

Sekvenční řízení modelu výtahu programovatelným automatem

Sequence control of elevator model by programmable logic
controller

Bc. Ondřej Vrzal

Diplomová práce
2010



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Ondřej VRZAL**
Studijní program: **N 3902 Inženýrská informatika**
Studijní obor: **Informační technologie**

Téma práce: **Sekvenční řízení modelu výtahu programovatelným
automatem**

Zásady pro vypracování:

1. Vypracujte literární rešerši na téma Sekvenční řízení se zaměřením na řízení a ovládání výtahu.
2. Proveďte návrh modelu výtahu umožňující uživateli řídit dva výtahy pohybující se mezi dvěma patry. Model zakomponujte do pracoviště obsahující i PC k programování.
3. Návrh modelu i celého pracoviště realizujte.
4. Vytvořte vzorové úlohy pro řízení vytvořeného modelu.
5. Realizujte vizualizaci řízeného objektu ve vybraném SCADA/HMI systému.

Rozsah práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **Martinásková, M., Šmejkal, L.: Řízení programovatelnými automaty, Vydavatelství ČVUT, Praha, 1998.**
2. **Martinásková, M., Šmejkal, L.: Řízení programovatelnými automaty II, Vydavatelství ČVUT, Praha, 2000.**
3. **Martinásková, M., Šmejkal, L.: Řízení programovatelnými automaty III, Vydavatelství ČVUT, Praha, 2003.**
4. **Šmejkal, L., Martinásková, M.: PLC a automatizace, Nakladatelství BEN – technická literatura, Praha, 1999.**
5. **Firemní literatura k vybranému SCADA/HMI software.**
6. **Firemní literatura k PLC Tecomat.**

Vedoucí diplomové práce:

Ing. Tomáš Sysala, Ph.D.

Ústav automatizace a řídicí techniky

Datum zadání diplomové práce:

19. února 2010

Termín odevzdání diplomové práce:

8. června 2010

Ve Zlíně dne 19. února 2010


prof. Ing. Vladimír Vašek, CSc.
děkan




prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

ABSTRAKT

Tématem této práce je návrh a následná konstrukce modelu výtahu pro potřeby výuky programovatelných automatů a také návrh programu dle normy IEC 61 131-3 pro ovládání tohoto modelu. Práce ve své první části obsahuje literární rešerši na téma sekvenční řízení se zaměřením na ovládání výtahu. Druhá část pak popisuje model a způsob jeho provedení a dále pak samotný program pro ovládání. Model je řízen programovatelným automatem řady Foxtrot od české firmy Teco a.s.

Klíčová slova: *PLC, Foxtrot, sekvenční řízení, model výtahu*

ABSTRACT

The main theme of Master thesis is design and construction of elevator model which will be mainly used for teaching of programmable logic controllers programming and also creation of program for control of the model which will be compatible with IEC 61 131-3. The theoretical part deals with the sequential control focused on elevator control problem. Next part of thesis describes the elevator model and way of its implementation and also program for control of model. Control process is carried out using programmable logic controller from Czech joint-stock company Teco.

Keywords: PLC, Foxtrot, sequential control, elevator model

Děkuji vedoucímu mé diplomové práce Ing. Tomáši Sysalovi, Ph.D. z Ústavu automatizace a řídicí techniky a panu Mgr. Milanovi Adámkovi Ph.D. z Ústavu bezpečnostního inženýrství za odbornou pomoc. Dále děkuji své rodině za poskytnutou podporu.

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 PROGRAMOVATELNÉ AUTOMATY OBECNĚ	11
1.1 HISTORIE PLC.....	11
1.2 SOUČASNOST PLC.....	12
1.3 KONSTRUKCE PLC.....	12
1.4 PRVKY PROGRAMOVATELNÉHO AUTOMATU.....	13
1.4.1 Centrální výpočetní jednotka.....	13
1.4.2 Uživatelská a systémová paměť.....	14
1.4.3 Binární vstupně výstupní jednotky.....	14
1.4.4 Analogové vstupně výstupní jednotky.....	14
1.4.5 Čítače.....	15
1.4.6 Komunikační jednotky.....	15
2 PLC FOXTROT	16
2.1 KOMUNIKACE.....	16
2.2 ZÁKLADNÍ PARAMETRY PLC FOXTROT.....	18
2.3 CENTRÁLNÍ JEDNOTKA.....	19
2.4 PLC FOXTROT PRO ŘÍZENÍ VÝTAHU.....	20
2.4.1 Modul binárních vstupů IB-1301.....	21
2.4.2 Modul binárních výstupů OS-1401.....	22
2.4.3 Analogové moduly IT-1601 a IT-1602.....	23
3 SEKVENČNÍ ŘÍZENÍ	25
3.1 POJEM SEKVENČNÍ LOGICKÉ FUNKCE.....	25
3.2 SEKVENČNÍ ŘÍZENÍ VÝTAHU.....	26
3.3 PROGRAMOVÁ REALIZACE.....	27
4 NORMA IEC 61 131	28
4.1 ČSN EN 61 131-3 PROGRAMOVATELNÉ ŘÍDICÍ JEDNOTKY - ČÁST 3: PROGRAMOVACÍ JAZYKY.....	28
4.1.1 Základní stavební bloky programu.....	29
4.1.2 Datové typy.....	30
4.1.3 Proměnné.....	31
4.1.4 Programovací jazyky.....	31
4.1.4.1 Jazyk seznamu instrukcí IL.....	32
4.1.4.2 Jazyk strukturovaného textu ST.....	32
4.1.4.3 Společné prvky grafických jazyků.....	34
4.1.4.4 Jazyk kontaktních schémat LD.....	36
4.1.4.5 Jazyk funkčního blokového schématu FBD.....	37
II PRAKTICKÁ ČÁST	38
5 NÁVRH MODELU	39
5.1 POŽADAVKY.....	39
5.2 VIZUALIZACE.....	39
6 REALIZACE MODELU	42

6.1	KONSTRUKCE MODELU	42
6.1.1	Stěny výtahu	42
6.1.2	Kovové konstrukce	43
6.1.3	Strop výtahu	44
6.1.4	Podložka výtahu	44
6.2	MOTORY	45
6.2.1	Motor pro zvedání kabiny	45
6.2.2	Motory pro otevírání a zavírání dveří	46
6.2.3	Ovládání motorů	46
6.2.4	Mechanické převody	48
6.3	DVEŘE	50
6.3.1	Konstrukce dveří	50
6.3.2	Logický obvod pro ovládání dveří	51
6.4	ČIDLA POLOHY	52
6.5	PRACOVNÍ MÍSTO	52
6.5.1	Uložení PLC	53
7	DOKUMENTACE MODELU	54
7.1	ADRESY VSTUPŮ A VÝSTUPŮ	54
7.2	OVLÁDÁNÍ MOTORŮ	55
7.3	ÚDRŽBA A PORUCHY MODELU	56
7.3.1	Převodní mechanismus kabin	56
7.3.2	Dveře	56
7.3.3	Kabina výtahu	57
7.3.4	Elektrické obvody	57
7.4	SCHÉMATA ELEKTRICKÝCH OBVODŮ	58
7.4.1	Zdroj napětí	58
7.4.2	Logický obvod pro ovládání dveří	58
7.4.3	H-můstek pro ovládání dveří	59
7.4.4	H-můstek pro motory kabiny	60
8	PROGRAMOVÁ ČÁST	61
8.1	UKÁZKOVÝ PROGRAM	61
8.1.1	Funkční blok dveří	61
8.1.2	Funkční blok motoru	62
8.1.3	Funkční blok kabiny	64
8.1.4	Funkční blok tlačítka	66
8.1.5	Funkce pro zařazení a vyřazení požadavků	67
8.1.6	Funkce podsvícení tlačítek	68
8.1.7	Ostatní funkce	68
8.2	VIZUALIZACE	69
	ZÁVĚR	70
	ZÁVĚR V ANGLIČTINĚ	71
	SEZNAM POUŽITÉ LITERATURY	72
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	73
	SEZNAM OBRÁZKŮ	74
	SEZNAM TABULEK	76
	SEZNAM PŘÍLOH	77

ÚVOD

Programovatelné automaty jsou dnes nedílnou součástí každodenního života člověka, a přesto si to málokdo uvědomuje. Příkladem budiž pražské metro, jehož bezpečný provoz a především bezpečnost cestujících, řídí právě programovatelný automat. A důvod proč si každý cestující neuvědomuje, co má na starosti jejich bezpečnou cestu je fakt, že programovatelné automaty nejsou vidět. Jsou totiž malé. Samozřejmě že jsou schované v rozvodných skříních, ale to nemění nic na faktu, že programovatelný automat je malé zařízení, které udělá obrovskou práci.

Je patrné, že programovatelné automaty už dávno nejsou pouze řídicí nástroje ve výrobním procesu, kde bychom je čekali nejčastěji, ale že se z průmyslové výroby přesunuli také do sféry řízení nevýrobních technologických procesů. Nutnost nasazení programovatelných automatů mimo průmysl si vyžaduje především technologický pokrok a stále větší a větší požadavky, které kladou důraz na komfort lidí. K tomuto faktu přispívá i skutečnost, že se programovatelné automaty neustále vyvíjejí a drží krok s dobou. První PLC totiž zpracovávaly pouze dvojkovou logiku řízení. Později s rozvojem polovodičových součástek se rozšířila použitelnost také na zpracování analogových signálů, matematických funkcí až po možnost realizace složitých systémů řízení obsahující zpracování binárních signálů, analogových hodnot, komunikaci s jinými systémy, přenos dat, archivaci naměřených hodnot, vlastní diagnostiku, tiskové výstupy atd.

Nasazení programovatelných automatů v řešení daného problému však v neposlední a mnohdy bohužel v první řadě, ovlivňuje finanční stránka věci. A to platí zejména v době hospodářské krize. Kvalitní a ucelený systém řízení pomocí PLC je nákladný, ale dobrý investor tuto možnost určitě pečlivě zváží a dojde k závěru, že správně navržený systém programovatelných automatů je dobrá návratná investice. Správně navržený systém s PLC je pak práce projektanta a programátorů.

Cíle této práce jsou proto zaměřeny na výchovu takových programátorů a spočívají v návrhu modelu, který se co nejvíce podobá situacím řešeným ve skutečnosti a který by těmto potenciálním programátorům PLC kladly stejné překážky.

I. TEORETICKÁ ČÁST

1 PROGRAMOVATELNÉ AUTOMATY OBECNĚ

1.1 Historie PLC

Myšlenka použití počítačů v přímém řízení je jen o málo mladší než samy počítače. Pokusy o konstrukci počítačů použitelných v automatizaci, a tedy vyhovujících požadavkům na činnost v reálném čase, se datují již od konce 50. let minulého století. Stále rostoucí výkonnost a spolehlivost počítačů při současně klesající ceně a požadavcích na provozní podmínky vedly na začátku 70. let k situaci, kdy bylo možné reálně uvažovat o efektivním a masovém uplatnění počítačů v automatizaci. V té době bylo také projektováno mnoho automatizovaných systémů vybavených počítači. Stále však šlo o etapu pokusů a ověřování. Podle výsledků statistických šetření – provedených tehdy ústavem INORGA – bylo v oboru těžkého strojírenství a hutnictví na celém světě zhruba 60 % projektů počítačově automatizovaných systémů neúspěšných. Od této doby však počet aplikací i počet úspěšných projektů a dokončených děl plynule rostl. V cestě širšímu uplatnění počítačů v přímém řízení stála relativně velká cena počítačových systémů. Bylo co zlepšovat i ve spolehlivosti, výkonnosti a dalších parametrech důležitých pro aplikace. Všechny tyto problémy v jisté míře přetrvávají dodnes, ale již nejsou hlavní překážkou aplikací. Hospodárnost ale byla v 70. letech dvacátého století důvodem, který vedl ke konstrukci specializovaných počítačových systémů, jež se i v tehdejších podmínkách již dokázaly efektivně uplatnit v automatizaci v průmyslu. [9]

Zmíněné specializované počítače našly uplatnění ve specifické oblasti automatizace – v ovládacích obvodech automatizovaných strojů a zařízení založených na řízení typu anone. Univerzální počítačové systémy byly příliš drahé a pro daný úkol zbytečně složité. To platilo pro technické i pro programové vybavení. V této situaci se na trhu objevily specializované logické procesory a specializované programové vybavení orientované pouze na realizaci ovládacích funkcí. Takové počítačové systémy byly tehdy vyvíjeny a některé z nich i použity v celkem úspěšných projektech i v našem státě – např. počítač PPC4, vyvinutý v ČKD. Postupem času však výrazně poklesla cena stále zdokonalovaných univerzálních procesorů, které postupně nahradily jednoúčelové přístroje dosud používané ve speciálních aplikacích. V průběhu 80. let dosáhl vývoj úroveň srovnatelné s dnešním stavem. V té době se také pro programovatelné automaty vybavené počítači ustálilo označení PLC. [1]

1.2 Současnost PLC

PLC jsou v současné době nejrozšířenějším typem řídicích systémů v průmyslu v mnoha aplikačních oblastech. Uplatňují se především v řízení dopravy nebo v oboru techniky budov. Moderních PLC se od prvních přístrojů tohoto druhu odlišují v mnoha směrech.

1.3 Konstrukce PLC

Z hlediska konstrukce lze programovatelné automaty dělit do skupiny kompaktních systémů a modulárních systémů.

Kompaktní systém (Obr. 1) obsahuje v jednom celku centrální procesorovou jednotku, digitální a analogové vstupně výstupní obvody. Rozšiřitelnost takových systémů je ale omezena.



Obr. 1. Kompaktní PLC [5]

Modulární systém (Obr. 2) je takový systém, kde jsou jednotlivé komponenty celku rozděleny do modulu. Modulární systém lze jednoduše rozšiřovat a to v nepoměrně větším rozsahu než u kompaktních systémů.



Obr. 2. Modulární PLC [5]

1.4 Prvky programovatelného automatu

Typické PLC z hlediska vnitřního uspořádání obsahuje centrální výpočetní jednotku, systémovou paměť, uživatelskou paměť, binární vstupy/výstupy, analogové vstupy/výstupy a několik dalších modulů jako čítače, komunikační moduly, speciální moduly a v neposlední řadě záložní paměťový modul. [2]

1.4.1 Centrální výpočetní jednotka

Centrální procesorová jednotka je jádrem celého programovatelného automatu a určuje jeho výkonnost. Bývá jednoprocesorová ale i víceprocesorová. U víceprocesorových systémů bývají někdy použity matematické koprocesory, vstupně výstupní procesory a někdy též komunikační procesory.

Důležitým charakteristickým parametrem centrální výpočetní jednotky je operační rychlost posuzovaná podle tzv. doby cyklu, což je doba zpracování 1000 logických instrukcí. Podle typu jednotky se pohybuje řádově od desítek milisekund až k desetínám milisekund u nejrychlejších jednotek [2].

1.4.2 Uživatelská a systémová paměť

Do uživatelské paměti se ukládá uživatelský program. Tato paměť bývá u starších modelů typu EPROM nebo EEPROM. Novější modely využívají již paměť FLASH. Paměti mají obvykle kapacitu řádově od desítek kB až po jednotky MB. A opět, novější modely již mají sloty pro paměťové karty.

V paměti dat, která musí být typu RAM, jsou umístěny uživateli dostupné uživatelské registry, zápisníkové registry, čítače, časovače a většinou i vyrovnávací registry pro obrazy vstupů a výstupů. Počet těchto registrů výrazně ovlivňuje možnosti programovatelného automatu. Adresovatelný prostor vymezený pro vstupy/výstupy omezuje počet připojitelných periferních jednotek. Důležitým parametrem jsou i rozsahy čítačů a časovačů. Většina modulárních automatů má dostupné i hodiny reálného času, případně i s kalendářem, které umožňují zahrnout do uživatelského programu úlohy využívající absolutní čas [2].

1.4.3 Binární vstupně výstupní jednotky

Binární vstupní jednotky slouží k připojování prvků pro tvorbu vstupů s dvouhodnotovým charakterem výstupního signálu, což mohou být např. tlačítka, přepínače, senzory doteku nebo přiblížení, dvouhodnotové senzory tlaku, hladiny, teploty apod.

K binárním výstupním jednotkám se obvykle připojují nejrůznější akční členy s dvouhodnotovým charakterem vstupního signálu. Mohou to být např. různá optická i akustická signalizační zařízení nebo cívky relé, stykačů, solenoidových ventilů, elektromagneticky ovládaných pneumatických či hydraulických rozvaděčů apod. [2]

1.4.4 Analogové vstupně výstupní jednotky

Analogové vstupní a výstupní jednotky zprostředkovávají kontakt PLC se spojitým prostředím. K analogovým vstupům lze připojit například snímače teploty, vlhkosti, tlaku, síly, hladiny, rychlosti, ale i většinu inteligentních přístrojů s analogovými výstupy. Důležitou součástí analogové vstupní jednotky je A/D převodník, který převádí analogové napěťové nebo proudové signály na číselné hodnoty.

Analogové výstupní jednotky slouží pro ovládání různých akčních členů se spojitým charakterem vstupního signálu, jako např. spojitě servopohony, frekvenční měniče, ale

třeba i ručkové měřicí přístroje apod. Analogová výstupní jednotka také obsahuje převodník, který převádí binární hodnoty na spojitý signál [2].

1.4.5 Čítače

Čítačové jednotky jsou určeny k čítání pulsů, jejichž perioda je srovnatelná nebo kratší, než je smyčka programu programovatelného automatu. Bývají k dispozici v provedení pro připojení univerzálních signálů, inkrementálních snímačů nebo absolutních snímačů. Všechny programovatelné automaty jsou také vybaveny softwarovými čítači, které se s výhodou používají v situacích, kdy není za potřebí použití čítačových jednotek [2].

1.4.6 Komunikační jednotky

Důležitou vlastností PLC systémů je schopnost komunikovat se vzdálenými moduly vstupů a výstupů, s podsystémy, se souřadnými i nadřizenými systémy, s operátorským panelem a s jinými inteligentními přístroji, s počítači a jejich sítěmi a tak vytvářet distribuované systémy. V drtivé většině používají moderní PLC ethernetové rozhraní pro komunikaci s vlastním okolím.

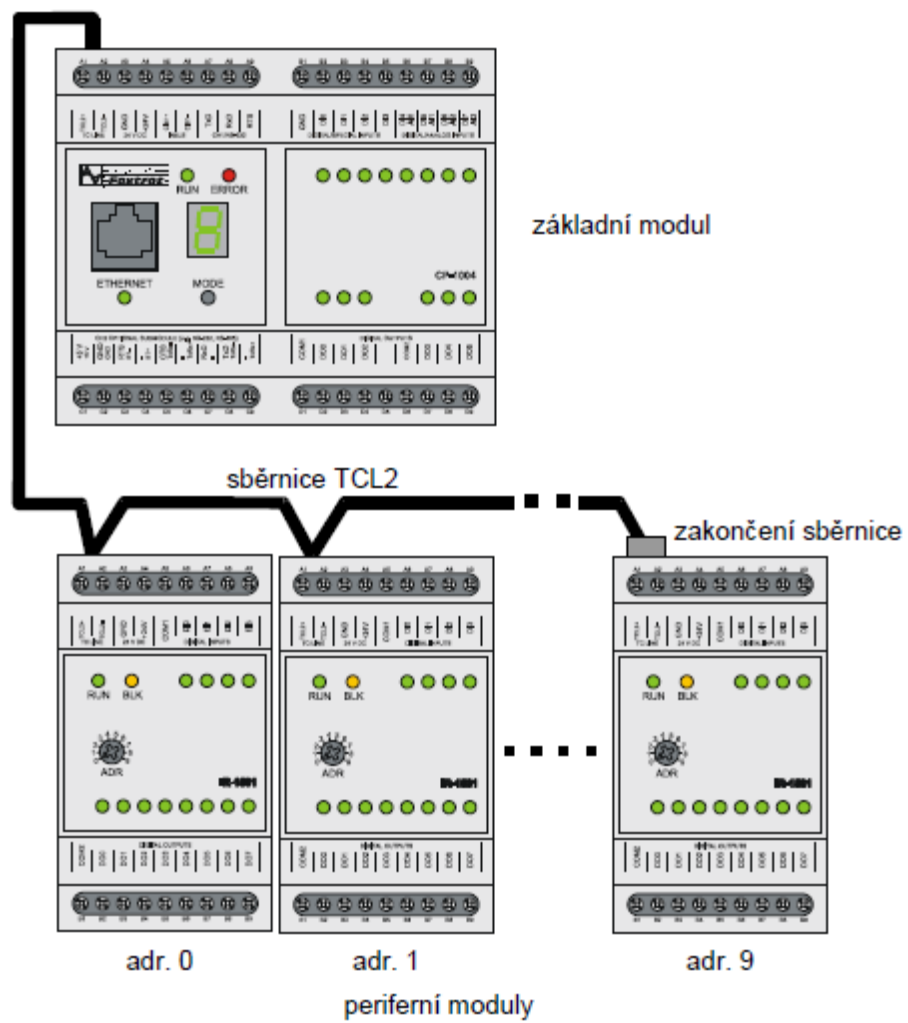
2 PLC FOXTROT

Pro řízení modelu výtahu bylo vybráno PLC od české firmy Teco Kolín s označením TECOMAT FOXTROT. Tyto programovatelné automaty představují malé kompaktní automaty s možností modulárního rozšíření. Spojují tak výhody kompaktních automatů co do velikosti a modulárních co do rozšiřitelnosti a variability. Jsou určeny pro řízení technologií v nejrůznějších oblastech průmyslu i v jiných odvětvích. Jednotlivé moduly systému jsou uzavřeny v plastových ochranných pouzdrech, které se montují na U lištu ČSN EN 50022. Díky tomu lze s nimi manipulovat bez nebezpečí poškození citlivých CMOS součástek. [7]

2.1 Komunikace

Datové komunikace mezi PLC a nadřizenými PC, mezi několika PLC, nebo mezi PLC a ostatními zařízeními jsou obvykle realizovány sériovými přenosy. Systémy FOXTROT podporují základní přenosy pomocí sítí Ethernet nebo průmyslové sítě EPSNET. Jeden asynchronní sériový kanál je pevně osazen rozhraním RS-232, druhý je volitelně osazen různými typy fyzických rozhraní (RS-232, RS-485, RS-422). Na jedné úrovni sítě EPSNET může být při použití rozhraní RS-485 až 32 účastníků a délka sériové linky až 1200 m. Volitelně jsou podporovány i jiné průmyslové protokoly a sběrnice, např. MODBUS, PROFIBUS DP, CAN, apod. Případně je možná asynchronní komunikace univerzálními přenosovými kanály ovládanými přímo z uživatelského programu. Všechny centrální jednotky jsou vybaveny rozhraním Ethernet 10/100 Mb umožňujícím provozovat současně více logických spojení. [7]

Rozšiřovací periferní moduly se k centrální jednotce připojují pomocí sériové sběrnice TCL2 (Obr. 3). Díky tomu mohou být jednotlivé části systému TECOMAT FOXTROT rozmístěny decentralizovaně tak, že jednotlivé moduly jsou umístěny přímo u ovládaných technologií a šetří tak silovou kabeláž. Celý systém může komunikovat s počítači standardu PC. Počítač tak může být využit k monitorování řízeného procesu a přitom je umístěn mimo průmyslové prostředí ve velínu nebo dispečinku. Počítač také slouží jako programovací přístroj pro PLC. Kromě PLC řady TECOMAT FOXTROT se komunikace mohou účastnit počítače standardu PC (prostřednictvím adaptéru sériového rozhraní), ale i další účastníci, kteří vyhoví požadavkům sítě EPSNET (další PLC TECOMAT, operátorské panely, apod.). [7]



Obr. 3 Připojení periferních modulů sběrnici TCL2 [7]

2.2 Základní parametry PLC Foxtrot

Typ centrální jednotky	CP-1004	CP-1014	CP-1005	CP-1015	CP-1016
Napájecí napětí (SELV)	24 V DC, +25%, -15%				
Interní jištění	vratná pojistka				
Maximální příkon	8 W				
Připojení	šroubové pevné svorky				vyjímatelné svorkovnice
typ svorek					
průřez vodiče	max.2,5 mm ²				
rozhraní Ethernet	konektor RJ-45				
Galvanické oddělení napájení od vnitřních obvodů	ne*				
Počet vstupů	8		6		15
z toho volitelně binárních / pro čítače	4		-		1
z toho volitelně binárních / analogových	4		6		13
z toho binární 230 V AC	-		-		1
Počet triakových výstupů	-		-		2
Počet reléových výstupů	6		6		10
Počet analogových výstupů	-		2		2
Uživatelský LCD displej (počet znaků)	-	4 x 20	-	4 x 20	4 x 20
Počet uživatelských tlačítek	-	6	-	6	6
Počet přidavných binárních vstupů / výstupů volitelně na submodulu kanálu CH2	7 / 0 (PX-7811) 4 / 3 (PX-7812)				-
Rozměry modulu	106 x 95 x 65 mm				159 x 95 x 65 mm

* Galvanicky oddělené jsou pouze binární vstupy 230 V, triakové a reléové výstupy a sériový kanál CH2 v závislosti na použitém submodulu realizujícím rozhraní

Tab. 1 Základní parametry základních modulů PLC Foxtrot [7]

PLC TECOMAT FOXTROT jsou konstrukčně řešeny pro montáž do skříní a stojanů. Základní parametry základních modulů jsou uvedeny v Tab. 1, základní charakteristika systému pak v Tab. 2. Všechny moduly sestavy PLC FOXTROT jsou opatřeny plastovým ochranným pouzdem a držákem pro osazení modulu na U lištu. Šířky všech modulů sestavy jsou vždy celým násobkem rozměru 17,5 mm označovaného písmenem M. Tato hodnota odpovídá zpravidla šířce jističů a dalších elektroinstalačních prvků osazovaných na U lištu. Šířka základních modulů tedy odpovídá hodnotám 6M (CP-10x4, CP-10x5), nebo 9M (CP-1016), šířka periferních modulů odpovídá hodnotám 3M nebo 1M. [7]

Vykonávání uživatelského programu
<ul style="list-style-type: none"> • cyklické, vícesmyčkové řízení s možností přerušení od času a chybových hlášení
Paměť uživatelského programu
<ul style="list-style-type: none"> • CMOS RAM, EEPROM
Základní režimy PLC
<ul style="list-style-type: none"> • RUN - vykonávání uživatelského programu • HALT - zastavení vykonávání uživatelského programu, programování PLC • možnost změny režimu příkazem po komunikačním kanálu
Blokování výstupů
<ul style="list-style-type: none"> • příkazem po komunikačním kanálu • automaticky po závažné chybě systému
Diagnostika hardwaru
<ul style="list-style-type: none"> • kontrola procesoru (watchdog) • hlídání napájecího napětí (power fail), ochrana dat při jeho výpadku • zabezpečení sériových komunikací • zabezpečení přenosu dat po I/O sběrnici
Diagnostika softwaru
<ul style="list-style-type: none"> • kontrola platnosti uživatelského programu • hlídání doby cyklu uživatelského programu • průběžná kontrola správnosti uživatelského programu (neexistující cíl skoku, přeplnění paměťových struktur, dělení nulou, neznámá instrukce, apod.)
Komunikace
<ul style="list-style-type: none"> • sériová v síti EPSNET, MODBUS, PROFIBUS DP, CAN • obecná sériová asynchronní • rozhraní Ethernet UDP / TCP / IP, RS-232, RS-485, RS-422
Další funkce
<ul style="list-style-type: none"> • automatické rozpoznávání konfigurace periferních modulů • programování EEPROM pro zálohování uživatelského programu • komunikační podpora pro monitorování dat nadřazeným systémem • možnost vykonávání uživatelského programu bez aktivace periferních modulů • přídatná paměť pro archivaci dat DataBox • RTC obvod • podpora pro analyzátor proměnných PLC • možnost fixace vstupů a výstupů periferních modulů • změna programu za chodu (online editace) • archivace projektu v paměti PLC • SDHC / SD / MMC karta se souborovým systémem FAT12 / FAT16 / FAT32 • integrovaný Web server

Tab. 2 Charakteristika PLC Foxtrot [7]

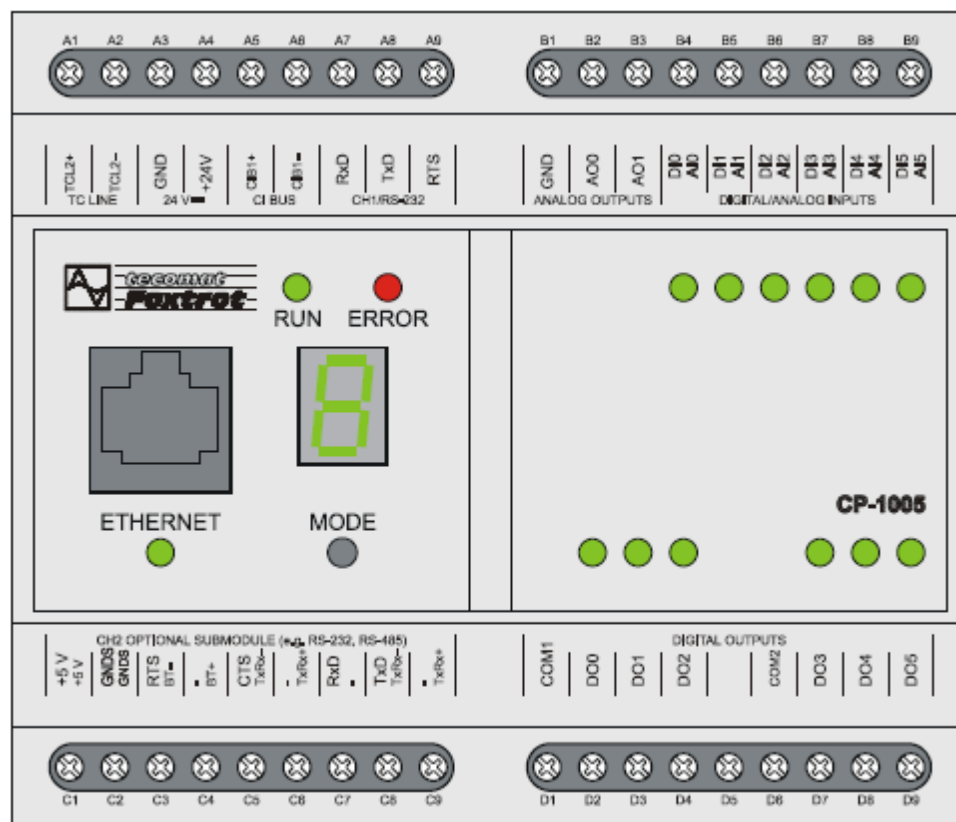
2.3 Centrální jednotka

Centrální jednotka provádí vlastní uživatelský program a obsahuje základní funkce, bez kterých se PLC neobejde. Z toho vyplývá, že centrální jednotku musí PLC obsahovat. Každá centrální jednotka má přidělené písmeno, které určuje řadu. Každá řada centrálních jednotek má své specifické vlastnosti důležité pro překladač uživatelského programu, jako například mapování a rozsah paměťového prostoru, rozsah instrukčního souboru, apod.

PLC Foxtrot obsahuje centrální jednotku řady K s instrukčním souborem, jehož součástí jsou i aritmetické operace s čísly v pevné řádové čárce o velikosti 32 bitů bez znaménka i se znaménkem, v pohyblivé řádové čárce (floating point single precision - 32 bitů a double precision - 64 bitů), instrukce PID regulátoru, podpora operátorských panelů (instrukce TER) a podpora vyššího programovacího jazyka. Režim a diagnostická hlášení jsou zobrazována na sedmsegmentovém zobrazovači nebo na displeji. [7]

2.4 PLC Foxtrot pro řízení výtahu

Model výtahu řešený v rámci této práce je řízen pomocí PLC Foxtrot se základním modulem s označením CP-1005 (Obr. 4). Submodul, neboli periferní část tohoto modulu, tvoří dva analogové a šest binárních výstupů. K dispozici je také šest vstupů. Přepínače ze zadní části celého modulu určují, jsou-li tyto vstupy analogové nebo binární.

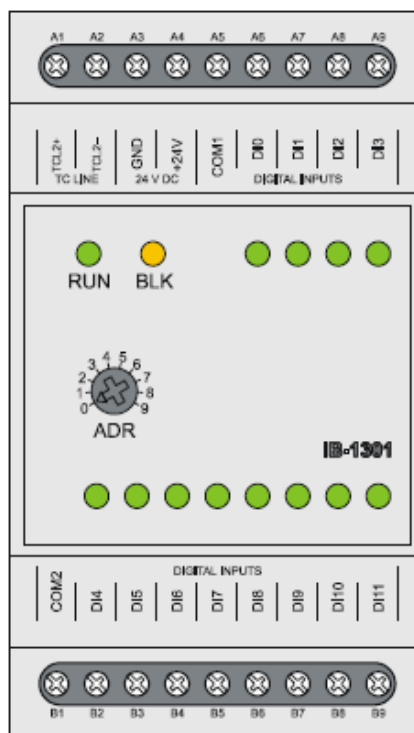


Obr. 4 Základní modul CP-1005 [7]

Počet periférií, které nabízí modul CP-1005, je pochopitelně nedostatečný pro řízení navrhovaného modelu výtahu. Z tohoto důvodu jsou k základnímu modulu připojeny další periferní moduly.

2.4.1 Modul binárních vstupů IB-1301

Periferní modul IB-1301 (Obr. 5) obsahuje 12 binárních vstupů 24 V DC. První čtyři vstupy DI0 - DI3 mohou být použity jako běžné binární vstupy nebo jako vstupy pro čítače. Všechny vstupy jsou galvanicky oddělené. Modul je napájen z vnějšího napájecího napětí 24 V DC, které není galvanicky oddělené od vnitřních obvodů. Základní parametry tohoto modulu zobrazuje Tab.3. [8]



Obr. 5 Modul IB-1301 [8]

Typ modulu	IB-1301
Počet vstupů	12
Počet vstupů ve skupině	4 + 8
Galvanické oddělení od vnitřních obvodů	ano
Diagnostika	signalizace vybuzeného vstupu na panelu
Společný vodič	minus / plus
Vstupní napětí pro log.0 (UL)	max. +5 V DC
	min. -5 V DC
pro log.1 (UH)	min. +15 V DC
	typ. +24 V DC
	max. +30 V DC
Vstupní proud při log.1	typ. 10 mA (DI0 - DI3)
	typ. 5 mA (DI4 - DI11)
Zpoždění z log.0 na log.1	5 μs (DI0 - DI3)
	5 ms (DI4 - DI11)
Zpoždění z log.1 na log.0	5 μs (DI0 - DI3)
	5 ms (DI4 - DI11)
Minimální šířka zachyceného pulzu	50 μs

Tab. 3 Základní parametry periferního modulu IB-1301 [8]

2.4.2 Modul binárních výstupů OS-1401

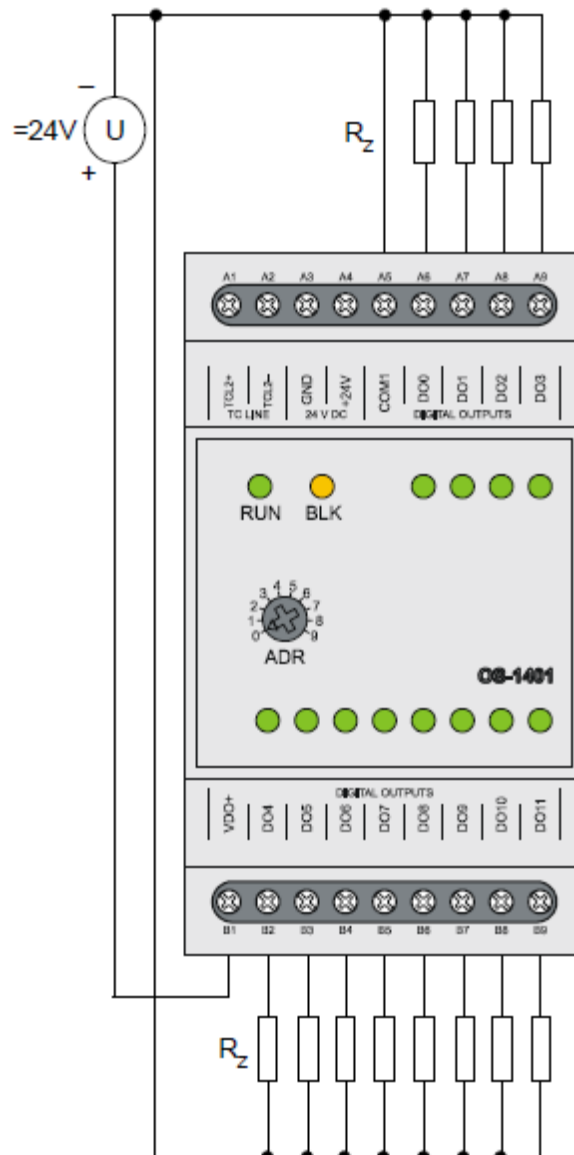
Periferní modul OS-1401 obsahuje 12 binárních výstupů 24 V DC. Všechny výstupy jsou galvanicky oddělené. Modul je napájen z napájecího napětí 24 V DC, které není galvanicky oddělené od vnitřních obvodů. [8]

Binární výstupy slouží k ovládání dvoustavových akčních a signalizačních prvků řízeného objektu napájených stejnosměrným napětím 24 V. Periferní modul OS-1401 obsahuje 12 binárních výstupů DO0 - DO11. Výstupy DO0 - DO3 umožňují spínat proudovou zátěž až 2 A každý (dohromady max. 4,4 A). Výstupy DO4 - DO11 umožňují spínat proudovou zátěž až 0,5 A každý. Výstupy jsou galvanicky odděleny od vnitřních obvodů PLC. Vybuzení (sepnutí) výstupu je signalizováno rozsvícením příslušné LED diody. Výstupy jsou uspořádány do jedné skupiny se společnou svorkou. Společná svorka skupiny má polaritu mínus. Základní parametry modulu jsou uvedeny v Tab. 4. [8]

Typ modulu	OS-1401
Počet výstupů	12
Počet výstupů ve skupině	12
Galvanické oddělení od vnitřních obvodů	ano
Diagnostika	signalizace vybuzení vstupu na panelu
Typ výstupů	tranzistorový výstup
Společný vodič	plus
Spínané napětí	9,6 až 28,8 V DC
Spínaný proud	max. 2 A (DO0 - DO3) max. 0,5 A (DO4 - DO11)
Proud společnou svorkou	max. 4,4 A (součet DO0 - DO3) max. 9 A (součet DO0 - DO11)
Zbytkový proud při rozeznutí	max 300 μ A
Doba sepnutí	max. 400 μ s
Doba rozeznutí	max. 400 μ s
Ochrana proti zkratu	ano
Omezení počátečního špičkového proudu	typ. 7,5 A
Doba odpojení počátečního špičkového proudu	typ. 4 ms
Omezení zkratového proudu	typ. 4 A
Ochrana proti přepólování	ano*
Ošetření induktivní zátěže	vnější RC člen, varistor, dioda

Tab. 4 Základní parametry periferního modulu OS – 1401 [8]

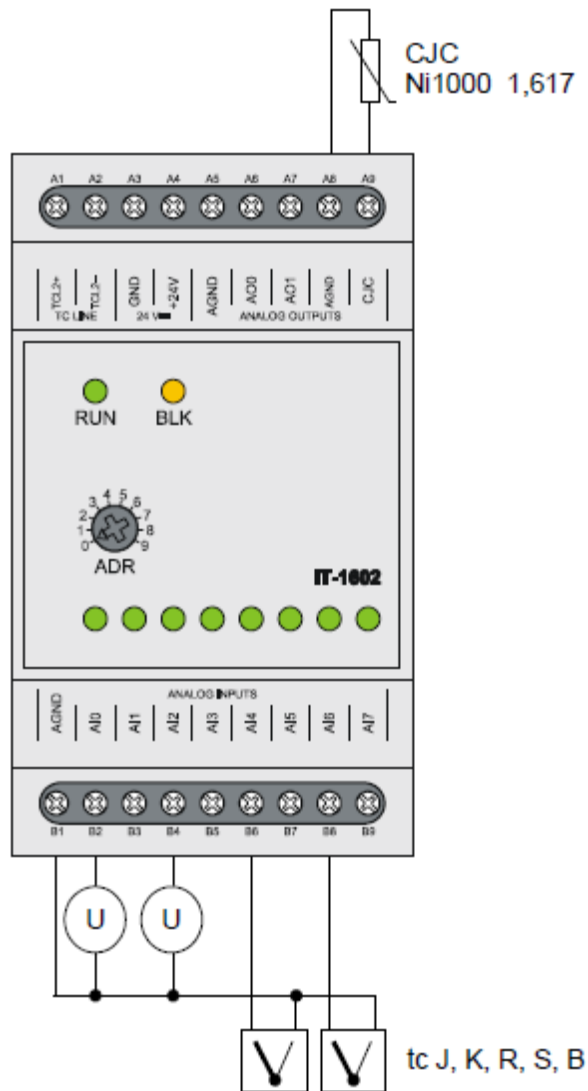
Binární výstupy jsou vyvedeny na svorky v polích DIGITAL OUTPUTS. Na Obr. 6 je schematicky naznačeno připojení zátěží napájených z nezávislých zdrojů. Výstupy jsou realizovány polovodičovými spínači s interní ochranou proti proudovému a teplotnímu přetížení. Napájení 24 V DC připojené na svorky VDO+ a COM1 je nutné pro správnou funkci výstupních spínačů. [8]



Obr. 6 Připojení zátěží k binárním výstupům modulu OS-1401 [8]

2.4.3 Analogové moduly IT-1601 a IT-1602

Periferní moduly IT-1601 a IT-1602 obsahují 8 analogových vstupů a 2 analogové výstupy. Všechny vstupy i výstupy jsou galvanicky oddělené. Modul je napájen z napájecího napětí 24 V DC, které není galvanicky oddělené od vnitřních obvodů. Oba moduly se mezi sebou liší pouze v rozlišovacích schopnostech analogových vstupů. Způsob zapojení modulů je zobrazen na Obr. 7. [8]



Obr. 7 Připojení signálů k analogovým vstupům modulu IT-1602 [8]

Mezi periferní moduly se také řadí další typy modulů, které lze označit za komunikační moduly. Např. modul UC-1203 slouží pro připojení regulačních pohonů firmy Belimo pomocí sběrnice MP-Bus, nebo Modul UC-1204 slouží pro připojení zařízení komunikujícího obousměrným protokolem OpenTherm. Firma Teco Kolín a.s., výrobce PLC Foxtrot, vyrábí celou řadu dalších periferních modulů na základě požadavku zákazníka, převážně určených pro požadovanou technologii.

3 SEKVENČNÍ ŘÍZENÍ

V otázce sekvenčního řízení výtahu je zapotřebí nejdříve objasnit základní pojem sekvenční logické funkce. Také je nutné poznamenat, že následující řádky nepojednávají o celé problematice ovládání výtahu, nýbrž pouze o základních otázkách, které je nutné si při řešení položit.

3.1 Pojem sekvenční logické funkce

Tento pojem lze vysvětlit na dvou nejčastěji používaných modelech sekvenčních logických funkcí – konečných automatech *Mooreova* a *Mealyho* typu. Oba jsou používány jako modely sekvenčních chování pro poměrně širokou třídu sekvenčních systémů. Podstatou sekvenčního chování je zpětná vazba. Uskutečňuje se prostřednictvím vnitřních proměnných (stavového vektoru), které kódují stav automatu. Stav automatu je výslednicí jeho vývoje od počátečního stavu do současného a předurčuje jeho následující vývoj. Charakterizuje určitou situaci modelového procesu. Při realizaci konečného automatu programem bývá stav interpretován jako kód stavu nebo jako soubor bitových příznaků aktivních stavů (každému stavu odpovídá jeden příznak – stavová proměnná). Pokud uvažujeme o situaci, kdy v konečném automatu je aktivován právě jeden stav – pak mluvíme o deterministickém konečném automatu. [3]

Automaty *Mooreova* a *Mealyho* typu jsou tvořeny dvěma částmi. V kombinační části je výstupní vektor *Mealyho* automatu závislý na vstupním vektoru a stavových proměnných. U *Mooreova* automatu je pak výstupní vektor přímým obrazem stavových proměnných. Jinými slovy u obou automatů je zpětná vazba realizována rozdílnými způsoby. Kombinační část lze totiž u *Mooreova* automatu rozdělit na část přechodovou, kdy automat generuje nový stav v závislosti na vstupním vektoru a část výstupní, kdy automat realizuje kombinační funkční závislost výstupního vektoru na nově vytvořeném stavu. *Mealyho* automat tyto kroky generuje v rámci jednoho bloku. Paměťové části je pak uchován aktuální stav systému v podobě stavových proměnných i v době, kdy informace ze vstupního vektoru již není k dispozici.

Oba typy automatů jsou navzájem převoditelné (pro stejnou funkci potřebuje ale *automat Mooreova* typu více stavů, než *automat Mealyho*). Ne vždy musí být výstupní vektor automatu interpretován jako kombinace hodnot výstupních proměnných, může mít i význam číselné hodnoty, kódové kombinace nebo v případě právě programovatelných

automatů jako odpovídající aktivity programu. Podobně může být interpretován i vnitřní stav. [3]

Přestože oba typy konečných automatů mají stejnou algoritmickou mohutnost a jsou navzájem převoditelné, má každý typ svou intuitivní interpretaci a specifickou oblast použití. *Automat Mooreova typu* je považován jako stacionární – většinu času setrvává v některém ze stavů, kterému je přiřazena hodnota výstupního vektoru. Přejed je interpretován jako bezvýznamná jednorázová (mžiková) událost, doprovázená změnou stavu – nové hodnoty výstupů budou odvozeny až od nového stavu. Tato interpretace odpovídá modelování a řízení mechanismů a technologických soustav. Naproti tomu *automat Mealyho typu* je vnímán jako dynamický – stále se v něm něco děje (při každé aktivaci se uskuteční nějaký přechod a s ním se vždy nově vygeneruje výstupní vektor). Tato interpretace vyhovuje především situacím, kdy jsou vyhodnocovány posloupnosti znaků, tisknutých tlačítek nebo jiných diskretních událostí (např. při realizaci komunikačních protokolů, překladech a interpretacích programovacích jazyků, při kontrole jejich syntaxe, při vyhodnocení příkazů operátora a kontrole jejich správnosti). Obvykle je automat aktivován jen při těchto událostech (např. při příjmu nového znaku, nebo při novém stisku tlačítka). Nezměněná hodnota vstupního vektoru zde znamená, že se opakovaně vyskytla stejná událost (např. druhý výskyt stejného znaku), zatímco u automatu Mooreova typu je stejná situace chápána jen jako nezměněné hodnoty vstupů. [3]

3.2 Sekvenční řízení výtahu

Obecně jsou konečné automaty, mezi které se samozřejmě řadí i *automaty Mealyho a Mooreova typu*, používány jako matematické modely pro popis funkce logických systémů. Poznatky shrnuté v předchozím bodě mohou být aplikovány při řešení problému ovládní výtahu. V tomto případě se však jedná o model a proto nemusí být kladen důraz na všechny aspekty, jejichž opomenutí by v případě řízení reálného výtahu mohlo vést k ohrožení zdraví přepravovaných osob. Na druhou stranu je model navržený a realizovaný v rámci této práce dostatečně propracovaný, aby bylo možné tvrdit, že se při řízení takového modelu setkáváme se stejnými otázkami, jako při řízení skutečného výtahu.

Na základě shrnutých poznatků lze pro popis řízení modelu výtahu jako matematický model použít konečný Mealyho automat, který je, jak již bylo řečeno, vnímán jako dynamický, ve kterém se stále něco děje a což lze bez zbytečných debat říci i o řízení výtahu.

Jako vstupní vektor, který vyvolává změnu vnitřního stavu systému, nám může při řízení výtahu figurovat hned několik prvků. Mezi ně patří hlavně stisknutí tlačítka pro přivolání výtahu, stisknutá tlačítka v kabině výtahu, změna směru chodu motoru, změna aktuálního podlaží kabiny atp. To vše je změna stavů, které je potřeba uchovávat jako stavové proměnné a na jejich základě pak rozhodovat (generovat výstupní vektor, resp. odezvu řídicího programu), kterým směrem se bude která kabina pohybovat. Správná optimalizace takového řízení (která je v praxi samozřejmě vyžadována hlavně kvůli minimalizaci čekací doby a snížení spotřeby elektrické energie a tím i nákladů na provoz) je už pak otázkou vhodně vyřešených sekvenčních logických funkcí a správné programové realizace.

3.3 Programová realizace

Jeden z možných způsobů takové realizace je vytvoření tabulky možných stavů celého systému a ke každému stavu přiřadit výstupní vektor. Toto řešení může být velmi elegantní zvláště tehdy, využijeme-li vlastních vhodných datových typů nebo tabulek. Ty pak uložíme do paměti PLC a jednoduchým programem budeme na základě aktuálního stavu vyhledávat patřičnou odezvu. Program se tím značně minimalizuje a zpřehlední. Je však důležité si uvědomit, kde je hranice takového řešení. Důležité je také určit všechny možné stavy a zvážit, jestli by výsledná tabulka nebyla příliš obsáhlá a složitá.

Je proto na programátorovi, zdali raději nezvolí intuitivní přístup a každé ošetření stavu naprogramuje zvlášť. Program pak sice nebude tak krátký a bude méně přehledný, nicméně základem je každý důležitý řádek okomentovat. Zárukou mu pak bude zachycení všech možných stavů a zajištění patřičné odezvy na ně.

4 NORMA IEC 61 131

Norma IEC 61 131 pro programovatelné řídicí systémy má pět základních částí a představuje souhrn požadavků na moderní řídicí systémy. Je nezávislá na konkrétní organizaci či firmě a má širokou mezinárodní podporu. Jednotlivé části normy jsou věnovány jak technickému tak programovému vybavení těchto systémů. [6]

V ČR byly přijaty jednotlivé části této normy pod následujícími čísly a názvy:

- ČSN EN 61 131-1 Programovatelné řídicí jednotky - Část 1: Všeobecné informace
- ČSN EN 61 131-2 Programovatelné řídicí jednotky - Část 2: Požadavky na zařízení a zkoušky
- ČSN EN 61 131-3 Programovatelné řídicí jednotky - Část 3: Programovací jazyky
- ČSN EN 61 131-4 Programovatelné řídicí jednotky - Část 4: Podpora uživatelů
- ČSN EN 61 131-5 Programovatelné řídicí jednotky - Část 5: Komunikace
- ČSN EN 61 131-7 Programovatelné řídicí jednotky - Část 7: Programování fuzzy řízení

V Evropské unii jsou tyto normy přijaty pod číslem EN IEC 61 131. [6]

4.1 Programovatelné řídicí jednotky - ČSN EN 61 131-3 Část 3:

Programovací jazyky

Programovací jazyky definuje norma IEC 61 131-3 (**dále jen norma**), která je třetí částí z rodiny norem IEC 61 131 a představuje první vážný pokus o standardizaci programovacích jazyků pro průmyslovou automatizaci. Na normu je možné pohlížet z různých hledisek, např. tak, že je to výsledek náročné práce sedmi mezinárodních společností, které do vypracování normy vložily svoji desetiletou zkušenost na poli průmyslové automatizace, nebo tak, že ve svém souhrnu obsahuje asi 200 stran textu, a asi 60 tabulek. Na jejím vytváření pracoval tým patřící do pracovní skupiny SC65B WG7 mezinárodní standardizační organizace IEC (International Electrotechnical Commission). Výsledkem je **specifikace syntaxe a sémantiky unifikovaného souboru programovacích jazyků, včetně obecného softwarového modelu a strukturujícího jazyka**. Tato norma byla přijata jako směrnice u většiny významných výrobců PLC. [6]

V následujících bodech budou popsány nejdůležitější pasáže normy. Tyto pasáže by měl znát každý programátor, který by podle normy chtěl programovat. Je ještě potřeba říci, že

programovat dle normy není vždy nutné a pokud to podmínky dovolí, tak může programátor využít způsoby programování, které ovládá nejlépe. Přínosem normy je však jedna nesporná výhoda a tou je univerzálnost napsaného programu. Program, který je podle ní napsán, by měl fungovat na PLC od každého výrobce, jehož zařízení normu podporuje. Výhodou je to, že tento program stačí napsat a odladit pouze jednou. Tím pak odpadají starosti při přechodu na jiné PLC nebo při použití více PLC od různých výrobců.

4.1.1 Základní stavební bloky programu

Základním pojmem při programování podle normy je termín *Programová Organizační Jednotka* nebo zkráceně *POU (Program Organisation Unit)*. Jak vyplývá z názvu, POU je nejmenší nezávislá část uživatelského programu. POU mohou být dodávány od výrobce řídicího systému nebo je může napsat uživatel. Každá POU může volat další POU a při tomto volání může volitelně předávat volané POU jeden nebo více parametrů. Existují tři základní typy POU :

- *funkce (function, FUN)*
- *funkční blok (function block, FB)*
- *program (program, PROG)*

Nejjednodušší POU je *funkce*, jejíž hlavní charakteristikou je to, že pokud je volána se stejnými vstupními parametry, musí produkovat stejný výsledek (funkční hodnotu). Funkce může vrátit pouze jeden výsledek.

Dalším typem POU je *funkční blok*, který si na rozdíl od funkce, může pamatovat některé hodnoty z předchozího volání (např. stavové informace). Ty pak mohou ovlivňovat výsledek. Hlavním rozdílem mezi funkcí a funkčním blokem je tedy schopnost funkčního bloku vlastnit paměť pro zapamatování hodnot některých proměnných. Tuto schopnost funkce nemají a jejich výsledek je tedy jednoznačně určen vstupními parametry při volání funkce. Funkční blok může také (na rozdíl od funkce) vrátit více než jeden výsledek.

Posledním typem POU je *program*, který představuje vrcholovou programovou jednotku v uživatelském programu. Centrální jednotka PLC může zpracovávat více programů a programovací jazyk ST obsahuje prostředky pro definice spouštění programů (v jaké periodě vykonávat program, s jakou prioritou, apod.). Norma pro POU dále popisuje její syntaxi. Především způsob zápisu deklarační a výkonné části POU. Více podrobností o POU lze najít v [6, strana 8]

4.1.2 Datové typy

Pro programování v některém z jazyků podle normy jsou definovány tzv. *elementární* (Tab. 5), předdefinované datové typy (Elementary data types).

Klíčové slovo	Anglicky	Datový typ	Bitů	Rozsah hodnot
BOOL	Boolean	Boolovské číslo	1	0,1
SINT	Short integer	Krátké celé číslo	8	-128 až 127
INT	Integer	Celé číslo	16	-32 768 až +32 767
DINT	Double integer	Celé číslo, dvojnásobná délka	32	-2 147 483 648 až +2 147 483 647
USINT	Unsigned short integer	Krátké celé číslo bez znaménka	8	0 až 255
UINT	Unsigned integer	Celé číslo bez znaménka	16	0 až 65 535
UDINT	Unsigned double integer	Celé číslo bez znaménka, dvojnásobná délka	32	0 až +4 294 967 295
REAL	Real (single precision)	Číslo v pohyblivé řádové čárce (jednoduchá přesnost)	32	±2.9E-39 až ±3.4E+38 Podle IEC 559
LREAL	Long real (double precision)	Číslo v pohyblivé řádové čárce (dvojnásobná přesnost)	64	Podle IEC 559
TIME	Duration	Trvání času	24d 20:31:23.647	
DATE	Date (only)	Datum	Od 1.1.1970 00:00:00	
TIME_OF_DAY nebo TOD	Time of day (only)	Denní čas	24d 20:31:23.647	
DATE_AND_TIME nebo DĚ	Date and time of day	„Absolutní čas“	Od 1.1.1970 00:00:00	
STRING	String	Řetězec	Max.255 znaků	
BYTE	Byte(bit string of 8 bits)	Sekvence 8 bitů	8	Není deklarován rozsah
WORD	Word (bit string of 16bits)	Sekvence 16 bitů	16	Není deklarován rozsah
DWORD	Double word (bit string of 32 bits)	Sekvence 32 bitů	32	Není deklarován rozsah

Tab. 5 Elementární datové typy [6]

Dále jsou definovány *rodové* datové typy (Generic data type) které vyjadřují vždy celou skupinu (rod) datových typů. Jsou uvozeny prefixem **ANY**. Např. zápisem **ANY_BIT** se rozumí všechny datové typy **DWORD**, **WORD**, **BYTE**, **BOOL**. Přehled rodových

datových typů je uveden v Tab. 6. Názvy rodových datových typů začínající **ANY_** nejsou podle normy klíčovými slovy. Slouží pouze k označení skupiny typů se stejnými vlastnostmi. [6]

ANY				
ANY_BIT	ANY_NUM		ANY_DATE	TIME STRING
BOOL BYTE WORD DWORD	ANY_INT		ANY_REAL	
		INT SINT DINT	UINT USINT UDINT	
			DATE DATE_AND_TIME TIME_OF_DAY	

Tab. 6 Rodové datové typy [6]

Norma ohledně datových typů dále popisuje uživatelsky definované datové typy. Více podrobností o odvozených datových typech v [6, strana 23]

4.1.3 Proměnné

Podle normy jsou *proměnné* v podstatě prostředkem pro identifikaci datových objektů, jejichž obsah se může měnit, tzn. dat přiřazených ke vstupům, výstupům nebo paměti PLC. Proměnná může být deklarována některým z elementárních datových typů nebo některým z odvozených (uživatelských) datových typů. Tím se programování podle normy přiblížilo k běžným zvyklostem. Místo dříve používaných hardwarových adres nebo symbolů jsou zde definovány proměnné tak, jak se používají ve vyšších programovacích jazycích. Proměnné jsou identifikátory (jména) přiřazené programátorem, které slouží v podstatě pro rezervaci místa v paměti a obsahují hodnoty dat programu. Více o proměnných v [6, strana 32]

4.1.4 Programovací jazyky

Norma definuje čtyři programovací jazyky pro programování PLC. Patří mezi ně jazyk seznamu instrukcí (IL, Instruction List) a jazyk strukturovaného textu (ST, Structured Text) jako textové programovací jazyky a jazyk kontaktních schémat (LD, Ladder Diagram, jazyk kontaktních schémat) a jazyk funkčního blokového schématu (FBD, Function Block Diagram) jako grafické programovací jazyky. Programátor by měl dobře znát možnosti daného jazyka, ale obecně lze říci, že s každým z jazyků lze daný problém naprogramovat. Je totiž možné v jednom projektu kombinovat více jazyků

dohromady. Platí však, že textové jazyky a to zejména objektově orientovaný Structured Text, jsou velmi výkonné vyšší programovací jazyky

4.1.4.1 Jazyk seznamu instrukcí IL

Jazyk seznamu instrukcí (Instruction List) (Obr. 8) je nízkourovňový jazyk typu assembler. Tento jazyk patří mezi řádkově orientované jazyky. *Seznam instrukcí* se skládá ze sekvence (posloupnosti) *instrukcí*. Každá instrukce (příkaz) začíná na novém řádku a obsahuje *operátor*, který může být doplněn *modifikátory*, a pokud je to pro konkrétní instrukce nutné, tak dále obsahuje jeden nebo více *operandů* oddělených čárkami. Na místě operandů mohou být libovolné reprezentace dat definované pro literály a proměnné.

Pro účely identifikace může být před instrukcí uvedeno *návěští*, za kterým následuje dvojtečka (:). Návěští slouží k označení místa v programu pro instrukce volání resp. skoku. Na posledním místě na řádku instrukce může být uveden komentář. Mezi instrukcemi mohou být vloženy prázdné řádky. V normě jsou také definovány způsoby vytváření funkcí a funkčních bloků a jejich volání pomocí jazyka IL. Norma dále definuje operátory, modifikátory a operandy. Více informací v [6, strana 63]

```
VAR_GLOBAL
  AT %X1.2      : BOOL;
  AT %Y2.0      : BOOL;
END_VAR

PROGRAM Example_IL
  VAR
    tmp1, tmp2  : BOOL;
  END_VAR

  Step1: LD      %X1.2 // load bit from PLC input
         AND     tmp1  (* AND temporary variable *)
         ST      %Y2.0 (* store to PLC output *)
         (* empty instruction *)
  Step2: (* label *)
         LDN     tmp2
END_PROGRAM
```

Obr. 8 Ukázka programu v jazyce IL [6]

4.1.4.2 Jazyk strukturovaného textu ST

Jazyk strukturovaného textu ST je velmi výkonný vyšší programovací jazyk, který má kořeny ve známých jazycích Ada, Pascal a C. Je objektově orientován a obsahuje všechny podstatné prvky moderního programovacího jazyka, včetně větvení (IF-THEN-ELSE a CASE OF) a iterační smyčky (FOR, WHILE a REPEAT). Tyto prvky mohou být vnořovány. Tento jazyk je vynikajícím nástrojem pro definování komplexních funkčních

bloků. Algoritmus zapsaný v jazyce ST lze rozdělit na jednotlivé *příkazy* (*statements*). Příkazy se používají pro výpočet a přiřazení hodnot, řízení toku vykonávání programu a pro volání resp. ukončení POU. Část příkazu, která vypočítává hodnotu, je nazývána *výraz*. Výrazy produkují hodnoty nezbytné pro provádění příkazů. Výraz je konstrukce, ze které se po vyhodnocení vygeneruje hodnota odpovídající některému z datových typů. Výraz se skládá z *operátorů* (Tab. 7) a *operandů*. Operandem může být literál, proměnná, volání funkce nebo jiný výraz.

Operátor	Operace	Priorita
()	Závorky	Nejvyšší
**	Umocňování	
- NOT	Znaménko Doplňek	
* / MOD	Násobení Dělení Modulo	
+ -	Sčítání Odčítání	
<, >, <=, >=	Porovnávání	
= <	Rovnost Nerovnost	
&, AND	Boolovské AND	
XOR	Boolovské exkluzivní OR	
OR	Boolovské OR	Nejnižší

Tab. 7 Operátory v jazyce strukturovaného textu ST [6]

Vyhodnocení výrazu spočívá v aplikaci operátorů na operandy a to s ohledem na prioritu vyjádřenou v Tab. 7. Operátory s nejvyšší prioritou ve výrazu jsou aplikovány nejdříve, pak následují další operátory směrem k nižší prioritě, dokud není vyhodnocování dokončeno. Operátory se stejnou prioritou se vyhodnocují tak jak jsou zapsány ve výrazu směrem odleva doprava. [6]

```

PROGRAM PRIKLAD
VAR                                     // lokální proměnné
  A      : INT := 2;
  B      : INT := 4;
  C      : INT := 5;
  D      : INT := 8;
  X, Y   : INT;
  Z      : REAL;
END_VAR

X := A + B - C * ABS(D);           // X = -34
Y := (A + B - C) * ABS(D);        // Y = 8
Z := INT_TO_REAL( Y );
END_PROGRAM

```

Obr. 9 Ukázka programu v jazyce ST [6]

Norma dále pro jazyk ST definuje příkaz přiřazení, příkazy podmíněné (IF, CASE) a iterační (FOR, WHILE, REPEAT) a způsoby volání funkčních bloků. Více o jazyce ST v [6, strana 69]

4.1.4.3 Společné prvky grafických jazyků

Stejně jako je tomu u textových jazyků obsahuje každá deklarace POU v grafickém jazyce deklarační a výkonnou část. Deklarační část je naprosto shodná s textovými jazyky, výkonná část je rozdělena do tzv. obvodů (v anglické literatuře označovaných jako networks). Každý obvod se skládá z následujících prvků (Obr. 10):

Návěští obvodu

Každý obvod může být opatřen návěstím, což je uživatelem definovaný identifikátor zakončený znakem dvojtečka. Návěští pak může být cílem skoku při větvení výkonného programu POU. Rozsah působnosti obvodu a jeho návěští je *lokální* v rámci té programové organizační jednotky, ve které je obvod umístěn. Návěští obvodu není povinné. To je generováno automaticky a slouží k lepší orientaci ve složitých POU. Při vložení nového obvodu se následující obvody automaticky přečíslovají. Grafický editor pak umožňuje rychle vyhledávat obvody v POU podle jejich čísel. [6]

Komentář obvodu

Mezi návěstím obvodu a grafikou obvodu může být umístěn komentář obvodu. Ten může být víceřádkový a může obsahovat znaky národních abeced. Komentář obvodu není povinný. [6]

Grafika obvodu

Grafika obvodu obsahuje grafické prvky propojené spojnicemi. Grafickým prvkem může být například spínací kontakt, blok časovače nebo výstupní cívka. Spojnice (propojovací čáry) určují tok informace, například z výstupu časovače na výstupní cívku. Každý grafický prvek může být volitelně opatřen komentářem. [6]

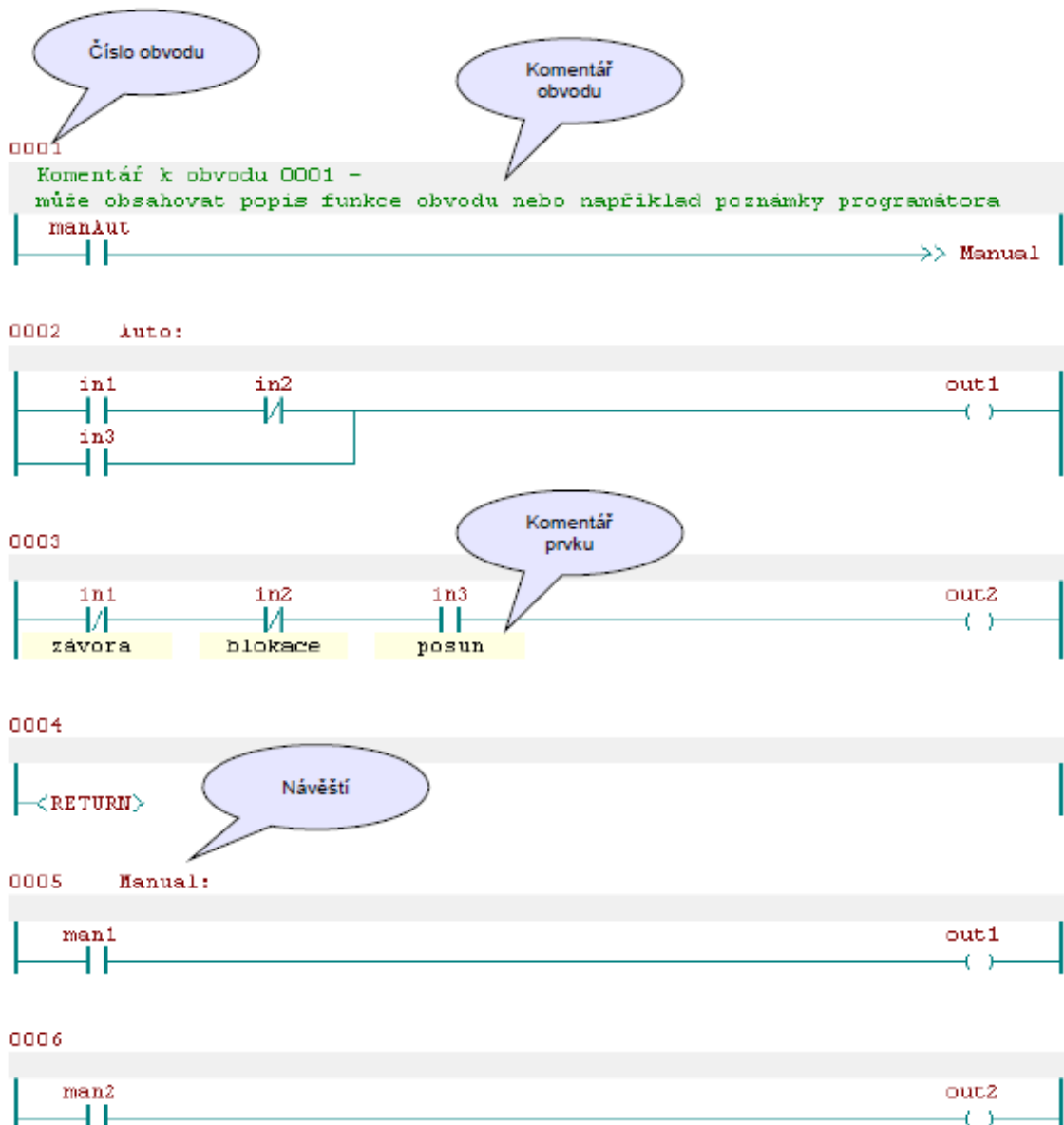
Směr toku v obvodech

Grafické jazyky se používají pro reprezentaci toku “myšleného množství” skrze jeden nebo více obvodů reprezentujících algoritmus řízení. Toto myšlené množství můžeme chápat jako:

- “*tok energie*”, analogický k toku elektrické energie v elektromechanických reléových systémech, který se běžně používá v reléových schématech
- “*tok signálu*”, analogický k toku signálů mezi prvky systému zpracovávajícího signály, který se běžně používá ve funkčních blokových diagramech (schématech)

Příslušné “myšlené množství” protéká podél čar mezi prvky sítě podle následujících pravidel:

- Tok energie v jazyku LD probíhá odleva doprava.
- Tok signálu v jazyku FBD probíhá od výstupu (na pravé straně) funkčního bloku ke vstupu (na levé straně) dalšího připojeného funkčního bloku [6]



Obr. 10 Společné prvky grafických jazyků [6]

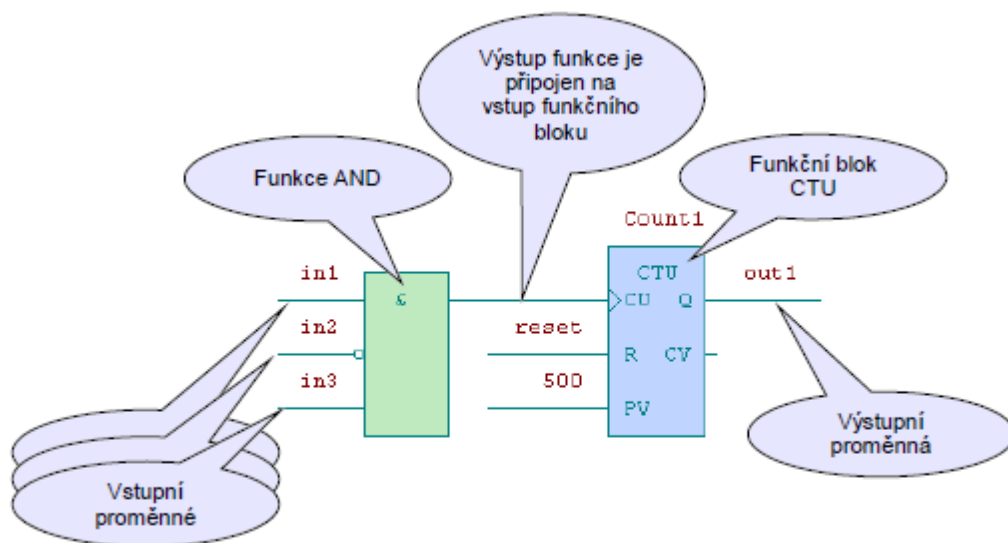
4.1.4.4 Jazyk kontaktních schémat LD

Jazyk kontaktních schémat (Ladder Diagram) pochází z elektromechanických reléových obvodů je založen na grafické reprezentaci reléové logiky. Tento jazyk je primárně určen pro zpracování booleovských signálů. Jak už bylo řečeno, výkonná část POU v jazyce LD je složena z obvodů (networks). Obvod je v jazyce LD ohraničen tzv. napájecími sběrnicemi (power rails) na levé a pravé straně. Z levé napájecí sběrnice „vede“ logická jednička (TRUE) do všech na ni připojených grafických prvků,

typicky spínacích a rozpínacích kontaktů. V závislosti na jejich stavu se pak logická jednička propouští nebo nepropouští do následujících prvků zapojených v obvodu. Poslední prvek vpravo bývá výstupní a je připojen na pravou napájecí sběrnici. Typickým představitelem výstupního prvku je cívka. Norma zde dále podrobněji popisuje jazyk LD a to zejména významy jednotlivých grafických prvků, způsoby jejich propojování a vyhodnocování. Více podrobností o jazyku LD v [6, strana 81]

4.1.4.5 Jazyk funkčního blokového schématu FBD

Jazyk funkčního blokového schématu (Function Block Diagram) je založen na propojování funkčních bloků a funkcí. Stejně jako v jazyce LD jsou i v jazyce FBD funkce a funkční bloky reprezentovány obdélníkem. Rozdíl je v tom, že v jazyce LD lze spojnicemi mezi prvky přenášet pouze hodnoty typu BOOL zatímco v jazyce FBD mohou spojnice mezi grafickými prvky přenášet hodnoty libovolného typu



Obr. 11 Grafika obvodu v jazyce FBD [4]

Více podrobností o jazyku LD a to zejména význam jednotlivých grafických prvků v [6, strana 90]

Norma prezentovaná v [4] je zpracována i s názornými příklady a lze tento materiál považovat za příručku pro programátora, který chce programovat podle normy IEC 61 131 – 3.

II. PRAKTICKÁ ČÁST

5 NÁVRH MODELU

V atriu budovy Fakulty aplikované informatiky se nachází osmipodlažní výtah se dvěma kabinami, který sloužil jako předloha pro navrhovaný model. Cílem práce je však hlavně vytvoření učební pomůcky pro výuku předmětu Programovatelné automaty a při návrhu modelu se z toho faktu vycházelo.

5.1 Požadavky

Prvním krokem při realizaci této práce bylo pochopitelně stanovení požadavků na výsledný model.

Model výtahu

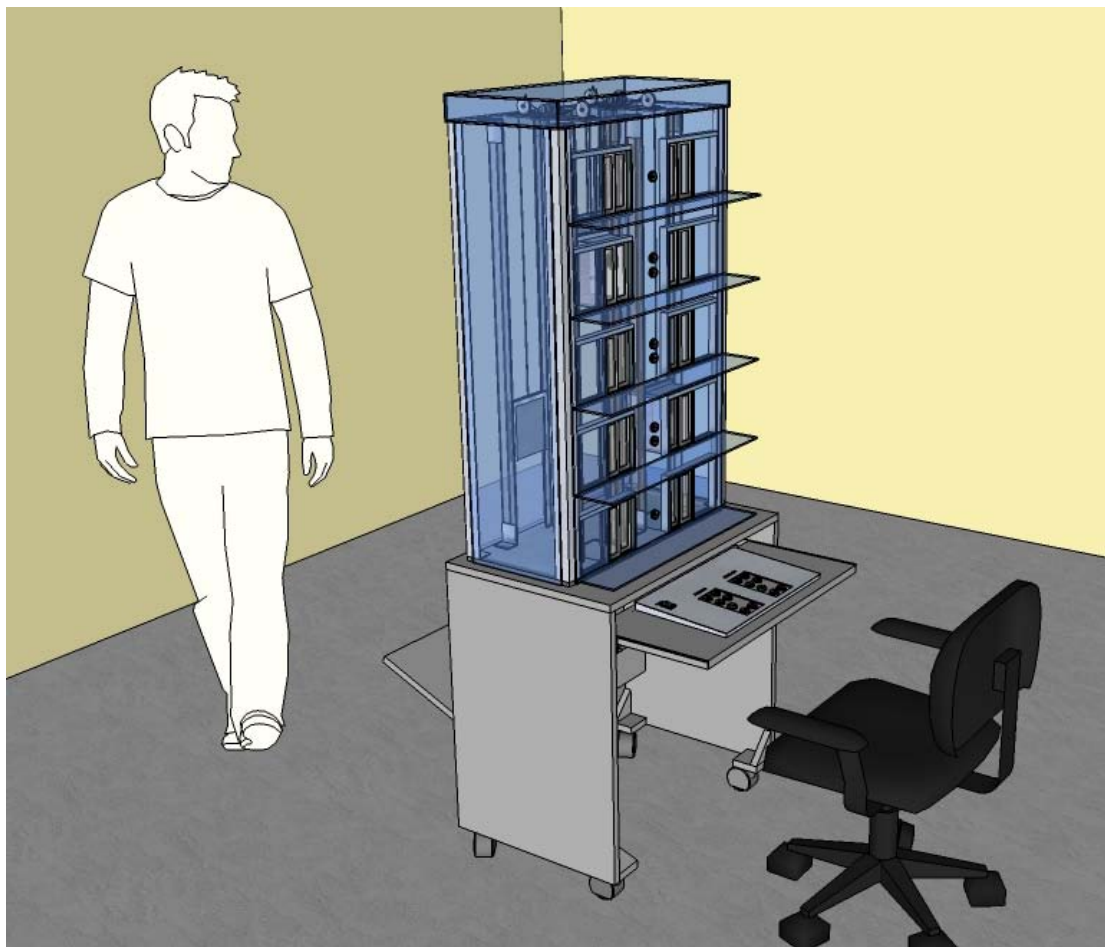
- má být pětipodlažní se dvěma kabinami
- bude řízen pomocí PLC Tecomat Foxtrot
- bude součástí uceleného pracoviště, které má být také navrženo
- má být co nejrealističtější a má obsahovat funkční prvky, především otevíratelné dveře na každém podlaží
- bude mít všechny aktivní prvky kryté proti manipulaci cizích osob

5.2 Vizualizace

Pro snadnější vyjádření mých představ řešení tohoto úkolu jsem vytvořil 3D model pomocí aplikace Google SketchUp (www.sketchup.google.com). Výstup z této aplikace je v podobě video ukázky součástí přílohy PII.

Hlavním důvodem vytvoření vizualizace návrhu výtahu ale byla potřeba konzultovat návrh výtahu s lidmi, kteří budou s modelem pracovat v rámci výuky.

Podle prvotního návrhu (Obr. 12) mělo pracoviště tvořit pouze samotný model se stolem přesně navrženým na rozměry modelu. PLC mělo být zabudováno ve spodní uzamykatelné části stolu. Stůl měl mít ještě vysouvací pojezd pro ovládací pult výtahových kabin a také kolečka pro snadnou manipulaci se stolem. Představa byla taková, že by se s celým modelem přijelo ke stolu s počítačem a s ním by se PLC spojilo pomocí síťového kabelu. Programátor by seděl u stolu s počítačem a svůj navržený program by pak ladil u stolu s modelem výtahu.



Obr. 12 Prvotní návrh

Po konzultacích s vedoucím práce se ale takové řešení ukázalo být nepřilíživě praktické a proto vznikl nápad pro vytvoření uceleného pracoviště. Návrh samotného modelu vyhovoval představám vedoucího a ten zůstal stejný. Rozšířil se však pracovní stůl a počítač určený pro programování PLC bude součástí celého pracoviště. Také umístění samotného PLC je oproti prvotnímu návrhu jiné. Počítá se s tím, že bude umístěno přímo pod deskou pracovního stolu ve speciální uzamykatelné skříňce. Konečný návrh modelu, ze kterého se pak vycházelo při jeho realizaci, je na Obr. 13;



Obr. 13 Konečný návrh

6 REALIZACE MODELU

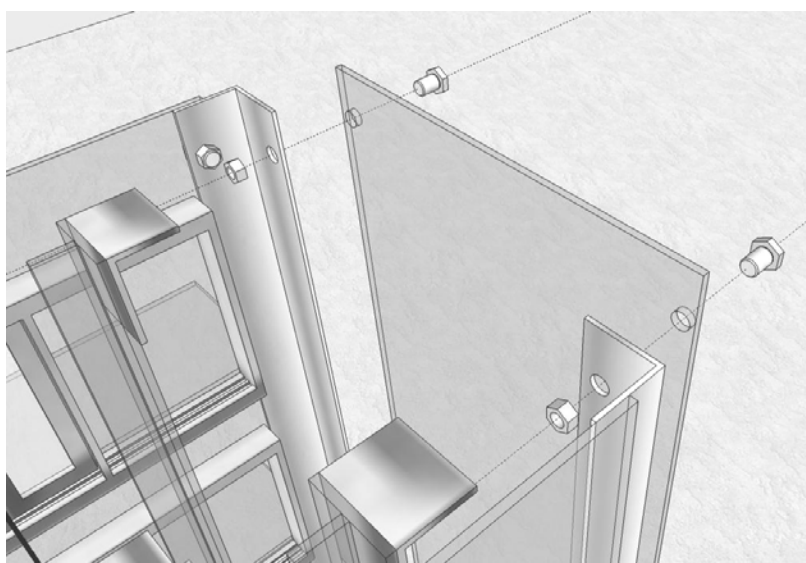
Při řešení návrhu představeného v předchozím bodě bylo zapotřebí stanovit otázky, které je nutné vyřešit. Každý bod v této kapitole představuje jednu takovou otázku a snaží se co nejlépe a nejpodrobněji danou problematiku vysvětlit.

6.1 Konstrukce modelu

Dominantním materiálem pro stavbu modelu je plexisklo. S výjimkou prvků pro upevnění, kolejnič kabin a podložky modelu je vše vyrobeno z tohoto materiálu. Základní rozměry modelu jsou Š/V/H 630mm/1000mm/300mm. Tloušťka plexiskla je 4mm.

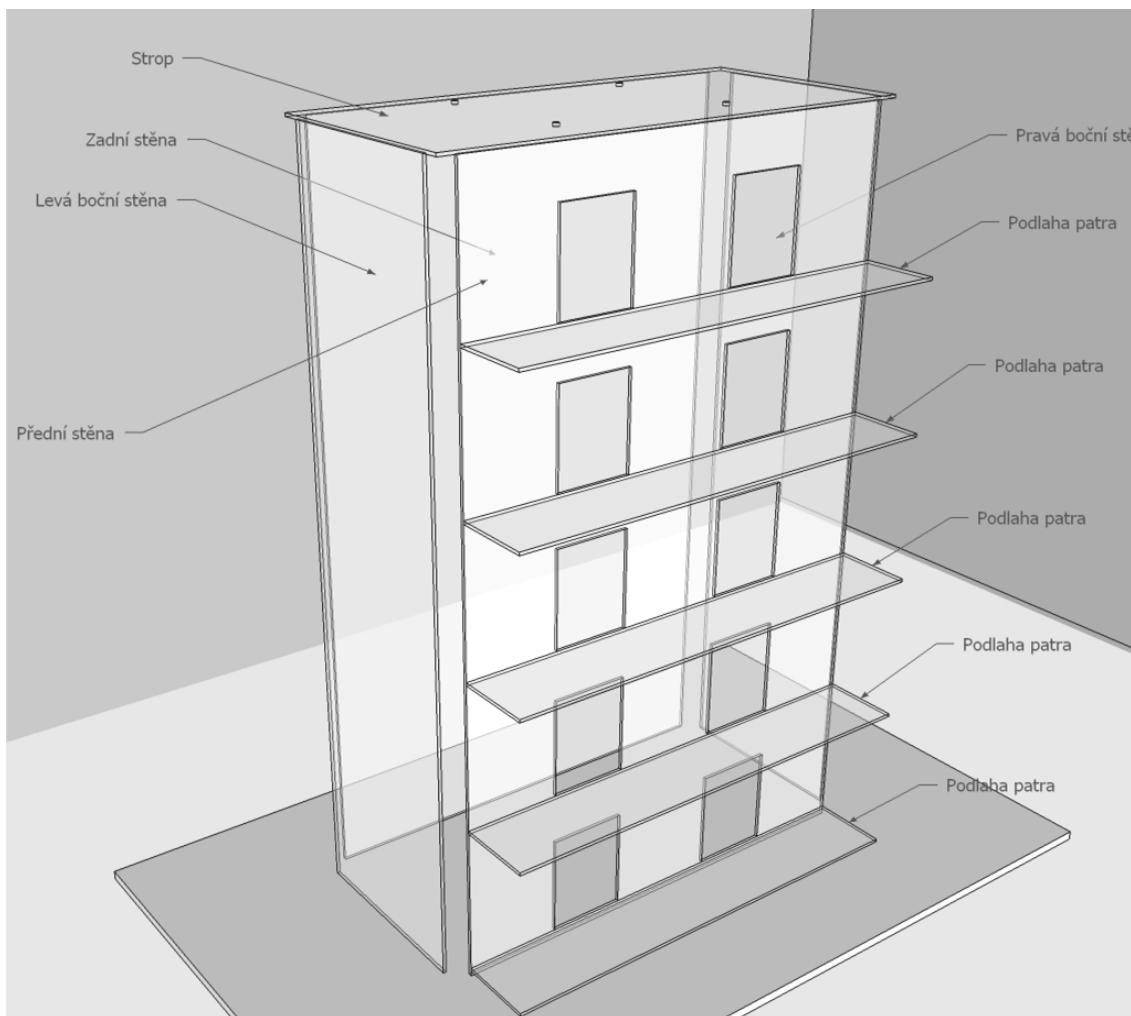
6.1.1 Stěny výtahu

Každá ze čtyř stěn, které tvoří korpus výtahu a které uzavírají prostor, ve kterém jezdí kabiny, je samostatně oddělitelná od kovové nosné konstrukce. Upevněná je pomocí šroubů se závitem M5 (Obr. 14). Tyto šrouby mají svou hlavu z vnější strany a to znesnadňuje přístup do vnitřního prostoru, který by neměl být volně přístupný. V případě nutného zásahu z důvodu nějaké poruchy je potřeba odtáhnout matky šroubů z vnitřního prostoru. Do něj je možné se dostat po odejmutí jedné z bočních stěn, která má hlavy upevňujících šroubů uvnitř a matky jsou volně přístupné z vnější strany. Všechny matky jsou na šroubech přitaženy pouze rukou, protože není nutné a ani žádoucí aby byly silně dotaženy. Přední a zadní stěna je ještě navíc přitažena ve spodní části uprostřed, protože se jedná o větší plochu, která by bez tohoto opatření odstávala od upevňovací konstrukce.



Obr. 14 Způsob upevnění stěn

Stěny jsou vyrobeny z plexiskla. Tento materiál se v domácích podmínkách velmi těžko opracovává a zejména přední stěna má kvůli výřezům pro dveře vysoké nároky na přesnost. Tyto díly (Obr. 15) proto byly vyrobeny firmou zpracovávající plasty.

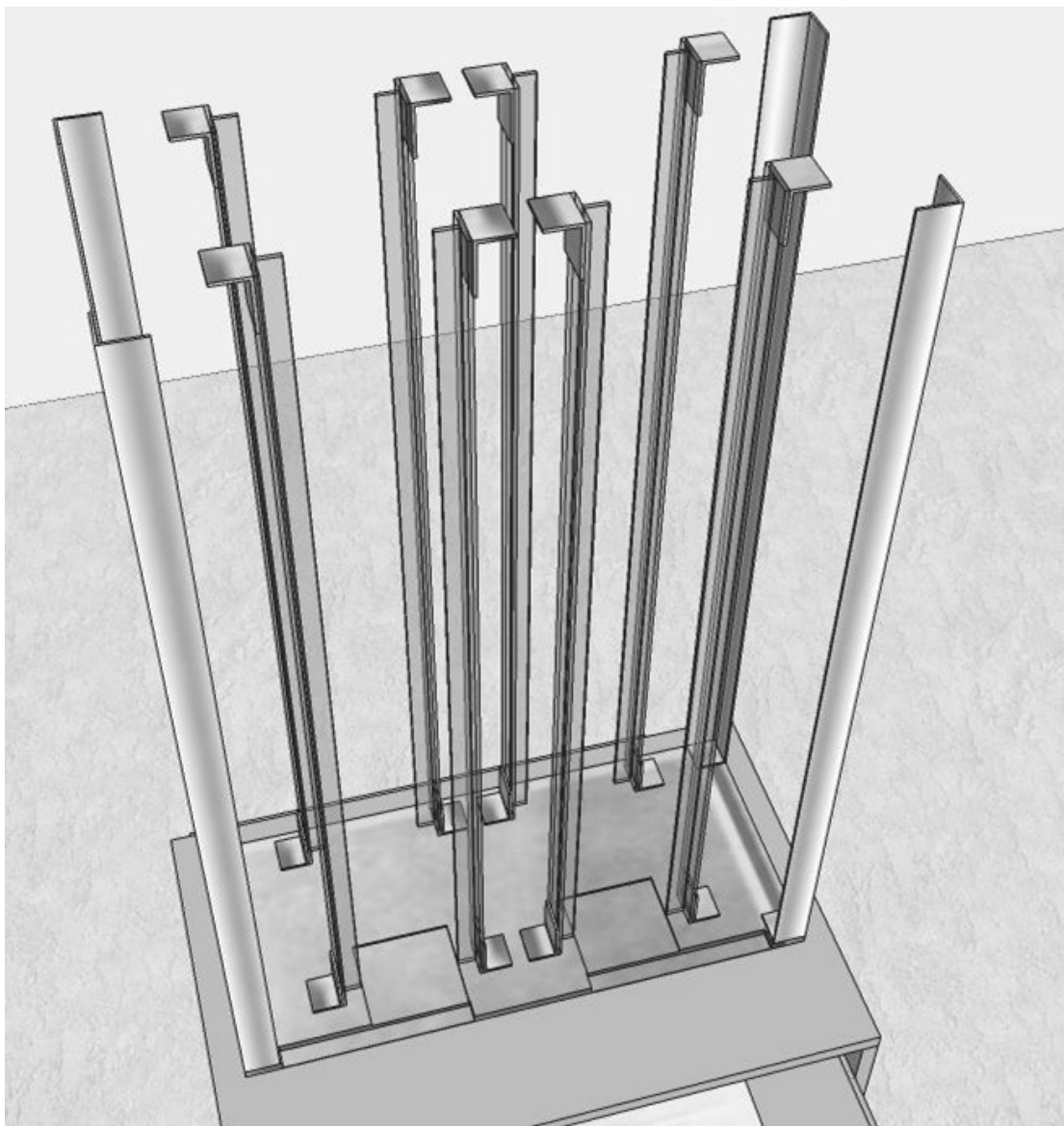


Obr. 15 Stěny výtahu

6.1.2 Kovové konstrukce

Stěny výtahu se připevňují ke kovovým konstrukcím (Obr. 16), které udržují tvar výtahu. Jsou vyrobeny z eloxovaných hliníkových L profilů o rozměrech profilu 40 x 40 mm. Tyto konstrukce jsou připevněny k podložce celého výtahu pomocí šroubů. Pravoúhlý tvar konstrukce zajišťuje deska s výřezy, která tvoří strop výtahu.

Dalšími kovovými prvky v modelu jsou kolejnice pro kabiny výtahu. Ty tvoří hliníkové T profily a jsou upevněny i v podložce i ve stropě výtahu a celou konstrukci tím zpevňují.



Obr. 16 Kovové konstrukce

6.1.3 Strop výtahu

Strop uzavírá celý vnitřní prostor a je posazen na kovových konstrukcích, ke kterým je přišroubován a tím je zpevňuje a drží jejich tvar. Na stropě výtahu je ale hlavně umístěna celá strojovna, která se skládá ze dvou motorů a převodů (viz bod 6.2.2).

6.1.4 Podložka výtahu

Podložka tvoří spodní část celého modelu. Je vyrobena z dřevěné spárovky tloušťky 18mm. Dřevěný materiál byl použit pro usnadnění opracování, protože v podložce je vyřezáno několik výřezů. Povrchovou úpravu tvoří barva imitující hliník.

6.2 Motory

V modelu jsou použity dva typy motorů. V obou případech se jedná o DC motory vybavené převodovkou. Použití převodovky si vyžaduje účel, pro jaký jsou motory použity, a sice potřeba většího točivého momentu a méně otáček. Takové motory se samozřejmě dají koupit, ale jsou cenově výrazně dražší než vysokootáčkové DC motory s převodem, které mají stejné vlastnosti.

6.2.1 Motor pro zvedání kabiny

Motor pro pohánění kabin má označení RB-35 s převodem 1:100. Jedná se o tichý a velmi odolný motor v precizním provedení.



Obr. 17 Motor RB-35 s převodcem 1:100 [10]

Technická data motoru RB-35 s převodem 1:100

- Jmenovité napětí: 12 V DC
- Odběr proudu naprázdno: cca 80 mA
- Průměr motoru: 35 mm, průměr převodovky: 37 mm
- Délka motoru s převodovkou: 51 mm
- Počet otáček naprázdno: 60 ot./min.
- Počet otáček při zatížení: 52 ot./min.
- Odběr proudu při zatížení: 350 mA
- Trvalý točivý moment: 6 kg/cm (krátkodobě 18 kg/cm)
- Účinnost: 66 %

6.2.2 Motory pro otevírání a zavírání dveří

Označení motoru pro ovládání dveří je GM12-N20VA. Jeho hlavní výhodou je jeho velikost. Jedná se o velmi malý motorek ovšem díky převodovce v poměru 1:72 i s velkým točivým momentem.



Obr. 18 Motor
GM12-N20VA [10]

Technická data motoru GM12-N20VA

- Jmenovité napětí: 5 V DC
- Odběr proudu naprázdno: cca 80 mA
- Průměr motoru: 12 mm
- Délka motoru s převodovkou: 25 mm
- Počet otáček naprázdno: 45 ot./min.
- Počet otáček při zatížení: 35 ot./min.
- Odběr proudu při zatížení: 200 mA
- Trvalý točivý moment: 2,6 kg/cm

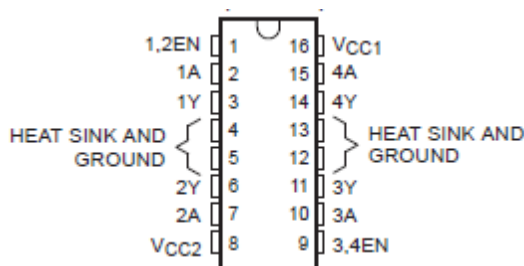
Pro představu velikosti jsou oba obrázky motorů (Obr. 17 a Obr. 18) ve skutečné velikosti. Potřeba malého motoru je z důvodu omezeného prostoru v konstrukci dveří (viz 6.3)

6.2.3 Ovládání motorů

Vyřešit otázku ovládání motorů byla jedna ze stěžejních otázek celé úlohy. Oba typy motorů totiž vyžadovaly možnost změny směru otáčení (dveře bylo potřeba otevřít ale i zavřít a kabina jezdí nahoru nebo dolů). Změnu směru otáčení DC motoru dosáhneme změnou polarity napájecího zdroje motoru. To samozřejmě v modelu nešlo realizovat mechanickou cestou.

Změnu směru otáčení motoru lze však také docílit i elektronickou cestou. Slouží k tomu obvod nazývaný H-můstek. V modelu je tento můstek použitý v provedení integrovaného obvodu (IO) s označením L293D.

L293D je dvoukanálový H-můstek, což znamená, že je možné k němu připojit dva motory a nezávisle na sobě je řídit. Pro řízení jednoho motoru, připojeného k jednomu kanálu jsou k dispozici tři vstupy a dva výstupy. Označení jednotlivých pinů je zobrazeno na Obr. 19.



Obr. 19 H-můstek L293D

Piny na levé straně IO jsou vstupy a výstupy prvního kanálu a na pravé straně pak druhého kanálu. Piny označené xA jsou vstupní piny a ovládají směr otáčení motoru (viz Tab. 8), xY výstupní piny. Třetím zmiňovaným vstupem je pin 1,2EN a 3,4EN. Tyto vstupy ovládají chod celého motoru (log. 1 otáčení, log. 0 stop). Je však možné pomocí nich řídit rychlost otáčení každého motoru pomocí pulzně šířkové modulace. Modulace může mít frekvenci až 20kHz. Další funkcí těchto vstupů je brzdění motorů. Zatímco při změně obou vstupů xA na stejnou úroveň, za předpokladu že xEN není na úrovni log. 0, dojde k okamžitému zastavení, při sestupné hraně signálu na xEN motor zastavuje pozvolna. Piny č. 8,16 a 4,5,12,13 slouží pro napájení. Pin 16 slouží k napájení IO, které je +5V, zatímco pin 8 je napájecí napětí pro motory v rozmezí +4,5 až +35V. Potenciál těchto napětí je oproti společné zemi připojené k pinům 4,5,12,13, kteří zároveň slouží jako odvod tepla z obvodu. Maximální proudový odběr pro jeden kanál je 0,6A.

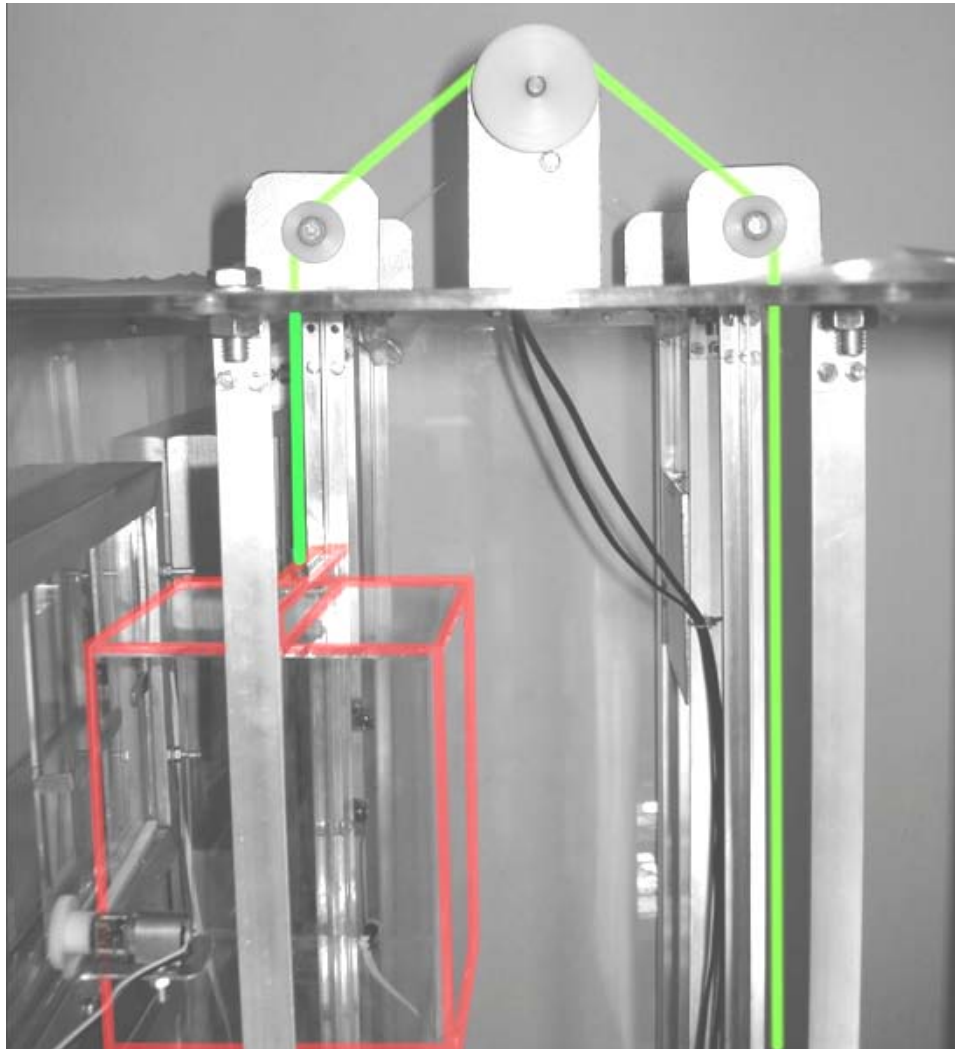
Vstupy		Funkce
1A, 3A	2A, 4A	
0	0	motor stojí (rychlá brzda)
0	1	otáčení vpravo
1	0	otáčení vlevo
1	1	motor stojí (rychlá brzda)

Tab. 8 Pravdivostní tabulka pro ovládání motoru pomocí H-můstku

6.2.4 Mechanické převody

Oba typy motorů popsaných v bodě 6.2.1 a 6.2.2 jsou součástí pohonných mechanismů.

Pohonný mechanismus kabin je zobrazen na Obr. 20.



Obr. 20 Pohonný mechanismus kabin

Skládá se ze tří kol, která vedou nosné lanko (na obrázku zvýrazněno zeleně) upevněné na střeše kabiny (na obrázku červeně). Na druhém konci lanka je upevněno protizávaží (není na obrázku). Na hřídeli motoru RB-35 s převodem 1:100 je nasazeno největší kolečku převodu (na obrázku úplně nahoře).

Pohonný mechanismus dveří je zobrazen na Obr. 21. a skládá se z pastorku motoru a hřebenového převodu (na obrázku zvýrazněn zeleně) upevněného k jednomu ze dvou křídel dveří (na obrázku červeně). Dveře a způsob jejich ovládní jsou podrobněji popsány v následujícím bodě.



Obr. 21 Pohonný mechanismus dveří

6.3 Dveře

Jeden z požadavků na vypracování modelu bylo vytvoření aktivních dveří. To znamená, že dveře nebudou pouze doplňovat konstrukci modelu, ale bude možné je otevírat a zavírat.

6.3.1 Konstrukce dveří

Dveře jsou zobrazeny na Obr. 22 a tvoří, stejně jako v předloze pro model, dvě prosklená křídla, která se při otevírání vysouvají do strany.



Obr. 22 Dveře zabudované v modelu

Tato křídla jsou osazena v dřevěném rámu s kovovým pojezdem. K jednomu z křídel dveří je připevněn hřebenový převod poháněný motorem. Tím je převáděn pohyb (rovnoměrný po kružnici) motoru na pohyb (rovnoměrný přímočarý) dveří. (Obr. 21). Toto křídlo je také osazeno kovovým háčkem a tím přenáší svůj pohyb na druhé křídlo a tím je zajištěno, že se pohybují obě křídla. Důležitou částí dveří jsou koncové body, jejichž signál se využívá pro ovládání v logickém obvodu.

6.3.2 Logický obvod pro ovládání dveří

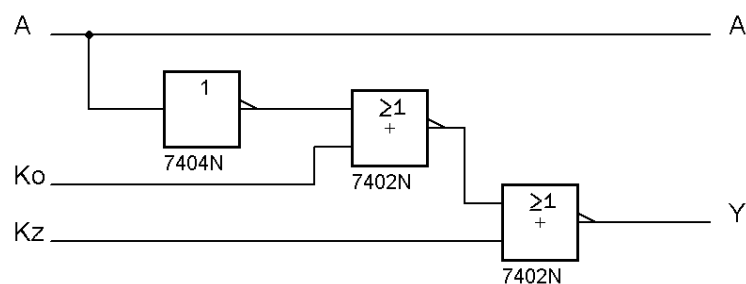
Motory dveří jsou řízeny H-můstkem (viz 6.2.3) pomocí tří vstupů. Vstupy H-můstku xEN jsou trvale připojeny k log. 1, protože není potřeba řídit rychlost otáčení. Další dva vstupy jsou ale proměnné a řídí směr otáčení. V modelu je deset dveří a proto by jen na dveře bylo potřeba dvacet číslicových výstupů z PLC. A pokud by se uvažovalo i s kontrolou stavu dveří (jestli jsou otevřeny nebo zavřeny), tak by bylo zapotřebí ještě dalších dvacet číslicových vstupů do PLC. Cílem proto bylo vymyslet způsob otevírání a zavírání pomocí jediného výstupního signálu z PLC (log. 0 zavřeno, log. 1 otevřeno) a kontrolu otevření popř. zavření ignorovat. Pro lepší pochopení záměru si představme následující situaci. PLC vyšle signál úrovně log. 1 pro otevření dveří. Jeden vstup H-můstku je nastaven na log. 1 přímo z PLC a druhý, který musí mít dle pravdivostní tabulky H-můstku (Tab. 8) opačnou hodnotu, se automaticky nastaví na tuto hodnotu (PLC se o to nestará). PLC nebude kontrolovat, jestli se dveře otevřely nebo se otevírají, nicméně nemůže nastat situace, že dveře dojedou do koncového bodu a motor stále pojede. Proto se musí druhý vstup H-můstku nastavit pro změnu na stejnou hodnotu, aby se motor zastavil. Takže nastane situace, kdy jsou dveře plně otevřeny, výstup PLC je nastaven na log. 1, ale motor stojí a vice versa. Bylo proto potřeba navrhnout takový logický obvod, který by svým chováním zajišťoval správný chod dveří.

Logický obvod pro ovládání dveří má tři vstupy, a sice jeden z PLC pro otevírání a zavírání a další dva od koncových bodů dveří a dva výstupy, které se rovnou připojují na vstupy H-můstku.

Logická funkce minimalizovaná pomocí metody Karnaughových map

$$y = \overline{(\overline{A} + K_o)} + K_z$$

Realizace funkce pomocí hradel NOR je na Obr. 23.



Obr. 23 Schéma logického obvodu pro ovládání dveří

Funkce je realizována pomocí méně často používaných hradel NOR, protože v tomto případě je stupeň minimalizace větší, než při použití hradel NAND. Pravdivostní tabulka funkce je v Tab. 9.

vstup H-místku 1A, 3A výstup PLC	koncový bod dveří pro otevřeno	koncový bod dveří pro zavřeno	vstup H-místku 2A, 4A	činnost motorů
A	Ko	Kz	Y	
0	0	0	1	Zavírá
0	0	1	1	Stojí
0	1	0	1	Zavírá
0	1	1	0	Stojí
1	0	0	0	Otevírá
1	0	1	0	Otevírá
1	1	0	1	Stojí
1	1	1	0	Otevírá

Tab. 9 Pravdivostní tabulka logické funkce pro ovládání dveří

6.4 Čidla polohy

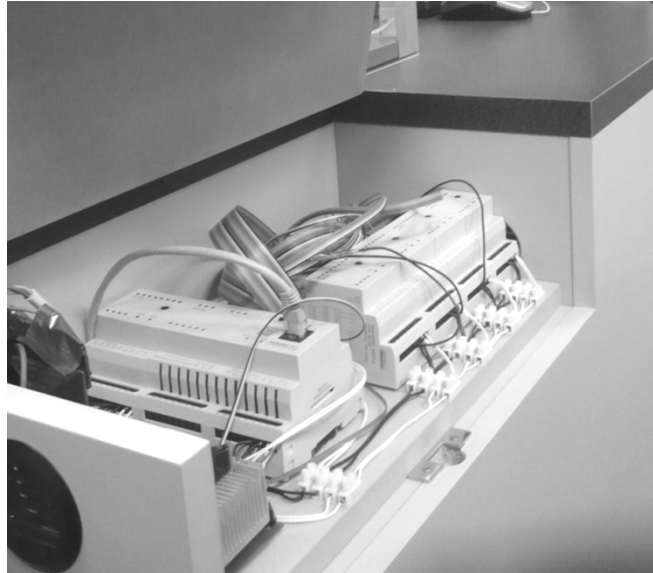
Pro určení polohy kabin výtahu slouží čidlo polohy tvořené IR diodou a fototranzistorem citlivým na IR záření. IR dioda je upevněna na spodní části kabiny a fototranzistor pak na každém podlaží. Při projetí kabiny podlažím se daný fototranzistor napojený na vstup PLC otevře a generuje signál log. 1.

6.5 Pracoviště

Celý model má být součástí pracoviště, které bude dále obsahovat PLC a počítač pro jeho programování. Pro pracoviště byl zhotoven návrh stolu (Obr. 13) podle kterého byl také odbornou firmou zhotoven.

6.5.1 Uložení PLC

Pro PLC byla ve stole vyrobena uzamykatelná skříňka (Obr. 24). Skříňka je zapuštěna v desce stolu a z vnitřní strany pod deskou má výřez pro přivedení kabeláže. Ta je z modelu vyvedena výřezem v podložce a v pracovní desce pod stůl.



Obr. 24 PLC zabudované ve stole

Skříňka má odvětrávací otvory protože mimo PLC je uvnitř uložen také napájecí zdroj, který je tvořen transformátorem a modulem s usměrňovačem a stabilizátory napětí.

7 DOKUMENTACE MODELU

Tato dokumentace slouží jako pomůcka nejen pro programování, ale také jako návod, jak s modelem pracovat. Některé pasáže již byly v práci popsány a budou se opakovat, nicméně tato kapitola bude samostatně vytištěna a přiložena k modelu.

7.1 Adresy vstupů a výstupů

		Svorka	Absolutní adresa podle IEC	Význam
Základní modul CP-1005		B4	r0_p3_D1.DI0	Rezervováno pro panel tlačítek kabin
		B5	r0_p3_D1.DI1	
		B6	r0_p3_D1.DI2	
		B7	r0_p3_D1.DI3	
		B8	r0_p3_D1.DI4	
		B9	r0_p3_D1.DI5	Rezervováno pro panel tlačítek kabin
		D2	r0_p3_D0.D00	
		D3	r0_p3_D0.D01	
		D4	r0_p3_D0.D02	
		D7	r0_p3_D0.D03	
		D8	r0_p3_D0.D04	nevyužito
		D9	r0_p3_D0.D05	
		B2	r0_p3_A00	nevyužito
		B3	r0_p3_A01	nevyužito
Binární výstupy OS-1401 Adresa: 0		B2	r1_p0_D0.D04	LED 2NP nahoru
		B3	r1_p0_D0.D05	LED 1NP
		B4	r1_p0_D0.D06	LED 2NP dolů
		B5	r1_p0_D0.D07	LED 3NP dolů
		B6	r1_p0_D0.D08	LED 3NP nahoru
		B7	r1_p0_D0.D09	LED 4NP dolů
		B8	r1_p0_D0.D010	LED 4NP nahoru
		B9	r1_p0_D0.D011	LED 5NP
		A6	r1_p0_D0.D00	Rezervováno pro panel tlačítek kabin
		A7	r1_p0_D0.D01	
A8	r1_p0_D0.D02			
Binární vstupy IB-1301 Adresa: 1		B2	r1_p1_D1.DI4	Podlaží 3NP pravé
		B3	r1_p1_D1.DI5	Podlaží 4NP pravé
		B4	r1_p1_D1.DI6	Podlaží 5NP pravé
		B5	r1_p1_D1.DI7	Podlaží 1NP levé
		B6	r1_p1_D1.DI8	Podlaží 2NP levé
		B7	r1_p1_D1.DI9	Podlaží 3NP levé
		B8	r1_p1_D1.DI10	Podlaží 4NP levé
		B9	r1_p1_D1.DI11	Podlaží 5NP levé
		A6	r1_p2_D1.DI0	Rezervováno pro panel tlačítek kabin
		A7	r1_p2_D1.DI1	
A8	r1_p2_D1.DI2			
		A9	r1_p2_D1.DI3	Podlaží 2NP pravé

Obr. 25 Přiřazení adres PLC - část první

		Svorka	Absolutní adresa podle IEC	Význam
Binární vstupy IB-1301 Adresa: 2		B2	r1_p2_D1.DI4	Tlačítko 5NP
		B3	r1_p2_D1.DI5	Tlačítko 4NP nahoru
		B4	r1_p2_D1.DI6	Tlačítko 4NP dolů
		B5	r1_p2_D1.DI7	Tlačítko 3NP nahoru
		B6	r1_p2_D1.DI8	Tlačítko 3NP dolů
		B7	r1_p2_D1.DI9	Tlačítko 1NP
		B8	r1_p2_D1.DI10	Tlačítko 2NP dolů
		B9	r1_p2_D1.DI11	Tlačítko 2NP nahoru
		A6	r1_p2_D1.DI0	Rezervováno pro panel tlačítek kabin
		A7	r1_p2_D1.DI1	
A8	r1_p2_D1.DI2			
Binární výstupy OS-1401 Adresa: 3		B2	r1_p3_DO.DO4	Motor 4A pravý
		B3	r1_p3_DO.DO5	Motor 3A pravý
		B4	r1_p3_DO.DO6	PWM pravý
		B5	r1_p3_DO.DO7	PWM levý
		B6	r1_p3_DO.DO8	Motor 1A levý
		B7	r1_p3_DO.DO9	Motor 2A levý
		B8	r1_p3_DO.DO10	Dveře 1NP pravé
		B9	r1_p3_DO.DO11	nevyužito
		A6	r1_p3_DO.DO0	nevyužito
		A7	r1_p3_DO.DO1	Dveře 2NP pravé
Binární výstupy OS-1401 Adresa: 4		A8	r1_p3_DO.DO2	nevyužito
		A9	r1_p3_DO.DO3	nevyužito
		B2	r1_p4_DO.DO4	Dveře 3NP pravé
		B3	r1_p4_DO.DO5	Dveře 4NP pravé
		B4	r1_p4_DO.DO6	Dveře 5NP pravé
		B5	r1_p4_DO.DO7	Dveře 1NP levé
		B6	r1_p4_DO.DO8	Dveře 2NP levé
		B7	r1_p4_DO.DO9	Dveře 3NP levé
		B8	r1_p4_DO.DO10	Dveře 4NP levé
		B9	r1_p4_DO.DO11	Dveře 5NP levé
A6	r1_p4_DO.DO0	Rezervováno pro panel tlačítek kabin		
A7	r1_p4_DO.DO1			
A8	r1_p4_DO.DO2			
A9	r1_p4_DO.DO3	nevyužito		

Obr. 26 Přiřazení adres PLC - část druhá

7.2 Ovládání motorů

Hlavní motory kabin se ovládají pomocí tří výstupů pro každý ze dvou motorů. Dvě dvojice výstupů a sice 1A na adrese r1_p3_DO.DO8 a 2A na adrese r1_p3_DO.DO9 pro levý motor a 3A na adrese r1_p3_DO.DO5 a 4A na adrese r1_p3_DO.DO4 pro pravý motor, řídí směr otáčení podle pravdivostní tabulky Tab. 10.

Vstupy		Funkce
1A, 3A	2A, 4A	
0	0	motor stojí (rychlá brzda)
0	1	otáčení vpravo
1	0	otáčení vlevo
1	1	motor stojí (rychlá brzda)

Tab. 10 Pravdivostní tabulka pro ovládání motorů

Rychlost otáčení levého motoru řídí výstup na adrese r1_p3_DO.DO7 a pravého motoru r1_p3_DO.DO6. Rychlost se řídí metodou pulzně šířkové modulace s frekvencí až do 20kHz.

7.3 Údržba a poruchy modelu

V případě poruchy ve vnitřním prostoru modelu je jedna stěna výtahu odstranitelná pomocí šroubů. V případě potřeby lze pak demontovat i další stěny.

7.3.1 Převodní mechanismus kabin

Pokud nastane problém s pohonem kabin, tak je nejprve potřeba překontrolovat převodový mechanismus na střeše výtahu. Zejména pak jsou-li lana ve vodících kolečkách. V případě prasknutí lana je potřeba je vyměnit.

7.3.2 Dveře

Dveře se mohou čas od času zasekávat při otevírání nebo zavírání. Pokud se tak děje, tak je nutné nejdříve zastavit motor. Buď vypnutím napájecího napětí, nebo ručním posunutím dveří do koncové polohy. Pak je potřeba zkontrolovat, není-li ve vodících kolejnicích nějaký předmět. Dále může být problém způsoben špatnou polohou motoru dveří (pastorek motoru nedoléhá správně na hřebenový převod dveří). V tomto případě lze polohu motoru nastavovat povolením úchytných šroubů. Pokud problém s dveřmi dále přetrvává, tak je vhodné dveře odpojit a demontovat a do doby než bude problém odstraněn je nepoužívat. Dlouhodobý chod motoru při zaseknutých dveřích může vést k jeho spálení nebo k vylovení zubů na hřebenovém převodu.

7.3.3 Kabina výtahu

Kolejnice kabiny jsou upevněny v podložce a ve stropu výtahu. Při manipulaci uvnitř prostoru se může nedopatřením s jednou kolejnicí pohnout. Špatná poloha kolejnice se projevuje zejména zaseknutím kabiny na místě (motor běží, ale kabina se nepohybuje). **Pozor! Stojící kabina v krajních polohách (ne v přízemí, ale úplně na podložce a ne v pátém podlaží, ale úplně u stropu) při chodu motoru není porucha!** Jedná se o bezpečnostní prvek, kdy se motor nezastaví v prvním nebo posledním podlaží a jede dál až do absolutních krajních poloh. Napětí v nosných lanech se v těchto krajních bodech uvolní a motor se otáčí naprázdno. V jiných polohách kabiny, kdy se motor otáčí a kabina se nepohybuje se, však jedná o poruchu a lze jí odstranit srovnáním kolejnic. Jedna z kolejnic zaseknuté kabiny se u stropu povolí, potáhne směrem ven a opět přišroubuje. Také je dobré kolejnice srovnat tak, aby s podložkou svíraly pravý úhel. Správnou polohu kolejnic pak lze ověřit ručním posunem kabiny přes všechna podlaží. Kabina se nesmí zadržávat a musí jet hladce.

7.3.4 Elektrické obvody

V případě, že jsou všechny mechanické prvky v pořádku, ale kabina nebo dveře nepracují tak jak mají, může být závada v elektronických prvcích modelu. Uvnitř skříňky, kde je umístěno PLC se nachází i napájecí zdroj. Pokud při zapnutém stavu nesvítí LED kontrolky na PLC, tak zkontrolujte pojistku. Nachází se u přívodního kabelu z elektrické sítě. **Pozor! Kontrolu pojistky provádějte pouze při odpojeném přívodním kabelu!** Pokud je však pojistka v pořádku, tak může být další problém v napájecích napětích. Pomocí voltmetru změřte napětí na výstupních svorkách zdroje. Hodnoty napětí jsou u každé svorky uvedeny. Pokud ani zde není problém, tak je potřeba zkontrolovat kabeláž, zejména pak v místech, kde je upevněna ve svorkovnicích.

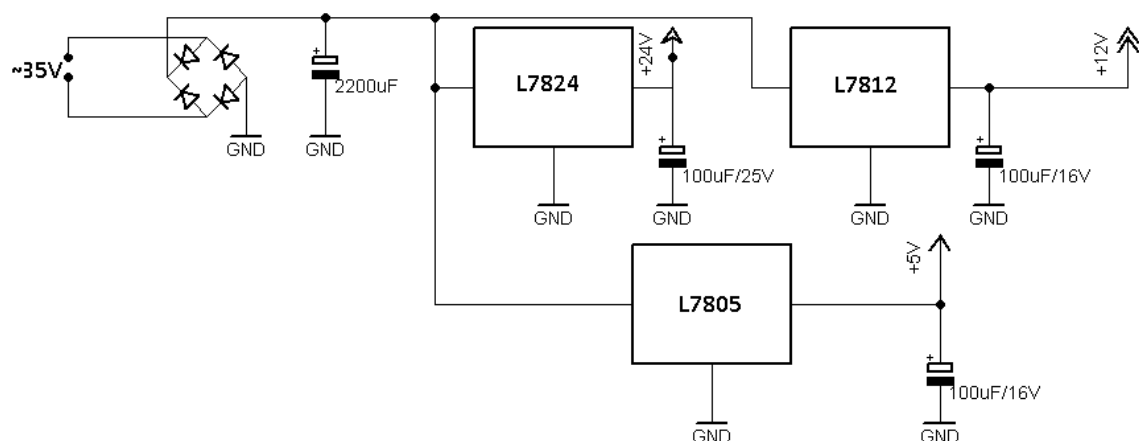
Dalším problémem může být vadná součástka v obvodu pro ovládání dveří a motorů. Pokud nefungují hlavní motory a kabeláž i napájecí napětí jsou v pořádku, tak může být vadný IO s označením L293D. Na desce plošného spoje se těchto součástek nachází celkem šest. Pět z nich ovládá dveře a jsou seřazeny v řadě. Šestá ovládá motory, je mimo řadu a je doplněna skupinou odporů. Možným řešením je zaměnit součástku za jednu z těch, které slouží pro otevírání dveří. Pokud se problém vyřeší, bude potřeba součástku vyměnit. Pokud ne, tak může být problém ve vadném motoru. Ten je možné odzkoušet externím zdrojem +12V, které přivedeme na svorky motoru. Tuto zkoušku však provádějte

pouze při vypnutém napájecím napětí modelu. U motoru dveří může nastat stejný problém se součástkami. Zde jsou navíc ze spodní strany součástky logického obvodu a test jejich funkčnosti můžete provést stejným způsobem jako u motorů. Pozor však na zaměnění součástek. Nachází se zde dva druhy integrovaných obvodů. Tato závada se navíc projevuje vyřazením dvou nebo čtyř dveří. Pokud nefungují pouze dvoje dveře a ostatní ano, vyměňte součástku s označením M74HC02. Pokud jich nefunguje více, tak součástku s označením M74HC04.

7.4 Schémata elektrických obvodů

7.4.1 Zdroj napětí

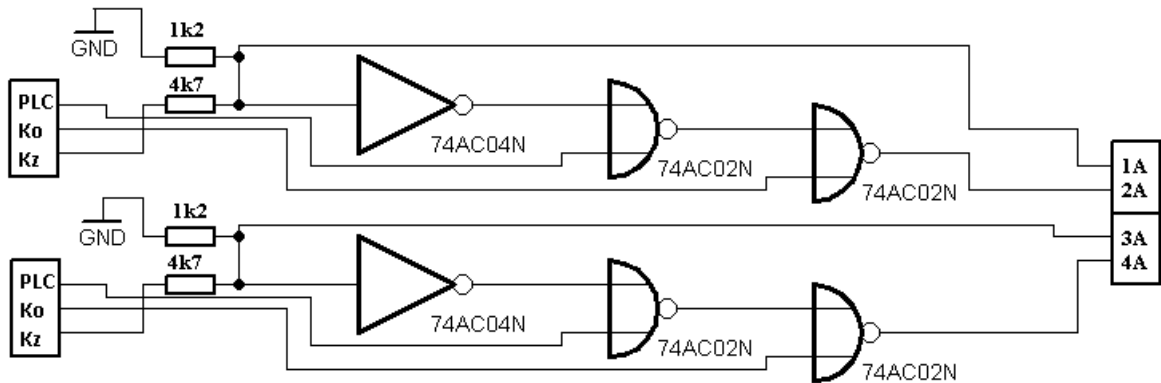
Zdroj napětí tvoří můstkový usměrňovač, filtrační kondenzátor 2200 $\mu\text{F}/53\text{V}$ a trojice stabilizátorů napětí pro 5, 12 a 24V s podpurnými kondenzátory. Všechny tři pro proudovou zátěž do 2A.



Obr. 27 Elektrické schéma zapojení zdroje napětí

7.4.2 Logický obvod pro ovládání dveří

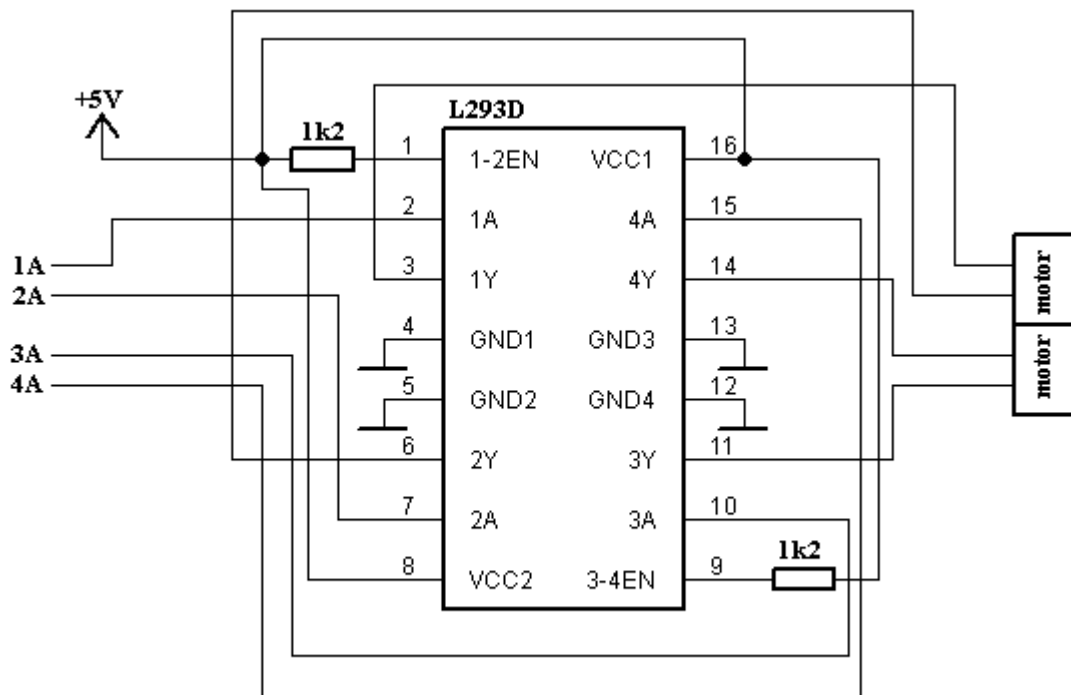
Logický obvod je složen z hradel IO M74HC02 a M74HC04. M74HC02 je čtveřice NOR hradel a IO M74HC04 je šestice negací. V obvodu je celkem 5x M74HC02 a 3x M74HC04. Schéma (Obr. 28) znázorňuje zapojení pro dvoje dveře. Odporů na vstupech od PLC tvoří odporový dělič. Z úrovně $+24\text{V}$ z PLC pro log. 1 vytvoří $+5\text{V}$ pro log. 1 TTL.



Obr. 28 Elektrické schéma zapojení logického obvodu pro ovládání dveří.

7.4.3 H-můstek pro ovládání dveří

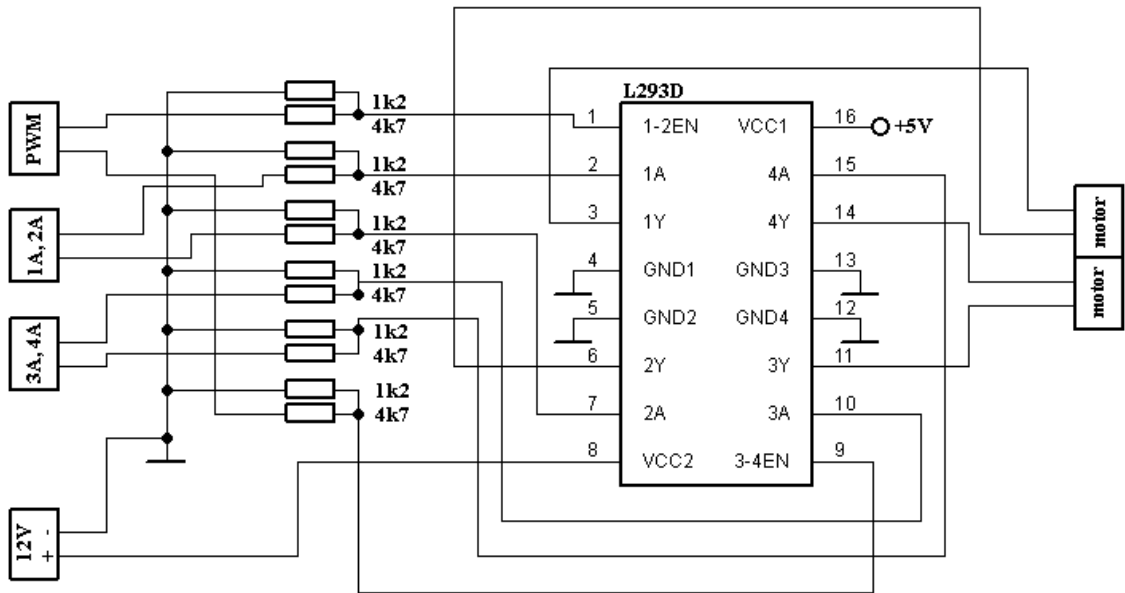
Výstupy z předchozího obvodu jsou přivedeny na vstupy H-můstku. Pro ovládání dveří jsou vstupy pro ovládání rychlosti otáčení trvale připojeny přes odpor 1k2 k log. 1.



Obr. 29 Elektrické schéma zapojení H-můstku pro ovládání dveří

7.4.4 H-můstek pro motory kabiny

Zapojení H-můstku pro motory kabiny se od zapojení pro ovládání dveří liší v tom, že nemá předstupeň tvořený logickým obvodem, ale na jeho vstupy jsou rovnou připojeny přes odporové děliče výstupy z PLC.



Obr. 30 Elektrické schéma zapojení H-můstku pro ovládání kabin

8 PROGRAMOVÁ ČÁST

V programové části bude popsán program pro ukázkou funkcí výtahu. Nejedná se proto o komplexní program pro ovládání výtahu, ale o program, který umožňuje ruční nastavení všech dostupných prvků skrze prostředí vizualizační aplikace Control Web 6. Funkční bloky popsané v dalším bodě lze využít pro vytvoření komplexního programu pro řízení výtahu.

8.1 Ukázkový program

Program je tvořen několika funkčními bloky a funkcemi jejichž některé vstupní proměnné jsou propojeny s Control Webem a umožňují tak jejich ovládání jiným programem. A protože je jazyk strukturovaného textu jazykem objektovým, napsal jsem funkční blok pro každý prvek, který se v modelu nachází.

8.1.1 Funkční blok dveří

Obsahuje funkci pro otevření a zavření dveří. Parametry předávané funkční bloku jsou adresa fyzického vstupu PLC předávaná jako ukazatel a řídicí proměnná typu boolean



Obr. 31 Funkční blok dveří – vstupy

Dveře mohou mít dva režimy ovládání. Obyčejný režim, kdy jsou dveře otevírány a zavírány podle proměnné *stav* : **BOOL** a nebo režim automatického zavření. Aktivní režim automatického zavření obsahuje časovač TOF, který změní proměnnou *stav* : **BOOL** 10 sekund od vzestupné hrany na úroveň log. 0 a tím dveře zavře. Režim automatického zavírání dveří je aktivní na základě podmíněných direktiv. Ty jsou v programu jako ukázkou jejich využití, protože by se zapnutí nebo vypnutí automatického režimu dalo řešit efektivněji použitím větvení IF, ELSIF a R_TRIG.

```

FUNCTION_BLOCK fbDvere
({$DEFINE automatickeZaviraniDveri} /**
VAR_INPUT
ptr_adresaDveri : PTR_TO BOOL;
({$IFDEF automatickeZaviraniDveri}
ovladani : BOOL R_EDGE;
{$ELSE}
ovladani : BOOL;
{$END_IF}
END_VAR
VAR
casovacZavreni : TOF;
END_VAR
VAR_OUTPUT
stav : BOOL;
END_VAR
VAR_TEMP
END_VAR

({$IFDEF automatickeZaviraniDveri}
casovacZavreni( IN := ovladani, PT :=T#10s, Q => stav );
ptr_adresaDveri^ := stav;
{$ELSE}
ptr_adresaDveri^ := ovladani;
stav := ovladani;
{$END_IF}

-
END_FUNCTION_BLOCK

```

Obr. 32 Funkční blok dveří - zdrojový kód

8.1.2 Funkční blok motoru

Tento blok ovládá hlavní motory výtahu. Jako parametr přebírá směr chodu kabiny 'N' - nahoru, 'D' - dolu, 'S' - stop. Dále přebírá rychlost otáčení, která je ovládaná pomocí pulzní šířkové modulace. Interval předávaného parametru je $< 0 - 10 >$ (* 10%)

Výstupní proměnné jsou pro motor dvě (Ix1 a Ix2), kde Ix1 odpovídá vstupu integrovaného obvodu řídicího motor (IO) označeného 1A nebo 3A a Ix2 odpovídá vstupu 2A nebo 4A. Dále pak jedna proměnná xEN pro řízení rychlosti otáčení.



Obr. 33 Funkční blok motoru – vstupy

Směr pohybu motoru je ovládán vstupem *smer* : *STRING*, který může obsahovat znak „S“ pro stání, znak „N“ pro jízdu nahoru (otáčení po směru hod. ručiček) a znak „D“ pro jízdu směrem dolů. Na základě této hodnoty se nastavují hodnoty výstupních proměnných *Ix1*, *Ix2* : *BOOL*. Proměnná předávaná do *stopStav* : *BOOL* je globální proměnná, která může být nastavena v jakémkoliv místě programu a která znamená okamžité zastavení otáčení motoru neohledně na proměnnou *smer* : *STRING*.

Proměnná *rychlost* : *REAL* slouží jako nastavení rychlosti pro pulzní šířkovou modulaci, která ovládá rychlost otáčení. V programu jsou definovány tři rychlosti jako konstanty označeny jako *M_rychlostKabiny_x*, kde *x* může být *rychlá* a odpovídá 100% rychlosti, *normální* odpovídá 30% rychlosti a *pomalá* odpovídá 10% rychlosti.

```

FUNCTION_BLOCK fbMotor
VAR_INPUT
    stopStav : BOOL;
    smer : STRING;
    rychlost : REAL := M_rychlostKabiny_Rychla; //konstanta
END_VAR
VAR
    i : REAL := 0;
END_VAR
VAR_OUTPUT
    Ix1 : BOOL;
    Ix2 : BOOL;
    xEN : BOOL;
END_VAR
VAR_TEMP
END_VAR

//ovladani smeru otaceni
IF stopStav OR smer = 'S' THEN
    Ix1 := false;
    Ix2 := false;
ELSIF NOT stopStav AND smer = 'D' THEN
    Ix1 := true;
    Ix2 := false;
ELSIF NOT stopStav AND smer = 'N' THEN
    Ix1 := false;
    Ix2 := true;
END_IF;

//ovladani rychlosti otaceni
IF i <= rychlost THEN xEN := true;
ELSE xEN := false;
END_IF;
i := i + 0.5;
IF i > M_rychlostKabiny_Rychla THEN i := 0.0;
END_IF;
END_FUNCTION_BLOCK

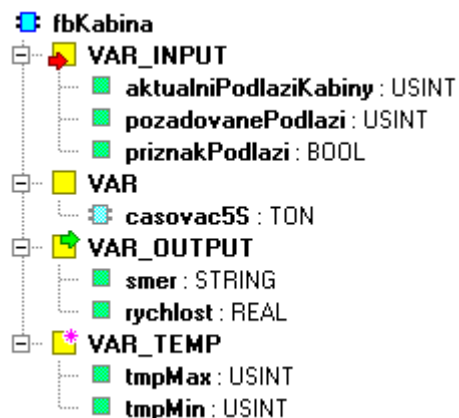
```

Obr. 34 Funkční blok motoru – zdrojový kód

Frekvence pulzně šířkové modulace pro ovládání rychlosti motoru lze nastavit změnou iterační proměnné i : *REAL*. Ta je teď nastavena na hodnotu 20 Hz (maximální rychlost $10.0 * \text{hodnota proměnné } i * 0.5$), což je nejnižší možná frekvence pulzně šířkové modulace, kterou lze přivést na vstup IO pro řízení otáček. I s touto hodnotou je pohyb kabiny krásně plynulý.

8.1.3 Funkční blok kabiny

Podle požadovaného podlaží předávaného jako vstupní parametr *požadovanePodlazi* : *UINT* přijede kabina do daného podlaží automaticky. Funkční blok sám rozhoduje na základě aktuální pozice kabiny, kterým směrem se kabina rozjede.



Obr. 35 Funkční blok motoru –
vstupy

Důležitou vlastností kabiny výtahu, která by měla fungovat i u skutečných výtahů, je přijetí do přízemí po obnově dodávky elektrického proudu. Program totiž obsahuje globální proměnné uložené v remanentní zóně PLC, které neustále obsahují aktuální číslo podlaží, ve kterém se nachází kabina. Tyto proměnné mají inicializační hodnotu 0, takže po studeném restartu (který v reálu může znamenat obnovení dodávky elektrického proudu) musejí kabiny sjet do přízemí. Funkční blok proto nejdříve ověří, jakou hodnotu mají tyto globální proměnné. Pokud jsou nulové, tak se kabina rozjede směrem nahoru. Zároveň se aktivuje 5ti sekundový časovač. Pokud se ani po těchto pěti sekundách nezmění stav globální proměnné pro aktuální podlaží kabin, znamená to, že se kabina nachází v absolutním horním krajním bodě a rozjede se směrem dolů až do prvního podlaží. Požadované podlaží je uloženo také v globálních proměnných, ne však remanentních a které mají inicializační hodnotu právě 1 pro první podlaží.


```

VAR_INPUT
    aktualniPodlaziKabiny : USINT;
    pozadovanePodlazi : USINT;
    priznakPodlazi : BOOL;
END_VAR
VAR
    casovac5S : TON;
END_VAR
VAR_OUTPUT
    smer : STRING;
    rychlost : REAL;
END_VAR
VAR_TEMP
    tmpMax : USINT;
    tmpMin : USINT;
END_VAR

//zastaveni kabiny v poslednich patrech
IF aktualniPodlaziKabiny = 1 AND smer = 'D' THEN
    smer := 'S';
END_IF;
IF aktualniPodlaziKabiny = 5 AND smer = 'N' THEN
    smer := 'S';
END_IF;
//ovladani kabiny na zaklade pozadavku patra. Nejdrive je zkontrolovano, jestli
//aktualni pozice kabiny neni nastavena na 0, coz je pocatecni stav promenne
//po studenem restartu PLC, kdy neni mozno zjistit aktualni pozici kabiny
//a kabina se v tomto stavu nastavy na plnou rychlost a rozjede se smerem dolu
//po dobu 5 sekund. Pokud se ani po tuto dobu stav nezmeni, tak se kabina
//rozjede smerem nahoru. Pote uz lze vytah ovladat normalnim zpusobem.
IF aktualniPodlaziKabiny = 0 THEN
    casovac5S( IN := true, PT :=T#5s);
    rychlost := M_rychlostKabiny_Rychla;
    IF casovac5S.Q THEN
        smer := 'N';
    ELSE
        smer := 'D';
    END_IF;
ELSE
    RM_STOPSTAV := false;
    IF pozadovanePodlazi < aktualniPodlaziKabiny THEN
        smer := 'D';
    ELSIF pozadovanePodlazi > aktualniPodlaziKabiny THEN
        smer := 'N';
    ELSIF pozadovanePodlazi = aktualniPodlaziKabiny AND priznakPodlazi THEN
        smer := 'S';
    END_IF;
    //automaticke zpomalovani kabin priblizeni se cilovemu podlazi. Dve podlazi od
    //ciloveho zpomali kabina na 30%, a jedno podlazi od konecneho na 10%
    tmpMax := MAX(pozadovanePodlazi, aktualniPodlaziKabiny);
    tmpMin := MIN(pozadovanePodlazi, aktualniPodlaziKabiny);
    IF (tmpMax-tmpMin) = 2 THEN
        rychlost := M_rychlostKabiny_Normalni;
    ELSIF (tmpMax-tmpMin) <= 1 THEN
        rychlost := M_rychlostKabiny_Pomala;
    ELSE rychlost := M_rychlostKabiny_Rychla;
    END_IF;
END_IF;
END_FUNCTION_BLOCK

```

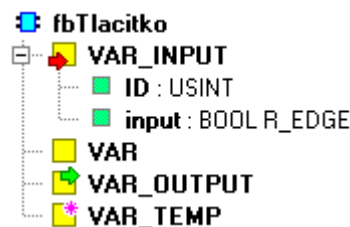
Obr. 36 Funkční blok kabiny - zdrojový kód

Po dobu, kdy se aktuální podlaží kabin rovná nule, je aktivován tzv. globální stop stav (Boolean), který je také uložen v remanentní zóně (protože když je v remanentní zóně aktuální podlaží, tak stop stav nemusí nastat vždy po každém restartu, pouze po studeném, protože proměnná pro stop stav má inicializační hodnotu True). Tato proměnná se pak může využít při úplném řízení výtahu, kdy výtah nepřijímá v tomto stavu žádné příkazy.

Funkční blok také reguluje rychlost. Pokud se kabina přiblíží k cílovému podlaží na dvě podlaží, zpomalí na 30% a na vzdálenost jednoho podlaží na 10%.

8.1.4 Funkční blok tlačítka

Funkční blok tlačítka na základě ID, které je přiřazeno každému reálnému tlačítku, volá funkci, která zařazuje požadavky (viz 8.1.5)



Obr. 37 Funkční blok
tlačítka – vstupy

```

FUNCTION_BLOCK fbTlacitko
VAR_INPUT
  ID : USINT;
  input : BOOL R_EDGE;
END_VAR
VAR
  cisloPatra : USINT;
END_VAR
VAR_OUTPUT
END_VAR
VAR_TEMP
END_VAR
IF input THEN
  CASE ID OF
(*1NP*)      1: fZaradPozadavek(pozadavek := '1');
(*2NP_D*)   2: fZaradPozadavek(pozadavek := '2');
(*2NP_N*)   3: fZaradPozadavek(pozadavek := '3');
(*3NP_D*)   4: fZaradPozadavek(pozadavek := '4');
(*3NP_N*)   5: fZaradPozadavek(pozadavek := '5');
(*4NP_D*)   6: fZaradPozadavek(pozadavek := '6');
(*4NP_N*)   7: fZaradPozadavek(pozadavek := '7');
(*5NP*)     8: fZaradPozadavek(pozadavek := '8');
  END_CASE;
END_IF;
END_FUNCTION_BLOCK

```

Obr. 38 Funkční blok tlačítka - zdrojový kód

8.1.5 Funkce pro zařazení a vyřazení požadavků

Funkce pro zařazení požadavků pracuje s globální proměnnou *M_pozadavky* : *STRING*. Ta uchovává číselný kód ID každého tlačítka na podlaží v pořadí, v jakém byly požadovány. Číselné kódy jsou definovány jako konstanty v paměti PLC. Funkce také zjišťuje, není-li již požadavek zařazen. Opačně pak funguje funkce pro vyřazení požadavku. Pole požadavků je typu *STRING* z důvodu snadné práce s tímto datovým typem (zejména funkce FIND, INSERT a DELETE).

```

FUNCTION fZaradPozadavek : BOOL
  VAR_INPUT
    pozadavek : STRING;
  END_VAR
  VAR
    k : INT;
  END_VAR
  VAR_TEMP
    M_pozadavky : STRING;
  END_VAR
  k := FIND(IN1 := M_pozadavky, IN2 := pozadavek );
  IF k = 0 THEN
    M_pozadavky := INSERT( IN1 := M_pozadavky, IN2 := pozadavek, P := LEN(M_pozadavky)+1 );
  END_IF;
END_FUNCTION

```

Obr. 39 Funkce pro zařazení požadavků

```

FUNCTION fVyradPozadavek : BOOL_
  VAR_INPUT
    pozadavek : STRING;
  END_VAR
  VAR
    k : INT;
  END_VAR
  VAR_TEMP
    M_pozadavky : STRING;
  END_VAR
  k := FIND(IN1 := M_pozadavky, IN2 := pozadavek );
  IF k <> 0 THEN
    M_pozadavky := DELETE( IN := M_pozadavky, L := 1, P := k );
  END_IF;
END_FUNCTION

```

Obr. 40 Funkce pro vyřazení požadavků

8.1.6 Funkce podsvícení tlačítek

Tlačítka, která se fyzicky nacházejí na modelu jsou podsvícená. Tato funkce periodicky kontroluje seznam požadavků, a zapíná nebo vypíná podsvícení daného tlačítka na základě toho, jestli je požadavek od daného tlačítka v seznamu nebo nikoliv.

```

FUNCTION fpodsviceniTlacitek : BOOL
VAR_INPUT
END_VAR
VAR
  i : USINT;
  podlazi : STRING;
  pozice : INT;
END_VAR
VAR_TEMP
END_VAR
FOR i:= 1 TO 8 DO
  podlazi := USINT_TO_STRING(i);
  pozice := FIND( IN1 := M_pozadavky, IN2 := podlazi );
  CASE i OF
    1: IF pozice = 0 THEN LED_1NP := false;
      ELSE LED_1NP := true;
      END_IF;
    2: IF pozice = 0 THEN LED_2NP_dolu := false;
      ELSE LED_2NP_dolu := true;
      END_IF;
    3: IF pozice = 0 THEN LED_2NP_nahoru := false;
      ELSE LED_2NP_nahoru := true;
      END_IF;
    4: IF pozice = 0 THEN LED_3NP_dolu := false;
      ELSE LED_3NP_dolu := true;
      END_IF;
    5: IF pozice = 0 THEN LED_3NP_nahoru := false;
      ELSE LED_3NP_nahoru := true;
      END_IF;
    6: IF pozice = 0 THEN LED_4NP_dolu := false;
      ELSE LED_4NP_dolu := true;
      END_IF;
    7: IF pozice = 0 THEN LED_4NP_nahoru := false;
      ELSE LED_4NP_nahoru := true;
      END_IF;
    8: IF pozice = 0 THEN LED_5NP := false;
      ELSE LED_5NP := true;
      END_IF;
  END_CASE;
END_FOR;
END_FUNCTION

```

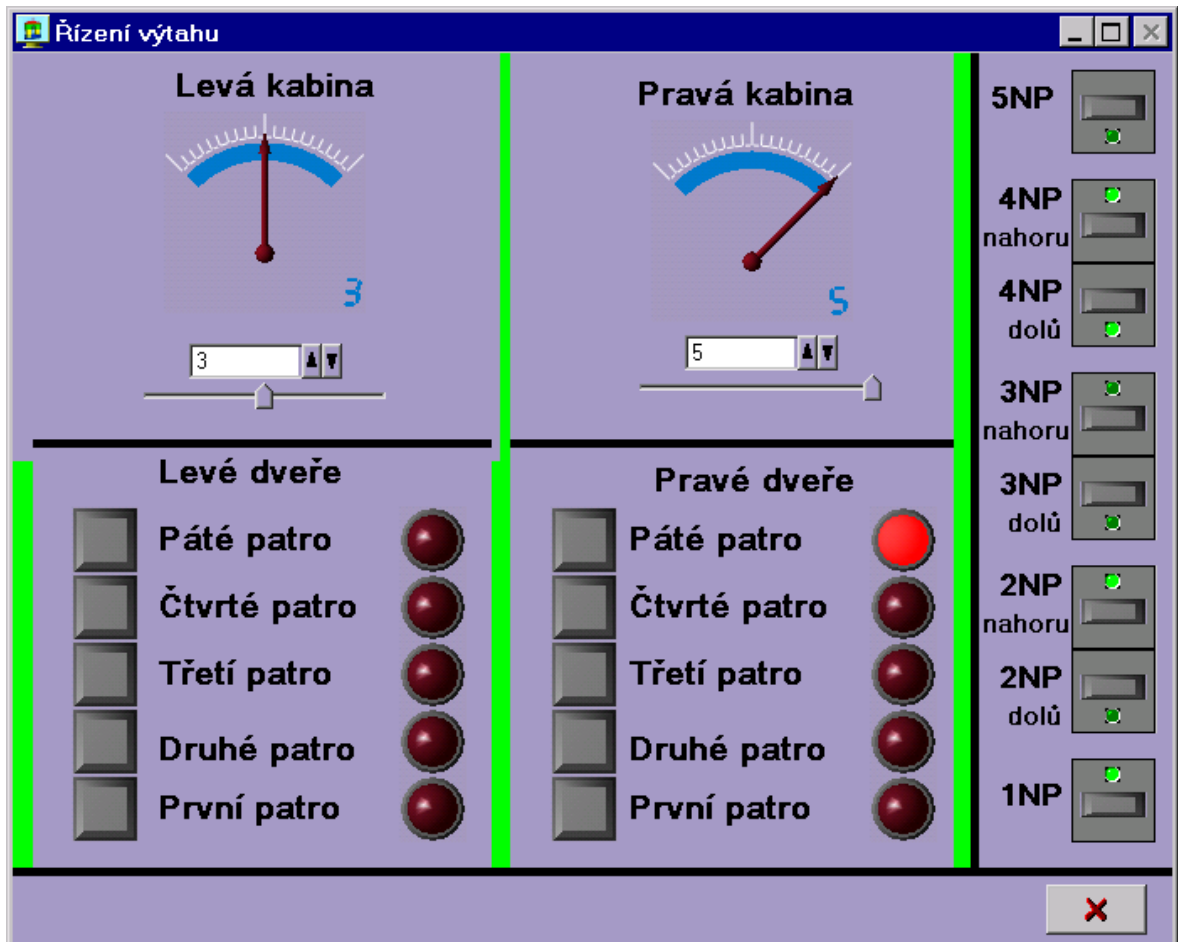
Obr. 41 Funkce podsvícení tlačítek

8.1.7 Ostatní funkce

Program obsahuje ještě další jednoduché funkce pro periodické nastavování aktuálního podlaží a funkci pro otevírání dveří.

8.2 Vizualizace

Vizualizace v prostředí Control Web 6 umožňuje pohodlně ovládat všechny nastavitelné prvky výtahu. V horní polovině okna programu se zobrazuje aktuální a nastavuje požadované podlaží. V dolní polovině se pak ovládají dveře. Červená kontrolka signalizuje stav dveří. Pravá krajní část pak obsahuje tlačítka pro každé podlaží.



Obr. 42 Okno vizualizačního programu

ZÁVĚR

Náplní této práce byl návrh a realizace modelu výtahu pro potřeby výuky předmětu programovatelné automaty. V prvním bodě teoretické části jsem se zabýval nejen základním popisem jednotlivých částí programovatelného automatu, ale i jeho historií. Také jsem popsal vlastnosti programovatelného automatu Tecomat Foxtrot, kterým je celý model řízen. Poslední část teorie pak nastínila problematiku sekvenčního řízení se zaměřením na ovládání výtahu a shrnula stěžejní body normy IEC EN 61 131-3. Praktická část pak popisuje nejpodstatnější otázky, které jsem musel řešit při realizaci navrženého modelu, které jsou dále doplněné fotkami pro jasnější představu o konečném řešení těchto otázek.

V úvodu této práce jsem mluvil o stále rostoucí potřebě člověka využívat programovatelné automaty. Člověk se vždy v historii snažil nahrazovat svou práci strojem a právě programovatelné automaty jsou tomu dalším důkazem. Navíc v případě programovatelných automatů se jedná o nahrazení mnohem více práce, než se na první pohled může zdát. Programovatelný automat například nahradí partu technologů, kteří by jinak museli sedět ve velíně a dohlížet na daný technologický proces, nebo se postarají o to, že ráno v sedm hodin zapnou v domě vytápění. Toto je zejména důvod, proč jsou programovatelné automaty tak rozšířené a tím je fakt, že se již dávno nejedná o nástroj v průmyslové výrobě, ale o nástroj, který usnadňuje činnosti člověka v každodenním životě.

Posláním této práce je proto především napomoci výchově nových programátorů programovatelných automatů tím, že jim je dán programovatelný nástroj, který se snaží co nejvíce napodobit překážky, které pak v pozdějším životě budou muset překonávat při řešení reálných úkolů.

ZÁVĚR V ANGLIČTINĚ

The main goal of presented Master thesis is design and realization of elevator model which will be mainly used for teaching of programmable logic controller programming. In the first paragraph of the theoretical part there are main parts of PLC described and also there is something about history of PLC. I also described qualities of PLC Tecomat Foxtrot which is used for the model control. Last part of theory is about sequential control focused on elevator model and also summarizes the key points of EN IEC 61 131 standard. Practical part of the Master thesis describes most important questions which had to be resolved when the model was implemented. There questions are also supplemented by pictures for a better understanding of the final resolution of these issues.

In the introduction of this thesis, I was talking about growing needs of human to use programmable logic controller. In history, there always have been efforts to replace human labor by machine. Moreover, in the case of programmable logic controllers is to replace a lot more work than at first glance may seem. For instance, PLC can replace a group of technologists who had to work at a control station or PLC can take care of heating the house at seven o'clock in the morning. This is main reason why are programmable logic controller so prevalent and it is fact, that it is not only tool in industrial production, but tool, that facilitates the work of humans in daily life.

The aim of this thesis is therefore primarily to assist training of new programmable logic controllers programmers, whom it was given a programmable tool. This tool tries to emulate as many obstacles that will later in life will have to overcome in addressing real challenges.

SEZNAM POUŽITÉ LITERATURY

- [1] CENDELÍN, Jiří. Historie programovatelných automatů a jejich současné efektivní použití. *Automa* [online]. 2004, roč. 3, č. 06 [cit. 2010-06-02]. Dostupný z WWW: <http://www.odbornecasopisy.cz/index.php?id_document=28831>
- [2] MARTINÁSKOVÁ, Marie, ŠMEJKAL, Ladislav. *Řízení programovatelnými automaty*. Praha: Vydavatelství ČVUT, 1998. 165 s.
- [3] MARTINÁSKOVÁ, Marie, ŠMEJKAL, Ladislav. *Řízení programovatelnými automaty II*. Praha: Vydavatelství ČVUT, 2000. 72 s.
- [4] MARTINÁSKOVÁ, Marie, ŠMEJKAL, Ladislav. *Řízení programovatelnými automaty III*. Praha: Vydavatelství ČVUT, 2003. 161 s.
- [5] TECOMAT : PLC for machine, process, technology, transport and building automation [online]. 2010 [cit. 2010-06-02]. *Produkty*. Dostupný z WWW: <<http://www.tecomat.com/index.php?a=cat.12>>.
- [6] TXV 003 21.01. *Programování PLC podle normy IEC 61 131-3 v prostředí Mosaic*.: Teco a.s., 2009. 103 s.
- [7] TXV 004 10.01. *Programovatelné automaty Tecomat Foxtrot*: Teco a.s., 2009. 185 s.
- [8] TXV 004 12.01. *Periferní moduly Tecomat Foxtrot*: Teco a.s., 2009. 101 s.
- [9] VRZAL, Ondřej. *Levitace balónku na vodním sloupci*. 2008. 48 s. Bakalářská práce. UTB ve Zlíně, Fakulta aplikované informatiky.
- [10] GM Electronic [online]. 2010 [cit. 2010-06-08]. *Servomechanismy*. Dostupné z WWW: <<http://www.gme.cz/cz/elektronicke-soucastky/servomechanismy/15072.html>>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

- PLC Programmable logic controller – anglický výraz pro programovatelný automat
- SPS Speicherprogrammierbare Steuerung – německý výraz pro programovatelný automat
- IO Integrovaný obvod
- DPS Deska plošného spoje

SEZNAM OBRÁZKŮ

Obr. 1. Kompaktní PLC.....	12
Obr. 2. Modulární PLC.....	13
Obr. 3 Připojení periferních modulů sběrnicí TCL2.....	17
Obr. 4 Základní modul CP-1005.....	20
Obr. 5 Modul IB-1301.....	21
Obr. 6 Připojení zátěží k binárním výstupům modulu OS-1401.....	23
Obr. 7 Připojení signálů k analogovým vstupům modulu IT-1602.....	24
Obr. 8 Ukázka programu v jazyce IL.....	32
Obr. 9 Ukázka programu v jazyce ST.....	34
Obr. 10 Společné prvky grafických jazyků.....	36
Obr. 11 Grafika obvodu v jazyce FBD.....	37
Obr. 12 Prvotní návrh.....	40
Obr. 13 Konečný návrh.....	41
Obr. 14 Způsob upevnění stěn.....	42
Obr. 15 Stěny výtahu.....	43
Obr. 16 Kovové konstrukce.....	44
Obr. 17 Motor RB-35 s převodem 1:100.....	45
Obr. 18 Motor GM12-N20VA.....	46
Obr. 19 H-můstek L293D.....	47
Obr. 20 Pohonný mechanismus kabin.....	48
Obr. 21 Pohonný mechanismus dveří.....	49
Obr. 22 Dveře zabudované v modelu.....	50
Obr. 23 Schéma logického obvodu pro ovládání dveří.....	51
Obr. 24 PLC zabudované ve stole.....	53
Obr. 25 Přiřazení adres PLC - část první.....	54
Obr. 26 Přiřazení adres PLC - část druhá.....	55
Obr. 27 Elektrické schéma zapojení zdroje napětí.....	58
Obr. 28 Elektrické schéma zapojení logického obvodu pro ovládání dveří.....	59
Obr. 29 Elektrické schéma zapojení H-můstku pro ovládání dveří.....	59
Obr. 30 Elektrické schéma zapojení H-můstku pro ovládání kabin.....	60
Obr. 31 Funkční blok dveří – vstupy.....	61
Obr. 32 Funkční blok dveří - zdrojový kód.....	62

Obr. 33 Funkční blok motoru – vstupy	62
Obr. 34 Funkční blok motoru - zdrojový kód	63
Obr. 35 Funkční blok kabiny – vstupy	64
Obr. 36 Funkční blok kabiny - zdrojový kód.....	65
Obr. 37 Funkční blok tlačítka – vstupy.....	66
Obr. 38 Funkční blok tlačítka - zdrojový kód.....	66
Obr. 39 Funkce pro zařazení požadavků	67
Obr. 40 Funkce pro vyřazení požadavků	67
Obr. 41 Funkce podsvícení tlačítek	68
Obr. 42 Okno vizualizačního programu	69

SEZNAM TABULEK

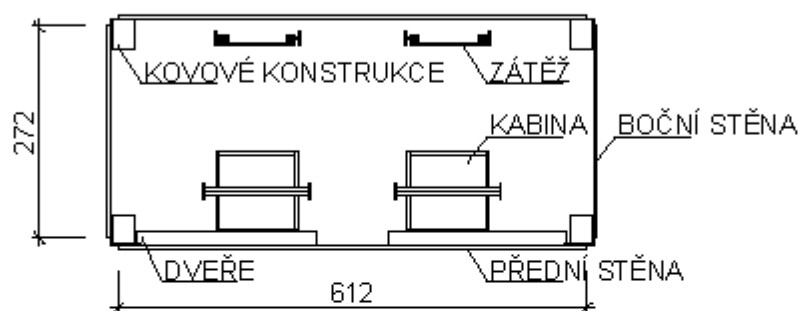
Tab. 1 Základní parametry PLC Foxtrot.....	18
Tab. 2 Charakteristika PLC Foxtrot.....	19
Tab. 3 Základní parametry periferního modulu IB-1301	21
Tab. 4 Základní parametry periferního modulu OS – 1401.....	22
Tab. 5 Elementární datové typy	30
Tab. 6 Rodové datové typy	31
Tab. 7 Operátory v jazyce strukturovaného textu ST	33
Tab. 8 Pravdivostní tabulka pro ovládání motoru pomocí H-můstku.....	47
Tab. 9 Pravdivostní tabulka logické funkce pro ovládání dveří	52
Tab. 10 Pravdivostní tabulka pro ovládání motoru pomocí H-můstku.....	56

SEZNAM PŘÍLOH

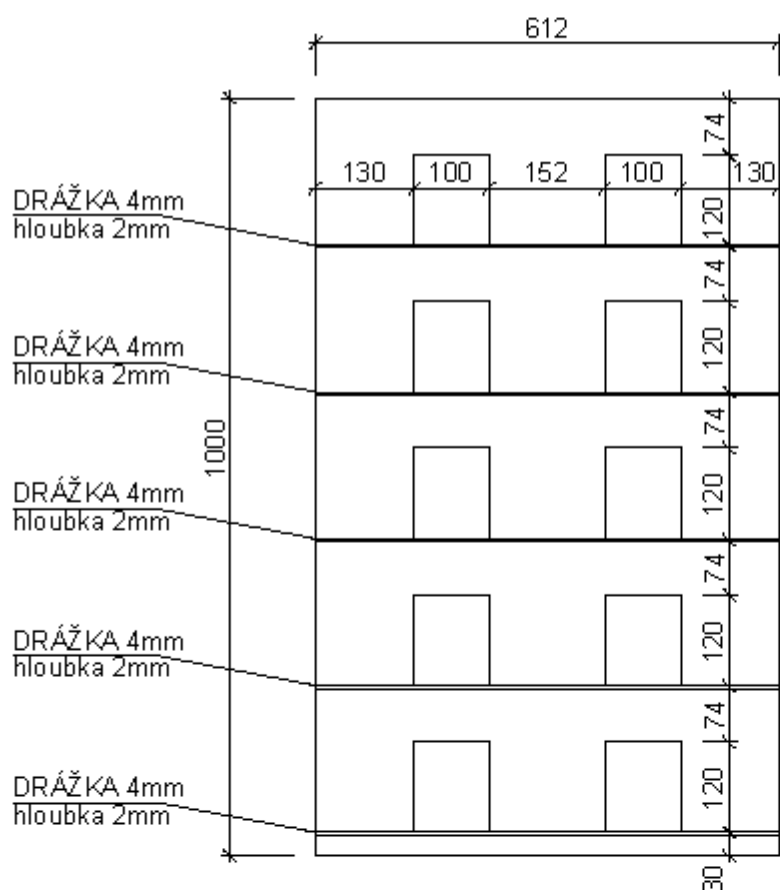
P I Model výtahu

P II CD-ROM Model výtahu

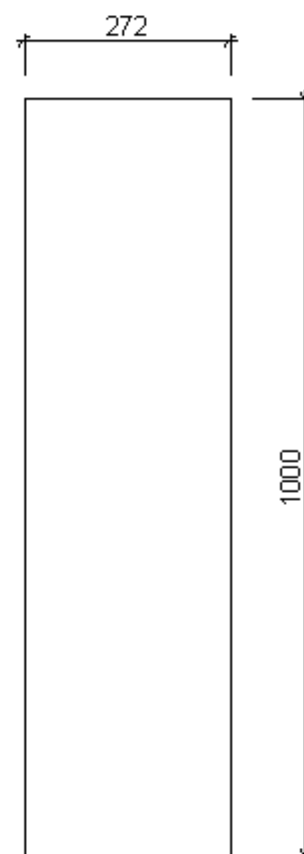
PŮDORYS



PŘEDNÍ STĚNA



BOČNÍ STĚNA



MODEL VÝTAHU		Univerzita Tomáše Bati ve Zlíně Fakulta aplikované informatiky	
DIPLOMOVÁ PRÁCE			
NÁZEV:	MODEL VÝTAHU	DATUM:	06/2010
VYPRACOVAL:	Bc.ONDŘEJ VRZAL	MĚŘÍTKO:	1:10