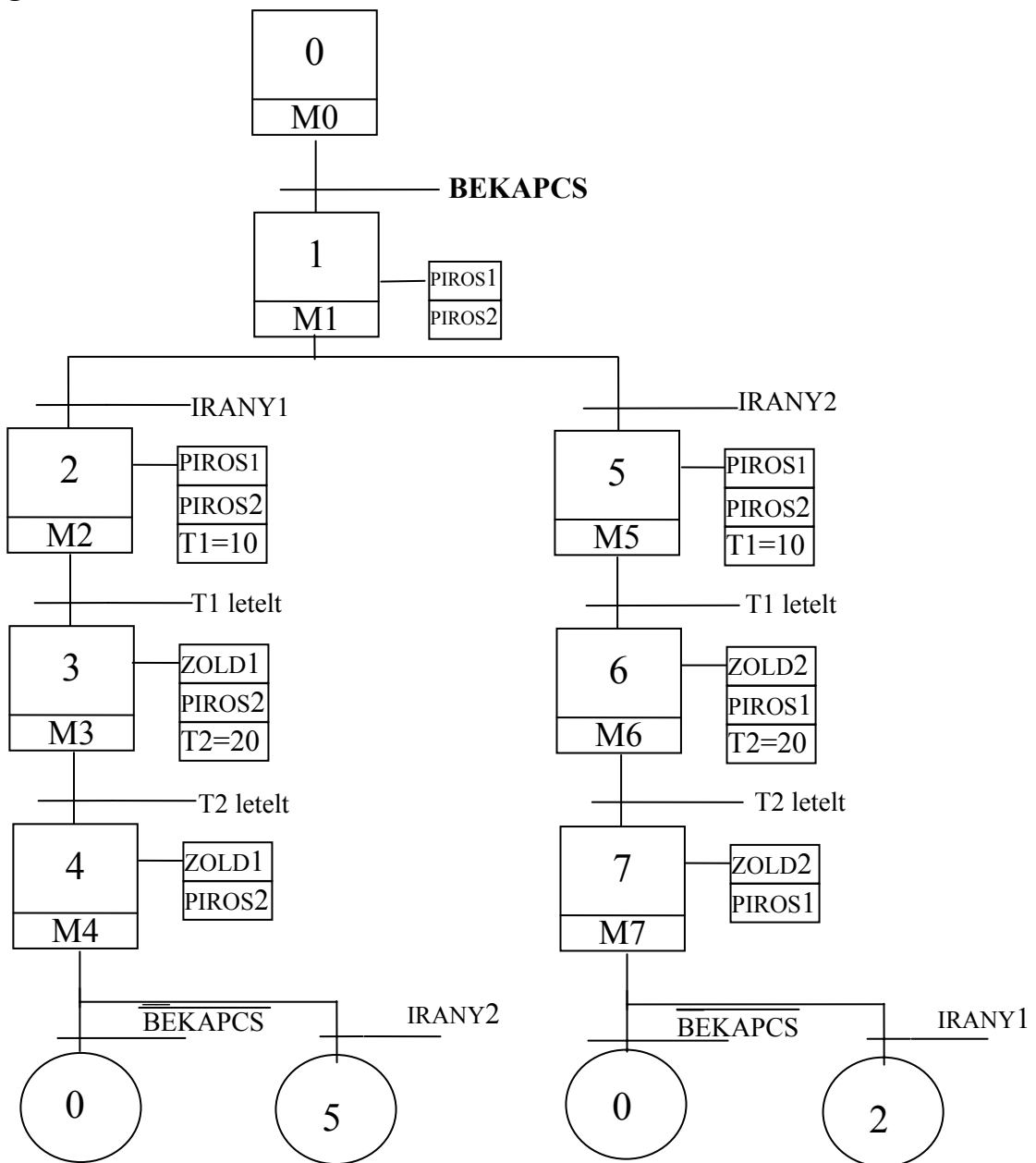
Bemenetek	Jel	Logikai összerendelés	Cím
BE/KI kapcsoló	BEKAPCS	bekapcsolva: BEKAPCS=1	I0.0
1. irány érzékelő	IRANY1	jelez: IRANY1=1	I0.1
2. irány érzékelő	IRANY2	jelez: IRANY2=1	I0.2
<b>Kimenetek</b>			
1. lámpa zöld	ZOLD1	világít: ZOLD1=1	Q0.0
2. lámpa zöld	ZOLD2	világít: ZOLD2=1	Q0.1
1. lámpa piros	PIROS1	világít: PIROS1=1	Q0.2
2. lámpa piros	PIROS2	világít: PIROS2=1	Q0.3

### **A vezérlés állapotai**

1. Alapállapot: semmit sem működtet, a bekapcsolási jelre vár
2. Mindkét lámpa piros
3. Adott ideig mindkét lámpa piros, jármű vár az 1. irányból
4. Adott ideig mindkét lámpa piros, jármű vár a 2. irányból
5. Adott ideig 1. lámpa piros, 2. lámpa zöld
6. Adott ideig 2. lámpa piros, 1. lámpa zöld
7. lámpa piros, 2. lámpa zöld
8. lámpa piros, 1. lámpa zöld

Az állapotok közötti kapcsolatokat, az egyik állapotból a másikba történő átlépés feltételeit állapotgráf segítségével írjuk le.

**Állapotgráf**

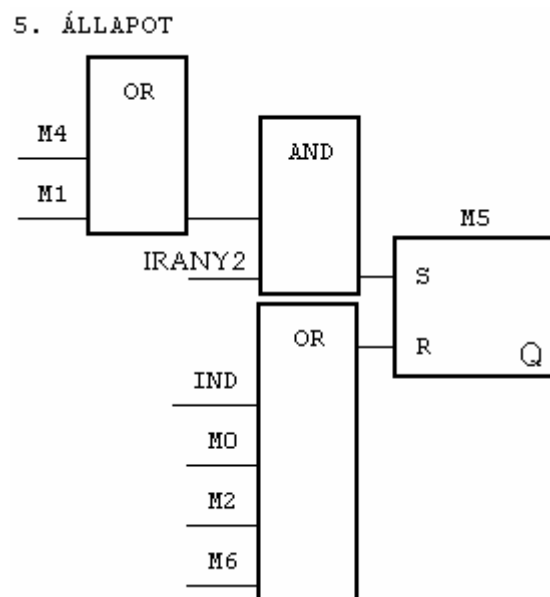
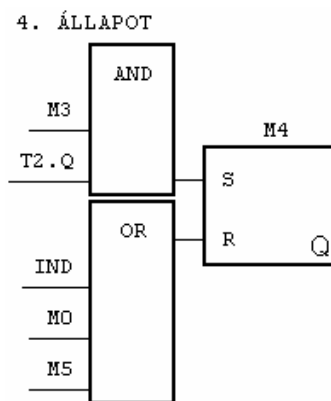
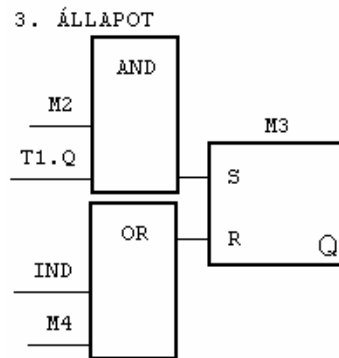
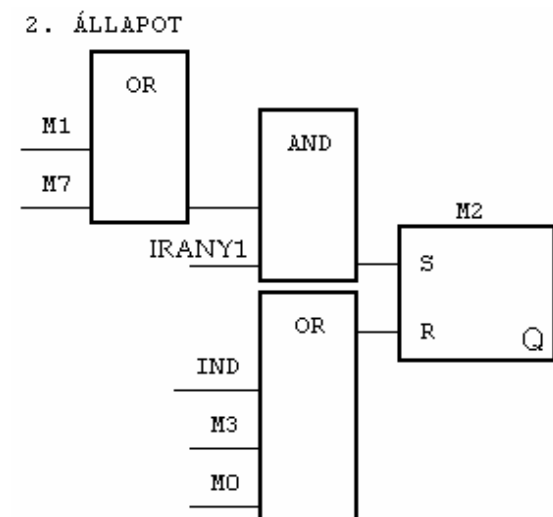
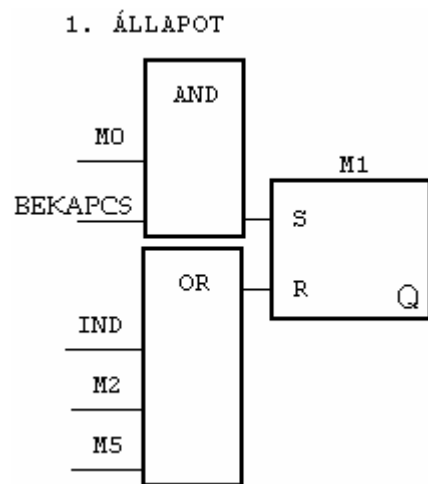
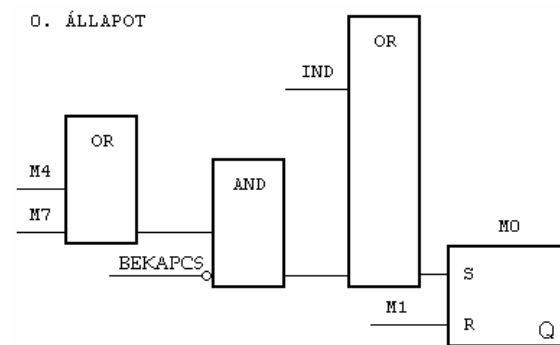


**IRANY1** és **IRANY2** egyidejűleg is jelezhet. (Mindkét irányból érkezhet egyszerre jármű.) Mivel egyszerre csak egy állapot lehet „aktív” -egyszerre csak egy iránynak lehet zöld jelzése-, el kell dönteni, hogy melyik irány élvezzen elsőbbséget. Legyen **IRANY1**-nek elsőbbsége, ekkor **M2** reteszeli **M5**-t. (Az **5. állapotot** a **2. RESET**-eli.)

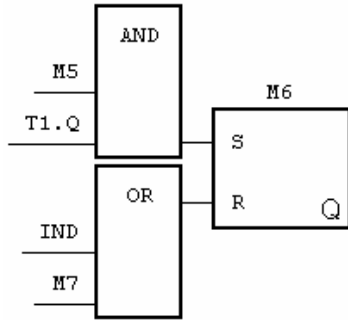
A **4.** és **7. állapot** után ismét kölcsönös reteszelés van. Itt a **0. állapot**nak van előnye. **M0** **M2**-t és **M5**-t is **RESET**-eli.

## Az állapotgráf átírása funkciótervbe illetve utasításlistába

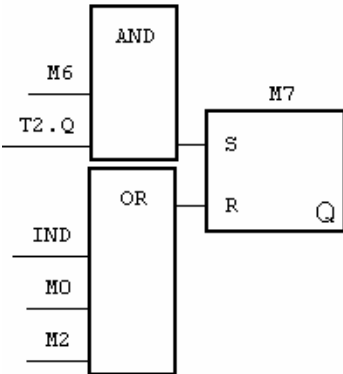
### Funkcióterv



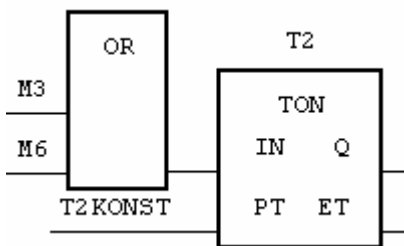
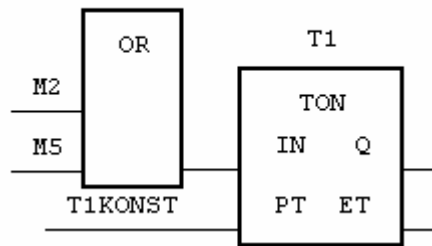
6. ÁLLAPOT



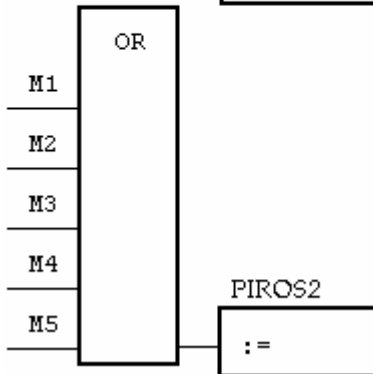
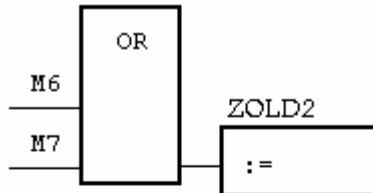
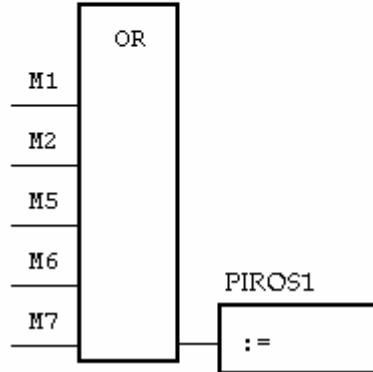
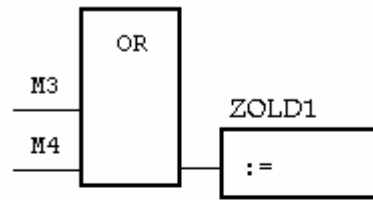
7. ÁLLAPOT



IDŐZÍTŐK



KIMENETEK



## Utasításlista

PROGRAM PRUTLEZAR

VAR\_GLOBAL

```
BEKAPCS AT %I0.0.0.0.0: BOOL;  
IRANY1 AT %I0.0.0.0.1:  BOOL;  
IRANY2 AT %I0.0.0.0.2:  BOOL;  
ZOLD1 AT %Q0.0.0.0.0:  BOOL;  
PIROS1 AT %Q0.0.0.0.1:  BOOL;  
ZOLD2 AT %Q0.0.0.0.2:  BOOL;  
PIROS2 AT %Q0.0.0.0.3:  BOOL;
```

END\_VAR

VAR

```
FGVBL:    UTLEZ1;  
IMPULZUS: BOOL;
```

END\_VAR

(\*INDÍTÓ IMPULZUS\*)

```
LD  ISA  
PLC_Message  
ST  IMPULZUS
```

(\*FÜGGVÉNYBLOKK HÍVÁSA\*)

```
CAL FGVBL(IND:=IMPULZUS)
```

END\_PROGRAM

FUNCTION\_BLOCK UTLEZ1

VAR\_INPUT

```
IND:  BOOL;
```

END\_VAR

VAR

```
M0:  BOOL;  
M1:  BOOL;  
M2:  BOOL;  
M3:  BOOL;  
M4:  BOOL;  
M5:  BOOL;  
M6:  BOOL;  
M7:  BOOL;  
T1:  TON;  
T2:  TON;
```

END\_VAR

VAR constant

```
T1KONST:  TIME := t#10s;
```

```

T2KONST:  TIME := t#20s;
END_VAR

```

```

VAR_EXTERNAL
  BEKAPCS:  BOOL;
  IRANY1:   BOOL;
  IRANY2:   BOOL;
  ZOLD1:    BOOL;
  PIROS1:   BOOL;
  ZOLD2:    BOOL;
  PIROS2:   BOOL;
END_VAR

```

```
(*0. ÁLLAPOT*)
```

```

LD  IND
OR(  M4
OR   M7
ANDNBEKAPCS
)
S   M0
LD  M1
R   M0

```

```
(*1. ÁLLAPOT*)
```

```

LD  M0
AND BEKAPCS
S   M1
LD  IND
OR  M2
OR  M5
R   M1

```

```
(*2. ÁLLAPOT*)
```

```

LD  M1
OR  M7
AND IRANY1
S   M2
LD  IND
OR  M3
OR  M0
R   M2

```

```
(*3. ÁLLAPOT*)
```

```

LD  M2
AND T1.Q
S   M3
LD  IND
OR  M4
R   M3

```

```
(*4. ÁLLAPOT*)
```

```

LD  M3
AND T2.Q
S   M4
LD  IND
OR  M0
OR  M5
R   M4

```

```
(*5. ÁLLAPOT*)
```

```

LD  M4
OR  M1
AND IRANY2
S   M5
LD  IND
OR  M0
OR  M2
OR  M6
R   M5

```

```
(*6. ÁLLAPOT*)
```

```

LD  M5
AND T1.Q
S   M6
LD  IND
OR  M7
R   M6

```

```
(*7. ÁLLAPOT*)
```

```

LD  M6

```

```
AND T2.Q
S M7
LD IND
OR M0
OR M2
R M7
```

(\*IDŐZÍTŐK\*)

```
LD M2
OR M5
ST T1.IN
LD T1KONST
ST T1.PT
CAL T1
```

```
LD M3
OR M6
ST T2.IN
LD T2KONST
ST T2.PT
CAL T2
```

(\*KIMENETEK\*)

```
LD M3
OR M4
ST ZOLD1
```

```
LD M1
OR M2
OR M5
OR M6
OR M7
ST PIROS1
```

```
LD M6
OR M7
ST ZOLD2
```

```
LD M1
OR M2
OR M3
OR M4
OR M5
ST PIROS2
```

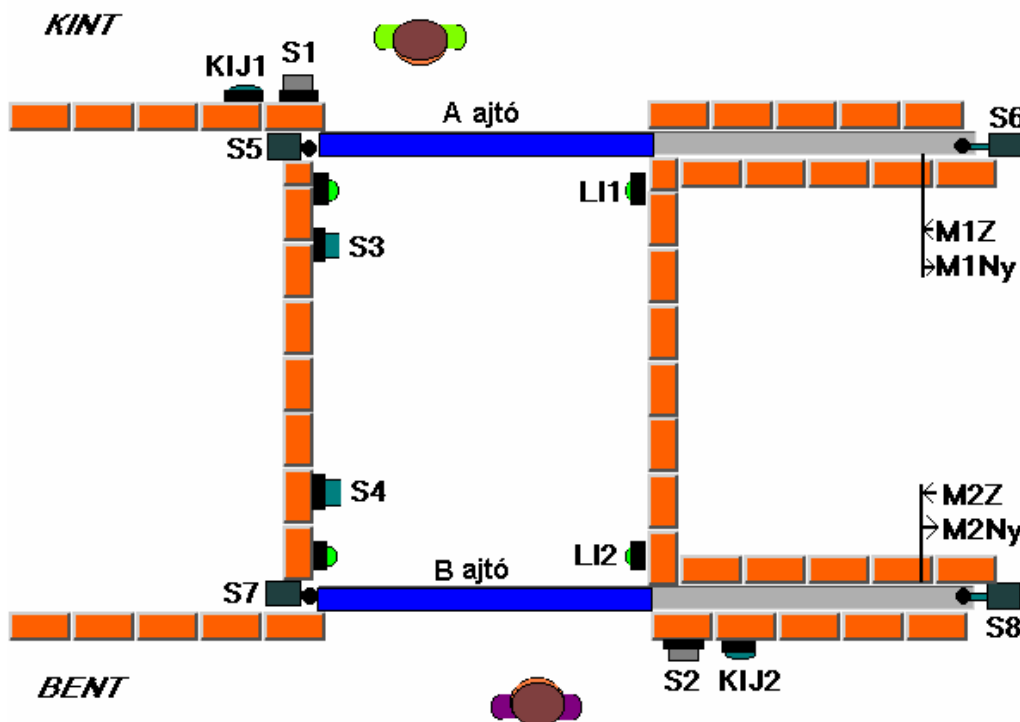
END\_FUNCTION\_BLOCK

### Jelek állapotgráfon kívüli feldolgozása

Előfordul sok esetben, hogy a jelforrás, a jeladó, (pl. nyomógomb) csak rövid ideig szolgáltat jelet a PLC bemenetén. Ha a vezérlést állapotgráffal oldjuk meg, és a vezérlés éppen olyan állapotban van, amelynek követőállapota egy másik vezérlőjel megjelenésére vár, akkor a rövid időre megjelenő jelzést elveszíthetjük. Ennek elkerülésére **szükséges a rövid ideig ható jeleket az állapotgráfon kívül eltárolnunk.** (Ez az ún. **előfeldolgozás.**)

### Zsilipajtók vezérlése

Egy helyiséget por- és szennyeződésmentesen kell tartani, így a bejárathoz kiépítettek egy zsilipkamrát, „A” és „B” tolóajtóval. Egyszerre, egy időben mindig csak az egyik ajtó lehet nyitva. A zsilipen az áthaladást kívülről az **S1** vagy **S2** nyomógombokkal lehet kezdeményezni. Kívülről belülről **S1** gomb megnyomása nyitja az „A” ajtót. Miután „A” kinyílt (**S6** jelez), még 3 s-ig nyitva marad, majd becsukódik. Ha „A” ajtó becsukódott (**S5** jelez), „B” ajtó automatikusan kinyílik, 3 s-ig nyitva marad, majd becsukódik. A másik irányból az áthaladás hasonló módon, fordított sorrendben, **S2** gomb megnyomására „B” majd „A” ajtó egymás után automatikusan nyílik és záródik. A nyomógomb melletti jelzőlámpa mutatja, hogy a vezérlés észrevette a gomb benyomását. Mindkét ajtónál végállás-kapcsolók jelzik az ajtó nyitott ill. zárt helyzetét. Az ajtókhoz tartozik egy-egy optikai érzékelő, ha a fényút megszakad, az ajtó zárását nem lehet megkezdeni, illetve, ha már záródik, azonnal vissza kell nyitni. Hasonlóan zárás közben vissza kell nyitni az ajtókat akkor is, ha az „A” ajtónál **S1** vagy **S3**, a „B” ajtónál **S2** vagy **S4** gombot megnyomják. 3s-os várakozás után a zárás ismét kezdeményezhető. A zsilipben a biztonság miatt elhelyeztek két nyomógombot, (**S3** és **S4**), amelyekkel a hozzájuk tartozó ajtók nyitását szükség esetén kezdeményezni lehet (pl. ha valaki véletlenül bennrekedt a zsilipben, mert úgy lépett be egy éppen nyitott zsilipajtón, hogy előzőleg nem nyomta meg az **S1** vagy **S2** gombot.)

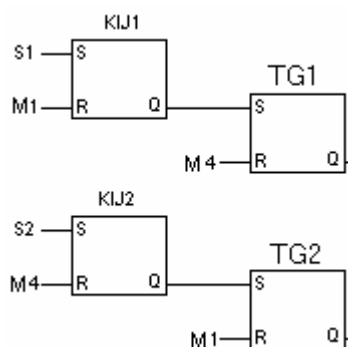


3. ábra Zsilipajtó

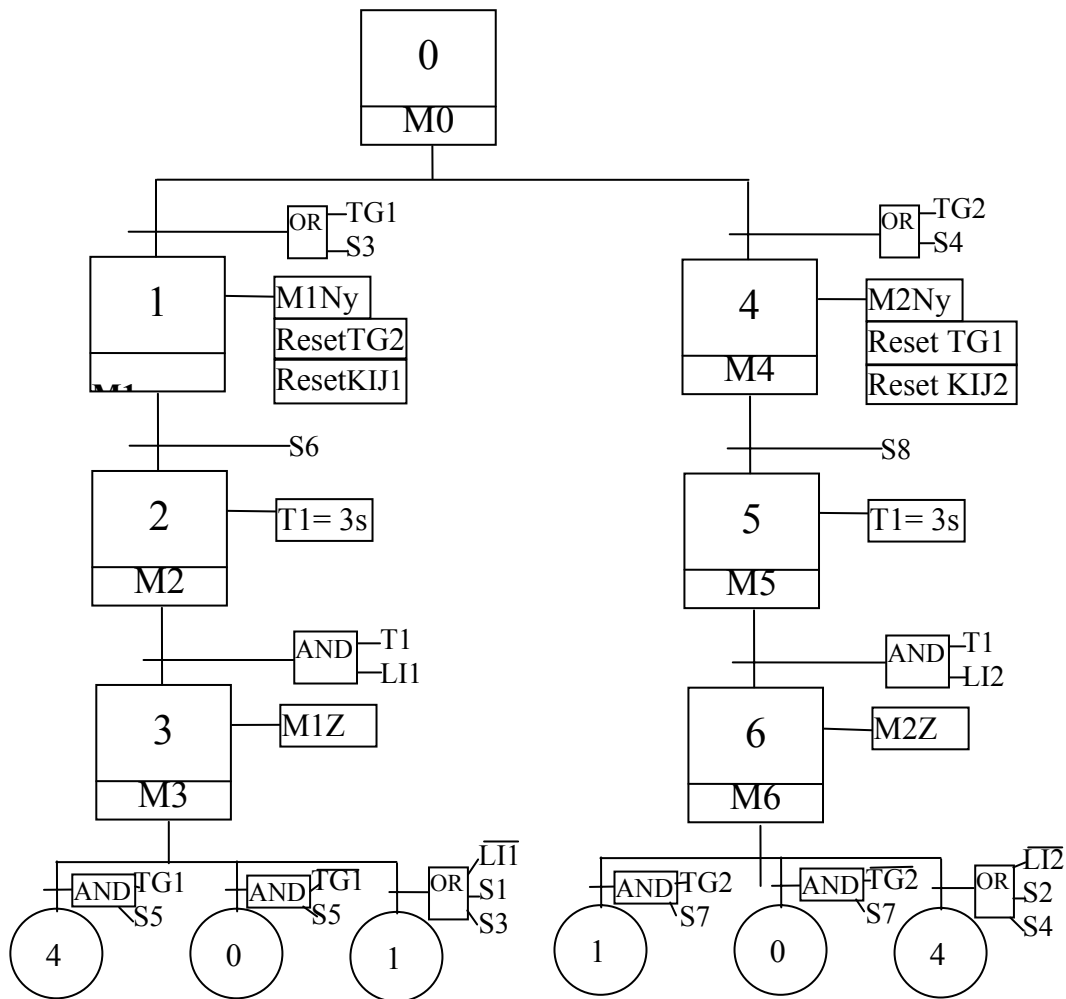
## Összerendelési táblázat

Bemenetek	Jel	Logikai összerendelés	Cím
A ajtó külső nyomógomb	S1	<b>benyomva:</b> S1=1	I0.0
B ajtó külső nyomógomb	S2	benyomva: S2=1	I0.1
A ajtó belső nyomógomb	S3	benyomva: S3=1	I0.2
B ajtó belső nyomógomb	S4	benyomva: S4=1	I0.3
A ajtó csukva	S5	jelez, ha: S5=1	I0.4
A ajtó nyitva	S6	jelez, ha: S6=1	I0.5
B ajtó csukva	S7	jelez, ha: S7=1	I0.6
B ajtó nyitva	S8	jelez, ha: S8=1	I0.7
A ajtó optikai érzékelő	LI1	ha a fényút megszakad: LI1=0	I1.0
B ajtó optikai érzékelő	LI2	ha a fényút megszakad: LI2=0	I1.1
Kimenetek			
A ajtó elektromotor nyitás irányba	M1Ny	működtetve: M1Ny=1	Q0.0
A ajtó elektromotor zárás irányba	M1Z	működtetve: M1Z=1	Q0.1
B ajtó elektromotor nyitás irányba	M2Ny	működtetve: M2Ny=1	Q0.2
B ajtó elektromotor zárás irányba	M2Z	működtetve: M2Z=1	Q0.3
A oldali visszajelző lámpa	KIJ1	világít, ha: KIJ1=1	Q0.4
B oldali visszajelző lámpa	KIJ2	világít, ha: KIJ2=1	Q0.5

### Rövid ideig ható jelek feldolgozása az állapotgráfon kívül



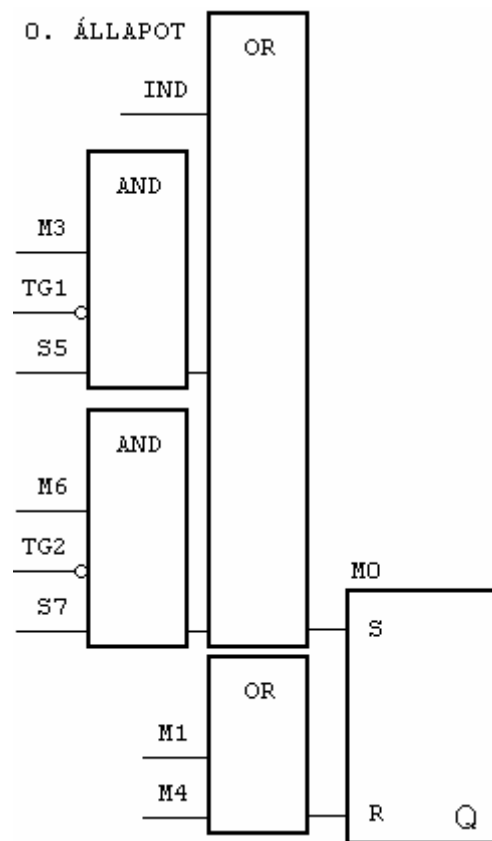
**Az állapotgráf**

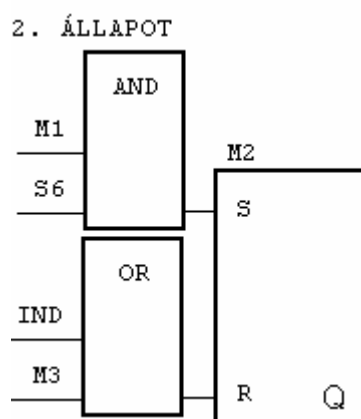
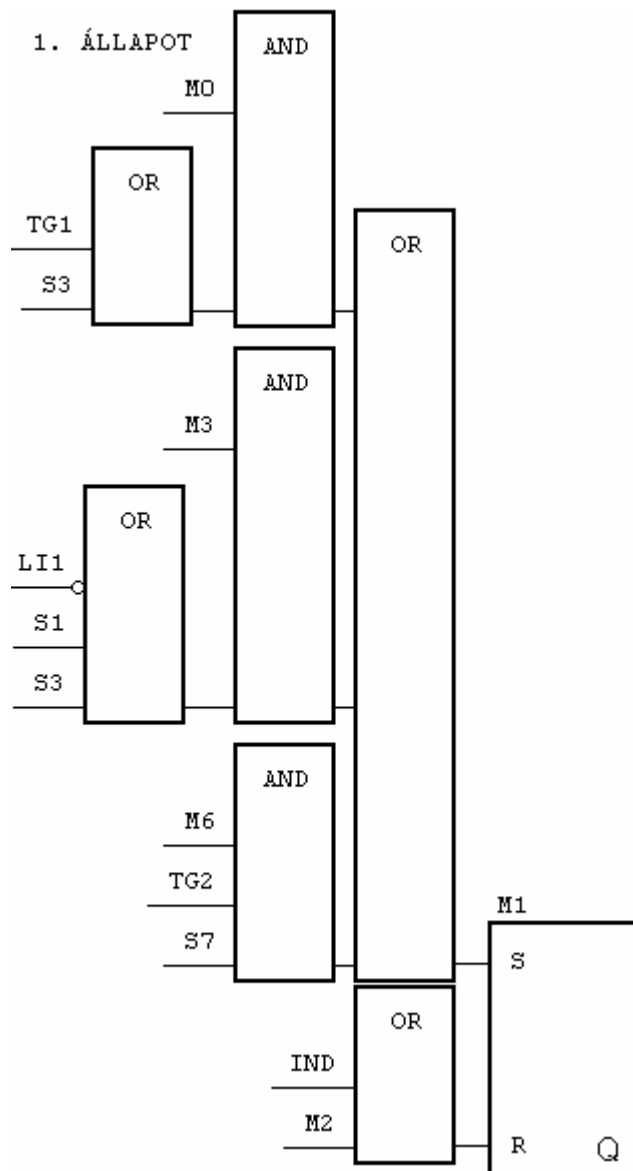


**Az állapotgráf átírása funkciótervbe illetve utasításlistába**

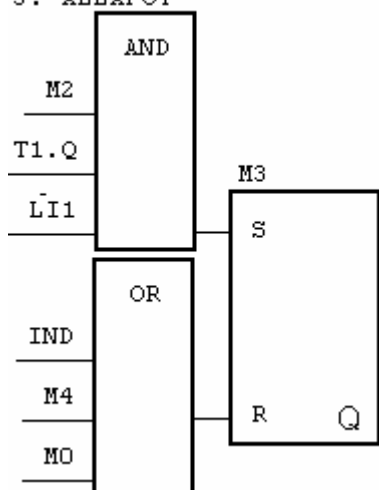
Az állapotgráf elágazásokat tartalmaz. Az elágazásokra vonatkozó szabályok figyelembevételével, rövid elemzés után észrevehetjük, hogy elegendő **M1 - M4** (vagyis a belépés – kilépés) közötti elsőbbséget meghatározni, ezáltal mindhárom elágazás problémája megoldódik. Legyen pl. a belépésnek elsőbbsége: ekkor **M1** reteszeli **M4**-et.

## Funkcióterv

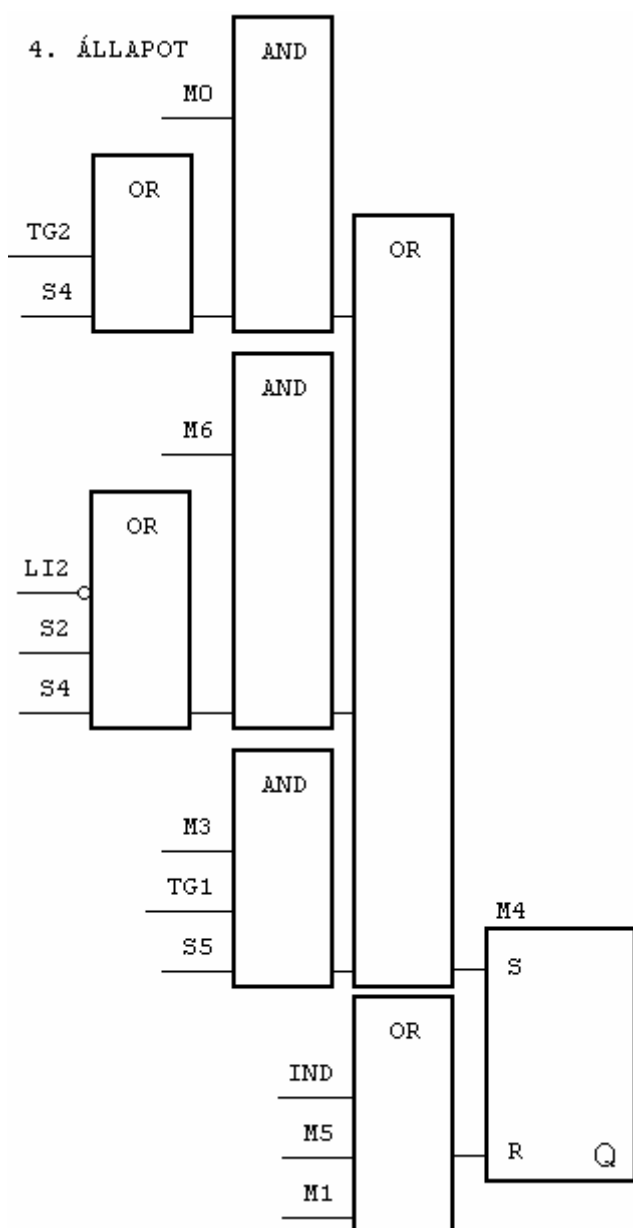




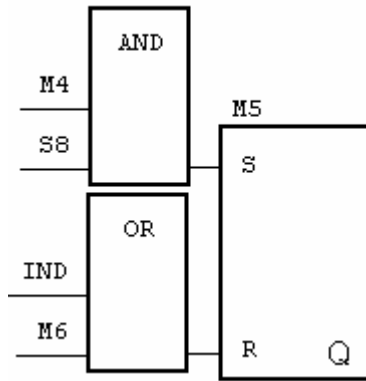
### 3. ÁLLAPOT



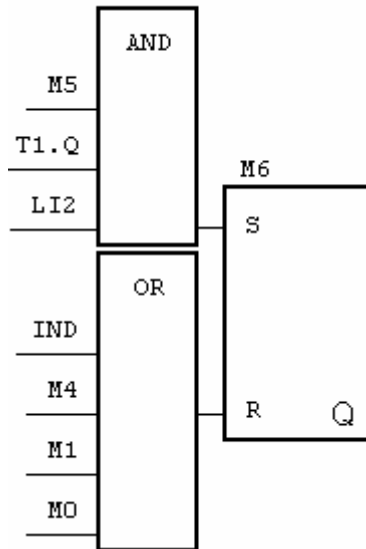
### 4. ÁLLAPOT



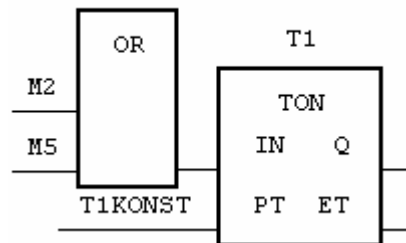
5. ÁLLAPOT



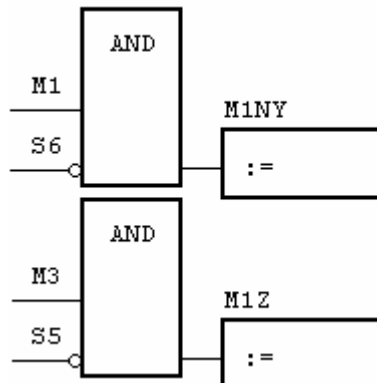
6. ÁLLAPOT

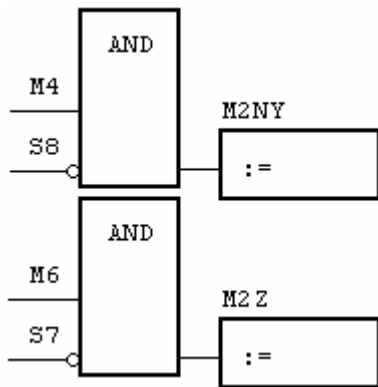


IDŐZÍTŐ



KIMENETEK





### A függvényblokk listája

FUNCTION\_BLOCK ZSLPALL

VAR\_INPUT

S1: BOOL;  
 S2: BOOL;  
 S3: BOOL;  
 S4: BOOL;  
 S5: BOOL;  
 S6: BOOL;  
 S7: BOOL;  
 S8: BOOL;  
 LI1: BOOL;  
 LI2: BOOL;  
 TG1: BOOL;  
 TG2: BOOL;  
 IND: BOOL;

END\_VAR

VAR\_OUTPUT

M1NY: BOOL;  
 M1Z: BOOL;  
 M2NY: BOOL;  
 M2Z: BOOL;  
 M1: BOOL;  
 M4: BOOL;

END\_VAR

VAR

M0: BOOL;  
 M2: BOOL;  
 M3: BOOL;  
 M5: BOOL;  
 M6: BOOL;  
 T1: TON;

END\_VAR

VAR constant

T1KONST: TIME := t#3s;

END\_VAR

(\*0. ÁLLAPOT\*)

```
LD  IND
OR(  M3
ANDNTG1
AND  S5
)
OR(  M6
ANDNTG2
AND  S7
)
S    M0
LD  M1
OR  M4
R   M0
```

(\*1. ÁLLAPOT\*)

```
LD  M0
AND( TG1
OR  S3
)
OR(  M3
AND( LI1
NOT
OR  S1
OR  S3
)
)
OR(  M6
AND  TG2
AND  S7
)
)
S    M1
LD  IND
OR  M2
R   M1
```

(\*2. ÁLLAPOT\*)

```
LD  M1
AND  S6
S    M2
LD  IND
OR  M3
R   M2
```

(\*3. ÁLLAPOT\*)

```
LD  M2
AND  T1.Q
AND  LI1
S    M3
LD  IND
OR  M4
OR  M0
R   M3
```

(\*4. ÁLLAPOT\*)

```
LD  M0
AND( TG2
OR  S4
)
OR(  M6
AND( LI2
NOT
OR  S2
OR  S4
)
)
OR(  M3
AND  TG1
AND  S5
)
)
S    M4
LD  IND
OR  M5
OR  M1
R   M4
```

(\*5. ÁLLAPOT\*)

```
LD  M4
AND  S8
S    M5
LD  IND
OR  M6
R   M5
```

(\*6. ÁLLAPOT\*)

```
LD  M5
AND  T1.Q
```

```

AND LI2
S M6
LD IND
OR M4
OR M1
OR M0
R M6

```

(\*IDŐZÍTŐ\*)

```

LD M2
OR M5
ST T1.IN
LD T1.KONST
ST T1.PT
CAL T1

```

(\*KIMENETEK\*)

```

LD M1
ANDNS6
ST M1NY

```

```

LD M3
ANDNS5
ST M1Z

```

```

LD M4
ANDNS8
ST M2NY

```

```

LD M6
ANDNS7
ST M2Z

```

END\_FUNCTION\_BLOCK

### A főprogram listája

PROGRAM przsilip

VAR

```

A_KULSO_NYG AT %I0.0.0.0.0: BOOL;
B_KULSO_NYG AT %I0.0.0.0.1: BOOL;
A_BELSO_NYG AT %I0.0.0.0.2: BOOL;
B_BELSO_NYG AT %I0.0.0.0.3: BOOL;
A_ZARVA AT %I0.0.0.0.4: BOOL;
A_NYITVA AT %I0.0.0.0.5: BOOL;
B_ZARVA AT %I0.0.0.0.6: BOOL;
B_NYITVA AT %I0.0.0.0.7: BOOL;
A_OPT AT %I0.0.0.1.0: BOOL;
B_OPT AT %I0.0.0.1.1: BOOL;
A_NYITAS AT %Q0.0.0.0.0: BOOL;
A_ZARAS AT %Q0.0.0.0.1: BOOL;
B_NYITAS AT %Q0.0.0.0.2: BOOL;
B_ZARAS AT %Q0.0.0.0.3: BOOL;
KIJ1 AT %Q0.0.0.0.4: BOOL;
KIJ2 AT %Q0.0.0.0.5: BOOL;
TG1: BOOL;
TG2: BOOL;
IMPULZUS: BOOL;
FGVBL: ZSLPALL;

```

END\_VAR

(\*INDÍTÓ IMPULZUS\*)

```
LD  ISA
PLC_Message
ST  IMPULZUS
```

(\*JELFELDOLGOZÁS AZ ÁLLAPOTGRÁFON KÍVÜL\*)

```
LD  A_KULSO_NYG
S   KIJ1
LD  FGVBL.M1
R   KIJ1
LD  KIJ1
S   TG1
LD  FGVBL.M4
R   TG1
```

```
LD  B_KULSO_NYG
S   KIJ2
LD  FGVBL.M4
R   KIJ2
LD  KIJ2
S   TG2
LD  FGVBL.M1
R   TG2
```

(\*FÜGGVÉNYBLOKK HÍVÁSA\*)

```
CAL  FGVBL(IND:=IMPULZUS,
      S1:=A_KULSO_NYG,
      S2:=B_KULSO_NYG,
      S3:=A_BELSO_NYG,
      S4:=B_BELSO_NYG,
      S5:=A_ZARVA,
      S6:=A_NYITVA,
      S7:=B_ZARVA,
      S8:=B_NYITVA,
      LI1:=A_OPT,
      LI2:=B_OPT,
      TG1:=TG1,
      TG2:=TG2
    )
```

(\* KIMENETEK \*)

```
LD  FGVBL.M1NY
ST  A_NYITAS
LD  FGVBL.M2NY
ST  B_NYITAS
LD  FGVBL.M1Z
ST  A_ZARAS
LD  FGVBL.M2Z
ST  B_ZARAS
END_PROGRAM
```

### Gyakorló feladat: utasításlista elemzése IV.

Írja át az alábbi utasításlistát funkciótervbe, majd próbálja meg felrajzolni az állapotgráfot.

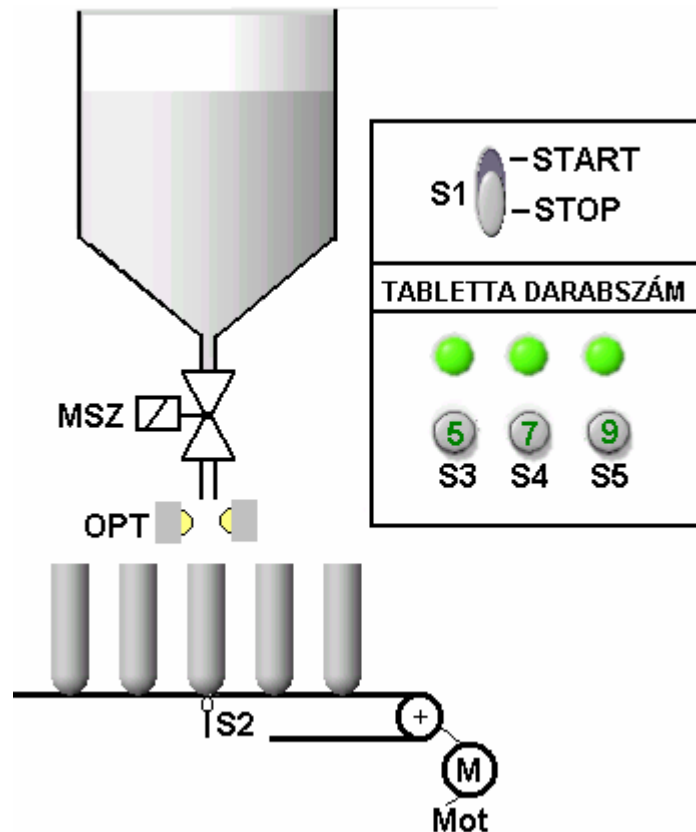
PROGRAM UTLELEM		S	M2
VAR		LD	M3
M60 : BOOL ;		R	M2
M61 : BOOL ;		LD	M2
M0 : BOOL ;		AND	I3
M1 : BOOL ;		S	M3
M2 : BOOL ;		LD	M0
M3 : BOOL ;		OR	M4
M4 : BOOL ;		R	M3
M5 : BOOL ;		LD	M1
M6 : BOOL ;		ANDN	I2
END_VAR		AND	I4
VAR		OR(	M3
I1 AT %I0.1 : BOOL ;		AND	I4
I2 AT %I0.2 : BOOL ;		)	
I3 AT %I0.3 : BOOL ;		S	M4
I4 AT %I0.4 : BOOL ;		LD	M5
Q1 AT %Q0.1 : BOOL ;		OR	M0
Q2 AT %Q0.2 : BOOL ;		R	M4
Q3 AT %Q0.3 : BOOL ;		LD	M4
END_VAR		AND	I3
		S	M5
LDN	M60	LD	M0
ST	M61	OR	M6
S	M60	R	M5
		LD	M1
LD	M61	AND	I1
OR(	I1	ANDN	I2
NOT		ANDN	I4
AND(	M3	OR(	M5
OR	M5	ANDN	I2
OR	M6	)	
)		S	M6
)		LD	M0
S	M0	R	M6
LD	M1		
R	M0	LD	M1
		OR	M2
LD	M0	OR	M3
AND	I1	OR	M4
S	M1	OR	M6
LD	M2	ST	Q1
OR	M4	LD	M2
OR	M6	OR	M3
R	M1	ST	Q2
		LD	M4
LD	M1	OR	M5
AND	I2	ST	Q3
ANDN	I4	END_PROGRAM	

### Komplex vezérlési feladat számlálóval

A számlálókat használó vezérlési feladatokban is szisztematikusabb feladatmegoldást jelent az állapotgráf bevezetése.

### Tablettaadagoló berendezés vezérlése

Egy elótárolóból bizonyos számú tablettát kell kémcsövekbe adagolni. A bekapcsolás után a kezelő a megfelelő nyomógomb működtetésével kiválasztja a kívánt darabszámot. A szalagmotor bekapcsol, és a kémcsövet töltési pozícióba továbbítja (S2 jelez). A motor kikapcsol és a mágnesszelep nyitásával a tabletták belesznek a kémcsőbe. A tablettákat az optikai érzékelő segítségével számolhatjuk. A megfelelő számú tabletta beesése után a szelep zár, a szalag továbbmegy és a következő kémcsőt hozza töltési helyzetbe. Ez a művelet sor ismétlődik, amíg a berendezést le nem állítják. A kívánt darabszámot működés közben is bármikor megváltoztathatják a kiválasztó nyomógomb benyomásával. A változtatás csak a következő kémcsőre érvényes, az éppen töltés alatt lévő kémcsőbe még annyi tabletta kerül, amennyivel a töltése kezdődött. A kikapcsoláskor az éppen folyamatban lévő adagolást még befejezi a vezérlés.



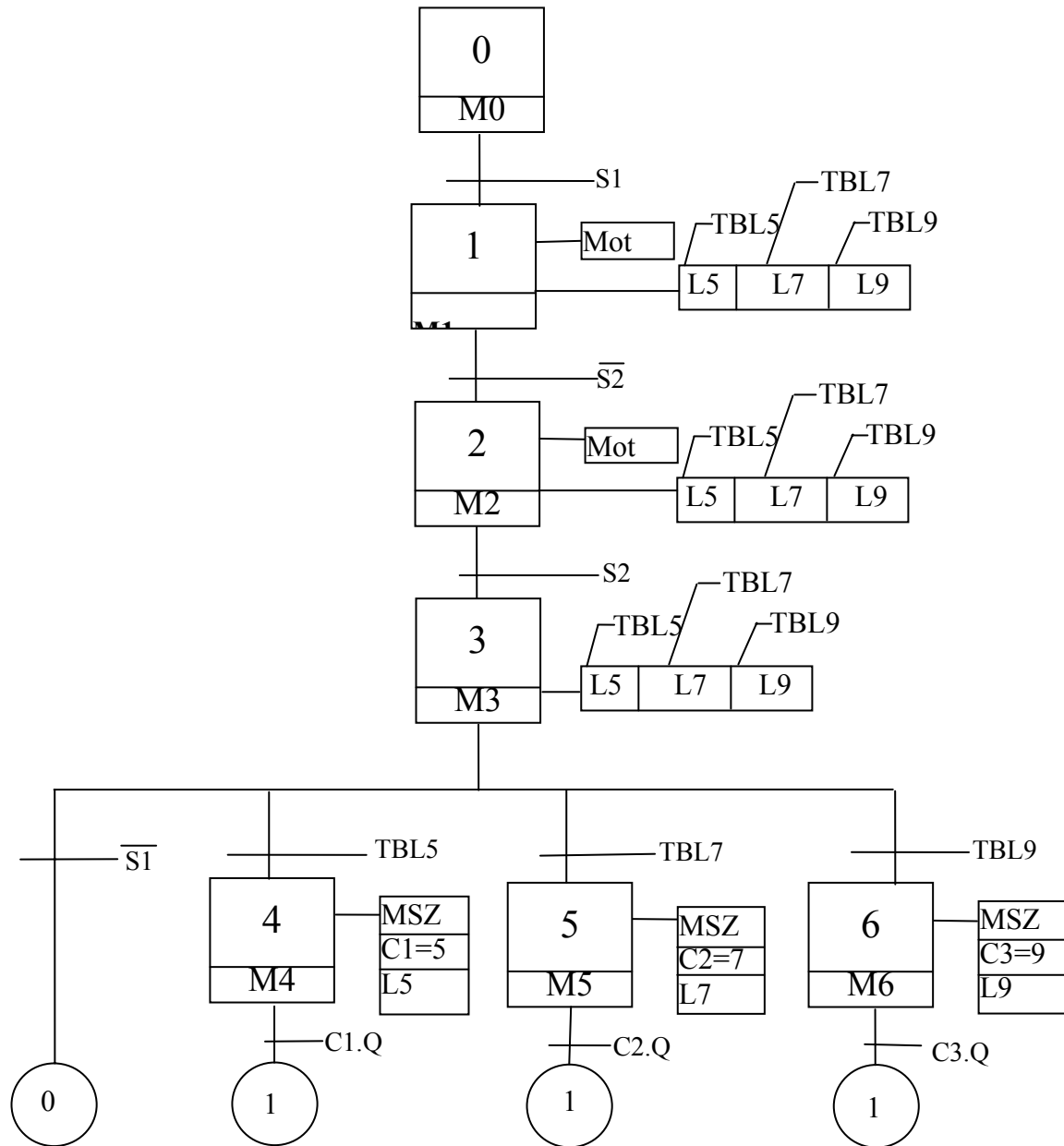
4. ábra Tablettaadagoló berendezés

A megoldásban a tabletta darabszámok kölcsönösen reteszelik egymást. Az egyik darabszámról a másikra átváltást mindig biztosítani kell, ezért szükséges a darabszámok tárolása, ezt RS – tárolókkal, ún. jelelfeldolgozással, az állapotgráfon kívül valósítjuk meg. A készülék kikapcsolása törli a tárolót.

## Összerendelési táblázat

Bemenetek	Jel	Logikai összerendelés	Cím
BE/KI kapcsoló	S1	<b>bekapcsolva:</b> S1=1	I0.0
A kémcső töltési pozícióban	S2	jelez, ha: S2=1	I0.1
5 DB TBL kiválasztó	S3	benyomva: S3=1	I0.2
7 DB TBL kiválasztó	S4	benyomva: S4=1	I0.3
9 DB TBL kiválasztó	S5	benyomva: S5=1	I0.4
optikai érzékelő	OPT	jelez, ha: OPT=1	I0.5
Kimenetek			
Motor	Mot	bekapcsolva: Mot=1	Q0.0
Szelep	MSZ	nyitva, ha: MSZ=1	Q0.1
5 TBL jelzőlámpa	L5	világít. ha: L5=1	Q0.2
7 TBL jelzőlámpa	L7	világít. ha: L7=1	Q0.3
9 TBL jelzőlámpa	L9	világít. ha: L9=1	Q0.4

**Az állapotgráf**

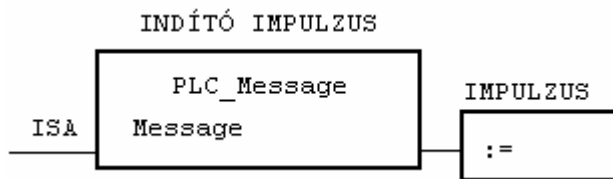


## A vezérlőalgorithmus felépítése

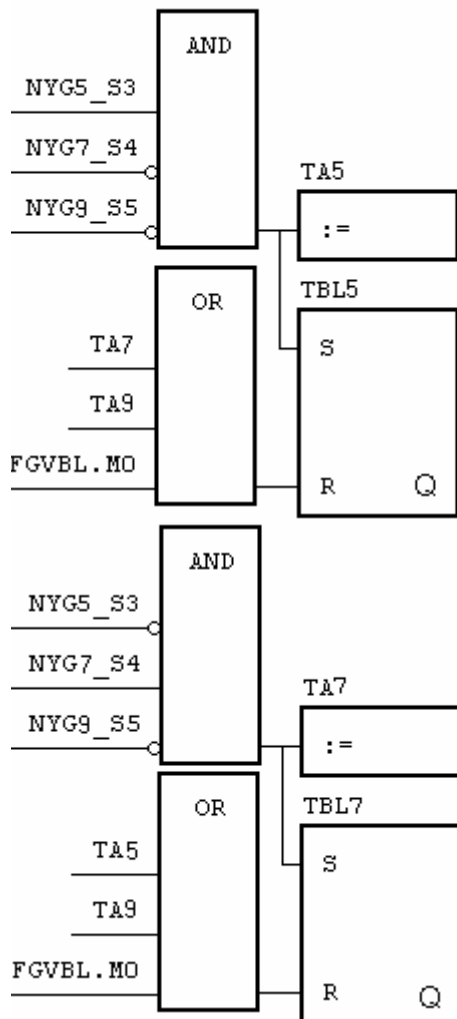
A vezérlést egy főprogram és egy függvényblokk segítségével valósítjuk meg. A főprogramban a ki/bemeneti jelek deklarálása mellett, még a függvényblokk-hívás előtt, feldolgozzuk a rövid ideig ható jeleket. Esetünkben ilyen jelnek számít a tableta darabszám kiválasztó nyomógomb. A függvényblokk az állapotgráfnak felel meg. A számlálók és a kimenetek kiszámítása is a függvényblokkban történik, a kimenetek beállítását a főprogram végzi.

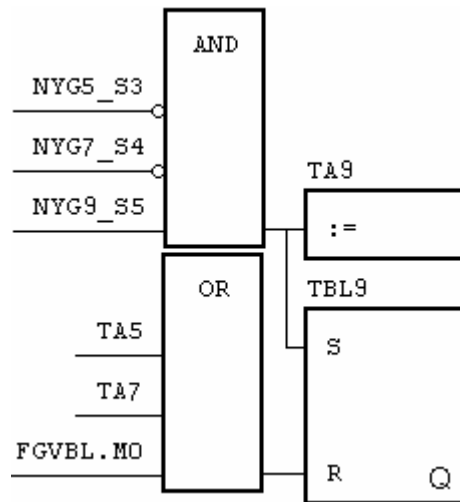
## A főprogram

Első programciklus lekérdezése:

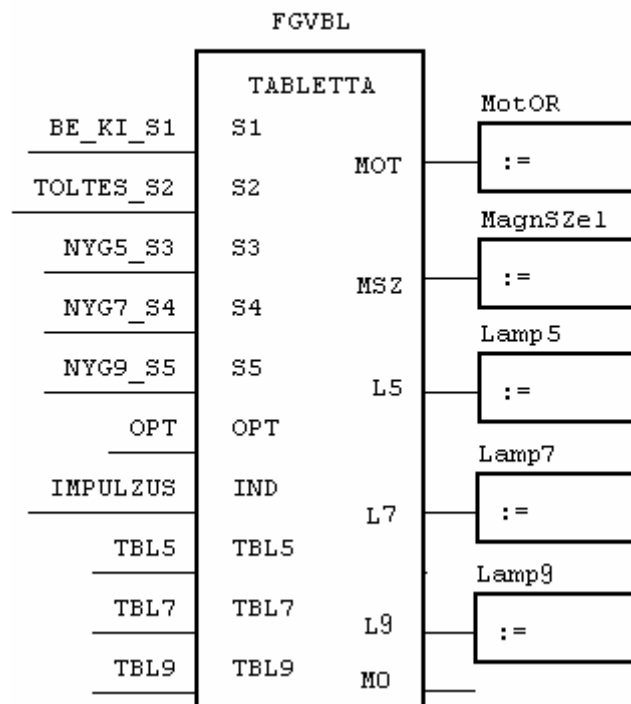


### JELFELDOLGOZÁS AZ ÁLLAPOTGRÁFON KÍVÜL





FÜGGVÉNYBLOKK HÍVÁSA



A funkcióterv átírása utasításlistába

PROGRAM PRTABL

VAR

```

BE_KI_S1 AT %I0.0.0.0.0: BOOL;
TOLTES_S2 AT %I0.0.0.0.1:   BOOL;
NYG5_S3 AT %I0.0.0.0.2:  BOOL;
NYG7_S4 AT %I0.0.0.0.3:  BOOL;
NYG9_S5 AT %I0.0.0.0.4:  BOOL;
OPT AT %I0.0.0.0.5:  BOOL;
MotOR AT %Q0.0.0.0.0:   BOOL;
MagnSZel AT %Q0.0.0.0.1:  BOOL;
Lamp5 AT %Q0.0.0.0.2:   BOOL;
    
```

```
Lamp7 AT %Q0.0.0.0.3:   BOOL;
Lamp9 AT %Q0.0.0.0.4:   BOOL;
TA5:  BOOL;
TA7:  BOOL;
TA9:  BOOL;
TBL5: BOOL;
TBL7: BOOL;
TBL9: BOOL;
IMPULZUS:  BOOL;
FGVBL:    TABLETTA;
END_VAR
```

(\*INDÍTÓ IMPULZUS\*)

```
LD  ISA
PLC_Message
ST  IMPULZUS
```

(\*JELFELDOLGOZÁS AZ ÁLLAPOTGRÁFON KÍVÜL\*)

```
LD  NYG5_S3
ANDNNYG7_S4
ANDNNYG9_S5
ST  TA5
S   TBL5
LD  TA7
OR  TA9
OR  FGVBL.M0
R   TBL5
```

```
LDN NYG5_S3
AND NYG7_S4
ANDNNYG9_S5
ST  TA7
S   TBL7
LD  TA5
OR  TA9
OR  FGVBL.M0
R   TBL7
```

```
LDN NYG5_S3
ANDNNYG7_S4
AND NYG9_S5
ST  TA9
S   TBL9
LD  TA5
OR  TA7
OR  FGVBL.M0
R   TBL9
```

(\*FÜGGVÉNYBLOKK HÍVÁSA\*)

```

CAL  FGVBL(IND:=IMPULZUS,
        S1:=BE_KI_S1,
        S2:=TÖLTÉS_S2,
        S3:=NYG5_S3,
        S4:=NYG7_S4,
        S5:=NYG9_S5,
        OPT:=OPT,
        TBL5:=TBL5,
        TBL7:=TBL7,
        TBL9:=TBL9
    )
    
```

(\* KIMENETEK \*)

```

LD  FGVBL.MOT
ST  MotOR
    
```

```

LD  FGVBL.MSZ
ST  MagnSZel
    
```

```

LD  FGVBL.L5
ST  Lamp5
    
```

```

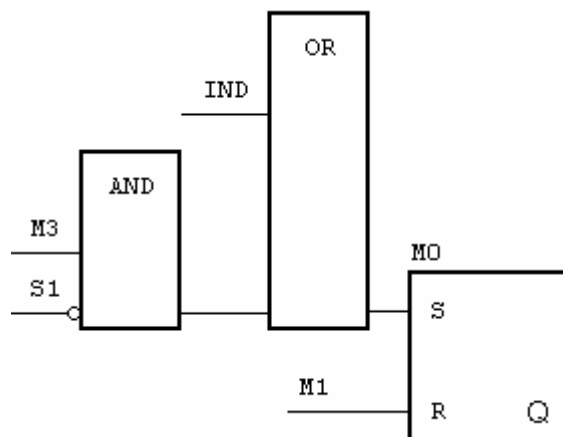
LD  FGVBL.L7
ST  Lamp7
    
```

```

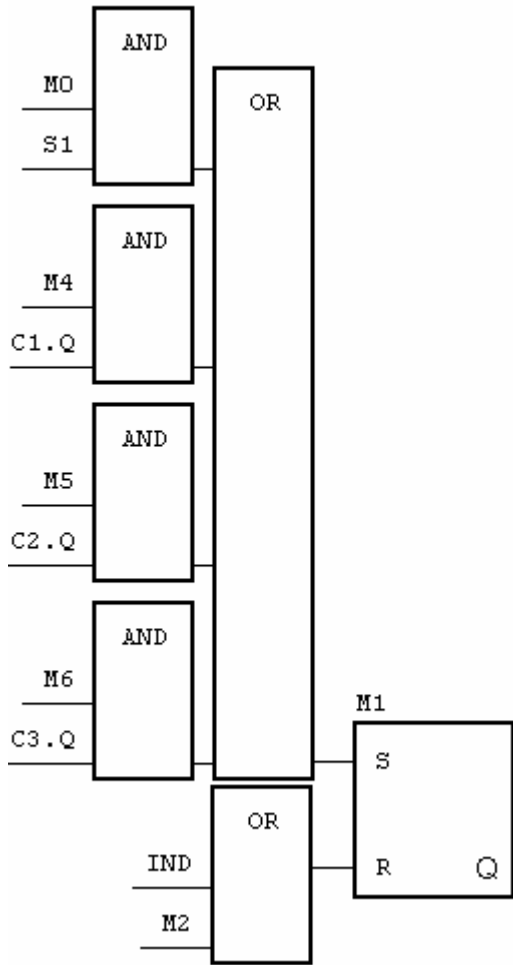
LD  FGVBL.L9
ST  Lamp9
END_PROGRAM
    
```

## A függvényblokk

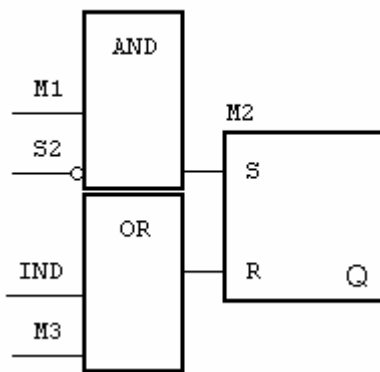
### Funkcióterv



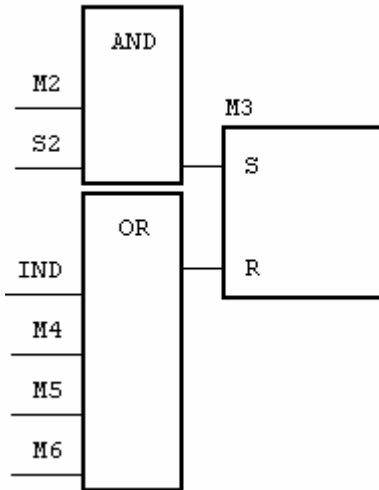
1. ÁLLAPOT



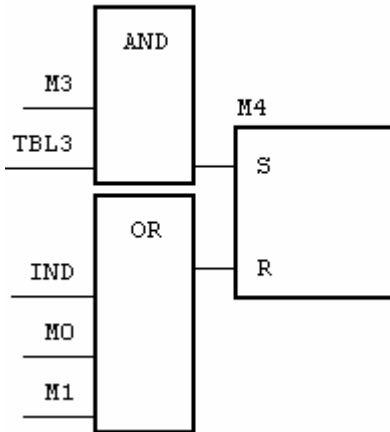
2. ÁLLAPOT



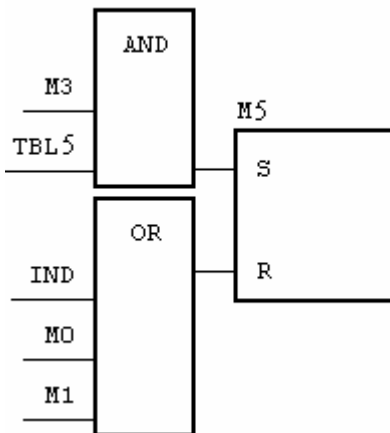
3. ÁLLAPOT



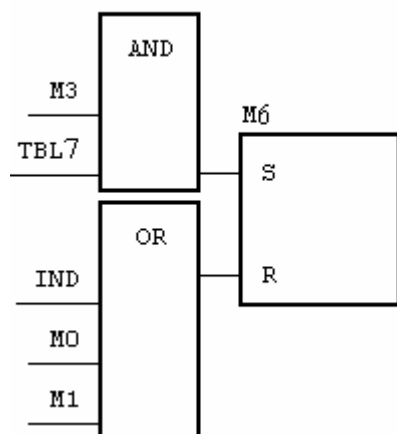
4. ÁLLAPOT



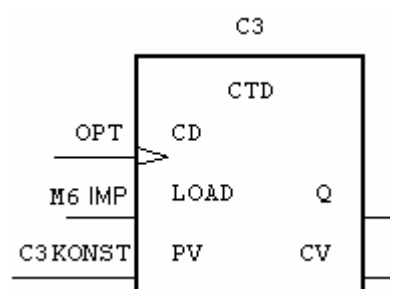
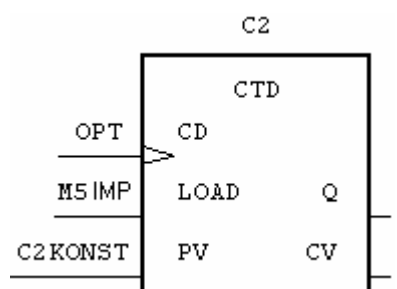
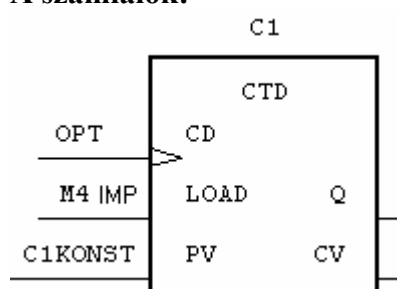
5. ÁLLAPOT

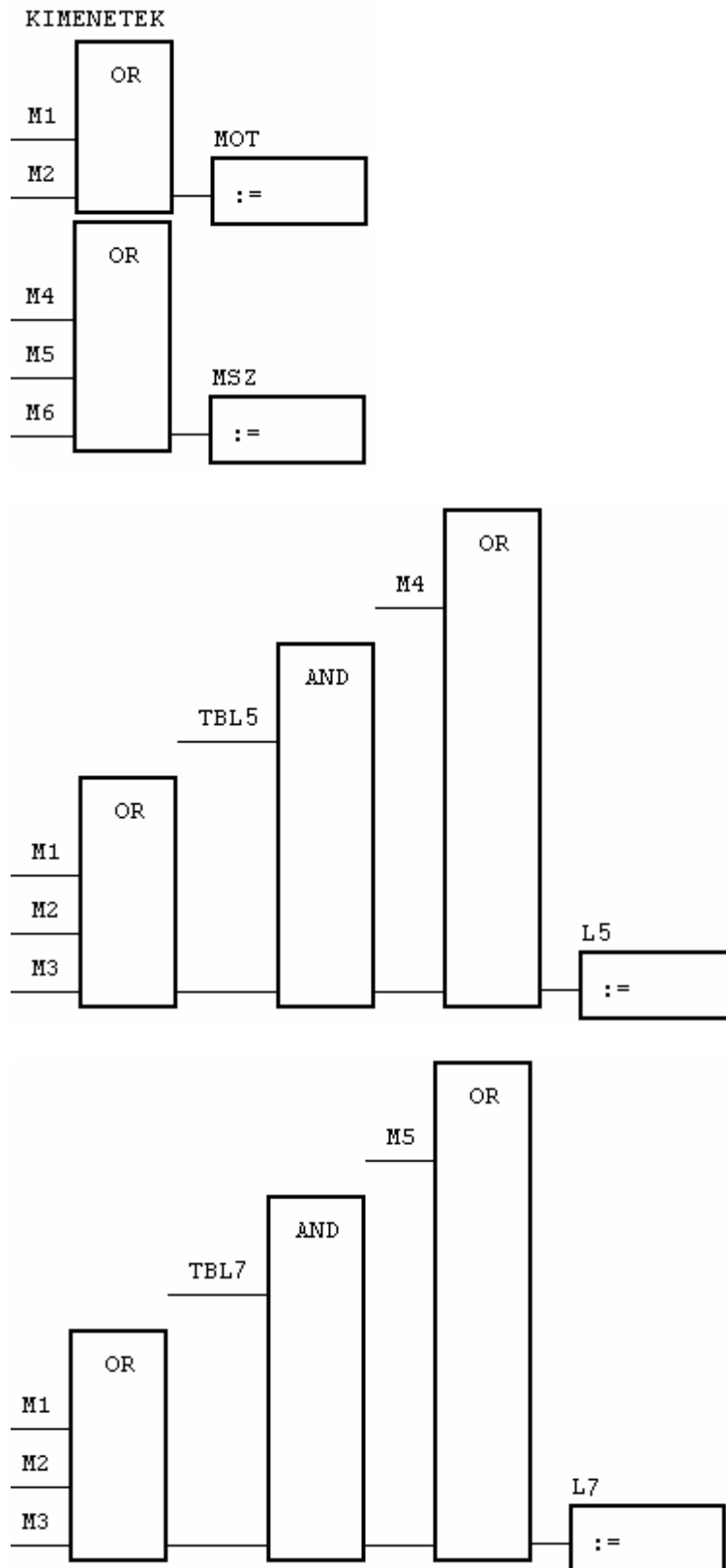


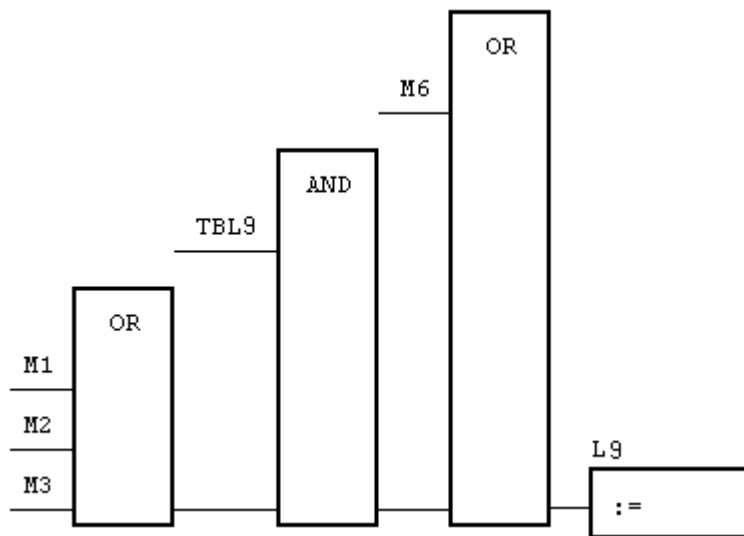
6. ÁLLAPOT



A számlálók:







### A függvényblokk utasításlista

FUNCTION\_BLOCK TABLETTA

VAR\_INPUT

S1: BOOL;  
 S2: BOOL;  
 S3: BOOL;  
 S4: BOOL;  
 S5: BOOL;  
 OPT: BOOL;  
 IND: BOOL;  
 TBL5: BOOL;  
 TBL7: BOOL;  
 TBL9: BOOL;

END\_VAR

VAR\_OUTPUT

MOT: BOOL;  
 MSZ: BOOL;  
 L5: BOOL;  
 L7: BOOL;  
 L9: BOOL;  
 M0: BOOL;

END\_VAR

VAR

M1: BOOL;  
 M2: BOOL;  
 M3: BOOL;  
 M4: BOOL;  
 M5: BOOL;  
 M6: BOOL;  
 C1: CTD;

END\_VAR

VAR constant

C1KONST: INT := 5;

```
END_VAR
```

```
VAR
```

```
    C1IMP:    R_TRIG;
```

```
    C2:    CTD;
```

```
END_VAR
```

```
VAR constant
```

```
    C2KONST: INT := 7;
```

```
END_VAR
```

```
VAR
```

```
    C2IMP:    R_TRIG;
```

```
    C3:    CTD;
```

```
END_VAR
```

```
VAR constant
```

```
    C3KONST: INT := 9;
```

```
END_VAR
```

```
VAR
```

```
    C3IMP:    R_TRIG;
```

```
END_VAR
```

```
(*0. ÁLLAPOT*)
```

```
LD  IND
```

```
OR(  M3
```

```
ANDNS1
```

```
)
```

```
S  M0
```

```
LD  M1
```

```
R  M0
```

```
(*1. ÁLLAPOT*)
```

```
LD  M0
```

```
AND S1
```

```
OR(  M4
```

```
AND C1.Q
```

```
)
```

```
OR(  M5
```

```
AND C2.Q
```

```
)
```

```
OR(  M6
```

```
AND C3.Q
```

```
)
```

```
S  M1
```

```
LD  IND
```

```
OR  M2
```

```
R  M1
```

```
(*2. ÁLLAPOT*)
```

```
LD  M1
```

```
ANDNS2
```

```
S  M2
```

```
LD  IND
```

```
OR  M3
```

```
R  M2
```

```
(*3. ÁLLAPOT*)
```

```
LD  M2
```

```
AND S2
```

```
S  M3
```

```
LD  IND
```

```
OR  M4
```

```
OR  M5
```

```
OR  M6
```

```
R  M3
```

```
(*4. ÁLLAPOT*)
```

```
LD  M3
```

```
AND TBL5
```

```
S  M4
```

```
LD  IND
```

```
OR  M0
```

```
OR  M1
```

```
R  M4
```

(\*5. ÁLLAPOT\*)

```
LD M3
AND TBL7
S M5
LD IND
OR M0
OR M1
R M5
```

(\*6. ÁLLAPOT\*)

```
LD M3
AND TBL9
S M6
LD IND
OR M0
OR M1
R M6
```

(\*SZÁMLÁLÓK\*)

```
LD OPT
ST C1.CD
LD M4
ST C1IMP.CLK
CAL C1IMP
LD C1IMP.Q
ST C1.LOAD
LD C1KONST
ST C1.PV
CAL C1
```

```
LD OPT
ST C2.CD
LD M5
ST C2IMP.CLK
CAL C2IMP
LD C2IMP.Q
ST C2.LOAD
LD C2KONST
ST C2.PV
CAL C2
```

```
LD OPT
ST C3.CD
LD M6
ST C3IMP.CLK
CAL C3IMP
LD C3IMP.Q
ST C3.LOAD
```

```
LD C3KONST
ST C3.PV
CAL C3
```

(\*KIMENETEK\*)

```
LD M1
OR M2
ST MOT
```

```
LD M4
OR M5
OR M6
ST MSZ
```

```
LD M4
OR( TBL5
AND( M1
OR M2
OR M3
)
)
ST L5
```

```
LD M5
OR( TBL7
AND( M1
OR M2
OR M3
)
)
ST L7
```

```
LD M6
OR( TBL9
AND( M1
OR M2
OR M3
)
)
ST L9
```

END\_FUNCTION\_BLOCK

## Ütemvezérelt lefutóvezérlések

### Közlekedési lámpa vezérlése

Egy közlekedési lámpa **3 időegységig piros**, ahol a **3. időegységben** a piros mellett a **sárga** lámpa is világít. Ezután **4 időegység zöld** fázis következik. A ciklus **1 időegység sárgával** zárul.

Az időegység legyen **5 s**.

### Összerendelési táblázat

Bemenetek	Jel	Logikai hozzárendelés	Cím
BE/KI kapcsoló	S0	bekapcsolva: S0=1	I0.0
Kimenetek			
Piros. lámpa	P	világít, ha: P=1	Q0.0
Sárga lámpa	S	világít, ha: S=1	Q0.1
Zöld. lámpa	Z	világít, ha: Z=1	Q0.2

### Megoldás 1. változat

Egy ciklus 8 időegységből áll. Az időütemet egy ütemadó adja, amelynek igen egyszerű a programja: az időzítő kimenőjelével vezérelt UTEM-merker legyen 0 állapotú. A merker negáltját az indítójellel (S0) együtt egy bekapcsolás-késleltetési időzítő IN bemenetéhez kapcsoljuk. Bekapcsoláskor elindul az időzítő, és a kimenetén 5s elteltével megjelenik az 1 jel, amely csak egyetlen ciklusideig „él”, mert a következő ciklusban az indítófeltétele nullára vált. Ez a ciklus ismétlődik mindaddig, amíg S0-al ki nem kapcsoljuk a vezérlést.

Az ütemgenerátor impulzusai egy számlálót inkrementálnak. Ha a számláló értéke eléri a 8-at, a számlálót RESET-elni kell. A számláló értéke megadja az ütemszámot, amely segítségével a kimenőjelek beállíthatók.

Ütem	Lámpafázisok			A számláló értéke	ütemmerker
1	<b>P</b>			0	M1
2	<b>P</b>			1	M2
3	<b>P</b>	<b>S</b>		2	M3
4			<b>Z</b>	3	M4
5			<b>Z</b>	4	M5
6			<b>Z</b>	5	M6
7			<b>Z</b>	6	M7
8		<b>S</b>		7	M8

## Utasításlista

PROGRAM KZLAMPÁ

VAR

```
S0 AT %I0.0.0.0.0:   BOOL;
PIROS AT %Q0.0.0.0.0:   BOOL;
SARGA AT %Q0.0.0.0.1:   BOOL;
ZOLD AT %Q0.0.0.0.2:   BOOL;
SZAMLALO: CTU;
UTEM_MAX: INT := 8;
T1:   TON;
UTEMIDO:   TIME := t#5S;
M1:   BOOL;
M2:   BOOL;
M3:   BOOL;
M4:   BOOL;
M5:   BOOL;
M6:   BOOL;
M7:   BOOL;
M8:   BOOL;
UTEM:BOOL;
USZAM:   INT;
```

END\_VAR

```
(*ÜTEMGENERÁTOR*)
LD    S0
ANDN  UTEM
ST    T1.IN
LD    UTEMIDO
ST    T1.PT
CAL   T1
LD    T1.Q
ST    UTEM

LD    USZAM
EQ    4
ST    M5
LD    USZAM
EQ    5
ST    M6
LD    USZAM
EQ    6
ST    M7
LD    USZAM
EQ    7
ST    M8

(*SZÁMLÁLÓ*)
LD    UTEM
ST    SZAMLALO.CU
LD    SZAMLALO.CV
GE    UTEM_MAX
ORN   S0
ST    SZAMLALO.RESET
CAL   SZAMLALO

(*ÜTEM-MERKEREK*)
LD    SZAMLALO.CV
ST    USZAM
EQ    0
ST    M1
LD    USZAM
EQ    1
ST    M2
LD    USZAM
EQ    2
ST    M3
LD    USZAM
EQ    3
ST    M4

(*PIROS LÁMPA*)
LD    M1
OR    M2
OR    M3
ST    PIROS

(*SÁRGA LÁMPA*)
LD    M3
OR    M8
ST    SARGA

(*ZÖLD LÁMPA*)
LD    M4
OR    M5
OR    M6
OR    M7
ST    ZOLD
END_PROGRAM
```

**Megoldás 2. változat, utasításlista**

PROGRAM KZLAMP2

VAR

```

S0 AT %I0.0.0.0.0:    BOOL;
PIROS AT %Q0.0.0.0.0:    BOOL;
SARGA AT %Q0.0.0.0.1:    BOOL;
ZOLD AT %Q0.0.0.0.2:    BOOL;
UTGEN AT %QB0.0.0.1:    BYTE;
FELFUTO:    R_TRIG;
FGVBL:    FBKLAMP;
FGVBL2:    FBSARGA;

```

END\_VAR

```

CAL FELFUTO(CLK:=S0)
LD    S0
JMPCN    VILLOG
LD    FELFUTO.Q
ST    FGVBL.INDIMP
CAL FGVBL
LD    FGVBL.P
ST    PIROS
LD    FGVBL.SA
ST    SARGA
LD    FGVBL.Z
ST    ZOLD
LD    FGVBL.UTEMB
ST    UTGEN
RET

```

```

VILLOG:
CAL    FGVBL2
LD    FGVBL2.P
ST    PIROS
LD    FGVBL2.SA
ST    SARGA
LD    FGVBL2.Z
ST    ZOLD
LD    FGVBL2.UTEMB
ST    UTGEN
RET
END_PROGRAM

```

FUNCTION\_BLOCK FBKLAMP

VAR\_INPUT

INDIMP: BOOL;

END\_VAR

VAR\_OUTPUT

```

P:    BOOL;
SA:    BOOL;
Z:    BOOL;
UTEMB:    BYTE;

```

END\_VAR

VAR

```

T1:    TON;
MB:    BYTE;
UTIMP:    BOOL;
UTEMIDO:    TIME := t#1S;

```

END\_VAR

```

LD    INDIMP
JMPCN    TOVABB
LD    1
ST    MB

```

```

TOVABB:
(*ÜTEMGENERÁTOR*)
LDN    UTIMP
ST    T1.IN

```



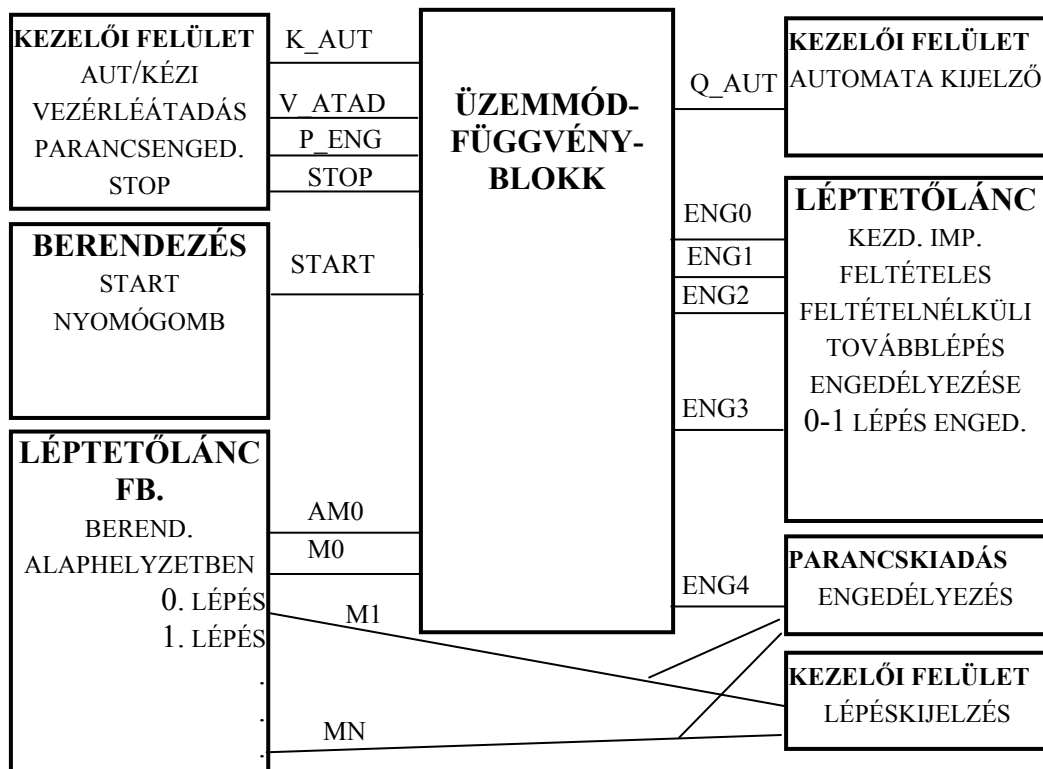
## Folyamatvezérelt lefutóvezérlések

### Az üzemmód programrész (függvényblokk)

Az üzemmód függvényblokk feldolgozza a kezelői felületről és a berendezés felől jövő parancsokat, jelzéseket és engedélyező jelek formájában továbbítja a léptetőlánc felé. Visszajelzi ezen kívül az üzemmódot, és engedélyezi a kimenetek működtetését.



5. ábra A kezelői felület



2. ábra Az üzemmód programrész kapcsolata a többi programrészsel, illetve a kezelői felülettel és a berendezéssel

### Az üzemmód függvényblokk

Készítsük el a fenti ábra előírásainak megfelelő függvényblokkot. A függvényblokk definiálásakor arra törekedtünk, hogy csak a konkrét technológiától független jeleket, a kezelői felület jeleit dolgozzuk fel. Azonos kezelői felület esetén így a későbbiekben változtatás nélkül felhasználhatjuk a függvényblokkot.



3. ábra A kezelői felület a függvényblokk változónév jelöléseivel

#### A függvényblokk utasításlistája:

```
FUNCTION_BLOCK UMODFGV
VAR_INPUT
    I1 : BOOL ;
    I2 : BOOL ;
    I3 : BOOL ;
    I4 : BOOL ;
    AM0 : BOOL ; (*BERENDEZÉS ALAPÁLLAPOTA*)
    M0 : BOOL ; (*0. LÉPÉS*)
    S0 : BOOL ; (*START a főprogramban*)
END_VAR
VAR_OUTPUT
    Q4 : BOOL ; (*AUTOMATA ÜZEMMÓD*)
    ENG0 : BOOL ; (*EN0 a főprogramban*)
    ENG1 : BOOL ; (*EN1 a főprogramban*)
    ENG2 : BOOL ; (*EN2 a főprogramban*)
    ENG3 : BOOL ; (*EN3 a főprogramban*)
    ENG4 : BOOL ; (*EN4 a főprogramban*)
END_VAR
VAR
    V_IMP : BOOL ;
    B11 : BOOL ;
    T_STOP : BOOL ;
END_VAR
    Impulzuskapcsoló a vezérlésátadás felfutó élére (V_IMP)
    LD    I2
    ANDN    B11
    ST    V_IMP
    LD    I2
    ST    B11

    Indító impulzus (ENG0)
```

```

LD  V_IMP
AND AM0
ANDN    Q4
AND  I1
ANDN    M0
ST  ENG0

```

Automata üzemmód kijelzés (Q4) és feltételes léptetés engedélyező jele (ENG1)

```

LD  AM0
AND V_IMP
AND M0
S   Q4
LDN I1
OR( T_STOP
AND M0
)
R   Q4
LD  Q4
ST  ENG1

```

Stop nyomógomb tárolása (T\_STOP)

```

LD  Q4
ANDN    I4
S   T_STOP
LDN  Q4
R   T_STOP

```

Feltétel nélküli továbblépés engedélyező jele (ENG2)

```

LD  V_IMP
ANDN    I1
ST  ENG2

```

Parancsengedélyezés (ENG4)

```

LD  S0
AND AM0
ST  ENG3
LD  Q4
OR(  I3
ANDN    I1
)
ST  ENG4

```

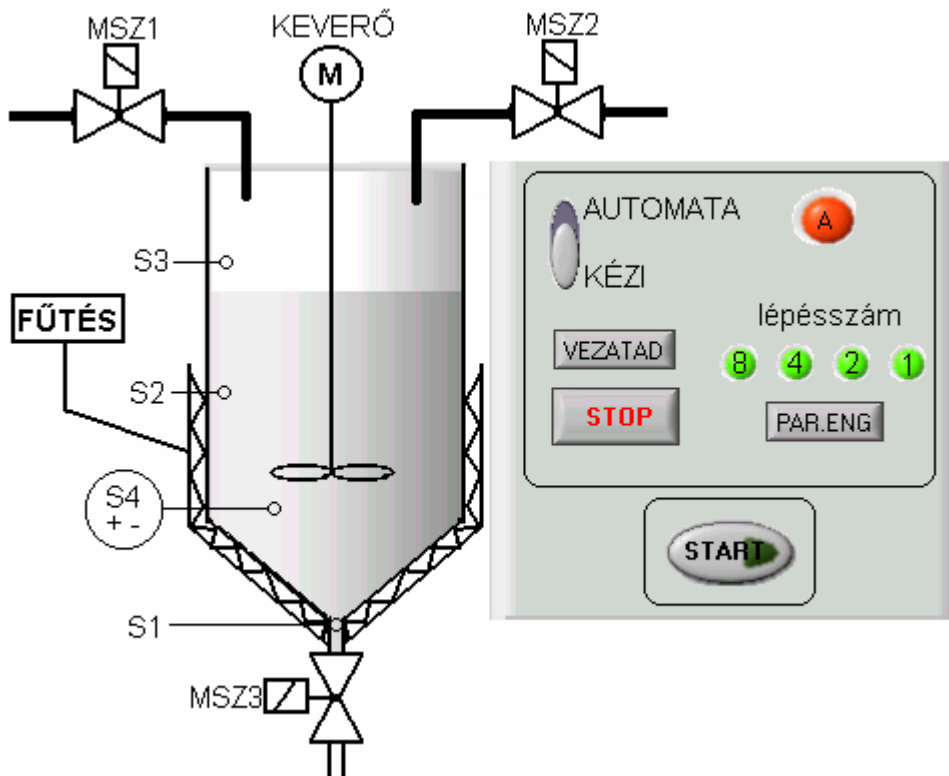
```

END_FUNCTION_BLOCK

```

### Szakaszos üzemű folyadékkeverő berendezés vezérlése

A technológiai feladat két különböző folyadék összelegyítése és adott hőmérsékletre melegítése.



6. ábra Keverőtartály a kezelőtáblával

A berendezés szakaszos üzemben működik. Alapállapotban a tartály üres, a szelepek zárva. Automata üzemmódban a technológia az alábbi lépések sorozata:

**START** nyomógomb benyomására, a **MSZ1** jelű mágnesszelep nyit, az 1. folyadék beáramlik a tartályba. Ha **S2** szintérzékelő jelez, a **MSZ1** szelep zár, **MSZ2** nyit, bekapcsol a keverés, és beáramlik a 2. folyadék. Ha **S3** szintérzékelő jelez, **MSZ2** zár, a keverő mellett bekapcsol a fűtés. Ha a tartályban lévő folyadék hőmérséklete elérte a kívánt értéket, **S4** jelez, leáll a fűtés és a keverés, az **MSZ3** mágnesszelep nyit, a tartály leürül. Ha **S1** jelez, **START** jelre ismét indulhat előlről a folyamat.

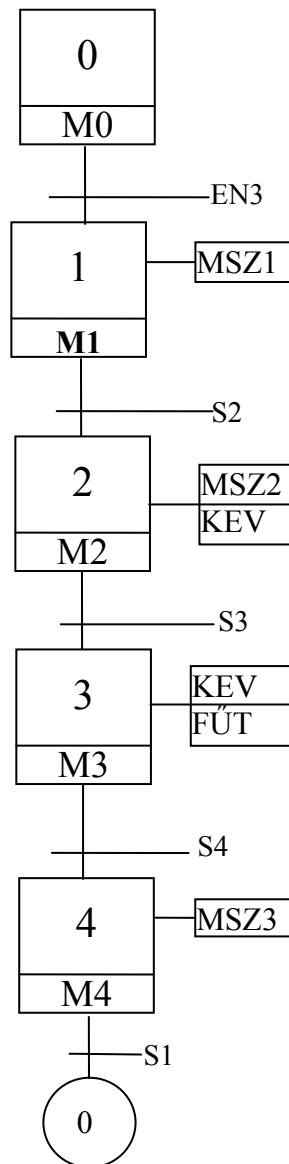
A berendezéshez tartozik egy olyan kezelői felület, amely biztosítja az üzemmód megválasztását, és különböző üzemmódokban a berendezés felügyeletét, a vezérlési lépések nyomon követését, a vezérlés tesztelését.

**Összerendelési táblázat**

A táblázatban felsoroljuk a technológiai jelek mellett a kezelői felület jeleit is.

<b>Bemenetek</b>	<b>Jel</b>	<b>Logikai összerendelés</b>	<b>Cím</b>
START	START	benyomva: START=1	I1.4
üres a tartály	S1	üres, ha: S1=1	I0.0
a tartály félig	S2	jelez, ha: S2=1	I0.1
a tartály tele	S3	jelez, ha: S3=1	I0.2
hőmérsékletérzékelő	S4	a mért hőmérséklet >= kívánt érték: S4=1	I0.3
Automata/kézi átkapcsoló	KAUT	automata: KAUT=1	I1.0
vezérlés átadás nyomógomb	VATAD	benyomva: VATAD=1	I1.1
parancsengedélyezés	PENG	benyomva: PENG=1	I1.2
STOP nyomógomb	STOP	benyomva: STOP=0	I1.3
<b>Kimenetek</b>	<b>Jel</b>	<b>Logikai összerendelés</b>	<b>Cím</b>
1. mágnesszelep	MSZ1	nyitva, ha: MSZ1=1	Q0.0
2. mágnesszelep	MSZ2	nyitva, ha: MSZ2=1	Q0.1
3. mágnesszelep	MSZ3	nyitva, ha: MSZ3=1	Q0.2
fűtés	FUT	bekapcsolva, ha: FUT=1	Q0.3
keverés	KEV	bekapcsolva, ha: KEV=1	Q0.4
lépéskijelző LED	LEP 0	világít, ha: LEP0=1	Q1.1
lépéskijelző LED	LEP1	világít, ha: LEP1=1	Q1.2
lépéskijelző LED	LEP2	világít, ha: LEP2=1	Q1.3
lépéskijelző LED	LEP3	világít, ha: LEP3=1	Q1.4
automata üzemmód jelzés	QAUT	világít, ha: LEP4=1	Q1.0

## A léptetőlánc



A léptetőlánc funkciótervbe való átírásakor figyelembe vesszük az üzemmód függvényblokk engedélyező jeleit is.

:

A vezérlés függvényblokkjai	prototípus	feladata
üzemmód	UMODFB	üzemmód beállítása, engedélyező jelek
léptetőlánc	LEPTL	meghatározza a berendezés alapállapotát és beállítja a lépésmerkereket
lépéskijelzés	LEPKIJ	a lépéskijelző LED-ek beállítása
parancskiadás	PARKIAD	a kimenetek beállítása

**A főprogram**

A főprogramot utasításlistában adjuk meg.

**A változódeklaráció:**

PROGRAM kevtart

VAR

```

KAUT AT %I0.0.0.1.0:   BOOL;      (* =1 :AUTO =0 :KÉZI *)
VATAD AT %I0.0.0.1.1:  BOOL;      (* =1 BENYOMVA *)
PENG AT %I0.0.0.1.2:   BOOL;      (* =1 BENYOMVA *)
STOP AT %I0.0.0.1.3:   BOOL;      (* =0 BENYOMVA *)
START AT %I0.0.0.1.4:  BOOL;      (* =1 BENYOMVA *)
QAUT AT %Q0.0.0.1.0:   BOOL;
LEP1 AT %Q0.0.0.1.1:   BOOL;      (* LEPESkijelzés *)
LEP2 AT %Q0.0.0.1.2:   BOOL;      (* LEPESkijelzés *)
LEP3 AT %Q0.0.0.1.3:   BOOL;      (* LEPESkijelzés *)

```

```

EN0:  BOOL;              (*indító impulzus*)
EN1:  BOOL;              (*feltételes továbblépés engedélyezése*)
EN2:  BOOL;              (*feltétel nélküli továbblépés engedélyezése*)
EN3:  BOOL;              (*0-1 lépés engedélyezése*)
EN4:  BOOL;              (*parancsengedélyezés*)

```

END\_VAR

VAR

```

S1 AT %I0.0.0.0.0:   BOOL;      (* 1.SZINT *)
S2 AT %I0.0.0.0.1:   BOOL;      (* 2.SZINT *)
S3 AT %I0.0.0.0.2:   BOOL;      (* 3.SZINT *)
S4 AT %I0.0.0.0.3:   BOOL;      (* HOMERSEKLET *)

```

```

MSZ1 AT %Q0.0.0.0.0:  BOOL;      (* 1.SZELEP *)
MSZ2 AT %Q0.0.0.0.1:  BOOL;      (* 2.SZELEP *)
MSZ3 AT %Q0.0.0.0.2:  BOOL;      (* 3.SZELEP *)
FUT AT %Q0.0.0.0.3:   BOOL;      (* FUTES *)
KEV AT %Q0.0.0.0.4:   BOOL;      (* KEVERO *)

```

```

UZEM:    UMODFB;
LANC:    LEPTL;
KIJELZ:  LEPKIJ;
KIMENET: PARKIAD;

```

END\_VAR

VAR\_GLOBAL

```

M0:  BOOL;      (* 0. lépés *)
M1:  BOOL;      (* 1.lépés *)
M2:  BOOL;      (* 2.lépés *)
M3:  BOOL;      (* 3.lépés *)
M4:  BOOL;      (* 4.lépés *)
AM0: BOOL;      (* berendezés alapállapota *)

```

END\_VAR

```
LD KAUT
ST UZEM.I1
LD VATAD
ST UZEM.I2
LD PENG
ST UZEM.I3
LD STOP
ST UZEM.I4
LD START
ST UZEM.S0
LD AM0
ST UZEM.AM0
LD M0
ST UZEM.M0
```

```
CAL UZEM
```

```
LD UZEM.Q4
ST QAUT
LD UZEM.ENG0
ST EN0
LD UZEM.ENG1
ST EN1
LD UZEM.ENG2
ST EN2
LD UZEM.ENG3
ST EN3
LD UZEM.ENG4
ST EN4
```

```
CAL LANC(ENG0:=EN0,ENG1:=EN1,
ENG2:=EN2,ENG3:=EN3,
S1:=S1,S2:=S2,
S3:=S3,S4:=S4)
```

```
CAL KIJELZ
```

```
LD KIJELZ.Q0
ST LEP1
LD KIJELZ.Q1
ST LEP2
LD KIJELZ.Q2
ST LEP3
```

```
CAL KIMENET(ENG4:=EN4)
```

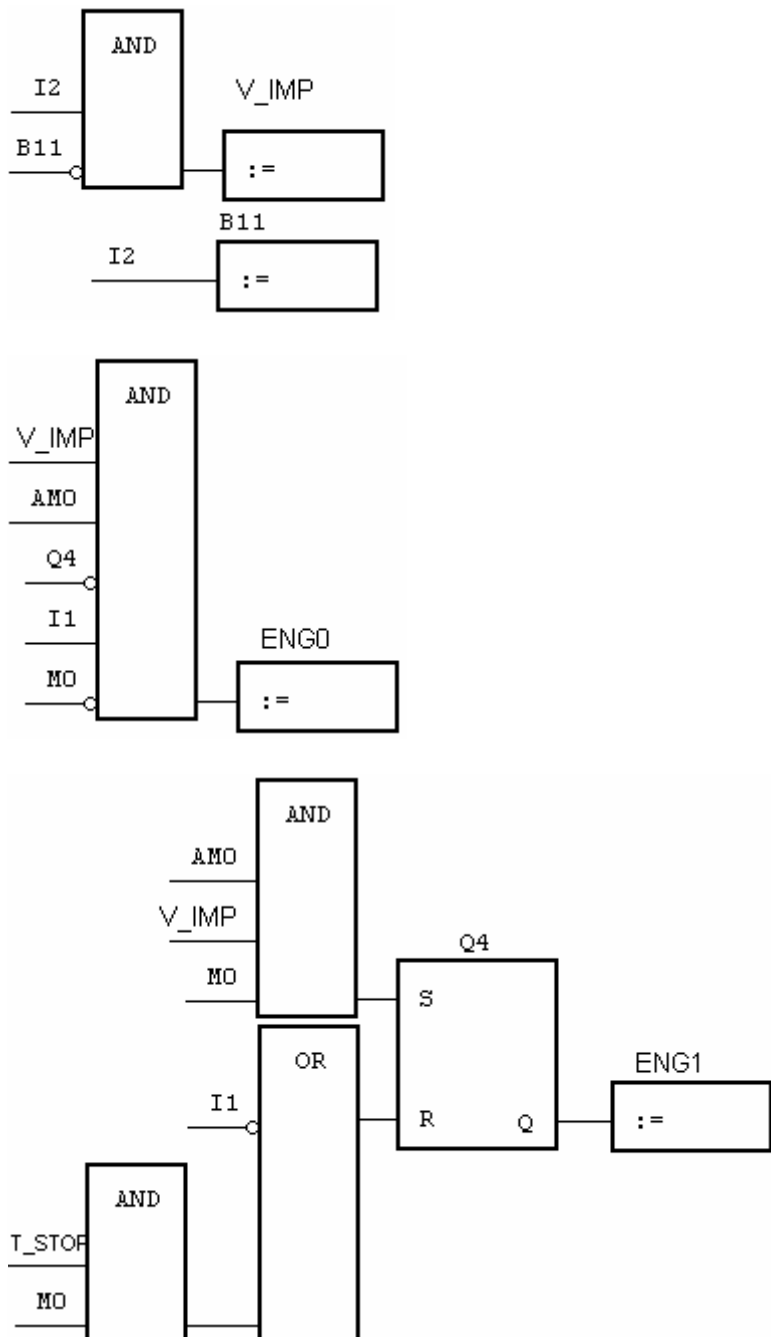
```
LD KIMENET.Y1
ST MSZ1
LD KIMENET.Y2
```

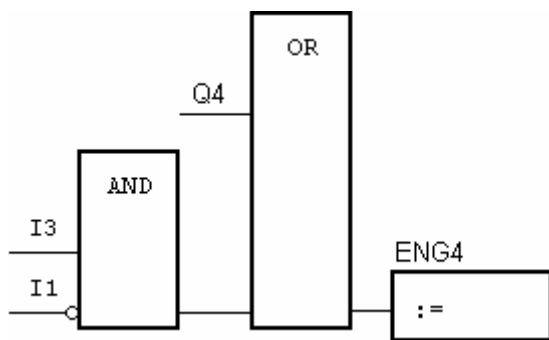
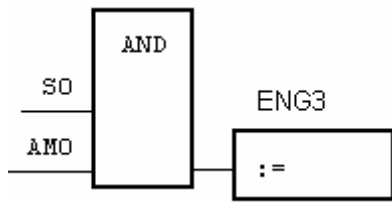
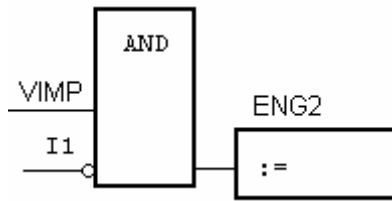
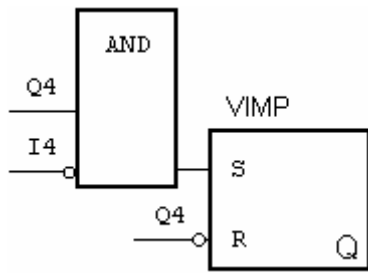
```

ST   MSZ2
LD   KIMENET.Y3
ST   MSZ3
LD   KIMENET.H
ST   FUT
LD   KIMENET.MOT
ST   KEV
END_PROGRAM
    
```

**Az üzemmód függvényblokk**

**Funkcióterv**





**Utasításlista**

FUNCTION\_BLOCK UMODFB

VAR\_INPUT

```

I1 : BOOL ;
I2 : BOOL ;
I3 : BOOL ;
I4 : BOOL ;
AM0 : BOOL ;
M0 : BOOL ;
S0 : BOOL ;

```

END\_VAR

VAR\_OUTPUT

```

Q4 : BOOL ;
ENG0 : BOOL ;
ENG1 : BOOL ;
ENG2 : BOOL ;
ENG3 : BOOL ;
ENG4 : BOOL ;

```

END\_VAR

VAR

```

V_IMP : BOOL ;
B11 : BOOL ;
T_STOP : BOOL ;

```

END\_VAR

```

LD  I2
ANDN      B11
ST  V_IMP

```

```

LD  I2
ST  B11

```

```

LD  V_IMP
AND  AM0
ANDN      Q4
AND  I1
ANDN      M0
ST  ENG0

```

```

LD  AM0
AND  V_IMP
AND  M0
S   Q4

```

```

LDN  I1
OR(  T_STOP
AND  M0
)
R   Q4

```

```

LD  Q4
ST  ENG1
LD  Q4
ANDN      I4
S   T_STOP

```

```

LDN  Q4
R   T_STOP

```

```

LD  V_IMP
ANDN      I1
ST  ENG2

```

```

LD  S0
AND  AM0
ST  ENG3

```

```

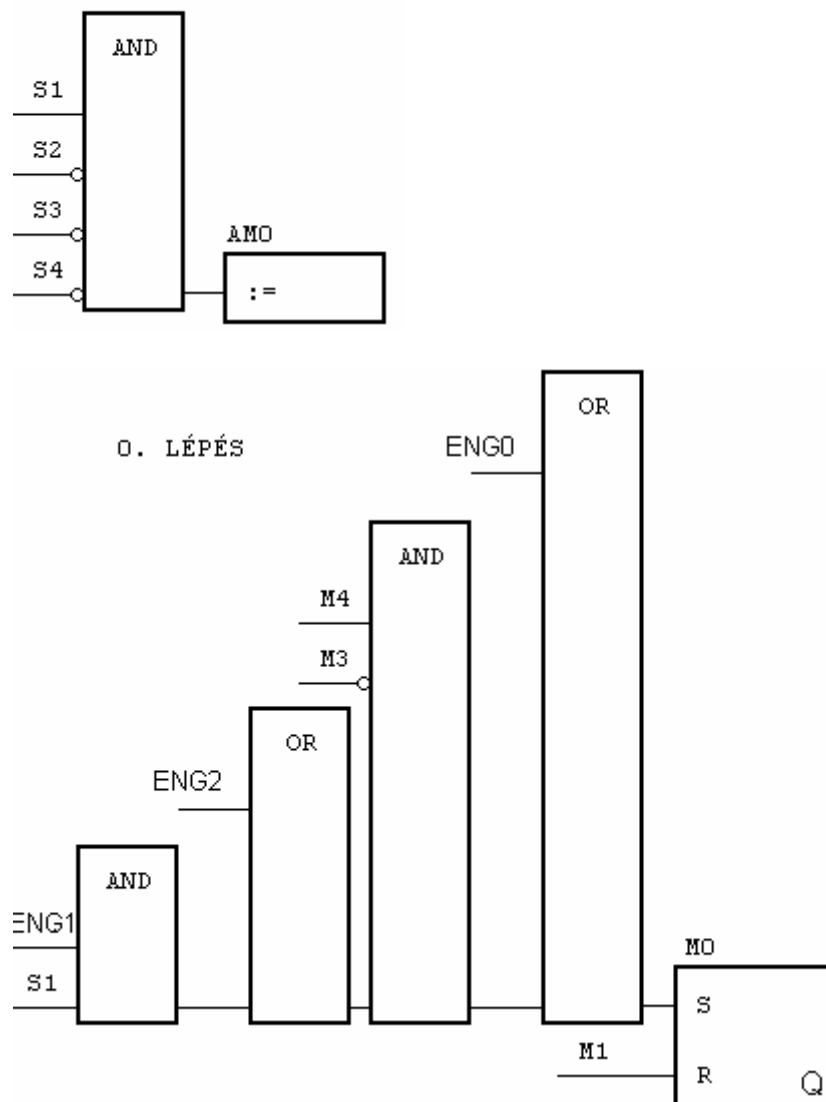
LD  Q4
OR(  I3
ANDN      I1
)
ST  ENG4
END_FUNCTION_BLOCK

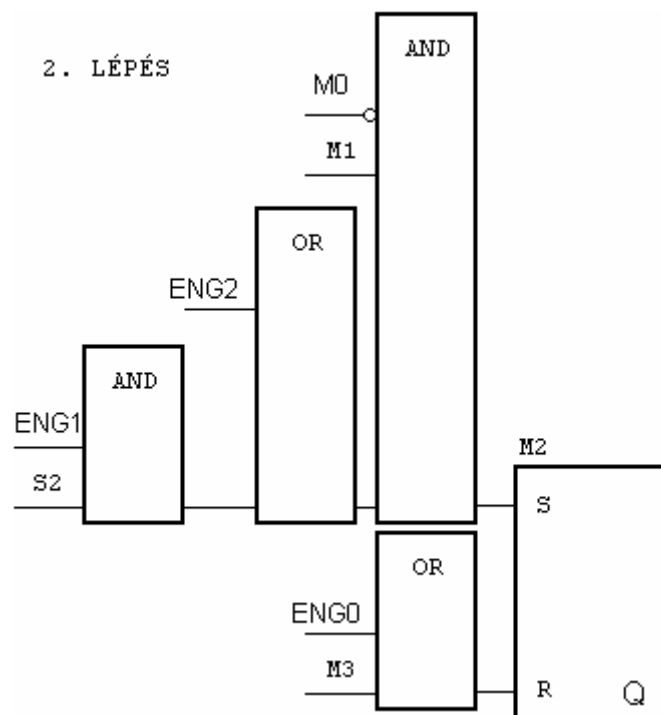
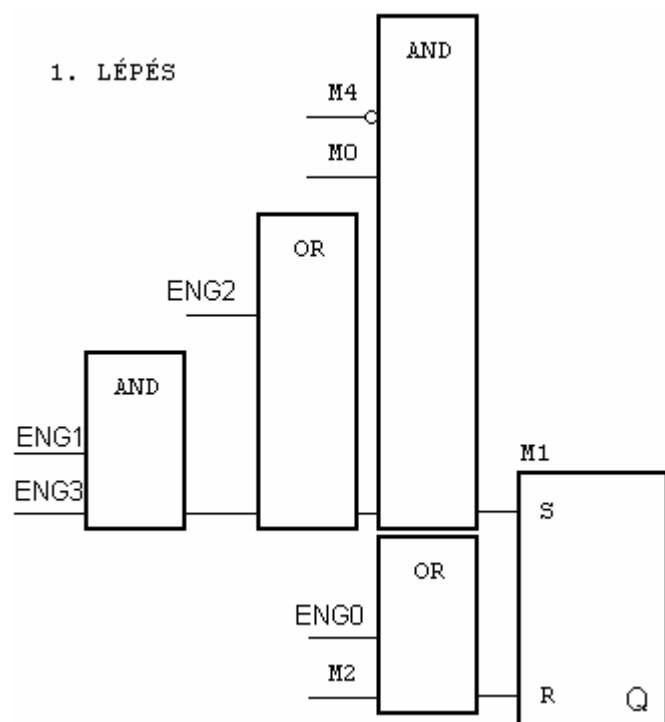
```

## A léptetőlánc függvényblokk

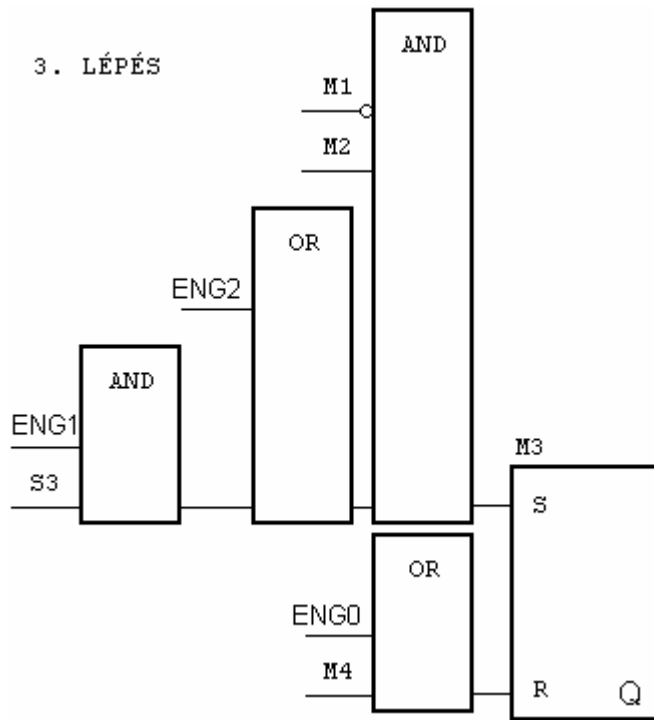
### Funkcióterv

BERENDEZÉS ALAPÁLLAPOTA

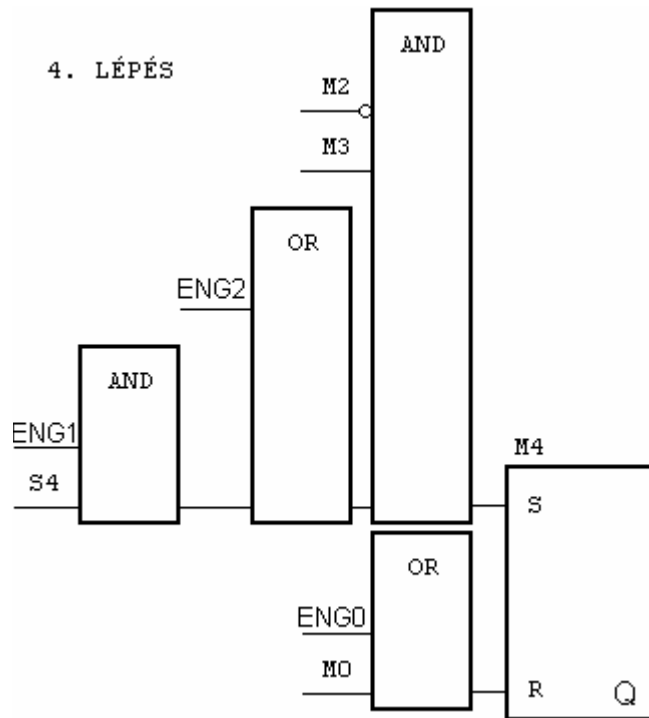




3. LÉPÉS



4. LÉPÉS



**Utasításlista**

FUNCTION\_BLOCK LEPTL

VAR\_INPUT

```

ENG0 : BOOL ;
ENG1 : BOOL ;
ENG2 : BOOL ;
ENG3 : BOOL ;
S1 : BOOL ;
S2 : BOOL ;
S3 : BOOL ;
S4 : BOOL ;

```

END\_VAR

VAR\_EXTERNAL

```

M0 : BOOL ;
M1 : BOOL ;
M2 : BOOL ;
M3 : BOOL ;
M4 : BOOL ;
AM0 : BOOL ;

```

END\_VAR

```

(*BERENDEZÉS
ALAPÁLLAPOTA*)

```

```

LD S1
ANDNS2
ANDNS3
ANDNS4
ST AM0

```

(\*0. LÉPÉS\*)

```

LD ENG0
OR ( M4
ANDNM3
AND ( ENG2
OR ( ENG1
AND S1
)
)
)
S M0

```

(\*1. LÉPÉS\*)

```

LDN M4
AND M0
AND ( ENG2
OR ( ENG1
AND ENG3
)
)
S M1
LD ENG0
OR M2
R M1

```

```

LD M1
R M0

```

(\*2. LÉPÉS\*)

```
LDN M0
AND M1
AND (   ENG2
      OR (   ENG1
          AND S2
        )
      )
S M2
LD ENG0
OR M3
R M2
```

(\*3. LÉPÉS\*)

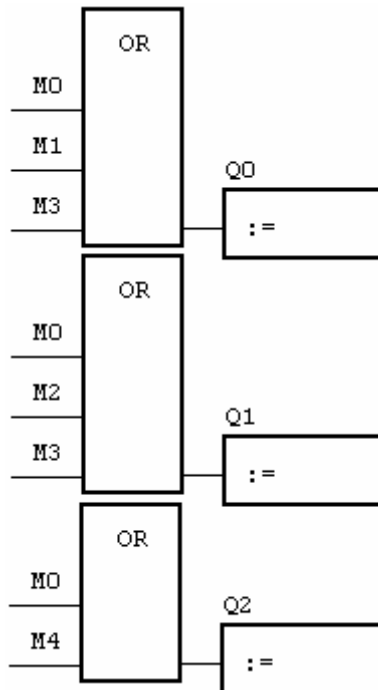
```
LDN M1
AND M2
AND (   ENG2
      OR (   ENG1
          AND S3
        )
      )
S M3
LD ENG0
OR M4
R M3
```

(\*4. LÉPÉS\*)

```
LDN M2
AND M3
AND (   ENG2
      OR (   ENG1
          AND S4
        )
      )
S M4
LD ENG0
OR M0
R M4
END_FUNCTION_BLOCK
```

## A lépkijelzés függvényblokk

### Funkcióterv



### Utasításlista

FUNCTION\_BLOCK LEPKIJ

VAR\_OUTPUT

Q0 : BOOL ;

Q1 : BOOL ;

Q2 : BOOL ;

END\_VAR

VAR\_EXTERNAL

M0 : BOOL ;

M1 : BOOL ;

M2 : BOOL ;

M3 : BOOL ;

M4 : BOOL ;

END\_VAR

LD M0

OR M1

OR M3

ST Q0

LD M0

OR M2

OR M3

ST Q1

LD M0

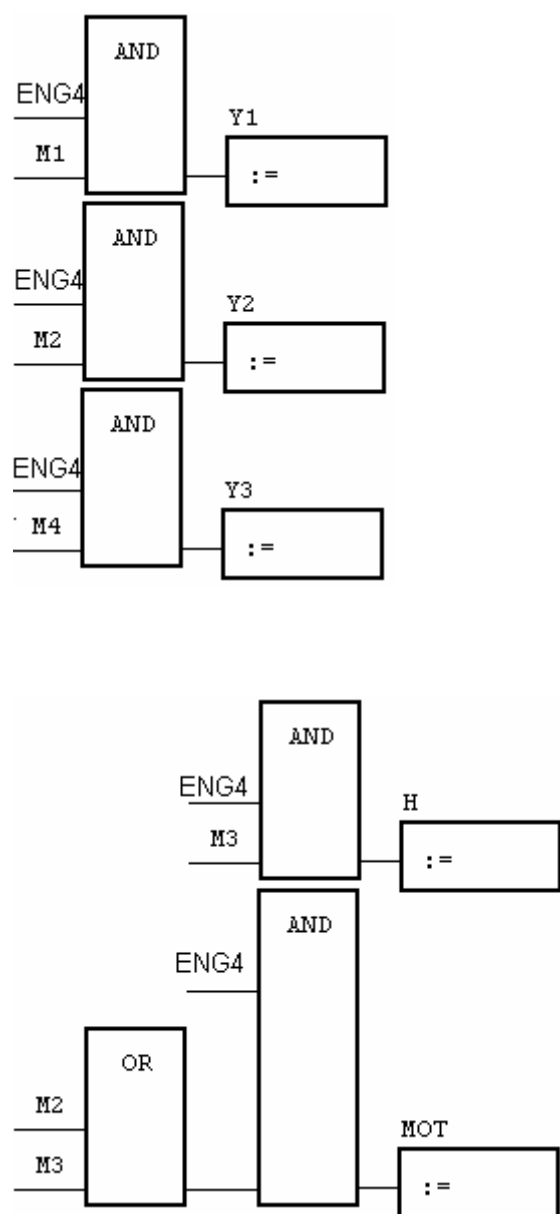
OR M4

ST Q2

END\_FUNCTION\_BLOCK

## A parancskiadás függvényblokk

### Funkcióterv



**Utasításlista**

```
FUNCTION_BLOCK PARKIAD
VAR_INPUT
    ENG4 : BOOL ;
END_VAR
VAR_OUTPUT
    Y1 : BOOL ;
    Y2 : BOOL ;
    Y3 : BOOL ;
    H : BOOL ;
    MOT : BOOL ;
END_VAR
VAR_EXTERNAL
    M1 : BOOL ;
    M2 : BOOL ;
    M3 : BOOL ;
    M4 : BOOL ;
END_VAR

    LD    ENG4
    AND  M1
    ST    Y1

    LD    ENG4
    AND  M2
    ST    Y2

    LD    ENG4
    AND  M4
    ST    Y3

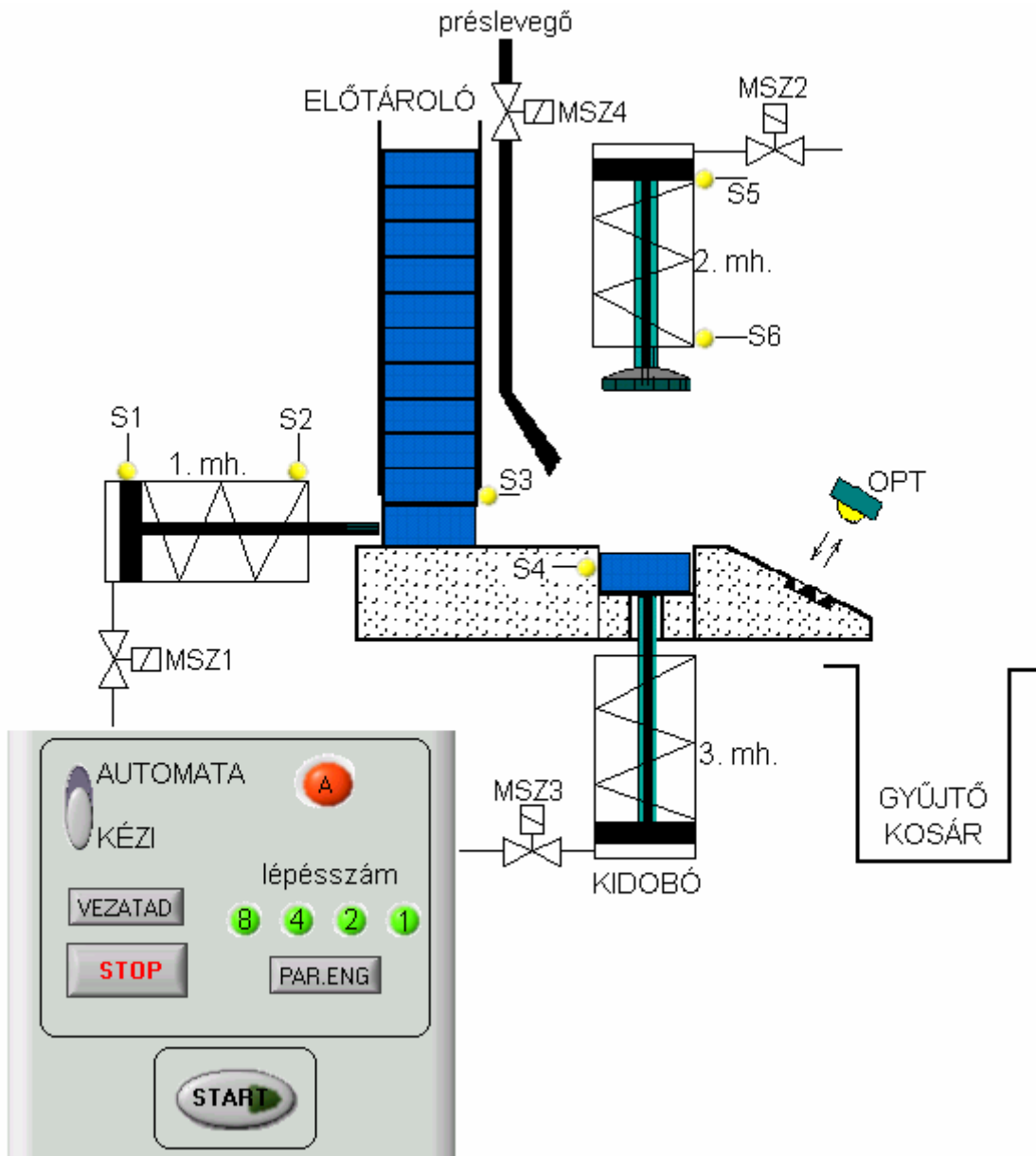
    LD    ENG4
    AND  M3
    ST    H

    LD    ENG4
    AND( M2
    OR   M3
    )
    ST    MOT
END_FUNCTION_BLOCK
```

## Példák lefutóvezérlésekre

### Présgép vezérlése

A présberendezéssel az előtárolóból kivezetett munkadarabokba jelzést préselnek. A pneumatikus munkahengerek úgy vannak kiképezve, hogy csak egyik irányban szükséges a működtető levegő, a másik irányba (alaphelyzetbe) rugó tolja vissza.



7. ábra Présgép a kezelőtáblával

#### A préselési ciklus:

Az előtárolóból (**S3** jelzi, hogy van készenlétben új munkadarab) 1. munkahenger tolattyúja kitolja a munkadarabot a présformába. ha **S4** jelzi, hogy megfelelő pozícióba került a munkadarab, a préselő munkahenger rányomja a présmintát, 2s-ig lenyomva tartja. Ezután **MSZ2** mágnesszelep zár, a rugó visszaviszi eredeti pozíciójába a préskart (**S5** jelez). A kidobó

munkahenger és a terelőlevegő segítségével a munkadarab a gyűjtőkosárba kerül. Ha az **OPT** érzékelő jelzi, hogy a munkadarab áthaladt, **MSZ3** és **MSZ4** mágnesszelepek zárnak, és indulhat a folyamat előről.

A préselés automata üzemmódban a **START** nyomógomb egyszeri lenyomására indul, és mindaddig fut ciklikusan, amíg van munkadarab az előtárolóban (ismételt végrehajtás engedélyezése!). (Az automata üzemmód természetesen a **STOP** gomb benyomásával a folyamatban lévő ciklus végén, a **Kézi** átkapcsolással pedig azonnal megszüntethető.)

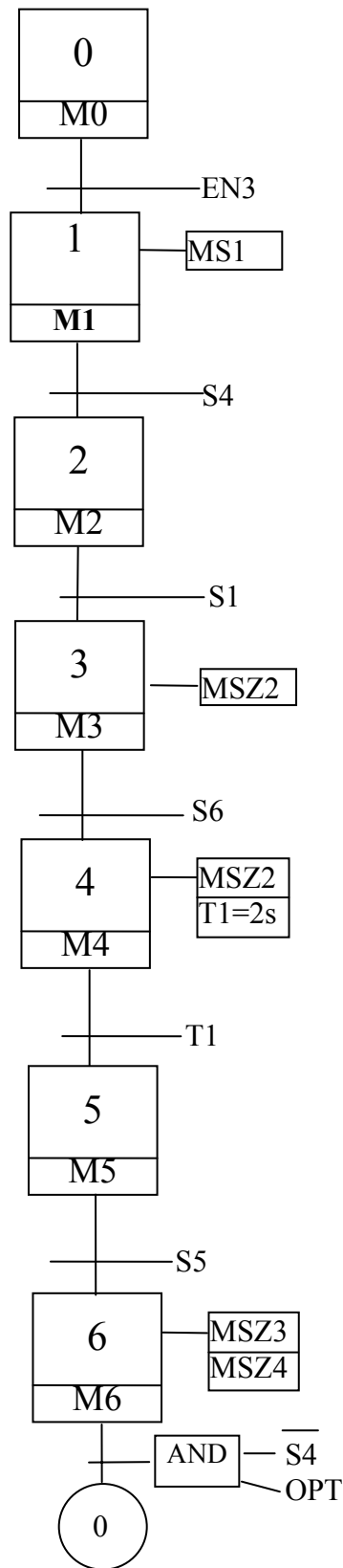
### Összerendelési táblázat

A táblázatban felsoroljuk a technológiai jelek mellett a kezelői felület jeleit is.

Bemenetek	Jel	Logikai összerendelés	Cím
START	START	benyomva: START=1	I1.4
1. mh. hátsó pozíció	S1	jelez, ha: S1=1	I0.0
1. mh. első pozíció	S2	jelez, ha: S2=1	I0.1
munkadarab az előtárolóban	S3	jelez, ha: S3=1	I0.2
munkadarab a pészformában	S4	jelez, ha: S4=1	I0.3
2. mh. hátsó pozíció	S5	jelez, ha: S5=1	I0.4
2. mh. első pozíció	S6	jelez, ha: S6=1	I0.5
optoérzékelő	OPT	a fényút megszakad: OPT=1	I0.6
Automata/kézi átkapcsoló	KAUT	automata: KAUT=1	I1.0
vezérlés átadás nyomógomb	VATAD	benyomva: VATAD=1	I1.1
parancsengedélyezés	PENG	benyomva: PENG=1	I1.2
STOP nyomógomb	STOP	benyomva: STOP=0	I1.3

<b>Kimenetek</b>	<b>Jel</b>	<b>Logikai összerendelés</b>	<b>Cím</b>
1. mh. mágnesszelep	MSZ1	nyitva, ha: MSZ1=1	Q0.0
2. mh. mágnesszelep	MSZ2	nyitva, ha: MSZ2=1	Q0.1
3. mh. mágnesszelep	MSZ3	nyitva, ha: MSZ3=1	Q0.2
4. mh. mágnesszelep	MSZ4	nyitva, ha: MSZ4=1	Q0.3
lépéskijelző LED	LEP0	világít, ha: LEP0=1	Q1.0
lépéskijelző LED	LEP1	világít, ha: LEP1=1	Q1.1
lépéskijelző LED	LEP2	világít, ha: LEP2=1	Q1.2
lépéskijelző LED	LEP3	világít, ha: LEP3=1	Q1.3
automata üzemmód jelzés	QAUT	világít, ha: QAUT=1	Q1.4

**Léptetőlánc**



A léptetőlánc funkciótervbe való átírásakor figyelembe vesszük az üzemmód függvényblokk engedélyező jeleit is. Az üzemmód függvényblokkot változatlanul átvehetjük az előző feladatból. A lépéskijelzést ki kell bővítenünk 6 lépésszám kijelzésére. A léptetőláncot és a parancskiadást meg kell feleltetnünk az új technológiának, a főprogramban pedig deklarálnunk kell a megfelelő ki/bemeneti jeleket. A főprogramban a függvényblokkok hívását és az adatátadást aktualizálni kell.

A vezérlés függvényblokkjai	prototípus	feladata
üzemmód	UMODFB	üzemmód beállítása, engedélyező jelek
léptetőlánc	LEPTET	meghatározza a berendezés alapállapotát és beállítja a lépésmerkereket
lépéskijelzés	PRESLEP	a lépéskijelző LED-ek beállítása
parancskiadás	PRESPAR	a kimenetek beállítása

### A főprogram

A főprogramot utasításlistában adjuk meg.

### A változódeklaráció:

PROGRAM PRESPR

VAR

```

KAUT AT %I0.0.0.1.0:   BOOL;      (* =1 :AUTO =0 :KÉZI *)
VATAD AT %I0.0.0.1.1:  BOOL;      (* =1 BENYOMVA *)
PENG AT %I0.0.0.1.2:   BOOL;      (* =1 BENYOMVA *)
STOP AT %I0.0.0.1.3:   BOOL;      (* =0 BENYOMVA *)
START AT %I0.0.0.1.4:  BOOL;      (* =1 BENYOMVA *)
QAUT AT %Q0.0.0.1.0:   BOOL;
LEP1 AT %Q0.0.0.1.1:   BOOL;      (* LEPESkijelzés *)
LEP2 AT %Q0.0.0.1.2:   BOOL;      (* LEPESkijelzés *)
LEP3 AT %Q0.0.0.1.3:   BOOL;      (* LEPESkijelzés *)
EN0:  BOOL;          (* indító impulzus *)
EN1:  BOOL;          (* feltételes továbblépés engedélyezése *)
EN2:  BOOL;          (* feltétel nélküli továbblépés engedélyezése *)
EN3:  BOOL;          (* 0-1 lépés engedélyezése *)
EN4:  BOOL;          (* parancsengedélyezés *)
S1 AT %I0.0.0.0.0:     BOOL;      (* 1.SZINT *)
S2 AT %I0.0.0.0.1:     BOOL;      (* 2.SZINT *)
S3 AT %I0.0.0.0.2:     BOOL;      (* 3.SZINT *)
S4 AT %I0.0.0.0.3:     BOOL;      (* HOMERSEKLET *)
S5 AT %I0.0.0.0.4:     BOOL;
S6 AT %I0.0.0.0.5:     BOOL;
OPT AT %I0.0.0.0.6:    BOOL;
MSZ1 AT %Q0.0.0.0.0:   BOOL;      (* 1.SZELEP *)
MSZ2 AT %Q0.0.0.0.1:   BOOL;      (* 2.SZELEP *)
MSZ3 AT %Q0.0.0.0.2:   BOOL;      (* 3.SZELEP *)
MSZ4 AT %Q0.0.0.0.3:   BOOL;      (* FUTES *)
UZEM:    UMODFB;
LANC:    LEPTETP;
KIJEZLZ: PRESLEP;

```

```

      KIMENET:  PRESPAR;
END_VAR

```

```

VAR_GLOBAL
  AM0:  BOOL; (* berendezés alapállapota *)
  M0:   BOOL; (* 0. lépés *)
  M1:   BOOL; (* 1.lépés *)
  M2:   BOOL; (* 2.lépés *)
  M3:   BOOL; (* 3.lépés *)
  M4:   BOOL; (* 4.lépés *)
  M5:   BOOL; (* 5.lépés *)
  M6:   BOOL; (* 6.lépés *)
END_VAR

```

**programtörzs:**

```

LD    KAUT
ST    UZEM.I1
LD    VATAD
ST    UZEM.I2
LD    PENG
ST    UZEM.I3
LD    STOP
ST    UZEM.I4
LD    START
ST    UZEM.S0
LD    AM0
ST    UZEM.AM0
LD    M0
ST    UZEM.M0

CAL   UZEM

LD    UZEM.Q4
ST    QAUT
LD    UZEM.ENG0
ST    EN0
LD    UZEM.ENG1
ST    EN1
LD    UZEM.ENG2
ST    EN2
LD    UZEM.ENG3
ST    EN3
LD    UZEM.ENG4
ST    EN4

CAL   LANC(ENG0:=EN0,ENG1:=EN1,
          ENG2:=EN2,ENG3:=EN3,
          S1:=S1,S2:=S2,
          S3:=S3,S4:=S4,
          S5:=S5,S6:=S6,
          OPT:=OPT)

CAL   KIJELZ

```

```

LD   KIJELZ.Q0
ST   LEP1
LD   KIJELZ.Q1
ST   LEP2
LD   KIJELZ.Q2
ST   LEP3

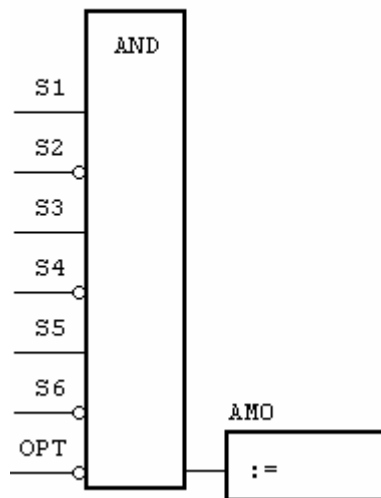
CAL  KIMENET(ENG4:=EN4)

LD   KIMENET.Y1
ST   MSZ1
LD   KIMENET.Y2
ST   MSZ2
LD   KIMENET.Y3
ST   MSZ3
LD   KIMENET.Y4
ST   MSZ4
END_PROGRAM

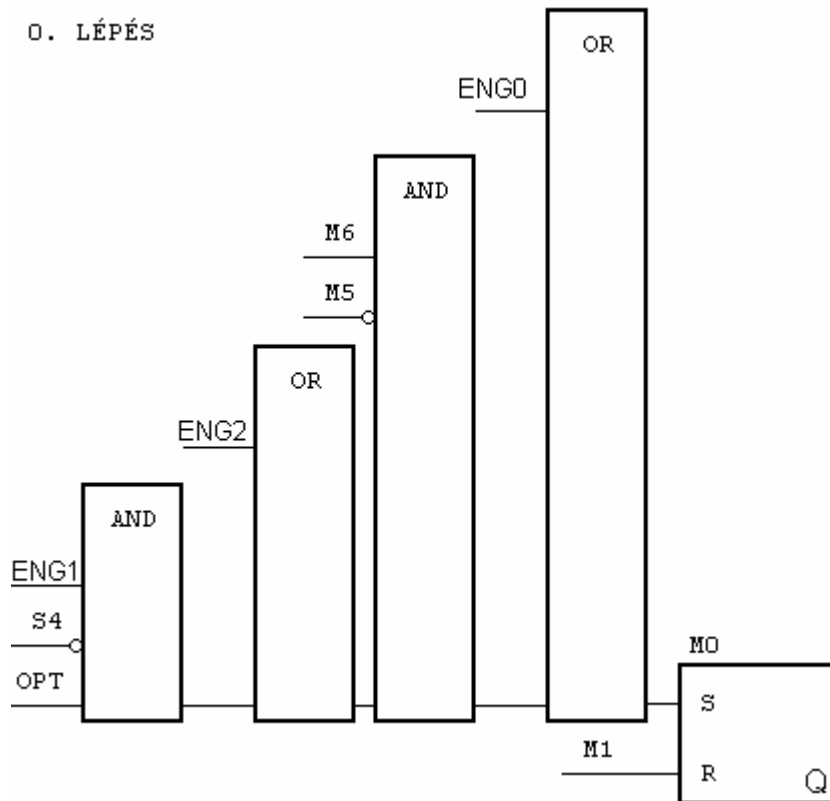
```

### A léptetőlánc függvényblokk funkciótervben

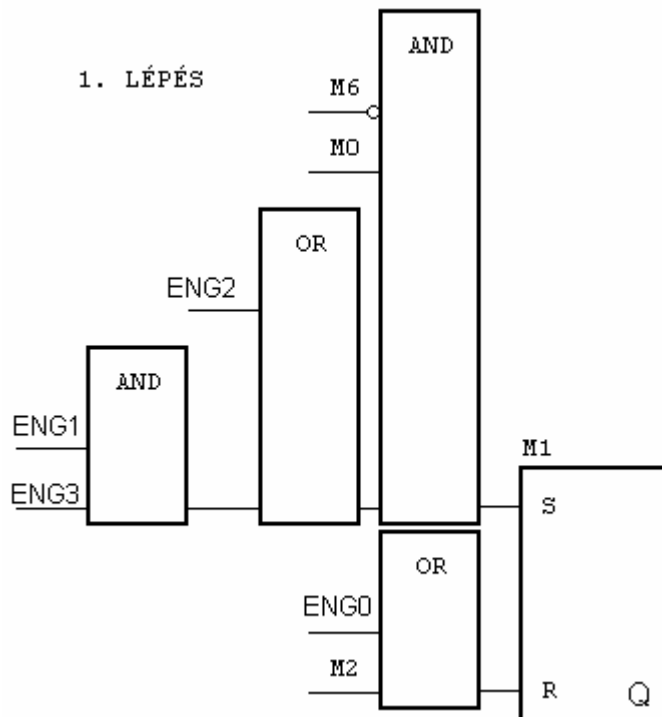
BERENDEZÉS ALAPÁLLAPOTA



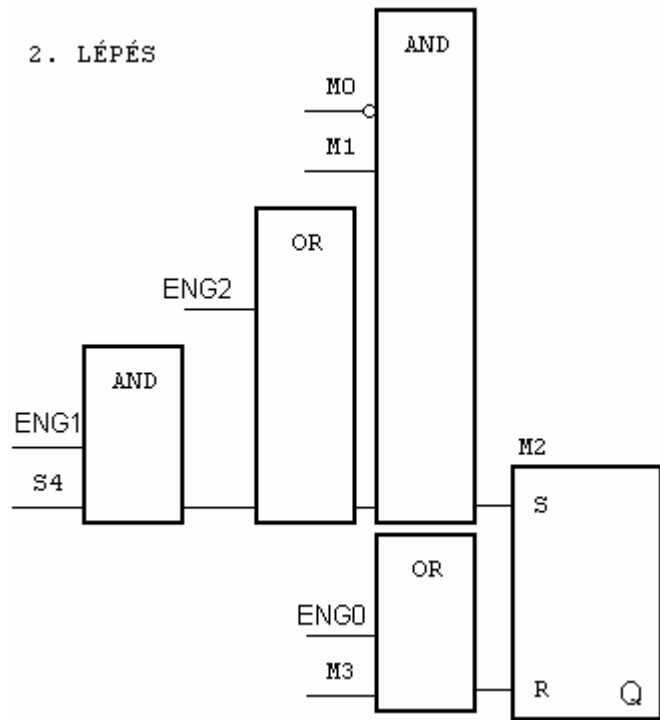
0. LÉPÉS



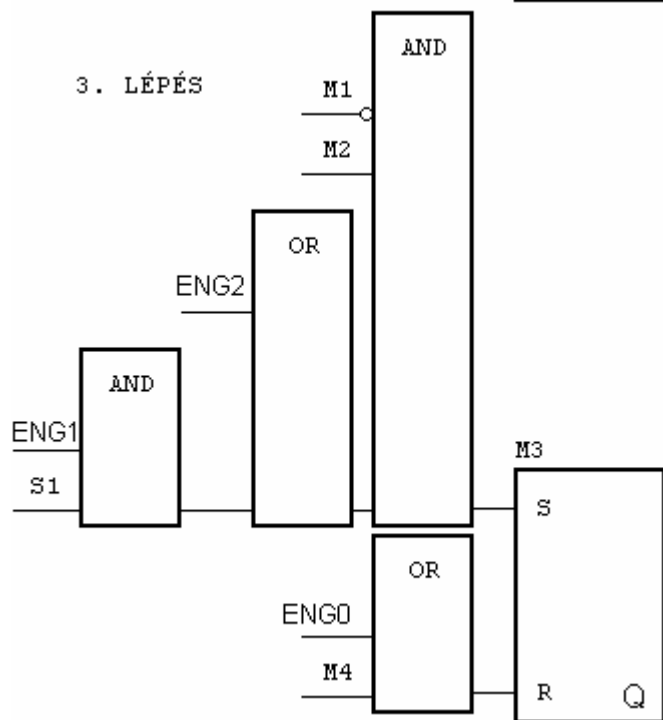
1. LÉPÉS

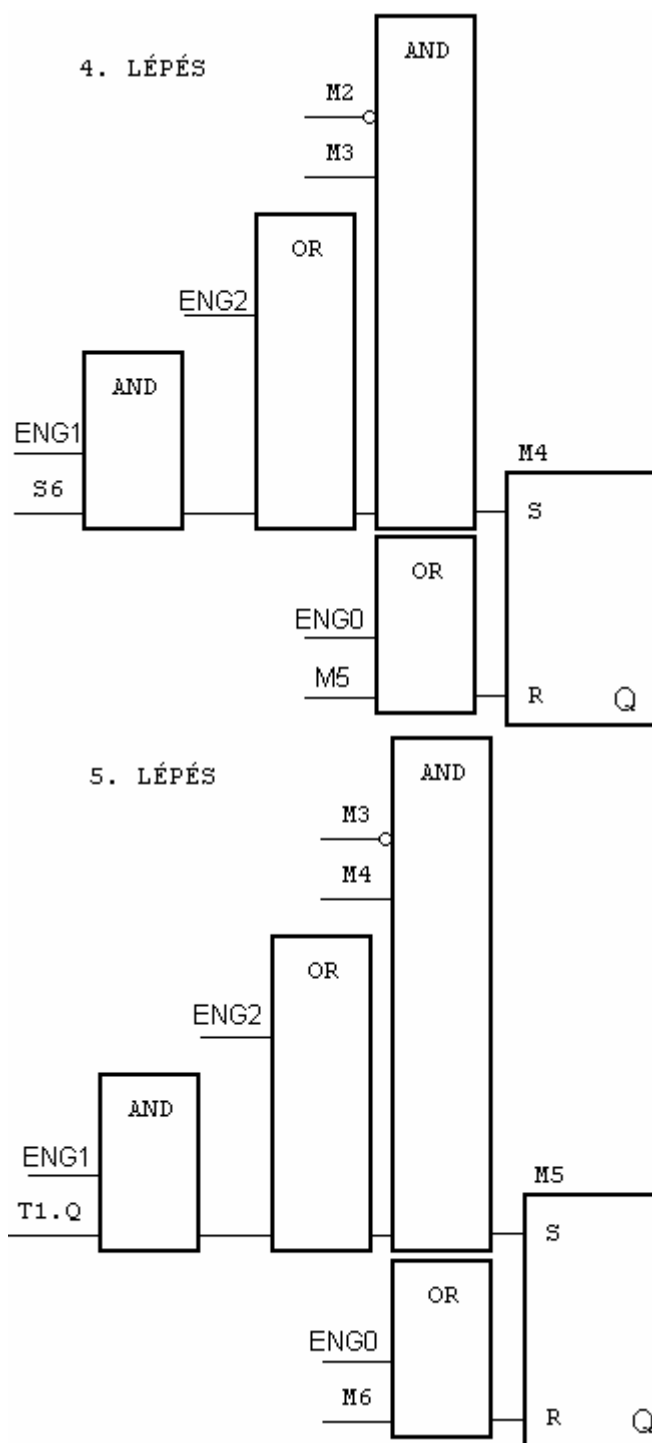


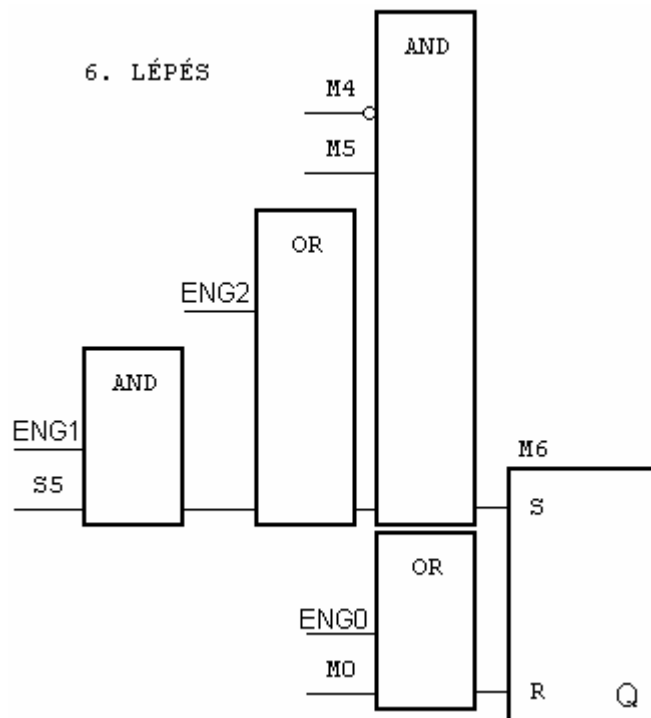
2. LÉPÉS



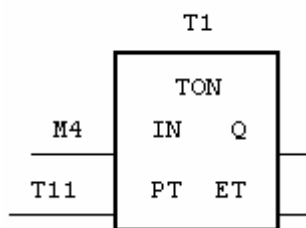
3. LÉPÉS







IDŐZÍTŐ



### Utasításlistában

FUNCTION\_BLOCK LEPTETP

VAR\_INPUT

ENG0: BOOL;  
 ENG1: BOOL;  
 ENG2: BOOL;  
 ENG3: BOOL;  
 S1: BOOL;  
 S2: BOOL;  
 S3: BOOL;  
 S4: BOOL;  
 S5: BOOL;  
 S6: BOOL;  
 OPT: BOOL;

END\_VAR

VAR

T1: TON;  
 T11: TIME := T#2s;

END\_VAR

```

VAR_EXTERNAL
  AM0: BOOL;
  M0:  BOOL;
  M1:  BOOL;
  M2:  BOOL;
  M3:  BOOL;
  M4:  BOOL;
  M5:  BOOL;
  M6:  BOOL;
END_VAR

```

(\*BERENDEZÉS ALAPÁLLAPOTA\*)

```

LD   S1
ANDNS2
AND  S3
ANDNS4
AND  S5
ANDNS6
ANDNOPT
ST   AM0

```

(\*0. LÉPÉS\*)

```

LD   ENG0
OR   (   M6
      ANDNM5
      AND (   ENG2
            OR (   ENG1
                  ANDNS4
                  AND OPT
                )
            )
      )
S    M0

```

```

LD   M1
R    M0

```

(\*1. LÉPÉS\*)

```

LDN  M6
AND  M0
AND  (   ENG2
      OR (   ENG1
            AND ENG3
          )

```

```
)  
S M1  
LD ENG0  
OR M2  
R M1
```

(\*2. LÉPÉS\*)

```
LDN M0  
AND M1  
AND ( ENG2  
      OR ( ENG1  
          AND S4  
        )  
      )  
S M2  
LD ENG0  
OR M3  
R M2
```

(\*3. LÉPÉS\*)

```
LDN M1  
AND M2  
AND ( ENG2  
      OR ( ENG1  
          AND S1  
        )  
      )  
S M3  
LD ENG0  
OR M4  
R M3
```

(\*4. LÉPÉS\*)

```
LDN M2  
AND M3  
AND ( ENG2  
      OR ( ENG1  
          AND S6  
        )  
      )  
S M4  
LD ENG0  
OR M5  
R M4
```

(\*5. LÉPÉS\*)

```
LDN M3
AND M4
AND (   ENG2
      OR (   ENG1
          AND T1.Q
        )
      )
S M5
LD ENG0
OR M6
R M5
```

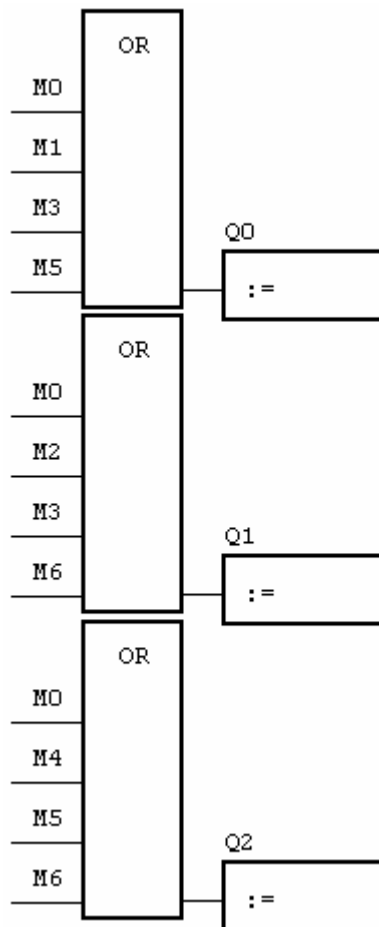
(\*6. LÉPÉS\*)

```
LDN M4
AND M5
AND (   ENG2
      OR (   ENG1
          AND S5
        )
      )
S M6
LD ENG0
OR M0
R M6
```

(\*IDŐZÍTŐ\*)

```
LD M4
ST T1.IN
LD T11
ST T1.PT
CAL T1
END_FUNCTION_BLOCK
```

## A lépéskijelzés függvényblokk funkciótervben



### Utasításlistában

FUNCTION\_BLOCK PRESLEP

VAR\_OUTPUT

Q0: BOOL;

Q1: BOOL;

Q2: BOOL;

END\_VAR

VAR\_EXTERNAL

M0: BOOL;

M1: BOOL;

M2: BOOL;

M3: BOOL;

M4: BOOL;

M5: BOOL;

M6: BOOL;

END\_VAR

LD M0

OR M1

OR M3

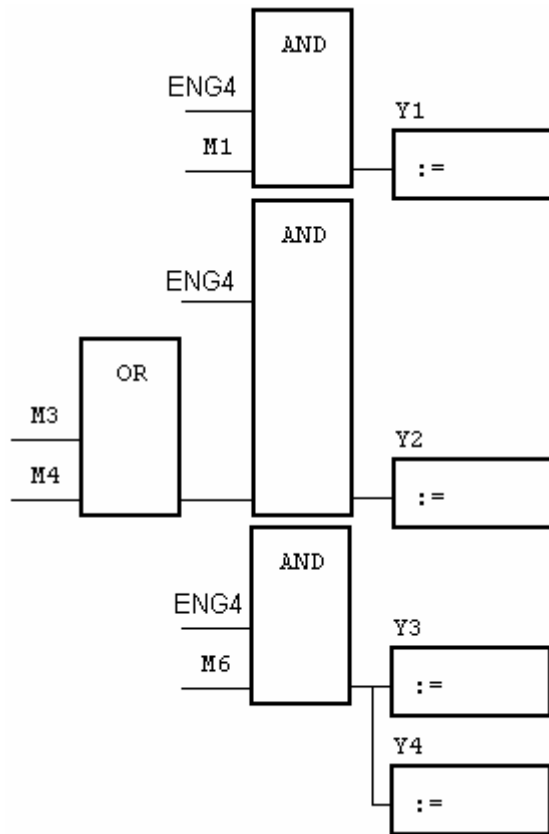
OR M5

ST Q0

```
LD M0
OR M2
OR M3
OR M6
ST Q1
```

```
LD M0
OR M4
OR M5
OR M6
ST Q2
END_FUNCTION_BLOCK
```

**A parancskiadás függvényblokk funkciótervben**



**Utasításlistában**

```
FUNCTION_BLOCK PRESPAR
VAR_INPUT
    ENG4 : BOOL ;
END_VAR
VAR_OUTPUT
    Y1 : BOOL ;
    Y2 : BOOL ;
    Y3 : BOOL ;
    Y4 : BOOL ;
```

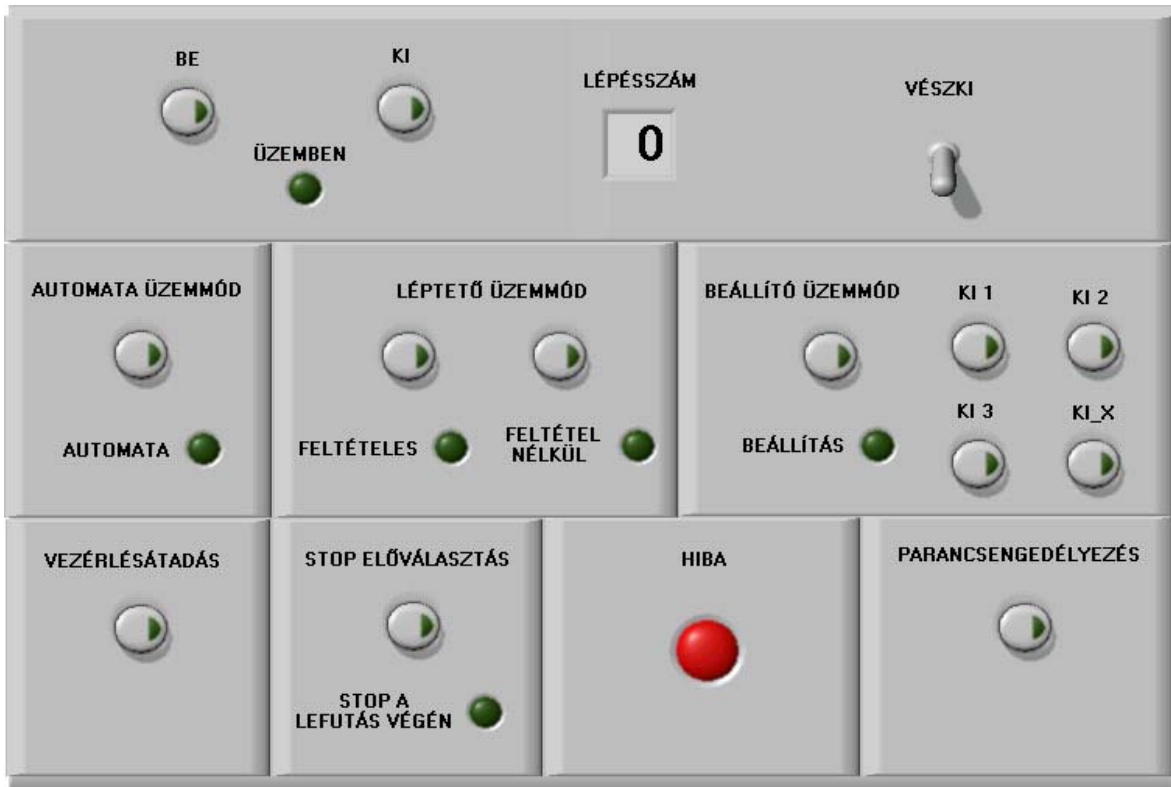
```
END_VAR
VAR_EXTERNAL
    M1 : BOOL ;
    M2 : BOOL ;
    M3 : BOOL ;
    M4 : BOOL ;
    M5 : BOOL ;
    M6 : BOOL ;
END_VAR
LD    ENG4
AND  M1
ST   Y1

LD    ENG4
AND( M3
OR   M4
)
ST   Y2

LD    ENG4
AND  M6
ST   Y3
ST   Y4
END_FUNCTION_BLOCK
```

### Kezelői felület VÉSZKI-kapcsolóval, többféle üzemmód választásának lehetőségével

Elképzelhető, hogy egy szakaszos üzemű technológia vezérléséhez az előző példákban alkalmazott kezelői felület nem elégséges. Összetettebb technológiáknál biztosítanunk kell a be/ki kapcsolás mellett a vészki kapcsolás lehetőségét is. Az automata üzemmódon kívül feltételes és feltétel nélküli kézi léptető üzemmód is igény lehet. A berendezés alapállapotának beállításához, vészleállás utáni beavatkozáshoz szükséges a beállító üzemmód. A technológia felől érkező hibajelzések, időtúllépések is leállíthatják az automata üzemmódot, hibajelzést adva. Egy ilyen kezelői felület egy lehetséges megvalósítása látható az alábbi ábrán.



8. ábra Kezelői felület

A megváltozott kezelői felület az üzemmód függvényblokk módosítását is megkívánja. Az alábbi utasításlista példa egy lehetséges megoldásra. A függvényblokkot úgy írtuk meg, hogy az engedélyező jelek, vagyis az illeszkedés a léptetőlánc és a parancskiadás függvényblokk felé változatlan marad, de alkalmassá tettük a kezelői felület felől érkező, megváltozott számú és funkciójú jelek fogadására illetve kiadására.

#### A függvényblokk utasításlistája

(Az előző feladatok üzemmód függvényblokkját az új függvényblokkra cserélve, azok az új kezelői felületről vezérelhetők. Természetesen a lépésszám-kijelzést a számjegyes kijelzőnek megfelelően módosítanunk kell.)

```
FUNCTION_BLOCK UMDOSSZ
VAR_INPUT
```

```

I1:   BOOL; (* VÉSZKI *)
I2:   BOOL; (* BE/KI, BE=1 *)
I3:   BOOL; (* AUTOMATA, BENYOMNVA=1 *)
I4:   BOOL; (* FELTÉTELES LÉPTETŐ ÜM, BENYOMNVA=1 *)
I5:   BOOL; (* FELTÉTEL NÉLKÜLI LÉPTETŐ ÜM., BENYOMNVA=1 *)
```

```

I6:    BOOL; (* BEÁLLÍTÓ ÜM., BENYOMNVA=1 *)
I7:    BOOL; (* VEZÉRLÉSÁTADÁS, BENYOMNVA=1 *)
I8:    BOOL; (* STOP, BENYOMNVA=1 *)
I9:    BOOL; (* PARANCS ENGEDÉLYEZÉS, BENYOMNVA=1 *)
AM0:   BOOL; (* BV: BERENDEZÉS ALAPÁLLAPOTA *)
AM1:   BOOL; (* BV: HIBA A TECHNOLÓGIÁBÓL *)
M0:    BOOL; (* 0. LÉPÉS MERKERE *)
SX:    BOOL; (* HIBAJEL A TECHNOLÓGIÁBÓL *)
TX:    BOOL; (* IDŐTÚLLÉPÉS A TECHNOLÓGIÁBÓL *)
END_VAR
VAR_OUTPUT
  Q1:   BOOL; (* ÜZEMEL LED *)
  Q2:   BOOL; (* AUTOMATA ÜZEMMÓD LED *)
  Q3:   BOOL; (* FELTÉTELES LÉPTETŐ ÜZEMMÓD LED *)
  Q4:   BOOL; (* FELTÉTEL NÉLKÜLI LÉPTETŐ ÜM. LED *)
  Q5:   BOOL; (* BEÁLLÍTÓ ÜZEMMÓD LED *)
  Q6:   BOOL; (* STOP ELŐJELZÉS LED *)
  Q7:   BOOL; (* HIBA LED *)
  Q8:   BOOL;
  B0:   BOOL; (* INDÍTÓ IMPULZUS *)
  B1:   BOOL; (* FELTÉTELES TOVÁBBLÉPÉS ENGEDÉLYEZÉSE *)
  B2:   BOOL; (* FELTÉTEL NÉLKÜLI TOVÁBBLÉPÉS ENGEDÉLYEZÉSE *)
  B3:   BOOL; (* 0-1 LÉPÉS ENGEDÉLYEZÉSE *)
  B4:   BOOL; (* KIMENET-ENGEDÉLYEZÉS *)
END_VAR
VAR
  B10:  BOOL;
  B11:  BOOL;
  B12:  BOOL;
  B13:  BOOL;
  B14:  BOOL;
  B15:  BOOL;
  B16:  BOOL;
END_VAR

```

(\*BE/KIKAPCSOLÁS IMPULZUS FELFUTÓ ÉLRE\*)

```

LD    I2
ANDN  B11
ST    B10

```

```

LD    I2
ST    B11

```

(\*ÜZEMEL KIJEZÉS\*)

LD B10  
S Q1

LDN I2  
ORN I1  
OR AM1  
R Q1

(\*AUTOMAT ÜZEM KIJEZÉS\*)

LD I3  
S Q2

LDN Q1  
OR Q3  
OR Q4  
OR Q5  
OR B12  
R Q2

(\*FELTÉTELES LÉPTETÉS ÜZEM KIJEZÉS\*)

LD I4  
S Q3

LDN Q1  
OR I5  
OR Q5  
OR B12  
R Q3

(\*FELTÉTEL NÉLKÜLI LÉPTETÉS ÜZEM KIJEZÉS\*)

LD I5  
S Q4

LDN Q1  
OR I4  
OR Q5  
OR B12  
R Q4

(\*STOP ELŐVÁLASZTÁS VISSZAJELZÉS\*)

LD I8  
S Q6

LDN Q2  
ANDN Q3  
ANDN Q4  
ANDN Q5  
R Q6

LD Q6  
AND M0  
ST B12

(\*INDÍTÓ IMPULZUS\*)

LD Q1

ANDN B13  
ST B0

LD B0  
S B13

LDN Q1  
R B13

(\*VEZÉRLÉSÁTADÁS IMPULZUS FELFUTÓ ÉLRE\*)

LD I7  
ANDN B15  
ST B14

LD I7  
ST B15

(\*FELTÉTELES TOVÁBBLÉPTETÉS ENGEDÉLYEZÉSE\*)

LD Q2  
OR ( Q3  
AND B14  
)  
ST B1

(\*FELTÉTEL NÉLKÜLI TOVÁBBLÉPTETÉS ENGEDÉLYEZÉSE\*)

LD Q4  
AND B14  
ST B2

(\*START-FELTÉTEL 0-1 LÉPTETÉS\*)

LD I7  
S B16

```
LDN  Q2
ANDN Q3
OR   I8
R    B16
```

```
LD   AM0
AND  B1
AND  B16
ST   B3
```

(\*PARANCSENGEDÉLYEZÉS\*)

```
LD   Q2
OR   (   I9
      AND (   Q3
            OR  Q4
            )
      )
ST   B4
```

(\*HIBAJELZÉS\*)

```
LD   Q2
AND  TX
OR   (   SX
      NOT AND (   Q2
                OR  Q3
                OR  Q4
                )
      )
S    AM1
```

```
LD   I7
R    AM1
```

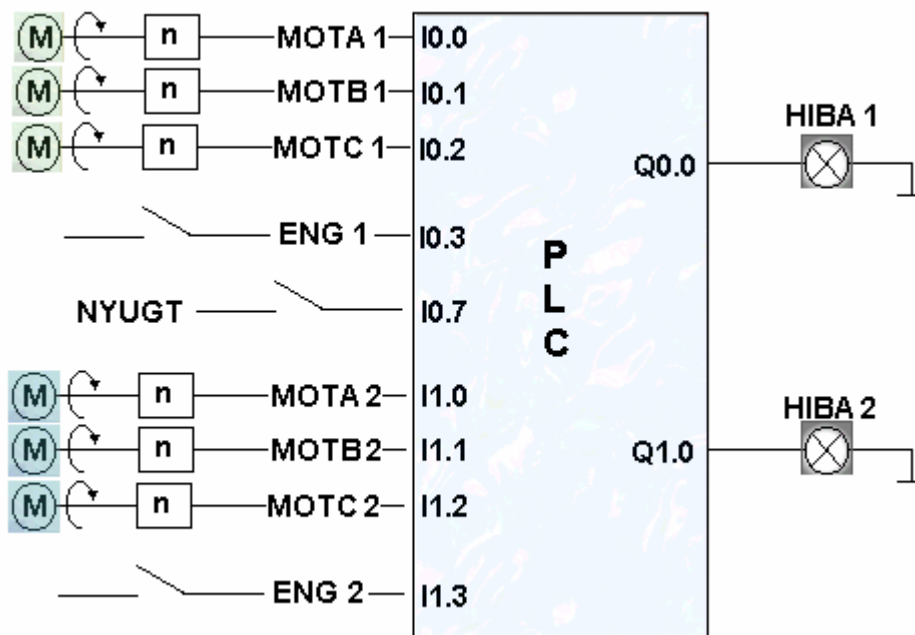
```
LD   AM1
ST   Q7
END_FUNCTION_BLOCK
```

**Feladat:** Írja át a fenti utasításlistát funkciótervbe és elemezze a működését!

## Digitális vezérlések

### Saját készítésű függvényblokk: Motorblokkok felügyelete

A feladat: 2 db, A,B,C motorból álló motorblokk felügyelete. Minden motorhoz tartozik egy fordulatszámjelző, amely folyamatos 1 jelet ad, ha a motor működik. Ha a motor leáll, az érzékelő-távadó kimenete 0-ra vált. A felügyelet akkor indul, vagyis a vezérlésnek akkor kell figyelni a motorok működését, ha a motorcsoporthoz tartozó engedélyező jelet a kezelőszemély bekapcsolta.



9. ábra Kettő, egyenként 3 db motorból álló motorblokk felügyelete

#### A hibajelzés feltételei:

1. eset: a hátromból két motor 5 s-nál hosszabb ideig leáll. (Időkésleltetett jelzés).
2. eset: mindhárom motor leáll. (Azonnali jelzés).

#### A hibajelzés megszüntetése:

1. eset: a hibajelzés magától megszűnik, ha valamelyik motor újra indul, azaz ismét legalább két motor fut..
2. eset: a kezelőnek a hiba elhárítása után meg kell nyomnia a nyugtázó gombot is ahhoz, hogy a hibajelzés megszűnjön.

**Megoldás:** mivel minkét motorbloknál ugyanazt a feladatot kell megoldani, ezt függvényblokkban írjuk meg, és a két motorcsoporthoz külön-külön egyedi névvel deklaráljuk. A főprogramban kell gondoskodni a be/kimenőjelek fizikai címekhez rendeléséről és a függvényblokkok aktuális paraméterekkel történő hívásáról.

**Összerendelési táblázat**

<b>Bemenetek</b>	<b>Jel</b>	<b>Logikai összerendelés</b>	<b>Cím</b>
A1 motor fordulatszámjelző	MOTA1	A1 motor fut: MOTA1=1	I0.0
A2 motor fordulatszámjelző	MOTA2	A2 motor fut: MOTA2=1	I1.0
B1 motor fordulatszámjelző	MOTB1	B1 motor fut: MOTB1=1	I0.1
B2 motor fordulatszámjelző	MOTB2	B2 motor fut: MOTB2=1	I1.1
C1 motor fordulatszámjelző	MOTC1	C1 motor fut: MOTC1=1	I0.2
C2 motor fordulatszámjelző	MOTC2	C2 motor fut: MOTC2=1	I1.2
1. csop. engedélyezés kapcsoló	ENG1	1. blokk bekakcsolva: ENG1=1	I0.3
2. csop. engedélyezés kapcsoló	ENG2	2. blokk bekapcsolva: ENG2=1	I1.3
NYUGTÁZÁS nyomógomb	NYUGT	benyomva: NYUGT=1	I0.7
<b>Kimenetek</b>			
HIBAJELZÉS 1	HIBA1	világít, ha: HIBA1=1	Q0.0
HIBAJELZÉS 2	HIBA2	világít, ha: HIBA2=1	Q1.0

**A függvényblokk formális paraméterei**

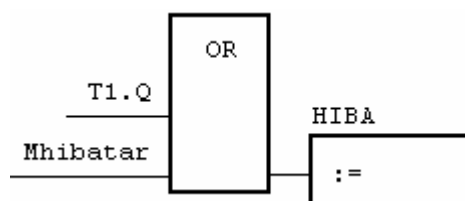
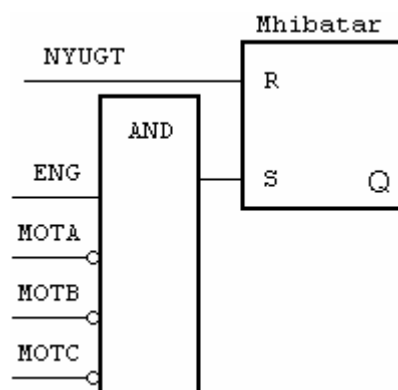
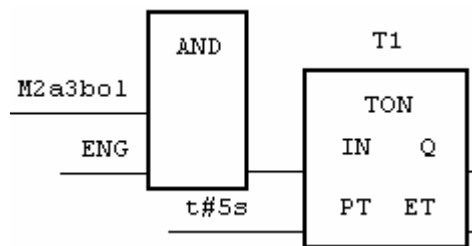
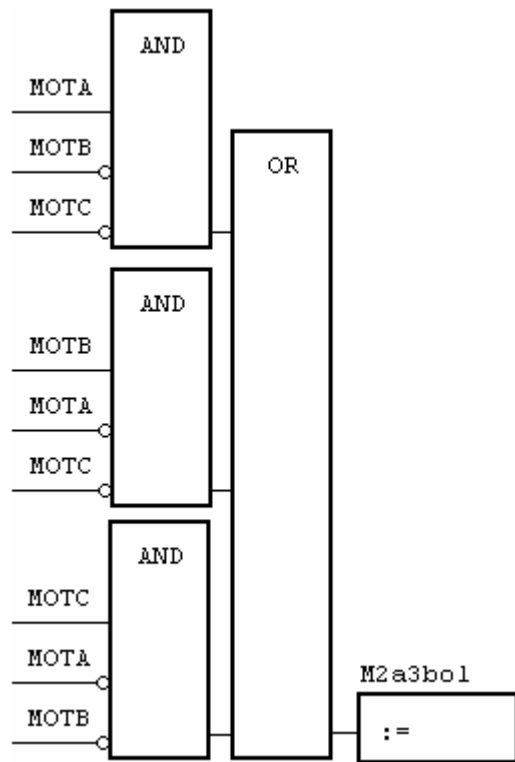
A feladatból (input, output): **MOTA, MOTB, MOTC, ENG, HIBA.**

Elrejtett (lokális): T1 időzítő a késleltetéshez: (TON)

3 motor meghibásodás átmeneti tároló: **Mhibatar**

3-ból 2 kiesést jelző segédmerker: **M2a3bol.**

## A függvényblokk funkciótervben



## Utasításlista

```

FUNCTION_BLOCK fordsz
VAR_INPUT
    MOTA : BOOL ;
    MOTB : BOOL ;
    MOTC : BOOL ;
    ENG : BOOL ;
    NYUGT : BOOL ;
END_VAR
VAR_OUTPUT
    HIBA : BOOL ;
END_VAR
VAR
    Mhibatar : BOOL ;
    M2a3bol : BOOL ;
    T1 : TON ;
END_VAR

LD MOTA
ANDN MOTB
ANDN MOTC
OR( MOTB
ANDN MOTA
ANDN MOTC
)
OR( MOTC
ANDN MOTA
ANDN MOTB
)
ST M2a3bol

LD M2a3bol
AND ENG
ST T1.IN
LD t#5s
ST T1.PT
CAL T1

LD NYUGT
R Mhibatar

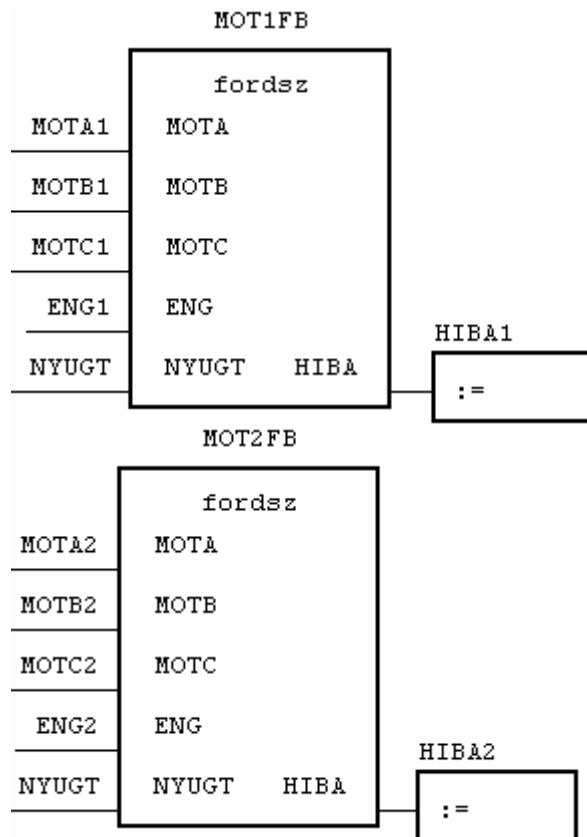
LD ENG
ANDN MOTA
ANDN MOTB
ANDN MOTC
S Mhibatar

LD T1.Q
OR Mhibatar
ST HIBA
END_FUNCTION_BLOCK

```

### Főprogram

Funkciótervben a függvényblokkok hívása az alábbi módon ábrázolható:



## A program utasításlistája

PROGRAM motorok

VAR

```
MOTA1 AT %I0.0.0.0.0: BOOL;  
MOTB1 AT %I0.0.0.0.1: BOOL;  
MOTC1 AT %I0.0.0.0.2: BOOL;  
ENG1 AT %I0.0.0.0.3:  BOOL;  
MOTA2 AT %I0.0.0.1.0: BOOL;  
MOTB2 AT %I0.0.0.1.1: BOOL;  
MOTC2 AT %I0.0.0.1.2: BOOL;  
ENG2 AT %I0.0.0.1.3:  BOOL;  
HIBA1 AT %Q0.0.0.0.0: BOOL;  
HIBA2 AT %Q0.0.0.0.1: BOOL;  
MOT1FB:      FORDSZ;  
MOT2FB:      FORDSZ;  
NYUGT AT %I0.0.0.0.7: BOOL;
```

END\_VAR

```
CAL MOT1FB(MOTA:=MOTA1,MOTB:=MOTB1,MOTC:=MOTC1,  
           ENG:=ENG1,NYUGT:=NYUGT)
```

```
LD  MOT1FB.HIBA  
ST  HIBA1
```

```
CAL MOT2FB(MOTA:=MOTA2,MOTB:=MOTB2,MOTC:=MOTC2,  
           ENG:=ENG2,NYUGT:=NYUGT)
```

```
LD  MOT2FB.HIBA  
ST  HIBA2
```

END\_PROGRAM

**Gyakorló feladat: utasításlista elemzése V.**

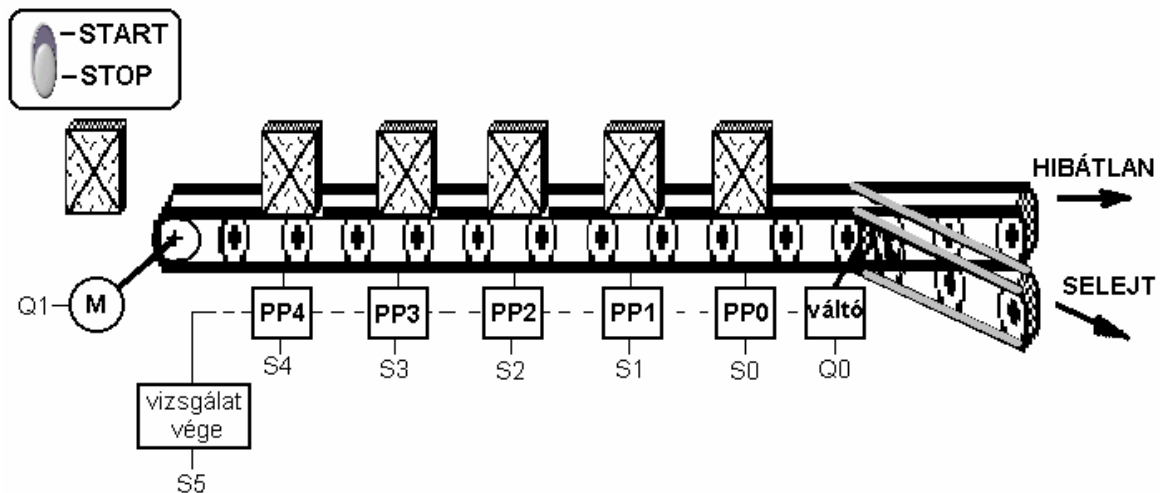
```
PROGRAM MASK1
VAR
    IB0 AT %IB0.0 : BYTE ;
    QB0 AT %QB0.0 : BYTE ;
END_VAR
VAR
    MB10 : BYTE ;
END_VAR
VAR
    IB1 AT %IB1.0 : BYTE ;
    QB1 AT %QB1.0 : BYTE ;
END_VAR
VAR
    MB5 : BYTE ;
END_VAR

LD MB10
XOR IB1
AND MB10
OR QB1
ST QB1

LD IB1
ST MB10
END_PROGRAM
```

## Minőségellenőrzés

Egy gyártási folyamat végén a munkadarabokat minőségellenőrzésnek vetik alá. A vizsgáló berendezés 5 db, egymástól egyenlő távolságra lévő ellenőrzőegységből (próbapad) áll (PP0..PP4). Szállítószalag gondoskodik a munkadarabok továbbításáról. Egyik részegységtől a másikig 5 s alatt ér a munkadarab. A szállítószalag először a bekapcsolás jelre indul el, majd S5 hatására, amely akkor jelez, ha az összes próbahelyen befejeződött az ellenőrzés. Ekkor a motor 5s-ig bekapcsol és egy vizsgálóhellyel továbblépteti a munkadarabokat. Ha a vizsgálandó munkadarab hibásnak bizonyul, a vizsgálóegység kimenetén 1-es jel jelenik meg. A szalag végén a munkadarab a vizsgálatok eredményétől függően vagy a HIBÁTLAN, vagy a SELEJT irányba halad tovább. (Átváltás a Q0 jellel.)



10. ábra Minőségellenőrző-sor vezérlése

## Összerendelési táblázat

Bemenetek	Jel	Logikai összerendelés	Cím
START/STOP kapcsoló	START	bekapcsolva, ha: SART=1	I1.0
1. próba eredménye	PP0	hibás, ha : PP0=1	I0.0
2. próba eredménye	PP1	hibás, ha : PP1=1	I0.1
3. próba eredménye	PP2	hibás, ha : PP2=1	I0.2
4. próba eredménye	PP3	hibás, ha : PP3=1	I0.3
5. próba eredménye	PP4	hibás, ha : PP4=1	I0.4
vizsgálat vége	S5	továbbléptethet, ha : S5=1	I0.5
<b>Kimenetek</b>			
szalagváltó	Q0	selejt irány: Q0=1	Q0.0
sz.szalag motor	Q1	bekapcsolva, ha: Q1=1	Q0.1

**Megoldás:** A munkadarabok hibás voltát egy **HIBA** nevű változóban (bájt) tároljuk. Minden munkadarabhoz egy-egy bit tartozik, amelyet a munkadarab léptetésével együtt léptetünk tovább (jobbra). Amikor a munkadarab a vizsgálatok befejeztével a szalag végére kerül, a **HIBA** merkerbájt legkisebb helyiértékű bitje jelzi, hogy valamelyik próbapadon a munkadarab hibásnak bizonyult-e, tehát selejtes, vagy mindegyik vizsgálatnak megfelelt, így a hibátlan darabok közé kerülhet.

A vizsgálat befejeztével S5 jelet ad. Ennek felfutó éle indítja az alábbi vezérlőalgoritmust:

- **HIBA** adatmerker-bájtot 1 helyiértékkel jobbra léptetni;
- az aktuális bemenőjel-bájttal kiegészítjük a hibatárolót (**HIBA**);
- az **HIBA** legalacsonyabb helyiértékű bitjének megfelelően **Q0** kimenet (szalagváltó) beállítjuk vagy töröljük.

### A vezérlőprogram

PROGRAM minellen

VAR

```
START AT %I0.0.0.1.0:   BOOL;
EREDM AT %IB0.0.0.0.0:  BYTE;
KESZ AT %I0.0.0.1.1:    BOOL;
Q0 AT %Q0.0.0.0.0:     BOOL;
Q1 AT %Q0.0.0.0.1:     BOOL;
HIBA AT %MB0.0.0.2.0:   BYTE;
T1:   TP;
KESZIMP:  R_TRIG;
M1:   BOOL;
M2:   BOOL;
BEIMP:   R_TRIG;
```

END\_VAR

VAR constant

```
T11:  TIME := T#0.5S;
```

END\_VAR

```
CAL BEIMP(CLK:=START)
```

```
CAL KESZIMP(CLK:=KESZ)
```

```
LD KESZIMP.IMP
```

```
JMPCN TOVABB
```

```
LD HIBA
```

```
SHR 1
```

```
OR EREDM
```

```
ST HIBA
```

```
AND 1 (*MASZK*)
```

```
EQ 1
```

```
ST Q0 (*SELEJT?*)
```

TOVABB:

```
LD BEIMP.IMP
```

```
OR KESZIMP.IMP
```

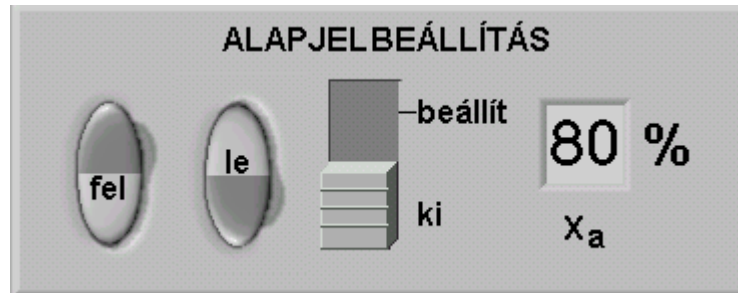
```
ST T1.IN
```

```
LD    T11
ST    T1.PT
CAL   T1
LD    T1.Q
ST    Q1
```

```
END_PROGRAM
```

## Alapjeladó

A kezelő egy szabályozás alapjelét 0..99% között változtathatja, a **fel** illetve **le** nyomógombok segítségével. A beállított alapjel visszajelzésre kerül. Írjuk meg az alapjel beállítását végző programrészletet. (A szabályozás most nem feladatunk, a kiszámított alapjelet más program használja fel.) Az alapjel csak akkor módosítható, ha előzőleg átkapcsolnak beállító üzemmódba. (A kapcsolót esetleg kulccsal is védhetik.)



11. ábra A kezelői felület

## Összerendelési táblázat

Bemenetek	Jel	Logikai összerendelés	Cím
beállító kapcsoló	BE	beállítás, ha: BE=1	I1.0
alapjel növelése	FEL	benyomva : FEL=1	I0.0
alapjel csökkentése	LEF	benyomva : LEF=1	I0.1
<b>Kimenet</b>			
alapjel értéke	QXA	alapjel értéke: QXA=byte	Q0.1

**Megoldás:** A nyomógombok benyomásával T1 időzítő ütemjelére folyamatosan inkrementálható ill. dekrementálható az alapjel értéke 0 és 99 % között. Ha mindkét nyomógombot benyomják, nem történik semmi.

## A vezérlőprogram

PROGRAM alapjALL

VAR

```
FEL AT %I0.0.0.0.0: BOOL;
LEF AT %I0.0.0.0.1: BOOL;
BE AT %I0.0.0.1.0:  BOOL;
QXA AT %QB0.0.0.0.0:  BYTE;
```

END\_VAR

VAR constant

```
T11:  TIME := T#0.25S;
```

END\_VAR

```
VAR
    T1:  TON;
    ALAPJ:INT;
END_VAR
```

```
LD  BE
ANDNT1.Q
ST  T1.IN
LD  T11
ST  T1.PT
CAL  T1
LD  T1.Q
RETCN
```

```
LD  FEL
AND  LEF
RETC
```

```
LD  FEL
JMPC NOVEL
```

```
LD  LEF
JMPC CSOKK0
RET
```

```
NOVEL:
LD  ALAPJ
EQ  99
RETC
```

```
LD  ALAPJ
ADD  1
ST  ALAPJ
JMP  VEGE
```

```
CSOKK:
LD  ALAPJ
EQ  0
RETC
LD  ALAPJ
SUB  1
ST  ALAPJ
```

```
VEGE:
LD  ALAPJ
INT_TO_BYTE
ST  QXA
RET
END_PROGRAM
```

### Tömbök használata a tároló nélküli követővezérlésekben

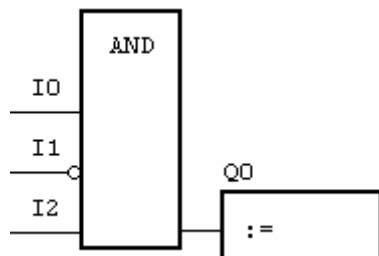
Ha a változók száma nem túl nagy, a tároló nélküli követővezérlés bemenő és kimenő változói közötti kapcsolat függvénytáblázattal leírható. A logikai függvényt nem egyszerűsítjük, hanem soronként felírjuk a függvénykódnak megfelelő digitális számot. Az így kapott vektort egy konstans tömbbe tároljuk. A bemeneteket és kimeneteket nem bitenként, hanem összefüggő bitsoportként (bájt vagy szó, a változók számától függően) kezeljük. Az éppen aktuális bemeneti érték meghatározza a vektornak azt az elemét, amelynek tartalmát a kimenetre írhatjuk. A főprogramban a be és kimeneti bájtokat maszkoljuk, hogy a PLC be/kimeneteire esetlegesen rákötött egyéb jelek ne befolyásolják a kiolvasott értéket, csak azokat a biteket dolgozzuk fel és írjuk felül, amelyek az adott feladathoz tartoznak.

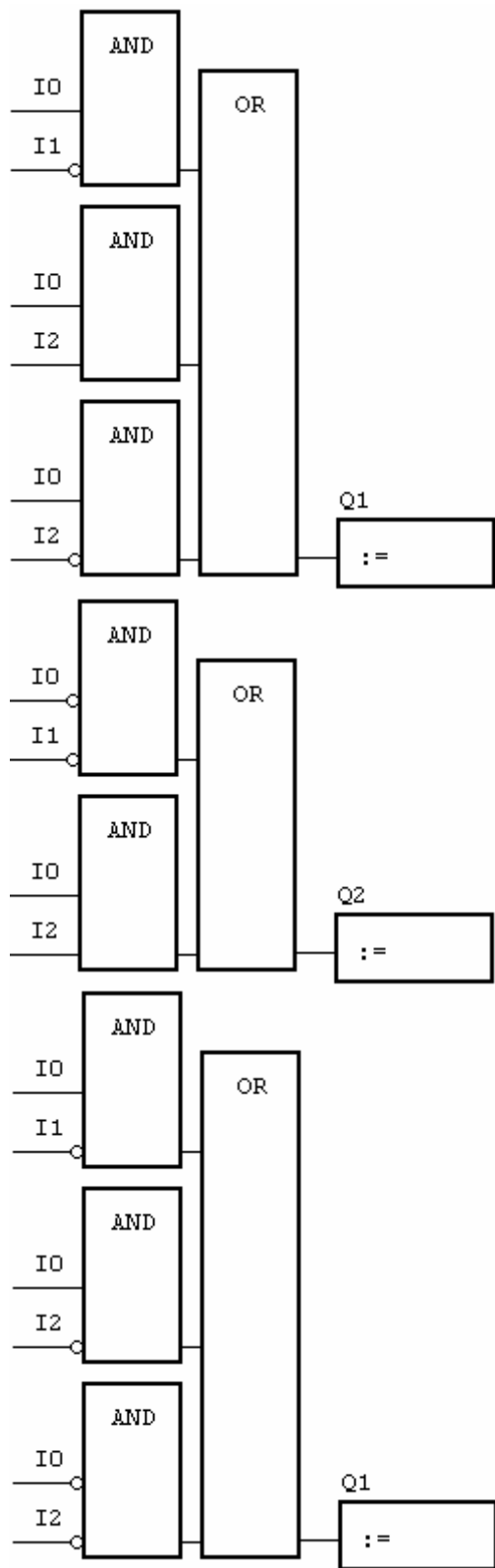
A program egyszerűen módosítható a tömb és a maszkok átírásával.

### Összerendelési táblázat

Bemenetek	Cím
I0	I0.0
I1	I0.1
I2	I0.2
Kimenetek	
Q0	Q0.0
Q1	Q0.1
Q2	Q0.2
Q3	Q0.3

### A be- és kimenetek közötti függvénykapcsolat





**A függvénytáblázat**

A függvénytáblázat felírásakor ügyeljünk arra, hogy a fizikai címeknek megfelelő növekvő sorrendben vegyük fel a be/kimeneti oszlopokat!

I2	I1	I0	Q3	Q2	Q1	Q0	KIMENET
0	0	0	1	1	1	0	14
0	0	1	1	0	1	0	10
0	1	0	0	0	1	0	2
0	1	1	1	0	0	0	8
1	0	0	0	1	0	0	4
1	0	1	1	1	1	1	15
1	1	0	0	0	0	0	0
1	1	1	0	1	1	0	6

### A vezérlőalgorithmus

PROGRAM prkowvez

VAR

IB0 AT %IB0.0.0.0: BYTE;

QB0 AT %QB0.0.0.0: BYTE;

TABLA: ARRAY[0..7] OF BYTE := [14,10,2,8,4,15,0,6];

END\_VAR

VAR constant

BEMASK: BYTE := 2#00000111;

KIMASK: BYTE := 2#11110000;

END\_VAR

VAR

M0: INT;

END\_VAR

LD IB0

AND BEMASK

BYTE\_TO\_INT

ST M0

LD QB0

AND KIMASK

OR TABLA[M0]

ST QB0

END\_PROGRAM

**Feladat:** Módosítsa a fenti programot úgy, hogy a mélygarázs szellőzésfelügyeletét valósítsa meg!

## Tömbök használata ütemvezérelt lefutóvezérléseknél

Egy útkereszteződésben a főút és a mellékutak kereszteződését közlekedési lámpával irányítják. A megadott ütemdiagram alapján kell a piros, sárga, zöld fázist kapcsolgatni. Az időegység (ütemegység) 5s. A berendezést **S0** kapcsolóval lehet bekapcsolni.

### Ütemdiagram

ütem:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0
<b>P1</b>																	
<b>S1</b>																	
<b>Z1</b>																	
<b>P2</b>																	
<b>S2</b>																	
<b>Z2</b>																	

### Összerendelési táblázat

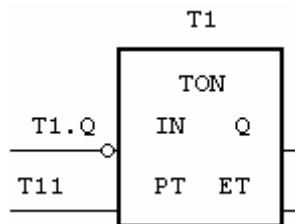
Bemenetek	Jel	Logikai összerendelés	Cím
BE/KI kapcsoló	S0	bekapcsolva: S0=1	I0.0
Kimenetek			
1. lámpa (főút) piros	P1	világít: P1=1	Q0.0
1. lámpa (főút) sárga	S1	világít: S1=1	Q0.1
1. lámpa (főút) zöld	Z1	világít: Z1=1	Q0.2
2. lámpa (mellékút) piros	P2	világít: P2=1	Q0.3
2. lámpa (mellékút) sárga	S2	világít: S2=1	Q0.4
2. lámpa (mellékút) zöld	Z2	világít: Z2=1	Q0.5

A megoldáshoz felhasználunk egy bekapcsolás-késleltetési időzítőt, az ütemjel generálására és egy számlálót az ütemek számlálására. A kimeneti byte értékeket egy tömbben tároljuk. A számláló értéke fogja megadni, hogy a tömb hányadik elemét írjuk ki a kimeneti byte-ba.

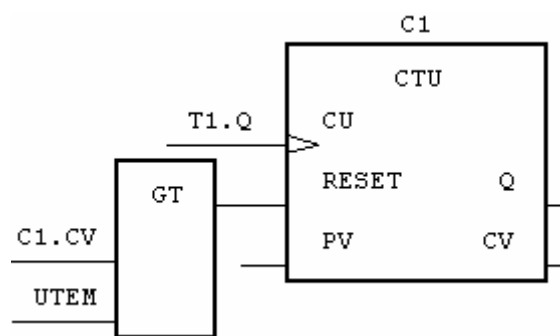
**Ha S0=1, a kimenetek ütemezése:**

számláló	Z2	S2	P2	Z1	S1	P1	kimeneti BYTE
0	0	0	1	1	0	0	12
1	0	0	1	1	0	0	12
2	0	0	1	1	0	0	12
3	0	0	1	1	0	0	12
4	0	0	1	1	0	0	12
5	0	0	1	1	0	0	12
6	0	0	1	1	0	0	12
7	0	0	1	1	0	0	12
8	0	0	1	0	1	0	10
9	0	1	1	0	0	1	25
10	1	0	0	0	0	1	33
11	1	0	0	0	0	1	33
12	1	0	0	0	0	1	33
13	1	0	0	0	0	1	33
14	0	1	0	0	0	1	17
15	0	0	1	0	1	1	11

**Az ütemgenerátor**



A számláló:



**A vezérlőalgorithmus**

PROGRAM TMBUTEM

VAR

S0 AT %I0.0.0.0: BOOL;

QB0 AT %QB0.0.0.0: BYTE;

TABLA:ARRAY[0..16] OF BYTE := [12,12,12,12,12,12,12,12,12,10,25,33,33,33,33,17,11,11];

T1: TON;

```

    C1:   CTU;
END_VAR

VAR constant
    T11:   TIME := t#2s;
    UTEM:  INT  := 15;
END_VAR

VAR
    KIMASK:   BYTE := 2#11000000;
    M0:       BOOL;
    S0IMP:    BOOL;

END_VAR

LD     S0
JMPCNKIKAPCS

LD     S0
ANDN   M0
ST     S0IMP
LD     S0
ST     M0

LDN    T1.Q
ST     T1.IN
LD     T11
ST     T1.PT
CAL    T1

LD     T1.Q
ST     C1.CU
LD     C1.CV
GT     UTEM
OR     S0IMP

ST     C1.RESET
CAL    C1

LD     QB0
AND    KIMASK
OR     TABLA[C1.CV]
ST     QB0
RET
KIKAPCS:

LD     0
ST     QB0
RET
END_PROGRAM

```

Feladatok:

A fenti program a kikapcsolás jel hatására azonnal lekapcsolja az összes lámpát. Hogyan módosítaná a programot, ha az lenne a feladata, hogy az adott ciklust még fejezze be?

Módosítsa úgy a fenti programot, hogy kikapcsolás után mindkét irányban a sárga lámpa villogjon 0,5 Hz frekvenciával!

Tervezzen reklámfényt, és az algoritmust a fenti példa szerint írja meg!

## Irodalomjegyzék

1. LabView is a registered trademark of National Instruments Corporation
2. Beuschel, J.: Prozesssteuerungssysteme : Einführung in die Informationsverarbeitung in Automatisierungsanlagen – München; Wien: Oldenbourg, 1994
3. infoteam OpenPCS programming System Ver. 4.1. User Manual 1. Edition infoteam Software GmbH, D-91088, Bubenreuth. 1996-2001. <http://www.infoteam.de>
4. Jakoby, W.: Automatisierungstechnik – Algorithmen und Programme: Entwurf und Programmierung von Automatisierungssystemen; Berlin; Heidelberg: Springer, 1997
5. Johns, K.- Tiegelkamp, M.: SPS-Programmierung mit IEC 1131-3 : Konzepte und Programmiersprachen, Anforderungen an Programmiersysteme, Entscheidungshilfen Berlin; Heidelberg: Springer, 1996
6. Sucosoft S 40 Programming Software: Language Elements for PS 4-150/-200/-300 and PS 416 02/00AWB 2700-1306 GB Bonn, Moeller GmbH, 2000
7. Sucosoft S 40 Programming Software: Training Guide 06/990AWB 27-137 GB Bonn, Moeller GmbH, 1999
8. Wellenreuther, G., Zastrow, D.: Steuerungstechnik mit SPS : Bitverarbeitung und Wortverarbeitung, Regeln mit SPS, Von der Steuerungsaufgabe zum Steuerungsprogramm; Braunschweig; Wiesbaden: Vieweg, 1995
9. Wellenreuther, G., Zastrow, D.: Lösungsbuch Steuerungstechnik mit SPS; Braunschweig; Wiesbaden: Vieweg, 1995