

# PLC-k programozása - alapok

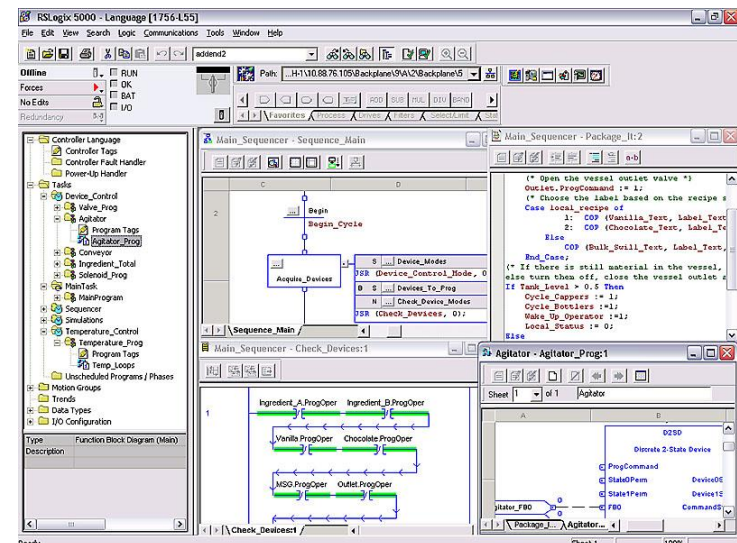
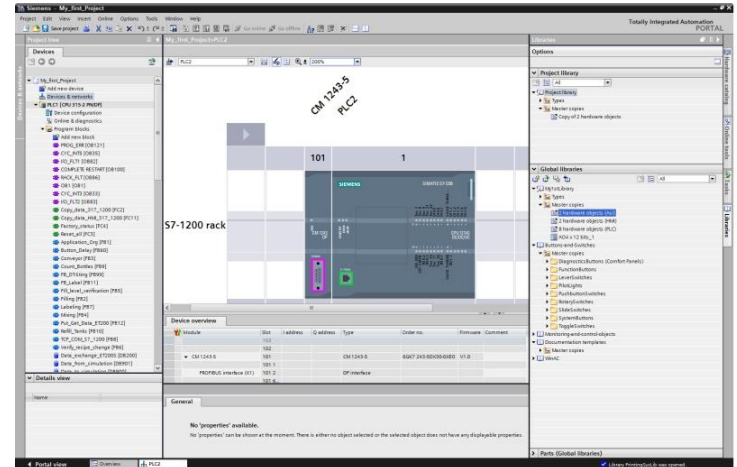
Programozható irányítóberendezések  
és szenzorrendszerek

KOVÁCS Gábor

gkovacs@iit.bme.hu

# A fejlesztői környezet feladatai

- PLC konfiguráció
- Alkalmazásfejlesztés
- Kapcsolat a PLC-vel
- Alkalmazás és adatok átvitele
- PLC monitorozása
- Szimuláció
- Dokumentáció támogatása



# PLC konfiguráció

- Az alkalmazás lehetőségei függnnek
  - A PLC típusától
  - Rendelkezésre álló I/Ok száma és típusától
  - Kommunikációs lehetőségektől
  - PLC firmware/OS típusától
- A fejlesztői környezet lehetővé teszi
  - A PLC konfiguráció megadását (modulok száma és típusa)
  - Firmware/OS frissítését

# Alkalmazásfejlesztés

- Az irányítási feladatot megvalósító alkalmazás fejlesztésének támogatása
- Több programozási nyelv támogatása
- Felhasználói program fordítása (*compile*)
- Felhasználóbarát megoldások
  - Könnyen használható felület
  - Autocomplete

# Kommunikációs kapcsolat a PLC-vel

- Nem szabványos kommunikációs interfész
- RS232/RS485 alapú megoldások
- USB
- Ethernet

# Alkalmazás- és adatátvitel

- Download - letöltés
  - PC → PLC
  - Alkalmazás és adatok
- Upload - feltöltés
  - PLC → PC
  - Alkalmazás és adatok
- Le- és feltöltés akár futás közben is!

# PLC monitorozás

- Memória online monitorozása
  - Rendszermemória
  - Bemeneti kép
  - Kimeneti kép
  - Felhasználói memória
- Programfutás monitorozása
- Kimenetek állítása (*force*)

# Szimuláció

- A PLC kód PC-s környezetű emulátoron fut
- Virtuális kimenetek megfigyelhetők, bemenetek állíthatók
- A fizikai ki- és bemenetek nem változnak
- Biztonságos módszer az alkalmazás tesztelésére



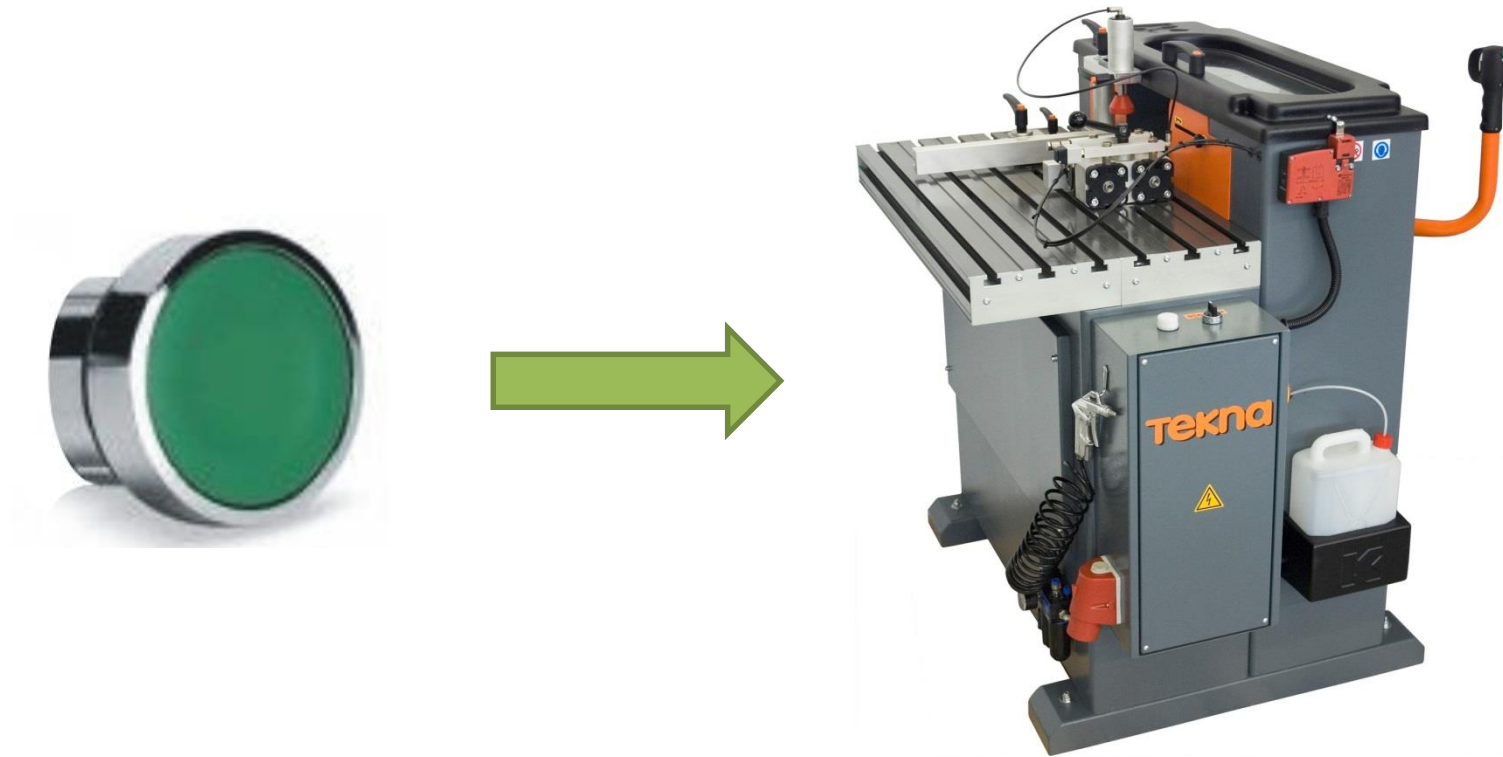
# Dokumentáció

- Ipari környezetben különösen fontos
- A PLC-k gyakran a kommenteket is tárolják a forráskód mellett
- Támogatás: modulok az automatikus dokumentációkészítéshez
- A dokumentáció részei
  - Konfigurációs beállítások
  - Memóriakonfiguráció (változók azonosítója, típusa, címe)
  - Programkód
  - Kommunikációs beállítások

# Fejlesztői környezetek

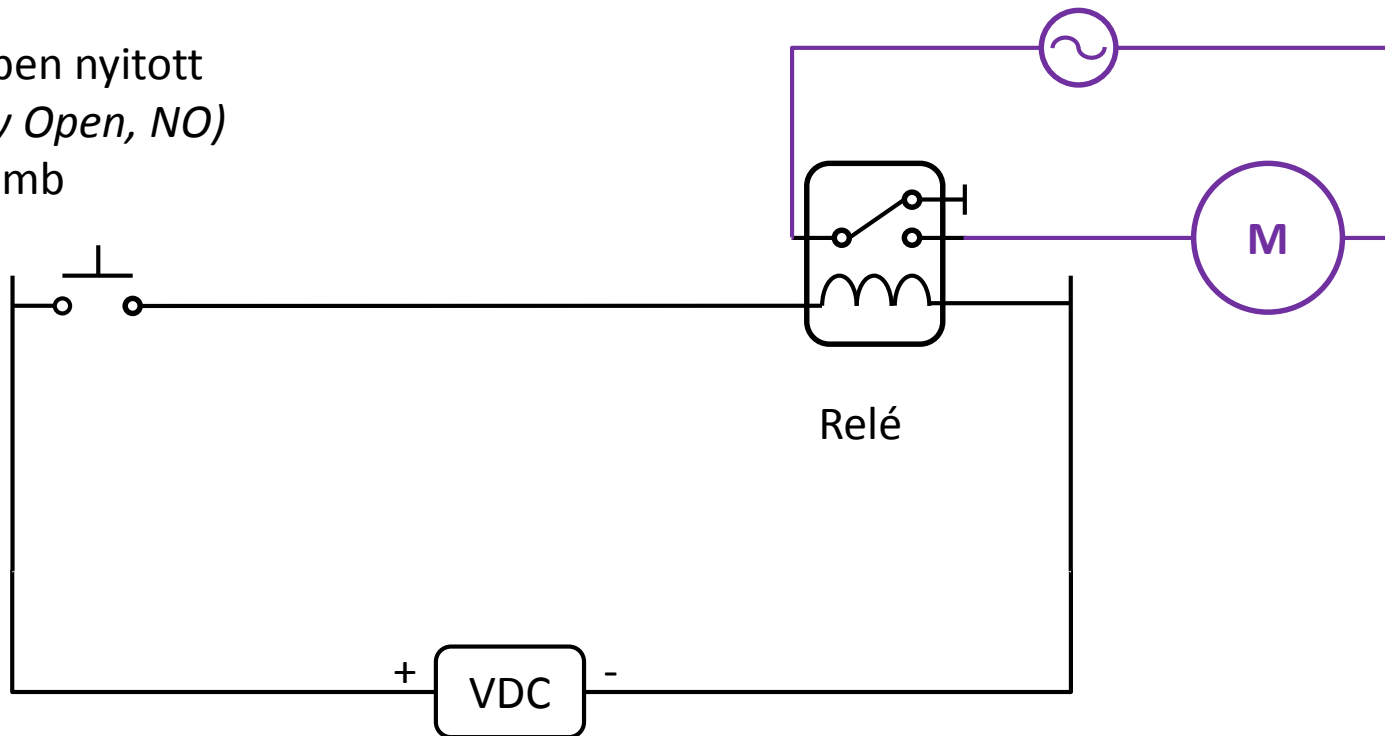
- Siemens: TIA portal (Step 7)
- Rockwell (Allen-Bradley): RSLogix, Studio 5000
- Schneider: Unity, TwidoSuite
- Mitsubishi: iQ Works
  
- CoDeSys
  - Gyártófüggetlen fejlesztői környezet
  - Több kisebb gyártó támogatja: Moeller, Beckhoff stb.

# A huzalozott logikától a PLC-kódig

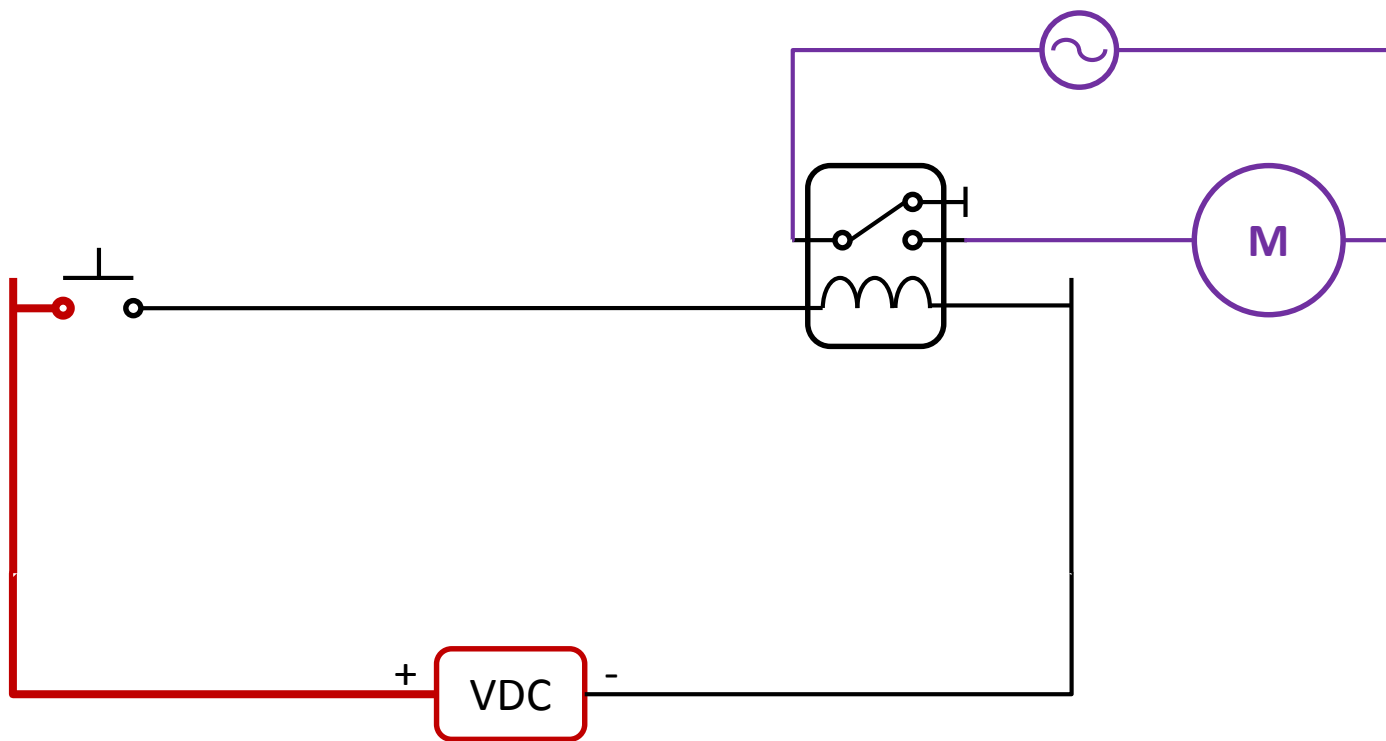


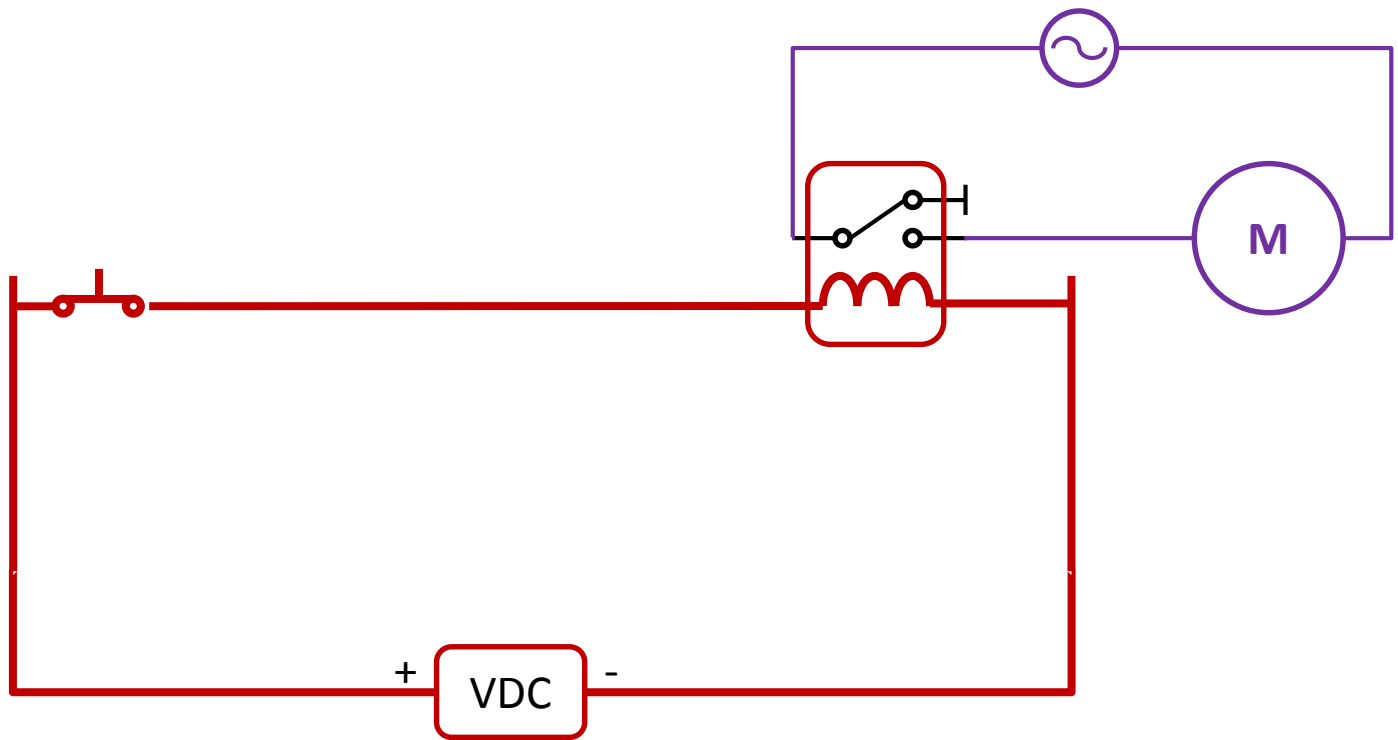
Szerszámgép: 380V AC

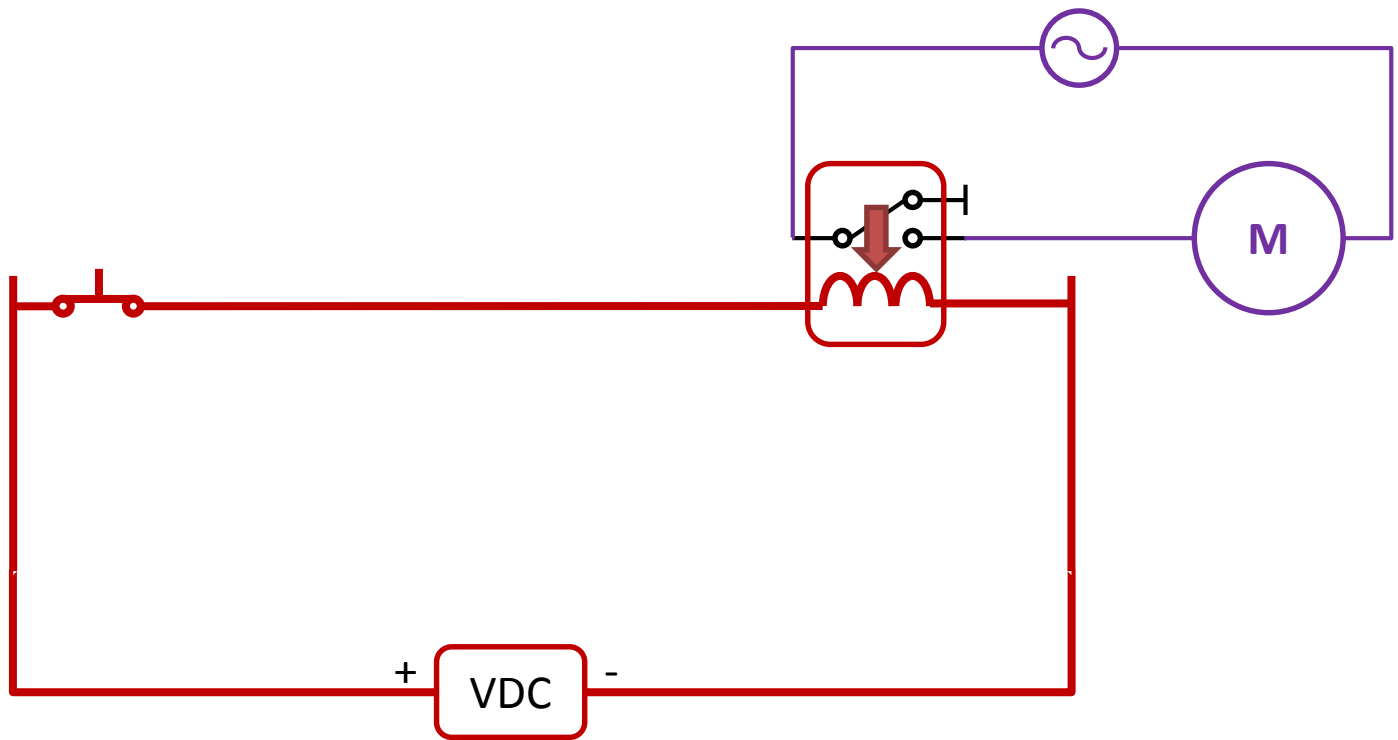
Alapesetben nyitott  
(Normally Open, NO)  
nyomógomb

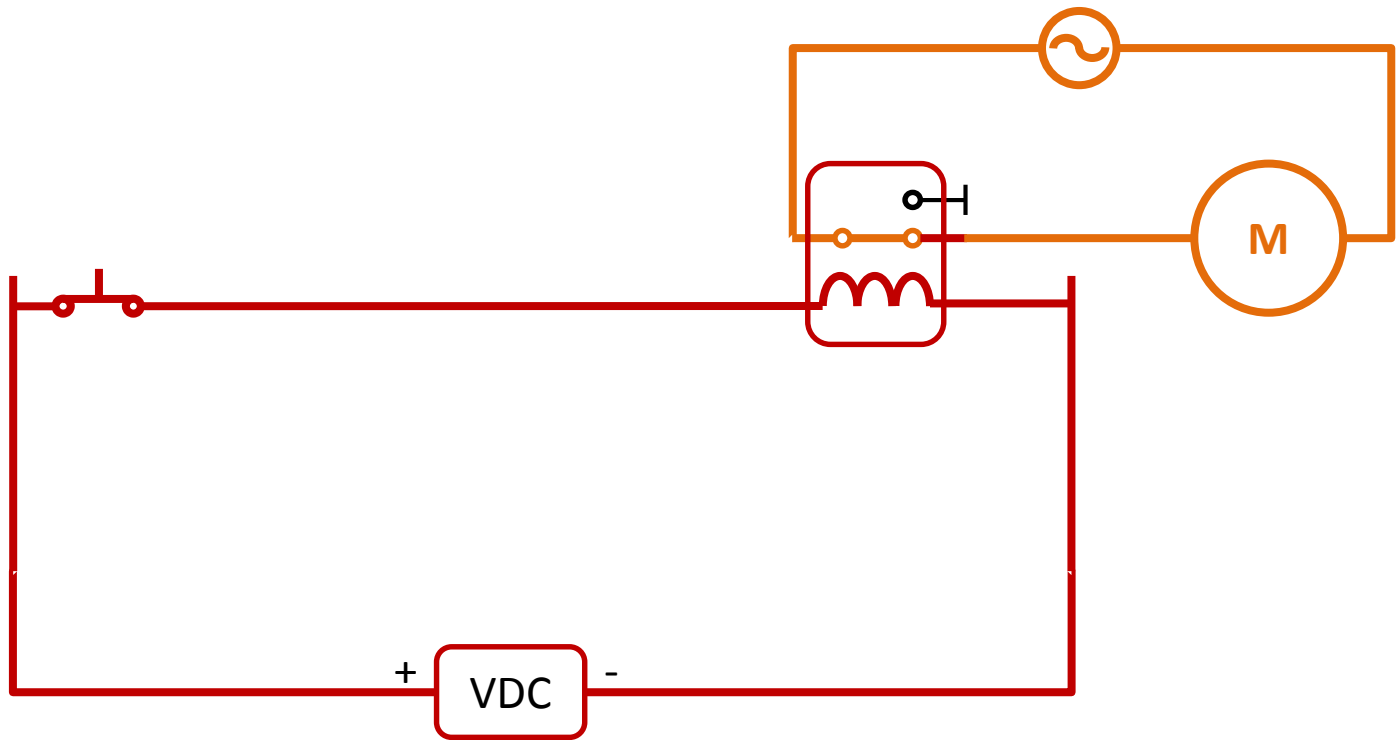


Logika: 24V DC





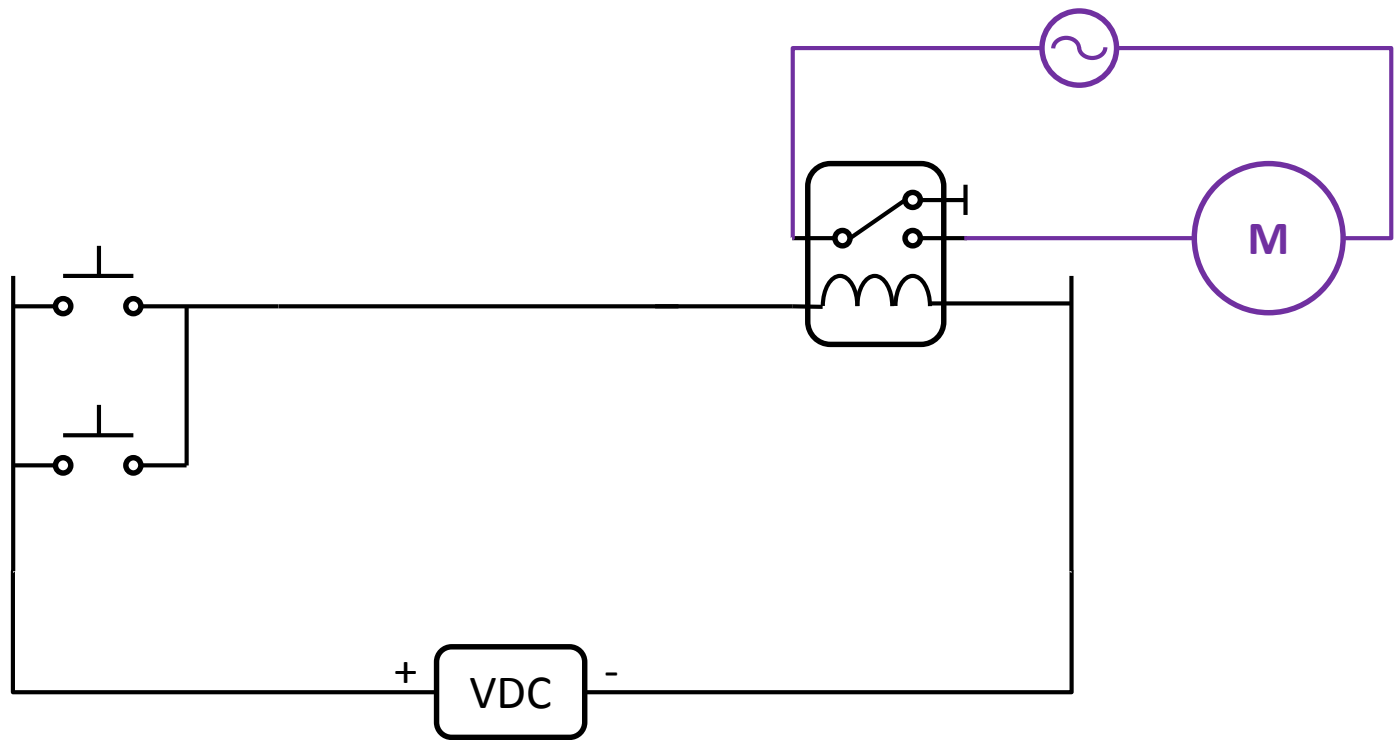


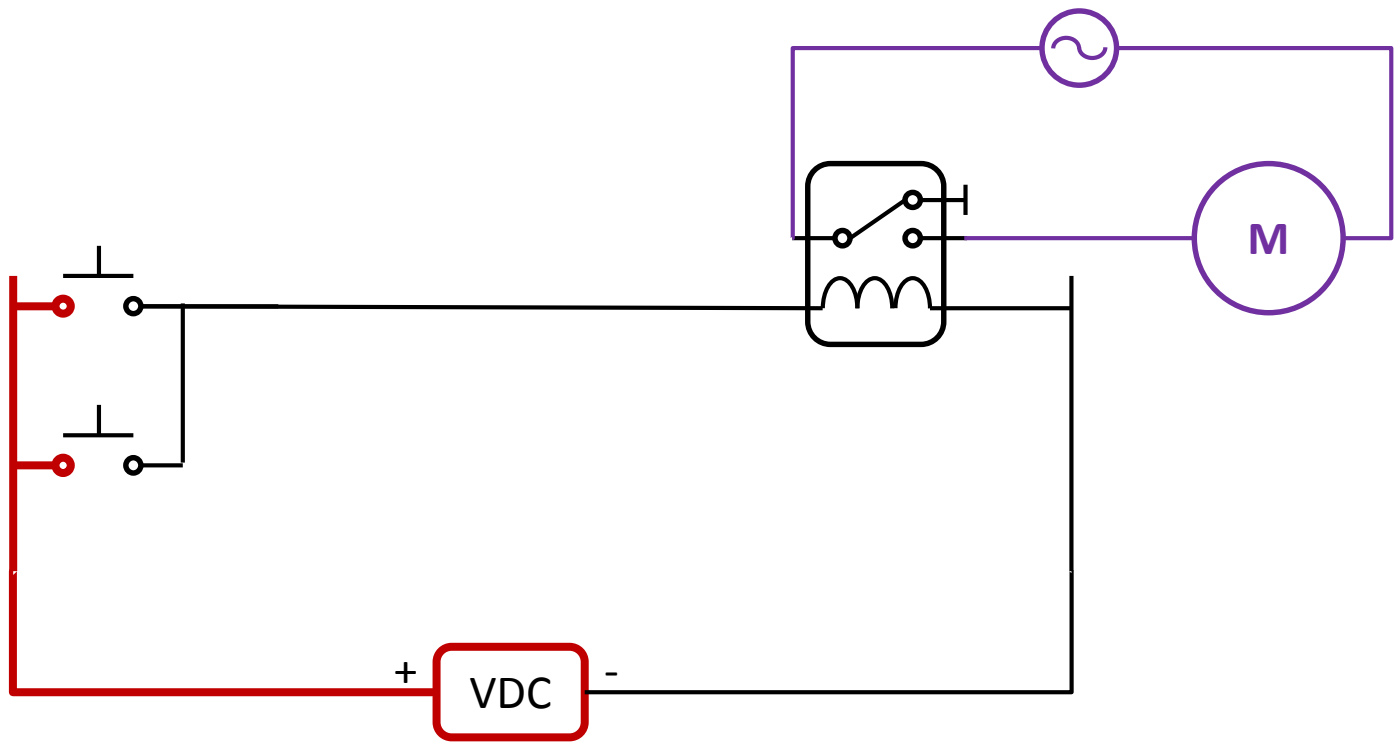


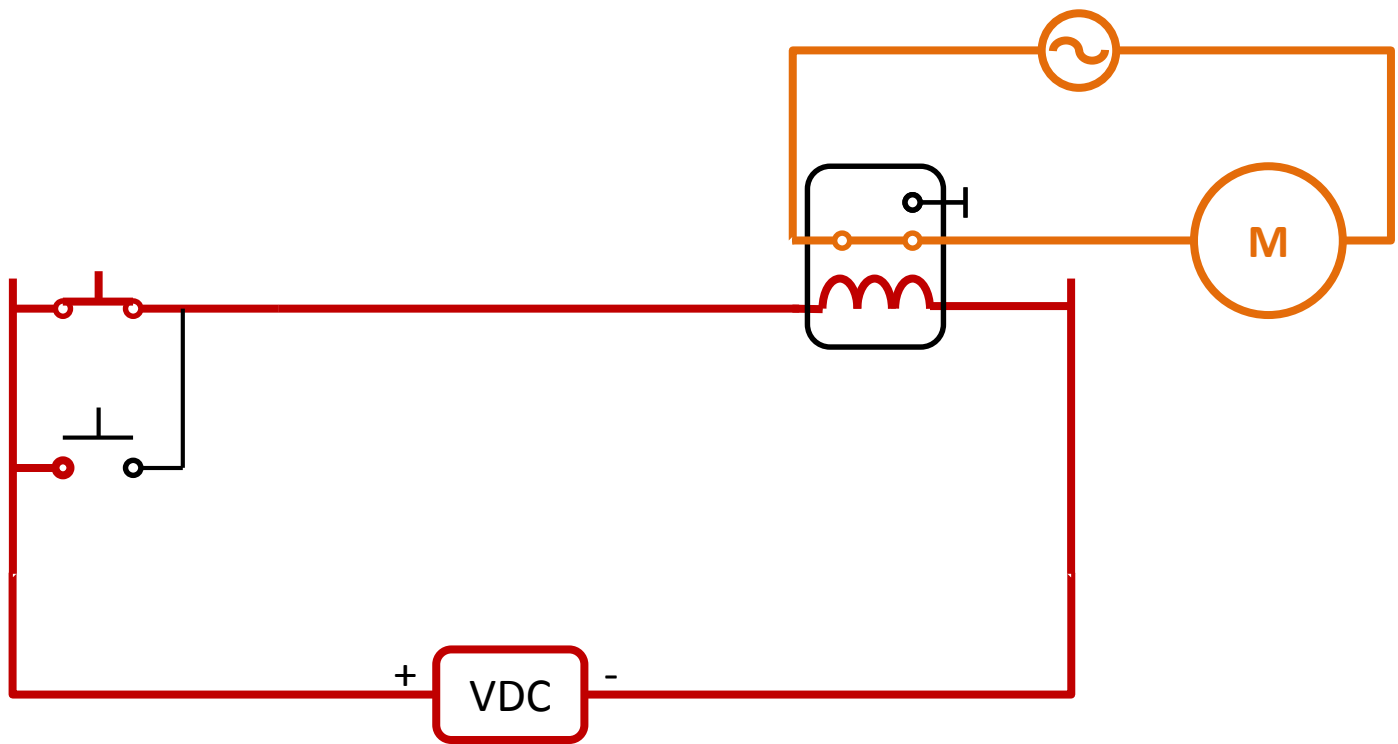


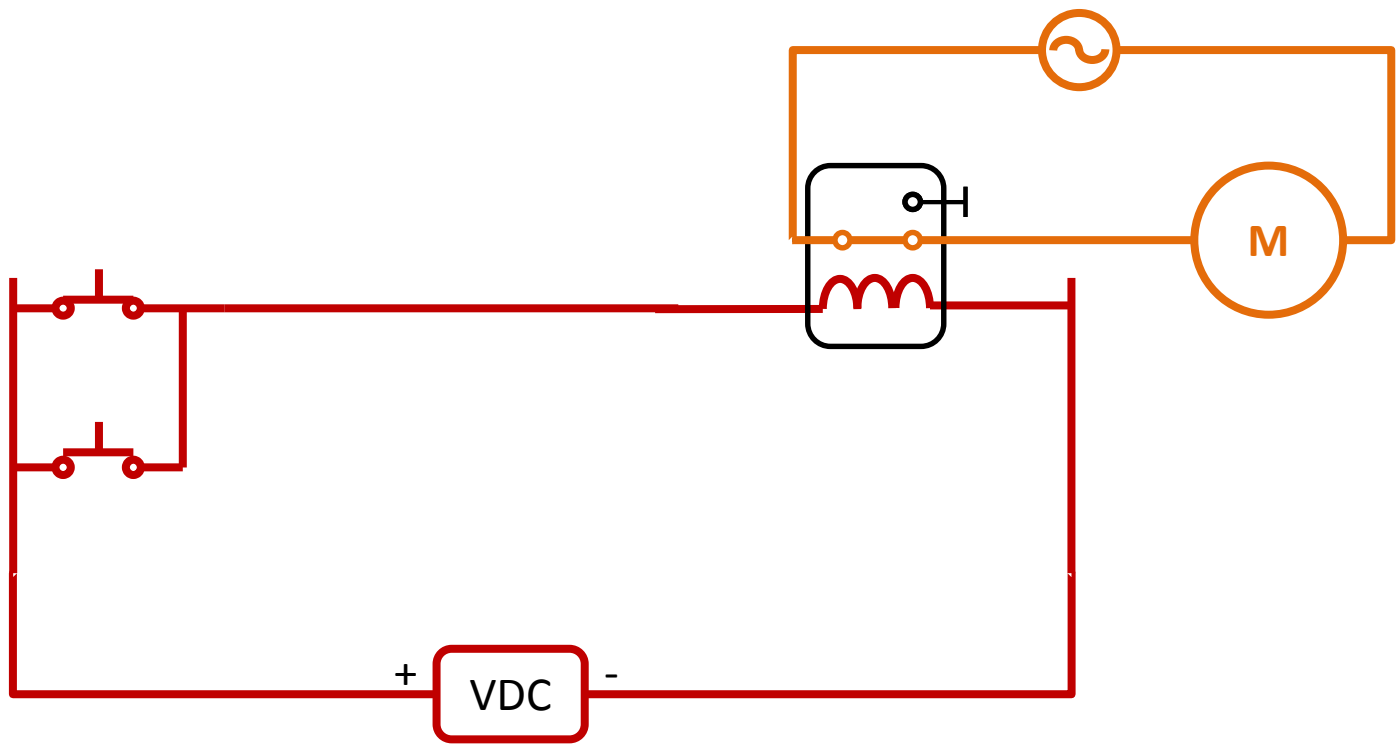
A szerszámgépet a két nyomógomb  
bármelyike működtetheti (VAGY)

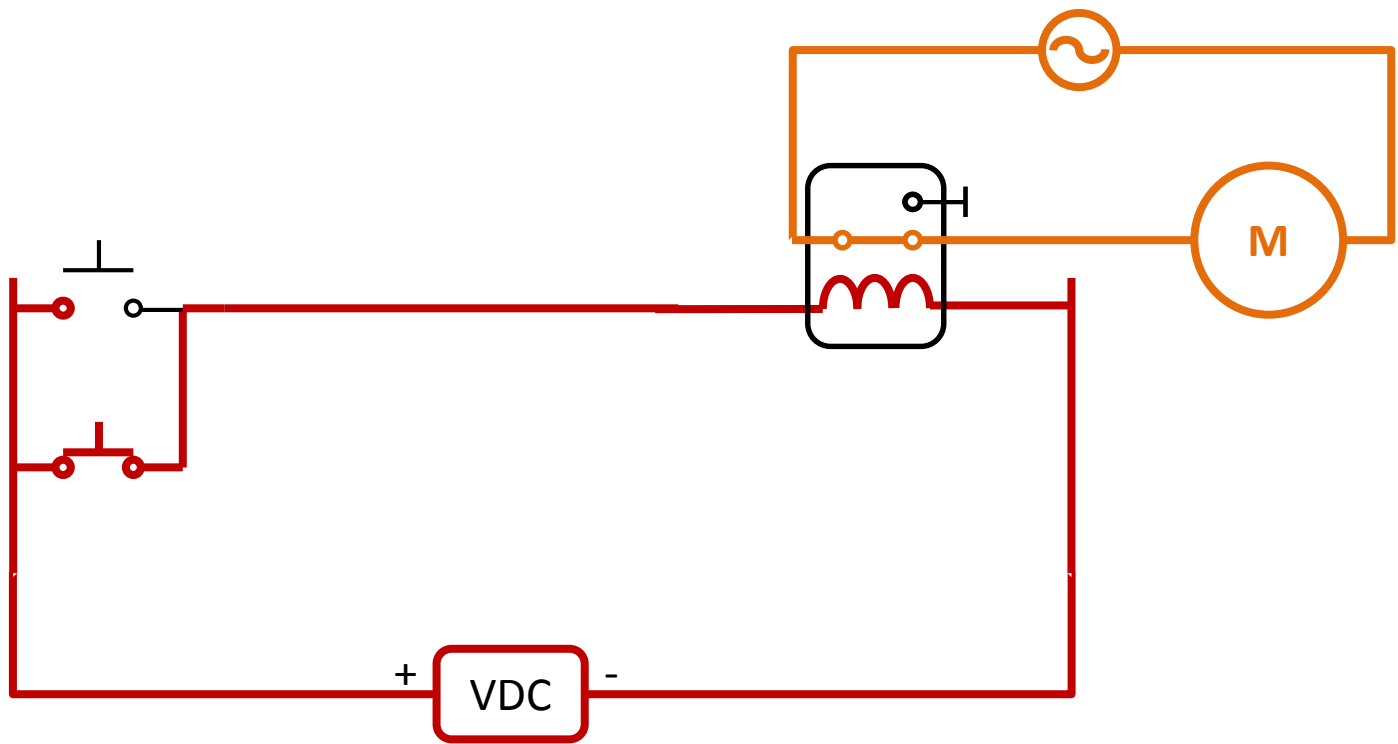




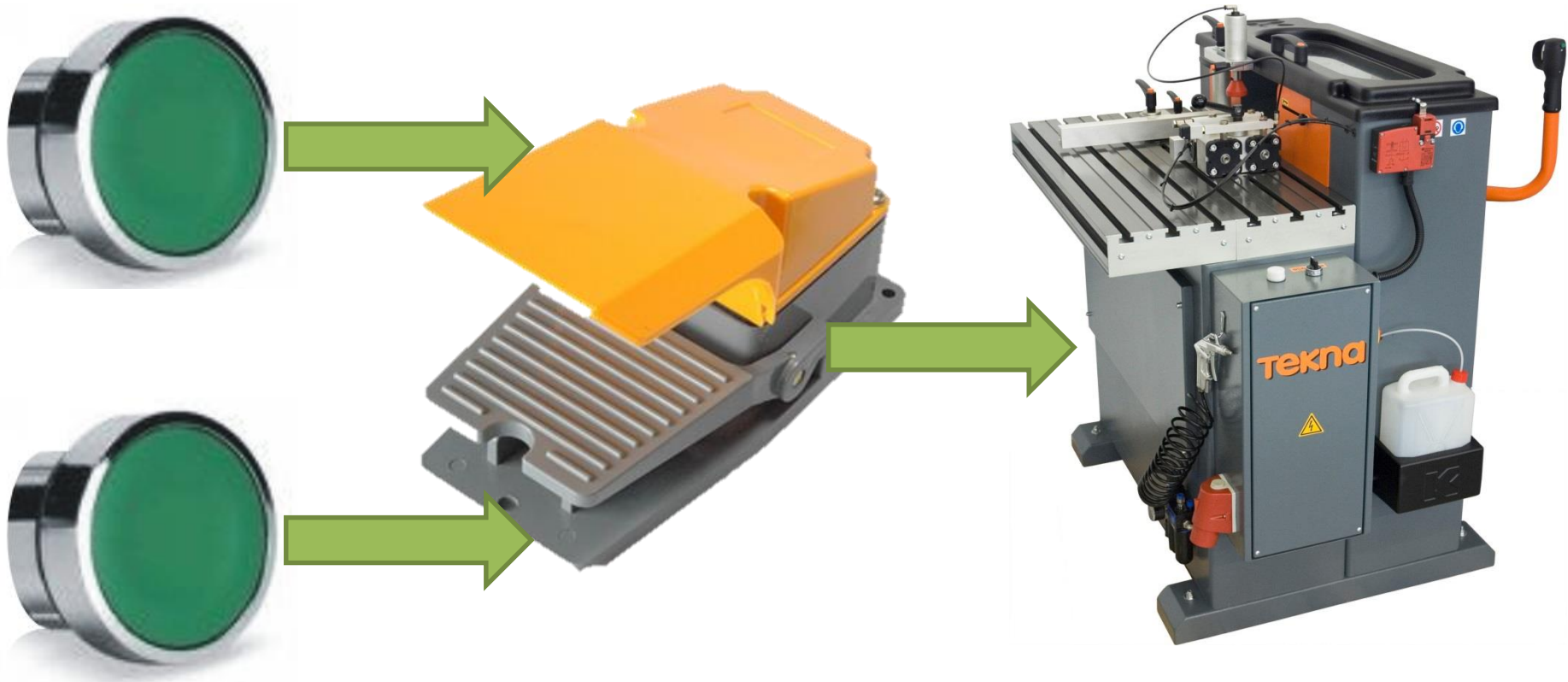


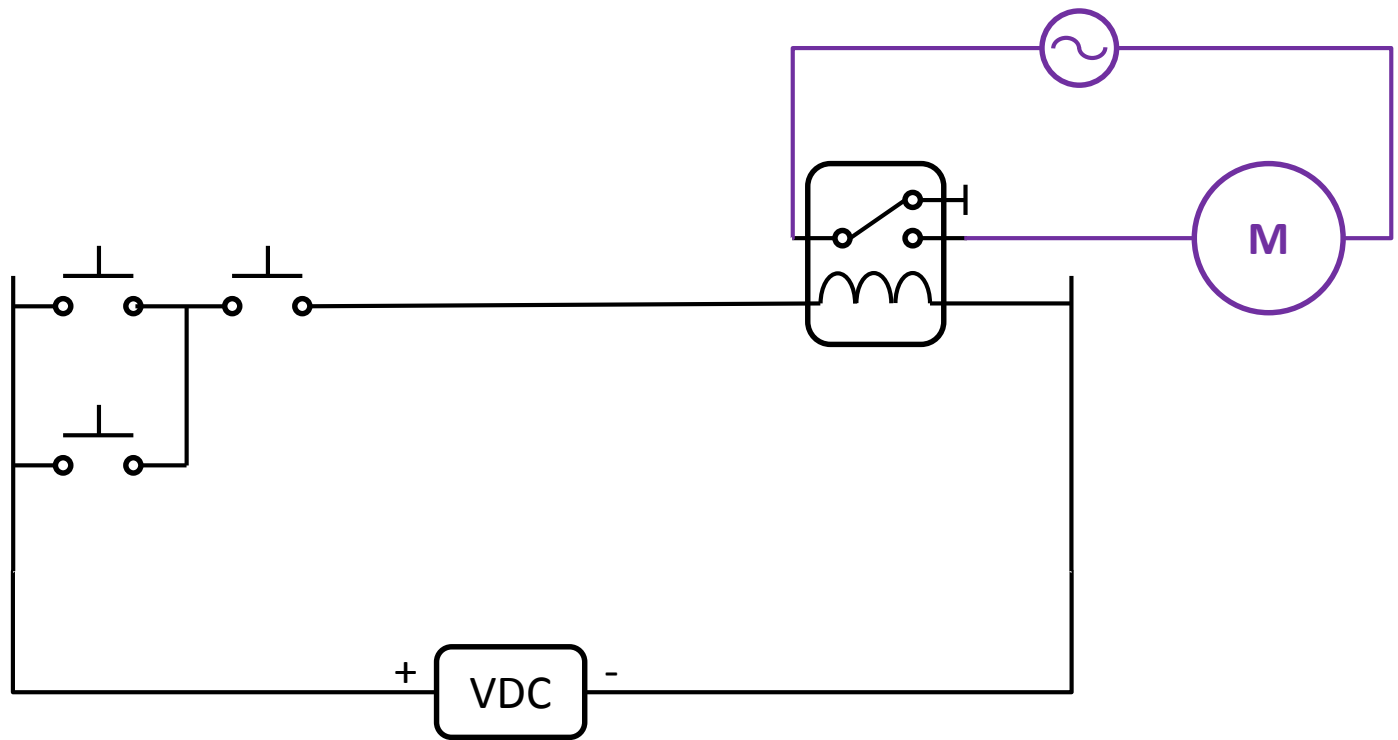




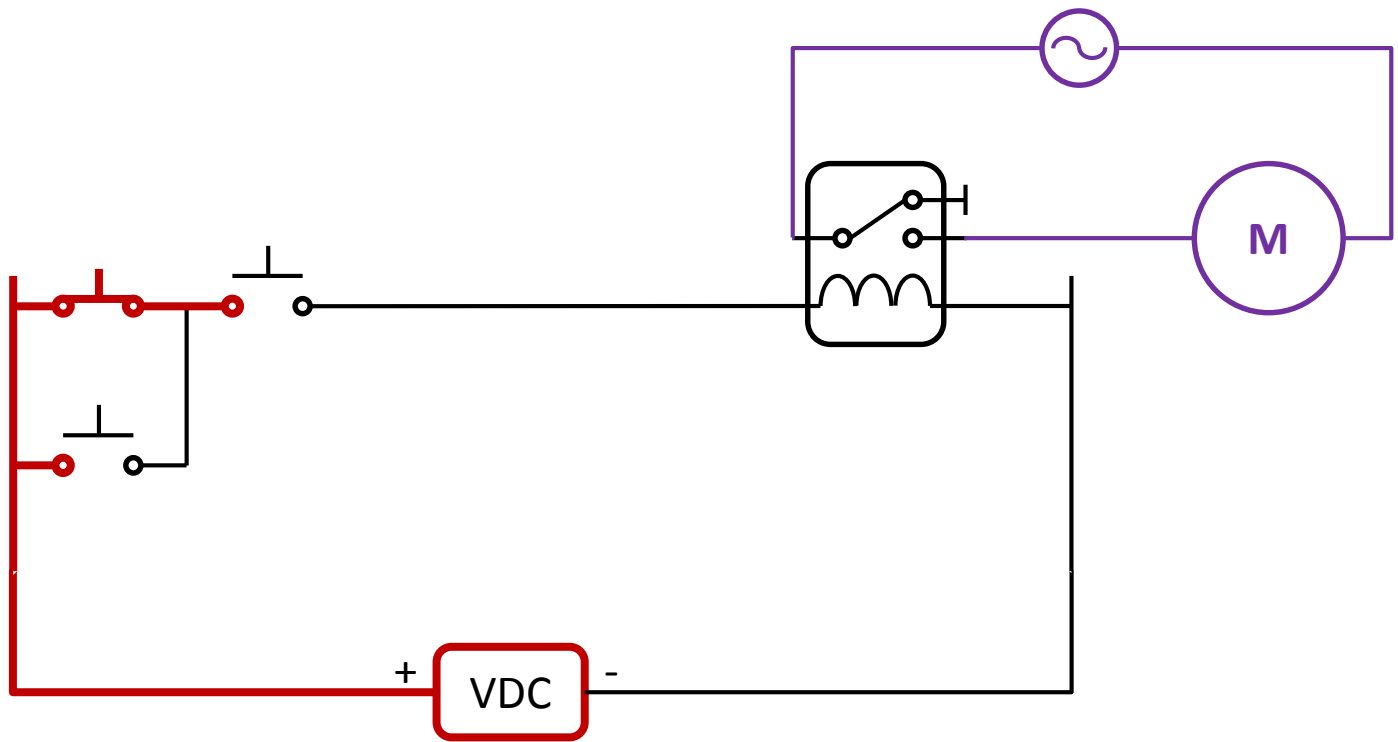


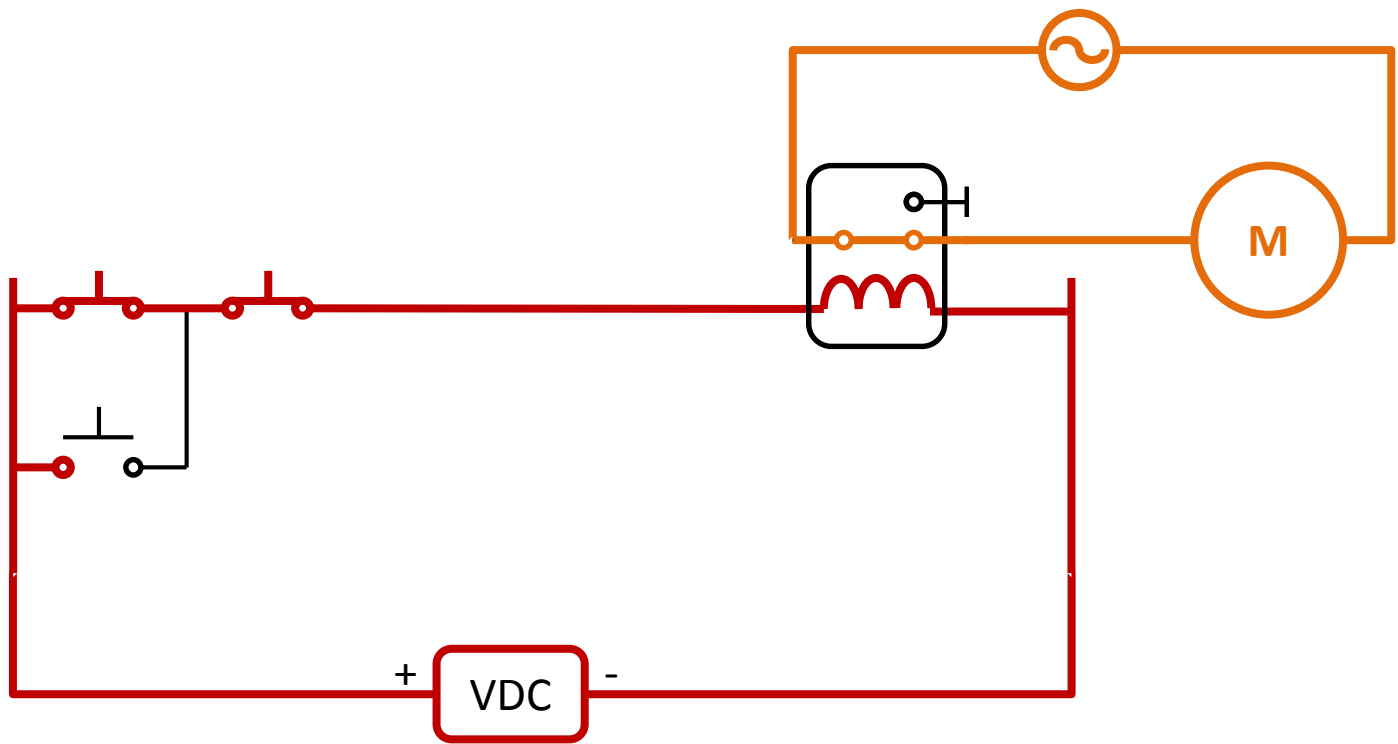
# Biztonsági kapcsoló (ÉS)

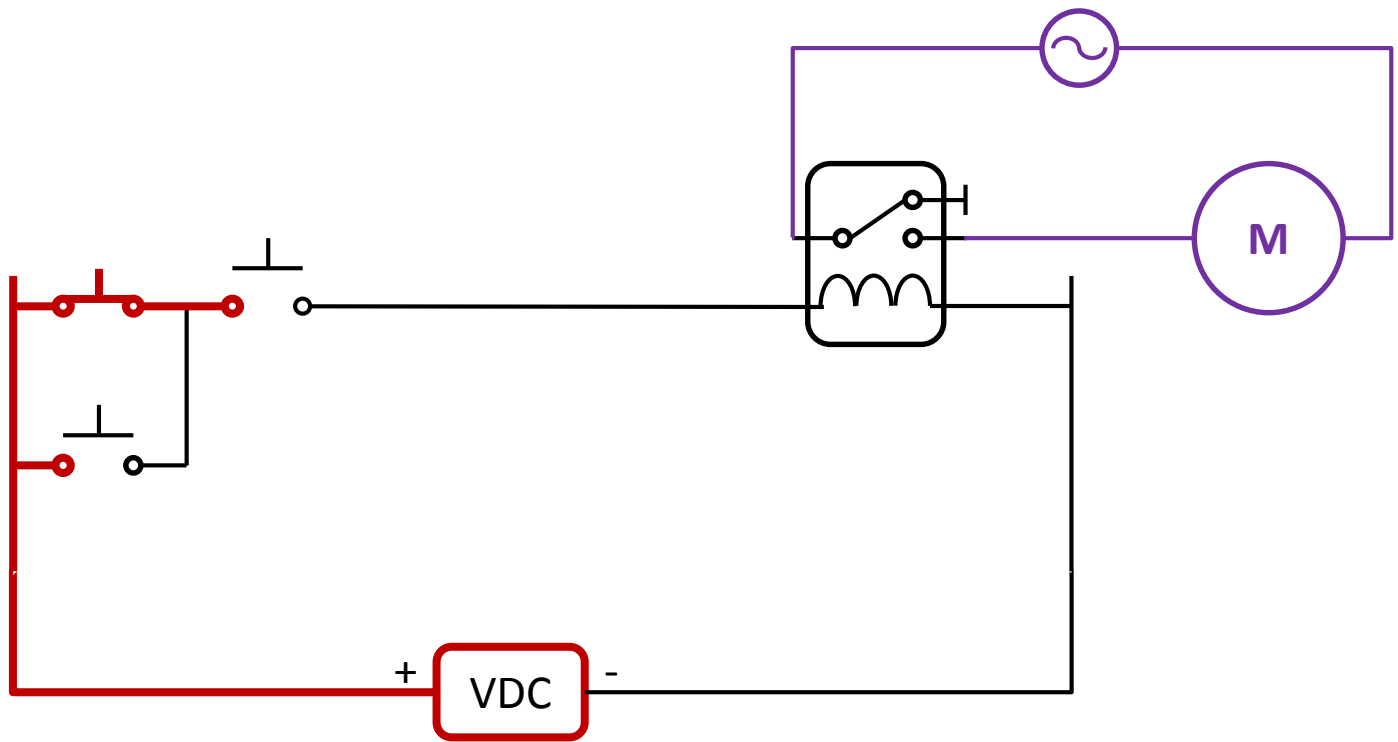




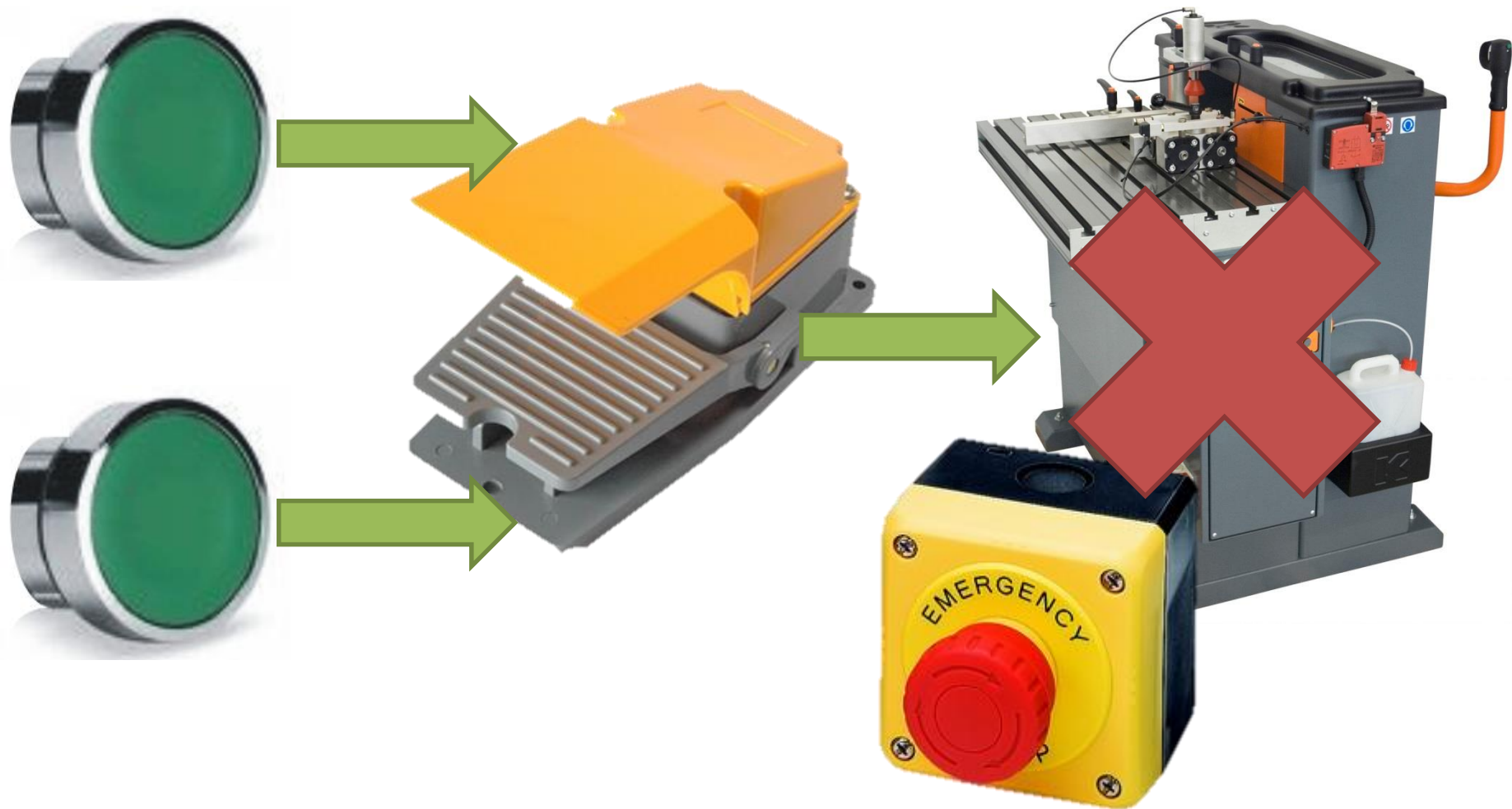




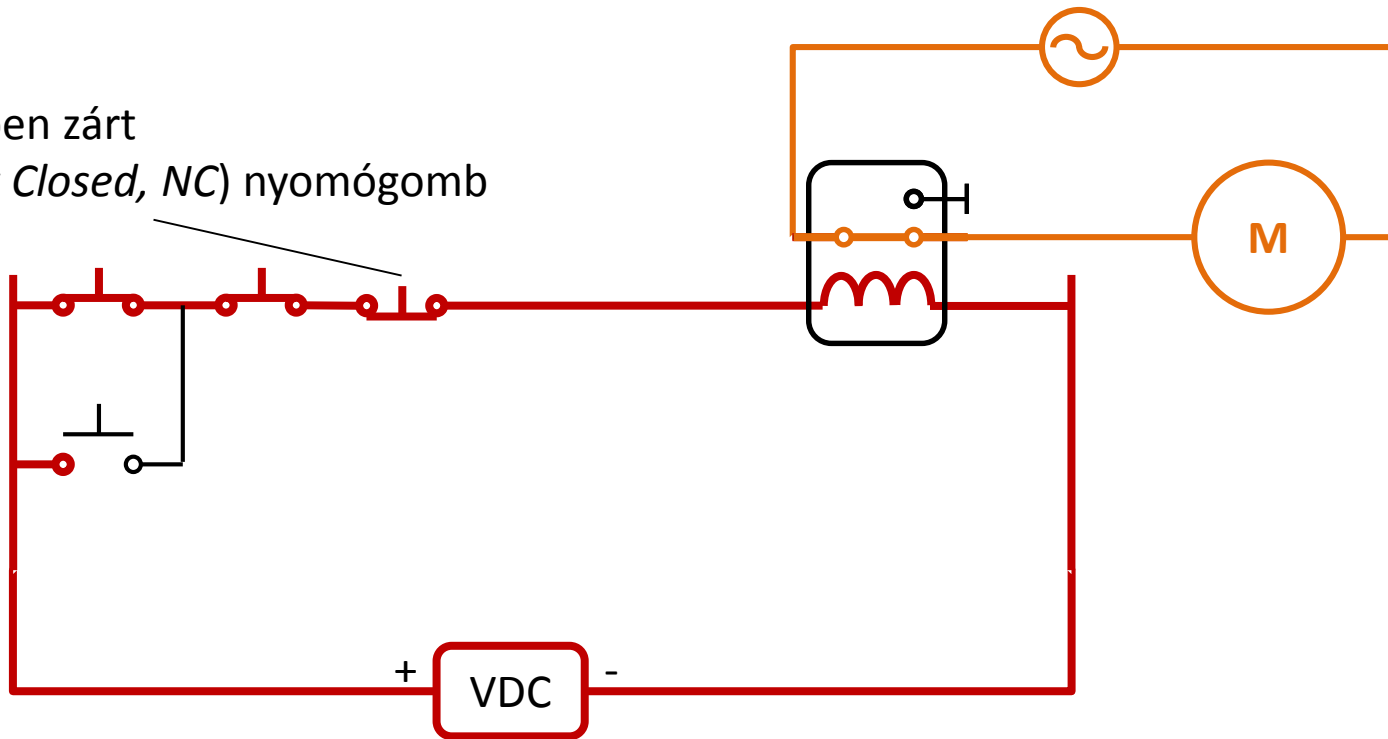


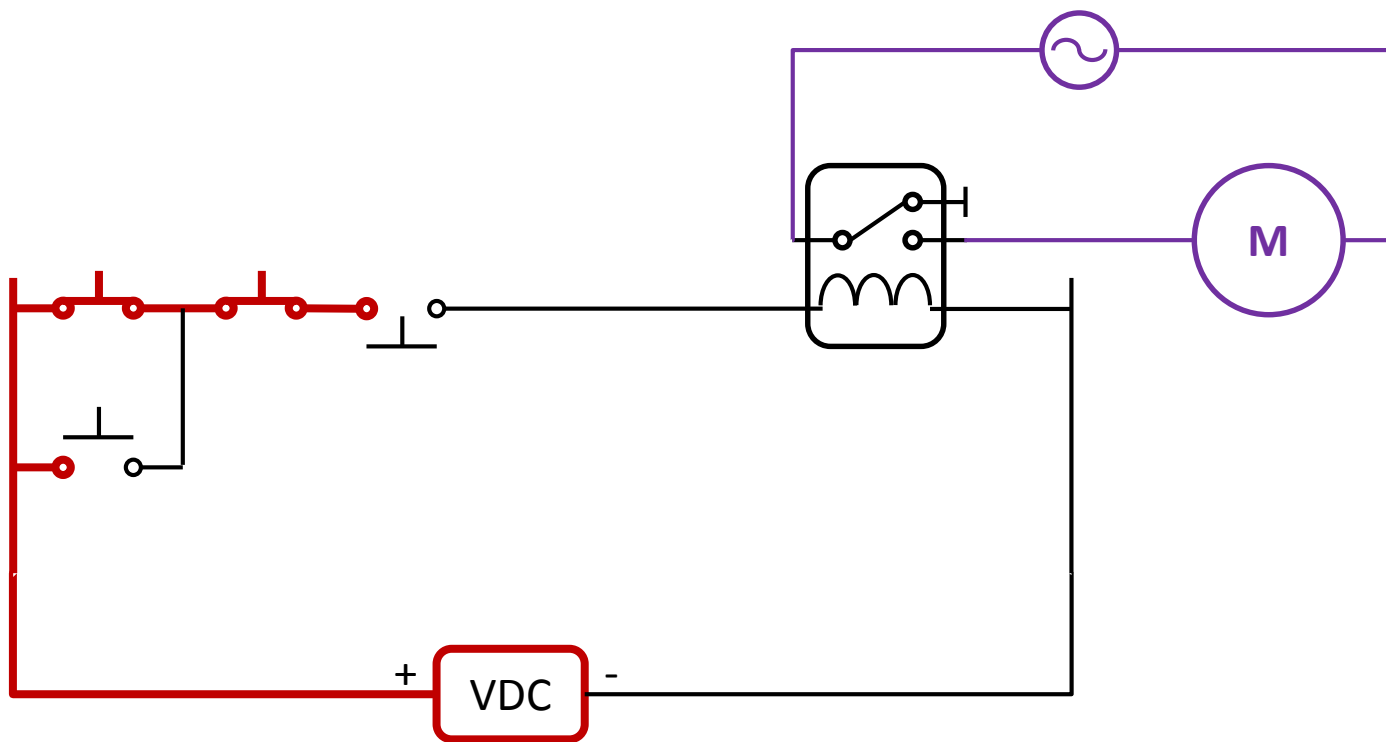


# Vészleállító gomb (NAND)

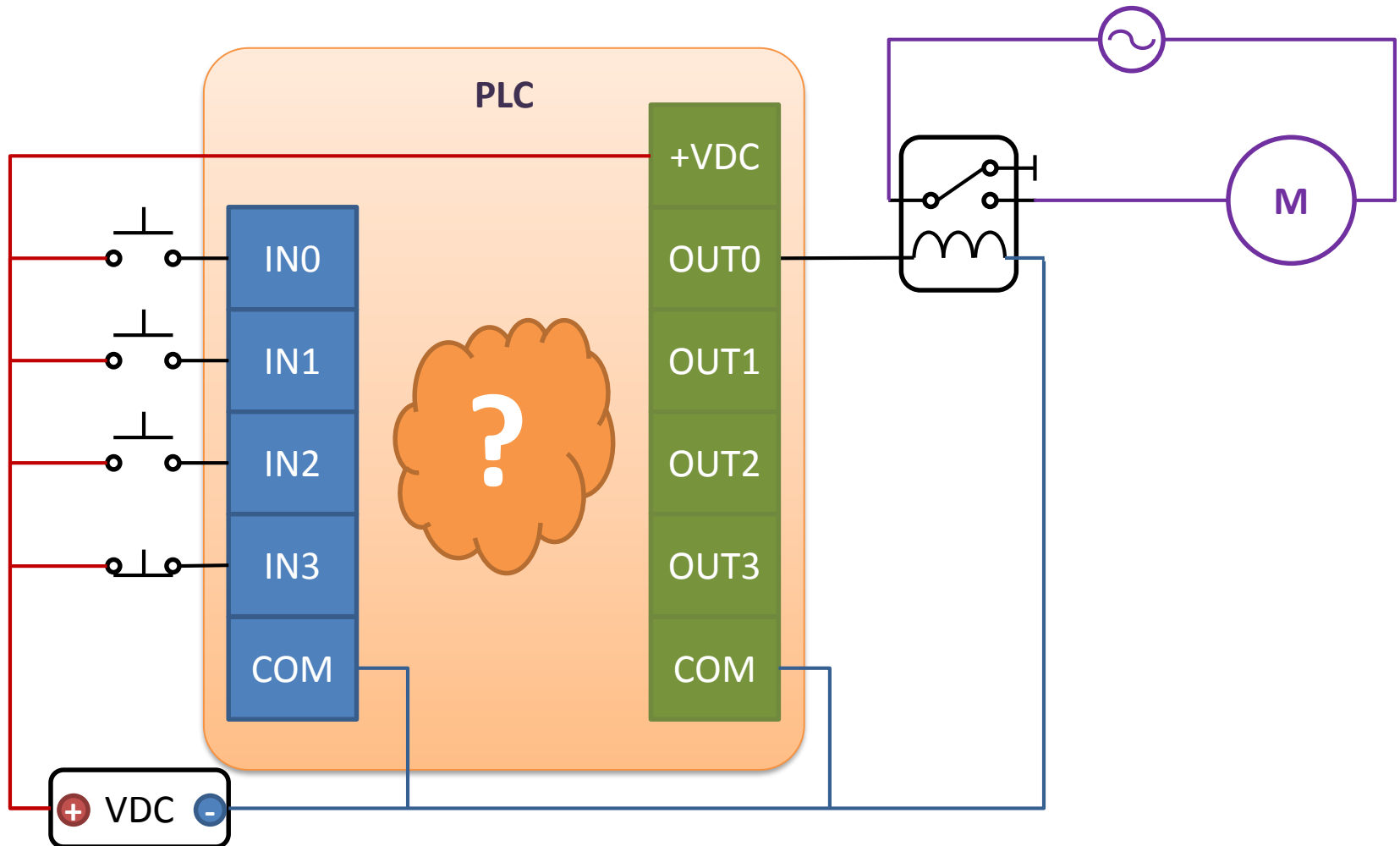


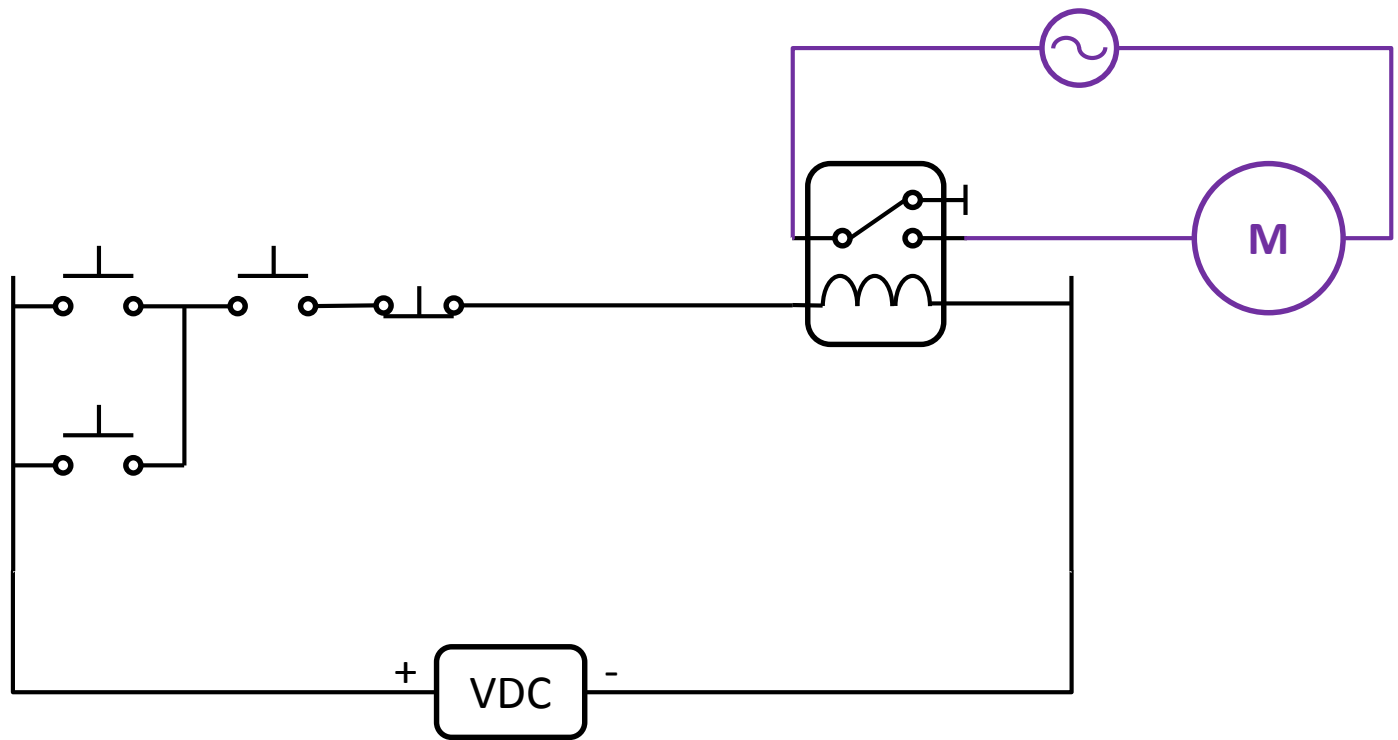
Alap esetben zárt  
(*Normally Closed, NC*) nyomógomb



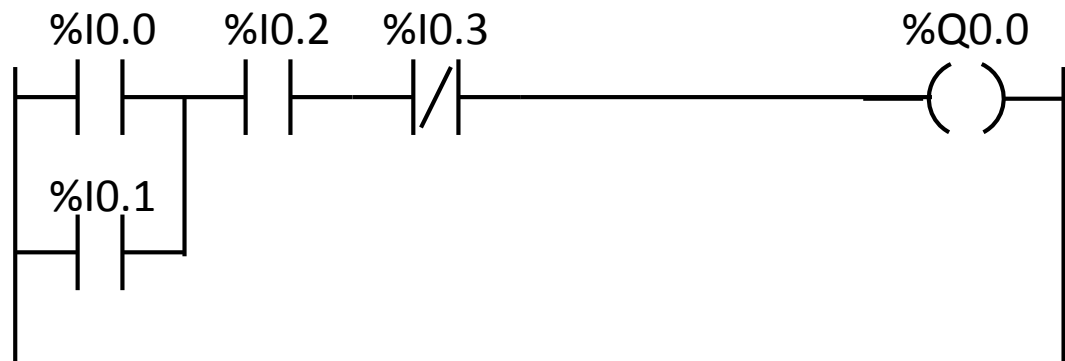
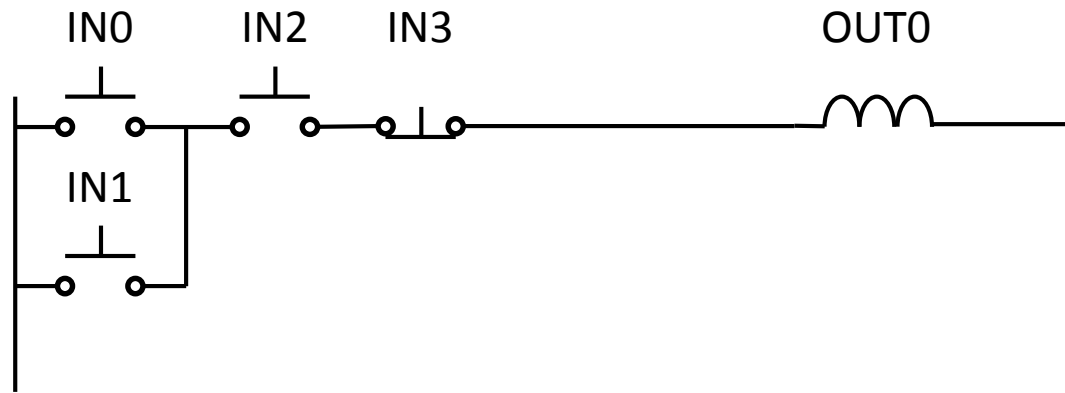


# PLC IO-k bekötése









# Létradiagram

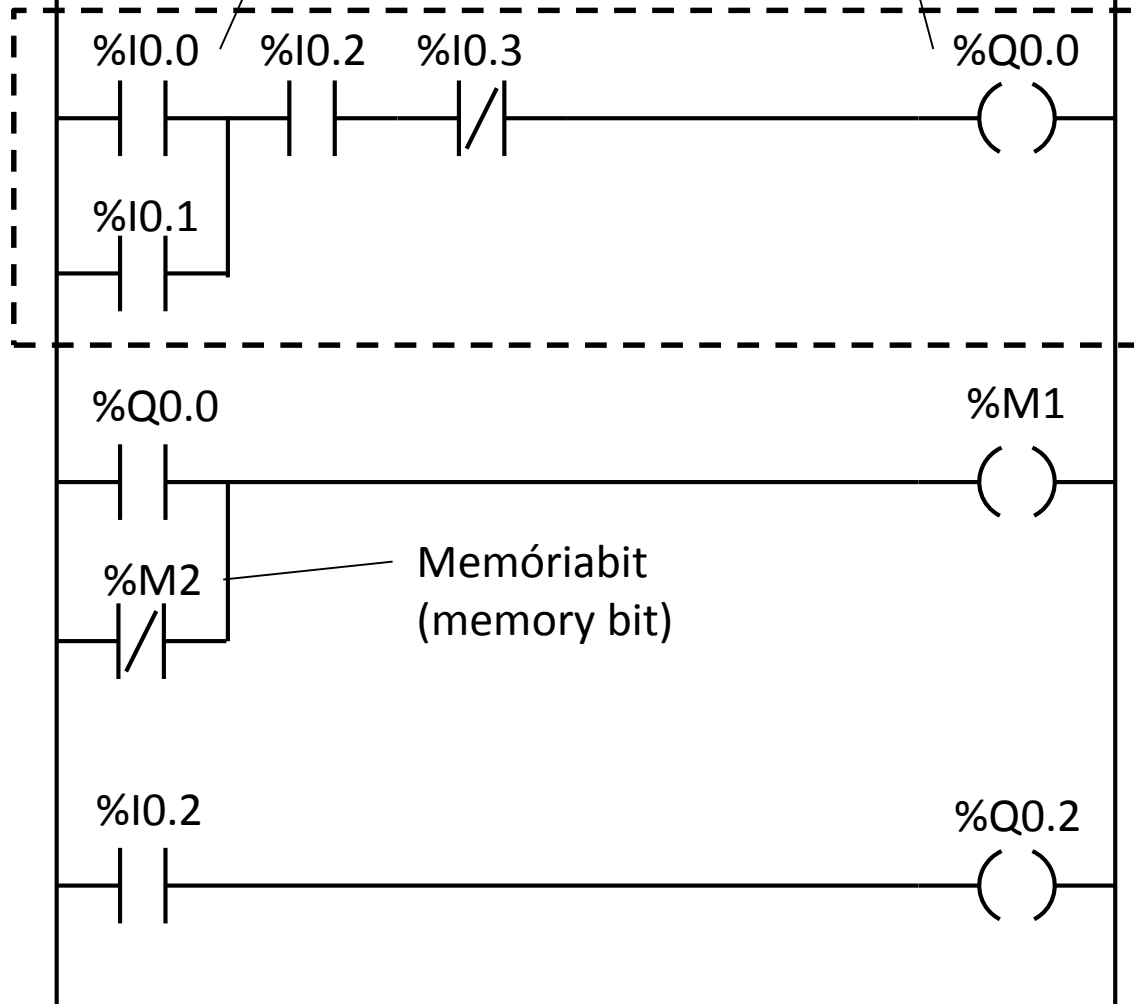
Pozitív sín  
(positive rail)

Föld sín  
(neutral rail)

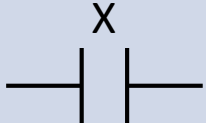
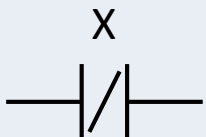
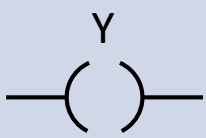
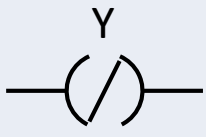
Bemeneti bit  
(input bit)

Kimeneti bit  
(output bit)

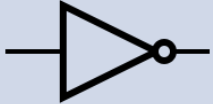
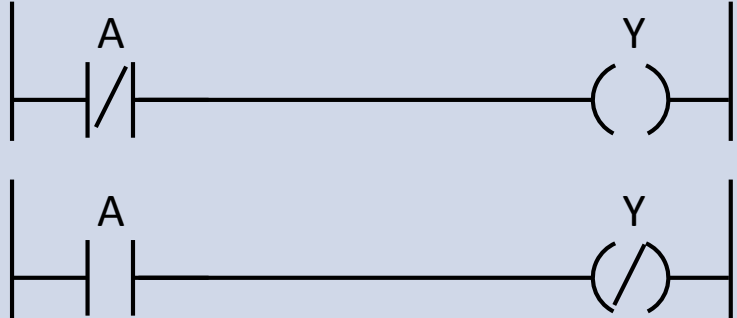

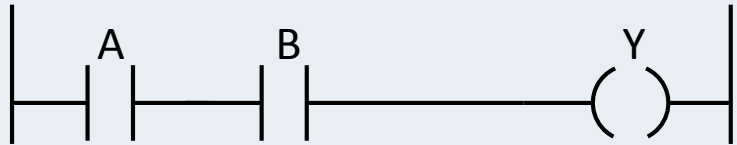

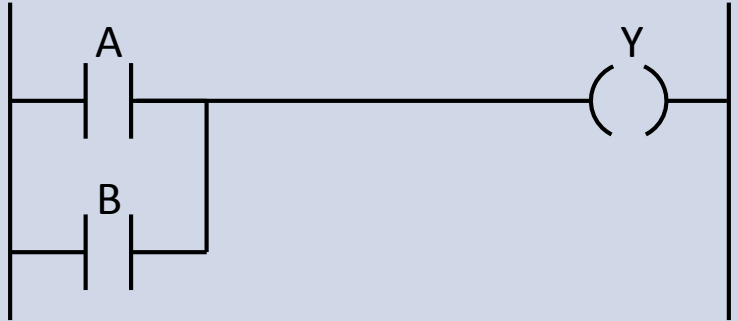
Létraszor  
(rung)




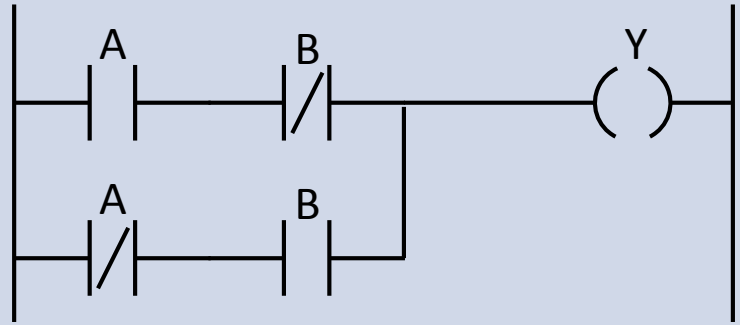

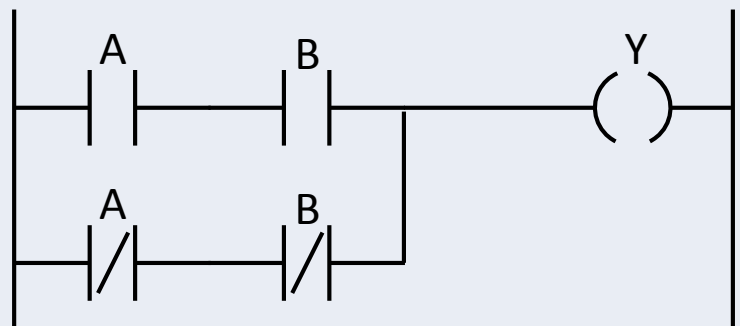
# Kontaktusok és tekercsek

Szimbólum	Megnevezés	Működés	Analógia
	Alaphelyzetben nyitott kontaktus [kontaktus] ( <i>NO contact, contact</i> )	„vezet” ha $X=1$	Alaphelyzetben nyitott nyomógomb
	Alaphelyzetben zárt kontaktus [negált kontaktus] ( <i>NC contact</i> )	„vezet” ha $X=0$	Alaphelyzetben zárt nyomógomb
	Alaphelyzetben nyitott tekercs [tekercs] ( <i>NO coil, coil</i> )	Y-t 1-be állítja ha „táplált”	Alaphelyzetben nyitott relé
	Alaphelyzetben zárt tekercs [negált tekercs] ( <i>NC coil</i> )	Y-t 0-ba állítja ha „táplált”	Alaphelyzetben zárt relé


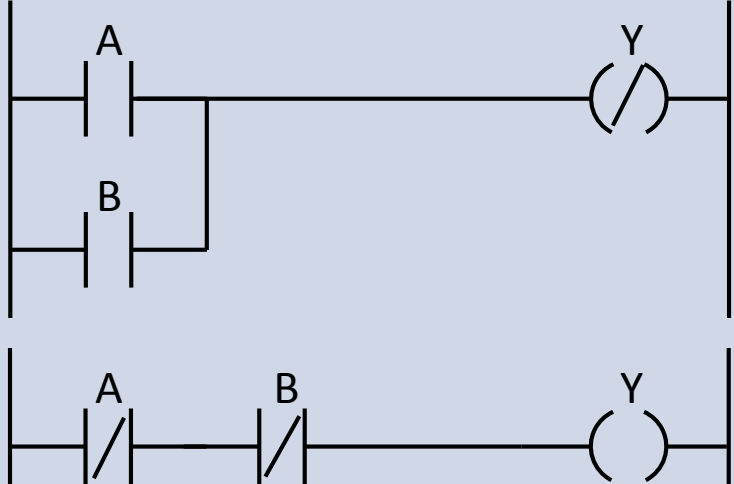

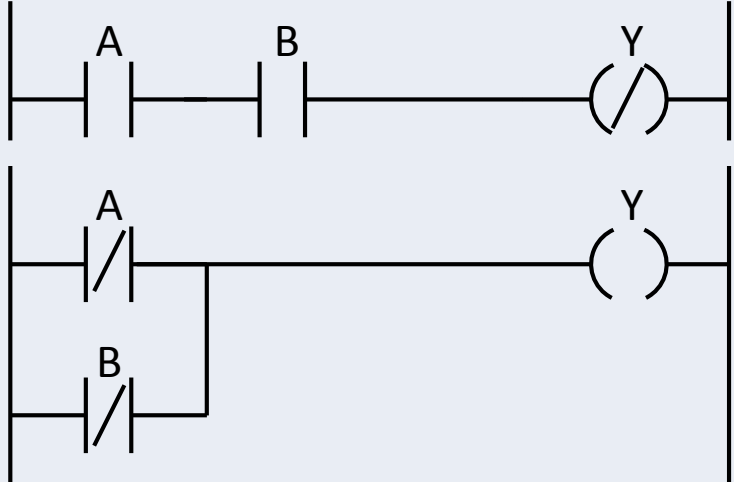
# Logikai műveletek

Művelet	Szimbólum	Algebrai jelölés	Létradiagramos megvalósítás
Negálás		$Y = \bar{A}$	
ÉS		$Y = A \cdot B$ $Y = A \& B$	
VAGY		$Y = A + B$	

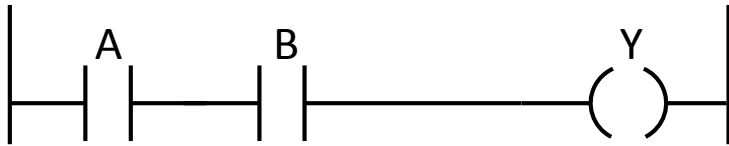
# Logikai műveletek

Művelet	Szimbólum	Algebrai jelölés	Létradiagramos implementációs
Antivalencia (XOR)		$Y = A \oplus B$	
Ekvivalencia (NXOR, EOR)		$Y = \overline{A \oplus B}$ $Y = A \odot B$	


# Logikai műveletek

Művelet	Szimbólum	Algebrai jelölés	Létradiagramos megvalósítás
NOR		$Y = \overline{A + B}$ $Y = \bar{A} \cdot \bar{B}$	
NAND		$Y = \overline{A \cdot B}$ $Y = \bar{A} + \bar{B}$	

# Létrásor = Logikai függvény



**IF (A=1) AND (B=1)**  
**THEN Y=1** 

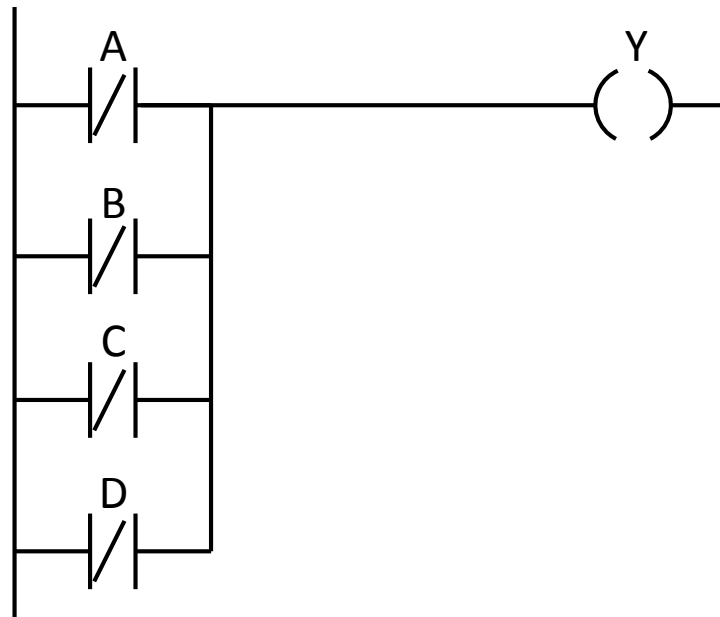
**IF (A=1) AND (B=1)**  
**THEN Y=1**  
**ELSE Y=0** 

# Hogyan egyszerűsítsünk?

$$Y = \overline{(A \cdot B)} + \overline{(C \cdot D)}$$



$$Y = \overline{A} + \overline{B} + \overline{C} + \overline{D}$$





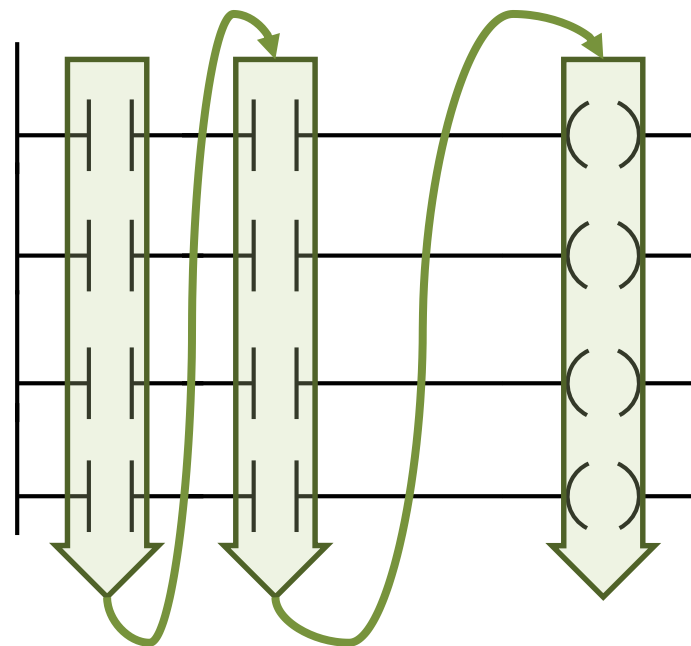
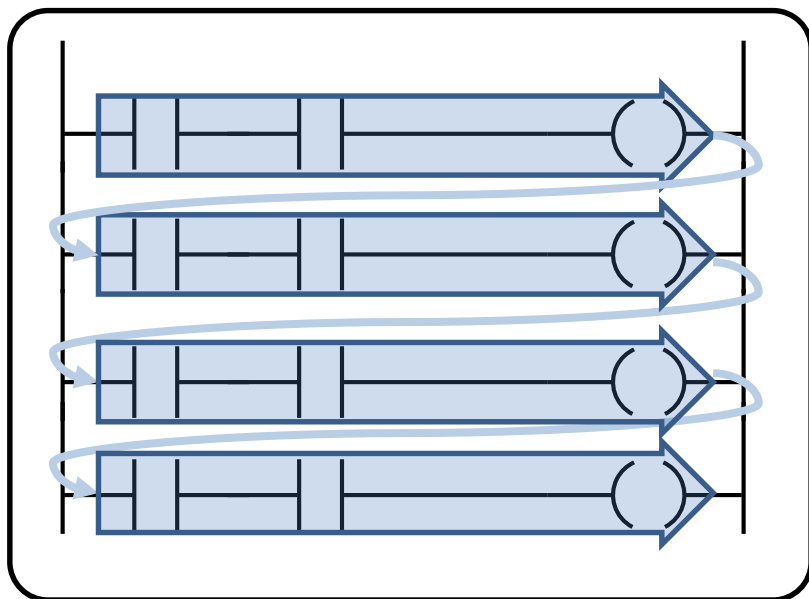
# A létradiagram kiértékelése

- A PLC ciklikus működésű
- A programvégrehajtás fázisában a teljes kód feldolgozásra kerül
- Minden egyes ciklusban a teljes létradiagram kiértékelésre kerül

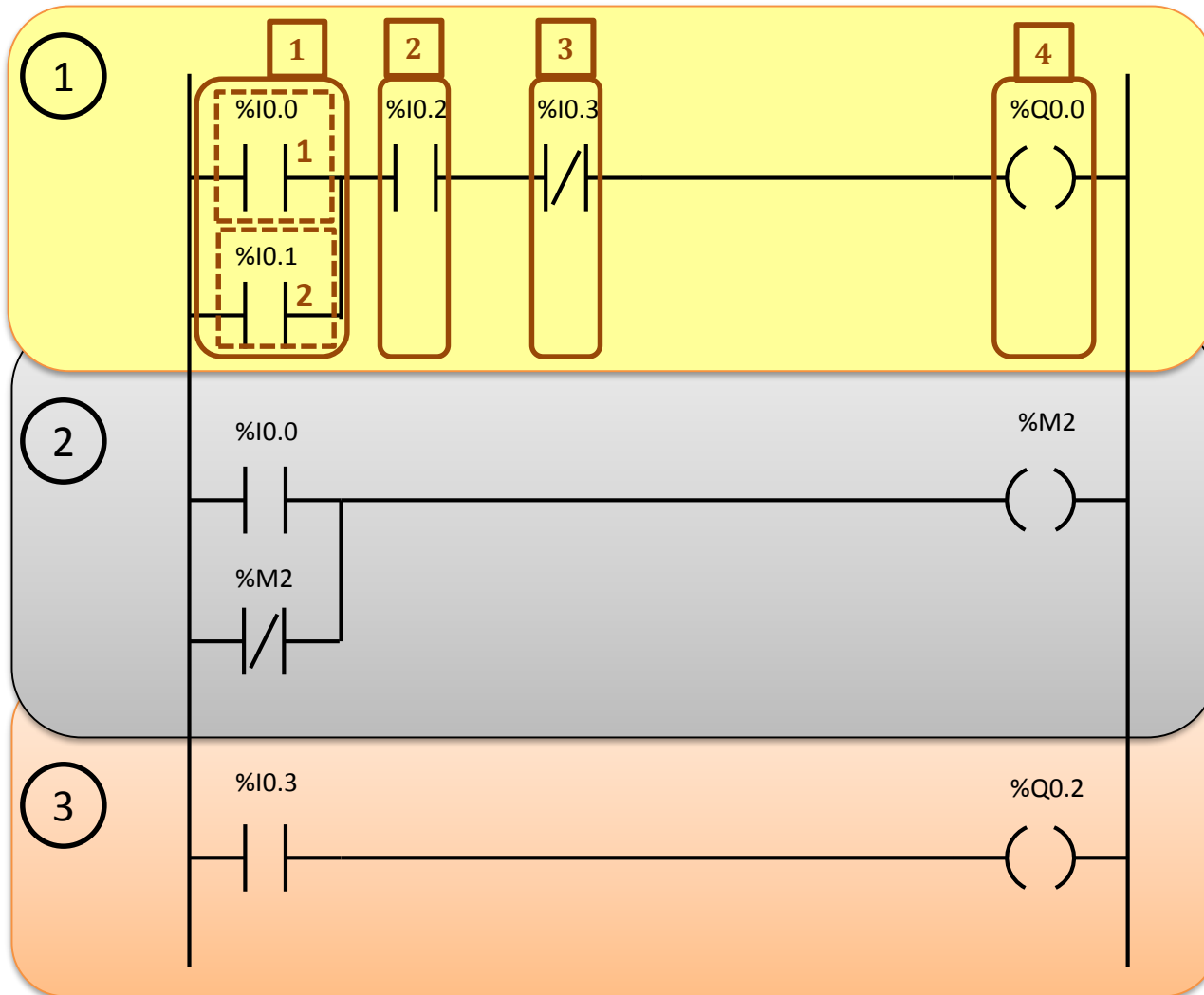


# A létradiagram kiértékelése

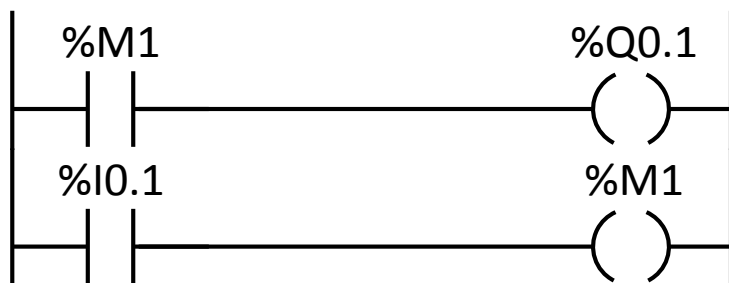
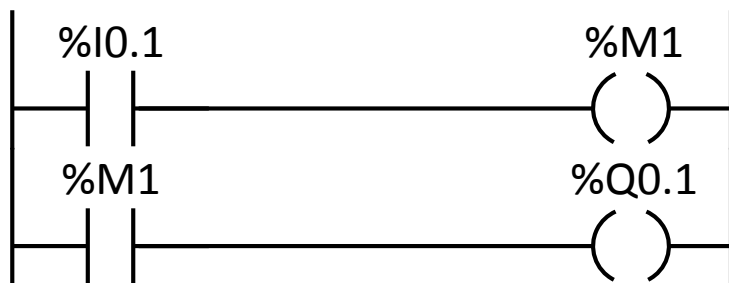
- Néhány nanoPLC-típus: létrasorok párhuzamos kiértékelése (ritka)
- Általános: **soros végrehajtás**
  - Soronként
  - Oszloponként (nagyon ritka)



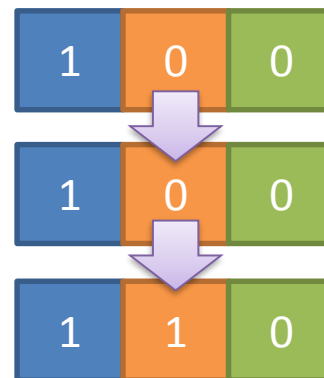
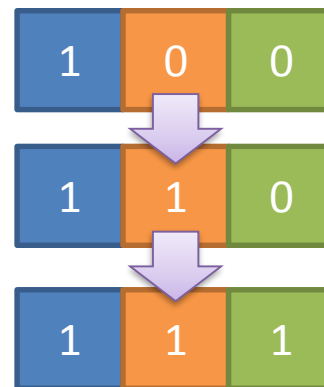
# A létradiagram kiértékelése



# A kiértékelési sorrend hatása

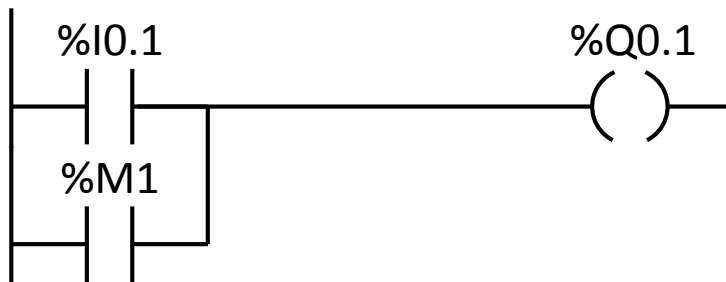
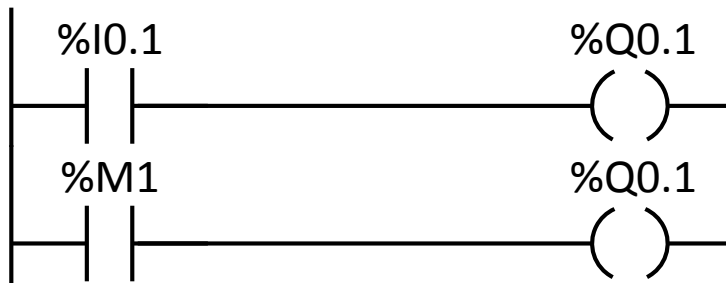


%I0.1 %M1 %Q0.1



# A kiértékelési sorrend hatása

$\%Q0.1 := \%I0.1 + \%M1$



$\%I0.1$   $\%M1$   $\%Q0.1$



Fizikai kimenetre

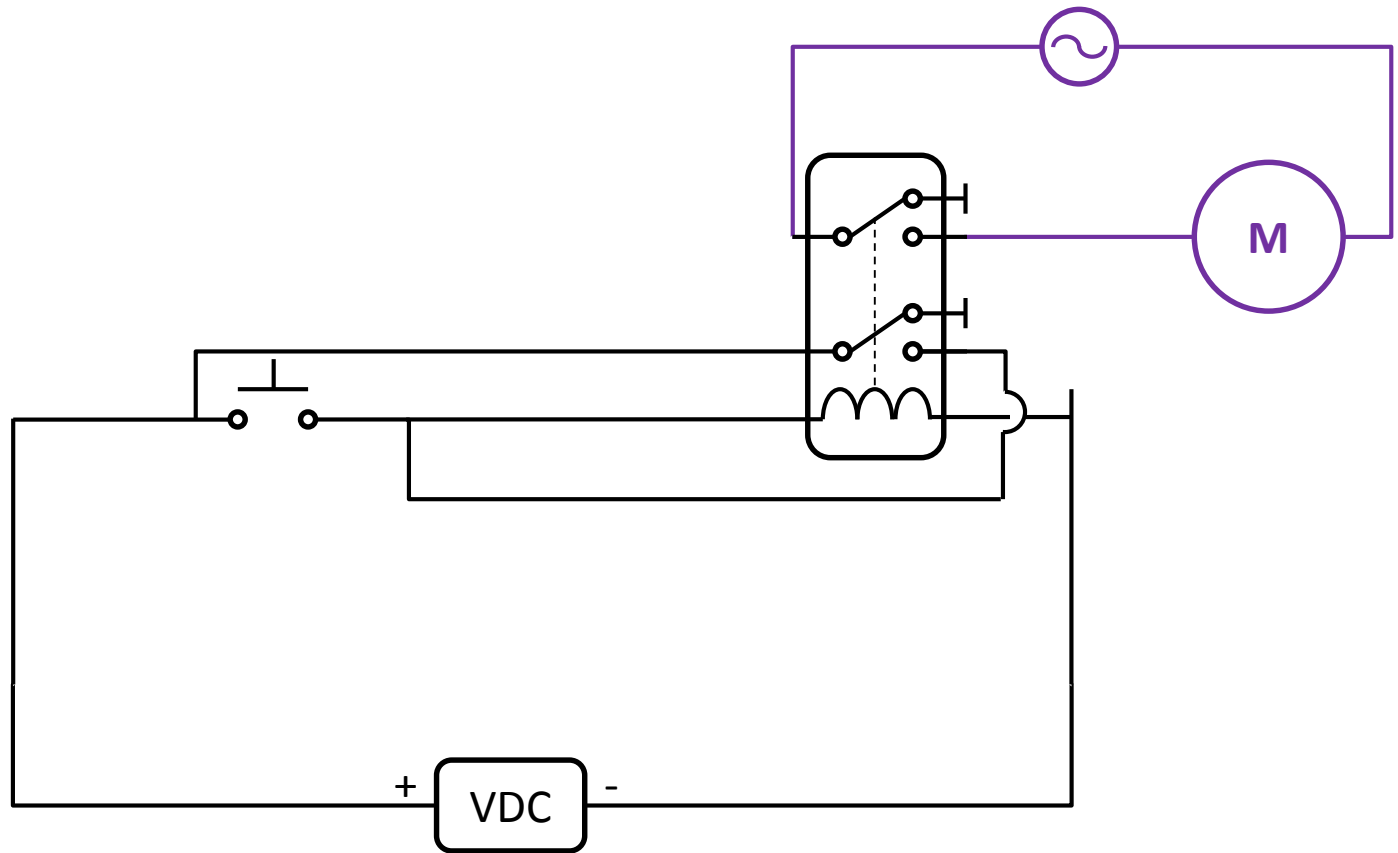


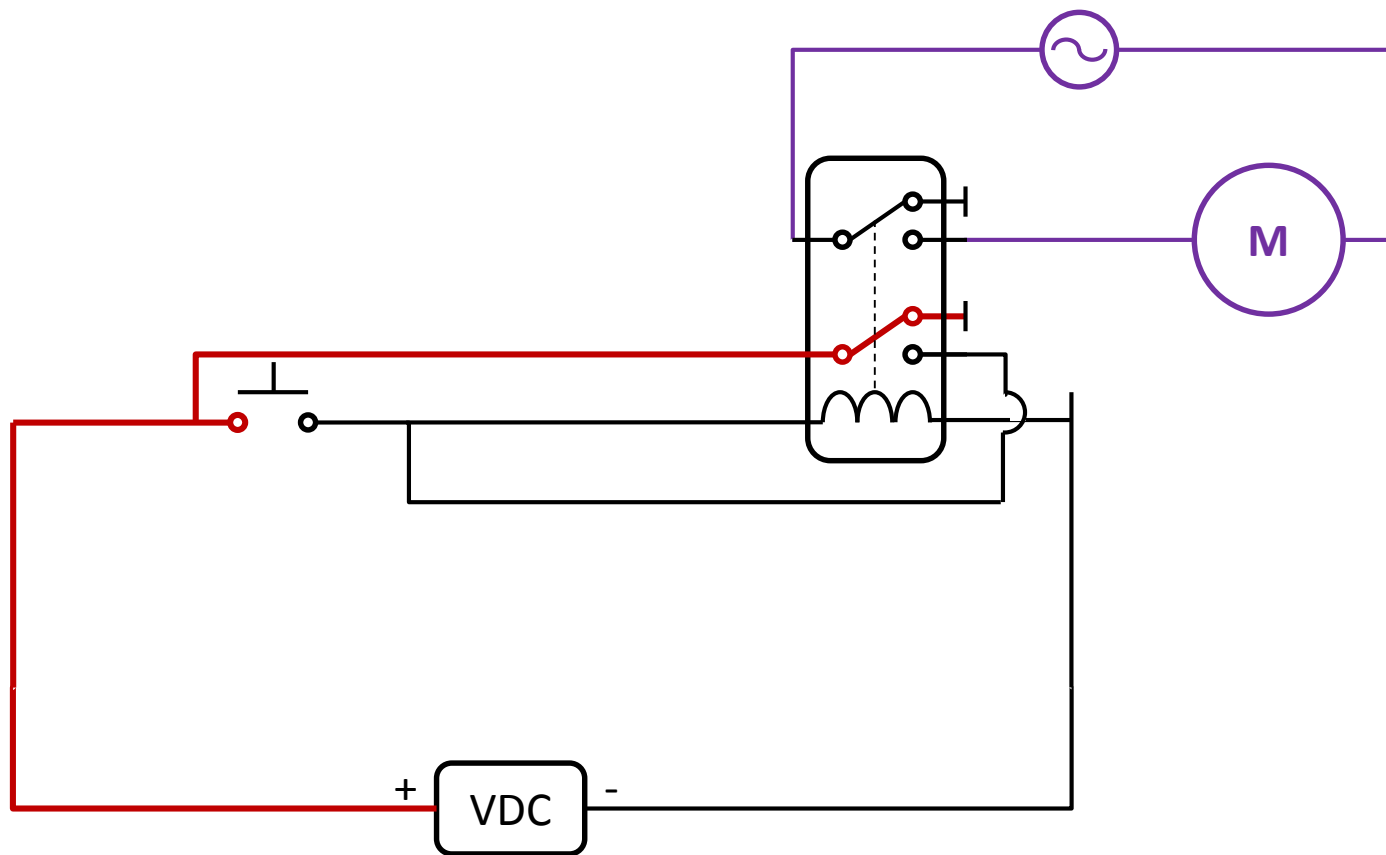
Fizikai kimenetre



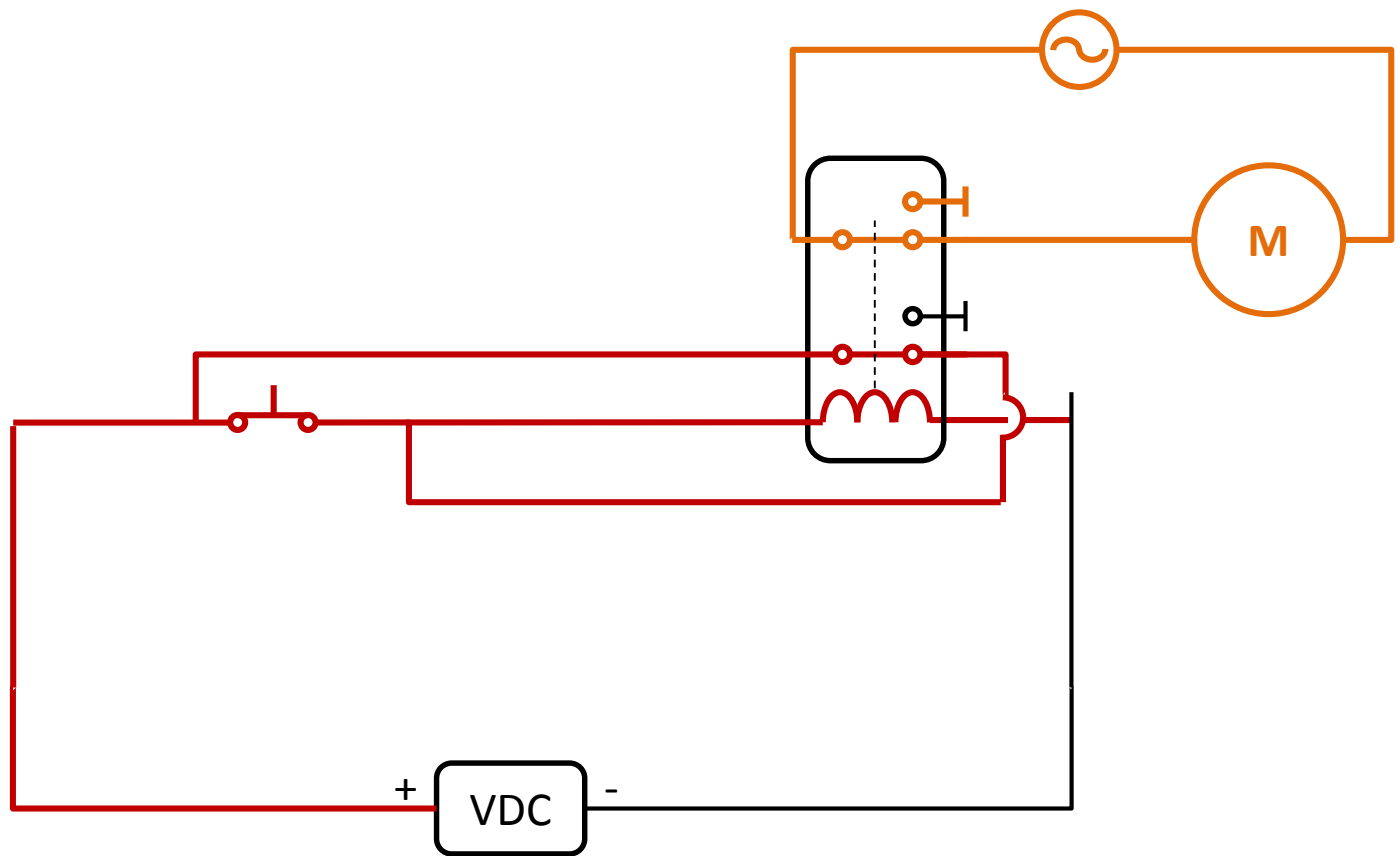
# Reteszelt tekercsek

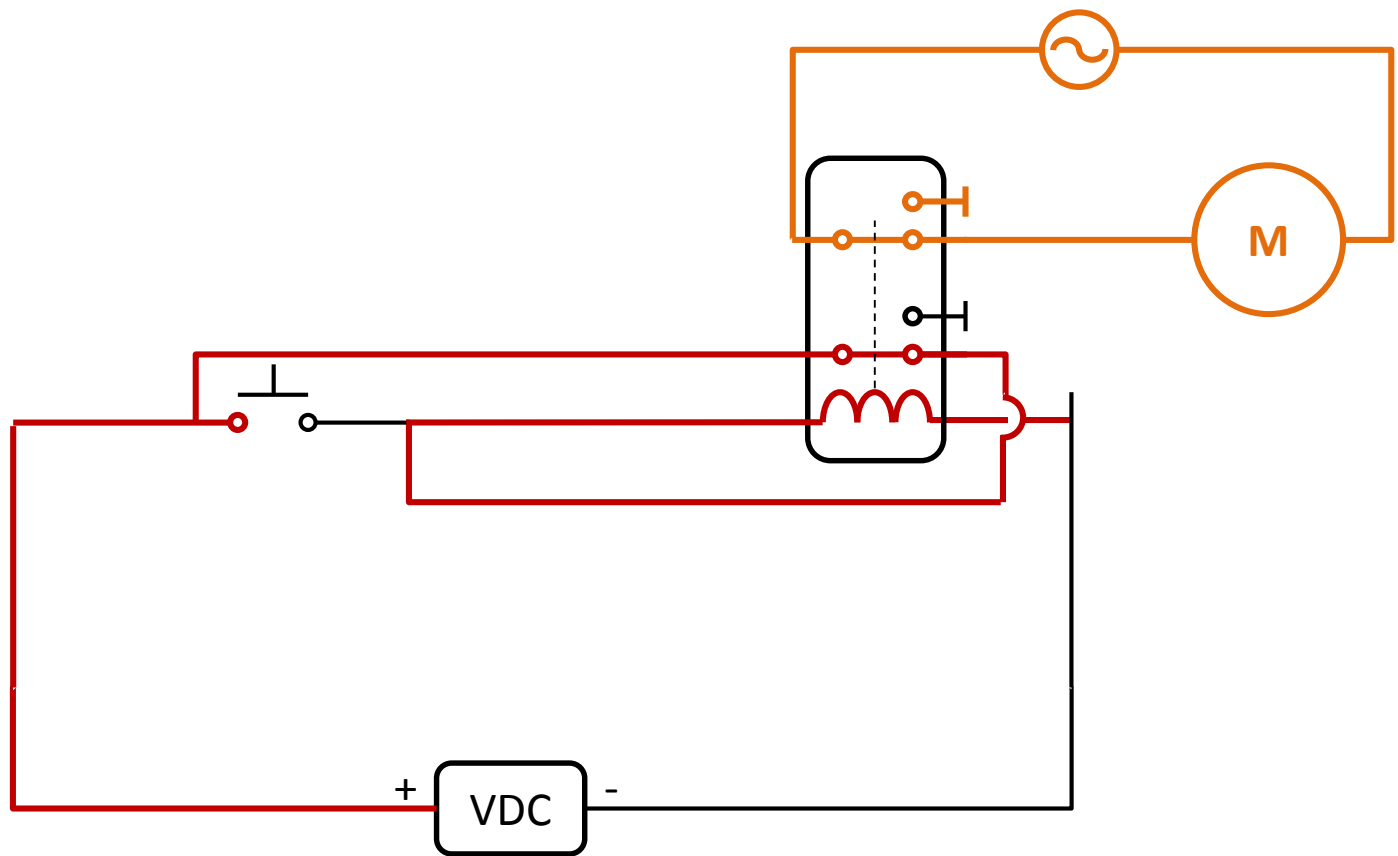
- Hogyan oldjuk meg, hogy a nyomógombunk reteszelje a kimenetet?





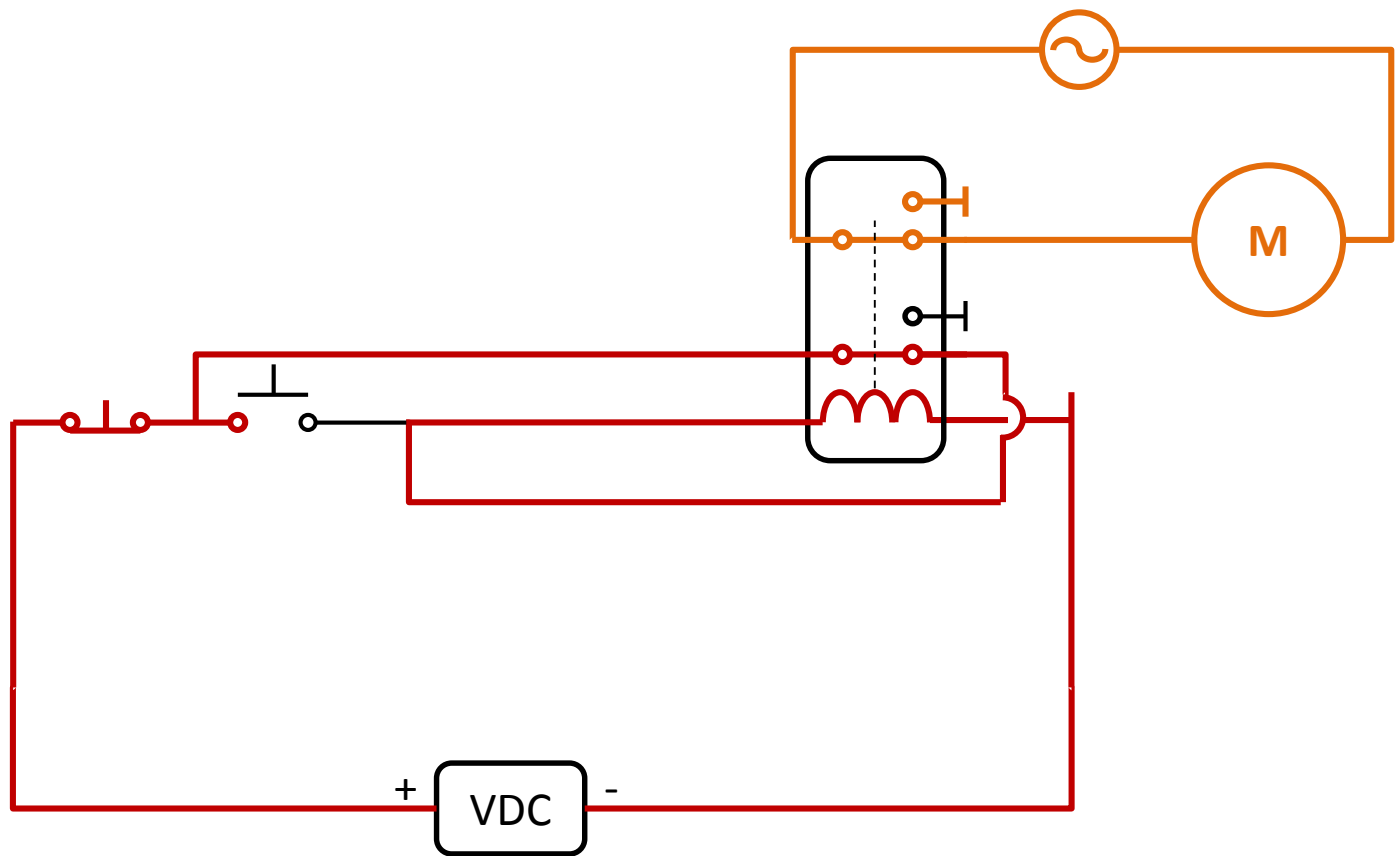


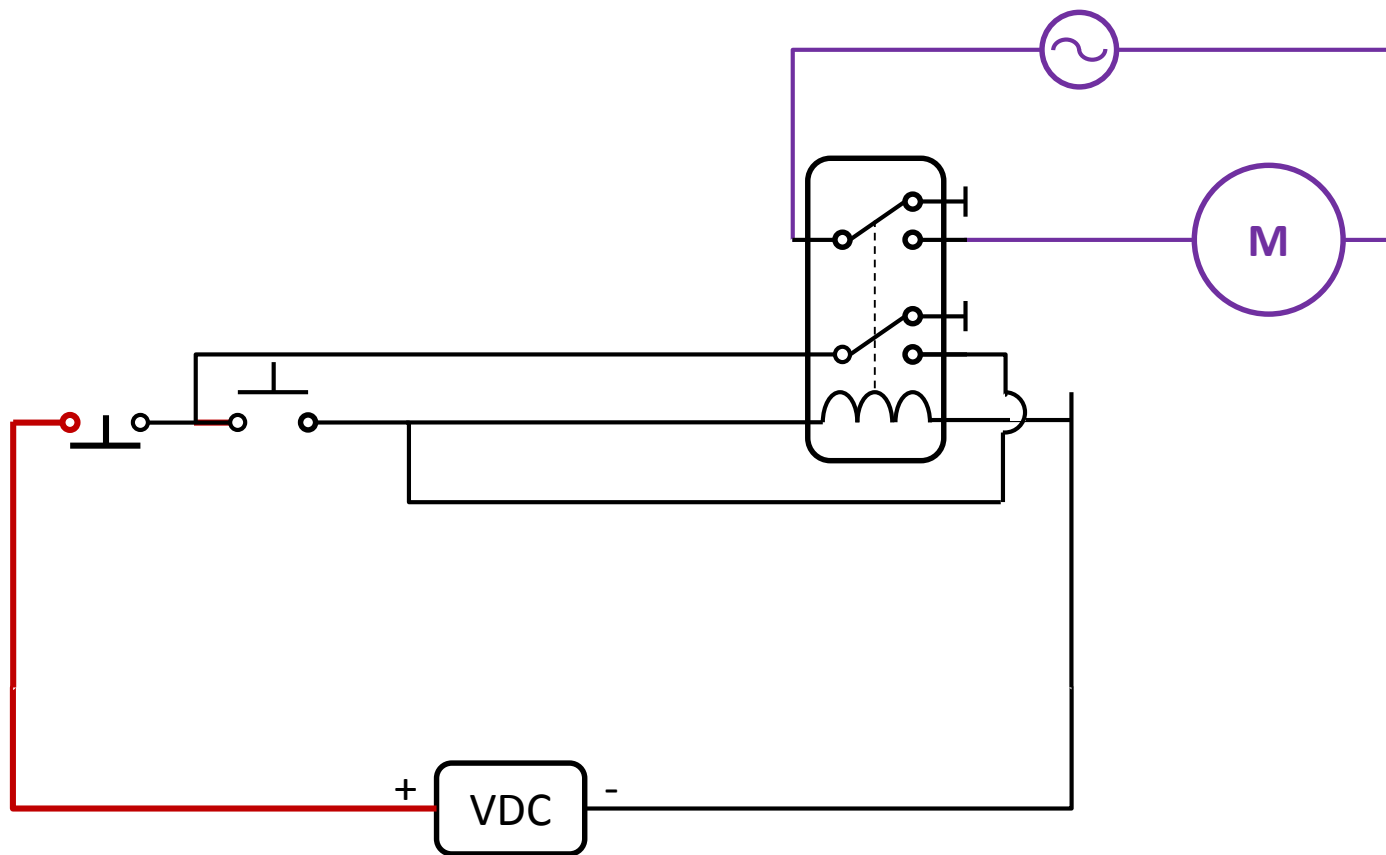




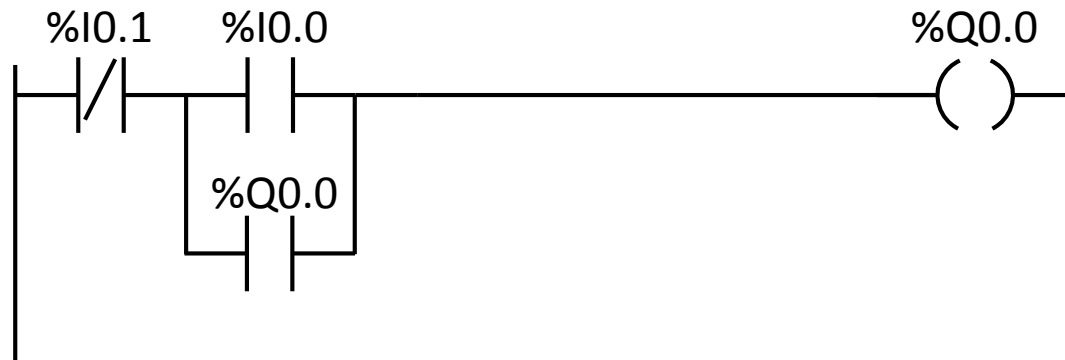
# Reteszelt tekercsek

- Megoldottuk, hogy a nyomógombunk reteszelje a kimenetet
- Hogyan oldjuk a reteszelést?







# Reteszelés





$$\%Q0.0 = \overline{\%I0.1} \cdot (\%I0.0 + \%Q0.0)$$

# Reteszelés

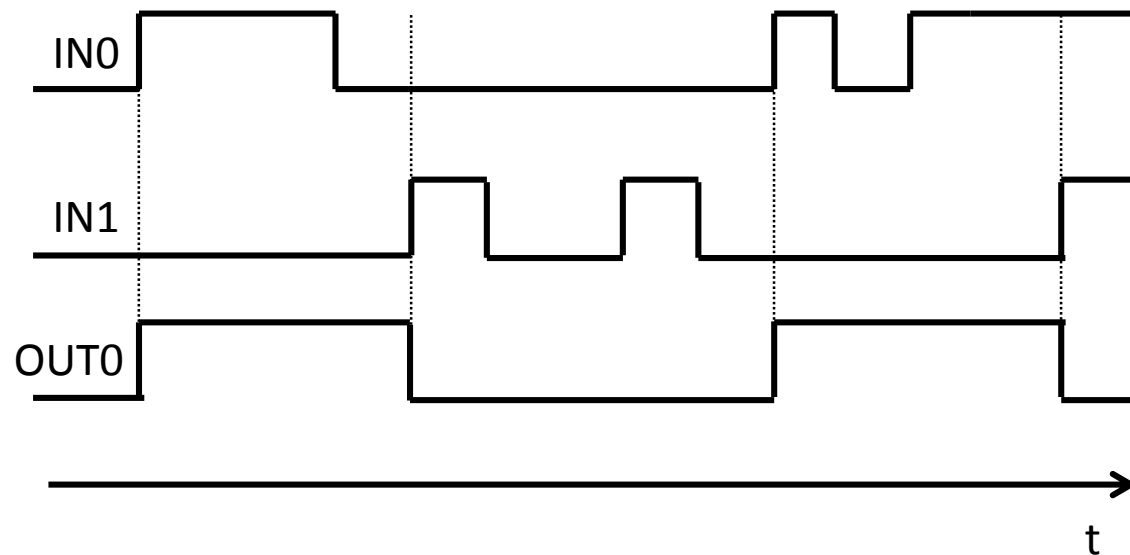
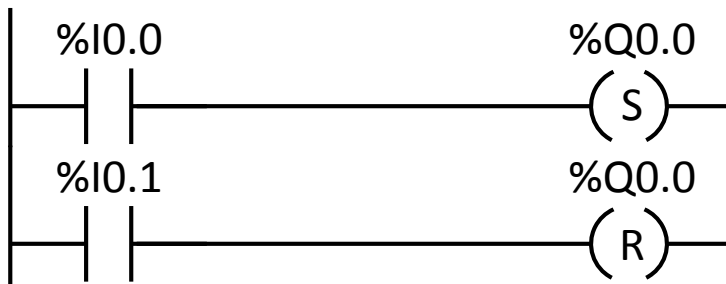
Szabványos jelölés

Symbol	Name
	Set tekercs
	Reset tekercs

RSLogix jelölés

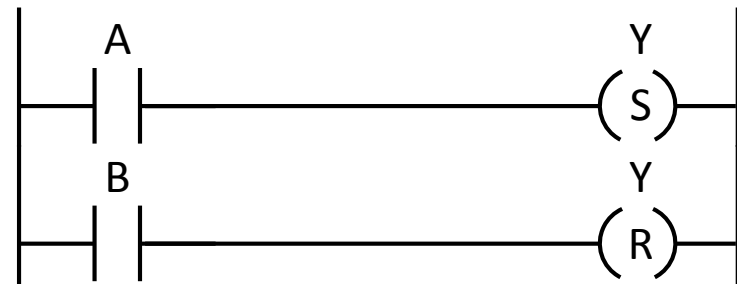
Symbol	Name
	Output Latch (OTL)
	Output Unlatch (OTU)

# Retteszelési műveletek





# A reteszelés értelmezése



**IF (A=1) THEN Y=1**

**IF (B=1) THEN Y=0**

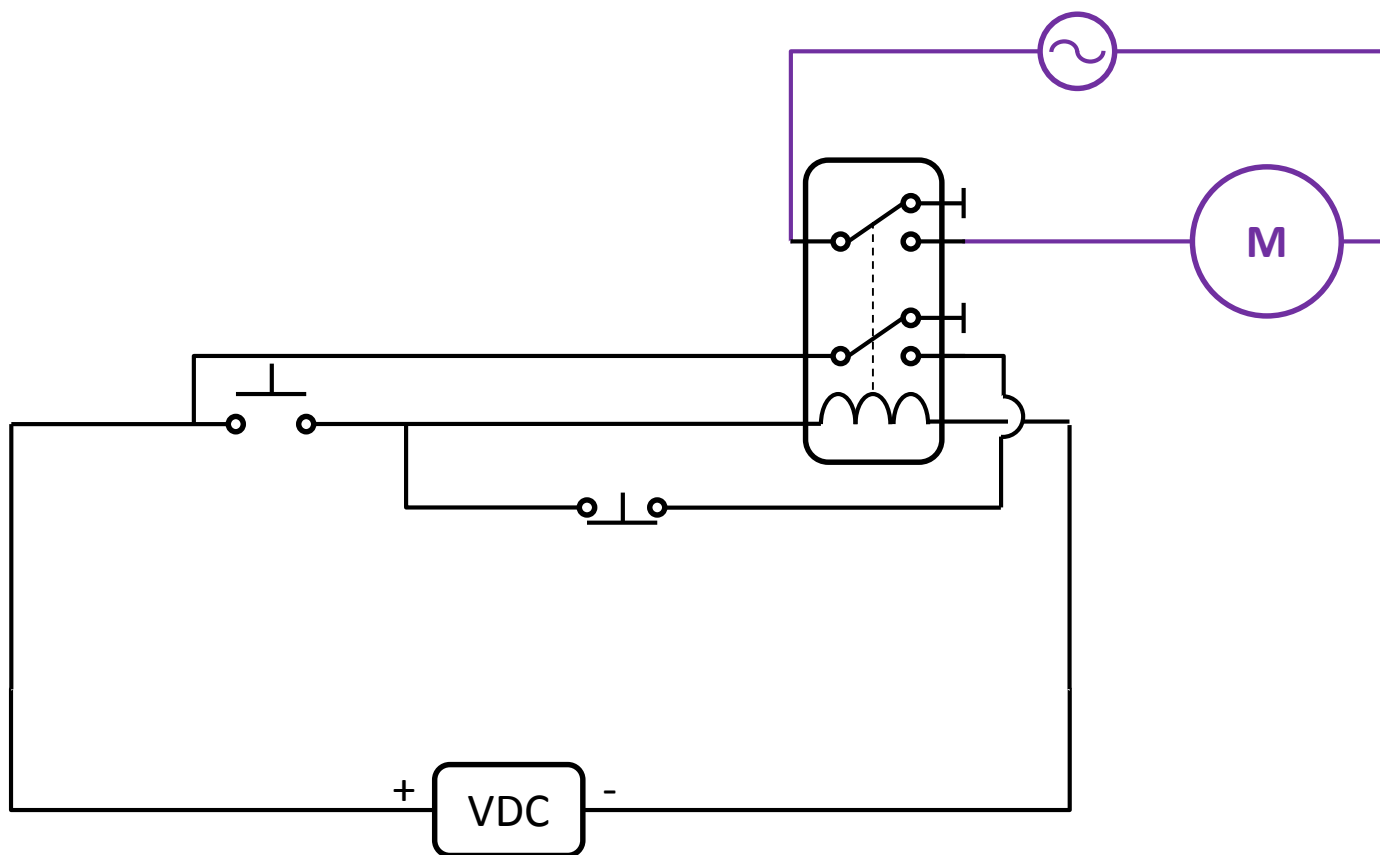


**IF (B=1) THEN Y=0**

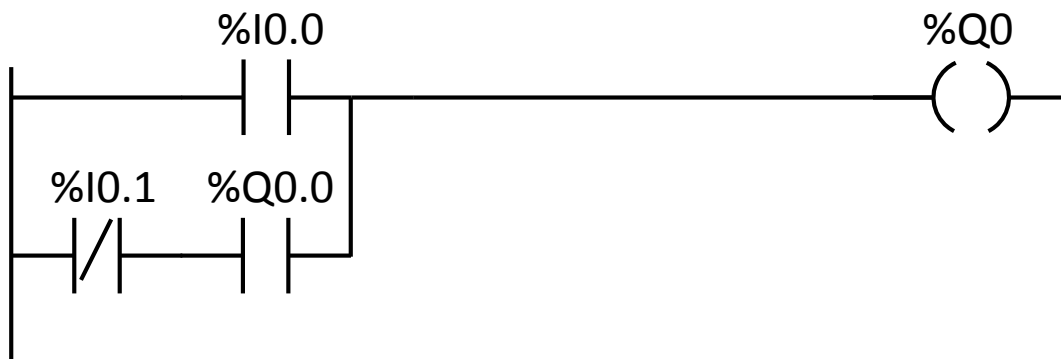
**ELSE**

**IF (A=1) THEN Y=1**

# Alternatív megoldás



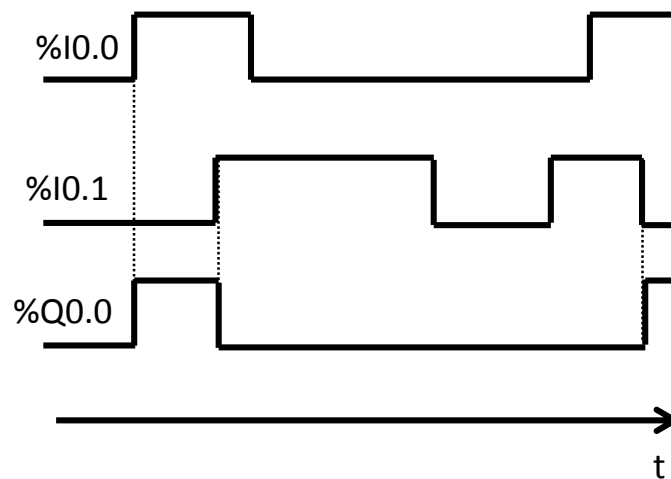
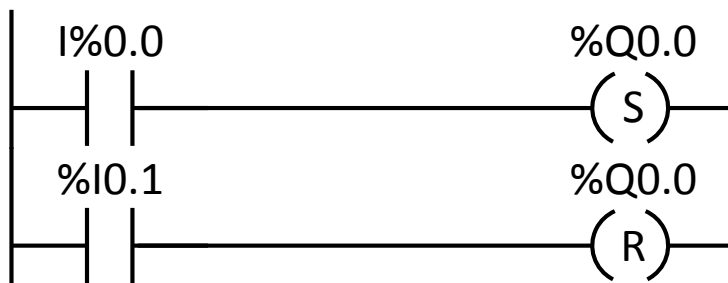
# Alternatív megoldás



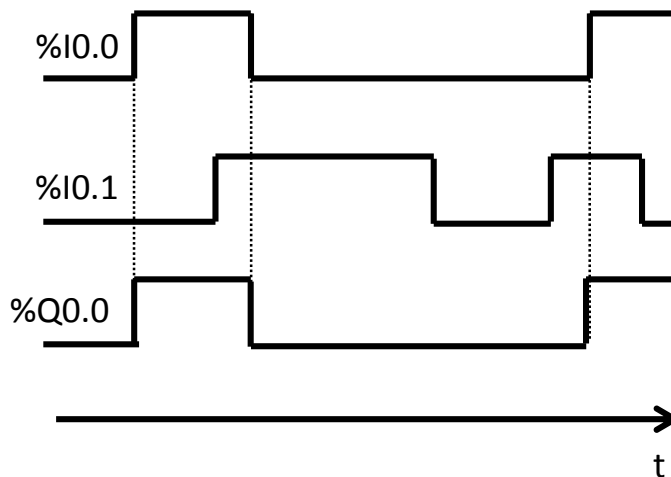
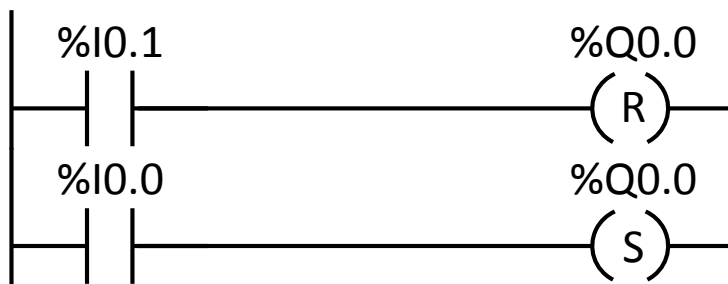
$$\%Q0 = \overline{\%I0.1} \cdot \%Q0.0 + \%I0.0$$

# Set / Reset prioritás

## Reset prioritás



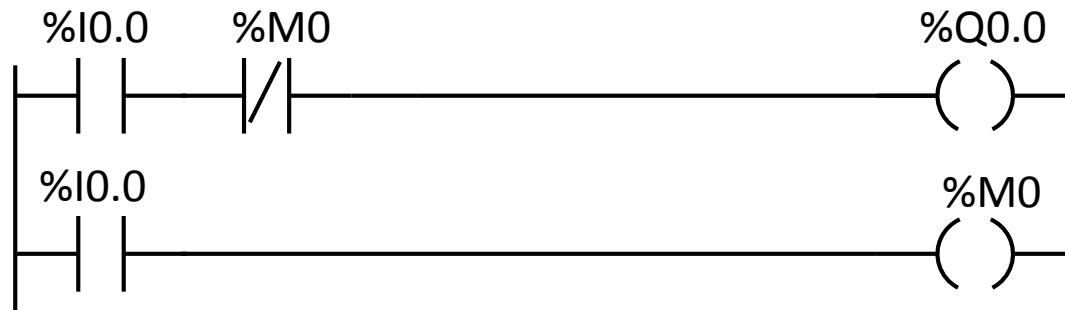
## Set prioritás



**Vigyázat! Az időben később végzett művelet írhatja felül a korábbi eredményét!**

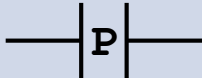
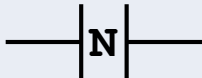
# Éldetektálás

- Hogyan érzékelhetjük egy bemenet fel- vagy lefutó élet?



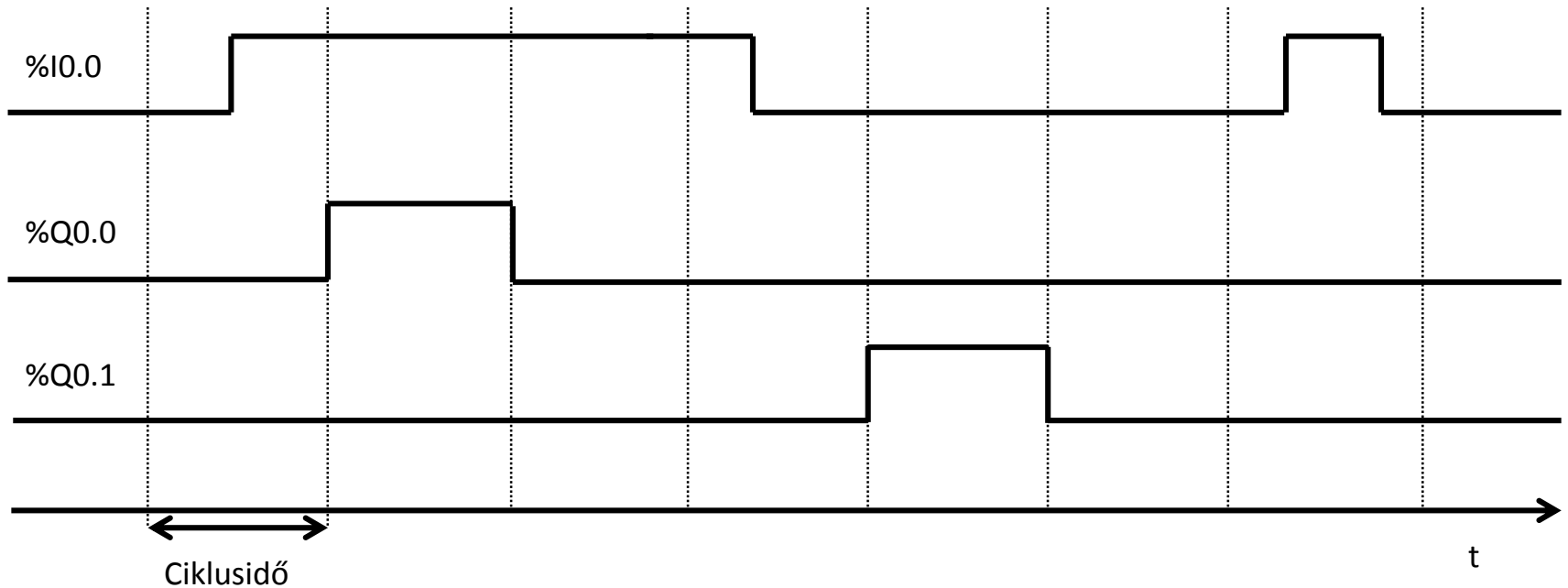
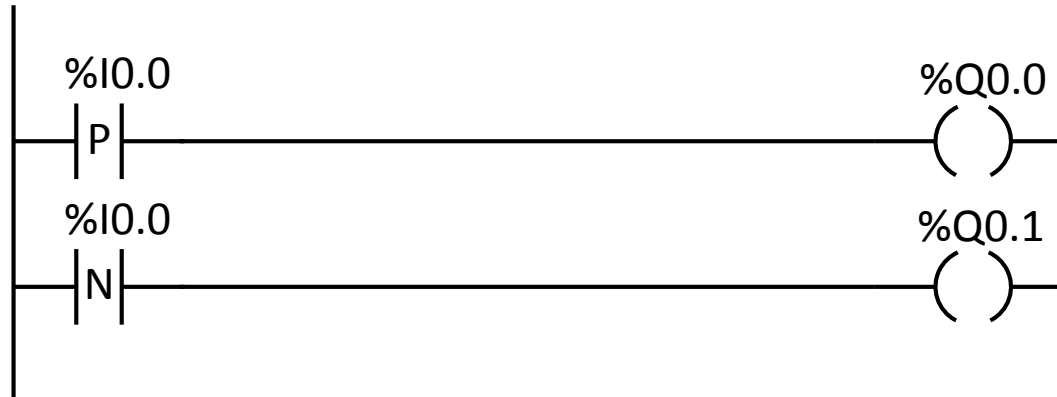
```
IF (%I0.0=1) AND (%M0=0) THEN %Q0.0:=1  
%M0:=%I0.0
```

# Érzékeny kontaktusok

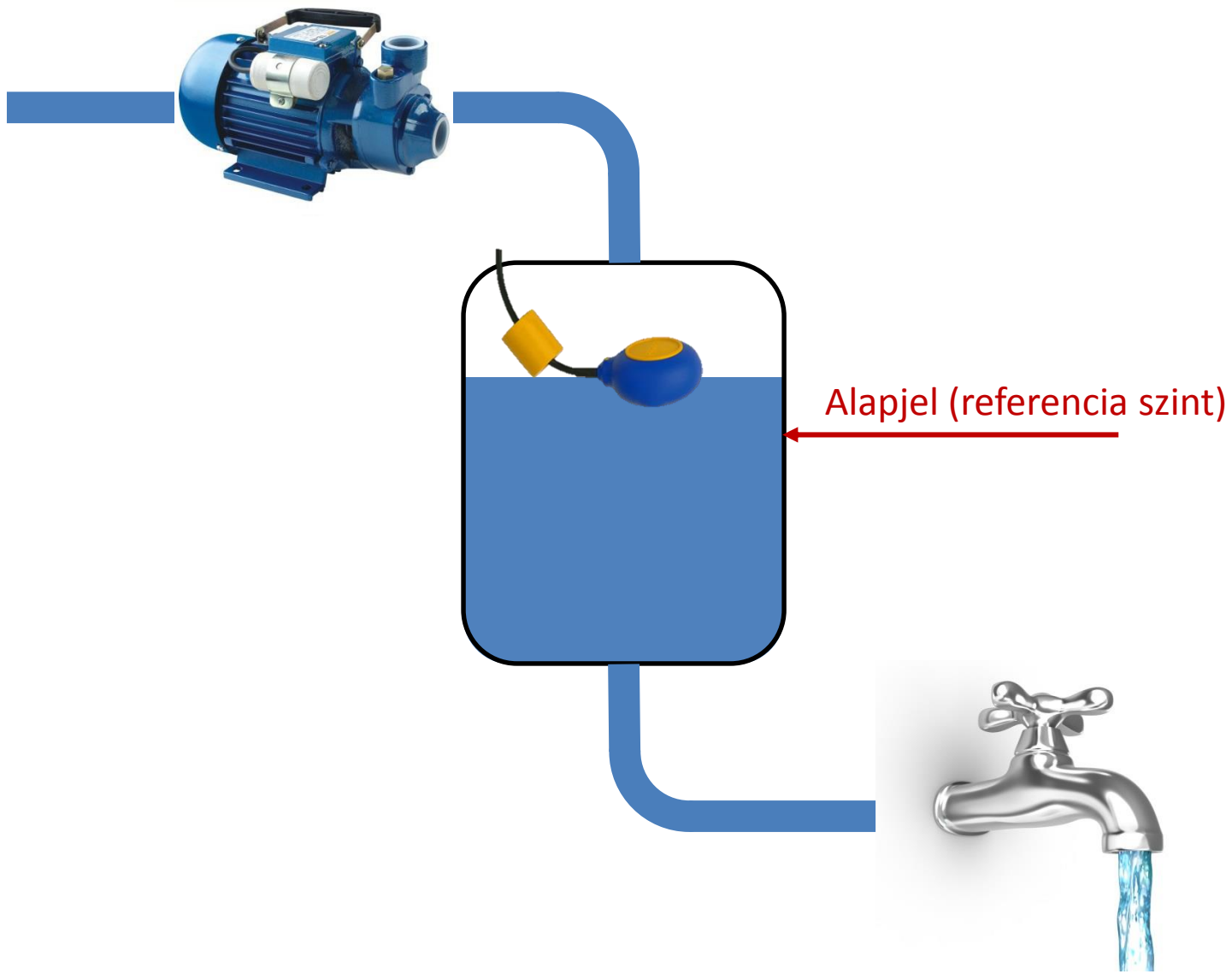
Szimbólum	Megnevezés
	Felfutó él érzékeny kontaktus <i>(rising edge contact , one shot rising)</i>
	Lefutó él érzékeny kontaktus <i>(falling edge contact, One Shot Falling)</i>

Megjegyzés: egyes fejlesztőkörnyezetekben az éldekektálás létradiagramban csak funkcióblokkok (ld. később) használatával oldható meg.

# Érzékeny kontaktusok

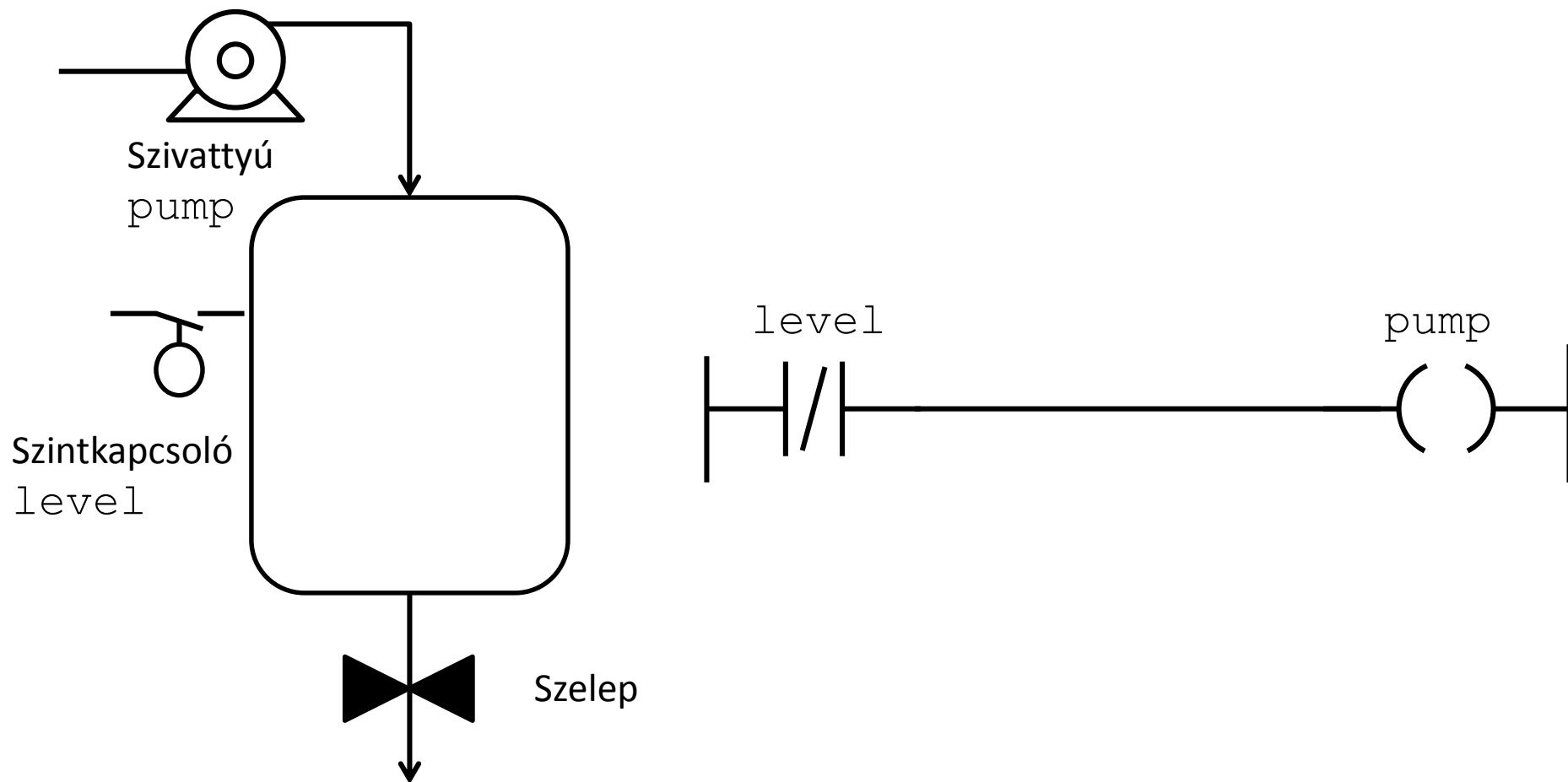


# Példa

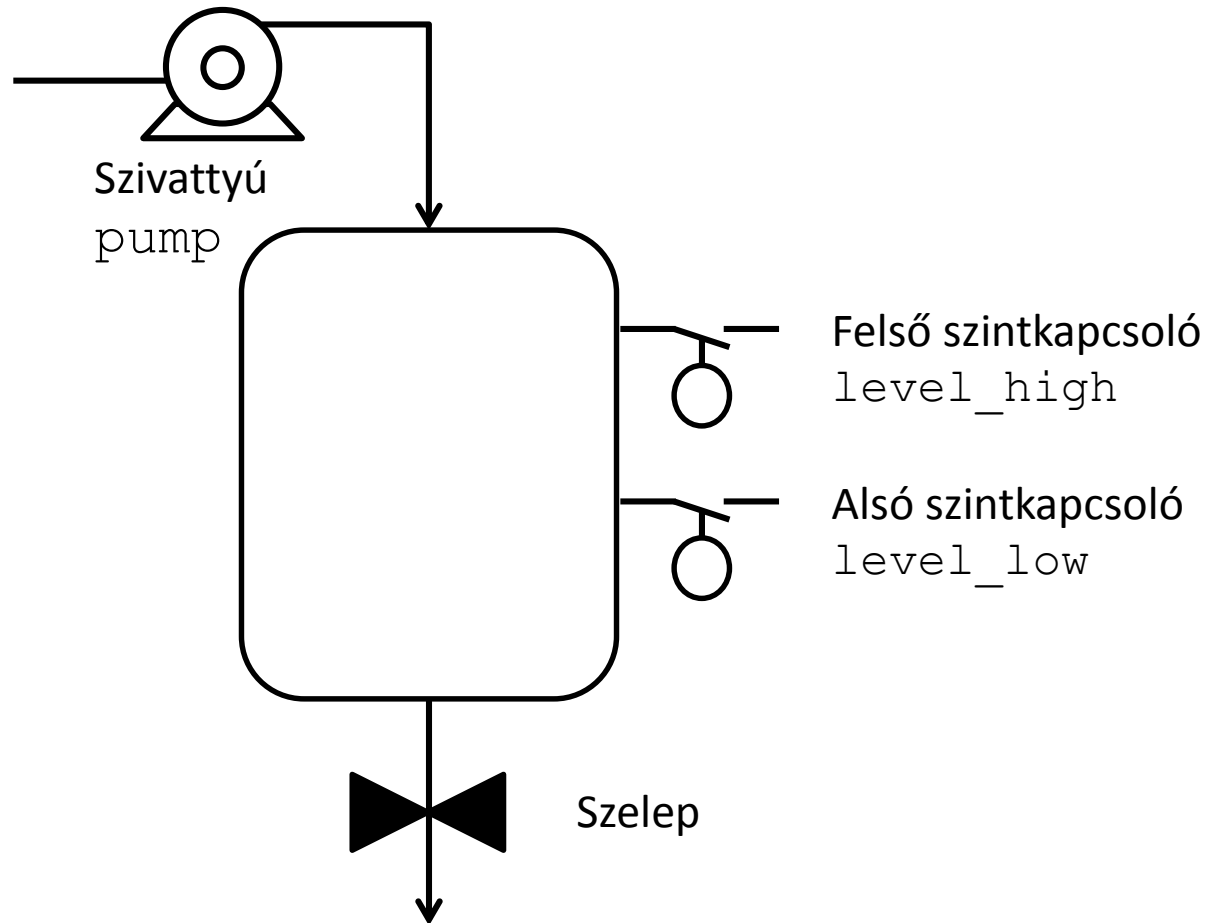




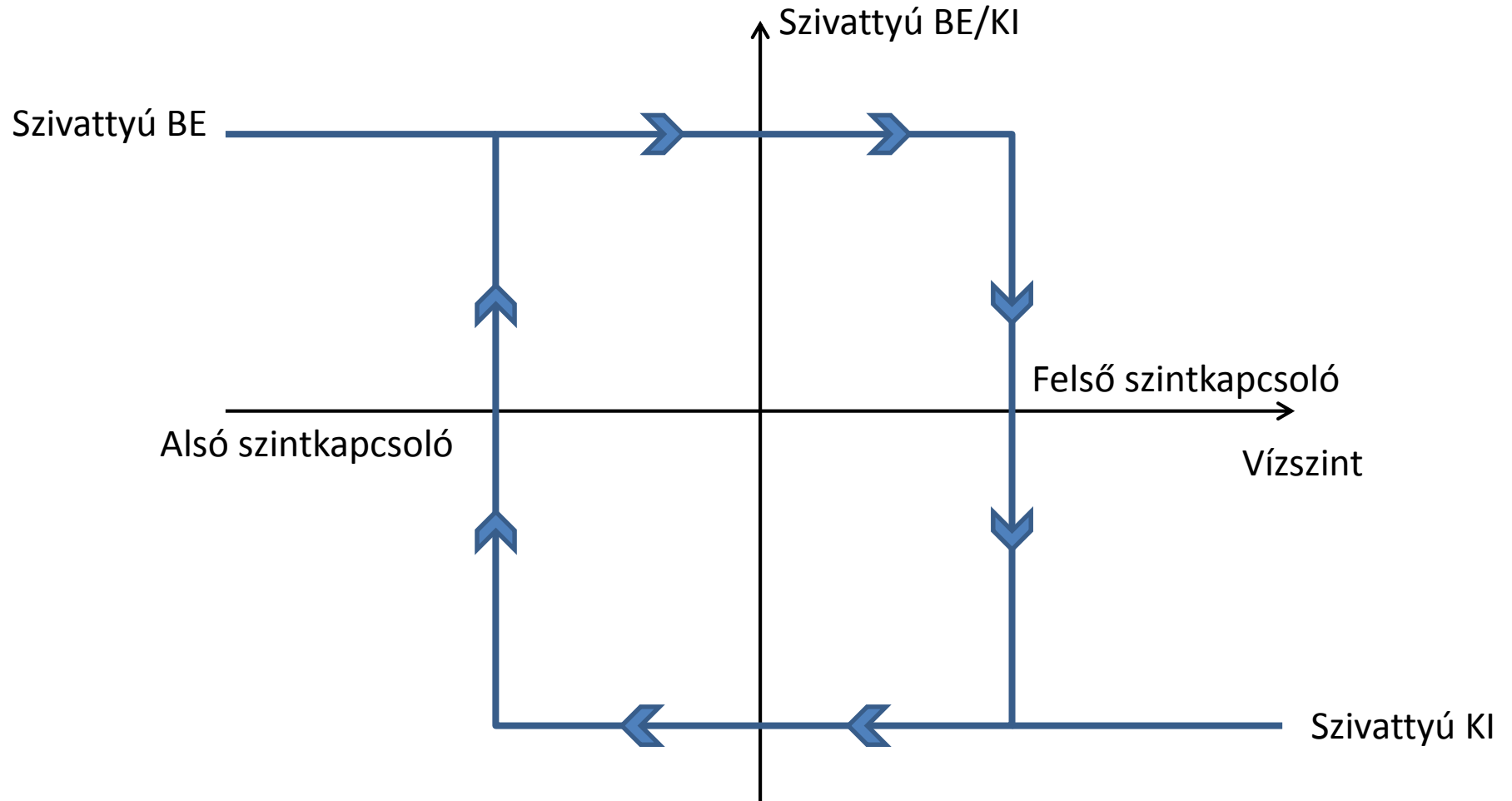
# Megoldás



# Példa – egy jobb megoldás



# Hiszterézis szabályozás



# Példa – egy jobb megoldás

