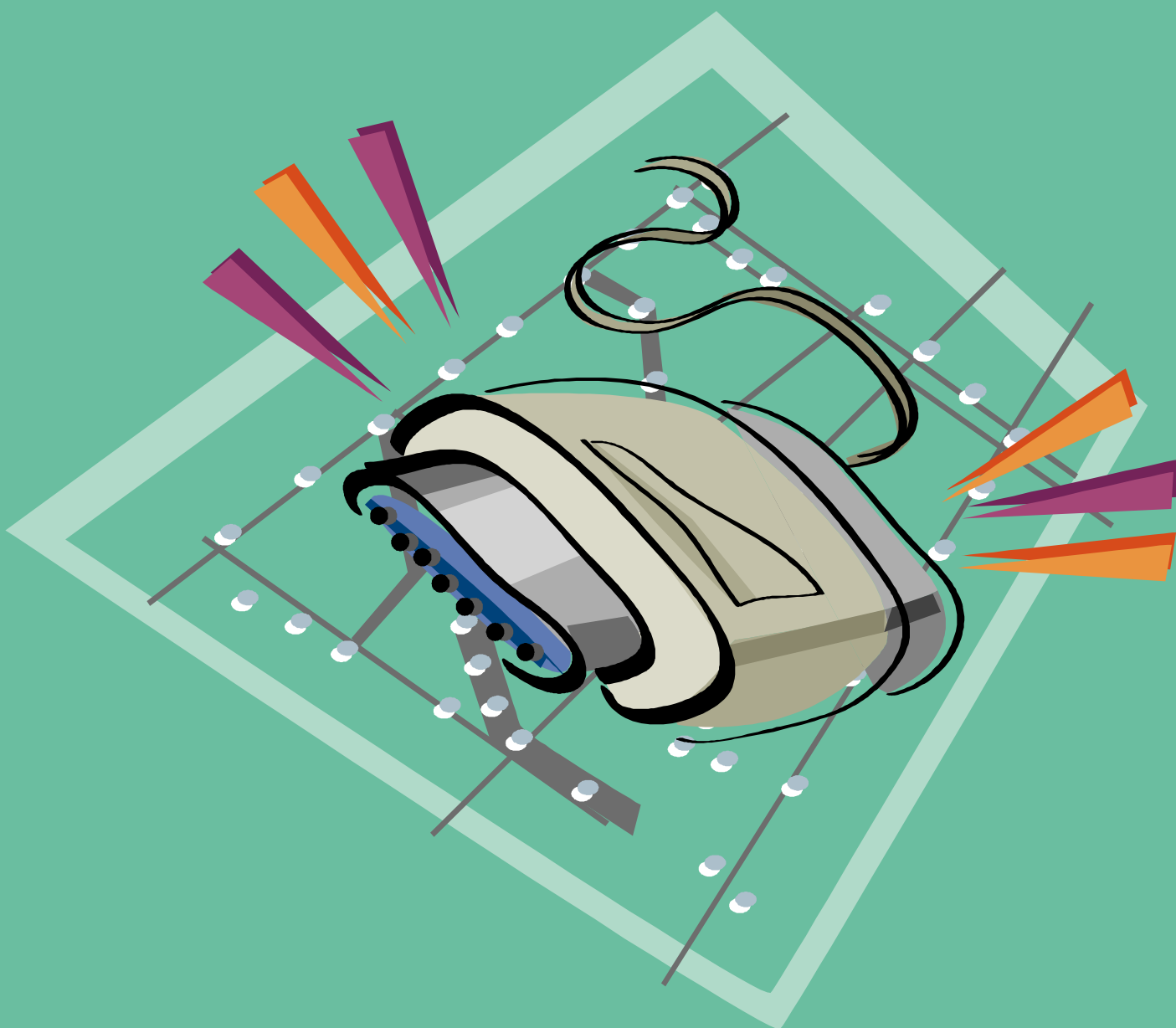


Maczik Mihály András

PLC ismeretek és példatár

IMO iSmart



Maczik Mihály András: PLC ismeretek és példatár

Lektorálta: Simén Gábor

Kiadja: Tech-Con Hungária Kft.

Felelős kiadó: Puha Imre ügyvezető

2012

1.	PLC ISMERETEK	1
1.1	Bevezetés.....	1
1.2	Történeti áttekintés	2
1.3	A PLC-k működésének alapjai.....	4
1.3.1	Villamos jelek.....	4
1.3.2	Digitális jelek	4
1.3.3	Analóg jelek	5
1.3.4	Frekvenciajelek.....	6
1.4	PLC-k fizikai alkotóelemei.....	7
1.4.1	Kiépítési forma	7
1.4.2	Tápegység	8
1.4.3	Fizikai I/O felületek	8
1.4.4	Távoli I/O felületek.....	11
1.4.5	Speciális I/O - felületek.....	11
1.4.6	Processzor (CPU).....	11
1.4.7	Memória.....	12
1.4.8	Kommunikáció	12
1.4.9	Programozó kábel	12
1.4.10	Visszajelző LED-ek	12
1.4.11	Ember-gép kapcsolat.....	13
1.5	PLC-k programozása	14
1.5.1	Utasításlista (Instruction List-IL).....	14
1.5.2	Létradiagram (Ladder Diagram-LD).....	15
1.5.3	Funkcióblokk diagram (Function Block Diagram-FBD).....	16
1.5.4	Strukturált szöveg (Structured Text-ST)	17
1.5.5	Sorrendi folyamatábra (Sequential Flow Chart – SFC, Grafcet).....	18
1.6	Egy PLC-vel megoldott vezérlés dokumentálása	20
1.6.1	Logikai vezérlések	20
1.6.2	Idővezérelt lefutó vezérlések	22
1.6.3	Összetett vezérlések	22
1.6.4	A biztonságos üzemeltetéssel kapcsolatos tervezési szempontok	23
1.7	Röviden az SMT okos reléről	24
1.8	Az SMT szoftver menürendszere, főbb beállításai létradiagram módban	25
1.8.1	File.....	25
1.8.2	Edit.....	26
1.8.3	Operation.....	27
1.8.4	View.....	31
1.8.5	HELP	32
1.9	Digitális eszközök, azok címei	33
1.9.1	Kontaktusok	33
1.9.2	I, Bemenetek (I01 – I08)	33
1.9.3	X, Bővítő modul bemenetek (X01 – X0C).....	33
1.9.4	Z, Billentyűzet bemenet (Z01 – Z04).....	33
1.9.5	Q, Kimenetek (Q01 – Q08)	33
1.9.6	Y, Bővítő modul digitális kimenetek (Y01 – Y0C)	34
1.9.7	M, Segédrelék (M01 - M3F).....	34
1.9.8	M, Különleges segédrelék (M31 – M33).....	34

1.9.9	N, Segédrelék (N típus, N01 – N3F).....	35
1.9.10	D, Differenciáló kontaktus (D típus).....	35
1.9.11	T, Timer, időzítő utasítás (T01 – T1F).....	36
1.9.12	C, Counter, számláló utasítások, (C01 – C1F).....	39
1.9.13	R, Real Time Clock, valós idejű óra utasítások (R01 – R1F).....	42
1.9.14	G, Comparator, komparátor, összehasonlítás (G01 – G1F).....	42
1.9.15	H, (Human-Machine Interfaces, ember-gép kapcsolat) kijelző utasítások	44
1.9.16	P, Pulse Width Modulation, impulzus-szélesség moduláció kimenet (P01 – P02).....	45
1.9.17	L, Link, adatkapcsolat kimenet (L01 – L08)	Hiba! A könyvjelző nem létezik.
1.9.18	S, SHIFT, elcsúsztatott kimenet (S01).....	51
1.10	Analóg eszközök	52
1.10.1	A, Analóg bemenet (A01 – A08).....	52
1.10.2	V, Analóg bemenő paraméter (V01 ~ V08).....	52
1.10.3	AT, Hőmérsékletmérő bemenet, (AT01 ~ AT04).....	52
1.10.4	AQ, Analogue Output, analóg kimenet (AQ01 – AQ04)	52
1.10.5	AS, Add-Subtract, összeadás, kivonás (AS01 – AS1F).....	52
1.10.6	MD, Multiplication - Division, szorzás, osztás (MD01 – MD1F).....	52
1.10.7	PID, arányos, integráló differenciáló szabályzó (PID01 – PID0F)	53
1.10.8	MX, multiplexer(MX01 – MX0F).....	53
1.10.9	AR, Analog Ramp, analóg rámpa (AR01 – AR0F).....	53
1.10.10	DR, Data Register, adattároló (DR01 – DR0F).....	53
1.10.11	MU, Modbus (MU01 – MU0F)	53
2	PÉLDATÁR	56
2.1	Gyakorló példák programozáshoz	56
2.1.1	feladat.....	57
2.1.2	feladat.....	57
2.1.3	feladat.....	57
2.1.4	feladat.....	57
2.1.5	feladat.....	58
2.1.6	feladat.....	58
2.1.7	feladat.....	58
2.1.8	feladat.....	59
2.1.9	feladat.....	60
2.1.10	feladat.....	60
2.1.11	feladat.....	61
2.1.12	feladat.....	61
2.1.13	feladat.....	62
2.1.14	feladat.....	62
2.1.15	feladat.....	63
2.1.16	feladat.....	63
2.1.17	feladat.....	65
2.1.18	feladat.....	65
2.1.19	feladat.....	66
2.1.20	feladat.....	67
2.1.21	feladat.....	67
2.1.22	feladat.....	69
2.1.23	feladat.....	71
2.1.24	feladat.....	72
2.1.25	feladat.....	74
2.1.26	feladat.....	75
2.1.27	feladat.....	76
2.1.28	feladat.....	77
2.1.29	feladat.....	78
2.1.30	feladat.....	79
2.1.31	feladat.....	81
2.1.32	feladat.....	83
2.1.33	feladat.....	85

2.1.34	feladat.....	87
2.1.35	feladat.....	88
2.2	Feladatok haladó programozóknak	89
2.2.1	feladat.....	90
2.2.2	feladat.....	92
2.2.3	feladat.....	94
2.2.4	feladat.....	98
2.2.5	feladat.....	100
2.2.6	feladat.....	102
2.2.7	feladat.....	104
2.2.8	feladat.....	106
3	FELHASZNÁLT IRODALOM.....	109

1. PLC ISMERETEK

1.1 Bevezetés

Az irányítástechnika ma már elképzelhetetlen korszerű félvezető alapú berendezések nélkül. Ezen eszközök egyik csoportját nevezzük PLC-knek, azaz programozható logikai vezérlőknek (Programmable Logic Control). Ez az írás rövid áttekintést szeretne adni róluk. Mivel a gyakorlatban sok cég sok típusa fordul elő, ezért itt általános áttekintést szeretnék adni, majd egy kiválasztott típuson (IMO iSmart) keresztül konkrét felhasználói ötleteket, programrészleteket, programokat szeretnék nyújtani a tisztelt olvasónak.

Relé működtetésekor a be- és kimeneti elemek közötti huzalozás határozza meg a vezérlés lefutását. Ezt huzalozott programozású vezérlésnek nevezzük. Ilyen vezérlés felépíthető elektromechanikus, elektropneumatikus, pneumatikus, hidraulikus elemekből, pl. mágneskapcsolókból, relékből. Megvalósítható azonban elektronikus elemekből, pl. digitális kapukból (ÉS, VAGY, NEM) is. A huzalozott programozású vezérlés működését a huzalozás határozza meg. A huzalozott programozású vezérlés a relékből, szelepekből, végállás-érzékelőkből és más elemekből álló kapcsolási rajz, logikai vázlat, vagy áramút-terv alapján készül. A rendszer a bemeneti változásokat ugyanabban a pillanatban átviszi a kimenetre. Ekkor párhuzamos jelfeldolgozásról beszélünk. PLC-vel irányított vezérlés esetén a működés soros feldolgozású, azaz a vezérlőutasítások végrehajtása egymás után programsoronként történik. Ha a program sok lépésből áll, akkor a tárolt programú vezérlés reakcióideje hosszabb lehet, mint a huzalozott vezérlésé.

Ha a vezérlésben bármi változtatást kell alkalmazni, nem kell a vezetékelést megbontani, elég csak új programot tölteni a PLC-be. Ezen kívül könnyebben megoldható a vezérlés paramétereinek üzem közbeni állítása, visszajelzése, naplózása, értékelése. Sokkal összetettebb vezérlések hozhatók így létre egy közös felügyelet alatt, a kezelés, karbantartás is egyszerűsödik. A teljesség igénye nélkül szeretnék felsorolni néhány gyártót, forgalmazót:

- DELTA
- FESTO
- KLÖCKNER MOELLER
- IMO
- MITSUBISHI
- MODICON- TELEMECANIQUE
- OMRON
- ROCKWELL (ALLEN-BRADLEY)
- SIEMENS

Hangsúlyozni szeretném, hogy ezek kiragadott példák alfabetikus sorrendben, ezen kívül még számtalan cég található meg a piacon különféle típusú termékeivel.

1.2 Történeti áttekintés

Az első ipari kivitelű vezérlőberendezések az 1960-as években jelentek meg. A PLC (Programmable Logical Controller / programozható logikai vezérlő) elnevezés az 1960-as évek végén jelent meg Amerikában (General Motors). Ezeknek a programozása még nem volt egyszerű, nem voltak szabványos programnyelvek kidolgozva.

Számos PLC, hasonlóan az akkori számítógépekhez, lyukkártyás módszerrel volt programozható, és fizikai méreteiben is inkább egy szekrényre hasonlított. A lyukkártyás módszeren kívül nagyon fontos volt a gépi kódú, valamint az assembly nyelvű programozás.

A számítógépek és a mikroprocesszorok fejlődésével párhuzamosan fejlődtek a PLC-k is. Az 1970-es években jelent meg az egyvonalas áramút-terv alapú programozás, a létradiagram használata, mely a mai napig is uralkodó programozási forma. Az áramút-tervezést eredetileg a relés vezérlőberendezésekhez használták. A relés kapcsolótáblákkal kombinációs és alapszintű sorrendi vezérléseket lehet megvalósítani.

Azonban az egyre bonyolultabb vezérlő algoritmusok, az időzítések, a kommunikáció igényének felmerülése, és nem utolsósorban a javíthatóság, a hibakeresés megvalósítása indokolta, hogy egy új eszköz szülessen, amely magában foglalja az egyszerű bővíthetőség és az általános felhasználhatóság tulajdonságát.

Az 1980-as években - a piaci követelményeknek megfelelően - a PLC-k is tovább fejlődtek. A kétállapotú vagy diszkrét jelek kezelése mellett bevezették az időben folytonos, azaz analóg jel feldolgozását is. Analóg jeleket szolgáltatnak például a hőmérséklet-, távolság-, a szintérzékelő, valamint a sebességmérő érzékelők is. Ez jelentős különbséget eredményezett a PLC előtti vezérlőkhöz képest, hiszen az analóg (0-10 V, 4-20 mA) jeleket digitális feldolgozásra alkalmassá kellett tenni, azaz digitalizálni kellett. A digitalizálás eredményeként az analóg jel a PLC-ben már kettes számrendszernek megfelelő adatsorként jelent meg.

Ezt az értéket azonban a kettes számrendszerben - amely a digitális technika alapja - nem lehet egy bittel leírni, azaz megjelentek a különböző adatábrázolási formák. Az adatokat, például a hőmérsékletértékeket össze kellett hasonlítani egy kívánt vagy előre meghatározott értékkel, amelyet a gyártástechnológia megkövetelt. Így a feldolgozandó jelek és adatok száma nagyobb teljesítményű eszközöket igényelt, tehát a hardver továbblépett a következő szintre, amit a mikrokontrollerek elterjedése nagyban elősegített. Az analóg jelek digitalizálása olyan speciális eszközöket kívánt meg, amelyek ára magas volt, ezért lehetővé tették, hogy külön alkalmazásonként lehessen megvásárolni őket.

A hardver teljesítményének növelésével lehetővé vált a mért adatok vagy üzemidők tárolása a PLC-kben. A PLC memóriájából azonban ezeket az adatokat a termelési statisztikák, illetve a hatékonyság elemzése érdekében egyre gyakrabban kellett kiolvasni, így megszületett az igény a kommunikációs felületek, azaz portok és protokollok használatára. Ennek, és a speciális modulok megjelenések következtében került a mai napig is használatos aszinkron soros kommunikációs port a PLC-re.

A gyártási technológiák komplexitása, a sebesség és a kapacitás növelése érdekében egyre több gyártási folyamatot kellett az automatizálni. Az automatizált részegységek összehangolásához, illetve a teljesen automatizált gyártósorok felügyeletéhez bonyolultabb számításokra, magasabb szintű vezérlés szervezésére volt szükség. Az 1990-es évekre tehető a PLC-k fejlődésének újabb mérföldköve, amikor is megjelentek a nagysebességű processzorok, az adatkapcsolati hálózatok, illetve a magas szintű matematikai funkciókkal és a komplex, több szabályzókörös algoritmusokkal felvértezett, akár több processzorral rendelkező vezérlő berendezések.

A vezérlőszekrények mérete folyamatosan csökkent, a kábelezési távolságokat gazdaságosan át kellett hidalni, így a PLC-k mérete is csökkent. Egyúttal egyre több kommunikációs lehetőséggel rendelkeznek, hogy távoli érzékelők jelei is minél kevesebb kábelezéssel eljuttathatók legyenek a vezérlőbe.

A PLC-k alkalmazása már túllépett a klasszikus ipari felhasználási területeken. Napjainkban egyre jobban növekszik a szerepük az épületautomatizálás, a háztartások területén, és egyéb speciális területeken. A modern, automatizált rendszerekhez, a PLC-khez helyi megjelenítést és beavatkozást biztosító érintőképernyős felületek kapcsolódnak, melyeken keresztül változtathatjuk például egy motor fordulatszámát, a hőmérsékletet, az anyagáramokat, a nyomtatékokat stb. Természetesen ma már nem csak egyszerűen vezérlőkről, hanem sokkal inkább komplex automatizált rendszerekről beszélhetünk, melyek integráltan lefedik a teljes gyártási technológiát, beépülnek a logisztikai, pénzügyi, minőségbiztosítási rendszerekbe. Ezen komplex alkalmazásokat ipari kommunikációs hálózatok fogják össze, melyek feladata az I/O kezeléstől egészen az adatgyűjtésig terjed.

Jelenleg a biztonsági, redundáns kialakítású, komplett megoldású technológiákat irányító automatizálási platformok egyre inkább előtérbe kerülnek a pár jelet lekezelő elszigetelt vezérlők helyett.

1.3 A PLC-k működésének alapjai

A PLC-k ipari környezetben, ipari jelekkel dolgoznak. A végálláskapcsolók, a szintérzékelők, az inkrementális jeladók, a látórendszerek, azaz a különféle szenzorok szolgáltatják a technológiáról az ipari bemeneti jeleket. A technológia irányába az indító jeleket, a sebesség-alapjeleket, engedélyezéseket, visszajelzéseket, tehát a kimeneti jeleket a PLC állítja elő.

1.3.1 Villamos jelek

Az ipari jelek és jelszintek jól meghatározott feszültség szintek, illetve áramértékek. Ezek a jelek az egész világon kisebb eltérésekkel egységesnek tekinthetők. Ily módon a PLC-k az egész világon univerzálisan alkalmazhatók. Az iparban használt villamos technológiai jelek lehetnek digitálisak, analóg- és frekvenciajelek.

1.3.2 Digitális jelek

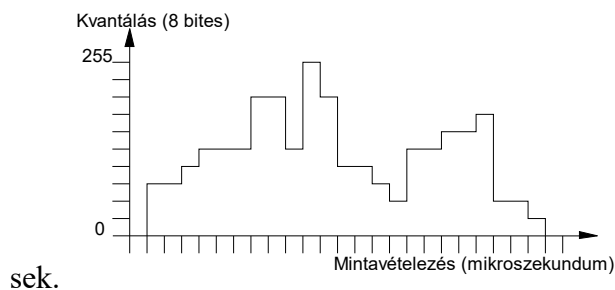
A digitális, azaz kétállapotú vagy diszkrét jelek leggyakrabban kapcsolóktól, nyomógomboktól érkező nyitó vagy záró kontaktusok. Az esetek túlnyomó többségében ezek a jelek 0-24 V egyenfeszültséget szolgáltatnak, amely vagy 0 V vagy 24 V. Ezen túlmenően használatosak még a 115/240 V váltófeszültséggel működő digitális jelek is. Mivel egy kontaktus vagy nyitott, vagy zárt állapotú, ezért ezekhez az állapotokhoz a logikai 1 és a logikai 0 jelet rendeljük.

Amennyiben pozitív logikát használunk, azaz alapesetben a kontaktus nyitott (NO - Normally Open) a 0 V-hoz a logikai 0 érték, míg a zárt kontaktushoz a 24 V tartozik, amelyet a logikai 1 értékkel jelölünk. Ellenkező esetben, negatív logika használatakor, amikor alapesetben a kontaktus zárt (NC - Normally Close) a 24 V egyenfeszültséghez értjük a logikai 0 állapotot, míg értelemszerűen a 0 V jelenti a logikai 1 állapotot.

Digitális bemenetre jó példa lehet egy villanykapcsoló, amelynek bekapcsolt állapota 1, kikapcsolt állapota 0. Digitális kimenet például egy villanymotor, amelynek bekapcsolt állapota 1, kikapcsolt állapota 0.

1.3.3 Analóg jelek

Az analóg jelek nem ábrázolhatók 0 vagy 1 logikai értékkel. Az analóg jelek időben folytonosak, és különböző értékeket vesznek fel. Az analóg jeleket a PLC számára feldolgozhatóvá kell tenni, azaz digitalizálni kell. A digitalizálás időbeli mintavételezést és kvantálást jelent. Az analóg jelek időbeli mintavételezése adja meg, hogy milyen gyakran mérjük meg a jel pillanatnyi értékét, és ez jellemzően néhány mikroszekundum nagyságrendű. A kvantálás a megmért pillanatnyi érték abszolút értékét adja meg, ez 8 biten 0...255, 10 biten 0...1023, 12 biten 0...4095 vagy 16 biten 0...65535 értéket jelent. Ezek az értékek az analóg-digitális jelátalakítótól függenek. A jelátalakítók tipikusan rendre 8, 10, 12, 16 vagy 32 bite-



Az iparban használt analóg jelértékek tipikusan 0-10 V egyenfeszültség, 4-20 mA vagy 0-20 mA egyenáram, ritkábban -10...0...10 V egyenfeszültség. Az analóg szenzorok jeleit ezekre a jelszintekre a távadók illesztik értékben és impedanciában. Minél pontosabban kell egy analóg jelet megmérni, annál nagyobb felbontású, azaz kvantálási bitszámú analóg-digitális jelátalakítóval rendelkező PLC-re vagy PLC modulra van szükség. A nagyobb felbontású analóg modulok általában nagyobb átalakítási idővel rendelkeznek. Amikor a technológiát automatizáljuk, a működési környezet ismeretén kívül nagyon fontos még a megfelelő felbontású és sebességű modul kiválasztása. Zajos környezetben, ahol a távadóból érkező jel nem elég stabil, nagy felbontású kártyával hamis jelértékeket is mérhetünk. Ez kihathat a teljes szabályzásra is. Például egy 8 bites analóg kártya a 40 mV alatti ingadozásokra érzéketlen, amíg egy 16 bites kártya 40 mV-os tartományban megközelítőleg 260 különböző értéket különböztet meg.

A 4-20 mA tartományba eső analóg jelek lehetőséget biztosítanak a kábelszakadások, a hibás szenzorok vagy távadók érzékelésére. Ugyanis az analóg modul a 4 mA áramerősséghez rendeli a 0 értéket, és a 20 mA-hez a legmagasabb értéket (255 - 8 bit, 1023 - 10 bit, 4095 - 12 bit, 65535 - 16 bit esetén), felbontástól függően. Kábelszakadás esetén, mikor megszűnik az áram, a fizikailag mért érték 4 mA alá, általában 0 értékre esik, ez egyértelműen hibás szen-

zorjelet jelent. Ha tehát a PLC-ben nem jelenik meg a mért érték, de egyéb úton tapasztalható vagy mérhető, akkor érdemes a kábelezést, illetve a szenzort vagy a távadót megvizsgálni, esetleg a távadó feszültségellátását ellenőrizni.

-10...0...10 V egyenfeszültség-jelek mérésekor az első bitet előjel bitként használjuk. Ilyenkor az értékek 8 bites felbontás esetén -64...0...63 decimális értékek között mozognak.

Analóg bementre jó példa egy hőmérő, melyből a hőmérsékletadatokat kapjuk, a mérés pontossága felbontás függő (8, 10, 12... bites). Ez a felbontás azt jelenti, hogy 2^n hatványa a felbontás, vagyis a 8 bites felbontás: $2^8 = 256$, a 10 bites: 1024 stb. Ez azt jelenti, ha a hőmérőnk 0-100 °C-ig mér, akkor ezt a felbontás függvényében egyenlő részre bonthatjuk: 8 bites felbontásnál 256, 10 bites felbontásnál 1024 egyenlő részre osztjuk. Ily módon 8 bites felbontásnál: $\frac{100}{256} \approx 0,4$ °C pontosság, míg 10 bites felbontásnál: $\frac{100}{1024} \approx 0,1$ °C pontosság érhető el.

1.3.4 Frekvenciajelek

A frekvenciajelek jellemzője nem az állapotuk, nem is az értékük, hanem az időbeli gyakoriságuk. Ezek a jelek általában négyszögjelek, amelyek 0-5 V egyenfeszültség között váltakozva szolgáltatnak 0 és logikai 1-es értéket. Ezen jelek mérendő jellemzője a frekvenciájuk, azaz egy másodperc alatti változási gyakoriságuk. A PLC-k általában kHz nagyságrendű jelváltozási sebességet képesek mérni. Számos esetben azonban nagyobb frekvenciák mérését is meg kell oldani, ez már különálló kártyát igényel. Az ilyen kártyákat gyorszámláló moduloknak hívjuk, amelyek akár a MHz-es tartományban is képesek működni. Frekvenciajeleket általában inkrementális jeladók szolgáltatnak.

Például ha egy 8 bites inkrementális forgó jeladó, amely egy szállítószalag tengelyéhez csatlakozik, másodpercenként 2560 impulzusjelet ad, akkor a tengely $\frac{2560}{256} \cdot 60 = 600$ 1/perc fordulatszámmal forog. Ebben a példában másodpercenként 2560 négyszögjelet kell megszámlálni, ami 2560 Hz, vagyis 2,56 kHz-es frekvenciajelet jelent.

A PLC-k kimeneti jelei megegyeznek a bementi villamos jelekkel, azaz digitális záró vagy nyitó érintkezőt, analóg értéket, vagyis 0-10 V, vagy 4-20 mA (0-20 mA), illetve frekvencia jelet képeznek. A PLC-k az érzékelt bemeneti jelek és a gépeket vezérlő kimeneti jelek között teremtenek kapcsolatot. A kimeneti jeleket a bemeneti jelek és a belső programjuk alapján állítják elő. A belső programjuk függhet előző bemeneti vagy kimeneti állapotoktól, előre beállított értékektől, belső relétől (merker), valós időtől, dátumtól, időzítőktől, számláló

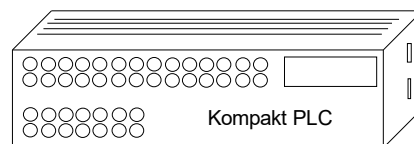
értékektől, reteszfeltételektől, speciális folyamatot lekezelő (megszakítás) programrész eredményétől stb.

1.4 PLC-k fizikai alkotóelemei

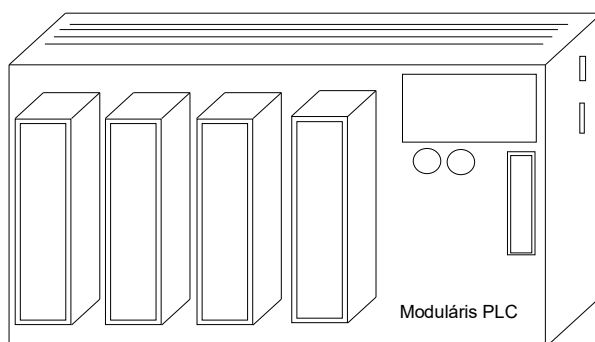
Mivel a PLC-k ipari, technológiai jelekkel dolgoznak, kialakításuk is ennek megfelelő. A PLC-k főbb logikai részei: tápegység, bemenetek, kimenetek (I/O-k), központi egység (CPU), memória, kommunikáció, programozó port, visszajelző LED-ek.

1.4.1 Kiépítési forma

Ennek alapján háromfelé tudjuk a ma fellelhető PLC családokat válogatni. Mindegyikre jellemző, hogy csavarokkal rögzíthető vagy DIN - sínen helyezhető el. Az első csoport a kompakt PLC-k, vagy okos relék családja, ahol minden funkcionális egység egy kisméretű házba van beépítve, és csak minimálisan (I/O egységgel, memóriakártyával) bővíthető, fejleszhető.



A második csoport a moduláris PLC-k családja, ahol van egy alapelemeket (CPU, memória, kommunikáció) tartalmazó keretszekrény, és vannak rajta fiókok (Rack), ahová a különféle bővítő modulok bepattinthatóak. Ez a rendszer közepesen bővíthető, fejleszhető.



A harmadik csoport az alaplapra építhető PLC-k családja. Itt a fő hátlap CPU-k, tápegységek, továbbá bemeneti, kimeneti és speciálismodulok, valamint kommunikációs modulok rögzítésére és csatlakoztatására szolgál. Ehhez szükség szerint kábelekkel bővítő hátlapok csatlakozhatnak, feladatuk a további modulok rögzítésének biztosítása. Ez a rendszer tetszőlegesen bővíthető, fejleszhető.

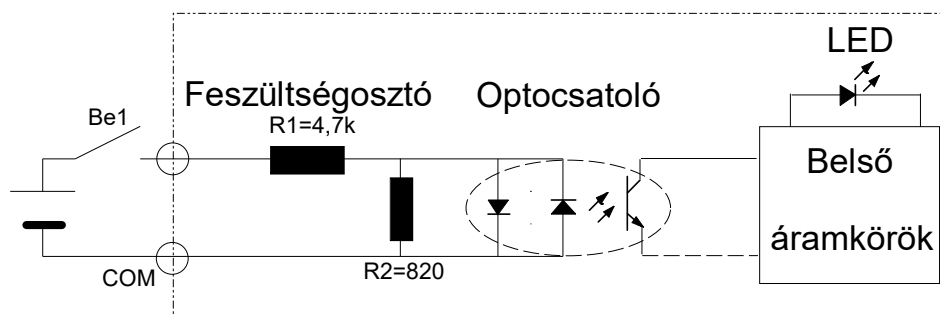
1.4.2 Tápegység

A tápegység az összes PLC részegység energiaellátását biztosítja. A kompakt, a keretes és a hátlapos kivitelű PLC-kben a tápegység helye kötött. Az egyéb PLC-k esetében a tápegységet és a PLC-t kábellel lehet összekötni. Ezek a külső tápegységek alkalmasak a fizikai I/O felületeken elhelyezett érzékelők, beavatkozók tápellátására, valamint a távoli eszközök tápellátására. A tápegységek kiválasztása egyszerű feladat: minden PLC modul teljesítményfelvételét megtaláljuk a katalógusokban. Ezek összege adja a PLC teljesítmény igényét. Ha a távoli eszközöket (érezelők, beavatkozók) is erről az eszköztől szeretnénk táplálni, akkor ezek teljesítményigényét is összegezzük a katalógusadatok alapján. Érdeemes a méretezés során legalább 10% tartalékot képezni. Sok helyen használnak megkettőzött (redundáns) tápegységeket a biztonságos működés érdekében.

1.4.3 Fizikai I/O felületek

Az I/O szó az angol Input (bemenet) és Output (kimenet) szavakból származik. A vezérelt gépekkel való kapcsolattartást az I/O felületek biztosítják, ezen keresztül lehet információkat kapni a technológiákról, és a beavatkozást biztosítani. A fizikai I/O felületeken valósulnak meg az alapvető digitális (kétállapotú) és analóg be- és kimeneti pontok. Ezeken kívül beszélhetünk még speciális I/O felületekről is (számlálók, hőmérsékletmérő bemenetek stb.).

A fizikai I/O felületeket a PLC belső részétől le kell választani. Ennek számos oka lehet. Rendkívül fontos, hogy a technológiában keletkező zavarok, túlfeszültségek, zárlatok ne tessenek kárt a PLC memóriájában, központi egységében (CPU), illetve ne tudjanak továbbterjedni más szenzorokra, beavatkozókra sem. A PLC-k be- és kimeneteit galvanikusan leválasztják nagy átütési szilárdságú szigeteléssel. A bemenetek esetében ez általában optikai leválasztást jelent (optocsatolók).



Az optikai csatolók közös pontját általában csoportonként kivezetik. A kivezetett közös pontokat vagy a +24 V egyenfeszültségre (NPN logika), vagy a 0 V-ra (PNP logika) kapcsolják. A bemeneti pontokra általában kontaktusok csatlakoznak, amelyek tápellátását külső táp-

egységgel vagy a PLC saját tápegységével oldhatjuk meg. Általában javasolt a külső, a PLC saját tápegységétől független áramforrás használata.

A kimenetekkel szemben támasztott legfőbb elvárások a kapcsolási szám, a kapcsolási sebesség és a kapcsolt teljesítmény. Ennek megfelelően a kétállapotú kimenetek lehetnek tranzistorosak, relések, szilárdtest-relések vagy triakosak. A tranzistoros kimenetek elméletileg végtelen számú kapcsolásra képesek, a megadott paraméterek között. Kapcsolási sebességük elérheti a kHz-es tartományt is, viszont általában legfeljebb 100 mA-rel terhelhetők és polaritás függők. Továbbá ügyelni kell a terhelés jellegére is, mivel a tranzistorok érzékenyek az induktív terhelésekre (villanymotorok tekercsei), ezért a tranzistoros kimeneteket védeni kell! A terhelhetőség-növelés érdekében, gyártótól függően, lehetőség nyílik több tranzistoros kimenet összekötésére, ilyen esetekben azonban a PLC- programban nagy figyelmet kell fordítani az összekötött kimenetek kezelésére.

A kimenetek gyakran használatosak lámpák, szelepek kapcsolásához, ilyenkor számos esetben igényelhetnek relés külső leválasztást. A relés kimenetek kiváló galvanikus leválasztást valósítanak meg, nagymértékben terhelhetőek, általában kimeneti pontonként 1-2 A-rel, a csoportok közös pontjain akár 8 A-rel is. A kimeneti pontok csoportokba rendezésével elérhető, hogy a különböző csoportok más-más feszültség szinten működjenek - ennek ellenére javasolt a relés kimenetek védelme is. Ezen kívül érzéketlenek az induktív terhelésekre, és polaritás függetlenek. A mozgó alkatrészek következtében a kapcsolási élettartalmuk rövidebb, beégésre hajlamosak, és sebességük a másodpercenkénti 5-10 kapcsolást nem haladja meg. Relés kimenetekkel nem nagy kapcsolás-igényű, általában váltóáramú rendszerek közvetlen meghajtását valósítjuk meg.

A szilárdtest relé egy jelfogóként működő félvezető lapka. A szilárdtest relé feladata, hogy nagyobb feszültséget, nagyobb áramerősséget viseljen el, mint egy egyszerű félvezető. Egy megfelelő szilárdtest relé például 5-9-12 V feszültséggel (tehát az általános digitális áramkörök feszültség szintjével) akár 230 V-os váltakozó feszültséget és több amperryi áramot tud vezérelni. A hagyományos elektromágneses vezérlésű (mechanikus) jelfogó összehasonlítása a „szilárdtest” jelfogóval:

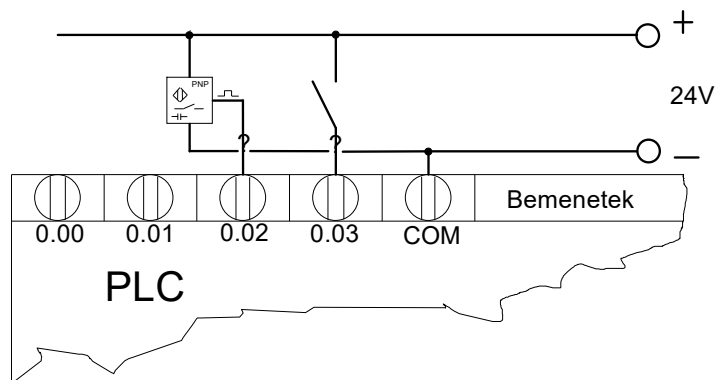
- A relé sok mozgó alkatrészt tartalmaz. A szilárdtest relé semmiféle mozgó elemet nem tartalmaz. Jobban bírja a rázkódást, véletlenül sem kapcsol ilyenkor hibásan.
- Egy hagyományos relénél egy szilárdtest elem élettartama lényegesen nagyobb (a kapcsolások számát tekintve, mivel nem tartalmaz mozgó alkatrészt), azonban az impulzusszerű túlterheléseket rosszabbul viseli.

- A hagyományos relé tizedmásodpercet is elérő késleltetéséhez képest a szilárdtest relé jelentősen gyorsabb.
- A hagyományos relék a kapcsolás pillanatában nagyfrekvenciás zajt is állítanak elő, amely a szilárdtest relénél nem jelentkezik.
- A szilárdtest reléknél nem jelentkezik a prell jelenség.
- A hagyományos jelfogókkal szemben hűtést igényelhetnek, ami kicsiny méretük, ill. kapcsolt teljesítmény nagyságától függően szükséges lehet.
- A hagyományos relék galvanikus leválasztás biztosítanak, míg a szilárdtest relék túlfeszültség hatására könnyebben átütnek (ez a veszély optocsatolással elhárítható).

A triakos kimenetek ötvözik a tranzistoros és relés kimenetek előnyeit. Elméletileg korlátlan a kapcsolási élettartalmuk, viszonylag nagy sebességgel képesek nagy áramok kapcsolására, mindemellett kevésbé érzékenyek az induktív terhelésekre, polaritás függetlenek.

A fizikai I/O felületeken kívül léteznek a kihelyezett sorkapcsok is, amelyek a PLC I/O - kártyáihoz - egy kábelkorbácson keresztül - közvetlenül csatlakoznak, de akár több 10 méter távolságban is lehetnek a PLC mellett lévő I/O - kártyájától.

Íme, egy példa egy passzív mechanikus záró érzékelő és egy aktív PNP kapacitív záró érzékelő bemeneti bekötésére:



1.4.4 Távoli I/O felületek

Funkciójukat tekintve megegyeznek a fizikai I/O - felületekkel, abban viszont eltérnek, hogy fizikailag nem a processzor mellett, hanem attól messzebb, akár több 100 méteres távolságban helyezkednek el, ipari kommunikációs hálózaton keresztül kommunikálnak. Egyik altípusuk a terepi I/O amely szélsőséges környezeti körülményeknek kitett helyeken is felszerelhető, a másik altípusuk a vezeték nélküli I/O egységek, amelyek olyan alkalmazásokban jelentenek megoldást, ahol nem használhatók kábelek, például daruk, automatikusan vezérelt járművek és robotok.

1.4.5 Speciális I/O - felületek

Ide tartoznak az analóg I/O kártyák, a hőelem- vagy hőmérsékletszenzorokat közvetlenül fogadni képes kártyák, a gyorszámláló modulok, szervo hajtásokat szinkronizáló pozícionáló modulok (CNC, robot vezérléséhez), kommunikációs modulok, biztonsági modulok, PID szabályzó modulok stb.

1.4.6 Processzor (CPU)

Ez a PLC motorja, központi egysége. A PLC-kben használatos processzorok alapvetően két csoportba sorolhatóak. Az első a gyors, nagy megbízhatóságú, de csökkentett utasításkészletű, vagyis RISC (Reduced Instruction Set Computer) processzorok, amelyekben a vezérlést direkt áramköri elemek valósítják meg. A másik csoport a lassabb, de rugalmas, számítógépekben is használatos teljes utasításkészletű CISC (Complex Instruction Set Computer) központi egységek, amelyeknél a vezérlést mikroprogram látja el. A mai korszerű központi egységek egy- vagy többprocesszoros formában érhetők el. Indításkor és üzem közben öndiagnosztikai funkciókat is ellátnak: CPU hiba észlelése, lefagyáskor újraindítás, elemhiba észlelése, memóriahiba észlelése, programellenőrzés, tápegység-hiba észlelése. A processzor vagy CPU futatja a vezérlőprogramot, kezeli az I/O - felületeket, a kommunikációs felületeket, adatokat szolgáltat ipari kommunikációs felületeken keresztül.

A processzorok másik fontos ismérve, a bennük futó operációs rendszer, amelyet firmwarenek nevezünk. A firmware határozza meg, hogy a PLC milyen funkciókat képes megvalósítani, és hogy milyen módon programozható. A fejlett PLC-kben a firmware cserélhető, ily módon újabb funkciók vagy programozási utasítások megjelenésekor nem szükséges a PLC rendszer cseréje.

1.4.7 Memória

A processzorokhoz csatlakozik a memória, ettől függ, hogy mekkora rendszerek építhetők ki belőlük, hány I/O - pont kezelésére alkalmasak, mennyi funkció megvalósítására képesek. Található itt RAM a programok futtatására, FLASH a programok tárolására, a teljes memóriakapacitás jellemzően 32 MB-ot. A modern processzorok memóriája kártyával bővíthető, így nehéz határt szabni a megvalósítandó feladatoknak.

1.4.8 Kommunikáció

A kommunikációs modulok a PLC-k egymás közötti, és a PLC-PC együttműködését biztosítják. Ha több hardver hálózati kommunikációja is lehetséges, „multi-point” kommunikációról beszélünk. A kommunikáció protokollokon keresztül valósul meg. A protokoll egy egyezmény, vagy szabvány, amely leírja, hogy a hálózat résztvevői miképp tudnak egymással kommunikálni. Ez többnyire a kapcsolat felvételét, kommunikációt, adat továbbítást jelenti.

Az elsők közül való protokoll az RS422 / RS485 volt, majd ezt a gyártók továbbfejlesztették, ma a PLC-k leginkább az alábbi protokollokat használják: Ethernet, CC-Link, Profibus, DeviceNet, AS-Interface, Melsecnet/H, Modbus, CANopen, SSCNET, Toolbus, SysmacWay...Egyes típusok univerzálisan használhatóak, míg mások csak adott gyártó termékeihez jók.

Két eszköz között (PLC-PC) lehet közvetlen, „pont-pont” kapcsolatot is létesíteni. Ehhez az egyik első protokoll az RS-232C soros kapcsolat volt, ma azonban inkább az USB csatlakozás az elterjedt.

1.4.9 Programozó kábel

A programozó kábelek általában úgynevezett aktív kábelek, ami azt jelenti, hogy tartalmaznak valamilyen elektronikát. Az elektronika átalakítást végez a PC és a PLC között. Tipikusan RS232C a PC oldalán és RS485, vagy TTL jel a PLC oldalán. Ma már gyakoriak a PC oldalán csak USB porttal rendelkező számítógépek, így megjelentek az USB csatlakozóval rendelkező programozó kábelek is.

1.4.10 Visszajelző LED-ek

A PLC-k döntő többségén megtalálhatók az alapfunkciójú visszajelző LED-ek. Ezek a futás (RUN), hiba (ERR) és a kommunikáció (COM). A futás jelzi, hogy a PLC jelenleg futtatja a felhasználói programot. A hiba LED valamilyen hibát jelez, amelyet a PLC érzékelt. A kommunikációs visszajelző általában a programozó kábelén történő adatforgalmat jelzi. Ezen

kívül a gyártótól függően számos egyéb LED is megtalálható a PLC-k központi egységén (akkumulátor-hiba - BAT, bővítő memória hiba - MEM). Továbbá a digitális be- és kimeneteket is ellátják apró LED-ekkel, amelyek segítenek ránézésre eldönteni, hogy a van-e jel a PLC valamelyik bemenetén, illetve a PLC bekapcsolta-e valamely kimenetét.

1.4.11 Ember-gép kapcsolat

A PLC-nek egyes konfigurációs, riasztási, jelentési, vagy mindennapi vezérlési műveletek során szükségük lehet az emberekkel való kapcsolattartásra. Erre a célra az ember-gép kezelőfelületet (Human-Machine Interface röviden HMI) alkalmazzák. HMI-ket úgy is emlegetik, mint MMI (Man Machine Interface) vagy GUI (Graphical User Interface).

Egy egyszerű rendszer nyomógombok és jelzőlámpák segítségével léphet kapcsolatba a felhasználóval.

Alkalmazhatnak egy-két soros monokróm LCD szöveges kijelzőt, amelynek a szélén nyomógombok vannak. Ennek egy speciális típusa a kompakt PLC-kre gyárilag kiépített kezelő - visszajelző felület, ahonnan annak menürendszere segítségével programozni is lehet.

Használhatnak grafikus színes érintőképernyőket is, ezek visszajelző és kezelő ábráit külön szoftverrel kell felprogramozni és illeszteni a PLC-hez.

Az összetettebb rendszerek egy számítógépre telepített programozó és ellenőrző szoftvert használnak, amihez egy kommunikációs kezelőfelülettel csatlakozik a PLC.

1.5 PLC-k programozása

A PLC-k programozása általában személyi számítógépeken (PC) keresztül történik. Azonban előfordulnak még külön erre a célra gyártott programozó készülékek (konzolok) is. A programozáshoz alapvetően szükséges a programozó szoftver, illetve a programozó kábel is. A programozó szoftverek elsősorban Microsoft Windows operációs rendszereken futnak. A programozó szoftverek vagy fejlesztőkörnyezetek általában licenzkötelesek.

Az IEC 61131-3:2003 vagy MSZ-EN 61131-3:2003 az a szabvány, amely meghatározza a PLC-k kötelező programozási nyelveit az alábbi táblázat szerint.

Leírás	Angol rövidítés	Német rövidítés	Megjelenés	Megjegyzés
Utasításlista	IL	AWL	karakteres	Egyszerű sorszámozott utasításokat, címeket, konstansokat tartalmaz, (Assembly)
Létradiagram	LD	KOP	grafikus	Olyan, mint egy elektromos kapcsolási rajz, amely 90 °-al el van forgatva
Funkcióblokk diagram	FBD	FBS	grafikus	Logikai szimbólumokat tartalmaz, különösen a Boole-algebrai feladatok megoldásához jó
Strukturált szöveg	ST	ST	karakteres	Hasonló, mint a magas szintű programozási nyelvek (C, Pascal, Basic)
Sorrendi folyamatábra	SFC	AS	grafikus	Egyfajta összetett folyamatábra (Grafcet)

1.5.1 Utasításlista (Instruction List-IL)

Az utasításlistával történő programozás során a működési egyenleteket előírt formában, soronként kell megadni. Az utasításlista tulajdonképpen a vezérlőutasítások egy sorozata. Beviteli egysége lehet kézi programozó konzol, vagy számítógép szoftver.

A sorszámozás általában automatikusan megtörténik az „ENTER” nyomógomb hatására és „0”-val kezdődik. Az utolsó sorszámu utasítás a programot lezáró „END” kell, hogy legyen. Egy programsor a következő elemekből áll:

Például:

Sorszám	Utasítás	Címterület	„ENTER”
0000	LD	0000	←

Bármely PLC-ről is van szó, az utasításkészlete a következő parancsokat mindenképpen tartalmazza:

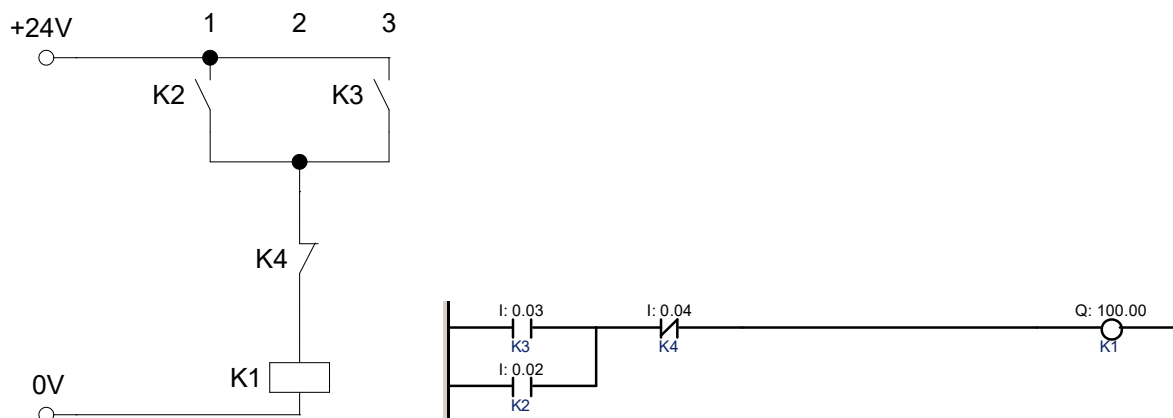
- Betöltés (LD)
- Logikai műveletek (NOT, AND, OR, AND NOT, OR NOT)
- Blokkok képzése, zárójelek (AND LD, OR LD)
- Kimenet megadása (OUT)
- Negált kimenet megadása (OUT NOT)
- Öntartás beírása és törlése (KEEP)
- Időzítő (TIM)
- Számláló (CNT)
- Differenciálás (DIFU, DIFD)
- Üres programsor (NOP)
- Program vége (END)

Egy utasításlistás programnak mindenképpen LD betöltés paranccsal kell kezdődnie.

1.5.2 Létradiagram (Ladder Diagram-LD)

A létradiagram a relés áramút-tervnek a PLC technikában alkalmazott egyszerűsített, áttekinthetőbb formája. Az áramút-tervtől abban különbözik, hogy az áramutakat vízszintesen rendezzi, és más szimbólumokat használ, amelyek a számítógép karaktereinek jobban megfelelnek.

Példa egy áramutas logikai összefüggés létradiagramos ábrázolására:



A programozás tehát általában a létradiagram elkészítésével kezdődik. A létradiagramot PC segédsoftver alkalmazásakor számítógépünk képernyőjén grafikusán megszerkesztjük.

A vezetékek (logikai vonalak) a baloldali, úgynevezett referenciavezetékéből indulnak ki. Ezután következnek a beiktatott érintkezők. Ezek lehetnek a bemenetekkel, kimenetekkel vezérelt vagy a belső segédrelékhez, tartórelékhez, időrelékhez tartozó záró-, ill. bontóérint-

kezők. A logikai vonal jobboldali végén a kimenetek, időrelék, számlálók stb. "tekercsei", vagy utasítások vannak.

Fontos szabály, hogy a létradiagramban és a hozzá tartozó programban az egyes kimenetek, tartórelék, időrelék, számlálók stb. csak egyszer szerepelhetnek (kivéve a lentebb ismertetett iSmart család SET/RESET utasításai). Ezek munkaérintkezői azonban programunkban, korlátlan számban felhasználhatók.

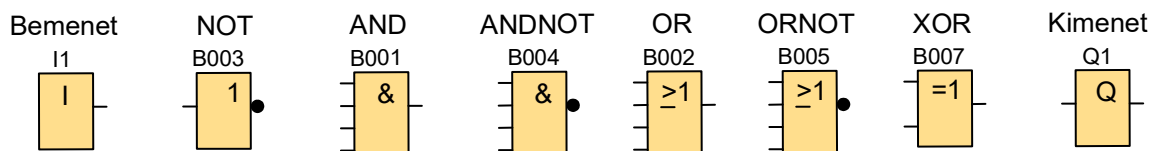
A program futásakor a vezérlőberendezés gyakorlatilag egyidejűleg figyeli (a valóságban ciklikusan letapogatja) a bemenetek állapotát, és ezeknek megfelelően állítja a kimeneteket.

Amennyiben több kimenetnek és/vagy utasításnak azonos a logikai feltétele, és az elágazás valamint a kimenetek/utasítások között nincs további érintkező, úgy ennek a logikai összefüggésnek a következő módon történő programozása megengedett:

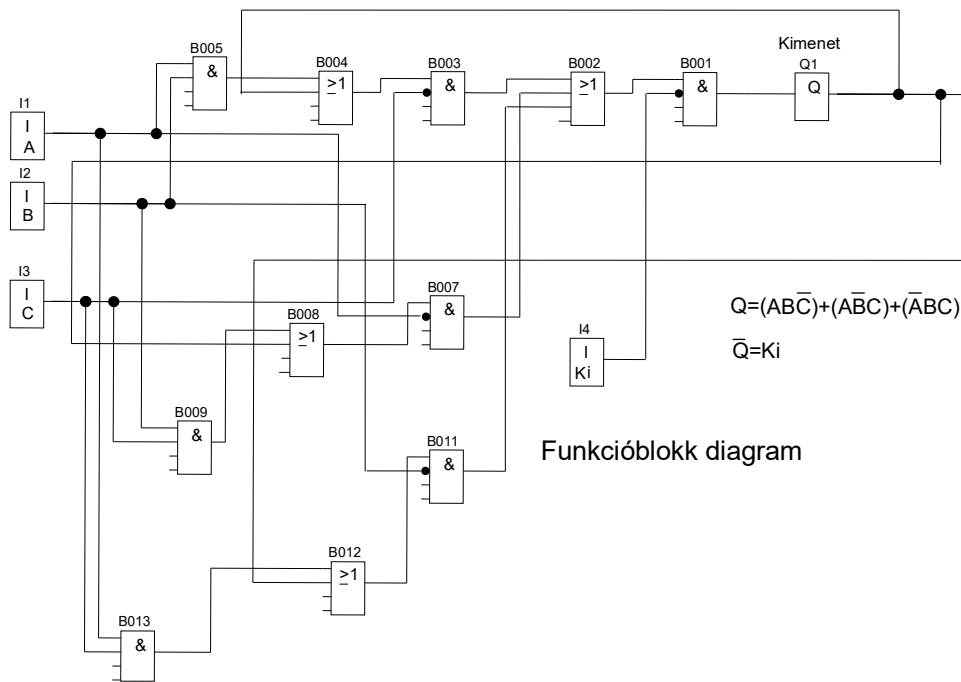


1.5.3 Funkcióblokk diagram (Function Block Diagram-FBD)

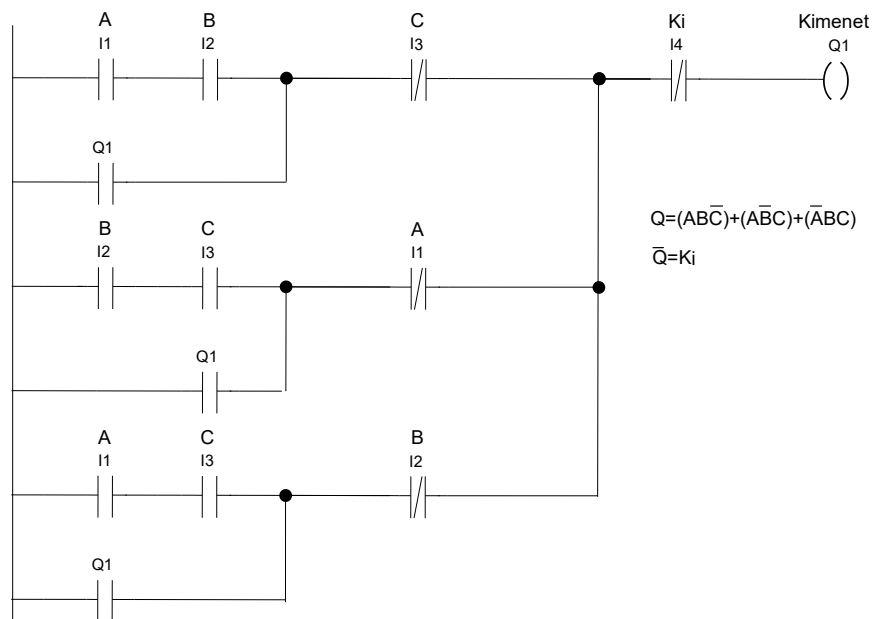
Funkcióblokkos programozási nyelv a huzalozott logikában használt, szimbólumokból kialakított nyelv. A funkcióblokkokon be-, és kimenetek találhatóak. A funkcióblokk bal oldalán mindig a bemenetek, a jobb oldalon a kimenetek találhatóak. A jelfolyam az előző fokozat kimeneteitől a bemenetek felé halad, azaz balról jobbra. Az alábbi rajzjelek néhány alapelemet mutatnak.



Általában a létradiagram és a funkcióblokk diagram szoftveresen átkonvertálható egymásba. Az alábbi példa ugyanazt a logikai feladatot mutatja kétféle ábrázolásban. Három nyomógombból (A, B, C) bármely kettőt lenyomva legyen aktív a Q kimenet, de mindhármát lenyomva ne! A kimenetet a „Ki” nyomógommbal kapcsolhatjuk ki.



Létradiagram



Láthatóan a létradiagramon egyszerűbb az eligazodás.

1.5.4 Strukturált szöveg (Structured Text-ST)

Strukturált szöveg használata esetén a felhasználói program valamilyen magasabb szintű programnyelvhez (C, Pascal, Basic) hasonlít. Általában fejlesztéseknél, illetve olyan esetekben használatos, ahol a kívánt feladatra nincs a fejlesztőkörnyezetben külön létra-jel, funkcióblokk jel. A matematikai eljárások programozhatóak így legegyszerűbben.

Például három bemeneti érték átlagát kiszámoljuk, majd összehasonlítjuk, hogy bizonyos határértékek között van-e.

```
AvgValue:=(Inpt1 + Input2 + Input3) / 3.0;  
IF ((AvgValue<=UpLimit) AND (AvgValue>=LowLimit)) THEN  
    Result:=TRUE;  
ELSE  
    Result:=FALSE;  
END_IF
```

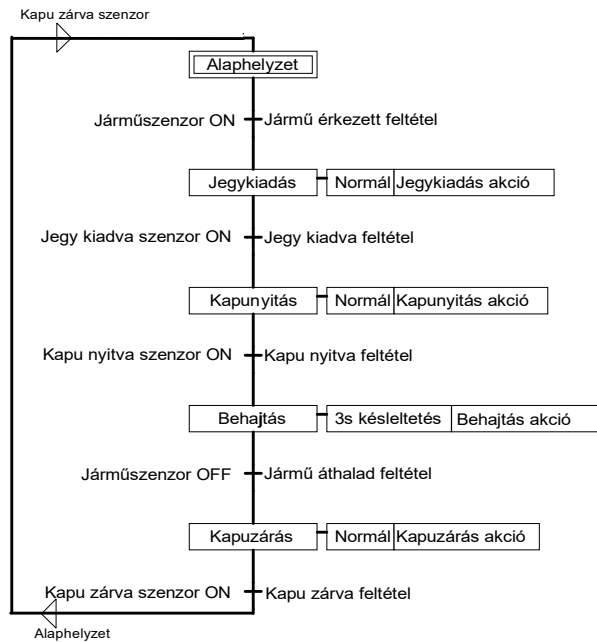
Minden fejlesztőkörnyezetnek van saját szintaktikája (helyesírási szabályrendszere), amit szigorúan be kell tartani. Ezeket a szoftver egyébként folyamatosan figyeli, ha kell, hibajelzést küld. Nem minden PLC programozható ezzel a módszerrel, és nem minden fejlesztőkörnyezet támogatja.

1.5.5 Sorrendi folyamatábra (Sequential Flow Chart – SFC, Grafcet)

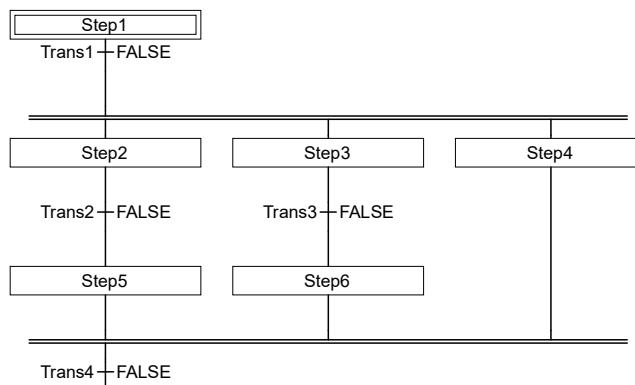
Az automatizálendő folyamatot grafikusán, lépésről lépésre írjuk le. Minden egyes lépéshez (Step) átmeneti feltételek (Translations) tartoznak. Minden lépéshez akciók (Actions) tartoznak, amelyek működésének eredményeképpen a folyamat a következő lépésbe lép, ha az átmeneti feltétel teljesül. Minden feltételhez és akcióhoz külön alprogram tartozik, amit meg lehet írni, utasításlistával, létradiagrammal, funkcióblokkokkal, vagy strukturált szöveggel. A feltételeknél általában a bemeneteket programozzuk, míg az akcióknál a kimeneteket. Ez a programozási forma is nagyon érzékeny a szintaktikára!

Az utolsó feltétel teljesülése után általában vissza kell térnie a programnak a kezdőlépéshez, ezt a ciklusosságot külön kell programoznunk.

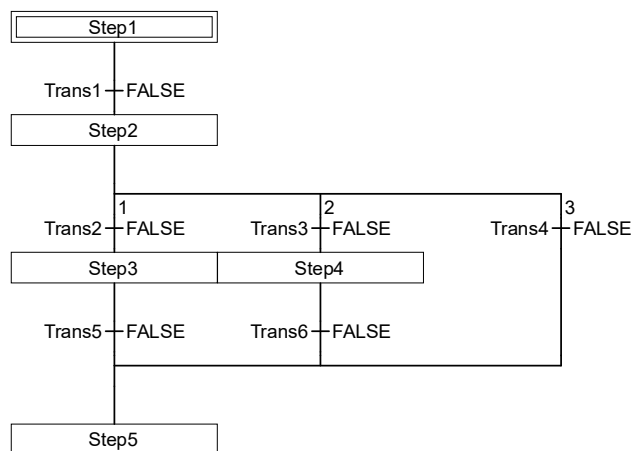
A sorrendi folyamatábra alapú nyelv lényegesen eltér a többi nyelvtől, mivel az SFC-t közvetlenül nem lehet a PLC-kbe letölteni. Az SFC-ből általában az eddig bemutatott nyelvek valamelyikén megvalósított programokat generál a szoftver.



A folyamat előrehaladása során lehet „ÉS” eseményeket programozni, rajzolni:



lehet „VAGY” eseményeket programozni, rajzolni:



A „VAGY” eseménynél megadható, hogy melyik feltételt vizsgálja először a program.

1.6 Egy PLC-vel megoldott vezérlés dokumentálása

A dokumentáció típusa attól függ, hogy milyen jellegű vezérlést kell megvalósítanunk. Ezért először foglaljuk össze, hogy milyenek is lehetnek a vezérléseink.

- Logikai vezérlések
- Folyamatvezérelt lefutó vezérlések
- Idővezérelt lefutó vezérlések
- Összetett vezérlések

1.6.1 Logikai vezérlések

Nevezik ezt a vezérlést kapcsolásvezérlésnek is, mert valamely berendezés be-, és kikapcsolásának logikáját tartalmazza, vagyis hogyan lehet bekapcsolni, illetve kikapcsolni egy gépet. A feladat megoldásakor a következőket kell dokumentálni:

- Boole algebrai egyenlete a bekapcsolt és a kikapcsolt állapotnak. Az esetek többségénél elegendő a bekapcsolást definiálni, mert a kikapcsolás ennek az ellentettje.

Például. $A \bullet B = Q$

Megjegyzés: a logikai alapl műveletek, és azok ábrázolása.

NEM	ÉS	VAGY
NOT	AND	OR
\bar{A}	$B \bullet C$ vagy $B \wedge C$	$E + F$ vagy $E \vee F$

- Igazságtáblázatot, amely szemléletesebb képet ad a logikáról.

A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

- Amennyiben lehetséges a Boole algebrai egyenlet egyszerűsítésének lépéseit vagy a grafikus egyszerűsítés (Karnough tábla, Veutch tábla) lépéseit. Egyszerűsítés után újra írjuk fel az egyenletet és az igazságtáblázatot!
- Az összerendelési táblázat
- A létradiagramot
- Az utasításlistát
- A PLC bekötési vázlatát

Összerendelési táblázat:

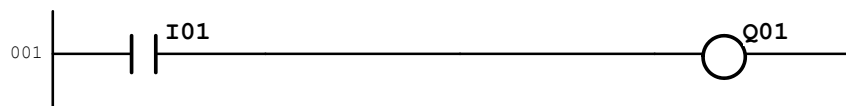
Bemenetek összerendelési listája												
Bemeneti cím	I01	I02	I03	I04	I05	I06	I07	I08	I09	I10	I11	I12
Funkció	Be- kapcs											
Megjegyzés												
Kimenetek összerendelési listája:												
Kimeneti cím	Q01	Q02	Q03	Q04	Q05	Q06	Q07	Q08				
Funkció	Lámpa											

Programlista:

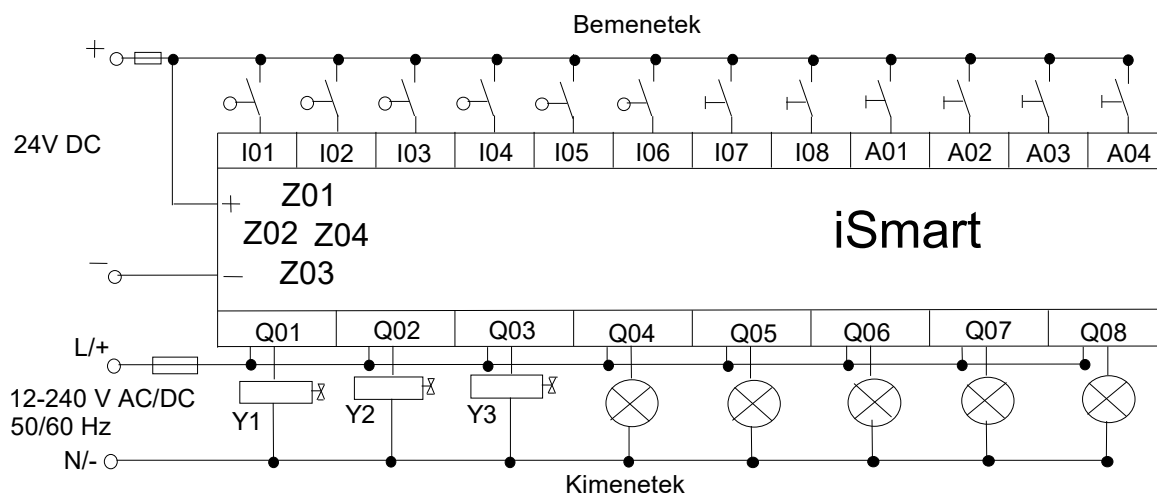
LD I01

OUT Q01

Létradiagram:



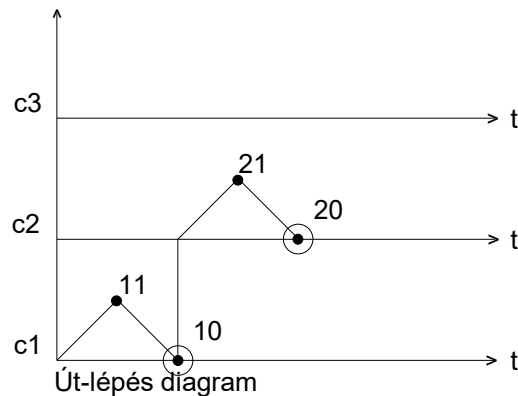
A PLC bekötési vázlatja:



Folyamatvezérelt lefutó vezérlések

Összetett mozgássorok tartoznak ide, ahol a következő eseményt az őt megelőző vége indítja. Általában ciklusosan ismétlődő mozgásokról van szó, ahol a mozgáselemek végét végállaskapcsolók, szenzorok jelzik. A mozgást legjobban az út-lépés követődiagramok szimbolizálják. A feladat megoldásakor a következőket kell dokumentálni:

- Út-lépés diagramot impulzusanalízissel



- Az összerendelési listát
- A létradiagramot
- Az utasításlistát
- A PLC bekötési vázlatát

1.6.2 Idővezérelt lefutó vezérlések

Azok a vezérlések tartoznak ide, ahol valamilyen belső óra ad meghatározott időközönként jeleket egy folyamat elemeinek be-, és kikapcsolásához. Ilyen például a futófény vagy a közlekedési jelzőlámpák vezérlése. A feladat megoldásakor a következőket kell dokumentálni:

- Táblázatot az idők számáról, sorrendjéről, funkcióiról

1. esemény T01	Piros fény	(60 s)
2. esemény T02	Piros-sárga fény	(20 s)
3. esemény T03	Zöld fény	(70 s)
4. esemény T04	Sárga fény	(20 s)

- Az összerendelési listát
- A létradiagramot
- Az utasításlistát
- A PLC bekötési vázlatát

1.6.3 Összetett vezérlések

Az előző három altípus keveréke, minden részét dokumentálni kell.

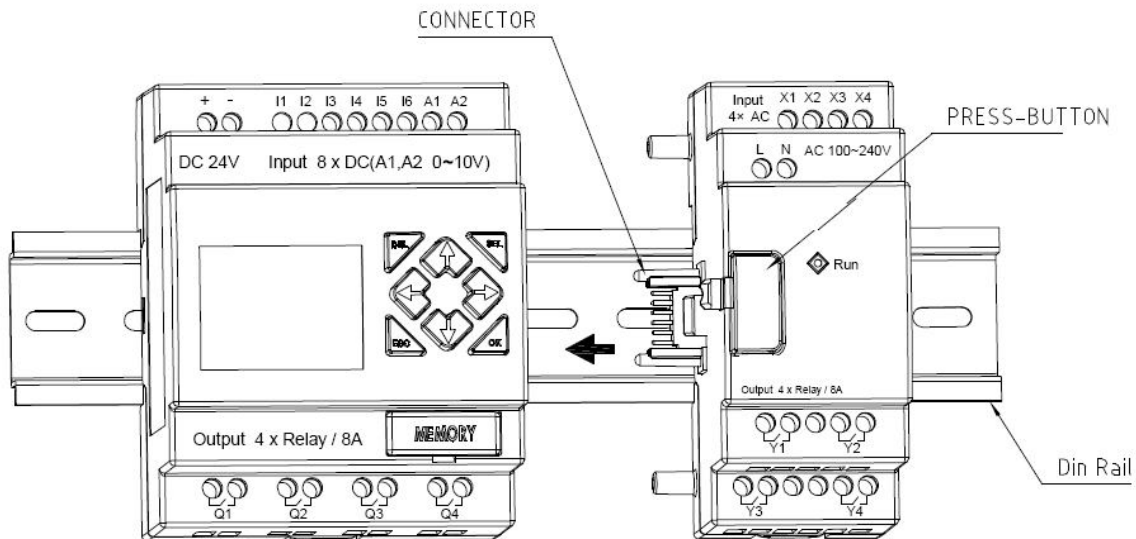
1.6.4 A biztonságos üzemeltetéssel kapcsolatos tervezési szempontok

A vezérlést úgy kell megvalósítani, hogy egy hiba fellépésekor ne veszélyeztesse a körülötte dolgozó embereket és környező berendezéseket. Ehhez a következő szabályokat kell betartani:

- A START gombok alaphelyzetben nyitottak legyenek! Ha alaphelyzetben zárt lenne, akkor vezetékszakadáskor a berendezés önmagától elindulna.
- A bekapcsoló gombok zöld színűek legyenek!
- A bekapcsolt állapotot zöld színű visszajelző jelezze!
- A STOP gombok alaphelyzetben zártak legyenek! Ha alaphelyzetben nyitott lenne, akkor vezetékszakadáskor a berendezést nem lehetne kikapcsolni.
- A kikapcsoló gombok piros színűek legyenek!
- A kikapcsolt állapotot piros színű visszajelző jelezze!
- Legyen a programban VÉSZSTOP!
- Tervezzünk ALAPHELYZET nyomógombot a rendszerbe, mellyel minden öntartás, számláló, időzítő alaphelyzetbe hozható, nullázható!
- Küszöböljük ki a prell hatást bemeneti szűrés segítségével! (Prell hatás: amikor egy kapcsolónál az érintkező zárásakor a visszarugózás miatt „0” és „1” jelek sorozata jut a PLC bemenetére.)

1.7 Röviden az SMT okos reléről

Az IMO (www.imopc.com) cég gyártja, Magyarországon a Tech-Con Hungária Kft. forgalmazza az iSmart, vagy SMT fantázianevét kompakt PLC-ket, más néven okos reléket. Szabványos DIN sínre szerelhetők.



Elnevezésük utal a kiépítettségükre, például SMT-ED-R20.

SMT	E	Bővíthető, LCD-s, kezelőgombos	A	AC 110-240V	R	Relés kimenet	10	I/O port összesen
	B	Bővíthető, nem LCD-s, nem kezelőgombos	D	DC 24 V	T	Tranzisztoros kimenet	12	
	C	Bővíthető, LCD-s, kezelőgombos, MODBUS	D12	DC 12V			20	
	M	Bővítő modul					8	





Egy okos relé maximum 44 db I/O-t tud kezelni. Egy bővítő modul 4 bemenetet és 4 kimenetet tartalmaz, ezekből legfeljebb 3 db fűzhető rá a központi egységre.

Létradiagramos módszerrel, vagy funkció-blokk diagramos módszerrel programozhatók a készülék tetején elhelyezett billentyűk, vagy számítógépes SMT szoftver segítségével.

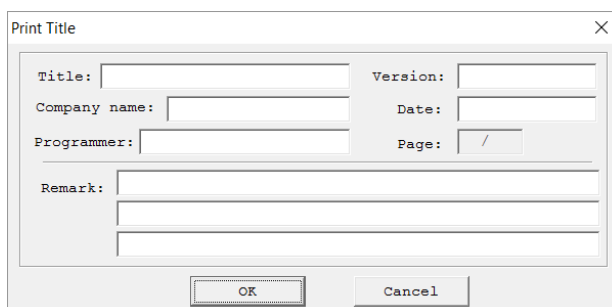
A számítógép valamelyik soros portjával az SMT-PC03 adatkábellel, USB portjával az SMT-USB adatkábellel lehet összekötni. Ha nincs a számítógépen soros port, bármely USB-soros átalakító használható hozzá. Ha Ethernet hálózattal kívánjuk összekötni, SMT-ENET bővítő modult kell hozzá csatlakoztatni.

1.8 Az SMT szoftver (ingyenesen letölthető a www.imopc.com címről)

1.8.1 File

A szokásos menüpontok találhatóak itt úgy, mint új program kezdése  (New), meglévő program betöltése  (Open...), program mentése  (Save), program mentése más néven (Save As...), nyomtatási előnézeti kép  (Print Preview), továbbá a legutóbb megnyitott programok listája. A mentett programok kiterjesztése: **.gen**.


A „Print Title” menüpont alatt a létradiagram fejlécének űrlapját tölthetjük ki.

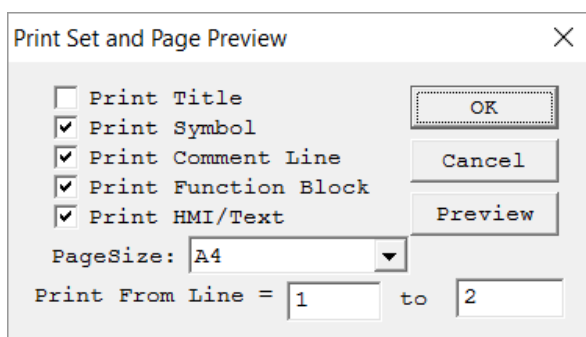


The 'Print Title' dialog box contains the following fields:

Title:	Version:
Company name:	Date:
Programmer:	Page: /
Remark:	

Buttons: OK, Cancel

A „Print... ” menüpont alatt bejelölhetjük, hogy a tervünk mely részeit kívánjuk kinyomtatni a létradiagramon kívül: fejléc (**Title**), a létradiagram-elemekhez társított magyarázó szövegek (**Symbol**), magyarázó szövegsorok a létrafokok között (**Comment Line**), funkcióblokkok tartalma (**Function Block**), és az LCD kijelzőre tervezett feliratok (**HMI/Text**). Az utóbbi kettőt külön lapokra készíti táblázatosan a program. (Hibajelenség a 3.49 verzióban, hogy a funkcióblokkok tartalmát csak akkor nyomtatja, ha a HMI/Text is be van kapcsolva.) Az is beállítható itt a papírméreten kívül (**Page Size**), hogy hányadik programsortól hányadik programsorig nyomtassunk.



The 'Print Set and Page Preview' dialog box contains the following options:

- Print Title
- Print Symbol
- Print Comment Line
- Print Function Block
- Print HMI/Text

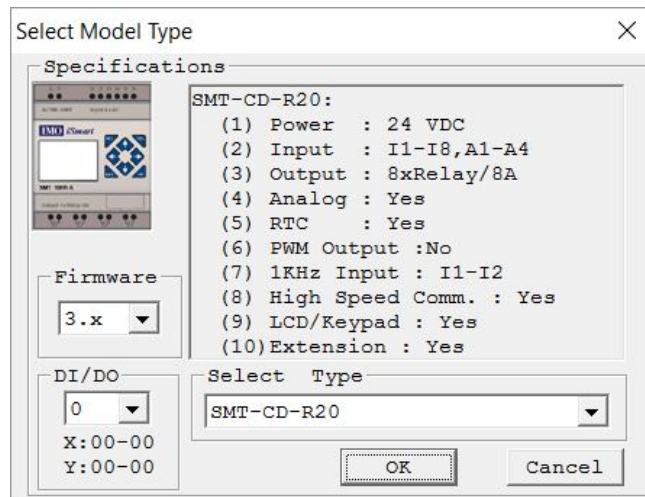
Buttons: OK, Cancel, Preview



Page Size: A4

Print From Line = 1 to 2

1.8.2 Edit

A „**Select Model...**” pontban iSMART modellt tudunk választani, illetve váltani. Itt megadható a használt firmware verziócsoporth (3.x – 2.x) és az esetlegesen használt DI/DO bővítőmodul sorszáma (0 – 3) is.




Ebben a menüben tudunk a létra-nézet  (**Ladder**), és az iSMART-látszati nézet  (**Keypad**) között váltani.


Ezt követik a szokásos mégsem  (**Undo**), mégis  (**Redo**) menüpontok.

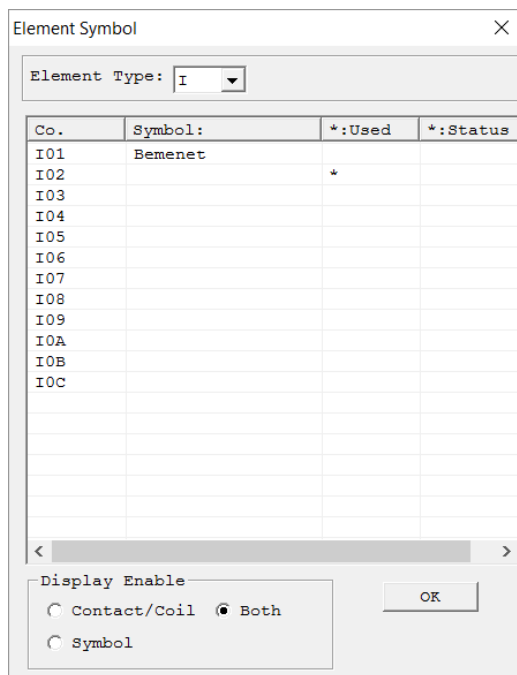
A következő pontban tudjuk a létradiagramhoz esetlegesen fűzött magyarázó szövegeinket törölni (**Clear Comments**) (egyszerre mindet).

A keresés (**Find...**) menüpontban a programban használt szimbólumokat vagy a szimbólumhoz társított szövegeket tudunk keresni. Figyeljünk arra, hogy hol áll a kurzor, és ehhez képest állítsuk be az irányt lefelé (**Down**) vagy fölfelé (**Up**)!

A csere (**Replace...**) menüpontban szimbólumokat (egyesével, vagy egyszerre mindet) tudjuk lecserélni. Itt is figyeljünk az irányra!

A „**HMI/Text...**”  menüpontban a HMI kimenetekhez tartozó szövegeket tudjuk létrehozni, szerkeszteni. (Részletezve később!)


A „**Symbol...**”  pontban a diagram elemeihez tudunk társítani rövid, (nyolc karakter hosszú) magyarázó szövegeket. Ezt egyébként egyenként már a diagram-elemek elhelyezésekor is megtehetjük a konfigurációs panel „Symbol” nyomógombjára kattintva. Itt azonban együtt, táblázatosan látszik, típusonként összegyűjtve. Azt is kiválaszthatjuk, hogy a létradiagramon a cím (**Contact/Coil**), a magyarázó szöveg (**Symbol**), vagy mindkettő (**Both**) látszódjon.




Az „I/O Display Set...” pontban azt állíthatjuk be, hogy mely változókat jelezze ki a PLC LCD-je.


„3-Contact---5-Contact” pontban azt választhatjuk ki, hogy a létradiagram milyen széles legyen, vagyis a bal oldalán 3 vagy 5 kontaktus férjen el. Ha már egyik féleképpen elkezdtünk tervezni, nem feltétlenül tudunk menet közben váltani közöttük oda-vissza! Az 5 kontaktusos beállítást a PLC LCD kijelzője nem tudja helyesen megjeleníteni!

1.8.3 Operation




A „Monitor ” menüpont csak akkor választható ki, ha a PC ONLINE módban van a PLC-vel. Bekapcsolva egyúttal „Run” üzemmódba is kerül a PLC. Ekkor megkérdezi a szoftver, hogy áttöltjük-e a PLC tartalmát a PC-re, ha igen, megkérdezi, mentjük-e a képernyőnkön lévő állapotot. Az áramutakat **rózsaszín** létrafok és szimbólumok jelölik.

Fordítva is igaz, ONLINE módban a „Run ”-t választva a „Monitor” üzemmód is bekapcsol.


A „Quit ” pontot választva a szoftver kilép a monitor üzemmódból, de a PLC „Run” állapotban marad (bár a státuszsor nem ezt mutatja).

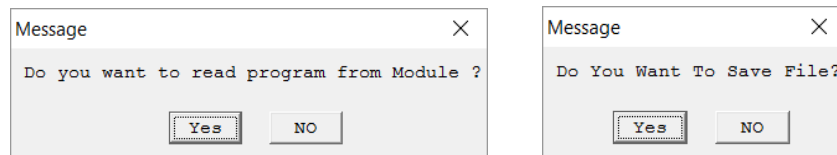
„Stop ”-ot választva a szoftver a „Monitor” üzemmódból és a PLC a „Run” üzemmódból is kilép.


„Monitor” módban a képernyő kontaktusokat szimbolizáló jelképei **nem reteszelvek** (nyomógomb funkció)!

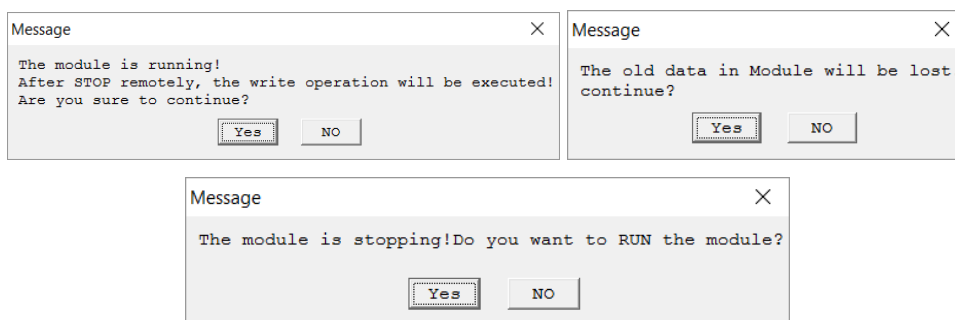
„**Simulator** ” üzemmód ONLINE és OFFLINE módban is választható. Ezt választva indul a szimulációs „**Run**” üzemmód a szoftverben, de természetesen a PLC- nem. Az áramutakat **zöld** létrafok és szimbólumok jelölik. Szimulátor módban a „**Quit**” és a „**Stop**” egyenértékűek. Szimulációs programfutáskor használható a „**Pause**”  menüpont, hogy a folyamatot egy pillanatra megállítsuk. Újra lenyomva, a folyamat tovább folytatódik. Ugyanitt található a „**Power** ” menüpont, amivel áramkimaradást tudunk szimulálni (bár ugyanazt teszi, mint a **Quit**)

Szimulátor módban a képernyő kontaktusokat szimbolizáló jelképei **reteszelvek** (kapcsoló funkció)!

A „**Read** ” menüpont segítségével olvassuk ki a programunkat a PLC-ből a PC-be. Előtte azért a képernyőn lévő programváltozatról egy mentési lehetőséget felkínál a szoftver.



A „**Write** ” menüpont segítségével írjuk ki a programunkat a PC -ből a PLC -be. Ha „**Run**” üzemmódban vagyunk, előtte azért figyelmeztet a szoftver, hogy a PLC „**Stop**” üzemmódba fog kapcsolni, illetve a PLC-n lévő program felül fog íródni. Írás után megkérdezi, hogy a PLC-t átváltjuk-e „**Run**” üzemmódra.



A „**Compare**” menüpont segítségével össze tudjuk hasonlítani a PC-ben, és a PLC-ben lévő programunkat, ha van különbség, azt táblázatosan kiírja a szoftver.

A „**Check Error**” menüpont segítségével lefuttathatunk egy online hibaellenőrzést.

Az „**RTC Set...**” menüpontban a készülék belső óráját tudjuk beállítani. Online módban a „**PC Clock**”, majd az „**OK**” gombra kattintva egyszerűen átvihető a PC órabeállítása a PLC-re. Ugyanezen panelon engedélyezhetjük, azután pedig beállíthatjuk a téli és a nyári időszámítás átváltási dátumát, óráját.

Az „Analog Set...” pontban az analóg bemenetek erősítési tényezőit és tőrésatárait tudjuk beállítani.

A „Password...” pontban lejelszavazhatjuk a készüléket. Az alapértelmezés, amikor nincs jelszó: **0000**. Kétfajta jelszó létezik. Az „A osztályú” jelszó értéke **0001-9FFF** közé eshet, a „B osztályú” jelszó értéke **A000-FFFF** közé eshet. Az alábbi táblázat mutatja, hogy melyik osztály milyen hozzáférést korlátoz.

Menu	A Class	B Class
LADDER	√	√
FUN.BLOCK	√	√
FBD	√	√
PARAMETER		√
RUN/STOP		√
DATA REGISTER		√
CLEAR PROG.	√	√
WRITE	√	√
READ	√	√
SET		√
RTC SET		
ANALOGUE SET		√
LANGUAGE		√
INITIAL	√	√

Ha elfelejtjük a jelszót, a PLC kikapcsolt állapotában behelyezünk egy olyan SMT PM04 (3rd) memóriakártyát a PLC kommunikációs nyílásába, ami gyári alaphelyzetben van, tehát nincs lejelszavazva. Ezután rákapcsoljuk a PLC-re a tápfeszültséget, ekkor a memóriakártyán lévő program és beállítások betöltődnek a PLC-be és végre is hajtódnak. Igaz, hogy a PLC-ből a benne lévő program törlődik, de a jelszó is törlődik, a készülék gyári alaphelyzetbe kerül.

„Language” pontban kezelői nyelvet tudunk váltani (magyar nincs).

A „**Module System Set**” fontos beállítási paramétereket tartalmaz.

Ha több PLC van RS 485 kommunikációs vonalon összekötve, itt beállíthatjuk azok azonosítóit (**Set ID (00-99)**).

Ha 2 db PLC van összekötve, úgy, hogy az egyik távoli I/O (**Remote I/O**), beállíthatjuk, hogy melyik a mester és melyik a szolga.

Ha van a PLC-hez bővítő modul kapcsolva, beállíthatjuk annak sorszámát (**Set expand I/O (0-3)**). Kérhetünk riasztást is (**I/O Alarm**), ha nem érhető el a bővítő modul.

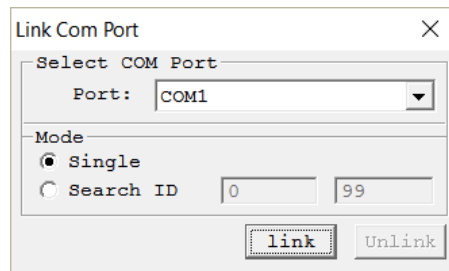
Talán a legfontosabb az „**Others**” pontban van, az „**M Keep**” segítségével bekapcsolhatjuk, hogy áramkimaradáskor az **M** típusú segédrelék őrizzék meg értéküket, de ezzel egyszerre azt is bekapcsoljuk, hogy a **T0E** és **T0F** időzítők is megtartsák értéküket.

A „**C Keep**” segítségével bekapcsolhatjuk, hogy áramkimaradáskor a számlálók őrizzék meg értéküket, a „**Back Light**” pontban bekapcsolhatjuk a készülék LCD paneljének az állandó háttérvilágítását (egyébként a kezelő gombokat nyomogatva világít, de utána 10 másodperc múlva kialszik). A „**Z Set**” pontban engedélyezhetjük a kezelőn lévő nyíl billentyűk Z1-Z4 bemenetkénti használatát.

Az „**RS-485 Set**” pontban átállíthatjuk az adatátvitel módját. Alapértelmezetten a kommunikációs mód (**Com. Mode**) **8/n/2**, vagyis 8 adatbit, 0 paritásbit, 2 stopbit van egy csomagban. Alapértelmezetten az átviteli sebesség (**Boud Rate**) 38400 bps.

A „**DR Format Set**” pontban, ha „**Unsigned**”-et választjuk, az adatregiszterek értékei 0-65535 közötti előjel nélküliek, „**Signed**” beállításban -32768 + 32767 közötti előjelesek.

A „**Link Com Port**” menüpont is nagyon fontos, itt kapcsolhatjuk PC-PLC rendszerünket **ONLINE (Link)** és **OFFLINE (Unlink)** állapotba.

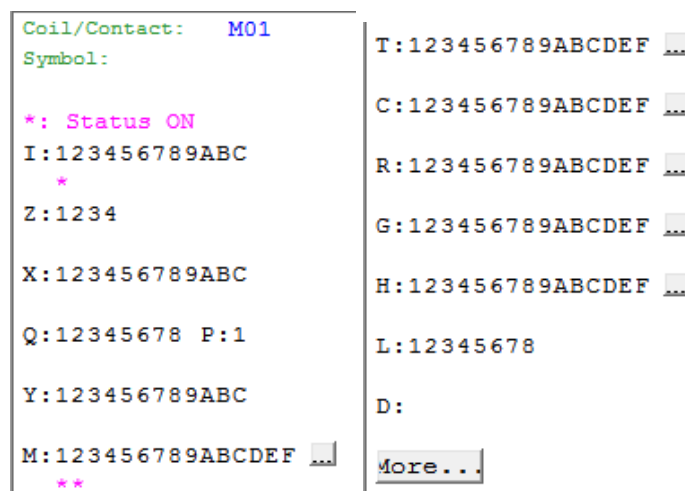


Azt, hogy melyik soros portot kell választani, megtudhatjuk a PC / Saját gép / Kezelés / Eszközkezelő / Portok menüpontban (Windows). Ha egy PLC kapcsolatunk van, a „Single” módot kell választani.

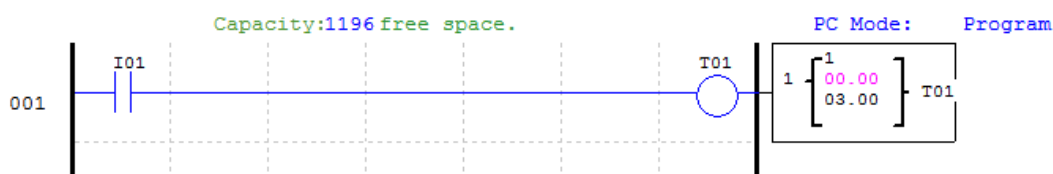
1.8.4 View

A képernyőn lévő panelek megjelenítését kapcsolhatjuk be-ki szimuláció, illetve monitor üzemmódokban.

Az „I/O” mutatja a különféle bemenetek, kimenetek, egyéb területek állapotát. Monitor üzemmódban amelyik ON állapotban van, az alatt egy kék * van, szimulátor üzemmódban amelyik ON állapotban van, az alatt egy rózsaszín * jel van. A „Coil/Contact” annak az elemnek mutatja a címét, és szimbólum-nevét, amelyen a kurzor áll. A sorok végén található gombokkal szimbólum- neveket adhatunk meg futtatás közben.

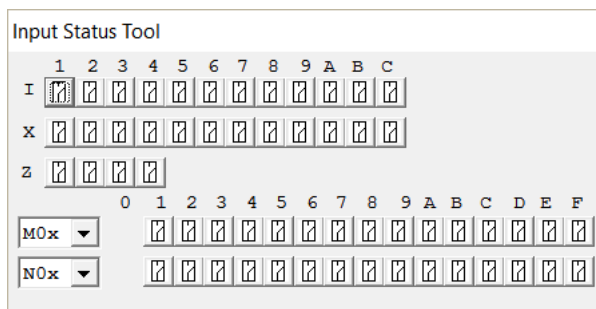


A „Function” mutatja az összes használt összetett utasítás beállítási paramétereit a létrától közvetlenül jobbra. Egyenként ezeket úgy lehet bekapcsolni, ha a tekercs-jelképre egyet kattintunk.

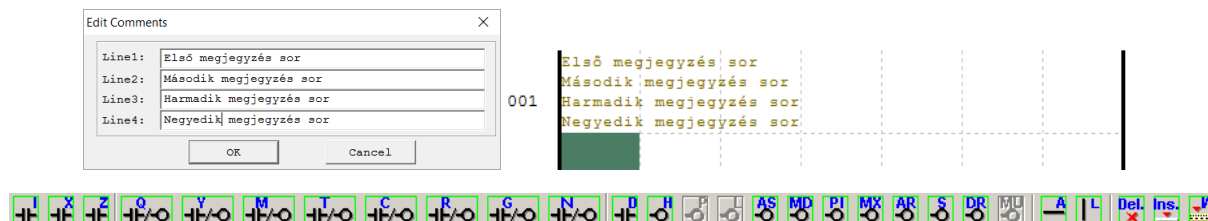


A „**Capacity**” megmutatja, hogy a létrában még hány üres jelképhely van (maximum 300 sor, soronként 4 jelkép, vagy 200 sor, soronként 6 jelkép, maximum 1200 db).

Monitor vagy szimulátor módban különféle eszköztárcák kapcsolhatók még be, ezek közül a legfontosabb az „**Input Status Tool**”, amivel tulajdonképpen a digitális bemeneteket, memóriaterületeket helyettesítjük.



A „**Ladder Toolbar**” a képernyő alján bekapcsolja az összes bemenet, kimenet szimbólumát. Bal egérgombbal rákattintva kiválaszthatjuk bármelyiket, ezután a létra tetszőleges üres helyére bal egérgombbal kattintva elhelyezhetjük a szimbólumot. Kilépni ebből a jobb egérgombbal való kattintással lehet. Az „**A**” betűvel vízszintes vonalat, az „**L**” betűvel függőleges vonalat rajzolhatunk. A „**Del**” törli a kurzor alatti elemet, az „**Ins**” beszúr egy sort a kurzor fölé. A „**W**” betűvel maximum négysoros megjegyzést írhatunk a sorok közé.



1.8.5 HELP

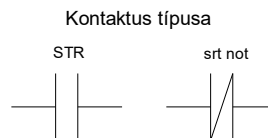
A „**HELP**” segítségével egyszerű angol nyelvű súgót kapunk, illetve megtudhatjuk a szoftver verziószámát, gyártójának adatait.



1.9 Digitális eszközök, azok címei

1.9.1 Kontaktusok

A kontaktusok (I, X, Z, Q, Y, M, N, D T, C, R, G,) lehetnek alaphelyzetben nyitottak, vagy zártak, és a létradiagram bal oldalán helyezkednek el. A nyitottakat a szoftver nagybetűvel, a zártakat kisbetűvel jelöli.



1.9.2 I, Bemenetek (I01 – I08)

A valós digitális bemeneti pontokat I típusú memóriák jelölik modelltől függően 6, 8, db bemenettel. Az A01 – A04 analóg bemenetek 24V-al használhatók digitális bemenetnek is, így kijön a 12 bemenet.

1.9.3 X, Bővítő modul bemenetek (X01 – X0C)

Ezeket a bemeneteket (12 db) csak akkor használhatjuk, ha csatlakoztatunk digitális bemeneti bővítő modult az alapkészülékhez.

1.9.4 Z, Billentyűzet bemenet (Z01 – Z04)

A kezelőgombokkal ellátott modelleknek a négy nyíl billentyűje bemeneti nyomógombként programozható ha a „Module System Set” potban a „Z Set” aktív.

(Z01 ▲, Z02 ◀, Z03 ▼, Z04 ▶)

1.9.5 Q, Kimenetek (Q01 – Q08)

A valós digitális Q kimenetektől modelltől függően 4 vagy 8 darab van.

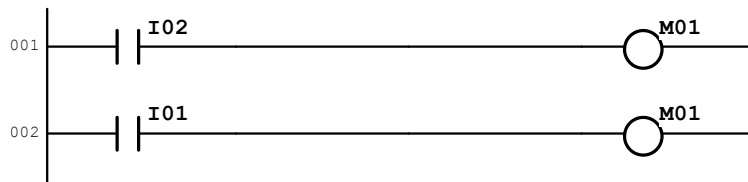


Addig van Q01 kimenet, amíg folyamatosan nyomjuk I01 bemeneti nyomógombot. Csak akkor működik helyesen a program, ha a létrafokban nincs szakadás! Az alábbi példa helytelen rajzot szemléltet.



Ha kétszer akarjuk megadni egy létradiagramban ugyanazt a funkcióblokk típusú (T, C, G, ...) kimenetet, a „**Double Used!**” hibaüzenetet kapjuk.

Érdekes módon ennél a készülécszaládnál, a (Q, Y, M, N) típusú kimenetekből ugyanaz többször is megadható, ilyenkor mindig a legelső létrafok utasítását hajtja végre, a példában I02-re nem kapcsol be M01, csak I01-re. Ez más PLC-knél általában hibaüzenetet generál.

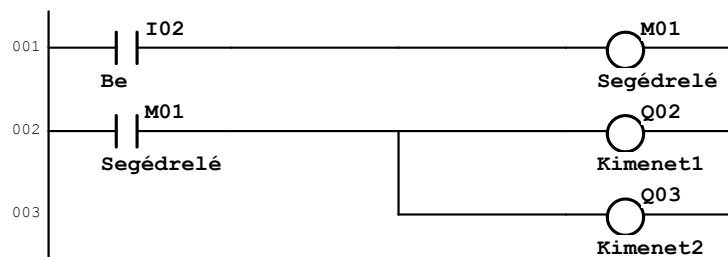


1.9.6 Y, Bővítő modul digitális kimenetek (Y01 – Y0C)

Ezeket a kimeneteket (12 db) csak akkor használhatjuk, ha csatlakoztatunk digitális kimeneti bővítő modult az alapkészülékhez.

1.9.7 M, Segédrelék (M01 - M3F)

63 db segédrelék (belső memória) használható egy programban belső tárolásra.



Például I02 nyomógomb bekapcsolja M01 segédrelét, majd M01 segédrelé bekapcsolja Q02, és Q03 valós kimeneteket.

A segédrelék meg fogják tartani állapotukat akkor is, ha a készüléket kikapcsoljuk, de csak akkor, ha a „**Options/Module System Set..** „ menüben az „**M Keep**” aktív.

1.9.8 M, Különleges segédrelék (M31 – M33)

M31: a felhasználói program indításakor az első ciklusban aktív, utána normál működésű. Felhasználható indítási alaphelyzet beállítására.

M32: 1s villogó kimenet 0,5s ON, 0,5s OFF

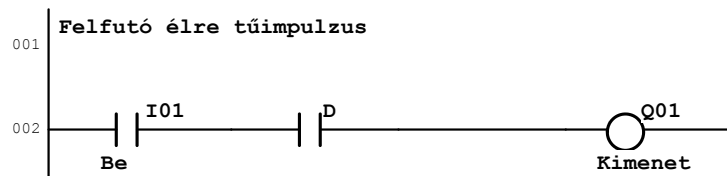
M33: nyári időszámításkor bekapcsol, téli időszámításkor kikapcsol.

1.9.9 N, Segédrelék (N típus, N01 – N3F)

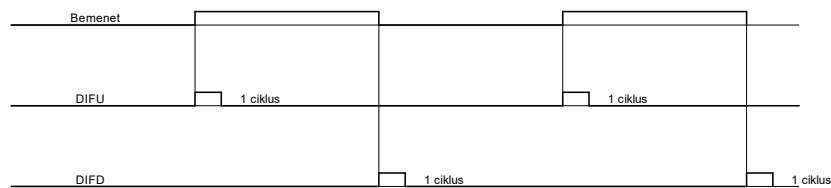
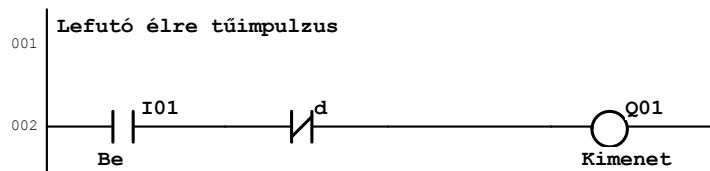
63 db, ugyanolyanok, mint az M típusok, de nem tartják meg állapotukat kikapcsoláskor.

1.9.10 D, Differenciáló kontaktus (D típus)

A bemenő jel felfutó élére egy ciklusidőnyi impulzus létrehozása



A bemenő jel lefutó élére egy ciklusidőnyi impulzus létrehozása



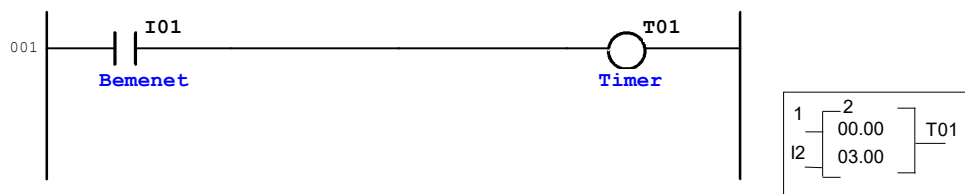
1.9.11 T, Timer, időzítő utasítás (T01 – T1F)

A programban 31 különálló időzítő lehet. A **T0E** és a **T0F** című időzítő képes áramkimaradáskor is megtartani értékét, ha az **"M-Keep"** beállítás aktív, a többi elfelejti aktuális állapotát. Az időzítés léptéke (**Time Base**) beállítható századmásodpercben, tizedmásodpercben, másodpercben, vagy percben. Az időzítő előre beállított értéke (**Preset Type**) lehet állandó, vagy más funkció aktuális változó értéke. Az alapértelmezett a szám (**N**).

2 és 6 módban feltétlenül definiálnunk kell „Reset” paramétert, különben csak a PLC „Stop” üzemmódba váltásával tudjuk nullázni az időzítőt! A Reset-et bekapcsolva nullázza az időzítő értékét, kikapcsolva, ha a feltételek teljesülnek, újratekdi az időzítést. A Reset-et elég definiálni, nem feltétlenül kell a létrába berajzolni! Ha mégis a létrában láthatóvá akarjuk tenni, azt is megtehetjük. **Például** az I01 bemenettel párhuzamosan rajzolt **I02**, ha az időzítő paraméterezésekor **Reset**-ként definiáljuk, nem párhuzamos bemenetként funkcionál! (Ezzel a „Reset” kontaktussal bármit kötünk sorba, az nem számít feltételnek!)

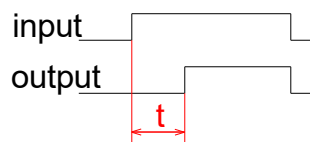


A két ábra egyenértékű



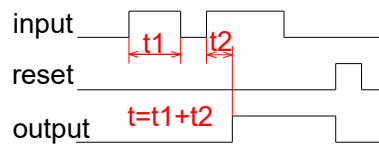
0. mód: egyszerű belső segédrelé tekercs, nem időzít valójában.

1. mód: Késve bekapcsoló időzítő. A bemeneti feltétel elindítja az időzítőt, előre beállított időértéknél bekapcsol a kimenet. A bemeneti feltétel megszűnése kikapcsolja a kimenetet. **Ez a legelterjedtebb alkalmazás!**

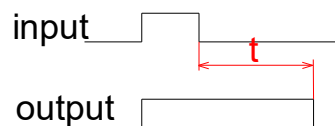


2. mód: Összegző késve bekapcsoló időzítő. A bemeneti feltétel elindítja az időzítőt, előre beállított időértéknél bekapcsol a kimenet. A bemenet megszakítható, ha újra van bemenet, folytatódik az időzítés. Ez alkalmassá teszi **T0E-T0F**-használva például **üzemóra**

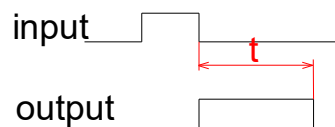
számlálására. Ha már bekapcsolt a kimenet, a bemeneti feltétel megszűnése nem nullázza, csak külső RESET.



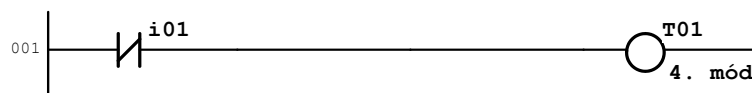
3. mód: Késve kikapcsoló időzítő 1. A bemeneti feltétel megléténel azonnal bekapcsol a kimenet, a bemeneti feltétel megszűnte után a beállított idővel kikapcsol a kimenet. RESET segítségével azonnal is kikapcsolható.



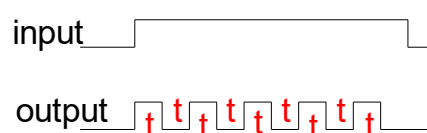
4. mód: Késve kikapcsoló időzítő 2. A bemeneti feltétel megléténel nem kapcsol be, hanem a bemeneti feltétel megszűntekor kapcsol be a kimenet, utána a beállított idővel kikapcsol. RESET segítségével azonnal is kikapcsolható.



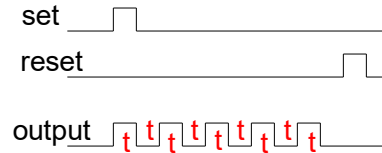
Ez a mód alkalmas **impulzus megszakításra**, ha a bemenet negáltját használjuk. Például egy végállaskapcsoló blokkoló jelének megszakítására kiválóan alkalmas így.



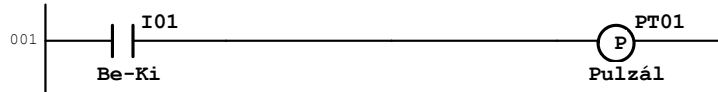
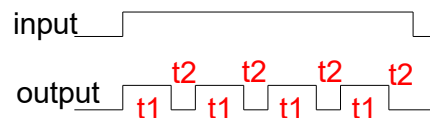
5. mód: Négyzetjel generátor (villogó) RESET nélkül. Be lehet állítani a periódusidő felét, ugyanaddig aktív és passzív. A feltételkontaktussal tudjuk be-, és kikapcsolni.



6. mód: Négyzetgjel generátor (villogó) RESET-el. Be lehet állítani a periódusidő felét, ugyanaddig aktív és passzív. A feltételkontaktus impulzusával tudjuk bekapcsolni, a RESET impulzusával tudjuk kikapcsolni.



7. mód: Négyzetgjel generátor (villogó) RESET nélkül. Az output típusát „P”-re kell állítani. Be lehet állítani a periódusidőn belül az aktív időszak idejét és a passzív időszak idejét külön-külön. A feltételkontaktussal tudjuk be-, és kikapcsolni.

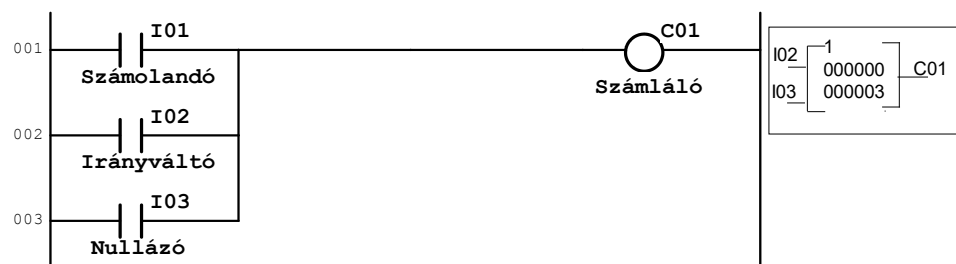


1.9.12 C, Counter, számláló utasítások, (C01 – C1F)

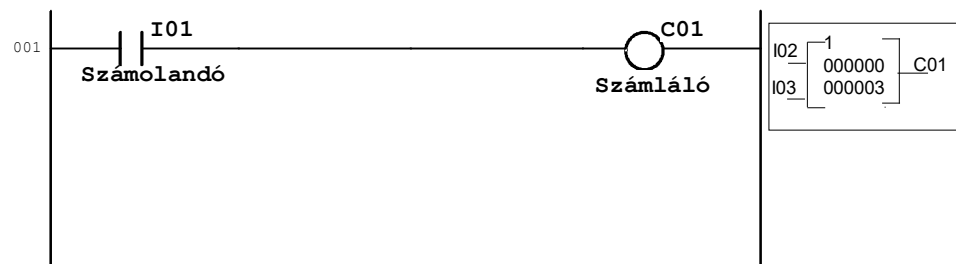
A szoftverben 31 db számlálót lehet használni, választható 9 féle üzemmód. Az előre beállított számláló értéke lehet állandó, vagy más funkció aktuális változó értéke.

A „**Direction set**” bekapcsolásával a számlálás iránya megfordítható, a „**Reset input**” bekapcsolásával a számláló alaphelyzetbe kerül, kimenete kikapcsolt lesz.

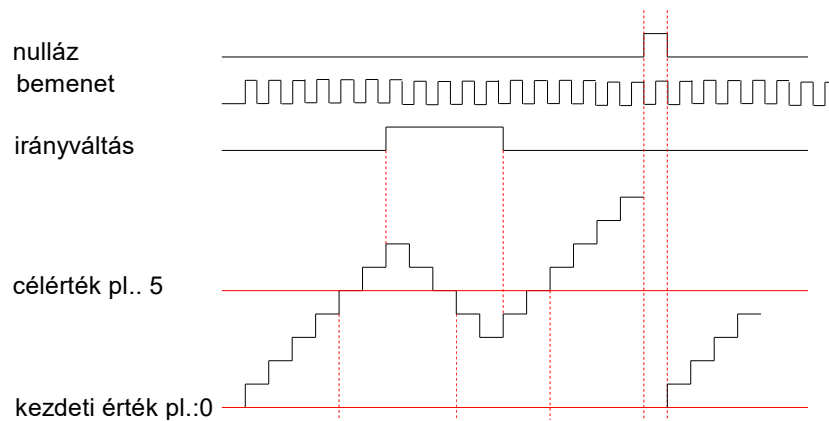
Furcsaság, hogy az I01 bemenettel párhuzamosan rajzolt I03, ha a paraméterezésekor Reset input-ként definiáljuk I02, ha a paraméterezéskor Direction set-ként definiáljuk, nem párhuzamos bemenet lesz!



Helyette általánosan az alábbi használható:



A példában I01-et számoljuk, 3 a határérték, ahol bekapcsol a számláló. I02-vel megfordítható a számlálás, ekkor 0-nál kapcsol be. I03-al (Reset) alaphelyzetbe hozható a számláló, ha a reset folyamatosan be van kapcsolva, akkor a többi bemenet nem aktív, a kimenet kikapcsolt marad.



0. mód: egyszerű belső segédrelé tekercs, nem számlál valójában.

1. mód: A számláló a bemenetére érkező impulzusokat számolja felfelé egy fix beállított értékig, a számolás megáll, ha a számláló értéke megegyezik a beállított értékkel, ekkor a számláló kimenete bekapcsol. Visszafelé is tud számolni, a Direction bemenetére érkező impulzusok hatására, de csak míg el nem érte a beállított értéket. 0 értéknél, és folyamatos Direction jelnél a kimenet bekapcsolt állapotú lesz. Ha a visszaszámlálás be van kapcsolva, akkor a Reset értéke nem 0, hanem a beállított érték lesz.

2. mód: Ugyan az, mint az 1. mód, de a beállított értéknél **tovább is számol**, előre számoláskor, ha a beállított értéket átlépi, a kimenet bekapcsolt marad.

3. mód: Ugyan az, mint az 1. mód, de a készülék kikapcsolásakor (áramkimaradás, STOP) megőrzi értékét (ha C Keep aktív (Operation/Module system Set...)).

4. mód: Ugyan az, mint az 2. mód, de a készülék kikapcsolásakor (áramkimaradás, STOP) megőrzi értékét (ha C Keep aktív (Operation/Module system Set...)).

5. mód: Ugyanaz, mint a 2. mód, vagyis a beállított értéknél **tovább is számol**, illetve a **beállított érték fölött visszafelé is tud számolni**, de visszafelé számolva, elérve a 0-t, nem kapcsol be. A Reset mindig nullázza.

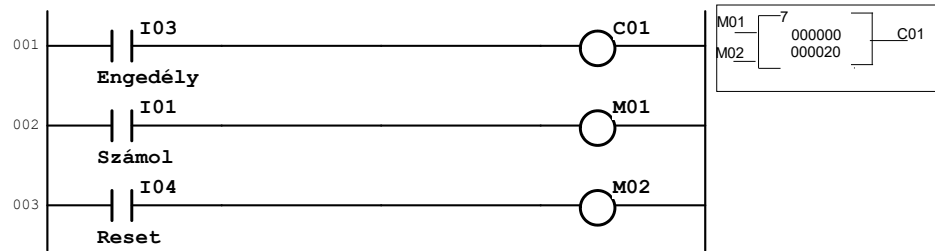
6. mód: Ugyan az, mint az 5. mód, de a készülék kikapcsolásakor (áramkimaradás, STOP) megőrzi értékét (ha C Keep aktív (Operation/Module system Set...)).

(Nagysebességű számlálók DC változat esetén)

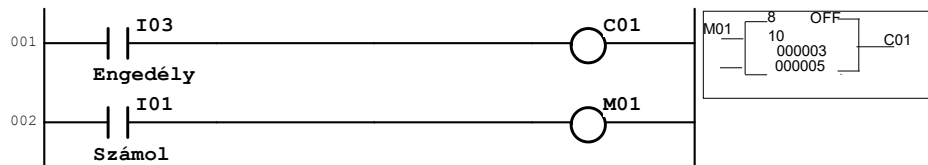
A DC típusú modell tartalmaz 2 db 1 kHz-es nagy sebességű bemenetet (I01 és I02). Ezeket lehet úgy is használni, mint általános célú digitális bemenet, vagy lehet úgy is, mint nagysebességű beviteli eszköz (pl.: fordulatszám jeladóhoz). A számlálót ekkor 7. vagy 8. módba kell kapcsolni.

7. mód: A számláló kimenete aktív lesz és úgy is marad, ha engedélyezve van (pl.: I03), a bemeneti impulzusszám (pl.: I01) eléri a beállított értéket (pl.: 20).

A számláló törlődik, amikor az engedélyező jel inaktív vagy a reset (I04) bemenet aktív.



8. mód: Be kell állítani a mérési időtartamot másodpercben (pl. 10s), továbbá azt az impulzusszámot, amikor már be kell, hogy kapcsoljon (pl. 3), illetve azt az impulzusszámot, ahonnan már ki kell, hogy kapcsoljon (pl. 5) a számláló ezen időtartam alatt.



1.9.13 R ,Real Time Clock, valós idejű óra utasítások (R01 – R1F)

Legelőször a programban az **Operation/RTC Set...** menüpontban be kell állítani a pontos évet, hónapot, napot, órát, percet, téli-nyári időszámítás váltási dátumait. Ezt egyébként a készülék kezelőgombjaival is meg tudjuk tenni. A programban ezt követően 31 különböző RTC utasítást lehet használni. A készülék 4 RTC üzemmódot ismer.

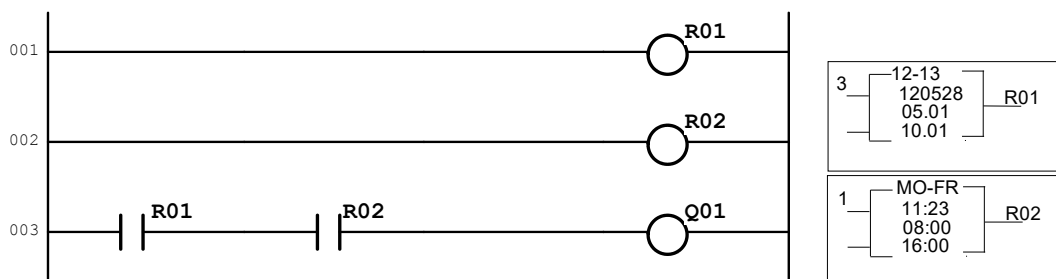
1. Mindennapos üzemmód. Meg lehet adni, hogy a hét melyik napjától melyik napjáig (pl.: hétfőtől péntekig) minden nap hány óra hány perckor kapcsoljon be és ki az RTC.

2. Hosszabb intervallum-mód. Meg lehet adni, hogy a hét melyik napján hány óra hány perckor kapcsoljon be és a hét egy másik napján hány óra hány perckor kapcsoljon ki az RTC.

3. Havi intervallum-mód: Meg lehet adni, hogy melyik év melyik hónap melyik napján kapcsoljon be, majd melyik év melyik hónap melyik napján kapcsoljon ki az RTC.

4. 30 másodperces mód: Meg lehet adni, hogy minden héten a hét egy kiválasztott napján hány óra hány perckor kapcsoljon be, és hány másodpercig legyen bekapcsolva. Ha 30 másodpercnél nagyobb értéket adunk meg, csak egy tüimpulzus kimenetet kapunk.

Az előre definiált RTC-k bemenetként kombinálhatóak.



A példában Q01 kimenet 2012. 05. 01.-től 2013. 10. 01-ig (3. mód) minden héten hétfőtől péntekig 08.00-16.00 időközben (1. mód) bekapcsol. Ezek a funkciók alkalmassá teszik a készüléket például iskolai csengető óra, vagy öntözőszivattyú vezérlésére.

1.9.14 G ,Comparator, komparátor, összehasonlítás (G01 – G1F)

A program 31 különálló komparátort tartalmazhat. Minden komparátor nyolc üzemmód szerint működhet. Az üzemmódokban három bemeneti paramétert (**Ax** bemenet, **Ay** bemenet, **Ref.** referencia érték) adhatunk meg.

A működési módok a következők: Az alábbi feltételekkor van G bekapcsolva.

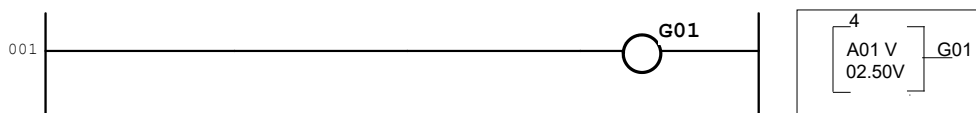
0. mód: egyszerű belső reléként használhatjuk

1. mód: $(Ay - \text{Ref.}) \leq Ax \leq (Ay + \text{Ref.})$;
2. mód: $Ax \leq Ay$;
3. mód: $Ax \geq Ay$;
4. mód: $Ax \leq \text{Ref.}$;
5. mód: $Ax \geq \text{Ref.}$;
6. mód: $Ax = \text{Ref.}$;
7. mód: $Ax \gg \text{Ref.}$;

1. példa: analóg jelek összehasonlítása

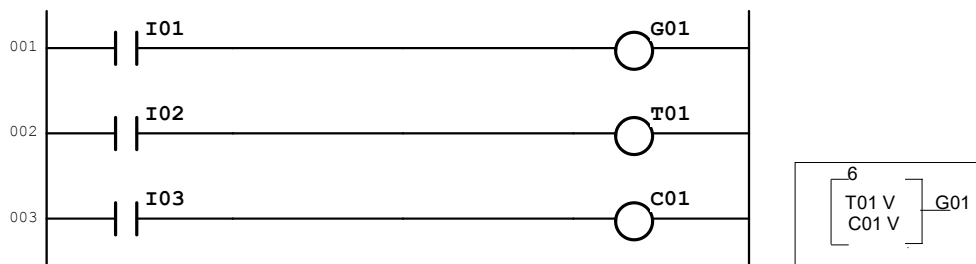
Az alábbi példában a 4. mód van kiválasztva ($Ax \leq \text{Ref.}$), amely összehasonlítja az $Ax=A01$ analóg bemenet értéket egy konstans (N) $\text{Ref.}=2,50 \text{ V}$ értékkel.

G1 tekercs akkor van bekapcsolva, amikor A01 analóg feszültség kisebb-egyenlő, mint 2,50 V konstans.



2. példa: Az időzítő és számláló jelenérték összehasonlítása

Az összehasonlítóval nem csak analóg bemeneti értékeket, referencia-értékeket, hanem időzítő, számláló, vagy más egyéb állandó, vagy változó értéket is összehasonlíthatunk.



Ebben a példában a komparátor 6. módban van ($Ax = \text{Ref}$), összehasonlítja az időzítő (T01) értékét a számláló (C01) értékével, és akkor kapcsol be G1 tekercs, ha (T01) késve behúzó tekercs ideje (szekundumban) éppen egyenlő a számláló aktuálisan elszámolt (darab) értékével pl.: I2 bekapcsolása óta 2 s telt el, és 2 db I3 bekapcsolás volt.

1.9.15 H, (Human-Machine Interfaces, ember-gép kapcsolat) kijelző utasítások

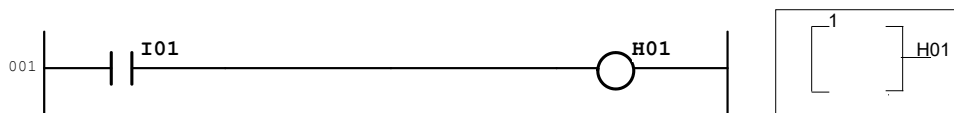
(természetesen, csak amelyik modellen van kijelző)

Az LCD kijelzőn maximum 4 sor, és egy soron 16 karakter jeleníthető meg.

Egy program összesen 31 db HMI utasítást tartalmazhat, amelyek két módot ismernek.

1. módban aktív a kijelző, 2. módban nem.

A HMI-nek a létradiagramban nem feltétlenül szükséges feltételt megadni, de ha megadunk, csak a feltétel teljesülésekor látható az LCD-n a kívánt karaktersorozat.

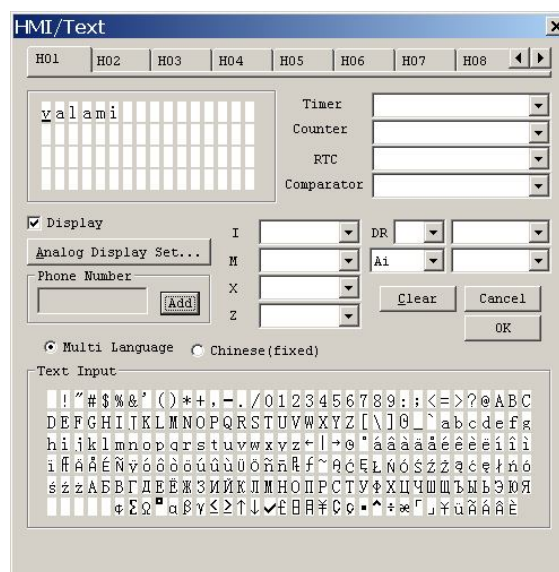


Azt, hogy mit jelezzen ki az LCD, az **Edit>>HMI/Text** menüpontban állíthatjuk be az alábbiak szerint.

A kijelzett karaktersorozat lehet tetszőleges szöveg, valamilyen feltétel esetén riasztási szöveg, a létradiagramban használt változók (bemenetek, kimenetek, belső memóriák, időzítők, számlálók, komparátorok, stb. előre beállított határértékei, vagy éppen aktuális értékei mértékegységgel, vagy nélküle). Az előre beállított határértékek (T, C, R, G és DR) a SEL gomb lenyomásával üzem közben is változtathatók, így például egy gyártási folyamat időzítésének megváltoztatásához egyszerűen itt lehet módosítani.

Megadható egy telefonszám is, amely valamely esemény hatására megjelenik az LCD-n, és segít a gépkezelőnek a szerviz elérésében.

A szöveg nyelve lehet többnyelvű (latin betűs, illetve a telepített PC billentyűzetkiosztásnak megfelelő) vagy kínai írásjeles.



1.9.16 P, Pulse Width Modulation, impulzus-szélesség moduláció kimenet (P01 – P02)

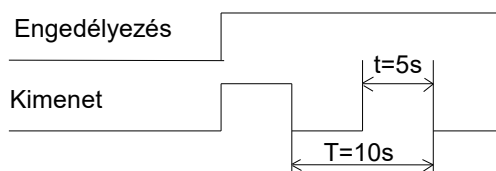
(csak a DC, tranzistoros kimenetű modelleken)

Bizonyos típusú villanymotorok fordulatszám-szabályzása megoldható úgy, hogy a tápfeszültséget csökkentjük. Ezt szabályzástechnikailag a legkönnyebb úgy megoldani, ha a négyzetű kimeneti jel kitöltési tényezőjét módosítjuk, csökkentjük.

Q01 kimenetet konfigurálhatjuk P01 PWM, vagy PLSY kimenetnek, Q02 kimenetet pedig csak P02 PWM kimenetnek.

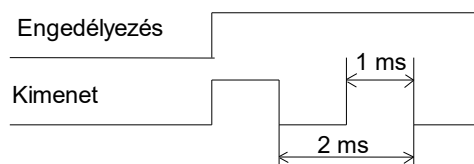
1. mód: PWM

Beállítható a négyzetű jel szélessége T (ms) és a kitöltött időszak t (ms). Ezek nem csak konstans értékek lehetnek, hanem a programban használt bármely változók, pl. számláló értéke.



2. mód: PLSY

Ebben az üzemmódban megadható a négyzetű jel frekvenciája (Hz), és a négyzetű jelek darabszáma, amit elérve a kimenet leáll. A kitöltési tényező itt 50%. Ilyen löketek sorozatával is szabályozhatók berendezések.



Például $PF = 500 \text{ Hz} = 2 \text{ ms}$, $PN = 5$ esetén Az engedélyező jel után a kimeneten 5 db 500 Hz-es 50% kitöltési tényezőjű négyzetű jel jelenik meg. PF és PN itt is lehet változó is.

1.9.17 Távoli I/O funkció

(csak SMT-CDxx modellek esetén)

Egy SMT PLC-hez legfeljebb további egy SMT egységet lehet beállítani, mint a távoli I/O pontot. Ekkor az egyik a mester, a másik a szolga PLC lesz.

A mester PLC-ben fut program, a szolgában viszont nem. Tulajdonképpen a szolga PLC úgy viselkedik, mintha a mester PLC kiterjesztett bemenete és kimenete lenne az alábbi összerendelési táblázat szerint.

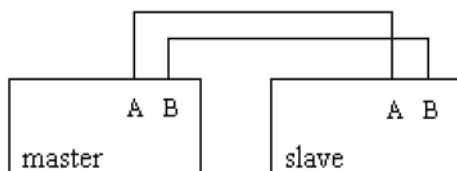
Természetesen a létradiagram készítésekor a kiterjesztett címeket kell beírni a programba, valójában azonban a szolga PLC be- és kimenetei fognak működni.

I/O cím	Mester	Szolga
Bemenetek	I01~I0C	
Kimenetek	Q01~Q08	
Kiterjesztett bemenetek	X01~X0C	I01~I0C
Kiterjesztett kimenetek	Y01~Y0C	Q01~Q08

Amikor a távoli I/O funkció engedélyezve van, nem használhatunk további bővítő DI / DO modulokat.

1.9.17.1 Hardveres beállítás

A PLC-k összekötése csavart érpárral az alábbi ábra szerint:



1.9.17.2 Szoftveres beállítás

Az **Operation»Module System Set** menüpontban

- Mindkét PLC ID azonosítója 01 legyen
- A mester PLC beállítása: Master
- A szolga PLC beállítása: Slave

Module System Set [X]

Set ID

Current ID:

New ID(00-99):

Remote I/O

NO

Master

Slave

Set Expand I/O

I/O Num: ▼

I/O Alarm

Others

M Keep

C Keep

Back Light

Z Set

RS485 Set

Comm. Mode: ▼

Baud Rate: ▼

DR Format Set

Unsigned

Signed

Coil Record

None M N

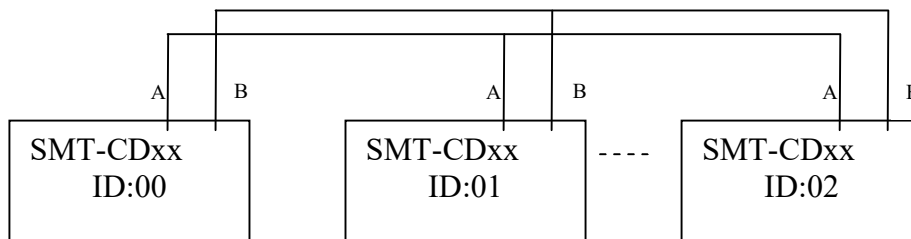
Range: ▼ SUM=0

1.9.18 IO Link funkció

Maximum 8 db SMT-CDxx típusú PLC-t tudunk ezzel a módszerrel hálózatba kötni RS 485 kapcsolaton keresztül.

1.9.18.1 Hardveres beállítás

A PLC-k összekötése csavart érpárral az alábbi ábra szerint:



1.9.18.2 Szoftveres beállítás

Az **Operation»Module System Set** menüpontban

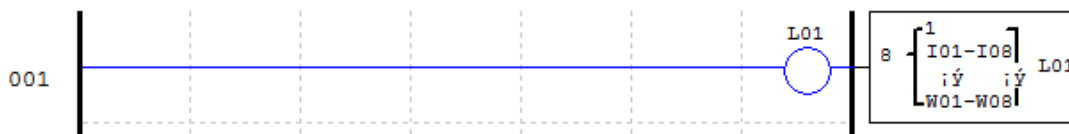
- Minden PLC-nek más ID azonosító kell.
- Az ID azonosítók csak sorban lehetnek, nem szabad kihagyni sorszámot pl.: ID:00, ID:01, ID:02, stb.
- A Remote I/O értéke: NO

A kapcsolattartás a „W” típusú memóriaterületeken keresztül történik. A 8 db PLC így összesen $8 \times 8 = 64$ címet használ az alábbi táblázat szerint.

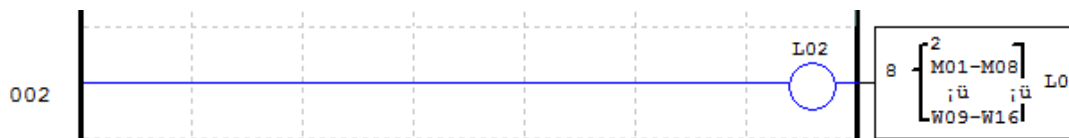
Minden létradiagramban egyszer szerepelhet a link 1. mód, vagyis a memóriába írás, de akárhányszor a link 2. mód, vagyis a kiolvasás.

ID:00	W01-W08
ID:01	W09 W16
ID:02	W17 W24
ID:03	W25 W32
ID:04	W33 W40
ID:05	W41 W48
ID:06	W49 W56
ID:07	W57 W64

Az alábbi ábrán az ID:00 azonosítójú PLC 1. módban kiírja az I01-I08 bemeneti értékeit a W01-W08 memóriaterületre. Nem feltétlenül fontos az összes bemenet átültetése”W” területre, megadható a kezdő sorszám és a darabszám is.



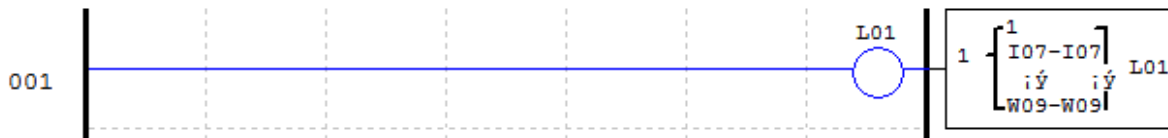
Ugyanezen ID:00 azonosítójú PLC az ID:01-es társából kiolvassa (2. mód) a W09-W16 memóriaterület értékeit, és áthelyezi M01-M08-ba. Onnan lehet látni, hogy az ID:01-ből olvas, hogy a címei W09-W16. Az így létrejövő M01-M08 értékeket a programban ezután bármilyen hivatkozásra felhasználhatjuk.



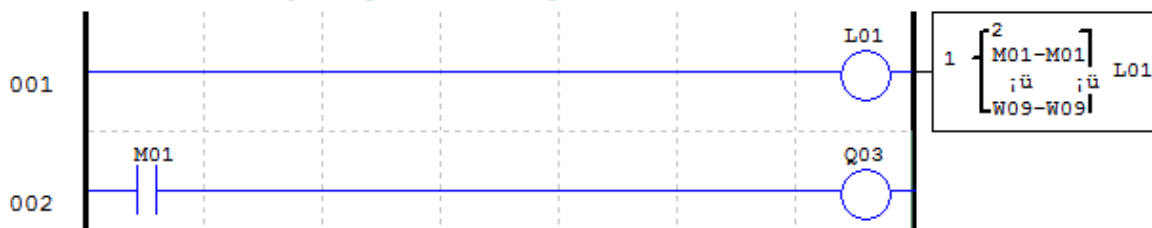
Példa

Összekötünk 2 db PLC-t I/O link segítségével. Ha az ID:01-es 7-es bemenetén egy nyomógombot lenyomok, kapcsoljon be az ID:00-ás készüléken a 3. kimenet!

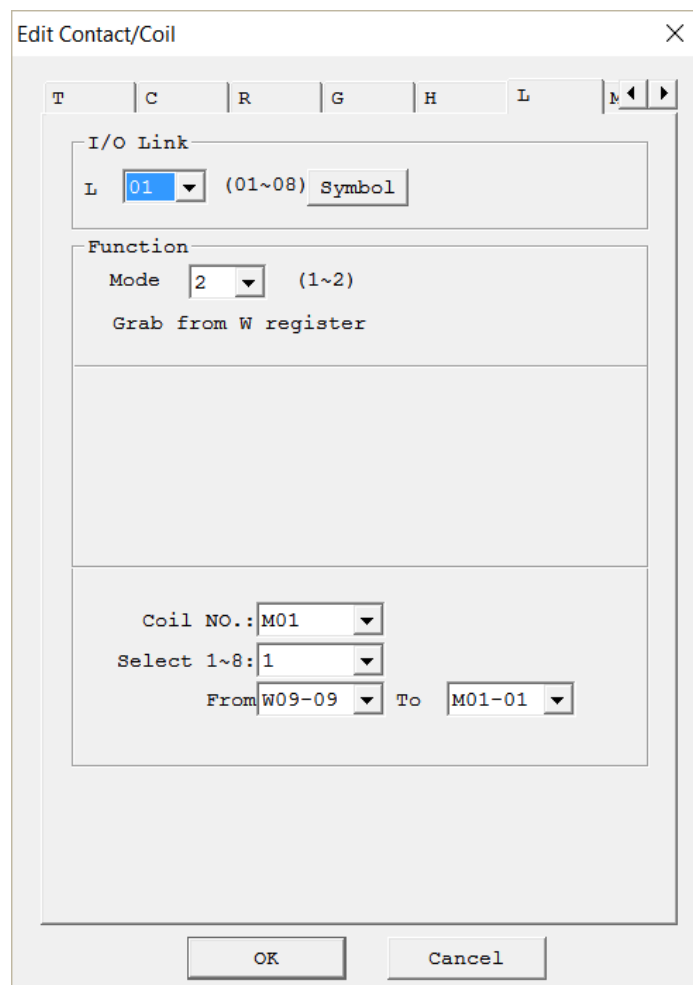
A program az ID:01-en:



A program az ID:00-en:

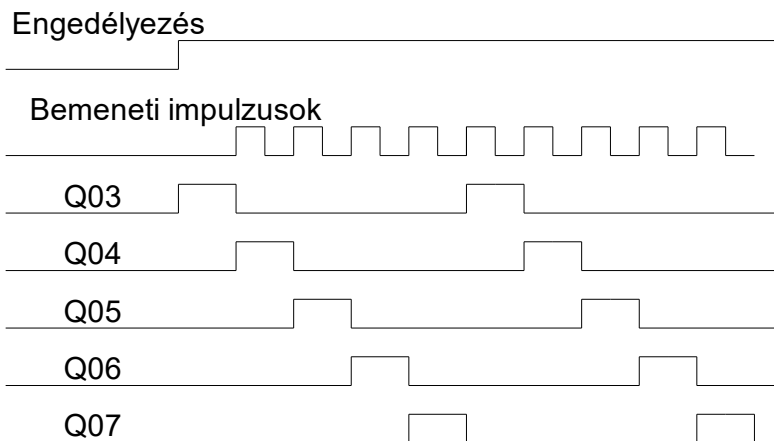
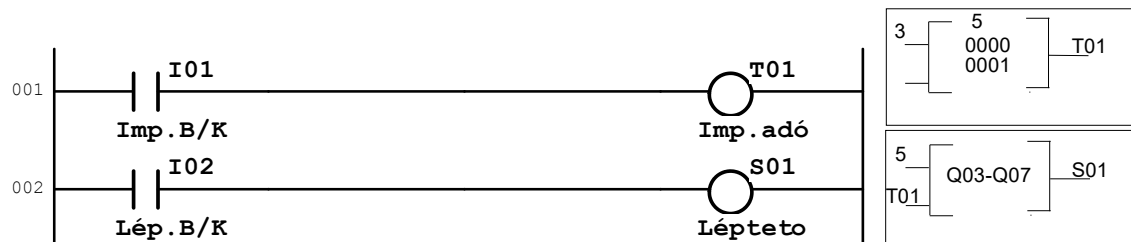


Nem feltétlenül fontos „I” értékét átírni „W”-be, lehet „I” ”Q” ”X” ”Y” ”M” ”N” is, továbbá „W” értékét beolvashatjuk „I” ”Q” ”X” ”Y” ”M” ”N”-be is.



1.9.19 S, SHIFT, elcsúsztatott kimenet (S01)

Különféle kimeneteket tud léptetni, de egy programban csak egy SHIFT utasítás használható. Meg kell adni a kimenet típusát (pl.: Q, M, N, ...), kezdő sorszámát (pl.: 03), és hogy hány db kimenet között legyen léptetés (pl.: 5). Ezen felül meg kell adni, hogy milyen bemenő jelre lépjen (pl.: T01 impulzus-adó).



1.10 Analóg eszközök

1.10.1 A, Analóg bemenet (A01 – A08)

Száma típustól változik (A01 ~ A08). Általában használhatók digitális bemenetnek is, ilyenkor minimum 10V feszültséget kell kapniuk.

1.10.2 V, Analóg bemenő paraméter (V01 ~ V08)

Lehet feszültség (0V – 10V), áram (0mA – 20mA) értékekre beállítani

1.10.3 AT, Hőmérsékletmérő bemenet, (AT01 ~ AT04)

Szabványos hőérzékelő modulok köthetők az analóg bemenetekre, ekkor ezekhez kell a szoftvert paraméterezni.

1.10.4 AQ, Analogue Output, analóg kimenet (AQ01 – AQ04)

Feszültség módban $AQ = 0-10V$ (AQ érték: 0 ~ 1000),

Áram módban $AQ = 0-20 \text{ mA}$ (AQ érték: 0 ~ 500)

AQ előre beállított értéke lehet állandó, vagy más funkció aktuális értéke.

1.10.5 AS, Add-Subtract, összeadás, kivonás (AS01 – AS1F)

A programban 31 db AS utasítás használható egyszerű összeadás, kivonás elvégzésére az egész számok körében.

A képlet: $AS = V1 + V2 - V3$

A paraméterek V1, V2, és V3 lehetnek állandók vagy más működési aktuális érték. A hibatekerics értéke akkor lesz ON, ha az eredmény túlsordul.

1.10.6 MD, Multiplication - Division, szorzás, osztás (MD01 – MD1F)

A programban 31 db MD utasítás használható egyszerű szorzás, osztás elvégzésére az egész számok körében.

A képlet: $MD = V1 * V2 / V3$

A paraméterek V1, V2, és V3 lehetnek állandók vagy más működési aktuális érték. A hibatekerics értéke akkor lesz ON, ha az eredmény túlsordul.

1.10.7 PID, arányos, integráló differenciáló szabályzó (PID01 – PID0F)

A P, PI, PD, PID szabályzókat különféle irányítási folyamatokban lehet felhasználni. Értelemszerűen, ha valamelyik tagot (P, I, D) nem kívánjuk használni, annak értékét nullán kell hagyni. A programban 15 PID utasítást lehet használni. A hibatekercs akkor kapcsol ON állásba, ha vagy $T_S = 0$ vagy $K_P = 0$.

1.10.8 MX, multiplexer(MX01 – MX0F)

Összesen 15 MX utasítást lehet használni egy programban. Az MX függvény lehetővé teszi, hogy egyszerű műveleteket lehessen elvégezni az egész számok körében. A paraméterek lehetnek állandók, vagy más funkció aktuális értékei.

1.10.9 AR, Analog Ramp, analóg rámpa (AR01 – AR0F)

Összesen 15 AR utasítást lehet használni egy programban. Az AR függvény lehetővé teszi, hogy egyszerű műveleteket lehessen elvégezni az egész számok körében. A paraméterek lehetnek állandók, vagy más funkció aktuális értékei.

1.11 DR, Data Register, adattároló (DR01 – DR0F)

Összesen 240 DR utasítást lehet használni egy programban. Bármilyen változó értékét le tudjuk tárolni ebben az átmeneti tárolóban.

1.12 MU, Modbus (MU01 – MU0F)

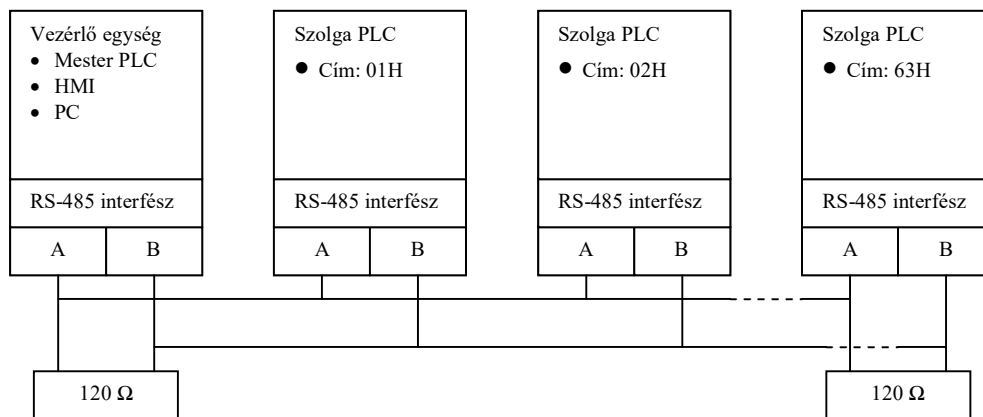
A Modbus a Modicon cég által fejlesztett hálózati kommunikációs protokoll, melyet elsősorban a programozható logikai vezérlők hálózatba kötéséhez terveztek. A Modbus protokollt főleg akkor érdemes használni, ha PC-t vagy HMI-t kötünk össze PLC-vel, és nem akarjuk a gyári kezelőszoftvereket használni, saját speciálist írunk helyette. Ebben az esetben legegyszerűbben ezen az úton jutunk a PLC adataihoz, illetve tudjuk azt vezérelni. A távoli I/O-val illetve a Link funkcióval ellentétben itt nem csak az I, Q, X, Y, M, N, pillanatnyi értékeihez, hanem **bármilyen adathoz** (pl.: időzítők pillanatnyi értékei, vagy beállított értékei) is hozzáférhetünk.

A Modbus egy központi lekérdezésű busz rendszer, amelyben egy Master vezérli az adatforgalmat a buszon. A Slave helyi egységek közötti keresztirányú adatforgalomra nincs lehetőség. Minden egyes adatcserét csak a Master kezdeményezhet kérés révén. A vezetéken egyidejűleg csak egy üzenet lehet. Slave nem kezdeményezhet adatátvitelt, hanem csak kérésre reagálhat, válasszal.

A Master és a Slave között két fajta párbeszéd lehetséges:

- A Master üzenetet küld az összes Slave-nek és nem vár arra választ (általános üzenetküldés, broadcast, ID=00).
- A Master üzenetet küld valamelyik Slave-nek és választ vár rá.

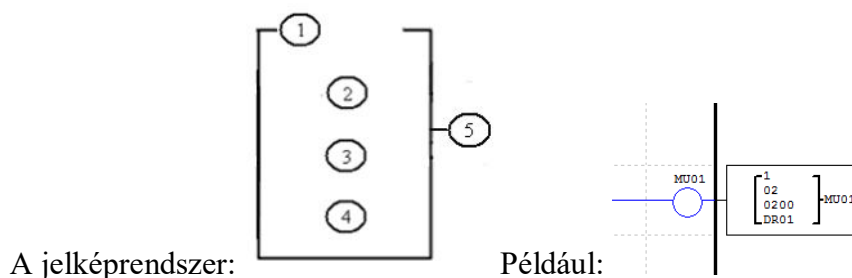
A Slave-ek hibaüzeneteit a Master ciklikusan kérdezi le. A reflexiók és az ahhoz kapcsolódó átviteli hibák elkerüléséhez a hálózati kábelt minden egyes fizikai végén 120 ohmos buszlezáró ellenállással ajánlatos lezárni.



Az üzenetek szerkezete a következőkből épül fel. A mester üzenete elején egy szolgál címet (**ID=01 ~ ID=7F**) vagy egy üzenetszóró címet (**ID=00**) tartalmaz, ezt követi egy funkció kód amely a szolgál által végrehajtandó utasítást írja elő (**mód**), majd a kommunikációs tartalom következik amiben a szolgálak küldhetünk adatot és az egészet egy hiba ellenőrzéshez szükséges blokk zárja (CRC-16).

A szolgál lekérdezésre történő válasza a következő: A mester által küldött parancs jóváhagyása, majd az igényelt adat, végül a hibaellenőrzéshez szükséges mező (CRC-16) zárja az üzenetet.

Modbusz mód	Kommunikáció fajtája
1	egy regiszter olvasása
2	egy regiszter írása
3	több regiszter írása
4	egy bit olvasása
5	egy bit írása



A példában MU01 azonosítójú Modbus utasítás egy regisztert olvas (1. mód), az ID 02 azonosítójú szolga PLC-ből, T1 időzítő aktuális értékét (0200) letárolja DR 01 regiszterben.

Szimbólum	Jelentés
①	MODBUS mód 1~5 (5 darab)
②	A szolga PLC ID száma , értéke 00~7F (128 darab)
③	Kommunikációs tartalom: 0000~FFFF közé eső konstans, ami <u>gépkönyvi táblázatokból</u> beazonosítható. Pl.: 0200 a T1 időzítő aktuális értékét, 0550 az I1~I24 digitális bemenetek aktuális állapotát mutatja...
④	DR 01~F0 (240 darab) adattároló a kiolvasott adatok letárolódik ezen a DR-en (read mód), vagy az adatok ebbe a DR-be íródnak (write mód)
⑤	MODBUS azonosító MU01~MU0F (240 darab)

Az értékeket mindig **hexadecimális** formában kell beírni!

1.13 Kompatibilitás az iSmart PLC készülék, memóriakártyák és az SMT Client szoftverek között

- A SMT-xx-x10 vagy SMT-xx-x12 I/O pontos készülékekre írt program kompatibilis a SMT-xx-x20 I/O készülékekkel, de fordítva nem!
- Az SMT-xA-xxx típusú készülékekre írt program kompatibilis a SMT-xD-xxx készülékekkel, de fordítva nem!
- Az relés kimeneti típusú (SMT-xx-Rxx) készülékekre írt program kompatibilis a tranzisztoros kimenetű (SMT-xx-Txx) készülékekkel, de fordítva nem!
- Az SMT-Ex-xxx típusú készülékekre írt program kompatibilis az SMT-Cx-xxx készülékekkel, de fordítva nem!
- A 2. szoftver-verzióban megírt program kompatibilis a 3. verzióban megírt programmal, de fordítva nem!
- Az SMT PM04 memóriakártya csak a 2. verzióval, míg az SMT PM04 (3rd) a 2. és a 3. verzióval is kompatibilis!

2 PÉLDATÁR

2.1 Gyakorló példák programozáshoz

A következő oldalakon néhány gyakorló feladatot szeretnék bemutatni a programozásához. Található közöttük kapcsolásvezérlés, Bool algebrai logikai-, idővezérelt lefutó-, folyamatvezérelt lefutó vezérlés is. A könnyebektől folyamatosan juthatunk el a nehezebbekig. A feladatokban szerepel:

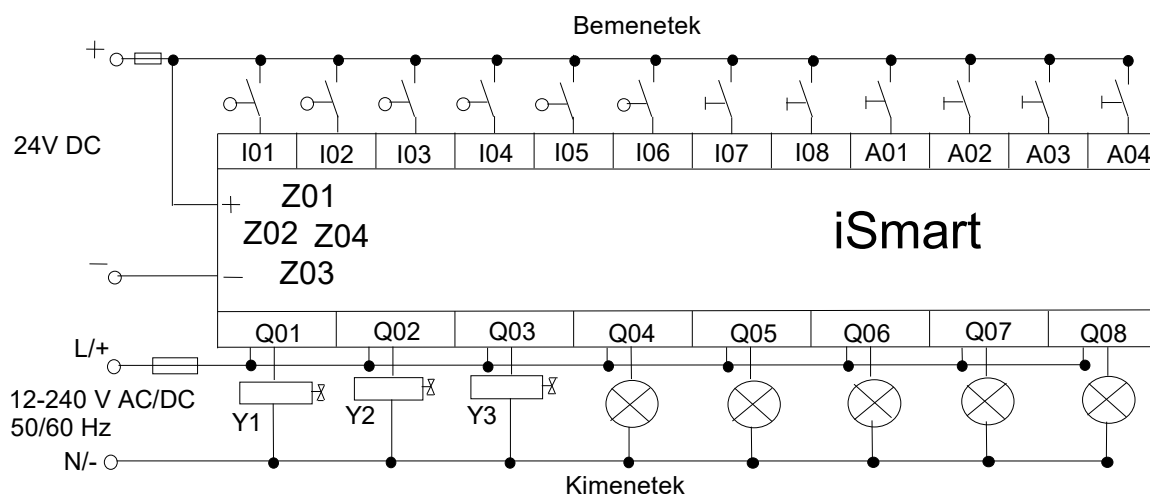
- A feladat sorszáma
- A probléma szöveges megfogalmazása
- A feladat típusa
- Az algebrai egyenlet (logikai példánál)
- Időrendi táblázat (idővezérelt lefutó példánál)
- Út-lépés követődiagram impulzusanalízissel (folyamatvezérelt lefutó példánál)
- Az összerendelési táblázat
- A létradiagram
- Szöveges magyarázat a nehezebben követhető dolgokhoz

A bekötési vázlat nem szerepel minden példa után, mert az összerendelési lista alapján és az általános bekötési vázlat példája alapján ugyanis mindenki könnyen elkészítheti. Ennél a készüléksaládnál néhány bemenetet többféleképpen jelölnek:

Digitális cím	I09	I10	I11	I12
Analóg cím	A01	A02	A03	A04
Hexadecimális cím	I09	I0A	I0B	I0C

Az analóg bemenetek 10V fölött úgy viselkednek, mint digitális bemenetek, a programok a hivatkozásokat így is, úgy is elfogadják.

Általános bekötési vázlat:



2.1.1 feladat

Igen kapcsolat (YES). Az I07-es bemeneti nyomógombot lenyomva legyen aktív a Q01-es kimeneten egy lámpa, amíg a gombot nyomva tartjuk!

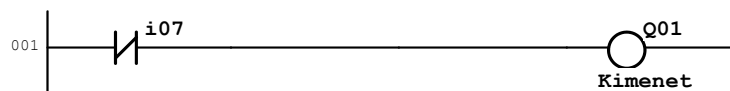
Egyenlet: $\{I07=Q01\}$;



2.1.2 feladat

Nem kapcsolat (NOT). Ha nem nyomom le az I07-es nyomógombot, legyen aktív a Q01-es kimeneten egy lámpa, amíg lenyomom, ne legyen aktív!

Egyenlet: $\{\overline{I07} = Q01\}$



2.1.3 feladat

Több kimenet egyidejű kapcsolása. Az I07-es bemeneti nyomógombot lenyomva legyen aktív a Q01-es és a Q02-es kimeneten egy-egy lámpa, amíg a gombot nyomva tartjuk!

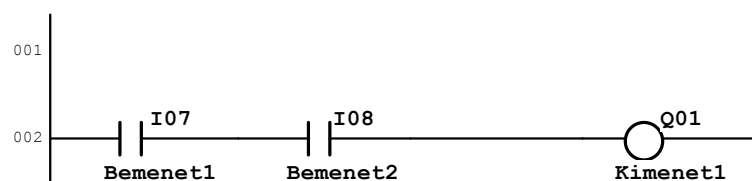
Egyenlet: $\{I07 = Q01 \bullet Q02\}$



2.1.4 feladat

És kapcsolat (AND). Az I07-es és az I08-as bemeneti nyomógombot lenyomva legyen aktív a Q01-es kimeneten egy lámpa, amíg a gombokat nyomva tartjuk!

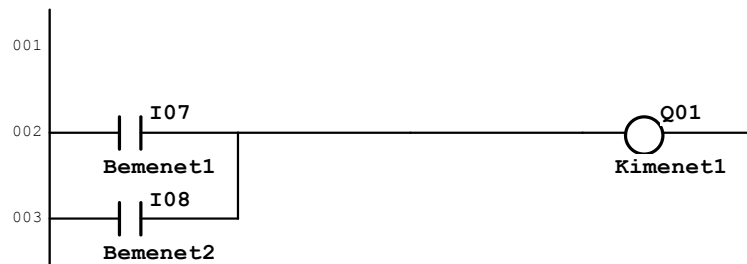
Egyenlet: $\{I07 \bullet I08 = Q01\}$



2.1.5 feladat

Vagy kapcsolat (OR). Az I07-es vagy az I08-as bemeneti nyomógombot lenyomva legyen aktív a Q01-es kimeneten egy lámpa, amíg a gombokat nyomva tartjuk!

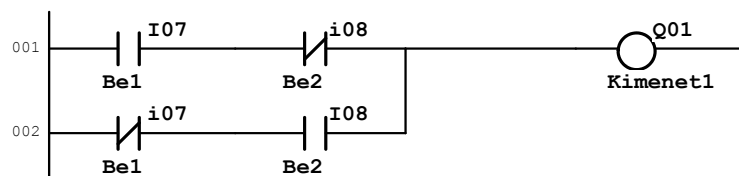
Egyenlet: $\{I07+I08=Q01\}$



2.1.6 feladat

Kizáró vagy kapcsolat (XOR). Az I07-es vagy az I08-as bemeneti nyomógombot lenyomva legyen aktív a Q01-es kimeneten egy lámpa, amíg a gombokat nyomva tartjuk, de ha mindkettőt lenyomjuk, ne!

Egyenlet: $\{(I07 \cdot \overline{I08}) + (\overline{I07} \cdot I08) = Q01\}$

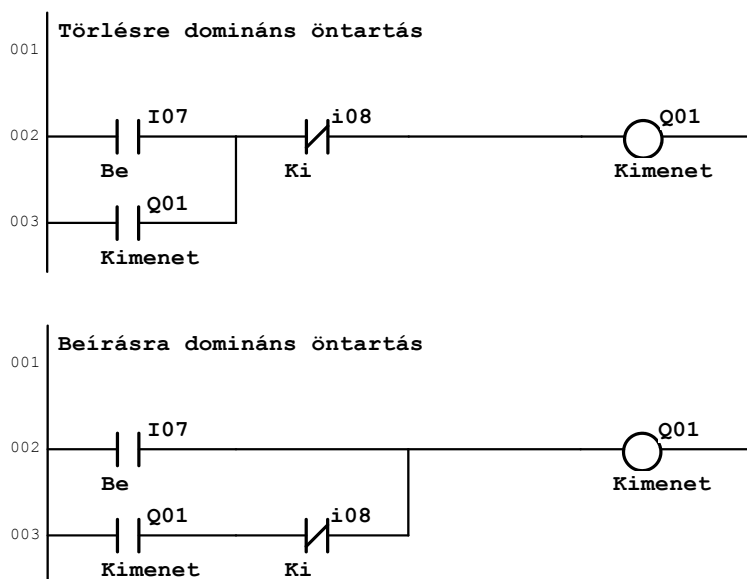


2.1.7 feladat

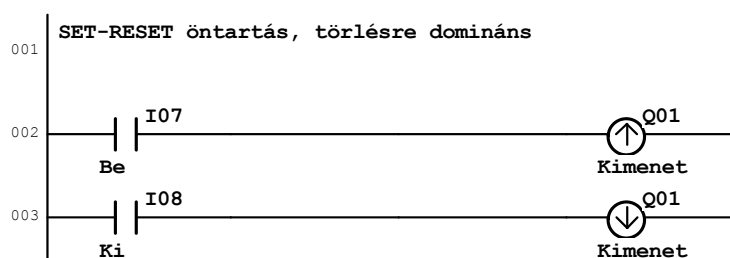
Öntartás, R-S memóriatároló (SET/RESET). Az I07-es bemeneti nyomógombot lenyomva legyen aktív a Q01-es kimeneten egy lámpa, és maradjon is úgy, amíg az I08-as bemeneti nyomógomb lenyomásával ki nem kapcsoljuk!

Egyenlet: $\{I07=Q01\}$ és $\{I08=\overline{Q01}\}$

A valós relés kapcsolásokhoz hasonlóan külön elemekből megépíthetjük a két tipikus öntartás kapcsolását.



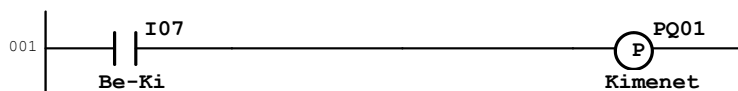
A másik mód a SET-RESET öntartás.



A SET bemenetre adott logikai "1" hatására a kimenetként megadott változó logikai "1"-be billen, és állapotát megtartja, míg a RESET bemenetre logikai "1"-t nem kap. A SET és RESET feltételek egyidejű teljesülése esetén a RESET-nek van prioritása. Megjegyzendő, hogy fordított paraméterezés esetén, vagyis ha a reset van felül, a set van alul, beírásra domináns lesz, mert mindig az utolsó utasítást hajtja végre a program!

2.1.8 feladat

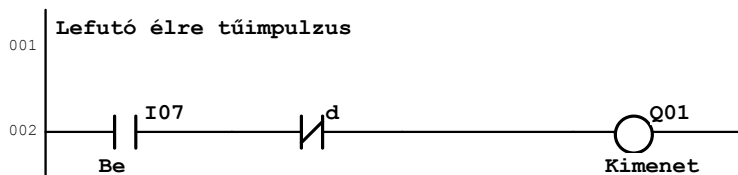
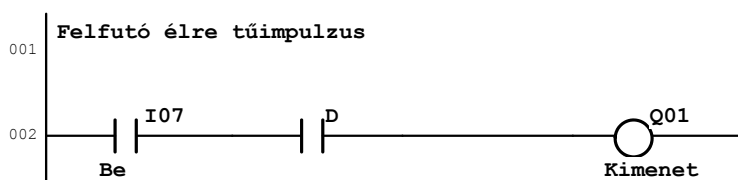
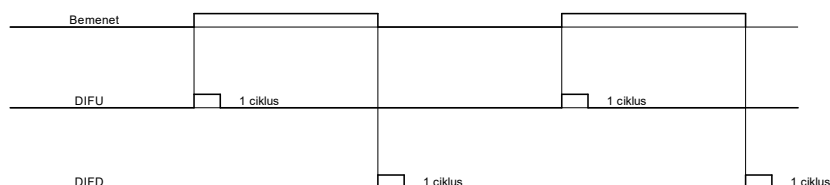
Flip-Flop. Csupán az I07-es nyomógombbal, egy lenyomással kapcsoljuk be Q01-et, míg a második lenyomással kapcsoljuk ki!



Ehhez a megoldáshoz a Q01 kimenetet „P” üzemmódba kell állítani.

2.1.9 feladat

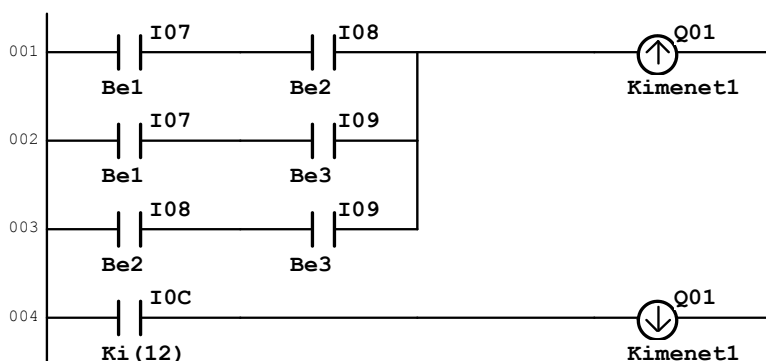
Felfutó, illetve lefutó élre túimpulzus. Hozzunk létre az I07 bekapcsolásakor egy ciklusidőnyi túimpulzust Q01-en, illetve kikapcsolásakor szintén egy ciklusidőnyi túimpulzust Q01-en az alábbi működési diagramok szerint!



2.1.10 feladat

Összetett Boole-algebrai feladat I. Az I07-es, I08-as, I09-es bemeneti nyomógombok közül bármely kettőt vagy mindhármat lenyomva legyen aktív a Q01s kimeneten egy lámpa, és maradjon is úgy, amíg az I12-es bemeneti nyomógomb lenyomásával ki nem kapcsoljuk!

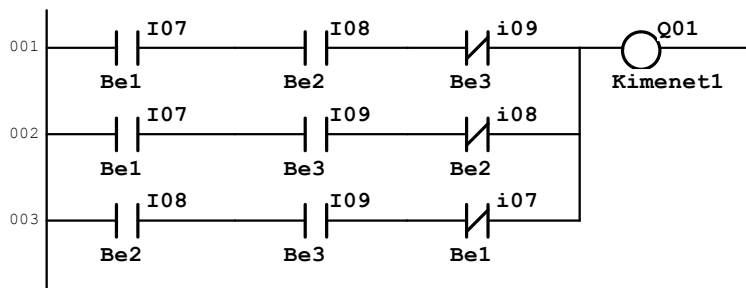
$$\text{Egyenlet: } \{(I07 \bullet I08) + (I07 \bullet I09) + (I08 \bullet I09) = Q01\}; I12 = \overline{Q01}$$



2.1.11 feladat

Összetett Boole-algebrai feladat II. Az I07-es, I08-as, I09-es bemeneti nyomógombok közül bármely kettőt lenyomva legyen aktív a Q01s kimeneten egy lámpa, de mindhármat lenyomva ne!

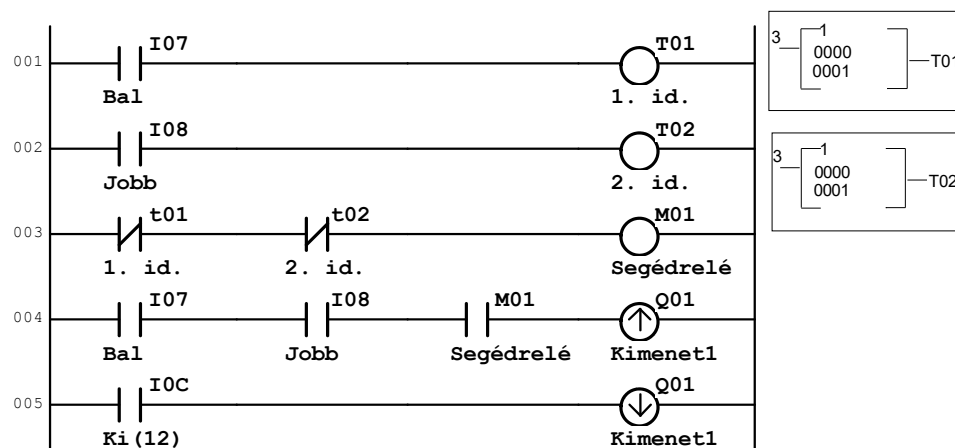
$$\text{Egyenlet: } \{(I07 \bullet I08 \bullet \overline{I09}) + (I07 \bullet I09 \bullet \overline{I08}) + (I08 \bullet I09 \bullet \overline{I07})\} = Q01$$



2.1.12 feladat

Biztonsági kétkezes indító. Az I07-es (bal kéz) és az I08-as (jobb kéz) bemeneti nyomógombot lenyomva legyen aktív a Q01-as kimeneten egy lámpa, és maradjon is úgy, amíg az I12-es bemeneti nyomógomb lenyomásával ki nem kapcsoljuk! A kimenet csak akkor legyen aktív, ha a két nyomógombot 1 másodpercen belül egyszerre nyomjuk le!

$$\text{Egyenlet: } \{I07 \bullet I08 \bullet \overline{T1} \bullet \overline{T2}\} = Q10 \text{ és } \{I12 = \overline{Q01}\}$$



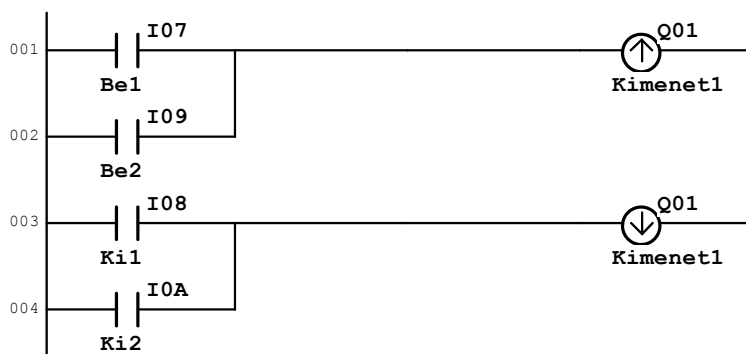
T01 és T02 időzítők 1-es módban vannak, időzítésük 1 s. Segédrelé azért kell, mert ennél a készüléktípusnál 4 db „ha” feltételt nem lehet sorba kötni (kevés a hely).

A feladathoz nem elegendő a két bemenet sorba kapcsolása, hanem még két időtagra is szükség van. Ezek negáltját sorba kötve a bekapcsoló gombokkal csak addig engedik a bekapcsolást, amíg a bennük beállított idő le nem telik. Célszerű a két időt azonosra venni.

2.1.13 feladat

Alternatív kapcsolás. Egy lámpát két helyről kell tudni bekapcsolni illetve, kikapcsolni. Az I07-es vagy I09-es bemeneti nyomógombot lenyomva legyen aktív a Q01-es kimeneten egy lámpa, és maradjon is úgy, amíg a I08-as vagy I10-es bemeneti nyomógomb lenyomásával ki nem kapcsoljuk! Az I07-es és az I08-as az egyik helyiségben van, az I09 és az I10 a mási helyiségben van.

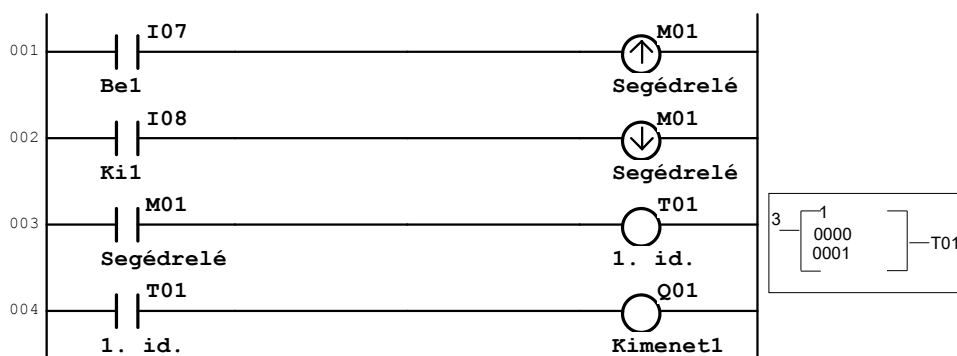
Egyenlet: $\{I07+I09=Q01\}$ és $\{I08+I10=\overline{Q01}\}$



2.1.14 feladat

Késleltetett bekapcsolás. Az I07-es bemeneti nyomógombot lenyomva T01 időkésleltetéssel legyen aktív a Q01-es kimeneten egy lámpa, és maradjon is úgy, amíg az I08-as bemeneti nyomógomb lenyomásával késedelem nélkül ki nem kapcsoljuk!

Egyenlet: $\{I07 \bullet T1=Q01\}$ és $\{I08=\overline{Q01}\}$

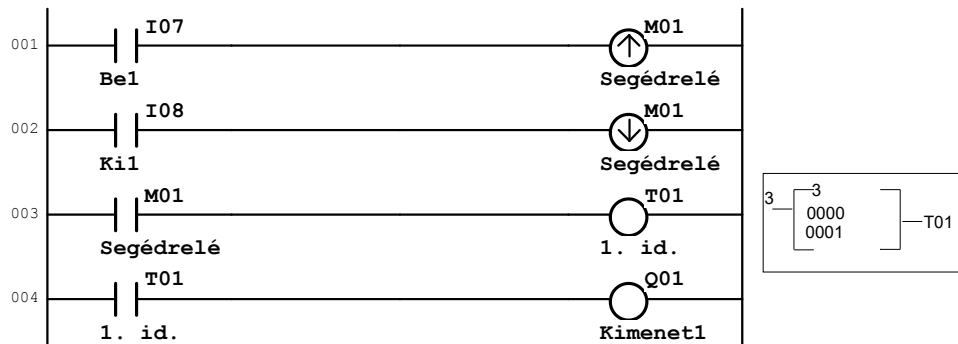


A T01 időrelé 1. módban van, 1 s a késleltetése.

2.1.15 feladat

Késleltetett kikapcsolás. Az I07-es bemeneti nyomógombot lenyomva késedelem nélkül legyen aktív a Q01-es kimeneten egy lámpa, és maradjon is úgy, amíg az I08-as bemeneti nyomógomb lenyomásával T1 időkésleltetéssel ki nem kapcsoljuk!

Egyenlet: $\{I07 = Q01\}$ és $\{I08 \bullet T1 = \overline{Q01}\}$

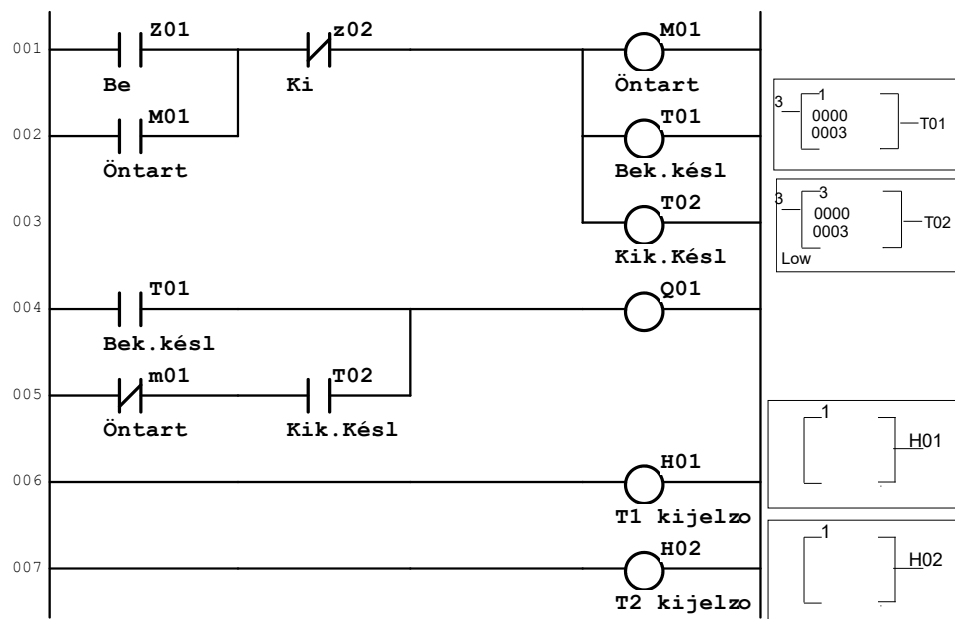


A T01 időrelé 3. módban van, 1 s késleltetéssel.

2.1.16 feladat

Késleltetett bekapcsolás és kikapcsolás. A Z01-es bemeneti nyomógombot lenyomva T01 = 3 s időkésleltetéssel legyen aktív a Q01-es kimeneten egy lámpa, és maradjon is úgy, amíg a Z02-es bemeneti nyomógomb lenyomásával T02 = 3 s időkésleltetéssel ki nem kapcsoljuk! T01 és T02 elnevezéseit, beállított és aktuális értékeit jelenítsük meg a készülék LCD kijelzőjén.

Egyenlet: $\{Z01 \bullet T01 = Q01\}$ és $\{Z02 \bullet T02 = \overline{Q01}\}$

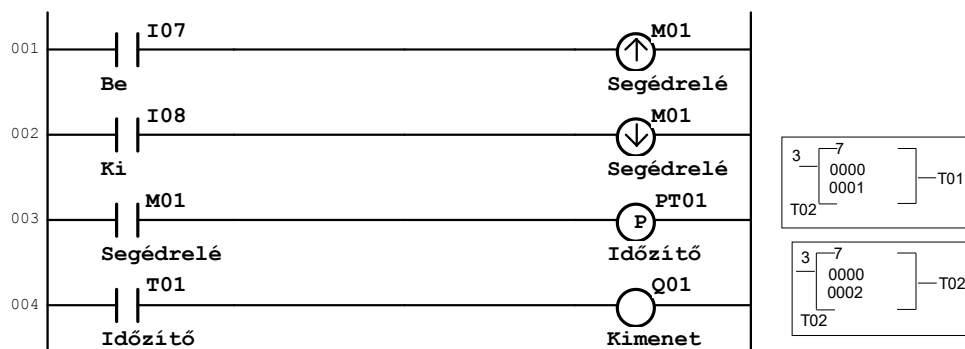


T01 1. módban van, értéke 3 s, T02 3. módban van, értéke 3 s, H01 és H02 engedélyezve van. H01 és H02 megjelenítése most nincs feltételhez kötve, a megjelenítendő szöveget, értékeket a HMI/Text felületen tudjuk beállítani. Ekkor a készülék kezelőfelületén átprogramozás nélkül állítható T01 és T02 kívánatos értéke.

2.1.17 feladat

Oszcillátor I. Az I07-es bemeneti nyomógombot lenyomva oszcilláljon a Q01-es kimeneten (1 s bekapcsolva, 2 s kikapcsolva), amíg az I08-as bemeneti nyomógomb lenyomásával kapcsoljuk!

Típus: Idővezérelt lefutó vezérlés

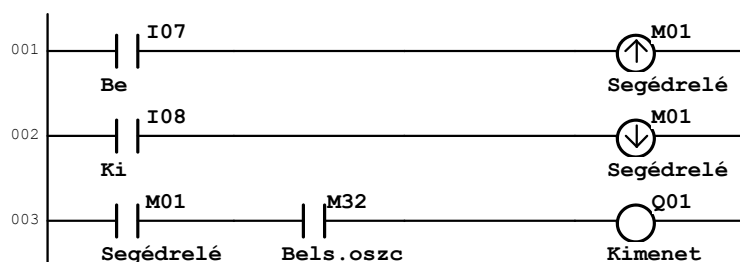


Az időzítő „P” üzemmódban úgy oszcillál, hogy külön állítható a bekapcsolási és a kikapcsolási ideje.

2.1.18 feladat

Oszcillátor II. Az I07-es bemeneti nyomógombot lenyomva oszcilláljon a Q01-es kimeneten (0,5 s bekapcsolva, 0,5 s kikapcsolva), amíg az I08-as bemeneti nyomógomb lenyomásával kapcsoljuk!

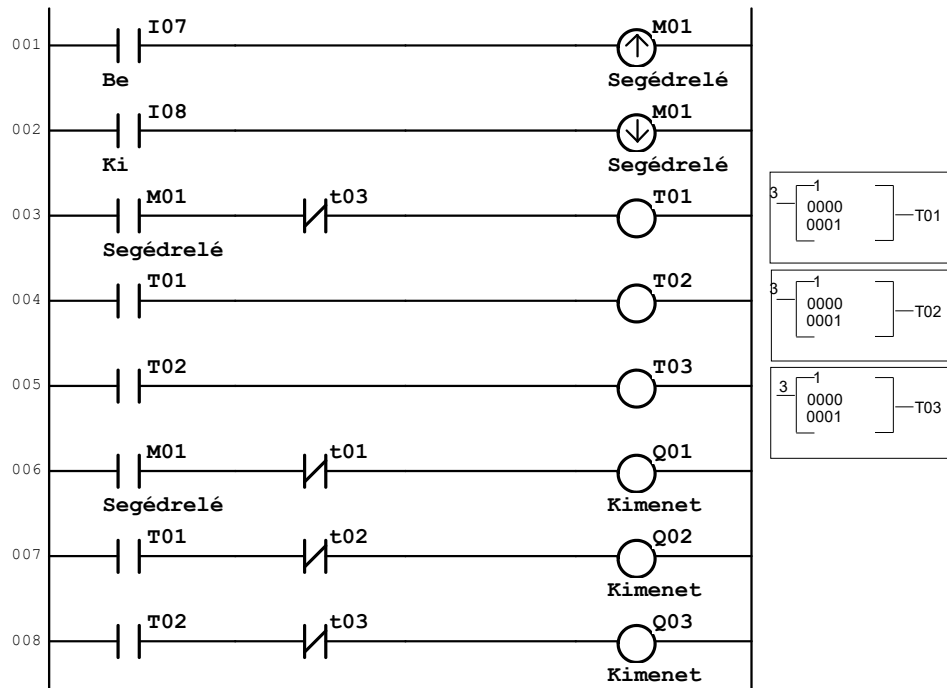
Típus: Idővezérelt lefutó vezérlés



Az M32 belső segédrelé 1 s periódusidejű oszcillátor.

2.1.19 feladat

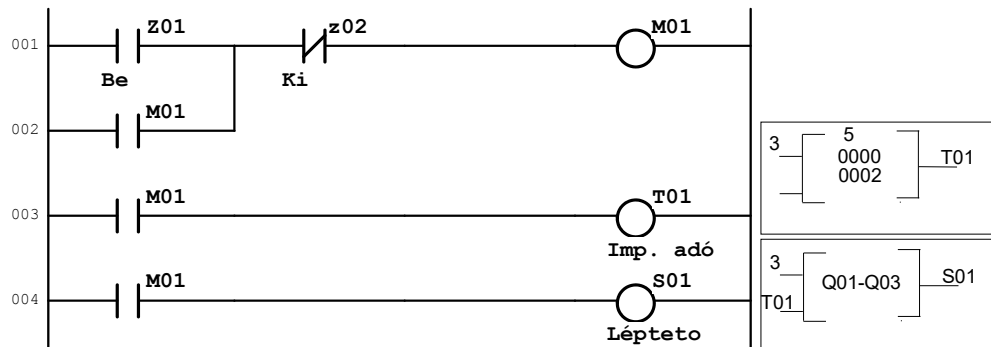
Futófény I. Készítsünk futófényt úgy, hogy a Q01, Q02, Q03 kimenetek egyenként kapcsolódjanak fel és le egymás után! A futófényt az I07 nyomógomb kapcsolja be, az I08 nyomógomb kapcsolja ki! **Típus:** Idővezérelt lefutó vezérlés



Mindhárom időzítő 1. módban van, értékük egyforma (1 s).

2.1.20 feladat

Futófény II. Készítsünk futófényt SHIFT utasítással úgy, hogy a Q01, Q02, Q03 kimenetek egyenként kapcsolódjanak fel és le egymás után! A futófényt az I07 nyomógomb kapcsolja be, az I08 nyomógomb kapcsolja ki! **Típus:** Idővezérelt lefutó vezérlés.



Z01 és Z02 a PLC homloklapján lévő nyílógombok. Ezek programozhatók nyitó-, és záróérintkezőnek is, így kalakítható velük egy öntartás kezelése is. T01 5. módban működő impulzusadó, beállított üteme a lépések közötti idő. S01 T01 ütemére léptet Q01-től Q 03-ig.

2.1.21 feladat

Készítsük el a **2 fényes gyalogos közlekedési jelzőlámpa** programját! A feladat 4 fő részre tagolható:

Be-, kikapcsolási rész.

Ezt egy öntartással célszerű megoldani.

Idő-léptetőlánc rész.

A működés során három esemény van: folyamatos piros fény, folyamatos zöld fény, villogó zöld fény. Ezért egy háromtagú idő - léptetőláncot kell alkalmazni. Ez azt teszi, hogy sorban, egymás után bekapcsolja az időzítőket, majd az utolsóval az egészet egyszerre lekapcsolja. Ekkor kezd az egészet, előlről. A programozott idők a valós jelzőlámpa időinél rövidebbek, tesztelésre valók, de barmikor megnövelhetők.

T01	5 s	Folyamatos piros
T02	2 s	Folyamatos zöld
T03	4 s	Villogó zöld

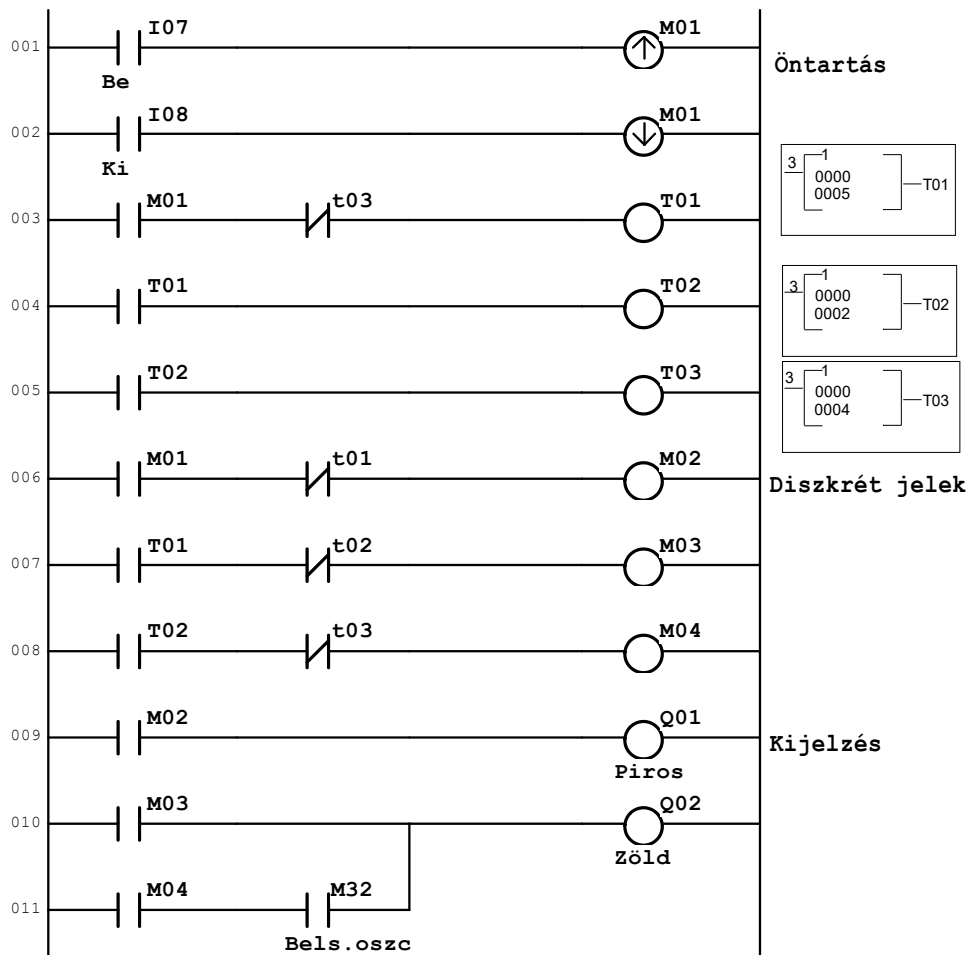
Diszkrét idők elkülönítése.

Erre a részre azért van szükség, hogy például a T01 idő elteltével, amikor T02 bekapcsolja a hozzá rendelt eseményt, a T01-hez rendelt esemény szűnjön meg. Ha ezt nem tennék meg, a ciklus végén egyszerre üzemelne minden esemény.

Itt arra kell vigyázni, hogy minden valós kimenetet csak egyszer programozhatunk, ezért az adott kimenetekhez tartozó feltételeket kell csokorba szedni VAGY utasításokkal.

Típus: Idővezérelt lefutó vezérlés

Bemenetek összerendelési listája												
Bemeneti cím	I01	I02	I03	I04	I05	I06	I07	I08	I09	I10	I11	I12
Funkció							Be	Ki				
Kimenetek összerendelési listája:												
Kimeneti cím	Q01	Q02	Q03	Q04	Q05	Q06	Q07	Q08				
Funkció	Piros l.	Zöld l.										



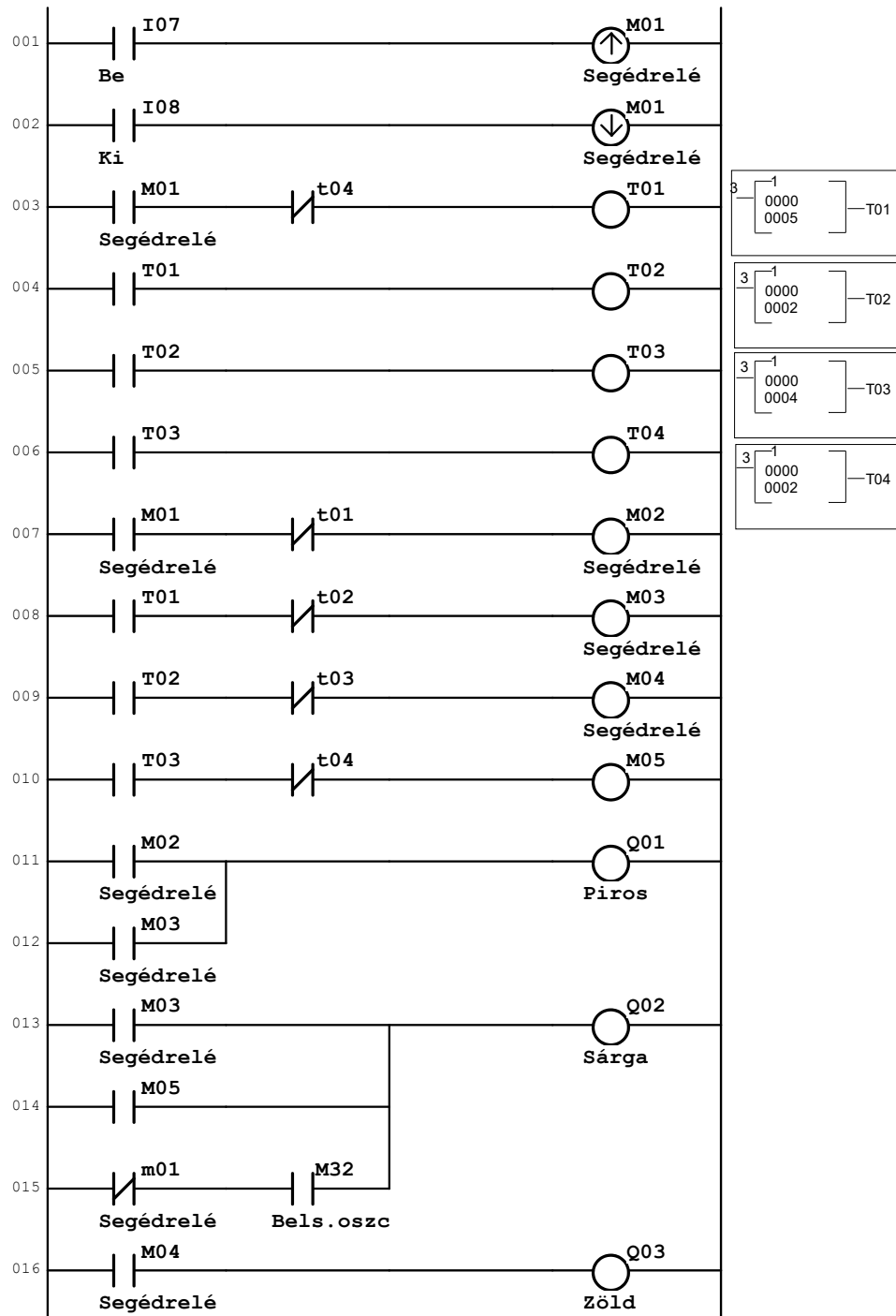
2.1.22 feladat

Készítsük el a **háromfényes közlekedési jelzőlámpa** programját! A feladat az előzőhöz hasonlóan négy fő részre tagolható. A feladat annyival bonyolultabb, hogy most négy esemény van, valamint kikapcsolt állapotban a sárga lámpának villognia kell.

T01	5 s	Piros
T02	2 s	Piros-sárga
T03	4 s	Zöld
T04	2 s	Sárga

Típus: Idővezérelt lefutó vezérlés. Minden időzítő 1. módban van.

Bemenetek összerendelési listája												
Bemeneti cím	I01	I02	I03	I04	I05	I06	I07	I08	I09	I10	I11	I12
Funkció							Be	Ki				
Kimenetek összerendelési listája:												
Kimeneti cím	Q01	Q02	Q03	Q04	Q05	Q06	Q07	Q08				
Funkció	Piros l.	Sárga l.	Zöld l.									



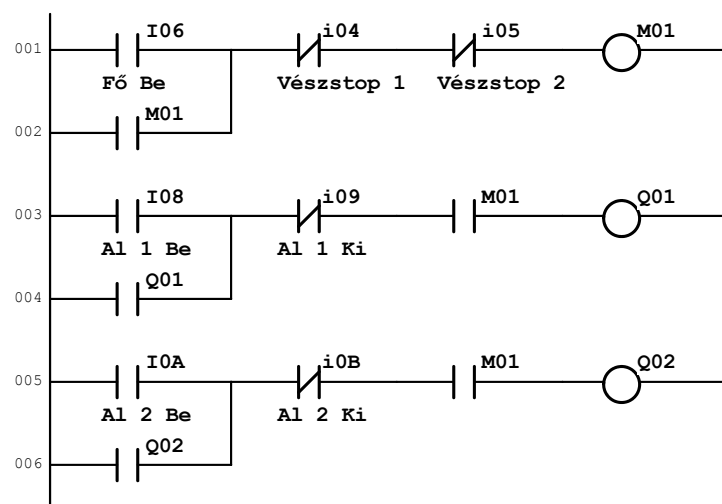
2.1.23 feladat

Főkapcsoló, vészstop kapcsolók, aláramkörök.

I06-os nyomógombbal lehessen bekapcsolni, I04-es, I05-ös vészstop nyomógombok közül bármelyikkel lehessen kikapcsolni egy főáramkört. Két egymástól független aláramkör legyen ez alá rendelve. Az elsőt az I08-as nyomógomb kapcsolja be, az I09-es ki, a másodikat az I10-es nyomógomb kapcsolja be, az I11-es ki. Az aláramkörök csak akkor kapcsolhatók be, ha a főáramkör már be van kapcsolva. A főáramkör kikapcsolásakor az esetlegesen bekapcsolt aláramkörök is kikapcsolódnak. Az aláramkörök bekapcsoltságát a Q01-es, illetve a Q02-es kimeneten egy lámpa jelezze!

Típus: Kapcsolásvezérlés

Bemenetek összerendelési listája												
Bemeneti cím	I01	I02	I03	I04	I05	I06		I08	I09	I10	I11	I12
Funkció				Fő ki	Fő ki	Fő be		A11 be	A11 ki	A12 be	A12 ki	
Kimenetek összerendelési listája:												
Kimeneti cím	Q01	Q02	Q03	Q04	Q05	Q06	Q07	Q08				
Funkció	Lámpa	Lámpa										



2.1.24 feladat

Kötött sorrendű bekapcsolások és kikapcsolások.

Q01-es kimenetet az I06-os kapcsolja be, az I11-es ki.

Q02-es kimenetet az I07-es kapcsolja be, az I10-es ki.

Q03-as kimenetet az I08-as kapcsolja be, az I09-es ki.

Q02-t csak akkor lehet bekapcsolni, ha a Q01-es már be van.

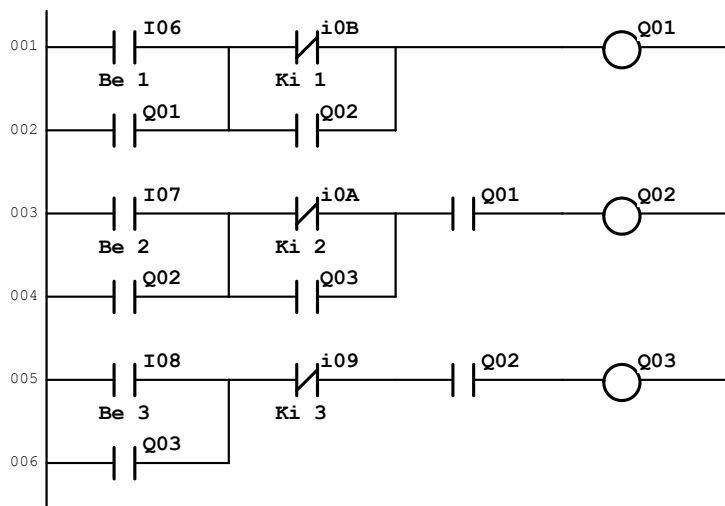
Q03-at csak akkor lehet bekapcsolni, ha a Q02-es már be van.

Q02-t csak akkor lehet kikapcsolni, ha a Q03-as már ki van.

Q01-et csak akkor lehet kikapcsolni, ha a Q02-es már ki van.

Típus: Kapcsolásvezérlés

Bemenetek összerendelési listája												
Bemeneti cím	I01	I02	I03	I04	I05	I06	I07	I08	I09	I10	I11	I12
Funkció						Be 1	Be 2	Be 3	Ki 3	Ki 2	Ki 1	
Kimenetek összerendelési listája:												
Kimeneti cím	Q01	Q02	Q03	Q04	Q05	Q06	Q07	Q08				
Funkció	Lámpa	Lámpa	Lámpa									

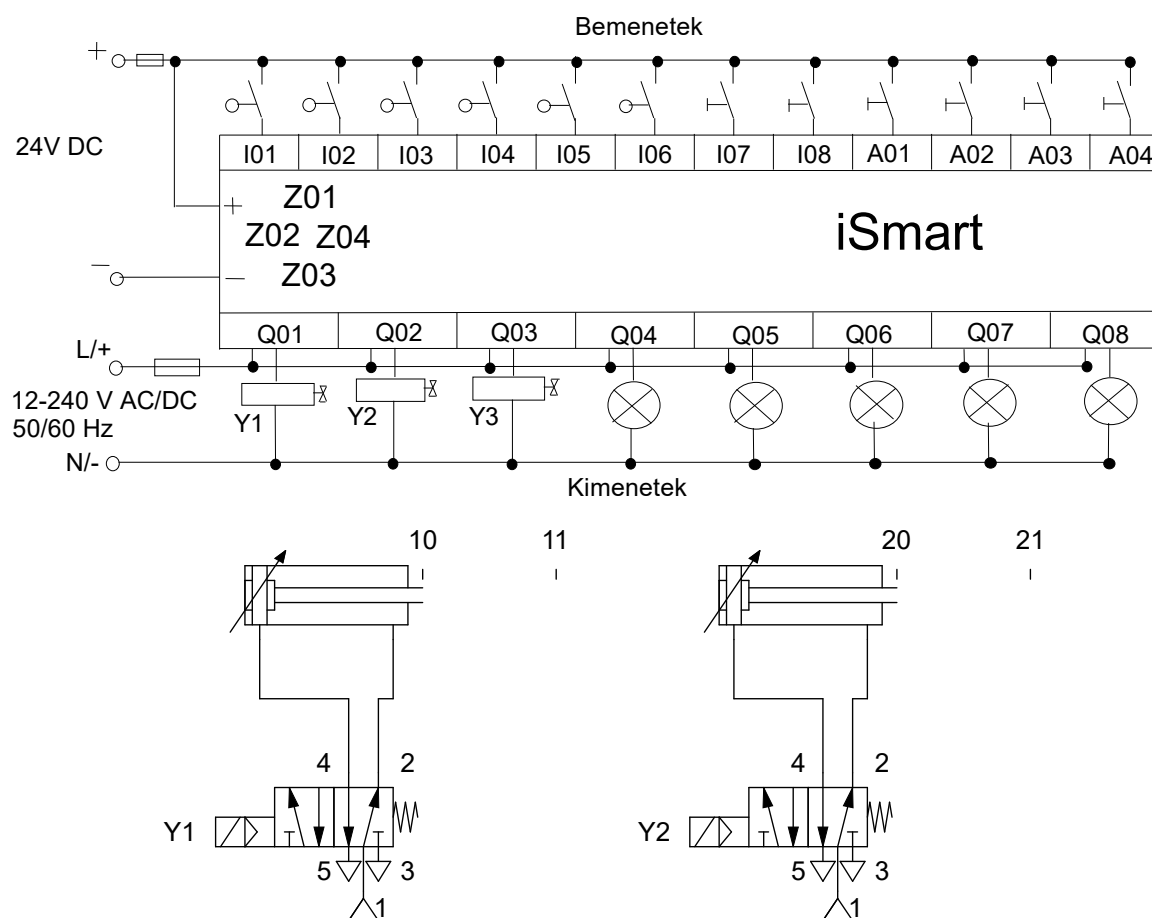


A következő néhány példában folyamatvezérelt lefutó vezérlések programjait mutatom be.

Közös jellemzőik:

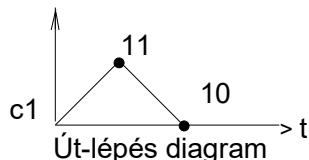
- A PLC huzalozását nem kell megbontani a különféle mozgások előállításához.
- A vezérlő nyomógombok funkcióit igyekeztem a különböző programokban nem változtatni, de ez csak egy általam kiválasztott lehetséges kiosztás, másképpen is lehetne, hiszen a bemenetek egymással egyenrangúak.
- A vezérlésekhez **monostabil 5/2-es főszelepeket** és kettősműködésű munkahengereket használtam végrehajtó elemként.
- Érzékelőknek egyszerű alaphelyzetben nyitott elektromos végállaskapcsolókat választottam.
- A mozgásokat a követődiagramjaikkal jellemeztem.
- A mozgások két munkahengeresek.

Az általános bekötési vázlat:



2.1.25 feladat

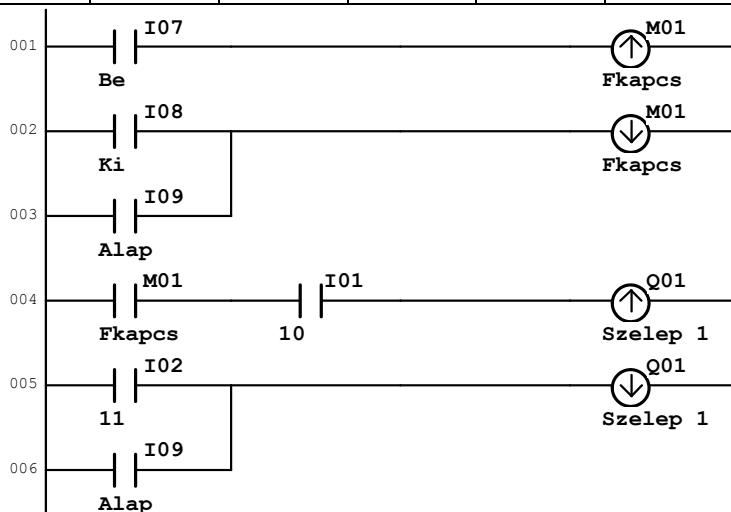
Egy dugattyú alternáló mozgása a következő út-lépés diagram alapján.



Az impulzussorrend: 10 11

Blokkoló jelek: Nincsenek

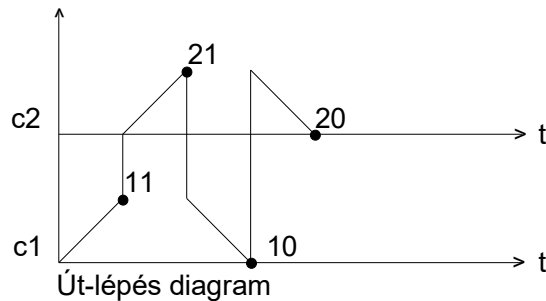
Bemenetek összerendelési listája												
Bemeneti cím	I01	I02	I03	I04	I05	I06	I07	I08	I09	I10	I11	I12
Funkció	10	11					Be	Ki	Alap			
Kimenetek összerendelési listája:												
Kimeneti cím	Q01	Q02	Q03	Q04	Q05	Q06	Q07	Q08				
Funkció	Y1											



A be-, és kikapcsolhatóságot az M01-es memóriaterület öntartásával oldhatjuk meg. Az öntartások RESET (alsó) lábaira párhuzamosan kötött I09-es alaphelyzetbe hozó nyomógombbal bármilyen üzemzavar esetén az egész rendszert alaphelyzetbe tudjuk hozni.

2.1.26 feladat

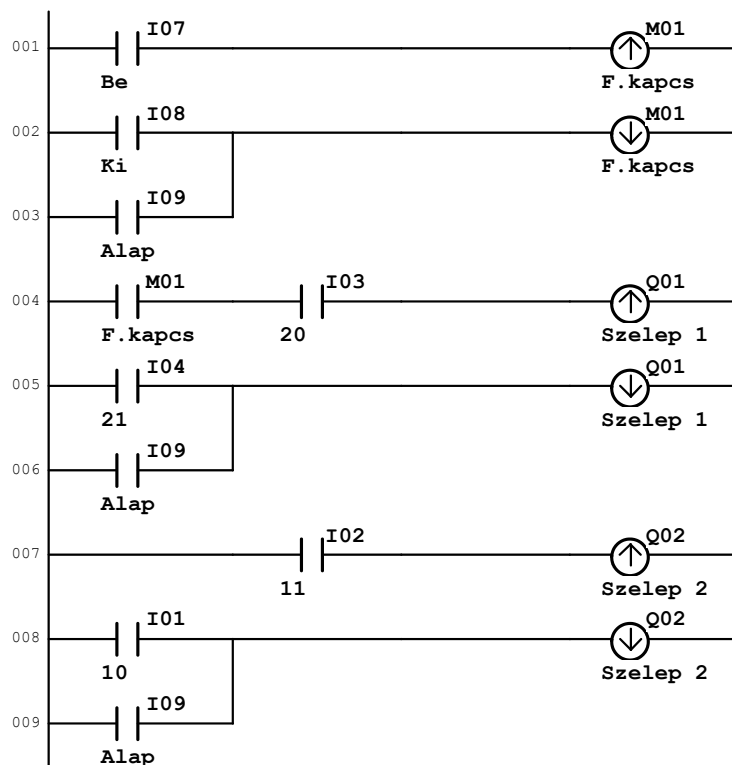
Két dugattyú alternáló mozgása a következő út-lépés diagram alapján.



Az impulzussorrend: 20 11 21 10

Blokkoló jelek: Nincsenek

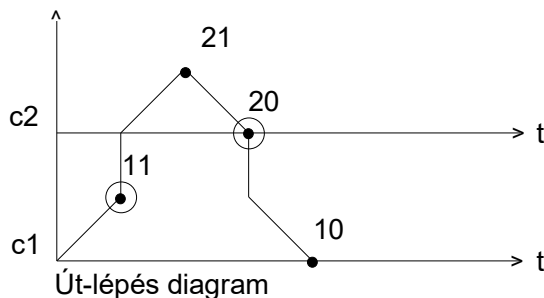
Bemenetek összerendelési listája												
Bemeneti cím	I01	I02	I03	I04	I05	I06	I07	I08	I09	I10	I11	I12
Funkció	10	11	20	21			Be	Ki	Alap			
Kimenetek összerendelési listája:												
Kimeneti cím	Q01	Q02	Q03	Q04	Q05	Q06	Q07	Q08	Q09	Q10	Q11	Q12
Funkció	Y1	Y2										



Az előző feladathoz képest más az impulzussorrend, és a két munkahengert két öntartással kell vezérelni.

2.1.27 feladat

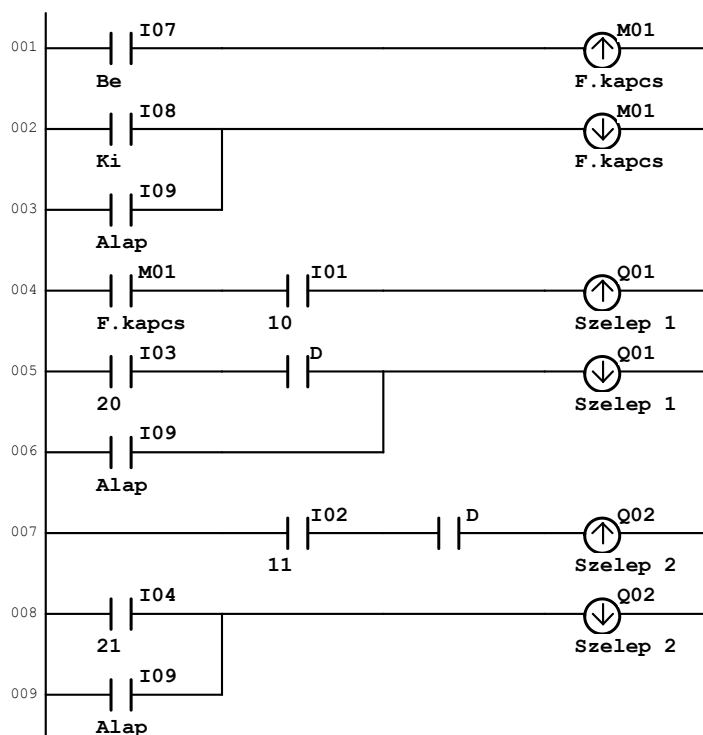
Két dugattyú alternáló mozgása a következő út-lépés diagram alapján.



Az impulzussorrend: 10 11 21 20

Blokkoló jelek: 11 20

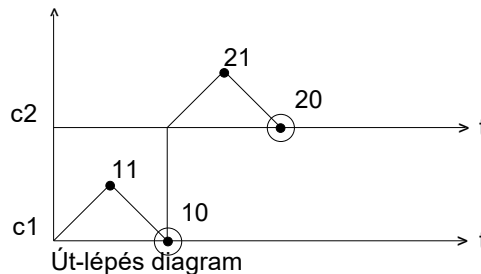
Bemenetek összerendelési listája												
Bemeneti cím	I01	I02	I03	I04	I05	I06	I07	I08	I09	I10	I11	I12
Funkció	10	11 Diff	20 Diff	21			Be	Ki	Alap			
Kimenetek összerendelési listája:												
Kimeneti cím	Q01	Q02	Q03	Q04	Q05	Q06	Q07	Q08				
Funkció	Y1	Y2										



Az előző feladathoz képest más az impulzussorrend és két blokkoló jel is megjelenik, amelyeket differenciáló utasításokkal lehet megszüntetni.

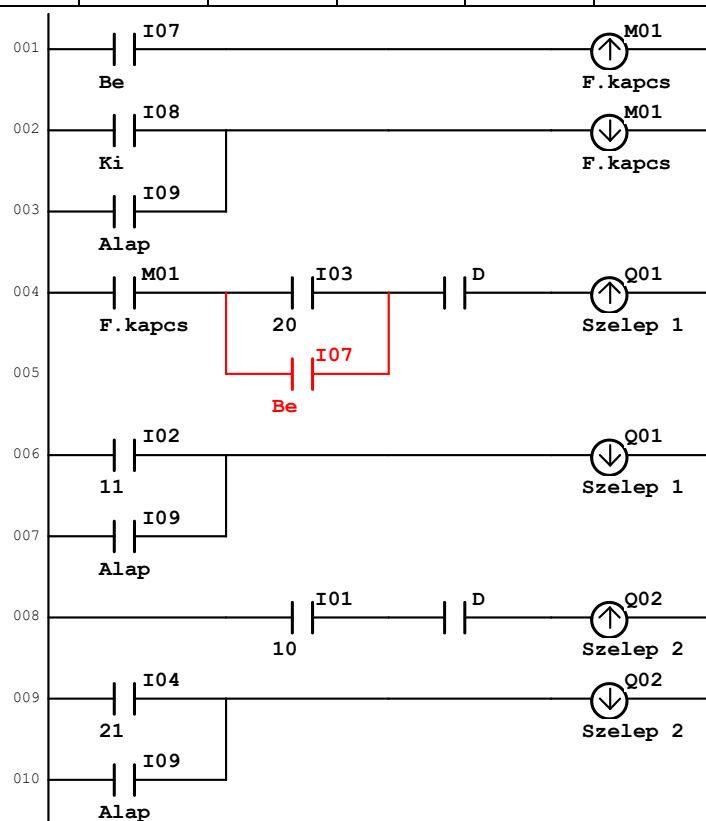
2.1.28 feladat

Két dugattyú alternáló mozgása a következő út-lépés diagram alapján.



Az impulzussorrend: 20 11 10 21
 Blokkoló jelek: 20 10

Bemenetek összerendelési listája												
Bemeneti cím	I01	I02	I03	I04	I05	I06	I07	I08	I09	I10	I11	I12
Funkció	10	11	20	21			Be	Ki	Alap			
Megjegyzés	Diff		Diff									
Kimenetek összerendelési listája:												
Kimeneti cím	Q01	Q02	Q03	Q04	Q05	Q06	Q07	Q08				
Funkció	Y1	Y2										

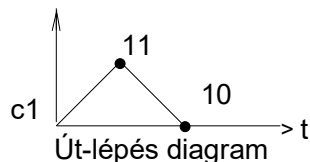


Az előző feladathoz képest más az impulzussorrend és két blokkoló jel is megjelenik, amelyeket differenciálással szüntetünk meg. További újdonság, hogy a sorban az első impulzus (I03) blokkoló, ezért az elindíthatósághoz szükséges, hogy vele párhuzamosan kössünk egy I07-es bekapcsoló kontaktust.

2.1.29 feladat

Egy dugattyú alternáló mozgása a következő út-lépés diagram alapján. A program három alternálás után automatikusan álljon le!

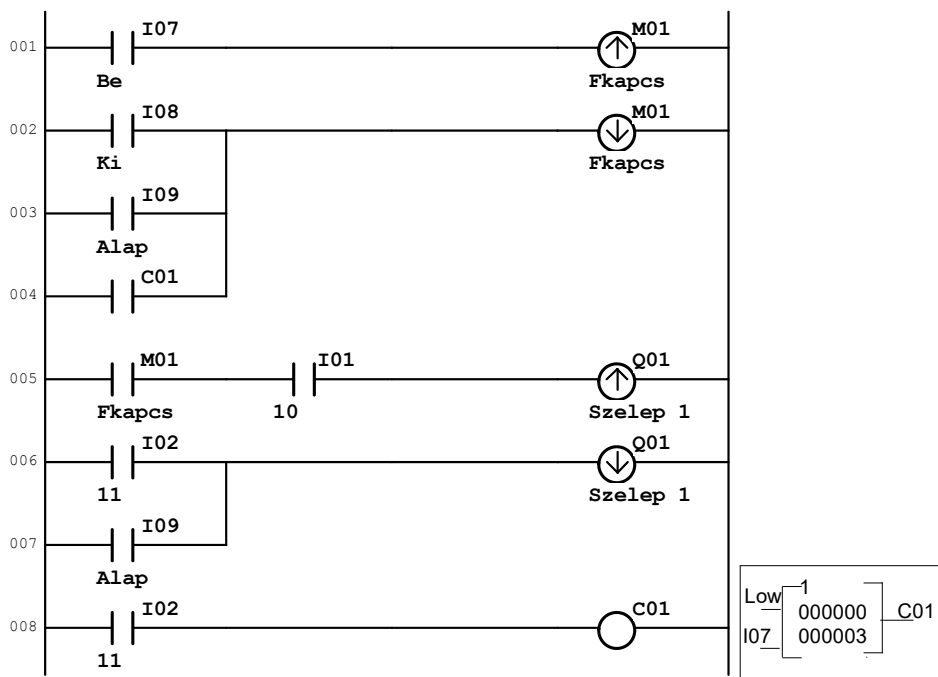
Típus: Ciklusok számlálása.



Az impulzussorrend: 10 11

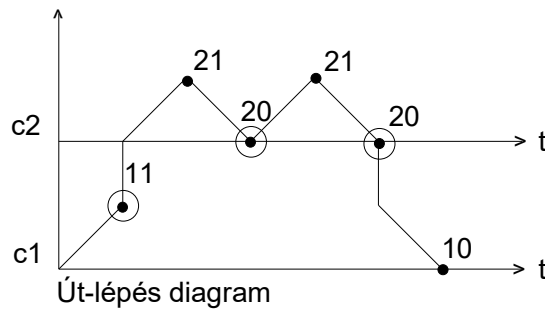
Blokkoló jelek: Nincsenek

Bemenetek összerendelési listája												
Bemeneti cím	I01	I02	I03	I04	I05	I06	I07	I08	I09	I10	I11	I12
Funkció	10	11					Be	Ki	Alap			
Kimenetek összerendelési listája:												
Kimeneti cím	Q01	Q02	Q03	Q04	Q05	Q06	Q07	Q08				
Funkció	Y1											



C01 számlálóval számoljuk, hányszor volt kint a dugattyú (11 nevű érzékelő). A számlálандó mindig tüimpulzus kell, hogy legyen! A számlálót (1.mód) a bekapcsoló gomb (I07) nullázza. Ha a számláló elszámol 3-ig, aktívva válik és lekapcsolja a fő (M01) öntartást.

2.1.30 feladat

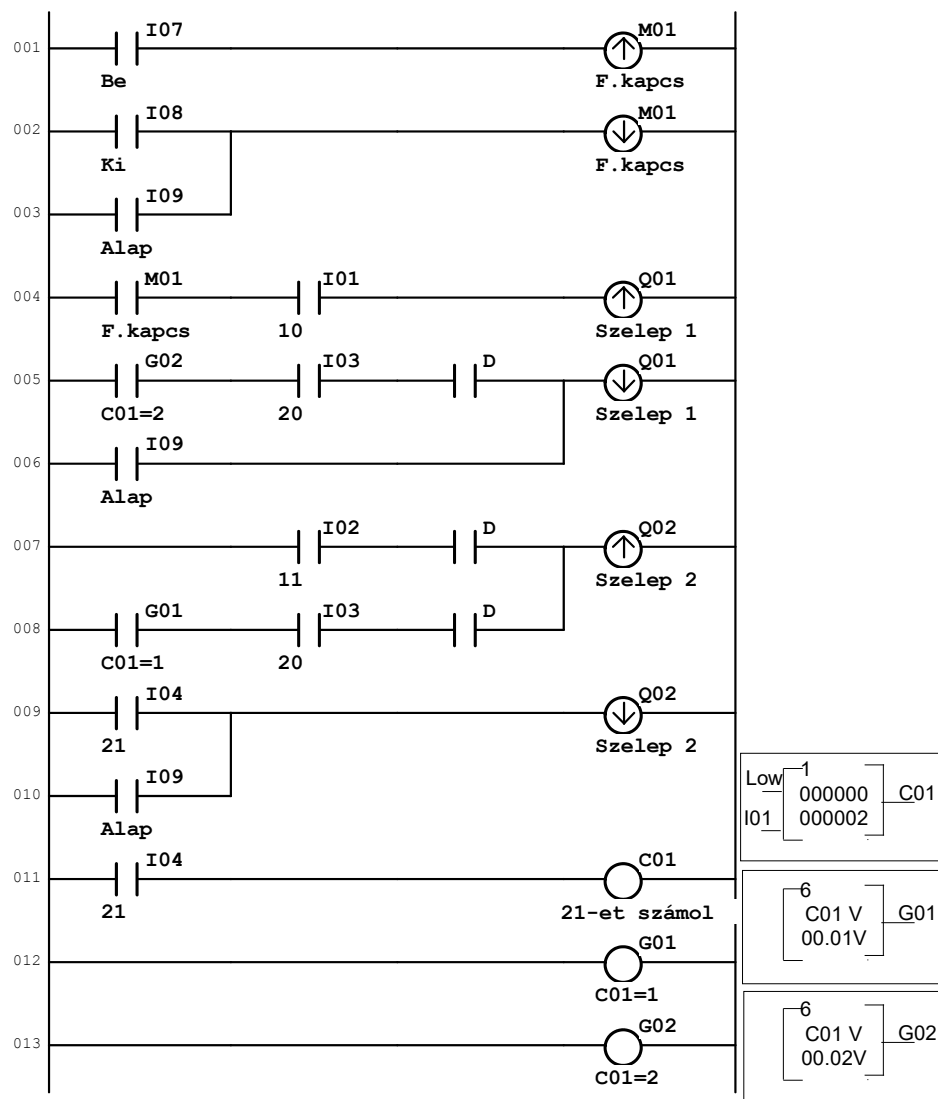


Két dugattyú alternáló mozgása a következő út-lépés diagram alapján. A mozgássorban a cikluson belül ismétlődés található, és ezt belső számláló alkalmazásával tudjuk a programban figyelembe venni.

Az impulzussorrend: C10 C11 C21 C20 C21 C20

Blokkoló jelek: C11 C20

Bemenetek összerendelési listája												
Bemeneti cím	I01	I02	I03	I04	I05	I06	I07	I08	I09	I10	I11	I12
Funkció	10	Diff 11	Diff 20	21			Be	Ki	Alap			
Kimenetek összerendelési listája:												
Kimeneti cím	Q01	Q02	Q03	Q04	Q05	Q06	Q07	Q08				
Funkció	Y1	Y2										

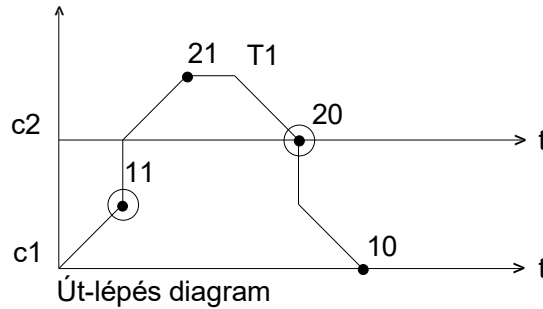


A belső számláló (C01) a 21 végálláskapcsoló lenyomásainak megfelelő I04-es bemeneti eseményeket számlálja (1. módban kettőig számol). A számláló nullázására bármely, a számlálandó ciklusrész után bekövetkező pillanatnyi impulzussal megoldható, jelen példában a 10-nek megfelelő I01-el.

A 20-as végállás-kapcsolónak megfelelő I03-as kétféle parancsot kell, hogy adjon, attól függően, hogy egyig, vagy kettőig számolt el a számláló. Ezt komparátorok alkalmazásával célszerű megoldani ((6. mód) G01: C01=1 és G02:C01=2). Ezért a megfelelő feltételeknél az I03-mal sorba kell kötni G01-et illetve G02-t. A blokkoló jelek feloldására differenciálást alkalmazunk.

2.1.31 feladat

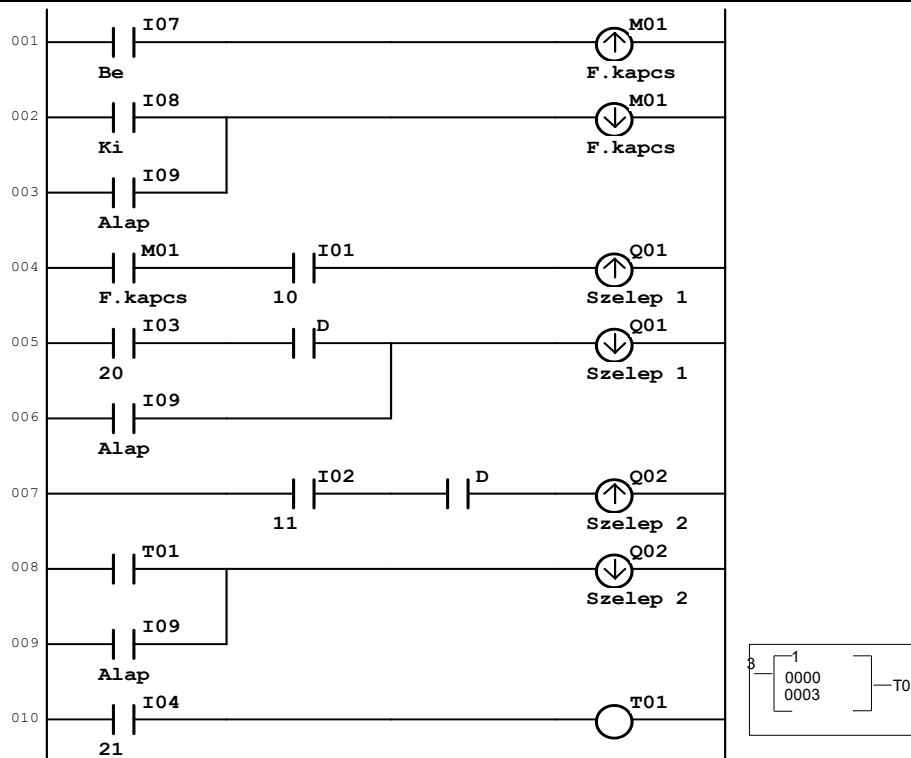
Két dugattyú alternáló mozgása a következő út-lépés diagram alapján. A cikluson belül T01= 3 s várakozás, időzítés is található.



Az impulzussorrend: C10 C11 C21 T1 C20

Blokkoló jelek: C11 C20

Bemenetek összerendelési listája												
Bemeneti cím	I01	I02	I03	I04	I05	I06	I07	I08	I09	I10	I11	I12
Funkció	10	Diff 11	Diff 20	21			Be	Ki	Alap			
Kimenetek összerendelési listája:												
Kimeneti cím	Q01	Q02	Q03	Q04	Q05	Q06	Q07	Q08				
Funkció	Y1	Y2										

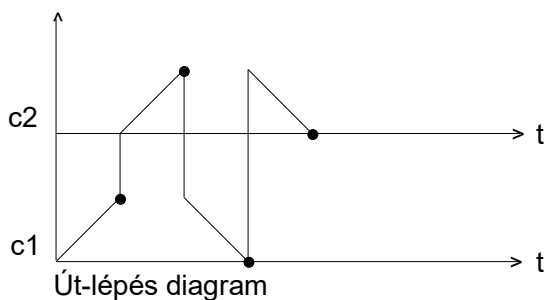


Az előző példákhoz képest az eltérés annyi, hogy nem az I04-es jel húzza vissza a Y2 dugattyút, hanem az csak egy időzítőt indít el. Ha már a beprogramozott idő eltelt, csak akkor indul vissza a Y2-es dugattyú. Tervezéskor figyelni kell a következőkre:

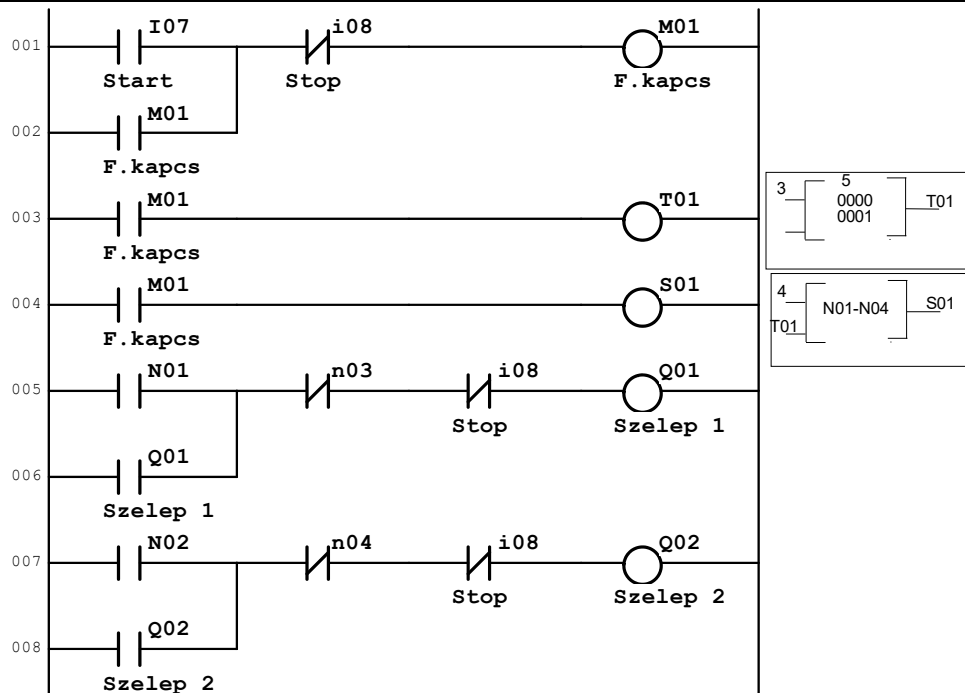
- Differenciálással előállított túimpulzus nem tudja a T01-et elindítani, mert a T01 bemenetének folyamatos jelre van szüksége az időzéshez.
- Ha a T01-et indító jel blokkoló, annak a blokkolóságát nem kell megszüntetni, hanem magát a T01 végeredményét kell majd differenciálással túimpulzussá alakítani.

2.1.32 feladat

Két dugattyú alternáló mozgása a következő út-lépés diagram alapján. A feladatot oldjuk meg idővezérelt léptetéssel (ilyenkor nincsenek végállás-érzékelők, beállított idők után lépnek a dugattyúk)! Használjuk ki a SHIFT utasítás lehetőségeit!

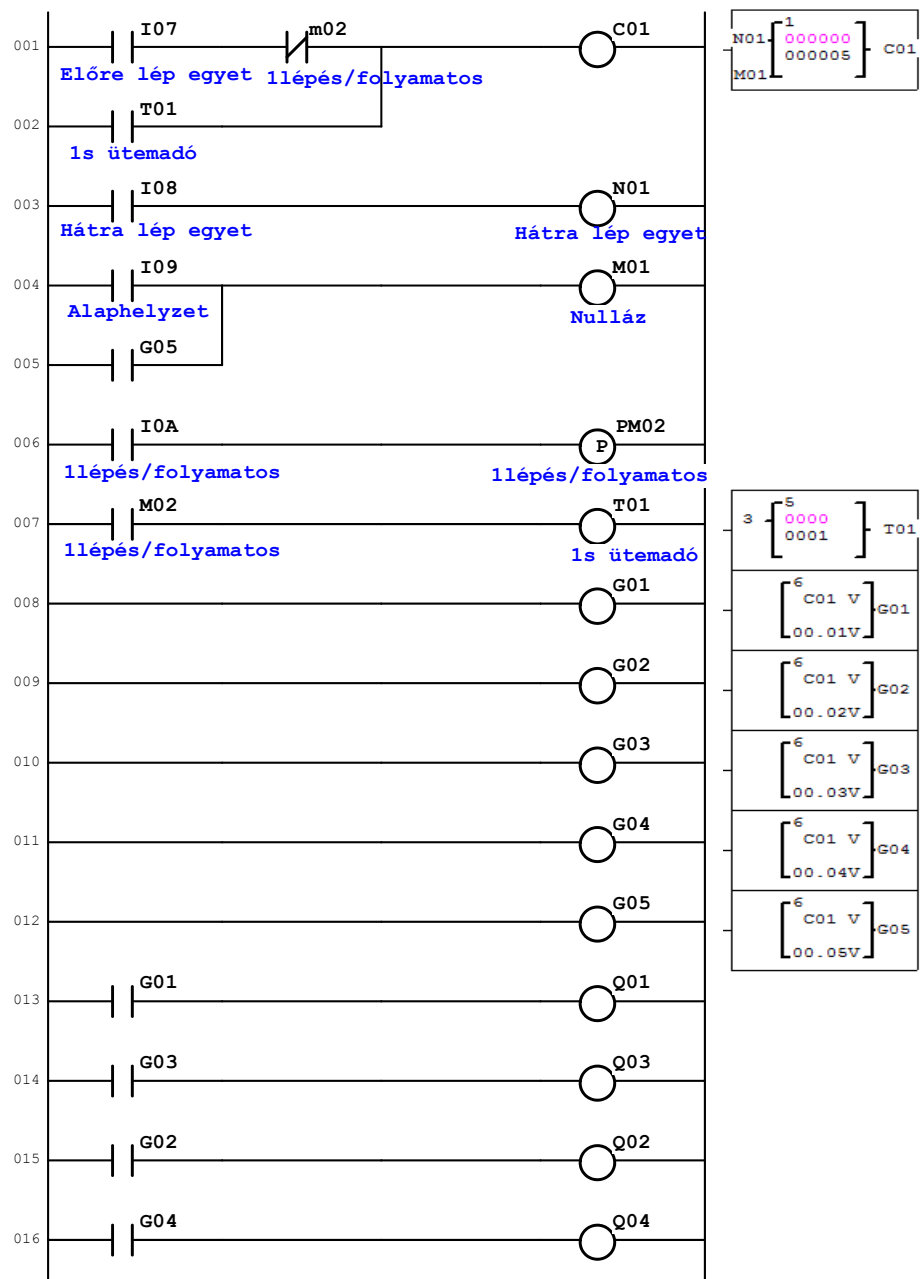


Bemenetek összerendelési listája												
Bemeneti cím	I01	I02	I03	I04	I05	I06	I07	I08	I09	I10	I11	I12
Funkció							Start	Stop				
Kimenetek összerendelési listája:												
Kimeneti cím	Q01	Q02	Q03	Q04	Q05	Q06	Q07	Q08				
Funkció	Y1	Y2										



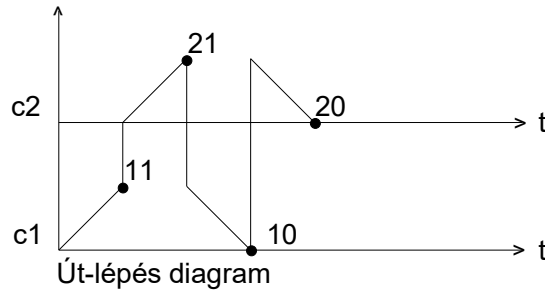
T01 5. módban van, értéke 1s. S01 T01 üteme alapján léptet N01-tő N04-ig.

Oldjuk meg a fenti mozgássort úgy, hogy az I07-es nyomógombot lenyomva mindig egy lépést tegyen előre, az I08-as nyomógombot lenyomva visszafelé lépjen, I09-es nyomógombra alaphelyzetbe álljon! **Ehhez a kapcsoláshoz használjunk két db bistabil 5/2-es szelepet!** Az utolsó lépés után lenyomva az I07-est, kerüljön ismét alaphelyzetbe! I10 nyomógomb hatására folyamatosan 1s időközönként lépjen előre, ekkor az I07 nem aktív, I08 és I09 viszont igen!



2.1.33 feladat

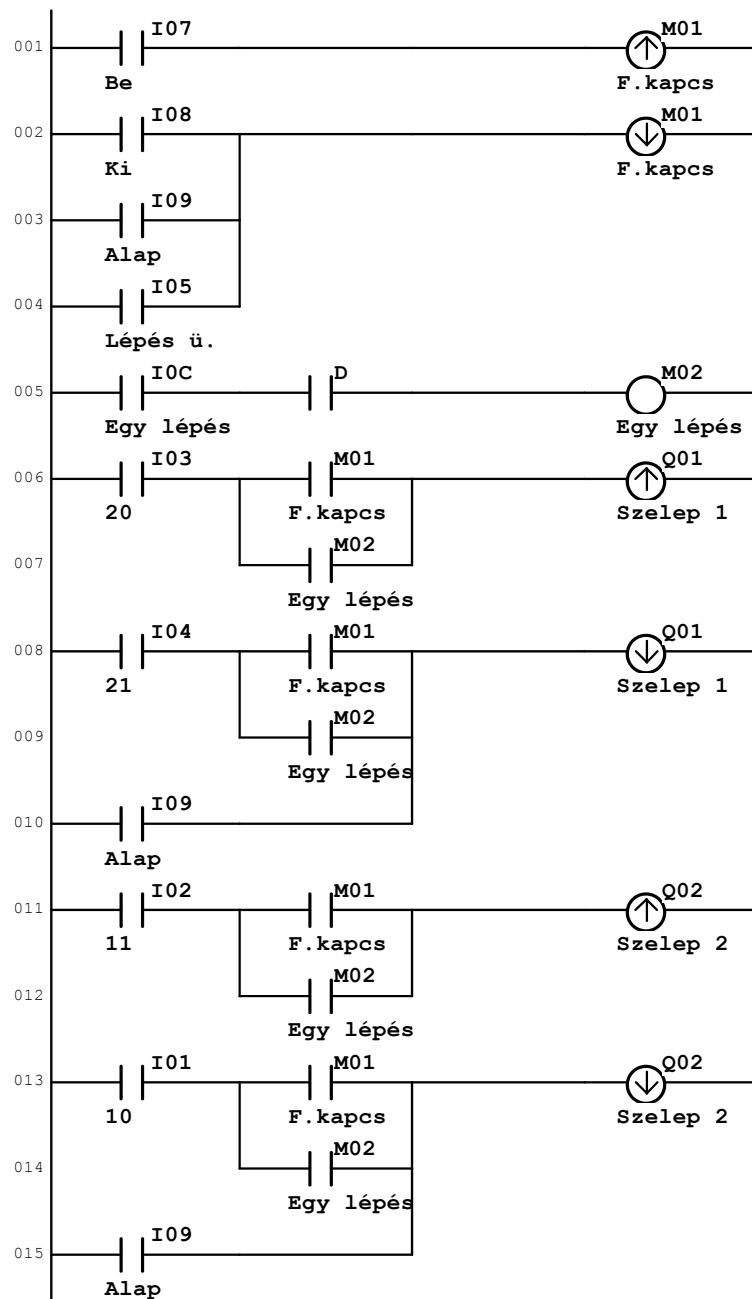
Két dugattyú alternáló mozgása a következő út-lépés diagram alapján. I05-ös kapcsoló átkapcsolásával le lesz tiltva az automatikus üzemmód, az I12-es gomb mindenkor lenyomásakor egy lépést tesz meg a program a ciklusból (kézi üzemmód).



Az impulzussorrend: C20 C11 C21 C10

Blokkoló jelek: Nincsenek

Bemenetek összerendelési listája												
Bemeneti cím	I01	I02	I03	I04	I05	I06	I07	I08	I09	I10	I11	I12
Funkció	10	11	20	21	Lépés /foly.		Be	Ki	Alap			Lépés indul
Megjegyzés												
Kimenetek összerendelési listája:												
Kimeneti cím	Q01	Q02	Q03	Q04	Q05	Q06	Q07	Q08				
Funkció	Y1	Y2										

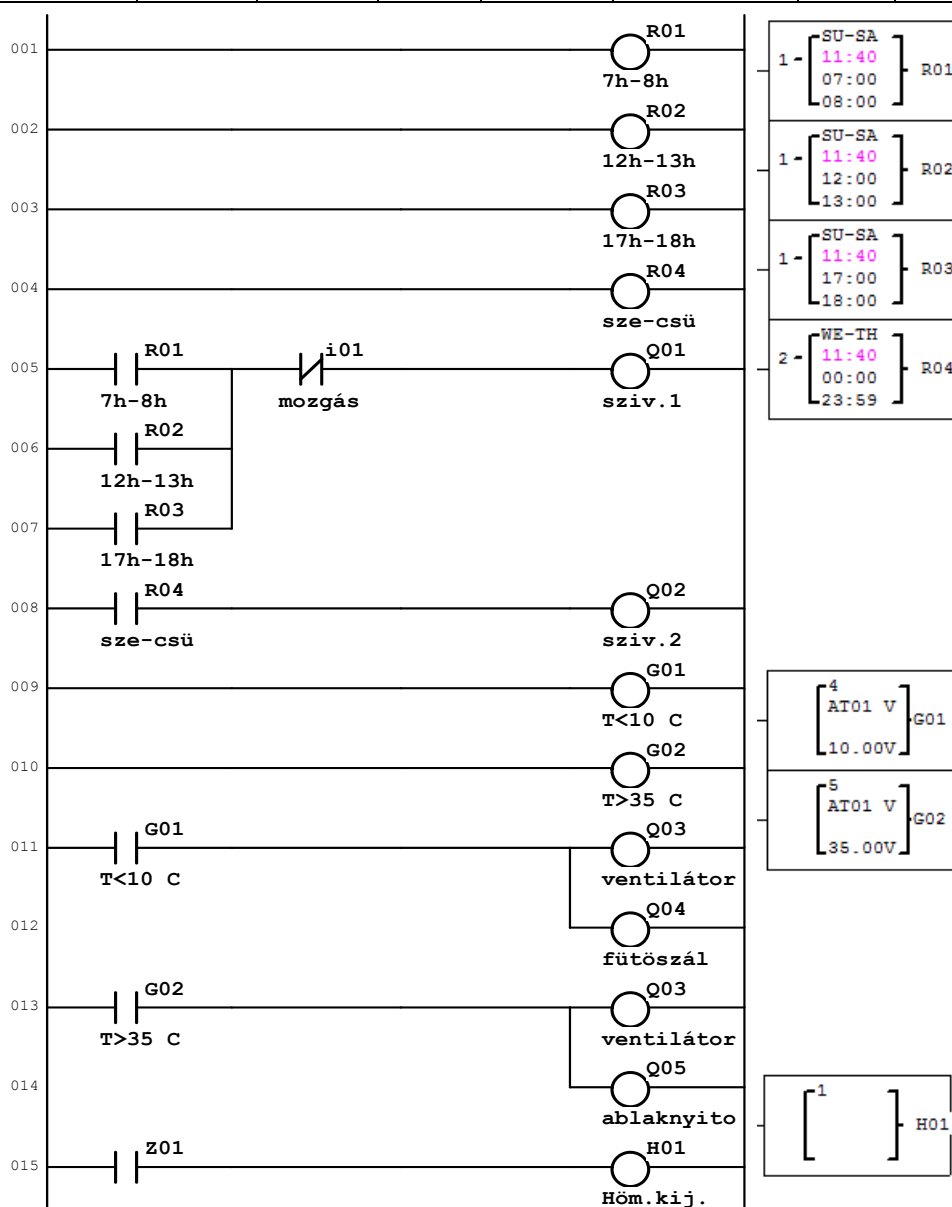


Az I05-ös üzemmódváltó kapcsoló - mivel a RESET domináns a SET ellenében - , nem engedi a M01-en keresztül a bekapcsolási feltétel teljesülést. Mivel azonban az I12-es (M02) párhuzamosan van kötve az M01-el, lenyomására egy lépés feltétele teljesül. Mivel az M02 felfutóél-vezérelt, ezért ha folyamatosan nyomjuk a gombot, akkor is csak egyet lép a cikluson belül.

2.1.34 feladat

Üvegház szabályozása. Egy üvegházban a növényeket egész évben, a hét minden napján naponta háromszor kell locsolni, az egyik szivattyúval 7.00-8.00, 12.00-13.00, 17.00-18.00 időpontokban. A másik szivattyú minden héten szerdától csütörtökig egész nap a gyökerekre csöpögtet vizet. Ha az üvegházban 35 °C-nál melegebb van, kapcsoljon be egy ventilátor és nyíljon ki egy szellőzőablak, ha 10 °C-nál hidegebb van, kapcsoljon be egy fűtőtest és a ventilátor. Ha az üvegházban tartózkodnak, ne induljon el az első szivattyú! Az LCD kijelző írja ki a mért hőmérsékletet, ha lenyomjuk a Z01 kezelőgombot!

Bemenetek összerendelési listája												
Bemenet	I01	I02	I03	I04	I05	I06	I07	I08	A01	A02	A03	A04
Funkció	Mozgásérzékelő	Ablak nyitva érzékelője							Hőmérséklet érzékelő			
Kimenetek összerendelési listája:												
Kimenet	Q01	Q02	Q03	Q04	Q05	Q06	Q07	Q08				
Funkció	1. szivattyú	2. sziv.	ventilátor	fűtőszál	ablaknyitó							

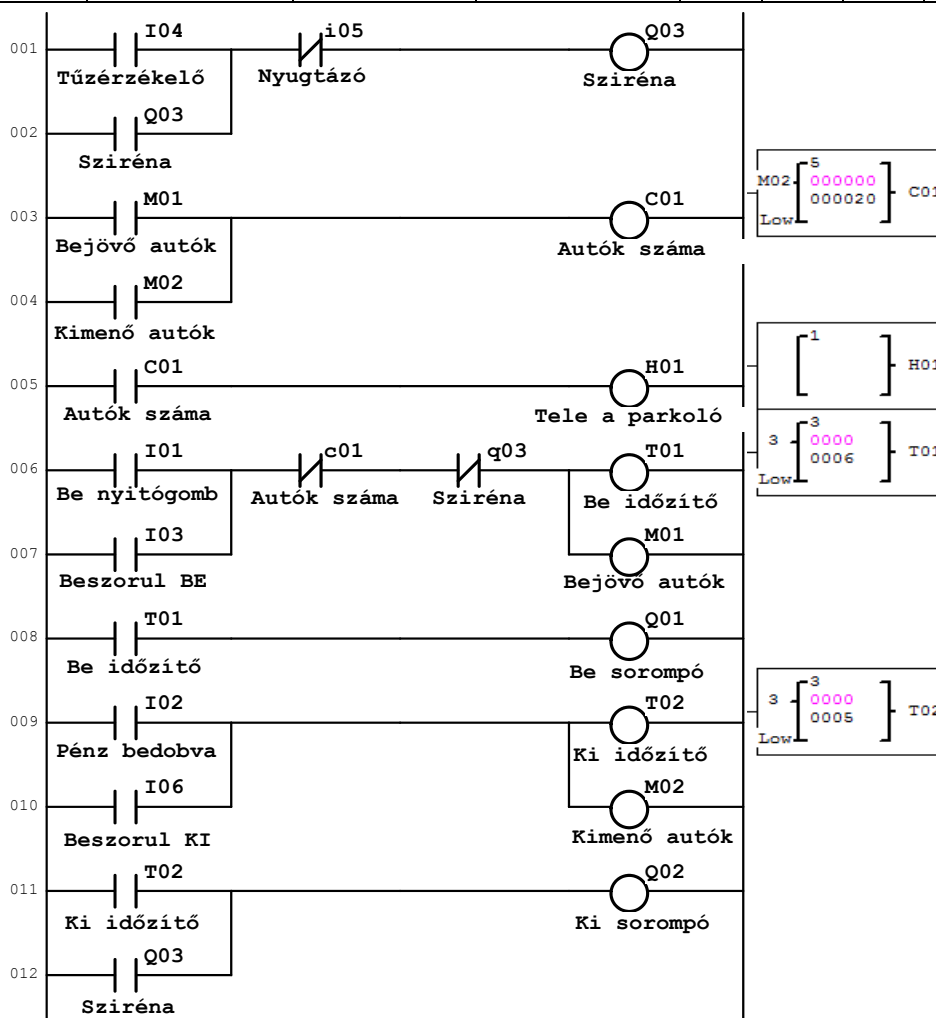


2.1.35 feladat

Parkolóház. Egy kis parkoló kapacitása 20 személyautó, a bejáratnál és a kijáratnál is sorompó van. A beléptető sorompó akkor nyílik, ha van üres hely a parkolóban, és megnyomjuk a beléptető gombot. Ezt követően T01=6 s idő után lecsukódik. A kiléptető sorompó akkor nyílik, ha bedobjuk a pénzt. Ekkor T= 5 s ideig nyitva van, utána lecsukódik. A személygépkocsi beszorulása esetén mindkét sorompó automatikusan kinyílik.

Tűz esetén a kiléptető sorompó automatikusan nyílik, a beléptető sorompó zárva marad, egy sziréna megszólal. Ezt az állapotot egy nyugtázó gombbal tudunk megszüntetni. Az LCD kijelző jelezze ki, ha tele van a parkoló!

Bemenetek összerendelési listája												
Bemeneti cím	I01	I02	I03	I04	I05	I06	I07	I08	I09	I10	I11	I12
Funkció	Autó érkezik	Kifizetve	Autó beszorult BE	Tűz érzékelő	Tűz nyugtázó	Autó beszorult KI						
Kimenetek összerendelési listája:												
Kimenet	Q01	Q02	Q03	Q04	Q05	Q06	Q07	Q08	Q09	Q10	Q11	Q12
Funkció	Beléptető nyit	Kiléptető nyit	Sziréna									



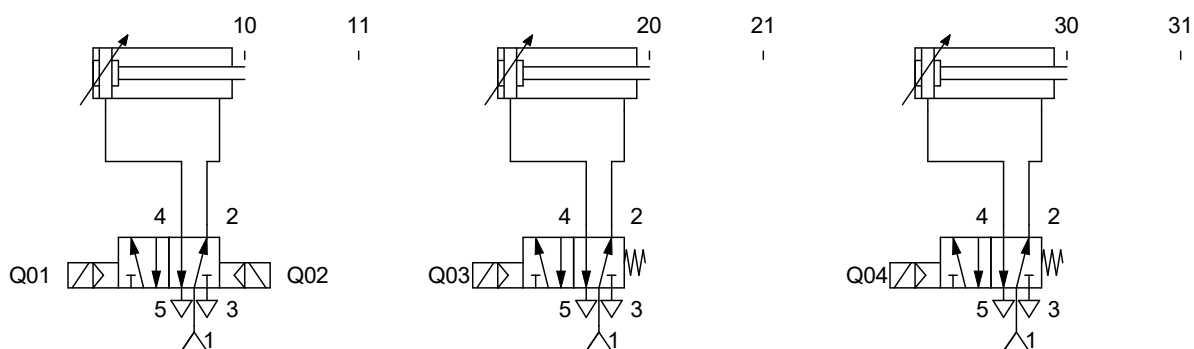
Az autók számolásánál a C01 5. módban van, M01-et előrefelé, M02-t visszafelé számolja.

Az időzítők (T01 és T02) késve kikapcsoló típusúak (3. mód).

2.2 Feladatok haladó programozóknak

Ezek a kapcsolások három munkahenger mozgásaira épülnek. A szelepek és a munkahengerek elrendezése mindig egyforma, ezért nem fogom minden feladatnál külön lerajzolni őket. Vegyes szelepelrendezést választottam, az első bistabil, a második és a harmadik monostabil, így mindkettő jellegzetességeire fel tudom hívni a figyelmet. A bemenetek és a kimenetek összerendelési listája is azonos, ha valahol eltérés van, azt külön jelzem.

A vezérelt szelepek elrendezése:

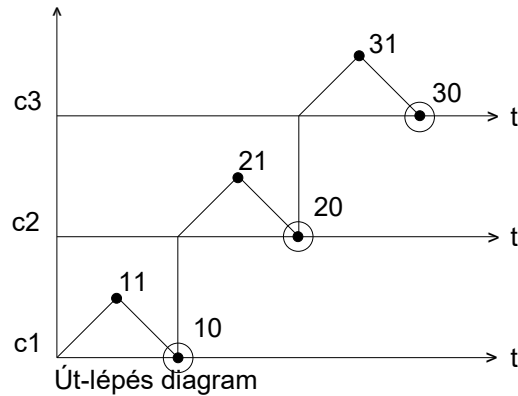


Az összerendelési lista:

Bemenetek összerendelési listája												
Bemenet	I01	I02	I03	I04	I05	I06	I07	I08	I09	I10	I11	I12
Funkció	10	11	20	21	30	31	Start gomb	Vészstop gomb	Üzem módváltó gomb egy ciklus/folyamatos			

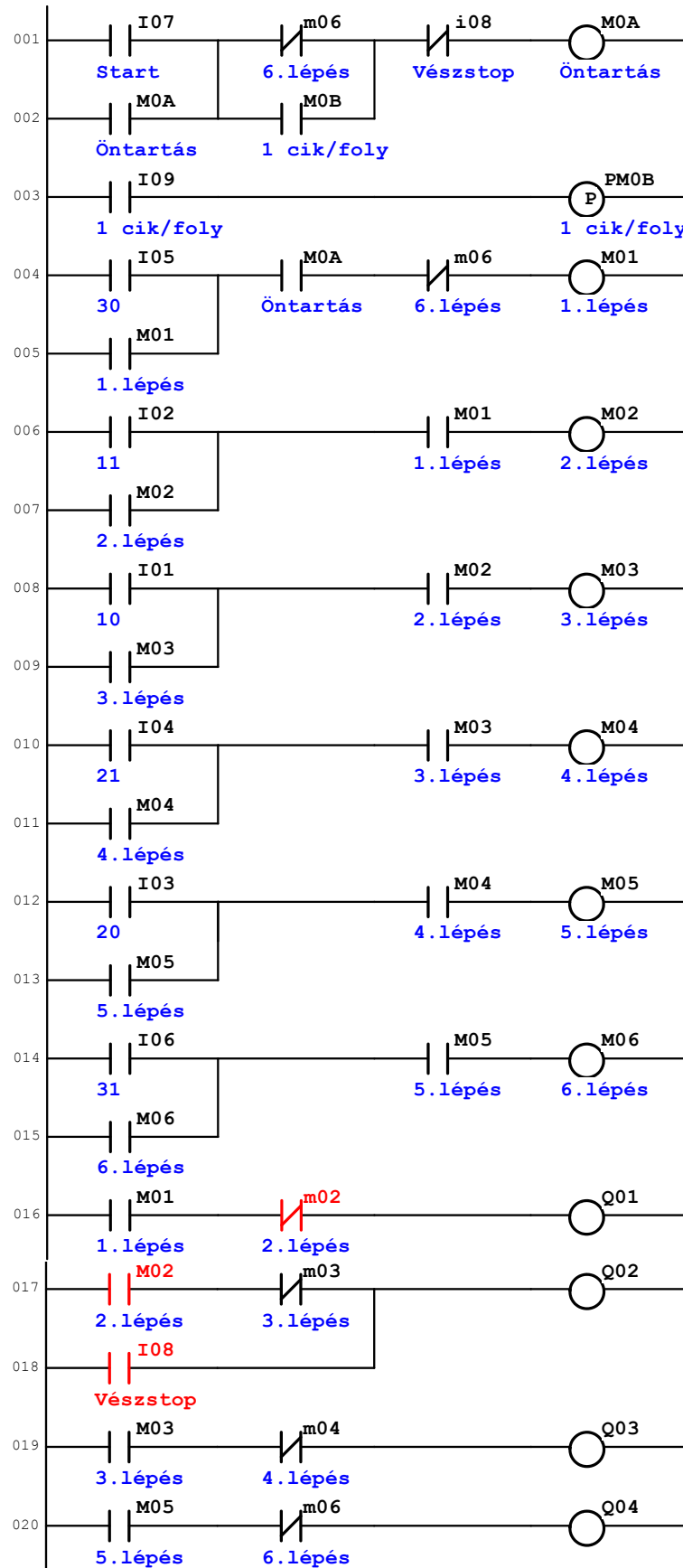
2.2.1 feladat

Három dugattyú ciklusos mozgása a következő út-lépés diagram alapján.



Az impulzussorrend: 30 11 10 21 20 31

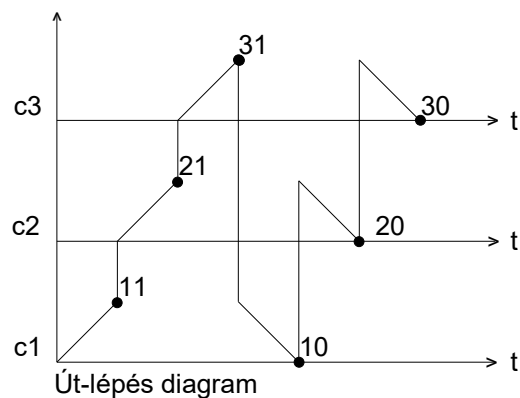
A blokkoló jeleket most nem vizsgáljuk, mert a feladat monostabil léptetőláncal lesz megoldva. Azért kell a monostabil léptetőláncot választani, mert vegyes típusú szelepeket alkalmazunk. A mozgás hat független lépésből áll, ezért hatelemű léptetőláncot építünk. Figyelni kell arra, hogy a bistabil szelepeknél külön kell megoldani a keresztreteszélést és az alaphelyzetbe állítást.



2.2.2 feladat

Három dugattyú ciklusos mozgása a következő út-lépés diagram alapján.

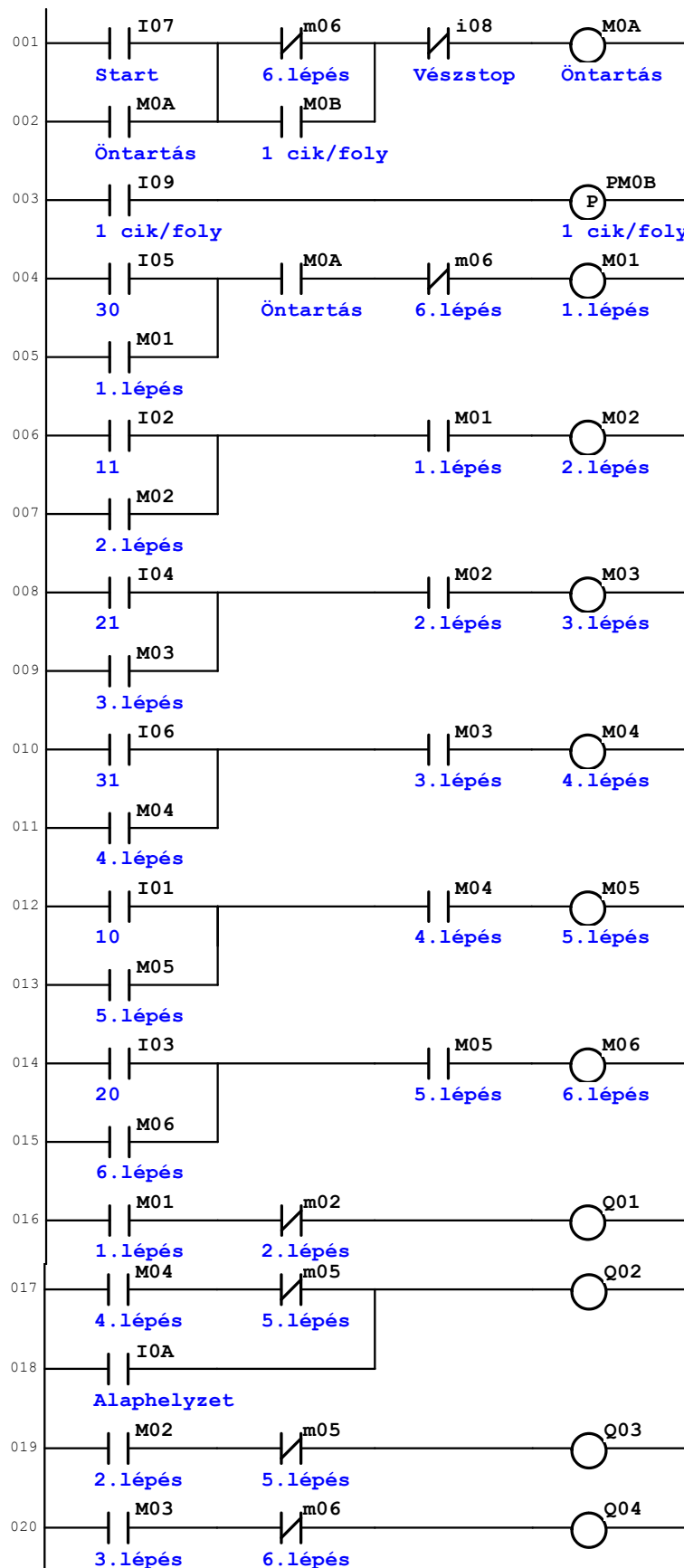
Bemenetek összerendelési listája														
Bemenet	I01	I02	I03	I04	I05	I06	I07	I08	I09			I10	I11	I12
Funkció	10	11	20	21	30	31	Start gomb	Vészstop gomb	Üzemmódváltó gomb egy ciklus/folyamatos					



Az impulzussorrend: 30 11 21 31 10 20

A feladatot most is monostabil léptetőláncsal oldjuk meg, a szelepbekötések és az összerendelési lista azonos, csak a mozgássorrend változik. Elegendő a létradiagram impulzus sorrendjén és a szelepvezérlő részén módosítani.

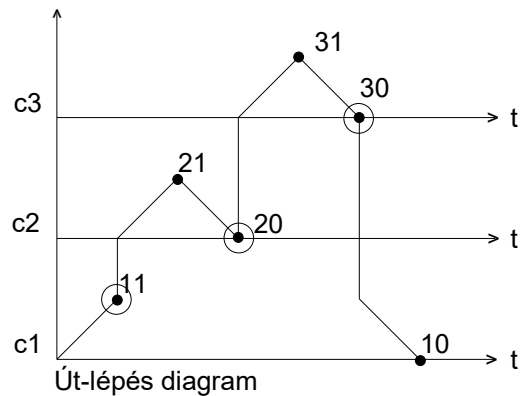
Ennél a mozgásnál egyébként nincsenek blokkoló jelek.



2.2.3 feladat

Három dugattyú ciklusos mozgása a következő út-lépés diagram alapján.

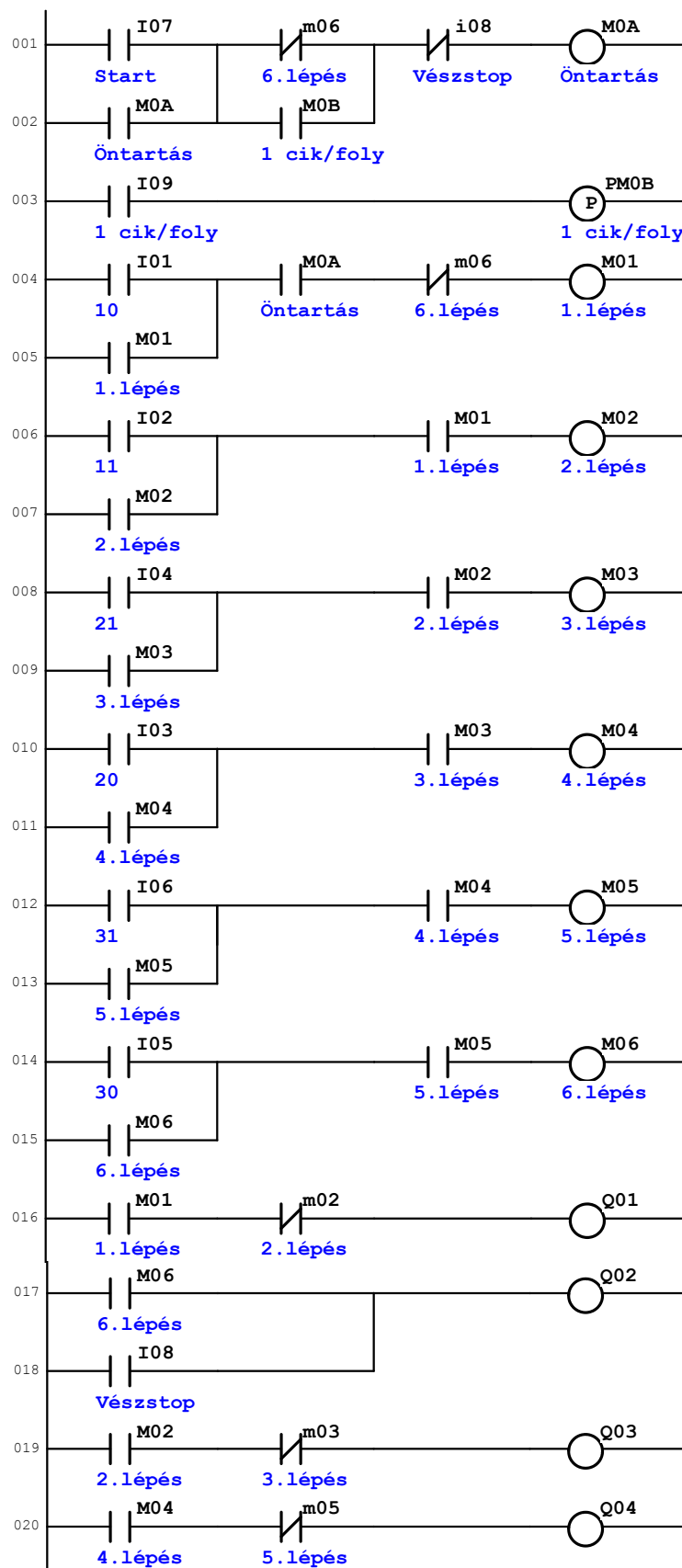
Bemenetek összerendelési listája														
Bemenet	I01	I02	I03	I04	I05	I06	I07	I08	I09			I10	I11	I12
Funkció	10	11	20	21	30	31	Start gomb	Vészstop gomb	Üzem módváltó gomb egy ciklus/folyamatos					



Az impulzussorrend: 10 11 21 20 31 30

A blokkoló jelek: 11 20 30

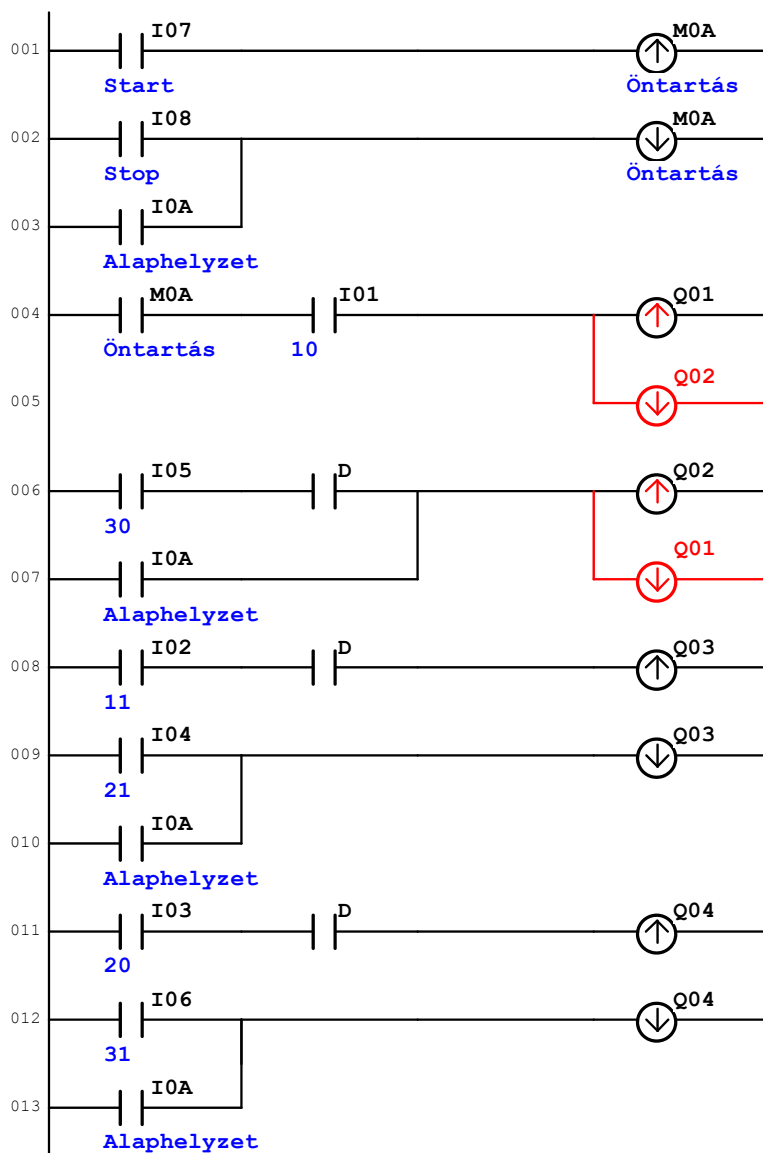
A feladatot most is monostabil léptetőláncsal oldjuk meg, a szelepbekötések és az összerendelési lista azonos, csak a mozgássorrend változik. Elegendő a létradiagram impulzus sorrendjén és a szelepvezérlő részén módosítani.



Oldjuk meg a feladatot elemi úton is! A Start gombra elindul a folyamatos üzemmód, a Stop gombra az éppen folyó ciklust befejezi, majd leáll. Ciklus közben az Alaphelyzet (Vész-stop) gombra azonnal minden dugattyú alaphelyzetbe áll.

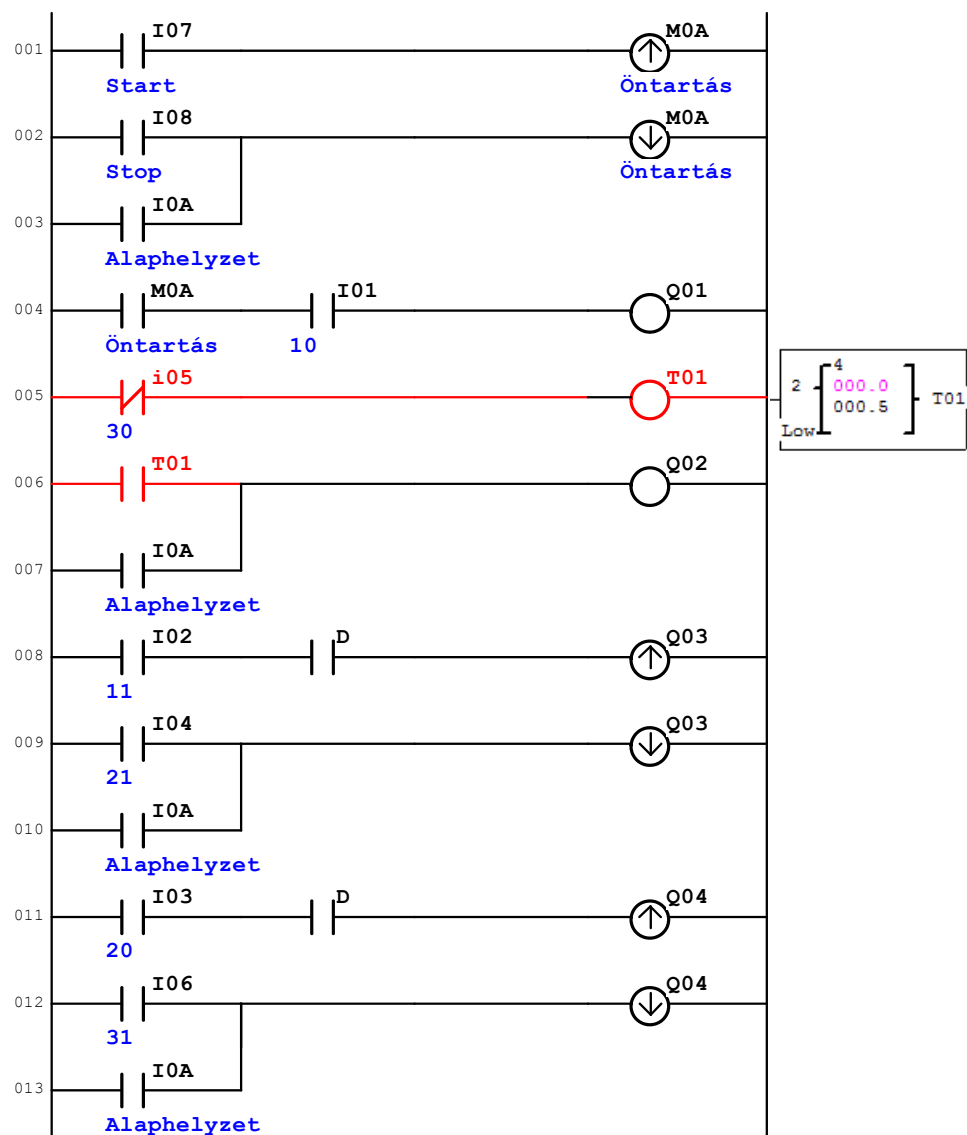
Bemenetek összerendelési listája												
Bemenet	I01	I02	I03	I04	I05	I06	I07	I08	I09	I10	I11	I12
Funkció	10	11	20	21	30	31	Start gomb	Stop gomb		Alaphelyzet gomb		

Monostabil szelepeknél a blokkoló jeleket differenciáló elemekkel küszöböljük ki. Alaphelyzetbe hozó nyomógombot minden munkahengert visszatérítő létrafokba terveznünk kell. Mivel a differenciált érzékelő jele olyan rövid ideig tart (1 ciklusidő), hogy képtelen bekapcsolni a bistabil szelepek tekercseléseit, ezért Q01 és Q02 esetén keresztreteszelt RS ki-meneteket kell alkalmaznunk.



A bistabil szelephez keresztreteszelés helyett, illetve egyszerű differenciálás helyett egy késve kikapcsoló időzítőt is tervezhetünk (T01, 4.mód), amelynek időzítése legalább akkora kell, hogy legyen, ami már képes a bistabil szelep tekercsét behúzni (pl.: 0,5 s).

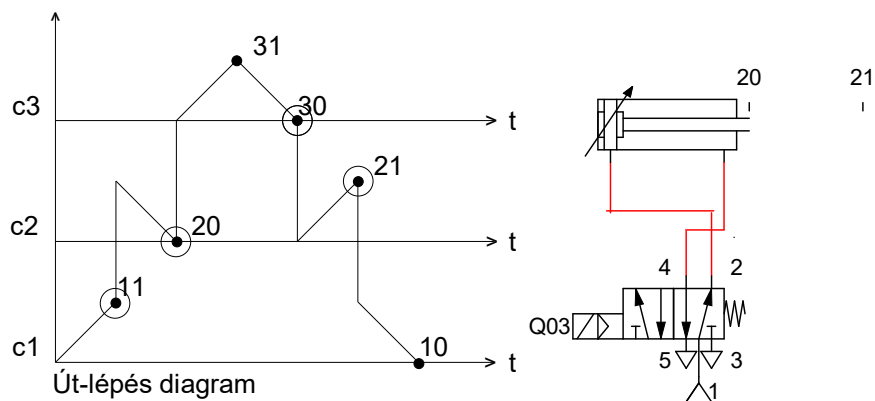
Az előző megoldásnál annyival jobb ez, hogy a bistabil szelep tekercseit nem terheli folyamatosan áram.



2.2.4 feladat

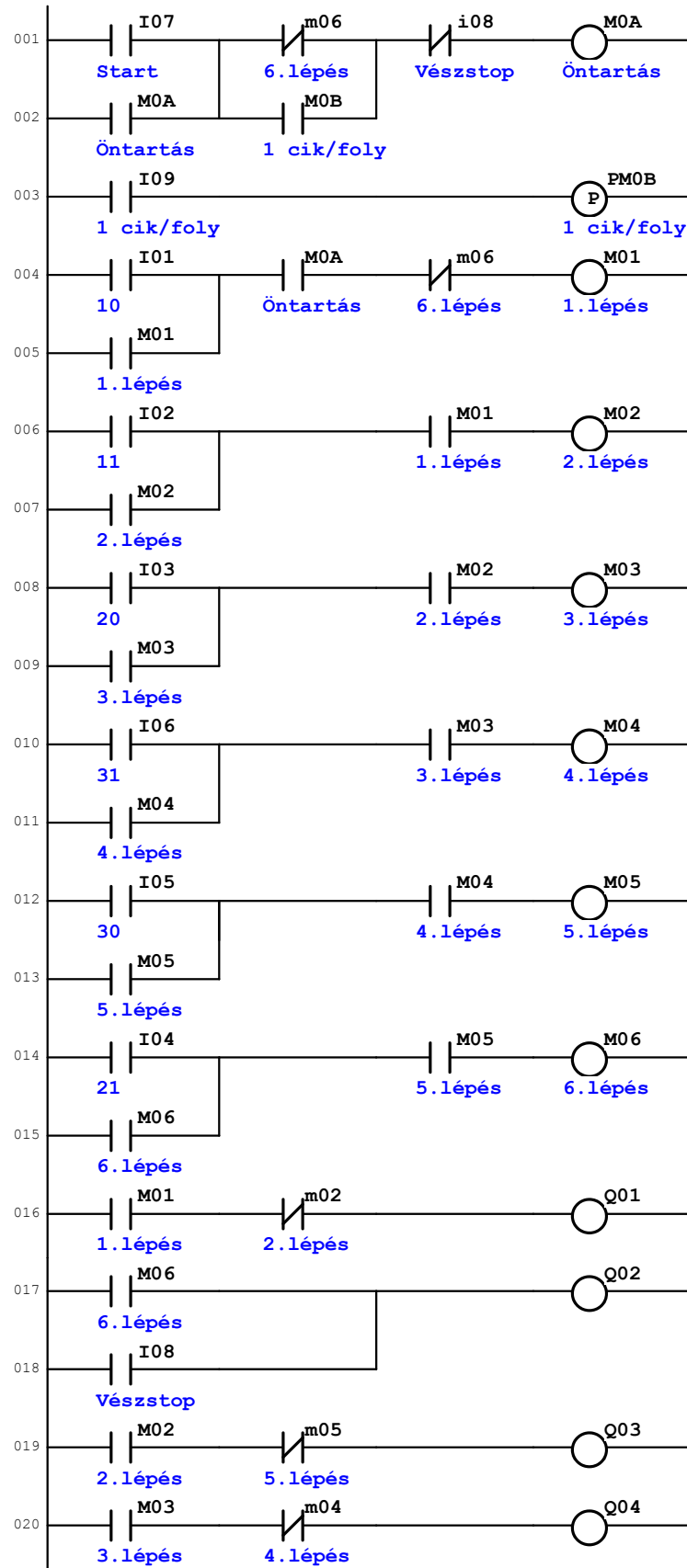
Három dugattyú ciklusos mozgása a következő út-lépés diagram alapján.

		Bemenetek összerendelési listája												
Bemenet	I01	I02	I03	I04	I05	I06	I07	I08	I09			I10	I11	I12
Funkció	10	11	20	21	30	31	Start gomb	Vészstop gomb	Üzem módváltó gomb egy ciklus/folyamatos					



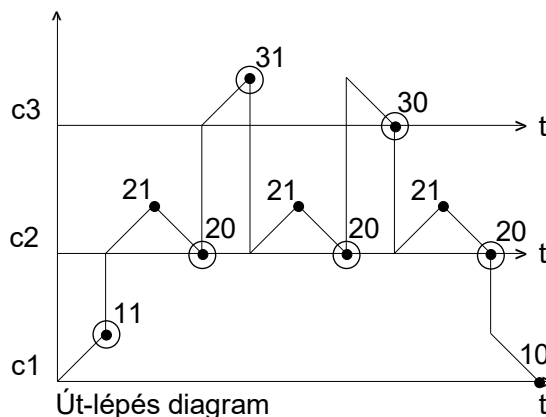
Az impulzussorrend: 10 11 20 31 30 21

A blokkoló jeleket nem vizsgálom, mert a feladat hatlépéses léptetőláncal lesz megoldva. Újdonság a feladatban, hogy a második munkahenger alaphelyzetben kinn van, erre külön figyelniünk kell. Ezt a problémát a legegyszerűbben úgy tudjuk megoldani, hogy a szelep és a munkahenger közötti két levegőcsövet felcseréljük.



2.2.5 feladat

Három dugattyú ciklusos mozgása a következő út-lépés diagram alapján.



Bemenetek összerendelési listája												
Bemenet	I01	I02	I03	I04	I05	I06	I07	I08	I09	I10	I11	I12
Funkció	10	11	20	21	30	31	Start gomb	Vészstop-Alaphelyzet gomb				

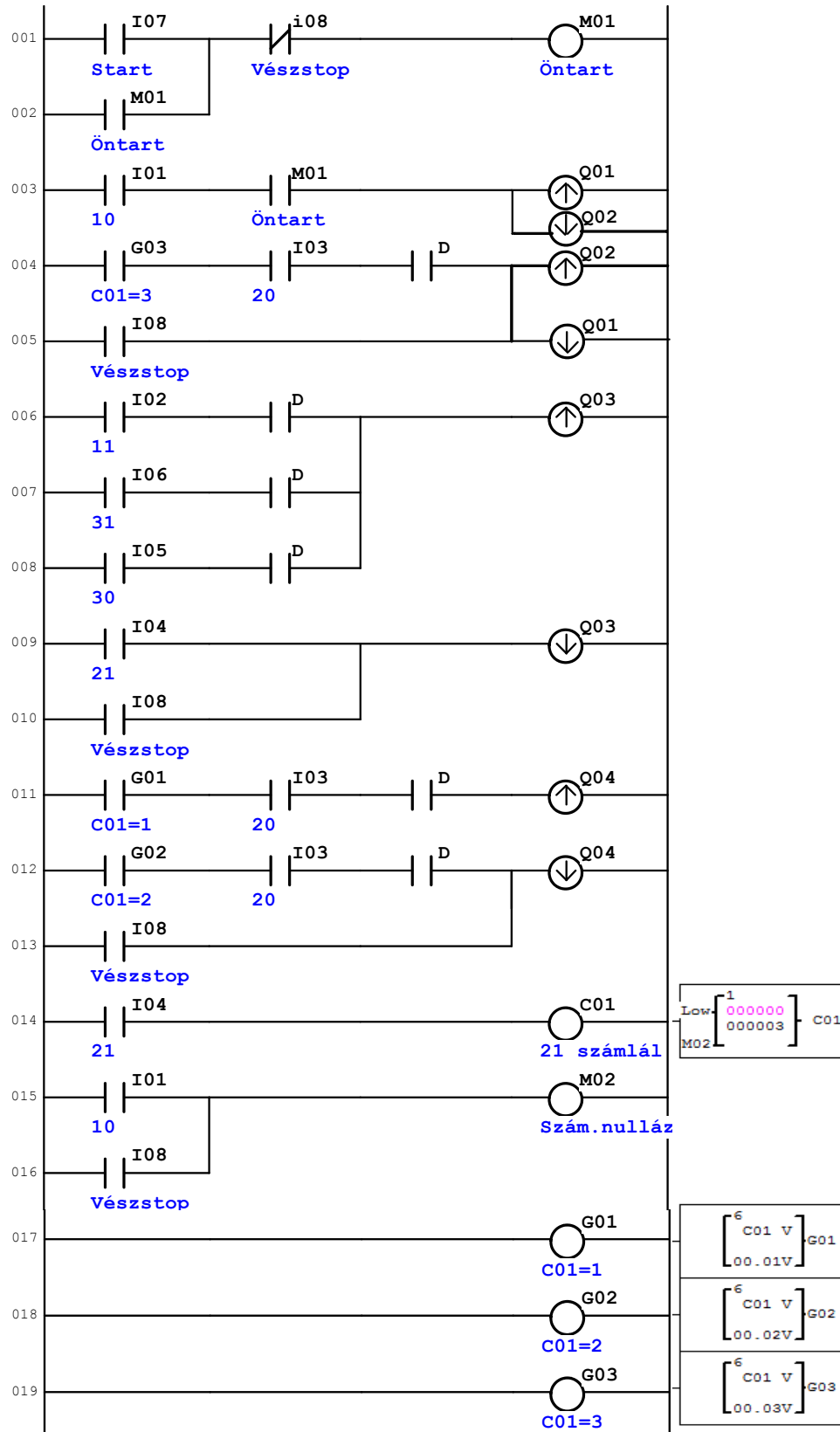
Az impulzussorrend: 10 11 21 20 31 21 20 30 21 20

A blokkoló jelek: 11 20 31 30

A feladatot elemi úton oldjuk meg. A blokkoló jeleket differenciálással szüntetjük meg. A 21 sorszámú végállaskapcsoló jeleit egy 1.módban lévő számlálóval számoltatjuk, a számlálót a 10-es végállaskapcsoló nullázza. A számláló értékeit (1, 2, 3), amit a cikluson belül felvehet, egy-egy komparátor (1, 2, 3) referenciaértékeivel hasonlítjuk össze, és amikor ezek az értékek megegyeznek, háromféle mozgást indítanak el.

A feladatban ügyelni kell „Vészstop” esetén a számláló nullázására, az R-S tárolók nullázására, a bistabil szelep alaphelyzetbe hozására.

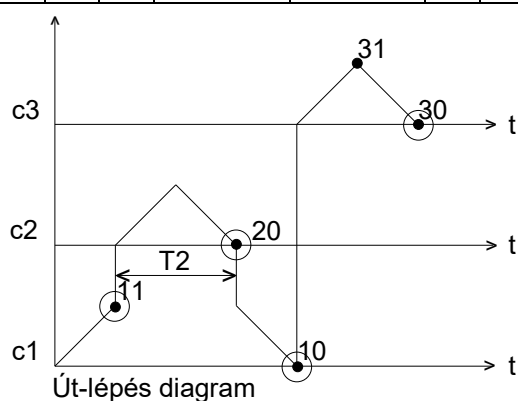
Természetesen a feladat megoldható lett volna hatelemű léptetőlánc segítségével is.



2.2.6 feladat

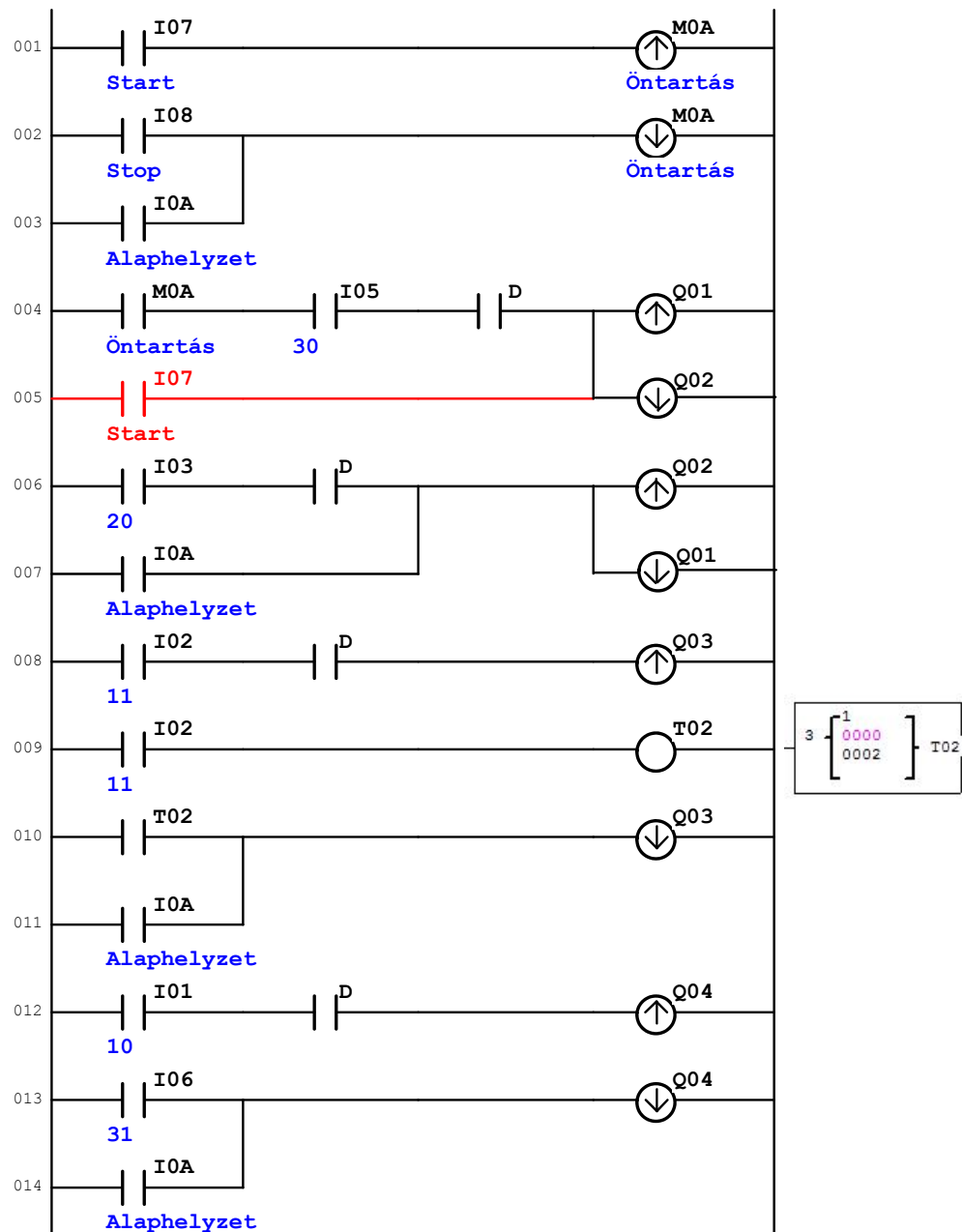
Három dugattyú ciklusos mozgása a következő út-lépés diagram alapján. Az egyik monostabil szelepnél (Q03) csak egy végállás-érzékelők áll rendelkezésre, így a hiányzó végállás-érzékelő (21) helyett időzítőt (T2) használunk.

Bemenetek összerendelési listája												
Bemenet	I01	I02	I03	I04	I05	I06	I07	I08	I09	I10	I11	I12
Funkció	10	11	20	21	30	31	Start gomb	Stop gomb		Alaphelyzet gomb		



Az impulzussorrend: 30 11 T2 20 10 31

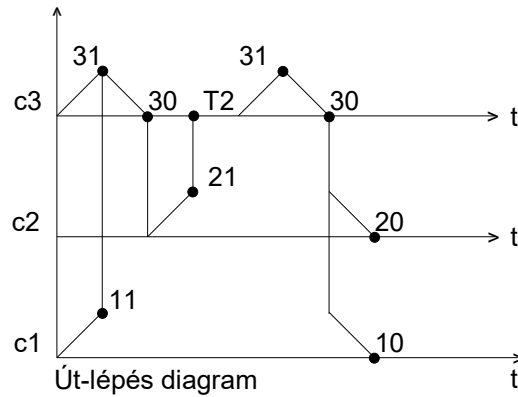
Blokkoló jelek: 30 11 20 10



A blokkoló jeleket differenciáló tagokkal oldjuk fel. Az idő megadásánál figyelniük kell arra, hogy legyen ideje a munkahengernek kimennie, majd bejönnie (T02 1. módban, értéke mondjuk 2 s). Arra is ügyelni kell, hogy a T02-t folyamatos jel vezérelje (I02). További probléma, hogy az indító jel blokkoló, amit ha differenciálunk, akkor önmagától nem fog a ciklus elindulni. Ilyenkor a kérdéses feltételekkel egy indító jelet (I07) kell párhuzamosan kötnünk.

2.2.7 feladat

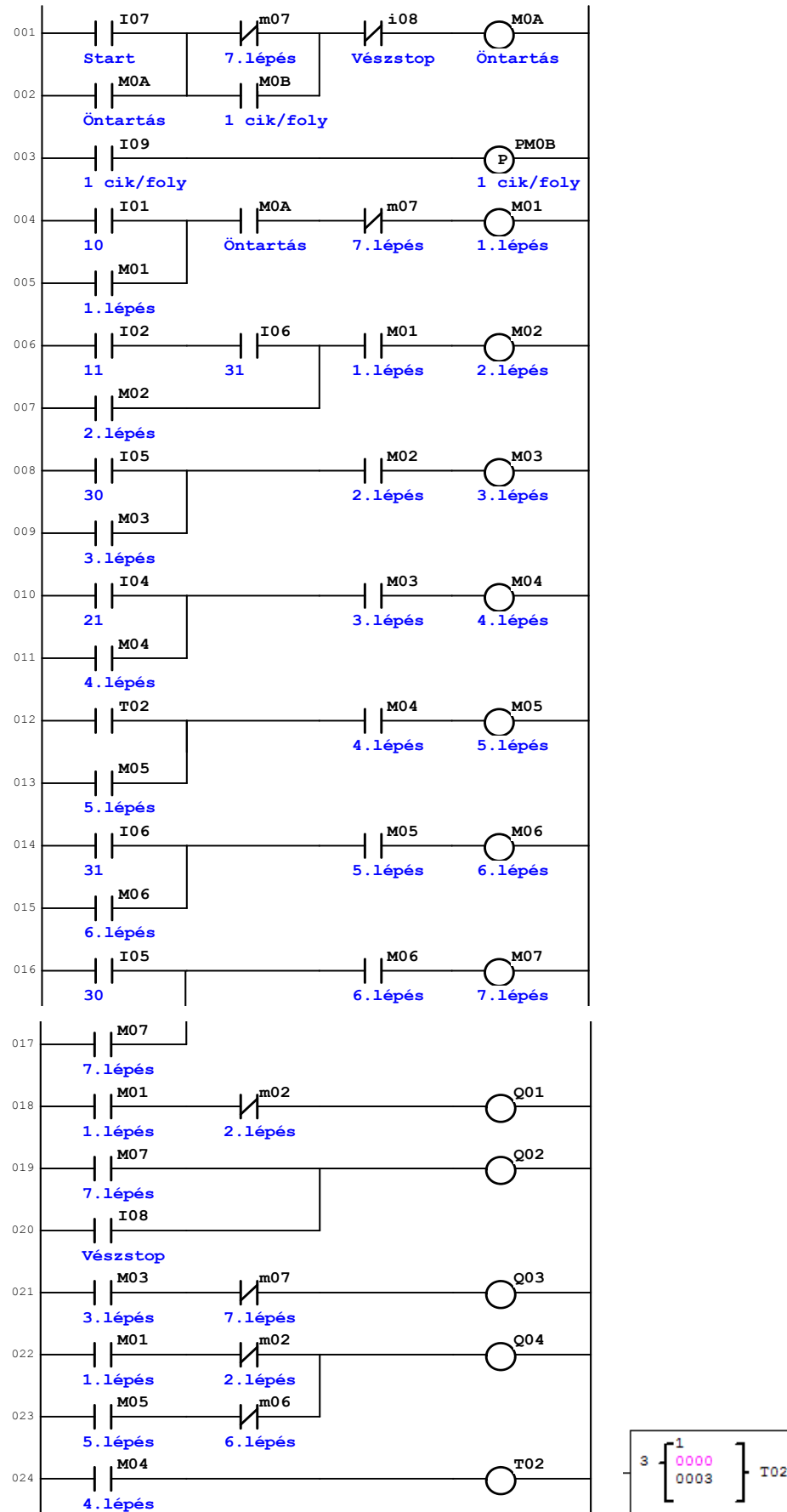
Három dugattyú ciklusos mozgása a következő út-lépés diagram alapján.



Az impulzussorrend: 10,20 együtt 11,31 együtt 30 21 T2 31 30

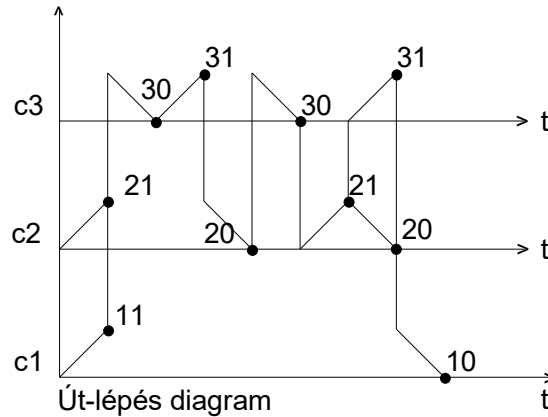
A blokkoló jeleket nem vizsgálom, mert a feladat léptetőláncsal lesz megoldva. A feladatban vegyes szeleptípusokat használunk. Az egyidejűségek miatt hét független eseményünk van.

Amely érzékelők egy időbe esnek, azokat a léptetőláncnál sorba kell kötnünk!



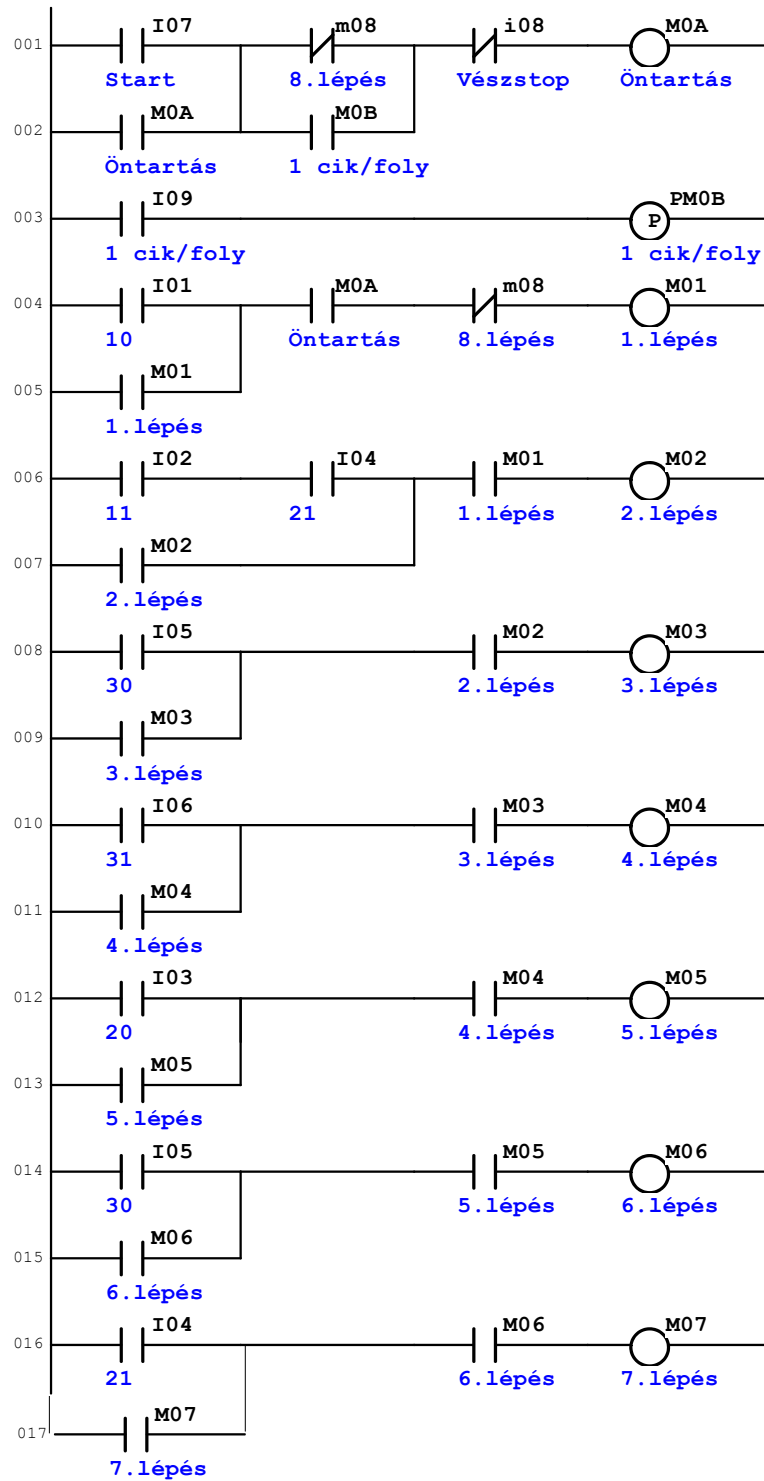
2.2.8 feladat

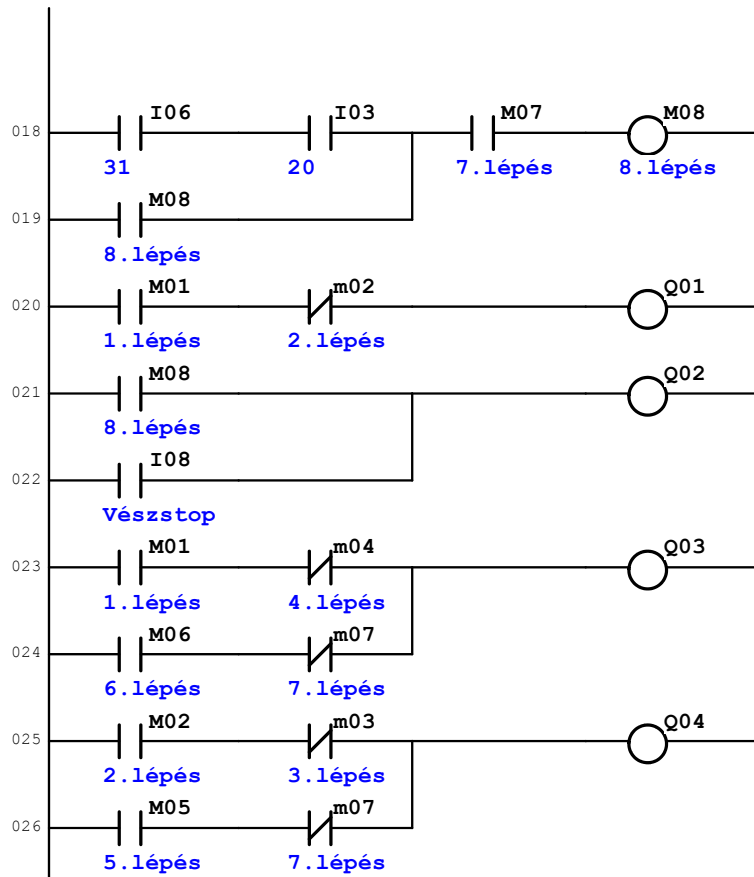
Három dugattyú ciklusos mozgása a következő út-lépés diagram alapján.



Impulzussorrend: 10, 11, 21 együtt 30 31 20 30 21 31, 20 együtt

A blokkoló jeleket és a belső ismétlődéseket nem vizsgálom, mert a feladat léptetőláncsal lesz megoldva. Nyolc független lépésből áll a mozgás a két egyidejűséget figyelembe véve. A C3 dugattyú inverzen mozog, ezt legkönnyebben úgy tudjuk elérni, ha a szelep és a dugattyú közötti két levegőcsövet felcseréljük.





3 FELHASZNÁLT IRODALOM

- Papp Géza, Nagylaki Csaba: A bütykös tengelytől a processzoros automatizálási platformokig, Villanyszerelők Lapja Internetes szócikk gyűjtemény
- ismart_manual_v3_0110.pdf
- Wikipedia internetes lexikon
- Borítóterv: Corel clipart