

MISKOLCI EGYETEM
GÉPÉSZMÉRNÖKI ÉS INFORMATIKAI KAR



Automatizálási és Infokommunikációs Intézet

Ipari automatizálás és kommunikáció

LIFT-MODELL IRÁNYÍTÁSTECHNIKAI REKONSTRUKCIÓJA, OKTATÁSBA INTEGRÁLÁSA

Szakdolgozat

DIÓ PÉTER

IPZ9T5

Miskolc, 2017

Tartalomjegyzék

1	Bevezetés.....	4
2	Célkitűzés	5
3	Lift-modell.....	6
3.1	Állapotfelmérés	7
3.2	Relék.....	8
3.3	Kódléc.....	10
3.4	Be- és kimenetek	12
4	Programozható logikai vezérlők.....	15
4.1	PLC alapok	15
4.1.1	PLC-k hardver komponensei.....	15
4.1.2	PLC programnyelvek	16
4.2	A PLC kiválasztása.....	17
4.3	S7-200 CPU	18
4.4	PLC illesztése a rendszerhez	23
5	Példafeladatok	27
5.1	Feladatismertetés	27
5.1.1	Bemenetek	27
5.1.2	Kimenetek	27
5.1.3	I/O tábla.....	28
5.1.4	Szintérzékelés.....	29
5.1.5	Hívógombok.....	29
5.2	Első feladat	30

5.3	Második feladat	32
5.4	Harmadik feladat	35
5.5	Negyedik feladat.....	37
5.6	Ötödik feladat	40
5.7	Hatodik feladat	41
5.8	Hetedik feladat.....	43
6	Fejlesztési lehetőségek	45
7	Összegzés	48
8	Summary	49
9	Irodalomjegyzék	50
10	Ábrajegyzék.....	51
11	Táblázatjegyzék.....	52

1 Bevezetés

A szakdolgozatom egy ötszintes lift-modell rekonstrukciójáról szól. El kell végeznem a rendszer állapotának felmérését, esetleges hibák feltárását és javításokat kell alkalmaznom rajta.

A liftek olyan szerkezetek, melyek függőleges irányban szállítanak terheket vagy személyeket szintek között. Rendkívül elterjedtek és nélkülözhetetlenek a mai ember számára. Az Automatizálási és Infokommunikációs Intézeti Tanszéken található lift-modell a valós szerkezetek működésének bemutatására készült. Ezt a funkcióját az utóbbi időben nem tudta biztosítani, így szükségessé vált a rekonstrukció. A szakdolgozatomban leírom mindazon észrevételeket és hibákat, melyeket a rendszer megismerése és átvizsgálása során észleltem. Leírom az alkalmazott javításokat és egy módszert arra, hogyan vezérelhetjük a modellt.

A témám kifejtése megoldást ad egy kevesebb bemeneti csatlakozóval rendelkező PLC-vel való programozáshoz is. Leírom, hogyan lehet ezen eszközöket programozni, és hogyan lehet velük a lift-modellt a valódi liftekhez hasonló módon vezérelni. Példafeladatok kiírása és bemutatása során elmagyarázom a feladatokhoz szükséges funkciókat és kulcslépéseket a megfelelő vezérlés eléréséhez. A feladatok részben egymásra épülnek és fokozottan bonyolódnak, míg végül a program írója teljes vezérlést ér el a lift-modell felett. Ezen feladatok megfelelőek arra, hogy az oktatásba bevehetővé tegyék a rendszert. Kihívást jelentenek mind kezdő és haladó programozási tudással rendelkező hallgatók számára, további feladatokkal a rendszer alkalmas lehet versenyekre való felkészülésre, különböző programozási módszerek oktatására. Az általam bemutatott fejlesztőkörnyezet három programozási nyelvet támogat, így tovább növeli a rendszer oktatásba integrálhatóságát.

A szakdolgozatom alapján az óraadó megismerkedhet a lift-modell egészével, a megszerzett ismereteket átadhatja a hallgatóknak. Így a hallgatók nem csak egy újabb szimulációs eszközt látnak a gépezetben, hanem megismerkedhetnek valós problémákkal és megoldásokkal. Emellett közelebb kerülnek egy kézzel fogható technológiához, teljesen új tapasztalatokat szerezhetnek. Szeretném abban a tudatban elkészíteni a szakdolgozatomat, hogy valami olyat viszek véghez, amely sokat segíthet a tanszéken zajló oktatás fejlesztésében és a hallgatók változatos tapasztalatainak megszerzésében.

2 Célkitűzés

Elsődleges feladatom az Automatizálási és Infokommunikációs Intézeti Tanszék lift-modelljének megismerése, irányítástechnikai korszerűsítése és a megfelelő átalakítások elvégzése annak érdekében, hogy az oktatásba integrálható legyen. A munkálatok után a modellnek alkalmasnak kell lennie arra, hogy kezdő és haladó szintű feladatokat is lehessen hozzá kiírni. További feladatom a rendszerhez illeszthető PLC kiválasztása valamint példafeladatok írása megoldással, így egy az oktatásba bevezethető technológiát kapunk.

Célom a lift-modell megépítésével az irányítástechnikai tantárgyak gyakorlatiasabb megközelítése, a technológiai megoldások és a hallgatók között létrehozni egy közvetlen kapcsolatot. Ezáltal találkozhatnak az iparban használatos eszközökkel és problémákkal.

3 Lift-modell

A lift-modell egy négyemeletes ház modellezésére alkalmas. Acél vázszerkezete és lemez borítása egy négy kerekcsatára van rögzítve. Hát- és oldallapjai egyszerre leszerelhetők, előlapja öt szinte egyforma szegmensre oszlik. A hátoldalon található egy DB50-es csatlakozó, amin keresztül a vezérlés megvalósítható, valamint a megtápláláshoz szükséges villásdugó. A megtáplálásához 230 V feszültség, 50 Hz szükséges. Minden előlapi szegmensen található egy ajtó, egy hívógomb, egy zöld és egy piros LED. A legfelső szint felett található a liftgépház. Ebben van a liftkabin mozgásáért felelős motor, motorvezérlő és egy három relés motorleállító. Ez utóbbi felel a motor leállításáért, amint a liftkabin elérte valamelyik végállást. Ilyen módon a motor megvédhető a szoftveresen nem megfelelően megírt programokból származó hibáktól.

A motor tengelyén található egy műanyag henger és egy lánckerék, így egy lánc segítségével mozgathatja a liftkabint és az ellensúlyt. Az alsó és felső szinten van felszerelve egy-egy végálláskapcsoló, ezek szolgáltatnak jelet a reléknek. A liftkabinban található öt darab nyomógomb, öt LED valamint három kétállású kapcsoló. A kabin hátoldalán van négy darab karos-görgős kapcsoló. Ezek szolgáltatják a jelet a szintérzékeléshez. Ez olyan módon működik, hogy a kabin mögött, a vázszerkezetre van rögzítve egy kódléc, így a kapcsolók nem csak a szinteken, de a szintek között is különböző kombinációjú jelet szolgáltatnak. A modellben található továbbá egy 24 V-os feszültségű tápegység, mely az érzékelőket táplálja.



1. ábra: Lift-modell

3.1 Állapotfelmérés

A lift-modellen állapotfelmérést végeztem, ennek eredményeit és észrevételeimet a következő alfejezet taglalja.

A lift-modell hátoldalán egy DB50-es anya csatlakozó található. Ennek a csatlakozónak 37 lába van bekötve, ezeknek a bekötéseknek az adatai megtalálhatóak a bekötési táblázatban. A felhasznált DB50-es CANNON apa csatlakozón 40 láb van bekötve. A bekötési adatok és színek szintén megtalálhatóak a bekötési táblázatban. A lift-modell (a technológia oldaláról tekintve) összesen 21 kimenettel és 13 bemenettel rendelkezik. A 20-as lábon lévő vezeték nincs bekötve a rendszerbe. Ez nem szükséges a modell megfelelő működéséhez. A 29-es kimenet bármilyen funkció tesztelése alatt 24 V potenciálon van.

A liftkabinban található három darab kétállású kapcsoló jelzései: T, TT, MT. Jelentésük: Terhelt, Túlterhelt, Megtelt. Ezek felhasználhatóak komplex programozási

feladatokhoz. A liftkabinban található öt darab LED szintjelzésre, ezek katódja közösítve van. Mellettük található öt darab nyomógomb a megfelelő szintekhez. A liftmodell előlapján látható szintenként egy hívógomb, egy zöld és egy piros LED. A LED-ek katódja közösítve van, az öt darab piros LED csak egyszerre vezérelhető. A zöld LED-ek hívás visszajelzésként alkalmazhatók. A kabin végálláskapcsolóinak jele akkor 0, ha a kabin épp végállásban van. A liftkabin kábelharisnyája felfelé és lefelé történő mozgásnál is beakad a vázszerkezetbe, ez a hiba a megfelelő működéshez elengedhetetlenül kiküszöbölendő. A liftkabin a tehetetlensége miatt a motor leállítását követően még továbbsiklik, így a pontos megállás érdekében minden szint elérése előtt már le kell kapcsolni a motort.

3.2 Relék

A liftgépházban található három darab Omron MY4-es elektromechanikus relé.



2. ábra: Omron MY4 24 VDC relé

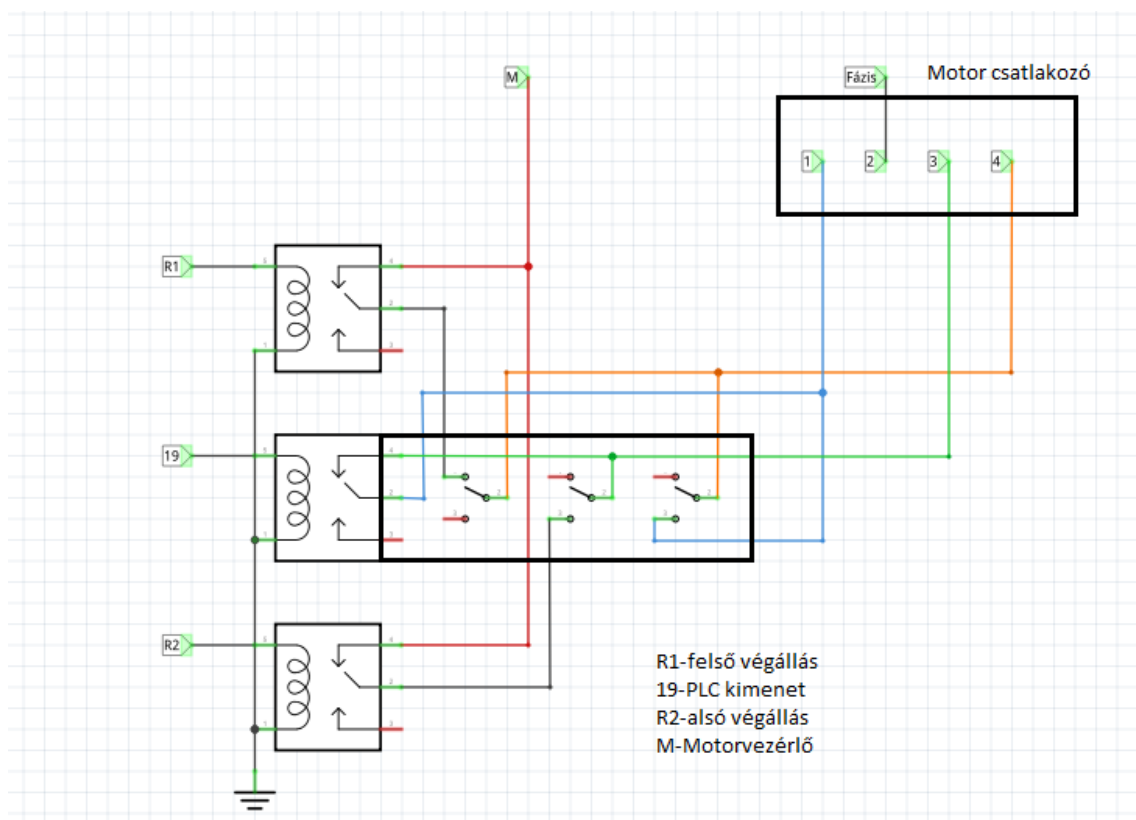
A relék lehetővé teszik, hogy egy jel vagy impulzus segítségével be-, illetve kikapcsolhassunk egy a jeltől független megáplálású elektromos eszközt. Gyakran egy kisebb feszültség vagy áram kapcsol egy nagyobb feszültséget vagy áramot. A viszonylag kis feszültségű/áramú jel egy egyszerű kapcsoló segítségével irányítható és az alacsony feszültség végett vékonyabb vezeték alkalmazható. Így a relé közel lehet az eszközhöz, a nagyobb áramot igénylő berendezés is távolról vezérelhető. Míg a szilárdtest relék gyorsabb kapcsolást tesznek lehetővé és megbízhatóbbak, az

elektromechanikus reléknek is van előnyük. Utóbbiak záró és bontó érintkezőkkel is rendelkeznek, egyszerre több ilyen kapcsolást tudnak kezelni és olcsóbbak, ha nagy feszültségek vagy áramok kapcsolására van szükség. [1]

Működésük

Az elektromechanikus relék tartalmaznak egy tekercset, armatúrát és legalább egy pár kontaktust. Ha áram folyik a tekercsben, az elektromágnesként viselkedik és mágneses teret képez. Ez magához húzza az armatúrát, amely zárja, illetve bontja az érintkezőket. A legelterjedtebb a nem reteszelő relé, amely a tekercs gerjesztése után egy rugó segítségével visszaáll a gerjesztés előtti állapotába. Ez a tulajdonsága rendkívül fontos olyan alkalmazásoknál, ahol egy előre meghatározott helyzetben kell lennie a relének áramkimaradás vagy a megáplálás megszűnése esetén.

Léteznek reteszelő relék is, ahol nincs alapállapot, a kapcsolásokhoz elég egy rövid impulzus és az állapotukat gerjesztés nélkül is megtartják. Viselkedésüket tekintve felfogható egy bistabil multivibrátornak. Egy egytekercses reteszelő relében a tekercsre adott feszültség polaritása határozza meg, hogy milyen állapotot vesz fel, míg egy kéttekercses kivitelben a második tekercs segítségével mozog az armatúra. [1]



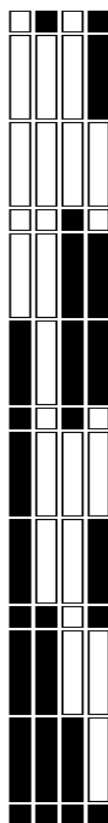
3. ábra: Relék kapcsolása

A 3. ábrán látható relék közül a felső, illetve alsó felel a motor leállításáért, amint elérte a végállást. Az R1 és R2 jeleket a szerkezetben felül valamint alul található karos-görgős végálláskapcsolók szolgáltatják. Ezek értéke addig logikai 1, amíg a liftkabin az adott véghelyzetbe nem ért. Akkor a megfelelő relé lekapcsolja a motorvezérlő csatlakozását a középső reléről. A 19-es jel a PLC oldaláról nézve tekintendő kimenetnek. Ez a jel felelős a motor irányának megadására. Ha a PLC kimenete nem aktív, a liftkabin felfelé fog mozogni, míg aktív kimenet esetén lefelé. A mozgás maga azonban egy külön kimeneten van vezérelve. Ha a középső relé gerjesztetlen állapotban van a motor csatlakozójának 1-es és 4-es pontja rövidre van zárva és 3-as láb a motorvezérlőre csatlakozik. Gerjesztett tekercs esetén a csatlakozó 1-es és 3-as pontja van rövidre zárva és a 4-es láb csatlakozik a motorvezérlőhöz. A 2-es számú kivezetés a motor megtáplálására szolgál, a fázisra van kötve. A csatlakozó érintkezőinek a száma nincs feltüntetve az eszközön, így ha balról jobbra rendre 1-nek, 2-nek, 3-nak és 4-nek nevezzük őket, akkor a következő színű vezetékek kapcsolódnak hozzájuk:

- 1: szürke, piros
- 2: barna
- 3: sárga, piros
- 4: sárga, piros

3.3 Kódléc

A lift-modell vázszerkezetének hátsó részére van rögzítve a kódléc. Ez négy darab különböző sávból áll, és ez szolgáltatja a jelet a liftkabin hátoldalán lévő négy karos-görgős kapcsolónak. A sávok kialakításuk végett alkalmasak kétállapotú jelek létrehozására, így vagy benyomódik a kapcsoló karja, vagy az pont egy barázdában van, így nem ad jelet. A négy jel tizenhárom kombinációban írja le a liftkabin jelenlegi helyzetét, minden szintnek, illetve szintek között két állapotnak egyéni jelkombinációja van. Az ábrán látható egy illusztráció a kódlécről, ahol a fehérén hagyott részeken a kapcsoló karját benyomja, a fekete téglalapok a barázdákat jelölik.



4. ábra: Kódléc illusztráció

A szintérezékelés a liftkabin pontos megállása érdekében elengedhetetlen. A későbbiekben tárgyalt változótábla részét képezik a szintérezékelők, melyek elnevezése G0, G1, G2, G3. A következő táblázat mutatja be a kapcsolók által szolgáltatott jelek értékeit a megadott szinteken.

1. táblázat: Szintérezékelés

	G3	G2	G1	G0
IV	1	0	1	0
-	1	1	1	0
-	1	1	1	1
III	1	1	0	1
-	1	1	0	0
-	0	1	0	0
II	0	1	0	1
-	0	1	1	1
-	0	1	1	0
I	0	0	1	0
-	0	0	1	1
-	0	0	0	1
FSZ	0	0	0	0

3.4 Be- és kimenetek

Minden ipari technológia, rendszer rendelkezik be-, illetve kimenetekkel. A következő táblázat tartalmazza a lift-modell összes be- és kimenetét. Ezek a PLC oldaláról tekintve értendők bemenetnek (I - Input) és kimenetnek (O - Output). Az ehhez hasonló rendszerek viszonylag alacsony számú be- és kimenetekkel rendelkeznek, ezeket könnyen lekezeli egy megfelelő moduláris PLC. A valódi technológiák szenzorjai akár több ezer bemeneti jelet szolgáltatnak az irányítástechnikai rendszernek, és akár több száz beavatkozó vezérlése is szükséges lehet. Az érzékelők jeleit dolgozza fel a PLC és azok értékeinek függvényében vezérli a kimeneteket.

A következő táblázatok a lift-modell hátoldali DB50-es anya csatlakozójának minden tűjét tartalmazzák, valamint a felhasznált CANNON kábel színek kódjait. A szimbólumok száma a csatlakozó érintkezőjének számával egyezik meg. A táblázatban szereplő első szín a vezeték alapszíne, második a csíkozás (ha van). Az I jelölés a PLC bemenete, az O jelölés a PLC kimenete, az NC nem használt, nem bekötött érintkezőt jelent (az angol Not Connected kifejezésből).

2. táblázat: Be- és kimenetek 1

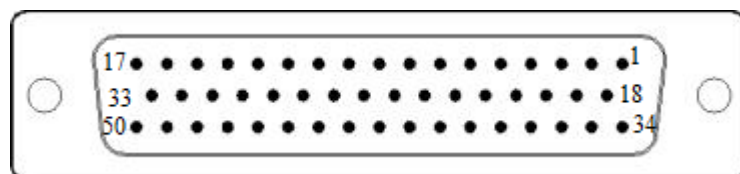
1	Barna	Zöld	FSZ LIFTBEN HG	I
2	Fehér	Sárga	I. LIFTBEN HG	I
3	Sárga	Barna	II. LIFTBEN HG	I
4	Fehér	Szürke	III. LIFTBEN HG	I
5	Zöld	Kék	IV. LIFTBEN HG	I
6	Szürke	Zöld	FSZ KINT HG	I
7	Rózsza	Piros	I. KINT HG	I
8	Sárga	Szürke	II. KINT HG	I
9	Szürke	Barna	III. KINT HG	I
10	Fehér	-	IV. KINT HG	I
11	Barna	-	Felső végállás INV	I
12	Rózsza	Zöld	Alsó végállás INV	I
13	Fekete	-	G0	I
14	Sárga	-	G1	I
15	Zöld	-	G2	I
16	Sárga	Fekete	G3	I
17	Szürke	-	Ajtók zárva	I
18	Fehér	Zöld	LIFT mozgatus	O
19	Barna	Piros	LIFT irány (0-fel, 1-le)	O
20	Fehér	Piros	-	NC
21	Barna	Fekete	FSZ KINT LED	O
22	Zöld	Fekete	I. KINT LED	O
23	Rózsza	Barna	II. KINT LED	O
24	Szürke	Kék	III. KINT LED	O
25	Rózsza	Kék	IV. KINT LED	O

A csatlakozó első 10 tűjén a hívógombok szerepelnek HG jelzéssel. Minden szintnek van hívógombja a modell előlapján és a liftkabinban egyaránt. A G0, G1, G2 és G3 bemenetek szolgáltatják az előzőleg tárgyalt szintérzékelő jeleket a PLC számára. A 17-es számú bemenet az ajtók zárását ellenőrzi. Az ajtók érzékelői sorba vannak kötve, így ha egy ajtó is nyitva van, a jel logikai 0 értéket vesz fel. A 18-as és 19-es kimenetek szolgálnak a liftkabin mozgatusáért. Ha a 18-as kimenet értéke 0, a liftkabin nem mozog. Ha az értéke logikai 1, akkor a 19-es kimenet értékének függvényében fog mozogni. Ha ennek az értéke logikai 0, a mozgás felfelé irányul, ha logikai 1 az értéke a liftkabin lefelé fog mozogni. A LED-ek funkciója feladatonként eltérhet.

3. táblázat: Be- és kimenetek 2

26	Fehér	Fekete	PIROS KINT LED	O	
27	Szürke	Piros		NC	
28	Sárga	Rózsza		NC	
29	Sárga	Kék	(eddig mindig 1)	I	
30	Zöld	Piros	0V	0V	
31	Fehér	Rózsza	T LIFTBEN	I	
32	Kék	-	MT LIFTBEN	I	
33	Rózsza	-	TT LIFTBEN	I	
34	Piros	Kék	FSZ LIFTBEN LED	O	
35	Szürke	Rózsza	I. LIFTBEN LED	O	
36	Barna	Kék	II. LIFTBEN LED	O	
37	Sárga	Piros	III. LIFTBEN LED	O	
38	Fehér	Kék	IV. LIFTBEN LED	O	
39	Lila	-		NC	
40	-	-		NC	
41	-	-		NC	
42	-	-		NC	
43	-	-		NC	
44	-	-		NC	
45	-	-		NC	
46	-	-		NC	
47	-	-		NC	
48	-	-		NC	
49	-	-		NC	
50	Piros	-	+24 V	+24 V	

Az előlapi öt piros LED csak együtt működtethető, ajtó nyitás jelzésére használható. A liftkabinban lévő LED-ek jelezhetik a jelenlegi szintet, a külső zöld LED-ek nyugtázást jelenthetnek. A 29-es érintkező közvetlenül a motorvezérlőhöz csatlakozik, ám funkciója ismeretlen. Tesztelések sorozata után elmondható róla, hogy minden esetben logikai 1 volt az értéke. A 30-as csatlakozó pont a 24 V-os tápegység és a PLC közös földpontja, az 50-es pont a +24 V.



5. ábra: DB50-es csatlakozó

4 Programozható logikai vezérlők

4.1 PLC alapok

A PLC megnevezés az angol Programmable Logic Controller, azaz programozható logikai vezérlő kifejezésből származik. A PLC egy ipari vezérlő eszköz, amely folyamatosan figyeli a bemeneti jeleinek állapotát, és a belső program logikája alapján vezérli a kimeneteit. A PLC-k elterjedése felváltotta a relés vezérléseket, ugyanazon funkciókat látták el. Akkoriban a relék hajlamosabbak voltak meghibásodni, a technikusoknak ilyen esetekben egy egész falnyi relé táblát kellett átvizsgálni a hiba megtalálásához.

A PLC-k robosztus kivitele segít abban, hogy ipari környezetben ellenálljanak magas hőnek, hidegnek, pornak és magas páratartalomnak egyaránt. A ma használt eszközök zöme moduláris kivitelű, így könnyedén módosíthatóak különböző feladatok elvégzésére. Programozásuk egyszerű, az IEC61131-3 szabvány által definiált öt programnyelv közül legelterjedtebb a létradiagramos programozási nyelv (LAD). A PLC-k hátránya, hogy nem teljesítenek megfelelően komplex adatfeldolgozás esetében. PLC-k mellett külső monitorok használata szintén gyakran előfordul, mivel önmagukban alkalmatlanok adatok megjelenítésére. [2]

4.1.1 PLC-k hardver komponensei

A PLC-k moduláris felépítésűek, így minden főbb funkciójukra külön modul áll rendelkezésre. Alapvető modulok a tápegység modul, amely a PLC megtáplálásához elengedhetetlen, a CPU modul, a be- és kimeneti modulok és a kommunikációs modul. A PLC agyaként a központi feldolgozó egység, angolul Central Processing Unit, azaz CPU szolgál. Ez a mikroprocesszor, amelyben helyet foglal a memória chip és integrált áramkörök, melyek vezérlik a logikát, a monitorozást és a kommunikációt.

A memória tárhelyet biztosít az operációs rendszernek, melynek adatait a CPU használja. A rendszer ROM-ja tárolja az adatokat az operációs rendszernek, míg a RAM-ban tárolódik a bemeneti és kimeneti eszközök állapot információja, valamint értékek a számlálók és időzítők számára. Ahhoz, hogy adatot írjunk a CPU-ba, a PLC-nek szüksége van egy programozó eszközre, például számítógépre. Vannak azonban programozó felülettel ellátott, úgynevezett mikro PLC-k. A PLC bemeneti jeleket olvas különböző bemeneti eszközökből, szenzorokból. Ezek lehetnek billentyűzetek,

kapcsolók, érzékelők. A bemeneti jel lehet digitális vagy analóg. Kimenatként szolgálhat egy motor, lámpa vagy szolenoid szelep.

A kimenetek kapcsolásához az egyik leggyakrabban használt elem a relés kimeneti modul. A relék nem polarizáltak, így váltakozó és egyenáram kapcsolására is alkalmasak. A relés kimenet hátránya, hogy lassú kapcsolásúak a tranzistoroshoz képest (5 – 50 ms), ám nagyobb áramok kapcsolhatóak velük. A tranzisztoros kimeneti modul nem csak gyorsabb kapcsolást eredményez, de az élettartama is hosszabb. Hátránya azonban, hogy csak egyenáram kapcsolható vele. [2]

4.1.2 PLC programnyelvek

Több programozási nyelv használható a PLC-k programozására. Ezek közül öt a nemzetközi IEC61131-3 szabvány által van definiálva. A létradiagramos programozási nyelv (LAD) a legelterjedtebb és szinte minden PLC támogatja. Ez a relés vezérlések tervezésére szolgáló áramút tervből alakult ki. Kézenfekvő megoldás volt ennek a nyelvnek a kifejlesztése, hiszen a relés vezérléseket hivatott kiváltani a PLC. Az áramút terv készítésében jártas szakemberek számára így egyszerűbb volt beletanulni az eszköz programozásába. Szimbólumok reprezentálják a nyitó és záró kontaktusokat, időzítőket, számlálókat, tekercseket és matematikai operációkat. Ezen szimbólumok soros, párhuzamos vagy vegyes kapcsolása során alakul ki a megfelelő logika alapján a program egy létrafoka, melyet network-nek nevezünk. A programozási szabályok megismerése után komplex programok írására és felügyelésére kiválóan alkalmas.

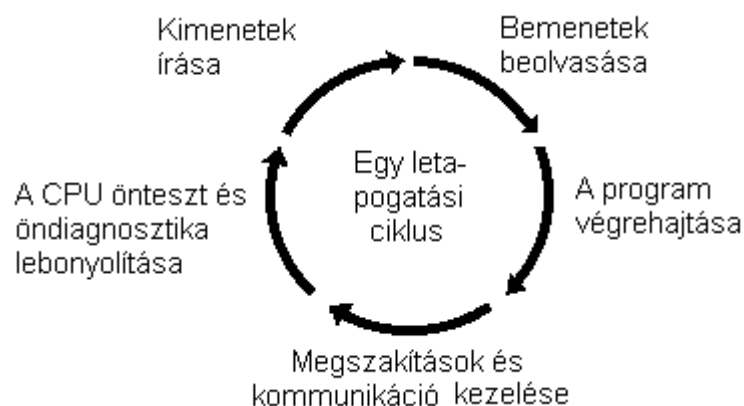
Egy másik programozási nyelv a funkcióblokkos programnyelv (FBD). Ebben a különböző funkciókhoz bemenet és kimenet között különböző blokkok tartoznak. Ezen blokkok bal oldalán találhatóak a bemenetek, jobb oldalán a kimenetek. A logika felépítéséhez a blokkokat vezetékekkel a megfelelő módon össze kell kapcsolni. A létradiagram kontaktusai és a funkcióblokkok gyakran egy programon belül is alkalmazhatóak, így rugalmasabb programozás hozható létre.

A strukturált szöveg (ST) egy magas szintű programozási nyelv. Ebben mondatszerű utasításokat használhatunk, így egy nagyon rugalmas programot kapva. Összetett utasításokat, ciklikus végrehajtást, feltételes végrehajtást és matematikai függvényeket is támogat. A programozók if/then/else vagy repeat/until utasításokat is használhatnak.

Az utasításlistás programozási nyelv (IL) egyidős a PLC-k kialakulásával. Programvezérlési utasítások egymás után történő meghívásából áll a program. Ez egy alacsony szintű programozási nyelv. Az utasítások műveleti részből és operandusból állnak. A műveleti rész határozza meg, hogy a processzor milyen műveletet hajtson végre, és az operandus határozza meg, hogy azt melyik adaton.

A sorrendi funkcióábra programozási nyelv (SFC) alkalmas komplex vezérlő rendszerek és szekvenciális hálózatok programozására. A sorrendi funkcióábra a vezérlési feladat folyamatát tartalmazza. Az ábra elemei a lépés, az átmenet és annak feltétele valamint a lépéshez tartozó akció.

Egy PLC a következő ábra szerinti funkciókat végzi el egy letapogatási ciklusban a következő sorrendben: bemenetek olvasása, program végrehajtása, kommunikációs kérések kiszolgálása, önteszt és más diagnosztikai funkciók ellátása és a kimenetek írása.



6. ábra: PLC letapogatási ciklus

4.2 A PLC kiválasztása

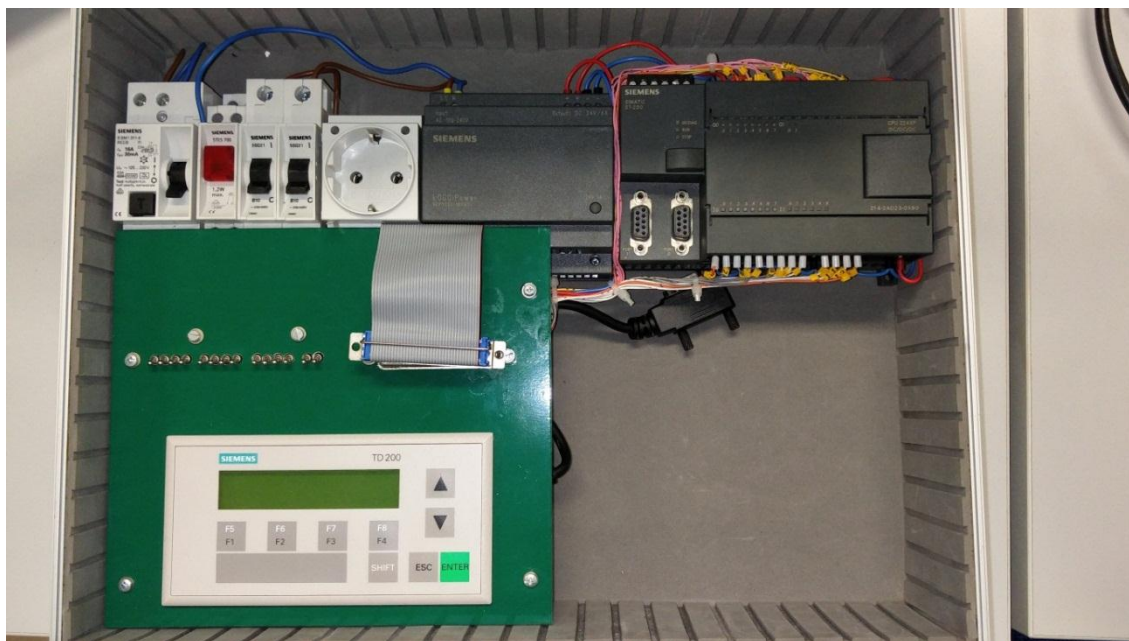
A megismert lift-modell a PLC oldaláról tekintve 20 db bemenettel és 13 db kimenettel rendelkezik. Kézenfekvő megoldás lenne egy elterjedt moduláris PLC választása a megfelelő bővítő modulokkal, azonban nekem az oktatásba integrálhatóság volt az elsődleges feladatom. Megoldható lenne, hogy egy PLC folyamatosan rá van kötve a modellre, és mindenki azt programozná, de így kihasználatlanok lennének a tanszéken lévő bördöndözött PLC-k, a hallgatók nem kerülnének elég közel a technológiához, és folyamatosan le lenne foglalva a szóban forgó PLC.

A tanszék birtokában van nyolc darab bőrdözt Siemens S7-200-as PLC, melyek korábbi feladatokra lettek tervezve, jelenleg azonban minimális a kihasználtságuk az oktatásban. A használandó PLC-k közül négy darab 224-es CPU-val és négy darab 224XP-s CPU-val rendelkezik. Azonban ezek a mikro PLC-k csak korlátozott be- és kimenettel rendelkeznek. Mindkét eszköznek 14 bemeneti és 10 kimeneti érintkezője van, azonban a bőrdbe épített szimulátor kialakítása miatt csak 10 db bemenet van bekötve a DB37-es csatlakozóba. Ebben az esetben már a lift-modell 10 db hívógombja lefoglalná az összes rendelkezésre álló bemenetet.

Elképzelhető egy csökkentett funkciókkal való programozás, azonban a feladatom egy olyan rendszer kialakítása, mely valós problémák megoldására taníthatja a hallgatókat. A fennálló problémára megoldást találtam egy segéd PLC formájában, melyet a későbbiekben fejtek ki.

4.3 S7-200 CPU

Az S7-200 CPU egy nagyteljesítményű mikro PLC, amely kisméretű tokban tartalmazza a mikroprocesszort, egy integrált tápegységet, a bemeneti és a kimeneti áramköröket. Külső kialakítása lehetővé teszi a felszerelést a szabványos (DIN) sínre, valamint az állapotjelző be- és kimeneti LED-ek meglétét. Sorkapocs csatlakozója levehető CPU 224-nél valamint CPU 224XP-nél egyaránt. Fizikai méretei CPU 224 esetén 120,5 x 80 x 62mm, CPU 224XP esetén 140 x 80 x 62mm. Mindkét CPU tápfeszültsége 24 VDC, 14 db 24 V-os bemenettel és 10 db relés 24 V-os kimenettel rendelkeznek. Különbség a két CPU között, hogy a 224-es csak egy, a 224XP-s viszont két darab RS-485 kommunikációs porttal rendelkezik. A felhasználható programméret futás közben szerkesztés nélkül CPU 224 esetén 12KB, CPU 224XP esetén 16KB. A CPU 224XP rendelkezik két darab analóg bemenettel és egy analóg kimenettel. Mindkét CPU-hoz hét darab bővítő modul csatlakoztatható. További közös jellemzőjük, hogy összesen 256 db időzítőt tudnak lekezelni. Ebből 4 időzítő 1 ms-os, 16 időzítő 10 ms-os és 236 időzítő 100 ms-os. 256 db számláló és 256 bit belső memória áll rendelkezésünkre a programozás során. [3]



7. ábra: CPU 224XP

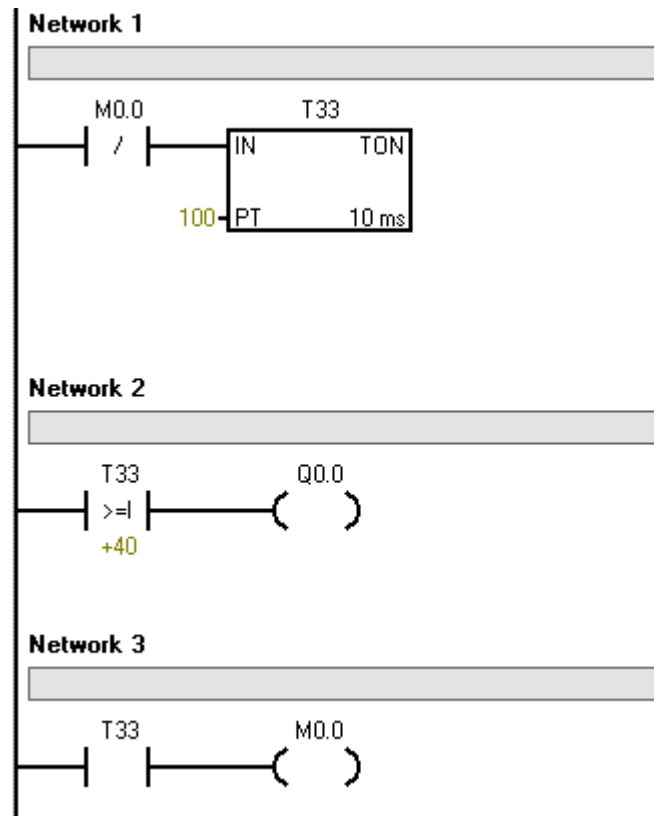
Az S7-224XP CPU mellett a bőrdobozban helyet foglal egy TD 200 kijelzőpanel. Ez egy olyan szövegkijelző egység, mely könnyedén csatlakoztatható az S7-200-hoz és alkalmas kétsoros, 20 karakteres üzenetek megjelenítésére. A TD 200 varázsló segítségével egyszerűen beprogramozhatjuk az S7-200-at, hogy a rendszerünkkel kapcsolatos szövegüzeneteket és egyéb adatokat jelenítsen meg. Ez illesztési lehetőséget biztosít az eszköznek, hogy megtekinthessük, figyeljük vagy megváltoztathassuk a rendszerrel kapcsolatos változókat.

Az S7-224 és S7-224XP CPU programozására alkalmas szoftver a STEP 7-Micro/WIN. Ez a fejlesztőkörnyezet kifejezetten az S7-200-as CPU-khoz lett kifejlesztve. A programozáshoz egy USB/PPI Multi Master kábelre van szükség, melynek USB csatlakozója a programozó eszközre, RS-485 csatlakozója pedig a CPU kommunikációs portjára csatlakozik. A megfelelő működéshez elengedhetetlen, hogy a Micro/WIN megfelelő verziója legyen feltelepítve és támogassa az USB-n keresztül történő programozást. Esetünkben a program verziószáma V4.0.9.25.

A következő néhány lépésben megmutatom, hogyan kell csatlakoztatni, programozni és futtatni egy S7-200-ast. Első lépés a CPU-t tápfeszültség forrásra csatlakoztatni. Esetünkben a CPU egyenfeszültséget igényel, így egy külső 24 V-os tápegység szükséges a megápláláshoz. A 24 V-os tápegység negatív pólusát az M jelzésű, pozitív pólusát az L+ jelzésű sorkapocsba kell csatlakoztatni. A következő lépés

a korábbiak alapján csatlakoztatni a CPU-t a programozó eszközhöz. Ezután el kell indítani a Micro/WIN-t és létre kell hozni egy új projektet (File/New). A képernyő bal oldalán láthatjuk a navigációs sávot, mely tartalmazza a projekt elemeket. A Set PG/PC Interface ikonra kattintva elérjük az elérési út felületet, ahol a legördülő menüben ki kell választani a PC/PPI kábelt. A kiválasztott kábel tulajdonságait a Properties... gombra kattintva tudjuk megváltoztatni. A PPI fülnél ellenőrizzük, hogy a címe 0-ra legyen állítva, valamint a Local Connection fülön belül ki kell választani az USB csatlakozást.

Ezek jóváhagyása után leellenőrizzük a megfelelő kommunikáció meglétét azzal, hogy bekérjük a CPU-tól a típusát. Ehhez az eszköztárból a PLC/Type... parancsra kell kattintani, majd a felugró felületen a PLC típusának kézi beállítása helyett a Read PLC gombra kattintunk. Ekkor a PLC Type mezőben CPU 224-nek vagy CPU 224XP-nek kell megjelennie, alatta a megfelelő verziószámmal. Jóváhagyás után a program megírása következik. A program hat utasítást három hálózatban (network) tartalmaz, így létrehozunk egy önindító időzítőt, ami törli önmagát. A programszerkesztő a navigációs sáv Program Block ikonjával érhető el. Ebben a példában létradiagramos programozási nyelvet használunk. A LAD utasításokat a programszerkesztő bal oldalán található könyvtárban vagy utasításfában találjuk meg. A program a 8. ábra alapján alakul. [3]



8. ábra: Időzítő példaprogram [3]

Az első hálózatban az M0.0 memóriabit alapértéke 0, így a bontóérintkező engedélyezi az időzítő bemenetét. Ez egy bekapcsolást késleltető időzítő 10 ms-os felbontással, amelyet T33-as címmel érhetünk el. Összesen 100*10 ms-ig, azaz egy másodpercig nem engedélyezi a kimenetét.

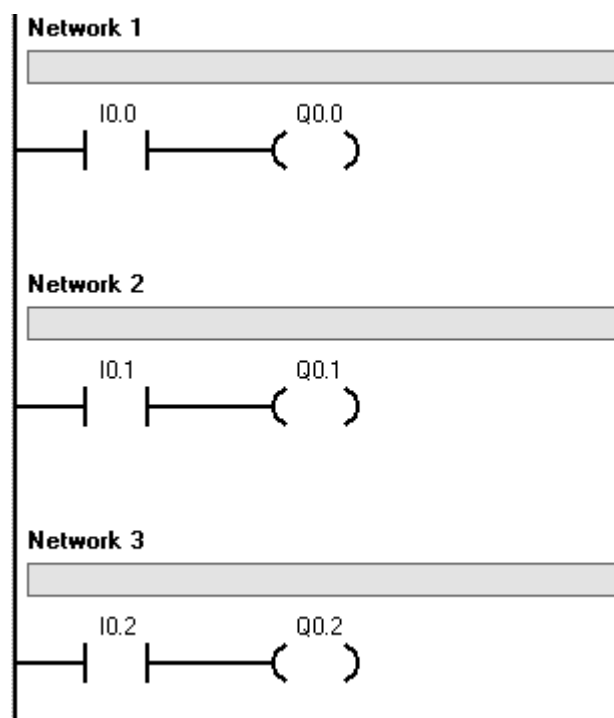
A második hálózatban egy komparátor és egy tekercs található. Amikor a T33-as időzítő értéke eléri a 40-et, ami a bekapcsolástól számított 400 ms, akkor az érintkező biztosítja a szabad áramfolyást a tekercs számára, így a CPU Q0.0-ás kimenete bekapcsol.

A harmadik hálózatban indítástól számított egy másodpercnél a T33-as időzítő kimenete igaz lesz, így a záró érintkező 0 értékről 1-es értékre állítja a memóriabitet. Mivel az első hálózatban az időzítés feltétele a memóriabit 0-s értéke, így az időzítő törlődik. Ezzel kaptunk egy 60%-os kitöltési tényezőjű, 1 s periódusidejű négyszögjelet a CPU Q0.0-ás kimenetén.

A mintaprojekt elmentését követően (File/Save as...) azt le kell tölteni a PLC-re. Ezt az eszközsávon belül a File/Download... paranccsal érhetjük el. A felugró letöltési panelen a Download gombra kattintva letöltjük a programot. Ügyelni kell arra, hogy letöltés közben a CPU STOP állapotban legyen, különben a folyamat nem megy végbe.

Ha a PLC üzemmód kapcsolója TERM állásban van, akkor a CPU szoftveresen RUN, azaz futó állapotba állítható. Azonban ha STOP állásban van a kapcsoló, akkor azt át kell állítani RUN vagy TERM állásba. Amint a PLC futó állapotba kerül a Q0.0-s kimenetének állapotjelző LED-je villogni kezd, ahogy a CPU futtatja a programot. A megírt program megfigyelhető futás közben a Debug/Program Status menüparanccsal.

A bőrdözött S7-200-as PLC-k mindegyike mellett található egy beépített szimulátor. Ez 14 db kétállású kapcsolóból áll, melyek segítségével szimulálhatjuk egy rendszer kimeneteit, így a PLC bemenetei a megfelelő értékeket kapják. A kapcsolók mellett található egy beépített DB37-es anya csatlakozó, mely közvetlenül a PLC be- és kimeneteihez van csatlakoztatva. A kétállású kapcsolók egy 37 eres szalagkábelhez csatlakoznak, a kábel végén egy DB37-es apa csatlakozó található. Így a szimulátor könnyedén csatlakoztatható vagy leválasztható a PLC-ről. Egyik CPU-nak sincs bekötve az összes bemenete erre a csatlakozóra. A használható 14 bemenetből csak 12 van bekötve, így 2 bemenet megmaradt későbbi, esetleg szimulátoron kívüli használatra. A kapcsolók által betöltött funkció elengedhetetlen bizonyos programok teszteléséhez, így fontos a működésük ellenőrzése. A következő, a CPU-ra feltöltendő program a szimulátor működését teszteli.



9. ábra: Bemenetek tesztelése (részlet)

A 9. ábrán látható az a programrészlet, amely segítségével meggyőződhetünk a szimulátor működéséről. A program egyszerű, amint egy bemeneti záró kontaktuson áram halad, az aktivizál egy tekercset. Minden hálózat egy bemenettel egy kimenetet tesztl le. A program feltöltése után a CPU fedőlapján lévő állapotjelző LED-ekkel, valamint a Debug/Program Status menüparancs segítségével egyesével ellenőrizhetjük a kapcsolatokat.

4.4 PLC illesztése a rendszerhez

Korábban említettem, hogy a lift-modellnek túl sok be- és kimenete van ahhoz, hogy S7-200-as PLC-vel bővítő modul nélkül programozhassuk. Megoldást kellett találnom arra, hogy lecsökkentsem a ki- és bemeneteket anélkül, hogy lényeges funkcióktól fosztanám meg a rendszert. A bőrröndözött S7-200-as CPU-khoz kötött DB37-es csatlakozó bekötése a következő.

4. táblázat: DB37-es csatlakozó

1	-	20	Q0.0
2	I0.0	21	Q0.1
3	I0.1	22	Q0.2
4	I0.2	23	Q0.3
5	I0.3	24	Q0.4
6	I0.4	25	Q0.5
7	I0.5	26	Q0.6
8	I0.6	27	Q0.7
9	I0.7	28	Q1.0
10	I1.0	29	Q1.1
11	I1.1	30	-
12	I1.2	31	-
13	I1.3	32	-
14	-	33	-
15	-	34	-
16	-	35	-
17	-	36	-
18	-	37	-
19	0 V		

A táblázatból látszik, hogy a PLC-knek csak 12 bemenete van bekötve. Emellett elengedhetetlen a közös földpont, valamint a 10 kimenet. Ezt kellett valahogy a lift-modellhez szükséges 20 db bemenethez és 13 db kimenethez illeszteni. Természetesen

néhány funkció elvesztése nem jelent problémát. A kimenetek kezelése egyszerűbb feladatnak bizonyult. Három kimenet megszüntetésére volt szükség, úgy döntöttem a liftkabinban lévő szintjelző LED-ek a legkevésbé fontosak. A lift szintjelzése megoldható a külső LED-ekkel, ha erre van igény egy feladatban. Így a kimenetek korlátozott száma nem jelent további problémát.

A bemenetek illesztése nagyobb fejtörést igényelt. Fontos volt a külső és belső hívógombok megtartása, a szintérzékelés és az ajtó nyitás érzékelése. Ezek összesen 15 db bemenetet igényelnének. Ebben már nincs benne a végállás érzékelése, hiszen ez megoldható a szintérzékeléssel, illetve a liftkabinban lévő kétállású kapcsolók.

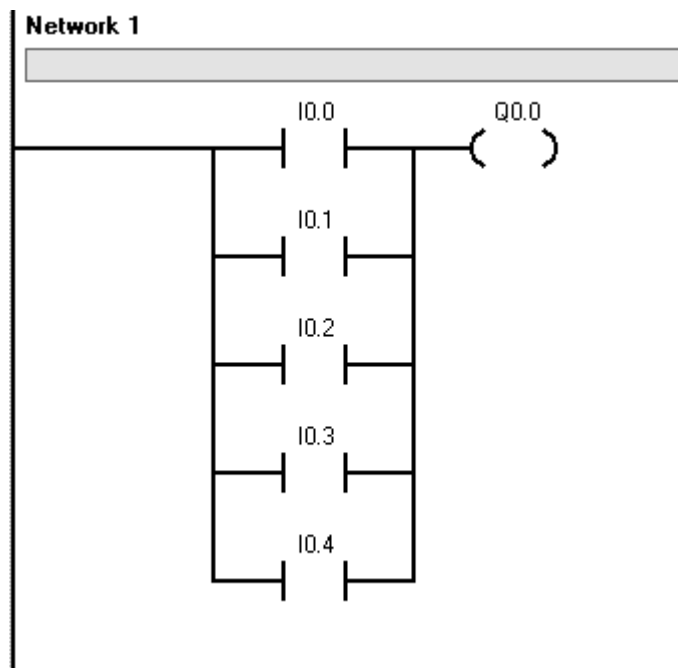
A problémára megoldást jelentett, ha valahogyan kódolom a 10 db hívógomb megnyomását kevesebb kimeneten. Ekkor született meg egy segéd PLC elképzelése. A segéd PLC kódoló funkciót lát el a lift-modell és a vezérlő PLC között. A 10 db hívógomb azzal együtt, hogy semmit nem nyomunk összesen 11 kombináció, így kódolható 4 biten. Ha a hívógombok kódolása 4 biten történik, a vezérlő PLC-nek 10 helyett csak 4 bemenetét foglalja le a szintadás. Ilyen módon további négy bemenet kell a szintérzékeléshez és egy bemenet az ajtók zárásának érzékeléséhez.

A segéd PLC kiválasztásának fő kritériuma volt a kis méret és a megfelelő mennyiségű ki-, illetve bemenet. Erre a feladatra alkalmasnak találtam a tanszék egyik S7-200-as CPU-ját. Ez a 224-es CPU annyiban eltér a bőrdözött változattól, hogy változó áramú a megáplálása. Ez további előnyt jelentett, mivel egy kisebb dobozt képzeltem el a lift-modell és a vezérlő PLC közé. A programjáról is megvolt addigra az elképzelésem. Egyszerűen, binárisan kódolom a különböző bemenetekre bekötött hívógombok jelét és ezt négy kimeneten kivezetem a vezérlő PLC számára. A kimeneteket S0, S1, S2 és S3 névvel illetem a táblázatban, és ezzel a névvel szerepelnek a feladatokhoz adott segédanyagban is.

5. táblázat: Segéd PLC be- és kimenetek

	S3	S2	S1	S0	Bemenet
Semmi	0	0	0	0	-
FSZ bent	0	0	0	1	I0.0
FSZ kint	0	0	1	0	I0.5
I bent	0	0	1	1	I0.1
I kint	0	1	0	0	I0.6
II bent	0	1	0	1	I0.2
II kint	0	1	1	0	I0.7
III bent	0	1	1	1	I0.3
III kint	1	0	0	0	I1.0
IV bent	1	0	0	1	I0.4
IV kint	1	0	1	0	I1.1
Kimenet	Q0.3	Q0.2	Q0.1	Q0.0	

A ki- és bemenetek az 5. táblázatban a segéd PLC-re vonatkoznak. A kimenetek azért is kaptak S jelölést, mert a későbbiekben összekavarható lenne a vezérlő és a segéd PLC kimenete. A segéd PLC programja négy hálózatot tartalmaz a következő részlet alapján.



10. ábra: Segéd PLC program (részlet)

Minden hálózat egy tekercset és a hozzá tartozó feltételeket tartalmazza. A feltételek a bemenetek egyszerű vagy kapcsolata a kódolási táblázat alapján. A Q0.1 és a Q0.2 esetén négy bemenet, a Q0.3 esetén három bemenet szerepel vagy kapcsolatban. A segéd PLC-re írt program nem kezeli azt a hibát, mely több hívógomb egyidejű megnyomásából adódna. Másik fontos dolog, hogy ha nem megfelelő időpillanatban engedünk el bizonyos gombokat, akkor hibás kombináció jelenik meg a kimeneten. Ennek a problémának a megoldására írtam elő legalább 100 ms-os gomb lenyomást a példafeladatokban. Ilyen hosszú idő nem lenne szükséges (5-10 ms is elegendő), de ez is kényelmesnek mondható és nem kell a korlátozott számú, nagyobb felbontású időzítőkből használni a programban. A példafeladatokhoz tartozó segédanyagban a segéd PLC teljes programja szerepel.

A program megírása után azt feltöltöttem a CPU-ra, majd teszteltem a funkciókat. Mivel minden megfelelően működött, így bekötöttem a rendszerbe, majd bedobozoltam. A dobozba bemegy egy villásdugó, amely a PLC-t táplálja, a lift-modell hátlapjáról a DB50-es csatlakozó, valamint a vezérlő PLC csatlakoztatására alkalmas DB37-es csatlakozó. A segéd PLC mellé került egy sorkapocs, amely a lift-modell és a vezérlő PLC közötti közvetlen csatlakozást teszi lehetővé a korábban tárgyalt be- és kimenetek esetében. A dobozba kerülő PLC összes bemenetét és kimenetét bekötöttem, RUN módba kapcsoltam, majd lezártam a dobozt. Így ha tápfeszültséget kap, alapértelmezetten futó üzemmódban indul, és folyamatosan futtatja a programot. Ezután csatlakoztattam a segéd PLC-t a lift-modell és a vezérlő PLC közé és különböző programokkal teszteltem a funkciókat. Minden tesztelés sikeresnek bizonyult, így következhetett a példafeladatok kidolgozása.

5 Példafeladatok

Az általam kidolgozott példafeladatoknak alkalmasnak kell lennie arra, hogy egy kezdő szintű PLC programozó számára fejlődési lehetőséget biztosítsanak és egy gyakorlottabb hallgató is kihívást találjon valamelyik feladatban. Programozási alapismeretek azonban szükségesek már a legegyszerűbb feladat megoldásához is. A kiírás rövid feladatismertetéssel kezdődik, majd a technológia kellő szintű bemutatásával éri el, hogy a hallgatók tisztában legyenek a rendszer működésével. A példafeladatok és az azok elvégzéséhez szükséges bemutató mellékletként megtalálható a szakdolgozatom mellett. A kiíráshoz megoldást is készítettem, valamint a megfelelő vezérléshez szükséges lépéseket, funkciókat is elmagyarázom ebben a fejezetben.

5.1 Feladatismertetés

A Lift-modell egy négyemeletes ház modellezésére alkalmas. A feladat ennek a szerkezetnek a megfelelő irányítása PLC-vel. Elsősorban a bőrröndözött S7 200-as PLC-t használjuk erre a feladatra, a segéd PLC is így van kialakítva. Lehetséges azonban a segéd PLC kihagyása és egy 16/16-os IO-val rendelkező vezérlő alkalmazása ugyanezekkel a funkciókkal, vagy egy legalább 20 bemenettel rendelkező PLC alkalmazásával az összes funkció programozható.

5.1.1 Bemenetek

A Lift-modell öt szintjének (FSZ, I, II, III, IV) érzékelésére négy darab mikrokapcsoló (G0, G1, G2, G3) szolgáltatja a jelet. A Gray kódolású szintérzékelés minden egész szint között további két különböző jelet szolgáltat. Így 13 kombinációjú jel érkezik a négy bemeneten. Az öt darab ajtónak egy közös érzékelője van, mely 1-es értékkel jelzi, ha minden ajtó zárva van. A liftkabinban található öt darab nyomógomb az összes szinthez, valamint minden szinten van egy hívógomb a modell előlapján. A 10 darab nyomógomb 4 biten van kódolva, ezt a kódolást végzi el a segéd PLC.

5.1.2 Kimenetek

Minden külső hívógomb mellett található egy zöld és egy piros LED. A piros LED-ek egy kimeneten vannak vezérelve, ezek jelzik, ha az ajtók valamelyike nyitva van. A zöld LED-ek a hívógomb nyugtázását igazolják, addig világítanak, amíg meg

nem érkezik a lift az adott szintre. A motor vezérlése 2 kimeneten történik. Az egyik kimenet adja a motornak az irányt, a másik kimenet vezérli a motor mozgását. A motor felfelé mozog, ha az irányának a kimenete logikai 0 és a mozgás logikai 1. Lefelé mozog, ha az irány és a mozgás értéke is logikai 1.

5.1.3 I/O tábla

6. táblázat: Feladat bemenetek

1	Hívógomb kódolás S0	I0.0
2	Hívógomb kódolás S1	I0.1
3	Hívógomb kódolás S2	I0.2
4	Hívógomb kódolás S3	I0.3
5	Szintérzékelés G0	I0.4
6	Szintérzékelés G1	I0.5
7	Szintérzékelés G2	I0.6
8	Szintérzékelés G3	I0.7
9	Ajtók zárva	I1.0

7. táblázat: Feladat kimenetek

1	Motor mozgás	Q0.0
2	Motor irány	Q0.1
3	LED FSZ	Q0.2
4	LED I	Q0.3
5	LED II	Q0.4
6	LED III	Q0.5
7	LED IV	Q0.6
8	LED Piros	Q0.7

5.1.4 Szintérzékelés

8. táblázat: Feladat szintérzékelés

	G3	G2	G1	G0
IV	1	0	1	0
-	1	1	1	0
-	1	1	1	1
III	1	1	0	1
-	1	1	0	0
-	0	1	0	0
II	0	1	0	1
-	0	1	1	1
-	0	1	1	0
I	0	0	1	0
-	0	0	1	1
-	0	0	0	1
FSZ	0	0	0	0

5.1.5 Hívógombok

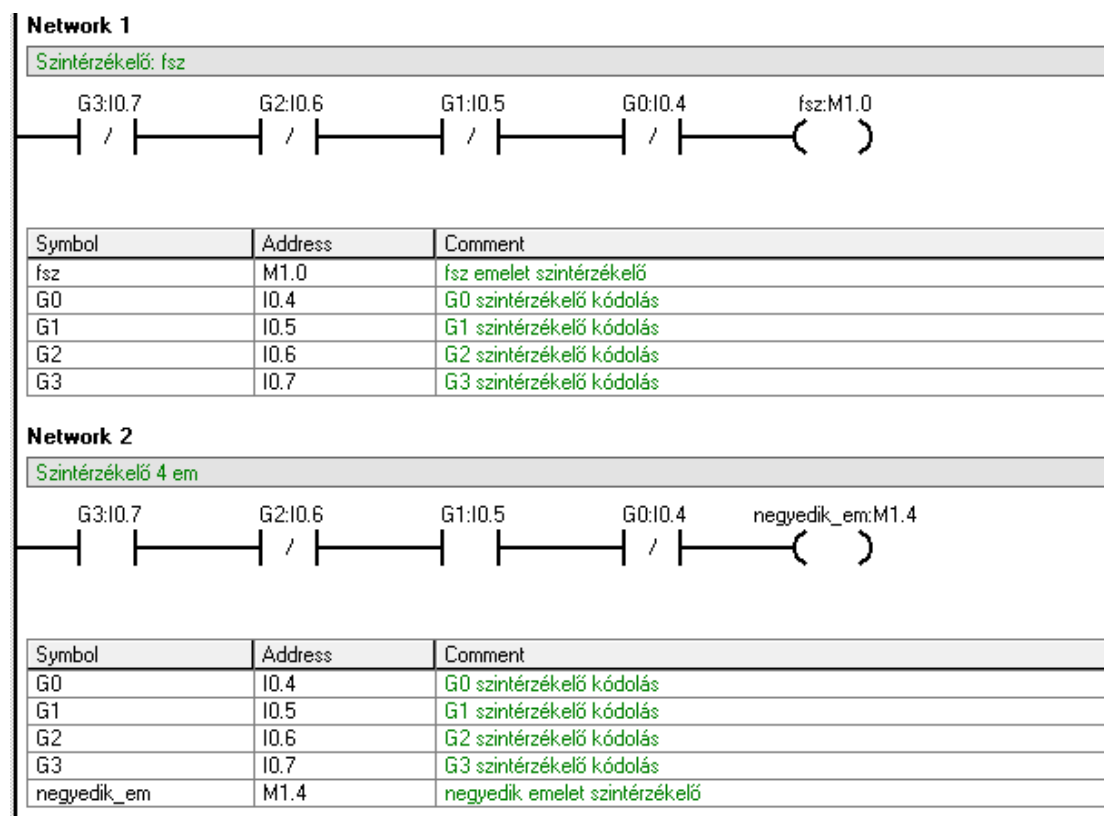
9. táblázat: Feladat hívógombok

	S3	S2	S1	S0
Semmi	0	0	0	0
FSZ bent	0	0	0	1
FSZ kint	0	0	1	0
I bent	0	0	1	1
I kint	0	1	0	0
II bent	0	1	0	1
II kint	0	1	1	0
III bent	0	1	1	1
III kint	1	0	0	0
IV bent	1	0	0	1
IV kint	1	0	1	0

5.2 Első feladat

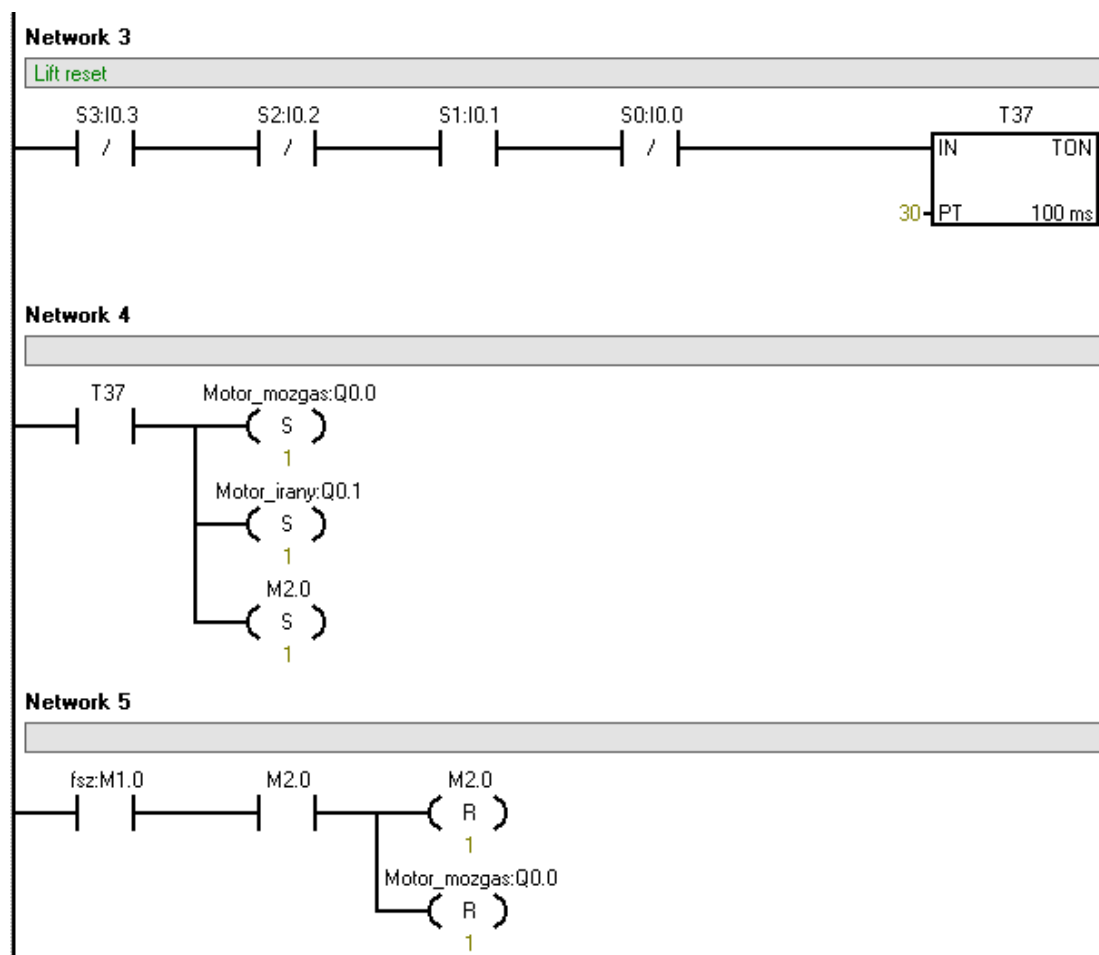
A liftkabin alaphelyzetben a földszinten van. A negyedik emeleti külső hívógomb megnyomásával a lift felmegy a negyedik emeletre, majd a földszinti külső hívógombbal visszahívhatjuk a földszintre. Minden mozgást csak akkor kezdhet meg a liftkabin, ha az már a végállásban megállt. A hívógombokat minimum 100 ms ideig nyomva kell tartani a megfelelő érzékeléshez. Az utat nem csak egyszer kell tudnia megtenni. A programba egy Reset funkciót kell beépíteni: ha a földszinti gombot 3 másodpercig nyomva tartjuk, a liftkabin minden ellenőrzés nélkül lejön.

A feladat megoldása és magyarázata során kezelendő funkciókról fogok írni. A példában a szintérzékelés bemeneteivel memóriabitekét aktiválok, így a működtető hálózatokban nem kell az összes szintérzékelő bemenetet megadni feltételként, elegendő a megfelelő memóriabitre hivatkozni. Erre ad példát a 11. ábra.



11. ábra: Első feladat szintérzékelés

Az ábrán látható, hogy a feladat megoldásához elegendő a földszint és a negyedik emelet szintjének használata, így két memóriabit használandó. A memóriák címe tetszőleges lehet, ebben az esetben M1.0 a földszint és M1.4 a negyedik emelet címzése. A hívógombok kezelésére is alkalmazható ez a módszer, de a példa kedvéért nem ilyen módon készítettem el a programot. Így a hívógombos feltételeknél a hívógomb kódolás szerinti összes feltételt meg kell adni minden hálózatban. Egy másik kezelendő funkció a Reset. Ennek az a jelentősége, hogy ha valami probléma lenne a megírt programban, akkor is le tudjuk hívni a liftkabint a földszintre. Ez az elsődleges kezelendő funkció és mindenképp ellenőrizni kell a program ezen részét feltöltés előtt. Ez az óraadó felelőssége, amennyiben a megfelelő ellenőrzést elmulasztja és hiba esetén nem hívható le a liftkabin, egy olyan program megírása válik szükségessé, mely a megfelelő kimenetek aktiválásával lehozza a kabint.



12. ábra: Első feladat Reset funkció

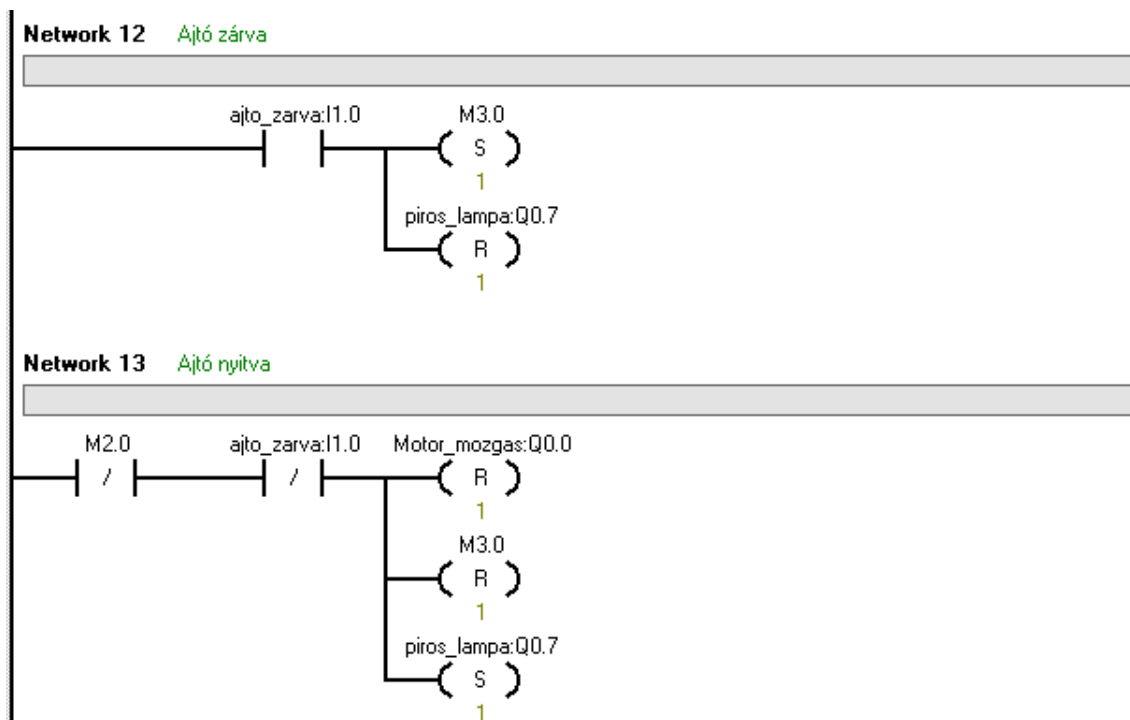
A programrészletben látható, hogy a hívógomb jele négy biten érkezik a vezérlőhöz, és mind a négy feltétel megadására szükség van. A külső hívógomb három másodpercig történő nyomva tartása aktiválja az időzítő kimenetét, így a negyedik hálózatban a liftkabin elindul a földszint felé, és beíródik az M2.0-ás memóriabit. A memóriabit 1-es állapota mellett amint a liftkabin a földszintre ér a motor leáll. A feladat megvalósításához már csak a liftkabin hívásának és megérkezésének kezelésére van szükség.

A liftkabin hívása nagyban hasonlít a 12. ábrán található harmadik és negyedik hálózathoz. A különbség csupán egy plusz feltétel az időzítő előtt, mivel a feladat kimondja, hogy csak abban az esetben indulhat el a liftkabin, ha végállásban van. Így az időzítő elé kerül valamelyik szintérzékelő memóriabit. Az időzítő lift hívása esetén nem három másodperces, hanem 100 ms-os, és a lejárt után is a megfelelő motorvezérlést kell kiadni annak függvényében, hogy a földszintre vagy a negyedik emeletre hívtuk a kabin. A megérkezésnél törölni kell a használt memóriabitet valamint a motorvezérlő kimenetet. A motorvezérlő irány törlése nem elengedhetetlen, mivel minden mozgatsnál a megfelelő állapotba írjuk.

5.3 Második feladat

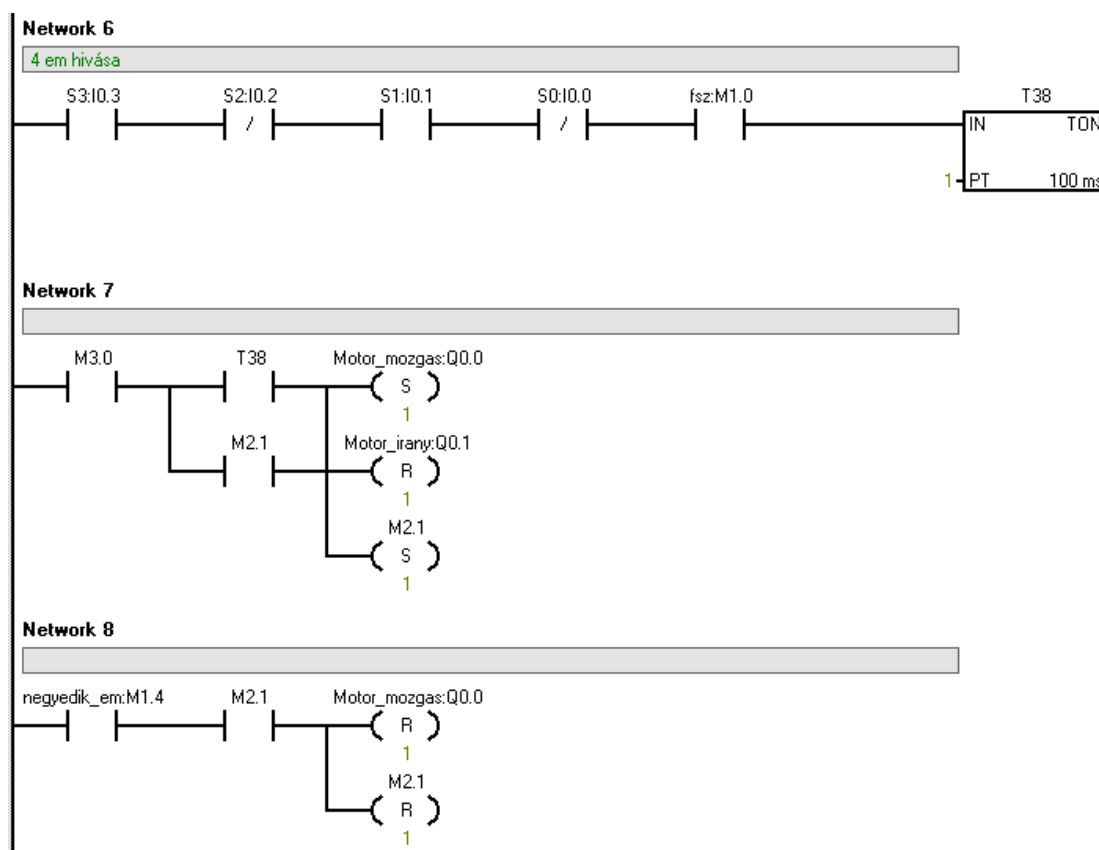
A modell működése alapján véve megegyezik az első feladattal. Hozzáadott funkció, hogy a liftkabin csak abban az esetben mozoghat, ha az ajtók biztonsági érzékelője jelez, hogy zárva vannak. Ha a kabin mozgása közben kinyílik egy ajtó, a liftkabinnak azonnal meg kell állnia, az ajtó bezárása után automatikusan folytatnia kell az útját. A nyitott ajtót a külső Piros LED-ekkel jelezni kell. A Reset funkció szükséges.

A programozónak le kell tudnia kezelni az ajtók nyitását és zárását. Fontos dolog, hogy a Reset funkció esetén az ajtók biztonsági érzékelőjének ellenőrzése kimaradjon, így akár nyitott ajtókkal is le kell jönnie a földszintre a liftkabinnak. A kabin hívásánál fontos különbség lesz az első feladathoz képest, hogy már nem használhatjuk csupán az adott időzítőt a motorok megindításánál, mivel a két végállás között is újra kell tudnia indulni.



13. ábra: Második feladat ajtók ellenőrzése

Az ábrán látható M2.0 memóriabit a Reset funkció tárolására szolgál. A 13-as hálózatot úgy alkottam meg, hogy csak abban az esetben állítsa le az ajtók nyitása a motor mozgását, ha éppen nem él a Reset funkció. A motor megállításával párhuzamosan a külső piros LED-eket felkapcsolja és az M3.0-ás bitet törli a látható feltétel kombináció. Az ajtók bezárásával a LED-ek kialszanak és az előbb említett memória beírásra kerül. Ez a memóriabit felel azért, hogy csak bezárt ajtóknál induljon meg a motor mozgása.



14. ábra: Második feladat vezérlés

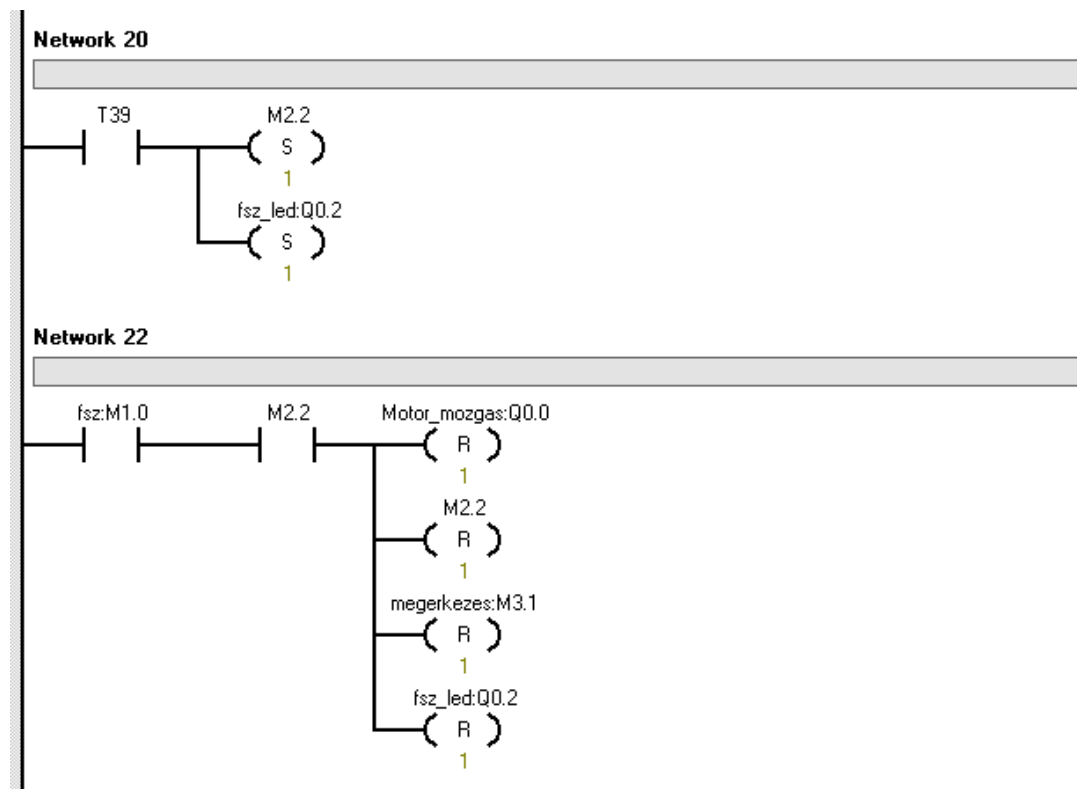
A 14. ábrán látható a negyedik emelet vezérlése. Ez magában foglalja a negyedik emelet hívásának kezelését, a motor megindítását az emelet felé és a szinten való megérkezést. A kabin hívása és a megérkezés ugyanolyan módon történik, mint az első feladatban. A különbség csupán annyi, hogy a motor megindításához alapfeltétel, hogy az ajtók zárva legyenek. Az ajtók bezárása beírta az M3.0-as memóriabitet, így a hetedik hálózatban arra hivatkozok. Használhattam volna az érzékelőből érkező jelet közvetlenül, de a példákon keresztül többféle megoldást szeretnék megmutatni.

A T38-as időzítő felel a 100 ms-os nyomva tartásért, így az idő letelte után beíródik a motor mozgása és iránya, valamint az M2.1-es memóriabit, mely a negyedik emelet hívását jelzi. A hetedik hálózatban a T38-as záró érintkezővel párhuzamosan be kellett raknom az M2.1-es bitet is feltételnek, mivel a motor elindítását abban az esetben is biztosítani kell, ha már nem a földszinten van a kabin, hanem valahol menet közben állt meg az ajtók kinyitása miatt. Ezzel a megoldással a program a feladatkiírásnak megfelelően működik.

5.4 Harmadik feladat

A modell működésének alapjául szolgál a második feladat. Minden az előbbi feladatokban megismert funkció kezelése szükséges. A működés további jellemzője, hogy a liftkabinnak csak abban az esetben szabad elindulnia, ha megtörtént az utasok ki- és beszállása a megfelelő szinten. Ez annyit tesz, hogy álló liftkabin esetén az ajtó egyszer kinyílt és bezárult. A másik kezelendő funkció a hívás nyugtázása. A liftkabin hívása után az adott szinten az előlapi LED-nek világítania kell, míg a kabin meg nem érkezik. A nyomógombok lenyomása változatlanul 100 ms és Reset funkció szükséges.

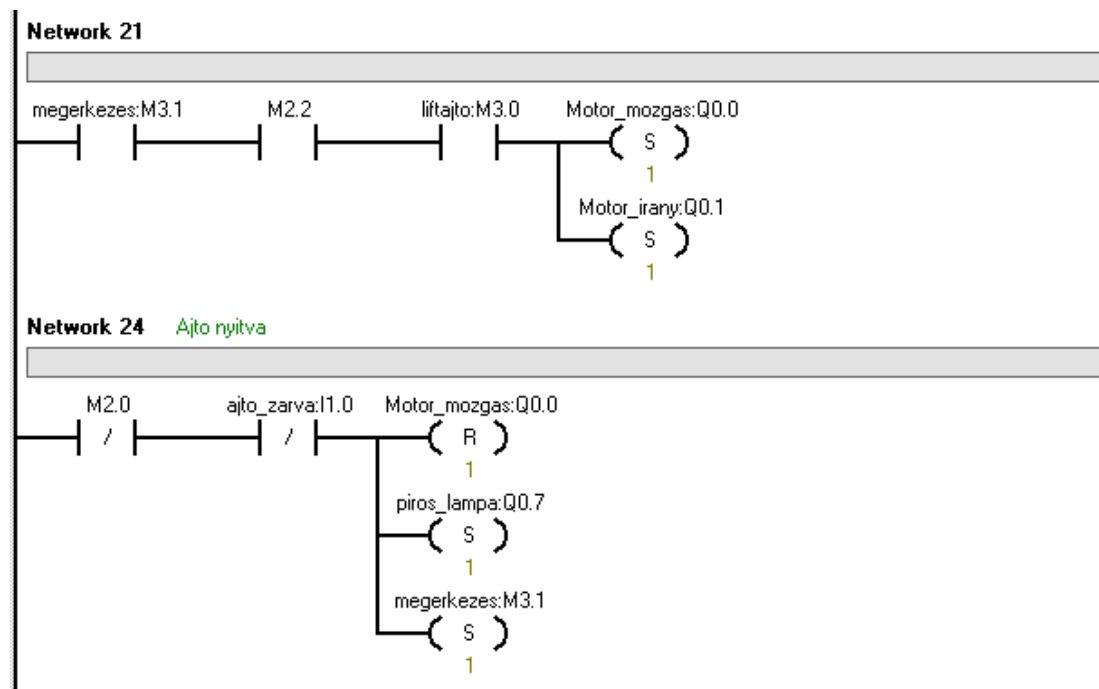
Amennyiben a programozó a feladatokkal sorrendben halad, akkor könnyen felismerheti, hogy nem kell nagy változtatást alkalmaznia a második feladaton. Az utasok ki- és beszállásának kezelése és a hívás nyugtázása is egyszerű feladat.



15. ábra: Harmadik feladat hívásnyugtázás

A 15. ábrán látható a hívásnyugtázás megfelelő kezelése. A 20-as hálózatban a földszinti nyomógomb időzítője a feltétel, és az nem csak az M2.2-es memóriabitet, hanem a Q0.2-es kimenetet is beírja. Az M2.2-es bit felel a földszintre vezérlésért, míg a Q0.2-es kimenet az előlapon található földszinti zöld LED-et vezérli. A 22-es hálózatban zajlik le a liftkabin megérkezésének kezelése. Megérkezéskor a földszinti

LED kimenete törlésre kerül. Ezzel párhuzamosan az M3.1-es, a kiszállásért felelős memóriabit is törlődik. Ez a gondolatmenet szükséges a negyedik emeleten történő hívásnyugtázásnál is.



16. ábra: Harmadik feladat utasok kiszállása

A 16. ábrán látható 24-es hálózatban beíródik az M3.1-es memória, ha jelenleg nem aktív a Reset funkció és az ajtók valamelyike ki van nyitva. Ez a bit jelzi, ha megtörtént a kiszállás. A 21-es hálózatban van megírva a liftkabin vezérlése a földszintre. Itt feltételként meg kell adni azt, hogy az utasok már kiszálltak. Fontos dolog, hogy az első elindulás előtt elfogadható, ha egyszer ki kell nyitni és be kell zárni az ajtót, hiszen ez nem egy sorrendi lefutó program, és feltételezzük a lift folyamatos működését. Ez elfogadható a további feladatok megvalósításánál is. Említésre méltó továbbá, hogy az ajtók érzékelői sorba vannak kapcsolva, így lényegtelen, hogy melyik ajtót nyitjuk vagy zárjuk.

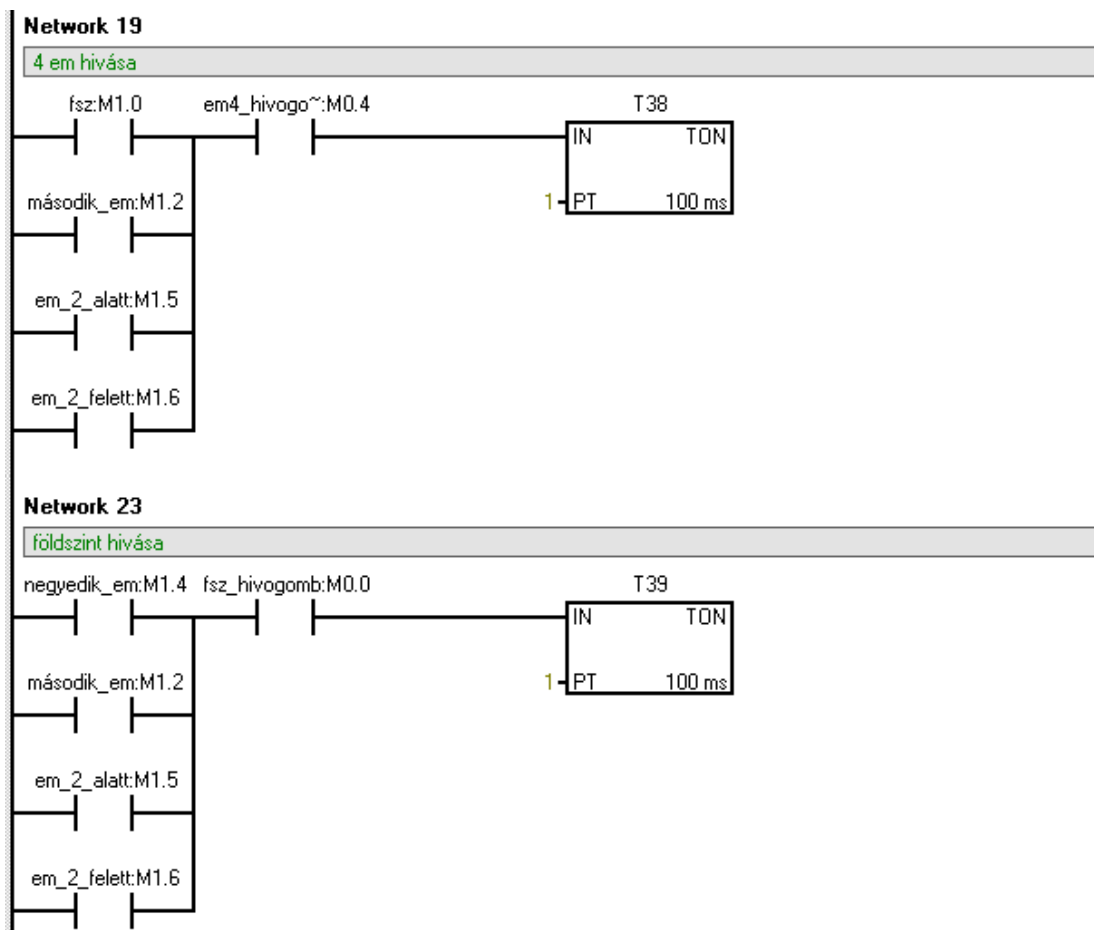
A szinteken való pontos megállás a liftkabin tehetetlensége miatt olyan bonyolult programot eredményezne, amely nem elvárható egy kezdő szintű programozótól. A pontos megállás kezeléséért felelős programrész figyelembe venné, hogy felfelé vagy lefelé közlekedik a kabin, valamint mennyi ideje volt felgyorsulni. Ez az ajtók bármelyik időpillanatban való kinyitása mellett szinte kivitelezhetetlen a

modellen. Így a ki- és beszállás megtörténe nem csak az egész szinteken lehetséges, tehát csak az ajtó nyitásától függ.

5.5 Negyedik feladat

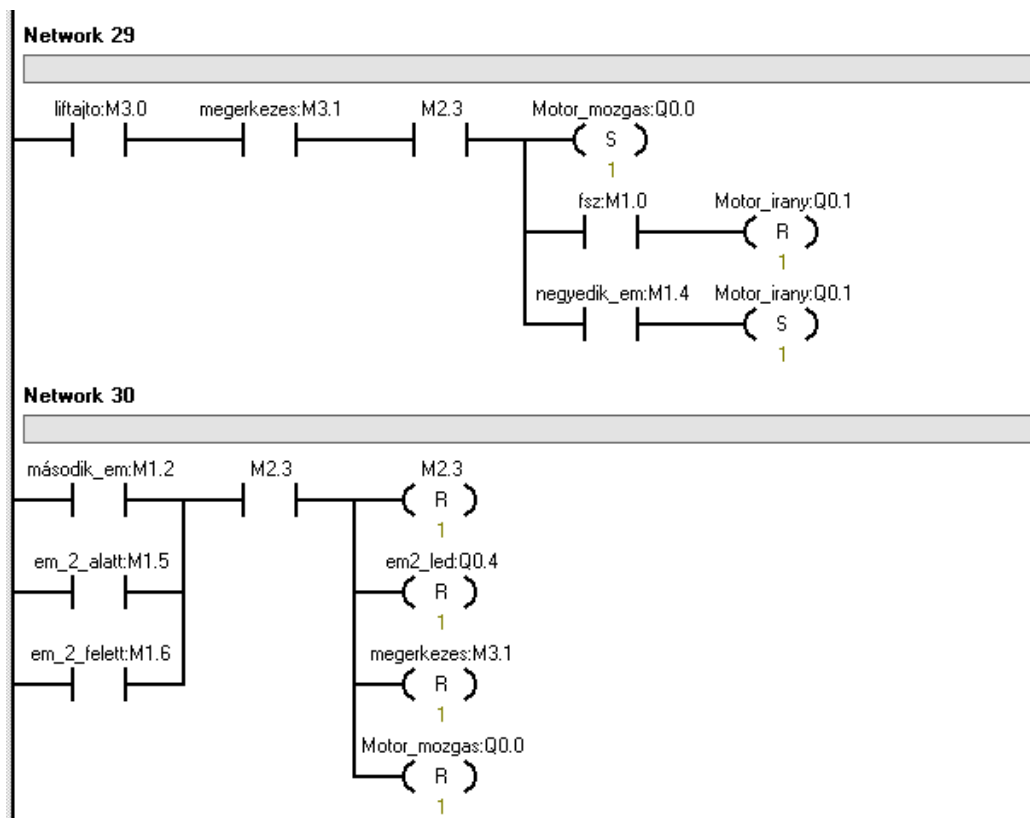
Az eddigi földszint és negyedik emelet mellett most a második emelet működtetése is szükséges. Csak külső hívógombokat használunk a vezérléshez, melyeket 100 ms ideig nyomva kell tartani. Az ajtó kinyitásával a kabin mozgása megáll, bezárásával újraindul. Reset funkció, a külső LED-ek megfelelő használata és az utasok ki- és beszállásának kezelése szükséges. A második emeleti megállást két szint között félúton már meg kell kezdeni a megfelelő pozíció eléréséhez.

A programozónak új szinteket kell tudnia érzékelni, valamint egy, az előzőktől eltérő vezérlést kell megalkotnia. A Reset funkció és a külső LED-ek vezérlése az előző megoldásokhoz hasonló módon történik. Ugyanez elmondható az ajtó nyitásáról és a beszállás kezeléséről. Nagyobb változtatás szükséges azonban a vezérlő hálózatokban. A második emelet irányítása nem azonos a földszinti vagy a negyedik emeleti vezérléssel, hiszen itt felülről, illetve alulról is érkezhet a kabin.



17. ábra: Negyedik feladat szintek hívása

A 17. ábrán látható, hogy a negyedik feladatban a földszint és a negyedik emelet hívása is megváltozott. Ennek a változásnak az az oka, hogy a liftkabin csak akkor indulhat el, ha valamelyik másik szinten várakozik. A 19-es és 23-as hálózatban megváltoztatásra kerültek a bemeneti feltételek a harmadik feladathoz képest. A földszint és negyedik emelet hívásának már nem csak az ellenkező szinten tartózkodás a feltétele, hanem a második szinten lévő is. Mivel a liftkabin pontos megállása nem megvalósítható egykönnyen, így a második emelet alatti és feletti szinteken való tartózkodás is egyenértékű magával az egész szinttel a program számára. Ezért tettem a földszint szintérzékelőjével párhuzamosan záró érintkezőket a második emeletnek megfelelő memóriabitek számára a 19-es hálózatban. Ugyanezen az elven haladva készült a 23-as hálózat is.



18. ábra: Negyedik feladat második emelet vezérlése

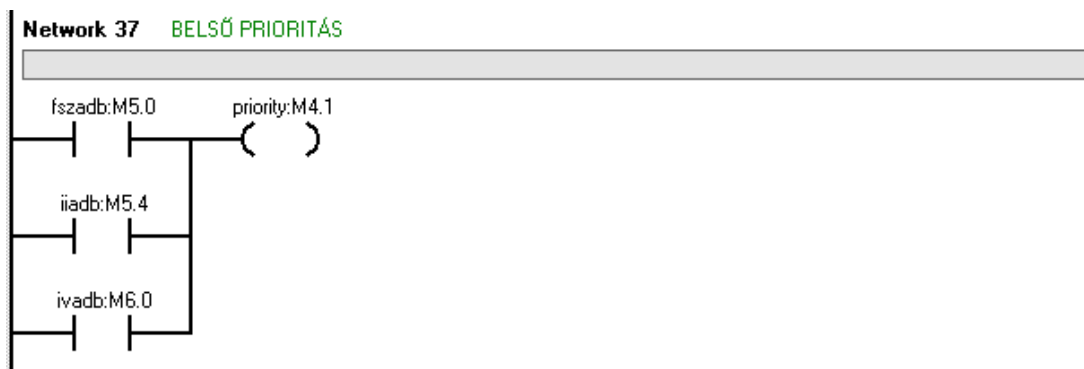
A 18. ábrán szemléltetem a második emelet vezérlésének egy részét. Az ábrán nem szerepel a nyomógomb megnyomása, és a késleltetett memóriabit beírása, ami a második emelet vezérléséért felelős (M2.3). A 29-es hálózatban a motor mozgásával párhuzamosan zajlik a motor irányának megadása. Az belátható, hogy a második emeletre felülről vagy alulról érkező csak a földszintről vagy a negyedik emeletről lehetséges. Így a motor mozgásának beírásával párhuzamosan van két feltétel. Ezek a szintérzékelésből adódnak, és ezek függvényében fog felfelé, illetve lefelé elindulni a liftkabin.

A 30-as hálózatban a második emeleti megérkezés látható. A liftkabin mozgásának megállítására már akkor ki kell adni a vezérlő jelet, ha az elérte a második emelet feletti vagy alatti szintet. Ez a megoldás elég pontos megállást eredményez ahhoz, hogy az könnyen kezelhető legyen. Megállás után biztosan tudjuk, hogy a kabin a második emeleten vagy közvetlenül alatta, illetve felette van. Azonban ez a megérkezés nem pontos eléggé ahhoz, hogy minden alkalommal a liftkabin pontosan a szint ajtaja mögött legyen. A belső nyomógombok megnyomása kissé kényelmetlen lehet, de a feladat egyszerűsége érdekében ez úgy gondolom vállalható. A megérkezés minden más tekintetben megegyezik az eddig ismerttetett módszerrel.

5.6 Ötödik feladat

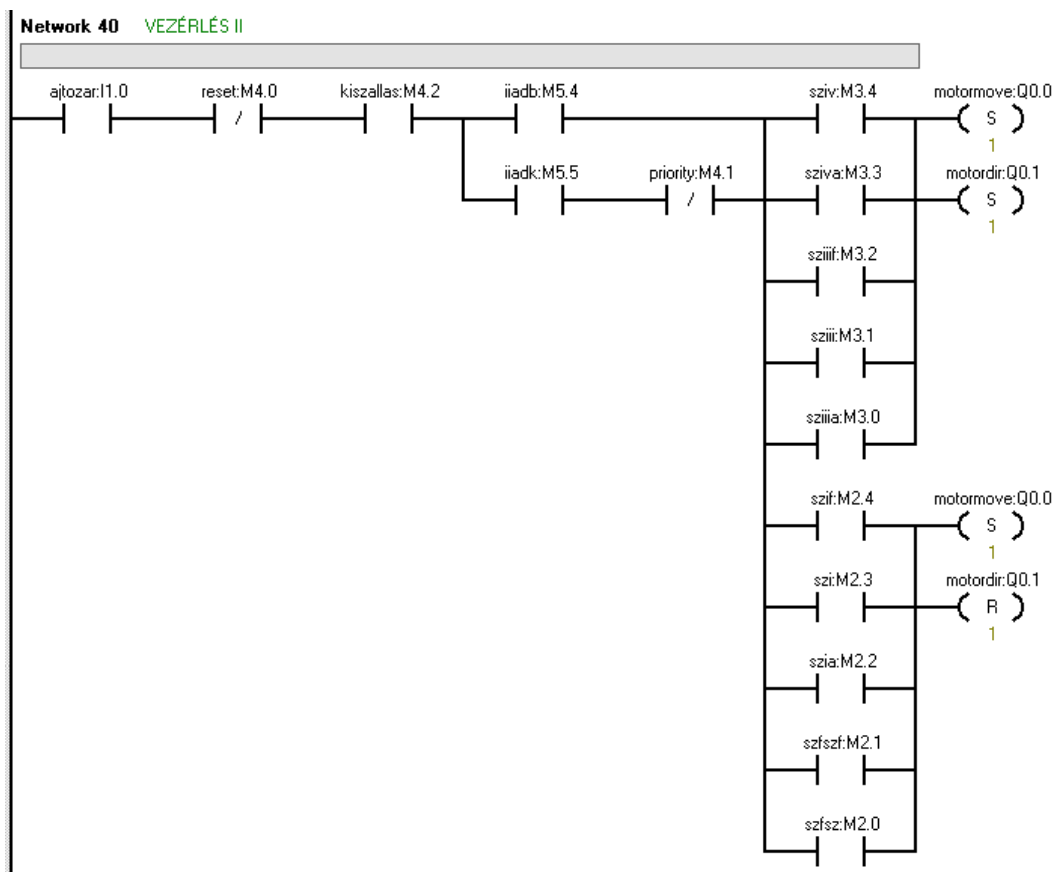
Az ötödik feladatban bevezetésre kerülnek a liftkabinban található nyomógombok. A használt szintek a földszint, második és negyedik emelet. A belső hívógombok élvezzenek elsőbbséget a külsőkkel szemben, és ezeknél is elvárt a 100 ms-os nyomva tartás. Ennek abban kell megnyilvánulnia, hogy ha egyszerre egy külső és egy belső hívás is történt, a liftkabinnak figyelmen kívül kell hagynia a külső kabinhívást, amíg a belső hívást nem teljesítette. A kabin nem indulhat el a belső nyomógomb által adott szintről, amíg a be- és kiszállás meg nem történt. A Reset funkció és a külső LED-ek megfelelő használata továbbra is szükséges. Aktív Reset állapot alatt is világítaniuk kell a külső piros LED-eknek.

A program megfelelő működéséhez a belső nyomógombok kezelése szükséges. Aktív belső hívás esetén célszerű egy memóriabitet aktív állapotba állítani, így a vezérlő hálózatokban hivatkozhatunk a belső gombok elsőbbségére. A feladat funkciói megegyeznek az előző feladatban ismertetettekkel, így a megvalósításuk is azonos.



19. ábra: Ötödik feladat belső gombok prioritása

A 19. ábrán látható az előbb említett belső elsőbbség érzékelése. Az M4.1-es memóriabit abban az esetben aktív, ha valamelyik szint a liftkabinban található nyomógomb megnyomásával volt hívva. A vezérlő hálózatokban erre a bitre hivatkozok, így csak abban az esetben megy a liftkabin a külső hívás szintjére, ha nincs belső elsőbbség aktívan.



20. ábra: Ötödik feladat második emelet vezérlése

A 20. ábrán látható a második emelet vezérlése. A hálózatban a program ellenőrzi az ajtók állapotát, a Reset funkciót, azt, hogy megvolt-e a kiszállás, azt, hogy külső vagy belső hívás történt és a liftkabin jelenlegi helyzetét. Ez utóbbi alapján indítja a vezérlés a motort a megfelelő irányba. A kabin megérkezésénél szintén fontos a belső hívógombok elsőbbsége. Csak abban az esetben szabad a kabinnak megállnia a második emeleten külső hívásra, ha más szinten nem történt belső hívás. Más tekintetben a liftkabin megérkezése megegyezik a negyedik feladatban bemutatott megoldással.

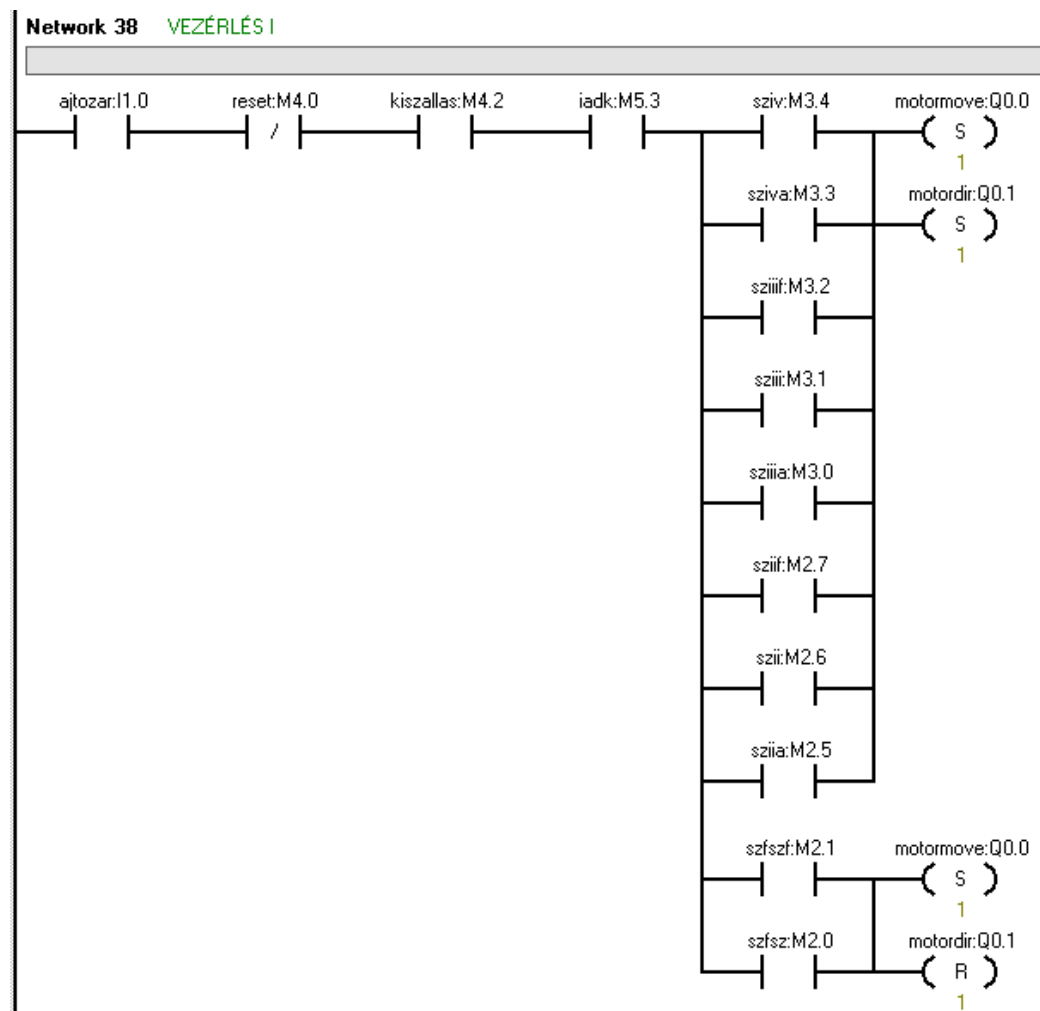
A külső piros LED-ek vezérlése is eltér a negyedik feladatban ismertetett módszertől. Ebben az esetben aktív Reset állapot mellett is világítaniuk kell. Ezt úgy oldottam meg, hogy a piros LED-eket vezérlő kimenet elé nem csak az ajtó állapota került feltételként, hanem azzal párhuzamosan a Reset funkció memóriabitje is.

5.7 Hatodik feladat

A hatodik feladatban bevezetésre kerül az összes szint. Csak a külső hívógombok kezelése szükséges 100 ms-os nyomva tartással. Az utasok be- és

kiszállásának ellenőrzése és a Reset funkció szükséges. A programnak akkor is megfelelően működnie kell, ha nem a földszinten kezd a kabin. A külső LED-ek vezérlése az ötödik feladattal azonos módon történjen. Nyitott ajtónál a liftkabin nem mozoghat, az ajtó bezárása után folytatnia kell az útját.

A hatodik feladat megoldása nem bonyolultabb az előzőnél, viszont hosszabb, így nagyobb a hiba esélye kezdő programozók esetében. Az első és harmadik emelet bevezetésével elérjük, hogy a kabin immár minden szintre hívható legyen. Így az összes külső hívógomb kezelése szükséges. További hozzáadott funkciót a feladat nem tartalmaz az ötödikhez képest, így a program nem bonyolódik. A bevezetett szintek vezérlése hasonlít a második emelet vezérléséhez. Egyedül abban van különbség, hogy a liftkabin jelenlegi helyzete másképp befolyásolja a motor vezérlésének irányát.



21. ábra: Hatodik feladat első emelet vezérlése

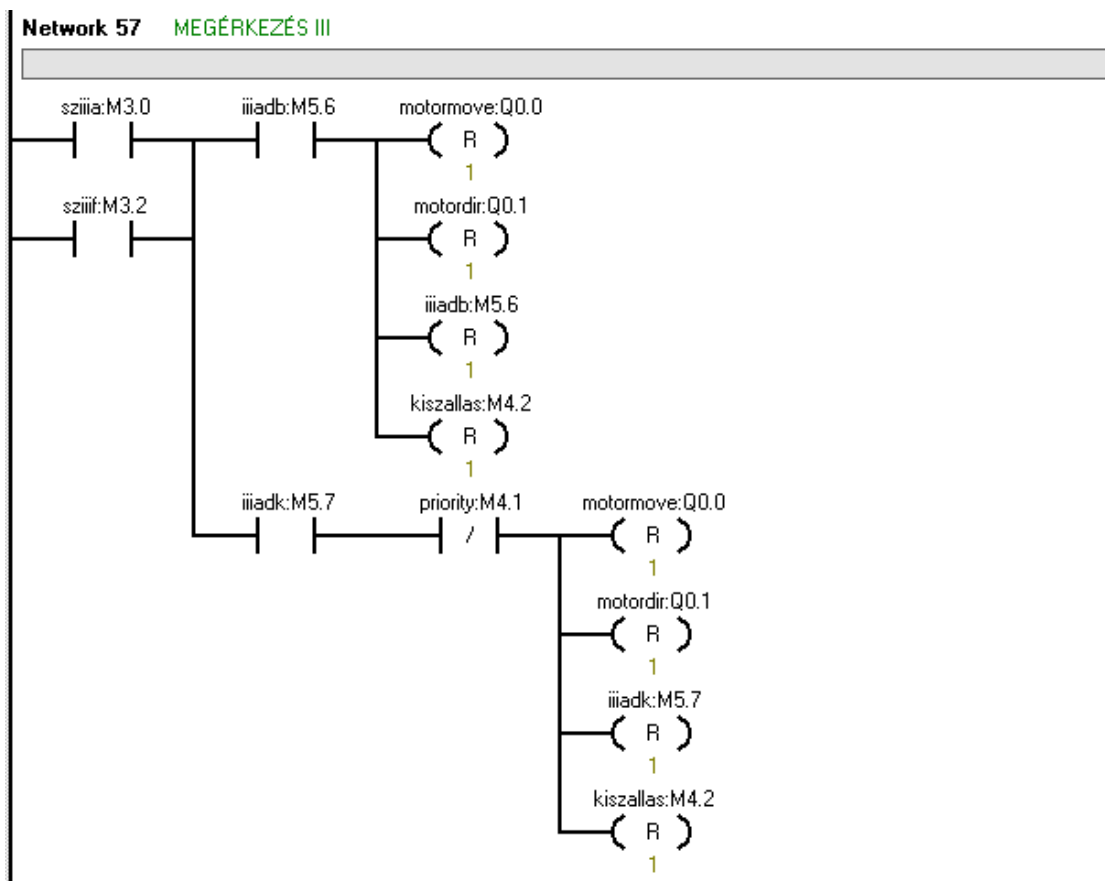
A 21. ábrán látható az első emelet vezérlése. A hálózatban a feltételek megadásának módja megegyezik a 20. ábrán láthatóval. Különbség azonban, hogy ebben az esetben nincs belső hívógomb, és mivel itt az első emelet vezérlése szerepel, így a szintek alapján beállított motorvezérlés is eltér.

5.8 Hetedik feladat

A hetedik feladatban az eddig tárgyalt összes funkció megfelelő programozása a cél. A liftkabinnak az összes szintre a megfelelő módon meg kell tudnia érkezni. A külső és belső hívógombok használata minden szinten az ötödik feladatban ismertetett módon szükséges. A programnak működnie kell, ha a kabin nem a földszinten van alaphelyzetben. Ajtók nyitásával és zárásával kapcsolatos elvárások az előző feladatokhoz hasonló módon. A Reset funkció, az utasok ki- és beszállása és a külső LED-ek működtetése szintén az előzőleg tárgyalt módszerrel kezelendők.

A hetedik feladat az ötödik és a hatodik feladat kombinációja. Az összes szint vezérlése szükséges külső és belső hívógombok használatával egyaránt. Ez kihívást jelenthet egy gyakorlottabb programozó számára is. A feladat megoldásában nem szerepel újabb funkció, így aki az előző feladatokat megoldotta, az egy kevés munkával ezt is könnyedén meg tudja oldani. Az általam elkészített program csak egy megoldás a sok lehetséges jó megoldás közül. Törekedtem egyszerű gondolatmenet szerint megoldani a feladatot, így nem a programkód rövidegére fektettem a hangsúlyt.

Az előző feladatokhoz képest főbb változás, hogy a belső hívógombok elsőbbsége immár minden szinthez meg van adva, azonban a memóriabit szerepe és felhasználása a programban változatlan. A vezérlő hálózatok logikája megegyezik az ötödik feladatban bemutatottal, különbség azonban, hogy itt minden szintet kezelni kell.



22. ábra: Hetedik feladat harmadik emelet megérkezése

A 22. ábrán látható a liftkabin megérkezésének kezelése a harmadik emeleten. Mint korábban említettem a megérkezésnél is törődni kell a belső hívógombok elsőbbségével. Az program 57-es hálózata állítja le a motor mozgását, amennyiben a kabin közvetlenül a harmadik emelet alatt vagy felett van. Ezt azonban csak abban az esetben teszi meg, ha belső hívógommbal küldtük oda, vagy más belső hívás nélkül, külső hívógommbal. Mindkét esetben csak a hívásnak megfelelő memóriabitet törli a hálózat.

6 Fejlesztési lehetőségek

Ebben a fejezetben a fejlesztési lehetőségek mellett néhány megjegyzést is teszek a lift-modell működésével kapcsolatban.

Nagy előrelépés lenne a programok bonyolítása terén, ha egy megfelelő számú ki- és bemenettel rendelkező PLC-vel vezérelnénk a lift-modellt. Újabb funkciókkal lehetne érdekesebbé tenni a feladatokat. Használatra kerülnének a liftkabinban lévő LED-ek, melyek a jelenlegi szintet jelezhetnék. Ezt nem lenne bonyolult programba foglalni, viszont sokkal látványosabb lenne a modell. A kabinban található három darab kétállású kapcsoló szerepe is bevezetésre kerülhetne. Ezek a Terhelt, Túlterhelt és Megtelt állapotokat jeleznék a vezérlőegység számára. Könnyen érdekesebbé tehetnek egy-egy példaprogramot ezek az állapotok, ha megfogalmazunk hozzájuk különböző viselkedést. Például túlterhelt állapotban villoghatnak a belső LED-ek és a kabin bezárt ajtókkal sem indulna el. Ez csak egy a sok érdekes lehetőség közül.

A példafeladatok további funkciója lehet akár az eddig is használt S7-200-as PLC-vel a több utas kezelése. A jelenlegi példaprogramok képtelenek egyszerre több belső hívógombbal adott parancsot kezelni. További példafeladat lehet egy olyan program, amely meghatározott számú belső parancsot tud tárolni és teljesíteni a megnyomások sorrendjében. Ugyanilyen módon érdekesebbé tehető a feladat, ha a külső hívógombokat egymás után több szinten megnyomhatjuk. A jelenlegi példafeladatok ezt sem tudják kezelni. A kabin mozoghat a hívások sorrendje alapján, vagy viselkedhet lefelé gyűjtő rendszerként. Ezek a funkciók bevezetése további kettő-három egyre bonyolultabb programot eredményezne, és egy lefelé gyűjtő vagy más módon optimalizált rendszer igazi kihívást jelenthet a hallgatóknak.

A liftkabin pontos megállítása késleltetést igényel. Korábban említettem, hogy a fentről érkező kabin megállításának kezdete a szint feletti fél szint elérésekor történik. Ezt fogalmaztam meg a feladatokban az egyszerűség kedvéért. Azonban ez nem eredményez pontos megállást, így a kabinban található gombok megnyomása kényelmetlen. Erre a problémára megoldást jelentene minden szinten a megállítási késleltetése. Tesztekre lenne szükség, hogy kikísérletezzük a pontos megállítási eredményező késleltetést, azonban ebből is lehet egy feladatot készíteni. Például a kabin feladata, hogy felmenjen a negyedik emeletre, majd lefelé álljon meg a másodikon egy

kisebbségi késleltetéssel. Az ajtót kinyitva a hallgató leellenőrzi a pontosságot, majd az ajtó bezárásával a liftkabin automatikusan újra elvégzi a mozgást egy nagyobb késleltetéssel. Ezt a folyamatot három-négy különböző értékkel tesztelhetjük, majd a megfelelő értéket kiválasztva megírhatjuk a programot.

Újabb funkció lehetne egy karbantartási állapot bevezetése. A nyomógombok megfelelő sorrendben történő megnyomása, vagy több nyomógomb egyidejű megnyomása hozná a lift-modellt ebbe az állapotba. Ebben az állapotban megfogalmazhatunk bizonyos viselkedést a modellnek. Például a kabin menjen a földszintre és villogjanak a külső piros LED-ek. A villogás periódusidejét és kitöltési tényezőjét is meghatározhatjuk. Közben a kabinnak semmilyen hívásra nem szabad reagálnia. A modellnek újabb nyomógomb kombinációval vissza kell állnia alapállapotba. Elképzelhető egy olyan feladat is, ahol megadott számú megtett útszakasz után hibát színlelne, vagy karbantartást kérne a lift-modell, és nem működhet, csak a karbantartási állapot után. Erre alkalmas lehet a belső kapcsolók egyike is.

Az általam elkészített programok, mint említettem, egy-egy megoldást jelentenek a feladatokra. Találkoztam más hallgatók által írt PLC programokkal (más eszközökre) és az enyémtől eltérő gondolkodásmódjuk miatt időbe telt, míg értelmeztem azokat. Senkitől nem elvárható, hogy egy bizonyos módon oldjon meg egy feladatot, azonban a segítség mellett lehetséges megoldásokat felkínálhatunk. Így szeretném egy másik megvalósítási ötletemet itt leírni. Az elképzelés az, hogy a vezérlő hálózatokban nem kell beírni a motor mozgásának tekerceit, hanem bizonyos memóriabiteket. Ezek mellett motorvezérlő hálózatokban az előbb említett memóriabitek valósítanak meg a motor mozgását. Így csak azt kell a vezérlésnél eldönteni, hogy a lefelé vagy a felfelé vezérlő hálózatot szeretnénk aktiválni. Ez nem egy bonyolult változtatás, de az előzőektől eltérő.

Szeretném itt megjegyezni, hogy a lift-modell építése során használt egyes alkatrészek mára kezdenek nem az elvárt módon működni. Ez leginkább a nyomógombokon érezhető. A megnyomás mértékétől függ, hogy adnak-e jelet. A nem eléggé megnyomott gomb nem ad jelet, valamint a túlzottan megnyomott sem. Ez nem minden nyomógombra igaz, azonban a közeljövőben ajánlatos lenne az összes nyomógomb lecserélése.

A feladatok megoldása során elfogadható, ha olyan megoldással készült el a program, hogy a kabin első elindulása előtt az ajtót ki kell nyitni, és bezárni. Ezt azért tartom elfogadhatónak, mert a lift-modell vezérlése során ez az üzembe helyezés állapot. A működése során további ehhez hasonló tevékenység nem fog történni, és ezután csak a folyamatos működés az elvárás.

A megírt programokat nem egy időben teszteltem, így a projektekben a beállított PLC típusa eltérhet. Ennek az az oka, hogy a projekt fájlokban a hozzájuk rendelt PLC típusa is el van tárolva, és ebben a legutóbbi mentésnél fennálló állapot van elmentve. Minden kipróbálás előtt a programszerkesztőben be kell olvasni, vagy be kell állítani a programozni kívánt PLC pontos típusát.

Az általam beillesztett ábrákon szereplő programrészletek nem tartalmazzák a változókhoz tartozó kommenteket. A kommentek megtekinthetők a projektekben, ahol pontosabb leírást kaphatunk a változókról. A programok projektfájlként és XPS-dokumentumként is megtekinthetők.

7 Összegzés

A szakdolgozatom témája egy, az oktatásba integrálandó lift-modellről szólt. A témát annak érdekében választottam, hogy az irányítástechnikai tantárgyak oktatását és tanulását érdekesebbé lehessen tenni. Fontos dolog, hogy a hallgatók kézzel fogható technológiákkal találkozzanak már az egyetemi éveik alatt is. Ezáltal megismerkedhetnek ipari problémákkal és megoldásokkal. A szakdolgozatom remélem jelentős változást hoz ezen tárgyak oktatásában. Bemutattam a modell fizikai felépítését és a vezérlésének lehetőségeit. Az ekkor fennálló hibáit kijavítottam. Ismertettem a modell egyes alkotóelemeit és az általam választott vezérlőegységet.

Megoldást találtam egy kevesebb ki- és bemenettel rendelkező PLC-vel való vezérlésre. Egy általam tervezett és megvalósított programmal futó segéd PLC rendszerbe kötésével lehetséges a lift-modell programozása S7-200-as CPU-val. A vezérlő PLC így részben közvetlen, részben közvetett módon irányítja a lift viselkedését. Példafeladatokat találtam ki az oktatás megkönnyítése érdekében. A feladatok kulcslépéseit ismertettem, a lépésről lépésre nehezedő programokat megvalósítottam és leteszteltem.

A programok többféle gondolkodásmód használatával készültek, így változatosak a megoldások. Az általam készített programokkal ellenőrizhetők a modell funkciói, segítséget nyújthatnak a feladatok megoldásában. A teljes feladatkiírás és az összes program projektfájlja és XPS-dokumentuma a mellékelt CD-n megtalálható. Javítási javaslatokat tettem a rendszer fejlesztéséhez. További funkciók és példafeladatok megalkotására tettem részletes javaslatokat. Leírtam a nem használt ki- és bemeneteket és az ezekkel továbbfejleszthető rendszer lehetőségeit. A kitűzött célokat elértem, a modell működőképes állapotban van, és remélhetőleg érdekesebbé fogja tenni az irányítástechnikai gyakorlati órákat mind az oktatók, mind a hallgatók számára.

8 Summary

The topic of my thesis is about an elevator-model that will be integrated into the education system. I chose this topic in order to make the teaching and learning of the control technologies subjects more interesting. It is important for the students to meet real industrial technologies during their university studies. This way they can get to know industrial problems and solutions. I hope my thesis will bring change to the teaching of these subjects. I presented the physical construction of the model and the possibilities to control it. I repaired the flaws of it. I introduced the parts of the model and the control unit that I chose.

I found a solution for the controlling of the model with a PLC with fewer inputs and outputs. It is possible to control the system with an S7-200 CPU with the help of a secondary PLC with a program that I planned and created. The primary PLC controls the model partly in an indirect way. I created example exercises in order to make the teaching easier. I presented the key steps to solve the exercises, I solved and tested the examples.

The program codes were made with different thinking methods, so the solutions are various. With the codes I wrote the functions of the elevator-model can be tested, and they can help in the solving of the examples. All of the examples with solutions in project files and XPS documents can be found in the CD attachment. I made suggestions to improve the system. I made detailed suggestions and ideas about additional functions and examples. I presented the unused inputs and outputs and the possibilities of an improved system with them. I reached my goals, the elevator-model is ready to use and I hope it will make the control technologies subjects more interesting for the students and the tutors.

9 Irodalomjegyzék

- [1] Charles Platt, Encyclopedia of Electronic Components Volume 1, O'Reilly Media, 2012
- [2] Engineering Essentials, <http://machinedesign.com/engineering-essentials/engineering-essentials-what-programmable-logic-controller>
(2016.október.02)
- [3] SIEMENS SIMATIC S7-200 Programozható vezérlőrendszer Kézikönyv,
Kiadás 2004/06, A5E00307987-01

10 Ábrajegyzék

1. ábra: Lift-modell	7
2. ábra: Omron MY4 24 VDC relé	8
3. ábra: Relék kapcsolása.....	9
4. ábra: Kódléc illusztráció	11
5. ábra: DB50-es csatlakozó	14
6. ábra: PLC letapogatási ciklus	17
7. ábra: CPU 224XP.....	19
8. ábra: Időzítő példaprogram	21
9. ábra: Bemenetek tesztelése (részlet)	22
10. ábra: Segéd PLC program (részlet).....	25
11. ábra: Első feladat szintérzékelés	30
12. ábra: Első feladat Reset funkció	31
13. ábra: Második feladat ajtók ellenőrzése.....	33
14. ábra: Második feladat vezérlés	34
15. ábra: Harmadik feladat hívásnyugtázás	35
16. ábra: Harmadik feladat utasok kiszállása.....	36
17. ábra: Negyedik feladat szintek hívása	38
18. ábra: Negyedik feladat második emelet vezérlése	39
19. ábra: Ötödik feladat belső gombok prioritása.....	40
20. ábra: Ötödik feladat második emelet vezérlése	41
21. ábra: Hatodik feladat első emelet vezérlése.....	42
22. ábra: Hetedik feladat harmadik emelet megérkezése	44

11 Táblázatjegyzék

1. táblázat: Szintérzékelés.....	11
2. táblázat: Be- és kimenetek 1	13
3. táblázat: Be- és kimenetek 2	14
4. táblázat: DB37-es csatlakozó.....	23
5. táblázat: Segéd PLC be- és kimenetek.....	25
6. táblázat: Feladat bemenetek.....	28
7. táblázat: Feladat kimenetek	28
8. táblázat: Feladat szintérzékelés.....	29
9. táblázat: Feladat hívógombok.....	29