

Model výtahu pro výuku předmětu Programovatelné automaty - přepracování stávajícího modelu

Elevator Model for PLC Education - Upgrade of the Existing Model

Jindřich Žaludek



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jindřich ŽALUDEK**
Osobní číslo: **A09041**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Informační a řídicí technologie**
Forma studia: **prezenční**

Téma práce: **Model výtahu pro výuku PLC – přepracování stávajícího modelu**

Zásady pro vypracování:

- 1. Popište stávající model výtahu a proveďte návrh jeho rozšíření o další čidla zlepšující jeho funkci a o modul simulující řídicí panely obou kabin.**
- 2. Návrh hardwarově realizujte. Snažte se minimalizovat počet užívaných binárních vstupů při zachování maximální funkčnosti modelu.**
- 3. Vytvořte programové vybavení pro řízení celého modelu, tj. dvě kabiny pohybující se po pěti patrech.**
- 4. Vypracujte vzorové úlohy pro výuku programování PLC zahrnující i využití řídicího panelu.**
- 5. Vytvořte webovou aplikaci, umožňující monitorovat a řídit model prostřednictvím zabudovaného webserveru v PLC.**

Rozsah bakalářské práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. ŠMEJKAL, Ladislav a Marie MARTINÁSKOVÁ. PLC a automatizace, 1.díl. 1. Vyd. Praha: BEN – technická literatura, 1999. ISBN 80-860-5658-9.
2. ŠMEJKAL, Ladislav. PLC a automatizace, 2.díl. 1. Vyd. Praha: BEN – technická literatura, 2005. ISBN 80-7300-087-3.
3. MARTINÁSKOVÁ, Marie a Ladislav ŠMEJKAL. Řízení programovatelnými automaty. Vyd. 2. Praha: Vydavatelství ČVUT, 2004. ISBN 80-010-2925-5.
4. MARTINÁSKOVÁ, Marie a Ladislav ŠMEJKAL. Řízení programovatelnými automaty II. Vyd. 1. Praha: ČVUT, Strojní fakulta, 2000. ISBN 80-010-2096-7.
5. TECOMAT. Programovatelné automaty TECOMAT FOXTROT. 13. Vyd. 2010.

Vedoucí bakalářské práce:

Ing. Tomáš Sysala, Ph.D.

Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce:

24. února 2013

Termín odevzdání bakalářské práce:

14. června 2013

Ve Zlíně dne 24. února 2013

prof. Ing. Vladimír Vašek, CSc.
děkan



prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

ABSTRAKT

Tématem této práce je rozšíření modelu výtahu pro výuku programovatelných automatů (PLC), návrh programu dle normy IEC 61 131-3 pro ovládání tohoto modelu a několik vzorových úloh. Teoretická část práce obsahuje popis programovatelných automatů, zejména řady Foxtrot od české firmy Teco a.s., který je použit k řízení modelu a popis normy IEC 61 131-3. Praktická část pak popisuje model a jeho rozšíření, programování a vizualizaci řízeného procesu.

Klíčová slova: PLC, Foxtrot, výtah

ABSTRACT

The theme of this work is to extend the elevator model for teaching of programmable logic controllers (PLC) programming, design program according to IEC 61 131-3 for control of this model and some sample tasks. Theoretical part contains a description of programmable logic controllers, especially Foxtrot from Czech company Teco a.s., which is used to control the model and a description of the IEC 61 131-3. Practical part describes the model, its extensions, programming and visualization of the controlled process.

Keywords: PLC, Foxtrot, elevator

Děkuji vedoucímu mé bakalářské práce Ing. Tomáši Sysalovi, Ph.D. za trpělivost, panu Petru Dvořákovi, za poskytnutý materiál, pomoc a rady týkající se jak bakalářské práce, tak rady do života. Dále děkuji Ing. Petru Dostálkovi, Ph.D. za propůjčení potřebného vybavení a v neposlední řadě své rodině za poskytnutou psychickou i finanční podporu.

„Technický vývoj směřuje vždy od primitivního přes komplikované k jednoduchému“

Antoine de Saint-Exupéry

Prohlašuji, že

- beru na vědomí, že odevzdáním 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 bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk 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 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 – 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 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 bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem 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 bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD.....	9
I TEORETICKÁ ČÁST	10
1 PROGRAMOVATELNÉ AUTOMATY.....	11
1.1 HISTORIE PLC	11
1.2 SOUČASNOST PLC.....	12
1.3 BUDOUCNOST PLC.....	12
1.4 KONSTRUKCE PLC	13
1.4.1 Kompaktní PLC.....	13
1.4.2 Modulární PLC.....	14
1.5 PROVEDENÍ PLC	15
1.5.1 Centrální procesorová jednotka.....	15
1.5.2 Paměťový prostor	16
1.5.3 Binární vstupní a výstupní jednotky.....	16
1.5.4 Analogové vstupní a výstupní jednotky	17
1.5.5 Čítačové jednotky.....	17
1.5.6 Komunikační jednotky	17
1.5.7 Speciální jednotky	17
2 PLC TECOMAT FOXTROT.....	18
2.1 VYKONÁVÁNÍ PROGRAMU	18
2.2 UŽIVATELSKÉ PROCESY	19
2.3 KOMUNIKACE.....	20
2.4 ZÁKLADNÍ MODUL.....	21
2.5 PERIFERNÍ MODULY	24
2.5.1 Modul binárních vstupů IB-1301	24
2.5.2 Modul binárních výstupů OS-1401	25
3 NORMA IEC 61 131	27
3.1 NORMA IEC 61 131-3	27
3.1.1 Společné prvky	27
3.1.1.1 Typy dat	27
3.1.1.2 Proměnné	29
3.1.1.3 Programové organizační jednotky	30
3.1.2 Programovací jazyky	31
3.1.2.1 Jazyk seznamu instrukcí IL.....	32
3.1.2.2 Jazyk strukturovaného textu ST.....	33
3.1.2.3 Jazyk kontaktních schémat LD	34
3.1.2.4 Jazyk funkčního blokového schématu FBD.....	35
3.1.2.5 Sekvenční funkční graf SFC	36
II PRAKTICKÁ ČÁST	38
4 MODEL VÝTAHU.....	39

4.1	ŘÍDICÍ PLC.....	40
5	ROZŠÍŘENÍ MODELU.....	42
5.1	ČIDLA POLOHY	42
5.2	PANEL TLAČÍTEK	42
5.3	ZAPOJENÍ PLC.....	43
6	PROGRAMOVÉ VYBAVENÍ.....	45
6.1	GLOBÁLNÍ PROMĚNNÉ	45
6.2	PROGRAMY	46
6.2.1	prgInits, prgInitt.....	46
6.2.2	prgMain	46
6.3	FUNKČNÍ BLOKY	47
6.3.1	fbDvere.....	47
6.3.2	fbKabinal, fbKabinap	47
6.3.3	fbKam.....	47
6.3.4	fbVlozit	48
6.3.5	fbVyjmout	49
6.4	FUNKCE.....	49
6.4.1	fceIn.....	49
6.4.2	fceMax, fceMin	49
6.4.3	fcePWM	50
7	VZOROVÉ ÚLOHY	51
7.1	ÚLOHA 1	51
7.2	ÚLOHA 2	51
7.3	ÚLOHA 3	51
8	WEBSERWER	52
8.1	WEBMAKER.....	52
8.2	ZABEZPEČENÍ	53
	ZÁVĚR	55
	ZÁVĚR V ANGLIČTINĚ.....	56
	SEZNAM POUŽITÉ LITERATURY.....	57
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	58
	SEZNAM OBRÁZKŮ	59
	SEZNAM TABULEK.....	61
	SEZNAM PŘÍLOH.....	62

ÚVOD

Programovatelné automaty i výtahy jsou dnes již běžnou součástí tohoto světa. Svět bez výtahů bychom si téměř už ani nedokázali představit. Slouží jak k přepravě osobní, tak i k přepravě nákladní. V minulosti byly výtahy poháněny vodou, párou nebo i lidskou silou, dnes převažuje elektrický pohon. K jejich řízení jsou použity převážně PLC a mikropočítače.

I. TEORETICKÁ ČÁST

1 PROGRAMOVATELNÉ AUTOMATY

Programovatelný automat je uživatelsky programovatelný řídicí systém přizpůsobený pro řízení průmyslových a technologických procesů nebo strojů, dříve specializovaný na úlohy převážně logického typu (převážně u starších typů nebo u nejmenších systémů). Prostřednictvím číslicových nebo analogových vstupů a výstupů získává a předává informace z a do řízeného zařízení. Algoritmy řízení jsou uloženy v paměti uživatelského programu, který je cyklicky vykonáván. Nejčastěji se označuje zkratkou PLC (Programmable logic controller), v německé literatuře se lze setkat s označením SPS (Speicherprogrammierbare Steuerung). Občas najdeme i označení PC (Programmable controller). Česká zkratka je PA (Programovatelný automat). [3, 5]

Původně byly programovatelné automaty navrženy k řešení úloh logického řízení, často jako náhrada pevné reléové logiky. V současných aplikacích se však zvyšuje podíl úloh regulačního typu, monitorování řízeného procesu i analogových měření. [3]

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 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. Do popředí však neustále, a v současné době čím dál tím více, vystupuje otázka ekonomické efektivity. Hospodárnost 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. [6]

Automatizované ovládání bylo v minulých dobách založeno na reléové technice, která byla v 60. letech postupně nahrazována polovodičovými integrovanými logickými obvody. U nás to byly např. stavebnice Transimat a Logizet. Matematickým modelem těchto obvodů jsou logické rovnice. U obvodů, u kterých se vystačí s relátky či s jednoduchými logickými funkcemi reprezentovanými jednotkami stavebnice, je skutečně možné celou úlohu popsat soustavou logických rovnic. Pro takovouto úlohu je univerzální programovací jazyk, v tehdejší době např. Fortran, zbytečně komplikovaný a těžkopádný. Naproti tomu stojí úloha „překladače logických rovnic“, kterou i v tehdejší době zvládl schopný analytik a programátor poměrně snadno. [6]

1.2 Současnost PLC

Samostatný vývoj hardwaru PLC se zdá být uzavřen a bude pravděpodobně sledovat další vývoj strojů na zpracování dat. Významný vliv na technické řešení PLC má dnes již převládající distribuované řešení počítačových automatizačních systémů. Pro PLC to znamená klesající požadavky na jejich složitost. Mohou zůstat jednoduchými stroji, do jisté míry specializovanými. Rostou pouze požadavky na schopnost komunikace a v souvislosti s tím na možnost jednoduchého zapojení do informačních sítí. [6]

V dnešní době se všechny používané programovací metody sešly v normě, která nese označení IEC 61131-3. Implementovat všechny metody zahrnuté v této normě na svých PLC se v současnosti snaží již všichni významní výrobci. Stále se sice lze setkat s množstvím odchylek a nepříjemných omezení, nicméně situace se soustavně zlepšuje. Zatímco v roce 1990 nabízel na našem trhu metodu GRAFCET jen jediný dodavatel (PEP Modular Computers), dnes ji nabízejí téměř všichni. [6]

1.3 Budoucnost PLC

PLC zřejmě mají i nadále zajištěnou budoucnost. Možnost distribuce funkcí na samostatné počítačové jednotky, vyplývající především z nízké ceny počítačů, poskytne i dále prostor pro specializované počítače. Jejich specializace však bude spočívat především v softwaru. Distribuované řešení přitom předpokládá, že budou k dispozici kvalitní, spolehlivé a rychlé

způsoby komunikace. I v tomto směru je současná situace dosti uspokojivá. Tím se ovšem nelze zbavit potíží vyplývajících ze složitosti řešených úloh. Do jisté míry specifickým produktem se svými specifickými problémy tedy budou PLC i v budoucnosti. Nebudou však ani tak problémem samy o sobě, problémem bude jejich začlenění do širšího celku a souvislostí. To bude vyžadovat, aby PLC byly vždy chápány jako neoddělitelná část celku, a tedy část, kterou nelze vyjmout ze souhrnného automatizačního projektu. [6]

1.4 Konstrukce PLC

Programovatelné automaty je možno třídit dle různých hledisek. Menší systémy bývají řešeny jako kompaktní (Obr. 1), větší jako modulární (Obr. 2). Princip činnosti kompaktních i modulárních programovatelných automatů a většinou i způsob programování je stejný, konstrukčním pojetím a uživatelskou koncepcí jsou však obě kategorie výrazně odlišné. [3]

1.4.1 Kompaktní PLC

Měly původně pevně danou konfiguraci integrovaných modulů a byly uzavřeny v jednom pouzdře. Toto pouzdro se montuje přímo do výrobku, nebo na rozváděcí lišty DIN do rozvaděče. V poslední době je i u kompaktních PLC patrná snaha o určitý stupeň modularity, takže je možno i u malých aplikací přizpůsobit sestavu PLC přesněji k potřebám konkrétní aplikace. [3]



Obr. 1 Kompaktní PLC [8]

1.4.2 Modulární PLC

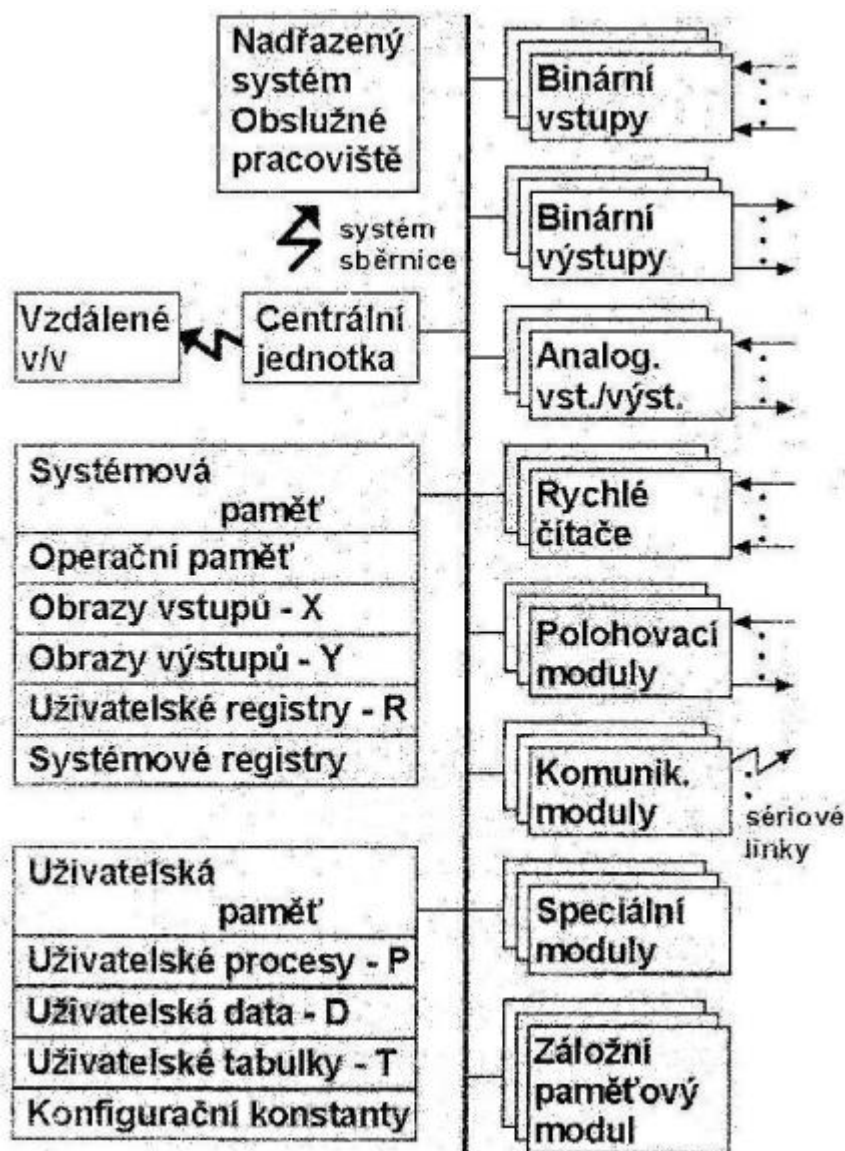
Jsou tvořeny v podstatě pevným procesorovým jádrem s napájecím zdrojem, umístěným v rámu či na nosné desce nebo liště, ke kterému se přes sběrnici připojují místní i vzdálené periferní jednotky zajišťující nejrůznější funkce a operace. Kromě běžných funkcí dostupných u Kompaktních PLC jako jsou binární i analogové vstupně výstupní jednotky zde bývá možnost z dalších jednotek pro rychlé čítání, pro polohování, pro nejrůznější typy komunikace (např. GSM), pro regulaci, i pro speciální funkce. [3]



Obr. 2 Modulární PLC [8]

1.5 Provedení PLC

PLC se v podstatě skládá z centrální procesorové jednotky, systémové a uživatelské paměti, souboru vstupních a výstupních jednotek pro připojení řízeného systému, čítačových modulů, souboru komunikačních jednotek pro komunikaci se souřadnými i nadřazenými řídicími systémy a různých speciálních modulů. Jednotky PLC jsou navzájem propojeny systémovou sběrnicí. [3, 7]



Obr. 3 Blokové schéma PLC [3]

1.5.1 Centrální procesorová jednotka

Je jádrem celého PLC a určuje jeho výkonnost. Bývá jednoprocessorová, ale i víceprocesorová. U víceprocesorových systémů bývají někdy použity matematické

koprocesory, vstupně výstupní procesory a někdy též komunikační procesory. Někdy se pro zpracování bitových logických operací s jednobitovými operandy pro urychlení práce používá rychlý bitový procesor. [3]

Důležitým charakteristickým parametrem centrální procesorové 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 desetinám milisekund. [3]

1.5.2 Paměťový prostor

Paměťový prostor se může dělit na paměť uživatelskou, systémovou a paměť dat.

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. [3, 7]

V systémové paměti je umístěn systémový program.

V paměti dat, která musí být typu RAM, jsou umístěny uživateli dostupné uživatelské 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. [3]

1.5.3 Binární vstupní a výstupní jednotky

Bývají někdy též označovány jako digitální, mají dvouhodnotový charakter. Vstupní jednotky slouží pro připojování prvků výstupního signálu, což mohou být např. tlačítka, přepínače, koncové spínače v nejrůznějším provedení, senzory doteku nebo přiblížení, dvouhodnotové senzory tlaku, hladiny, teploty apod.

Výstupní jednotky souží k připojování nejrůznějších akčních členů výstupního signálu. Mohou to být např. různá optická i akustická signalizační zařízení nebo cívky relé, stykačů, solenoidových ventilů, elektromagneticky ovládaných pneumatických či hydraulických rozvaděčů, elektromagnetických spojek apod. [3]

1.5.4 Analogové vstupní a 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, nebo třeba měřené napětí či výstup z potenciometru.

Důležitou součástí analogové vstupní jednotky je A/D převodník, který převádí analogové napěťové nebo proudové signály na číselné hodnoty se kterými můžeme pracovat. Rozsah převodníku (obvykle 8 až 12 bitů) mimo jiné určuje přesnost převodu analogové veličiny na číselný údaj.

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. Jejich nezbytnou součástí je D/A převodník, který převádí číselný údaj na analogový signál. [3]

1.5.5 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 nebo absolutních snímačů. [3]

1.5.6 Komunikační jednotky

Důležitou vlastností PLC systémů je schopnost komunikovat se vzdálenými moduly vstupů a výstupů, s podsystémy, se souřadnými i nadřizenými systémy, s operátorskými panely a s jinými inteligentními přístroji, s počítači a jejich sítěmi a tak vytvářet distribuované systémy.

Komunikační jednotky většinou rozšiřují počet asynchronních sériových komunikačních kanálů. U některých systémů jsou k dispozici i jednotky dálkových přenosů umožňující dálkové přenosy dat přes modem, GSM, nebo radiomodem. [3]

1.5.7 Speciální jednotky

U některých výrobců PLC je možné setkat i se specializovanými moduly pro řešení regulačních úloh nebo pro řešení úloh s využitím fuzzy logiky a fuzzy regulace.

2 PLC TECOMAT 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. [5, 7]

2.1 Vykonávání programu

Řídicí algoritmus programovatelného automatu je zapsán jako posloupnost instrukcí v paměti uživatelského programu. Centrální jednotka postupně čte z této paměti jednotlivé instrukce, provádí příslušné operace s daty v zápisníkové paměti a zásobníku, případně provádí přechody v posloupnosti instrukcí, je-li instrukce ze skupiny organizačních instrukcí. Jsou-li provedeny všechny instrukce požadovaného algoritmu, provádí centrální jednotka aktualizaci výstupních proměnných do výstupních periferních jednotek a aktualizuje stavy ze vstupních periferních jednotek do zápisníkové paměti. Tento děj se stále opakuje v programové smyčce a nazýváme jej cyklem programu. Na rozdíl od jiných programovatelných systémů se programátor PLC nemusí starat o to, aby po konci programu vrátil jeho vykonávání opět na začátek - zajistí to již systémový program. Naopak každé dlouhodobé setrvání programu v programové smyčce je "fatální chybou" a systém jej hlásí jako "překročení doby cyklu". [9, 10]



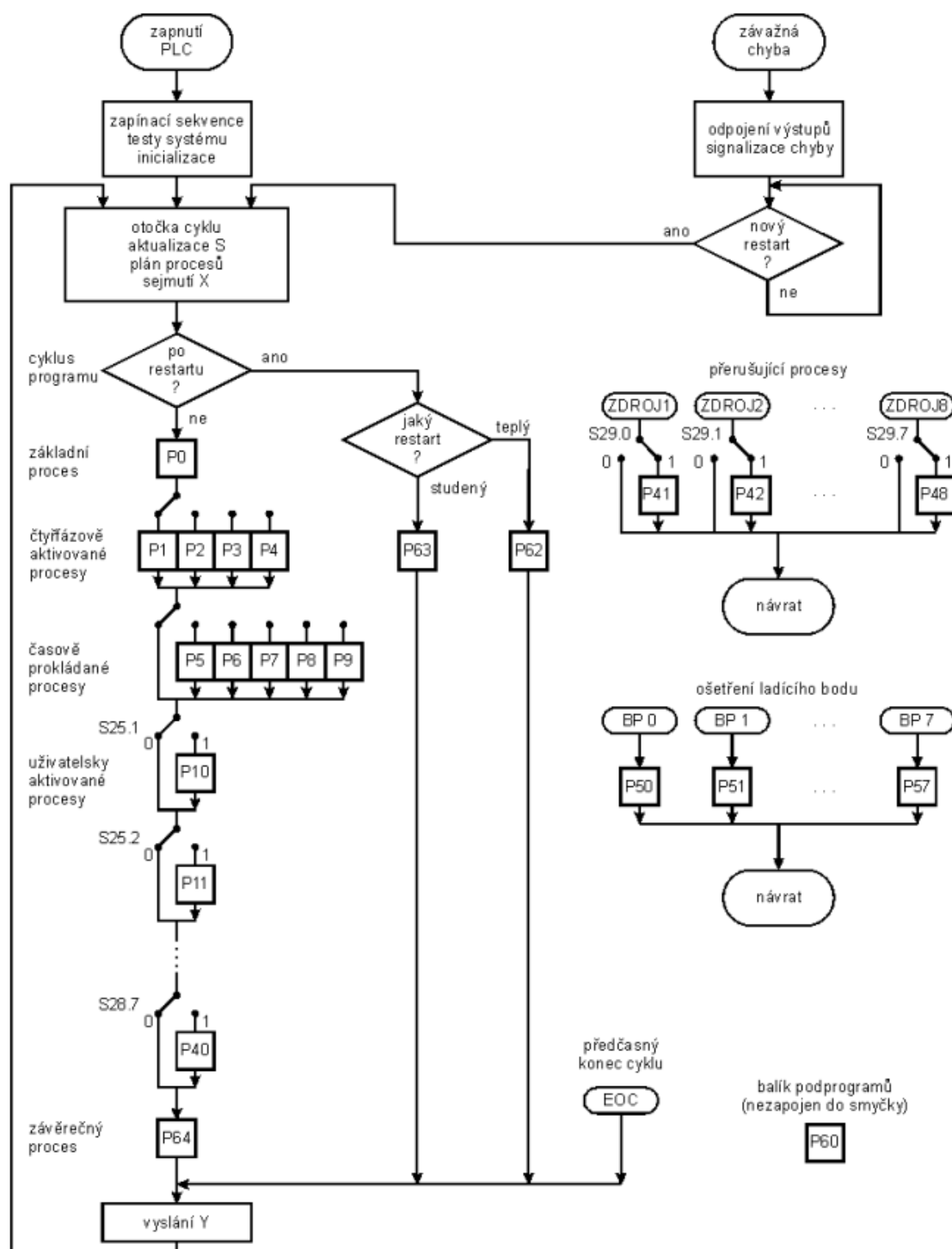
Obr. 4 Cyklus PLC [5]

2.2 Uživatelské procesy

Uživatelský program se skládá z uživatelských procesů. Teoreticky jich smí být až 65 (P0 až P64), prakticky jich bývá výrazně méně. Na rozdíl od tradičních operačních systémů reálného času pro počítače zde uživatel nemá takovou volnost při ovládání procesů. Procesy jsou aktivovány podle předem definovaných pravidel. V rámci těchto pravidel můžeme dodatečně ovlivnit aktivaci většiny procesů v průběhu uživatelského programu. Proces P0 se aktivuje vždy po otočce cyklu, P64 naopak vždy před otočkou, procesy P1, P2, P3, P4 se v aktivaci cyklicky střídají, P5 až P9 se aktivují v časových periodách (každých 0,4 s; 3,2 s; 25,6 s; 3,4 min a 27,2 min). O aktivaci procesů P10 až P40 rozhoduje programátor tím, jak nastaví hodnoty aktivačních bitových proměnných v systémových registrech S25 až S29. Procesy P62 a P63 se aktivují jednorázově po zapnutí nebo po restartu systému (inicializace při teplém a studeném restartu). Procesy P41 až P48 jsou aktivovány jako odezva na přerušující událost. [9, 10]

Procesy	Určení
P0	základní proces
P1 až P4	čtyřfázově aktivované procesy
P5 až P9	časově aktivované procesy
P10 až P40	uživatelsky aktivované procesy
P41 až P48	přerušovací procesy
P49	systémový proces - nepoužívat!
P50 až P57	ošetření ladicího bodu
P58, P59	systémové procesy - nepoužívat!
P60	balík podprogramů
P61	systémový proces - nepoužívat!
P62	teplý restart
P63	studený restart
P64	závěrečný proces cyklu

Tab. 1 Přehled procesů uživatelského programu a jejich určení [9]



Obr. 5 Schéma aktivace procesů [9]

2.3 Komunikace

Datové komunikace mezi PLC a nadřízenými 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 podle volby zákazníka (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. Všechny centrální jednotky jsou vybaveny rozhraním Ethernet 10/100 Mb umožňujícím provozovat současně více logických spojení.

Rozšiřovací periferní moduly se k centrální jednotce připojují pomocí sériové sběrnice TCL2. 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 i další účastníci, kteří vyhoví požadavkům sítě EPSNET (operátorské panely, apod.). [5]

2.4 Základní modul

Všechny základní moduly systému FOXTROT se skládají z několika částí. První část tvoří centrální jednotka s hlavním procesorem systému, dvěma sériovými kanály, rozhraním Ethernet a systémovou sběrnici TCL2 pro komunikaci s periferními moduly. Druhou část tvoří procesor zajišťující komunikaci na sběrnici CIB s moduly rodiny CFox. Třetí část je periferní.

Centrální jednotka provádí vlastní uživatelský program a obsahuje základní funkce, bez 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.

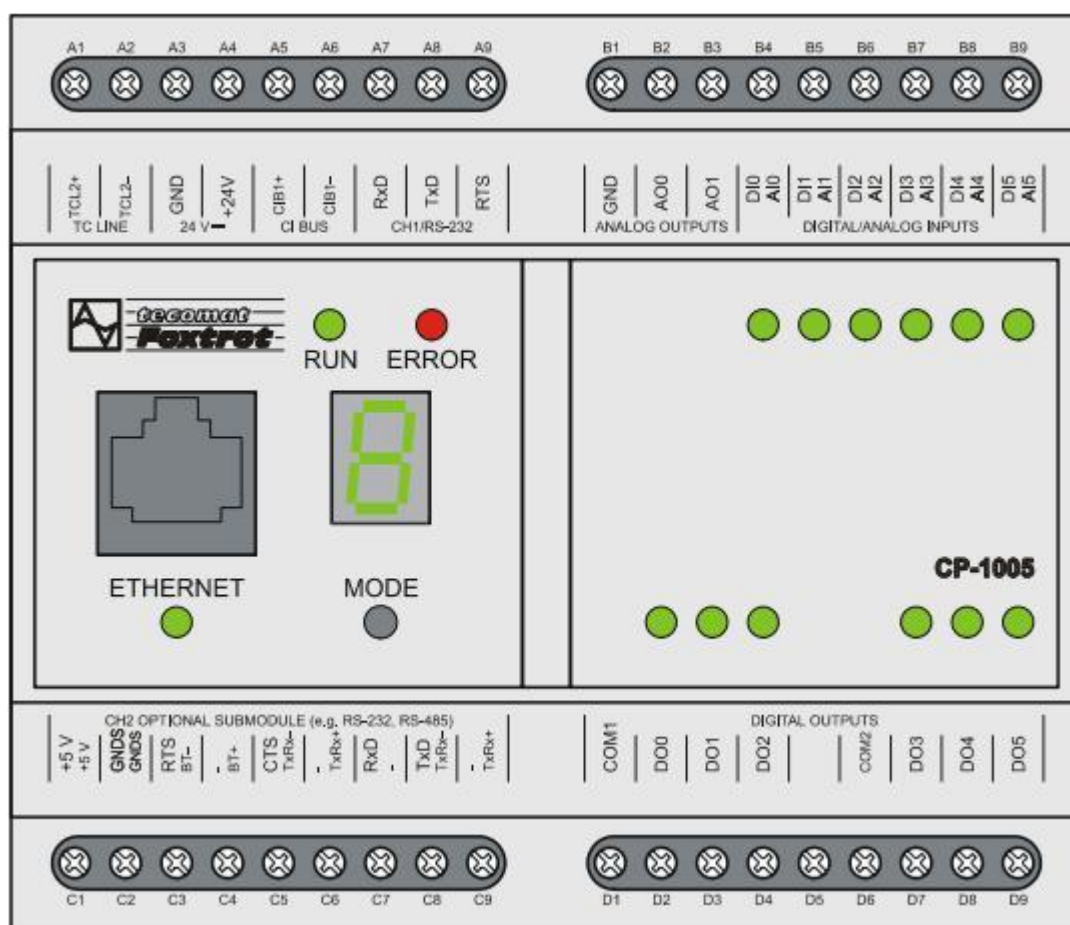
FOXTROT obsahuje centrální jednotku řady K s instrukčním souborem, jehož součástí jsou i aritmetické operace s čísly v pevné řádové čárce o velikosti 32 bitů bez znaménka i se znaménkem, v pohyblivé řádové čárce (floating point single precision - 32 bitů a double

precision - 64 bitů), instrukce PID regulátoru, podpora operátorských panelů a podpora vyššího programovacího jazyka. Režim a diagnostická hlášení jsou zobrazována na sedmisegmentovém zobrazovači nebo na displeji. [5]

Pro řízení modelu výtahu byl vybrán základní modul CP-1005.

CP-1005	<p>centrální jednotka řady K</p> <p>6 volitelných vstupů - binární 24 V / analogové (unipolární napětové a proudové rozsahy, pasivní odporové snimače, 16 bitů)</p> <p>6 reléových výstupů 250 V</p> <p>2 analogové výstupy 0 - 10 V (10 bitů)</p> <p>2 sériové kanály (CH1 - RS-232, CH2 - volitelné rozhraní)</p> <p>1 rozhraní Ethernet 10/100 Mb</p> <p>1 linka sběrnice TCL2 pro připojení periférií</p> <p>1 linka sběrnice CIB</p> <p>slot paměťové karty SDHC / SD / MMC</p> <p>možnost osazení submodulu s binárními vstupy a výstupy</p> <p>možnost osazení submodulu s dalšími 2 sériovými kanály</p>
---------	--

Tab. 2 Popis základního modulu CP-1005 [5]



Obr. 6 Základní modul CP-1005 [5]

Typ modulu	CP-10x5 (IR-1056)
Počet vstupů	6
Počet vstupů ve skupině	6
Galvanické oddělení od vnitřních obvodů	ne
Diagnostika	signalizace vybuzeného vstupu na panelu
Společný vodič	minus
Vstupní napětí	
pro log.0 (UL)	max. +5 V DC
pro log.1 (UH)	min. +12 V DC
	typ. +24 V DC
	max. +30 V DC
Vstupní proud při log.1	typ. 5 mA
Zpoždění z log.0 na log.1	1 ms
Zpoždění z log.1 na log.0	1 ms
Počet výstupů	6
Počet skupin x počet výstupů ve skupině	2 x 3
Galvanické oddělení od vnitřních obvodů	ano (i skupiny navzájem)
Diagnostika	signalizace vybuzeného výstupu na panelu
Typ výstupů	elektromechanické relé, nechráněný výstup
Typ kontaktu	spínací
Spínané napětí	max. 250 V
	min. 5 V
Spínaný proud	max. 3 A
	min. 10 mA
Krátkodobá přetížitelnost výstupu	max. 4 A
Proud společnou svorkou	max. 10 A
Doba sepnutí kontaktu	typ. 10 ms
Doba rozepnutí kontaktu	typ. 4 ms
Mezní hodnoty spínané zátěže	
pro odporovou zátěž	max. 3 A při 30 V DC nebo 230 V AC
pro induktivní zátěž DC13	max. 3 A při 30 V DC
pro induktivní zátěž AC15	max. 3 A při 230 V AC
Frekvence spínání bez zátěže	max. 300 sepnutí / min.
Frekvence spínání se jmenovitou zátěží	max. 20 sepnutí / min.
Mechanická životnost	min. 5 000 000 cyklů
Elektrická životnost při maximální zátěži	
pro odporovou zátěž	min. 100 000 cyklů
pro induktivní zátěž DC13	min. 100 000 cyklů
pro induktivní zátěž AC15	min. 100 000 cyklů
Ochrana proti zkratu	není
Ošetření induktivní zátěže	vnější - RC člen, varistor, dioda (DC)
Izolační napětí	
mezi výstupy a vnitřními obvody	3750 V AC
mezi skupinami výstupů navzájem	3750 V AC

Tab. 3 Parametry modulu CP-1005 [5]

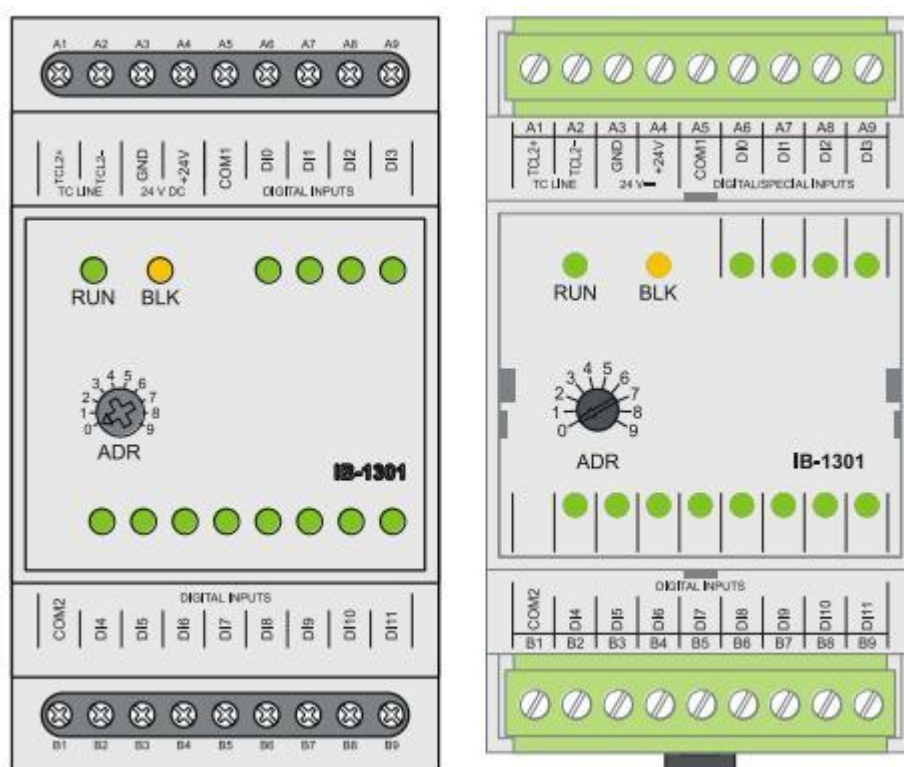
Počet periférií, které nabízí modul CP-1005, je pochopitelně nedostatečný pro řízení navrhovaného modelu výtahu. Z tohoto důvodu jsou k základnímu modulu připojeny další periferní moduly. [7]

2.5 Periferní moduly

Pro řízení modelu výtahu byly použity periferní moduly binárních vstupů IB-1301 a binárních výstupů OS-1401.

2.5.1 Modul binárních vstupů IB-1301

Periferní modul IB-1301 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. Vybuzení (sepnutí) vstupu je signalizováno rozsvícením příslušné LED diody. Modul je napájen z vnějšího napájecího napětí 24 V DC, které není galvanicky oddělené od vnitřních obvodů. Od roku 2011 je modul vyráběn v novém mechanickém provedení, kde pevné svorky nahradila vyjímatelná svorkovnice. Elektrické parametry modulu zůstaly shodné. [11]



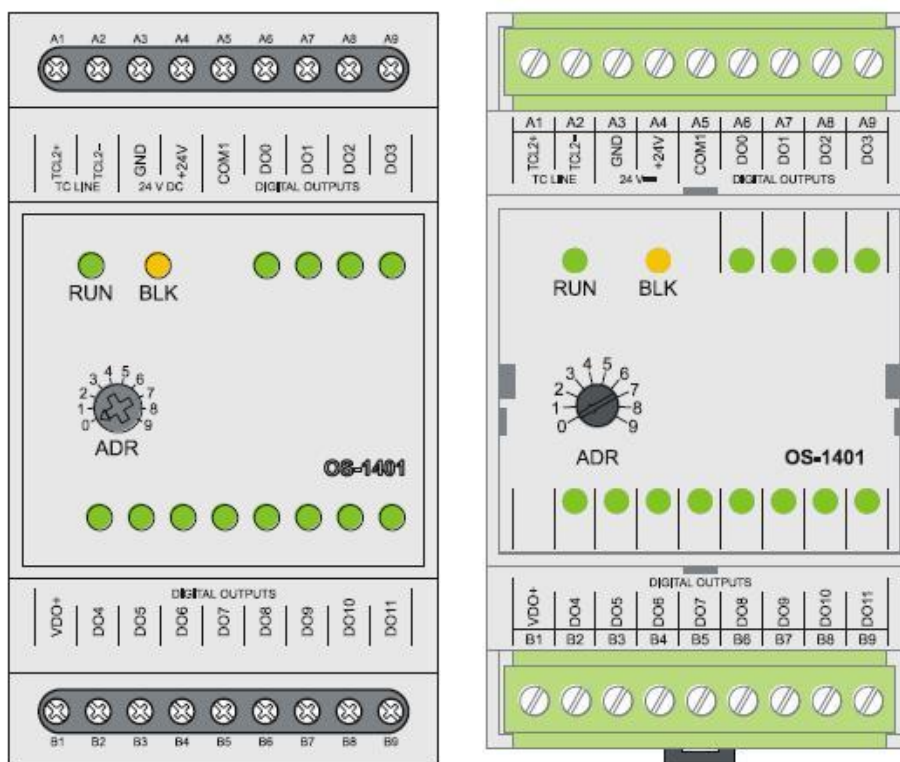
Obr. 7 Periferní modul IB-1301 (staré a nové provedení) [11]

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	max. +5 V DC	
pro log.0 (UL)	min. -5 V DC	
pro log.1 (UH)	min. +15 V DC	
	typ. +24 V DC	
	max. +30 V DC	
	vstupy DI0 - DI3	vstupy DI4 - DI11
Vstupní proud při log.1	typ. 10 mA	typ. 5 mA
Zpoždění z log.0 na log.1	5 μ s	5 ms
Zpoždění z log.1 na log.0	5 μ s	5 ms
Minimální šířka zachyceného pulzu	50 μ s	-

Tab. 4 Parametry modulu IB-1301 [11]

2.5.2 Modul binárních výstupů OS-1401

Periferní modul OS-1401 obsahuje 12 binárních výstupů 24 V DC. Všechny výstupy jsou galvanicky oddělené. Vybuzení (sepnutí) výstupu je signalizováno rozsvícením příslušné LED diody. Modul je napájen z napájecího napětí 24 V DC, které není galvanicky oddělené od vnitřních obvodů. Od roku 2011 je modul vyráběn v novém mechanickém provedení, kde pevné svorky nahradila vyjímatelná svorkovnice. Elektrické parametry modulu zůstaly shodné. [11]



Obr. 8 Periferní modul OS-1401 (staré a nové provedení) [11]

Typ modulu	OS-1401	
Počet výstupů	12	
Počet výstupů ve skupině	12	
Galvanické oddělení od vnitřních obvodů	ano	
Diagnostika	signalizace vybuzeného vstupu na panelu	
	výstupy DO0 - DO3	výstupy DO4 - DO11
Typ výstupů	tranzistorový výstup	
Společný vodič	plus	
Spínané napětí	9,6 až 28,8 V DC	
Spínaný proud	max. 2 A	max. 0,5 A
Proud společnou svorkou	max. 4,4 A	max. 9 A
Zbytkový proud při rozepnutí	max. 300 μ A	
Doba sepnutí	max. 400 μ s	
Doba rozepnutí	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. 5 Parametry modulu OS-1401 [11]

3 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.

Programovací jazyky definuje norma IEC 61 131-3, která je třetí částí z rodiny norem IEC 61 131 a představuje první vážný pokus o standardizaci programovacích jazyků pro průmyslovou automatizaci. [12]

3.1 Norma IEC 61 131-3

Na normu 61 131-3 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. [12]

Dělí se v podstatě na dvě základní části:

- Společné prvky
- Programovací jazyky

3.1.1 Společné prvky

3.1.1.1 Typy dat

V rámci společných prvků jsou definovány typy dat. Definování datových typů napomáhá prevenci chyb v samém počátku tvorby projektu. Je nutné definovat typy všech použitých parametrů. Pro programování v některém z jazyků podle normy IEC 61 131-3 jsou definovány tzv. **elementární**, předdefinované datové typy, dále jsou definovány **rodové** datové typy pro příbuzné skupiny datových typů. A konečně je k dispozici mechanismus, kterým může uživatel vytvářet vlastní **odvozené** (uživatelské) datové typy. Tímto způsobem

můžeme např. definovat jako samostatný datový typ analogový vstupní kanál a opakovaně ho používat pod definovaným jménem. [12]

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 DT	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. 6 Elementární datové typy [12]

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

Tab. 7 Rodové datové typy [12]

3.1.1.2 Proměnné

Jsou v podstatě prostředkem pro identifikaci datových objektů, jejichž obsah se může měnit. Proměnná může být deklarována některým z elementárních nebo některým z odvozených (uživatelských) datových typů. Tím se programování podle IEC 61 131-3 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 přiřazené programátorem, které slouží v podstatě pro rezervaci místa v paměti a obsahují hodnoty dat programu.

Proměnné mohou být přiřazeny explicitně k hardwarovým adresám (např. vstupům, výstupům) pouze v konfiguracích, zdrojích nebo programech. Tímto způsobem je dosaženo vysokého stupně hardwarové nezávislosti a možnosti opakovaného využití softwaru na různých hardwarových platformách.

Oblast působnosti proměnných je běžně omezena pouze na tu programovou organizační jednotku, ve které byly deklarovány (proměnné jsou v ní lokální). To znamená, že jejich jména mohou být používána v jiných částech bez omezení. Tímto opatřením dojde k eliminaci řady dalších chyb. Pokud mají mít proměnné globální působnost, např. v rámci celého projektu, pak musí být jako globální deklarovány. Aby bylo možné správně nastavit počáteční stav procesu nebo stroje, může být parametrům přiřazena počáteční hodnota při startu nebo studeném restartu. [12]

Třída proměnné	Význam	Určení
VAR_INPUT	vstupní	Pro předávání vstupních parametrů do POU Tyto proměnné jsou viditelné z ostatních POU a jsou z nich také nastavovány
VAR_OUTPUT	výstupní	Pro předávání výstupních parametrů z POU Tyto proměnné jsou viditelné z ostatních POU, kde je možné provádět pouze jejich čtení Změnu hodnoty těchto proměnných lze provádět pouze v rámci POU, ve které byly proměnné deklarovány
VAR_IN_OUT	vstup / výstupní	Pro nepřímý přístup k proměnným ležícím vně POU Proměnné lze číst i měnit jejich hodnotu uvnitř i vně POU
VAR_EXTERNAL	globální	Proměnné definované v mnemokódu PLC
VAR_GLOBAL	globální	Proměnné, které jsou dostupné ze všech POU
VAR	lokální	Pomocné proměnné používané v rámci POU Z ostatních POU nejsou viditelné, to znamená, že je lze číst resp. měnit jejich hodnotu pouze v rámci POU, ve které jsou deklarovány Tyto proměnné mohou uchovávat hodnotu i mezi jednotlivými voláními příslušné POU
VAR_TEMP	lokální	Pomocné proměnné používané v rámci POU Z ostatních POU nejsou viditelné Tyto proměnné vznikají při vstupu do POU a zanikají po ukončení POU – nemohou tedy uchovávat hodnotu mezi dvěma voláními POU

Tab. 8 Třídy proměnných [12]

3.1.1.3 Programové organizační jednotky

Základním pojmem při programování podle normy IEC 61 131-3 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 vracet 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) vracet 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.).

Každá POU se skládá ze dvou základních částí: **deklarační** a **výkonné**. V deklarační části POU se definují proměnné potřebné pro činnost POU. Výkonná část pak obsahuje vlastní příkazy pro realizaci požadovaného algoritmu. [12]

3.1.2 Programovací jazyky

V rámci standardu jsou definovány čtyři programovací jazyky. Jejich sémantika i syntaxe je přesně definována a neponechává žádný prostor pro nepřesné vyjadřování. Zvládnutím těchto jazyků se tak otevírá cesta k používání široké škály řídicích systémů, které jsou na tomto standardu založeny.

Programovací jazyky se dělí do dvou základních kategorií:

Textové jazyky

IL - Instruction List - jazyk seznamu instrukcí

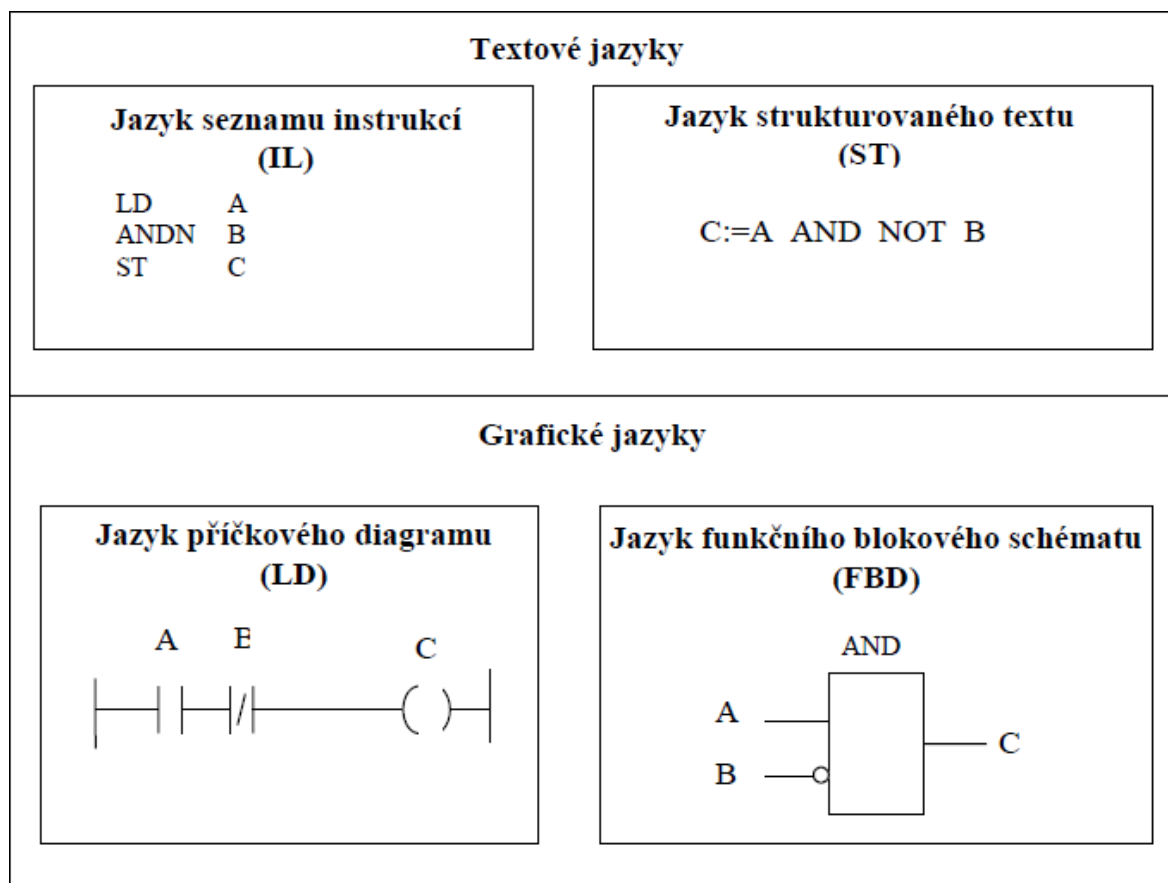
ST - Structured Text - jazyk strukturovaného textu

Grafické jazyky

LD - Ladder Diagram - jazyk kontaktních schémat

FBD - Function Block Diagram - jazyk funkčního blokového schématu

Pro první přehled je na Obr. 9 stejná logická funkce, a to součin proměnné **A** a negované proměnné **B** s výsledkem ukládaným do proměnné **C**, vyjádřen ve všech čtyřech programovacích jazycích. [12]



Obr. 9 Logická funkce ANDN ve čtyřech základních jazycích [12]

Volba programovacího jazyka je závislá na zkušenostech programátora, na typu řešeného problému, na úrovni popisu problému, na struktuře řídicího systému a na řadě dalších okolností, jako jsou např. typ odvětví průmyslu, zvyklosti firmy implementující řídicí systém, zkušenosti spolupracovníků v týmu apod. Všechny čtyři základní jazyky jsou vzájemně provázány. [12]

3.1.2.1 Jazyk seznamu instrukcí IL

Jazyk seznamu instrukcí je nízkoúrovňový jazyk typu assembler. Tento jazyk patří mezi řádkově orientované jazyky. *Seznam instrukcí* se skládá ze sekvence *instrukcí*. Každá instrukce 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. 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. Ukázka programu v jazyku IL je uvedena na Obr. 10. [12]

```
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:
        LDN     tmp2
  END_PROGRAM
```

Obr. 10 Program v jazyce IL [12]

3.1.2.2 Jazyk strukturovaného textu ST

Jazyk strukturovaného textu 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 a iterační smyčky. 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**. 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 se skládá z **operátorů a operandů**. Operátory jazyka strukturovaného textu ST jsou přehledně uspořádány v Tab. 9. [12]

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. 9 Operátory v jazyce strukturovaného textu ST [12]

Vyhodnocení výrazu spočívá v aplikaci operátorů na operandy a to s ohledem na prioritu vyjádřenou v Tab. 9. 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. Názorně je to zobrazeno na Obr. 11. [12]

```

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

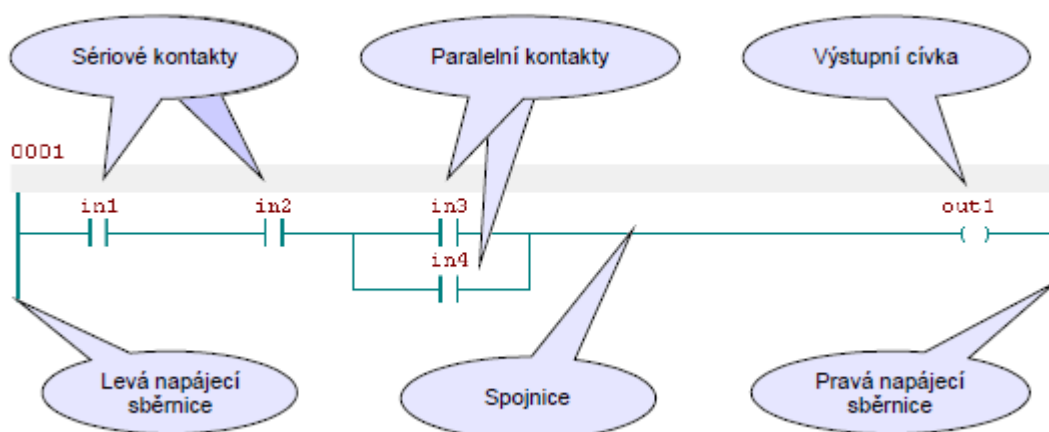
```

Obr. 11 Program v jazyce ST [12]

3.1.2.3 Jazyk kontaktních schémat LD

Jazyk kontaktních schémat pochází z elektromechanických reléových obvodů. Je založen na grafické reprezentaci reléové logiky. Tento jazyk je primárně určen pro zpracování

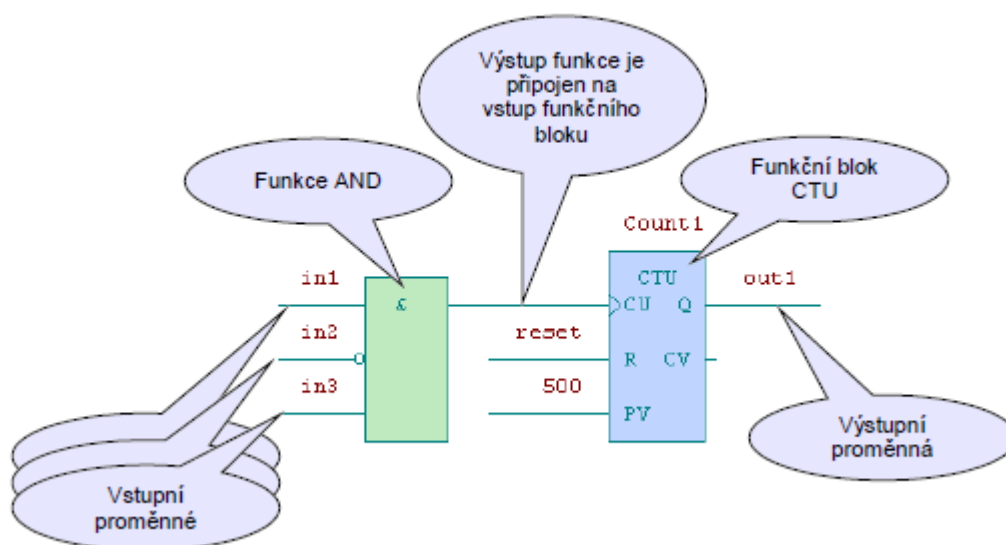
booleovských signálů. Výkonná část POU v jazyce LD je složena z obvodů. Obvod je v jazyce LD ohraničen tzv. napájecími sběrnicemi na levé a pravé straně. Z levé napájecí sběrnice „vede“ logická jednička (TRUE) do všech na ni připojených grafických prvků, typicky spínacích a rozpínacích kontaktů. V závislosti 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 sběrnicí. Typickým představitelem výstupního prvku je cívka. [12]



Obr. 12 Program v jazyce LD [12]

3.1.2.4 Jazyk funkčního blokového schématu FBD

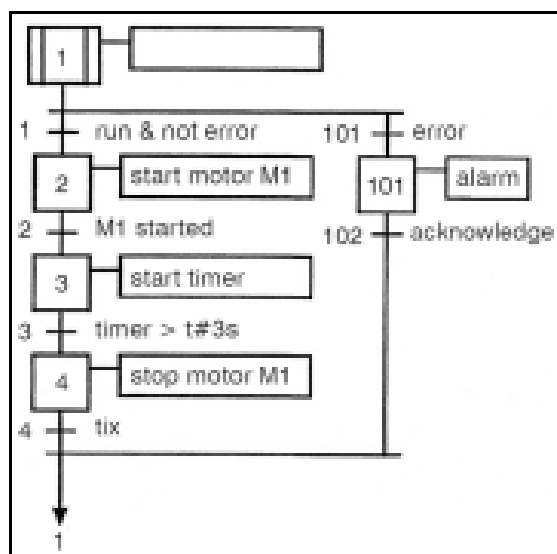
Jazyk funkčního blokového schématu je založen na propojování funkčních bloků a funkcí. Funkce a funkční bloky jsou reprezentovány obdélníkem. Rozdíl je v tom, že v jazyce LD lze spojnicemi mezi prvky přenášet pouze hodnoty typu BOOL zatímco v jazyce FBD mohou spojnice mezi grafickými prvky přenášet hodnoty libovolného typu. Jazyk FBD neobsahuje žádné další grafické prvky, jako jsou kontakty nebo cívky v jazyce LD. Prvky jazyka FBD se propojují spojnicemi toku signálu. Výstupy funkčních bloků se spolu nepropojují. [12]



Obr. 13 Grafika obvodu v jazyce FBD [12]

3.1.2.5 Sekvenční funkční graf SFC

Někdy též označován jako GRAFCET. Základem dobrých vlastností metody sekvenčních funkčních grafů je myšlenka dekompozice jako účinného nástroje ke zjednodušování složitých problémů. Složitý systém se dekomponuje na jeho části. Modulární dekompozice rozkládá systém na moduly. Obecně platí, že menší moduly jsou jednodušší. Dekompozice chování je dekompozicí uvedeného souhrnného procesu. Dílčí procesy nemusejí obhospodařovat všechny stavové proměnné daného modulu ve všech stavech, ale dotýkají se jen těch, které spadají pod působnost daného dílčího procesu. Zjednodušení z toho plynoucí je zřejmé. Metoda stavů a přechodů (state-transition) vychází ze znázornění chování systému stavovým diagramem (síťovým grafem). Jeho uzly jsou stavy, hrany představují přechody mezi nimi. Jednotlivým událostem mohou být přiřazeny různé přechody v závislosti na dodatečné podmínce. Každý přechod mezi stavy může být spojen s akcí, kterou přechod aktivuje. Nevýhoda je v tom, že je třeba vyšetřit všechny možné přechody mezi stavy, přičemž se snadno ztrácí přehled o jejich vzájemné koordinaci. Jednotlivé uzly a hrany jsou reprezentovány algoritmy, které mohou být napsány ve kterémkoliv z ostatních jazyků. [6]



Obr. 14 Příklad SFC [13]

II. PRAKTICKÁ ČÁST

4 MODEL VÝTAHU

Pracoviště tvoří stůl s modelem pětipatrového dvoukomorového výtahu s otevíratelnými dveřmi řízeného PLC Tecomat Foxtrot a počítač určený pro programování PLC. PLC je umístěno přímo pod deskou pracovního stolu ve speciální uzamykatelné skřínce. Podrobnější popis modelu a práce s ním je uveden v diplomové práci Ondřeje Vrzala [7].



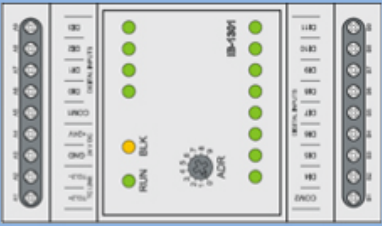
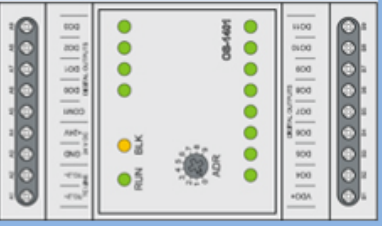
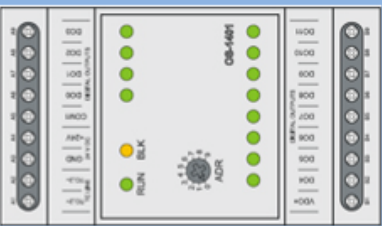
Obr. 15 Pracoviště s modelem výtahu [7]

4.1 Řídicí PLC

Řídicí PLC tvoří základní modul rozšířený o pět dalších periferních modulů. Tři moduly jsou pro binární výstupy a dva moduly pro binární vstupy. První ze tří výstupních modulů je napojen na LED diody v jednotlivých patrech, zbylé dva se zabývají ovládáním motorů pro otevírání dveří a jízdu kabin mezi patry. Jeden ze vstupních modulů je napojen na tlačítka v patrech, druhý potom na čidla polohy kabiny.

		Svorka	Absolutní adresa podle IEC	Význam
Základní modul CP-1005		B4	r0_p3_D1.D10	Rezervováno pro panel tlačítek kabin
		B5	r0_p3_D1.D11	
		B6	r0_p3_D1.D12	
		B7	r0_p3_D1.D13	
		B8	r0_p3_D1.D14	
		B9	r0_p3_D1.D15	
		D2	r0_p3_D0.D00	Rezervováno pro panel tlačítek kabin
		D3	r0_p3_D0.D01	
		D4	r0_p3_D0.D02	
		D7	r0_p3_D0.D03	
		D8	r0_p3_D0.D04	
		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		A9	r1_p0_D0.D03	
		B2	r1_p1_D1.D14	Podlaží 3NP pravé
		B3	r1_p1_D1.D15	Podlaží 4NP pravé
		B4	r1_p1_D1.D16	Podlaží 5NP pravé
		B5	r1_p1_D1.D17	Podlaží 1NP levé
		B6	r1_p1_D1.D18	Podlaží 2NP levé
		B7	r1_p1_D1.D19	Podlaží 3NP levé
		B8	r1_p1_D1.D110	Podlaží 4NP levé
		B9	r1_p1_D1.D111	Podlaží 5NP levé
		A6	r1_p2_D1.D10	Rezervováno pro panel tlačítek kabin
		A7	r1_p2_D1.D11	
		A8	r1_p2_D1.D12	
		A9	r1_p2_D1.D13	Podlaží 2NP pravé

Obr. 16 Přiřazení adres PLC - část první [7]

		Svorka	Absolutní adresa podle IEC	Význam
Binární vstupy IB-1301	Adresa: 2			
Binární výstupy OS-1401	Adresa: 3			
Binární výstupy OS-1401	Adresa: 4			

Obr. 17 Přiřazení adres PLC - část druhá [7]

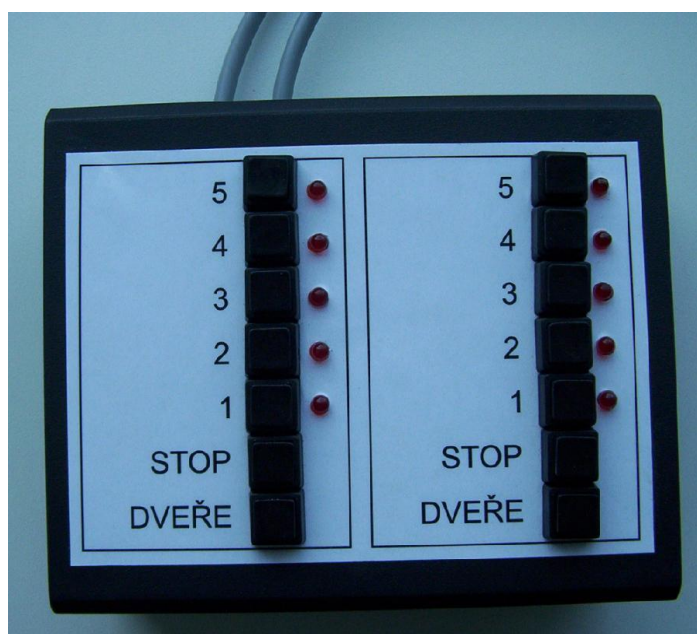
5 ROZŠÍŘENÍ MODELU

5.1 Čidla polohy

K výtahu byla přidána do každého mezipatra čidla, která budou tvořit pro každou kabinu jeden společný vstup do PLC. Pomocí těchto čidel bude moci student například lépe řídit rychlost kabiny při dojezdu. Čidla polohy jsou tvořena infračervenou diodou a fototranzistorem citlivým na infračervené záření. 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.

5.2 Panel tlačítek

Dalším krokem byl návrh modulu simulujícího řídicí panely obou kabin. Panel obsahuje pět LED diod a sedm tlačítek pro každou kabinu. Pět pro jízdu do jednotlivých pater a dvě například pro STOP tlačítko, nebo otevírání dveří. Pro nedostatek vstupů se řídicí PLC doplnil o další modul binárních vstupů, aby se řídicí panel mohl připojit. Panel je připojen dvěma kabely přes patnáctipinové Canon konektory (VGA), takže jej lze odpojit a uložit, pokud se s modelem výtahu nepracuje. Jedním kabelem je vedené napětí do panelu přes tlačítka a zpět na vstupy PLC. Druhým kabelem jsou vedeny výstupy z PLC na LED diody a jejich uzemnění. Diody jsou v sérii zapojeny s odporem 1,1 k Ω .



Obr. 18 Řídicí panel kabin

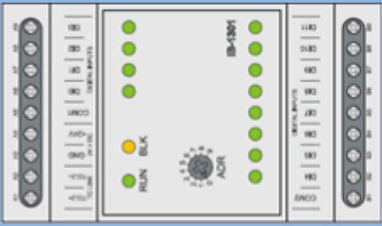
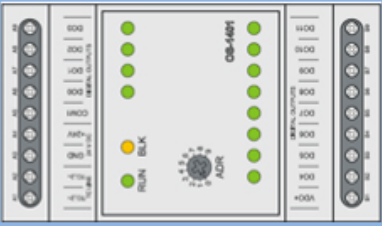
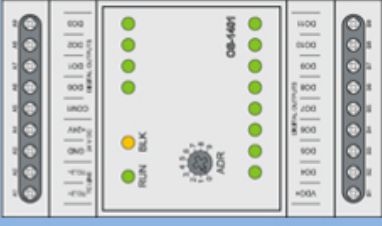

5.3 Zapojení PLC

Při zapojování LED diod jsem se snažil vyhnout základnímu modulu, protože jsou výstupy spínány pomocí relé, na rozdíl od periferního modulu binárních výstupů, kde jsou tyto spínány pomocí tranzistorů.

Po připojení nových čidel a řídicího panelu kabin vypadá zapojení řídicího PLC následovně.

		Svorka	Absolutní adresa podle IEC	Význam
Základní modul CP-1005		B4	r0_p3_D1.DI0	nevyužito
		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	nevyužito
		D2	r0_p3_DO.DO0	
		D3	r0_p3_DO.DO1	
		D4	r0_p3_DO.DO2	
		D7	r0_p3_DO.DO3	
		D8	r0_p3_DO.DO4	
		D9	r0_p3_DO.DO5	
Binární výstupy OS-1401 Adresa: 0		B2	r0_p3_A00	nevyužito
		B3	r0_p3_A01	nevyužito
		A6	r1_p0_DO.DO0	Kabina LED 5NP levá
		A7	r1_p0_DO.DO1	Kabina LED 5NP pravá
		A8	r1_p0_DO.DO2	nevyužito
		A9	r1_p0_DO.DO3	nevyužito
		B2	r1_p0_DO.DO4	LED 2NP nahoru
		B3	r1_p0_DO.DO5	LED 1NP
		B4	r1_p0_DO.DO6	LED 2NP dolů
Binární vstupy IB-1301 Adresa: 1		B5	r1_p0_DO.DO7	LED 3NP dolů
		B6	r1_p0_DO.DO8	LED 3NP nahoru
		B7	r1_p0_DO.DO9	LED 4NP dolů
		B8	r1_p0_DO.DO10	LED 4NP nahoru
		B9	r1_p0_DO.DO11	LED 5NP
		A6	r1_p1_D1.DI0	Mezipatro pravé
		A7	r1_p1_D1.DI1	Mezipatro levé
		A8	r1_p1_D1.DI2	Podlaží 1NP pravé
		A9	r1_p1_D1.DI3	Podlaží 2NP pravé
		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

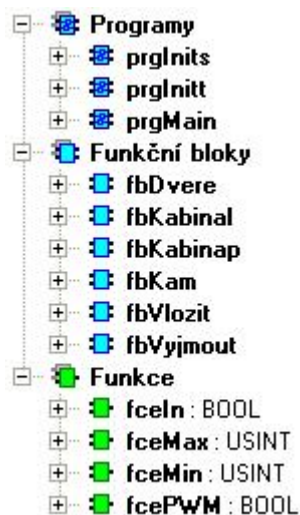
Obr. 19 Zapojení PLC – část první [7]

		Svorka	Absolutní adresa podle IEC	Význam
Binární vstupy IB-1301 Adresa: 2		A6	r1_p2_D1.DI10	nevyužito
		A7	r1_p2_D1.DI11	nevyužito
		A8	r1_p2_D1.DI12	Kabina dveře pravé
		A9	r1_p2_D1.DI13	Kabina dveře levé
		B2	r1_p2_D1.DI14	Tlačítko 5NP
		B3	r1_p2_D1.DI15	Tlačítko 4NP nahoru
		B4	r1_p2_D1.DI16	Tlačítko 4NP dolů
		B5	r1_p2_D1.DI17	Tlačítko 3NP nahoru
		B6	r1_p2_D1.DI18	Tlačítko 3NP dolů
		B7	r1_p2_D1.DI19	Tlačítko 1NP
Binární výstupy OS-1401 Adresa: 3		A6	r1_p3_DO.DO0	Kabina LED 1NP levá
		A7	r1_p3_DO.DO1	Kabina LED 2NP levá
		A8	r1_p3_DO.DO2	Kabina LED 3NP levá
		A9	r1_p3_DO.DO3	Kabina LED 4NP levá
		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
Binární výstupy OS-1401 Adresa: 4		A6	r1_p4_DO.DO0	Kabina LED 1NP pravá
		A7	r1_p4_DO.DO1	Kabina LED 2NP pravá
		A8	r1_p4_DO.DO2	Kabina LED 3NP pravá
		A9	r1_p4_DO.DO3	Kabina LED 4NP pravá
		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
Binární vstupy IB-1301 Adresa: 5		A6	r1_p5_DI.DI10	Kabina tl. STOP levé
		A7	r1_p5_DI.DI11	Kabina tl. 1NP levé
		A8	r1_p5_DI.DI12	Kabina tl. 2NP levé
		A9	r1_p5_DI.DI13	Kabina tl. 3NP levé
		B2	r1_p5_DI.DI14	Kabina tl. 4NP levé
		B3	r1_p5_DI.DI15	Kabina tl. 5NP levé
		B4	r1_p5_DI.DI16	Kabina tl. STOP pravé
		B5	r1_p5_DI.DI17	Kabina tl. 1NP pravé
		B6	r1_p5_DI.DI18	Kabina tl. 2NP pravé
		B7	r1_p5_DI.DI19	Kabina tl. 3NP pravé
		B8	r1_p5_DI.DI110	Kabina tl. 4NP pravé
		B9	r1_p5_DI.DI111	Kabina tl. 5NP pravé

Obr. 20 Zapojení PLC – část druhá [7]

6 PROGRAMOVÉ VYBAVENÍ

Program pro ovládání výtahu je psán podle normy IEC 61 131-3 v jazyce strukturovaného textu a skládá se z následujících POU.



Obr. 21 POU

6.1 Globální proměnné

Pro tento program byly definovány následující globální proměnné a datové typy.

```

TYPE
  TUsintArray : ARRAY[1..5] OF USINT;           //definice typu pole
END_TYPE

VAR_GLOBAL
  init : BOOL := 1;                             //inicializace
  otvp : BOOL := 0;                             //otevirani pravych dveri
  otvl : BOOL := 0;                             //           levych
  kp : USINT := 1;                              //poloha prave kabiny
  kl : USINT := 1;                              //           leve
  kamp : USINT := 0;                            //kam ma prava kabina prijet
  kaml : USINT := 0;                            //           leva
  i : USINT;                                    //na PWM
  prava : TUsintArray := [0,0,0,0,0];           //pozadavky na pravou kabinu
  leva : TUsintArray := [0,0,0,0,0];           //           levou
  dol : ARRAY[1..5] OF BOOL := [0,0,0,0,0];    //pozadavky pro jizdu dolu
  nah : ARRAY[1..5] OF BOOL := [0,0,0,0,0];    //           nahoru
END_VAR
  
```

Obr. 22 Globální proměnné

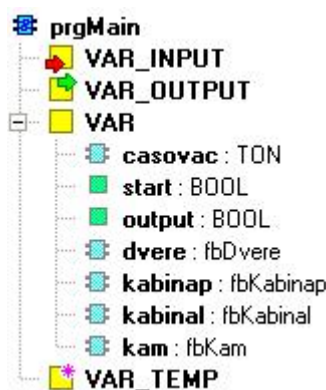
6.2 Programy

6.2.1 prgInits, prgInitt

Tyto programy se vykonají pouze jednou a to vždy při restartu PLC (studený – *prgInits*, teplý – *prgInitt*). Spustí oba motory směrem dolů, pokud ovšem nejsou již obě kabiny v prvním patře. Pak se motory nespustí a indikátor inicializace (proměnná *init*) nastaví na hodnotu 0. Neobsahuje žádné lokální proměnné, pracuje pouze s globálními a se vstupy a výstupy PLC.

6.2.2 prgMain

Hlavní program. Obsahuje inicializační časovač a jeho řídicí proměnné, dále pak proměnné ke spouštění funkčních bloků.



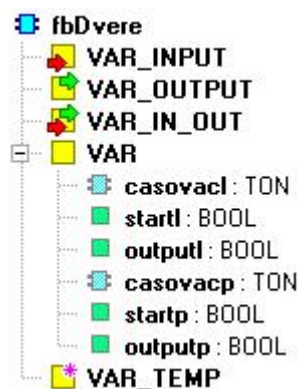
Obr. 23 prgMain

Program se dělí na inicializační a hlavní část. Inicializační část se vykonává, pokud je proměnná *init* nastavena na hodnotu 1 (implicitně) a zařizuje, aby obě kabiny přijeli do prvního patra. Jakmile tam obě kabiny jsou, *init* se nastaví na hodnotu 0 a již probíhá hlavní část programu. V hlavní části se postupně spouští funkční blok *fbKam*, který řeší stisknutí tlačítek a kam má která kabina přijet, a funkční bloky kabin (*fbKabinap* a *fbKabinal*), které podle požadavků řídí jízdu jednotlivých kabin. Tyto funkční bloky by mohly být psány přímo v hlavním programu, ale pro přehlednost jsou psány zvlášť. Dále program volá funkční blok *fbDvere* při stisknutí příslušného tlačítka.

6.3 Funkční bloky

6.3.1 fbDvere

Obsahuje časovače pro otevírání a zavírání jednotlivých dveří a jejich řídicí proměnné.



Obr. 24 fbDvere

Funkční blok na základě indikátoru otevírání dveří spustí příslušný časovač a vybere dveře, které se mají otevřít, podle toho, ve kterém patře se daná kabina nachází. Po uplynulém čase dveře zase zavře.

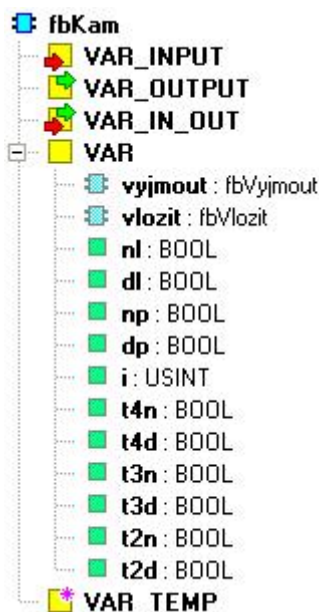
6.3.2 fbKabinal, fbKabinap

Obsahují proměnnou pro volání funkčního bloku **fbVyjmout** pro vyřazení aktuálního patra z pole požadavků pro danou kabinu.

Funkční bloky podle směru otáčení motorů a čidel polohy v patrech určují aktuální polohu kabiny. Dále na základě aktuální polohy kabiny a aktuálního požadavku na danou kabinu spouští příslušné motory.

6.3.3 fbKam

Obsahuje proměnné pro volání funkčních bloků **fbVyjmout** pro vyřazení a **fbVložit** pro zařazení požadavku do pole, proměnné pro indikaci jízdy jednotlivých kabin nahoru/dolů (nl-dp), pomocnou proměnnou pro různé cykly (i) a pomocné proměnné pro detekci náběžné hrany tlačítek (t4n-t2n).

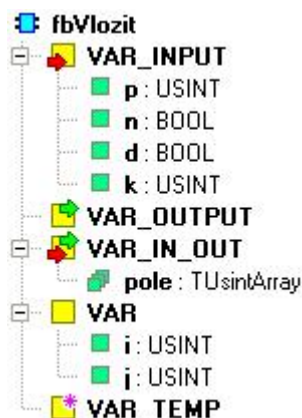


Obr. 25 fbKam

Funkční blok podle směru otáčení motorů nastavuje příslušné indikátory jízdy. Podle stisknutí tlačítek určuje která kabina má kam přijet. Pokud žádná kabina aktuálnímu požadavku nevyhovuje, požadavek čeká v poli požadavků pro jízdu nahoru/dolů, dokud se nenajde vhodná kabina. Při stisku tlačítka STOP funkční blok zastaví danou kabinu a její požadavky předá druhé kabině. Tento funkční blok řeší také rozsvěcování a zhasínání příslušných LED diod.

6.3.4 fbVlozit

Jako vstupní proměnné přebírá požadované patro (p), indikátory jízdy nahoru/dolů (n,d) a aktuální polohu příslušné kabiny (k). Jako vstupně-výstupní proměnnou přebírá pole požadavků na danou kabinu, do kterého bude požadované patro vkládat a obsahuje lokální pomocné proměnné pro cykly.



Obr. 26 fbVlozit

Funkční blok vkládá požadavky do pole požadavků na příslušnou kabinu podle směru jízdy tak, aby kabina nejezdila zbytečně nahoru a dolů, ale uspořádá požadavky tak, aby kabina po cestě jedním směrem zastavila ve všech požadovaných patrech.

6.3.5 fbVymout

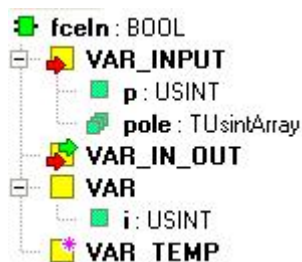
Jako vstupně-výstupní proměnnou přebírá pole požadavků na danou kabinu, ze které má být odstraněn již splněný požadavek a obsahuje lokální pomocnou proměnnou pro cykly.

Funkční blok posune všechny prvky v poli na index o jedna nižší, tím se zbaví prvního prvku, a na poslední pozici vloží 0.

6.4 Funkce

6.4.1 fceIn

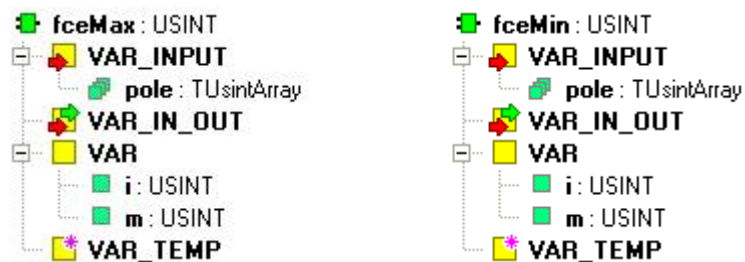
Jako vstupní proměnnou přebírá patro a pole požadavků a obsahuje pomocnou proměnnou pro cykly. Funkce vrací hodnotu 1, pokud se patro v seznamu nachází a hodnotu 0, pokud ne.



Obr. 27 fceIn

6.4.2 fceMax, fceMin

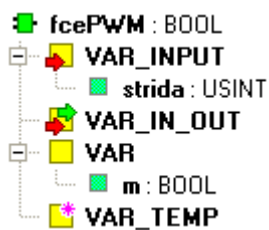
Jako vstupní proměnnou přebírají pole požadavků a obsahují pomocné proměnné pro cykly a uložení dočasné hodnoty. Funkce *fceMax* vrací nejvyšší a funkce *fceMin* nejnižší patro v seznamu požadavků.



Obr. 28 fceMax, fceMin

6.4.3 fcePWM

Obsahuje pomocnou proměnnou pro uložení dočasné hodnoty a jako vstupní proměnnou přebírá střidu, pomocí níž se na výstupu generuje určitou dobu hodnoty 1 a 0.



Obr. 29 fcePWM

7 VZOROVÉ ÚLOHY

Vzorové úlohy jsou psány podle normy IEC 61 131-3 strukturovaným textem. Celkem 3 úlohy jsou psány od nejlehčí, až po těžší. Každá další úloha je rozšířením té předchozí. Vzorové úlohy se nachází na CD přiloženém k práci.

7.1 Úloha 1

Vytvořte program, který pomocí modulu řídicího panelu kabin přivolá příslušnou kabinu do zadaného patra. Cílové patro indikujte rozsvícením LED diody na panelu.

7.2 Úloha 2

K programu připojte funkční blok (část programu, dle vlastního uvážení), který po přijetí do zadaného patra otevře a zavře dveře.

7.3 Úloha 3

Rozšiřte program tak, aby kabina sbírala patra, která jsou po cestě a případné jiné požadavky správně zařadila.

8 WEB SERVER

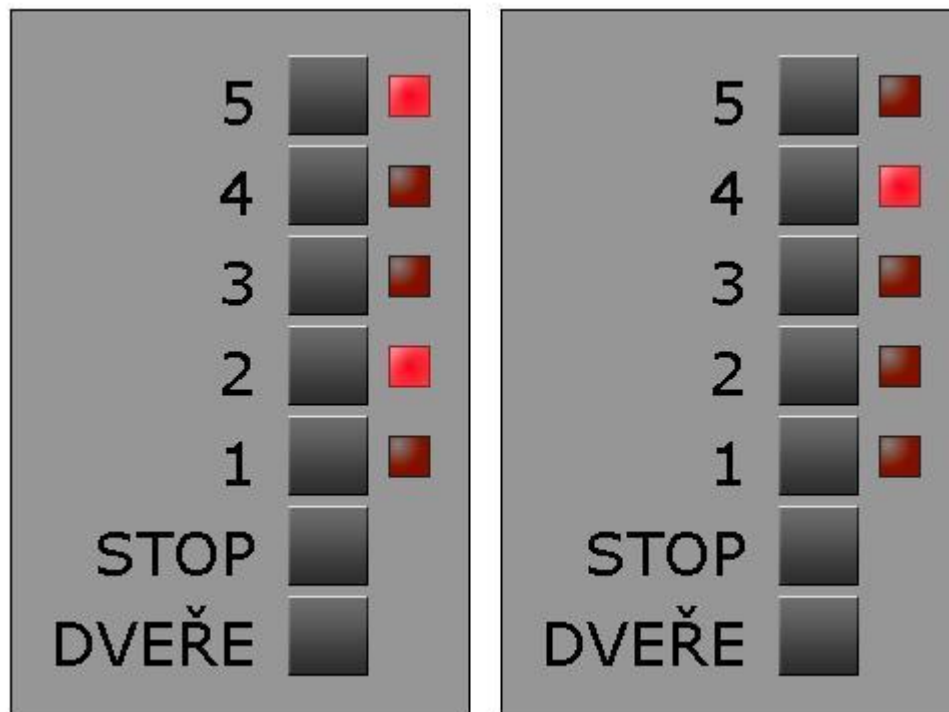
PLC Tecomat řady Foxtrot obsahuje také zabudovaný web server, který umožňuje sledování a řízení stavu procesu pomocí běžných internetových prohlížečů. Pro jeho využití je potřeba paměťová karta pro uložení souborů webových stránek.

Pro vytváření stránek ve vývojovém prostředí Mosaic se používá nástroj WebMaker, který obsahuje grafický editor umožňující vkládání obrázků, textů a proměnných z uživatelského programu v PLC.

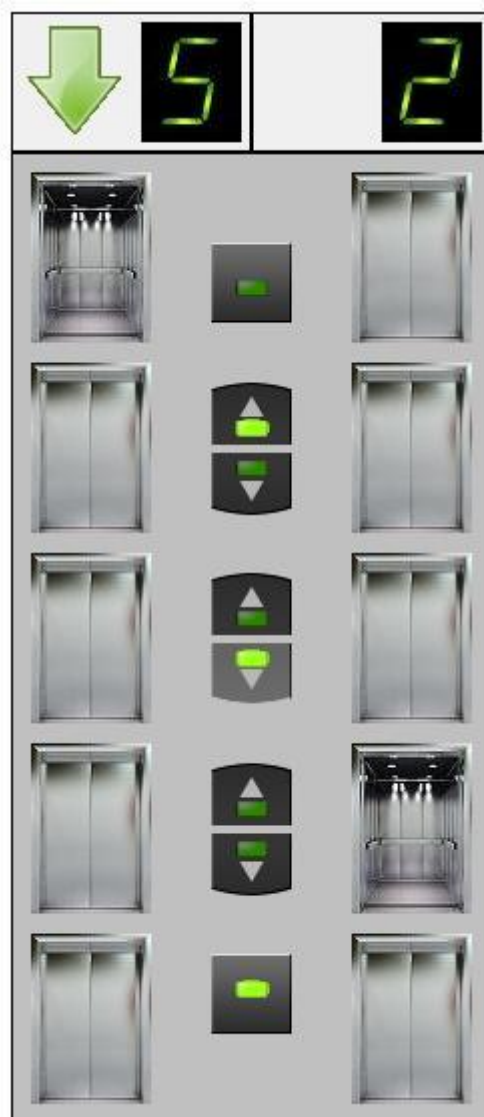
8.1 WebMaker

WebMaker slouží tvorbě XML stránek pro webový server a k zobrazování a pohodlnému nastavování vybraných proměnných v programu.

Pomocí nástroje WebMaker jsem vytvořil stránky pro řízení pomocí simulace řídicího panelu z pohledu cestujícího ve výtahu a pomocí vizualizace modelu z pohledu čekajícího na výtah.



Obr. 30 Vizualizace řídicího panelu



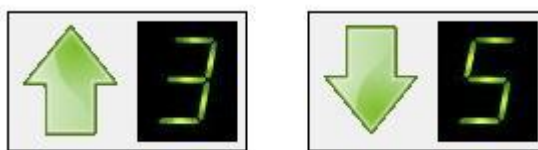
Obr. 31 Vizualizace modelu

8.2 Zabezpečení

WebMaker umožňuje nastavit 10 dvojic Uživatelské jméno – Heslo. Tyto údaje budou vyžadovány při přístupu k web serveru přes webový prohlížeč. Každé dvojici lze nastavit úroveň přístupu (0 – 9, vyšší číslo znamená vyšší práva) a výchozí stránku, která se objeví při přihlášení. Hodnota -1 značí neaktivní položku.

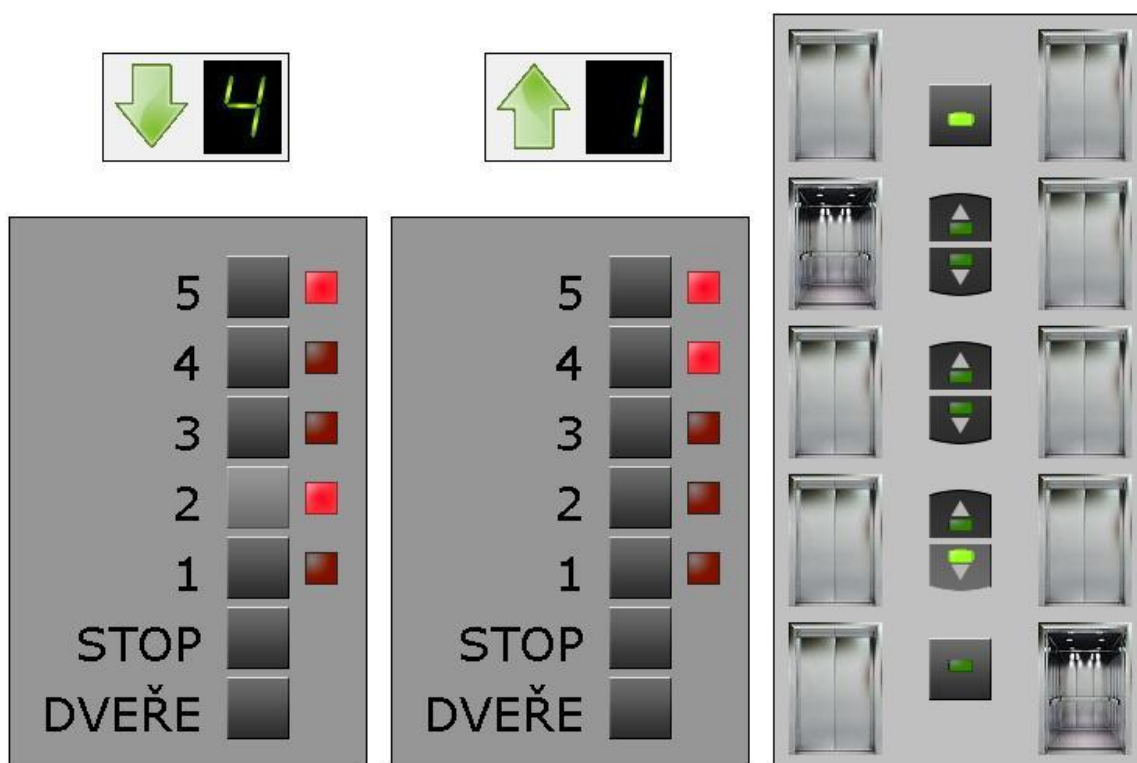
Pro tuto vizualizaci jsem definoval čtyři různé přístupové úrovně s prioritami 0, 1 a 2 pro uživatele a prioritou 9 pro administrátora.

Uživatel s prioritou 0 může vidět pouze informativní okénko.



Obr. 32 Informační okénko

Uživatel s prioritou 1 může již využít vizualizaci panelu, nebo vizualizaci modelu a uživatel s prioritou 2 může sledovat a řídit vše dohromady.



Obr. 33 Souhrnný pohled

ZÁVĚR

Cílem této práce bylo rozšířit model výtahu pro výuku PLC o další čidla a modul simulující řídící panely obou kabin, jeho naprogramování, vypracování vzorových úloh a vizualizace modelu.

V teoretické části zabýval popisem a rozdělením programovatelných automatů a jejich vývojem. Podrobněji jsem popsal PLC Tecomat Foxtrot, který je použit k řízení modelu výtahu, použité periferní moduly a základní jednotku. V poslední části teorie je popsána norma IEC 61 131-3, její společné prvky a programovací jazyky. Program pro řízení výtahu je psán podle této normy.

V praktické části jsem se zabýval popisem modelu výtahu, jeho vstupů, výstupů a rozšíření. Model byl rozšířen o čidla v mezipatrech a o modul simulující řídící panely obou kabin. Byly vypracovány na sebe navazující úlohy pro studenty. Byla také vytvořena webová aplikace pomocí zabudovaného web serveru, pomocí kterého to lze model sledovat i řídit na dálku.

ZÁVĚR V ANGLIČTINĚ

The object of this work was to extend the elevator model for learning PLC by another sensors and module simulating control panels of both cabins, its programming, develop sample tasks and visualization of the model.

In theoretical part I was involved in description a sorts of PLC and its evolution. I have described in detail PLC Tecomat Foxtrot, which is used to control model of the elevator, its peripheral modules and basic module. Last part of theory is about standard IEC 61 131-3, its common elements and programming languages. Program for control the elevator model is written according to it.

In the practical part is I was involved in describing the elevator model, its inputs, outputs and its extension. The model was extended to include sensors in the mezzanine and module simulating control panels of both cabins. At each follow-up tasks for students were developed. There also was created a Web application using the built-in web server for remote watching and control the model.

SEZNAM POUŽITÉ LITERATURY

- [1] ŠMEJKAL, Ladislav a Marie MARTINÁSKOVÁ. PLC a automatizace, 1.díl. 1. Vyd. Praha: BEN - technická literatura, 1999. ISBN 80-860-5658-9.
- [2] ŠMEJKAL, Ladislav. PLC a automatizace, 2.díl. 1. Vyd. Praha: BEN - technická literatura, 2005. ISBN 80-7300-087-3.
- [3] MARTINÁSKOVÁ, Marie a Ladislav ŠMEJKAL. Řízení programovatelnými automaty. Vyd. 2. Praha: Vydavatelství ČVUT, 2004. ISBN 80-010-2925-5.
- [4] MARTINÁSKOVÁ, Marie a Ladislav ŠMEJKAL. Řízení programovatelnými automaty II. Vyd. 1. Praha: ČVUT, Strojní fakulta, 2000. ISBN 80-010-2096-7.
- [5] TECOMAT. Programovatelné automaty TECOMAT FOXTROT. 13. Vyd. 2010.
- [6] CENDELÍN, Jiří. Historie programovatelných automatů a jejich současné efektivní použití. *Automa* [online]. 2003, roč. 4, č. 6 [cit. 2012-06-01]. Dostupné z: http://www.odbornecasopisy.cz/index.php?id_document=28831
- [7] VRZAL, Ondřej. *Sekvenční řízení modelu výtahu programovatelným automatem*. Zlín, 2010. Diplomová práce. UTB ve Zlíně, Fakulta aplikované informatiky.
- [8] Produkty. *Tecomat, PLC for machine, process, technology, transport and building automation* [online]. 2009 [cit. 2012-06-01]. Dostupné z: <http://www.tecomat.com/kategorie-12-produkty.html>
- [9] TECOMAT. Příručka programátora PLC TECOMAT. 13. Vyd. 2007.
- [10] KOVÁŘ, Josef, Zuzana PROKOPOVÁ a Ladislav ŠMEJKAL. *Programování PLC*.
- [11] TECOMAT. Periferní moduly PLC TECOMAT FOXTROT. 4. Vyd. 2011.
- [12] TECOMAT. Programování PLC podle normy IEC 61 131-3 v prostředí Mosaic. 10. Vyd. 2007.
- [13] URBAN, Luboš. Programování PLC podle normy IEC EN 61131-3 – víc než jednotné jazyky. *Automa* [online]. 2005, roč. 6, č. 2 [cit. 2012-06-01]. Dostupné z: http://www.odbornecasopisy.cz/index.php?id_document=30310

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

PLC	Programmable logic controller – anglický výraz pro programovatelný automat
IEC	International Electrotechnical Commission
SPS	Speicherprogrammierbare Steuerung – německý výraz pro programovatelný automat
PC	Programmable controller – jiný anglický výraz pro programovatelný automat
PA	Programovatelný automat
GRAFCET	Graphe Fonctionnel de Connexion Etapes Transitions
EPROM	Erasable Programmable Read-Only Memory
EEPROM	Electrically Erasable Programmable Read-Only Memory
RAM	Random-access memory
GSM	Groupe Spécial Mobile – standard pro mobilní komunikaci
PID	Proportional-integral-derivative
POU	Program Organisation Unit
IL	Instruction List – jazyk seznamu instrukcí
ST	Structured Text – jazyk strukturovaného textu
LD	Ladder Diagram – jazyk kontaktních schémat
FBD	Function Block Diagram – jazyk funkčního blokového schématu
SFC	Sequential Function Chart – sekvenční funkční graf
LED	Light-emitting diode – světelná dioda

SEZNAM OBRÁZKŮ

Obr. 1 Kompaktní PLC [8]	14
Obr. 2 Modulární PLC [8]	14
Obr. 3 Blokové schéma PLC [3]	15
Obr. 4 Cyklus PLC [5]	18
Obr. 5 Schéma aktivace procesů [9]	20
Obr. 6 Základní modul CP-1005 [5]	22
Obr. 7 Periferní modul IB-1301 (staré a nové provedení) [11]	24
Obr. 8 Periferní modul OS-1401 (staré a nové provedení) [11]	26
Obr. 9 Logická funkce ANDN ve čtyřech základních jazycích [12]	32
Obr. 10 Program v jazyce IL [12]	33
Obr. 11 Program v jazyce ST [12]	34
Obr. 12 Program v jazyce LD [12]	35
Obr. 13 Grafika obvodu v jazyce FBD [12]	36
Obr. 14 Příklad SFC [13]	37
Obr. 15 Pracoviště s modelem výtahu [7]	39
Obr. 16 Přiřazení adres PLC - část první [7]	40
Obr. 17 Přiřazení adres PLC - část druhá [7]	41
Obr. 18 Řídicí panel kabin	42
Obr. 19 Zapojení PLC – část první [7]	43
Obr. 20 Zapojení PLC – část druhá [7]	44
Obr. 21 POU	45
Obr. 22 Globální proměnné	45
Obr. 23 prgMain	46
Obr. 24 fbDvere	47
Obr. 25 fbKam	48
Obr. 26 fbVlozit	49
Obr. 27 fceIn	49
Obr. 28 fceMax, fceMin	50
Obr. 29 fcePWM	50
Obr. 30 Vizualizace řídicího panelu	52
Obr. 31 Vizualizace modelu	53

Obr. 32 Informační okénko.....	54
Obr. 33 Souhrnný pohled.....	54

SEZNAM TABULEK

Tab. 1 Přehled procesů uživatelského programu a jejich určení [9]	19
Tab. 2 Popis základního modulu CP-1005 [5].....	22
Tab. 3 Parametry modulu CP-1005 [5].....	23
Tab. 4 Parametry modulu IB-1301 [11].....	25
Tab. 5 Parametry modulu OS-1401 [11].....	26
Tab. 6 Elementární datové typy [12]	28
Tab. 7 Rodové datové typy [12].....	29
Tab. 8 Třídy proměnných [12].....	30
Tab. 9 Operátory v jazyce strukturovaného textu ST [12].....	34

SEZNAM PŘÍLOH

P I CD bakalářské práce

PŘÍLOHA P I: CD BAKALÁŘSKÉ PRÁCE

Na CD se nachází bakalářská práce, vzorové úlohy, program pro řízení modelu a jeho vizualizaci.