

PLC programozás 1.

- I. PLC alapismeretek
- II. Egyszerű logikai vezérlések
- III. Flag
- IV. Timer

1.1. PLC, bevezetés

Általános jellemzők

- Programmable Logic Controller - programozható logikai vezérlő
- célszámítógép ipari környezetbe → irányítástechnikai feladatokra (ipari folyamatok vezérlésére, szabályozására)
- ipari kivitelű mikroszámítógép, a régi relés vagy elektronikus huzalozott vezérlések kiváltására → előnye a kevesebb huzalozás, egyszerűbb módosítás
- sok cég gyárt PLC-eket, pl. Siemens, Omron, Festo, Möeller, Schneider, ...
- első PLC → Modicon 1966

Típusai

- méret alapján
 - kicsi, 10/20 be- és kimenet
 - közepes, néhány száz be- és kimenet
 - nagy, több mint ezer be- és kimenet
- felépítés szempontjából
 - kompakt, a hardver nem módosítható, fix számú bemenet, kimenet
 - moduláris, a hardver rugalmasan bővíthető modulokból áll → be- és kimenetek száma jelentősen bővíthető

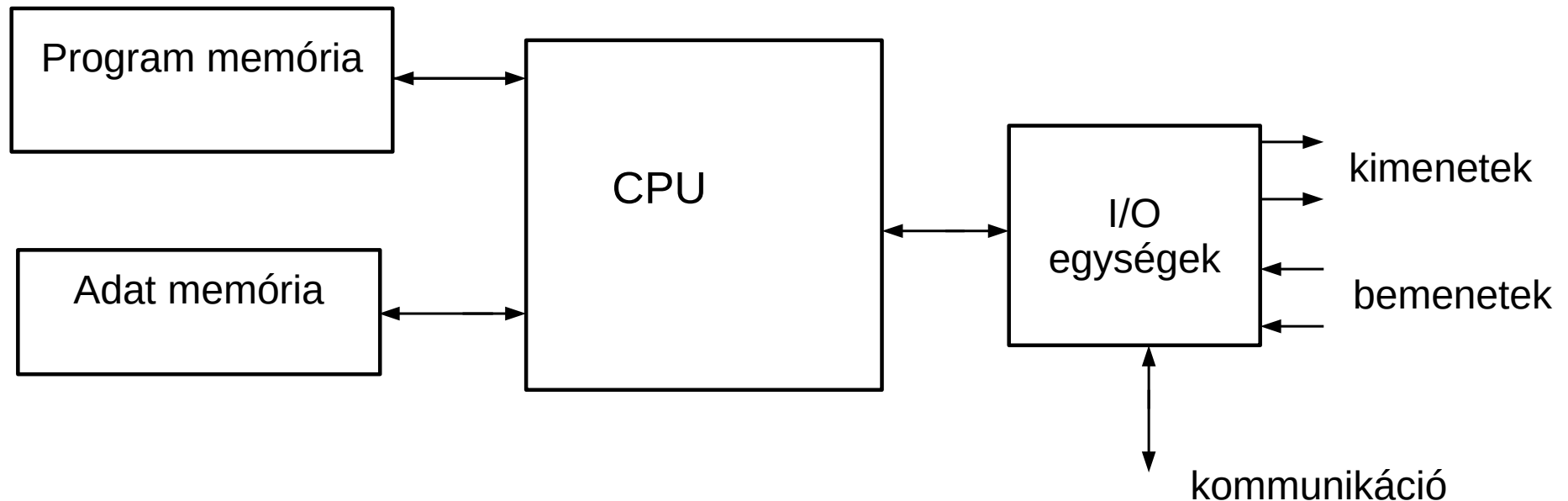
Szabványok

IEC 1131-3 programozási nyelvek, PLC projektek felépítése

1.2. PLC, bevezetés

Hardver felépítés

Nem teljesen a hagyományos számítógép felépítést követik →
az adat és program memória külön van választva → Harvard architektúra



CPU central processing unit (központi feldolgozó egység)

két részből áll: műveletvégző egység (ALU arithmetic and logic unit)

+ vezérlő egység (CU control unit)

1.3. PLC, bevezetés

Műveletvégző egység

- ALU (arithmetic and logic unit),
- matematikai és logikai műveletek elvégzése

Vezérlőegység

- CU (control unit), vezérli az egész számítógép működését
- a tárolt program utasításait egyenként, sorban lehívja, dekódolja, majd a szükséges vezérlő jeleket előállítja (ALU, memória, I/O felé)

Perifériák

- Input, output vezérlő, illesztő egységek → kapcsolat a külvilággal
- a bemenetek, kimenetek kapcsolatát biztosítja a rendszerrel

Kommunikációs interfészek

- kapcsolat PLC és PC között → PLC-re program feltöltése
Lehet: soros port, USB, ...
- PLC-k összekötése egymással, ipari buszok (MODBUS, PROFIBUS, ...) → működés összehangolása, felügyelet, mérés- adatgyűjtés, ...

1.4. PLC bemenetek, kimenetek

Bemenetek

Az irányítás bemenő jelei

a, kapcsolók, nyomógombok (záró, bontó)

b, érzékelők

pozíció meghatározás, véghelyzet érzékelés

- mechanikus végállás kapcsolók

- közelítő kapcsolók (érintésmentes)

induktív (vezető anyag érzékelésére)

kapacitív (fémek, szigetelők érzékelése)

optikai (fénysorompó)

mágneses (reed)

c, szenzorok

jelátalakítók → valamilyen mennyiséget villamos jellé alakítanak át

lehetnek:

- analógak

- digitálisak

mágneses (HALL szenzor)

termikus (termisztor,...)

optikai

mechanikai (nyomás,távolság,sebesség,...)

kémiai (nedvesség, por, ..)

1.5. PLC bemenetek, kimenetek

Kimenetek

Az irányítás kimenő jelei

- jelző készülékek (lámpa, hangjelző, ...)
- beavatkozó készülékek
relék, mágnesszelepek, motorok, ...

PLC-k bemenetei, kimenetei

lehetnek:

- digitálisak (két állapot !) → jellemzően 24V DC (vagy 24V AC, 120/230V AC)

- analógok

feszültség

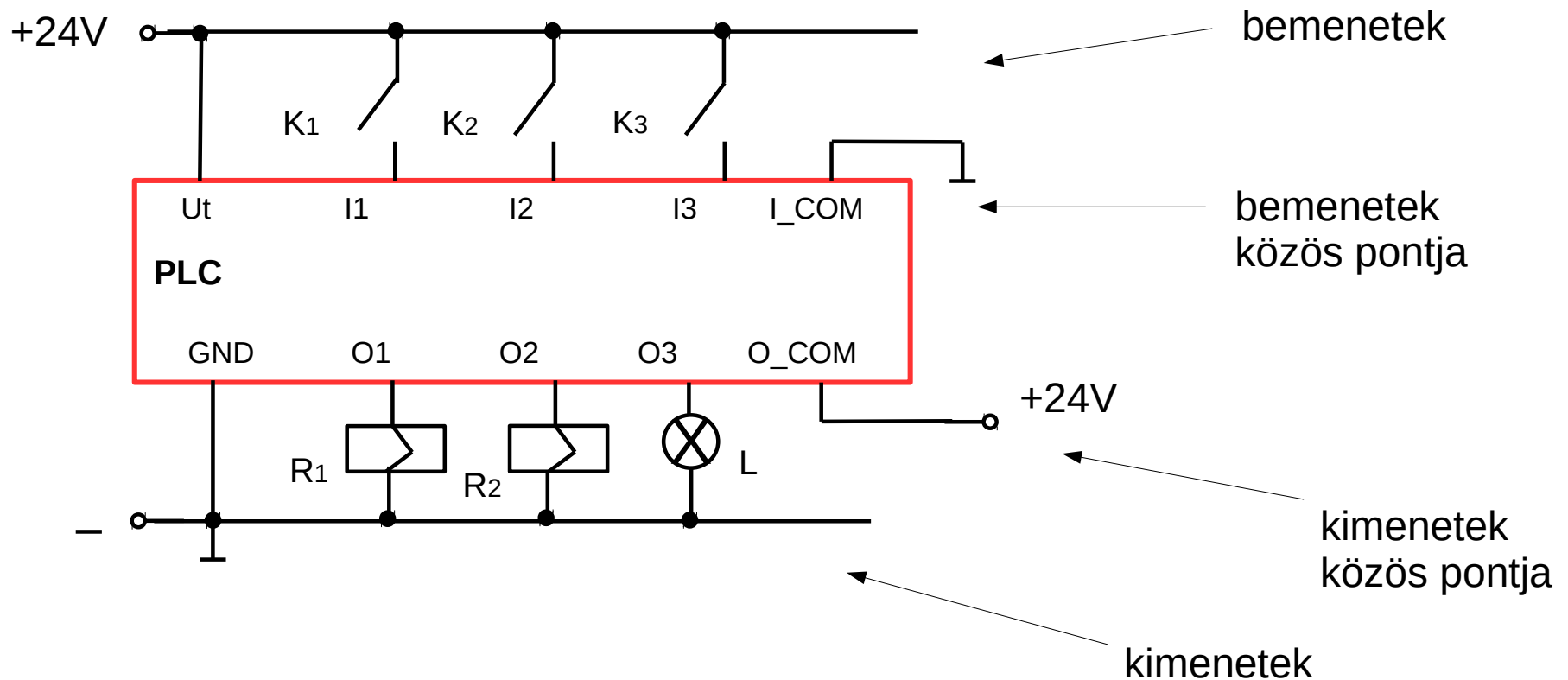
- 0-10 V
- 0-5 V
- ...

áram

- 0-20 mA
- 4-20 mA
- ...

1.6. PLC bemenetek, kimenetek

Bemenetek, kimenetek bekötése

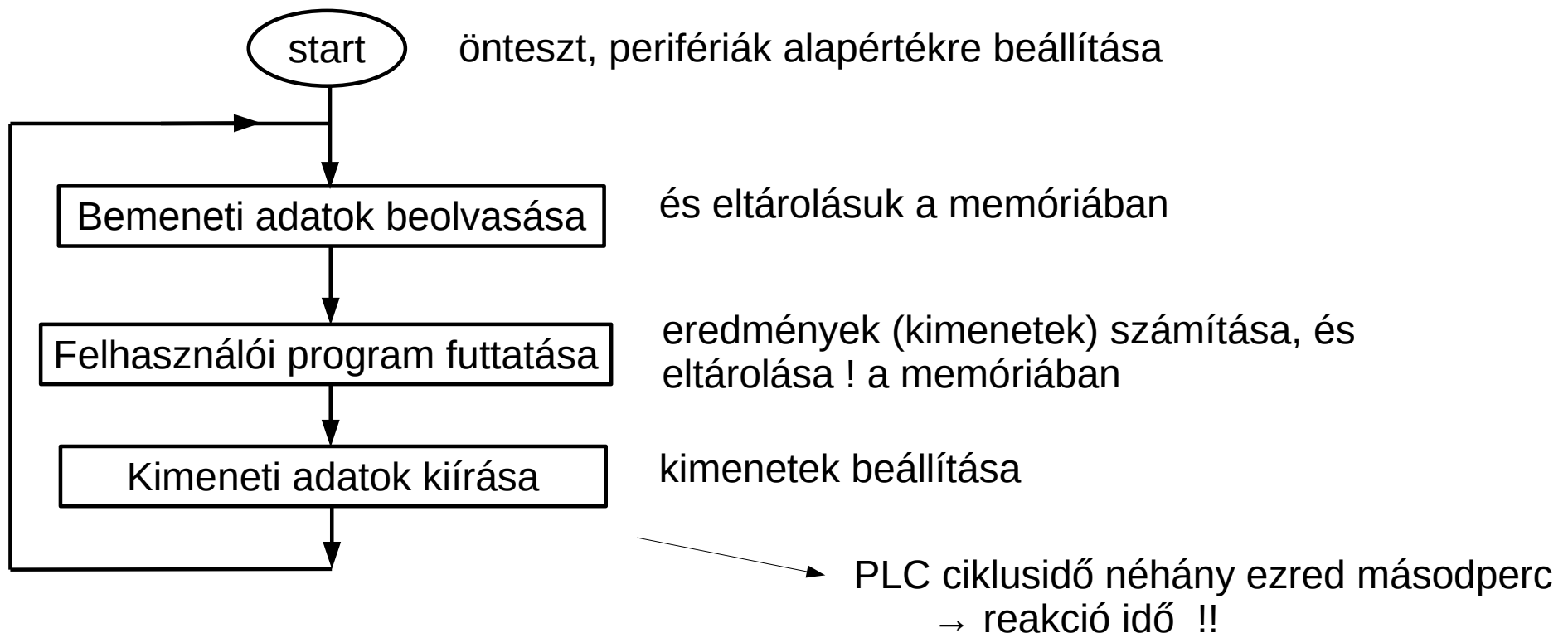


1.7. PLC programok

PLC programok

- operációs rendszer (monitor program) — futtatja a felhasználói programokat
 - felhasználói program
az adott vezérlési feladatot hajtja végre
- beolvasás, kiírás irányítása
 - program fejlesztési funkció
 - kommunikációs vonalak kezelése
 - megszakításkezelés

PLC ciklikus program feldolgozása



1.8. PLC címzések

Adatok címzése

- az adatmemóriában tárolt változók:
 - bemenetek beolvasott értékei
 - kimenetek számított értékei
 - belső változók (flag, merker)
 - speciális változók (időzítő, számláló, ...)
 - rendszer változók
- ezeket címezni kell
(hivatkozni kell rájuk)
- címzés lehet: bitenként,
vagy bájtos, szavas, ...
- címzésnél megkülönböztetjük a különböző típusú változókat, általában betűkkel →
 - bemenet – I (input) kimenet – Q vagy O (output)
 - flag/merker – F vagy M vagy MB
 - időzítő – T (timer) számláló – C (counter)
- de a különböző típusú változók megkülönböztethetők tisztán számokkal is !
egyes számtartományok meghatározott típusú változókhoz vannak rendelve, pl.
 - 000-100 bemenetek
 - 101-200 kimenetek
 - 201-300 flag-ek
 -

1.9. PLC címzések

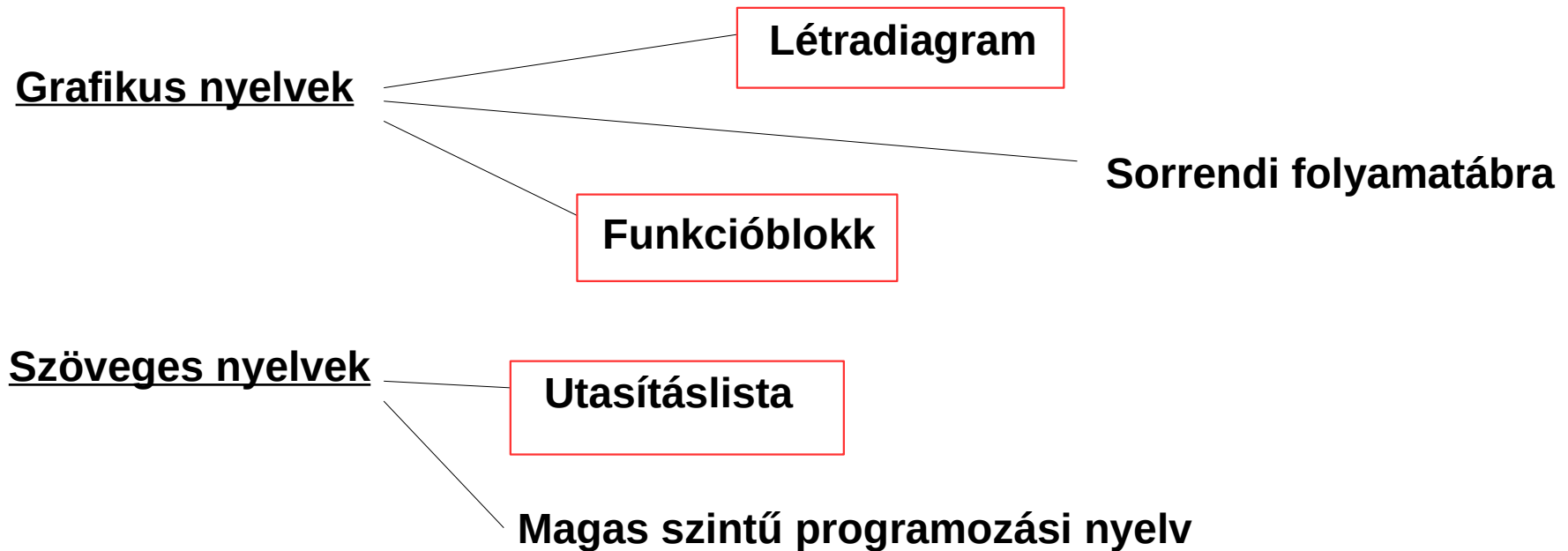
Jellemző címzés minták (Festo, Siemens)

- bit címzés: I1.3 → bemeneti memória 1. byte 3. bitje
I0.7 → bemeneti mem. 0. byte 7. bitje
O2.0 → kimeneti memória 2. byte 0. bitje
F3.6 → flag 3. byte 6. bitje
- byte címzés: IB3 → bemeneti mem. 3. byte
OB2 → kimeneti 2. byte
- szavas címzés: IW4 → bemeneti 4. szó (jellemzően 16 bit)
OW3 → kimeneti 3. szó
- Timer T3 → 3. impulzus időzítő
TON5 → 5. bekapcsolásidőzítő

Jellemző címzés minták (Unitronics)

- I1 → 1. digitális bemenet
- I27 → 27. digitális bemenet
- O2 → 2. digitális kimenet
- O16 → 16. digitális kimenet
- MB4 → 4. memory bit → flag !
- MB18 → 18. memory bit → flag !
- SB2 → 2. rendszer bit (system bit)
- TD3 → 3. Timer (bekapcsolásidőzítő)

1.10. PLC programozási nyelvek



1. Feladat

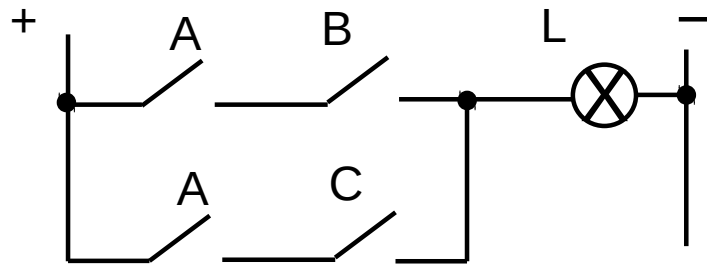
Az **L** izzó akkor világítson (legyen logikai 1-es értékű) ha **A**, **B** és **C** kapcsolók közül **A ÉS B** zárt egyszerre, **VAGY A ÉS C** zárt egyszerre !!

Kapcsolási rajz ? → kapcsolókkal
→ Logikai kapuáramkörökkel

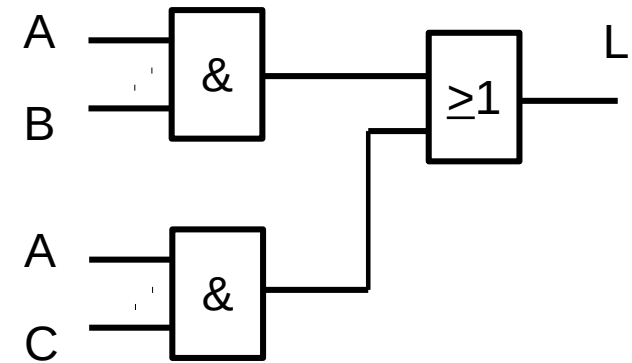
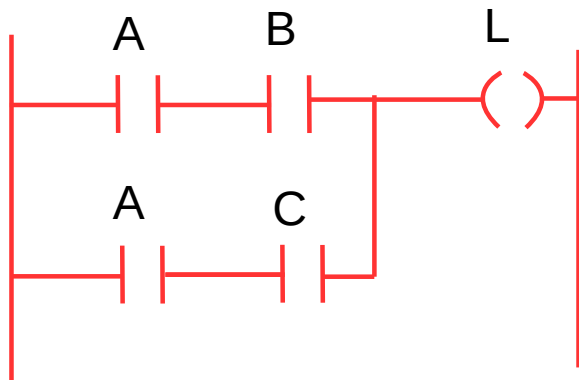
1.11. PLC programozási nyelvek

1. feladat megoldásai

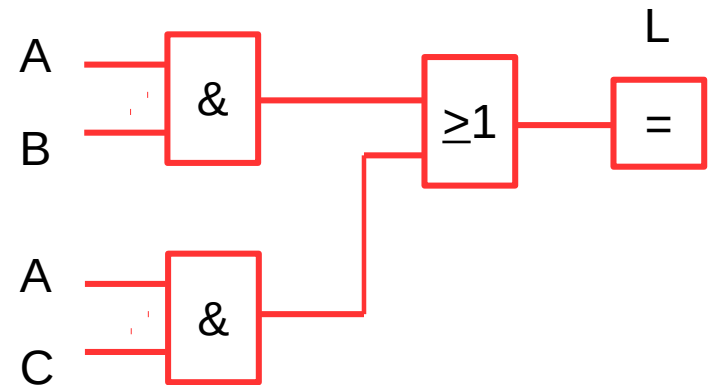
$$L = A * B + A * C$$



Létradiagram (LAD)



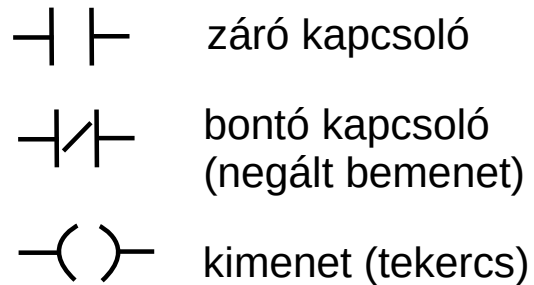
Funkcióblokk (FB)



1.12. PLC programozási nyelvek

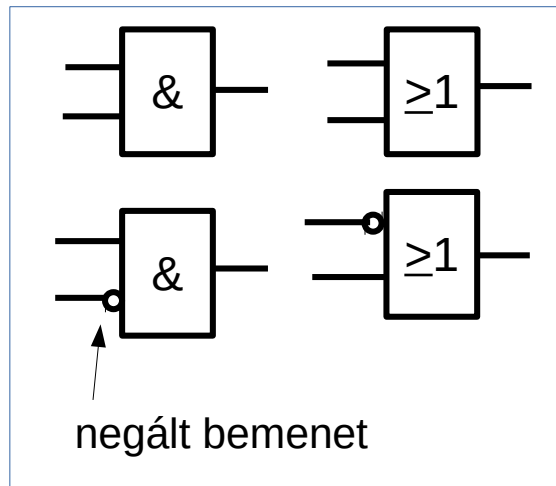
Grafikus nyelvek

Létradiagram
(LD vagy **LAD** vagy KOP)



ÉS fv. → soros kapcsolók
VAGY fv. → párhuzamos kapcs.

Funkcióblokk
(**FB** vagy FUP)



Sorrendi folyamatábra
(SFC)

Szöveges nyelvek

Utasításlista
(**STL** vagy IL v. AWL)

**Magas szintű
programozási nyelv**
(ST vagy SCL)

2.1. Egyszerű logikai vezérlések

1. AND (ÉS) függvény

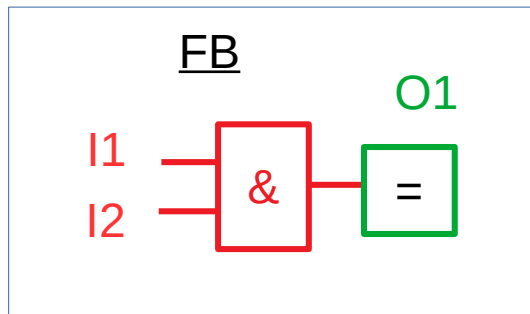
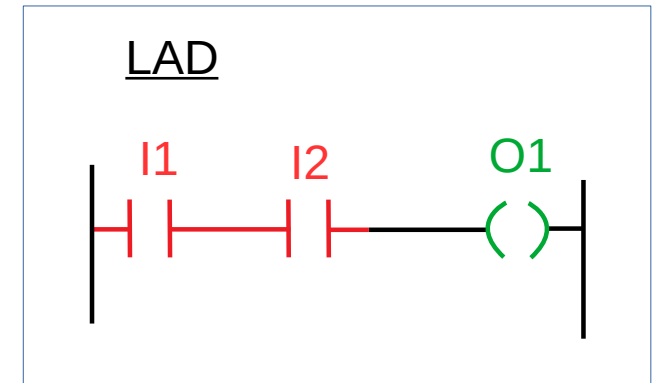
Az O1 kimenet csak akkor legyen 1-es értékű ha I1 és I2 bemenet értéke egyszerre 1-es

Igazságtáblázat:

I1	I2	O1
0	0	0
0	1	0
1	0	0
1	1	1

Logikai függvény:

$$O1 = I1 * I2$$



STL (FESTO FST)

IF I1 AND I2
THEN SET O1
OTHRW RESET O1

IF feltétel
THEN (ha a feltétel igaz)
OTHRW(ha a feltétel hamis)

SET bit → bit=1
RESET bit → bit=0

2.2. Egyszerű logikai vezérlések

2. OR (VAGY) függvény

Az O1 kimenet akkor legyen 1-es értékű ha I1 **vagy** I2 bemenet értéke 1-es

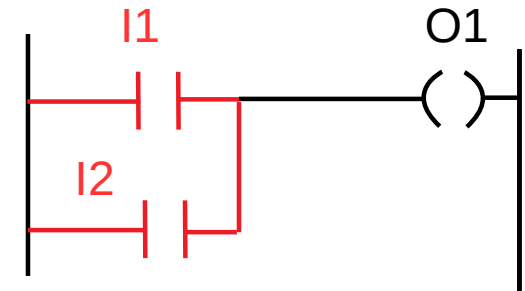
Igazságtáblázat:

I1	I2	O1
0	0	0
0	1	1
1	0	1
1	1	1

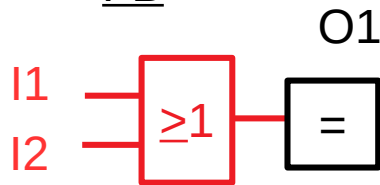
Logikai függvény:

$$O1 = I1 + I2$$

LAD



FB



STL (FESTO FST)

IF I1 OR I2
THEN SET O1
OTHRW RESET O1

2.3. Egyszerű logikai vezérlések

3. NOT (NEM, tagadás) függvény

Az O1 kimenet akkor legyen 1-es értékű ha I1 bemenet értéke 0

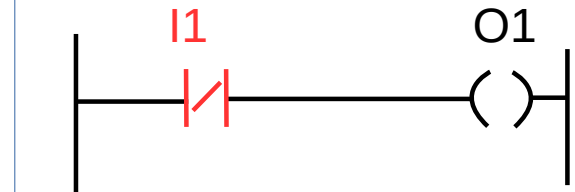
Igazságtáblázat:

I1	O1
0	1
1	0

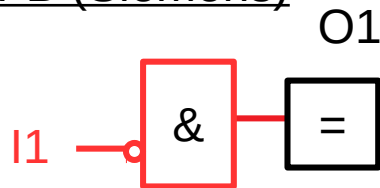
Logikai függvény:

$$O1 = \overline{I1}$$

LAD



FB (Siemens)



STL (FESTO FST)

```
IF N I1  
THEN SET O1  
OTHRW RESET O1
```


2.4. Egyszerű vezérlési feladatok

1. Feladat

Az O1 kimenet akkor legyen 1-es értékű ha I1 I2 I3 bemenetek közül pontosan 2db értéke 1-es (digitális bemenetek, kimenetek)

Igazságtáblázat:

I1	I2	I3	O1
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

→ $\overline{I1} * I2 * I3$

→ $I1 * \overline{I2} * I3$

→ $I1 * I2 * \overline{I3}$

Logikai függvény:

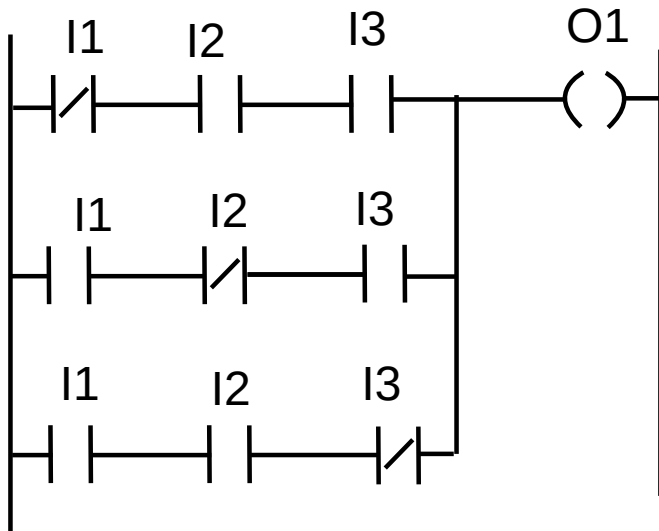
$$O1 = \overline{I1} * I2 * I3 + I1 * \overline{I2} * I3 + I1 * I2 * \overline{I3}$$

2.5. Egyszerű vezérlési feladatok

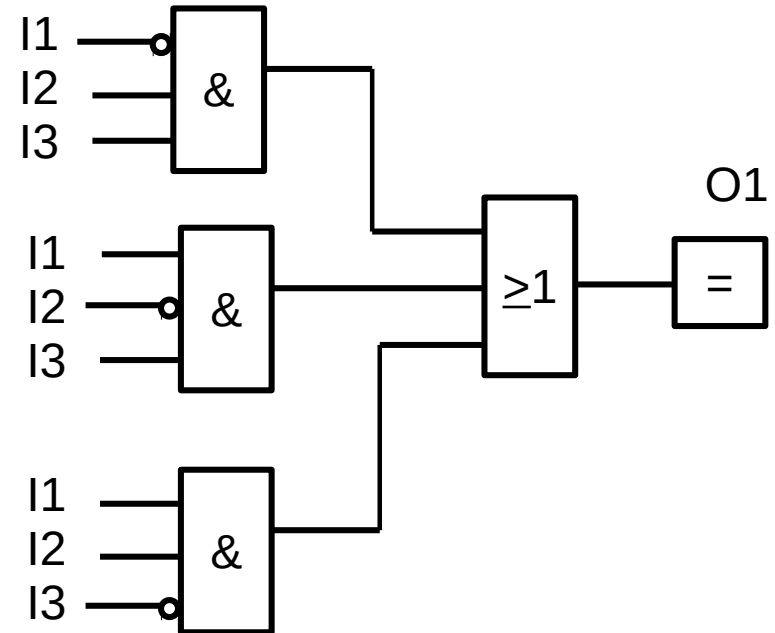
1. feladat megoldásai

$$O1 = \overline{I1} * I2 * I3 + I1 * \overline{I2} * I3 + I1 * I2 * \overline{I3}$$

Létradiagram (LAD)



Funkcióblokk (FB)



Utasításlista (STL, FESTO)

```
IF N I1 AND I2 AND I3
  OR I1 AND N I2 AND I3
  OR I1 AND I2 AND N I3
THEN SET O1
OTHRW RESET O1
```

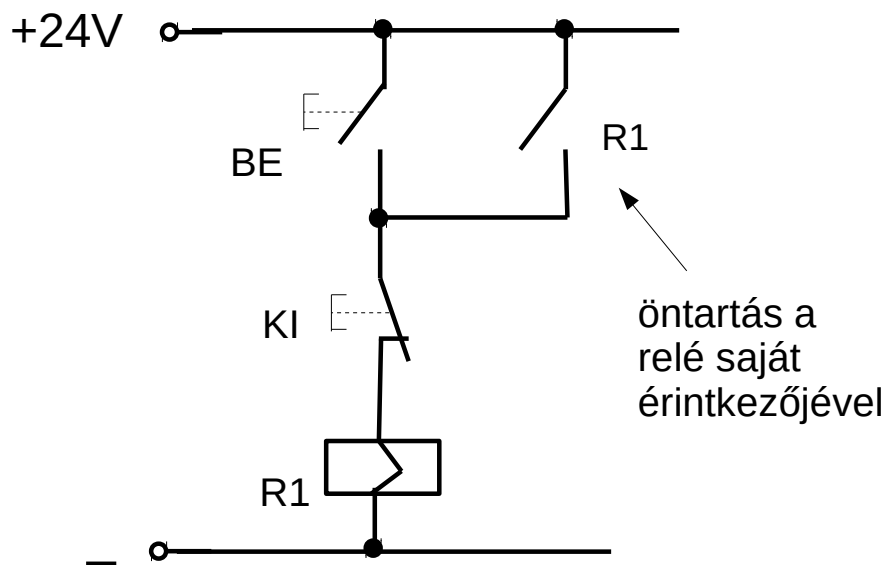
2.6. Egyszerű vezérlési feladatok

2. Öntartás vezérlés

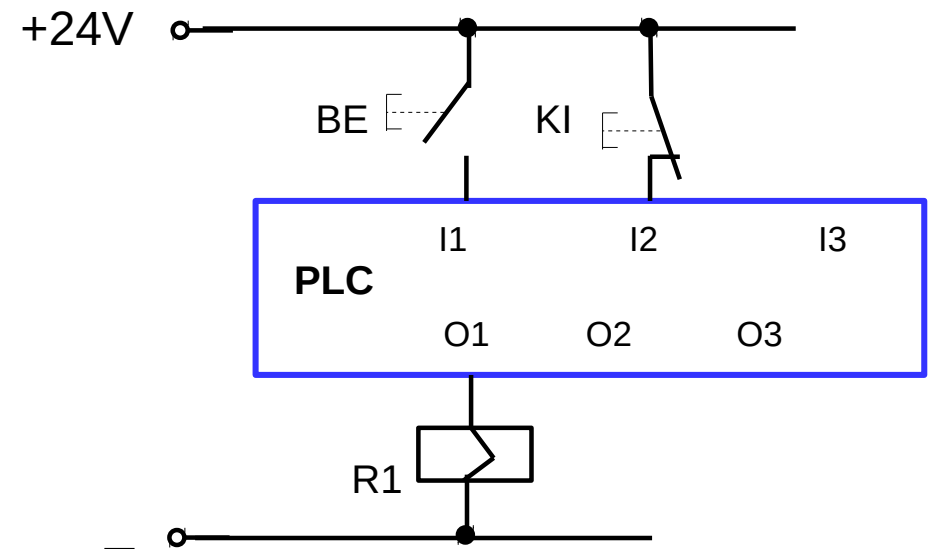
Minta feladat: a „BE” nyomógombot lenyomva „R1” relére feszültséget kapcsolunk, meghúz és feszültség alatt marad ha a nyomógombot felengedjük !

Kikapcsolása „KI” nyomógomb megnyomásával történik.

↓ Relés megvalósítás



↘ PLC-vel:



Logikai függvény:

$$O1 = (I1 + O1) * I2$$

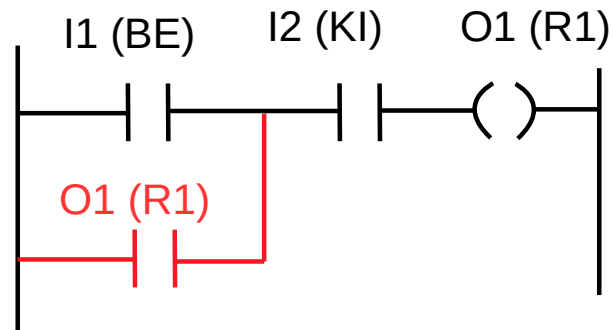
2.7. Egyszerű vezérlési feladatok

Öntartás vezérlés, megoldás

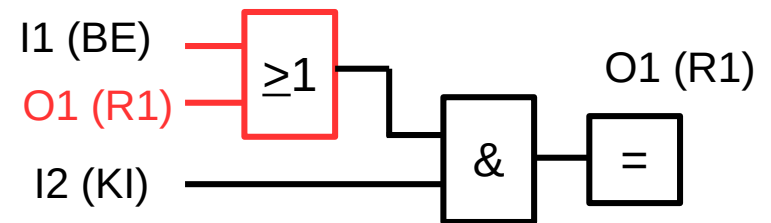
PLC esetén minden bemenet és kimenet aktuális állapota a memóriában tárolva van. → Bármikor lekérdezhetőek ! A kimenetek is (mintha bemenetek lennének)!!

A programok:

LAD



FB



STL (FESTO)

```
IF ( I1 OR O1 ) AND I2  
THEN SET O1  
OTHRW RESET O1
```

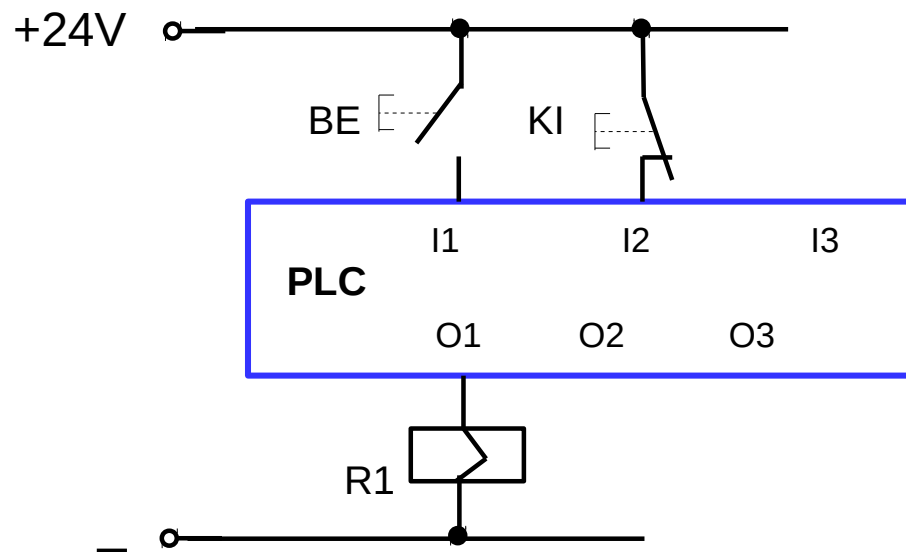
2.8. Egyszerű vezérlési feladatok

3. Öntartás vezérlés, másképp

Öntartás leprogramozható SET és RESET tekercsek (LAD) vagy SR, RS tárolóval (FB).
SET → a kimenet 1 értékű lesz és úgy marad ! RESET → a kimenet 0 értékű lesz és úgy marad !

SR, RS tároló működése:

- ha $S=1$ (és $R=0$) → Set → kimenet 1, és így marad (amíg nincs újra Reset)
- ha $R=1$ (és $S=0$) → Reset → kimenet 0, és így marad (amíg nincs újra Set)
- ha $S=R=1$ (elvileg tiltott !) gyakorlatilag ilyenkor az egyik a domináns („erősebb”)
SR tároló → dominánsan Set RS tároló → dominánsan Reset



Logikai függvények:

$$O1_{\text{set}} = I1$$

$$O1_{\text{reset}} = \overline{I2}$$

↓
nevekkel

$$R1_{\text{set}} = BE$$

$$R1_{\text{reset}} = \overline{KI}$$

↘
más
jelöléssel

$$O1_+ = I1$$

$$O1_- = \overline{I2}$$

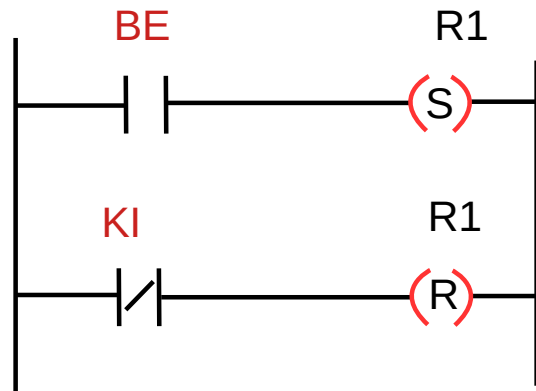
2.9. Egyszerű vezérlési feladatok

3. Öntartás vezérlés másképp, megoldás

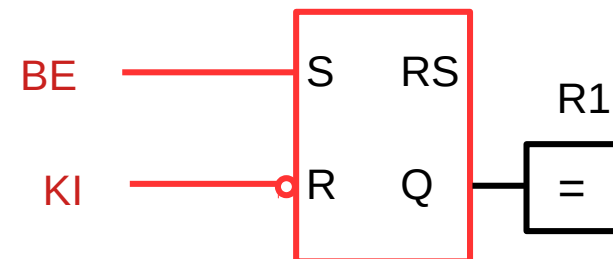
$R1_{\text{set}} = \text{BE}$

$R1_{\text{reset}} = \overline{\text{KI}}$

LAD



FB



STL (FESTO)

IF **BE** THEN SET R1

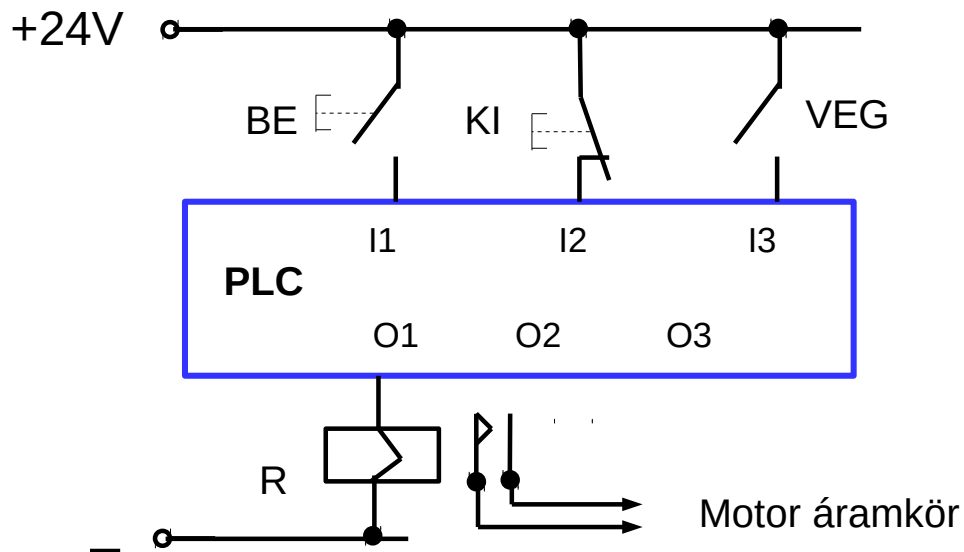
IF N **KI** THEN RESET R1

2.10. Minta feladatok

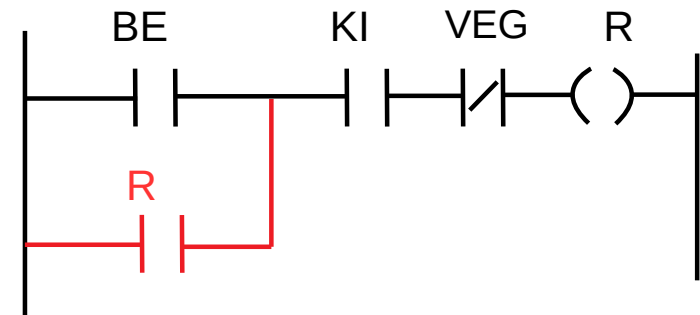
1. mintafeladat

- a „BE” (I1) nyomógombot lenyomva „R” (O1) meghúz (öntartás!) és indít egy motort.
- „KI” (I2) nyomógomb lenyomására „R” elenged (motor leáll).
- „VEG” (I3) végálláskapcsoló jelzésére a motor szintén leáll.

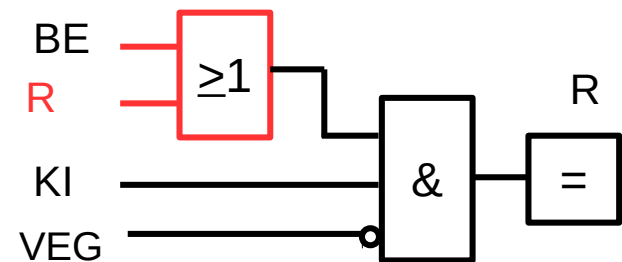
A hardver:



LAD



FB



STL (FESTO)

```
IF ( BE OR R ) AND KI AND N VEG
  THEN SET O1
  OTHRW RESET O1
```

2.11. Minta feladatok

2. mintafeladat

Az alábbi igazságtáblázat alapján írd meg O1 és O2 kimenetek vezérlésének programját, több nyelven is !

I1	I2	I3	O1	O2
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	0	0
1	1	1	1	0

egyszerűsítünk,
ha lehet

O1

		I2 I3			
I1		00	01	11	10
	0	0 ₀	0 ₁	1 ₃	1 ₂
	1	0 ₄	0 ₅	1 ₇	0 ₆

$$O1 = \overline{I1} * I2 + I2 * I3$$

O2

		I2 I3			
I1		00	01	11	10
	0	0 ₀	1 ₁	0 ₃	0 ₂
	1	1 ₄	1 ₅	0 ₇	0 ₆

$$O2 = I1 * \overline{I2} + \overline{I2} * I3$$

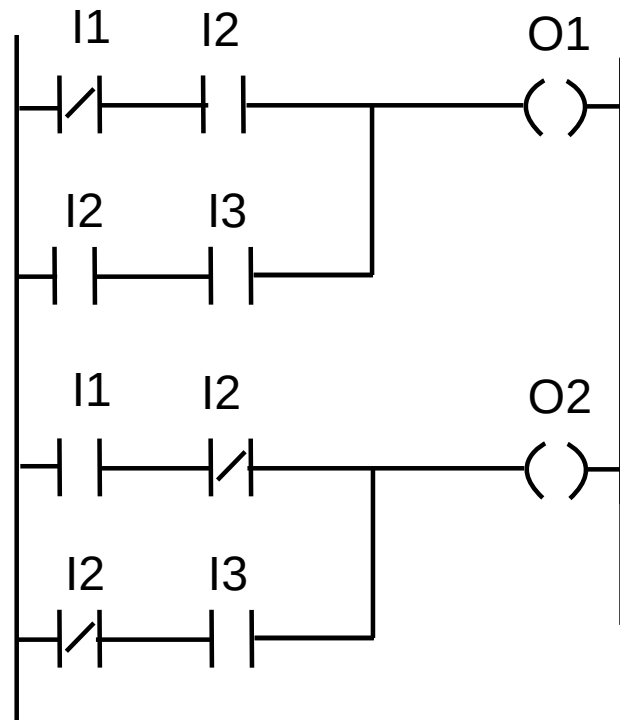
2.12. Minta feladatok

2. mintafeladat, megoldás

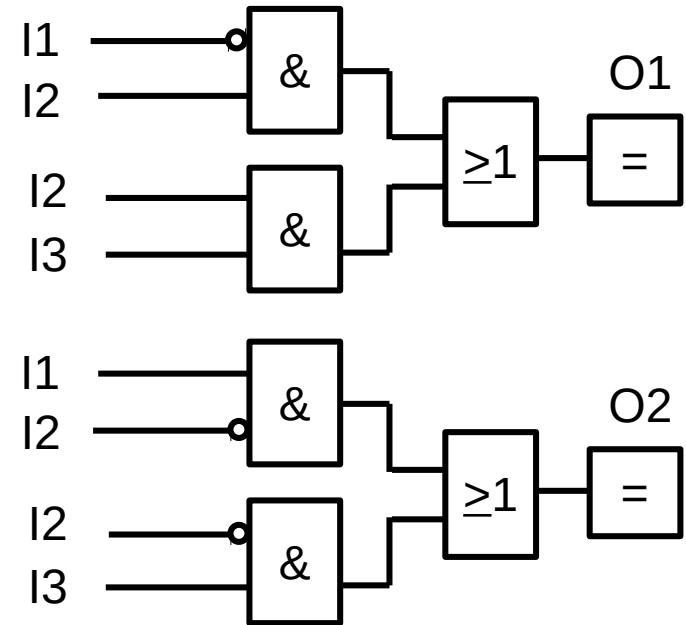
$$O1 = \bar{I1} * I2 + I2 * I3$$

$$O2 = I1 * \bar{I2} + \bar{I2} * I3$$

LAD



FB



STL (FESTO)

IF **N** I1 **AND** I2 **OR** I2 **AND** I3
THEN SET O1
OTHRW RESET O1

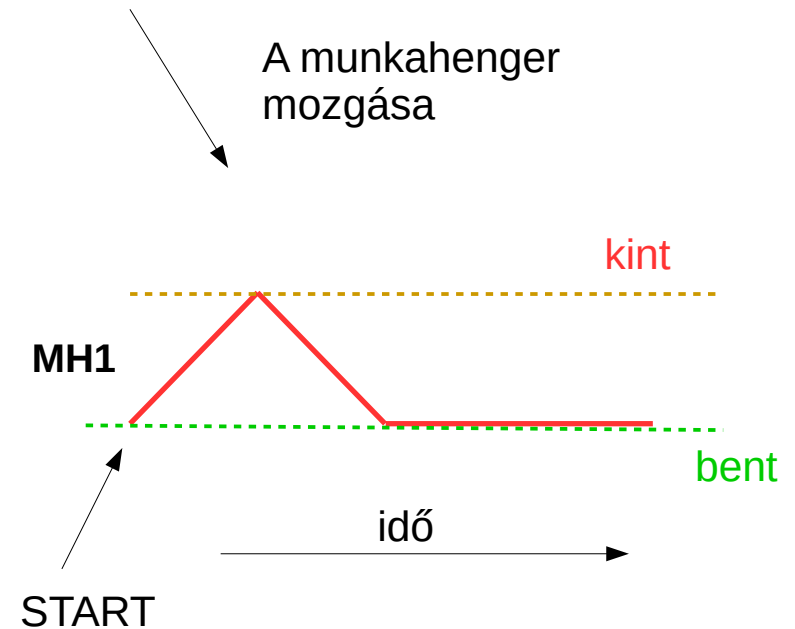
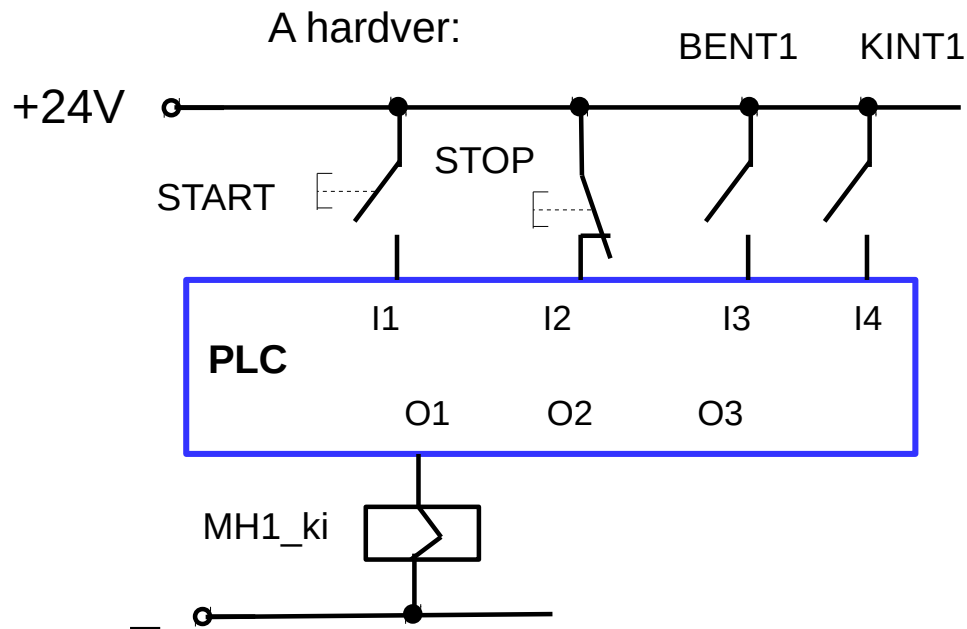
IF I1 **AND** **N** I2 **OR** **N** I2 **AND** I3
THEN SET O2
OTHRW RESET O2

2.13. Minta feladatok

3. mintafeladat

Egy pneumatikus munkahengert (MH1) kell a következőképpen működtetni:

- kettős működésű munkahenger, monostabil 5/2-es útváltóval vezérelve
- az útváltó mágnesszelepére (MH1_ki, O1) feszültséget kapcsolva a munkahengert kitoljuk, a feszültséget lekapcsolva a munkahengert visszahelyezzük alaphelyzetbe.
- a munkahenger két véghelyzetét két végálláskapcsoló jelzi → BENT1 (I3) és KINT1 (I4)
- „START” (I1) nyomógomb lenyomására a munkahengert toljuk ki (mágnesszelepre feszültség).
- ha a munkahengert teljesen kitoltuk → automatikusan azonnal menjen vissza alaphelyzetbe (mágnesszelepről a feszültség lekapcsolása)
- „STOP” (I2) nyomógomb lenyomására a munkahenger azonnal menjen alaphelyzetbe !



2.14. Minta feladatok

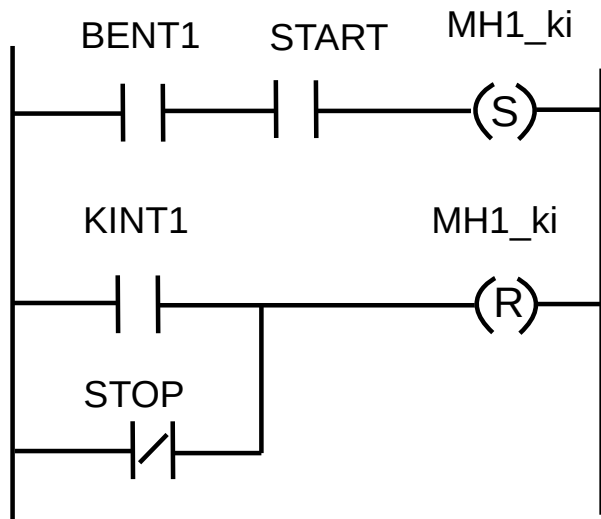
3. mintafeladat, megoldás

Itt is kell öntartás !! Mert a végálláskapcsolók csak a végpozíciókban jeleznek, és ne kelljen egyfolytában nyomni a Start ill. Stop gombot

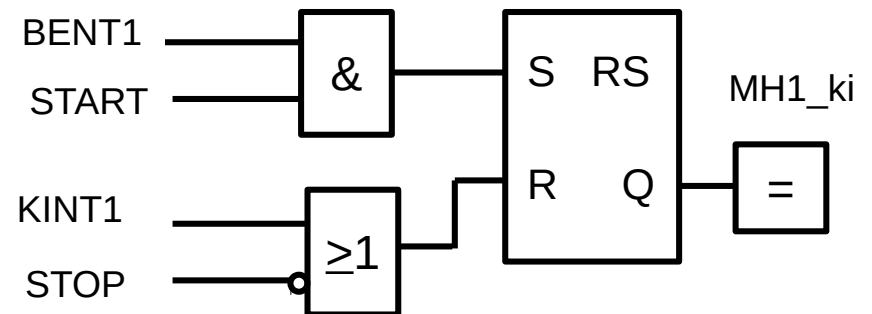
MH1



LAD



FB



STL (FESTO)

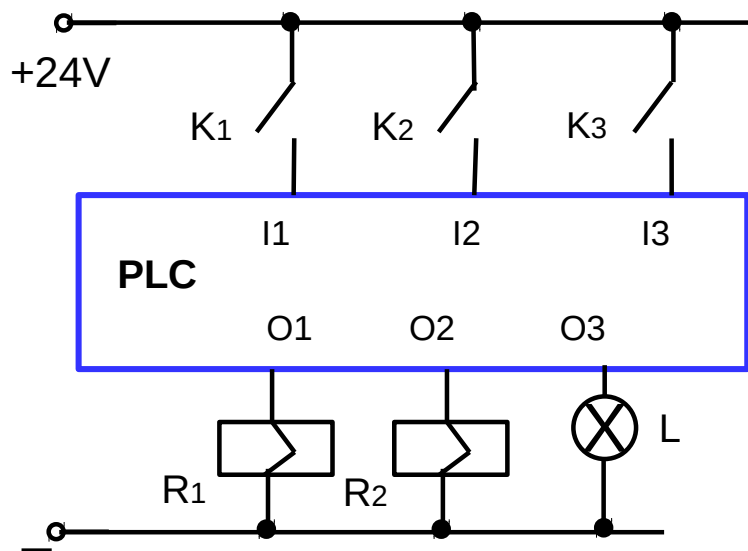
```
IF BENT1 AND START
    THEN SET MH1_ki
IF KINT1 OR STOP
    THEN RESET MH1_ki
```

2.15. Minta feladatok

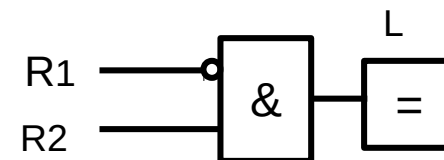
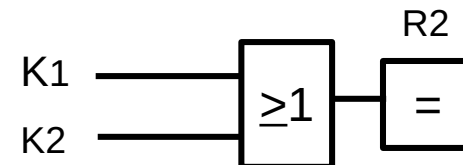
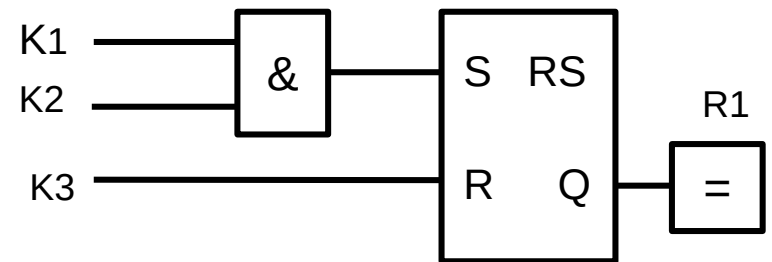
4. mintafeladat

- Ha K1 és K2 kapcsolókat zárjuk → R1 relé feszültséget kap, a relé meghúz és csak akkor enged el ha K3 kapcsolót zárjuk ! (öntartás)
- R2 relére feszültséget kell kapcsolni ha K2 vagy K1 zárva
- L izzó világít ha R2 relé működtetve van, de R1 nem

A hardver:



Megoldás, FB

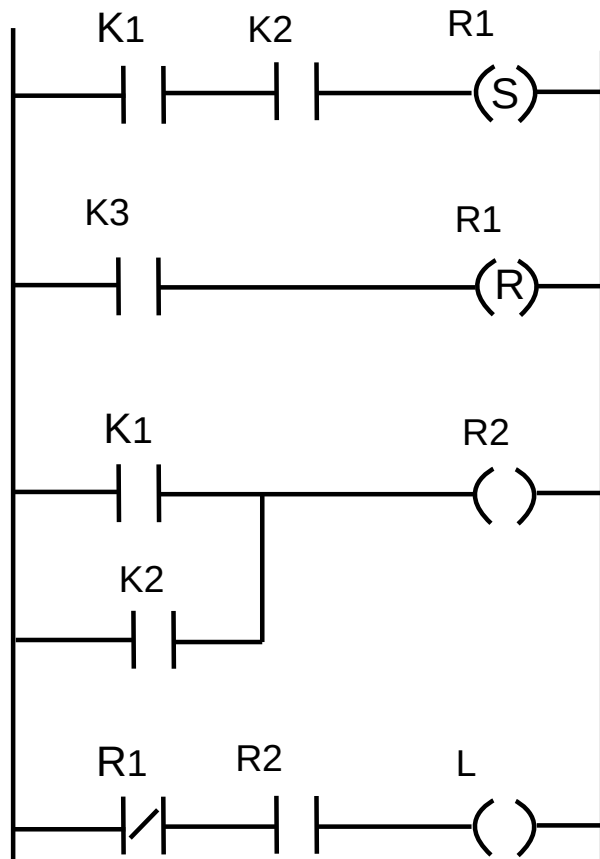


2.16. Minta feladatok

4. mintafeladat, folytatás

- Ha K1 és K2 kapcsolókat zárjuk → R1 relé feszültséget kap, a relé meghúz és csak akkor enged el ha K3 kapcsolót zárjuk ! (öntartás)
- R2 relére feszültséget kell kapcsolni ha K2 vagy K1 zárva
- L izzó világít ha R2 relé működtetve van, de R1 nem

Megoldás,LAD



Megoldás,STL (FESTO)

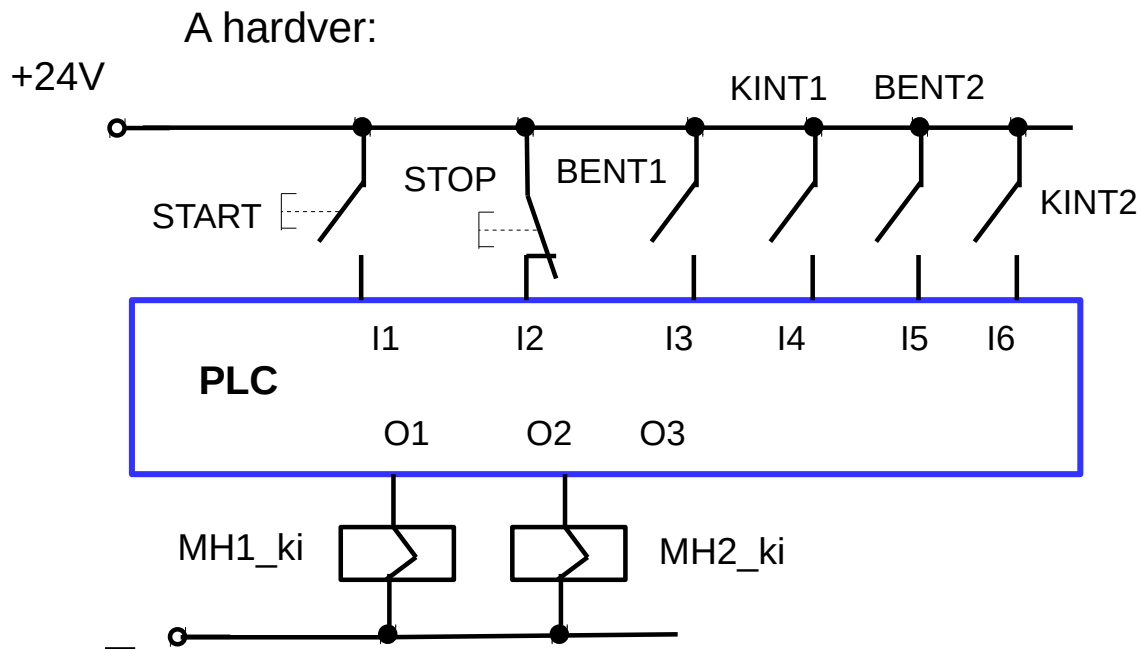
```
IF K1 AND K2
    THEN SET R1
IF K3
    THEN RESET R1
IF K1 OR K2
    THEN SET R2
    OTHRW RESET R2
IF NOT R1 AND R2
    THEN SET L
    OTHRW RESET L
```

2.17. Minta feladatok

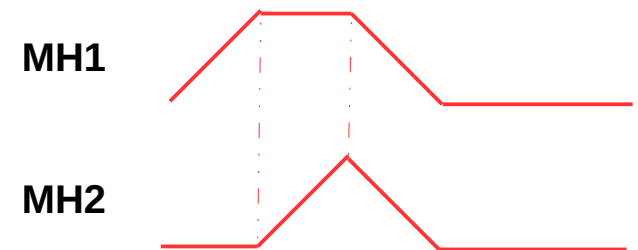
5. mintafeladat

Két pneumatikus munkahengert (MH1, MH2) kell a következőképpen működtetni:
(kettős működésű munkahengerek, monostabil 5/2-es útváltókkal vezérelve)

- az útváltók mágnesszelepei → MH1_ki (O1) és MH2_ki (O2)
- a munkahengerek két véghelyzetét jelző végálláskapcsolók →
BENT1 (I3) KINT1 (I4) BENT2 (I5) és KINT2 (I6)
- „START” (I1) nyomógomb lenyomására az MH1 munkahengert toljuk ki.
- ha MH1 munkahengert teljesen kitoltuk → automatikusan menjen ki a másik munkahenger is (MH2)
- ha mindkét munkahengert teljesen kitoltuk → automatikusan azonnal menjenek alaphelyzetbe
- „STOP” (I2) nyomógomb lenyomására mindkét munkahenger azonnal menjen alaphelyzetbe !



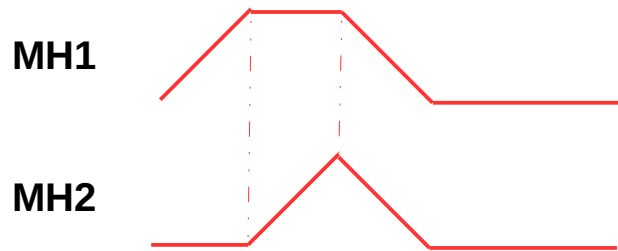
A munkahengerek mozgása (ciklus diagram)



2.18. Minta feladatok

5. mintafeladat, megoldás

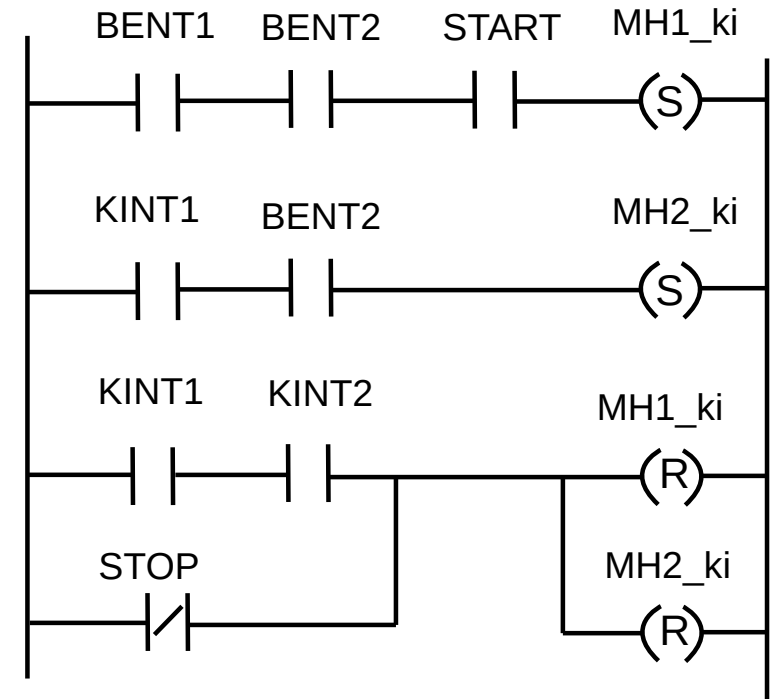
A munkahengerek
mozgása (ciklus diagram)



Működtető függvények

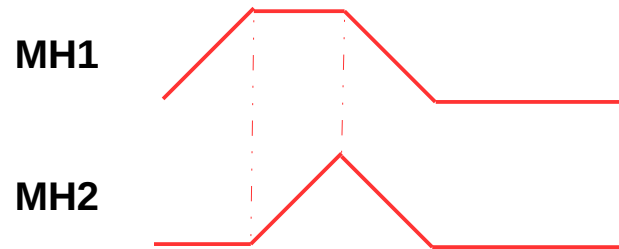
- MH1 meghúzatása:
 $MH1 + = START * BENT1 * BENT2$
- MH2 meghúzatása:
 $MH2 + = KINT1 * BENT2$
- MH1 elengedése:
 $MH1 - = KINT1 * KINT2 + \overline{STOP}$
- MH2 elengedése:
 $MH2 - = KINT1 * KINT2 + \overline{STOP}$

LAD



2.19. Minta feladatok

5. mintafeladat, megoldás folytatás



Működtető függvények

- MH1 meghúzatása:

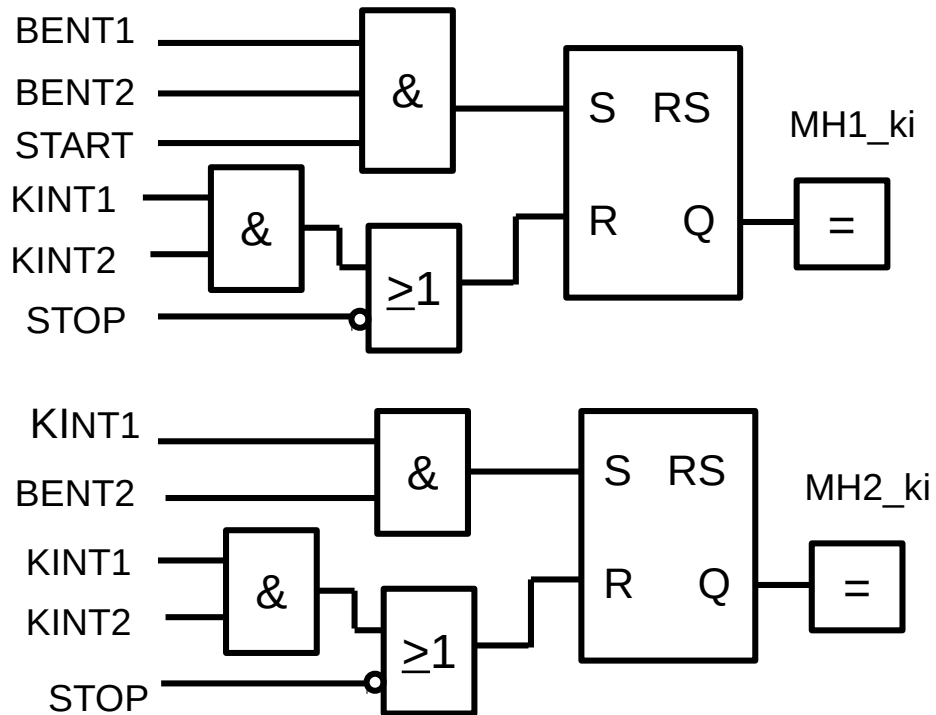
$$MH1 + = START * BENT1 * BENT2$$
- MH2 meghúzatása:

$$MH2 + = KINT1 * BENT2$$
- MH1 elengedése:

$$MH1 - = KINT1 * KINT2 + \overline{STOP}$$
- MH2 elengedése:

$$MH2 - = KINT1 * KINT2 + \overline{STOP}$$

FB



STL (FESTO)

```

IF BENT1 AND BENT2 AND START
    THEN SET MH1_ki
IF KINT1 AND BENT2
    THEN SET MH2_ki
IF KINT1 AND KINT2 OR NOT STOP
    THEN RESET MH1_ki
    RESET MH2_ki
    
```


2.20. Feladatok

1. feladat

Az alábbi igazságtáblázat alapján
írd meg O1 és O2 kimenetek
vezérlésének programját,
több nyelven is !

I1	I2	I3	O1	O2
0	0	0	1	1
0	0	1	1	0
0	1	0	0	0
0	1	1	1	1
1	0	0	0	1
1	0	1	0	0
1	1	0	0	1
1	1	1	0	1

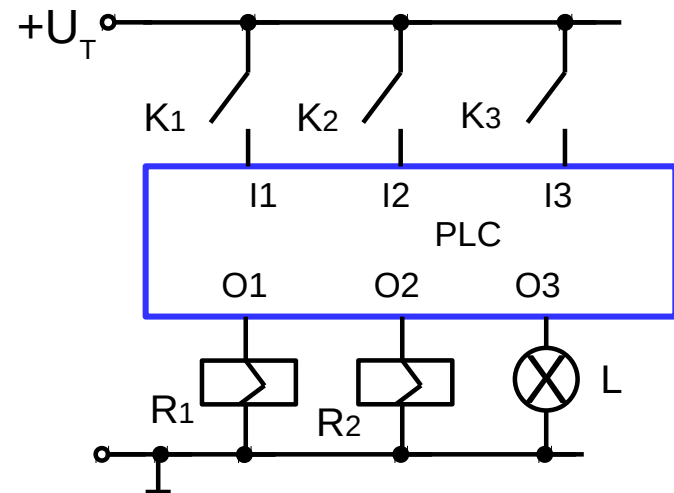
2. feladat

Az alábbi kapcsolásra írd meg
a következő vezérlést:

R2 relére feszültséget kell kapcsolni ha
K1 zárva És K2 nyitva És K3 zárva
R1 relére feszültséget kell kapcsolni ha
(K1 nyitva És K3 zárva) Vagy K2 zárva
L izzó világít ha
R1 És R2 relé működtetve van

A program elkészítése

- létra diagramban
- funkcióblokk programnyelven

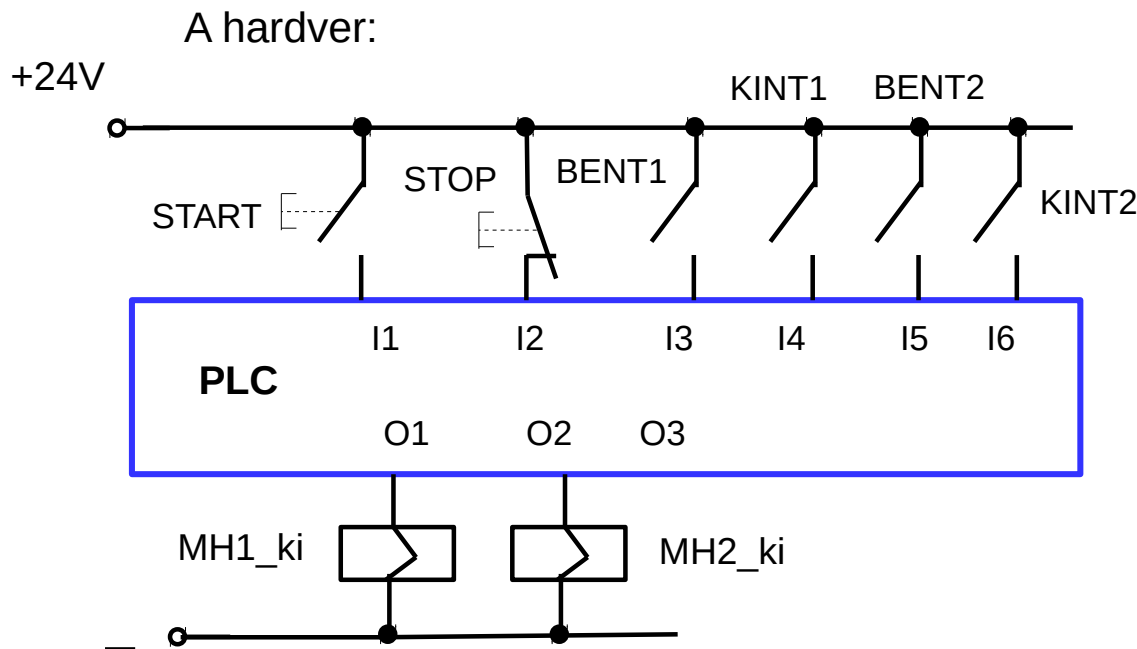


2.21. Feladatok

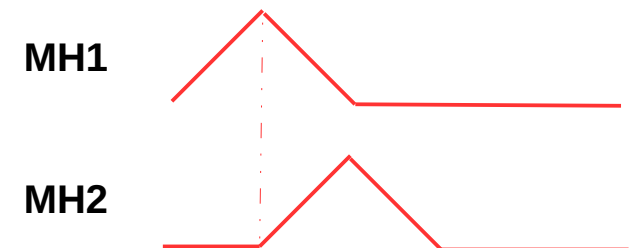
3. feladat

Két pneumatikus munkahengert (MH1, MH2) kell a következőképpen működtetni:
(kettős működésű munkahengerek, monostabil 5/2-es útváltókkal vezérelve)

- az útváltók mágnesszelepei → MH1_ki (O1) és MH2_ki (O2)
- a munkahengerek két véghelyzetét jelző végálláskapcsolók →
BENT1 (I3) KINT1 (I4) BENT2 (I5) és KINT2 (I6)
- „START” (I1) nyomógomb lenyomására az MH1 munkahengert toljuk ki.
- ha MH1 munkahengert teljesen kitoltuk →
MH1 menjen vissza alaphelyzetbe, és
automatikusan menjen ki a másik munkahenger (MH2)
- ha MH2 munkahengert teljesen kitoltuk → automatikusan azonnal menjen vissza alaphelyzetbe



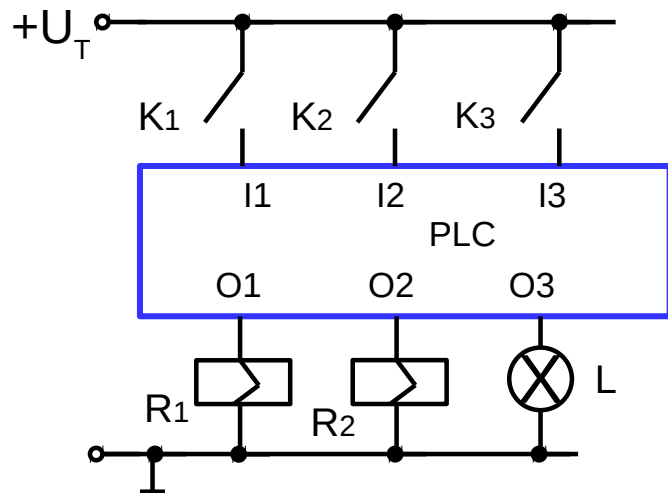
A munkahengerek mozgása (ciklus diagram)



2.22. Feladatok

4. feladat

Az alábbi kapcsolásra írd meg
a következő vezérlést:



- Ha K_1 És K_2 kapcsolókat zárjuk \rightarrow
R1 relére feszültséget kapcsolunk, a relé
meghúz és csak akkor enged el ha
 K_3 kapcsolót zárjuk ! (öntartás)
- R2 relére feszültséget kell kapcsolni ha
 K_2 Vagy K_1 zárva
- L izzó világít ha R2 relé működtetve van,
de R1 nem

A program elkészítése

- létra diagramban
- funkcióblokk programnyelven

2.23. Feladatok megoldásai

1. feladat

Az alábbi igazságtáblázat alapján írd meg O1 és O2 kimenetek vezérlésének programját, több nyelven is !

I1	I2	I3	O1	O2
0	0	0	1	1
0	0	1	1	0
0	1	0	0	0
0	1	1	1	1
1	0	0	0	1
1	0	1	0	0
1	1	0	0	1
1	1	1	0	1

egyszerűsítünk,
ha lehet

O1

	I2 I3			
	00	01	11	10
I1				
0	1 ₀	1 ₁	1 ₃	0 ₂
1	0 ₄	0 ₅	0 ₇	0 ₆

$$O1 = \bar{I}_1 * \bar{I}_2 + \bar{I}_1 * I_3$$

O2

	I2 I3			
	00	01	11	10
I1				
0	1 ₀	0 ₁	1 ₃	0 ₂
1	1 ₄	0 ₅	1 ₇	1 ₆

$$O2 = I_1 * I_2 + I_2 * I_3 + \bar{I}_2 * \bar{I}_3$$

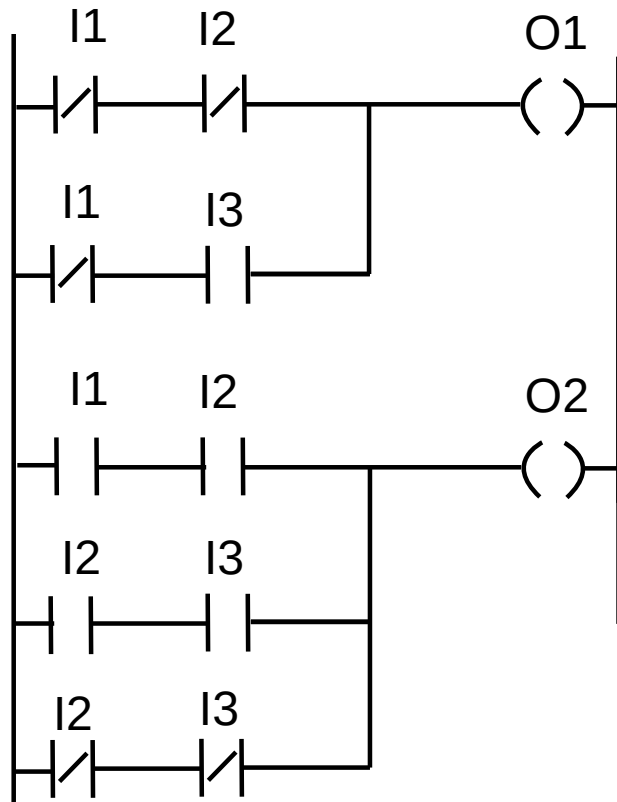
2.24. Feladatok megoldásai

1. feladat, megoldás

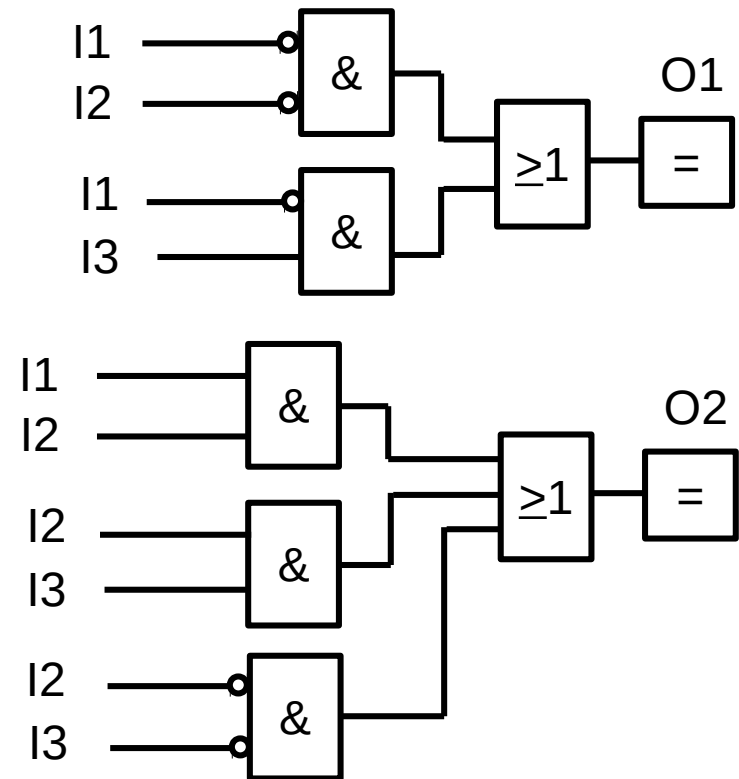
$$O1 = \bar{I1} * \bar{I2} + \bar{I1} * I3$$

$$O2 = I1 * I2 + I2 * I3 + \bar{I2} * \bar{I3}$$

LAD



FB



STL (FESTO)

IF **N** I1 **AND** I2 **OR** **N** I1 **AND** I3
THEN SET O1
OTHRW RESET O1

IF I1 **AND** I2 **OR** I2 **AND** I3
OR **N** I2 **AND** **N** I3
THEN SET O2
OTHRW RESET O2

3.1. Flag (Merker) használata

Flag

Flag (Merker): belső segédváltozó (se nem bemenet, se nem kimenet)

Felhasználása:

- számolásra (egy számérték tárolására)
- állapot tárolására !

Például Festo FC34 PLC → 10000db 16 bites flag (FW0, FW1, FW2,FW9999),

De ezek használhatók bitenként is → F0.0 F0.1F0.15

F1.0 F1.1 ...F1.15

.....
F9999.0 F9999.15

1 bites Flag

- állapot tárolására → pl. valami már megtörtént, vagy még nem
- lekérdezhető, mint egy digitális bemenet
- beállítható (Set), törölhető (Reset) mint egy digitális kimenet

Festo és Siemens PLC-k esetén F0.0 F0.1 ... F0.7 F1.0 F1.1 ...

Unitronics PLC-k esetén a flag helyett a **memory bit** elnevezés van, és sorba számozza a biteket !

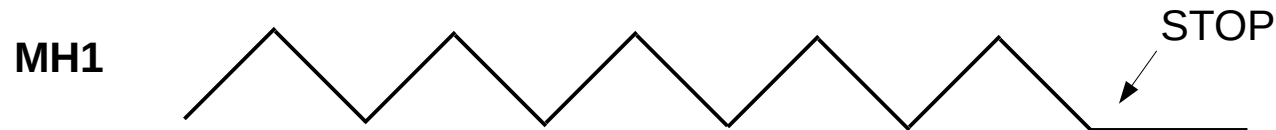
MB0 MB1 MB2 MB3 MB21 MB22

3.2. Flag (Merker) használata

1. mintafeladat

Egy pneumatikus munkahengert (MH1) kell a következőképpen működtetni:
(kettős működésű munkahenger, monostabil 5/2-es útváltóval vezérelve)

- a munkahenger két véghelyzetét két végálláskapcsoló jelzi → BENT1 (I3) és KINT1 (I4)
- „START” (I1) nyomógomb lenyomása után ! a munkahenger folyamatosan menjen ki-be
- „STOP” (I2) nyomógomb lenyomására a munkahenger menjen alaphelyzetbe, és álljon le !



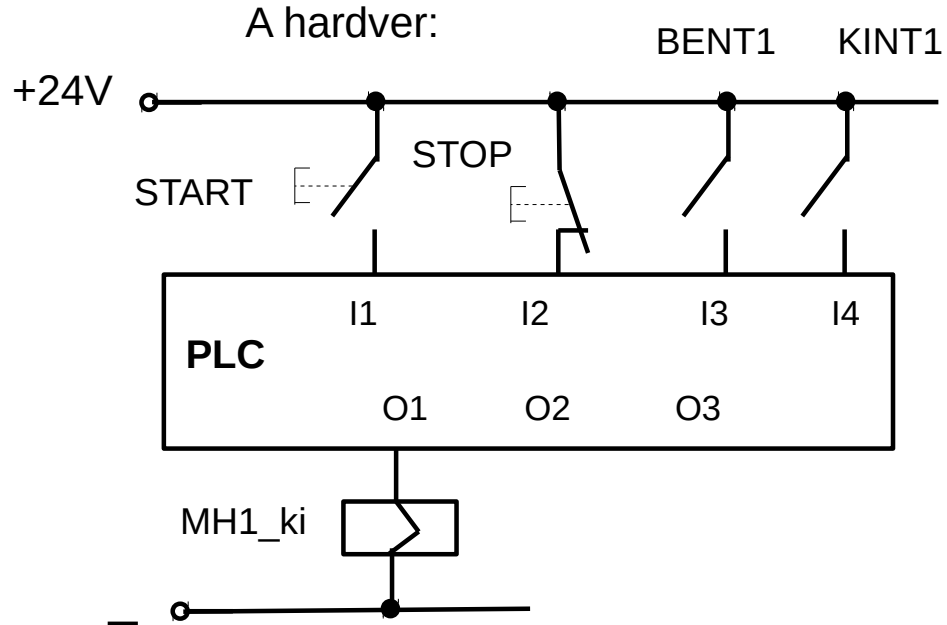
Megoldás Flag (memory bit) felhasználásával:

- egy Flag-et használunk arra hogy tároljuk: a START le volt-e már nyomva
- induláskor a Flag 0 értékű !! (ez nem biztos, célszerű program induláskor törölni !!)
→ ez fogja azt jelenteni, hogy még nem nyomtuk le a START gombot
- „START” nyomógomb lenyomására → a Flag értékét 1-be állítjuk → munkahenger mozgása
- „STOP” nyomógomb lenyomására a Flag értékét 0-ba állítjuk → munkahenger leállítása
- a munkahengert csak akkor küldjük kinti állásba, ha benti végállásban van ÉS a flag értéke 1-es !

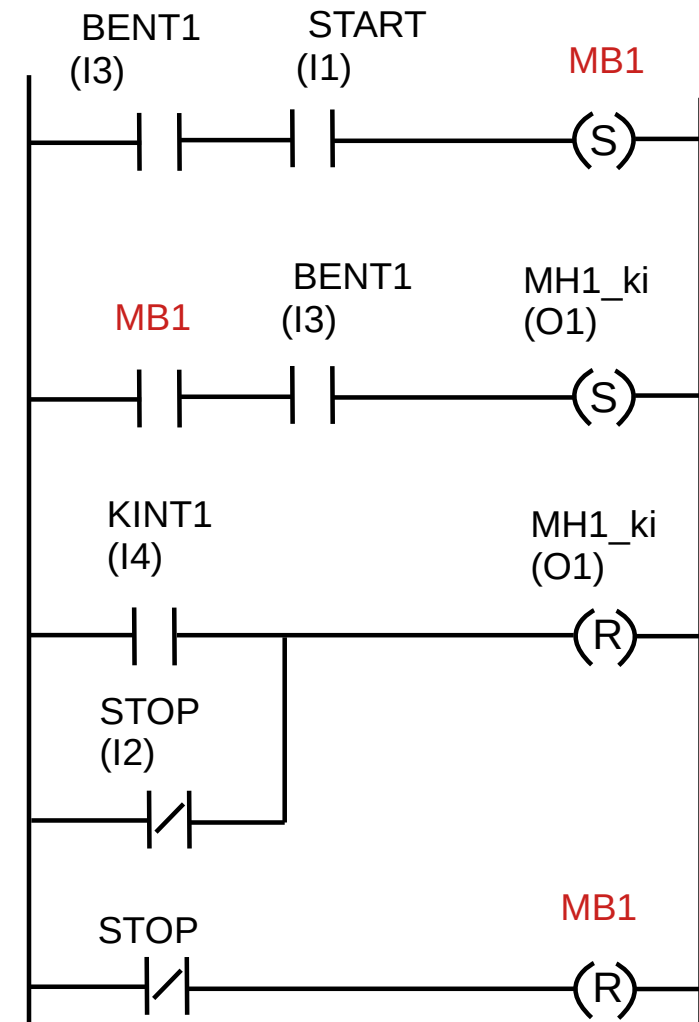
3.3. Flag (Merker) használata

1. mintafeladat megoldása

Unitronics PLC esetén



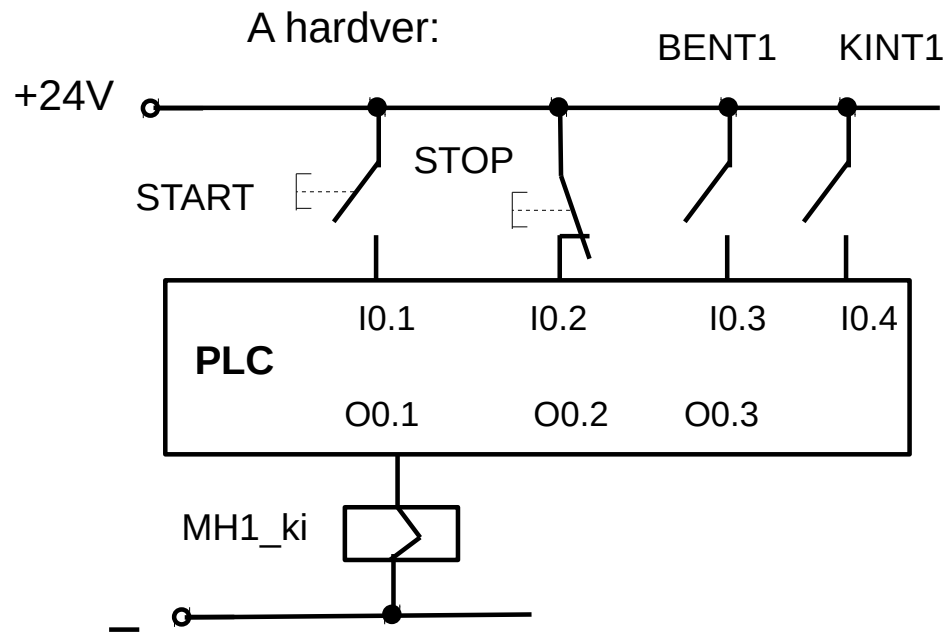
LAD



3.4. Flag (Merker) használata

1. mintafeladat megoldása

Festo PLC esetén



STL

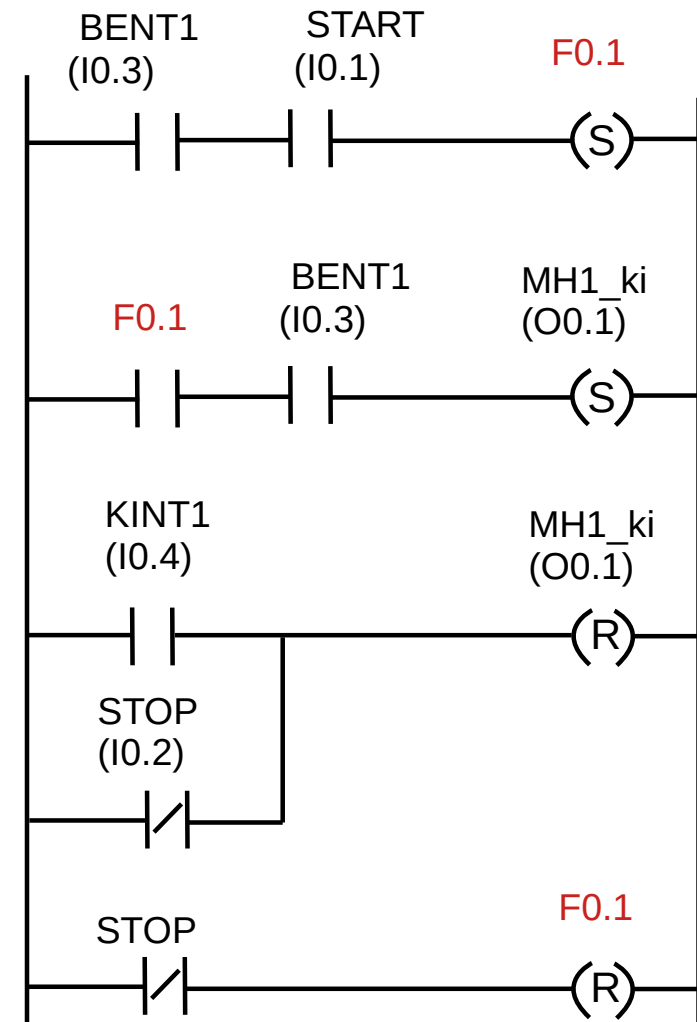
```
IF BENT1 AND START
    THEN SET F0.1

IF F0.1 AND BENT1
    THEN SET MH1_ki

IF KINT1 OR N STOP
    THEN RESET MH1_ki

IF N STOP
    THEN RESET F0.1
```

LAD

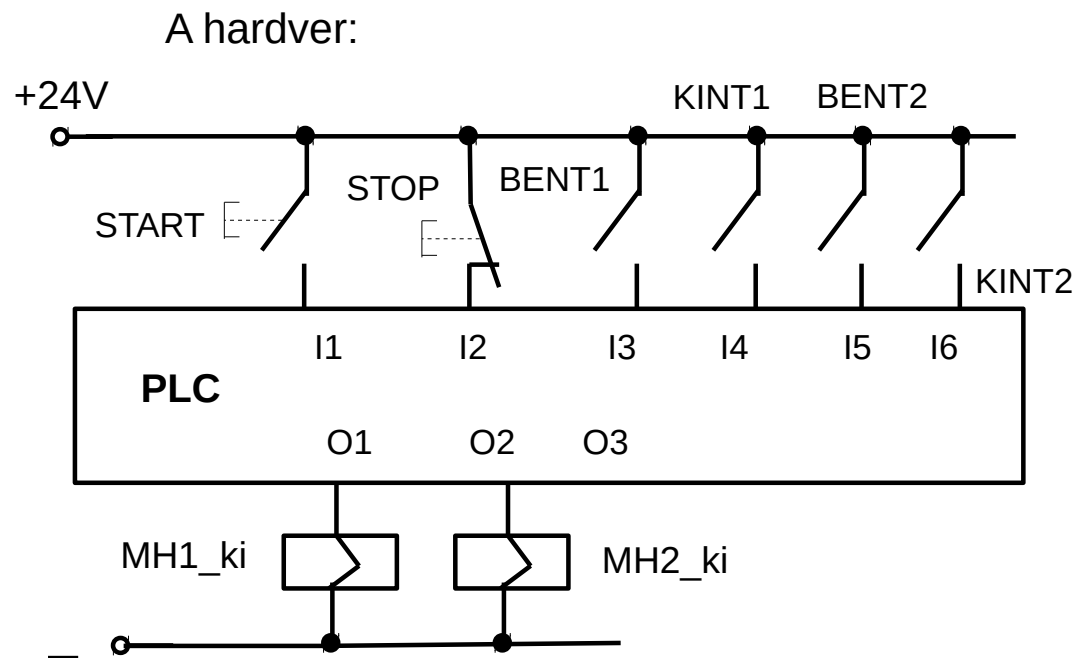


3.5. Flag használata, mintafeladatok

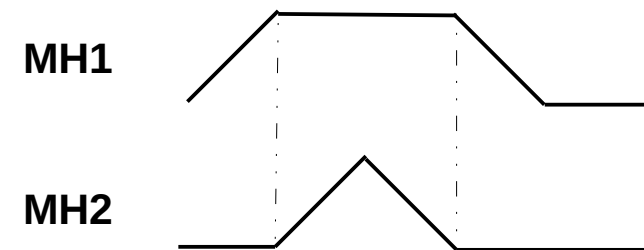
2. mintafeladat

Két pneumatikus munkahengert (MH1, MH2) kell a következőképpen működtetni:
(kettős működésű munkahengerek, monostabil 5/2-es útváltókkal vezérelve)

- „START” (I1) nyomógomb lenyomására az MH1 munkahengert toljuk ki.
- ha MH1 munkahengert teljesen kitöltük → automatikusan menjen ki a másik munkahenger is (MH2)
- ha MH2 munkahengert teljesen kitöltük → automatikusan azonnal menjen alaphelyzetbe
- ha MH2 munkahenger alaphelyzetbe ment → menjen alaphelyzetbe a másik munkahenger is (MH1)



A munkahengerek mozgása



Működtető függvények

- MH1 meghúzatása:
 $MH1 + = START * BENT1 * BENT2$
- MH2 meghúzatása:
 $MH2 + = KINT1 * BENT2 * \overline{FLAG}$
- MH1 elengedése:
 $MH1 - = KINT1 * BENT2 * FLAG$
- MH2 elengedése:
 $MH2 - = KINT2$

3.6. Flag használata, mintafeladatok

2. mintafeladat megoldása

A problémát az okozza, hogy flag nélkül, csak a bemeneti jelek alapján nem tudjuk megkülönböztetni **t1** és **t2** időpontokat (mindkettőnél MH1 kint van és MH2 bent) pedig mást kell csinálni azután !!

(t1 bekövetkezésekor → MH2-t ki kell tolni, míg t2 bekövetkezésekor → MH1-t vissza kell tolni)

- egy Flag-et használunk arra hogy tároljuk: MH2 már „kiment” → így a két idő pillanat megkülönböztethető

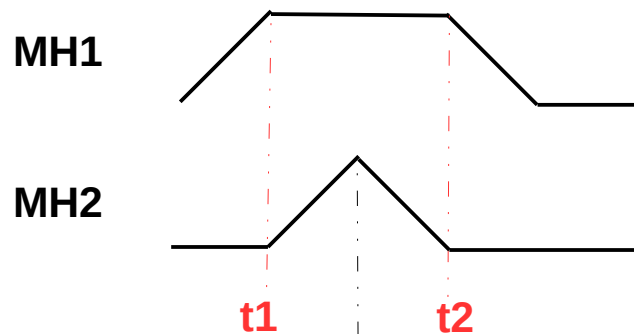
- induláskor a Flag 0 értékű !! →

„START” nyomógomb lenyomására →

a Flag értékét 0-ba állítjuk

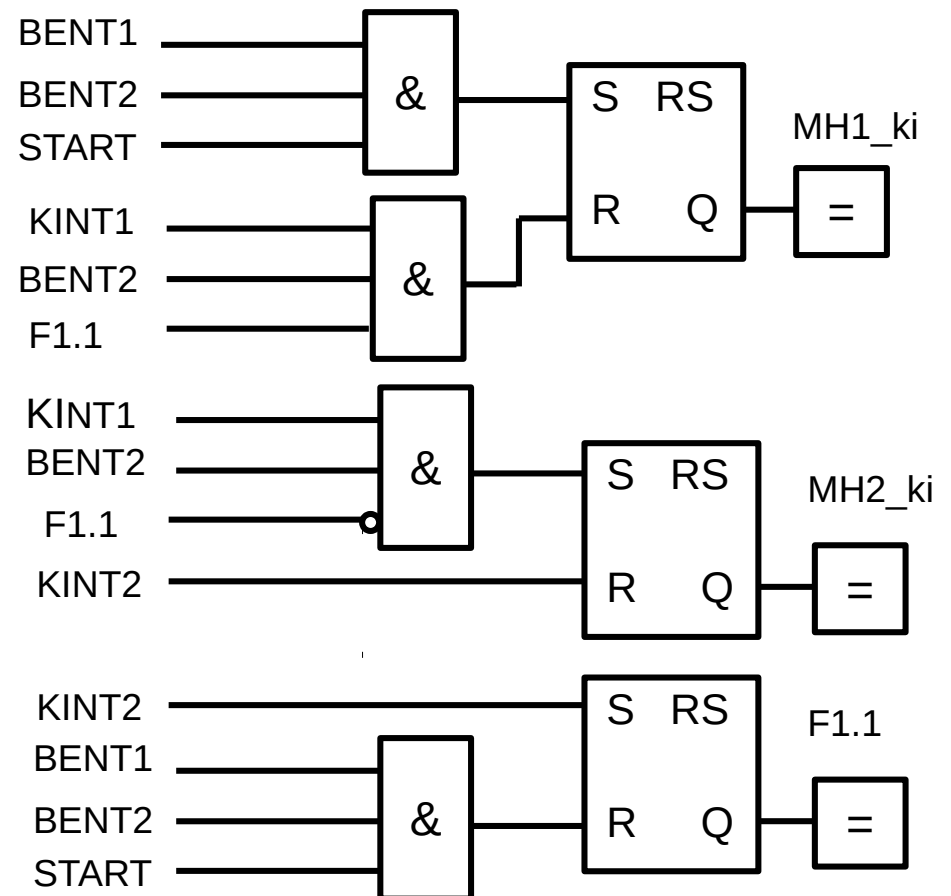
- ha MH2 már „kiment” a Flag értékét 1-be állítjuk

A munkahengerek
mozgása



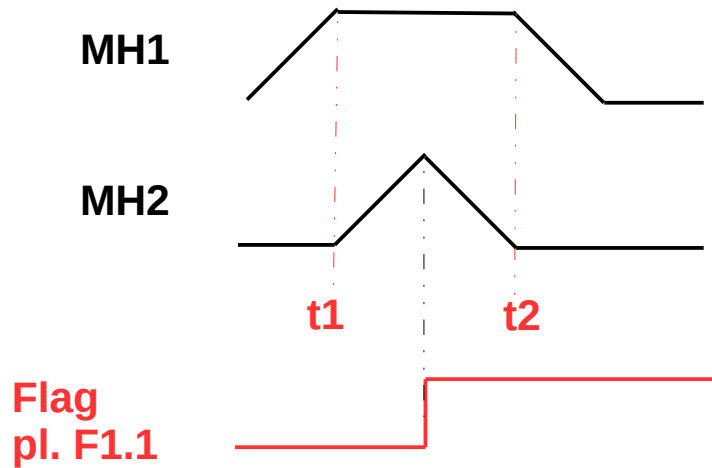
Flag
pl. F1.1
vagy MB1

FB



3.7. Flag használata, mintafeladatok

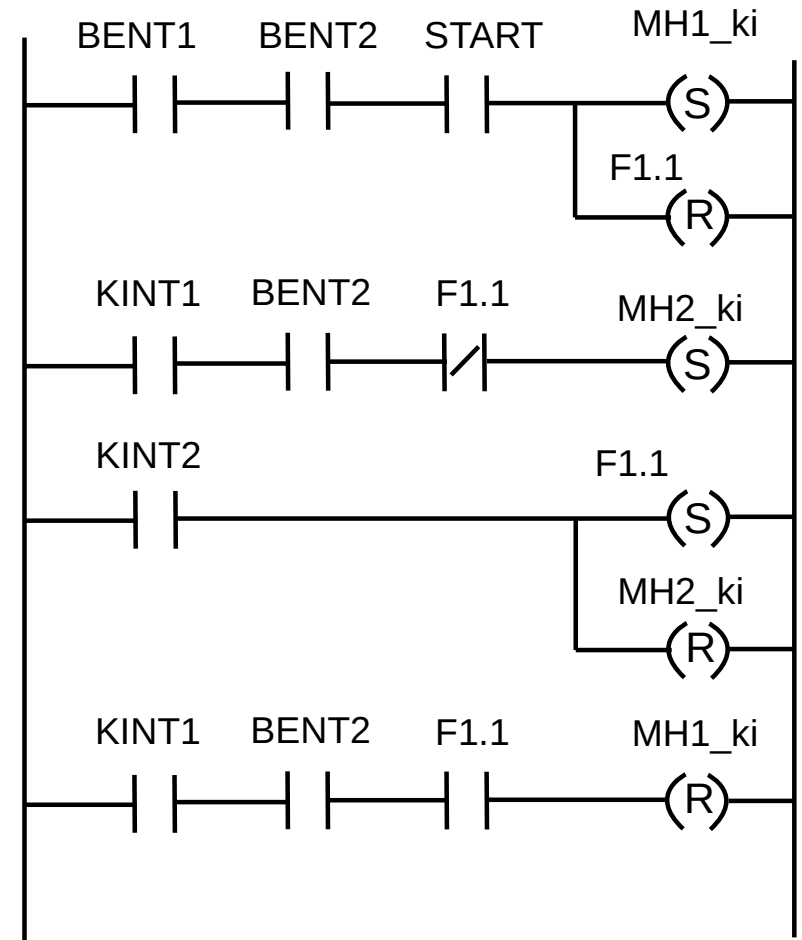
2. mintafeladat megoldása, folytatás



STL

```
IF BENT1 AND BENT2 AND START
  THEN SET MH1_ki
  RESET F1.1
IF KINT1 AND BENT2 AND N F1.1
  THEN SET MH2_ki
IF KINT2
  THEN RESET MH2_ki
  SET F1.1
IF KINT1 AND BENT2 AND F1.1
  THEN RESET MH1_ki
```

LAD

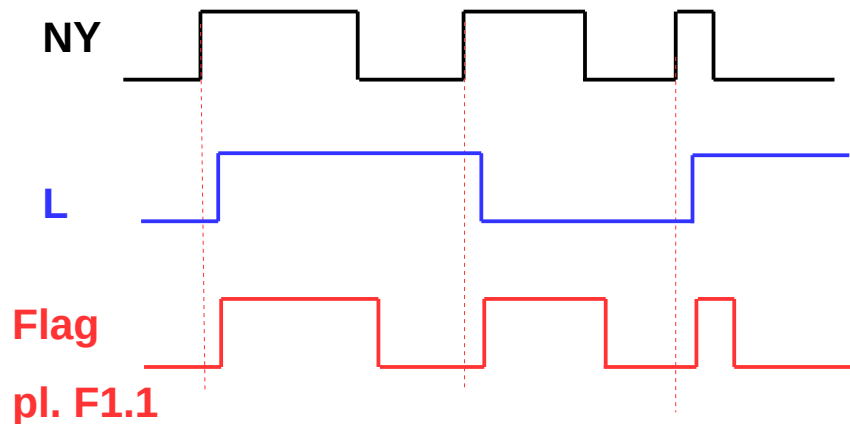


3.8. Flag használata, mintafeladatok

3. mintafeladat

Egy nyomógombbal (NY) vezérlünk egy lámpát (L)

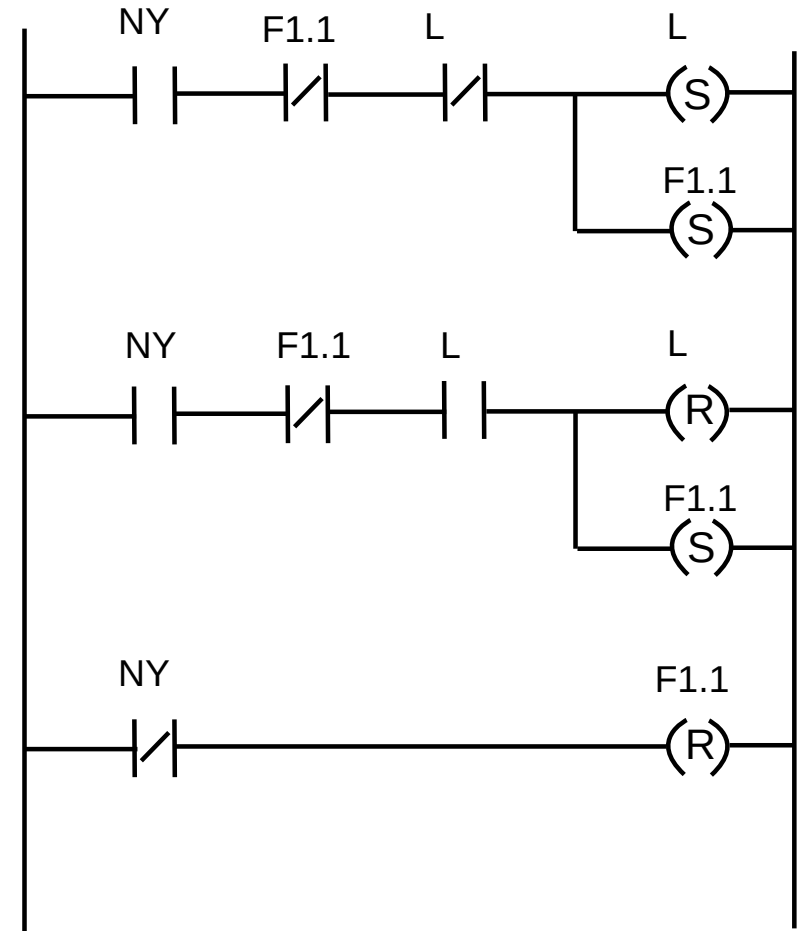
- a nyomógomb minden lenyomására a lámpa állapotot vált !
(ha le volt kapcsolva akkor felkapcsolódik, ha fel volt kapcsolva akkor lekapcsolódik)



Megoldás:

- Egy flag segítségével tároljuk a nyomógomb állapotát
- kihasználjuk, hogy a ciklikus program feldolgozás miatt a bemeneti változások érzékelése nem azonnali !! →
- ha NY=1 de Flag=0 (még) → NY felfutó él !!
→ ilyenkor kell a lámpát invertálni

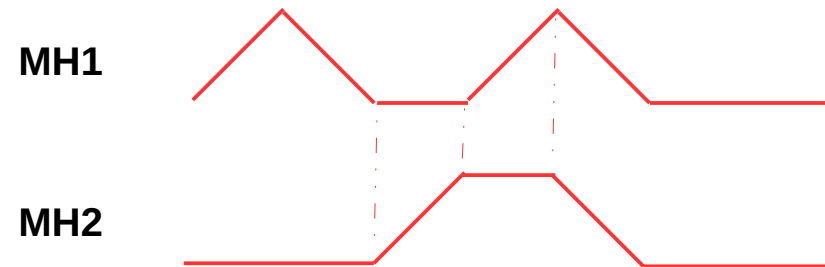
LAD



3.9. Feladatok

1. feladat

Két munkahengert (MH1, MH2) kell a következőképpen működtetni:



- kettős működésű munkahengerek, monostabil 5/2-es útváltóval vezérelve
- az útváltók vezérlése mágnesszelepekkel, a következő PLC kimenetekre kötve →
O0.0 (MH1_ki) O0.1 (MH2_ki)
- a munkahengerek helyzetét végálláskapcsolók jelzik, a következő PLC bemenetekre kapcsolva →
I0.0 (MH1_bent) I0.1 (MH1_kint) I0.2 (MH2_bent) I0.3 (MH2_kint)
- a munkavégzés „START” nyomógomb megnyomására kezdődik
START → a PLC I1.0 bemenetére kötve
- a „STOP” nyomógomb megnyomására a munkavégzés leáll, és minden alaphelyzetbe megy vissza
STOP → a PLC I1.1 bemenetére kötve

A program elkészítése

- létra diagramban
- és utasításlistában

3.10. Kezdeti beállítások

A PLC berendezések bemeneti, kimeneti, belső, speciális memória változói bekapcsoláskor meghatározott alapértékeket vesznek fel.

Gyakran van szükség arra, hogy a változók közül néhány más kezdeti értéket vegyen fel a program indulásakor. → Ennek megvalósításához a PLC rendelkeznek speciális (rendszer) változókkal

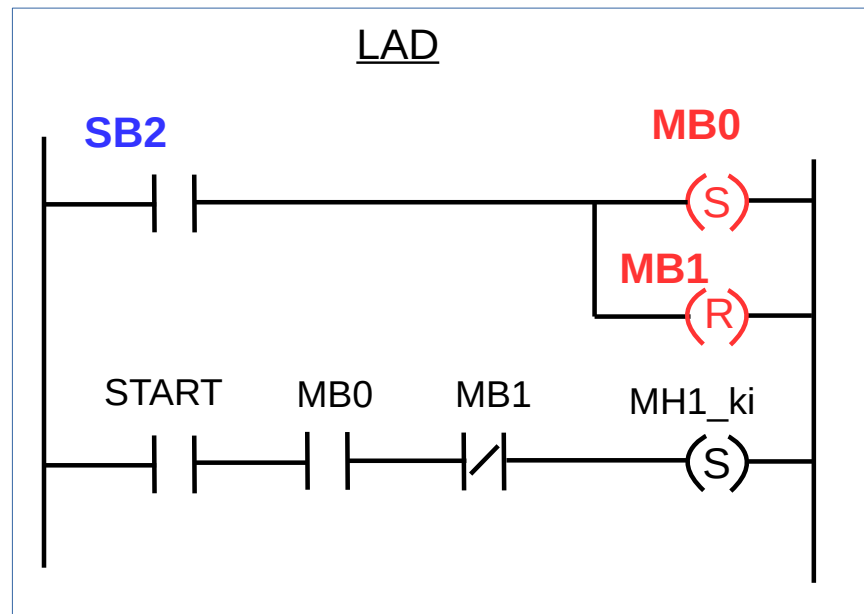
pl. Festo FC34 PLC

FI inicializációs flag → csak a legelső program ciklusban 1-es az értéke, azután mindig 0 !!

Unitronics Samba SM43-J-T20 PLC

SB 2 2. system bit (power-up bit)
→ csak a legelső program ciklusban 1-es az értéke, ezután mindig 0 !!

Felhasználási példa



← csak a legelső ciklusban hajtódik végre

4.1. Időzítő (Timer)

Timer

Időzítő, késleltető → megvalósítás számlálóval

- késleltetési idő = $n * \text{időalap}$
- n beállításával tudjuk megadni a késleltetést

$$t_k = n * T_i$$

Egyes PLC típusok esetén csak egyféle időalapú timer van, de van amelyik PLC többféle időalapú időzítővel rendelkezik !!

- például a Festo FC34, FC20 PLC-k → 256 db timer (T0, T1, T2,T255), de az időalap mindegyiknél 10ms
- a Siemens S7-200 típusú PLC esetén → szintén 256 db timer, de ezek nem egyformák, háromféle T_i van → 1ms, 10ms, 100ms

Használatuk

- használatuk előtt be kell állítani az időzítést
 - vagy n értékét kell megadni, vagy közvetlenül a késleltetési időt
- valamilyen logikai feltétellel el kell indítani (mintha kimenet lenne !)
- „figyelni” kell, hogy letelt-e az idő ? → lekérdezés (mintha bemenet lenne !)

Időzítő típusok

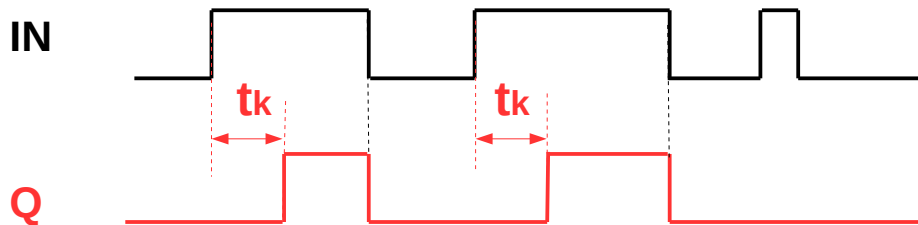
- bekapcsolás késleltetés → On-Delay Timer
- kikapcsolás késleltetés → Off-Delay Timer
- impulzus időzítő → Pulse Timer
- * Extended Pulse Timer (hasonló mint a Pulse Timer)
- * Accumulated Timer (részidők összegzése ?)

4.2. Időzítő (Timer)

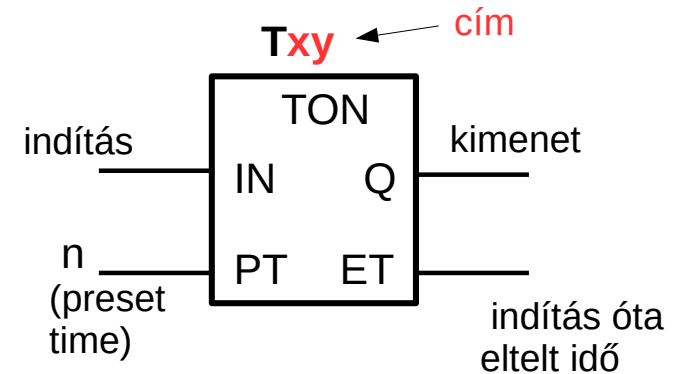
Bekapcsolás késleltetés (TON)

On-Delay Timer

A kimenet csak az indítás után t_k idő múlva lesz 1-es



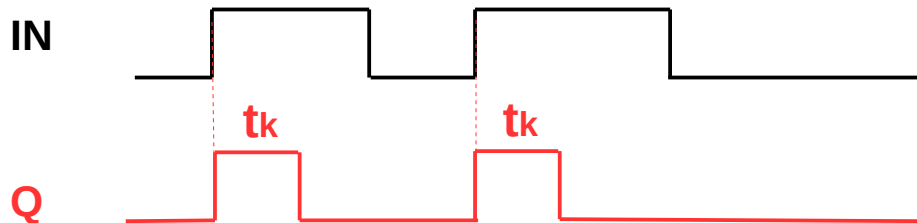
TON jelölése általában
(FB ill. LAD)



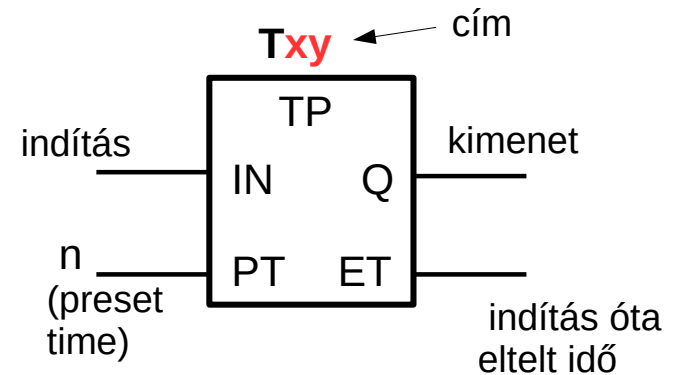
Impulzus időzítő (TP)

Pulse Timer

Indítás után fix szélességű (t_k) impulzust ad a kimenete



TP jelölése általában
(FB ill. LAD)

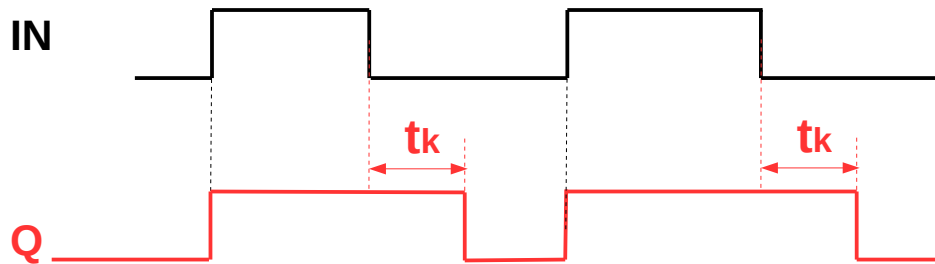


4.3. Időzítő (Timer)

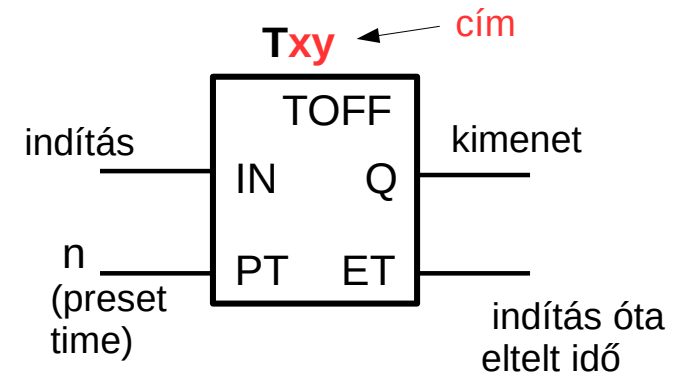
Kikapcsolás késleltetés (TOFF)

Off-Delay Timer

A kimenet csak az indítás megszűnése után t_k idővel lesz újra 0-ás



TOFF jelölése általában (FB ill. LAD)

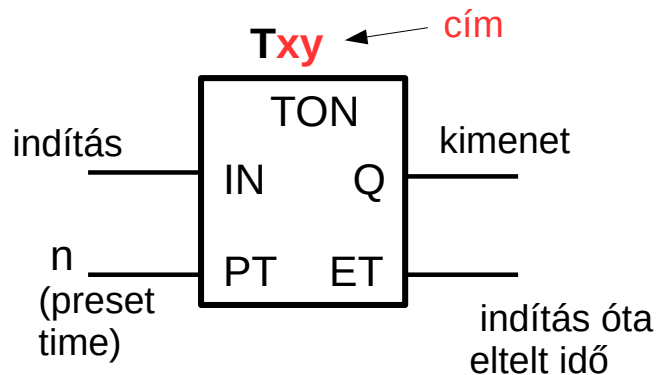


4.4. Időzítő, TON

Bekapcsolás késleltetés (TON)

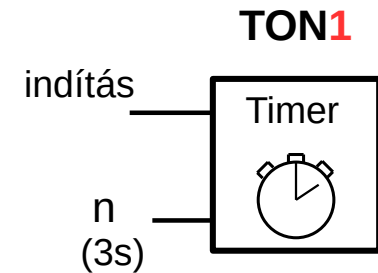
On-Delay Timer, (TON vagy ODT vagy SD vagy TD)

A kimenet csak az indítás után **tk** idő múlva lesz 1-es



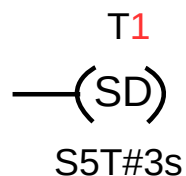
De egyes PLC-k esetén a jelölések eltérőek egy picit

pl. Festo FST (LAD)

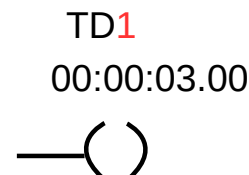


Lehet egyszerűsített jelölése is
→ speciális kimenet
On-delay timer coil

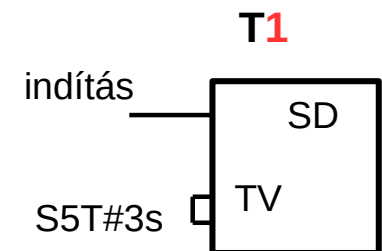
pl. Siemens
STEP7 (LAD)



pl. Unitronics
VisiLogic (LAD)



pl. Siemens STEP7
(FB)

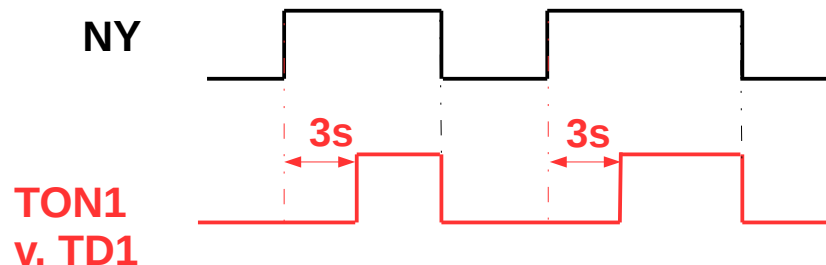
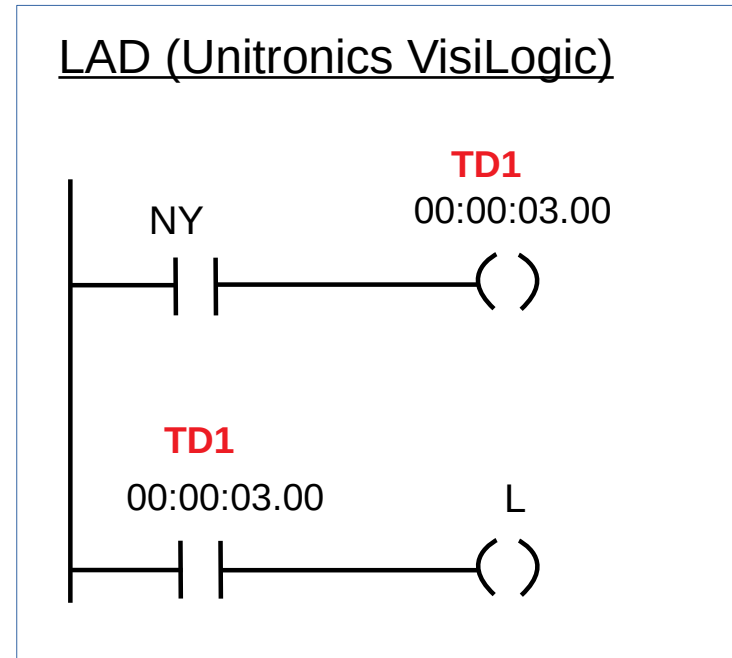
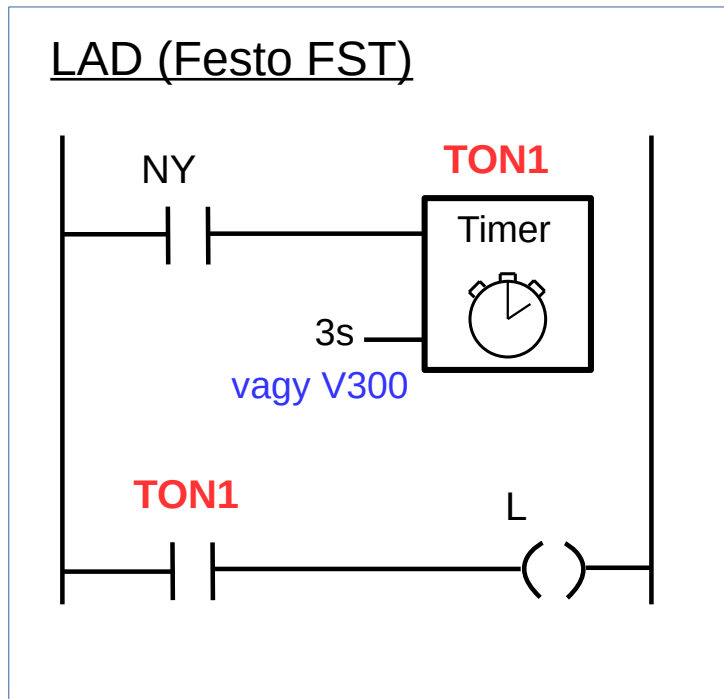


4.5. Időzítő, TON

1. mintafeladat (bekapcsolás késleltetés)

Egy nyomógombbal (NY) vezérlünk egy lámpát (L)

- a nyomógomb lenyomása után (azt lenyomva tartva) csak 3s leteltével kapcsoljon fel a lámpa
- a nyomógomb elengedésekor azonnal kapcsoljon le a lámpa



V300 értelmezése:
 $n=300$

$$t_k = 300 * T_i$$

$$t_k = 300 * 10\text{ms} = 3\text{s}$$

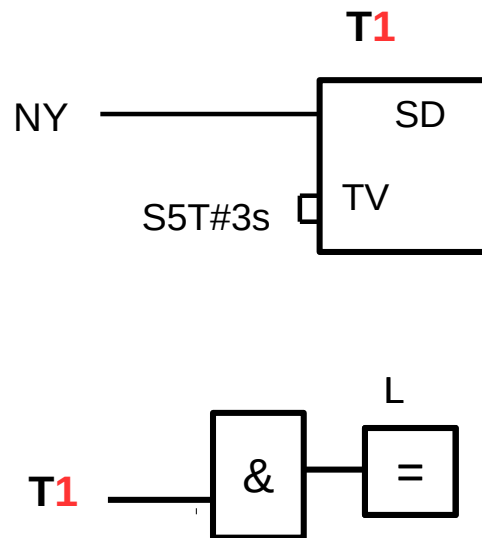
4.6. Időzítő, TON

1. mintafeladat, folytatás (bekapcsolás késleltetés)

Egy nyomógombbal (NY) vezérlünk egy lámpát (L)

- a nyomógomb lenyomása után (azt lenyomva tartva) csak 3s leteltével kapcsoljon fel a lámpa
- a nyomógomb elengedésekor azonnal kapcsoljon le a lámpa

FB (Siemens STEP7)

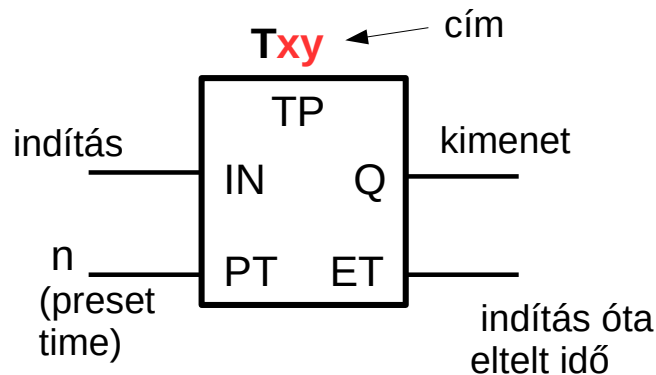


4.7. Időzítő, TP

Impulzus időzítő (TP)

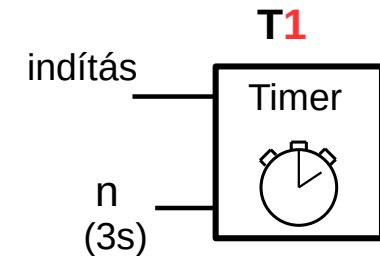
Pulse Timer, (TP vagy T vagy SP vagy TE)

Indítás után fix szélességű (tk) impulzust ad a kimenete



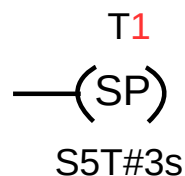
De egyes PLC-k esetén
a jelölések eltérőek egy
picit

pl. Festo FST (LAD)

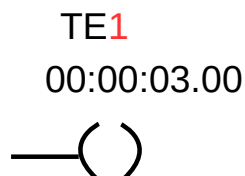


Lehet egyszerűsített jelölése is
→ speciális kimenet
Pulse timer coil

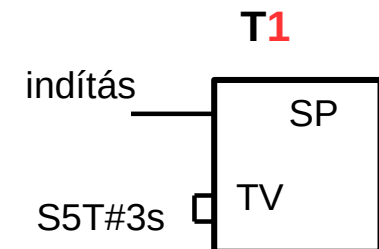
pl. Siemens
STEP7 (LAD)



pl. Unitronics VisiLogic (LAD)
Extended pulse timer!



pl. Siemens STEP7
(FB)

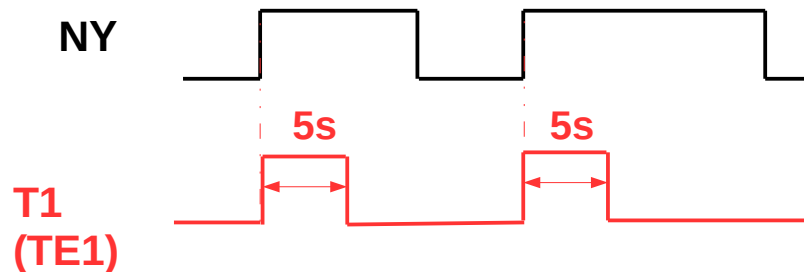
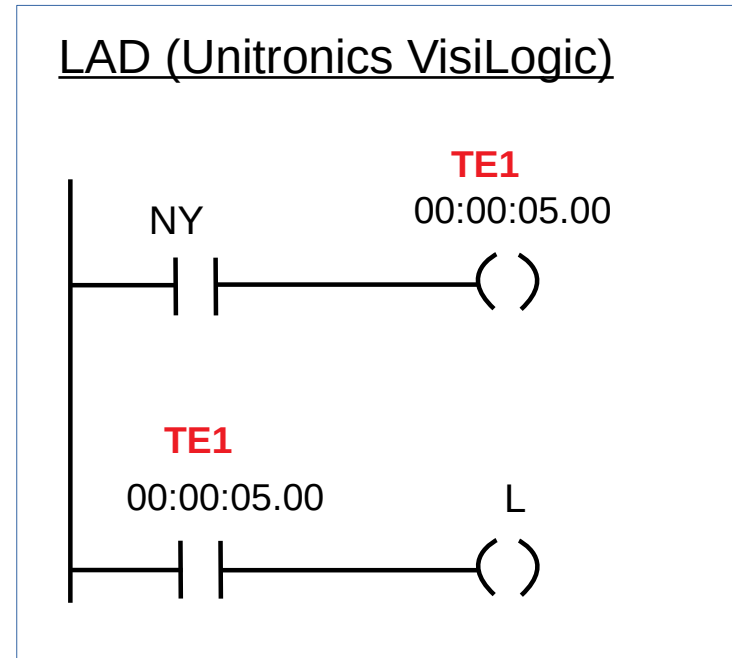
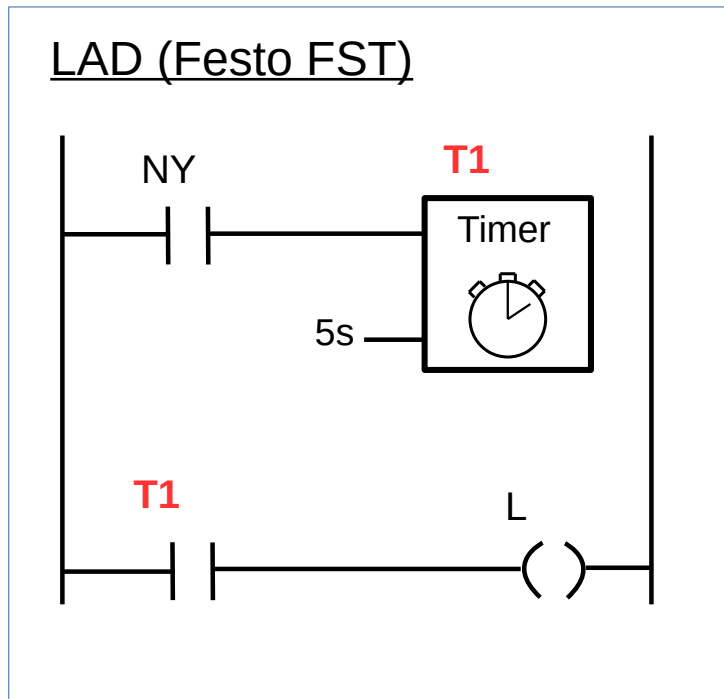


4.8. Időzítő, TP

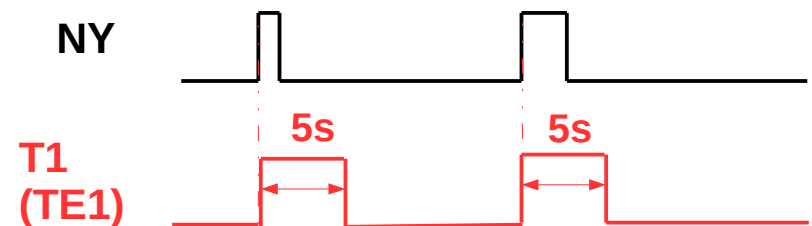
2. mintafeladat (impulzus időzítő)

Egy nyomógombbal (NY) vezérlünk egy lámpát (L)

- a nyomógomb lenyomása után (azt lenyomva tartva) azonnal kapcsoljon fel a lámpa, de → 5 másodperc után kapcsoljon le !! (mindegy, hogy a nyomógombot előbb, vagy később engedjük fel !)



Vagy!!



4.9. Időzítő, TP

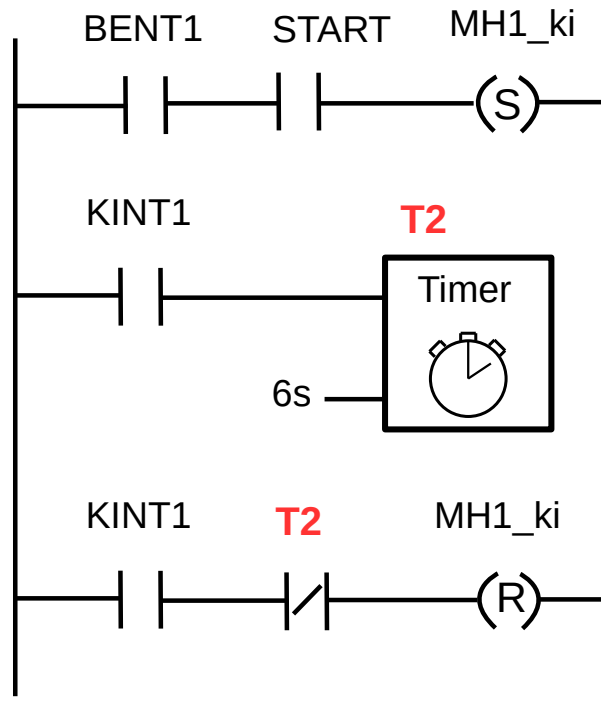
3. mintafeladat

Egy pneumatikus munkahengert (MH1) kell a következőképpen működtetni:
(kettős működésű munkahenger, monostabil 5/2-es útváltóval vezérelve)

- „START” (I1) nyomógomb lenyomására az MH1 munkahengert (O1) toljuk ki.
- ha MH1 munkahengert teljesen kitöltük → várakozzon 6 másodpercig →
→ majd ezután automatikusan menjen alaphelyzetbe

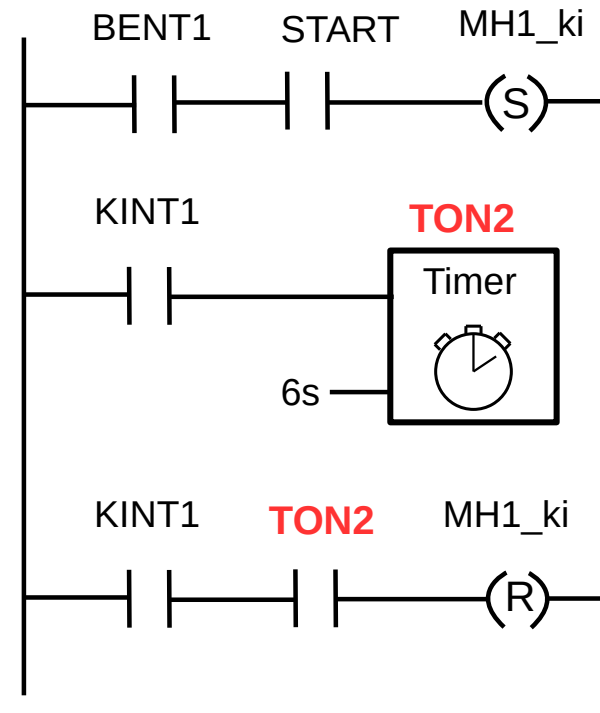
Megoldás impulzus időzítővel

LAD (Festo FST)



Megoldás bekapcsolás késleltetéssel

LAD (Festo FST)

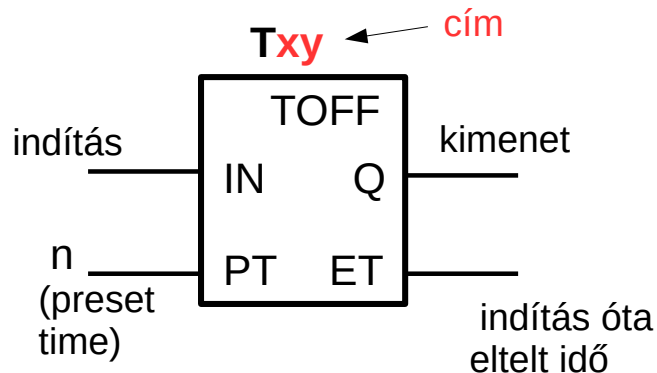


4.10. Időzítő, TOFF

Kikapcsolás késleltetés (TOFF)

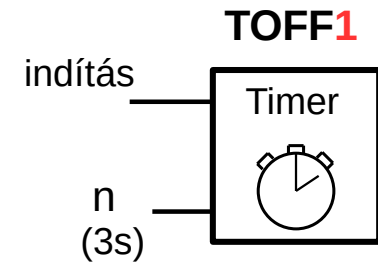
Off-Delay Timer, (TOFF vagy SF)

A kimenet csak az indítás megszűnése után **tk** idővel lesz újra 0-ás



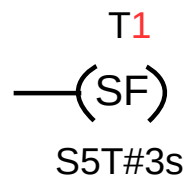
De egyes PLC-k esetén a jelölések eltérőek egy picit

pl. Festo FST (LAD)

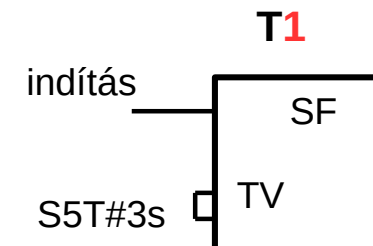


Lehet egyszerűsített jelölése is
→ speciális kimenet
Off-delay timer coil

pl. Siemens STEP7 (LAD)



pl. Siemens STEP7 (FB)

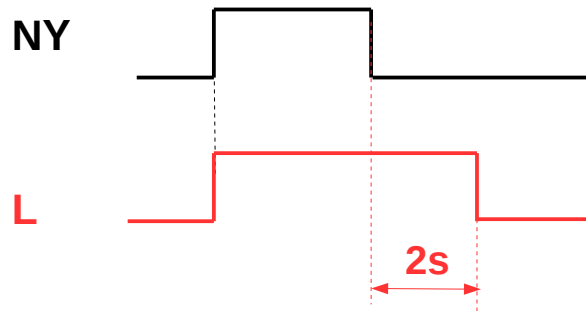


4.11. Időzítő, TOFF

4. mintafeladat (kikapcsolás késleltetés)

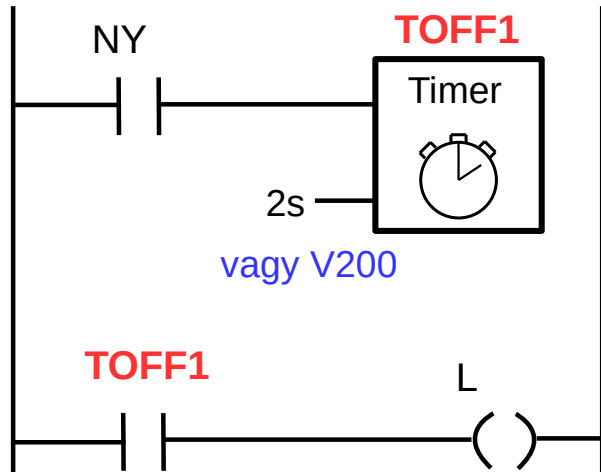
Egy nyomógombbal (NY) vezérlünk egy lámpát (L)

- a nyomógomb lenyomása után azonnal kapcsoljon fel a lámpa
- a nyomógomb elengedésekor nem azonnal, hanem csak 2s elteltével kapcsoljon le a lámpa



Megoldás kikapcsolás késleltetéssel

LAD (Festo FST)



Megoldás bekapcsolás késleltetéssel

LAD (Unitronics VisiLogic)

