

Řízení modelu kolejiště pomocí programovatelného automatu

Jiří Kubáček

Bakalářská práce
2005



Univerzita Tomáše Bati ve Zlíně
Fakulta technologická

ABSTRAKT

Cílem této práce bylo připravit model kolejiště pro použití ve výuce, konkrétně pro výuku ovládání programovatelného automatu SAIA.

Bylo potřeba obnovit připojení automatu jak ke kolejišti, tak i k vývojovému a vizualizačnímu PC. Dále pak vytvořit přehledný nástěnný obraz kolejiště s popsányými vstupy a výstupy i s jejich adresami, na kterých jsou k automatu připojeny. Byly také vytvořeny vzorové programy s návody na jejich řešení.

V neposlední řadě byly provedeny opravy nefunkčních částí kolejiště, zejména výhybek.

Při řízení modelu kolejiště byly ukázány možnosti využití programovatelného automatu.

The purpose of this thesis was to prepare model of railway for educational use, exactly for learning handling of programmable controller SAIA.

There was necessary to re-join controller to railway and to PC, too. Then it was needy to make comprehensible picture of railway with designation of inputs and outputs and their addresses at controller. Several of example programmes has been made too.

Last but not least there have been made some repairing of un-function parts of railway, mostly rail-switches.

There have been presents possibilities of using programmable controller.

Děkuji vedoucímu práce ing. Tomáši Sysalovi, PhD. za odborné vedení a pomoc v průběhu řešení této práce.

Souhlasím s tím, že s výsledky mé práce může být naloženo podle uvážení vedoucího diplomové práce a ředitele ústavu. V případě publikace budu uveden jako spoluautor.

Prohlašuji, že jsem na celé diplomové práci pracoval samostatně a použitou literaturu jsem citoval.

Ve Zlíně dne 3.6.2005

.....

podpis

OBSAH

OBSAH	5
ÚVOD.....	8
I. TEORETICKÁ ČÁST	9
1 ŘÍZENÍ POMOCÍ PLC	10
1.1 ÚVOD DO PROBLEMATIKY PLC	10
1.1.1 Základní pojmy	10
1.1.2 Zařazení programovatelného automatu mezi řídicí systémy	11
1.1.3 Hlavní charakteristiky programovatelných automatů	12
1.1.4 Typy programovatelných automatů z hlediska velikosti a konstrukčního uspořádání	14
1.2 RODINA AUTOMATŮ SAIA PCD	15
1.3 PROGRAMOVATELNÝ AUTOMAT SAIA PCD2	16
1.3.1 Architektura PCD2.....	16
1.3.2 Dodávané typy vstupně-výstupních modulů	17
1.3.3 Typy datových prostředků využitelných v programech.....	18
1.3.4 Struktura uložení programu.....	19
2 VÝVOJOVÉ PROSTŘEDÍ PG4.....	21
2.1.1 Instruction List (IL).....	23
2.1.2 Function Plan - FUPLA (FBD)	23
2.1.3 Graftec(SFC)	25
3 POPIS MODELU KOLEJIŠTĚ A JEHO PROPOJENÍ S PLC	26
3.1 FUNKČNÍ USPOŘÁDÁNÍ	26
3.2 OVLÁDACÍ SKŘÍŇ	26
3.3 PROPOJENÍ ŘÍDICÍHO PLC S OVLÁDACÍ SKŘÍŇÍ MODELU	26
3.4 OVLÁDÁNÍ PRVKŮ MODELU KOLEJIŠTĚ.....	27
3.4.1 Význam logických hodnot	25
3.4.2 Seznam adres prvků	25
3.5 PROPOJENÍ ŘÍDICÍHO PLC S PC	29
4 POUŽITÉ MODULY	31
4.1 MODULY OVLÁDÁNÍ ŽELEZNIČNÍCH PŘEJEZDŮ	31
4.2 MODULY OVLÁDÁNÍ RYCHLOSTI A SMĚRU JÍZDY VLAKU	31
II. PRAKTICKÁ ČÁST.....	33
5 VZOROVÉ PROGRAMY	34
5.1 PŘESUN VLAKU V RÁMCI NÁDRAŽÍ CHYBA! ZÁLOŽKA NENÍ DEFINOVÁNA.	
5.2 JÍZDA DVOU VLAKŮ PO OKRUHU SE SPOLEČNOU KOLEJÍ	35
ZÁVĚR.....	36
SEZNAM POUŽITÉ LITERATURY.....	37
SEZNAM OBRÁZKŮ	38
SEZNAM TABULEK.....	39

SEZNAM PŘÍLOH.....	40
---------------------------	-----------

ÚVOD

V dnešní době se setkáváme s automatickým řízením téměř na každém kroku. Významná část různých procesů je řízena programovatelnými automaty (PLC), a díky tomu jsou stále žádanější programátoři těchto zařízení. Jedním z úkolů naší fakulty je takové programátory školit. A protože navzdory všem změnám stále platí, že hrou se člověk nejlépe učí, byl zde před několika lety vytvořen model kolejiště. Ten je neustále obdivován nejen studenty, ale i zaměstnanci a návštěvníky školy, proto jistě umožní učinit výuku příjemnou a zábavnou.

Výroba i zdokonalování kolejiště proběhly v rámci několika diplomových prací, na které jsem volně navazoval a některých jejich výsledků využíval – například využitím modulů pro řízení rychlosti jízdy a ovládání výstražných světel.

Úkolem této práce bylo připravit model pro již zmíněnou výuku programování PLC, v tomto případě šlo o kvalitní švýcarský automat SAIA PCD2. K tomu bylo zapotřebí zejména opravit některé nefunkční části kolejiště, obnovit připojení automatu ke kolejišti a připojit jej k osobnímu počítači za účelem vývoje aplikací a jejich případné vizualizace. Dále chyběl obrázek kolejiště s popsány vstupy a výstupy a adresami, na kterých jsou připojeny k automatu.

Aby samotné řešení nabylo smyslu, musí jeho výsledky být uvedeny do praxe. Poslední částí této práce byla proto příprava několika demonstračních úloh.

I. TEORETICKÁ ČÁST

1 ŘÍZENÍ POMOCÍ PLC

1.1 ÚVOD DO PROBLEMATIKY PLC

1.1.1 Základní pojmy

Programovatelný automat je uživatelsky programovatelný řídicí systém přizpůsobený pro řízení průmyslových a technologických procesů nebo strojů, mnohdy specializovaný na úlohy převážně logického typu (obzvláště u starších typů nebo u nejmenších systémů). 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, která se teprve začíná používat, je PA (Programovatelný automat).

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

Každý programovatelný automat se v podstatě skládá z centrální procesorové jednotky, systémové paměti, souboru vstupních a výstupních jednotek pro připojení řízeného systému a souboru komunikačních jednotek pro komunikaci se souřadnými i nadřazenými řídicími systémy. Jednotky programovatelného automatu jsou navzájem propojeny systémovou sběrníci.

Řídicí algoritmy jsou realizovány uživatelským programem, který může být zapsán v různých programovacích jazycích a po přeložení je uložen v uživatelské paměti programovatelného automatu. Program obsahuje posloupnost instrukcí, které procesor vykonává cyklicky.

Chování programovatelného automatu je tedy dáno v podstatě zaměnitelným programem, zatímco u reléových systémů bylo chování určeno strukturou zapojení jednotlivých komponent, která byla po realizaci reléového systému téměř nezměnitelná. Proto se zvláště v počátcích vývoje programovatelných automatů hovořilo o tzv. volně programovatelné logice a mnoho programovatelných automatů se označovalo zkratkou FPC (Free Programmable Controller).

1.1.2 Zařazení programovatelného automatu mezi řídicí systémy

Relé a kontakty versus programovatelné automaty

Relé, stykače a tlačítka jsou v některých případech nenahraditelné a nemá smysl se bránit jejich použití i v případech, kdy je k řízení použit programovatelný automat. Z bezpečnostních důvodů se takto realizují bezpečnostní okruhy, např. obvod CENRAL STOP. Nemá smysl bránit se příležitostnému vytváření logických funkcí prostým propojením kontaktů, obzvláště, pokud tím ušetříme počty vstupů a výstupů programovatelného automatu a snížíme tak jeho cenu. Rozsáhlejší funkce se kontaktní a reléovou technologií již nerealizují a svěříme je důsledně programu programovatelného automatu.

Regulátory versus programovatelné automaty

Rozdíl mezi těmito dvěma tradičními kategoriemi výrobků - mezi regulátory a programovatelnými automaty - se postupně stírá. Obě dvě skupiny těchto dnes vyráběných výrobků pracují číslicově. Hranice je tak neostrá, že je mnohdy obtížné rozhodnout, zda produkt je ještě regulátorem nebo již programovatelným automatem.

Výrobci programovatelných automatů postupně expandují do aplikačních oblastí dosud vyhrazených pro regulátory, a to jak aplikacemi tradičních PLC, tak i vývojem nových výrobků, které se mnohdy odklánějí od původní koncepce PLC. Na druhé straně tradiční výrobci regulátorů naopak své programovatelné regulátory uzpůsobují i pro logické řízení. Vyrábějí např. inteligentní regulátory, které mají přidány logické funkce a jsou schopny pracovat v různých režimech podle stavu řízeného systému, a pronikají s nimi do oblastí typických pro nasazení programovatelných automatů.

Často je tedy rozlišení mezi programovatelným automatem a regulátorem v dnešní době spíše formální a je záležitostí tradice a firemní obchodní politiky.

Personální počítače versus programovatelné automaty

Někdy se setkáváme s přímým řízením technologických procesů standardním PC, mnohdy umístěným přímo v technologii. Toto řešení je přinejmenším riskantní a diskutabilní.

Běžný počítač kategorie PC je produkt spotřební elektroniky a je konstruován pro provoz v prostředí domácností, laboratoří a kanceláří, kde obvykle funguje s vyhovující spolehlivostí. V drsných průmyslových podmínkách mnohdy selhává (bývá málo spolehlivý, je citlivý na rušení a přepětí, nemá potřebnou životnost). Problémy vznikají už s pouhým připojením většího počtu vstupních a výstupních vodičů a s jejich odrušením.

Průmyslové počítače (IPC, IC) se někdy používají při přímém řízení strojů a technologií, někdy jen v roli inteligentního operátorského panelu nebo komunikačního adaptéru. Problémem při jejich nasazování je vysoká cena. Jsou tedy účelné jen tam, kde je zdůvodněna, zejména při archivaci a zpracování velkých objemů dat, při využití obrazovky a standardního počítačového ovládání, při využívání standardních programových produktů, při využívání výkonných komunikací, při řešení geometrických a jiných výpočetně náročných úloh.

Přímé řízení počítačem je dnes účelné jen v laboratorních podmínkách, pro potřeby výuky a řízení laboratorních a modelových úloh. Je-li v průmyslových podmínkách použití PC nezbytné, pak je nutno použít průmyslový typ. Standardním řešením je použití distribuovaného systému, kdy osobní počítač je použit ve velínu nebo na dispečerském pracovišti a do drsného průmyslového prostředí jsou předloženy programovatelné automaty.

Někteří výrobci nabízejí v sortimentu modulů svého programovatelného automatu i počítačový modul kompatibilní s PC. V něm lze odpovídajícími prostředky řešit úlohy příslušející počítači (složitě a rychlé výpočetní algoritmy, grafické a geometrické úlohy, zpracovávání a archivace velkého množství dat, databázové úlohy, výkonné komunikace, napojení do počítačové sítě).

1.1.3 Hlavní charakteristiky programovatelných automatů

VÝHODY: Rychlá realizace

Hlavní předností programovatelných automatů je možnost rychlé realizace systému. Technické vybavení nemusí uživatel vyvíjet. Stačí navrhnout a včas objednat vhodnou sestavu modulů programovatelného automatu (konfiguraci) pro danou aplikaci, vytvořit projekt, napsat a odladit uživatelský program - a pak to vše realizovat a uvést do chodu.

Spolehlivost, odolnost, diagnostika

Technické vybavení programovatelných automatů je navrženo tak, že jsou extrémně spolehlivé i v drsných průmyslových podmínkách, jsou odolné proti rušení i poruchám, vyznačují se robustností a spolehlivostí. Programovatelné automaty bývají vybaveny i vnitřními diagnostickými funkcemi, které průběžně kontrolují činnost systému a včas zjistí případnou závadu, lokalizují ji, bezpečněji ošetří a usnadní její odstranění.

Snadná přizpůsobitelnost řešení (nekončící změny v zadání)

Jen výjimečně se podaří, že první varianta řešení zůstane tou poslední a konečnou. Představy zadavatele a koncového uživatele, ale i projektanta a programátora postupně zrají, požadavky se průběžně vyvíjejí a rozšiřují. Při uvádění do provozu je třeba všechny funkce důkladně prověřit a odstranit mnohé chyby a slabá místa. Dodatečné požadavky a zadání nových funkcí vznikají i po mnohých měsících a letech rutinního provozu.

Schopnost komunikace

K neopomenutelným výhodám programovatelných automatů patří jejich schopnost komunikace s nejrůznějšími systémy a zařízeními jak v podřízené úrovni, v takzvaném poli, což je oblast senzorů, měřících zařízení a akčních členů, tak i v souřadné úrovni s ostatními programovatelnými automaty či jinými řídicími systémy, a v neposlední řadě i směrem k systémům nadřízeným. Právě tato schopnost komunikace umožňuje stavbu distribuovaných nebo i hierarchických systémů řízení z nejrůznějších komponent a od různých výrobců.

NEVÝHODY: Prodloužení odezvy

Řídicí systémy s pevnou logikou se od systémů s PLC liší v době odezvy, tj. v době, za kterou zareagují výstupy na změnu na vstupech systému. V pevné logice jsou všechny logické členy trvale aktivní, algoritmus systému se realizuje paralelně a ve spojitém čase. Odezva na změnu vstupů je dána jen celkovým zpožděním logických členů v nejdelší větvi. U integrovaných obvodů to bývají řádově nanosekundy až mikrosekundy, u reléových systémů jednotky, desítky, někdy i stovky milisekund.

Odezva PLC bývá delší a je dána dobou průchodu programu. Závisí na rychlosti procesoru a na délce aktivní větve programu. Typicky nabývá hodnot v řádu jednotek až

desítek milisekund (někdy stovek ms), což pro běžné aplikace postačuje. Je však třeba s touto skutečností počítat, aby v některých případech nebyla příčinou nečekaných překvapení.

Nespojitost v čase

Dalším důležitým znakem programovatelných systémů je časová nespojitost zpracování. Algoritmus je vykonáván cyklicky, vždy jen v určitých okamžicích. Uvnitř intervalu mezi okamžiky aktivace systém nereaguje na změny vstupních hodnot. Tuto skutečnost je třeba respektovat při návrhu a programování systému, jinak může být příčinou hazardů a chyb, ztráty krátkého vstupního impulsu, nevyhodnocení hrany signálu apod.

Postupnost zpracování

Program PLC je vykonáván v pořadí, v jakém je zapsán, nikoliv v pořadí „toku signálů“ v odpovídajícím logickém schématu. Je-li možné zapsat PLC program sousledně s tokem signálů (aby pořadí instrukcí sledovalo tok signálů směrem od vstupů k výstupům), nebývají problémy. U složitých a nepřehledných logických funkcí (a zejména v případech mnoha změn a vsuvek do programu) se to vždy nepodaří. V lepším případě je následkem prodloužení odezvy systému (k ustálení hodnoty výstupu je zapotřebí několika cyklů PLC programu). V případě nesystematického návrhu sekvenčních funkcí (se zpětnými vazbami) může být následkem i chybná funkce programu nebo jeho zdánlivě nahodilé chyby (hazardy).

1.1.4 Typy programovatelných automatů z hlediska velikosti a konstrukčního uspořádání

Mikro PLC

Nejmenší a nejlevnější kompaktní PLC systémy (mikro PLC) nabízejí uživateli pevnou sestavu vstupů a výstupů, obvykle jen binárních, v konkrétních počtech podle modelu. Uživatel se v tomto případě může rozhodnout pro jeden typ systému, který již nemůže dodatečně rozšiřovat. Svým kompaktním provedením, malými rozměry a nízkou cenou se mikro PLC řadí do kategorie spotřebního materiálu. Jejich funkční a

programátorský komfort je obvykle redukován na nezbytné minimum, komunikační možnosti mnohdy chybějí.

Kompaktní PLC

Ostatní PLC v kompaktním provedení nabízejí určitou, i když omezenou variabilnost ve volbě konfigurace. Uživatel může k základnímu modulu připojit jeden nebo několik přídatných modulů z omezeného sortimentu s pevnou kombinací vstupů a výstupů. Některé kompaktní systémy se navíc vyznačují ještě vnitřní modulárností, kdy konfiguraci základního modulu lze sestavit osazením základní desky násuvnými moduly vhodného typu, tzv. „piggybacky“. Piggyback bývá řešen jako malý plošný spoj s jedním nebo několika integrovanými obvody a konektorem.

Modulární PLC

Nesrovnatelně větší volnost ve volbě konfigurace poskytují modulární programovatelné automaty. Do základního rámu lze zasouvat libovolné moduly i rozšiřovací

moduly k připojení dalších rámu i stovky metrů vzdálených. Tak lze vytvářet různě strukturované distribuované systémy.

Programovatelné pracovní stanice

V nedávné době představila řada výrobců své kompaktní výrobky, které v sobě sdružují funkce PLC a operátorského panelu. Integrace funkcí a praktické konstrukční provedení a výhodný poměr cena/výkon poskytuje těmto systémům široké možnosti uplatnění - a to i tam, kde bylo použití tradičního PLC s odděleným operátorským panelem dosud cenově nedostupné. Je otázkou diskuse, zda tyto výrobky zařadit mezi PLC, mezi operátorské panely nebo zda tvoří samostatnou kategorii. Pro odlišení zavádějí někteří výrobci označení „pracovní stanice“, někdy s přívlastkem „inteligentní“ nebo „programovatelná“.

1.2 RODINA AUTOMATŮ SAIA PCD

Pro aplikaci řízení vlaků byl vybrán programovatelný automat SAIA PCD2 švýcarské firmy SAIA-Burgess Electronics Ltd.

Modelová řada automatů SAIA PCD (Process Control Device) má společnou základní architekturu, stejnou instrukční sadu, shodný přístup k prostředkům a perifériím. Program vytvořený pro jednu verzi PCD poběží bez úprav i na všech ostatních (samozřejmě s přihlédnutím k dostupnosti prostředků, např. počtu procesorů). Jednotlivé modely se liší velikostí a stupněm modularity:

- **PCD1** je kompaktní přístroj se čtyřmi pozicemi pro instalaci plochých vstupně-výstupních modulů v jedné rovině se základní deskou. V základním provedení je osazen jedním procesorem, 17 kB paměti rozšiřitelné až na 140 kB, může obsahovat 1 až 2 sériové kanály. Je v úspornou verzi jinak velmi podobného modelu PCD2.
- **PCD2** obdobně jako PCD1 je plochý, kompaktní, s možností osazení osmi modulů, čtyři na horním i spodním okraji základní desky. Paměť pro jediný procesor je 32 - 536 kB velká, má 1 - 4 sériová rozhraní.
- **PCD4** Již v pravém smyslu modulární, tvořen bloky montovanými na DIN lištu. 1 nebo 2 procesory, paměť 64 - 428 kB, 1 - 4 sériové porty.
- **PCD6** je tvořen moduly ve „vanách“ vkládaných do 19" skříní. 1 až 7 procesorů má k dispozici od 256 kB do 1 MB paměti, 4 až 28 sériových kanálů.

Všechny modely shodně používají procesor Motorola 68340, instrukční sada obsahuje přes 120 instrukcí včetně obsluhy komunikací a aritmetiky s plovoucí desetinnou čárkou, což je významným kritériem při realizaci číslicových regulátorů. Provedení PCD2 bylo vybráno vzhledem ke své velikosti odpovídající rozsahu aplikace.

1.3 PROGRAMOVATELNÝ AUTOMAT SAIA PCD2

1.3.1 Architektura PCD2

SAIA PCD2 je programovatelný automat kompaktní konstrukce. Základní jednotka obsahuje hlavní desku se všemi aktivními prvky, za ní je sběrnice deska s konektory pro připojení vstupně-výstupních modulů - čtyři na horním okraji a čtyři na dolním. Celek je uzavřen v ploché skřínce, jež umožňuje přístup jen ke konektoru PGU (Programming Unit) pro připojení programovacího zařízení a svorkovnicím jednotlivých modulů na stranách

přístroje. Světlovody přivádějí od LED diod na plošných spojích na povrch optickou indikaci stavu základní jednotky a jednotlivých vstupů a výstupů.

Po sejmutí krytu se zpřístupní svorkovnice pro napájení přístroje 24V $\pm$, komunikační rozhraní a kontakty relé watchdogu. Nyní lze na kontakty sběrnice desky zasunout I/O moduly. Na hlavní desce jsou také dvě pozice k instalaci rozšiřujících piggybacků s komunikačními a zobrazovacími obvody a patice pro rozšíření paměti (RAM i EPROM). Při vypnutí napájení jsou paměti zálohovány knoflíkovým článkem. Viz obr. 14 v příloze A.

Datová komunikace mezi hlavní deskou a přídatnými moduly probíhá pomocí adresovaných kanálů. Každý I/O modul má vyhrazeno 16 adres, podle pozice, na níž je instalován. Adresace začíná v levém horním rohu sběrnice desky od báze adresy 0 a pokračuje ve směru hodinových ručiček. Pro každý modul je báze adresy zvyšována vždy o 16. Toto pravidlo platí vždy, nezáleží na tom, zda modul využívá 16, 8, 6 nebo jen 4 adresy. V případě potřeby připojení více než osmi modulů lze použít rozšiřující jednotku, která obsahuje sběrnici s dalšími osmi pozicemi s báze adresami 128 - 240.

Pozn.: Protože obvodu watchdog na hlavní desce je interně přiřazena adresa 255, nesmí být na adrese 240 instalován modul, který používá všech 16 adres.

1.3.2 Dodávané typy vstupně-výstupních modulů

- komunikační rozhraní (RS 422, RS 485, RS 232, proudová smyčka 20mA) (jako piggyback)
- Hodiny reálného času + displej (jako piggyback)
- Dvuhodnotové vstupní a výstupní moduly - 4 až 8 vstupů nebo výstupů, 24V $\pm$ nebo 250 V $\sim$, proud až 2A. Galvanicky neoddělené i optočleny oddělené vstupy, výstupy s tranzistory MOSFET nebo relé.
- Analogové vstupní a výstupní moduly s 4 až 8 kanály s rozlišením A/D, D/A převodníků
8 až 12 bitů pro signály 0 - 10V, -10 - 0V, -10 - 10V, 0 - 20mA, -20 - 0mA, -20 - 20mA, 4 - 20mA a pro přímé připojení teploměrů Pt, Ni 100, 1000
- Čítači a polohovací moduly
- Zobrazovací moduly a terminály

1.3.3 Typy datových prostředků využitelných v programech

Data a hodnoty, které program používá jsou striktně rozděleny do datových typů. Do paměti nelze přistupovat libovolně, ale vždy jen k prvku určitého typu a pořadového čísla (adresy) v rámci typu. Tabulka 1 obsahuje přehled použitelných typů dat.

Tab. 1. Přehled datových typů prostředků programovatelného automatu SAIA

Typ kódu	Typ	Popis	Rozsah adres
I	Vstup (Input)	Hodnota fyzického dvouhodnotového vstupu I/O modulu (adresa podle umístění modulu)	0..8191
O	Výstup (Output)	Slouží k nastavení nebo přečtení stavu fyzického dvouhodnotového výstupu I/O modulu (adresa podle umístění modulu). I a O sdílejí stejné adresy.	0..8191
F	Flag	Binární proměnná	0..8191
T	Časovač (Timer)	Proměnná k nastavení nebo čtení stavu časovače	0..450
C	Čítač (Counter)	Proměnná k nastavení nebo čtení stavu čítače. T a C sdílejí adresy, je možno si zvolit, co bude čítač a co časovač.	0..1599
R	Registr (Register)	Obecný číselný registr (32 bitů). Uložená hodnota může být celé nebo reálné číslo	0..4095
K	Konstanta (K constant)	Podle přítomnosti desetinné tečky se rozlišují celá a reálná čísla	
X	Text	Slouží k ukládání řetězců do paměti	0..7999
DB	Datový blok (Data Block)	Slouží k ukládání bloků dat do paměti. Výhodné pro větší množství dat, používají se speciální instrukce. X a DB sdílejí stejné adresy, je nutno zvolit, zda půjde o řetězec nebo data.	0..7999

Uvedený rozsah adres platí vždy jen v rámci typu (kromě výjimek sdílení adres). Např. Flag F15 je uložen jinde než registr R15. V programech lze samozřejmě využívat možnosti přiřazení symbolických označení (např. direktivou EQU v Instruction List).

1.3.4 Struktura uložení programu

Podobně jako k datům a proměnným lze přistupovat pouze systematicky pomocí adres v rámci datového typu, program se musí zapisovat formou bloků různých typů podle určení. Tabulka 2 obsahuje přehled použitelných bloků.

Tab. 2. Přehled typů programových bloků

Typ kódu	Typ bloku	Popis	Rozsah číslování
COB	Cyklický organizační blok (Cyclic Organization Block)	Části programu jsou cyklicky prováděny (postupně se vykoná COBO, COB1 až po COB 15 a znova od COBO)	0..15
XOB	Blok výjimky (Exception Organization Block)	K jednorázovému spuštění bloků tohoto typu dochází při výjimečných událostech přiřazených každému číslu XOB. Např. XOBO se provede při výpadku napájení, XOB13 při výskytu chyby, XOB 16 při studeném startu PCD	0..31
PB	Programový blok (Program Block)	Blok podprogramu (procedure). Volá se z jiného bloku a po provedení se vrací zpět.	0..299
FB	Funkční blok (Function Block)	Funkce - použití stejné jako u PB, při volání lze navíc předávat parametry.	0..999

SB	Sekvenční blok (Sequential Block)	Blok pro programování sekvenčních algoritmů (Graftec), obsahuje IST, ST a TR	0..31
IST	Úvodní krok (Initial Step)	Provádí se na začátku sekvence definované SB	0..1999
ST	Krok (Step)	Blok obsahující krok - dílčí část sekvence	0..1999
TR	Přechod (Transition)	Blok obsahující podmínky, na jejichž základě se rozhoduje o přechodu na další krok	0..1999

V programu se nesmějí vyskytovat bloky stejného čísla v rámci typu .

[1]

2 VÝVOJOVÉ PROSTŘEDÍ PG4

K tvorbě programů pro rodinu programovatelných logických automatů SAIA PCDx výrobce dodává vývojové prostředí PG4 běžící na PC s operačním systémem MS Windows 3.x, 95, NT. Jedná se o ucelený balík nástrojů pro tvorbu jednotlivých rutin v různých programovacích jazycích, jejich kompilování, správu knihovny takto vzniklých objektů, linkování programů až správu celých projektů. Pomocí PG4 lze navíc on-line připojením k PCD měnit konfiguraci logického automatu, nahrávat do jeho operační paměti vytvořené programy, tyto spouštět a za chodu monitorovat na obrazovce PC či přímo krokovat až na úrovni nejzákladnějších funkčních bloků.

Knihovna projektů

Po spuštění vývojového prostředí SAIA PG4 se objeví okno knihovny projektů (SAIA Projects Library). Toto je nejvyšší úroveň, zde se k další práci vybírá jeden ze samostatných již vytvořených projektů nebo se zakládá nový.

Po zvolení jednoho z projektů s ním lze začít pracovat ve Správci projektu (SAIA Project Manager)

Správce projektu

Okno správce projektu obsahuje seznam programových modulů v tomto projektu obsažených, z nichž se sestavuje výsledný program. V případě potřeby lze samozřejmě určitý modul ze sestavování vyjmout, aniž by se musel zcela odstranit z projektu.

Správce je jádrem PG4. Odtud lze linkovat program, připojit se on-line k PCD, nahrát do něj program a následně jej spustit, krokovat. K dispozici jsou zde také nástroje ke konfigurování paměťových zdrojů PCD - rozvržení prostoru RAM pro dynamické proměnné, čítače, časovače, uživatelské texty a datové bloky. Online Configurator umožňuje přímo nastavovat systémové prostředky logického automatu - vnitřní obvod reálného času, parametry komunikačních rozhraní, případně zablokovat další přístup k paměti PCD heslem (pro zabránění neoprávněného kopírování či modifikace software).

Možné programovací jazyky

AWL - Instruction List (assembler), editor AWL (textový editor)

FBD/LD - Function Block Diagram/Ladder Diagram (schéma funkčních bloků), editor
FUPLA (Function Plan)

SFC – Graftec - grafický jazyk pro programování sekvencí bloků

Dalšími možnostmi jsou:

OBJ - předem zkompileovaný objekt k zahrnutí do výsledného programu

RIO - Remote I/O Network

FMS - Profibus FMS Network - objekty připojené ze vzdáleného zdroje po síti

2.1.1 Instruction List (IL)

Instruction List (výpis instrukcí) je forma zápisu programu přímo v základních instrukcích. K zápisu slouží obyčejný textový editor, řádky se nečíslují, komentáře se uvozují středníkem.

Bloky se specifikují úvodními a koncovými instrukcemi podle typu bloku.

Příklad:

XOB 16 ; Studený start - začátek bloku

EXOB ; Konec bloku

K přiřazení symbolického označení slouží direktiva EQU.

Příklad: Registr R 14 se nadále může vyjadřovat symbolem „Teplota“:

Teplota EQU R 14

Aby byl definovaný symbol i se svým významem použitelný v jiných objektech (souborech), z nichž se sestavuje výsledný program, musí se zveřejnit direktivou PUBL.

Příklad:

Teplota EQU R 14

PUBL Teplota

Aby mohl být v aktuálním kódu použit symbol definovaný a zveřejněný v jiném objektu, zavede se direktivou EXTN jako externí. Příklad:

EXTN Teplota

2.1.2 Function Plan - FUPLA (FBD)

FUPLA (Function Plan) je grafický jazyk založený na makrech s výhodou dynamického přiřazování datových prostředků.

nahradí konkrétními čísly datových prostředků. Opět se čerpá z oblastí rezervovaných pro dynamické přiřazování.

Striktně se rozlišují číselné typy hodnot - binární, celočíselné a s plovoucí desetinnou čárkou - a to i barvou spojnic. Např. celočíselný výstup nelze přivést na vstup očekávající reálné číslo, musí se použít konverzní funkce.

Každý soubor v jazyce FUPLA může obsahovat více bloků. K vybrání jednoho z nich ke zobrazení v editačním okně slouží seznam bloků, vyvolaný z menu File->Block List... . Volba Properties... umožňuje nastavit typ (COB, XOB, PB ...) a číslo bloku, New... zakládá nový blok.

Každý blok může mít rozsah více stran. Listování mezi nimi, přidávání nových a jejich organizaci umožňují funkce z nabídky Page .

Jazyk FUPLA je přehledný, programování vněm je jednoduché, ale některé nedostatky přinášejí omezení:

- nelze použít skoky v programu, podmíněné ani nepodmíněné, což zcela znemožňuje realizaci cyklu
- ze stejného důvodu lze podmíněně jen volat podprogramy, i jednoduchý program se tak může stát velmi nepřehledným
- při volání funkčních bloků (FB) jim nelze předat parametry, což je degraduje na procedury (PB)

Jako vyšší jazyk přirozeně FUPLA oproti Instruction List vytváří redundantní kód, to je však samozřejmá daň za používání knihoven funkcí k univerzálnímu použití.

2.1.3 Graftec (SFC)

Graftec je grafický jazyk pro zápis sekvenčních algoritmů. Vývojový diagram určuje postup provádění dílčích kroků a podmíněných přechodů mezi nimi. Procedury kroků (ST) i vyhodnocování přechodů (TR) jsou napsány buď v jazyce IL nebo FUPLA.

Po provedení kroku program stojí, dokud není splněna podmínka ukončení přechodu na následující krok (např. doběhnutí časovače nebo změna stavu vstupu). Z tohoto důvodu je v případě realizace číslicových regulátorů s potřebou současného provádění více procesů Graftec nevhodný. [1]

3 POPIS MODELU KOLEJIŠTĚ A JEHO PROPOJENÍ S PLC

Kolejiště je modelem třídy H0. Plocha modelu je přibližně 2 m². Model znázorňuje situaci se dvěma nádražími. Hlavní nádraží, označované N1 disponuje třemi kolejemi. Nádraží N2 má dvě koleje. Železniční trať je situována do členitého terénu s kopci, tunelem a mostem. Model je osazen standardními prvky, kterými disponují i skutečné železniční tratě. Jmenovitě se jedná o výhybky, semaforey, výstražná světla na železničních přejezdech, osvětlení budov a prostoru nádraží.

3.1 FUNKČNÍ USPOŘÁDÁNÍ

Kolejiště je rozděleno do deseti samostatně napájených úseků. Toto rozdělení umožňuje individuální regulaci rychlosti a směru jízdy. Pro monitorování pohybu vlaku jsou po trati rozmístěna čidla. Fyzicky se jedná o jazýčková relé a vláčky, které jsou opatřeny magnety při přejetí čidla vyvolají sepnutí kontaktů těchto relé. Vzniklý impuls je dále zpracován.

3.2 OVLÁDACÍ SKŘÍŇ

Elektronické ovládání prvků kolejiště je integrováno do společné ovládací skříně modulárního uspořádání, kdy jednotlivým prvkům odpovídají samostatné panely. Skříň je osazena několika kontrolními přepínači pro rychlý test správného propojení s modelem. Ovládací skříň poskytuje rozhraní pro připojení řídicího počítače. Jedná se o dva 48 kolíkové konektory, které jsou mezičlánkem pro binární vstup a výstup. Výstupem (z pohledu skříně) jsou impulsy čidel. Kdy každému z čidel odpovídá jeden kolík konektoru. Podobně jsou implementovány i vstupy. Každému ovládacímu prvku kolejiště odpovídá jeden z kolíků vstupního konektoru. Vstupním logickým signálem ovládací skříně je 5V TTL. Bližší popis kolejiště, funkce a zapojení ovládací skříně viz [2, 3].

3.3 PROPOJENÍ ŘÍDICÍHO PLC S OVLÁDACÍ SKŘÍŇÍ MODELU

Podle požadavků modelu byl PLC SAIA osazen vstupními a výstupními moduly.

Jednotlivé moduly jsou propojeny s ovládací skříní dvěma svazky kabelů zakončených standardním konektorem FRB, viz [3].

Požadavek na vybavení I/O modely:

24 vstupů	-	čidla (informace o pozici vlaku)
47 výstupů	-	signalizační a řídicí prvky (např.: semaforey, výhybka, apod.)

Použité moduly:

PCD2.E110	-	vstupní modul (8 vstupů)	1x
PCD2.E165	-	vstupní modul (16 vstupů)	1x
PCD2.A465	-	výstupní modul (16 výstupů)	3x

3.4 OVLÁDÁNÍ PRVKŮ MODELU KOLEJIŠTĚ

Pro ovládání výhybek, výstražných světel železničních přejezdů, závor, semaforů a osvětlení postačuje dvouhodnotová logika. Tyto prvky se spínají přímým zápisem logické hodnoty na adresu výstupního modulu. Pro větší přehlednost funkčních schémat knihovna nabízí seznam symbolů a konstant. Adresování lze tedy provádět na „jmenné adresy“.

3.4.1 Význam logických hodnot

Osvětlení:	log. 0 = osvětlení vypnuto;	log. 1 = osvětlení zapnuto
Semaforey:	log. 0 = červené světlo;	log. 1 = zelené světlo
Přejezdy:	log. 0 = výstražná funkce vypnuta;	log. 1 = výstražná funkce zapnuta
Závory:	log. 0 = závory vytaženy	log. 1 = závory spuštěny
Jednoduché výhybky:	log. 0 = vyhybka je přepnuta pro přímý směr; log. 1 = přepnuto na zabočení	

Křížová výhybka:

Tab. 3 Možné kombinace nastavení křížové výhybky

Požadovaný směr	K1 – C,D	K1 – A,B
BD	0	0
AD	0	1
BC	1	0
AC	1	1

Příklad:

Vyžadujeme-li nastavení výhybky V1 pro přímý směr provedeme zápis log. 0 na adresu O63. Vyžadujeme-li, aby křížová výhybka K1 byla uzpůsobena pro propojení části A a D provedeme zápis log. 0 na adresu O57 a log. 1 na adresu O56. V případě využití předdefinovaných symbolů knihovny může nastavení vypadat asi takto: ROVNE na adresu V1

3.4.2 Seznam adres prvků

Tab. 4 Semaforey

označení	S1	S2	S3	S4	S5	S6	S11	S12	S13	S14	S15	S16
adresa	O55	O54	O53	O52	O51	O50	O49	O48	O47	O46	O45	O44

Tab. 5 Výhybky

označení	V1	V2	V3	V4	V5	V6	K1-C,D	K1-A,B
adresa	O63	O62	O61	O60	O59	O58	O57	O56

Tab. 6 Přejezdy

označení	P1	P2	P3
adresa	O36	O37	O38

Tab. 7 Ostatní prvky

označení	Závory	osvětlení N1	osvětlení N2	osvětlení OD
adresa	O39	O41	O42	O43

Tab. 8 Čidla

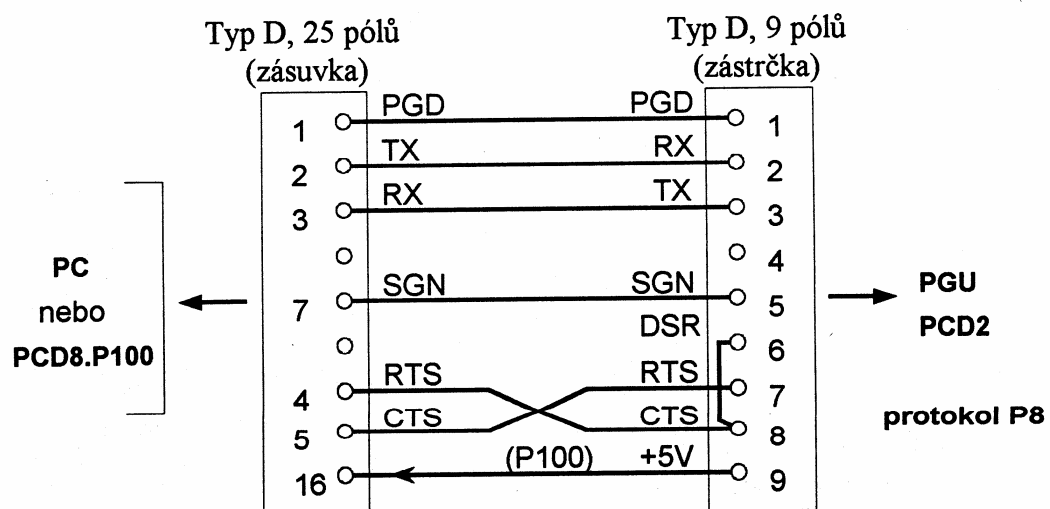
označení	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
adresa	I64	I65	I66	I67	I68	I69	I70	I71	I72	I73	I74	I75
označení	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23	C24
adresa	I76	I77	I78	I79	I0	I1	I2	I3	I4	I5	I6	I7

[4]

3.5 PROPOJENÍ ŘÍDICÍHO PLC S PC

Pro vývoj aplikací, jejich vizualizaci a případně i ovládání je třeba propojit řídicí PLC s vývojovým a vizualizačním PC. Toto propojení je realizováno sériovým kabelem. Na straně PLC musí být konektor s devíti PIN-y, na straně PC máme možností více (viz [5]). V našem případě zůstal jediný volný konektor, a to sériový port COM 1 s 25 PIN-y. Proto bylo potřeba podle obr. 2 vyrobit propojovací kabel.

Propojovací kabel PCD8.K100 (pro protokol P8)



Obr. 2. Zapojení použitého propojovacího kabelu.

4 POUŽITÉ MODULY

Pro ovládání složitějších funkcí jsem využil modulů zpracovaných v diplomové práci [4]. Jedná se o moduly pro ovládání rychlosti a výstražných světel na přejezdech, vždy ve verzích pro společné ovládání všech úseků i pro ovládání každého úseku zvlášť. Moduly je potřeba do vlastního projektu nejdříve importovat, to se provádí z okna Správce projektu (Project Manager) příkazem File – Import...

Moduly pro společné a individuální ovládání nemohou být v jednom projektu zároveň.

4.1 MODULY OVLÁDÁNÍ ŽELEZNIČNÍCH PŘEJEZDŮ

Úkolem těchto modulů je zapouzdření obsluhy sepnutí/vypnutí výstražné funkce světel železničních přejezdů v okamžiku, kdy se blíží vlak. Stejně tak o jejich vypnutí v případě kdy vlak přejezd opustil. Jedná se o obsluhu událostí čidel které přiléhají k železničnímu přejezdu. V případě signalizace čidla v závislosti na směru jízdy dojde k sepnutí výstražné funkce a podržení tohoto stavu až do okamžiku signalizace čidlem umístěným za železničním přejezdem.

O činnost těchto modulů se programátor nemusí dále starat.

Modul pro společné ovládání má název „prejezdy.fup“, při individuálním ovládání použijeme moduly „prejezd1.fup“, „prejezd2.fup“ a „prejezd3.fup“.

4.2 MODULY OVLÁDÁNÍ RYCHLOSTI A SMĚRU JÍZDY VLAKU

Tyto moduly slouží k regulaci rychlosti a směru jízdy vlaku. Při ovládání směru dochází k transformaci parametru na výstup PLC, kdy log. 1 udává směr jízdy zpět (proti směru otáčení hodinových ručiček) a log. 0 jízdu vpřed. Regulace rychlosti je softwarovou implementací jednoduché šířkové modulace. V případě log. 1 na výstupu PLC je úsek kolejiště napájen, log. 0 vypíná napájení. Program PLC je vykonáván cyklicky s vysokou periodou (v závislosti na množství instrukčním kódů a komunikačních zařízeních). Za impuls pro regulaci rychlosti je uvažováno 5 průběhů programové smyčky. Parametr rychlosti (celočíslná hodnota) udává při kolika průbězích programové smyčky bude na výstupu log.1 (ve zbývajících průbězích bude výstup log. 0).

Modul společného ovládání má název „Useky.fup“, vstupními parametry jsou „Smer“ (směr jízdy vlaku) a „Rychlost“ (rychlost jízdy).

Moduly individuálního ovládání mají názvy i vstupní parametry obdobné, např. „Usek1.fup“, „Smer_U1“ a „Rychl_U1“.

Pro jízdu vlaku tedy stačí jen nastavit rychlost a směr jízdy.

Rychlost jízdy je celočíselná hodnota v rozmezí $\langle 0,5 \rangle$. Při nulové rychlosti vlak stojí, při jízdě tedy máme k dispozici 5 stupňů rychlosti. Impulzní řízení rychlosti by se mohlo negativně projevit na životnosti technického zařízení, proto pro běžnou jízdu doporučuji maximální rychlost.

II. PRAKTICKÁ ČÁST

5 VZOROVÉ PROGRAMY

Programy jsem tvořil výhradně v jazyce FBD/LD (Function Block Diagram/Ladder Diagram - schéma funkčních bloků), používal jsem proto editor FUPLA (Function Plan).

5.1 PŘESUN VLAKU V RÁMCI NÁDRAŽÍ

Tato úloha je velice jednoduchá, proto je vhodná pro seznámení se s vybavením a jeho ovládáním.

Zadání

Na začátku je vlak v nádraží N1 na koleji U5. Úkolem je přemístit jej na kolej U6.

Návod na vypracování

Na začátku programu je třeba nastavit výhybky a směr jízdy, k tomu slouží blok XOB16 (nastavení ve File - Block Properties – Type, Block number). Zbytek programu je standardně v bloku COB0, ten není třeba nastavovat. Pak už můžeme rozjet vlak - nastavíme rychlost (např. střední). Po přejetí čidla U5 změníme směr jízdy a nastavíme výhybky tak, aby vlak přijel na kolej U6.

Jedna z možných realizací této úlohy je v příloze [1].

5.2 JÍZDA DVOU VLAKŮ PO OKRUHU SE SPOLEČNOU KOLEJÍ

Tato úloha podstatně složitější, pro výukové účely doporučuji ovládat jen jeden z vlaků.

Zadání

Na začátku jsou oba vlaky v nádraží N1, vlak 1 na koleji U5 a vlak 2 na koleji U6. Vlak1 vyjíždí po vnějším okruhu (koleje U2, U1,U9) na nádraží N2, po několika vteřinách vyjede vlak2 po vnitřním okruhu (U2, U10), také na nádraží N2. Oba vlaky zde několik sekund počkají a pak vyjíždí na společnou kolej (U8,U3,U1,U7), poté se opět rozdělí na U5, resp.

U6, na několik sekund zde zastaví a cyklus se opakuje. Je třeba ošetřit, aby se vlaky nesrazily (a to jak jízdou na společné koleji těsně za sebou, tak vlak 1 při projíždění křížové výhybky s vlakem 2 při projíždění společného okruhu).

Návod na vypracování

Je třeba ošetřit, aby se vlaky nesrazily (a to jak jízdou na společné koleji těsně za sebou, tak vlak 1 při projíždění křížové výhybky s vlakem 2 při projíždění společného okruhu). Toho docílíme použitím pomocných proměnných, např. `k_volna` (spol. kolej volna) a `v_volna` (křížová výhybka volná).

Další problém je s přepnutím výhybky V5; Zvolil jsem pomocnou proměnnou, která určuje počet vlaků na spol. koleji. Pokud na ni vlak vyjíždí jako první, přepne si výhybku pro svůj směr, jinak ji nechá v původní poloze. V každém případě se po vyjetí ze společného úseku přepne výhybka na opačnou stranu.

Jedna z možných realizací této úlohy je v příloze [2].

ZÁVĚR

Úkolem této práce bylo zejména připravit model kolejiště pro výuku programování PLC.

Toho bylo dosaženo jednak realizací propojení programovatelného automatu SAIA PCD2 s kolejištěm a počítačem, dále vytvořením nástěnného obrazu se zvýrazněnými aktivními prvky kolejiště doplněným adresami, na které jsou jednotlivé prvky připojeny k řídicímu automatu. Posledně zmiňovaný fakt je důležitý pro rychlé a pohodlné programování, protože dříve bylo nutné najít v obrázku (popř. přímo na kolejišti) označení prvku a poté v tabulce jeho adresu. Dále měla být provedena vizualizace v programovém prostředí Control Web 2000, zde se nepodařilo oživit propojení vstupů s programovatelným automatem.

.....

SEZNAM POUŽITÉ LITERATURY

- [1] Jan Horných - Aplikace PLC automatů pro řízení enzymatické hydrolýzy koželužských odpadů, Diplomová práce, rok 1999.
- [2] Vladimír Hnilica – Počítačové řízení vlakového kolejiště, Diplomová práce, rok 2002.
- [3] Vladimír Hnilica – Učební pomůcka pro výuku – vlakové kolejiště, Bakalářská diplomová práce, rok 1999.
- [4] Viliam Dolinay – Modulární programové vybavení pro řízení kolejiště, Diplomová práce, rok 2003
- [5] SAIA - Burgess Electronics Ltd. - Manuál pro řady PCD1 a PCD2, rok 1997
- [6] Martinásková, M., Šmejkal, L.: Řízení programovatelnými automaty, Vydavatelství ČVUT, Praha, 1998..
- [7] Martinásková, M., Šmejkal, L.: Řízení programovatelnými automaty II, Vydavatelství ČVUT, Praha, 2000.
- [8] Šmejkal, L., Martinásková, M.: PLC a automatizace, Nakladatelství BEN - technická literatura, Praha, 1999.

SEZNAM OBRÁZKŮ

Obr. 1. Okno editoru Fupla..... **Chyba! Záložka není definována.**

Obr. 2. Zapojení použitého propojovacího kabelu. **Chyba! Záložka není definována.**

SEZNAM TABULEK

Tab. 1. Přehled datových typů prostředků programovatelného automatu SAIA **Chyba! Záložka není d**

Tab. 2. Přehled typů programových bloků **Chyba! Záložka není definována.**9

Tab. 3. Možné kombinace nastavení křížové výhybky .. **Chyba! Záložka není definována.**

Tab. 4. Semaforey..... **Chyba! Záložka není definována.**

Tab. 5. Výhybky **Chyba! Záložka není definována.**

Tab. 6. Přejezdy **Chyba! Záložka není definována.**

Tab. 7. Ostatní prvky **Chyba! Záložka není definována.**

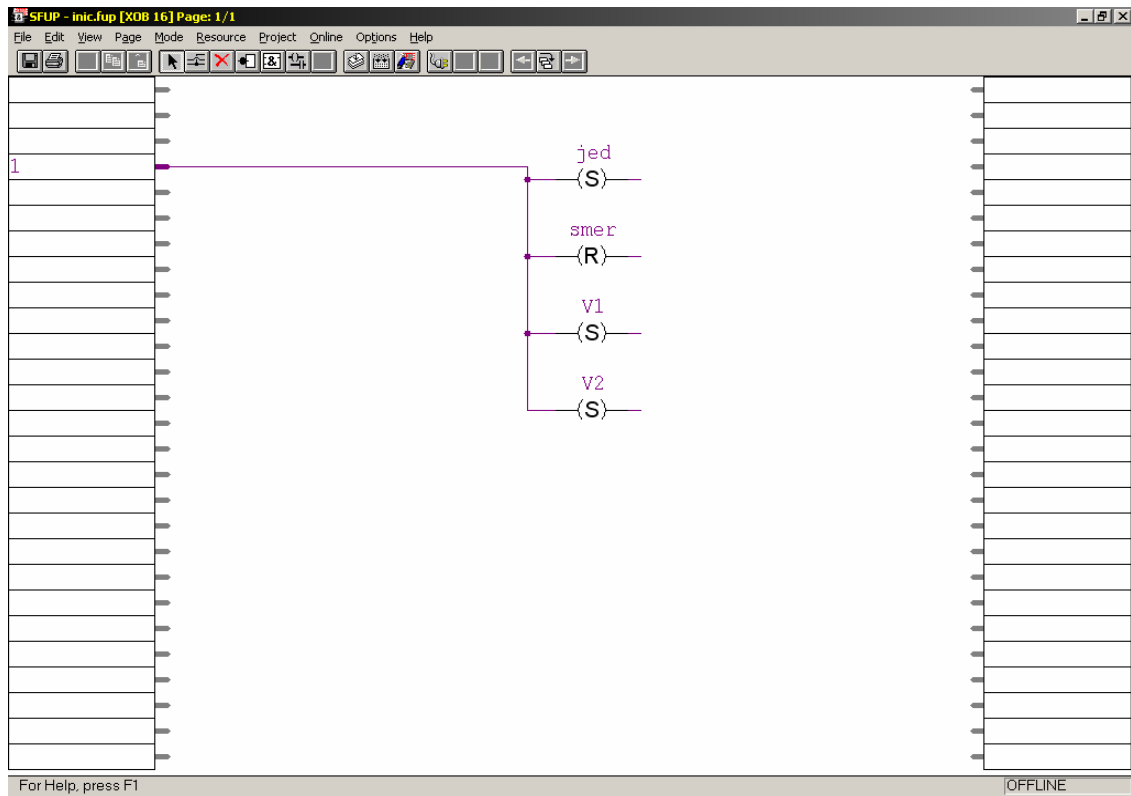
Tab. 8. Čidla..... **Chyba! Záložka není definována.**

SEZNAM PŘÍLOH

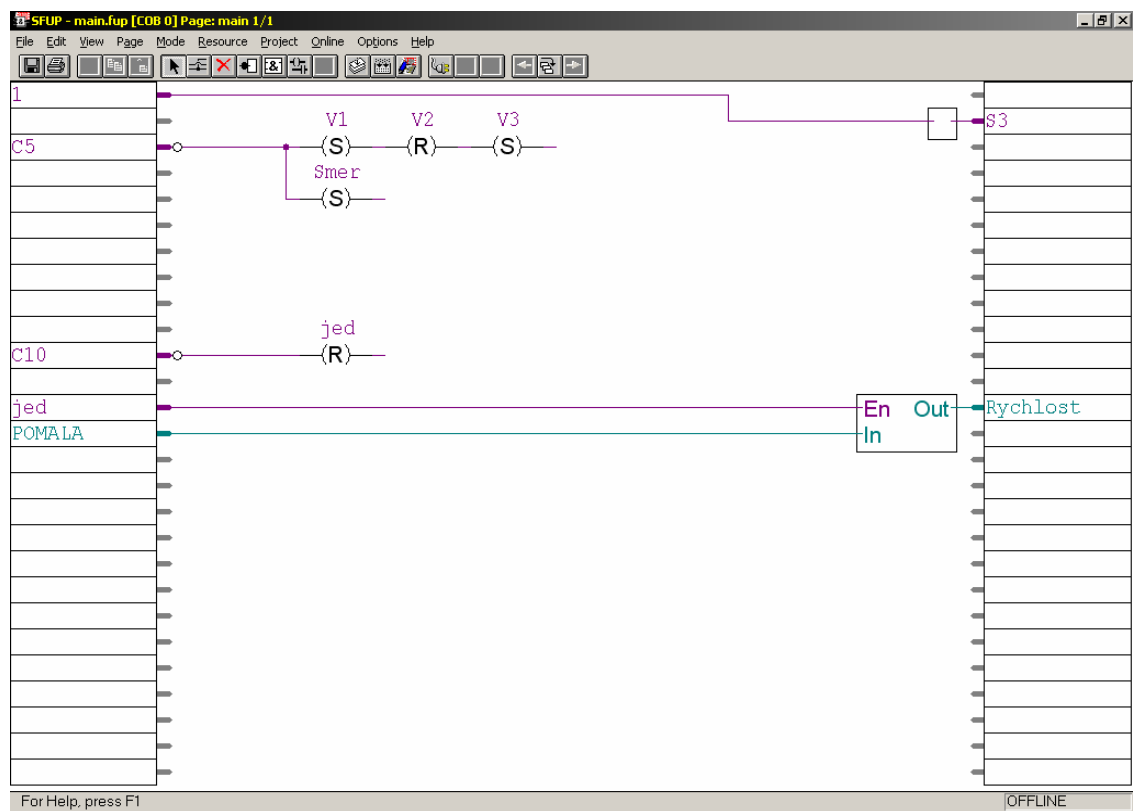
- P I Přesun vlaku v rámci nádraží.
- P II Jízda dvou vlaků po okruhu se společnou kolejí.

PŘÍLOHA CHYBA! NENALEZEN ZDROJ ODKAZŮ.: PŘESUN VLAKU V RÁMCI NÁDRAŽÍ

Jedna z možných realizací



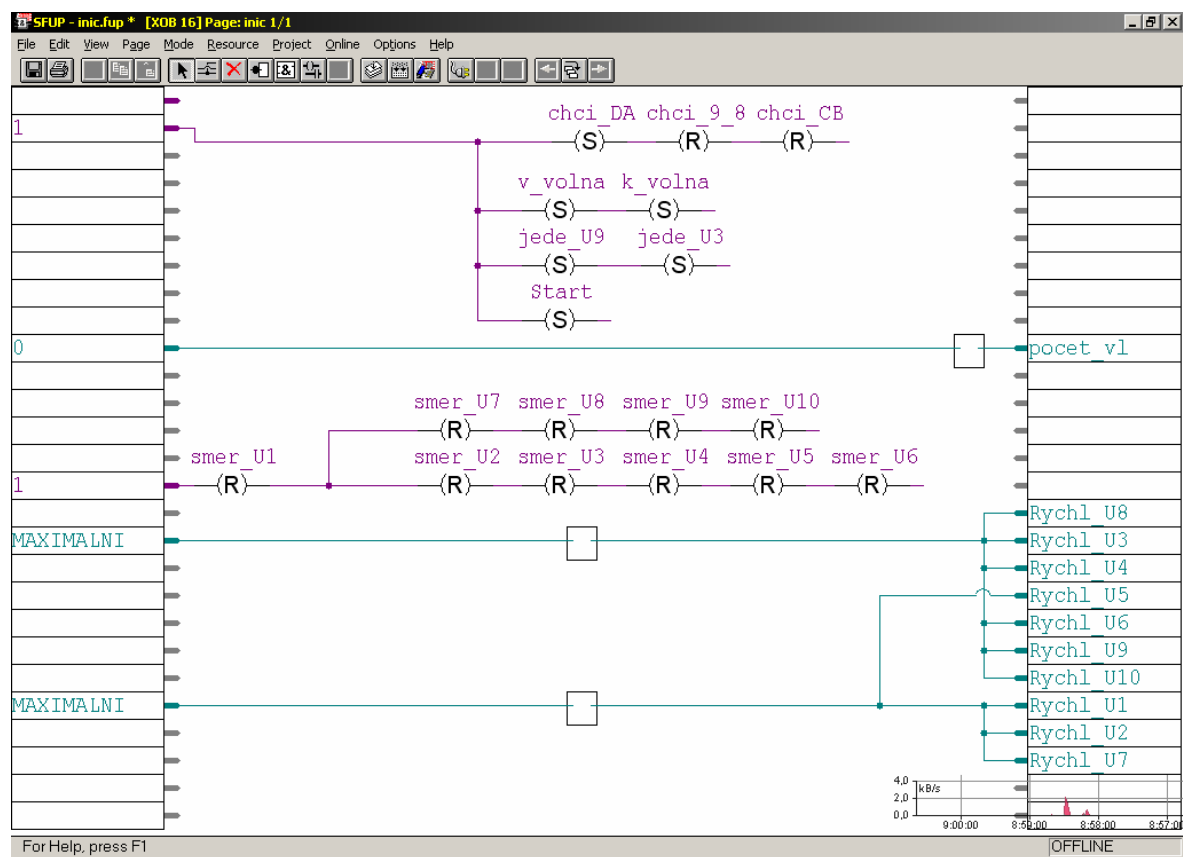
Obr. PI – 1. Blok XOB 16 - inicializace



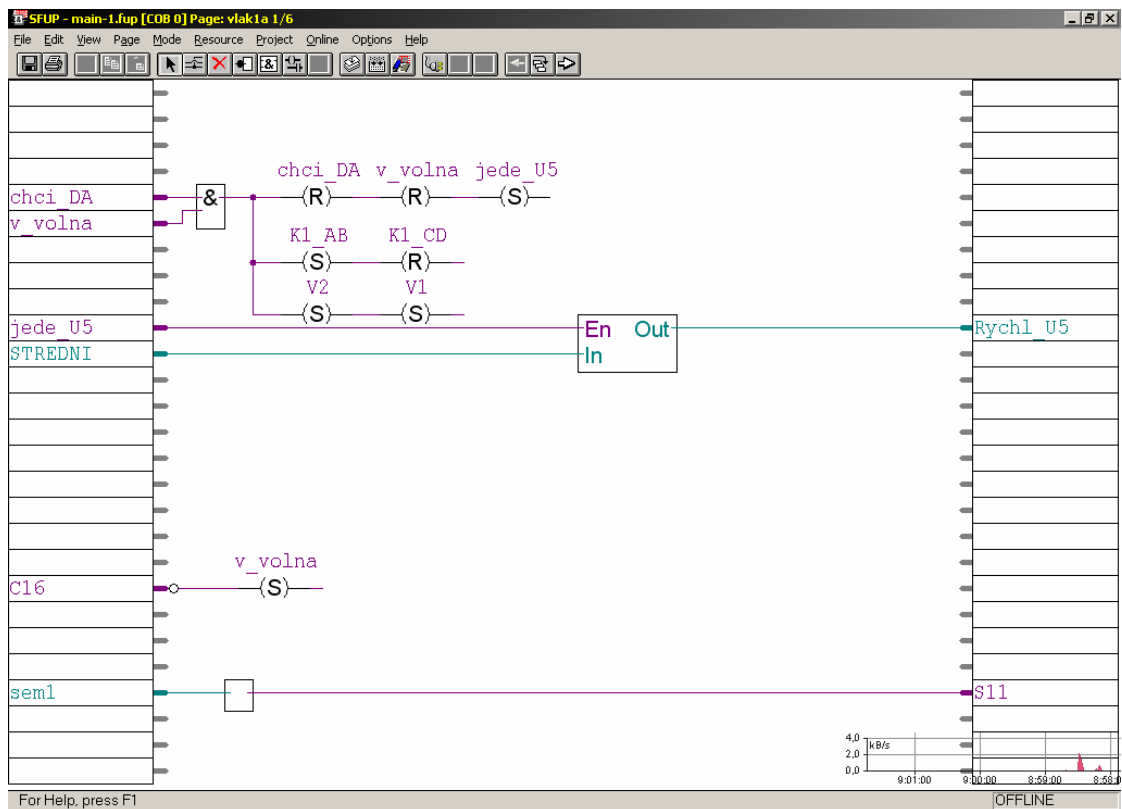
Obr. PI – 2. Blok COB 0 – vlastní program

PŘÍLOHA CHYBA! NENALEZEN ZDROJ ODKAZŮ.: JÍZDA DVOU VLAKŮ PO OKRUHU SE SPOLEČNOU KOLEJÍ

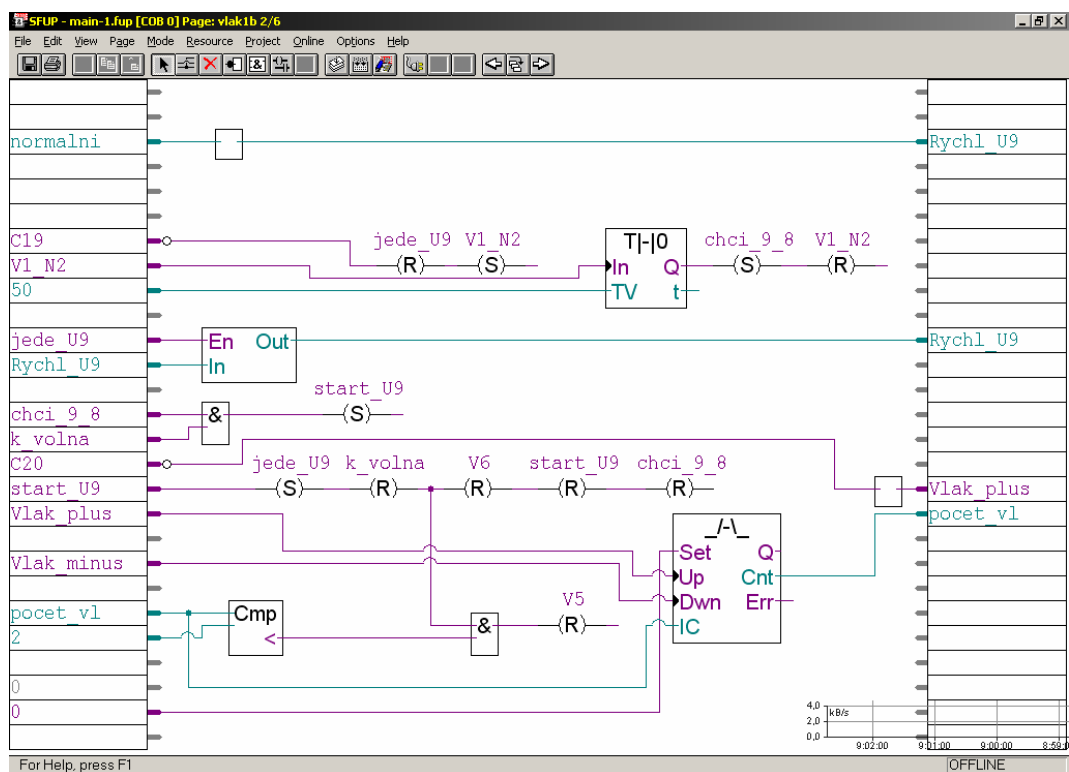
Jedna z možných realizací



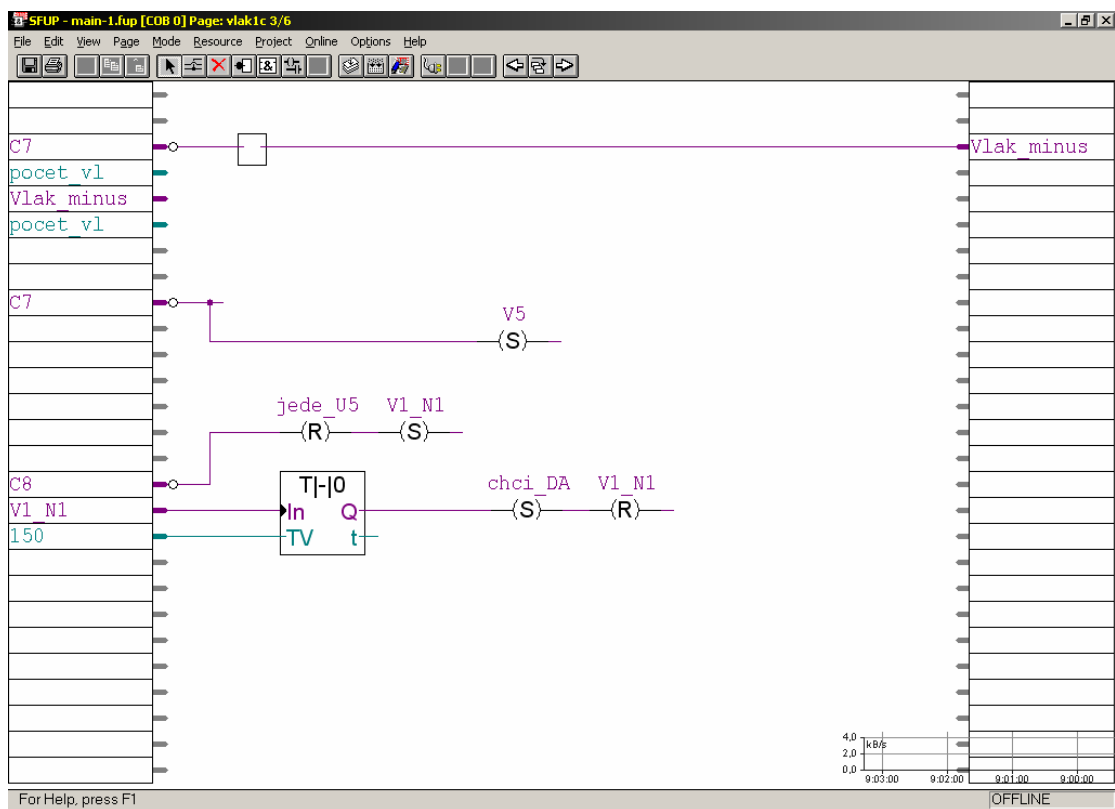
Obr. PII – 1. Blok XOB 16 - inicializace



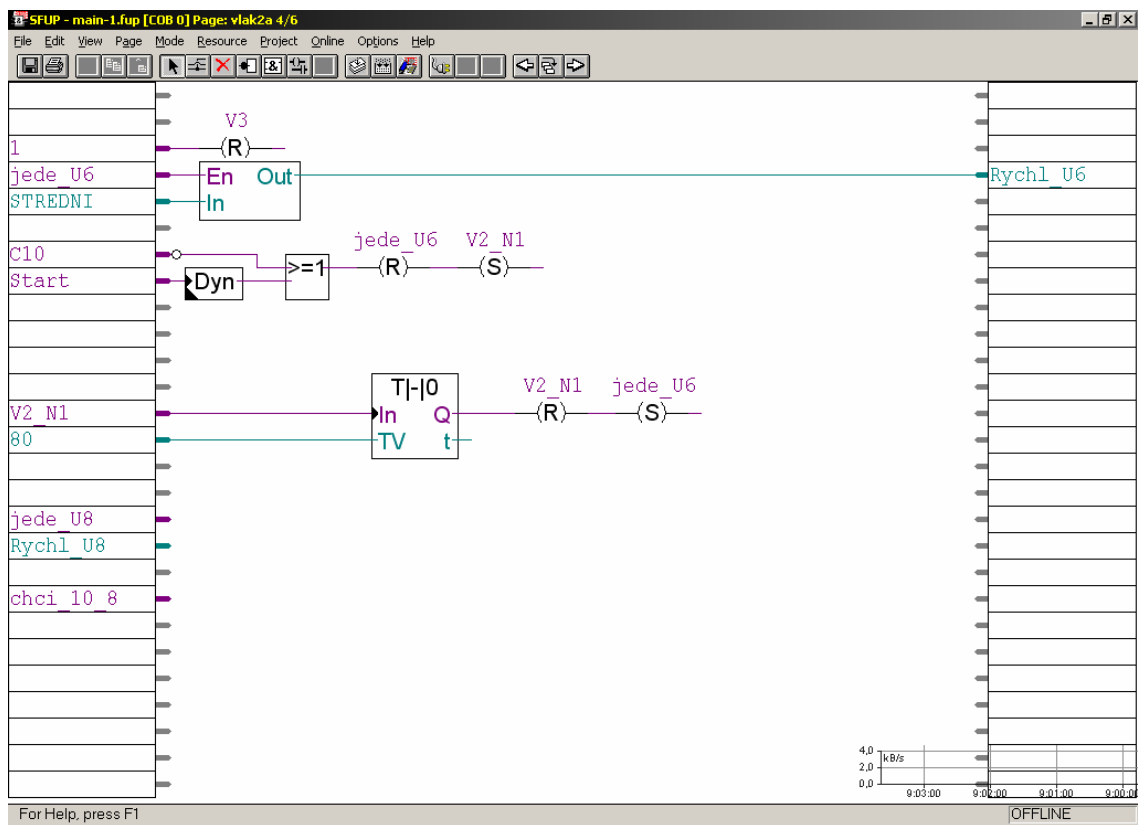
Obr. PII – 2. Blok COB 0 – vlastní program, 1. část trasy vlaku 1



Obr. PII – 2. Blok COB 0 – vlastní program, 2. část trasy vlaku 1



Obr. PII – 2. Blok COB 0 – vlastní program, poslední část trasy vlaku 1



Obr. PII – 2. Blok COB 0 – vlastní program, 1. část trasy vlaku 2

