

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA STROJNÍ



BAKALÁŘSKÁ PRÁCE

**Moderní nástroje analýzy a syntézy při návrhu řízení
pomocí PLC**

Praha, 2011

Autor: Petrásek Zbyšek

Prohlášení

Prohlašuji, že jsem předloženou bakalářskou diplomovou prací zpracoval samostatně a souhlasím s tím, že její výsledky mohou být dále využity dle uvážení vedoucího této práce Ing. Marie Martináskové, PhD. jako jejího spoluautora. V případě publikace si přeji být uveden jako spoluautor.

Datum.....

Podpis.....

Poděkování

Chtěl bych tímto poděkovat všem, kteří mi při vytváření této práce pomáhali. Především děkuji svému vedoucímu bakalářské práce, Ing. Marii Martináskové, PhD. za věnovaný čas a připomínky, dále spolupracovníkům z laboratoře č. 109 za pomoc při realizaci této práce a Ing. Aleši Kocinovi z firmy Festo za technickou podporu.

Abstrakt

Bakalářská diplomová práce obsahuje popis prostředků pro analýzu řízení zadané výukové úlohy z potravinářského průmyslu. Práce dále obsahuje popis realizace řešení zadané úlohy prostřednictvím HW CPX-CEC a SW CoDeSys firmy Festo. Tento popis slouží jako návod pro další budoucí uživatele, kteří budou realizovat různé druhy úloh pomocí tohoto HW a SW.

Abstract

Bachelor thesis contains description of instruments for analysis of controlling the assigned training task from food-processing industry. Thesis furthermore consists of description of the realization of the assigned task through the Festo hardware CPX-CEC and software CoDeSys. This description serves as a guide for the next users who will implement various tasks with this hardware and software.

Obsah

1	ÚVOD DO PROBLEMATIKY ŘÍZENÍ POMOCÍ PLC	1
1.1	PLC automaty	1
1.2	Petriho sítě, distribuované systémy	1
1.3	Grafcet.....	4
2	SOFTWARE PRO APLIKACI PETRIHO SÍTÍ.....	7
3	ANALÝZA ŘÍZENÍ TECHNOLOGICKÉHO PŘÍPRAVKU	14
3.1	Zadání.....	14
3.2	Výběr software	16
3.3	Vytvoření algoritmu ovládacího automatu.....	16
4	ŘÍZENÍ PŘÍPRAVKU V PROSTŘEDÍ CODESYS	26
4.1	Vytvoření programu v SW CoDeSys	26
4.2.1	Řídicí program.....	26
4.2.2	Vizualizace	37
4.2.3	Rozšíření funkcí operátorského panelu	43
4.3	HW - řídicí jednotka CPX-CEC a SW - CoDeSys	44
4.3.1	Charakteristika jednotky CPX-CEC.....	44
4.3.2	Technická realizace řízení	46
5	ZÁVĚR.....	52
	SEZNAM ZDROJŮ.....	53
	SEZNAM OBRÁZKŮ.....	55
	SEZNAM TABULEK	57

Seznam použitých zkratek v angličtině

PLC	Programmable Logic Controller
CPU	Central Processing Unit
SW	Software
HW	Hardware
C/E	Condition/Event
P/T	Place/Transition
IEC	International Electrotechnical Commission
SFC	Sequential Function Chart
TON	Time On Delay
LD	Ladder Diagram
FBD	Function Block Diagram
IL	Instruction List
SIPN	Signal Interpreted Petri Nets
POU	Program Organization Unit
IP	Internet Protocol

1 Úvod do problematiky řízení pomocí PLC

1.1 PLC automaty

Pro binární logiku řízení se zpočátku realizovaly řídicí obvody pomocí reléových automatů. S rostoucími nároky však postupem času začala být znáta omezenost této realizace a zejména pak rozvoj polovodičových součástek uspořádal příchod nových kompaktních celků pro řízení technických aplikací, PLC automatů.

Programovatelný logický automat (*Programmable Logic Controller*) je počítač, používaný pro automatizaci procesů v reálném čase, zpracovávající jak analogové, tak digitální vstupy a výstupy. Oproti běžným počítačům program vykonává cyklicky a jeho periferie jsou přímo uzpůsobeny pro napojení na technologické procesy.

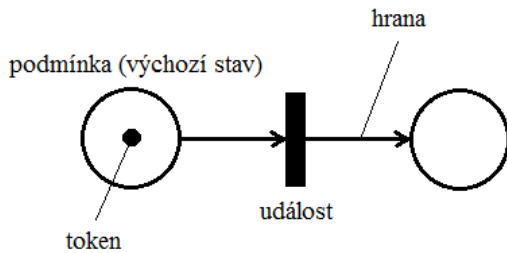
Kromě značného pohodlí, které nabízí programátorovi dnešní softwarové prostředí, je i možné jednoduše propojovat automaty s dalšími moduly, např. pro polohování a komunikaci. V současné době už také ani není výjimkou, že CPU automatu obsahuje WWW server a lze tak s automatem komunikovat na dálku. Sortiment a cenový rozsah PLC systémů je dnes skutečně rozsáhlý (menší automat je otázkou řádově tisíců Kč) a i pro množství svých značných výhod je tedy dnes výhodným řešením pro realizaci jak jednodušších logických úloh, tak i pro velmi komplexní aplikace v nejrůznějších odvětvích průmyslu.

1.2 Petriho sítě, distribuované systémy

Distribuovaný systém je systém rozdělený na určitý počet menších jednodušších celků za účelem zjednodušení a urychlení celého procesu nebo výpočtu. Podmínkou pro vytvoření distribuovaného systému je pochopitelně možnost s jeho jednotlivými částmi pracovat paralelně = vykonávat více úloh v jeden okamžik. Pro znázornění a modelování distribuovaných (diskrétních) systémů se dnes hojně používá tzv. Petriho síť.

Petriho síť se poprvé objevila v disertační práci *Kommunikation mit Automaten* německého matematika a počítačového vědce Carla Adama Petriho roku 1962. Petri v ní graficky reprezentoval paralelismus distribuovaného systému diagramem s prvky znázorňujícími události (angl. *events*) a podmínky (*conditions*), které jsou propojeny

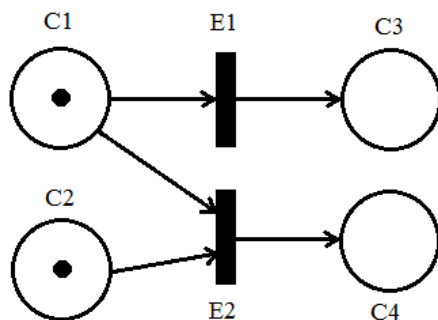
orientovanými hranami (*arcs*). Počáteční stav a jednotlivé kroky algoritmu jsou pak znázorněny značkami v uzlech (*tokens*).



Obr. 1.2.1 Prvky Petriho sítě

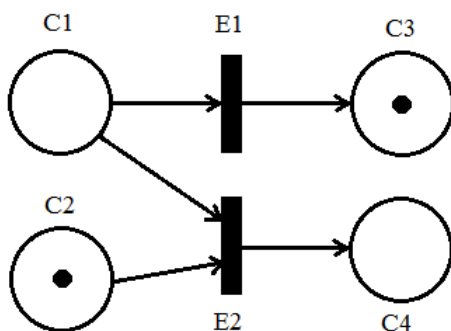
Petriho sítě od dob svého vzniku prošly svým vývojem a především změnami, které umožňují ještě více zpřehlednit názornost algoritmu pro ten daný případ, pro které jsou vypracovány. Dle [4] je tak lze rozdělit na několik základních typů:

- a) *C/E síť* – původní model použitý Petrim; vyjadřuje průběh událost (*event*) a podmínky (*condition*), za které událost proběhne; token vyjadřuje pravdivost podmínky ([4])



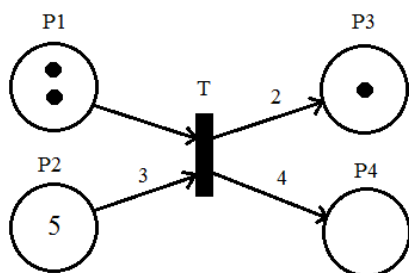
Obr. 1.2.2 C/E síť před přechodem

- událost E1 proběhne, jsou-li splněny všechny její podmínky (C1)
- pro události E1 a E2 jsou splněny všechny podmínky, nastat však může pouze jedna



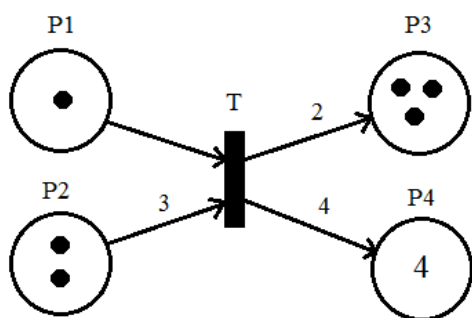
Obr. 1.2.3 C/E síť po přechodu

- b) *P/T síť* – rozšíření původního modelu; vyjadřuje průběh místo/stav systému (*place*) – přechod (*transition*); místa mají max. kapacitu tokenů, které se mohou v daném stavu nacházet; váha hrany určuje počet tokenů, které se po hraně přesouvají



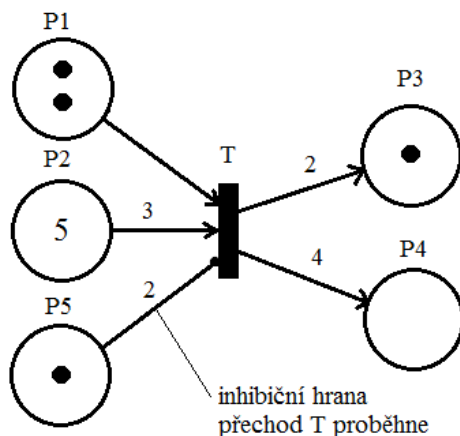
Obr. 1.2.4 P/T síť před přechodem

- přechod T proběhne, je-li na výchozích místech (P1, P2) minimálně tolik tokenů, kolik jsou váhy hran vstupujících do přechodu; na výstupních místech se objeví o tolik tokenů více, kolik jsou váhy hran z přechodu vystupujících



Obr. 1.2.5 P/T síť po přechodu

- C/E síť je druh P/T síť s nekonečnou kapacitou míst a vahami hran 1 (nezakresluje se) => každou P/T síť lze převést na síť C/E
- *inhibiční hrana* – hrana vedoucí pouze z místa do přechodu; přechod proběhne jen tehdy, proběhnul-li by bez přítomnosti inhibiční hrany, a zároveň, je-li v místě, z kterého inhibiční hrana vede, méně tokenů, než kolik má inhibiční hrana váhu



Obr. 1.2.6 Inhibiční hrana

- *P/T síť s prioritami* – každému přechodu je přiřazeno celé nezáporné číslo, udávající jeho prioritu; přechod s prioritou proběhne, proběhl-li by bez přiřazené priority, a zároveň žádný jiný povolený přechod nemá vyšší prioritu

c) *Časované Petriho síť* – zahrnují do procesu i časový faktor

- trvání dějů: deterministické (DPN), stochastické (SPN), kombinované (GSPN)
- zavedení časů: přechodu, místa, hrany, tokenů

d) *Barevné Petriho sítě* – P/T síť, která umožňuje pracovat s více typy tokenů (více barev tokenů), každému místu je pak přiřazen typ tokenů (třída), které se zde mohou nacházet; umožňuje přiřadit podmínky přechodu (*guard*), jež po vyhodnocení dává pravdivostní hodnotu 0, 1; hranám se přiřazuje hranový výraz, jenž po vyhodnocení dává tzv. multimnožinu tokenů, která umožňuje výskyt stejného prvku vícekrát

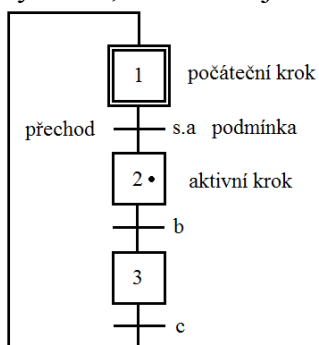
- přechod proběhne, je-li vstupní multimnožina alespoň stejně velká, jako multimnožina vypočtená po vyhodnocení příslušného hranového výrazu

Pro systém řízení pomocí PLC jsou nejčastěji používány především P/T sítě, jejichž systém „stav-přechod systému za určitých podmínek“ je snadno implementovatelný do dalších nástrojů, jako jsou např. stavové diagramy či grafcet.

1.3 Grafcet

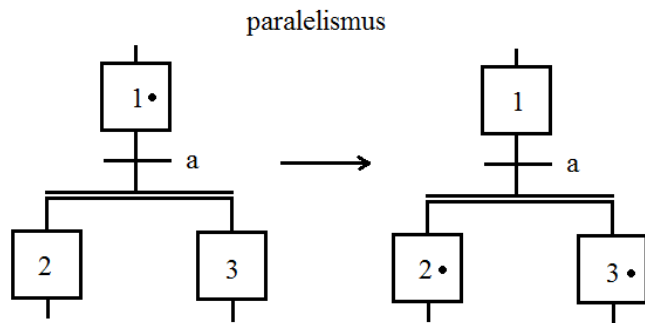
Grafcet (z francouzského *Grphe Fonctionnel de Connexion Etapes Transitions*) je návrhový grafický nástroj pro matematický popis automatu. V letech 1975-1977 byl navržen francouzskou organizací AFCET zabývající se výzkumem a aplikacemi v oblastech automatického řízení a informatiky a v roce 1987 se stal mezinárodním standardem ([5]). Vychází z Petriho P/T sítí a pro jednoznačnou interpretaci vstupů a výstupů je vhodným prostředkem pro návrh algoritmů PLC automatů.

Obdobně jako Petriho sítě využívá Grafcet hranami propojené kroky a přechody, přičemž aktuální stav je zde reprezentován aktivitou kroku či kroků. Každý přechod má svojí podmínku uskutečnitelnosti přechodu. Její platnost bývá často vyjádřena logickým výrazem, ovšem lze ji i interpretovat slovně či graficky.

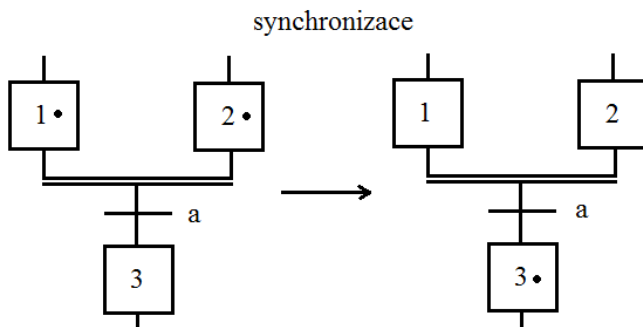


Obr. 1.3.1 Prvky Grafcet

Pro znázornění souběžných operací byl zaveden výraz *paralelismus* (divergentní AND) a *synchronizace* (konvergentní AND):

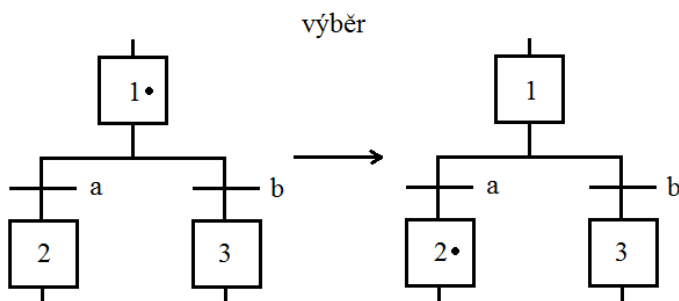


Obr. 1.3.2 Paralelismus

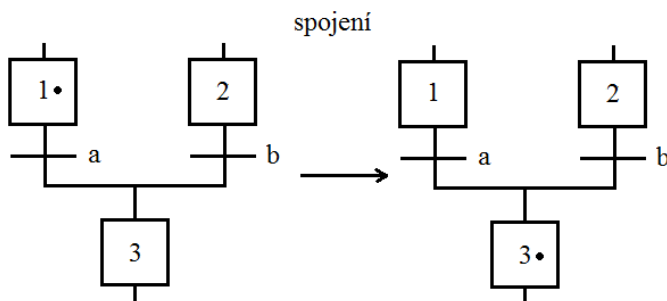


Obr. 1.3.3 Synchronizace

Pro znázornění více přechodů, z nichž po splnění dané podmínky proběhne pouze jeden, se používá *výběr* (divergentní OR). Pokud více přechodů s různými podmínkami navazuje na jeden krok, použije se *spojení* (konvergentní OR).



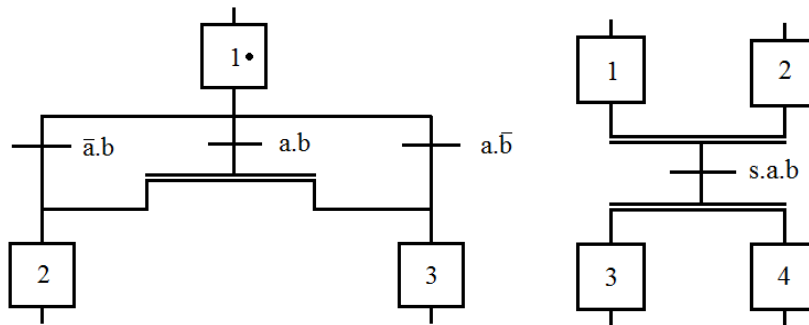
Obr. 1.3.4 Výběr



Obr. 1.3.5 Spojení

Jednou z výhod Grafketu je právě toto zobrazení, které umožňuje snadnou viditelnost sekvencí a případné rozšíření diagramu. To zároveň s sebou přináší i

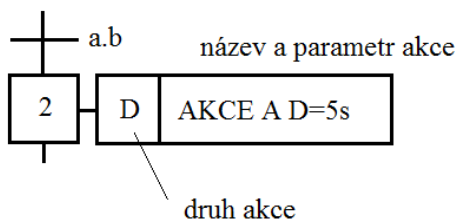
požadavek na určitou preciznost zavedení podmínek, kdy je třeba zaznamenat všechny možné situace (týká se zejména operace výběr). Zde jsou ukázky konstrukce přechodů:



Obr. 1.3.6 Konstrukce přechodů

V grafetu podmínky uskutečnitelnosti představují vstupy automatu a jejich kombinaci, v krocích jsou pak přiřazeny výstupy do technologických (většinou) prvků v akcích.

V diagramu v poli akce se značí písmenem druh akce, v druhé části pole je pak zadáván název a parametr akce.



Obr. 1.3.7 Akce

Druhy akcí a jejich značení:

- se spožděním (*D, delay*) - akce bude vykonána po uplynutí prodlevy od aktivace kroku; parametr: $D=\dots s$
- s časovým limitem (*L, limit*) – akce bude vykonávána po určitý čas od okamžiku aktivace kroku; parametr: $L=\dots s$
- s podmínkou (*C, condition*) – akce je vykonána, je-li krok aktivní a je-li splněna určitá podmínka; parametr: if “b” ($b=1, \text{TRUE}$)
- s pamětí (*S, start/stop*) – akce je vykonávána od aktivace parametrem START až po krok, kde se nachází tatáž akce s parametrem STOP
- s pamětí s časovým limitem (*SL/S, start limit/stop*) – akce je vykonávána počátkem aktivace kroku a ukončena buďto časovým limitem, nebo aktivací kroku, ve kterém je tatáž akce s parametrem STOP

2 Software pro aplikaci Petriho sítí

Kromě běžných požadavků na software pro konstrukci Petriho sítí, mezi které patří především srozumitelnost a uživatelská přívětivost, je pak důležitá především dostupnost produktů, která umožní podporu výuky a rozšíření vědomostí v problematice. S ohledem na rozvinutý trh se dá očekávat přítomnost množství volně šiřitelných programů oproti programům licenčním, které pak budou nejspíše sloužit jako doplněk té konkrétní aplikace a součást většího „balíku“ nástrojů.

Softwaru pro Petriho sítě, který vyhovuje výše zmíněným požadavkům, je možno v dnešní době vyhledat značné množství. Velká část těchto aplikací vznikla jako výsledek činnosti pracovníků a studentů vysokých škol technického směru, což je pro potřeby této práce vhodný předpoklad pro výběr několika zástupců, které zde budou uvedeny jednak jako příklad realizace, a jednak z nich byl zároveň proveden i výběr nejvhodnějšího prostředku pro vyjádření zadaného technologického procesu.

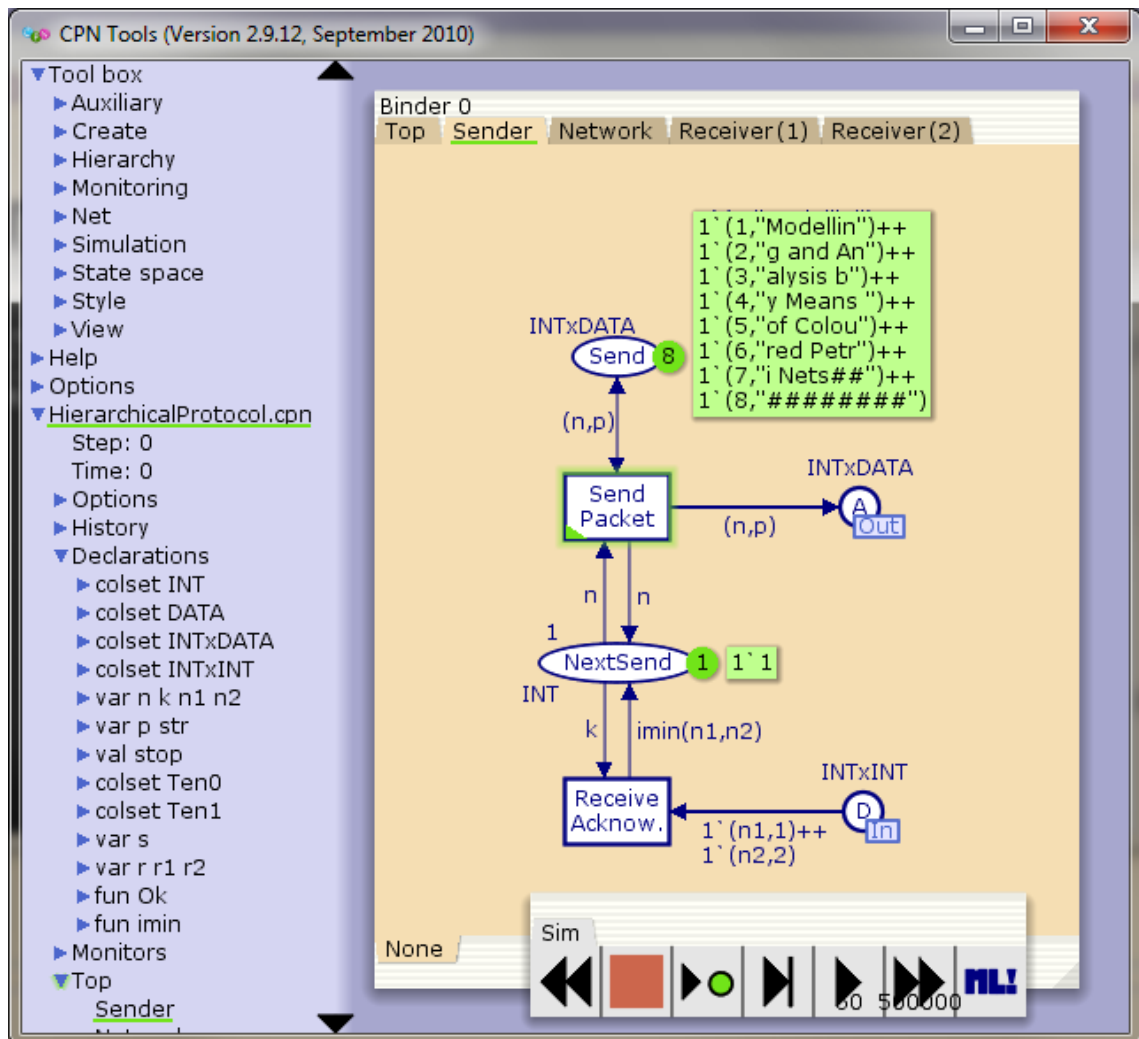
Pro další zkoumání vhodnosti pro podporu syntézy řízení PLC byly vybrány následující programy:

- 1) CPN Tools
- 2) CESim
- 3) PESim
- 4) Netlab
- 5) PIPE
- 6) INA
- 7) PNtalk
- 8) SIPN Editor

1) CPN Tools [1]

Nástupce rozšířeného SW Design/CPN pro konstrukci barevných Petriho sítí je produktem skupiny CPN Group na Aarhus University v Dánsku. V dnešní době je k dispozici verze 3.0.3, která kromě jiného podporuje i 64-bitový operační systém. Firma, která SW vyvíjela, je dosti orientována na zákaznickou podporu, na webových stránkách nabízí různé tutoriály a walkthrough-videa. Přestože má CPN Tools netradičně (ale užitečně) řešený interface, zachovává si technický vzhled. Další výhodou CPN Tools je jeho bezplatné šíření bez nutné registrace, avšak oproti dalším

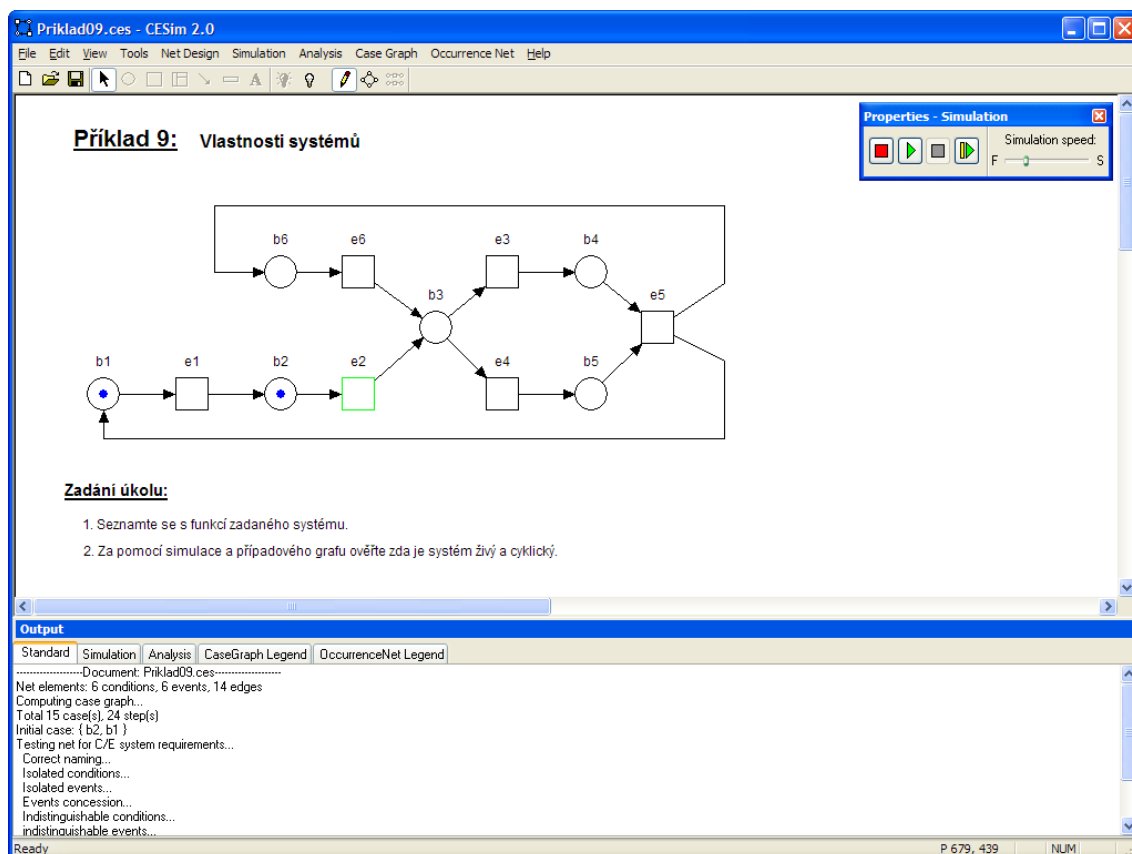
SW pro konstrukci Petriho sítí, které fungují „portable“ (bez instalace), je však nutné program pro spuštění instalovat.



Obr. 2.1 Obrazovka SW prostředí CPN Tools; [s1]

2) CESim [i2]

Tvůrcem CESim – „nástroje pro návrh a simulaci C/E Petriho sítí“ - je Ing. Petr Novosad z Fakulty informačních technologií na VUT v Brně. Nástroj má tak celkem blízko k samotným studentům, kteří konstruují a monitorují navrhovaný proces řízení tak, jak jej viděli před chvílí na „na papíře“. Na webovém odkazu, kde je volně ke stažení i samotný software, se nacházejí vypracované ukázkové příklady.



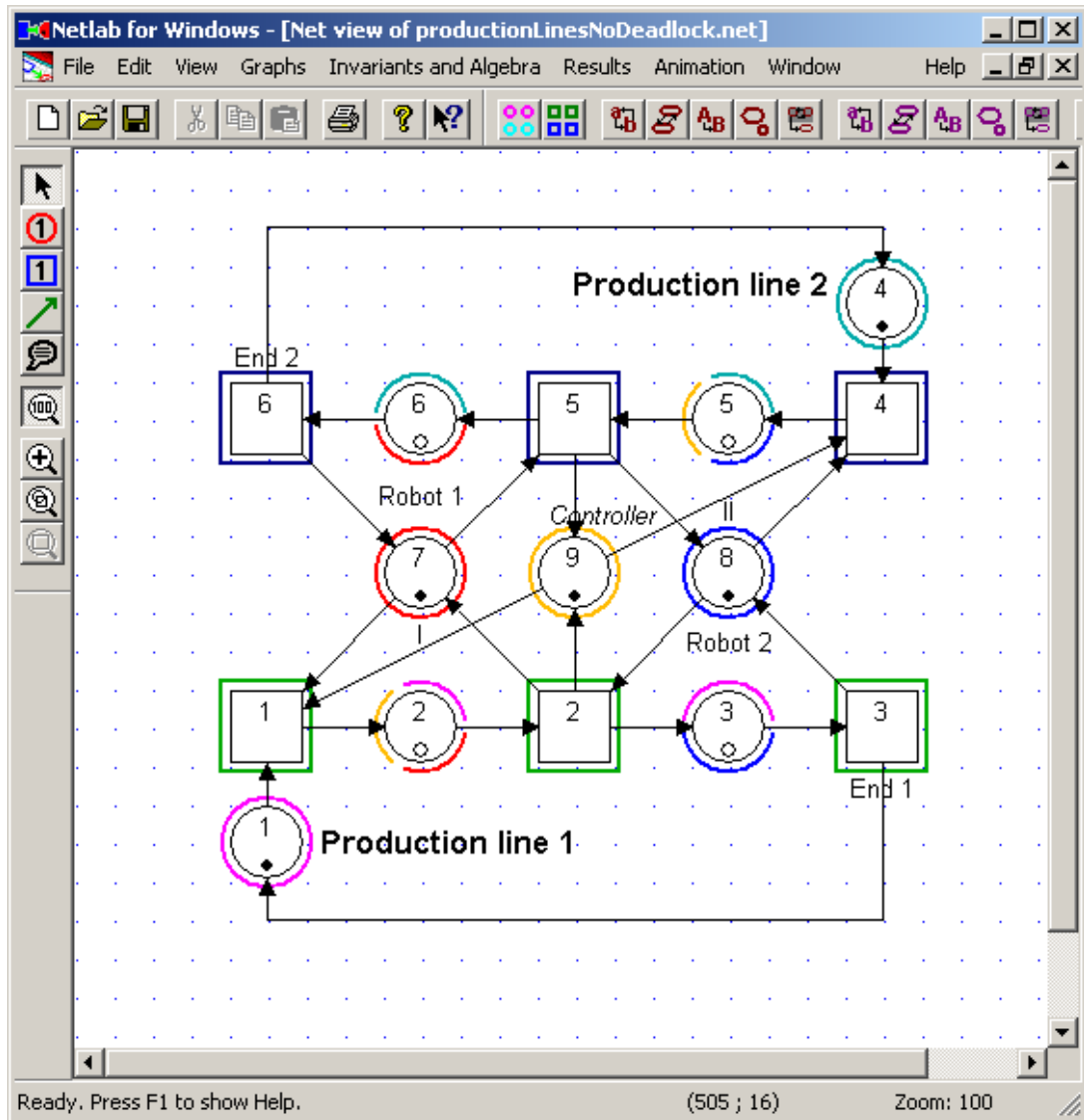
Obr. 2.2 Příklad v SW prostředí CESim; [s2]

3) Pesim [i3]

Dalším nástrojem od kolegů z FIT VUT v Brně je Pesim, který dle popisu slouží k simulaci a analýze Petriho P/T sítí a stochastických sítí. Obsahuje řadu nástrojů, jakými je např. simulace krok za krokem, pravděpodobnostní model, apod.

4) Netlab [i4]

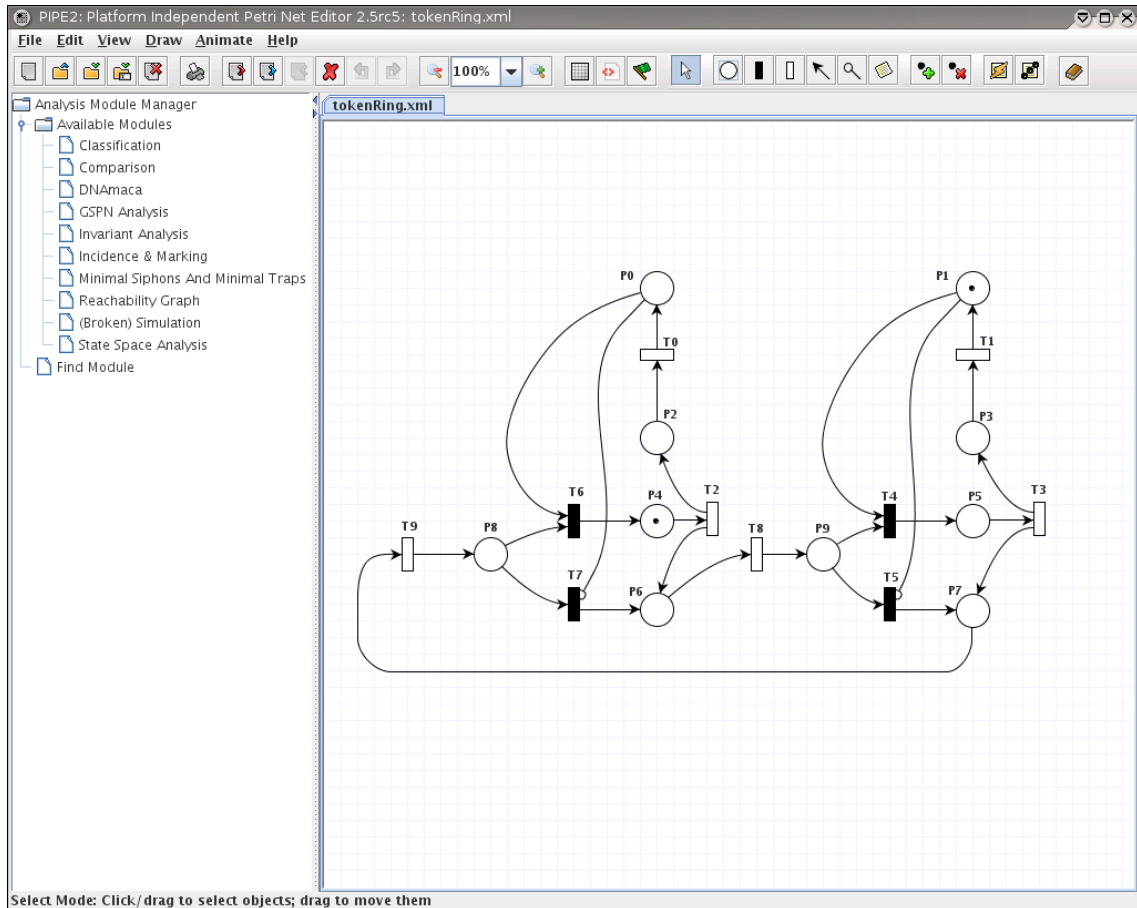
Free SW Netlab je výsledkem práce RWTH Aachen University v Německu. Nabízí sadu obvyklých funkcí a jedná se o další alternativu výukového SW Petriho sítě. Podpora a literatura pro Netlab komunikuje výhradně v německém jazyce.



Obr. 2.3 Obrazovka SW prostředí Netlab; [s3]

5) PIPE [i5]

Platform Independent Petri Net Editor je volně šiřitelný nástroj pro návrh Petriho P/T sítí a je výsledkem projektu MSc. Group na Department of Computing, Imperial College London.Co. Oproti ostatním ovšem nabízí navíc velký potenciál v možném rozsahu plnění funkcí - umožňuje například zoom oblastí, není nutné umisťovat prvky na rastr aj. SW PIPE je šířen jako open-source a přístupný tak k možným modifikacím.



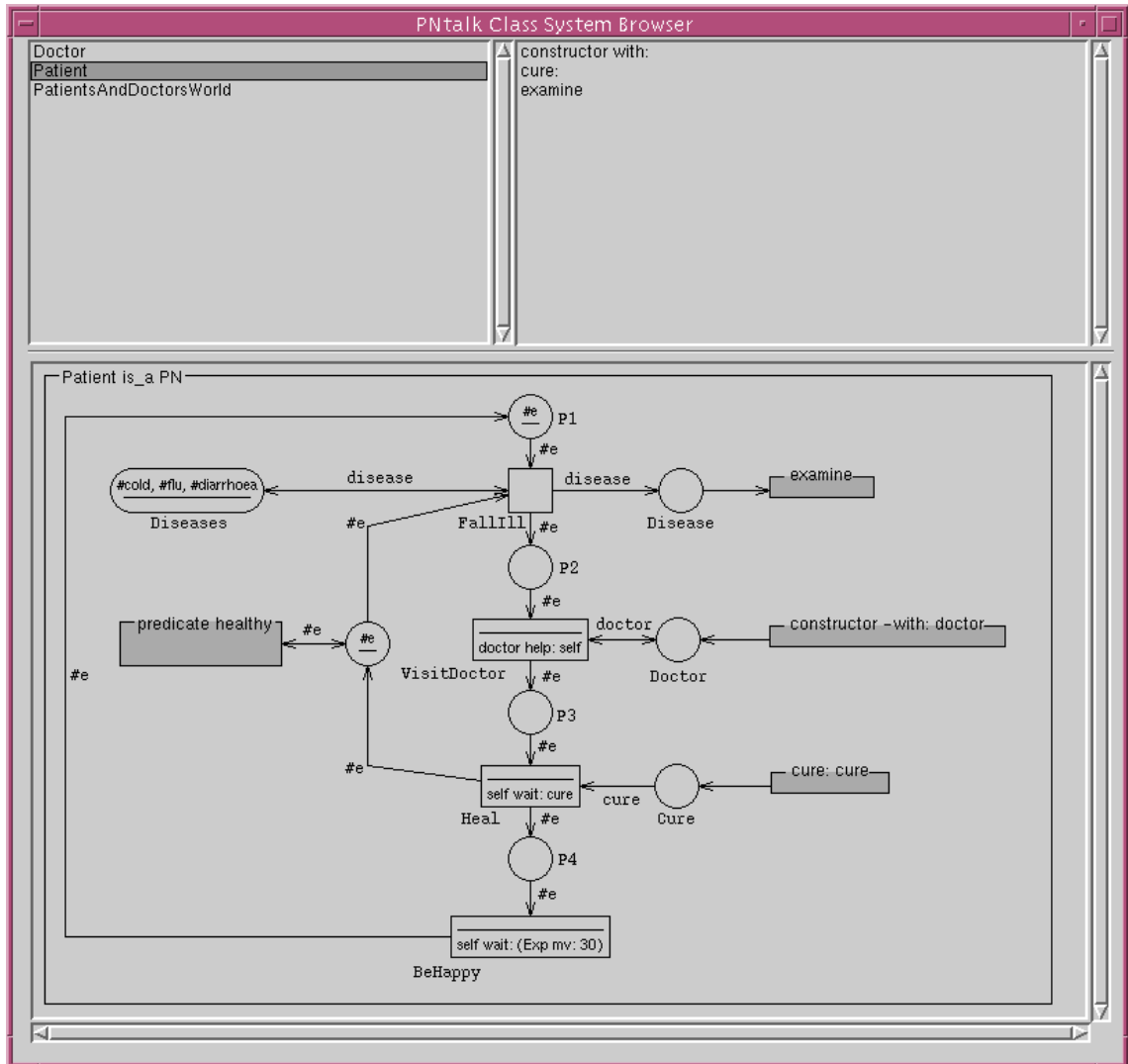
Obr. 2.4 Aplikace v PIPE 2.0; [s4]

6) INA [i6]

INA neboli *Integrated Net Analyzer* představuje další free nástroj pro P/T a barevné Petriho sítě. Poslední verze je z roku 2003, literatura v němčině, SW nejspíše nebude mít již žádnou další podporu. Program je open-source.

7) PNTalk [i7]

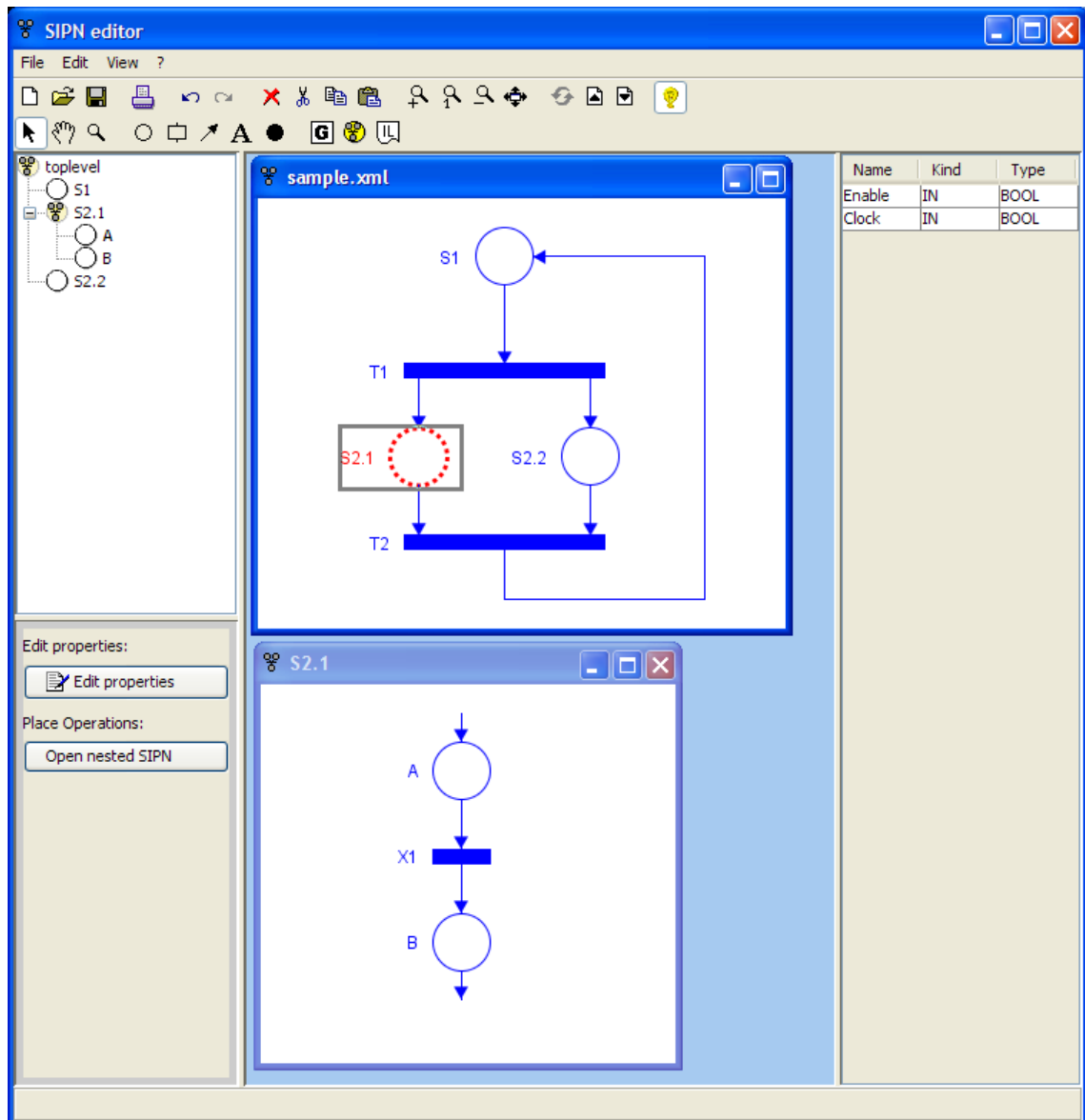
Spíše pro zajímavost než-li pro skutečný výběr softwaru je zde uveden dlouhodobý projekt Ph.D. Vladimíra Janouska a později i dalších, na VUT v Brně. PNTalk představuje nástroj pro tvorbu programově orientovaných Petriho sítí, jehož účelem je umožnit tvorbu sítí pomocí prostředků bližších běžným programovacím jazykům. První verze vyšla v roce 1996, další je dle oficiálních informací stále ve vývoji.



Obr. 2.5 Aplikace v PNTalk; [i7]

8) SIPN Editor [i8]

SIPN Editor je uveden jako grafický nástroj pro konstrukci, analýzu a překlad Petriho sítí užívajících vyjádření signály a je výsledkem práce Georga Freye, Karla Heize a Marka Minase na University of Erlangen v Německu. Je primárně určen pro programátory PLC a tak i definice podmínek přechodů a kroků je v jazyku IL. Výsledný kód je možné nechat přeložit pro další účely, jako např. simulace a vizualizace, do kódu PLC.



Obr. 2.6 Aplikace v SIPN; [i8]

3 Analýza řízení technologického přípravku

3.1 Zadání

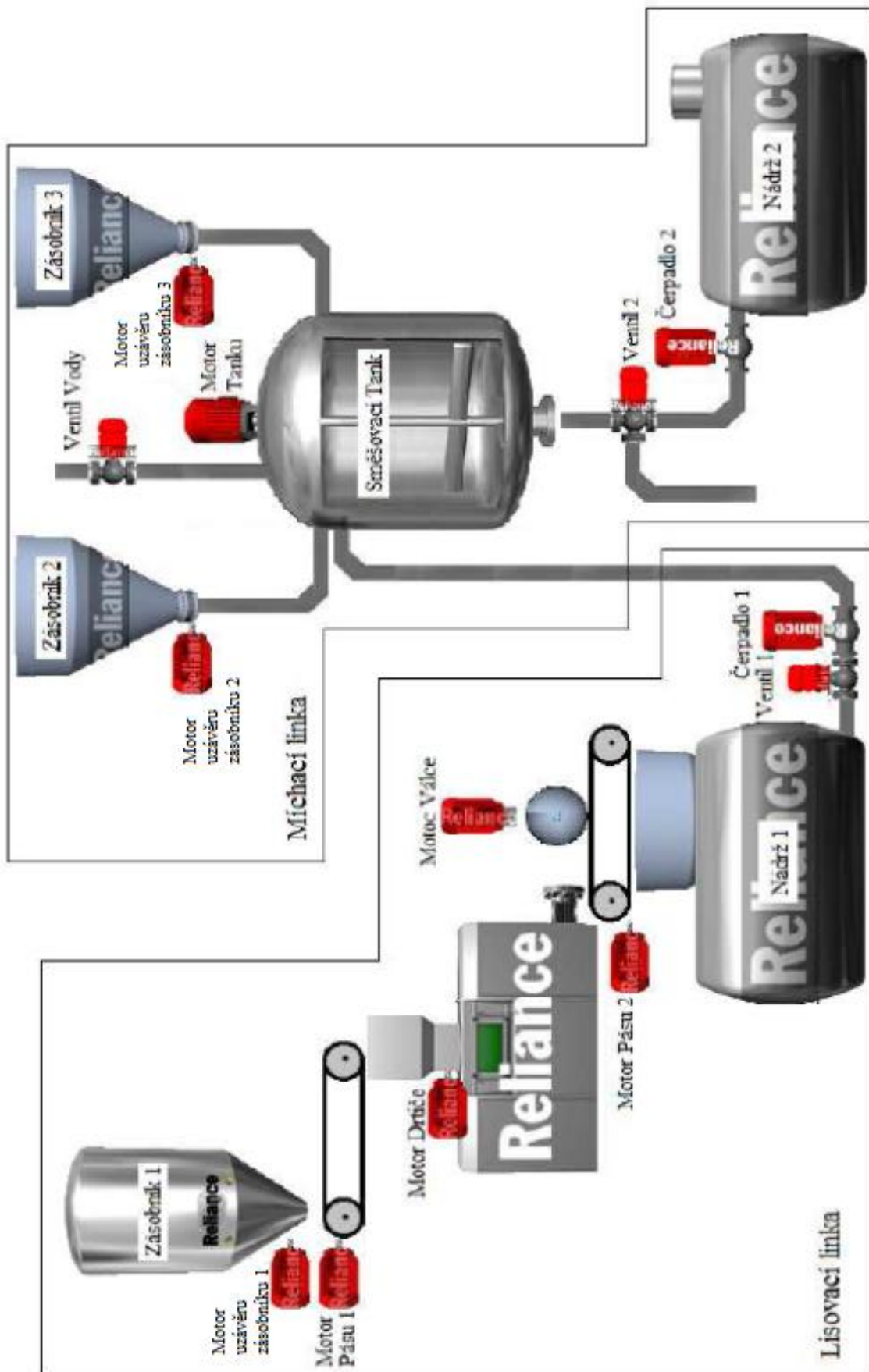
Úkolem bylo navrhnout řízení pro zadanou modelovou technologickou soustavu a simulovat její provoz. Navrhnutá soustava je fiktivní, ale v praxi (v potravinářském průmyslu) bychom našli případy, které se jí podobají.

Technologická soustava se skládá ze dvou technologických linek. Každá z nich může pracovat nezávisle na druhé.

První levá část se nazývá lisovací linka. Skládá se ze zásobníku suroviny, dopravních pásů, drtiče, lisovacího válce, nádrže 1 a ventilu 1 s čerpadlem 1. Úkolem této části je rozdrtit surovinu a vymáčet z ní šťávu. Ta je skladována v nádrži. Zbytky suroviny po drcení se vyhazují a na další zpracování se s nimi nepočítá. Po vylisování válcem jsou pásem 2 dopravovány mimo výrobní halu do kontejnerů a následně vyváženy.

Druhá pravá část se nazývá míchací linka. Obsahuje směšovací tank s míchačkou. Do něj je přečerpána z nádrže 1 vymáčkaná šťáva. Když je tank naplněn do poloviny objemu, je přívod šťávy uzavřen. Dále se do tanku přidají dvě přísady ze zásobníku 2 a 3 v poměru 20% a 15 % objemu tanku. Zbýlých 15 % objemu tanku se doplní vodou. Poté je uvedena do činnosti míchačka. Suroviny se míchají po stanovenou dobu. Po uplynutí této doby je směs hotová a vypouští se do nádrže 2. Po dokončení vypouštění je nádoba tanku vypláchnuta vodou a je připravena na další cyklus. Hotová směs se z nádrže 2 odčerpává k dalšímu zpracování.

Následující strana obsahuje obrázek technologické soustavy s popisem jejích prvků:



Obr. 3.1.1 Zadaná technologická soustava

3.2 Výběr software

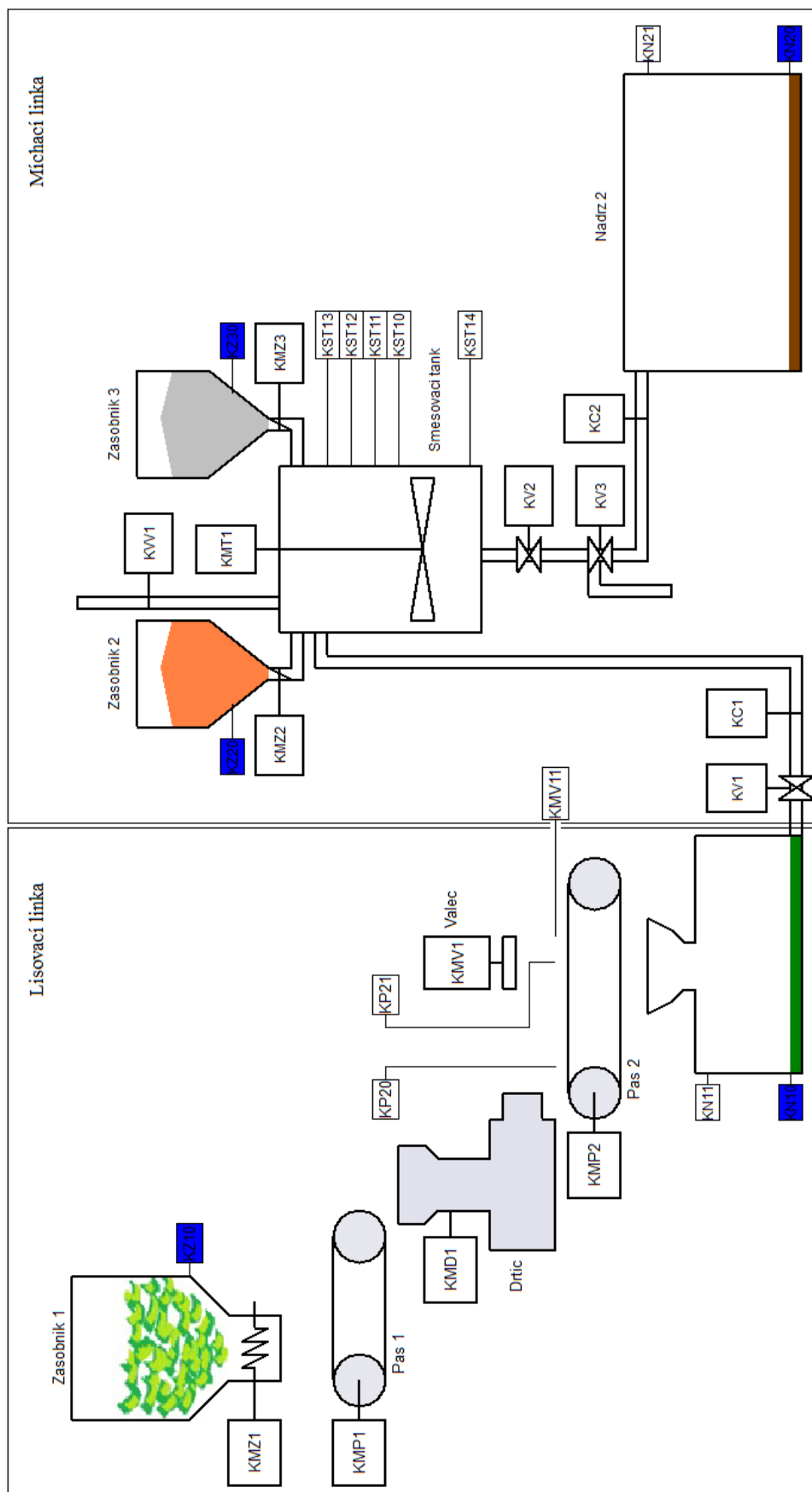
Pro Petriho síť bylo původně zamýšleno znázornění činnosti a simulace sekvencí technologických linek vytvořit využitím programu CPN Tools, který se pro svou variabilitu zdál být vhodným kandidátem pro tuto úlohu, ovšem nepředvídatelné technické problémy programu během pokročilých voleb simulace znemožnily pokračování v práci. Po vyzkoušení dalších zástupců bylo nakonec rozhodnuto vytvořit Petriho síť v programu SIPN.

Pro vytvoření algoritmu grafcet a sestrojení schémat zapojení byl vybrán SW FluidSIM (verze 4.0) fy Festo.

Pro samotné řízení úlohy bylo k dispozici PLC CPX-CEC a k jeho naprogramování SW prostředí CoDeSys, jehož součástí je i vizualizační prostředí.

3.3 Vytvoření algoritmu ovládacího automatu

Pro návrh řízení a jeho simulaci bylo nutno zadání upřesnit. Nebylo zapotřebí zásahů do technologie, jen určitých modifikací pro snazší realizaci řízení. Jedná se zejména o doplnění snímačů poloh pásu a hladin v nádržích a směšovacím tanku, a pak zejména nahrazení ventilu směšovacího tanku („Ventil 2“) dvěma ventily V2 (uzavírací) a V3 (dvoucestný). Ventil 1 a čerpadlo 1 byly přesunuty do linky 2. Schéma na další straně znázorňuje provedené změny a umístění prvků potřebných pro realizaci:



Obr. 3.3.1 Schéma technické soustavy s akčními členy a senzory; [s6]

	Označení	Význam	Komentář
Vstupy	S1	tlačítko „Start“ stisk: S1 = 1	start linky 1
	S2	tlačítko „Start“ stisk : S2 = 1	start linky 2
	A1	tlačítko „Auto“ stisk: A1 = 1	zapnutí režimu auto linky 1
	A2	tlačítko „Auto“ stisk: A2 = 1	zapnutí režimu auto linky 2
	P20	materiál v poloze 0: P20 = 1	levá krajní poloha pásu 1
	P21	materiál v poloze 1: P21 = 1	střední poloha pásu 1
	MV11	válec v dolní poloze: MV11 = 1	koncová poloha válce (spodní)
	N10	hladina na hodnotě 0: N10 = 1	prázdná nádrž 1
	N11	hladina na hodnotě 1: N11 = 1	plná nádrž 1
	ST10	hladina na hodnotě 0: ST10 = 1	tank plný na 50 %
	ST11	hladina na hodnotě 1: ST11 = 1	tank plný na 70 %
	ST12	hladina na hodnotě 2: ST12 = 1	tank plný na 85 %
	ST13	hladina na hodnotě 3: ST13 = 1	tank plný na 100 %
	ST14	hladina na hodnotě 4: ST14 = 1	tank prázdný
	N20	hladina na hodnotě 0: N20 = 1	prázdná nádrž 2
	N21	hladina na hodnotě 1: N21 = 1	plná nádrž 2
	Z10	materiál v poloze 0: Z10 = 1	zásobník 1 není prázdný
	Z20	materiál v poloze 0: Z20 = 1	zásobník 2 není prázdný
	Z30	materiál v poloze 0: Z30 = 1	zásobník 3 není prázdný
	Výstupy	I1	I1 = 1: kontrolka „Ready“ svítí
I2		I2 = 1: kontrolka „Ready“ svítí	linka 2 připravena na další cyklus
MZ1		MZ1 = 1: motor KMZ1 běží	doprava suroviny 1
MP1		MP1 = 1: motor KMP1 běží	pás 1 spuštěn
MD1		MD1 = 1: motor KMD1 běží	drtič spuštěn
MP2		MP2 = 1: motor KMP2 běží	pás 2 spuštěn
MV1		MV1 = 1: válec KMV1 vyjede MV1 = 0: válec KMV1 zajede	válec lisuje válec se vrací
V1		V1 = 1: ventil KV1 otevřen	otevření nádrže 1
C1		C1 = 1: čerpadlo KC1 běží	přečerpávání z nádrže 1 do tanku
MZ2		MZ2 = 1: motor KMZ2 běží	doprava suroviny 2
MZ3		MZ3 = 1: motor KMZ3 běží	doprava suroviny 3
VV1		VV1 = 1: ventil KVV1 otevřen	napouštění vody
MT1		MT1 = 1: motor KMT1 běží	míchání
V2		V2 = 1: ventil KV2 otevřen	vypouštění tanku
V3		V3 = 1: ventil KV3 otevřen dolu V3 = 0: ventil KV3 otevřen vlevo	vypouštění tanku do nádrže 2 vypouštění tanku ven
C2		C2 = 1: čerpadlo KC2 běží	přečerpávání z tanku do nádrže 2

Tab. 1 Tabulka proměnných

	krok	1	2	3	4	5	6	7	8	9	10	11	12	13
Linka 1	MZ1	1	0	0	1	0	0							
	MP1	1	0	0	1	0	0							
	MD1	1	0	0	1	0	0							
	MV1	0	0	0	1	0	0							
	MP2	0	1	0	0	1	0							
	I1	0	0	1	0	0	1							
Linka 2	V1				1	0	0	0	0	0	0	0	0	1
	C1				1	0	0	0	0	0	0	0	0	1
	MZ2				0	1	0	0	0	0	0	0	0	0
	MZ3				0	0	1	0	0	0	0	0	0	0
	VV1				0	0	0	1	0	0	1	0	0	0
	MT1				0	0	0	0	1	0	0	0	0	0
	V2				0	0	0	0	0	1	0	1	0	0
	V3				0	0	0	0	0	0	0	1	0	0
	C2				0	0	0	0	0	1	0	0	0	0
	I2				0	0	0	0	0	0	0	0	1	0

Tab. 2 Tabulkový zápis činností

Tabulka znázorňuje chod linek při prvním spuštění (např. po instalaci, po údržbě):

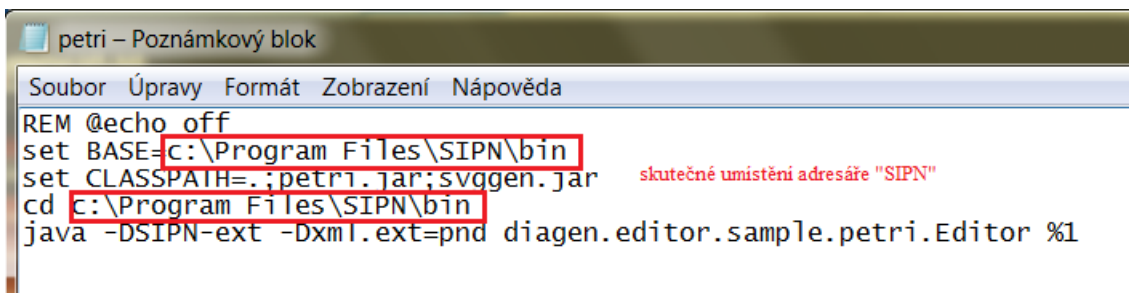
Na lince 1 se v prvním kroku naplní pás 2, dále dojde k posunu pásu 2 a pak teprve lisování v kroku 2 zároveň s dalším plněním pásu 2. Odpad je dopraven pásem vpravo (např. do beden). Na konci cyklu je rozsvícena indikace konce cyklu I1. V dalším cyklu (zvýrazněné sloupce) je pás již naplněn z cyklu předchozího a lisování nastává okamžitě. Zda bude válec MV1 lisovat hned v prvním cyklu bude rozhodovat podmínka přítomnosti materiálu v poloze pod válcem.

Linka 2 může začít pracovat až v kroku 4 linky 1 (po vylisování šťávy, nádrž 1 je alespoň z části naplněna). V prvním kroku se přečerpává vylisovaná šťáva z nádrže 1 do směšovacího tanku. Po naplnění do poloviny se postupně přidávají suroviny 2, 3 a voda. Po určité době míchání v tanku je otevřen ventil V2, V3 a puštěno čerpadlo C2 (krok 9), směs přečerpána do nádrže 2. V2 a V3 se uzavřou/přesměrují, tank je vymyt (zcela naplněn vodou), voda vypuštěna otevřením V2 ven. V2 je uzavřen a je rozsvícena indikace konce cyklu I2. Začátek dalšího cyklu (zvýrazněný sloupec) je beze změny (rozdíl oproti lince 1).

Petriho síť

Petriho síť byly vytvořeny v programu SIPN editor, který umožňuje Petriho síť zkonstruovat a znázornit. Další odstavce mohou posloužit jako jednoduchý návod na obsluhu.

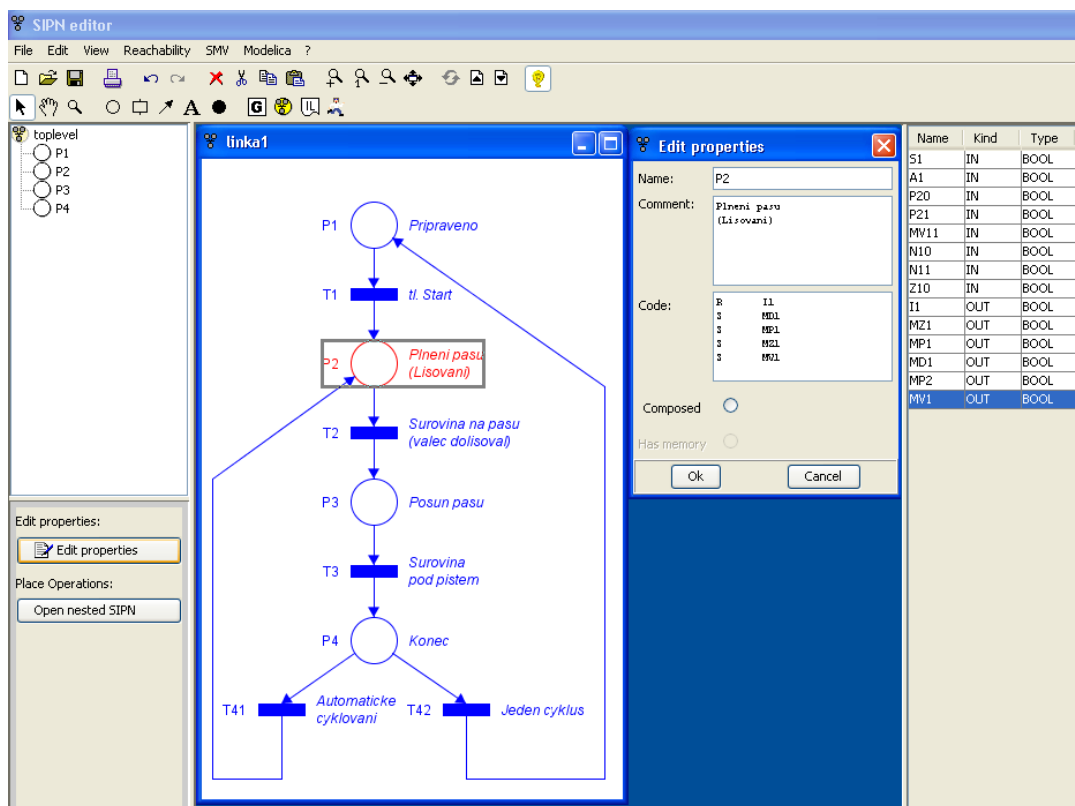
Instalace programu spočívá v přepsání cesty zdroje na skutečné umístění extrahovaného adresáře SIPN. Cesta zdroje se nachází v dávkovém souboru „Petri.bat“ a lze ji upravit pomocí textového editoru, např. Poznámkového bloku v OS Windows:



```
REM @echo off
set BASE=C:\Program Files\SIPN\bin
set CLASSPATH=.;petri.jar;svggen.jar
cd C:\Program Files\SIPN\bin
java -DSIPN-ext -Dxml.ext=pnd diagen.editor.sample.petri.Editor %1
```

Obr. 3.3.2 Instalace SIPN

Po samotném spuštění souboru „Petri.bat“ je možno bez dalšího nastavování vytvářet Petriho síť.



Obr. 3.3.3 Obrazovka v SIPN; [s5]

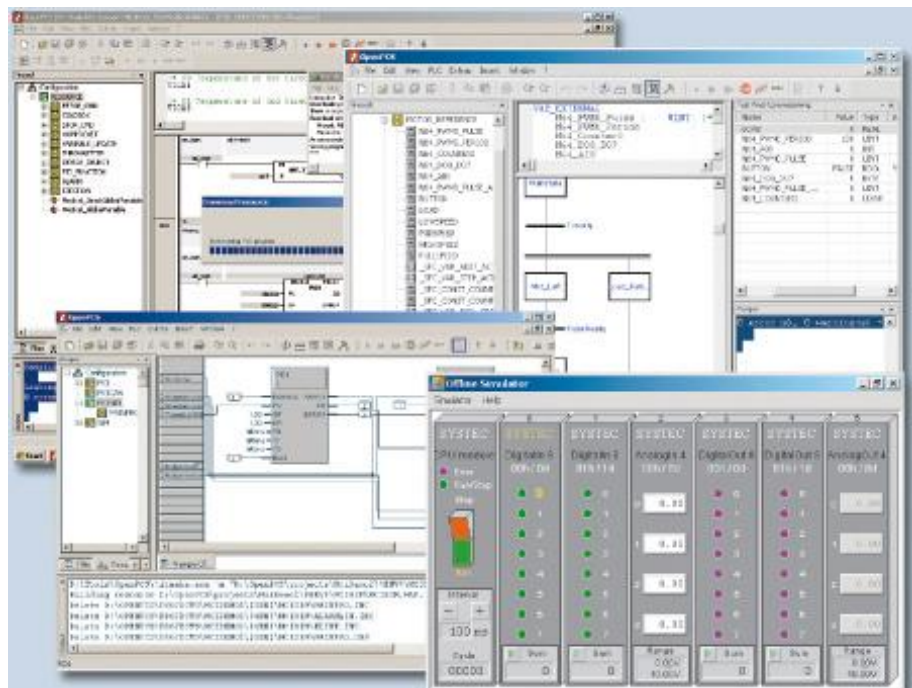


Obr. 3.3.4 Panel nástrojů v SIPN; [s5]

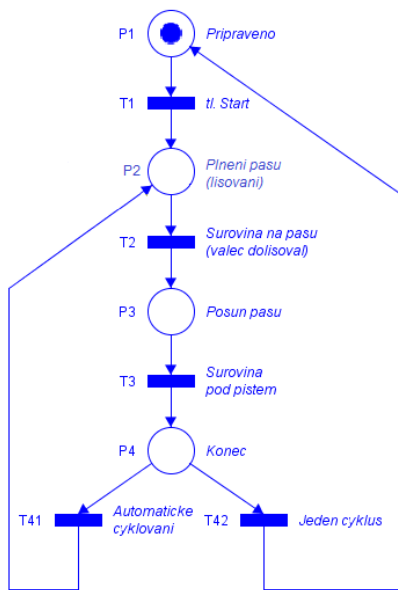
Všechny potřebné objekty lze vkládat z horního panelu nástrojů označením ikonky a kliknutím levého tlačítka myši do pracovního prostoru. Označením vloženého prvku a kliknutím na „Edit properties“ se editují podmínky přechodů a akce míst v jazyce IL. Protože SW SIPN slouží ke generování kódu pro PLC, podmínky a akce jsou reprezentovány prostřednictvím funkcí globálních proměnných, které lze definovat rovněž v panelu nástrojů a přiřadit jim adresy. Jejich výpis se nachází v pravé části obrazovky.

Pozn.: Hrany se vkládají kliknutím levého tlačítka myši na výchozí prvek a pravého tlačítka na prvek cílový.

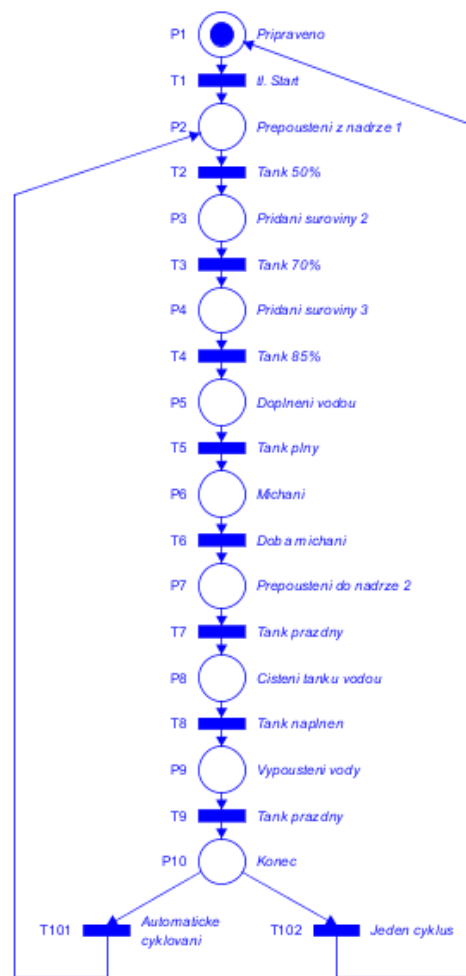
Pro další použití je třeba sítě přeložit do jazyka IL nebo SMV (v panelu nástrojů). Simulace a vizualizace takto vytvořených Petriho sítí je pak možná v dalších programech, jako je např. autory SW SIPN doporučovaný multifunkční nástroj OpenPCS.



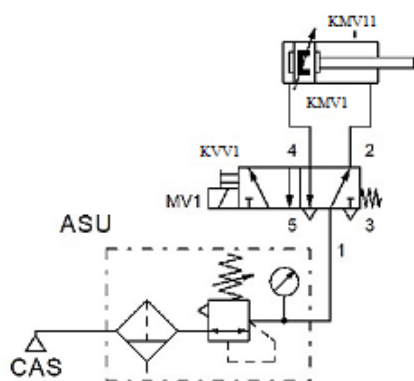
Obr. 3.3.5 Obrazovka prostředí OpenPCS; [i10]



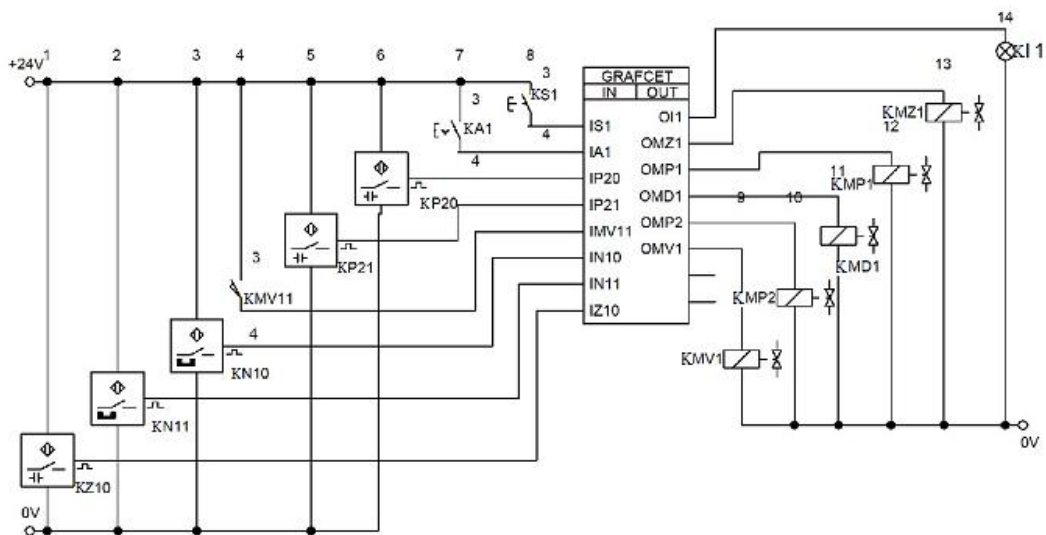
Obr. 3.3.6 Linka 1 Petriho síť; [s5]



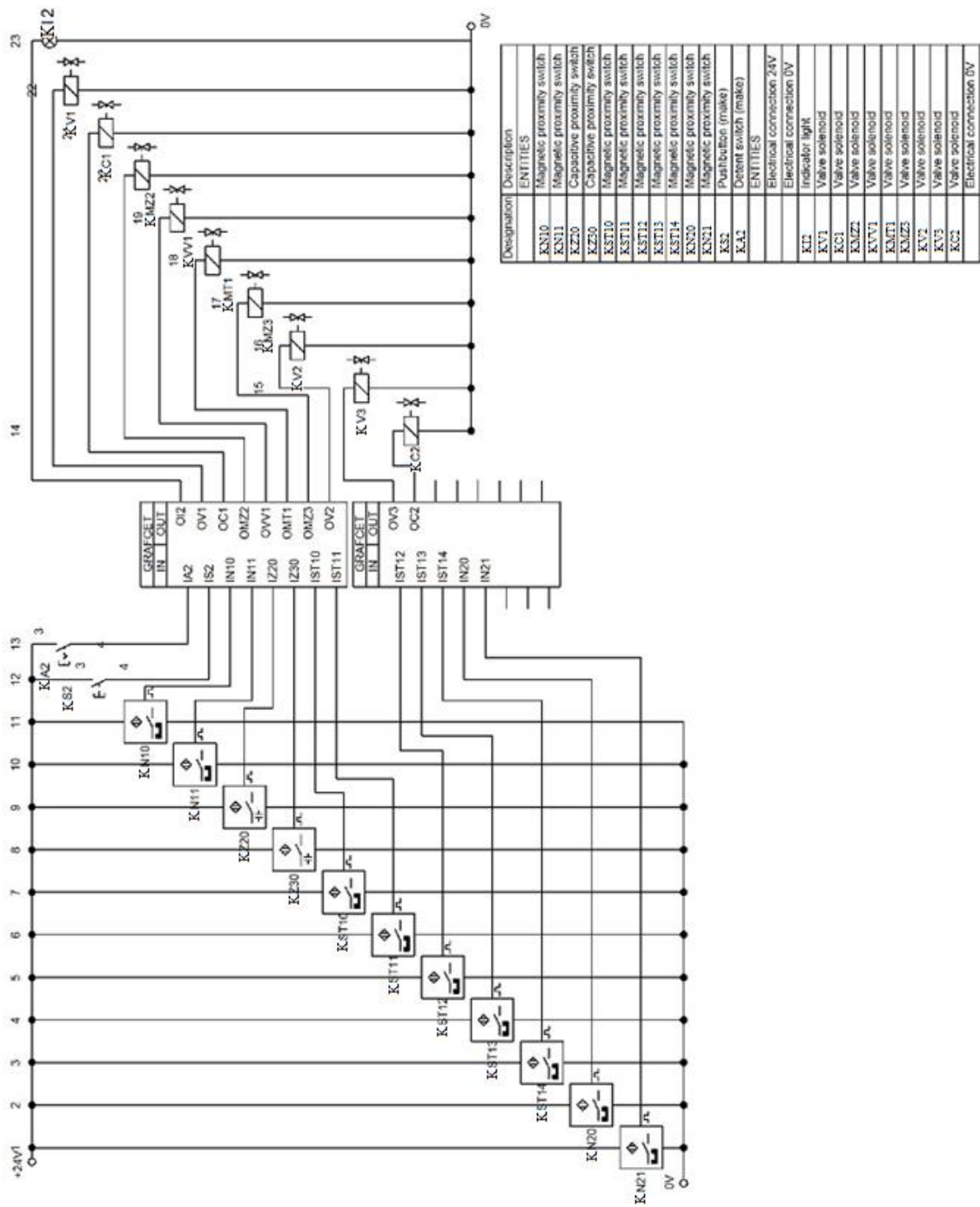
Obr. 3.3.7 Linka 2 Petriho síť; [s5]



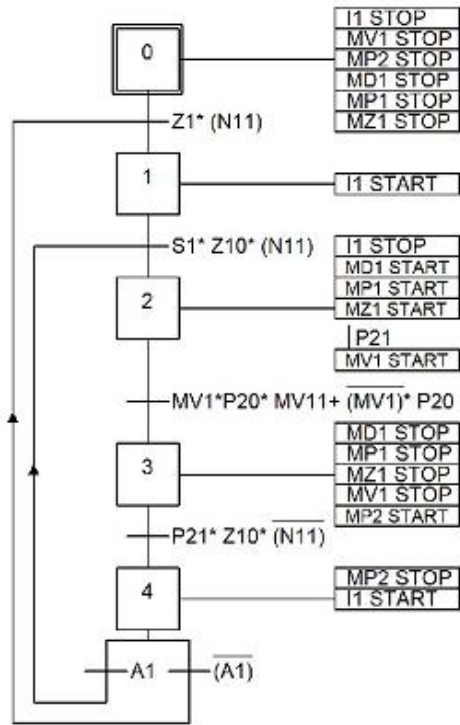
Designation	Description
KMV1	Double acting cylinder
	Distance rule
KMVV1	5/2-way solenoid valve
ASU	Air service unit
CAS	Compressed air supply
ENTITIES	
KS1	Pushbutton (make)
KA1	Detent switch (make)
KP20	Capacitive proximity switch
KP21	Capacitive proximity switch
KMV11	Make switch
KN10	Magnetic proximity switch
KN11	Magnetic proximity switch
KZ10	Capacitive proximity switch
KI1	Indicator light
KMZ1	Valve solenoid
KMP1	Valve solenoid
KMD1	Valve solenoid
KMP2	Valve solenoid
KMV1	Valve solenoid
	Electrical connection 24V
	Electrical connection 0V
	Electrical connection 0V



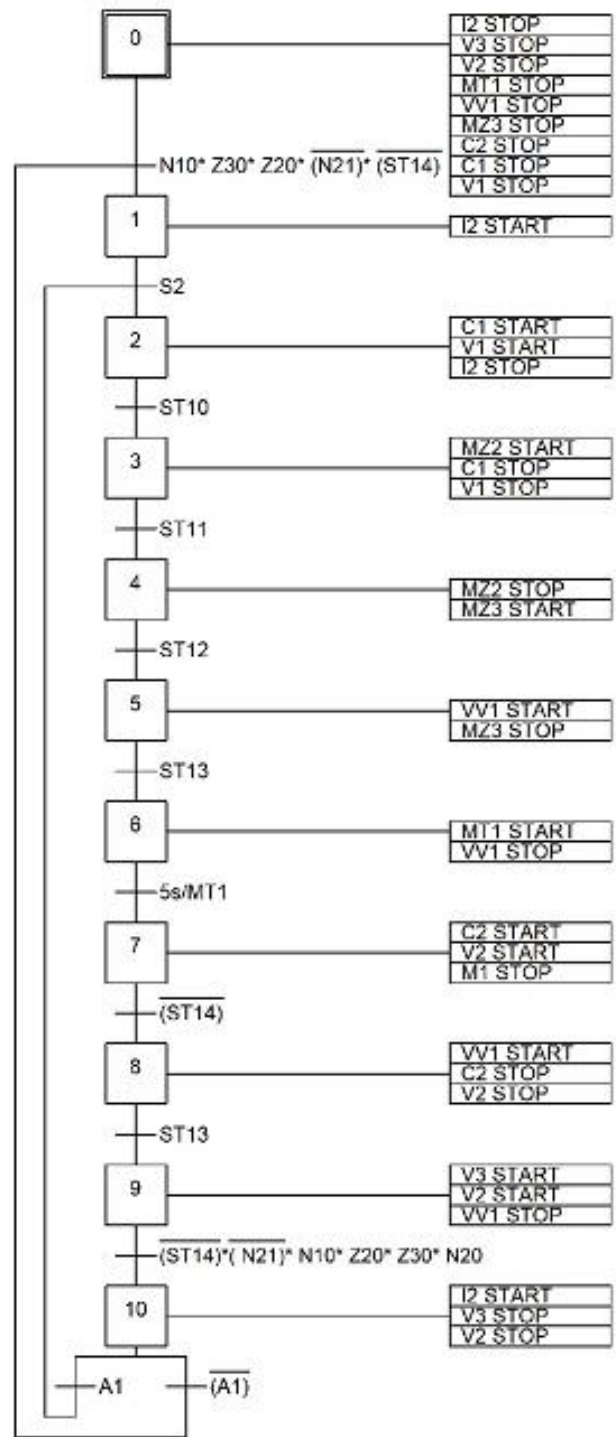
Obr. 3.3.8 Elektropneumatické a obvodové schéma linky 1; [s7]



Obr. 3.3.9 Obvodové schéma linky 2; [s7]



Obr. 3.3.10 Linka 1 grafcet; [s7]



Obr. 3.3.11 Linka 2 grafcet; [s7]

4 Řízení přípravku v prostředí CoDeSys

Pro danou technologii pro řešení této BDP zadáno řízení v SW CoDeSys, což je multifunkční řídicí a simulační nástroj společnosti 3-S Software pro řídicí jednotky Festo typu CPX-CEC. Jeho verze 2.3.9.19 je volně ke stažení na [i9], a přestože se jedná o dnes již starší verzi, bohatě pokrývá potřeby pro řešení automatizačních úloh a i samotná firma Festo s touto verzí při nabídce svých novějších produktů počítá.

CoDeSys obsahuje všechny jazyky normy IEC 1131-3 a lze je mezi sebou různě kombinovat prostřednictvím vnořených vrstev, které mezi sebou pomocí deklarovaných proměnných navzájem komunikují. Funkci jednotlivých prvků, jakými jsou např. funkční bloky, lze též celou ručně naprogramovat.

Nedílnou součástí programu je rovněž vizualizace, kterou lze sestavit z předdefinovaných vložitelných objektů, kterým lze přiřadit pomocí dialogových oken různé vlastnosti a funkce. Mezi tyto objekty patří i vytváření diagramů či měření reálného času procesu apod.

4.1 Vytvoření programu v SW CoDeSys

Hlavní účel mé práce spočívá v seznámení budoucího uživatele se SW a v implementování technických znalostí při tvorbě funkčního programu pro kompletní obsluhu technologického procesu prostřednictvím reálného PLC.

Jednak kvůli požadavkům na dokumentaci tvorby programu a pak i zejména pro absenci použitelné nápovědy přímo v CoDeSys bude v následujících odstavcích podrobněji popsán vývoj program, který může sloužit jako návod. Samotný algoritmus a soupis potřebných proměnných byl již uveden na předchozích stranách a slouží jako teoretický základ řešení, který je vhodné vytvořit si ještě před spuštěním samotného vývojového prostředí.

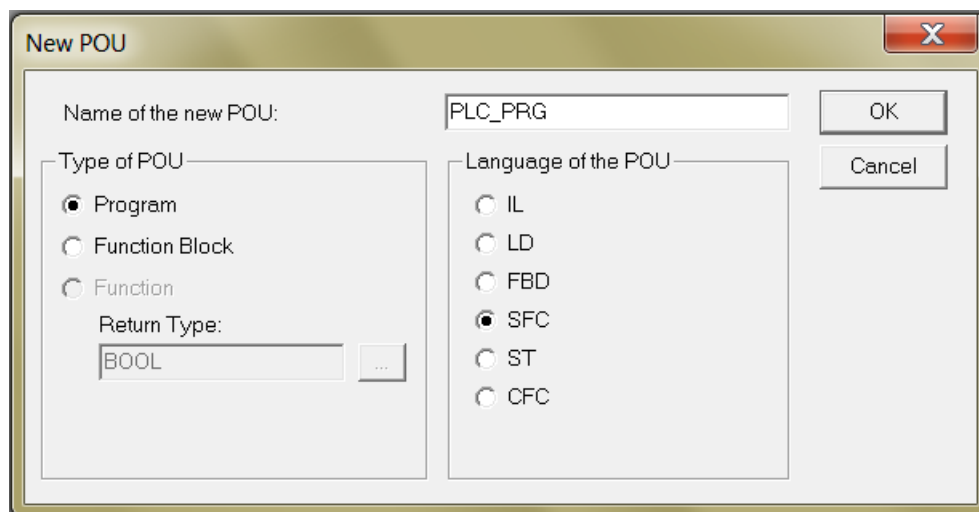
4.2.1 Řídicí program

Před vysvětlením ovládání SW CoDeSys budou zavedeny tyto pojmy:

LT – kliknutí levým tlačítkem myši

PT – kliknutí pravým tlačítkem myši

Po spuštění instalovaného SW CoDeSys (verze 2.3.9.19) a výběru hardwarové konfigurace v menu *File->New*, kterou lze i případně ručně vytvořit, následuje vytvoření první POU (pozn. *Program Organization Unit*, hierarchicky sestavitelná část programu) v některém z jazyků dle normy IEC 1131-3. V našem případě zvolíme typ POU *Program* a jazyk *SFC* pro hlavní řídicí program PLC, který bude organizačně téměř shodný s dříve uvedeným algoritmem v Grafcet.



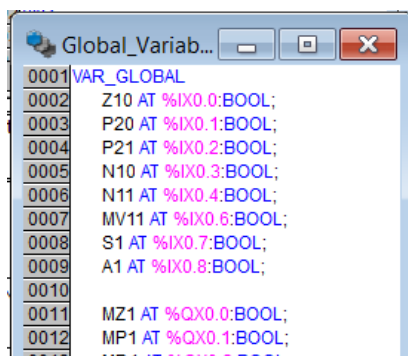
Obr. 4.2.1 Nová POU; [s6]

V levé dolní části okna jsou 4 záložky – *POUs*, *Data types*, *Visualisations* a *Resources*.



Obr. 4.2.2 Záložky; [s6]

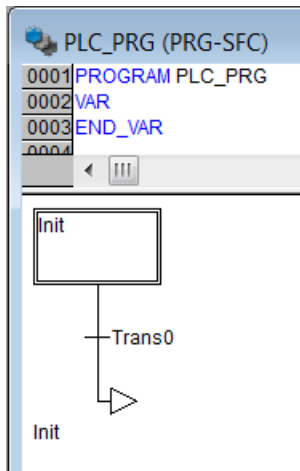
To je seznam všech vytvořených a použitých prvků. Hned na začátku je vhodné deklarovat globální proměnné pro vstupy a výstupy na PLC a jejich adresy, abychom s nimi mohli pracovat hned ze začátku. 4. záložka *Resources->Global Variables->Global Variables* (2xLT), získáme okno, kam zapíšeme proměnné v následujícím tvaru:



Z10 – název proměnné
 %IX0.0 – adresa 0.0 na IN u PLC
 (%QX0.0 na OUT)
 BOOL – datový typ Bool pro bin. řízení

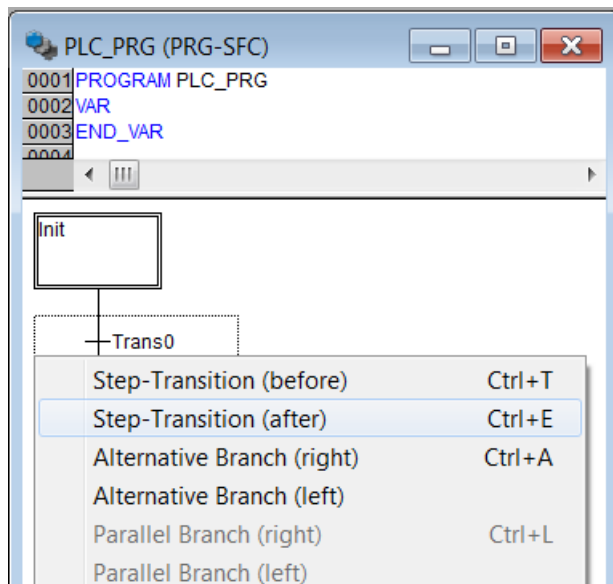
Obr. 4.2.3 Globální proměnné; [s6]

Po vytvoření první POU se v okně s názvem POU a typem jazyka zvoleného pro programování v záorce „(PRG-SFC)“ se objeví první krok *Init* s prvním přechodem *Trans0*. Šipka pod přechodem znázorňuje tzv. skok (angl. *jump*), což je v podstatě návratová větev odkazující na některý krok. Pro správnou funkci skoku musí být skok shodně pojmenovaný s odkazovaným krokem; lze jej přejmenovat (LT na jeho název).



Obr. 4.2.4 Hlavička POU; [s6]

Pomocí 2xLT na přechod/krok získáme menu podobné volbě nové POU, kde je výběr jazyka, ve kterém chceme daný přechod/krok naprogramovat. Pomocí PT na krok nebo přechod se rozbálí menu, ve kterém lze nastavit další krok, přechod, možnosti, paralelní větve apod. Pro náš program budeme vytvářet kroky a přechody v jazyce LD a FBD.

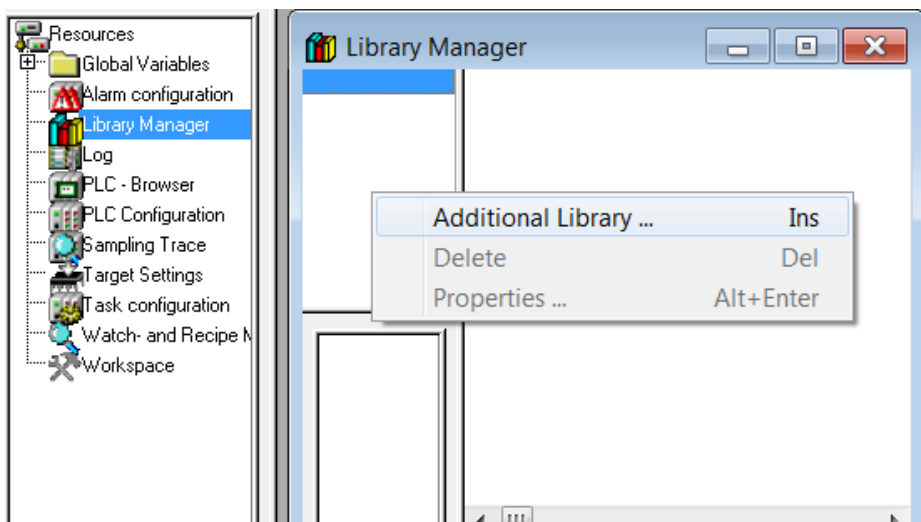


Obr. 4.2.5 Vytvoření nového přechodu; [s6]

Pozn.: Při vytváření přechodů v jazyce LD musíme mít na paměti, že přechod nesmí obsahovat cívky. V jazyce FBD se vstupy a výstupy přiřazují výběrem v menu bloku (PT na daný blok). Pro přechody se nepřičítají proměnné na výstup. Pro jednoduché

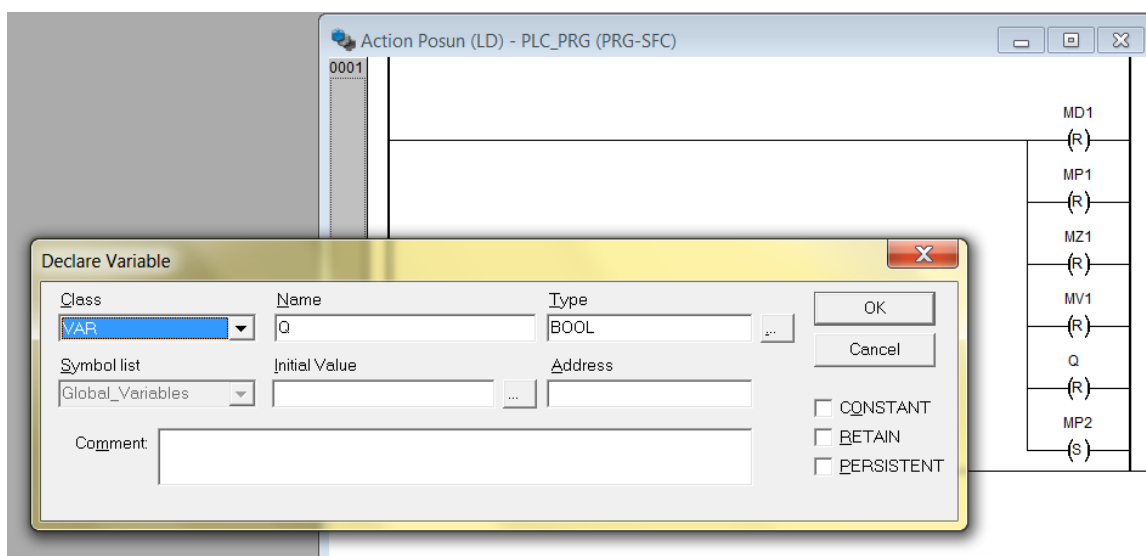
přechody (jedna podmínka) můžeme také přechod pojmenovat právě jeho podmínkou a pokud bude proměnná hodnoty TRUE (=1), přechod proběhne.

Pro použití funkčních bloků, jako je časovač typu TON apod., je zapotřebí do projektu importovat knihovnu „Standard.LIB“ v kořenovém adresáři v *Library manager* umístěného ve stejné záložce, kde jsou globální proměnné:



Obr. 4.2.6 Přidávání knihoven; [s6]

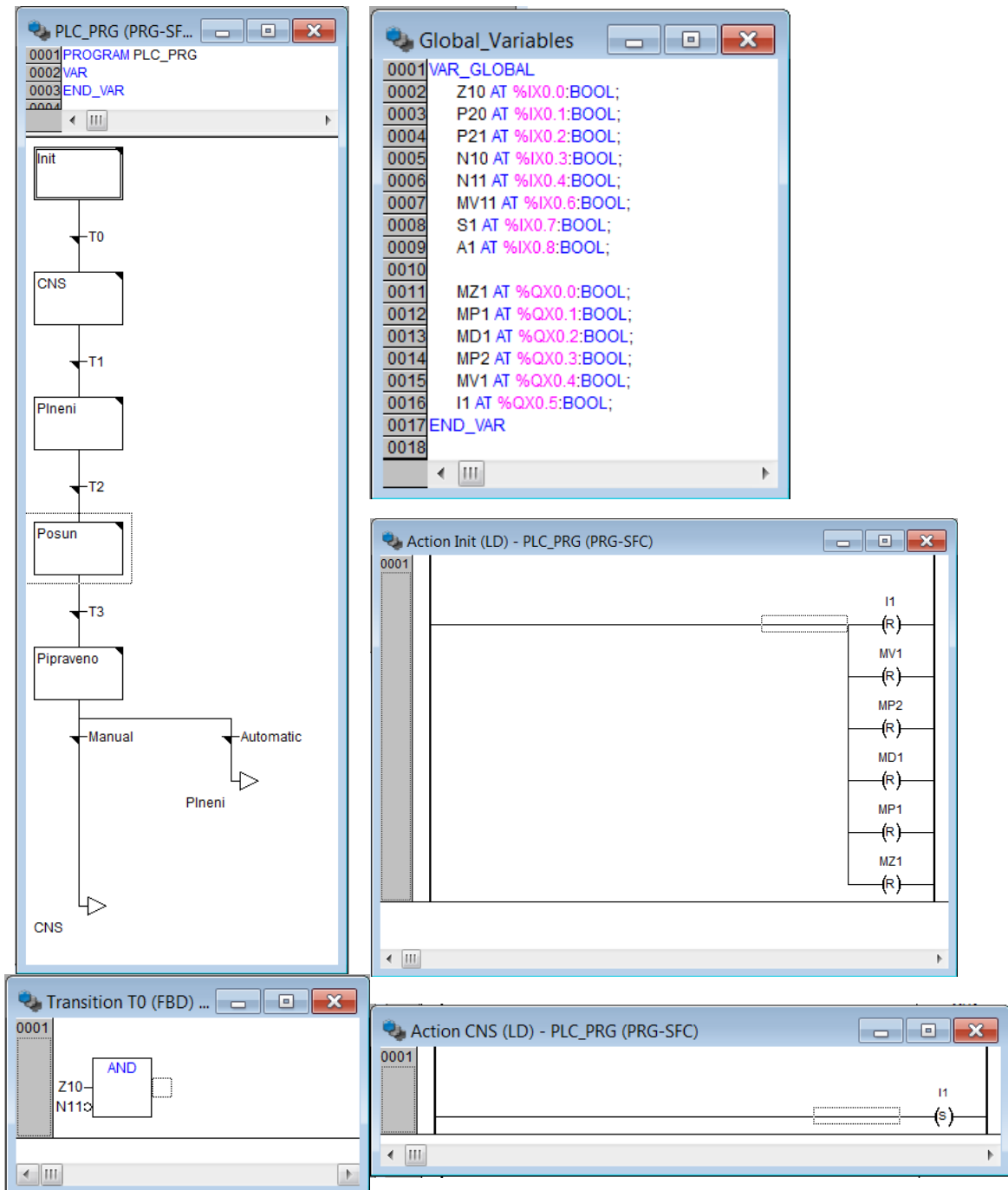
Nyní víme všechny potřebné informace a můžeme začít tvořit řídicí program. Pokud potřebujeme novou proměnnou, i tu která není globální a využijeme ji jen v rámci dané POU (ovšem lze se na ni i v nadřazených POUs odvolávat jejím použitím v nadřazené POU), jednoduše jejím jménem nazveme daný prvek (např. cívku v jazyce LD) a hned po skončení jeho editace se automaticky objeví okno pro výběr typu proměnné, adresy apod. SW ji pak sám deklaruje do deklarační části POU.



Obr. 4.2.7 Deklarace proměnných; [s6]

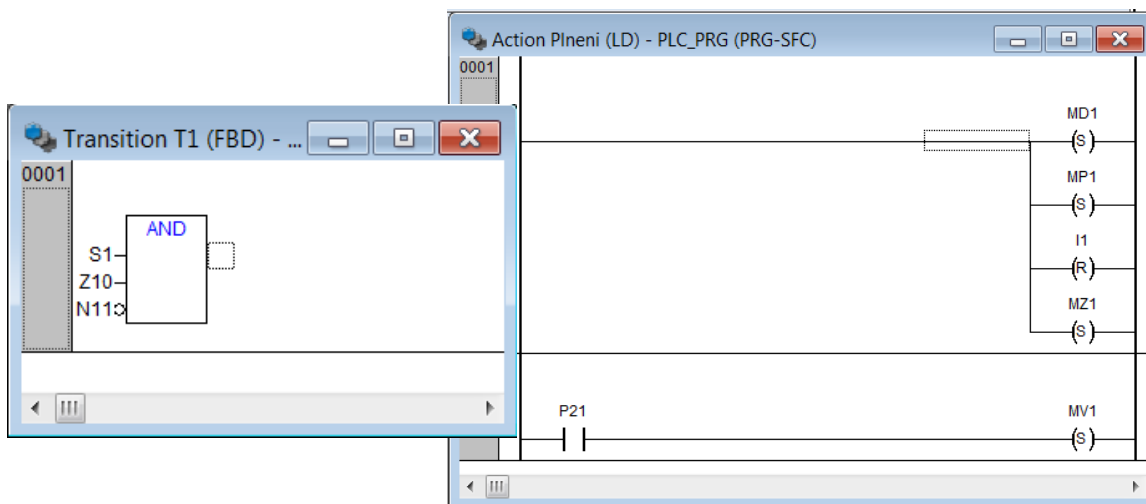
Pokud budeme vytvářet složitější struktury a budeme potřebovat více POU (není náš případ), kliknutím PT do prostoru záložce *POUs* vložíme novou POU tak jako při vytváření řídicího programu pro PLC.

Tímto způsobem dostaneme následující řídicí program pro linku 1:



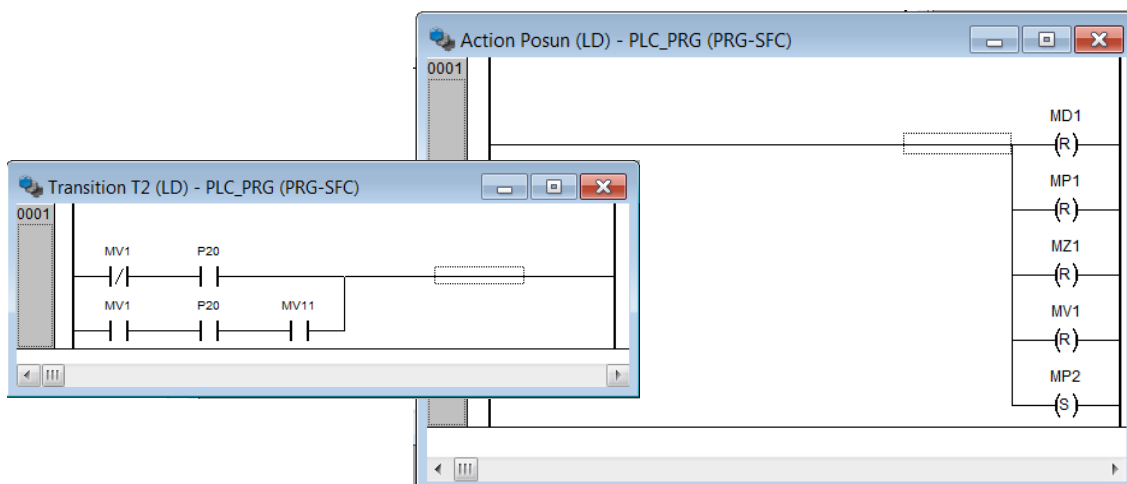
Obr. 4.2.8 Řídicí program linky 1, část 1; [s6]

V levé části je hlavní řídicí program v jazyce SFC, v pravé horní části výpis globálních proměnných s přiřazenými adresami. Dolní část zobrazuje detail prvních dvou kroků „Init“ a „CNS“ a první přechod T0 v jazycích LD a FBD.



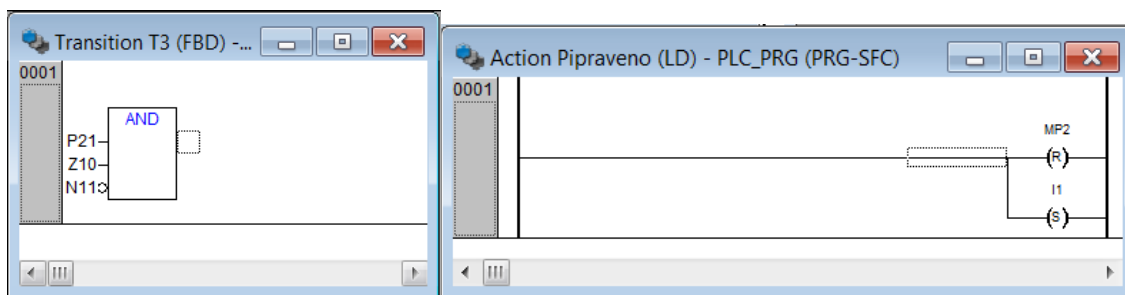
Obr. 4.2.9 Řídicí program linky 1, část 2; [s6]

V kroku „Plneni“ je realizován signál pro vysunutí válce 1 prostřednictvím podmínky přítomnosti lisované drtě pod jeho pístem. V přechodu T2 je pak výběr (logický součet po kroku č.2 v dříve uvedeném algoritmu Grafcet na str. 21) ze dvou větví pro situaci, kdy materiál pod pístem není (1. cyklus) a kdy je (2. cyklus a další). Tím je vyřešen správný sled operací linky při automatickém chodu ($A1 = 1$).



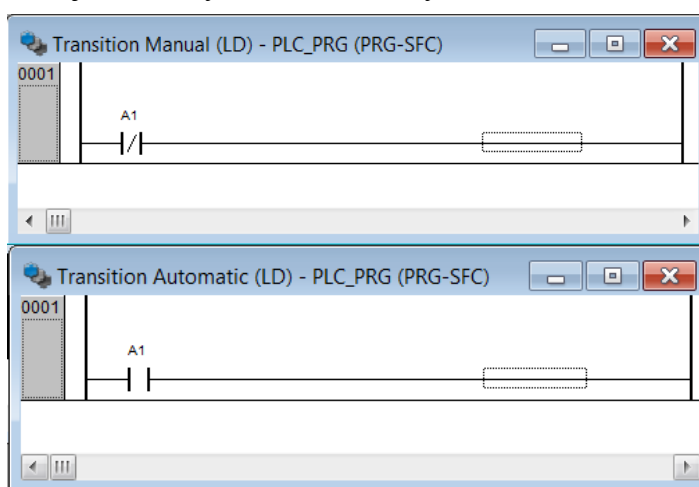
Obr. 4.2.10 Řídicí program linky 1, část 3; [s6]

V posledním kroku cyklu „Pripraveno“ je po splnění podmínek rozsvícena kontrolka linky KI1 signalizující připravenost linky pro další cyklus. KI1 je rozsvícena i při splnění podmínek při 1. spuštění linky (krok „CNS“). Zhasnuta je automaticky začátkem cyklu.



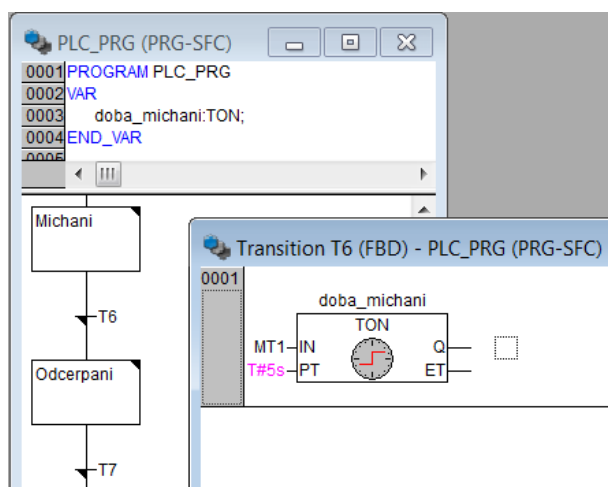
Obr. 4.2.11 Řídicí program linky 1, část 4; [s6]

Výběr „Automatic-Manual“ pak představuje podmínku (tlačítko KA1) pro skok buďto do kroku „CNS“, kdy program čeká na stisknutí tlačítka KS1, nebo do kroku „Plneni“, kdy bude zahájen další cyklus automaticky.



Obr. 4.2.12 Řídicí program linky 1, část 5; [s6]

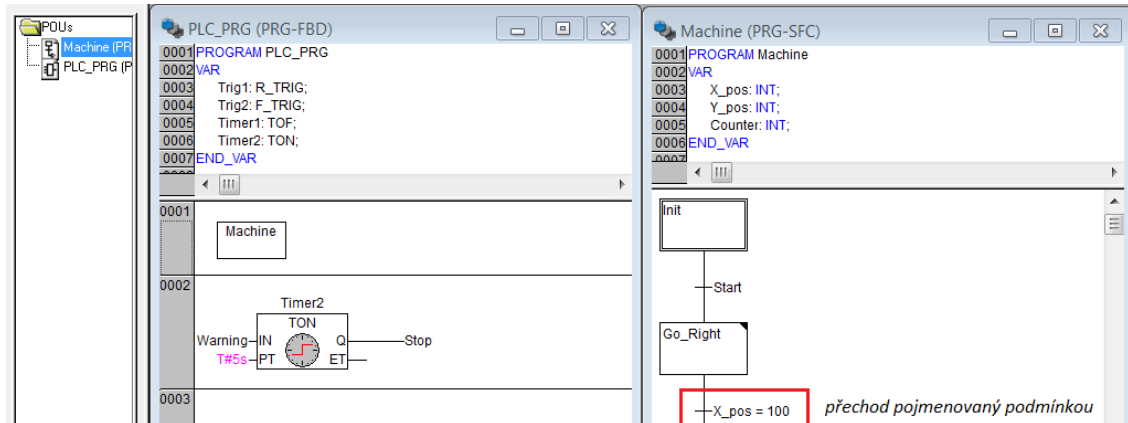
Obdobným způsobem pak budeme vytvářet i řídicí program pro linku 2. Pro přidání časovače typu TON po kroku „Michani“ v přechodu T6 jej musíme deklarovat v hlavním řídicím programu jako proměnnou typu TON:



Obr. 4.2.13 Časovač typu TON; [s6]

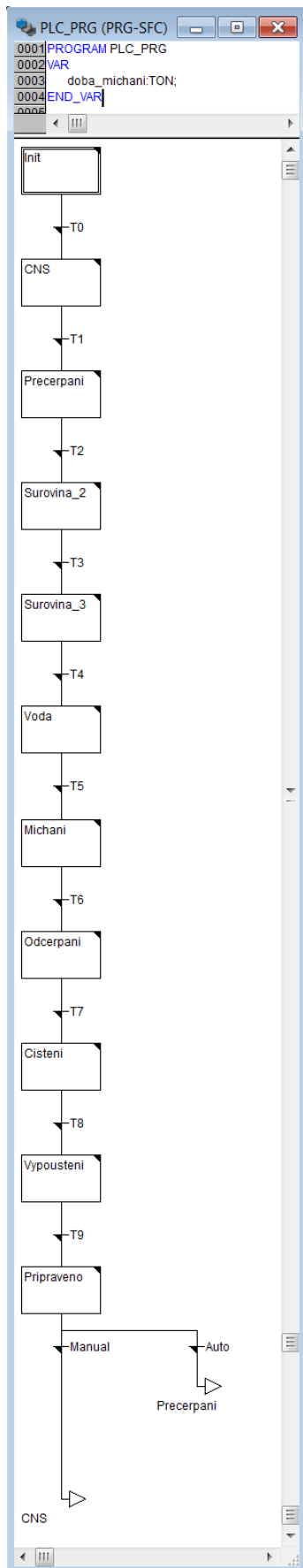
Globální proměnná MT1 aktivuje časovač, po uplynutí doby zadané v proměnné PT (=5s) se aktivuje výstup Q a přechod T6 proběhne.

Podobně jako časovač typu TON s názvem „doba_michani“ pak může vypadat i vnořený program do programu hlavního, který bude sestaven např. v jazyku FBD:



Obr. 4.2.14 Pojmenovaný přechod; [s6]

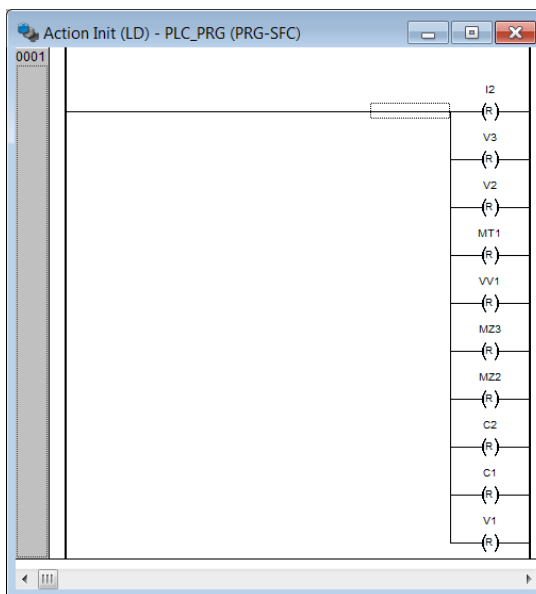
Navržený řídicí program pro linku 2 vypadá následovně:



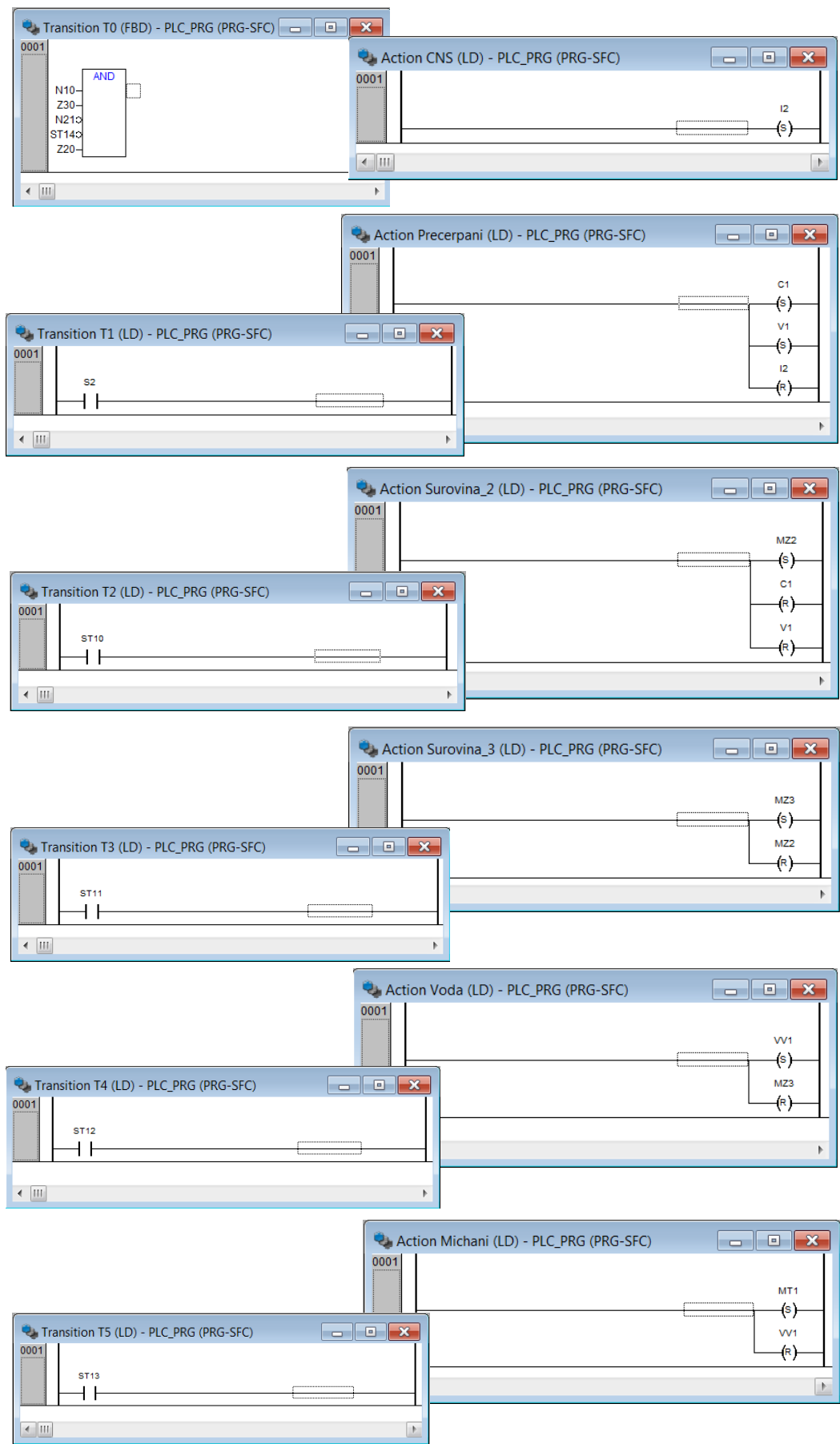
```

Global_Variables
0001 VAR_GLOBAL
0002 N10 AT %IX0.0:BOOL;
0003 N11 AT %IX0.1:BOOL;
0004 Z20 AT %IX0.2:BOOL;
0005 Z30 AT %IX0.3:BOOL;
0006 ST10 AT %IX0.4:BOOL;
0007 ST11 AT %IX0.5:BOOL;
0008 ST12 AT %IX0.6:BOOL;
0009 ST13 AT %IX0.7:BOOL;
0010 ST14 AT %IX0.8:BOOL;
0011 N20 AT %IX0.9:BOOL;
0012 N21 AT %IX1.0:BOOL;
0013 S2 AT %IX1.1:BOOL;
0014 A2 AT %IX1.2:BOOL;
0015
0016 V1 AT %QX0.0:BOOL;
0017 C1 AT %QX0.1:BOOL;
0018 C2 AT %QX0.2:BOOL;
0019 MZ2 AT %QX0.3:BOOL;
0020 MZ3 AT %QX0.4:BOOL;
0021 VV1 AT %QX0.5:BOOL;
0022 MT1 AT %QX0.6:BOOL;
0023 V2 AT %QX0.8:BOOL;
0024 V3 AT %QX0.9:BOOL;
0025 I2 AT %QX1.0:BOOL;
0026 END_VAR
0027

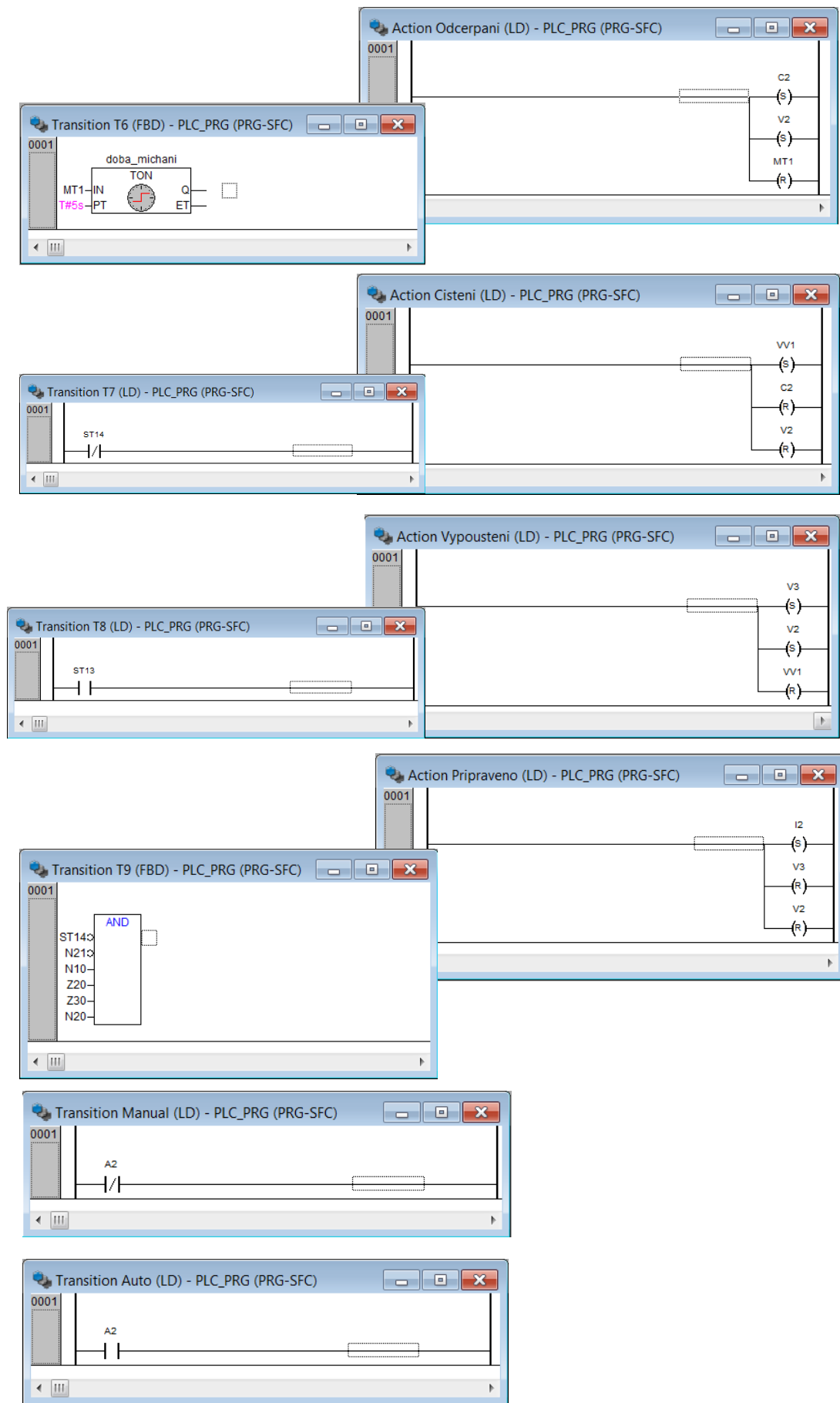
```



Obr. 4.2.15 Řídicí program linky 2, část 1; [s6]



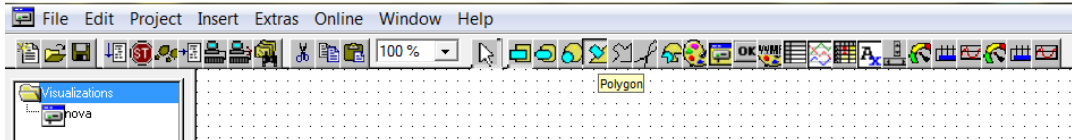
Obr. 4.2.16 Řídicí program linky 2, část 2; [s6]



Obr. 4.2.17 Řídicí program linky 2, část 3; [s6]

4.2.2 Vizualizace

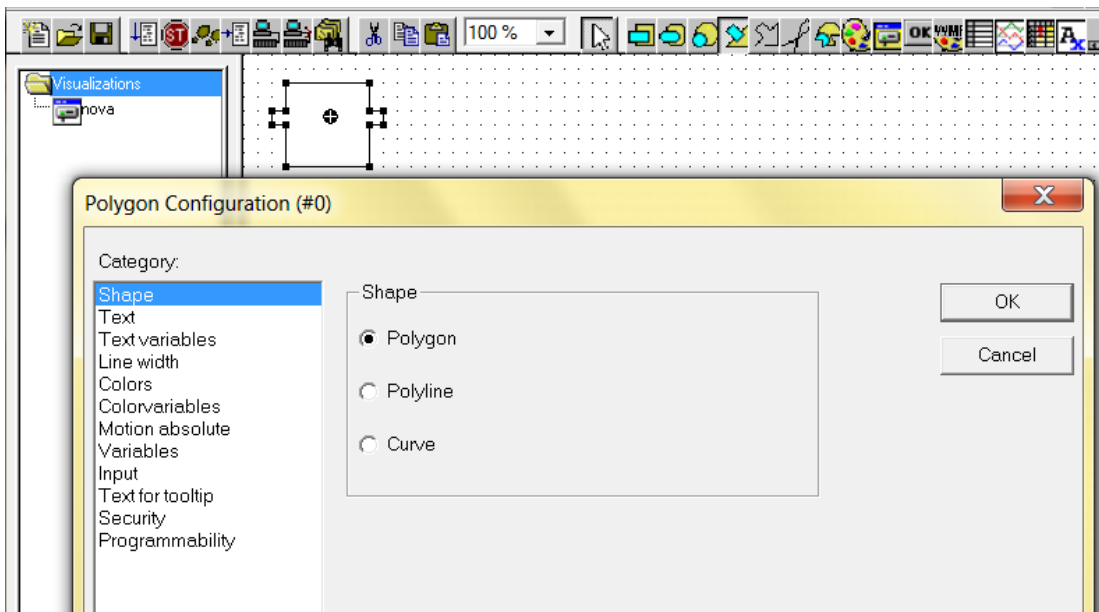
Když je řídicí program hotový, můžeme začít vytvářet vizualizaci, ve které je možno následně proces simulovat. K tomuto účelu slouží záložka *Visualisations* v levé dolní části obrazovky. Tak jako přidáváme novou POU v záložce *POUs*, klikem PT přidáme a pojmenujeme novou vizualizaci. Otevře se bílý pracovní prostor s bodovým rastrem, na který se vkládají prvky vizualizace. V tuto chvíli pro nás bude důležitý zejména panel prvků (konkrétně prvek *Polygon*, základní stavební prvek vizualizace).



Obr. 4.2.18 Panel prvků; [s6]

Pozn.: Při vytváření prvku se můžeme pohybovat pouze po rastru; pokud bude vizualizace složena z jednoduchých vytvářených prvků, nejmenší velikost prvku může být 1x1 mezera mezi body.

Po vytvoření tvaru daného prvku mu lze přiřadit funkci (2xLT). Podívejme se na možnosti prvku *Polygon*:



Obr. 4.2.19 Vlastnosti Polygon; [s6]

Zde je na výběr z několika nastavitelných parametrů:

Shape – jak je vytvořen tvar prvku

Text – obsahuje popisku prvku (*content*) a její umístění

Text variables – zde se lze odkázat na proměnnou v procesu, podle které se změní vlastnost textu při změně její hodnoty

Line width – parametry obrysu prvku

Colors – barevná výplň polygonu a barva jeho obrysu; počáteční (*Color*), při změně (*Alarm color*)

Colorvariables – proměnné pro barvy

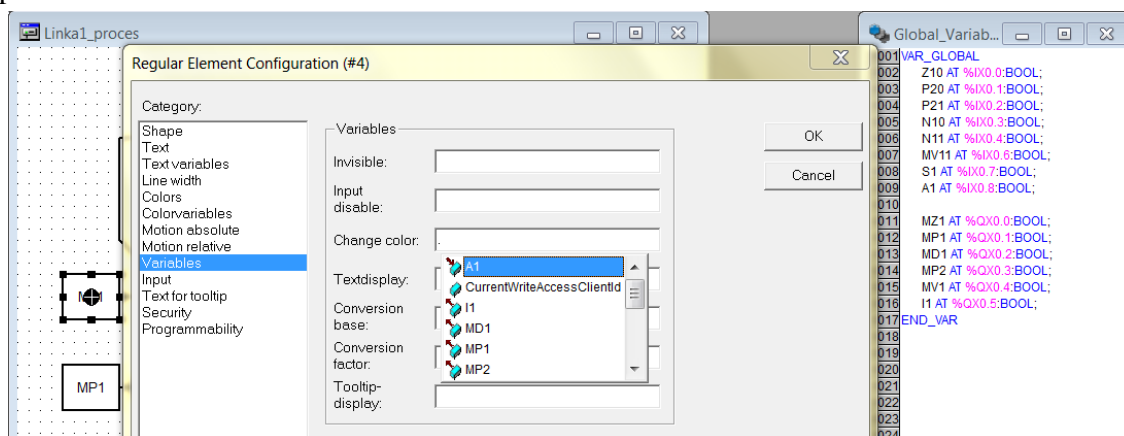
Motion absolute – parametry pro pohybující se prvek

Variables – zde se lze odkázat na proměnné v procesu, jejichž změna bude mít přímou vazbu na změnu samotného prvku (zneviditelnění, změna barvy, zobrazení textu apod.); nastavení, které budeme využívat celkem často

Input – funkce pro tlačítka; *Toggle* – aktivní po stisknutí, *Tap* – aktivní po dobu stisku; obsahuje zároveň i pokročilé volby jako je např. spouštění programu, skok na jinou vizualizaci apod.

Text for tooltip, *Security*, *Programmability* – pokročilé volby pro naprogramování prvku a nastavení oprávnění

Při odkazování na proměnné vepíšeme do požadovaného pole název POU, ve které máme proměnnou deklarovanou a pak znak „.“, případně jen onu tečku pro proměnnou globální. Program sám dá na výběr, kterou proměnnou chceme prvku přiřadit:



Obr. 4.2.20 Přiřazení proměnné; [s6]

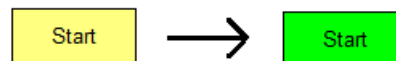
Po nastavení a potvrzení parametrů získáme již funkční prvek.

Příklady prvků a jejich konfigurace:

Tlačítko Start s funkcí tlačítka (I=zelená) měnícím hodnotu proměnné S1

Shape: Rectangle

Text:content: Z10



Colors: Color: inside: žlutá, frame: černá

Alarm color: inside zelená, frame: černá

Variables: Change color: .S1 (glob. proměnná)

Input: Tap variable: .S1

Motor pásu 1 mění barvu aktivací proměnné MP1 (1=zelená)

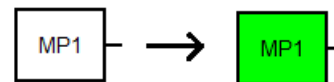
Shape: Rectangle

Text: MP1

Colors: Color: inside: bílá, frame: černá

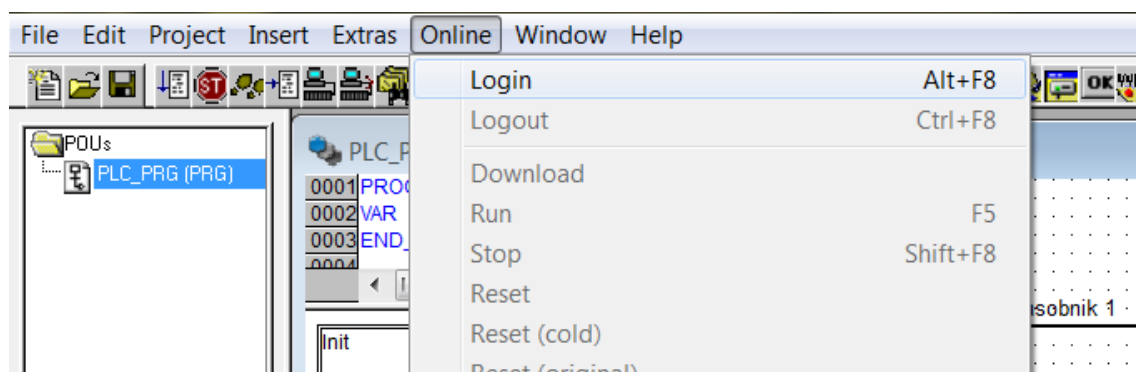
Alarm color: inside zelená, frame: černá

Variables: Change color: .MP1 (glob. proměnná)



Panel nástrojů vizualizace dále obsahuje množství různých dalších prvků jako jsou diagramy, hodiny apod.

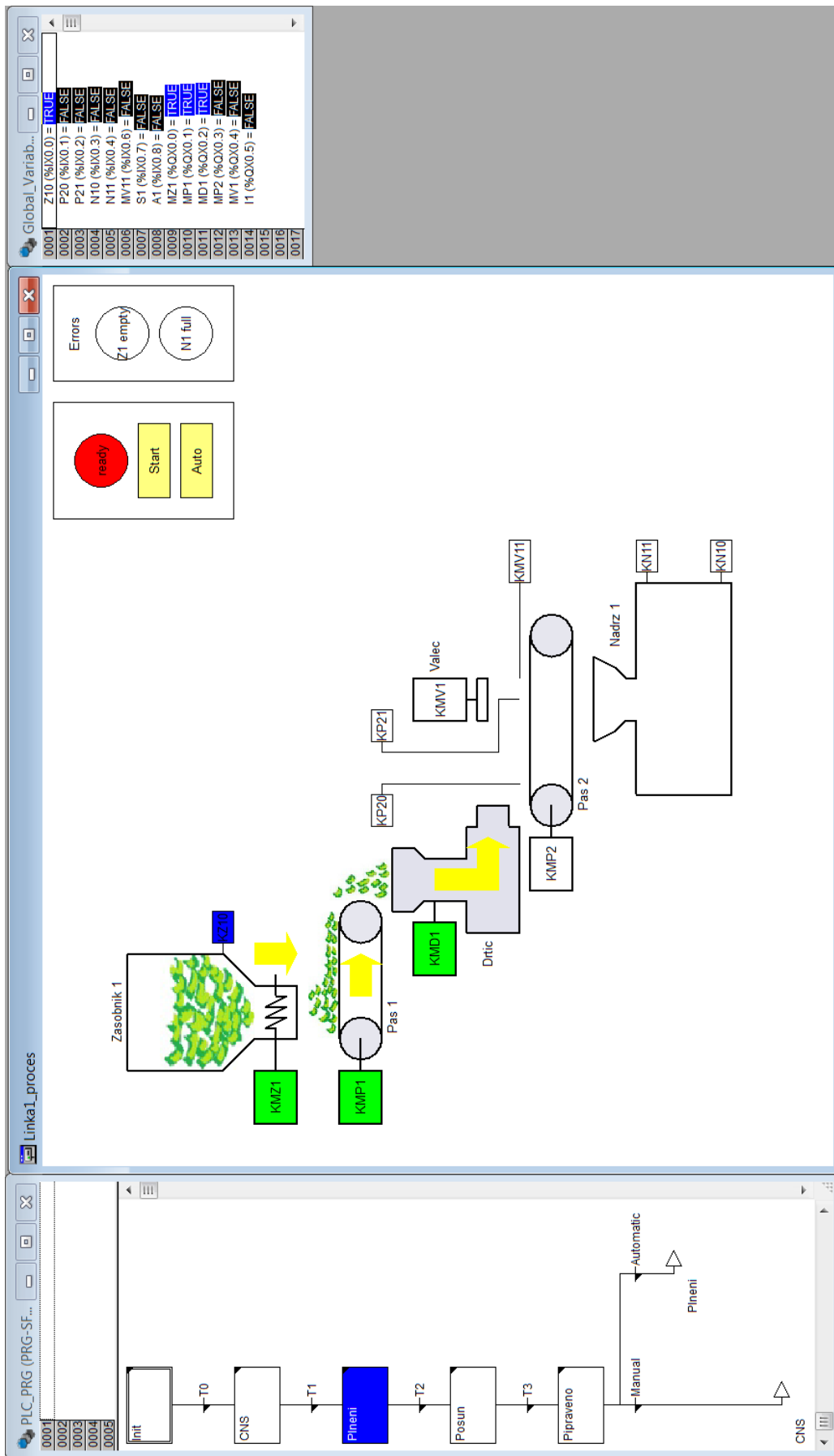
Po vytvoření vizualizace je program připravený ke spuštění. Najdeme příkaz *Login* v menu *Online* v horní liště programu (případné chyby jsou ohlášeny v *message window*) a poté příkazem *Run* program spustíme.



Obr. 4.2.21 Spuštění vizualizace; [s6]

A pak už lze pozorovat chování soustavy v závislosti na zásazích operátora. Na levé straně se nachází hlavní program v SFC s barevným vyznačením aktivního kroku a přechodu (obojí lze dále podrobněji sledovat). V prostřední části obrazovky je vizualizační schéma a v pravém horním rohu je jednoduchý operátorský panel. V pravém okně je seznam globálních proměnných s jejich aktuálními hodnotami. Pozn.: Tímto způsobem se spouští program s konfigurací HW, která ovšem není v tuto chvíli ještě vytvořena; simulace s virtuálním PLC se spouští výběrem *Simulation Mode* ve stejné záložce

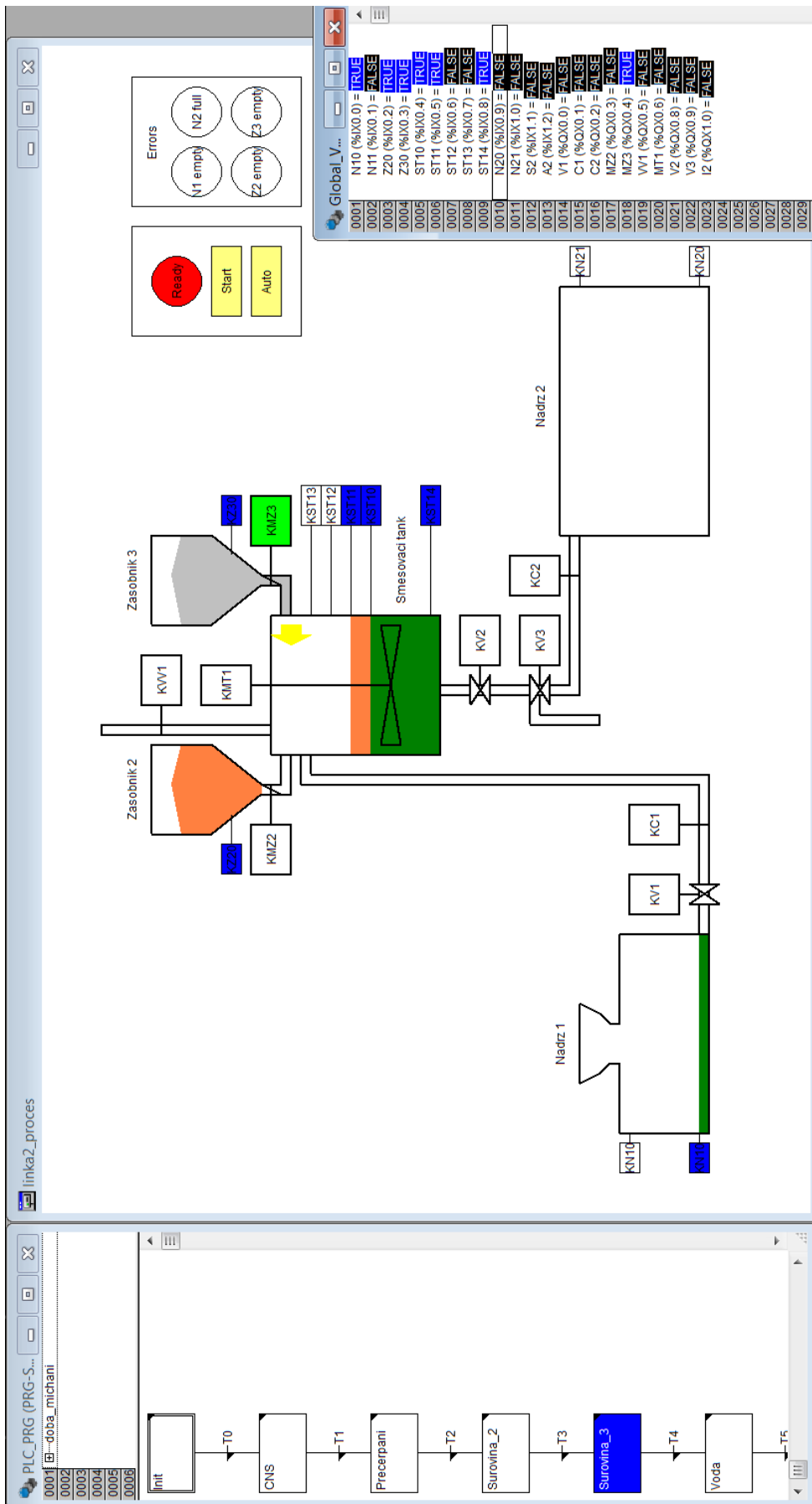
Simulace linky 1 v kroku „Plnění“ 1.cyklu vypadá následovně:



Obr. 4.2.22 Linka 1 vizualizace; [s6]

Schéma vizualizace bylo vytvořeno takto: šipky jsou tvořeny polygony přesunutými do popředí a viditelnými jen při aktivaci příslušné proměnné a prvku (př. motoru KMP1), surovina ze zásobníku 1 je vložena ručně kreslená bitmapa. Tyto prvky jsou překryty polygonem stejného rozměru, který má definovanou základní neprůsvitnou barvu (=bílou) a při změně proměnné se jeho barva změní na zcela průhlednou (=žádnou), takže prvek pod ním (šipka) je zcela viditelná. Jistě existuje elegantnější řešení (např. v programu pomocnou proměnnou hodnoty negace proměnné, která bude prvek řídit), ale tento způsob je celkem rychlý a jednoduchý.

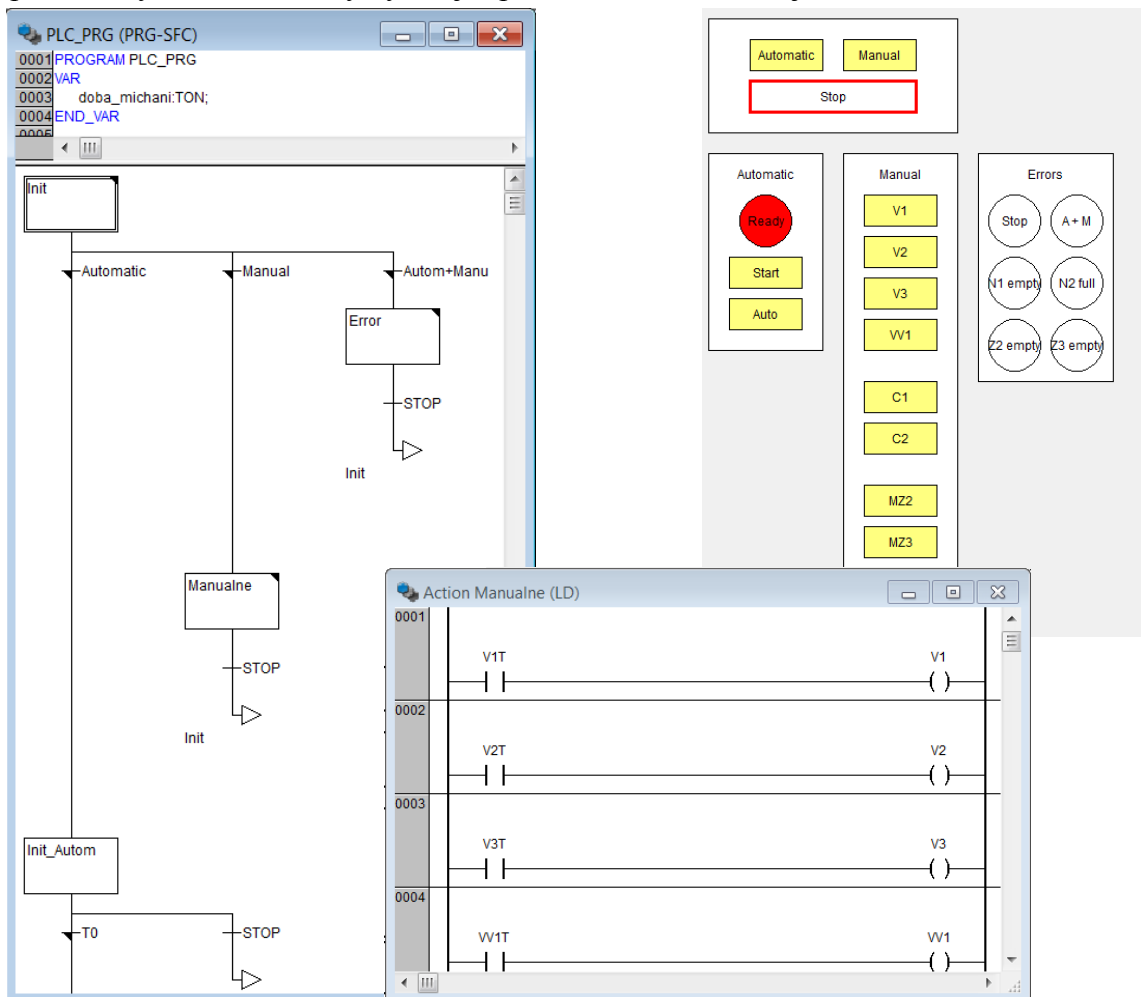
A stejným způsobem se sestrojí vizualizace pro linku 2. Potrubí, která vedou mezi nádržemi a tankem, mění barvu podle toho, jaká surovina jimi zrovna proudí, a tvoří je vlastně polygony tvaru „L“ se změnou barvy podle změny proměnné.



Obr. 4.2.23 Linka 2 vizualizace; [s6]

4.2.3 Rozšíření funkcí operátorského panelu

K běžným funkcím operátorského panelu patří rovněž plně manuální ovládání všech akčních členů. Ovládací algoritmus byl tedy upraven tak, že prvním krokem po zapnutí samotného systému je výběr mezi ovládáním ručním a automatickým, jenž obsluha potvrzuje příslušným tlačítkem. Po stisku tlačítka „Automatic“ pro automatický chod se spustí výše uvedená sekvence řízení jednoho chodu linky nebo cyklické řízení. Při stisku tlačítka „Manual“ je možno pomocí tlačítek ovládat jednotlivé akční prvky systému. Pro ukončení režimu „Manual“ i přerušení činnosti v „Automatic“ slouží vypínač „Stop“, který navrátí systém do kroku výběru mezi režimy. Slouží tak zároveň i jako okamžité vypnutí. Tato akce je realizována alternativní přechodovou větví se skokem na počáteční krok („Init“) v každém dalším kroku SFC diagramu obou linek. Pozn.: tlačítko „Stop“ je typu *toggle* (=klopné) a jeho odblokování je podmínkou pro znovuspuštění procesu; rovněž i tlačítka „Automatic“ a „Manual“ jsou tohoto typu, přičemž výběr mezi režimy vylučuje spuštění obou režimů v jednom okamžiku



Obr. 4.2.24 Upravený algoritmus a operátorský panel linky 2; [s6]

4.3 HW - řídicí jednotka CPX-CEC a SW - CoDeSys

4.3.1 Charakteristika jednotky CPX-CEC

K technické realizaci terminálu CPX se 2 moduly bylo využito řídicí jednotky CPX-CEC s přípojovacím modulem CPX AB-8-KL-4POL.

Základní funkce:

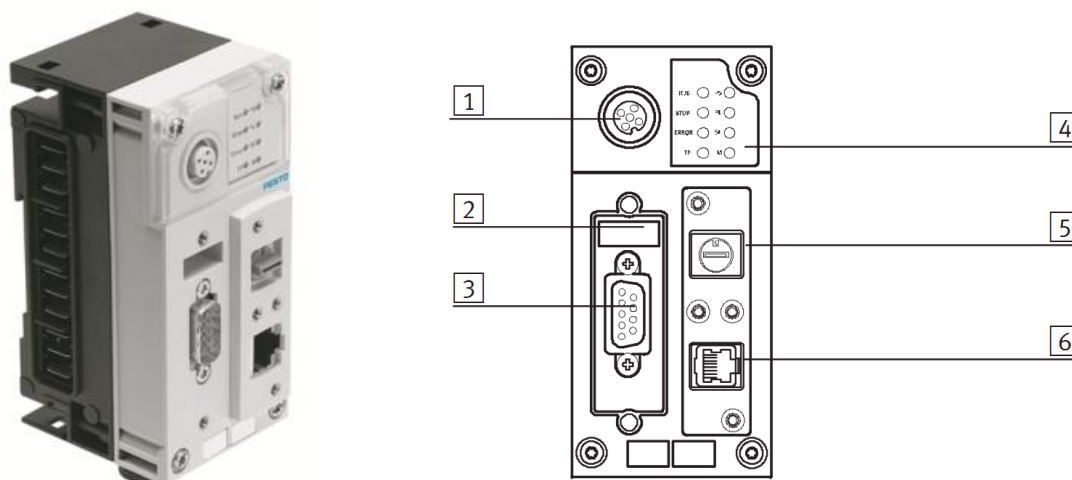
- programovatelné v CoDeSys (IEC 61131-1)
- komunikace přes Ethernet (Modbus/TCP, EasyIP, TCP/IP)
- možnost připojení dalších modulů (např. FED či modulů pro připojení až 512-ti I/O)
- rozhraní RS232 pro komunikaci s externími zařízeními

Hlavní technické údaje CPX-CEC:

Komunikační protokol	CoDeSys level 2, EasyIP, ModbusTCP, TCP/IP
CPU data	32 MB RAM, 32 MB flash, 400 MHz processor
Doba zpracování	200 µs/1000 instrukcí
Přenosová rychlost	10/100 bps při IEEE 802.3 (10BaseT) nebo 802.3u (100BaseTx)
Programovací software	Festo CoDeSys
Programovací jazyk	SFC, IL, FBD, LD, ST (IEC 61131-3), přídatný CFC
Jazyk komunikace	německy, anglicky
Paměť programu	4 MB uživatelský program
Flags	30 kB remanentní paměť, 8 MB glob. data, CoDeSys systém proměnných
Diagnostiky zařízení	paměťová, kanálová a modulově zaměřená, podpěťová a zkratová modulová
LED indikace	TP: Link/traffic RUN: PLC status STOP: PLC status ERR: Runtime error PS: elektronický a senzorový přívod PL: přívod signálu SF: systémová chyba M: editace/zesílení aktivní
Parametrizace	CoDeSys
Podpora konfigurace	CoDeSys
Nastavení IP adresy	DHCP, CoDeSys, MMI
Ovládací prvky	otočný přepínač RUN/STOP
Přídavné funkce	Komunikační funkce RS232
Jmenovité pracovní napětí	24 V DC
Jmenovité pracovní napětí silového napájení	24 V DC 18–30 V DC (bez pneumatiky) 21,6–26,4 V DC (pneumatika typu midi/maxi) 20,4–26,4 V DC (pneumatika typu CPA) 18–30 V DC (pneumatika typu MPA)
Přemostění výpadku energie	10 ms

Příkon při 24 V DC	85 mA
Třída ochrany	IP65, IP67
Rozměry š x d x v	50 x 107 x 55 mm
Hmotnost	155 g
Teplota okolí/skladovací	-5 až +50 °C/-20 až +70 °C
Relativní vzdušná vlhkost	95 % (bez kondenzace)
Třída odolnosti proti korozi dle normy Festo 940 070	2

Tab. 3 Technické údaje; [6]



- 1 – CPX-MMI připojení
- 2 – DIL přepínač
- 3 – rozhraní RS232 (socket, Sub-D, 9-pin)
- 4 – LED indikace
- 5 – RUN/STOP přepínač
- 6 – rozhraní Ethernet (RJ45, socket, 8-pin)

Obr. 4.3.1 Jednotka CPX-CEC; [6]

Webový monitor CPX:

- online diagnostika systému a stavů podle modulů a kanálů s alarmy prostřednictvím e-mailu nebo SMS, ruční nastavování výstupů, trasování chyb (sledování analogových hodnot, hlášky pomocí prostého textu o povaze chyby pro efektivní zásah)
- primárně určen pro údržbu a kontrolu na dálku (web-serveru je přiřazena IP adresa, se zařízením lze komunikovat prostřednictvím PC a dalších inteligentních zobrazovacích a obslužných zařízení)
- požadavky: PC kompatibilní s IBM; OS MS-Windows 98, ME, 2000, XP; IE 5.5+, Mozilla Firefox 1.0; Java Runtime Environment 1.3+

Možnosti připojení funkčních modulů k CPX AB-8-KL-4POL:

- digitální vstupní moduly: CPX 4DE, CPX 8DE, CPX 8DE-D, CPX 8NDE
- digitální vstupní moduly a kombinované moduly vstupy/výstupy: CPX 4DA, CPX 8DA, CPX 8DA-H, CPX 8DE-8DA
- analogové funkční moduly se vstupy/výstupy: CPX 2AE-U-I, CPX 4AE-I, CPX 4AE-T, CPX 4AE-TC, CPX 2AA-U-I



Obr. 4.3.2 Kryt modulu CPX AB-8-KL-4POL; [7]

kryty s připojením vstupů/výstupů	CPX-8DE-8DA	
	X1.0: 24 V _{SEN}	X5.0: Output x+4
	X1.1: 0 V _{SEN}	X5.1: 0 V _{OUT}
	X1.2: Input x	X5.2: Output x
	X1.3: FE	X5.3: FE
	X2.0: Input x+4	X6.0: Output x+5
	X2.1: Input x+5	X6.1: 0 V _{OUT}
	X2.2: Input x+1	X6.2: Output x+1
	X2.3: FE	X6.3: FE
	X3.0: 24 V _{SEN}	X7.0: Output x+6
	X3.1: 0 V _{SEN}	X7.1: 0 V _{OUT}
	X3.2: Input x+2	X7.2: Output x+2
	X3.3: FE	X7.3: FE
	X4.0: Input x+6	X8.0: Output x+7
	X4.1: Input x+7	X8.1: 0 V _{OUT}
	X4.2: Input x+3	X8.2: Output x+3
	X4.3: FE	X8.3: FE

Obr. 4.3.3 Zapojení vstupně-výstupního modulu; [7]

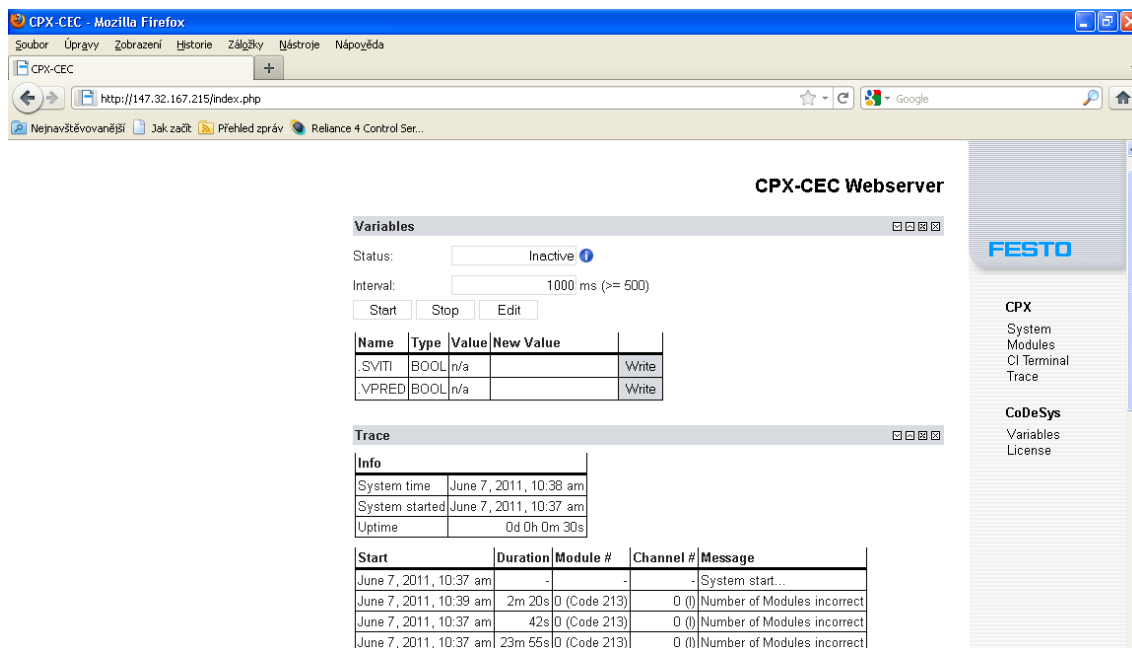
4.3.2 Technická realizace řízení

Nezbytnou součástí simulace a zároveň jeden z cílů práce je technická realizace řízení modelové soustavy pomocí reálného PLC. Text zde napsaný může posloužit jako první seznámení a návod na zapojení samotné jednotky CPX-CEC a řízení technologického procesu pomocí PLC prostřednictvím uživatelského programu vytvořeného v SW prostředí CoDeSys.

Po sestavení terminálu CPX a připojení napájecího a síťového kabelu zapneme PLC pomocí RUN/STOP přepínačem do libovolné polohy, která není výchozí (poloha

0). Tyto polohy lze dle fy Festo údajně i využít v řídicím programu, v úloze však toto realizováno nebylo a na poloze, která byla nastavena, nezáleží. Kryt přepínače můžeme opět upevnit na své místo, PLC bude zapínáno a vypínáno napájením.

Zapneme zdroj terminálu a zkontrolujeme síťovou komunikaci na IP adrese jednotky CPX-CEC. Naše IP adresa byla: 147.32.167.215 a po jejím zadání do adresového řádku v internetovém prohlížeči jsme ověřili aktivní status webserveru.



Obr. 4.3.4 Webserver jednotky CPX-CEC

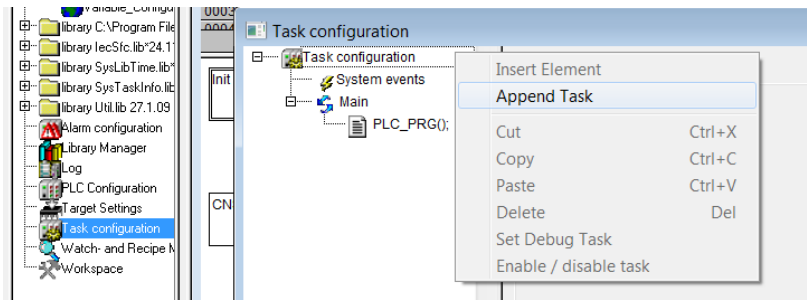
Webserver může hlásit 3 stavy. Zde jejich popis a řešení:

„*Not running*“ znamená, že není nastavena konfigurace připojeného PLC v prostředí CoDeSys. Pokud tak nebylo učiněno hned na začátku při vytvoření nového projektu, lze nastavit konfiguraci v záložce *Resources->Target Settings (2xLT)*. Pro možnost výběru konfigurace je nutno importovat knihovnu „Festo_CPX.lib“ v *Library Manageru (2xLT)* ve stejné záložce. Knihovna se nalézá v adresáři:

(umístění instalace)/Festo/CoDeSys/Targets/Festo/CPX-CEC/Library/

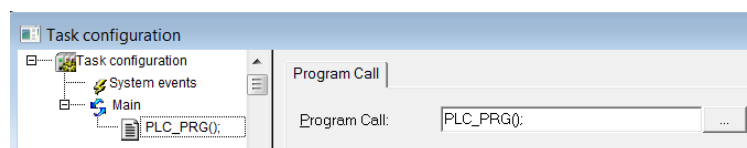
Na místě *Resources->Target Settings->General* je nutno zaškrtnout volbu „Download symbol file“; to je soubor, prostřednictvím kterého komunikuje webserver se SW prostředím CoDeSys.

Dále je třeba nastavit automatické úlohy v *Resources->Task configuration (PT na Task Configuration)*:



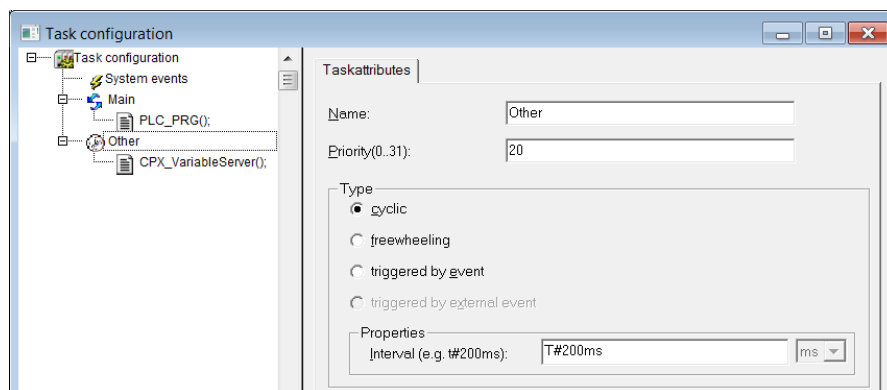
Obr. 4.3.5 Nastavení úloh; [s6]

Úlohy budou dvě; název první je „Main“, její typ je *freewheeling* a bude spouštět hlavní program „PRG_SFC“. Program se přidává podobným způsobem jako úloha (PT na „Main“) a jeho zápis je znázorněn na obrázku:



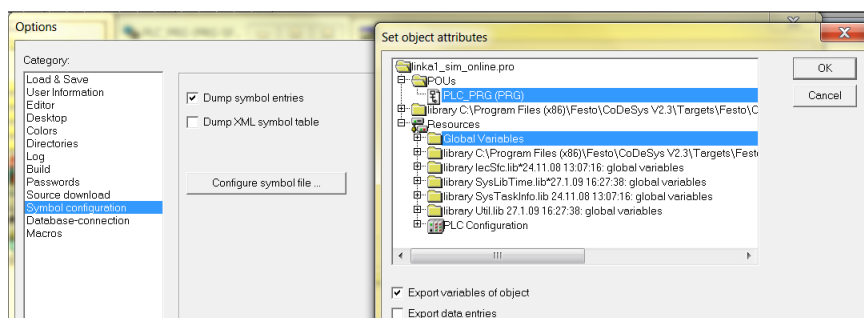
Obr. 4.3.6 Volání programu; [s6]

Druhá úloha nese název „Other“, její typ je *cyclic* (interval např. 200 ms) a bude spouštět program „CPX_VariableServer();“ pro příjem a odesílání aktualizovaných hodnot proměnných mezi webserverem a prostředím CoDeSys.



Obr. 4.3.7 Nastavení úlohy Other; [s6]

Posledním krokem je nastavení *Symbol file* v záložce hlavní horní lišty *Project-Options* tak, jak je vidět na následujícím obrázku:



Obr. 4.3.8 Symbol file;[s6]

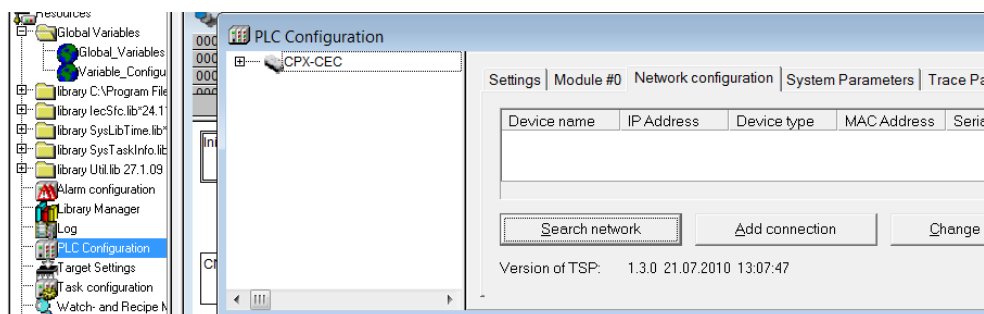
„*Inactive*“ je označení stavu, kdy je nastavení webserveru a komunikace v pořádku, ale žádné hodnoty nejsou na webserver uploadovány. Tento problém se odstraní spuštěním programu.

„*Active*“ je požadovaný normální provozní stav.

Jakmile je komunikace nastavena, lze na webserveru a v SW CoDeSys sledovat stav proměnných. Během nastavení anebo při komunikaci samotné se mohou vyskytnout chyby. Zde jsou příklady těch nejčastějších:

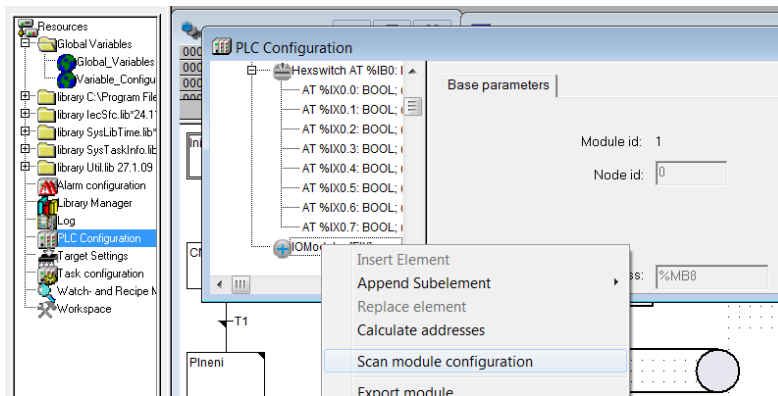
„*No variables are configured*“ – (původ: webserver) není nastaveno sledování proměnných; odstraní se tlačítkem *Edit*, po zvolení proměnných např. pomocí filtru se potvrdí tlačítkem *Apply*

„*Communication Error (#0): Logout Performance*“ – není nastavena správně komunikace; ve většině případů stačí restartovat PLC (stává se po změně programu) nebo je třeba nechat vyhledat PLC v záložce *Resources->PLC Configuration->Network Configuration->Search network*. Pokud není schopen SW najít sám připojené PLC, je možno jeho adresu ručně zadat pomocí *Add connection*.



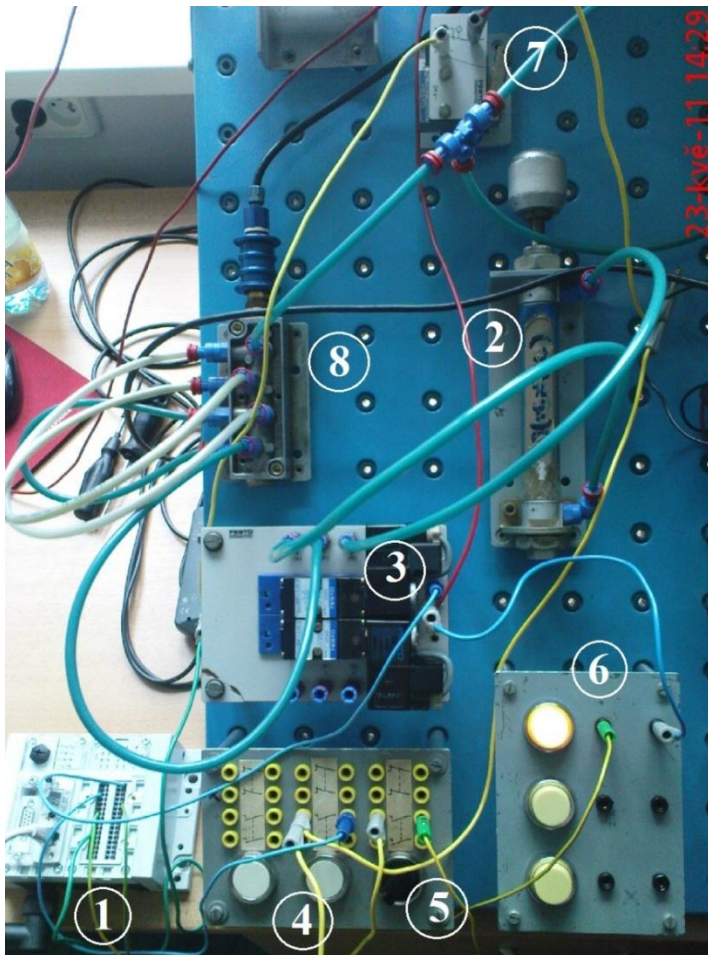
Obr. 4.3.9 Konfigurace PLC; [s6]

„*#4096: module: ,Code=4' is found but is not a member of the plc configuration*“ – tato hláška znamená, že systém rozpoznal PLC i vstupně-výstupní modul, ale neví, že operují spolu; webserver hlásí: „*Number of modules incorrect*“; problém je třeba vyřešit v záložce *Resources->PLC Configuration->I/O Modules[FIX] (PT)->Scan module configuration*. Program by pak měl být schopen sám rozpoznat modul a propojit jej s PLC:



Obr. 4.3.10 Vyhledávání modulu; [s6]

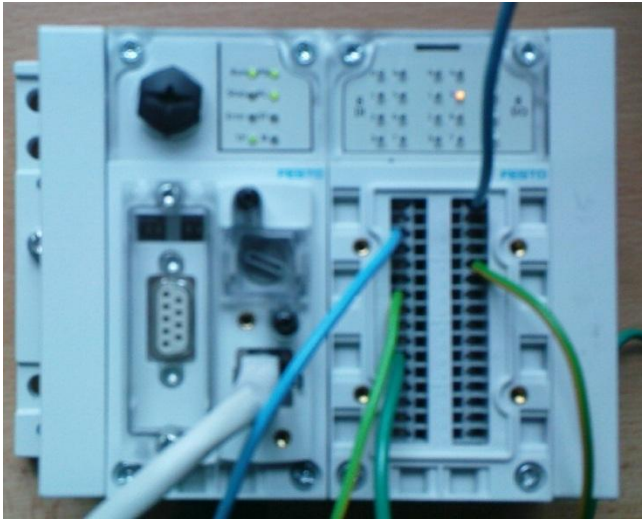
Jakmile jsou všechny chyby ošetřeny, je možno začít fyzicky propojovat PLC s akčními prvky. K tomuto účelu byla k dispozici stavebnice pro elektropneumatické úlohy Festo Didactic v laboratoři č.109.



- 1 – terminál CPX
- 2 – válec KMV1
- 3 – rozvaděč KVV1
- 4 – tlačítko „Start“
- 5 – tlačítko „Auto“
- 6 – kontrolka „Ready“
- 7 – koncový senzor KMV11
- 8 – přívod vzduchu

Obr. 4.3.11 Simulace linky 1 s reálným PLC

Samotné PLC bylo zapojeno dle manuálu (viz. Obr. 4.3.3 Zapojení vstupně-výstupního modulu; [7]). V hlavním řídicím programu bylo ještě třeba upravit adresy globálních proměnných tak, aby vstupy začínaly číslicí 2 (2.0,2.1,...). Adresy výstupů se nezměnily.



X1.2: S1 AT %IX2.0
X2.2: A1 AT %IX2.1
X3.2: MV11 AT %IX2.2
X5.0: MV1 AT %QX0.4
X6.0: I1 AT %QX0.5

Obr. 4.3.12 Reálné zapojení PLC pro linku 1

Simulace s reálným PLC proběhla pro linku 1.

5 Závěr

Pro zadanou modelovou soustavu technologického procesu byla vytvořena analýza pomocí Petriho sítí v programu SIPN Editor, mezi jehož přednosti patří dostupnost, jednoduchost a schopnost přeložit kód pro další možné použití. Návrh algoritmu a obvodového zapojení byl vytvořen v SW FluidSIM (verze 4.0).

Řídicí program pro jednotku CPX-CEC byl zkonstruován pro obě části modelové soustavy v SW prostředí CoDeSys (verze 2.3.9.19), které umožnilo kromě samotného vytvoření hlavního řídicího programu v jazyku SFC a jeho částí v jazycích LD a FBD i vytvoření vizualizace procesu a jeho simulace pro virtuální PLC. Simulace pro PLC reálné proběhla úspěšně pro linku 1, pro linku 2 jen na virtuálním PLC kvůli nedostačující kapacitě vstupně-výstupního modulu. Pro reálné zapojení byla k dispozici stavebnice Festo Didactic v laboratoři č.109.

Celý proces realizace byl podrobně zaznamenán a sepsán v této bakalářské práci, která tak může sloužit jako návod pro obsluhu a práci s tímto novým hardwarovým a softwarovým vybavením produktů firmy Festo – CPX-CEC a CoDeSys.

Seznam zdrojů

Literatura

- [1] MARTINÁSKOVÁ, M., ŠMEJKAL, L.: Řízení programovatelnými automaty I, skriptum ČVUT FS. Praha, 2004
- [2] MARTINÁSKOVÁ, M., ŠMEJKAL, L.: Řízení programovatelnými automaty II, skriptum ČVUT FS. Praha, 2000
- [3] MARTINÁSKOVÁ, M., ŠMEJKAL, L.: Řízení programovatelnými automaty III, skriptum ČVUT FS. Praha, 2003
- [4] HANZÁLEK, Z.: Když se řekne...Petriho síť. Automatizace 9., 2001
- [5] HOFSSAES, C.: The French Society of Computer Scientists: AFCET. Annals of the History of Computing 12., 1990
- [6] katalog produktu CPX-CEC fy Festo, 2009
- [7] katalog pluginu fy Festo, 2009

Software

- [s1] CPN Tools
- [s2] CESim
- [s3] Netlab
- [s4] PIPE V2.0
- [s5] SIPN Editor
- [s6] CoDeSys V2.3.9.19
- [s7] FluidSIM V4.0

Internetové zdroje

- [i1] Webové stránky CPNTools: <http://cpntools.org/>
- [i2] Webové stránky CESim: <http://www.fit.vutbr.cz/~novosad/CESim/>
- [i3] Webové stránky Pesim:
<http://www.fit.vutbr.cz/study/courses/PES/public/Pomucky/pesim.zip>

- [i4] Webové stránky Netlab: <http://www.irt.rwth-aachen.de/index.php?id=101>
- [i5] Webové stránky PIPE: <http://pipe2.sourceforge.net/>
- [i6] Webové stránky INA: <http://www2.informatik.hu-berlin.de/~starke/ina.html>
- [i7] Webové stránky PNtalk: <http://perchta.fit.vutbr.cz:8000/projekty/12>
- [i8] Webové stránky SIPN Editor: <http://www.eit.uni-kl.de/litz/ENGLISH/software/SIPNEditor.htm>
- [i9] Webové stránky firmy Festo: www.festo.com
- [i10] Webové stránky OpenPCS: <http://www.systec-electronic.com/html/imdex.pl/>

Seznam obrázků

Obr. 1.2.1 Prvky Petriho sítě.....	2
Obr. 1.2.2 C/E sítě před přechodem.....	2
Obr. 1.2.3 C/E sítě po přechodu.....	2
Obr. 1.2.4 P/T sítě před přechodem	3
Obr. 1.2.5 P/T sítě po přechodu	3
Obr. 1.2.6 Inhibiční hrana	3
Obr. 1.3.1 Prvky Grafcet.....	4
Obr. 1.3.2 Paralelismus	5
Obr. 1.3.3 Synchronizace	5
Obr. 1.3.4 Výběr.....	5
Obr. 1.3.5 Spojení	5
Obr. 1.3.6 Konstrukce přechodů	6
Obr. 1.3.7 Akce	6
Obr. 2.1 Obrazovka SW prostředí CPN Tools; [s1].....	8
Obr. 2.2 Příklad v SW prostředí CESim; [s2].....	9
Obr. 2.3 Obrazovka SW prostředí Netlab; [s3].....	10
Obr. 2.4 Aplikace v PIPE 2.0; [s4].....	11
Obr. 2.5 Aplikace v PNTalk; [i7]	12
Obr. 2.6 Aplikace v SIPN; [i8].....	13
Obr. 3.1.1 Zadaná technologická soustava.....	15
Obr. 3.3.1 Schéma technologické soustavy s akčními členy a senzory; [s6].....	17
Obr. 3.3.2 Instalace SIPN.....	20
Obr. 3.3.3 Obrazovka v SIPN; [s5]	20
Obr. 3.3.4 Panel nástrojů v SIPN; [s5].....	21
Obr. 3.3.5 Obrazovka prostředí OpenPCS; [i10]	21
Obr. 3.3.6 Linka 1 Petriho sítě; [s5].....	22
Obr. 3.3.7 Linka 2 Petriho sítě; [s5].....	22
Obr. 3.3.8 Elektropneumatické a obvodové schéma linky 1; [s7]	23
Obr. 3.3.9 Obvodové schéma linky 2; [s7]	24
Obr. 3.3.10 Linka 1 grafcet; [s7].....	25
Obr. 3.3.11 Linka 2 grafcet; [s7]	25
Obr. 4.2.1 Nová POU; [s6].....	27

Obr. 4.2.2 Záložky; [s6]	27
Obr. 4.2.3 Globální proměnné; [s6]	27
Obr. 4.2.4 Hlavička POU; [s6].....	28
Obr. 4.2.5 Vytvoření nového přechodu; [s6]	28
Obr. 4.2.6 Přidávání knihoven; [s6].....	29
Obr. 4.2.7 Deklarace proměnných; [s6]	29
Obr. 4.2.8 Řídicí program linky 1, část 1; [s6]	30
Obr. 4.2.9 Řídicí program linky 1, část 2; [s6]	31
Obr. 4.2.10 Řídicí program linky 1, část 3; [s6]	31
Obr. 4.2.11 Řídicí program linky 1, část 4; [s6]	32
Obr. 4.2.12 Řídicí program linky 1, část 5; [s6]	32
Obr. 4.2.13 Časovač typu TON; [s6]	32
Obr. 4.2.14 Pojmenovaný přechod; [s6]	33
Obr. 4.2.15 Řídicí program linky 2, část 1; [s6]	34
Obr. 4.2.16 Řídicí program linky 2, část 2; [s6]	35
Obr. 4.2.17 Řídicí program linky 2, část 3; [s6]	36
Obr. 4.2.18 Panel prvků; [s6]	37
Obr. 4.2.19 Vlastnosti Polygon; [s6].....	37
Obr. 4.2.20 Přiřazení proměnné; [s6].....	38
Obr. 4.2.21 Spuštění vizualizace; [s6]	39
Obr. 4.2.22 Linka 1 vizualizace; [s6].....	40
Obr. 4.2.23 Linka 2 vizualizace; [s6].....	42
Obr. 4.2.24 Upravený algoritmus a operátorský panel linky 2; [s6].....	43
Obr. 4.3.1 Jednotka CPX-CEC; [6].....	45
Obr. 4.3.2 Kryt modulu CPX AB-8-KL-4POL; [7]	46
Obr. 4.3.3 Zapojení vstupně-výstupního modulu; [7].....	46
Obr. 4.3.5 Nastavení úloh; [s6]	48
Obr. 4.3.6 Volání programu; [s6].....	48
Obr. 4.3.7 Nastavení úlohy Other; [s6]	48
Obr. 4.3.8 Symbol file;[s6]	48
Obr. 4.3.9 Konfigurace PLC; [s6].....	49
Obr. 4.3.10 Vyhledávání modulu; [s6]	50
Obr. 4.3.11 Simulace linky 1 s reálným PLC	50
Obr. 4.3.12 Reálné zapojení PLC pro linku 1	51

Seznam tabulek

Tab. 1 Tabulka proměnných	18
Tab. 2 Tabulkový zápis činností	19
Tab. 3 Technické údaje; [6]	45