

# Értékünk AZ **EMBER**

Humán erőforrás-fejlesztési Operatív Program



Hodossy László

## PROGRAMOZOTT VEZÉRLÉSEK I.



SZÉCHENYI ISTVÁN  
EGYETEM  
GYŐR

Magyarország célba ér



Készült a HEFOP 3.3.1-P.-2004-09-0102/1.0 pályázat támogatásával.

Szerző: dr. Hodossy László  
főiskolai docens

Lektor: dr. Keresztes Péter  
egyetemi docens

# A dokumentum használata

## Mozgás a dokumentumban

A dokumentumban való mozgáshoz a Windows és az Adobe Reader megszokott elemeit és módszereit használhatjuk.

Minden lap tetején és alján egy navigációs sor található, itt a megfelelő hivatkozásra kattintva ugorhatunk a használati útmutatóra, a tartalomjegyzékre, valamint a tárgymutatóra. A ◀ és a ▶ nyilakkal az előző és a következő oldalra léphetünk át, míg a Vissza mező az utoljára megnézett oldalra visz vissza bennünket.

## Pozicionálás a könyvjelzőablak segítségével

A bal oldali könyvjelző ablakban tartalomjegyzékfa található, amelynek bejegyzéseire kattintva az adott fejezet/alfejezet első oldalára jutunk. Az aktuális pozíciókat a tartalomjegyzékfában kiemelt bejegyzés mutatja.

## A tartalomjegyzék és a tárgymutató használata

### Ugrás megadott helyre a tartalomjegyzék segítségével

Kattintsunk a tartalomjegyzék megfelelő pontjára, ezzel az adott fejezet első oldalára jutunk.

### A tárgymutató használata, keresés a szövegben

Keressük meg a tárgyszavak között a bejegyzést, majd kattintsunk a hozzá tartozó oldalszámok közül a megfelelőre. A további előfordulások megtekintéséhez használjuk a Vissza mezőt.

A dokumentumban való kereséshez használjuk megszokott módon a Szerkesztés menü Keresés parancsát. Az Adobe Reader az adott pozíciótól kezdve keres a szövegben.

# Tartalomjegyzék

<b>1. Előszó .....</b>	<b>5</b>
<b>2. Az irányítástechnika alapfogalmai és csoportosítása .....</b>	<b>7</b>
2.1. Az irányítások hierarchikus rendszere .....	9
2.2. Az irányítórendszerek fejlődése .....	12
<b>3. Programozható vezérlők.....</b>	<b>19</b>
3.1. Egy kis PLC-történelem.....	19
3.2. A PLC funkcionális egységei.....	21
3.3. A PLC-k szerkezeti felépítése .....	25
3.4. A bitszervezésű PLC működése .....	27
3.5. Mikroprocesszor alapú PLC-k hardverfelépítése .....	33
<b>4. A PLC-k programozása .....</b>	<b>47</b>
4.1. A PLC-ben futó programok és feladataik .....	47
4.2. Alapszoftver.....	47
4.3. Felhasználói programok.....	49
4.4. PLC programnyelvek.....	51
4.5. A felhasználói programok végrehajtásának módjai.....	62
<b>5. Programozható vezérlők hálózatba kapcsolása .....</b>	<b>71</b>
5.1. A hálózattechnika alapjai .....	73
5.2. Az adatátvitel elve .....	73
5.3. Hálózati topológiák .....	74
5.4. Átviteli közeg.....	77
5.5. Buszhozzáférési eljárások .....	79
5.6. Telegram felépítése .....	83
5.7. Adatvédelmi módszerek.....	86
5.8. Átviteli módok.....	89
5.9. Buszprotokoll .....	90
5.10. Hálózati elemek.....	93
5.11. Irányítástechnikai hálózatok .....	94
<b>6. PLC programozási példák.....</b>	<b>96</b>
<i>Irodalomjegyzék.....</i>	<i>119</i>
<i>Tárgymutató .....</i>	<i>120</i>

# 1. Előszó

A programozható vezérlők világába vezető olvasmányt tart kezében a kedves olvasó. A programozható vezérlők megjelenésüktől napjainkig jelentős fejlődésen mentek keresztül, s ma szerte a világon a legkülönbözőbb ágazatokban egyre szélesebb körű alkalmazásra találnak. A szabadon programozható vezérlők nagyszámú elterjedését az okozza, hogy a legszélesebb alkalmazási területek találkoztak ezzel az új technológiával. Így a villanyszerelőktől a magasan kvalifikált számítástechnikai szakemberekig csábít a felhasználás lehetősége.

A közös probléma gyakran az eltérő szakmai felkészültségekben rejlik. A műszerész vagy villanyszerelő általában ismeri a Programozható Logikai Vezérlő (PLV vagy PLC) be- és kimeneti egységeire kapcsolt berendezéseket, de nem ismeri részleteiben a PLC belső felépítését és működését, míg a számítástechnikai szakember a hardverek és szoftverek világában járatos. A két ismeretkör áthidalását célozza meg ez a jegyzet, azaz alapvető ismereteket kíván nyújtani azon főiskolai és egyetemi hallgatók számára, akik némi előismeretek birtokában szándékoznak megismerni ezen készüléktípus belső felépítését, alapvető működését, programozási lehetőségeit és utalást találnak a hálózatba kapcsolásuk módjára is.

A jegyzet első része az irányítórendszerek fejlődésével, csoportosításával foglalkozik. A következőkben a programozott vezérlők kialakulásáról és fejlődéséről illetve az alapvető felépítésről olvasható ismertetés. Ezt követően a Programozható Logikai Vezérlők (PLC-k) működésének majd programozási lehetőségeinek ismertetése következik, amelyet a hálózati kialakítás lehetőségeinek és jellemzőinek rövid bemutatása követ. A jegyzet utolsó részében Siemens S7 sorozatú PLC-kre készült néhány programozási példa segíti a korábbi fejezetek elméleti ismereteinek gyakorlatban történő alkalmazását.

A jegyzet 3. (Programozható vezérlők) és 4. (A PLC-k programozása) fejezete Dr. Ajtonyi István, Dr. Gyuricza István: „Programozható irányítóberendezések, hálózatok és rendszerek” című könyve [2] alapján készült. Ebből a könyvből kerültek átvételre a jegyzet alábbi számú ábrái: 3.1.-3.15., 3.17.-3.23., 4.1.-4.11., 4.14.-4.17., 4.19.

A jegyzet egyik vagy másik fejezete az olvasó számára bizonyára ismert. Azonban a teljes áttekintés céljából az elmélet éppen olyan fontos, mint a programozási eljárás ismertetése. A jegyzet tanulmányozása elekt-

ronikai, digitálistechikai, mikroprocesszoros és irányítástechnikai ismereteket tételez fel. Természetesen bizonyos témakörök részletesebb tanulmányozásához a gyártó cégek kiadványainak, katalógusainak ismerete is szükséges.

Remélem, hogy minden olvasó talál a jegyzetben számára érdekes és új ismeretet.

A szerző ezúton mond köszönetet Dr. Keresztes Péter egyetemi docens lektornak (Széchenyi István Egyetem, Automatizálási Tanszék) az anyag összeállítása és egységesítése érdekében közölt és részletekbe menő nagy számú, igen értékes és hasznos szakmai tanácsaiért, észrevételeiért.

Kívánom minden kedves olvasómnak, hogy elérje célját munkám kézbe vételével. Észrevételeivel, javaslataival, ha vannak, kérem, keressen meg. Jó munkát, jó tanulást!

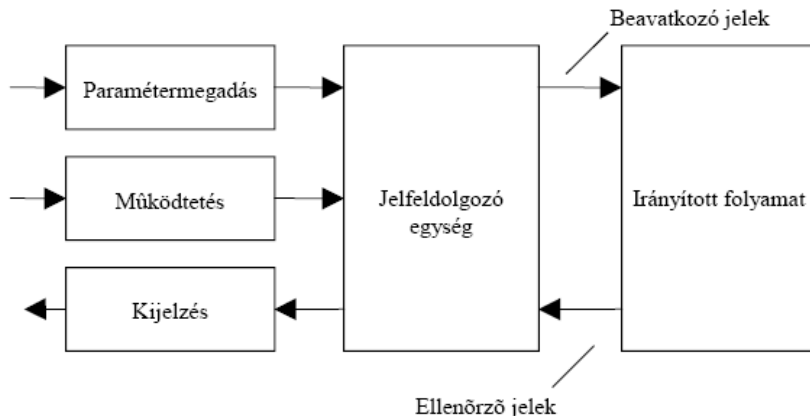
2006. szeptember

a szerző  
[hodossy@sze.hu](mailto:hodossy@sze.hu)

## 2. Az irányítástechnika alapfogalmai és csoportosítása

Az irányítástechnika, s ezen belül a szabályozástechnika, a vezérléstechnika és a felügyelet fogalmát szabványok rögzítik (Magyarországon az MSZ 18 450 szabvány).

Az önműködő irányítás esetén valamennyi irányítási művelet és annak minden részművelete (a folyamat elindítása, fenntartása, megváltoztatása vagy megállítása) kezelői beavatkozás nélkül megy végbe. Az irányítás részműveletei: érzékelés, ítéletalkotás, rendelkezés, beavatkozás. Az irányítás tárgyát képező technológiai folyamat, berendezés, gép, stb. jelenti az irányított rendszert. Az irányítórendszer pedig mindazon szervek és készülékek együttese, amelyek együttműködése révén az irányított rendszer irányítása megvalósul. Egy adott technológia irányítása annak jellegétől függően kétféle módon lehetséges: vezérléssel illetve szabályozással. A kétféle irányítási mód a hatáslánc felépítésében különbözik. Az irányítási folyamat általános blokkvázlata látható az alábbi ábrán:



2.1. ábra. Az irányítási folyamat általános blokkvázlata

A paramétermegadás a rendszer megkívánt működéséhez szükséges adatok (alajelek, technológiai jellemzők) megadását jelentik. A működtetés a berendezés működtetése céljából szükséges be- és kikapcsolás, vészleállítás stb. jeleit jelenti. A működtető jel általában emberi beavatkozás eredménye, de származhat az irányító vagy az irányított berendezéstől is. A kijel-

zés a működés fázisainak és egyéb jellemzőinek visszajelzése a kezelő, ill. felügyelő személyzet részére. Általában a kijelzések jelentős segítséget nyújtanak az esetleges hibák helyeinek felismerésében is. Az ellenőrző jeleket a technológiai folyamatban elhelyezett érzékelők szolgáltatják. Az itt használt érzékelők vagy kétállapotú jeleket szolgáltatnak, vagy analóg értékeket, melyeket sokszor a digitális feldolgozhatóság érdekében digitalizálunk. Az irányítójelek (beavatkozójelek) kiadása a beavatkozószerveken keresztül történik. Ezek általában jelfogók, mágneskapcsolók, mágnesszelepek stb. A jelfeldolgozó egység, amely a vizsgálatunk szempontjából a legfontosabb, alapvetően kétféleképpen működhet: sorrendi (szekvenciális) és kombinációs hálózatként.

Felépítését tekintve a feldolgozóegység huzalozott programú vagy tárolt programú lehet. A huzalozott programú vezérléseknél az egyes elemek (jelfogók vagy félvezető logikai elemek) összekapcsolása huzalozással oly módon történik, hogy ez határozza meg az egység működését, a bemenetek és a kimenetek közötti logikai kapcsolatot. Ezek az összeköttetések legtöbbször nem, vagy csak igen nehezen módosíthatók, vagyis a huzalozott logikai kapcsolat megváltoztatása komoly nehézséget jelent. Vannak olyan berendezések, ahol a huzalozott logikai kapcsolat változtatását dugaszolással teszik lehetővé. A tárolt programú vezérlőberendezéseknél a bemeneti és kimeneti jelek közötti kapcsolatot (ezek logikai függvényeit) egy tárolt program (felhasználói program) határozza meg. A tároló áramkör lehet pl. csak olvasható memória (ROM). Ekkor, ha az áramkör nem cserélhető, akkor rögzített, ha cserélhető, akkor cserével változtatható a berendezés programja. Más tárolóáramkörök alkalmazása esetén (pl. RAM, EEPROM) a tárolt program újraírható. Esetenként a tárban egymástól függetlenül több programot is elhelyeznek, amelyek közül a kívántat külső jelre vagy időtől függően aktivizálják.

A vezérlések felépítéséhez használt elemeket és a vezérlések struktúráját tekintve az alábbi felosztást tehetjük:

- jelfogós (relés) vezérlések
- félvezető logikai elemekre épülő vezérlések
- számítógépes vezérlések
- programozható logikai vezérlők.

A szabályozás hatáslánca zárt, azaz a szabályozott jellemző kívánt értéke az irányított szakasztól visszacsatolással valósul meg. Erre utal a szabályozás angol megnevezése is: closed loop control azaz zárt hurkú irányítás. A



zárt hurkú irányítás kedvező tulajdonsága, hogy különböző zavaró hatások kiküszöbölésére is alkalmas. Hátránya ugyanakkor a visszacsatolásból származik, hiszen bizonyos feltételek esetén a rendszer instabillá válhat.

A vezérlés nyílt hatáslánccal rendelkezik (angol megnevezése: open loop control, azaz nyílt hurkú irányítás), ugyanis nincs visszacsatolás a vezérelt szakasról, tehát a rendelkezőjel a vezérelt jellemzőtől függetlenül jön létre. A nyílt hatásláncú vezérlés csak a rendszer tervezésekor számításhoz vett zavarójeleket képes kiküszöbölni, strukturálisan mindig stabil működésű.

Régebben a szabályozó- illetve vezérlőkészülékek mind felépítésükben, mind kivitelükben is elkülönültek egymástól. Napjainkra a funkcionális integrálódás a jellemző és a merőben eltérő alkalmazások esetén is gyakorlatilag ugyanazon a hardver bázison kerülnek megvalósításra, csak a működtető szoftverekben van különbség.

A technológiai folyamatok jellegükből adódóan három csoportba sorolhatók: folyamatos, diszkrét és vegyes technológia.

Folyamatos technológia esetén a folyamat nem osztható időben ismétlődő részfolyamatokra. Az ilyen technológiák irányításánál a szabályozási feladatok a meghatározóak. Leginkább a vegyipar, a gyógyszeripar, az élelmiszeripar stb. területén lehet ilyen folyamatokkal találkozni.

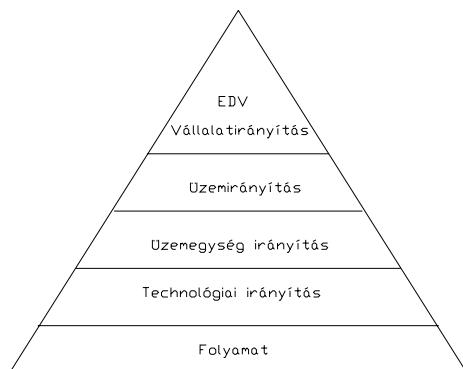
A diszkrét technológiai folyamatok időben ismétlődő részfolyamatokból állnak, s irányításukban a vezérlési műveletek a meghatározóak (pl. gépipar).

Az összetett, komplex folyamatokban általában a vegyes jelleg dominál. Amennyiben az irányításukban a szabályozási és vezérlési funkciók időben is tagozódnak, akkor szakaszos (batch) technológiáról beszélünk.

## 2.1. Az irányítások hierarchikus rendszere

Az egyes iparágak irányítástechnikai igényei, a szabályozási, vezérlési és felügyeleti funkciók aránya, az alkalmazott technológia veszélyességi szintje, a készülékek funkcionális és térbeli elhelyezkedése alapján eltérhetnek egymástól, de ugyanazon iparágon belül általában nagy hasonlóságot mutatnak.

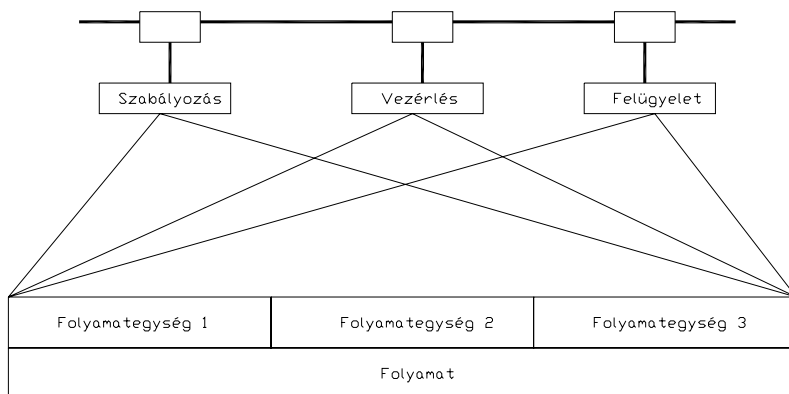
Egy vállalatban belüli irányítórendszereket a kommunikációs kapcsolatok – egymás mellé- és fölrendelésekkel – egy piramisszerűen ábrázolható hierarchikus rendszerré fogják össze. A folyamatközeli szinteket technológiai irányításnak, míg az üzem- és vállalatirányítási szinteket termelésirányításnak szokás nevezni.



2.2. ábra. Hierarchikus irányítás piramisa

Egy helyesen felépített, strukturált rendszer egyes alrendszerei különböző feladatokat láthatnak el. Hatáskörük kiterjedhet akár az egész folyamat minden működésére is, de lehet, hogy tartalékként állnak és egy esetlegesen meghibásodott egység szerepét veszik át egy időre.

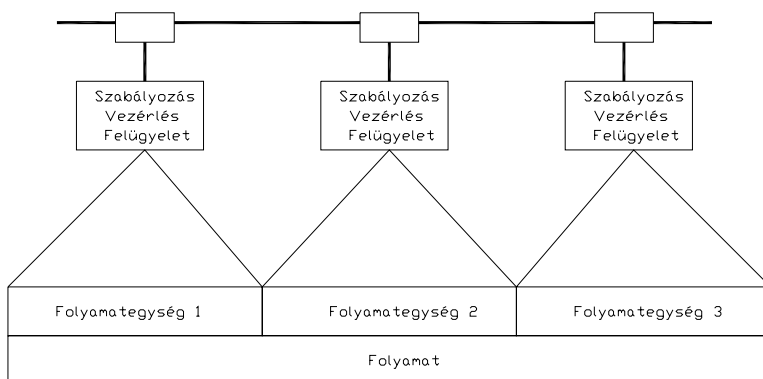
Az irányítandó folyamat az analízis során a térbeli egybetartozás szerint is részekre osztható. A térbelileg (pl. egy helyiségben vagy üzembrészben illetve egy technológiai alrendszerben) összetartozóan elhelyezkedő elemek együttesét egy folyamategységnek lehet tekinteni. Az irányítórendszer részei ezek után a hatáskörük kiterjedése és viselt funkciójuk alapján osztályozható.



2.3. ábra. Funkcionálisan osztott irányítási rendszer

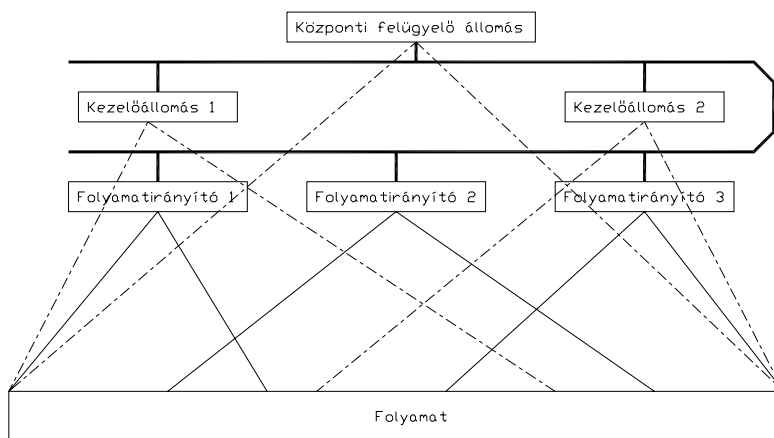
Ha az irányítási hierarchia alsó szintjén álló készülékek hatásköre a folyamat egészére kiterjed, de a három fő funkció – szabályozás, vezérlés, felügyelet – közül csak egyiket látják el, akkor az irányítási rendszer funkcionálisan osztott, ahogy az alábbi ábra ezt mutatja.

Ha az irányítási hierarchia alsó szintjén álló készülékek hatásköre csak egy folyamategységre terjed ki, de arra nézve minden feladatot ellát, akkor az irányítási rendszer térbelileg osztott.



2.4. ábra. Térbelileg osztott irányítási rendszer

Egy nagy folyamat komplex irányítási rendszerében a térbeli és funkcionális elosztás általában összetetten jelentkezik, hiszen az egymásra következő hierarchikus szintek is különböző módon kapcsolódhatnak egymáshoz.



2.5. ábra. Összetett irányítási struktúra

## 2.2. Az irányítórendszerek fejlődése

Az irányítástechnikai feladatok megoldása során különböző generációs rendszerek alakultak ki a technika fejlődése során, melyek mindegyike valamilyen jellemző tulajdonsággal rendelkezik. Az irányítórendszerek egy lehetséges csoportosítását az alábbi táblázat mutatja:

Fix huzalozású rendszerek	Típus	Programozható rendszerek	Típus
Jelfogós logika	D	Digitális számítógép	D/A
Digitális elektronikus logika	D	Mikrovezérlő	D/A
(Elektro)pneumatikus logika	D	PLC	D/A
(Elektro)hidraulikus logika	D	PLC-SCADA	D/A
Analóg elektronika	A	DCS	D/A

### 2.1. táblázat. Irányítórendszerek csoportosítása

A táblázatban szereplő rövidítések jelentése:

PLC: **P**rogrammable **L**ogic **C**ontroller, Programozható logikai vezérlő

SCADA: **S**upervisory **C**ontrol and **D**ata **A**cquisition, **Felügyeleti irányítás és adatgyűjtés**

PLC-SCADA: olyan rendszer, ahol a folyamatjeleket PLC-k vagy egyéb intelligens egységek kezelik és illesztik, az ember – gép kapcsolatot viszont PC vagy munkaállomás biztosítja és ezeket az eszközöket valamilyen ipari lokális hálózat köti össze, vagy **röviden PLC-s központi adatgyűjtés és folyamatvizualizálás**

DCS: Distributed Control System, osztott intelligenciájú folyamatirányító rendszer

D: digitális

A: analóg

### 2.2.1. Jelfogós vezérlések

A kétállapotú jeleket feldolgozó vezérléstechnikában korábban szinte kizárólagosan a jelfogós kapcsolásokat használták. A jelfogók a jelek fogadására, elosztására, tárolására, feldolgozására, jelek kiadására, galvanikus szétválasztásra képesek. Mindezek mellett a jelfogós vezérlések alkalmazása számos hátránnyal is jár. Egyrészt minden változtatás nehézkesen végezhető el rajtuk, másrészt nehezen integrálhatóak elektronikus rendszerekbe.

A jelfogós vezérléseket áramutas rajzokkal, pl. ún. létradiagrammal lehet ábrázolni, amely szimbólumai segítségével a működés szemléletesen követhető. Ezen személeteség az egyik oka annak, hogy ezt az ábrázolási módot a programozható logikai vezérlőknél is használják.

### 2.2.2. Félvezető elemekre épülő vezérlések

A nagyobb megbízhatóságra és kedvezőbb megvalósíthatóságra való törekvés vezetett az érintkezőmentes elemek alkalmazásához. Ezek a félvezető alapú dióda, a tranzisztor és az integrált áramkör. Az ilyen vezérlések előnye a viszonylag nagyfokú integráltság és ebből következően a kis teljesítmény- és helyigény.

### 2.2.3. Számítógépes alapú vezérlések

A személyi számítógépek megjelenésekor került előtérbe a számítógépek folyamatirányításra történő felhasználásának gondolata. Mivel a számítógépek rendelkeznek kétállapotú jelek fogadására, ill. kiadására alkalmas egységekkel, digitális vezérlések és/vagy szabályozások megvalósítására ideálisak. Ugyanakkor a jel fogadó- és kiadóegységek szintjei nem illeszkednek az ipari szintekhez, és az ipari hatásokkal szemben sem ellenállóak. A kezdeti időszakban a számítógépek ára is igen magas volt, így nem lehet csodálkozni azon, hogy megjelentek a speciálisan folyamatirányításra kifejlesztett mikroszámítógépek, a PLC-k, és nagyrészt ki is szorították a számítógépeket. Manapság az egészen bonyolult, nagy adathalmazmal operáló, nagy sebességigényű folyamatvezérléseknél használnak speciális folyamatirányító számítógépeket.

Az egyes rendszertípusok néhány lényeges szempont szerinti összevetése látható a 2.2. táblázatban.

Az összehasonlításból kitűnik, hogy egyedül a működési sebesség és a zavarérzékenység tekintetében nem a legkedvezőbb a PLC-s rendszer az ipari célú irányítórendszerek között. Az összetett műveletek végzése tekintetében a hagyományos számítógép talán előnyösebbnek tűnik egy megfelelő egységekkel ugyan jól felszerelt PLC-nél, ez azonban csak az adott funkció létrehozásánál érvényes, az alkalmazásnál már nem. A PLC-k ugyanis sokkal hatékonyabban működnek, hiszen a megfelelő adatokat átadva egy megfelelő modulnak, az elvégzi a kívánt vezérlési és/vagy szabályozási feladatot függetlenül a főprocesszor működésétől (többprocesszoros rendszer). Ezt a hátrányt ezenkívül a PLC-be dugaszolható PC kártya is kiküszöböli (PC a PLC-ben) vagy a PLC olyan kialakítása, amely közvetlenül egy ipari kivitellő PC-be dugaszolható (PLC a PC-ben).

Jellemző	Jelfogós rendszer	Fix huzalozású elektronikus rendszer	Számítógép	PLC
Ár/funkció	Közepes	Alacsony	Magas	Alacsony
Méret	Nagy	Kicsi	Közepes	Kicsi
Működési sebesség	Lassú	Nagyon gyors	Közepesen gyors	Gyors
Zavarérzékenység	Kiváló	Jó	Közepes	Jó
Üzembe helyezés	Időigényes tervezés és szerelés	Időigényes tervezés	Programozás időigényes	Egyszerű programozás és szerelés
Összetett műveletek végzése	Nem lehetséges	Lehetséges	Lehetséges	Lehetséges
Változtatás a működésben	Nagyon nehéz	Nehéz	Viszonylag egyszerű	Nagyon egyszerű
Karbantartás, javítás	Nehéz a sok érintkező miatt	Nehéz, ha az IC-k be vannak forrasztva	Nehéz a sokféle kártyatípus miatt	Egyszerű a tipizált kártyák miatt

**2.2. táblázat.** Különböző rendszertípusok összehasonlítása néhány szempont alapján

Az irányítási rendszerek más szempontok alapján is osztályozhatók. A műszaki fejlődés és fejlesztés révén különböző generációk alakultak ki. A rendszerek generációkba sorolásának főbb szempontjai a következők lehetnek:

- a rendszer térbeli tagoltsága,
- a kommunikáció egységessége és formája,
- a készüléktechnológiai tagoltság vagy integráltság mértéke,
- a részrendszerek együttműködésének lehetősége,
- a kezelőhely fejlettsége.

Ezen feltételek alapján az irányítási rendszerek öt generációját lehet megkülönböztetni Telkes Zoltán meghatározása szerint [1]. Az egyes rendszergenerációk megjelenése időben természetesen egymás után következett és a bonyolultságuk is növekedett. Ebből azonban nem következik, hogy a következő generáció teljesen kiszorította az előzőt, hiszen az egyszerűbb feladatokra az alacsonyabb generációjú rendszer is alkalmas

#### **2.2.4. Első generációs irányítási rendszerek**

Az első generációs irányítási rendszerek az irányítástechnikai elemek és készülékek megjelenésekor alakultak ki. Első generációs rendszertechnikájuk az iparban a helyi szint-, nyomás- és hőmérséklet-szabályozások, de ilyen pl. a központi fűtésű helyiségek egyedi hőmérséklet-szabályozása is.

Az első generációs irányítási rendszer legfontosabb tulajdonságai:

- térbelileg nem tagolt, központilag nem áttekinthető, autonóm, helyi, egyedi, kizárólag folyamatközeli, gyakran egybeépített irányításokból tevődik össze;
- nincs benne egységes készülék és jel;
- az elemek és készülékek nagyrészt segédenergia nélkül vagy pneumatikus, esetleg villamos segédenergiával működnek.

#### **2.2.5. Második generációs irányítási rendszerek**

A második generációs irányítási rendszerek jellemzői:

- térbelileg tagolt, központilag megfigyelhető, autonóm, egymással kapcsolatban nem lévő irányításokból tevődnek össze;
- egységes készülékek és egységes jelek, a szabályozási körben analóg (pl. 0...20mA), vezérlés esetén főként kétállapotú, pneumatikus vagy villamos jelek vannak;
- a rendszerelemek nagyrészt villamos vagy pneumatikus segédenergiával működnek;
- az egységes jelek és méretek miatt a rendszerelemek csereszabatosak;
- moduláris elven épülnek fel, így a különböző gyártmányok illeszkednek a rendszerbe;
- megjelenik a központi információkezelés és a mért értékek regisztrálásának igénye.

### 2.2.6. Harmadik generációs irányítási rendszerek

A harmadik generációs irányítási rendszerek jellemzői:

- Térbelileg tagolt, központilag megfigyelhető, autonóm, egymással kapcsolatban nem lévő, de központi irányítóberendezés (digitális számítógép) segítségével, speciális kommunikációval koordinálható szabályozási körökből, illetve vezérlőkből épülnek össze.
- Egységes készülékek (modulok) és főként egységes analóg pneumatikus vagy villamos jelek vannak (pl.: 4..20mA).
- A készülékek főként villamos (esetleg pneumatikus) segédenergiával működnek és egy részük alkalmas a központi irányítóberendezéssel való kommunikációra.
- Az egységes jelek és rendszerint az egységes modulok következtében az elemek csereszabatosak.
- Építőköckaelven épülnek fel, a különböző gyártmányok illeszthetők a rendszerbe.
- Megjelenik a központi adatgyűjtés, az ember–gép kapcsolatot megvalósító kezelőpult, illetve sématabla.
- Biztonsági célt szolgál a kézi/automata átkapcsolóval váltható háttéreszközök (szabályozó, vezérlő) megléte.

Ide sorolható minden olyan irányítás, amelynek egyedi szabályozási és vezérlési körei vannak, de ezek központilag koordinálhatók a különböző típusú és felépítésű számítógéppel.

A digitális (intelligens) kompakt szabályozók és az egyedi PLC-k is lehetnek e rendszer vagy generáció részei.

### 2.2.7. Negyedik generációs irányítási rendszerek

Az előző rendszertechnikai kialakításban egyetlen centralizált számítógép biztosítja az irányítási, adatfeldolgozási és megjelenítési feladatokat, ami a következő hátrányokkal jár:

- A számítógép meghibásodása a teljes technológia esetén a tartalékirányításra való áttérést igényel. A tartalékirányítás egyedi eszközi megvalósítása költségigényes, és az esetek többségében nem is biztosítja az elvárt mutatókat.
- A számítógépnek sokféle (irányítási, adatfeldolgozási, megjelenítési) feladatot kell ellátnia, így a programrendszer túlságosan szerteágazó, nehezen tipizálható. Általában egy számítógép hardverteljesítménye csak kompromisszumok árán elegendő a sokféle feladat ellátásához.



- A technológiából származó és a technológia felé juttatandó jelek esetenként nagy távolságra kell továbbítani, így a kábelezési költség igen tekintélyes.

Alapvetően a felsorolt okok miatt alakították ki a decentralizált folyamatirányító rendszereket, ahol a felmerülő feladatokat egymással kommunikációs kapcsolatban lévő számítógépek oldják meg, amelyek valamilyen logikai hierarchiában vannak szervezve. Egy-egy számítógép feladata viszonylag jól körülhatárolható.

A negyedik generációs irányítási rendszerek jellemzői:

- Térbelileg tagolt, a központi irányítóteremben elhelyezkedő része egységes, integrált irányítóberendezés.
- Digitális működésű és a belső kommunikáció is digitális, a folyamatközeli készülékek analóg működésűek.
- Egyedi digitális készülékek és digitális belső busz (ami a központi feldolgozóegység(ek) párhuzamos sínrendszerét jelenti, pl. VME, MULTIBUS), a központi irányítóberendezés és a folyamatközeli elemek (folyamatperifériák) között analóg (pneumatikus vagy villamos) egységes jelek vannak.
- A központi készülékek digitális elvűek, a folyamatperifériák villamos (esetleg pneumatikus) segédenergiával működnek és egy részük alkalmas a központi irányítóberendezéssel való kommunikációra.
- A központi rész egyedi rendszer, más rendszerrel általában nem kombinálható, a folyamatközeli elemek csereszabatosak.
- A rendszer ezen része építőköckaelven épül fel, az egységes analóg jelek és rendszerint az egységes méretű készülékek következtében a különböző gyártmányok illeszkednek a rendszerbe.
- Külön sajátosság a nagy megbízhatóság, a valós idejű adatfeldolgozás (az osztott erőforrás és adatbázis-kezelés, valamint a redundáns hardverek és szoftverek miatt).

### 2.2.8. Ötödik generációs irányítási rendszerek

Az ötödik generációs rendszerek kialakulását az intelligens (SMART) távadók, valamint az intelligens beavatkozók és a terepi buszok megjelenése váltotta ki. Ezek az eszközök egy – a nagy hálózatoktól (pl. ETHERNET) eltérő – kommunikáció csatornán a feladat jellegéhez igazodó protokoll szerint kezelhetők. A kommunikációs vonalra (szokásos szóhasználat szerint fieldbusra) korlátozott számú intelligens eszköz fűzhető fel. Ez a

technika további kábelezési költségcsökkentést jelent, még a technológia közelébe telepítet (pl. kihelyezett I/O) eszközökhöz képest is.

Az intelligens eszközök mellett általánossá vált, hogy a folyamatirányító rendszerek szerves részét képezik a PLC-berendezések még a DCS-rendszerek többségében is. Ezek ugyancsak egy kommunikációs vonalon kapcsolódnak az irányítórendszer folyamatállomásaihoz. A PLC-k kommunikációs vonalai és protokolljai (pl. MODBUS, MODBUS PLUS, SINEC-L, SINEC-H, PROFIBUS) különböznek a számítógépek között szokásos kommunikációs protokolltól, de különböznek az intelligens eszközök protokolljaitól is.

Az ötödik generációs irányítórendszer tulajdonságai:

- Térbelileg tagolt, a központi irányítóteremben elhelyezkedő része egységes, de funkciója átalakult és mérete csökkent. Az integrált központi irányítóberendezés digitális működésű, a belső és a terepi kommunikáció is digitális.
- Egyedi digitális készülékek és digitális belső busz (sín), a központi irányítóberendezés és a folyamatközeli elemek (folyamatperifériák) között digitális egységes terepbuszjelek vannak.
- A készülékek csak villamos segédenergiával működnek, digitális elvűek, jeleik digitálisak és a jelilleszkedésen túl alkalmasak a központi irányítóberendezéssel való kétirányú kommunikációra.
- A központi rész egyre kevésbé egyedi, inkább csereszabatos ipari számítógép.
- A folyamatközeli elemek csereszabatosak, a rendszernek ez a része moduláris felépítésű.
- Az esetenként egységes digitális jelek miatt a különböző gyártmányok egy része illeszkedik a rendszerbe.
- Kommunikációs lehetőséget biztosít az internetre, így bizonyos funkciói teljes mértékben nyitottá válnak a világ bármely pontjáról.
- Az érzékelő- és beavatkozószervek egyre intelligensebbek ezért adatfeldolgozási műveletek végrehajtására és hálózati kommunikációra alkalmasak.
- Az intelligens távadók és beavatkozók, valamint a terepi buszok révén az adatfeldolgozás osztott jellege kiteljesedik, hiszen pl. a szabályozási funkció az intelligens beavatkozón realizálódik, tehát a korábbi terepi, ill. központi funkció megszűnik.

## 3. Programozható vezérlők

A programozható vezérlők az 1970-es évektől kezdődően terjedtek el és ma csaknem kizárólagos alkalmazást nyertek az ipari folyamatok vezérlésében. A programozható vezérlőberendezések, a **vezérlési (esetleg szabályozási) funkciókat** szoftver útján valósítják meg és beviteli, kiviteli egységeik révén a technológiai folyamatok **tárolt programú vezérlésére** közvetlenül alkalmasak.

A programozható vezérlők szokásos elnevezései:

PC: Programmable Controller (programozható vezérlő, UK)

PLC: Programable Logic Controller (programozható logikai vezérlő, USA)

PBS: Programmable Binary System (programozható bináris rendszer, svéd)

SPS: Speicherprogrammierbare Steuerung (tárolt programú vezérlés, német)

PV: Programozható Vezérlő (magyar)

PLV: Programozható Logikai Vezérlő (magyar).

Mivel szakmai körökben leginkább a PLC elnevezés terjedt el, ezért a továbbiakban ez a jegyzet is ezt a rövidítést használja.

### 3.1. Egy kis PLC-történelem

1968-ban az amerikai General Motors cég pályázatot hirdetett a régi, meglévő vezérlőinek kiváltására abból a célból, hogy olyan programozható vezérlőberendezést fejlesszenek ki, **amely ötvözi a relés, a félvezetős és a számítógépes vezérlés előnyeit.**

A pályázati kiírásban az alábbi szempontok szerepeltek:

- egyszerű, moduláris felépítés, kis méret;
- mozgó alkatrészt ne tartalmazzon;
- galvanikusan leválasztott bemeneti/kimeneti fokozatok (24 Vdc-től 240 Vac)
- könnyű programozhatóság és újraprogramozás;
- valós idejű működés max. 0,1 s válaszidővel;
- nagy megbízhatóság, minimális karbantartás;
- versenyképes ár.

A pályázatra a **Modicon**, ill. az **Allen-Bradley** cégek pályáztak, amelyek ma is vezető cégek a PLC-k piacán. A PLC-k fejlődésének főbb állomásait a kezdeti időszakban az alábbiakban foglaljuk össze.

- 1968. A PLC-konceptió kidolgozása a General Motors felhívására
- 1969. Az első Modicon PLC megjelenése huzalozott CPU-val, 1K memóriával és 128 I/O-val
- 1971. A PLC első alkalmazása az autóiparban
- 1973. Az első intelligens (smart) PLC megjelenése aritmetikai funkcióval, nyomtatóvezérléssel, mátrixműveletekkel, képernyőkijelzéssel
- 1974. Az első többprocesszoros PLC gyártása időzítő- és számlálófunkcióval, 12K memóriával és 1024 I/O-val
- 1975. Az első PID algoritmussal ellátott PLC kibocsátása
- 1976. A távoli modulkezelés (remote control) kidolgozása és a hierarchikus konfiguráció bevezetése az integrált gyártórendszerben
- 1977. A mikroprocesszor bázisú PLC bevezetése
- 1980. Intelligens kommunikációs modulok kifejlesztése, valamint a nagy sebességű, nagy pontosságú pozicionáló interfész kifejlesztése
- 1981. A Data Highway kommunikáció alkalmazása, 16 bites mikroprocesszor bázisú PLC színes monitorral
- 1983. Olcsó „mini” PLC-k megjelenése
- 1985. PLC hálózatok kifejlesztése

Természetesen a fejlődés nem állt meg 1985-ben, azóta is töretlen s napjainkban a gyártók a felhasználók igényeinek legmegfelelőbb készülékeket kínálják egyre kisebb méretben, egyre több intelligenciával s általában nem növekvő árakon.

A mai PLC-ket kivitelük alapján **kompakt és moduláris** felépítésű csoportba sorolhatjuk.

A kompakt PLC jellemzője, hogy **hardverstruktúrája nem módosítható**, kizárólag megfelelő védettségű ipari tokozásban készül és kis helyigényű. Felhasználási területei a sorozatban gyártott gépek, berendezések automatikái, illetve a PLC műszaki jellemzői által lefedhető egyedi vezérlések. **A kompakt PLC-k egy különleges típusát** jelentik az ún. **mikro-PLC-k**, amelyek az ember-gép kapcsolat hardver- és szoftverfeltételét is tartalmazzák.

Az utóbbi időben néhány ismert, PLC-ket gyártó cég kompakt kivitelű, egyszerűen programozható **logikai kapcsoló/vezérlőmodulokat fejlesztett ki (mikro-PLC-k)**. Ezek a készülékek **kisebb vezérlési feladatok** megoldására alkalmasak, tartalmaznak időzítő és számláló funkciókat is. Progra-

mozásuk a készülék nyomógombjaival vagy PC-vel lehetséges. Ismert készüléktípusok:

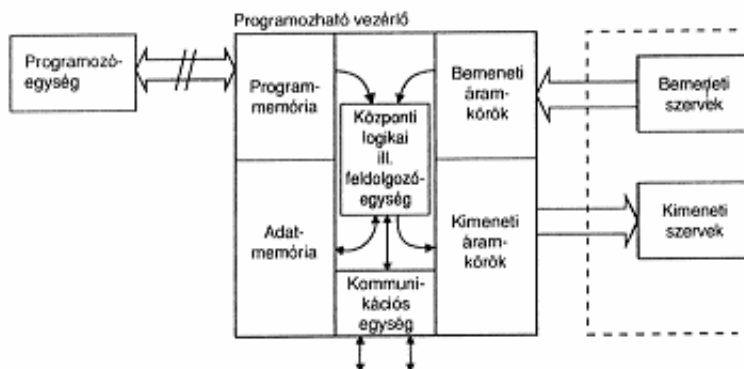
- LOGO (Siemens)
- ZELIO Logic (Telemecanique)
- EASY (Moeller)
- ZEN (Omron)

A moduláris felépítésű programozható logikai vezérlők jellemzője, hogy a vezérlőberendezés valamely speciális funkciót önmagában ellátó modulokból épül fel. A modulok fizikai megjelenése rendszerint az áramköri kártya, dugaszolható csatlakozóval. A modulok ún. rack-be (tartókeret) dugaszolhatók, ezért a rendszer konfigurációja tág határokon belül bővíthető. A rack-ek megfelelő védettségű műszerdobozba vagy műszerszekrénybe szerelhetők. A moduláris felépítésű PLC-ket közepes, ill. nagyméretű rugalmas gyártórendszerek vagy ipari folyamatok irányítására fejlesztették ki.

### 3.2. A PLC funkcionális egységei

A következőkben sorra vesszük az általános felhasználási célú PLC-k funkcionális elemeit. Moduláris kialakítás esetén ezek a funkcionális egységek általában külön modulokban helyezkednek el, míg kompakt kivitelnél akár az egész PLC egy közös egységben található. Egy PLC rendszerben az alábbi egységeket találhatjuk meg:

- Központi logikai és feldolgozó egység (CPU)
- Tápegység
- Bemeneti és kimeneti egységek (I/O, digitális illetve analóg)
- Intelligens egységek
- Kommunikációs egységek



3.1. ábra. A PLC funkcionális felépítése

### 3.2.1. A központi feldolgozó egység (Central Processing Unit, CPU)

A központi feldolgozó egység a PLC „agya”. Futtatja a felhasználói programot s vezérli a további egységeket. A felhasználói program vagy RAM-ban, vagy EPROMban van tárolva. A program fejlesztése személyi számítógépen történik, és a kész programot (már a CPU processzorának gépi kódjában) viszik át a későbbi tárolóeszközbe. Egyes PLC-k speciális, előlapi programozási lehetőséggel is rendelkeznek. A központi vezérlőegység címezi a kimeneti és a bemeneti egységeket, parancsokat ad a rendszerben lévő intelligens feldolgozóegységeknek.

### 3.2.2. A tápegység

A tápegység feladata, hogy a rendszert megfelelő feszültséggel ellássa, a hálózati feszültséget a PLC számára átalakítsa és stabilizálja. A legtöbb esetben külön telepeket is tartalmaz, hogy feszültség-kimaradás esetén a RAM tartalma megőrizhető legyen.

### 3.2.3. Bemeneti és kimeneti egységek

A PLC-k egyes be- és kimeneti pontjai szinte minden esetben galvanikusan le vannak választva a belső buszról, illetve a CPU egységtől.

#### Digitális bemeneti egységek

A digitális bemeneti egységek feladata olyan jelek értelmezése, illetve illesztése a PLC belső szintjéhez, melyek csak két lehetséges állapotot vehetnek fel. Az iparban használatos feszültségekhez igazodva megtalálható-

ak a választékban egyen- ill. váltakozófeszültséget érzékelő egységek is. A bemeneti egységek feszültségtartománya is széles skálán mozog, a 24 V-os névleges feszültségűtől a 230 V-os névleges feszültségűig.

### Digitális kimeneti egységek

A digitális kimeneti egységek feladata a PLC belső jeleinek átalakítása a környezet számára. Alapvetően kétféle változat található:

**Relés kimenetű:** itt a CPU egy relét vezérel, amelynek az érintkezője van kivezetve. Előnye, hogy segítségével olcsón lehet nagy áramokat kapcsolni, és az áramkörökbe bárhova beilleszthető (ha az egyik pontja nincs pl. földelve). Hátránya, hogy a megvalósítható kapcsolási frekvencia kicsi, kisebb, mint 10 Hz, és a reakcióideje is nagy.

**Elektronikus kimenetű:** itt valamilyen vezérelt félvezető elem kapcsolja a kimeneti feszültséget. Előnye, hogy viszonylag gyors kapcsolásra képes (100 Hz körül) és rövid a reakcióideje. Hátránya, hogy csak megadott helyre illeszthető, pl. az egyik pontot földelni kell, illetve nagy áramokat csak viszonylag drága elemekkel lehet kapcsolni.

### Analóg bemeneti egységek

Az analóg bemeneti egységek A/D átalakítók segítségével alakítják át (konvertálják) digitális kóddá a bemenetre kapcsolt analóg jelet. Az ipari jeltartományokhoz illeszkedve a bemeneti feszültség vagy áramtartomány több lépcsőben változtatható.

### Analóg kimeneti egységek

A PLC a felhasználói program végrehajtása során számolt digitális értéket alakítja át D/A konverter segítségével analóg jellé.

### Kommunikációs egységek

Ezen egységek segítségével valósítható meg a kapcsolattartás más eszközökkel, pl. folyamatirányító számítógéppel vagy nyomtatóval. Szabványos soros vagy párhuzamos interfészt biztosítanak, vagy hálózati kommunikációt tesznek lehetővé.

### Intelligens egységek

Ezen egységek valamilyen nagy sebességű előfeldolgozást végeznek a bemenő jeleken, nagymértékben tehermentesítve a központi feldolgozó egységet. Ilyen előfeldolgozás lehet pl. számlálás, pozíciófigyelés, hőmérsékletszabályozás, tengelyvezérlés stb. A legkülönbözőbb ipari célokra ké-

szülnek intelligens egységek. Az alábbi felsorolásban a gyakrabban előforduló intelligens periféria egységek közül említünk néhányat:

- Tengelyvezérlő egység – fordulatszám szabályozás, útmérés, pozicionálás
- Léptetőmotor vezérlő egység – fordulatszám vezérlés, útmérés, pozicionálás
- Számláló (pozicionáló) digitális időzítő egység – programozható időzítés, impulzusszámlálás, pozicionálás
- Hőmérséklet szabályzó egység – hőmérsékletmérés, analóg hőmérsékletszabályozás
- Adagoló szelepvezérlő egység – áramlásmérés, durva és finom adagolás vezérlés
- Proporcionális és szervó szelepvezérlő egység – digitális vagy analóg követőjel képzés, követő szabályzás
- Univerzális analóg szabályzó egység – analóg szabályzás digitális paraméter megadással
- Univerzális digitális vezérlő egység – impulzusszámlálás, komparálás, kimenet vezérlés
- Hajtásszabályzó egység – egyenáramú motorok fordulatszám-szabályozása
- Be/kimeneti processzor – a központi egységet tehermentesítő gyors periféria vezérlés
- Állásos és impulzus kimenetű szabályzó egység – digitális vezérlésű végrehajtó szervek vezérlése
- Mérlegillesztő egység – automatikus mérlegelés, adagolás
- Kódolvásó egység – vonal vagy induktív kód beolvasása
- Real time Óra – naplózás, időzítési funkciók
- In rack számítógép – adaptív szabályozás, grafikus jelfeldolgozás, archiválás, stb.
- Hálózatfigyelő egység – tápellátási, energiagazdálkodási feladatok

A programozható vezérlőkre vonatkozó IEC-1131-1 szabvány szerint a vezérlő az alábbi funkciók ellátására képes:

- jel/adat feldolgozási funkció (signal/data processing);
- technológiai interfészfunkció az érzékelők kezelésére, ill. beavatkozók működtetésére;



- kommunikációs funkciók (PLC-PLC; PLC-számítógép; PLC-hálózat);
- ember-gép interfészfunkció (man-machine interface, MMI);
- programozási, tesztelési, dokumentálási funkció;
- tápellátási funkció.

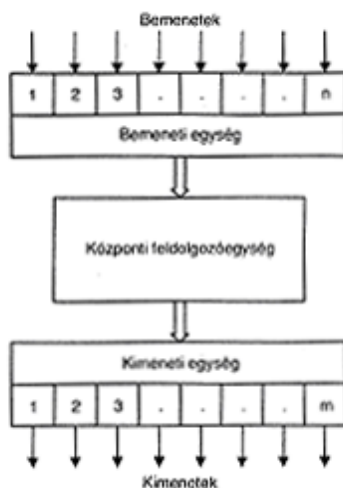
A fejlődés során a programozható vezérlők funkciói nagymértékben közeledtek a számítógép funkcióihoz. Így mára a PLC olyan ipari számítógépnek tekinthető, amely speciális hardveregységei és felhasználói programja révén a technológiai folyamatok tárolt programú vezérlésére, szabályozására és intelligens kommunikációs felülete révén hierarchikus és/vagy osztott folyamatirányító rendszerek létrehozására alkalmas.

A programozható vezérlők előnyei: a szabad strukturálhatóság, a gyakorlatilag végtelen kapcsolási szám, a telepítési költségek csökkenése, a rendszerbe szervezhetőség lehetősége. A szabad strukturálhatóság felhasználói programozhatóságot jelent, amelynek révén a felhasználó tárolt, egyedi program révén az univerzális hardvert a feladatra alkalmassá teszi. A PLC-k alkalmazásával a telepítési, beüzemelési idő nagymértékben lerövidíthető.

### 3.3. A PLC-k szerkezeti felépítése

A programozható vezérlők hardvere univerzális. Fő rendeltetése a vezérlési program végrehajtása, amihez az adatok beolvasására, feldolgozására és az eredmény kivitelére van szükség. Ezt a három műveletet az alábbi hardveregységek végzik: bemeneti egység, központi feldolgozó egység és kimeneti egység. A felsoroltak közül a központi feldolgozóegység fejlődött a legdinamikusabban, és főként ez határozza meg a PLC szolgáltatásait.

Az első PLC-k központi feldolgozóegysége még kis- és/vagy közepes mértékben integrált digitális áramkörökből (Small Scale Integration, SSI; Medium Scale Integration, MSI) épült fel. Az integrálási technológia fejlődésével a huzalozott logikájú központi egységeket felváltották az ún. bit-processzorok (szokásos elnevezések még: Boole-processzor, logikai processzor), amelyek funkcionálisan nem, csak méretbeli és áramköri jellemzők szempontjából jelentettek előnyt az előzőekhez képest. Ezek alkották a programozható vezérlők első generációját.

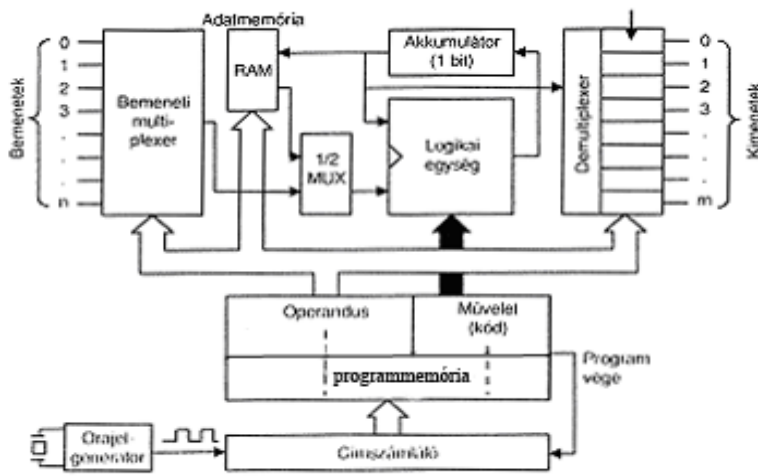


**3.2. ábra.** A programozható vezérlők három fő egysége

A nagymértékben integrált áramkörök (Very/Large Scale Integration, V/LSI) elterjedésével az általános célú mikroprocesszorok váltak a PLC-k központi feldolgozóegységévé, ami egyben minőségi változást is jelentett. A bájtt-, ill. szóprocesszorok alkalmazása révén a Boole-műveletek mellett a következő funkciók váltak általánossá a programozható vezérlőkben: aritmetikai műveletek végzése, szabályozási funkció ellátása, szabványos kommunikáció biztosítása stb. A mikroprocesszor alapú programozható vezérlők, amelyek napjainkra szinte egyeduralmukodóvá váltak, tekinthetők a PLC-k második generációjának.

A bitprocesszor alapú programozható vezérlők jellemzői:

- csak egybites, Boole-jellegű logikai műveletek végzése;
- kevés számú utasítás;
- kisméretű memóriakezelési lehetőség;
- mikroprogramozott vezérlési architektúra;
- névkódon alapuló (esetleg gépi kódú) programozás;
- alacsony szintű pont-pont jellegű kommunikáció.



3.3. ábra. Egy bitprocesszor alapú PLC felépítése

Az egyes egységek funkciói:

- Bemeneti multiplexer: a bemeneti logikai változók kiválasztása és az adat kapuzása a programmemóriában tárolt bemeneti cím alapján.
- Logikai egység: bemenetére jutó bitek között a programmemóriában tárolt mikrokód által meghatározott logikai művelet végzése.
- Akkumulátor: egy bites operandus- és eredményregiszter.
- Kimeneti demultiplexer és tároló: a LU által végrehajtott logikai művelet eredményének (1 bit) kijuttatása a programmemória által meghatározott kimenetre és az adat tárolása.
- Adatmemória: a logikai műveletek részeredményeinek tárolása.

A bitprocesszor alapú PLC külső elemei és azok funkciói:

- Programmemória: a vezérlési algoritmust realizáló program tárolása.
- Programszámláló: a programmemória egymás utáni címkombinációinak előállítására az óragenerátorról kapott impulzusok hatására.

### 3.4. A bitszervezésű PLC működése

A programmemóriában tárolt operandusmező közvetlenül címezi a bemeneti, kimeneti, ill. RAM-elemeket, a műveleti kód pedig a programozható logikai egység által végzendő műveletet definiálja. A logikai egység mindig az akkumulátor és a másik operandus (bemenet, vagy RAM bit) között

hozza létre az előírt logikai műveletet, például a kétváltozós ÉS kapcsolatot. A programmemóriában tárolt programot a PLC az órajel-generátor által működtetett számláló révén, növekményes sorrendben egymás után ciklusosan hajtja végre. A kimenetre juttatott eredményeket a flip-flopok tárolják két feldolgozási ciklus között.

A programozható vezérlőkben kétféle funkciójú memóriát alkalmaznak: program- és adatmemóriát.

A programmemória a vezérlőprogramot tartalmazza, amely a vezérlési algoritmust realizálja.

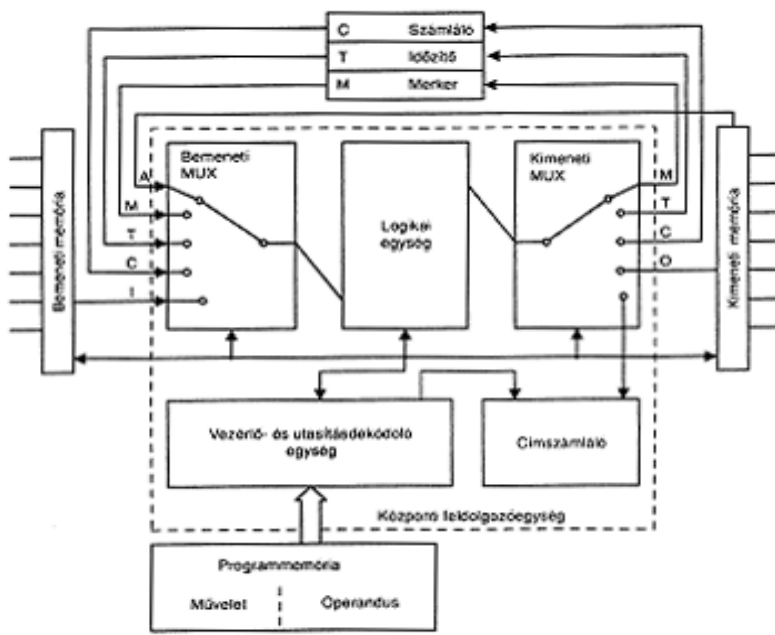
Programmemóriaként korábban egyszer programozható, csak olvasható memóriát használtak (Programmable Read-Only Memory, PROM), míg napjainkban törölhető, programozható, csak olvasható (Erasable PROM, EPROM) memóriát alkalmaznak. Utóbbi előnye, hogy a program módosítása a memóriacsip kivétele nélkül közvetlenül megoldható (in system programming). Főként a felhasználói programok belövésekor jól használhatók programmemóriaként a teleppel védett CMOS RAM-ok (Complementary Metal-Oxid Semiconductor, CMOS).

A programfutás közben keletkező változók tárolására írható-olvasható memóriák (RAM) szolgálnak, amelyek a tápfeszültség bekapcsolásakor teljes védelem nélkül tetszőleges értéket (0, ill. 1) vehetnek fel.

A RAM-memóriák a következő célorientált funkciókat látják el a programozható vezérlőkben:

- a közbenső adattárolók funkciója hasonló a hagyományos vezérlés segédreléihez. Ezek a tárolók valósítják meg az ún. MERKER funkciókat (a programban M vagy F betűvel jelölik). Erre van szükség a diszjunktív alakú függvényekben szereplő VAGY kapcsolatok részeredményeinek tárolásánál vagy a sorrendi hálózatokban a lefutó vezérlés belső állapotainak 1 az n-ből jellegű állapot kódolására. A MERKER memória bitprocesszor alapú PLC-k esetén bitszervezésű;
- az I/O RAM funkció a bemeneti, ill. kimeneti változók tárolását jelenti, szintén bitszervezésű;
- az időzítő (timer) és számlálók (counter) értékének átmeneti tárolása (bájt vagy szó jellegű).

Napjainkban szinte kizárólag teleppel védett CMOS RAM csipeket használnak RAM-memóriaként. Egy tipikus bitprocesszor alapú PLC felépítését szemlélteti az alábbi ábra.



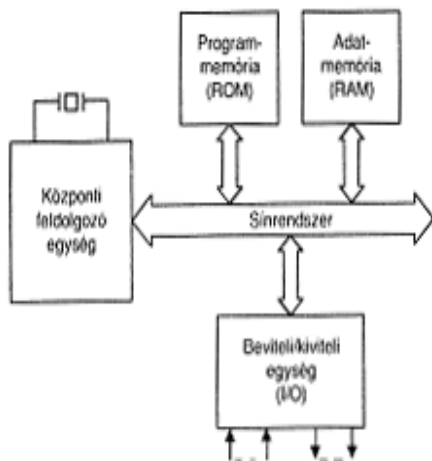
3.4. ábra. Bitszervezésű PLC felépítése

Látható, hogy a logikai egység operandusai lehetnek: egy bemenet (I), egy kimenet (O), egy merkerbit (M), egy időzítőkimenet (T) vagy számlálókimenet (C). A bemeneti változót kivéve valamennyi változó a logikai egységgel, a program futtatásával módosítható.

A programozható vezérlők szolgáltatásai az általános célú mikroprocesszorok (bájt- vagy szóprocesszorok) alkalmazásával minőségileg megváltoztak. A szóprocesszorok felépítése nagymértékben hasonlít a bitprocesszorokéhoz.

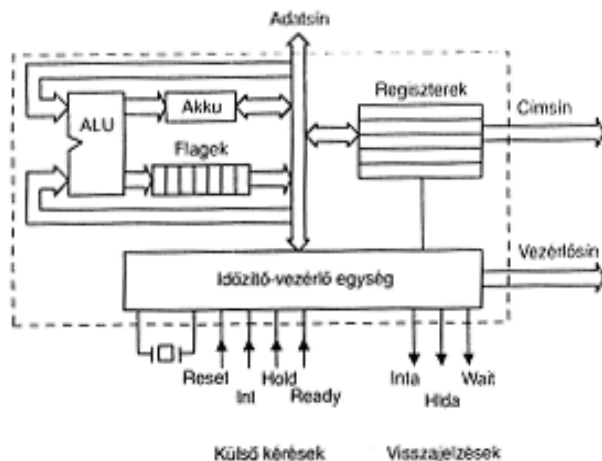
Az első mikroprocesszorok a 70-es évek elején jelentek meg. A mikroprocesszorok a digitális számítógépek központi feldolgozóegység funkcióinak ellátására alkalmasak, nagymértékben integrált áramkörök, egyetlen lapkán kialakítva. Az első mikroszámítógép-rendszer már a Neumann-féle modell valamennyi elemét tartalmazta:

- központi feldolgozóegység (Central Processor Unit, CPU);
- memóriák (RAM, ROM);
- beviteli/kiviteli egység (Input/Output, I/O);
- sínrendszer.



3.5. ábra. A mikroszámítógép felépítése

A mikroprocesszor a számítógép funkcióit ellátó digitális, igen nagy mértékű integráltságú áramkör (Very Large Scale Integration, VLSI), amelynek három fő része van: időzítő-vezérlő egység, aritmetikai-logikai egység (Arithmetical and Logical Unit, ALU) és regiszterek:



3.6. ábra. A mikroprocesszor fő egységei

### Az egyes részegységek funkciói:

Az időzítő-vezérlő egység feladata a program utasításai vagy a külső kérések (megszakítás, tartás, várakozás) alapján a gép részeinek irányítása. Ez egyrészt az ALU műveleteinek vezérlését, az egyes adatútvonalak nyitását, zárását, a sínek működtetését, másrészt a külső egységek: a memória és az I/O egységek vezérlését jelenti.

A vezérlőegység fontos része az utasításregiszter, amely a programmemóriából az utasításlelívási ciklusban beolvasott utasítás kódját tárolja, amíg az utasításdekódoló és értelmező logika meghatározza az elvégzendő műveletet és elindítja a végrehajtást vezérlő mikroprogramot.

A vezérlőegység fontos funkciója a különböző aszinkron jellegű kérések (program-megszakítás, tartáskérés, várakozáskérés) fogadása és az ezekhez tartozó vezérlés.

A mikroprocesszor másik fontos egysége az aritmetikai-logikai egység (ALU), amely az utasításokban meghatározott aritmetikai és logikai műveleteket hajtja végre.

Az ALU-hoz szorosan hozzátartozik az akkumulátorregiszter, ill. az állapotregiszter (flagregiszter). A processzorok általában csak néhány aritmetikai műveletet (összeadás, kivonás, szorzás) képesek elvégezni. Ezért a korszerű mikroprocesszorokhoz ma már nélkülözhetetlenül hozzárendelnek egy aritmetikai társprocesszort (co-processzor). A mikroprocesszorok speciális és általános célú regisztereket tartalmaznak.

A regiszterek a tápfeszültség bekapcsolásakor véletlenszerű értéket vehetnek fel, ezért van szükség bekapcsoláskor ezek inicializálására. A CPU fontos részét képezi a sínrendszer. Ezen a belső egységek, valamint a külső egységek (memória, I/O) közötti adatforgalom bonyolódik. A sínrendszer funkcionálisan háromféle sít foglал magában: adatsín (data bus), címsín (address bus) és vezérlősín (control bus).

A sínrendszer funkciói, jellemzése:

- Címsín: Megoldja az adatforgalomban részt vevő eszközök kijelölését. Egyirányú, háromállapotú, a processzortól függően 16/32 bit szélességű (azaz ennyi vezeték), amely meghatározza a címezhető memória és I/O tartományt.
- Adatsín: Biztosítja az adatok útját. Kétirányú, háromállapotú, a processzortól függően 8/16/32 bit szélességű. Az adatsínvezérlés meghatározza az adatátvitel irányát.

- Vezérlősín: Összehangolja a kapcsolatban részt vevő eszközök működését. Egyirányú, háromállapotú, a processzortól függően 5-15 bit szélességű (azaz ennyi vezeték).

A legegyszerűbb vezérlősín ötbités:

- Memóriaolvasás (Memory Read, MR);
- Memóriaírás (Memory Write, MW);
- Beviteli/kiviteli eszköz írása (Input/Output Write, I/OW);
- Megszakítás (interrupt).

A vezérlősín révén lehet az azonos címen lévő memória-, illetve I/O műveleteket megkülönböztetni. A külső sínrendszer lehet helyi sín (local bus), amely a processzorhoz közvetlenül kapcsolódik, ill. lehet rendszer-sín (system bus), amely a processzor sínmeghajtásán keresztül kapcsolódik a rendszer elemeihez. A sínrendszer használatának előnye, hogy a szabványosított jel és vezetékkiosztás miatt az egyes részegységek könnyen cserélhetők. A rendszer bemenetei hardverjelleggel terhelik a sínrendszert, ezért kell bizonyos számú modul esetén sínmeghajtást használni. A mikroprocesszoros rendszerekben használatos sín párhuzamos sínnek tekinthető.

A CPU működése ciklikus: utasításlehívás, végrehajtás, lehívás, végrehajtás stb. Ezt a pontos sorrendiséget a rendszeróra vezérli. A CPU működésében a legegységesebb időegység a gépi állapot, amely rendszerint egy órajel periódusa alatt játszódik le. Egy gépi állapothoz egy jól definiált művelet tartozik: pl. a címinformáció kijuttatása a címsínre. Általában több gépi állapot alkot egy gépi ciklust, ami egy összetettebb műveletet jelent.

Tipikus gépi ciklusok: egy memóriarekesz olvasása (MR), ill. írása (MW) vagy (I/O) eszköz írása, ill. olvasása (I/OW, I/OR), utasításlehívás (fetching) stb. Egy utasítás lehívásának és végrehajtásának együttes művelete az utasításciklus, amely 1...8 gépi ciklusból állhat az utasítás bonyolultságától függően. Általában egy utasításciklus annyi gépi ciklusból áll, ahányszor a CPU-nak a memóriához vagy I/O-hoz kell fordulnia. Minden utasításciklus utasításlehívási gépi ciklussal kezdődik (fetch), a további gépi ciklusok olvasási vagy írási jellegűek.



## A processzor állapotai:

Egy processzor működése rendszerint a következő állapotokból áll:

- futó (run) állapot, amikor a processzor a programmemória által meghatározott utasításokat egymás után hajtja végre;
- várakozó (wait) állapot, amely a gépi cikluson belül valósul meg;
- tartás (hold) állapot, amely gépi ciklusok között aktualizálható;
- leállítás (halt) állapot, amikor egy HALT állapot utasítás hatására a processzor leáll, nem végez műveletet és ezen állapotból csak engedélyezett megszakítás hatására lép ki.

A mikroszámítógép beviteli, ill. kiviteli elemei a központi feldolgozó egység és a külvilág (ember, gép, technológia, számítógép) közötti kapcsolat kialakításának lehetőségét biztosítják. A beviteli, ill. kiviteli elemek típustól függően párhuzamos, ill. soros kommunikációra alkalmasak.

Közös jellemzőik: biztosítják a be/ki elem csatlakoztatását a mikroszámítógép buszrendszeréhez a szükséges adat-, cím- és vezérlővezetékkel; rendszerint programozható felépítésük miatt igen rugalmasan alkalmazkodnak a csatlakoztatandó eszközökhöz.

A be-/kiviteli eszközök beállítását (inicializálását) a rendszer bekapcsolása után a programból kell letölteni. Néhány be/kiviteli áramkör: programozható párhuzamos perifériaillesztő egység (Programmable Peripheral Interface, PPI); programozható soros illesztőegység (Universal Synchronous Receiver Transmitter, USART); programozható időzítő- és számlálóegység (COUNTER/TIMER).

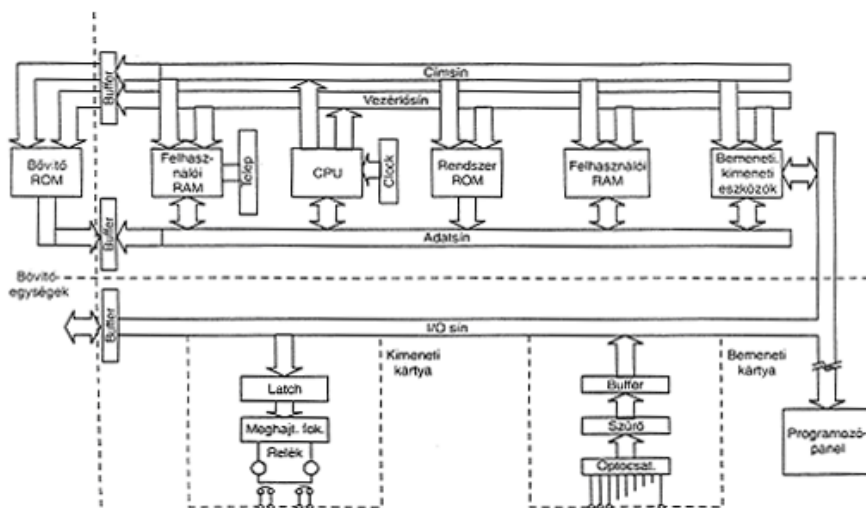
## 3.5. Mikroprocesszor alapú PLC-k hardverfelépítése

A mikroprocesszor bázisú PLC központi egysége 8, 16, ill. 32 bites általános célú processzor vagy mikrovezérlő (mikrokontroller) egyaránt lehet. Ehhez szükség van a mikroszámítógép szokásos elemeire (CPU, RAM, ROM), valamint a külvilággal való kapcsolattartás eszközeire. A bemeneti, ill. a kimeneti vonalak kezelésére négyféle módszer terjedt el:

- a bemeneti/kimeneti eszközök a processzor párhuzamos perifériaillesztőin keresztül kapcsolódnak a cím-, adat- és vezérlőszínre;
- a bemeneti/kimeneti vonalak kezelésére egy külön I/O sítint állítanak elő kifejezetten az I/O kezelésére, tekintettel a moduláris felépítés be/ki vonalainak nagy számára, a terhelés viszonyaira stb.;

- távoli I/O kezelés
- terepi, soros jellegű buszrendszer szervezésű I/O kezelés.

Az első megoldást főként kompakt PLC-khez használják, ahol a kevés be/ki vonal miatt a külön I/O sín kialakítása nem indokolt. Egy tipikus, mikroprocesszor alapú PLC hardverét mutatja az alábbi ábra:



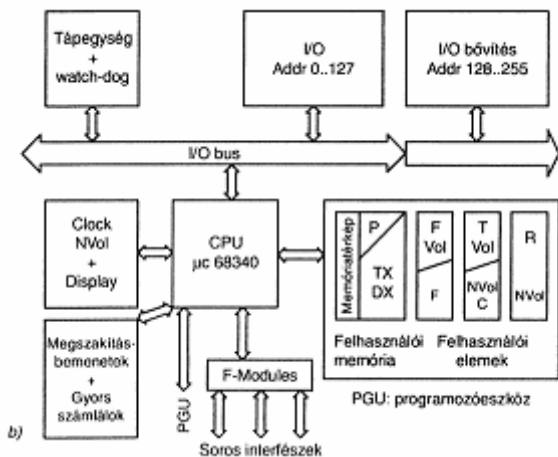
**3.7. ábra.** Egy mikroprocesszor bázisú PLC általános felépítése

Látható, hogy a be/ki vonalak kezelésére egy külön I/O sín használatos. Az ábra egyes blokkjai korábban egy-egy fizikai egységet alkottak (egy-egy kártya), ma viszont az integrálási technológia fejlődésével elérhető, hogy a CPU, RAM, ROM, I/O meghajtót egyetlen kártyán helyezik el, miáltal a CPU buszrendszere előnyösebben és biztonságosabban alakítható ki.

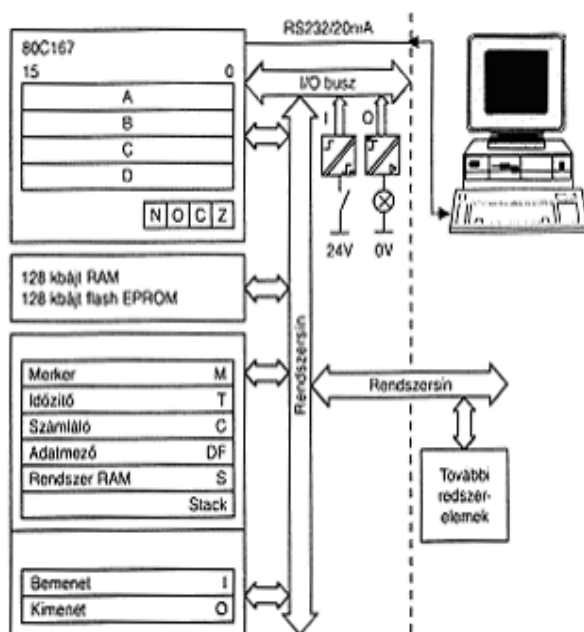
Az általános célú mikroprocesszor alapú PLC-k hardverét szemlélteti a 3.8. ábra.

Az ábrából jól kivehetők a mikroszámítógép-rendszer fő elemei: CPU, memóriák, flegék, időzítők, számlálók, megszakításkezelő egység, valós idejű óra, be/ki elemek stb. A be/ki vonalak kezelését itt is a külön I/O BUS-szal oldották meg.

A moduláris felépítésű programozható vezérlőre példaként egy 80C167 típusú mikroprocesszorral gyártott típus hardverét mutatja az alábbi ábra, amelyen a négy 16 bites általános célú regiszter és a négy jelzőbit (N előjel, O túlcordulás, C végső átvitel, Z zérus) is található, amelyek a programozó által elérhetők.



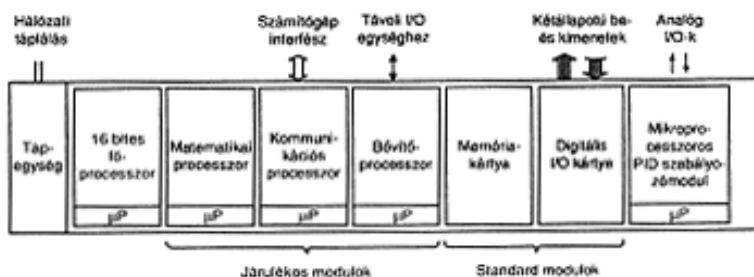
3.8. ábra. Egy 68340 típusú mikroprocesszoros kompakt PLC hardver felépítése



3.9. ábra. A 80C167 típusú mikroprocesszorral gyártott PLC hardver felépítése

A rendszer alapfigurációja 128 kb-ot RAM-ot és 128 kb-ot Flash EPROM-ot tartalmaz. A RAM az alábbi funkciók ellátására szolgál: merker (M), időzítés (T), számlálók (C), adatmező (DB), rendszer-RAM funkció, stack-RAM funkció. A be/ki egységek itt is külön I/O buszra csatlakoznak. A moduláris PLC rack-jének hátlapján kialakított rendszersín (SYSTEM BUS) további hardveregységek csatlakoztatására szolgál. A központi egység három megszakításkérés kezelését biztosítja. A példabeli PLC a be-, ill. a kimeneteket I/O RAM egységen keresztül kezeli.

A nagyméretű és bonyolult rendszerek irányításához rendszerint több processzort alkalmaznak, amelyek egy-egy speciális funkciót látnak el. Egy ilyen többprocesszoros PLC felépítését szemlélteti a következő ábra, ahol a főprocesszor (Main Processor) 16 bites szóprocesszor.

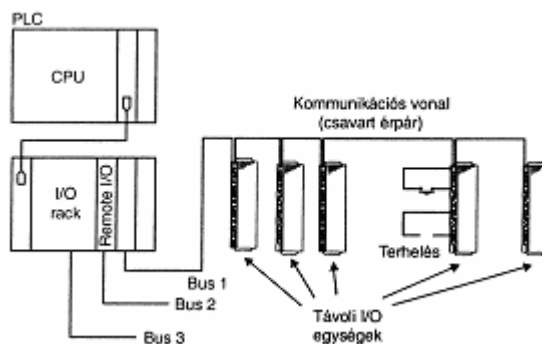


**3.10. ábra.** A többprocesszoros PLC felépítése

Matematikai műveleteket a matematikai processzor, a kommunikációs funkciókat a kommunikációs processzor vezérli. Emellett a nagyszámú távoli I/O kezelést és a PID szabályozási algoritmust is külön processzor végzi. Ezek a processzorok rendszerint mester-szolga (master-slave) kapcsolatban állnak a főprocesszorral. A mester-szolga rendszerű kommunikáció esetén a szolgaprocesszorok csak a mesterrel állnak kapcsolatban, egymással nem. Napjainkban a decentralizált irányítási módszerek kerülnek előtérbe.

A programozható vezérlők speciális kiegészítő egységei a távoli be/ki (I/O) modulok. A távoli I/O kezelés (Remote I/O, RIO) a nagyszámú be/ki vonal esetén indokolt, különösen akkor, ha az I/O eszközök távol vannak a PLC-től. Ilyenkor a központi PLC-rack rendszerint tartalmaz egy távolsági I/O rack-et (remote I/O rack), amely soros kommunikációval

kezeli a távoli be/ki vonalakat. A távoli I/O modulokkal ellátott PLC felépítését szemlélteti az alábbi ábra.



3.11. ábra. Távoli be/ki vonalak kezelése (RIO)

### 3.5.1. Digitális bemeneti és kimeneti egységek

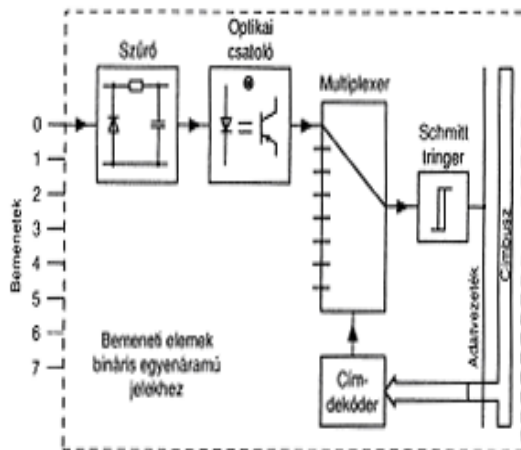
A PLC-k központi műveletvégző egysége kétfokozatú beviteli/kiviteli egységen keresztül tartja a kapcsolatot a külvilággal, azaz a technológiai folyamat jeladó ill. beavatkozószerveivel. A CPU-val rendszerint egy belső be/ki egység (multiplexer vagy I/O port) révén, a sínrendszeren keresztül történik az adatforgalom. A belső I/O egységek tápfeszültsége megegyezik a CPU tápfeszültségével.

A belső I/O egységek bemeneteire kapcsolódhatnak:

- a PLC-hez kapcsolódó kétállapotú eszközök, amelyek táplálása közvetlenül a CPU tápfeszültségéről történik, ezért galvanikus leválasztást nem igényelnek (pl. kezelőszervek, billentyűzet);
- a technológiáról érkező terepi jellegű eszközök, amelyekre jellemző, hogy a táplálásuk nem egyezik meg a CPU táplálásával. Amíg a CPU központi processzora rendszerint 5-15 V, addig a működtetőszervek ettől rendszerint eltérő táplálást igényelnek (pl. 24Vdc, 48 Vdc, ill. 24 Vac, 230 Vac). A különböző feszültségek keveredésének megakadályozása céljából galvanikus leválasztású bemeneti, ill. kimeneti illesztőket használnak. A galvanikus leválasztást optocsatolóval oldják meg. Ennek főleg zavarvédelmi, villamos és biztonságtechnikai (érintésvédelmi) előnye van. A bitprocesszor, ill. szóprocesszor alapú PLC-k bemeneti, ill. kimeneti egységei csak a szervezés módjában térnek el. A bit-

processzor a be- ill. kimeneteket bitenként, a szóprocesszor bájtonként/szavanként kezeli.

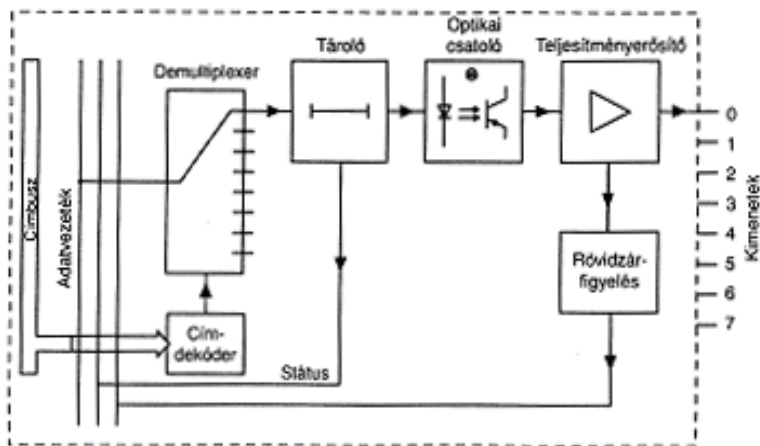
Bitszervezésű kétfokozatú bemeneti egységet szemléltet az alábbi ábra.



**3.12. ábra.** Kétfokozatú bitszervezésű beviteli egység felépítése

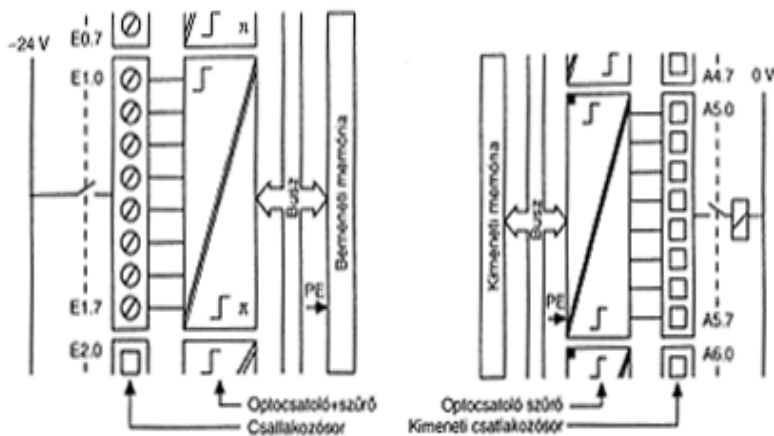
A kétállapotú kapcsolók jele a zavarójeleket szűrőfokozaton át az optoleválasztású bemeneti fokozaton keresztül továbbítja a PLC által címezhető beviteli elemre, amely lehet port vagy multiplexer. Látható, hogy a bemeneti fokozat nem tartalmaz tárolóelemet. Tárolóelem akkor szükséges, ha fontos az azonos idejű mintavételezés. Ilyenkor egy parancs (utasítás) hatására a PLC összes bemenetének állapotát egy RAM memória tárolja. Ilyen bemeneti fokozatot használnak a gyorsan változó jelek feldolgozásakor.

Bitszervezésű kétfokozatú kimeneti egységet mutat a következő ábra, ahol az adat a CPU-tól a mikroprocesszor kiviteli pontján át egy tárolóelemre kerül (D flipflop, D latch). A kimeneti fokozathoz a tárolóelem elengedhetetlen, ez tárolja ugyanis a program által előírt és a CPU által létrehozott eredményt (1 bit) két mintavételezés között.



3.13. ábra. Bitszervezésű kétfokozatú kimeneti egység felépítése

A bájt szervezésű processzorral felépített PLC bemeneti és kimeneti egységek láthatók az alábbi ábrán.

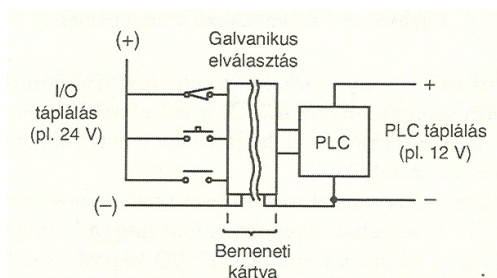


3.14. ábra. Bájt szervezésű bemeneti és kimeneti egységek

A fenti ábrán látható, hogy mind a bemeneteket, mind a kimeneteket az ún. I/O RAM-on keresztül kezelik. Az I/O RAM-ok írása a program végén PE (page-end character, lap vége) jellel ciklikusan történik.

A galvanikus elválasztás – azaz a működtető feszültségek közös vonatkoztatási pontjának szétválasztása – zavarvédelmi tulajdonságát a következők alapján érthetjük meg. A bemeneti, ill. kimeneti eszközök rendszerint a technológiai berendezések (motorok, mágneskapcsolók, szelepek stb.) közelében (a terepen) vannak elhelyezve, így induktív és kapacitív úton különböző zavarójelek szuperponálódnak ezen vezetésekre. Galvanikus elválasztás nélkül ezek a zavarójelek a CPU tápfeszültségére jutnának. A CPU-t egy kristályoszillátor működteti. Galvanikus elválasztás nélkül a zavaróimpulzusok katasztrofális hibát okozhatnak a CPU működésében (pl. akaratlan memóriairás, -olvasás).

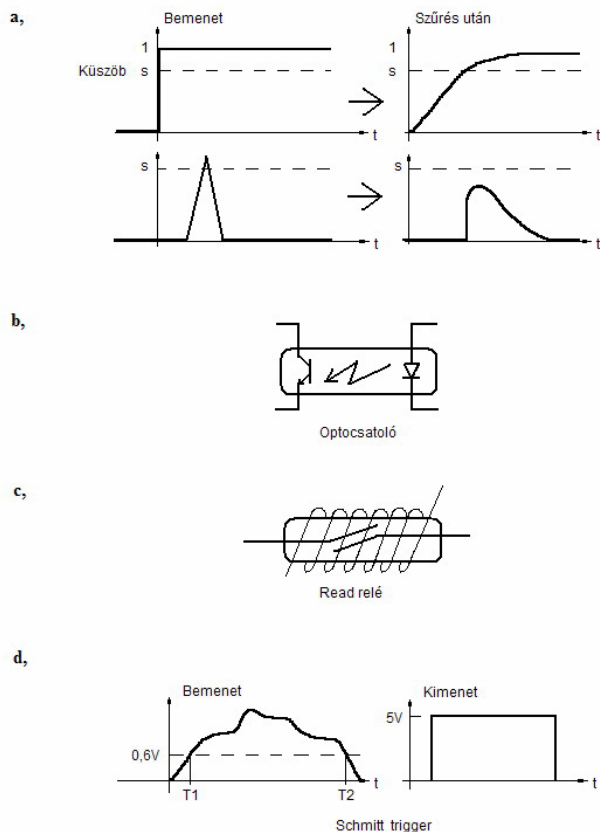
Az alábbi ábra szerinti szétválasztás a bemeneti és kimeneti oldalon védetté teszi a CPU működését a külső zavaroktól és véd az esetleges áramütéstől.



**3.15. ábra.** A galvanikus elválasztás elve

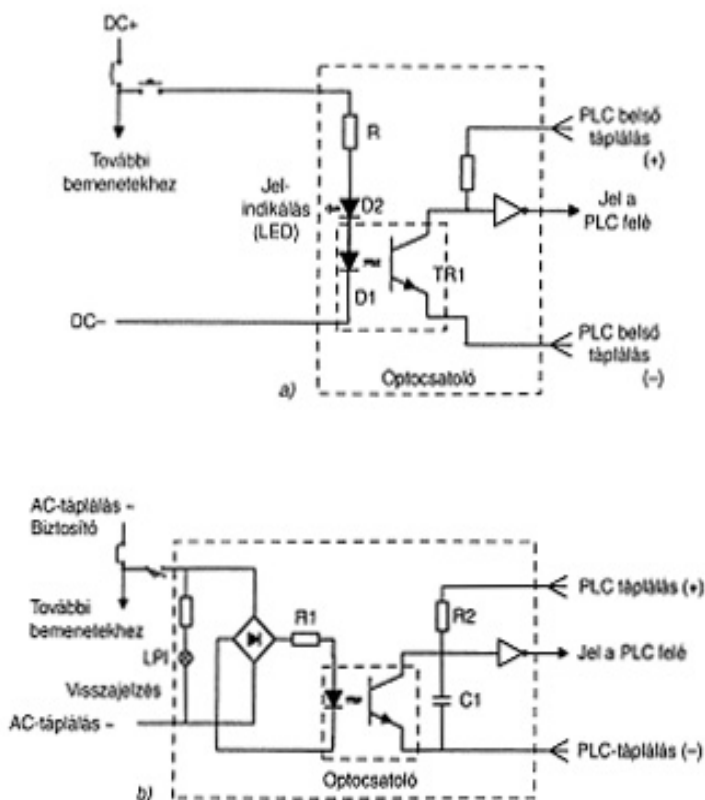
A digitális bemeneteken jelentkező problémák elvi megoldási lehetőségeit mutatja a következő ábra:





**3.16. ábra.** Elvi megoldások a digitális bemeneti problémákra

A szinte minden be-, kimeneti egységnél használt optocsatoló (optoizolátor) egy közös tokba épített fényemittáló diódából és egy fototranzisztorból áll. Optoelválasztású bemeneti fokozatot szemléltet a következő ábra.

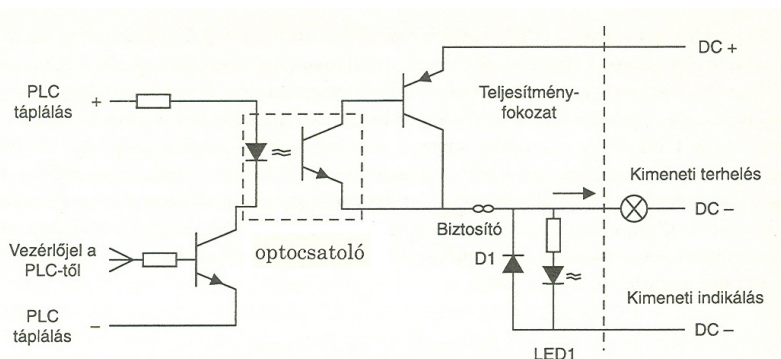


**3.17. ábra.** Egyenáramú (a) és váltakozóáramú (b) bemeneti fokozat

Az (a) ábra az egyenáramú galvanikus elválasztást mutatja. Az R ellenállás a tápfeszültség értékétől függ és a LED dióda áramának beállítására szolgál. Rendszerint 5 V, 12 V, 24 V, ill. 48 V tápfeszültséget használnak.

A (b) ábra váltakozó áramú jel galvanikus elválasztását mutatja. Az R1 a megfelelő áram beállítására szolgál, míg az R2, C1 a váltakozó komponens szűrését végzi (50 ms). A PLC bemeneti kártyájának előlapján a bemeneti állapot információját rendszerint LED-diódával visszajelzik.

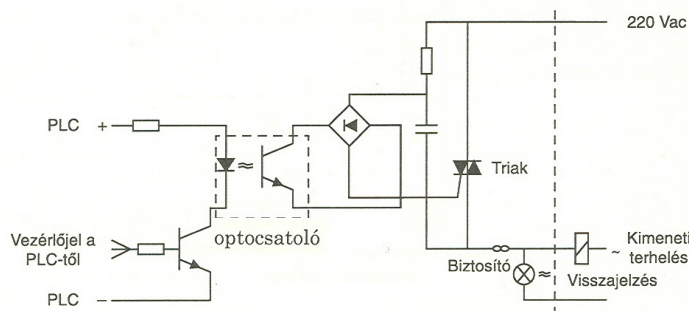
Galvanikus elválasztású tranzisztoros kimeneti fokozatot szemléltet az alábbi ábra.



**3.18. ábra.** Galvanikus elválasztású tranzisztoros kimeneti fokozat

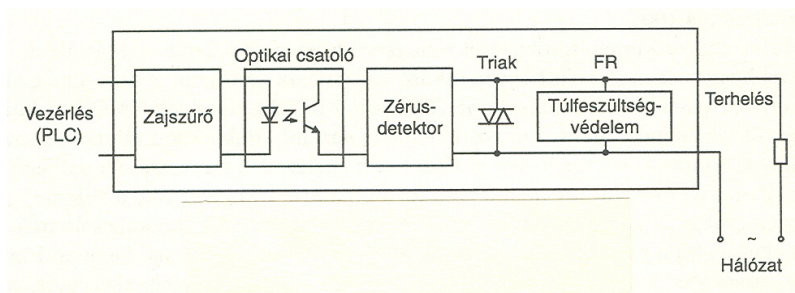
A megfelelő erősítéshez Darlington-fokozatot alkalmaznak. A D1 dióda az ellentétes áramimpulzus vágását végzi. A PLC I/O kártyán rendszerint 16 vagy 32 ilyen kimeneti egységet helyeznek el. A különböző kimeneti szervek villamos sajátosságait a felhasználónak figyelembe kell vennie, pl. izzólámpa esetén a hőmérséklettel változó ellenállás, induktív terhelés (relé, mágneskapcsoló) esetén a kikapcsoláskor fellépő feszültség-visszalökés.

Váltakozó áramú galvanikus leválasztású kimeneti fokozatot szemléltet a következő ábra, amelyen kimeneti kapcsolóként triakot alkalmaznak. A triak a váltakozó feszültség „0” átmeneténél kapcsol ki, így csökkenti az induktív terhelés kikapcsolásánál fellépő nemkívánatos zavarójeleket.



**3.19. ábra.** Galvanikus elválasztású triakos kimeneti fokozat

A váltakozóáramú hálózatokban egyre elterjedtebben használják a félvezető relét (Solid State Relay, SSR). A félvezető relék (FR) egyfázisú kivételben készülnek. Blokksémászerű felépítésüket az alábbi ábra szemlélteti.



**3.20. ábra.** Félvezető relével megvalósított kimeneti fokozat

A vezérlőkör a zajszűrőkből és optikai csatolóból áll. A szűrőkör feladata annak megakadályozása, hogy a rövid idejű zavaróimpulzusok bekapcsolják az FR-t. A vezérlőjel (3...24 V közötti egyenfeszültségű jel) az optocsatolón át a zérusdetektorra jut, amely annak megjelenését követően a hálózati feszültség első nullátmeneténél kapcsolja be a triak vagy a szembekapcsolt tirisztor vezérlésével a terhelést. Az FR a hálózati feszültségre szuperonált túlfeszültség-impulzusok korlátozása céljából túlfeszültségvédelemmel is el van látva.

A közepes teljesítményű félvezető kapcsolóelemek közül régebben a bipoláris tranzisztort alkalmazták a leggyakrabban. Különösen előnyösek az egy tokba integrált kapcsolóerősítők.

A tranzisztoros kimeneti fokozatot rendszerint rövidzárvédelemmel látják el.

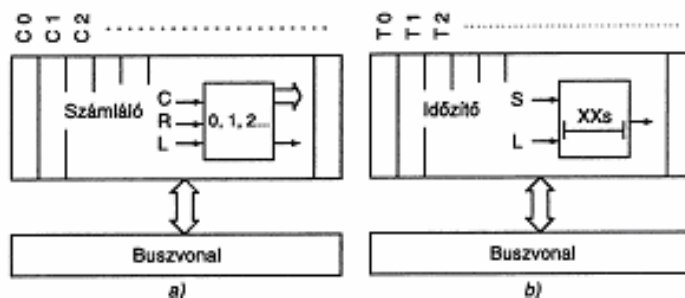
A bipoláris tranzisztorokat egyre inkább kiszorítják az utóbbi években megjelent térvezérlésű teljesítménytranzisztorok (POWER FET-ek), amelyek VMOS technikával készülnek és igen jelentős műszaki előnyei vannak. Bekapcsolt állapotban az ellenállásuk hőmérséklet-tényezője pozitív, ezért melegeedés hatására ellenállásuk nő és korlátozza az áramot, így nem léphet fel a bipoláris tranzisztorokra jellemző „termikus megfutás”. Ezek a tranzisztorok közvetlenül párhuzamosan kapcsolhatók. További előny a több nagyságrenddel nagyobb áramerősítés és a nagy bemeneti impedancia miatt igen kicsi (mikrowatt nagyságrendű) vezérlési teljesítményigény, ezért logikai jelekkel közvetlenül vezérelhetők.

### 3.5.2. Számlálók, időzítők

A számlálási funkció megvalósítására a PLC-k fejlődése során szoftver ill. hardver úton megvalósuló számlálók születtek. Ez a két megoldás a struk-

turális különbözőségeen túl a számlálható frekvencia maximális értékében is eltér a hardveres megoldás javára.

A szoftverúton történő számláló hardverfeltétele egy, a mikroproceszor által kezelt RAM-memória (esetleg regiszter). Ezt szemlélteti a következő ábra, amelyen egy-egy számlálót (C0, C1, C2...) a RAM-memória bájta vagy szava reprezentál. Az ábrán a számláló működtetéséhez szükséges funkciók is láthatók: számlálóbemenet (C), törlésbemenet (R), párhuzamos feltöltés (L).

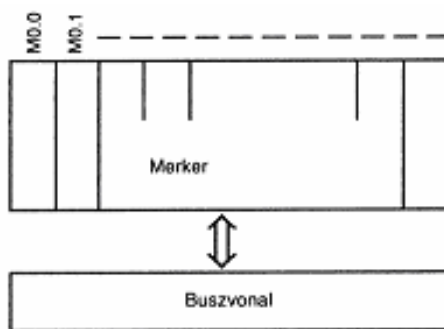


3.21. ábra. Számláló (a) és időzítő (b) funkciók a RAM-ban

Különböző megoldásokat dolgoztak ki az időzítési funkciókra is. Korábban a PLC kimenete által működtetett analóg/digitális időzítőket, hosszabb idejű késleltetéshez kapcsolóórákat alkalmaztak.

Napjainkban az időzítési funkciókat a PLC óragenerátor frekvenciájának szoftverúton történő osztására vezetik vissza. Ennek hardverfeltétele megegyezik a számlálóéval, azaz a CPU által kezelt és erre a célra fenntartott RAM memóriaterület a hozzá tartozó indítási (S), ill. a kezdetiérték-betöltési (L) funkció biztosításával (b ábra részlet).

A számlálási, ill. időzítési funkciók hardverfeltételéhez nagymértékben hasonlít a MERKER-memória kezelése is. Ezt szemlélteti az alábbi ábra.



3.22. ábra. MERKER-memória illesztése a buszvonalhoz

A fenti funkciókat rendszerint a RAM-memória részeként adják meg. Természetesen a RAM-térképek további funkciókat is tartalmaznak.

M0 ⋮ n	Merkerok
T, C0 ⋮ n	Időzítők (T) és számlálók (C)
	Egyéb funkciók

3.23. ábra. A PLC RAM térképe

## 4. A PLC-k programozása

A PLC-k hardvere univerzális, amely önmagában nem, csak a felhasználói programmal együtt válik konkrét irányítási feladatra alkalmassá. Ebből következik, hogy a programozható logikai vezérlők alkalmazásának egyik legfontosabb kérdése a felhasználói programok készítése. Már a General Motors által 1968-ban kiírt pályázatban szerepelt a felhasználóbarát, vezérléstechnika-orientált programozási nyelv.

### 4.1. A PLC-ben futó programok és feladataik

A korszerű PLC-k szoftvere a feladatuk alapján alapszoftverre és felhasználói programcsoportra osztható. Az alapszoftver az *állandó* (rezidens), a felhasználói program pedig a *változó* részt képviseli.

### 4.2. Alapszoftver

A PLC alapszoftverét – hasonlóan valamennyi mikroszámítógépes berendezéshez – az *operációs rendszer* biztosítja. A PLC alapszoftvere igen erősen gyártó-, ill. típusfüggő, így egyedi. Ennek ellenére megfogalmazhatók a következő közös funkciók, amelyek szinte valamennyi korszerű típusnál felfedezhetők.

Az *interpreter funkció* a felhasználói program értelmezésére és végrehajtására alkalmas szoftver. Az interpreter a kódolt felhasználói programot utasításonként veszi elő, értelmezi és végrehajtja.

A *státusszó-generálás funkció*, amely szinte valamennyi mikroszámítógépes berendezésben megtalálható. A státusszó-generálás célja a processzor műveleteiről történő információszolgáltatás.

Az *önteszt funkció* a PLC egyes funkcióinak ellenőrzését végzi, különösen a biztonsági PLC-k alkalmazásakor nagy jelentőségű.

A *kommunikációs vonalak* kezelése a soros pont-pont, ill. hálózati kommunikációs funkciók ellátása. Napjainkban e funkció jelentősége a PLC-hálózatok, terepi buszok szerepének növekedésével rohamosan nő.

Az *ember-gép kapcsolat* terén a PLC egyik alapvető funkciója a kezelő és a PLC közötti kommunikáció biztosítása.

A *programfejlesztési funkció* típustól függően lehet a PLC operációs rendszerének sajátossága, de lehet külön a fejlesztőrendszeré is. Ma már a programfejlesztési funkcióit egyre inkább a személyi számítógépek veszik át.

Az előbbiek illusztrálására példa a korábbiakban vázolt 80C167 processzor alapú PLC alapszoftverének felépítése, amely a Siemens által kifejlesztett STEP-5, ill. STEP-7 nyelv sajátossága.

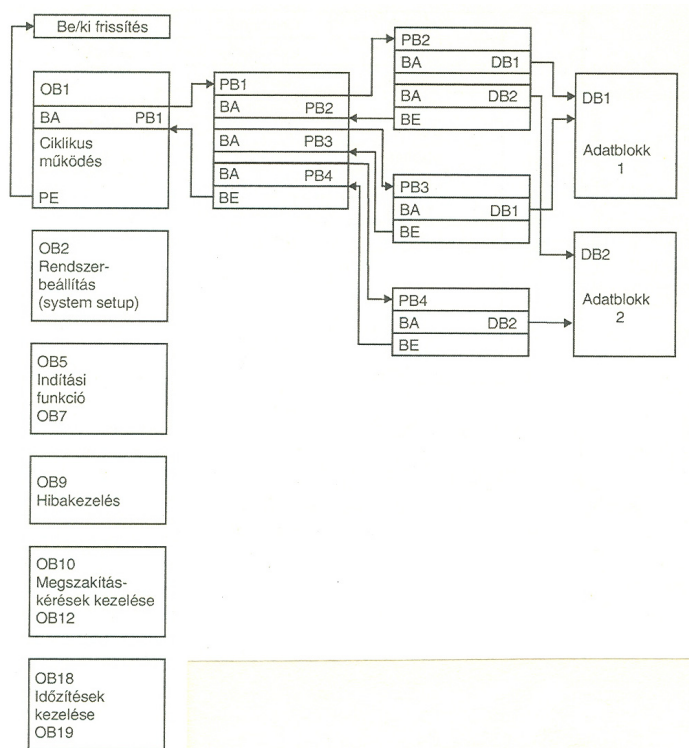
A PLC operációs rendszere 3 fő szoftver modul típust tartalmaz: szervezőblokkok (OB), programblokkok (PB) és adatblokkok (DB).

A Siemens rendszerű alapszoftver tíz szervező szoftverblokkból épül fel, amelyek biztosítják:

- OB1 a ciklus működését
- OB2 a rendszer beállítását (set up)
- OB5 és OB7 az újraindítási funkciókat
- OB9 a hibakezelést
- OB10, OB11, OB12 a három programmegszakítást
- OB18, OB19 az időzítések kezelését

Az OB1 szervezőblokk tartalmazza az *interpretert* és a felhasználói program végrehajtását biztosító *executive* részt. A felhasználói programok a PB (vagy FC) blokkokban vannak és korlátozott számú alprogramot (szubrutin), valamint két adatblokkot kezelnek. Ezt szemlélteti az alábbi ábra.





4.1. ábra. Egy PLC alapszoftverének a felépítése (STEP5)

A PLC ún. *hardvertesztel* indul (memóriateszt, telepteszt stb.), majd az OB2 rendszerbeállító (system setup) funkció révén az OB5 vagy az OB7 blokkon keresztül jut el az OB1 ciklikus üzemmódot biztosító szoftverblokkba. A be/ki memóriát az OB1 blokk végén a PE jel minden ciklus befejezésekor frissíti. A PLC-vel kapcsolatos hibakezelési funkciót az OB9 blokk látja el. A három megszakítási szintet az OB10...OB12 szoftverblokkok kezelik. A programmegszakítások egyike a soros kommunikációhoz van hozzárendelve.

### 4.3. Felhasználói programok

A fentiekben leírt alapszoftver a PLC-ben futó programok állandó része és minden azonos típusú programozható vezérlőben egyforma. Ezzel szemben a felhasználói programok a PLC programok változó részét jelentik és segítségükkel válik alkalmassá a PLC az adott vezérlési feladatra. A fel-

használói programok speciális, vezérléstechnikai, ill. irányítástechnikai orientáltságú programnyelven íródnak.

A felhasználói programokkal kapcsolatos, hogy a bitprocesszor alapú PLC-k esetén a hardverstruktúra és a program felépítése között igen szoros a kapcsolat, ezért interpreterre nem volt szükség. Az ilyen PLC utasítások programmemóriabeli elhelyezkedését szemlélteti az alábbi táblázat.

1	Utasítás	Operandus / cím	<b>1. logikai egyenlet</b>
2	Utasítás	Operandus / cím	
3	Utasítás	Operandus / cím	
4	Utasítás	Operandus / cím	<b>2. logikai egyenlet</b>
5	Utasítás	Operandus / cím	
6	Utasítás	Operandus / cím	
7	Utasítás	Operandus / cím	
.	.	.	
.	.	.	
	Utasítás	Operandus / cím	<b>Utolsó logikai egyenlet</b>
	Utolsó utasítás	Kezdő cím	

#### 4.1. táblázat. Utasítások elhelyezkedése a programmemóriában bitszervezésű PLC esetén

Gyakran a mai nem bitszervezésű PLC-k felépítését is sematikus és a programozási nyelvhez illeszkedően úgy ábrázolják, mintha bitprocesszor alapú lenne.

A mai bájtt-, ill. szóprocesszor alapú PLC-kben a felhasználói program felépítését az interpreter határozza meg. A bájtt-, ill. szóprocesszor felépítésű programozható vezérlőkben a Boole jellegű műveletek végzése körülményesebb, ugyanis ezen processzorok 8, ill. 16 bites szavak között végeznek aritmetikai, logikai vagy adatmozgatási műveleteket.

Az interpreter tehát egy közbenső szoftvereszköz a vezérléstechnikai nyelv és a PLC processzora között. Valamennyi felhasználói programnyelv a vezérléstechnikai (irányítástechnikai) feladatnak az interpreter számára érthető formába szervezéséhez szükséges szabályok összefoglalása.

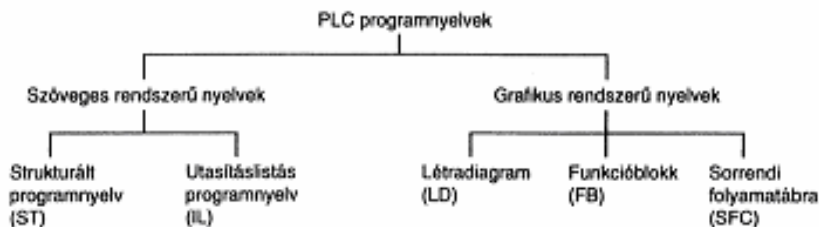
## 4.4. PLC programnyelvek

A PLC-k fejlődése során számos programozási nyelvet fejlesztettek ki. Ezek közös jellemzője az „ahány gyártó, annyiféle programnyelv” volt. Az IEC 1131-3 számú nemzetközi szabvány az egész világra egyesíteni kívánja a felhasználói programnyelveket és ezek jelöléseit. Ez a szabvány nem új programnyelveket hoz létre, hanem a korábbi, közös jellemvonású nyelveket igyekszik egyesíteni. Az IEC 1131 szabvány a PLC felhasználói programnyelveket két osztályba sorolja: szöveges rendszerű és grafikus szimbólumokat alkalmazó programnyelvek.

A szöveges rendszerű programnyelveknek két megvalósítási formáját engedélyezik. Az egyik a magas szintű programnyelvekkel (Pascal vagy C) támogatott strukturált felhasználói programnyelv, amelynek angol és német jelölése egyaránt ST (angol: Structured Text, német: Strukturierter Text). Ez esetben a vezérlési feladatot megvalósító felhasználói program leírása hasonlít a Pascal vagy C nyelven megírt program szintaktikájához. E módszer célja, hogy a magas szintű nyelvet ismerők képesek legyenek PLC program készítésére. Ennek ellenére ez a programozási nyelv a PLC technikában eddig nem terjedt el, de egyre népszerűbb.

A másik szöveges programnyelv az utasításlistás felhasználói programnyelv, amelynek jelölése angolul Instruction List, IL, németül Anweisungsliste, AWL. Ez a programnyelv az assembly nyelvű programozásból alakult ki, és a bitszervezésű PLC-knél erősen kötődött a hardverstruktúrához.

Az IEC 1131-3 szabvány háromféle grafikus szimbólumot alkalmaz: létradiagramos, funkcióblokkos és sorrendi folyamatábrán alapuló programnyelvet. A szabvány a következő PLC nyelveket definiálja és ajánlja:



#### 4.4.1. Strukturált programozási nyelv

A strukturált programozási nyelv (SI) egy magas szintű nyelv a vezérlési és szabályozási funkciók leírására az IF... THEN, CASE, FOR, WHILE, REPEAT stb. parancsok segítségével.

#### 4.4.2. Utasításlistás programozási nyelv

Az utasításlistás programozási nyelv (IL) a PLC kialakulásával egyidős és a bitszervezésű programozható vezérlők esetén az utasításkészlet nagymértékben függött az adott feldolgozóegységtől (processzortól). Az ilyen felépítésű PLC-kben egy-egy utasítás a memória egy címén helyezkedett el. Az utasítások száma széles határok között változott. Gyártottak 8, 16, ill. 32 utasítást értelmező vezérlőket.

Az utasításlistás programnyelv felhasználói programvezérlési parancsok (utasítások) sorozatából áll. Egy-egy utasítás a műveleti (operációs) részből és az operandusrészből áll.

A műveleti rész (utasítás) azt határozza meg, hogy a CPU-nak milyen műveletet kell végeznie. Az egyes műveleteket rendszerint az utasítás nevének rövidítésével jelölik (pl. OR, LD, stb.).

Az operandusrész arra a kérdésre ad választ, hogy a műveleti részben definiált műveletet mivel kell elvégezni.



4.2. ábra. Az utasításlistás programnyelv egy utasításának felépítése

A példa szerint az utasítás a 34-es bemeneti (I) bit (B) negáltjának (N) logikai ÉS kapcsolatát végzi az akkumulátor tartalmával.

A be- és kimeneti jeleket (változókat), a belső változókat (merkerek), valamint az időzítőket, számlálókat címezni kell.

A leggyakrabban előforduló műveletek és szokásos jelölésük:

Műveleti elem	Jele (német)	Jele (angol)
Bemenet	E (=Eingang)	I (=Input)
Kimenet	A (=Ausgang)	O / Q =( Output / Quit)
Merker (belső változó)	M (=Merker)	F (=Flag)
Állandó	K (=Konstans)	K (=Constant)
Időtag	T (=Timer)	T (=Timer)
Számláló	Z (=Zähler)	C (=Counter)
Program egység	P	P
Hálózat	NW (=Netzwerk)	NW (=Network)
Működési egység	F	F

4.2. táblázat.

**Bitműveletek:**

Művelet	Jele (német)	Jele (angol)
Logikai összeadás	O (=Oder)	O (=Or)
Logikai szorzás	U (=Und)	A (=And)
Logikai tagadás	N (= Nicht)	N (=No)
Betöltés	L (=Laden)	L (=Load)
Nullművelet	NOP (=Nulloperation)	NOP (=No Operation)
Hozzárendelés kimenethez	=	=

4.3. táblázat.

A felhasználói programszerkesztés legfontosabb szabályai:

- a programozandó folyamatot lépésekre kell bontani (gyakori szóhasználat a lépésekre a „hálózat” is)
- a lépéseknek sorszámot és nevet kell adni (Megjegyzés: a „név” programonként változó! Pl. Step, Schritt, Netzwerk, Network, Rung, Segment stb.)
- programfutás a lépések sorrendjében történik, kivéve, ha valamelyik lépésben egy későbbi lépésre ún. ugró utasítást adunk

- minden lépés elején általában feltételeket adhatunk meg logikai kapcsolatokkal: leggyakoribb az ÉS, VAGY, NEM
- a lépések második részében végrehajtó utasítások adhatók: leggyakoribb a kimenet(ek) be-/kikapcsolása, idő vagy számláló tag indítása
- a ciklikus ismétlődés érdekében a program végét jelző szó vagy rövidítése szükséges, pl. End, PE (Programm end), BE (Bausteinende)

A fenti szabályok általános érvényűek, természetesen eltérés a gyártók szerint előfordulhat. A programírás szintaktikai („nyelvhelyességi”) szabályai szintén gyártófüggőek, amelyek a programok kezelési utasításaiból vagy a „Help”-ből megismerhetők.

Az utasításlistás programozási nyelv elsősorban azoknak ajánlható, akiknek gépközeli programozási ismereteik vannak.

#### 4.4.3. Létradiagramos programozási nyelv

A létradiagramos programozási nyelv (Ladder Diagram, LD vagy LAD, KOP) az áramútervből alakult ki, így a létradiagramok a vezérlési áramútervek szoftver megfelelői. A jelfogós áramutas tervhez hasonló programozási nyelv kidolgozásának igénye már a General Motors 1968-as pályázatban is szerepelt, de széles körű elterjedését csak a grafikai lehetőségekkel ellátott fejlesztőrendszerek eredményezték. Szükségességét főként a jelfogós hálózatok tervezéséhez és üzemeltetéséhez értő szakemberek könnyebb átképzése indokolta. A létradiagramos programozási nyelv alkalmazása az áramúterv bizonyos mértékű ismeretét feltételezi.

#### Létradiagramok

Bár a létradiagramok (LAD, KOP) az áramútervek programozási megfelelői, lényeges különbség közöttük, hogy csak logikai egyenletek leírására szolgálnak, nem tartalmaznak speciális, és csak az érintkezős hálózatokra jellemző összevonásokat és elemei csak az adott PLC funkcionális szoftverelemei lehetnek.

A létradiagramok főbb elemei:

- kontaktusok
- huzalozás
- logikai kimenetek (Output, Flag, MERKER)
- időzítők
- számlálók
- különleges elemek (pl. léptetőregiszter, PID blokk).

Az IEC szabvány főbb LAD grafikus szimbólumai az alábbiakban láthatók:

–   –	Záró- (munka-) érintkező
– / –	Bontó- (nyugalmi) érintkező
– P –	0 → 1 átmenetet adó érintkező
– N –	1 → 0 átmenetet adó érintkező
–( )–	Tekercs
–(/)–	Negált működésű tekercs
–(S)–	RS FF beírótekercs
–(R)–	RS FF törlőtekercs
–(M)–	Tápfeszültség- kimaradáskor állapotát megtartó tekercs
–(SM)–	Tápfeszültség- kimaradáskor állapotát megőrző RS FF beírótekercs
–(RM)–	Tápfeszültség- kimaradáskor állapotát megőrző RS FF törlőtekercs
–(P)–	0 → 1 élre működő (ON) tekercs
–(N)–	1 → 0 élre működő (OFF) tekercs

A létradiagram szimbólumai a vezérlési áramúterv rajzjeleiből alakultak ki. Fontos különbség, hogy a létradiagramon az egyes elemek csak a funkció, ill. viselkedés alapján vannak megkülönböztetve, hiszen a létradiagram szimbólumai nem valódi elemek (érintkezők, beavatkozók), hanem szoftverelemek. Ezért nincsenek megkülönböztetve, pl. a nyomógombok, a jelfogó érintkezők vagy a hőkioldó érintkezője, ill. a jelfogó és a mágneskapcsoló.

A létradiagram készítésének szabályai hasonlóak az áramúterv készítésének szabályaihoz, kivéve a speciális hardvervonatkozásokat:

- a létradiagramon a működtető és a működtetett elemek szimbólumai-ból álló áramutakat két függőleges vonal között vízszintes egyenes mentén ábrázoljuk;
- a működtetett elem egyik sarkát mindig a jobb oldali függőleges egyenesbe kötjük be;
- az összetartozó működtető és működtetett elemeket azonos betűjellel látjuk el, több ilyen elem esetén számozást alkalmazunk;
- a vezetékek kereszteződését kerülni kell;
- az egyes áramutakat a sorrendi működés szerint ábrázoljuk;

- a létradiagram egyes elemeit feszültségmentes állapotban ábrázoljuk;
- a létradiagram egyes elemeihez további adatok is írhatók (pl. időzítésre vonatkozó konstans, számláló-határérték, stb.);
- a speciális funkció sohasem csatlakozhat közvetlenül a bal oldali referenciavezetékre, csak logikai feltételeken keresztül.

#### 4.4.4. Funkcióblokkos/Funkcióterves programozás

A funkcióblokk-diagram (Function Block Diagram, FBD, németül FUP) tulajdonképpen a huzalozott logikában az SSI, MSI áramköröknél használt szimbólumokból kialakított, erősen hardverorientált nyelv. Egy funkcióblokk bal oldalán a bemenetek, jobb oldalán a kimenetek vannak feltüntetve. A jelfolyam iránya az előző fokozat kimenetétől a következő fokozat bemenete felé (balról jobbra). Így az FBD szintaktikai szabályai a huzalozott, feszültséglogikájú hálózatok hardverkialakítási szabályaival egyeznek meg, néhány kivétellel (pl. belső terhelési viszonyok, FAN OUT, FAN IN)

Az FBD/FUP programozási nyelv a feszültség logikájú logikai tervezésben jártas személyek részére ajánlható.

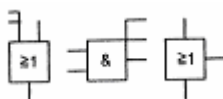
#### 4.4.5. Sorrendi folyamatábrás programozás

Már a huzalozott vezérlések idején felmerült egy szoftverorientált vezérlési feladatleírás mód igénye, mivel az áramútrajz, ill. a logikai szimbólumos leírás módok erősen kötődtek egy-egy megvalósítási módhoz. Kifejlesztették a Grafcet francia nemzeti szabványt. A PLC-k programozására használatos sorrendi folyamatábra (Sequential Flow Chart, SFC) tulajdonképpen a Grafcet továbbfejlesztése és a PLC-k programozásánál mindkettő használatos.

#### Grafcet

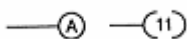
A Grafcet feladatleírás mód szert a számítástechnikában használt folyamatábra és a digitális technika jelölésrendszere alapján dolgozták ki. A működési tervek tartalmazzák a digitális technika szabványos jelöléseit, vagyis a logikai eszközök jele négyzet vagy téglalap alakú, a hatásvonal fentről lefelé, ill. balról jobbra halad. Az ettől eltérő haladási irányt nyíllal kell jelölni. Az alábbi ábra bemutat néhány, a szabvány által megengedett, a logikai áramköri ábrázolásokban nem szokványos, a működési tervekben gyakori jelölési módot. Látható, hogy a jel bármely oldalára együttesen is kerülhet be-, ill. kimenet, valamint a jelölésben össze lehet vonni azokat.





4.3. ábra. A vezérléstechnikában használt, nem szokványos jelölések

A számítástechnikai folyamatábra készítésekor használt kapcsolójelek itt is alkalmazhatók, ha az áttekinthetőség azt kívánja:

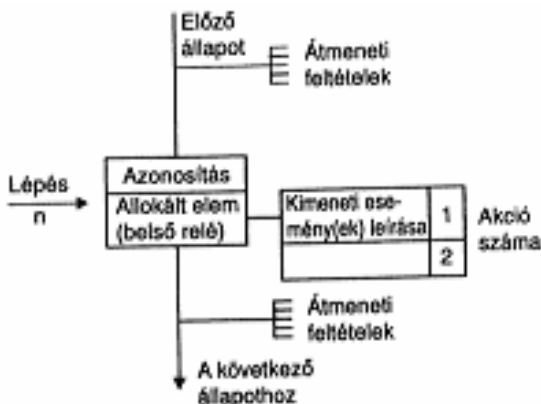


4.4. ábra. A számítástechnikai folyamatábrából átvett jelek

(Megjegyzés: A Grafcet programozás további ismertetése terjedelmi korlátok miatt itt nem lehetséges.)

### Sorrendi folyamatábra

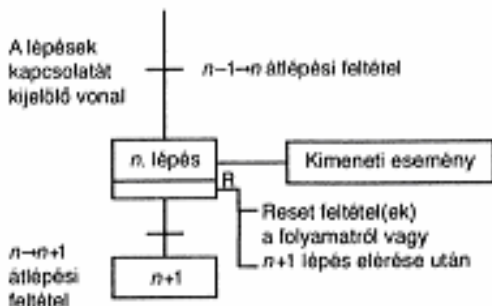
Az IEC 1131 szabvány szerinti sorrendi folyamatábra (SFC) igen hasonlít a Grafcethez. Az SFC i-edik sejtjét szemléletli a következő ábra.



4.5. ábra. A sorrendi folyamatábra általános eleme

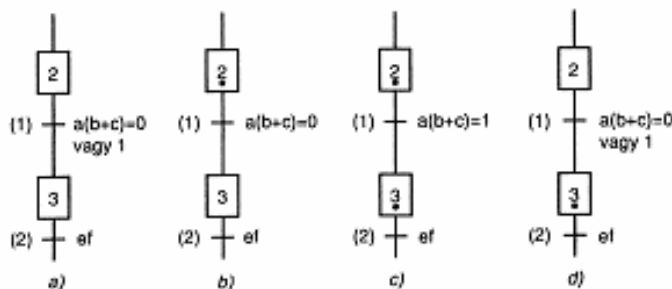
A folyamatábrában a sorrendi hálózat egy-egy belső állapotát, az ezen állapotba kerülés feltételeit, továbbá az állapothoz rendelt kimeneti eseményeket tüntetik fel. Ez a módszer az „1 az n”-ből típusú állapotkódolásból fejlődött ki. Minden lépést (belső állapotot) egy RS Flip-Flop (FF) reprezentál. Ha egy állapot előáll, akkor az előkészíti a következő állapotot és

törli az előző állapotot realizáló FF-t. A mai PLC-kben az RS flipflopokat a RAM memóriabitjei helyettesítik és az átmenetek feltételeinek meghatározását a CPU szoftverúton végzi. Az  $n$ -ik lépés (MERKER) visszaállítást/törlését szemlélteti az alábbi ábra.



4.6. ábra. Az  $n$ -edik lépés és törlése a következő lépéssel

A törlési feltétel rendszerint a következő állapot ( $n+1$ ) beírási feltétele, de elvileg további feltételei is lehetnek az  $n$ -ik lépés törlésének. Az  $n$ -ik lépés törlését csak az utóbbi esetben szokták feltüntetni. A folyamatábrás leírás esetén szigorú szabályok rögzítik az  $n$ -ik lépésről az  $(n+1)$ . lépésre történő előrehaladás rendjét. Egy átmenet átlépését jelentő előrehaladás akkor jöhet létre, ha az adott átmenet érvényes (elért) és a hozzá kapcsolódó átmeneti feltétel teljesül. Ezek együttes teljesülésekor az átlépés kötelező. Átlépéskor a PLC ciklusideje alatt az átmenetet követő lépés aktívvá válik és az átmenetet megelőző lépés inaktíválódik. Egy átmenet átlépésének megfelelő előrehaladást jól szemlélteti a következő ábra:



4.7. ábra. Az  $(n+1)$ -ik állapotba kerülés lehetséges esetei

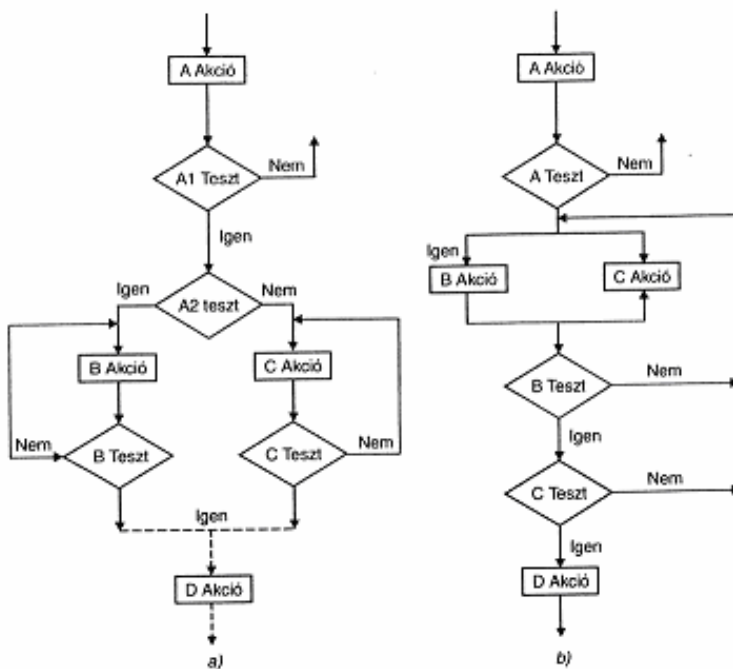
A fenti ábra (a) részén az átmenet nem érvényes, mert a 2. lépés még nem aktív, így a hozzá kapcsolódó feltétel logikai értéke közömbös.

A b ábrán a 2. lépés aktív (\*), az átment tehát elért, de nem léphető át, mert a hozzá kapcsolódó átmeneti feltétel  $a(b \vee c)=0$ .

A c ábrán vázolt átmenet átléphető attól a pillanattól kezdve, hogy az  $a(b \vee c)=1$  feltétel teljesül. Ez esetben az átmenet kötelező.

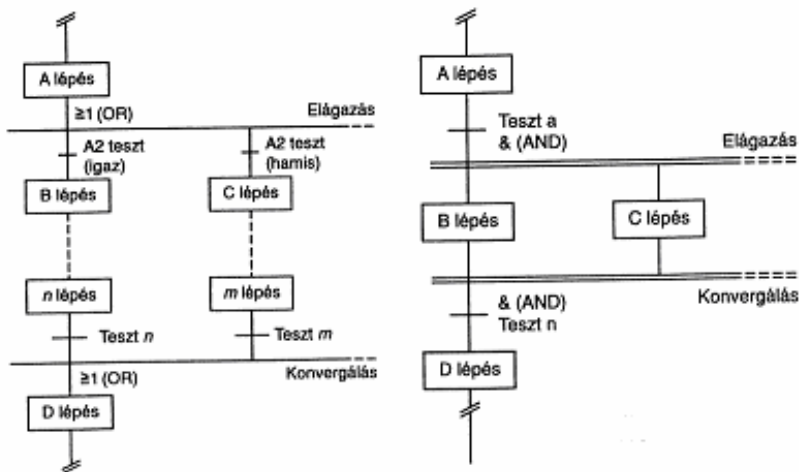
A d ábra azt az esetet szemlélteti, amikor a 3. állapot aktiválása és a 2. állapot inaktiválása megtörtént és a 3. átmenet elért.

Az előzőekben leírt átmenetek időben sorosan folyó események leírásánál fordulnak elő. A szabvány lehetőséget biztosít időben párhuzamosan folyó események leírására is. Ez VAGY, ill. ÉS jellegű elágazással oldható meg. A VAGY (a), ill. ÉS (b) elágazást szemlélteti az alábbi ábra hagyományos szimbólumokkal.



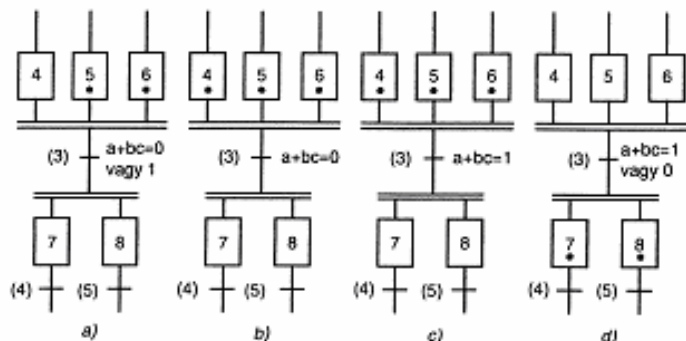
4.8. ábra. VAGY (a) ill. ÉS (b) elágazás jelölése

Az IEC szabvány szerint a VAGY jellegű elágazást egyvonalas, az ÉS jellegű elágazást kétvonalas átmenettel szokás jelölni:



4.9. ábra. IEC szabvány szerinti VAGY ill. ÉS elágazás

Több lépés közötti ÉS elágazással szinkronizált átmenetet mutat a következő ábra.



4.10. ábra. Több lépés közötti ÉS elágazással szinkronizált átmenet

A fenti ábra (a) része szerinti helyzetben az átmenet nem jöhet létre, mert bár az 5., ill. 6. lépés aktív( ), de a 4. lépés nem. Ez ÉS jellegű átmenetnél az átmenet tiltását jelenti. Ilyenkor a (3) feltétel állapota közömbös.

A b ábrán a 4.,5.,6. lépések aktívak, ezért az átmenet elért, de nem de nem léphető át, mert a (3) feltétel nem teljesül.

A c ábrán mindkét feltétel teljesül az átlépéshez, így az átlépés, kötelező.

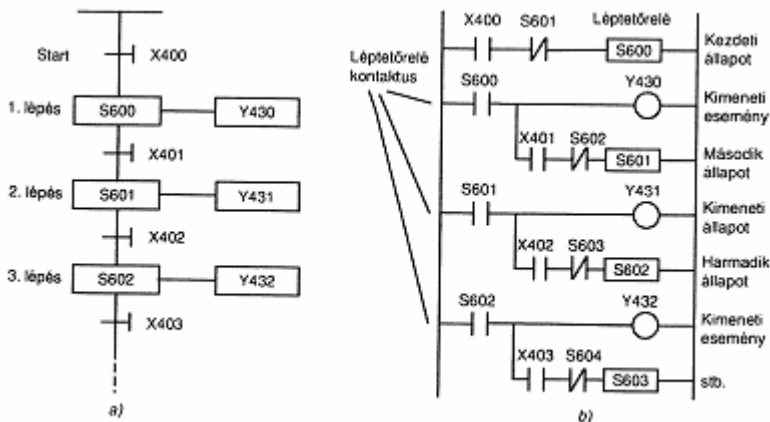
A d ábra az átmenet bekövetkezése utáni állapotot szemlélteti. Az át-  
lépés párhuzamosan megy végbe, amelynek eredményeként:

- a 7., ill. 8. lépés aktiválódik,
- a 4.,5., ill. 6. lépések inaktiválódnak (törlődnek);
- a 7.,8., lépések elértté válnak.

VAGY jellegű elágazásnál egyidejűleg csak az egyik ágon folytatódhat a  
lépések közötti előrehaladás. Ezt úgy érik el, hogy minden ág belépésénél  
eltérő (kizáró) feltételeket határoznak meg.

Az SFC abban tér el az előzőekben ismertetett programozási nyelvek-  
től (ST, IL, LAD, FBD), hogy az SFC nem jelent a PLC-be közvetlenül  
betölthető programot. Az SFC-ből rendszerint az előző négy programozá-  
si nyelv valamelyikén megfogalmazott programot generálják.

Az SFC igényli a legkevesebb speciális ismeretet a feladat megoldásá-  
hoz. Az SFC-vel megírt programok programozásához egyes gyártók a  
létdiagramos programozás keretében ún. steplader (STL), léptetőrelé  
funkciót vezettek be. Az alábbi ábra az SFC (a ábra) és a LAD (b ábra)  
közötti kapcsolatot szemlélteti.



**4.11. ábra.** A léptetőrelé funkció az SFC-ben (a) és realizálása  
létradiagramban (b)

A fenti ábrában az S600 lépés az X 400 bemenettel aktiválódik és az S601  
lépéssel törlődik. Ehhez a lépéshez az Y430 kimenet van rendelve . A tör-

lési funkciót megvalósító érintkezőket, hasonlóképpen az SFC szimbólumhoz, rendszerint nem tüntetik fel a létdiagramon.

## 4.5. A felhasználói programok végrehajtásának módjai

A mai PLC-k olyan speciális mikroszámítógépek, amelyek a programjukkal és speciális be/kimeneti eszközeikkel az irányítási, főként vezérlési feladatok közvetlen végrehajtására alkalmasak. Napjainkban egyre népszerűbb a személyi számítógépek központi egységének használata PLC funkciókhoz. A PLC-k és a számítógépek között a négy legfontosabb különbség a következő: valós idejű működés, környezeti feltételek, programozási nyelvek és a programvégrehajtás módja.

Valós idejű működés: a PLC-k valós idejű (real-time) operációs rendszerrel vannak ellátva, amelynek fő prioritása a be/ki eszköz állapotának lekezelése egy meghatározott válaszidő alatt.

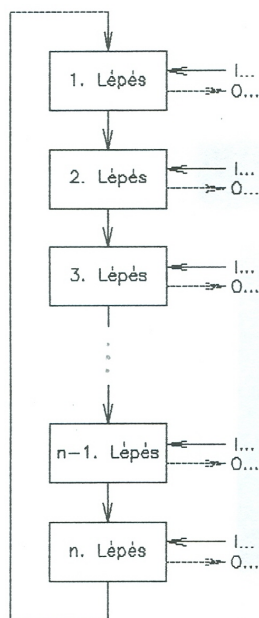
Környezeti feltételek: A PLC-eket ipari környezetben előforduló körülmények (hőmérséklet, páratartalom, rezgés, egyéb zavarok, stb.) közötti működésre tervezték és kiviteleztek.

Programozási nyelvek: A PLC-k speciális, irányítástechnika-orientált nyelvezzel rendelkeznek.

Programvégrehajtás módja: A PLC-k és a PC-k között alapvető különbség a programvégrehajtás módja. A számítógépek ma már korszerű operációs rendszerek révén az ún. multitaskos program-végrehajtási módot, míg a PLC-k a szekvenciális végrehajtási módot alkalmazzák.

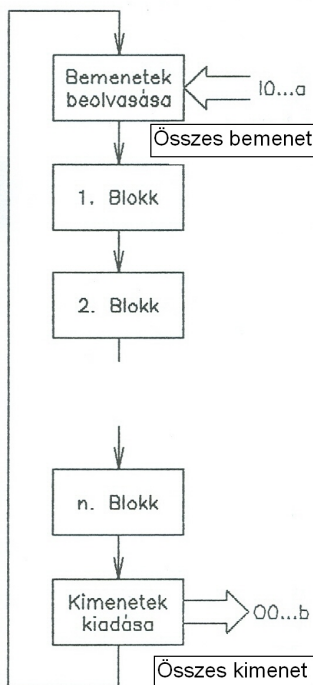
A programozható vezérlők fejlődése során háromféle utasításfeldolgozási móddal találkozhatunk: lépésorientált sorrendi, ciklikus és aciklikus működés.

Lépésorientált sorrendi működés esetén a PLC csak a következő lépés kialakulásának feltételeit vizsgálja. Ilyen felépítésű PLC tehát nem vizsgálja ciklikusan az összes bemenetet (lásd az alábbi ábra). Az ilyen készülék esetén, ha például a következő sorrendi lépés (pl. 14.) feltétele az X4&X5, akkor a 13. lépés teljesülése után a PLC csak X4, X5 bemeneteket olvassa be és vár az ÉS kapcsolat teljesülésére. Az ilyen PLC-k ma már csak ritkán használatosak.



**4.12. ábra.** A lépésorientált működés elve

A ciklikus működésű PLC-k a leggyakoribbak. Ez a program-végrehajtási forma valamennyi folyamateseményt programozottan figyeli a program ciklikusan ismételt végrehajtásával. Ennek a feldolgozási módnak az az előnye, hogy egyszerűbb hardvert és szoftvert igényel, hátránya viszont, hogy a ciklusidő és a reakcióidő függ a felhasználói program hosszától és az utasítások típusától. A ciklikus szervezésű programnak két változata ismert: a lineáris és strukturált programvégrehajtás.



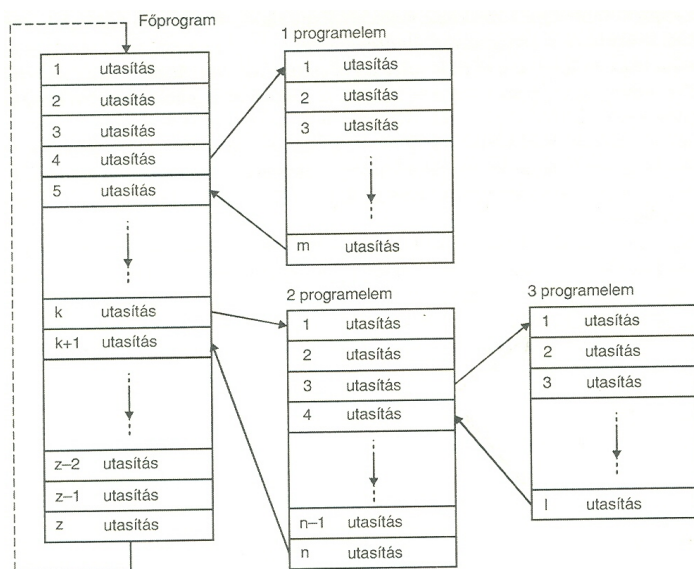
4.13. ábra. A ciklikus működés elve

Lienáris végrehajtásúnak tekinthetők azok a PLC-k, amelyek vezérlésátadó utasításokat nem alkalmaznak, így a program utasításait növekményes, kötött sorrendben hajtják végre. Előnyük, hogy a válaszidők viszonylag könnyen megadhatók, hátrányuk, hogy bonyolult programok esetén igen megnő a letapogatási idő és nincs lehetőség az ismétlődések kihasználására. Általában az egyszerűbb, rövid programokat igénylő feladatok esetén szokás ezt a programszerkesztési módot alkalmazni.

Összetettebb, bonyolultabb, több üzemmódot igénylő alkalmazások esetén a strukturált programszerkesztést szokás használni. A strukturált szervezésű programok főprogramból és az ehhez tartozó alprogrammodulokból állnak. A programmodulok paraméterezhetők, többször is hívhatók és egymásba ágyazhatók. A programozási feladat könnyen felosztható, könnyebb a program tesztelése és módosítása is.

A strukturált szervezésű program elvi felépítése az alábbi ábrán látható.





4.14. ábra. Strukturált programszerkesztés

A fenti ábrán jól látható, hogy a program végrehajtása alapvetően ciklikus az első modulhívó utasításig. A modulhívó utasítás lehet feltételhez kötött, vagy nem. A modulok ún. egymásba foglalási mélysége kötött (általában 8, 16, stb. értékű). A strukturált ciklikus feldolgozású PLC-k előnye, hogy a modulok ismétlődő programrészek egyszerű programozására adnak lehetőséget. Hátrányuk, hogy a programozásuk mélyebb programozási ismereteket igényel és a válaszügy meghatározása nehézkes. Ilyen feldolgozásra csak olyan PLC-k alkalmasak, amelyek utasításkészlete szubrutinhívást, megszakításkezelést, stb. tartalmaz. A strukturált ciklikus PLC-eket némely irodalomban aciklikus program- végrehajtású PLC.-eknek nevezik.

#### 4.5.1. Ciklusidő és reakcióidő

A lineáris, ciklikus működésű PLC tehát az utasításokat ciklikusan, egymás után hajtja végre. Az utolsó utasítás végrehajtása után visszatér a program elejére. A program egyszeri végrehajtási idejét nevezzük programletapogatási időnek vagy ciklusidőnek. Ez az idő több tényező függvénye: függ a program méretétől, az alkalmazott utasítások típusától és a processzor sebességétől. A ciklusidő tipikusan 1-5 ms/K, ahol K a programmemória 1024 rekeszenkénti mérete. Az átlagos letapogatási idő rendszerint 10-50ms között van a program hosszától függően.

A ciklusidő mellett használatos a reakcióidő fogalma is. A PLC-k reakcióidejét ciklusidőn kívül a be – és kimeneti jelkésleltetési idők befolyásolják. A maximális reakcióidő értéke az alábbi képlet alapján határozható meg:

$$T_{\text{rmax}} = 2T_c + T_{\text{bj}} + T_{\text{kj}}$$

ahol

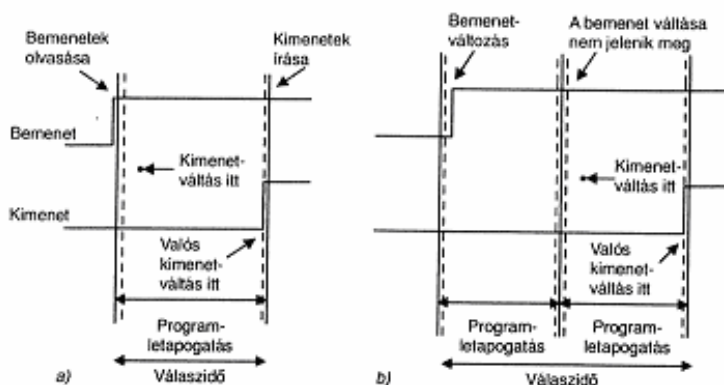
$T_{\text{rmax}}$ : a maximális reakcióidő,

$T_c$ : a PLC ciklusideje,

$T_{\text{bj}}$ : bemeneti jelkésleltetési idő,

$T_{\text{kj}}$ : kimeneti jelkésleltetési idő.

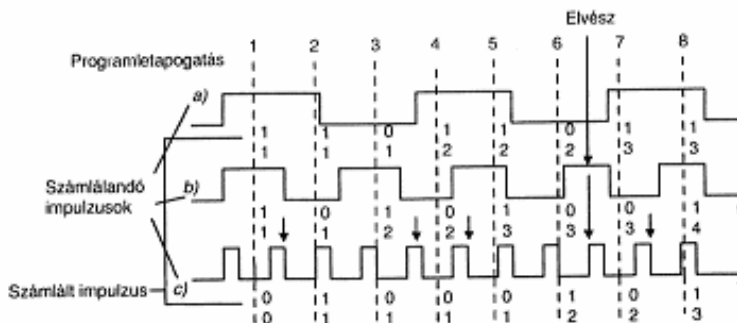
A képletben szereplő kettes szorzó szerepe az alábbi ábra alapján indokolható. A bemeneteken történt jelváltozás és a kimeneteken megjelenő válasz közötti reagálási idő minimális (a) és maximális (b) értéke jól látható az ábrán.



**4.15. ábra.** A ciklusidő értelmezése a ciklikus működésű PLC-k esetén

A reakcióidő képletében szereplő jelkésleltetési idők közül a bemeneti jelkésleltetés a nagyfrekvenciás jelek szűrése, míg a kimeneti jelkésleltetés az induktív terhelések áramlökéseinek csillapítására alkalmazott RC szűrők miatt lép fel. A hálózatba szervezett PLC-k esetén még hálózat adatforgalmához szükséges időt is figyelembe kell venni. A készülékek jelentős részénél a bemeneti jelkésleltetési idők tipikus értéke 2-3 ms, míg a kimeneti jelkésleltetési időké 1 ms. A bemeneti és kimeneti jelkésleltetés főként

analog jelfeldolgozás esetén növekedhet. A PLC ciklusideje meghatározza a PLC-vel feldolgozható jelek frekvenciáját. Ezt szemlélteti az alábbi ábra.

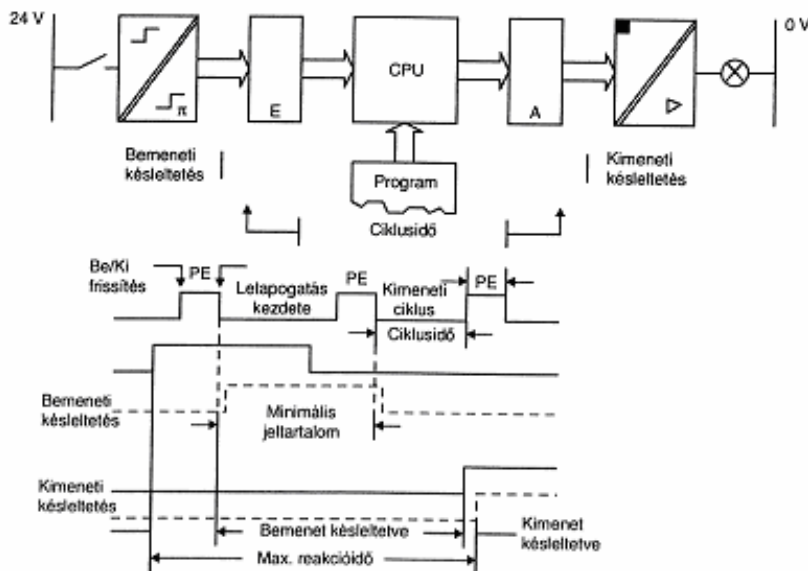


**4.16. ábra.** Eltérő frekvenciájú jelek feldolgozása és a mintavételezési frekvencia kapcsolata

A fenti ábra (a) esetében a vázolt frekvenciájú jelek hibátlanul feldolgozhatók a PLC-vel, mert a jel frekvenciája jóval kisebb, mint a letapogatási frekvencia. A b ábrán a jel a frekvenciája alig nagyobb, mint a PLC mintavételezési frekvenciája, így a két mintavételezés közötti időben egy impulzus elvész. A c esetben a jel frekvenciája jóval nagyobb, mint a letapogatási frekvencia. Az utóbbi két esetben a jelsorozat nem dolgozható fel hibátlanul, a nyíllal jelzett impulzusok elvesznek.

A leírtakból az következik, hogy a számlálást egy bizonyos frekvencia felett rendszerint speciális számlálómodulokkal oldják meg. A folyamatos I/O kezelésű PLC-k esetén lehetőség van a letapogatási frekvencia növelésére úgy, hogy az impulzusbemenet lekérdezését végző programrészt többször is beiktatjuk egy programcikluson belül.

Az előzőek szemléltetésére a korábban bemutatott 80C167 típusú mikroprocesszorral működő PLC jellemző adatainak értelmezése látható az alábbi ábrán.



4.17. ábra. A ciklusidő és a reakcióidő értelmezése

A PLC-k központi processzorának sebességét az utóbbi időben sikerült jelentősen növelni. Ez lehetővé teszi, hogy azonos ciklusidő alatt mind több művelet lehessen végrehajtani.

Az előzőekben leírtakat összefoglalva a ciklikus programvégrehajtás folyamata:

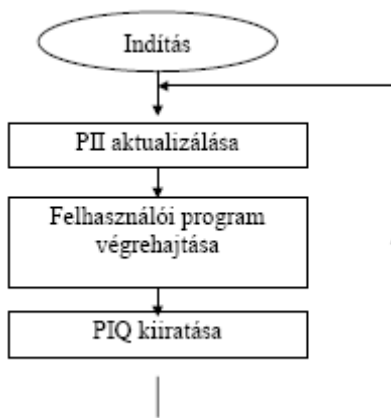
A logikai vezérlő bekapcsolása után azonnal egy olyan ciklusba kerül, amelyből csak kikapcsoláskor fog kilépni. A ciklus a következő 3 tevékenységből áll:

1. A bemenetek értékeinek beolvasása és tárolása a memória egy előre rögzített, a rendszer által kezelt területén. Ez a művelet azzal az előnnyel jár, hogy a felhasználói program futása közben hiába változnak meg a környezet jelei, ezt a program nem érzékeli, így nem léphetnek fel házárdjelenségek. A memória azon területének a neve, ahol a beolvasott értékeket a rendszer tárolja, „bemeneti folyamatkör”, angolul Process Input Image (PII).
2. A felhasználói program végrehajtása. A processzor a végrehajtandó utasításokat egymás után olvassa ki a memóriából. A felhasználói program a bemenetek értékét a PII-ből veszi, és a számolt eredménye-

ket nem közvetlenül írja ki a kimeneti egységekre, hanem szintén a memória egy területére teszi le. Ennek a területnek a neve „kimeneti folyamattükör”, angolul Process Output Image (PIQ).

3. A PIQ-ban tárolt értékek kiírása a kimeneti egységekre. Így minden kimenet egyszerre vált értéket, és elkerülhetők a hazárdok.

Ha időzíteni vagy egyéb okokból szükség van egy bemenet adott pillanatbeli értékének használatára, vagy egy kimenet azonnali beállítására, ez megtehető a PII és a PIQ megkerülésével. Ezekre az esetekre speciális utasításokat használhatunk.



4.18. ábra. A ciklikus programvégrehajtás folyamata

A programmegszakítások kiszolgálása nem része a ciklusidőnek, hanem eseményvezérelt módon (aciklikusan) hajtódik végre. A ciklusidő betartása a valós idejű működés szigorú feltétele. A ciklusidő állandó és változó részből tevődik össze. A változó részt vagy korlátozással látják el vagy maga a PLC méri a ciklusidőt. Egyes PLC típusok esetén lehetőség van a ciklusidő határértékének a megadására. Ezzel az egyes perifériák kiszolgálási ideje, valamint a teljes ciklusidő beállítható és a túllépés lekezelhető.

**Fontos:** a ciklusidő mindig kisebb legyen, mint a ciklusellenőrzési idő! Ellenkező esetben a vezérlő STOP üzemmódba kerül.

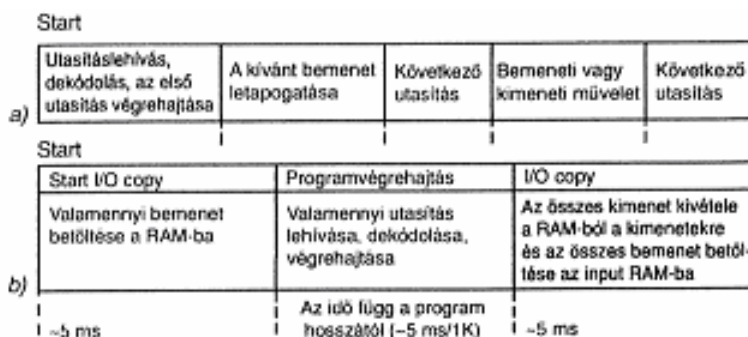
#### 4.5.2. A be-, ill. kimenetek kezelési módja

A be-, ill. kimenetek feldolgozása rendszerint kétféle: folyamatos egyenkénti I/O kezeléssel, vagy blokkos I/O kezeléssel.

A folyamatos I/O kezelés esetén az egyes be-; ill. kimenetek a program végrehajtása közben más–más időpillanatban kerülnek beolvasásra a PLC mintavételezési idejétől és a be/kimenetek programban elfoglalt helyétől függően. A módszer hátránya, hogy gyorsan változó jelek esetén egy mintavételezési cikluson belül ugyanazon változó két mintavételezés között értéke válthat, ami esetleg hibás működést okoz.

A blokkos I/O kezelés esetén az I/O elemek kezelése egy közbenső I/O RAM közreműködésével történik. Az ilyen PLC-k működése két fázisra bontható: I/O kezelés ill. programvégrehajtás. Ez esetben valamennyi be-; ill. kimenet mintavételezése egy időben történik.

A kétféle I/O kezelési módot szemlélteti az alábbi ábra.



4.19. ábra. A folyamatos (a) és a blokkos (b) I/O kezelés időviszonyai

Blokkos I/O kezelésnél a CPU és az I/O RAM közötti adatforgalom rendszerint egy I/O copy szubrutin révén valósul meg. A kétféle I/O kezelés között lényeges különbség, hogy a (b) ábra szerinti I/O kezelésnél garantáltan nem változnak a be-; ill. kimenetek egy program-végrehajtási ciklusban. Ha egy bemeneti jel megváltoztatja állapotát az I/O copy rutin után, akkor az csak a következő I/O copy műveletnél kerül felismerésre és érvényesítésre. Így a programvégrehajtási ciklusban a végrehajtandó logikai műveletek egy mintavételezett állapotra vonatkoznak. Ez a feltétel a folyamatos I/O kezelés esetén nem teljesül.

Gyorsan változó folyamatok esetén a blokkos feldolgozási mód ajánlott.

## 5. Programozható vezérlők hálózatba kapcsolása

Az irányítás (szabályozás, vezérlés és felügyelet) alapvető feltétele a szükséges információk rendelkezésre állása. Minden egyes információhoz általában vezetékre van szükség; másképpen fogalmazva ez azt jelenti, hogy sok információhoz sok vezeték illetve kábel kell. Ez a probléma annál nagyobb, minél komplexebb az a létesítmény, amelyet figyelni vagy irányítani akarunk.

Ha egy mai tipikus gyártólétesítményben ilyen irányító központot akarnánk létrehozni, ez vastag kábelkötegekkel és nagy kábelezési költséggel járna.

Éppen itt jelentkezik a gyártásautomatizálásban a hálózatba kapcsolás igénye. Mit jelent ez a fogalom? Egyszerűsítve azt, hogy az információkat nem egymás mellett küldjük egy vastag kábelkötegen, hanem egymás után egy vékonyabb kéteres kábelben. Erre a kábelre csatlakoznak rá a felhasználók, akik egymással kommunikálni kívánnak. Esetünkben a folyamatvezérlők akarnak egymással folyamatadatokat stb. cserélni.

A részletek mellőzésével is könnyen belátható, hogy egy ilyen vezeték fektetése lényegesen egyszerűbb. A kábelfektetés és maga a kábel is lényegesen olcsóbb. A hálózatba kapcsolás során azonban járulékos költségeket jelentenek a speciális csatolók, amelyek a „busz”-ra való csatlakozást lehetővé teszik.

Mit jelent a „busz”? Leegyszerűsítve busznak nevezzük azt a kéteres vezetékkel, amelyre a különböző résztvevők csatlakoznak. A kétfajta megoldást a költségek alapján összehasonlítva azt találjuk, hogy még kis, áttekinthető létesítmények esetén is előnyösebb.

Az előnyök azonban nem pusztán a költségek terén jelentkeznek. Im már minden szükséges adatot központilag gyűjthetünk és dolgozhatunk fel. Ez lehetővé teszi a munka gyorsabb és pontosabb előkészítését, a folyamatok gyorsabb optimalizálását és módosítását központi előírt értékek segítségével stb. Az összes előny felsorolása túllépné jelen könyv terjedelmét.

A megoldás tehát mind gazdasági, mind szolgáltatási szempontból előnyösebb. Igen fontosak a műszaki előnyök: az üzembiztonság és a teljesítmény.

Az üzembiztonság nő, mivel a hibákat idejekorán fel lehet ismerni, az ok a meglévő adatok alapján felismerhető és egyértelműen azonosíthatóak a létesítmény problémás pontjai.

A teljesítőképesség nő, mert az elosztott intelligencia révén, azaz helyszíni önálló automatikai készülékekkel részfolyamatok önállóan vezérelhetőek és párhuzamosan futtathatók. Ez tehát decentralizált, gyors folyamatvezérlést jelent a létesítménynél, valamint centralizált figyelést és alapadatmegadást.

A jobb áttekintés kedvéért összefoglaljuk a hálózatba kapcsolás előnyeit:

- jelentősen kisebb kábelköltségek
- kisebb méretű kapcsolószekrény, mert a ki/bemeneti egységek decentralizáltak
- kisebb mennyiségű járulékos hagyományos technika, pl. sorozatkapcsok
- kisebb telepítési ráfordítás (bérköltségek)
- kisebb szervizköltségek
- nagyobb üzembiztonság és teljesítmény

Az eddigieket összefoglalva azt mondhatjuk, hogy a hálózatba kapcsolás mindenképpen előnyös az alábbi esetekben:

- összetett, többszörösen tagolt a technológiai folyamat,
- az átlagosnál több be-, kimenet kezelése egy központi helyről,
- egymástól távol lévő önálló egységek irányítása,
- analóg jelek átvitele nagy távolságra.

A hálózatba kapcsolás feladatainak jobb megértése érdekében tekintsünk most egy példát. Képzeljünk el egy gyártóművet, amely három elképzelt síkra oszlik. A legalsó síkon fut a gyártási folyamat, amely több részfolyamatból áll. Ezeknek a folyamatoknak minden részinformációja buszrendszereken keresztül eljut a 2. síkra. A 2. síkon található az irányító központok, amelyek átfogó vezérlési funkciókkal rendelkeznek. Itt történik központi a részfolyamatok koordinálása, valamint a létesítményfigyelés.

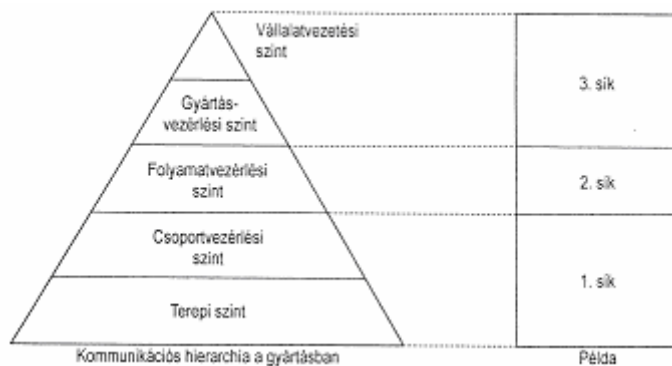
A gyártási folyamat ezen a síkon ér véget.

A továbbiak az irodai síkon zajlanak. Az irodai szint, amely sémánkon a 3. szint, a 2. szinttel kommunikál. Itt futnak össze a logisztikai adatok is, amelyek a szabályozott be- és kiszállítást biztosítják. Jelentős mennyiségű információt kell kicserélni ahhoz, hogy ez a folyamat problémamentesen



működjön. Rugalmas gyártásról beszélünk akkor, ha a gyártásban részt vevő összes készülék – a beszerzéstől a gyártáson át a raktározásig – adatokat tud cserélni. Ebben az esetben az irodából egyszerűen lehet például teljes receptúrákat indítani a gyártósoron, ahol például új keverési aránnyal vagy más alkotóelemek hozzáadásával új termék jön létre. A gyártás során a másik irányban haladnak az üzemi adatok: az iroda felé, minőségbiztosítási célból.

A gyártásban olyan buszrendszerekre van szükség, amelyek kis adatmennyiségeket igen gyorsan visznek át. Ezzel szemben az irodában nagy mennyiségű adatok átvitelére van szükség. Itt a sebesség és a bekapcsolási költség alárendelt szerepet játszik. Az ezekből az eltérő követelményekből adódó buszrendszereket a hálózattechnikában síkokba rendezik, amelyek egy lehetséges elrendezését az alábbi ábra mutatja.



5.1. ábra. A kommunikációs hierarchia síkjai a gyártásban

## 5.1. A hálózattechnika alapjai

A programozható vezérlők üzemszerűen több információforrással állnak kapcsolatban. A leggyakrabban a PLC és a technológiai folyamat, PLC és PLC, PLC és számítógép, PLC és kezelő, valamint PLC és valamilyen periféria közötti kommunikációra van szükség.

## 5.2. Az adatátvitel elve

Adatok átvitelére két alapvetően különböző módszer létezik: a soros és a párhuzamos adatátvitel. Mindkét eljárásnak vannak előnyei és hátrányai, és ezért speciális alkalmazási területei. A következőkben röviden bemutatjuk mindkét módszert.

### 5.2.1. Párhuzamos busz

A párhuzamos busz több párhuzamosan futó vezetékből áll. Ezek mindegyike egy adott funkcióhoz van hozzárendelve. A vezetékek egy-egy csoportjához vannak hozzárendelve a címadatok, a hasznos adatok és a vezérlőadatok. Ha nyolc adatvezetékét használunk, akkor a buszon párhuzamosan le tudunk kérdezni egy bájtot (=8 bit). Ezért az előnyért, azaz az adatok igen gyors átviteléért sok vezetékkel fizetünk. Ebből közvetlenül következik a párhuzamos buszok alkalmazási területe. Az elektronikus adatfeldolgozásban igen nagy szükség van rövid távolságokon igen gyors adatátvitelre, így a párhuzamos buszt itt használják a legelterjedtebben. Párhuzamos buszra tipikus példa egy személyi számítógép belső busza. Itt összesen 62 vezeték vezetnek párhuzamosan: ezek közül 8 adatvezeték, 20 címvezeték, a többi pedig vezérlőjelekhez, valamint táp- és földvezetéként használják.

### 5.2.2. Soros busz

Soros adatátvitel esetén az adatok bitenként, a kiegészítő, ellenőrző jelekkel együtt, időben egymás után egy vezetéken (érpáron) kerülnek továbbításra. Minden információ, tehát az adat-, cím- és vezérlőadatok egymás után mennek át az adatvezetéken. Ezért meg kell állapodni a jelsorozat formátumában, azaz az üzenetformátumban, hogy az adatokat megérkezésük helyén helyesen lehessen értelmezni.

Azt is meg kell említeni, hogy minden használatos soros busz két adatvezetékét használ, hogy így különbségi jelet tudjon átvinni, ugyanis a különbségi jel a normál jelátvitelhez képest jelentősen nagyobb zavarvédelmet biztosít.

A fenti tulajdonságok okán a soros buszt használják nagyobb távolságoknál, például elektronikus adatfeldolgozó berendezések összekapcsolásánál. A soros busz hátránya általában a kisebb adatátviteli sebesség. Ezt a hátrányt viszont különböző megoldásokkal (árnyékolás, nagy integráltságú lapkák alkalmazása, stb.) meg lehet szüntetni.

## 5.3. Hálózati topológiák

Ha több résztvevőt, ill. készüléket kapcsolunk össze adatcsere céljából, akkor különféle struktúrákat lehet kialakítani. Ezek főként a hozzáférés módjában térnek el egymástól. Mielőtt ezekkel a struktúrákkal foglalkoznánk, nézzük meg az adatcsere fogalmát általánosságban.

Az adatcsere különböző módokon és különböző síkokon történik. Itt két fő terület van:

- adatcsere egyazon adatfeldolgozó egységben, pl. egy tárolt programú vezérlésben (PLC) vagy egy PC-ben;
- adatcsere térben különálló készülékek között, pl. összekapcsolt PLC-k között vagy egy bankfiók személyi számítógépe és a központ nagygépe között.

Az adatcsere első fajtája egy rendszerbusznak is nevezett belső buszon történik. Ez kapcsolja össze a lehető legrövidebb útvonalon a készülékben lévő eszközöket. Erre a célra párhuzamos buszt használnak.

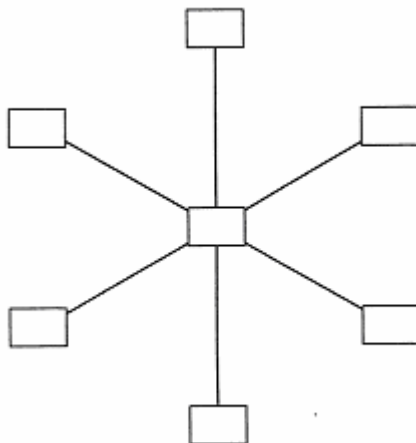
Az adatcsere második fajtájánál a busz térben egymástól különálló készülékek között húzódik. Ilyen esetekben többnyire soros buszt használnak. A busszal összekapcsolt készülékek közötti távolság lehet mindössze 50 m, de akár 1000 km is. Ez a terület tovább bontható nagy és kis távolságú adatátvitelre.

A nagy távolságú adatátvitel pl. a nyilvános telefonhálózaton vagy ehhez hasonló hálózaton történik. Ezeknek a nagy távolságú adatátvitelt lehetővé tevő hálózatoknak a neve Wide Area Network (WAN). Mi a kis, azaz a néhány kilométeres távolságú adatátvitellel fogunk foglalkozni. Ezeket a hálózatokat helyi hálózatnak (Local Area Network = LAN) nevezik. Egy ilyen LAN rendszer összekapcsolhat PLC-ket, PC-ket vagy általánosságban adatfeldolgozó egységeket. Azt, hogy a vezeték hogyan kapcsolja össze a hálózati résztvevőket, a hálózati struktúra határozza meg. Az alábbi alapstruktúrák léteznek: csillag, gyűrű és a busz illetve a fa struktúra.

### 5.3.1. Csillagstruktúra

A csillagstruktúra alapelve az, hogy minden résztvevő állomás a csillag középpontjában lévő központi számítógépre csatlakozik. Minden résztvevő saját átviteli vezetékén csatlakozik a központhoz. A központi számítógép vezérli a teljes kommunikációt, amely ezért viszonylag egyszerű. A rendszer hátrányai nyilvánvalóak:

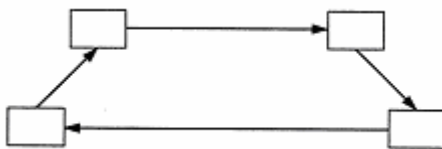
- a központi számítógép kiesése esetén nem lehetséges a kommunikáció,
- magas a kábelezési költség és a
- központi számítógép terhelése nagy, mivel rajta keresztül zajlik az állomások egymás közötti kommunikációja is.



5.2. ábra. Csillagstruktúra

### 5.3.2. Gyűrűstruktúra

Ebben a rendszerben minden állomás egy gyűrűvé záródó átviteli közeggel van összekapcsolva. Ez a gyűrű áthalad minden egyes állomáson, egy adón és egy vevőn keresztül. Így mindegyik állomás része az adatátviteli útnak. Az adatok pontosan egyszer mennek körbe a gyűrűn, állomásról állomásra. Mindegyik állomás megvizsgálja, hogy neki szólnak-e az adatok. Ha igen, beolvassa az adatokat saját tárába. A gyűrűstruktúrára példa az IBM által kifejlesztett Token Ring vagy az InterBus S.



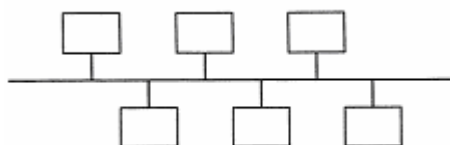
5.3. ábra. Gyűrűs hálózati struktúra

### 5.3.3. Buszstruktúra

A buszstruktúra felvágott gyűrűnek is tekinthető. Az átviteli út itt nem halad át az állomáson, hanem az állomás igen rövid vezetéken csatlakozik a buszra.

A buszon bármelyik állomás kommunikálhat bármelyik másikkal. A csatlakoztatott állomások alapesetben azonos jogosultságúak, ezért a hoz-

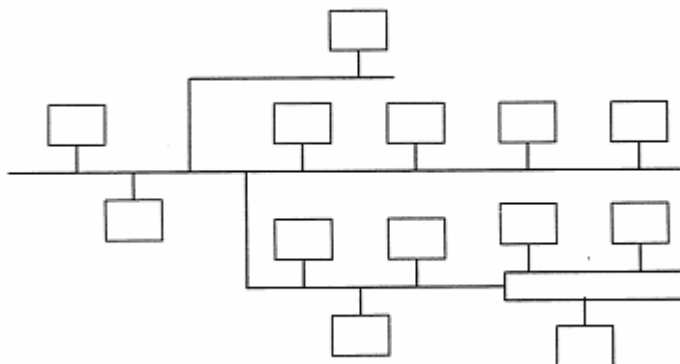
záférést szabályozó eljárásra van szükség. Ez a hozzáférési eljárás szabályozza a busz „forgalmát”. Erre a struktúrára példák a Token Bus, a Master-Slave vagy a CSMA/CD, amelyeket a későbbiekben részletezünk. Rugalmassága okán ez a busztípus a legelterjedtebb a fenti hálózati formák közül.



5.4. ábra. Buszstruktúra

### 5.3.4. Fastruktúra

A buszstruktúra fontos továbbfejlesztése a fastruktúra. Ezzel lehet a legjobban illeszteni a lokális hálózatot egy adott épület konkrét igényeihez.



5.5. ábra. Fastruktúra

## 5.4. Átviteli közeg

Az információ átviteléhez szükség van átviteli közegre. Az átviteli közeg köti össze az összes állomást, amelyek egymással adatokat kívánnak cserélni. Mi lehet átviteli közeg? Különböző lehetőségek közül választhatunk, amelyek főként költség, adatátviteli sebesség, és fektethetőség terén térnek el egymástól. Szokványos átviteli közeg például a sodrott érpár, a koaxiális kábel vagy a fénykábel (fiber-optic cable), de gyakran használnak vezeték

nélküli átvitelt is. Minél közelebb kerül a hálózat az automatizálási folyamathoz, annál lényegesebbé válnak a költségek. A legolcsóbb módszer a sodrott érpár. Ezért főleg ezt használják a gépekben, a folyamatokban vagy a berendezésekben, hogy a legkisebb egységeket is csatlakoztatni lehessen. A koaxiális kábelt főleg az irodákban és a komplex automatizálási készülékeknél alkalmazzák.

#### **5.4.1. Sodrott érpár**

A sodrott (csavart) érpár messze a legkedvezőbb árú módszer a hálózatba kapcsolásra. További előnye a vezeték könnyű fektetése. Hátránya az általában kisebb adatátviteli sebesség (max. 1Mbaud) és a zavarérzékenység. Járulékos árnyékolással mindkét hátrány csökkenthető, ez viszont nyilván rontja a költségelőnyt.

#### **5.4.2. Koaxiális kábel**

Az egyértelműen drágább és nehezebben fektethető koaxiális kábel nagy adatátviteli sebességet (10 Mbaud) tesz lehetővé és egyúttal a legtöbb esetben kielégítő zavarvédelmet nyújt. Már említett hátránya, hogy nehezen fektethető, már kis fektetési hibák (törés, csavarodás) is zavaróan hathatnak.

#### **5.4.3. Fénykábel (fiber-optic cable)**

A fénykábel az átviteli közegek legújabb generációját képviseli. Páratlan zavarvédelem és adatátviteli sebesség, de viszonylag magas költség is jellemzi, bár ez utóbbi csökkenő tendenciát mutat. A fénykábeleket elsősorban erős villamos és/vagy mágneses zavaró terekben alkalmazzák, mivel ezekre teljesen érzéketlen. Átviteli sebessége néhány 100 Mbaud. A fénykábelek anyaga lehet műanyag vagy üvegszál, ezek egymástól eltérő tulajdonságokat mutatnak.

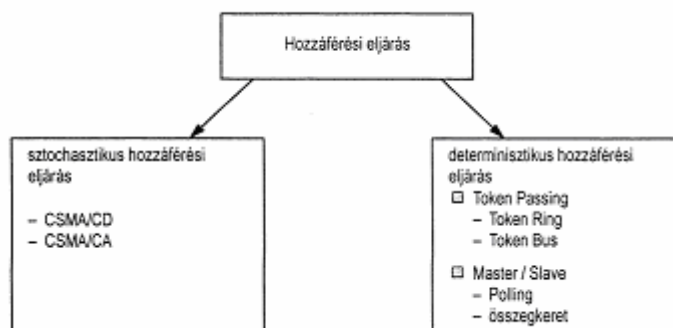
#### **5.4.4. Vezeték nélküli átvitel**

A levegő (illetve az elektromágneses hullám) is átviteli közeg: ezt használják például a rádió- és tévéműsorok, valamint a mobiltelefonia. Nagy előnye az, hogy nem kell kiépíteni az átviteli utat, viszont az átvitel során előfordulhatnak zavarok. Ennek ellenére pl. szervizalkalmazásoknál a vezeték nélküli átvitel nagyon érdekes alternatívája az időigényes és ezért drága helyszíni kiszállásnak.

## 5.5. Buszhozzáférési eljárások

A hálózati kommunikáció esetén egy vezetérendszerre (buszra) több állomás is kapcsolódik, ezért a hibátlan adatkapcsolathoz szükség van az átvitelt vezérlő eljárásra azaz szabályoznunk kell, hogy milyen módon „beszélgetnek” egymással az egyes készülékek. Ezt a szabályozást hívjuk buszhozzáférési eljárásnak. Ezen eljárás vagy szabályok híján a buszon leírhatatlan káosz keletkezne. Az információk egymásra íródna, és kölcsönösen megsemmisítenék egymást az ún. sínkonfliktus miatt. Ezért feltétlenül szükség van egy buszhozzáférési eljárásra a rendezett adatáramlás biztosítása érdekében.

Megkülönböztetünk sztochasztikus (véletlenszerű) és determinisztikus (meghatározott) hozzáférési eljárást.



5.6. ábra. A buszhozzáférési eljárások csoportosítása

Most vegyük sorra a legfontosabb eljárásokat.

### 5.5.1. Sztochasztikus (véletlenszerű) hozzáférési eljárások

#### CSMA/CD

A CSMA/CD a következő angol megjelölés kezdőbetűiből áll: „Carrier Sense Multiple Access with Collision Detection”. Magyar jelentése: vivőfigyelés többszörös hozzáféréssel és ütközésfelismeréssel, amely a következőket jelenti: A buszra csatlakozó állomások állandóan figyelik a buszon lévő üzeneteket. Ha egy állomás adni akar, kivárja, amíg szabadnak látja a közeget. Ebben a pillanatban megeshet, hogy egyidejűleg két állomás is szabadnak detektálja a buszt, és így mindkettő elkezd adni az adatokat.

Ezek az adatcsomagok ekkor ütköznek, ezt a két állomás felismeri és megszakítja az adást. Mindegyik állomásban van egy véletlengenerátor, amely meghatározza, hogy mennyi idő elteltével szabad újra megkísérelni az adást. A véletlengenerátor tehát csökkenti az újabb ütközés valószínűségét.

A CSMA/CD mára erősen elterjedt az irodai kommunikációban, de használják az automatizálás egyes részterületein is.

## CSMA/CA

Alapvetően ugyanaz az alapelv, mint a CSMA/CD esetén, de ez az eljárás másképpen reagál az ütközésekre. A „CA” a „Collision Avoidance”, azaz ütközésselkerülés rövidítése. Míg a CSMA/CD megengedi az ütközést, a CSMA/CA az ütközést felismeri, és a „kevésbé fontos” üzenetet küldő adó abbahagyja az adást, míg a fontosabb üzenetet küldő folytatja. Ez azért lehetséges, mert a fellépő ütközés nem semmisíti meg az üzeneteket. A CSMA/CA tehát hatékonyabb hozzáférési módszer, mivel nincs kényszervárakozás az ismételt adási kísérlet előtt. Példa erre a hozzáférési módra a CAN busz.

### 5.5.2. Determinisztikus (meghatározott) hozzáférési eljárások

#### Token Passing

A Token Passing (vezérljel/zseton adogatás) hozzáférési mód az ún. token (zseton/vezérljel) adogatásán alapul. A token egy rövid üzenetet (bitmintát) jelent, amely az adási jogosultságot hordozza. Amelyik állomásnál a zseton van, az adhat. Adás után a résztvevő köteles a tokent a következő résztvevő felé továbbítani. Amennyiben nem kíván adni, akkor azonnal továbbküldi a tokent a következő résztvevőnek. Az utolsó résztvevő után ismét az első kerül sorra. Ha egyik résztvevő sem kíván adni, akkor a token körbejár. Ez az általános token eljárás tovább osztható a Token Ring és a Token Bus eljárásra.



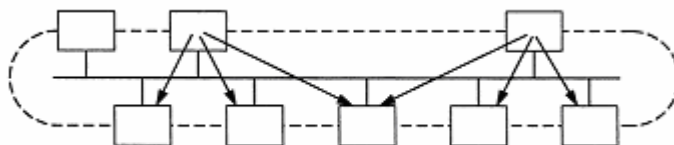
## Token Ring



5.7. ábra. Token Ring buszhozzáférési mód

Az állomások fizikailag egy gyűrűn keresztül vannak összekötve. A zseton a kör mentén állomásról állomásra jár. Ha küldeni akaró állomáshoz ér, az lefoglalja a zsetont és hozzáfűzi az üzenetét. Az üzenet állomásról állomásra adódik tovább. Az állomások egy kódból felismerik, hogy nekik szól-e az üzenet. Ha igen, akkor az illető állomás átveszi az üzenetet. Amikor az üzenet körbejut újra a küldőhöz, az leveszi az üzenetet a gyűrűről, és a zseton így ismét szabaddá válik.

## Token Bus



5.8. ábra. Token Bus buszhozzáférési mód

A Token Bus hozzáférési mód a Token Passing módszer alkalmazása busztopológiájú hálózatokra. Az eljárás során a zsetonelv megmarad. Szemben a gyűrűvel, ahol a sorrendet a körkörös csatlakozás fixen meghatározza, a busz esetén csak „logikai gyűrű” van. Létezik egy lista, amely meghatározza az állomások sorrendjét. Itt tehát a zseton egy listával definiált gyűrű mentén halad. A gyűrűn belül a folyamat ugyanaz, mint a Token Ring esetén, de az üzenet nem állomásról állomásra jut tovább, hanem a buszon keresztül közvetlenül a célállomásra kerül.

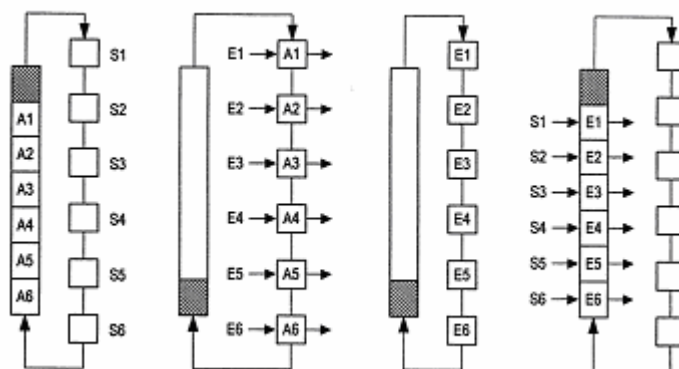
### Master/Slave (polling)



5.9. ábra. Master/Slave hozzáférési mód

Lekérdezéses (polling) eljárás esetén a főállomás (master/mester) sorban egymás után mindegyik mellékállomáshoz (slave/szolga) olyan üzenetet küld, amelyből megtudhatja, hogy azok akarnak-e üzenetet küldeni. Ha a lekérdezett állomásnak van üzenete, akkor azt elküldi a főállomáshoz, amelyik továbbítja ahhoz a résztvevőhöz, amelyiknek a címét az üzenet tartalmazta. Ha a lekérdezett állomásnak nincs továbbítandó üzenete, akkor nemleges választ küld a master lekérdezésére. Amikor a főállomás befejezte az adott állomással való párbeszédet, akkor a lekérdezési lista alapján a következő alállomást kérdezi le. Az elsőtől az utolsó slave-ig tartó ismétlődő lekérdezési ciklust polling ciklusnak nevezzük.

### Master/slave (összesítő keret)



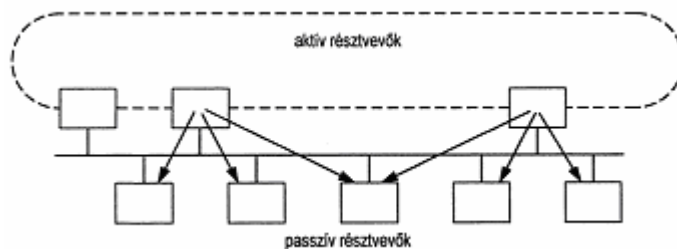
5.10. ábra. A Master/Slave összesítőkeret

A Master/Slave eljárás speciális formája a gyűrűstruktúra és az összesítőkeret összekapcsolása. Az eddig leírt eljárásoknál mindig a vevő

kapott üzenetet, amelyre maga is üzenettel válaszolt. Az összesítő keret különlegessége az, hogy az összes üzenetet egyetlen üzenetben foglalja össze. Ezáltal ez az eljárás jelentősen hatékonyabb, mint a külön-külön üzenetekkel dolgozó eljárások. Az eljárás menete a következő: a master kiküldi az üzenetét (1. fázis), azaz elküldi az első készülékre a gyűrűn, majd addig „tolja” tovább, amíg az üzenet fejrésze a gyűrű mentén vissza nem érkezik hozzá (2. fázis). Most minden üzenet ott van a kívánt vevőknél. A slave-ek kiolvassák az adatokat, beírják az új üzeneteket (3. fázis), majd a master ismét addig tolja az üzenetkeretet, amíg az új üzenet teljes egésze meg nem érkezik (4. fázis). Ezen az elven alapul az InterBus S.

### Hibrid hozzáférési eljárások

Ha két hozzáférési eljárást összekapcsolunk, akkor az ún. „hibrid hozzáférési eljárás” alkalmazására kerül sor. Ennek a csoportnak a legismertebb képviselője a „multi-master” módszer. Ez a Token Passing és a Master / Slave eljárás kombinációja, azzal a céllal, hogy egyesítsük a két módszer előnyeit és csökkentsük a hátrányait. A résztvevők lehetnek aktív vagy passzív résztvevők. A passzív résztvevők csak slave-ként működhetnek, ezért zsetont (token) nem kapnak. Az aktív résztvevők a „logikai Token Ring” gyűrűn vannak, és akár master, akár slave üzemmódban működhetnek.



5.11. ábra. Hibrid hozzáférési eljárás (Multi-master)

## 5.6. Telegram felépítése

Az előző szakaszokban már valamennyire megismertük a busz „KRESZ” szabályait. Ennek során többször előfordult az „üzenet” fogalma. Természetesen ahhoz, hogy az üzenetet a résztvevők megértsék, az üzenetnek előre egyeztetett formátumúnak kell lennie. A pont-pont össze-

köttetéssel szemben ehhez az üzenetfajta-hoz nagyobb szervezési munka szükséges.

Pont-pont összeköttetés van például a PC és a nyomtató között, vagy általánosságban két egymással telefonáló ember között. Nyilvánvaló, hogy ilyenkor az üzenet megértéséhez nem kell megadni az üzenet célpontját vagy időtartamát. A buszon történő kommunikációnál meg kell adni a formátumot, tehát az üzenet felépítését.

A telegram gyakorlatilag az üzenet kerete, és ennek a keretnek általában három része van:

- a fej (header): ebben van a célpont címe és egy vezérlőinformáció, pl. egy hosszadat;
- az adattest (data unit, „DU”): ez tartalmazza az átviendő információkat;
- az adatellenőrző rész (trailer)

Telegramfej (header)	Adattest (body)	Adatellenőrző rész (trailer)
<ul style="list-style-type: none"> <li>– kezdőpont-határoló</li> <li>– cím(ek)</li> <li>– vezérlő-információ</li> </ul>	<ul style="list-style-type: none"> <li>– nettó hasznos adatok</li> </ul>	<ul style="list-style-type: none"> <li>– ellenőrző információ</li> </ul>

5.12. ábra. Egy telegram általános felépítése

### Header

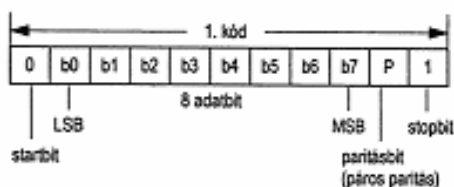
A header (fejrész) tartalmazza a cílcímet (annak a résztvevőnek a címét, amelynek az üzenet szól) és a vezérlőinformációt. A vezérlőinformáció tartalmazhatja például a telegram hosszát és/vagy a telegram típusát. Az adattestben (Data Unit) lévő információk ettől a vezérlőinformációtól függően mást és mást jelenthetnek.

### Data Unit

A Data Unit tartalmazza a tulajdonképpeni üzenetet. Az üzenet hossza lehet telegramtípusonként fix, vagy a fejrészben (header) lévő hosszadattal megadott.

## Trailer

Általános példatelegramunk utolsó része, a trailer ellenőrző információt tartalmaz. A küldő a megelőző telegramelemkből egy adott algoritmus szerint kiszámít egy értéket (ellenőrző kódot), és ezt írja be a trailerbe. A vevő a vett telegramelemkből ugyanazon algoritmus alapján maga is kiszámítja az ellenőrző kódot, és ezt összehasonlítja a küldőtől kapottal. Ezek a telegramrészek maguk is telegramkódokból állnak, amelyek felépítése szintén rögzített. Egy ilyen kód tipikus felépítése látható az alábbi ábrán.



5.13. ábra. Egy telegramkód tipikus felépítése

Az ábrában használt jelölések:

LSB = Least Significant Bit (legkisebb helyértékű bit)

MSB = Most Significant Bit (legnagyobb helyértékű bit)

A kód egy startbittel kezdődik, ez jelöli a kód kezdetét. Ezt követi nyolc biten a tulajdonképpeni információ, majd egy paritásbit. A kód végét a stopbit jelzi. Az aszinkron adatátviteli eljárásoknál a startbit és a stopbit határolja az egyes kódokat. Ezzel szemben a szinkron eljárásoknál csak a teljes telegram elejét és végét kell jelölni. A paritásbitről a következő szakaszban részletesebben szólnunk. A telegramkód tehát 8 adatbit esetén összesen 11 bitből áll. Ez a viszony, valamint a telegram felépítéséből adódó viszony fontos mennyiség az effektív adatátviteli sebesség kiszámításánál. Miért? Azonos adatátviteli sebességnél nagy vezérlőrész és kis adattest mellett nyilvánvalóan kevesebb hasznos adatot tudunk átvinni, mint kis vezérlőrész és nagy adattest alkalmazásával. Vezérlőrészen értjük a headert és trailert, valamint a telegramkód három bitjét. A hasznos adatokat a mindegyik adatblokkban (Data Unit) jelenlévő 8 bit képviseli.

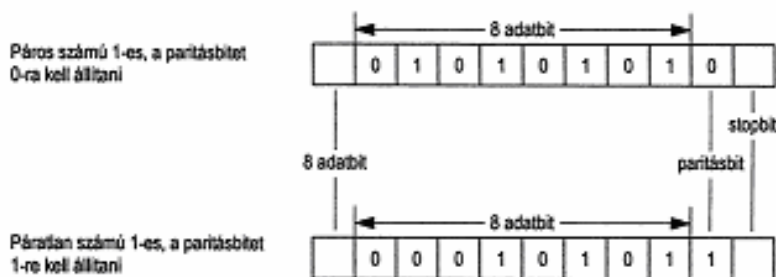
## 5.7. Adatvédelmi módszerek

Az adatátvitel során bármikor felléphetnek zavarok. Meghibásodhat a közege, pl. a kábel vagy hibákat okozhat egy csatlakozón az esetleges túl nagy átmeneti ellenállás. Okozhat zavart a túlfeszültség vagy az erős mágneses mező is. Ezeket és az egyéb átviteli hibákat ki kell szűrní. A telegram felépítéséről szóló szakaszban már szóltunk arról, hogy ennek a funkciónak az elvégzéséhez be kell vezetni egy ellenőrző kódot. Ennek alapján képes a vevő megállapítani, hogy hibátlan volt-e az üzenet átvitele, vagy sem. Ennek a kódnak az alapján persze nem tudjuk rekonstruálni az üzenetet. Ebből következik, hogy megállapítottan hibás átvitel esetén az üzenetet meg kell ismételni. Most azonban néhány szót magáról az adatvédelemről. Az adatátviteli védelmi módszerek közül az alábbi négygel fogunk foglalkozni:

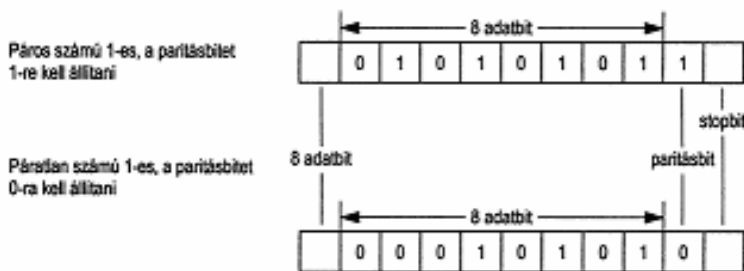
- VRC: keresztparitás
- LRC: hosszparitás
- kombinált paritás
- CRC: ciklikus redundanciakód

### 5.7.1. VRC – Vertical Redundancy Check, keresztparitás

Az átvitt adatok védelmének legismertebb módja a keresztparitás. Ez lehet páros vagy páratlan paritás. Mindkét esetben az adatbitek egy paritásbittel egészítjük ki. Az adatbitekben és a paritásbitben lévő 1-esek számának páros paritás esetén párosnak, páratlan paritás esetén páratlannak kell lennie. Ha tehát páros keresztparitás használata esetén az adatbitek közül három bit 1-es, akkor a paritásbitbe 1-est kell írni, hogy a paritás (az egyesek száma) páros legyen.



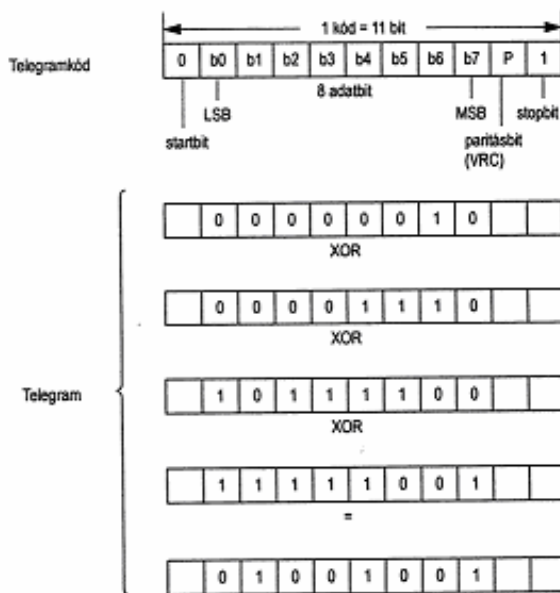
5.14. ábra. Páros keresztparitás



5.15. ábra. Páratlan keresztparitás

### 5.7.2. LRC – Logitudinal Redundancy Check, hosszparitás

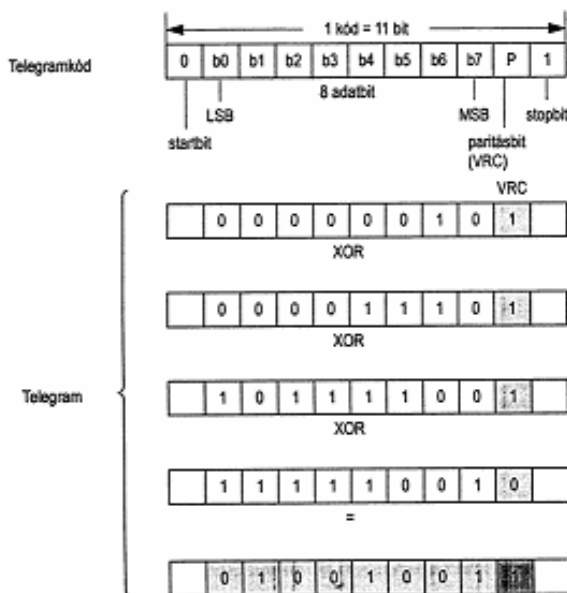
A hosszparitást hasonlóan képezzük, mint a keresztparitást. Ugyanúgy megkülönböztethetjük a páros és a páratlan hosszparitást. A hosszparitást azonban a keresztparitással szemben nem egyetlen kódra képezzük, hanem több telegramkód azonos helyiértékű bitjeire. Így annyi paritásbitet kapunk, ahány adatbit van, és ezeket a paritásbiteket egyetlen kódba foglaljuk össze. Ezt az ellenőrző kódot nevezzük blokkellenőrző kódnak vagy FCS-nek is (Frame Check Sequence)



5.16. ábra. Páros hosszparitás

### 5.7.3. Kombinált paritás

A fentebb leírt két eljárás (LRC/VRC) kombinálása a kombinált paritás. Ennek során mindkét módszert egyidejűleg alkalmazzuk, azaz minden egyes kódra kiszámítunk egy paritásbitet, és kiszámítunk egy paritáskódot a teljes blokkra is.



5.17. ábra. Kombinált paritás

### 5.7.4. CRC – Cyclic Redundancy Check, ciklikus redundanciakód

A paritásos elven alapuló hibaellenőrzés előnye az egyszerűsége, de nagy hátránya az alkalmazhatóság korlátozottsága. A két Hamming-távolságot biztosító paritásos módszer a kereszthibák esetén nem nyújt kellő védeltséget. Ezzel szemben a CRC eljárás jóval nagyobb védelmet biztosít.

A CRC hibaellenőrző kódot is a teljes adatblokkra képezzük. Ennek során az adatblokkot egy hosszú, összefüggő egységnek tekintjük, amelyet elosztunk egy többnyire szabványosított polinommal. Az osztás maradéka az ellenőrző szó, amelyet az adatblokkhoz hozzáfűzünk. A CRC költsége-sebb, mint a két előző eljárás, de mintegy 1000-szer pontosabban ismeri



fel a hibát. A karakterorientált protokollokban használt ellenőrző módszerekkel szemben a CRC-t többnyire bitorientált protokollokban használják.

Nagy hibafelismerési arány miatt a CRC eljárás magasan a legbiztonságosabb. Az átviteli hibákat igen nagy valószínűséggel felismeri.

## 5.8. Átviteli módok

Ejtsünk néhány szót az átviteli módról. Két módszer használatos a jelek fizikai kommunikációs közegen való átvitelére: az alapsávú és a szélessávú eljárás.

### 5.8.1. Alapsávú eljárás

Az alapsávú átvitel esetén az adatjeleket diszkrét villamos ill. fényimpulzusok formájában viszik át. Az ilyen átvitelnél az adó az adatimpulzusokat közvetlenül a kommunikációs csatornán át továbbítja, a vevő pedig ezeket érzékeli. A közeg ezért egyidejűleg csak egy jelet (bitet) tud átvinni.

Mivel az adatimpulzusok a kommunikációs csatornán haladnak, azon torzulást szenvedhetnek. Így a csatorna végén megjelenő jel formája és nagysága már nem feltétlenül az eredeti. Ha a vonal túl hosszú, akkor a vett jel túl gyenge lehet, vagy ha az átviteli sebesség túl nagy, akkor a vett jel felismerhetetlenné válhat. Ezen problémák leküzdésére jelismétlő egységeket (repeater) alkalmaznak, amelyek helyreállítják a jelet és megszüntetik a csatornában keletkezett zajt. Alapsávú átvitel esetén a zaj és a zavar általában nem okoz gondot, kivéve, ha a zavar annyira elrontja a jelet, hogy a 0-s bit 1-ként értelmezhető (vagy viszont). Az alapsávú átviteli módnál a csatorna kapacitását egyetlen adatjel továbbítására használják.

### 5.8.2. Széles sávú eljárás

A széles sávú eljárásnál az adatokat előbb egy vivőjelre modulálják, és ez a modulált jel kerül a közegre. A moduláció során a vivőhullám három jellemzőjének valamelyikét egy adatjelnek megfelelően változtatják, ennek megfelelően van amplitúdó-, frekvencia- és fázismoduláció. A vevőnek a vett jelet demodulálnia kell, hogy hozzáférjen a tulajdonképpeni adatokhoz. Különböző frekvenciájú vivőjelek alkalmazásával a közegen egyidejűleg több jelet is át lehet vinni. Ilyenkor átviteli csatornákról beszélünk. Ezek az átviteli csatornák a rádió- és tévécsatornákhöz hasonlóak.

## 5.9. Buszprotokoll

Mindaddig a hálózatechnikának csak egyes részeivel foglalkoztunk: a hozzáférési eljárással, a telegrammal és az átvitt adatok hibavédelmével. Puskán a közeg vagy a telegram persze még nem elegendő az adatátvitelhez, még kevésbé a buszműködtetéshez. Ez csak az összes szükséges megállapodás együttese révén lehetséges. Ezt az együttest nevezzük buszprotokollnak.

A buszprotokoll definíciója a következő:

*A vezérlési eljárások és üzemi előírások összességét protokollnak nevezzük. A protokoll rögzíti a kódot, az átvitel módját, irányát, formátumát, a kapcsolat felépítését és a kapcsolat elbontását.*

Az irodalomban gyakran keverik a „protokoll” és az „eljárás” fogalmát. Eligazít a DIN 44 300 szabvány, amely kimondja: „eljárás” egy adott protokollal lefolytatott adatátvitel.

### 5.9.1. Az ISO/OSI modell

Az ISO/OSI modell jelentőségének jobb megértése érdekében tekintsük a következő, talán első közelítésben távolinak tűnő helyzetet: Egy arab lótenyésztő lovakat szeretne vásárolni egy argentin méneseből. Itt alapjában véve két probléma lép fel. Először is az arab és az argentin nem ugyanazt a nyelvet beszéli. Másodszor a két személy között nagy földrajzi távolság van, mivel mindkettő otthonról szeretné lebonyolítani az üzletet. Hogyan lehet ezt a problémát megoldani?

Az arab lótenyésztő átadja igényeit egy fordítónak. Ez lefordítja az igényeket angolra. Az angol szöveget most küldönccel el kell küldeni egy Argentínában élő fordítóhoz, aki a szöveget lefordítja spanyolra. Ezután az argentin már el tudja olvasni és ki tudja elégíteni az arab kívánságait.

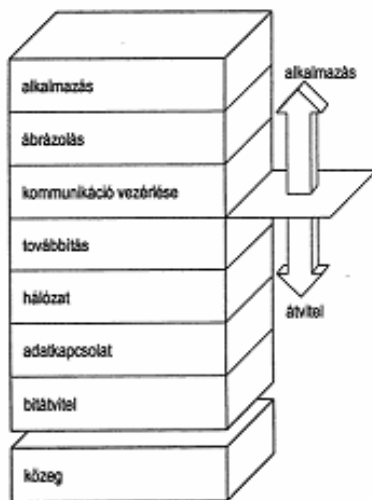
Ez a példa első hallásra egyszerűnek tűnik, de jól illusztrálja a kommunikáció gyakran fellépő problémáit. A levél, a küldönc és az angol nyelv formalizmusának híján a kapcsolat azonnal megszakadna. Ha a két fél valamelyike a kommunikáció nyelvét angolról például magyarra változtatná, a másik már nem értené meg az üzenetet.

A folyamatot több szakaszra felosztva három síkot (réteget) ismerhetünk fel:

- átviteli szint (küldőnc)
- fordítási szint (fordító)
- tárgyalási szint (lótenyésztő)

Ez a felosztás leképezhető az adatkommunikáció műszaki folyamataira is. Annak érdekében, hogy különböző gyártók készülékei tudjanak egymással kommunikálni, létrejött az ISO (International Standard Organization) és megalkotta az ISO/OSI referenciamodellt (OSI = Open System Interconnection), mely 1978-ban lett „a nyílt rendszerek közötti kommunikáció referenciamodellje”. Ez a modell az adatkommunikáció folyamatát hét rétegre osztja, amelyek két fő területre tagolhatók:

- átviteli rétegek (1-4. réteg)
- alkalmazási réteg (5-7. réteg)



5.18. ábra. Az ISO/OSI hétrétegű referenciamodellje

Az ISO/OSI modell rétegei:

**1. réteg** Fizikai vagy bitátviteli réteg

Ez a réteg definiálja a fizikai csatoló mechanikai, villamos és funkcionális paramétereit (átviteli sebesség, karakterhossz, feszültség, közeg, stb.)

## 2. réteg Adatkapcsolati réteg (Data Link Layer)

Az adatkapcsolati réteg nem csak az átvitel biztosítását tartalmazza, hanem meghatározza a telegramszerkezetet, a hozzáférési eljárást, a résztvevők címzését és az adatáramlás vezérlését is.

## 3. réteg Hálózati réteg (Network Layer)

Ez a réteg kezeli az útválasztást (routing), a hálózati kapcsolatok multiplexelését és az adatfolyam kezelését.

## 4. réteg Továbbítási réteg (Transport Layer)

Ez a réteg biztosítja a hibafelismerést és -javítást, valamint az üzenetisméltés eljárásait.

## 5. réteg Kommunikációvezérlő réteg (Session Layer)

Ez a réteg definiálja a kapcsolat felépítésének és lebontásának eszközeit. Ebben a rétegben történik a párbeszédvezérlés is.

## 6. réteg Ábrázolási réteg (Presentation Layer)

Ez a réteg definiálja a konvertálási és formátumillesztési szabályokat, amelyek lehetővé teszik az adatok helyes értelmezését.

## 7. réteg Alkalmazási réteg (Application Layer)

Az alkalmazási réteg a tulajdonképpeni csatolófelület a felhasználó és a hálózat között. Ebben a rétegben vannak definiálva a tulajdonképpeni hálózati szolgáltatások, pl. a fájlátvitel.

Maga az átviteli közeg a szakirodalomban néha az 1. rétegben van, de többnyire egy további rétegben, a 0. rétegben találjuk.

Az ISO/OSI modell csak a kommunikáció folyamatát definiálja. Nem definiál protokollokat, hanem szabványokról beszél. Ezért egy rétegben egymás mellett több protokoll, ill. szabvány lehet. Pl. az ISO/OSI modell 1. rétegébe tartozik többek között az RS 232 és az Ethernet protokoll.

Attól tehát, hogy egy berendezés kielégíti az ISO/OSI szabványt, még nem biztos, hogy képes kommunikálni egy másik olyan berendezéssel, amely szintén eleget tesz a szabványnak.

Ha példánkat ebben a modellben akarjuk értelmezni, akkor az átviteli sík az 1. és a 2. réteg, a fordítás síkja a 6. réteg, a tárgyalási sík pedig a 7.

réteg. Látható, hogy nem vesszük igénybe a modell összes rétegét. Ennek oka az, hogy a konkrét követelmények szerint egy-egy rétegnek csak bizonyos „szolgáltatásait” vesszük igénybe. Ha pl. csak egyetlen hálózaton belül mozgunk, akkor nincs több hálózat közötti összeköttetés és útválasztás, azaz nem kell a 3. réteg.

## 5.10. Hálózati elemek

Mindeddig csak a hálózatról és annak aktív elemeiről szóltunk. Vannak azonban más elemek is, amelyek feltétlenül szükségesek a hálózatépítéshez, ha a hálózat bizonyos határokat túllép. A hálózatok általában címzés (résztevők száma) és méret szempontjából korlátozottak. Korlát az is, hogy nem tudnak olyan más hálózatokkal kommunikálni, amelyek más protokollt használnak. Ezek a korlátok különböző új elemekkel szüntethetők meg. Ezeknek az elemeknek négy alaptípusa van:

- jelismétlő (repeater)
- híd (bridge)
- útválasztó (router)
- átjáró (gateway)

### 5.10.1. Jelismétlő (repeater)

A jelismétlő csak egyforma hálózatok esetén használható, az egyes hálózati szegmensek összekapcsolására szolgál. A hálózat fizikai hatókörét növeli meg azáltal, hogy az egyik oldalról vett jeleket reprodukálja vagy felerősíti, és továbbítja a másik oldalra. A hálózati specifikációban a távolságtól és az állomások számától függően írják elő repeater-ek használatát, de az egymás után kötött jelismétlők számát is korlátozzák. Elsősorban a busztopológiájú hálózatokhoz szokás használni. Gyűrűtopológiájú hálózat esetén ugyanis minden állomás jelismétlő is, ezért külön repeater-re nincs szükség.

### 5.10.2. Híd (Bridge)

A híd eltérő protokollt használó hálózatokat (pl. egy Token-Ring és egy Ethernet hálózatot) tud összekapcsolni. Ez az összeköttetés-típus az üzeneteket egy kis időre a hálózati hídban tárolja, és utána átküldi a másik hálózatba. A híd ezen kívül a hálózati terhelés csökkentésére is szolgál, mivel képes a címszűrésre is. Tekintsünk egy hálózatot, amely két szegmensből áll és híddal van összekötve:

Ha egyazon szegmensben lévő két résztvevő kommunikál egymással, ez a másik szegmensben nem érzékelhető. Azaz a híd csak akkor továbbít üzeneteket, ha valóban áthalad rajta az adatút, tehát a címzési tartomány szerint képes megkülönböztetni a kétfajta esetet.

### 5.10.3. Útválasztó (router)

Az útválasztónak a híddal szemben saját hálózati címe van, ezáltal „eggyel magasabb szintű”, mint a funkcionálisan egyébként vele egyenértékű híd. A hálózati modell fogalmaival élve két hálózat híd esetén legalsó két rétegében, útválasztó esetén legalsó három rétegében különbözhet.

### 5.10.4. Átjáró (gateway)

Az átjáró a legáltalánosabb szintű olyan készülék, amely különböző hálózatok között átjárást biztosít. Teljesen eltérő hálózatokat is össze tud kötni, azaz az egyik hálózat mind a hét rétege eltérő lehet a másiktól. Az átjáró feladata a két architektúra közötti összes átalakítás elvégzése: üzenetformátum, cím- és protokoll átalakítás.

## 5.11. Irányítástechnikai hálózatok

Az irányítástechnikában bizonyos rendszerbonyolultság felett a háromszintű informatikai rendszer terjedt el. Ez a hármas tagozódás a feldolgozási igényekhez igazodik. Ennek megfelelően az alsó szinten történik az érzékelők jelének valós idejű feldolgozása, ill. a beavatkozók jelének kiadása. Ezzel a funkcióval kapcsolatos információk átvitelére az érzékelő, beavatkozó terepi buszokat használják (eszköz szint). Középső szinten (automatizálási szint), a PLC-k, CNC-k, digitális szabályozók, stb. közötti kommunikáció zajlik, szintén terepi buszrendszeren. A felső szinten a termeléssel, gyártással kapcsolatos információfeldolgozás folyik, amihez a munkaállomások, számítógépek, stb. közötti nagy mennyiségű információ gyors és nagy távolságú átvitelére alkalmas ETHERNET hálózatot használnak (információs szint). Ezen a szinten a valós idejű feldolgozás igénye nem merül fel.

A különböző készülékgyártók a különböző informatikai és gyártási szintekhez igazodóan számos hálózati rendszert dolgoztak ki, amelyek közül több szabványosításra is került. Terjedelmi korlátok miatt e jegyzet keretében nincs lehetőség a különböző gyártóspecifikus buszrendszerek bemutatására, a gyorsan terjedő terepi buszrendszerek ismertetésére. Külön jegyzetben tervezzük bemutatni az ETHERNET, a PROFIBUS, a

PROFINET, az INTERBUS, az ASI-bus, a CAN-bus, stb. legfontosabb jellemzőit, alkalmazási lehetőségeit.

Ugyancsak nem tértünk ki a PLC-s vezérlések megbízhatóságának növelési lehetőségeire illetve a biztonsági PLC-k bemutatására, a biztonság fokozásának lehetőségeire.

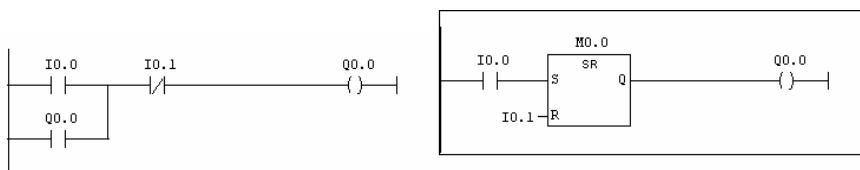
## 6. PLC programozási példák

Jelen fejezetben néhány konkrét feladat PLC programjának bemutatásával segítséget kívánunk nyújtani az olvasónak a PLC-k működésének jobb megértéséhez illetve a programozásuk begyakorlásához. A feladatokat az egyszerűtől a bonyolultabb felé haladva mutatjuk be. A feladatok PLC programja a Siemens cég STEP 7 Professional 12/2002 számú verziójával készült.

### 1. feladat

Készítsük el egy motorvezérlés nyomógombos vezérlőprogramját tartóáramkörös illetve merkeres változatban!

A megoldások létradiagramos illetve utasításlistás formában:



```

A (
O   I   0.0
O   Q   0.0
)
AN  I   0.1
=   Q   0.0

```

```

A   I   0.0
S   M   0.0
A   I   0.1
R   M   0.0
A   M   0.0
=   Q   0.0

```

Megjegyzés: Biztonságtechnikai megfontolások miatt célszerűbb az SR tároló alkalmazása az RS helyett.

### 2. feladat

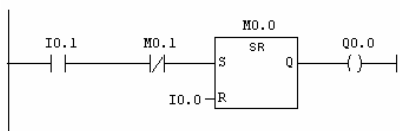
Készítsük el egy irányváltós motorvezérlés nyomógombos vezérlőprogramját!

A megoldások létradiagramos illetve utasításlistás formában:



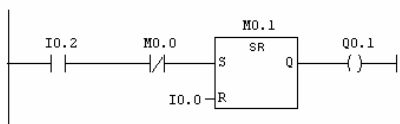
Network 1: Title:

Jobbra forgás



Network 2: Title:

Balra forgás



Network 1: Title:

Jobbra forgás

A	I	0.1
AN	M	0.1
S	M	0.0
A	I	0.0
R	M	0.0
A	M	0.0
=	Q	0.0

Network 2: Title:

Balra forgás

A	I	0.2
AN	M	0.0
S	M	0.1
A	I	0.0
R	M	0.1
A	M	0.1
=	Q	0.1

Megjegyzés: A jelen megoldási változatban a forgásirányváltást meg kell, hogy előzze egy kikapcsolás az ún. keresztreteszelés kialakítása miatt. Más kialakításban természetesen lehetséges a kikapcsolás nélküli forgásirányváltás is.

### 3. feladat

Készítsük el egy 4 motort és 3 jelzőlámpát tartalmazó rendszer vezérlő-programját!

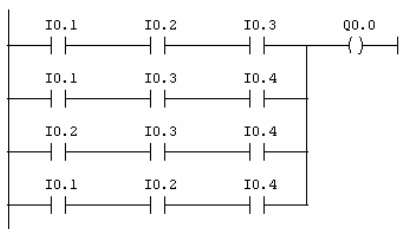
A működés jellemzői:

- Az 1. lámpa (zöld) világít, ha mind a 4 motor üzemel vagy bármelyik 3.
- A 2. lámpa (sárga) világít, ha csak 2 motor üzemel, (de bármelyik 2).
- A 3. lámpa (piros) világít, ha egy motor sem üzemel vagy csak 1 (de bármelyik a 4 közül).

Egy lehetséges megoldás létradiagramos illetve utasításlistás formában:

**Network 1:** Title:

1. lámpa (zöld)



OE1 : "Main Program Sweep ("

Comment:

**Network 1:** Title:

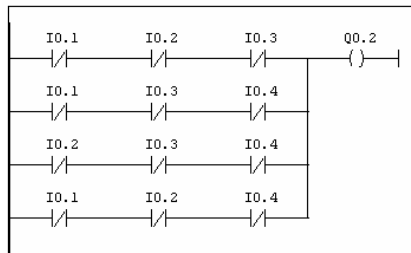
1. lámpa (zöld)

```

A   I   0.1
A   I   0.2
A   I   0.3
O
A   I   0.1
A   I   0.3
A   I   0.4
O
A   I   0.2
A   I   0.3
A   I   0.4
O
A   I   0.1
A   I   0.2
A   I   0.4
=   Q   0.0
    
```

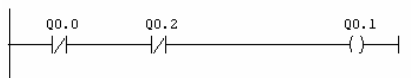
**Network 2:** Title:

3. lámpa (piros)



**Network 3:** Title:

2. lámpa (sárga)



**Network 2:** Title:

3. lámpa (piros)

```

AN  I   0.1
AN  I   0.2
AN  I   0.3
O
AN  I   0.1
AN  I   0.3
AN  I   0.4
O
AN  I   0.2
AN  I   0.3
AN  I   0.4
O
AN  I   0.1
AN  I   0.2
AN  I   0.4
=   Q   0.2
    
```

**Network 3:** Title:

2. lámpa (sárga)

```

AN  Q   0.0
AN  Q   0.2
=   Q   0.1
    
```

**4. feladat**

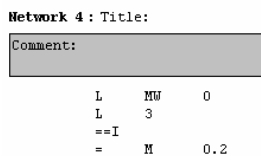
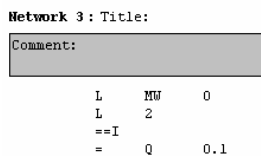
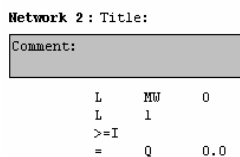
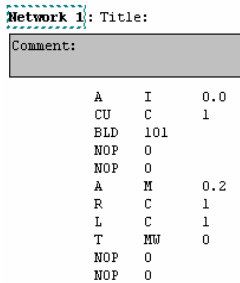
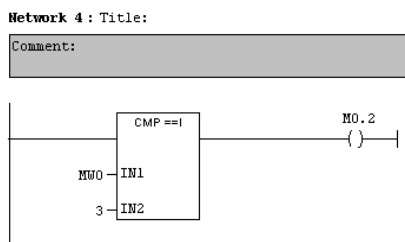
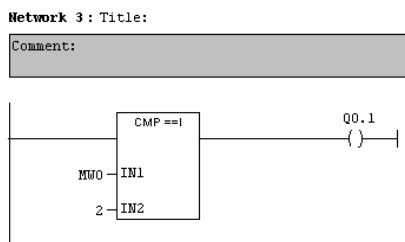
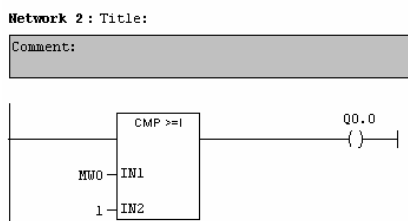
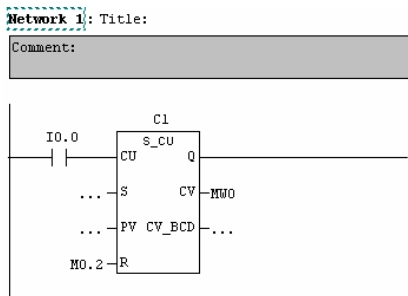
Készítsük el egy 1 nyomógombot és 2 jelzőlámpát tartalmazó rendszer vezérlőprogramját!

A működés jellemzői:

- A nyomógomb 1. megnyomásakor az 1. lámpa kigyullad és égve marad.
- A nyomógomb 2. megnyomásakor a 2. lámpa is kigyullad és égve marad (mindkét lámpa világít).

- A nyomógomb 3. megnyomásakor mindkét lámpa elalszik.
- A nyomógomb 4. megnyomásakor a folyamat előlről kezdődik.

Egy lehetséges megoldás létradiagramos illetve utasításlistás formában (számláló és komparátorok használatával):

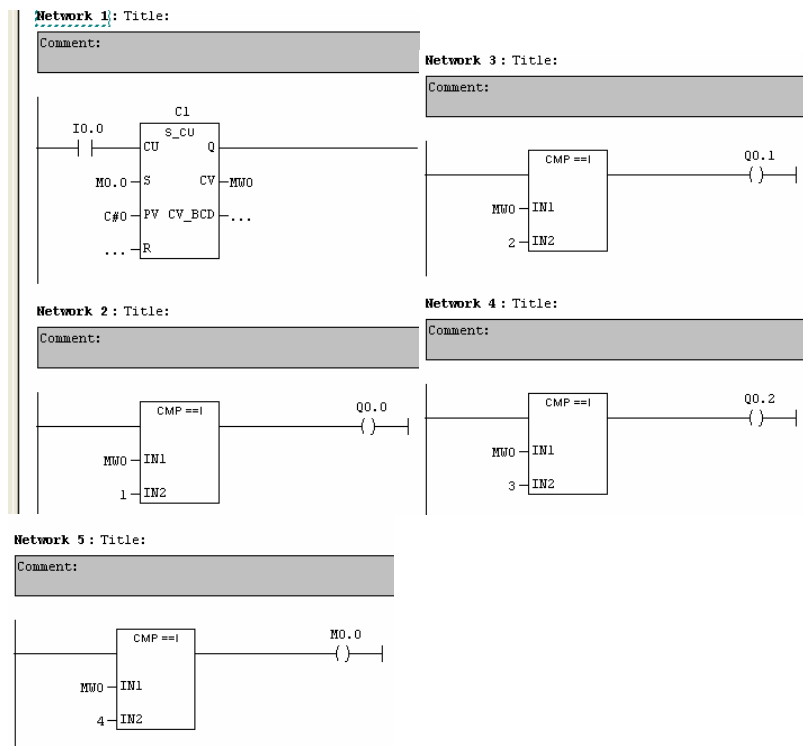


### 5. feladat

Készítsük el egy 1 nyomógombot és 3 jelzőlámpát tartalmazó rendszer vezérlőprogramját!  
A működés jellemzői:

- A nyomógomb 1. megnyomásakor az 1. lámpa kigyullad és égve marad.
- A nyomógomb 2. megnyomásakor a 2. lámpa kigyullad és az 1. elalszik.
- A nyomógomb 3. megnyomásakor a 3. lámpa kigyullad és a 2. elalszik.
- A nyomógomb 4. megnyomásakor a folyamat előlről kezdődik (1. lámpa kigyullad, stb.).

Egy lehetséges megoldás létradiagramos illetve utasításlistás formában (számláló és komparátorok használatával):



```

Network 1: Title:
Comment:
    A   I   0.0
    CU  C   1
    BLD 101
    A   M   0.0
    L   C#0
    S   C   1
    NOP 0
    L   C   1
    T   MW  0
    NOP 0
    NOP 0
    
```

```

Network 2: Title:
Comment:
    L   MW  0
    L   1
    ==I
    =   Q   0.0
    
```

```

Network 3: Title:
Comment:
    L   MW  0
    L   2
    ==I
    =   Q   0.1
    
```

```

Network 4: Title:
Comment:
    L   MW  0
    L   3
    ==I
    =   Q   0.2
    
```

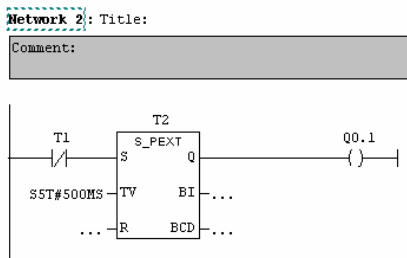
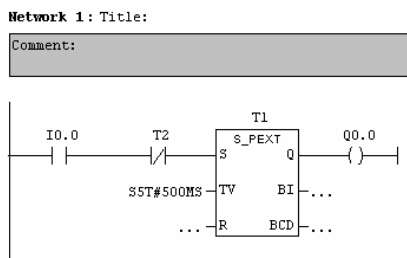
```

Network 5: Title:
Comment:
    L   MW  0
    L   4
    ==I
    =   M   0.0
    
```

## 6. feladat

Készítsünk villogót 2 lámpa segítségével!

Egy lehetséges megoldás létradiagramos illetve utasításlistás formában (időzítők használatával):



```

Network 1: Title:
Comment:
    A   I   0.0
    AN  T   2
    L   S5T#500MS
    SE  T   1
    NOP 0
    NOP 0
    NOP 0
    A   T   1
    =   Q   0.0
    
```

```

Network 2: Title:
Comment:
    AN  T   1
    L   S5T#500MS
    SE  T   2
    NOP 0
    NOP 0
    NOP 0
    A   T   2
    =   Q   0.1
    
```

### 7. feladat

Készítsünk futófényt 4 lámpa segítségével!

Egy lehetséges megoldás létradiagramos illetve utasításlistás formában időzítők használatával:

**Network 1:** Title:  
Comment:

**Network 2:** Title:  
Comment:

**Network 3:** Title:  
Comment:

**Network 4:** Title:  
Comment:

**Network 1:** Title:  
Comment:

```

A      I      0.0
AN     T      4
L      S5T#100MS
SE     T      1
NOP    0
NOP    0
NOP    0
A      T      1
=      Q      0.0
        
```

**Network 2:** Title:  
Comment:

```

AN     T      1
L      S5T#100MS
SE     T      2
NOP    0
NOP    0
NOP    0
A      T      2
=      Q      0.1
        
```

**Network 3:** Title:  
Comment:

```

AN     T      2
L      S5T#100MS
SE     T      3
NOP    0
NOP    0
NOP    0
A      T      3
=      Q      0.2
        
```

**Network 4:** Title:  
Comment:

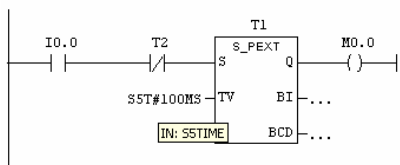
```

AN     T      3
L      S5T#100MS
SE     T      4
NOP    0
NOP    0
NOP    0
A      T      4
=      Q      0.3
        
```

A feladat megoldható számláló és komparátorok segítségével is. A megoldás létradiagramos formában:

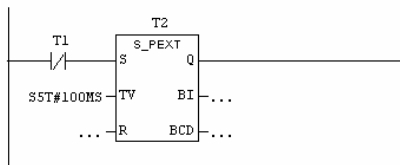
**Network 1:** Title:

Comment:



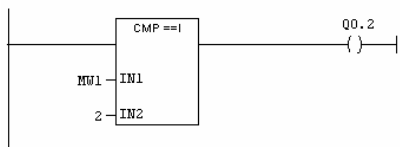
**Network 2:** Title:

Comment:



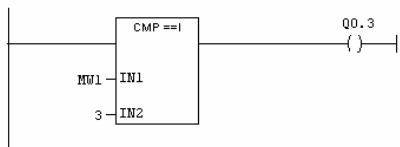
**Network 5:** Title:

Comment:



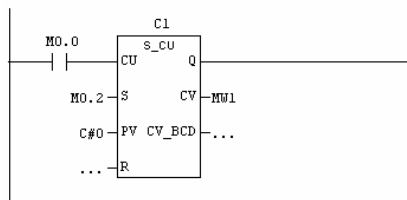
**Network 6:** Title:

Comment:



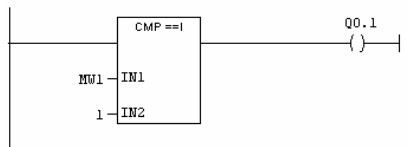
**Network 3:** Title:

Comment:



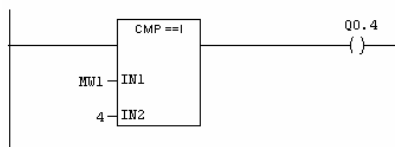
**Network 4:** Title:

Comment:



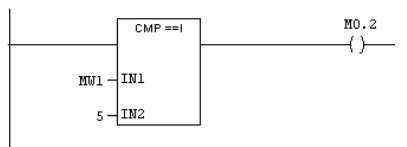
**Network 7:** Title:

Comment:



**Network 8:** Title:

Comment:



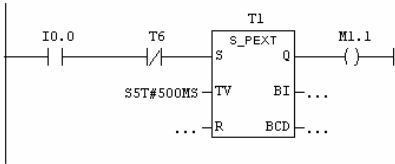
### 8. feladat

Készítsük el egy 4 lámpát tartalmazó oda-vissza futófény vezérlőprogramját!

Egy lehetséges megoldás létradiagramos illetve utasításlistás formában (figyeljük meg a kimenetek vezérlési megoldását):

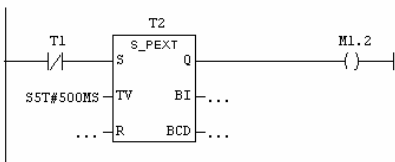
**Network 1:** Title:

Comment:



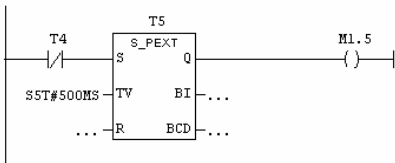
**Network 2:** Title:

Comment:



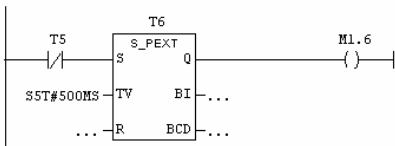
**Network 5:** Title:

Comment:



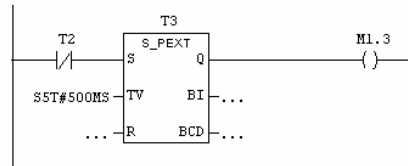
**Network 6:** Title:

Comment:



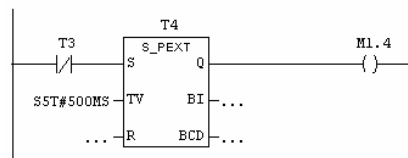
**Network 3:** Title:

Comment:



**Network 4:** Title:

Comment:



**Network 7:** Title:

Comment:



**Network 8:** Title:

Comment:





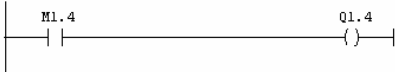
**Network 9** : Title:

Comment:



**Network 10** : Title:

Comment:



**Network 1** : Title:

Comment:

```

A   I   0.0
AN  T   6
L   S5T#500MS
SE  T   1
NOP 0
NOP 0
NOP 0
A   T   1
=   M   1.1
    
```

**Network 3** : Title:

Comment:

```

AN  T   2
L   S5T#500MS
SE  T   3
NOP 0
NOP 0
A   T   3
=   M   1.3
    
```

**Network 5** : Title:

Comment:

```

AN  T   4
L   S5T#500MS
SE  T   5
NOP 0
NOP 0
NOP 0
A   T   5
=   M   1.5
    
```

**Network 2** : Title:

Comment:

```

AN  T   1
L   S5T#500MS
SE  T   2
NOP 0
NOP 0
NOP 0
A   T   2
=   M   1.2
    
```

**Network 4** : Title:

Comment:

```

AN  T   3
L   S5T#500MS
SE  T   4
NOP 0
NOP 0
NOP 0
A   T   4
=   M   1.4
    
```

**Network 6** : Title:

Comment:

```

AN  T   5
L   S5T#500MS
SE  T   6
NOP 0
NOP 0
NOP 0
A   T   6
=   M   1.6
    
```

<b>Network 7 : Title:</b>		
Comment:		
A	M	1.1
=	Q	1.1
<b>Network 8 : Title:</b>		
Comment:		
0	M	1.2
0	M	1.6
=	Q	1.2
<b>Network 9 : Title:</b>		
Comment:		
0	M	1.3
0	M	1.5
=	Q	1.3
<b>Network 10 : Title:</b>		
Comment:		
A	M	1.4
=	Q	1.4

## 9. feladat

Készítsük el egy lépcsőházi világítás vezérlőprogramját!

A működés jellemzői:

- A nyomógomb egyszeri rövid megnyomásával a világítás T1 időre bekapcsol.
- A nyomógomb egyszeri hosszú megnyomásával a világítás T2 időre bekapcsol ( $T2 > T1$ ).
- A nyomógomb egymást követő kétszeri megnyomásával a világítás állandó jelleggel bekapcsolódik és a nyomógomb újbóli megnyomásakor kapcsol ki.

Egy lehetséges megoldás létradiagramos illetve utasításlistás formában:

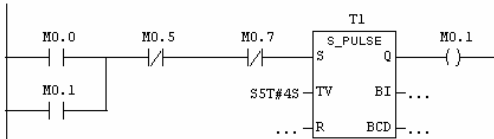
**Network 1:** Title:

Comment:



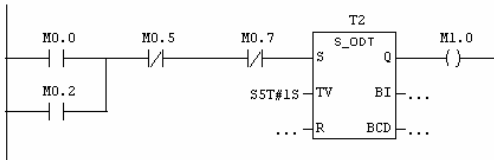
**Network 2:** Title:

Comment:



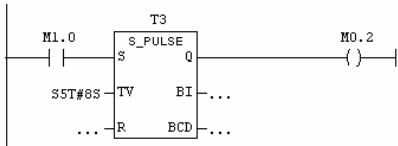
**Network 3:** Title:

Comment:



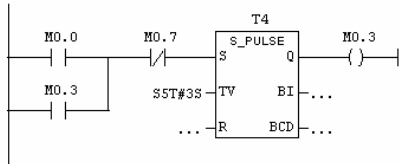
**Network 4:** Title:

Comment:



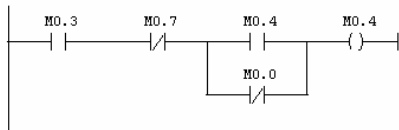
**Network 5 : Title:**

Comment:



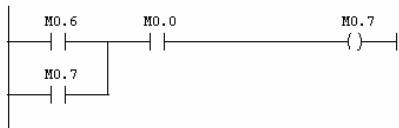
**Network 6 : Title:**

Comment:



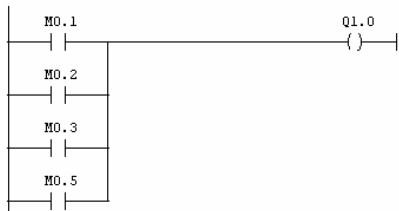
**Network 9 : Title:**

Comment:



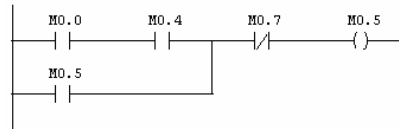
**Network 10 : Title:**

Comment:



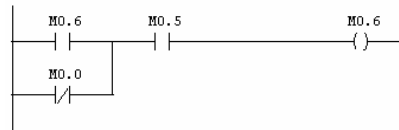
**Network 7 : Title:**

Comment:



**Network 8 : Title:**

Comment:



```

Network 1: Title:
Comment:
    A    I    1.0
    =    M    0.0

Network 2: Title:
Comment:
    A(
    O    M    0.0
    O    M    0.1
    )
    AN   H    0.5
    AN   M    0.7
    L    S5T#4S
    SP   T    1
    NOP  O
    NOP  O
    NOP  O
    A    T    1
    =    H    0.1

Network 3: Title:
Comment:
    A(
    O    M    0.0
    O    M    0.2
    )
    AN   M    0.5
    AN   M    0.7
    L    S5T#1S
    SD   T    2
    NOP  O
    NOP  O
    NOP  O
    A    T    2
    =    M    1.0

Network 4: Title:
Comment:
    A    M    1.0
    L    S5T#8S
    SP   T    3
    NOP  O
    NOP  O
    NOP  O
    A    T    3
    =    M    0.2

Network 5: Title:
Comment:
    A(
    O    M    0.0
    O    M    0.3
    )
    AN   M    0.7
    L    S5T#3S
    SP   T    4
    NOP  O
    NOP  O
    NOP  O
    A    T    4
    =    M    0.3

Network 6: Title:
Comment:
    A    M    0.3
    AN   M    0.7
    A(
    O    M    0.4
    ON   M    0.0
    )
    =    M    0.4

Network 7: Title:
Comment:
    A(
    A    M    0.0
    A    M    0.4
    O    M    0.5
    )
    AN   M    0.7
    =    H    0.5

Network 8: Title:
Comment:
    A(
    O    H    0.6
    ON   M    0.0
    )
    A    M    0.5
    =    M    0.6

Network 9: Title:
Comment:
    A(
    O    M    0.6
    O    M    0.7
    )
    A    M    0.0
    =    M    0.7

Network 10: Title:
Comment:
    O    M    0.1
    O    M    0.2
    O    M    0.3
    O    M    0.5
    =    Q    1.0
    
```

### 10. Feladat

Készítsük el egy közúti félútelzáró jelzőlámpa vezérlőprogramját!

A rendszer „tartozékai”:

1 db főkapcsoló (S0 >> I0.0)

2 db járműérezkelő (E1, E2 >> I0.1, I0.2)

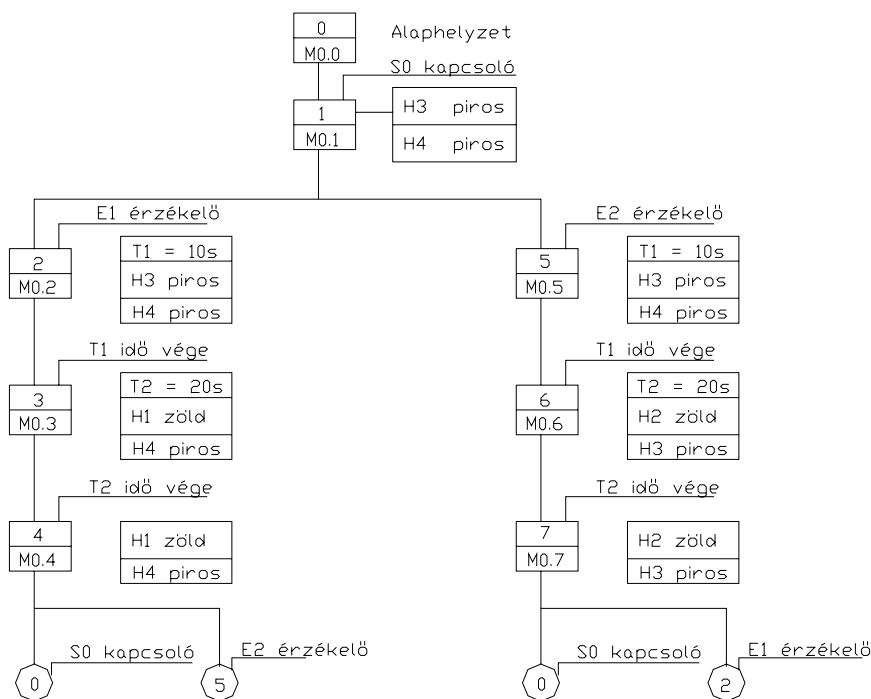
2x2 db jelzőlámpa (H3, H4: piros >> Q0.3, Q0.4 és H1, H2: zöld >> Q0.1, Q0.2)

A működés jellemzői:

- A rendszer bekapcsolásakor mindkét irányban piros lámpa világít.
- Amennyiben valamelyik irányból jármű érkezik, akkor  $T1=10s$  múlva az adott irányban a zöld lámpa világít. A zöld fázis ideje legalább  $T2=20s$ .
- Ha a másik irányból is érkezik jármű, akkor a zöld fázist követően mindkét irányban a piros lámpa világít. Ezt követően 10s múlva most a másik irányban lesz zöld fázis.
- Amennyiben egyik irányból sem érkezik jármű, akkor az utolsó zöld fázis aktív maradhat.
- A berendezés kikapcsolása csak valamelyik irány zöld fázisa után történhet meg.

A feladat kiválóan alkalmas a sorrendi folyamatábra alapján történő programozás gyakorlására.

A működési leírás alapján a sorrendi folyamatábra:



A sorrendi folyamatábra alapján a vezérlőprogram könnyen elkészíthető létradiagramos illetve utasításlistás formában:

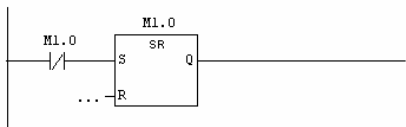
**Network 1:** Title:

Comment:



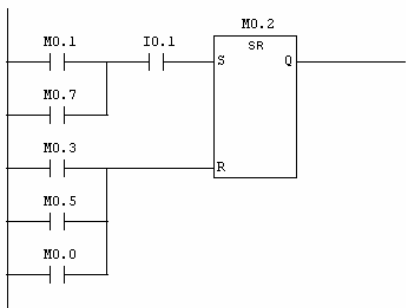
**Network 2:** Title:

Comment:



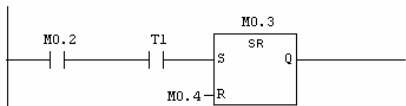
**Network 5:** Title:

Comment:



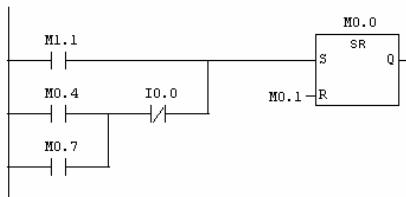
**Network 6:** Title:

Comment:



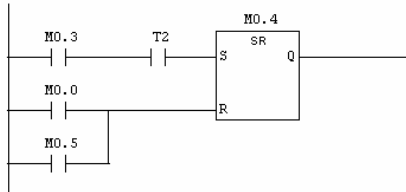
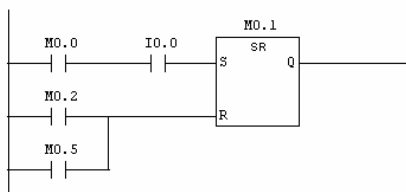
**Network 3:** Title:

Comment:



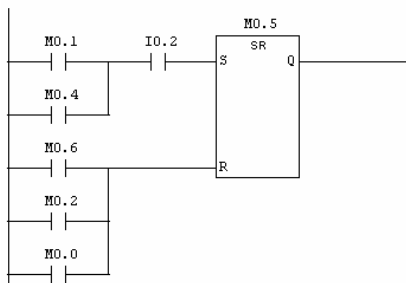
**Network 4:** Title:

Comment:



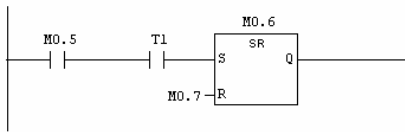
**Network 8:** Title:

Comment:



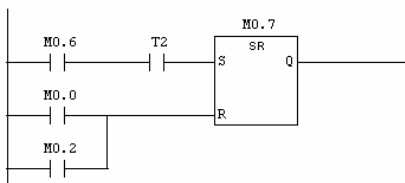
**Network 9 : Title:**

Comment:



**Network 10 : Title:**

Comment:



**Network 13 : Title:**

Comment:



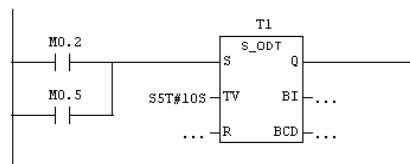
**Network 14 : Title:**

Comment:



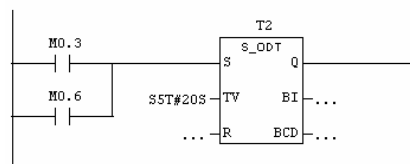
**Network 11 : Title:**

Comment:



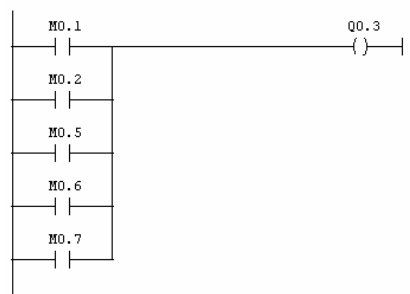
**Network 12 : Title:**

Comment:



**Network 15 : Title:**

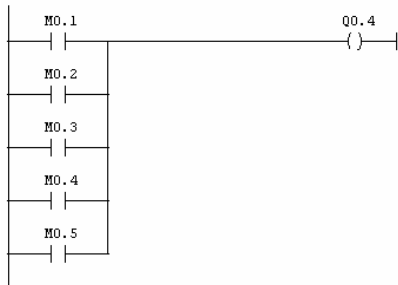
Comment:





**Network 16 : Title:**

Comment:



**Network 1 : Title:**

Comment:

```

AN M 1.0
= M 1.1
    
```

**Network 4 : Title:**

Comment:

```

A M 0.0
A I 0.0
S M 0.1
A(
O M 0.2
O M 0.5
)
R M 0.1
NOP O
    
```

**Network 2 : Title:**

Comment:

```

AN M 1.0
S M 1.0
NOP O
NOP O
    
```

**Network 6 : Title:**

Comment:

```

A M 0.2
A T 1
S M 0.3
A M 0.4
R M 0.3
NOP O
    
```

**Network 3 : Title:**

Comment:

```

A(
O M 1.1
O
A(
O M 0.4
O M 0.7
)
)
AN I 0.0
)
S M 0.0
A M 0.1
R M 0.0
NOP O
    
```

**Network 5 : Title:**

Comment:

```

A(
O M 0.1
O M 0.7
)
A I 0.1
S M 0.2
A(
O M 0.3
O M 0.5
O M 0.0
)
)
R M 0.2
NOP O
    
```

**Network 7 : Title:**

Comment:

```

A M 0.3
A T 2
S M 0.4
A(
O M 0.0
O M 0.5
)
)
R M 0.4
NOP O
    
```

11/11/2011

**Network 8 : Title:**

```

Comment:
A(
O M 0.1
O M 0.4
)
A I 0.2
S H 0.5
A(
O M 0.6
O M 0.2
O M 0.0
)
R M 0.5
NOP 0
    
```

**Network 10 : Title:**

```

Comment:
A M 0.6
A T 2
S M 0.7
A(
O M 0.0
O M 0.2
)
R M 0.7
NOP 0
    
```

**Network 11 : Title:**

```

Comment:
A(
O M 0.2
O M 0.5
)
L S5T#10S
SD T 1
NOP 0
NOP 0
NOP 0
NOP 0
    
```

**Network 9 : Title:**

```

Comment:
A M 0.5
A T 1
S H 0.6
A M 0.7
R M 0.6
NOP 0
    
```

**Network 15 : Title:**

```

piros lámpa
O M 0.1
O M 0.2
O M 0.5
O M 0.6
O H 0.7
= Q 0.3
    
```

**Network 16 : Title:**

```

piros lámpa
O M 0.1
O M 0.2
O M 0.3
O M 0.4
O M 0.5
= Q 0.4
    
```

**Network 12 : Title:**

```

Comment:
A(
O M 0.3
O M 0.6
)
L S5T#20S
SD T 2
NOP 0
NOP 0
NOP 0
NOP 0
    
```

**Network 13 : Title:**

```

zöld lámpa
O M 0.3
O M 0.4
= Q 0.1
    
```

**Network 14 : Title:**

```

zöld lámpa
O M 0.6
O M 0.7
= Q 0.2
    
```

## 11. Feladat

Készítsük el egy csomagológép vezérlőprogramját!

A működés jellemzői:

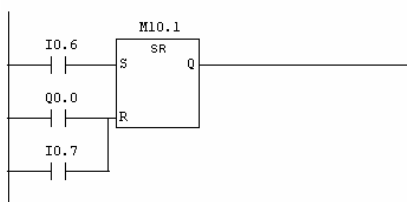
- A folyamat egy nyomógomb megnyomásával indítható (N1 >> I0.6).
- Egy dobozba pontosan 10 db terméket kell elhelyezni. A doboz megteltét egy jelzőlámpa jelzi (H1 >> Q0.0) és ezzel a folyamatnak vége.
- A termékek 2 irányból érkeznek, érkezésüket az E1 (I0.1) és E2 (I0.2) érzékelők érzékelik.

- A folyamatot az N1 nyomógomb ismételt megnyomásával lehet indítani.
- Az N2 Reset funkciójú nyomógombbal a folyamat bármikor leállítható illetve az alaphelyzetbe hozható (N2 >> IO.7).

Egy lehetséges megoldás létradiagramos illetve utasításlistás formában aritmetikai összegző alkalmazásával:

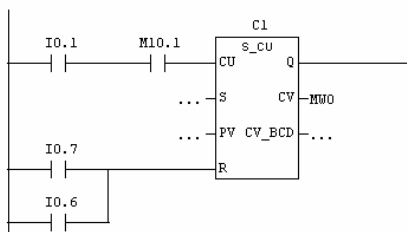
Network 1: Title:

Comment:



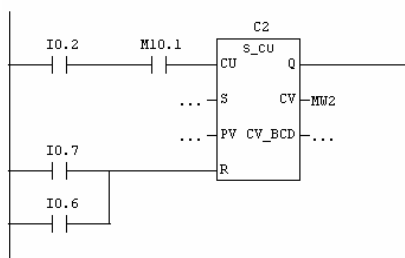
Network 2: Title:

Comment:



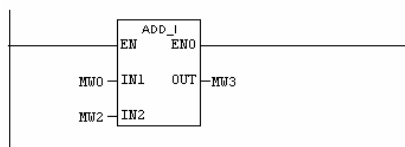
Network 3: Title:

Comment:



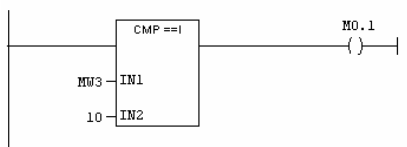
Network 4: Title:

Comment:



Network 5: Title:

Comment:



Network 6: Title:

Comment:



**Network 1 : Title:**

Comment:

```

A   I   0.6
S   M   10.1
A(
O   Q   0.0
O   I   0.7
)
R   M   10.1
NOP 0

```

**Network 2 : Title:**

Comment:

```

A   I   0.1
A   M   10.1
CU  C   1
BLD 101
NOP 0
NOP 0
A(
O   I   0.7
O   I   0.6
)
R   C   1
L   C   1
T   MW  0
NOP 0
NOP 0

```

**Network 3 : Title:**

Comment:

```

A   I   0.2
A   M   10.1
CU  C   2
BLD 101
NOP 0
NOP 0
A(
O   I   0.7
O   I   0.6
)
R   C   2
L   C   2
T   MW  2
NOP 0
NOP 0

```

**Network 4 : Title:**

Comment:

```

L   MW  0
L   MW  2
+I
T   MW  3
NOP 0

```

**Network 5 : Title:**

Comment:

```

L   MW  3
L   I   10
==I
=   M   0.1

```

**Network 6 : Title:**

Comment:

```

A   M   0.1
=   Q   0.0

```

A következő feladatok vezérlőprogramjának elkészítését gyakorlási célból javasoljuk.

Készítsen vezérlőprogramot az alábbi feladatokhoz!

**12. Feladat**

Egy kétállású kapcsolóval lehessen egy lámpa villogási frekvenciáját változtatni 2 rögzített érték között.

**13. Feladat**

Kétkezes motorvezérlés: A két nyomógombot 0,5s időtartamon belül kell megnyomni és folyamatosan nyomva tartani. A kimenet addig aktív, ameddig a 2 nyomógombot nyomva tartjuk. Bármelyik nyomógomb elengedése esetén újból kell kezdeni a folyamatot.

**14. Feladat**

Nyomógombos motorvezérlés üzemidő számlálóval. (Pl. másodperces pontossággal határozzuk meg egy motor üzemidejét.)

### 15. Feladat

Nyomógombos motorvezérlés indítás-számlálóval. (Számológó segítségével figyeljük a motor indításainak a számát.)

### 16. Feladat

Nyomógombos motorvezérlés indítás-számlálóval és tiltással. (Pl. 5 indítás után ne lehessen többször indítani a motort.)

### 17. Feladat

Nyomógombos motorvezérlés üzemidő számlálóval és tiltással. (Pl. összesen 2 percnyi üzemidő után ne lehessen többször indítani a motort.)

### 18. Feladat

Nyomógombos motorvezérlés indítás és üzemidő számlálóval és tiltással. (Pl. adott időtartamban történt 5 indítás után ne lehessen többször indítani a motort.)

### 19. Feladat

Nyomógombos forgásirányváltós motorvezérlés üzemidő számlálóval és tiltással. (Pl. az egyik forgásirányban 2 perc üzemidő után ne lehessen többször indítani a motort, a másik irányban nincs korlátozás.)

### 20. Feladat

Nyomógombos forgásirányváltós motorvezérlés indítás-számlálóval és tiltással. (Pl. az egyik forgásirányban csak kétszer lehessen indítani a motort, a másik irányban nincs korlátozás.)

### 21. Feladat

Egy lámpa villogási frekvenciáját az N1 nyomógombbal növelni, az N2 nyomógombbal csökkenteni lehessen pl. 0,1 s értékkel.

### 22. Feladat

3 db kézi ürítésű tartályt (1-2-3) 3 csővezetéken keresztül mágnesszelepek segítségével töltünk. Minden tartály rendelkezik egy alsó és felső szintérzékelővel. Egyszerre csak egy tartályt tölthetünk. A töltési sorrendet a tartályok leürülési sorrendje határozza meg, pl. ha 2-1-3 volt a tartályok leürülési sorrendje, akkor ugyanilyen sorrendben kell a töltést is elvégezni (egyszerre csak egy tartály tölthető).

### 23. Feladat

Három motor működését kell egy jelzőlámpa segítségével jelezni. A jelzőlámpa folyamatosan világít, ha 2 vagy 3 motor üzemel. A lámpa lassan villog, ha csak 1 (bármelyik 1) motor üzemel, és gyorsan villog, ha egy motor sem működik.

### 24. Feladat

Egy ajtó automatikus működtetését kell biztosítani. Az ajtó min két oldalán egy-egy mozgásérzékelő (E1, E2) található, az ajtó véghelyezeteit végállaskapcsolók (S1, S2) érzékelik. Amint az E1 vagy E2 mozgásérzékelők egyike személyt érzékel, akkor az ajtót nyitni kell. Ha a két mozgásérzékelő érzékelési területe egy adott időre szabad, akkor az ajtót zárni kell.

### 25. Feladat

Három nyomógomb (Ny1, Ny2 és Ny3) segítségével készítsen „kódzárát” az alábbiak szerint:

- Csak 3 gombnyomás fogadható el, a helyes sorrend Ny2-Ny1-Ny3.
- Helytelen sorrend vagy 3-nál több gombnyomás esetén 5s-ig újabb próbálkozást nem szabad elfogadni, még helyes kombinációt sem! A letiltás idején az L1 jelzőlámpa villogjon (0,5s-0,5s időzítéssel).
- 2 gombnyomás között legfeljebb 2s idő telhet el, ellenkező esetben 5s-ig újabb próbálkozást nem szabad elfogadni, még helyes kombinációt sem! A letiltás idején az L1 jelzőlámpa villogjon (0,5s-0,5s időzítéssel).

A megadott feltételek teljesülése esetén a „zárát” ki kell nyitni (PLC kimenet magas szintre állítása) 5s időtartamra.

## Irodalomjegyzék

- [1] Telkes Z.: *Az irányítások generációs tulajdonságai* (1. rész), Budapest, Elektronet, 1996/6.
- [2] Ajtonyi I., Gyuricza I.: *Programozható irányítóberendezések, hálózatok és rendszerek*. Műszaki Könyvkiadó, Budapest, 2002.
- [3] Szabó G.: *Programozható logikai vezérlők*. Budapesti Műszaki Egyetem, Közlekedésmérnöki Kar, Budapest, 1995.
- [4] Klöckner-Moeller Hungária Kft.: *Automatizálási rendszerek hálózatba kapcsolása*. Klöckner-Moeller Hungária Kft. Budapest, 1997.
- [5] Kóbor J.: *Programozható logikai vezérlők programozása*. Széchenyi István Egyetem, Győr, 2002.
- [6] Gilles Michel: *Programmable Logic Controllers*. John Wiley & Sons, 1990
- [7] Ian G. Warnock: *Programmable Controllers*. Prentice Hall, New York, 1988
- [8] K. Lechner, B. Staufer: *Rugalmas automatizálás*. Műszaki Könyvkiadó, Budapest, 1988.
- [9] Demmel, Molnár, Török, Vágvolgyi: *Programozható logikájú vezérlések*. Budapesti Műszaki Egyetem, Mérnöki Továbbképző Intézet, Budapest, 1983.
- [10] Siemens SIMATIC S7-300 Programmable Controller System Manual, 2000
- [11] Siemens LOGO! Ipari és háztartási alkalmazások, Siemens AG 2000
- [12] Hodossy L.: *Mikroprocesszoros rendszerek biztonságtechnikai alkalmazási lehetőségei anyagmozgatási folyamatok automatizálásában*. Egyetemi doktori értekezés, Budapest, 1995.
- [13] Hodossy L.: *Programozható logikai vezérlők*. Oktatási segédlet, Széchenyi István Egyetem, Győr, 2004.

## Tárgymutató

### A, Á

Ábrázolási réteg 92  
Adatkapcsolati réteg 92  
Adatvédelmi módszerek 86  
Alapsávú eljárás 89  
Alkalmazási réteg 92  
Átjáró (gateway) 94  
Átviteli közeg 77  
Átviteli módok 89

### B

Bemeneti és kimeneti egységek 22  
bitműveletek 53  
bitszervezésű PLC 27  
Buszhozzáférési eljárások 79  
Buszprotokoll 90  
Buszstruktúra 76

### C

ciklikus működés 64  
ciklikus programvégrehajtás 69  
ciklusidő és a reakcióidő 68  
Ciklusidő és reakcióidő 65  
CRC – Cyclic Redundancy Check,  
ciklikus redundanciakód 88

### Cs

Csillagstruktúra 75  
CSMA/CA 80  
CSMA/CD 79

### D

Data Unit 84  
DCS 12

### E, É

előforduló műveletek 52  
Első generációs irányítási  
rendszerek 15

### F

Fastruktúra 77  
Fénykábel (fiber-optic cable) 78  
Fizikai vagy bitátviteli réteg 91  
Funkcióterves programozás 56

### G

galvanikus elválasztás 40  
Grafcet 56

### Gy

Gyűrűstruktúra 76

### H

Hálózati elemek 93  
Hálózati réteg 92  
Hálózati topológiák 74  
Harmadik generációs irányítási  
rendszerek 16  
Header 84  
Hibrid hozzáférési eljárások 83  
Híd (Bridge) 93  
Hierarchikus irányítás 10

### I, Í

interpreter 50  
*interpreter funkció* 47  
irányítási folyamat 7  
irányítástechnika alapfogalmai 7  
ISO/OSI modell 90

### J

Jelfogós vezérlések 12  
Jelismétlő (repeater) 93

### K

Koaxiális kábel 78  
Kombinált paritás 88  
kommunikációs hierarchia 73



*kommunikációs vonalak* 47

Kommunikációvezérlő réteg 92  
központi feldolgozó egység 22

## L

lépésorientált működés 63  
Létradiagramos programozás 54  
LRC – Logitudinal Redundancy  
Check, hosszparitás 87  
LSB 85

## M

Második generációs irányítási  
rendszerek 15  
Master/slave (összesítő keret) 82  
Master/Slave (polling) 82  
maximális reakcióidő 66  
MERKER-memória 46  
Mikroprocesszor alapú PLC 33  
mintavételezési frekvencia 67  
MSB 85

## N

Negyedik generációs irányítási  
rendszerek 16

## O, Ó

OB1 szervezőblokk 48

## Ö, Ő

*önteszt funkció* 47  
Ötödik generációs irányítási  
rendszerek 17

## P

Párhuzamos busz 74  
PLC 12  
PLC funkcionális egységei 21  
PLC programnyelvek 51  
PLC RAM térképe 46  
PLC-SCADA 12

processzor állapotai 33

## R

relével megvalósított kimeneti  
fokozat 44

## S

SCADA 12  
Sodrott érpár 78  
Soros busz 74  
Sorrendi folyamatábra 57  
Sorrendi folyamatábrás  
programozás 56  
*Státuszó-generálás funkció* 47  
Strukturált programszerkesztés 65

## Sz

Számítógépes alapú vezérlések 13  
Számlálók, időzítők 44  
Széles sávú eljárás 89

## T

tápegység 22  
Telegram felépítése 83  
Token Bus 81  
Token Passing 80  
Token Ring 81  
Továbbítási réteg 92  
Trailer 85  
tranzisztoros kimeneti fokozat 43  
triakos kimeneti fokozat 43

## U, Ú

Utasításlistás programozás 52  
Útválasztó (router) 94

## V

Vezeték nélküli átvitel 78  
VRC – Vertical Redundancy  
Check, keresztparitás 86