

UNIVERZITA PARDUBICE
FAKULTA ELEKTROTECHNIKY A
INFORMATIKY

BAKALÁŘSKÁ PRÁCE

2013

PATRIK HORKÝ

Univerzita Pardubice

Fakulta elektrotechniky a informatiky

Laboratorní úlohy pro PLC Siemens

Patrik Horký

Bakalářská práce

2013

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2012/2013

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Patrik Horký**
Osobní číslo: **I10441**
Studijní program: **B2646 Informační technologie**
Studijní obor: **Řízení procesů**
Název tématu: **Laboratorní úlohy pro PLC Siemens**
Zadávající katedra: **Katedra řízení procesů**

Z á s a d y p r o v y p r a c o v á n í :

Cílem práce bude tvorba laboratorních úloh pro PLC automat Siemens. Laboratorní úlohy budou sestaveny pro simulovaná zařízení laboratoře specializace. Výsledkem práce bude výukový materiál s popisem metod programování PLC automatu a podrobnou dokumentací k jednotlivým laboratorním úlohám, včetně jejich vzorového řešení.

Teoretická část:

- seznámení s PLC automatem Siemens,
- seznámení s hardwarovým řešením laboratorních úloh.

Implementační část:

- navrhnete soubor laboratorních úloh,
- navržené laboratorní úlohy realizujete,
- vypracujete podrobnou dokumentaci k laboratorním úlohám.

Požadavky na studenta:

Základní orientace v programování PLC automatů.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

Martinásková M., Šmejkal L., PLC a automatizace, BEN - technická literatura, Praha 2002

Šmejkal L., PLC a automatizace, BEN - technická literatura, Praha 2005

Tecomat, PLC for machine, process, technology, transport and building automation [online]. Teco a.s. [cit. 28.10.2012]. Dostupné

z <http://www.tecomat.com>

firemní dokumentace k laboratorním úlohám

Vedoucí bakalářské práce:

Ing. Libor Havlíček, Ph.D.

Katedra řízení procesů

Datum zadání bakalářské práce:

28. listopadu 2012

Termín odevzdání bakalářské práce:

10. května 2013



prof. Ing. Simeon Karamazov, Dr.
děkan



L.S.



Ing. Daniel Honc, Ph.D.
vedoucí katedry

V Pardubicích dne 29. března 2013

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 3. 3. 2013

Patrik Horký

Poděkování

Tímto děkuji Ing. Liboru Havlíčkovi, Ph.D. za odborné vedení při psaní bakalářské práce a Ing. Stanislavu Pšeničkovi za odborné konzultace. Dále děkuji Ing. Stanislavu Pšeničkovi za zapůjčení modulárního PLC Siemens.

Anotace

Práce se zabývá problematikou řízení procesů pomocí programovatelných logických automatů. Výsledkem snažení je vytvoření výukového materiálu pro programování PLC Siemens. Po osvojení dovedností, jež práce předkládá, bude čtenář schopen naprogramovat základní software pro PLC a získá přehled o vývojovém prostředí Step7 a WinCC Flexible 2008. Součástí práce je i vzorové řešení dané úlohy.

Klíčová slova

Výukové materiály, automatizace, PLC, Siemens

Title

Laboratory tasks for PLC Siemens

Annotation

Bachelor thesis deals with automatic control via programmable logic controllers. The result of efforts is creating of teaching materials for PLC programming of Siemens logic controllers. After mastering the skills offered by the thesis, the reader will be able to program basic PLC software and obtain basic ability of working with Step7 and WinCC Flexible 2008 development tools. Sample solution is included.

Keywords

Teaching materials, automation, PLC, Siemens

Obsah

0 Úvod.....	12
1 Programovatelné logické automaty	13
1.1 Řízení pomocí PLC	13
1.2 HW a konfigurace	15
1.2.1 Mikro PLC	15
1.2.2 Kompaktní PLC	15
1.2.3 Modulární PLC	15
2 Výukové materiály – Teoretický úvod do programování	18
2.1 Logické operátory a operace	18
2.1.1 AND	19
2.1.2 OR	19
2.1.3 Negace.....	20
2.1.4 Schefferův funktor	20
2.1.5 Piercův funktor.....	21
2.2 Booleova algebra.....	21
2.2.1 Příklad minimalizace funkce:	22
2.3 Programovací jazyky PLC.....	24
2.3.1 Jazyk mnemokódů	24
2.3.2 Ladder Diagram	25
2.3.3 Function Block Diagram.....	25
2.3.4 Structured Text Language.....	26
2.4 Vykonávání programu v PLC.....	26
3 Výukové materiály – Praktický úvod do programování.....	28
3.1 Vývojové prostředí Step7.....	28
3.1.1 HW Config.....	29
3.1.2 NetPro	32
3.1.3 Programování.....	33
3.1.4 Datové bloky	35
3.1.5 Tabulka proměnných	35
3.1.6 Simulátor S7-PLCSIM1	36
3.2 Vývojové prostředí WinCC Flexible 2008.....	38
4 Zadání samostatných projektů	44

5 Závěr	45
Přílohy.....	1

Seznam obrázků

Obrázek 1: Blokové schéma zpětnovazebního obvodu	13
Obrázek 2: Blokové schéma přímovazebního obvodu	13
Obrázek 3: Jednoduchý uzavřený regulační obvod	14
Obrázek 4: Blokové schéma PLC	14
Obrázek 5: PLC typu MPC300 s instalačním rámem (přední pohled)	17
Obrázek 6: PLC typu MPC300 s instalačním rámem (zadní pohled)	17
Obrázek 7: Schéma funkce f před minimalizací	23
Obrázek 8: Schématické značky logických hradel	23
Obrázek 9: Schéma funkce f po minimalizaci	24
Obrázek 10: Příklad kódu v jazyce mnemkódů	24
Obrázek 11: Příklad kódu v jazyce LD	25
Obrázek 12: Příklad kódu v jazyce FBD	25
Obrázek 13: Příklad kódu v jazyce STL]	26
Obrázek 14: Vkládání objektů v Step7	29
Obrázek 15: HW Config - Simatic 300(1) v Step7	30
Obrázek 16: Nastavení časovače CPU na byte 255	31
Obrázek 17: Prostředí NetPro	32
Obrázek 18: Prostředí NetPro s nastavenou sítí industrial ethernet (zelená)	33
Obrázek 19: Rozhraní pro programování ve funkci FC1	34
Obrázek 20: Simulátor S7-PLCSIM1	36
Obrázek 21: Compile and Download Objects v Step7	37
Obrázek 22: Monitorování proměnných v režimu online	38
Obrázek 23: Prostředí WinCC Flexible 2008	39
Obrázek 24: Nabídka Events v prostředí button	40
Obrázek 25: SetBit ve WinCC Flexible 2008	40
Obrázek 26: Nastavení záložky Appearance v rámci prvku Rectangle	41
Obrázek 27: Zjednodušené schéma algoritmu řízení	2
Obrázek 28: Zjednodušené schéma algoritmu řízení funkce FC2	2
Obrázek 29: Podoba vizualizace ve vzorovém řešení	2

Seznam tabulek

Tabulka 1: Základní vlastnosti vybraných CPU	16
Tabulka 2: Základní vlastnosti vybraných CPU	16
Tabulka 3: Logické, napěťové úrovně vybraných CPU	18
Tabulka 4: Pravdivostní tabulka operátoru AND	19
Tabulka 5: Pravdivostní tabulka operátoru OR	19
Tabulka 6: Pravdivostní tabulka negace	20
Tabulka 7: Pravdivostní tabulka operátoru NAND	20
Tabulka 8: Pravdivostní tabulka operátoru NOR	21
Tabulka 9: Specifikace CPU IM151-8 PN/DP	28

Seznam zkratok a značek

PLC - programmable logic controller, programovateľný logický automat

HW - hardware

SW - software

CPU - central processing unit

MPI - multi – point interface

FEI – Fakulta elektrotechniky a informatiky

MCU – microcontroller unit

VŠ – vysoká škola

0 Úvod

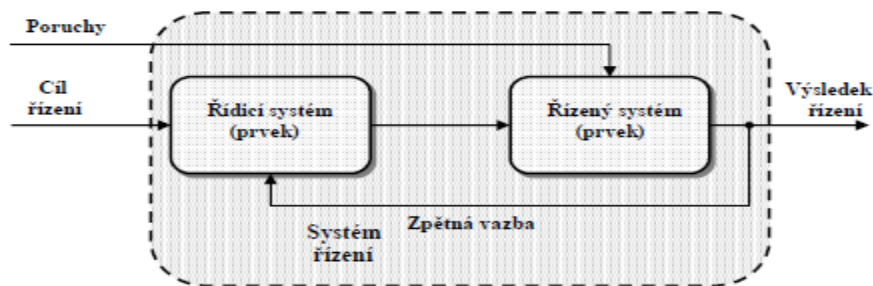
Cílem této bakalářské práce je poskytnout základní výukové materiály pro programování PLC systémů Simatic společnosti Siemens. Práce předkládá jak základní informace z oblasti teorie řízení či zpracování logických funkcí a booleovi algebry, tak systematický návod pro práci ve vývojovém prostředí Step7 a WinCC Flexible 2008. Tato bakalářská práce by měla sloužit jako doplňkový materiál při studiu programovatelných logických automatů. Součástí jsou zadání laboratorních úloh pro vybavení laboratoře automatizace na FEI (fakulta elektrotechniky a informatiky). V práci je zahrnuto vzorové řešení zadaného projektu, realizované na CPU IM151-8 PN/DP

1 Programovatelné logické automaty

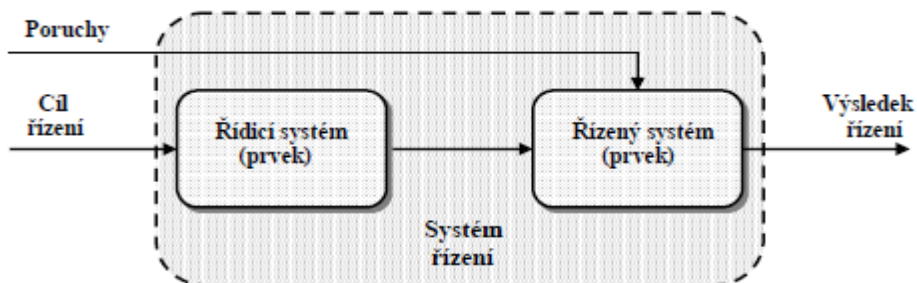
„Programovatelný automat je uživatelsky programovatelný řídicí systém přizpůsobený pro řízení průmyslových a technologických procesů, mnohdy specializovaný na úlohy převážně logického typu (obzvláště u starších typů nebo u nejmenších systémů). Nejčastěji se označuje zkratkou PLC (Programmable Logic Controller), v německé literatuře se lze setkat s označením SPS (Speicherprogrammierbare Steuerung)“ [1].

1.1 Řízení pomocí PLC

Projekty průmyslové automatizace lze řešit mnoha způsoby podle preferencí zákazníka. Řízení pomocí PLC lze pojmout formou ovládání, tedy přímého řízení [1, str. 33], kdy automat daný systém pouze ovládá, bez jakékoliv zpětné informace o stavu procesu. Také lze systémy navrhovat pro práci podle principu zpětné vazby. Tzv. zpětnovazební regulace pracuje na základě informace o výstupu systému. Jinak řečeno další vstup je nastaven v závislosti na předchozím výstupu.

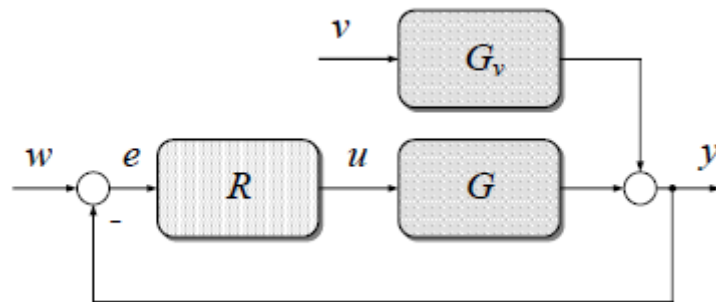


Obrázek 1: Blokové schéma zpětnovazebního obvodu [2]



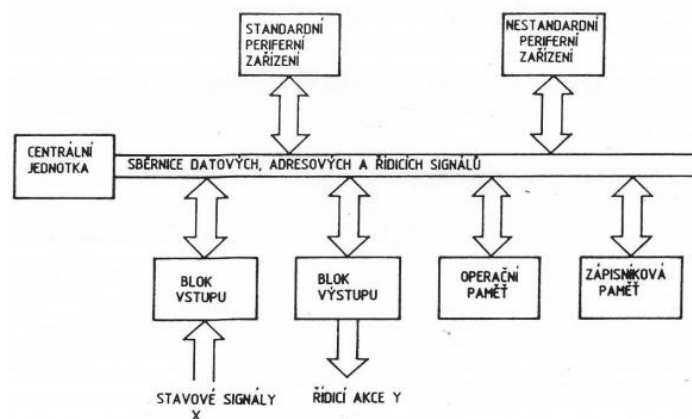
Obrázek 2: Blokové schéma přímovazebního obvodu [2, str. 22]

„V jednoduchém regulačním obvodu na obrázku 3 G zohledňuje dynamické vlastnosti řízeného systému vzhledem k akční veličině a G_v vzhledem k poruše (dynamika může být rozdílná), R je regulátor, $y(t)$ řízený výstup, $u(t)$ akční vstup, $v(t)$ porucha, $w(t)$ žádaná hodnota výstupní veličiny (referenční signál), a $e(t)$ regulační odchylka.“ [2, str 62].



Obrázek 3: Jednoduchý uzavřený regulační obvod [2, str. 62]

Automatizace pomocí PLC je spolehlivé řešení, automat je navrhován tak, aby díky svým vlastnostem a uspořádáním zachoval odolnost a spolehlivost, která se po něm požaduje v průmyslovém prostředí. Použití PLC je vhodné také zejména díky své jednoduchosti. Již při návrhu systému bychom měli mít představu o tom, jaké CPU a jaké moduly jsou vhodné pro realizaci daného projektu [1, str. 38]. Pojem modul bude vysvětlen v textu níže. Základní otázkou, při plánování řízení pomocí PLC je HW konfigurace použitého automatu.



Obrázek 4: Blokové schéma PLC [3]

1.2 HW a konfigurace

Každý projekt průmyslové automatizace vyžaduje jiné zacházení s veličinami, jiný přístup k řízení, jiné vlastnosti PLC. Typy logických automatů bychom tedy rozdělili do tří skupin, a to na základě velikosti a komplexnosti řídicího systému, který jsme za pomoci PLC schopni realizovat. Jsou jimi mikro PLC, kompaktní PLC, modulární PLC [1, str. 40].

1.2.1 Mikro PLC

Mikro PLC jsou automaty, vhodné k realizaci nejméně technologicky náročných dějů, nabízená sestava je pevná, bez možnosti dodefinování případných modulů. Obvykle tato varianta řešení poskytuje 6 vstupů a 6 výstupů, sestava je vhodná pro nejmenší systémy, „*pro větší pak sestavy 4/4, 8/6, 8/8, 12/12, ... atd.*“ [1, str. 40].

1.2.2 Kompaktní PLC

Pokud bychom při návrhu řešení uvážili, že použití mikro PLC je nevhodné či nedostačující, další možností bude typ kompaktní PLC. U tohoto typu PLC již má návrhář systému jistou volnost. Je totiž možné, na rozdíl od mikro PLC, přidat některé potřebné moduly. Výběr je omezený. Jako příklad modulů lze uvést např. modul rychlých čítačů, analogový vstupní či výstupní modul nebo modul regulátoru. [1, str. 41] Pokud pro komplexnost není kompaktní PLC dostačující, volíme poslední typ, kterým je modulární PLC.

1.2.3 Modulární PLC

Tento typ se používá při nejnáročnějších aplikacích v průmyslové praxi. Modul PLC je samostatná součástka, se svým konkrétním účelem, kompatibilní se zvoleným CPU (central processing unit). HW řešení je realizováno pomocí nasouvání PLC modulů na kovové rámy „(v délkách 11“, 13“, a 19““ [1, str. 41] Pro doplnění PLC společnosti Siemens nabízí tyto produktové řady: S7-200, S7-1200, S7-1500, S7-300, S7-400. Např. v případě S7-200, lze řešení systému založit na řadě CPU. Jsou jimi standardní výpočetní jednotky, s označením CPU 221, CPU 222, CPU 224 a další [4]. Dále podle potřeby analogové i digitální vstupní moduly, výstupní moduly. Moduly pro ovládání servomotorů, mimo jiné také modemy pro GSM komunikaci či napájecí moduly.

Součástí řešení bývají často operátorské panely Simatic HMI station. Tyto se nenasouvají na rám, ale komunikují se zbytkem PLC pomocí datových spojů technologií profibus, industrial ethernet nebo vlastní sběrnici společnosti Siemens, MPI. Pro úplnost, MPI sběrnice je typ vyhovující standardu RS – 485.

Tabulka 1: Základní vlastnosti vybraných CPU

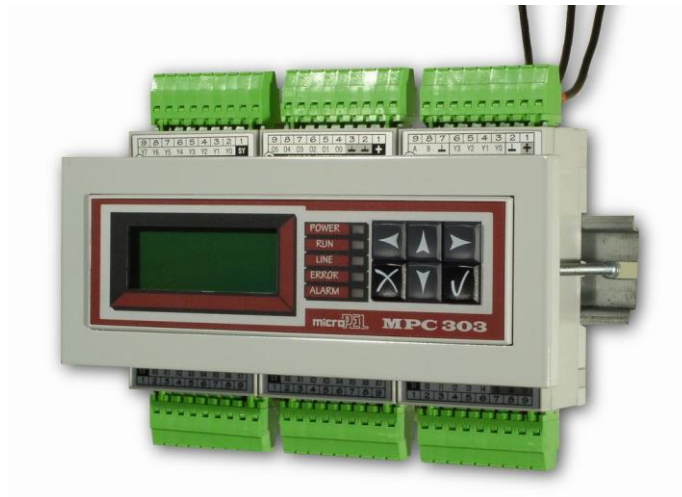
Název	Typ	I/O	Data memory
CPU 221	kompaktní	6x DI, 4x DO	2 KB
CPU 1211C	kompaktní	6x DI, 4x DO, 2x AI (0 - 10V)	30 KB
CPU 1511-1 PN	modulární	dle modulů	1 MB
CPU 312C	kompaktní	10x DI, 6x DO	64 KB
CPU 412	modulární	dle modulů	144 KB

Tabulka 2: Základní vlastnosti vybraných CPU

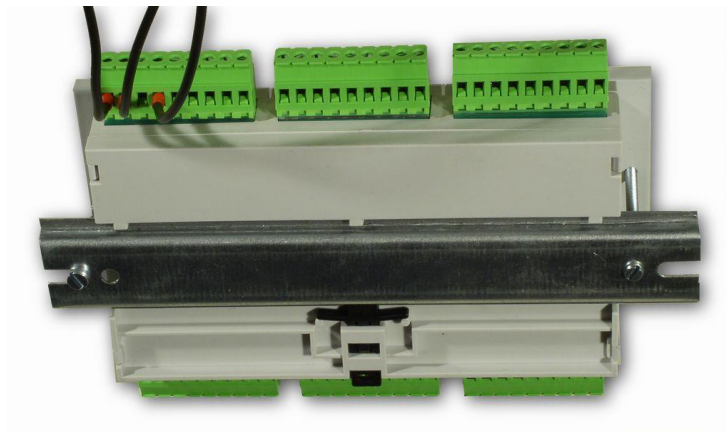
Název	Output	Order number
CPU 221	DC 24 V AC 85 - 264 V / DC	6ES7 211-0AA23-0XB0
CPU 1211C CPU 1511-1	24	6ES7 211-1BE31-0XB0
PN	DC 24 V	6ES7 511-1AK00-0AB0
CPU 312C	DC 24 V	6ES7 312-5BF04-0AB0
CPU 412	DC 24 V	6ES7 412-1XJ05-0AB0

Co se týká HW konfigurace, i ta musí být zahrnuta ve vývojovém prostředí Step7, to je zde pouze předesláno, práce se tím bude zabývat v samostatné kapitole, a to včetně síťové komunikace, nastavení detailů, jako časovače a jiné. Všechny použité moduly i CPU, které reálně instalujeme na kovové rámy, musí být v programu Step7 definovány a nastaveny, buď podle nastavení default nebo podle vlastního nastavení. Cílem první fáze tvorby projektu by měl tedy být kompletní návrh HW řešení, to je výběr typu PLC, výběr případných modulů PLC i dalších součástí jako servomotory, pneumatické či hydraulické prvky, to vše podle potřeb regulačních dějů.

Zde na obrázcích 5 a 6 vidíme kovové rámy, které slouží pro ukotvení PLC systému v průmyslové praxi. Nejčastěji bývají konstrukčně umístěny na dobře přístupné místo, tak aby se k nim snadno dostal servisní technik nebo programátor, a zároveň tak, aby byl PLC systém dostatečně chráněn před vlivem okolního prostředí.



Obrázek 5: PLC typu MPC300 s instalačním rámem (přední pohled) [4]



Obrázek 6: PLC typu MPC300 s instalačním rámem (zadní pohled) [4]

2 Výukové materiály – Teoretický úvod do programování

V následující kapitole pojedná práce o teorii, jejíž osvojení je nutnou podmínkou k programování logických automatů. Nejprve práce zmíní základy logických operátorů a funkcí a booleovské algebry, poté informuje o vykonávání programu v PLC a o konkrétních programovacích jazycích, ve kterých lze vytvářet řídicí SW. Náplní druhé části výukového materiálu bude seznámení s prostředím Step7 a s prostředím WinCC Flexible 2008. Součástí je i zadání samostatných projektů a jako příloha práce potom vzorové řešení.

2.1 Logické operátory a operace

Informace je v číslicových přístrojích uložena ve formě tzv. binárního kódu. Jedná se o číselný kód v tzv. dvojkové soustavě, kdy každý z prvků může nabývat pouze hodnot 0 nebo 1. V PLC (i jiných číslicových přístrojích) to znamená, že se v HW velmi rychle mění napěťové úrovně. Pokud bychom problematiku vztáhli na PLC, které většinou pracují na 24 voltové logice, potom to znamená, že logická nula je reprezentována napěťovou úrovní asi 0 – 10 V a jako logickou 1 chápe PLC napětí v rozmezí asi 10 – 24 V.

Tabulka 3: Logické, napěťové úrovně vybraných CPU

Název	Log. 0	Log. 1
CPU 221	0 - 5 V	15 V
CPU 1211C	5 V	15 V
CPU 1511-1 PN	-	-
CPU 312C	-3 - 5 V	15 - 30 V
CPU 412	-	-

Práce nyní obeznámí s konkrétními logickými operátory a operacemi, které se v souvislosti s číslicovými přístroji hojně využívají. Jsou jimi: konjunkce, disjunkce, negace, Schefferův funktor, Piercův funktor [1, str. 88]. Dále se zde vyskytnou pravdivostní tabulky pro zvolené operátory a operace, čtenář by je měl chápat jako pravidla pro logické funkce a jejich vyhodnocování.

2.1.1 AND

Tabulka 4: Pravdivostní tabulka operátoru AND

A	B	AND
0	0	0
0	1	0
1	0	0
1	1	1

Operátor AND přísluší právě operaci logického součinu. Jak je vidět z tabulky 1, logický součin je roven jedné, právě tehdy když jsou všechny jeho operandy rovny jedné. „Ve výrazech (v teoretické literatuře) se označuje symbolem „^“ nebo „&“, častěji však jen symbolem pro násobení, „.“ (tečkou) nebo „*“, někdy se symbol násobení v zápisu vypouští.“ [1, str. 81] Funkce logického součinu se využívá v programech tam, kde je třeba splnit všechny zadané podmínky. Velmi zjednodušeně se dá říci, že pokud platí A & platí B & platí C, potom nastane situace D. Je zde vidět, že situace D nastane pouze tehdy, platí – li současně A, B i C. Tedy, jestliže je buď jen jedna z podmínek A, B, C nepravdivá, nikdy nemůže situace D nastat.

2.1.2 OR

Tabulka 5: Pravdivostní tabulka operátoru OR

A	B	OR
0	0	0
0	1	1
1	0	1
1	1	1

Operátor OR vyjadřuje operaci logického součtu. Logický součet je roven jedné, je-li alespoň jeden z jeho operandů roven jedné. To znamená, je nulový pouze tehdy, jsou-li všechny jeho operandy nulové. „Logický součet si lze představit jako volnou analogii ke sčítání dvojkových číslic s tím rozdílem, že výsledek omezíme na hodnotě 1. Ve výrocích odpovídá logickému součtu spojka „nebo“ (odtud symbol OR). V teoretické literatuře se pro logické výrazy

používá symbol „ \vee “ nebo jen symbol pro sčítání (+)“ [1, str. 84]. Tedy Pokud platí $A + B + C$, potom nastane situace D. To tedy znamená, že proto, aby nastala situace D, je nutné, aby platilo alespoň jedno z následujících: A, B nebo C. Samozřejmě mohou platit dvě z uvedených, nebo i všechny. Jistě je zřejmé, že jediná situace, kdy nenastane D, je ta, když neplatí ani jeden z operandů A, B, C.

2.1.3 Negace

Tabulka 6: Pravdivostní tabulka negace

A	B	NEG A	NEG B
0	1	1	0

Logická, bitová negace je, jak je vidět v tabulce 3, změna hodnoty logického operandu. Velmi jednoduše, pokud je hodnota operandu 1, jeho znegováním se změní na 0 a naopak. Logická negace se také znázorňuje pomocí znaků „!“ , „~“ , „-“ , „ \bar{x} “ nebo „NEG“.

2.1.4 Schefferův funktor

Tabulka 7: Pravdivostní tabulka operátoru NAND

A	B	NAND
0	0	1
0	1	1
1	0	1
1	1	0

Schefferův funktor, nebo také NAND, je operátor negovaného součinu, tedy jeho výsledek je opačný, k výsledku logického součinu. Opět tedy jinak řečeno. Výsledkem Schefferova funktoru je 1 ve všech případech, kromě případu, kdy jsou oba operátory rovny jedné.

2.1.5 Piercův funktor

Tabulka 8: Pravdivostní tabulka operátoru NOR

A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0

Již podle označení Piercova funktoru NOR, je zřejmé, že výsledek operace Piercova funktoru je negovaný výsledek logického součtu. Tedy jednoduše. Piercův funktor je roven 0 v každém případě, kromě případu, kdy oba operandy jsou rovny 0.

2.2 Booleova algebra

„Libovolnou logickou funkci je možné popsat výrazem s použitím operátorů AND, OR a NOT – booleovských operátorů. Uvádíme soubor pravidel pro Booleovu algebru. Pravidla uvádíme po dvojicích, protože jsou navzájem duální (doplňková) – pokud v jednom zaměníme operátor AND za OR a naopak OR za AND a navzájem změněme konstanty 0 a 1, dostaneme druhé (duální) pravidlo. To je důsledkem operátorů AND a OR.“ [1, str. 93] Pravidla pro booleovu algebru jsou velmi přehledně popsána v citované literatuře, proto zde v práci uvede totožný seznam, ovšem pouze ve variantě logického součtu.

1. Komutativní zákon

$$a + b = b + a$$

2. Distribuční zákon

$$(a + b)c = ac + bc$$

3. Vyloučení třetího

$$a + \bar{a} = 1$$

4. Neutrálnost konstant

$$a + 0 = a$$

5. Agresivnost konstant

$$a + 1 = 1$$

6. Idempotence

$$a + a = a$$

7. Asociativná zákon

$$(a + b) + c = a + (b + c)$$

8. Dvojitá negace

$$\bar{\bar{a}} = \overline{(a)} = a$$

9. Absorpce

$$a + ab = a$$

$$a + \bar{a}b = a + b$$

10. De Morganovy zákony

$$\overline{a + b} = \bar{a}\bar{b}$$

$$\overline{\bar{a} + \bar{b}} = ab$$

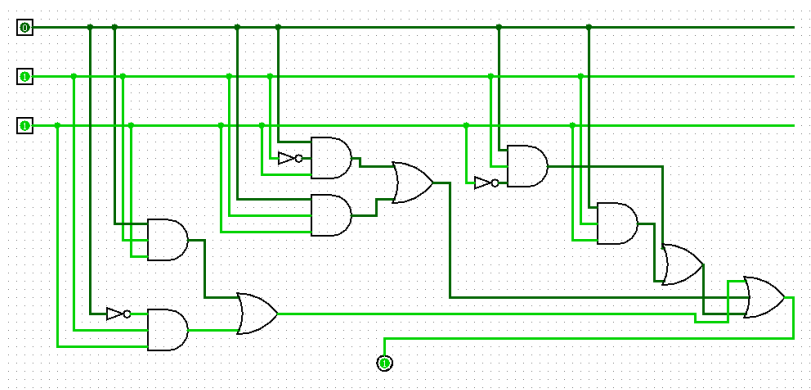
Logické funkce jsou mnohdy složité, a proto je z hlediska úspory a zjednodušení lze minimalizovat. Toto se provádí pomocí uplatnění pravidel booleovy algebry. Za využití výše popsaných pravidel nyní tedy dojde k minimalizaci logické funkce.

2.2.1 Příklad minimalizace funkce:

$$\begin{aligned} f &= (\bar{a}bc + abc) + (a\bar{b}c + abc) + (ab\bar{c} + abc) = \\ &= bc(\bar{a} + a) + ac(\bar{b} + b) + ab(\bar{c} + c) = \\ &= 1bc + 1ac + 1ab = \\ &= bc + ac + ab \text{ [1, str. 101]} \end{aligned}$$

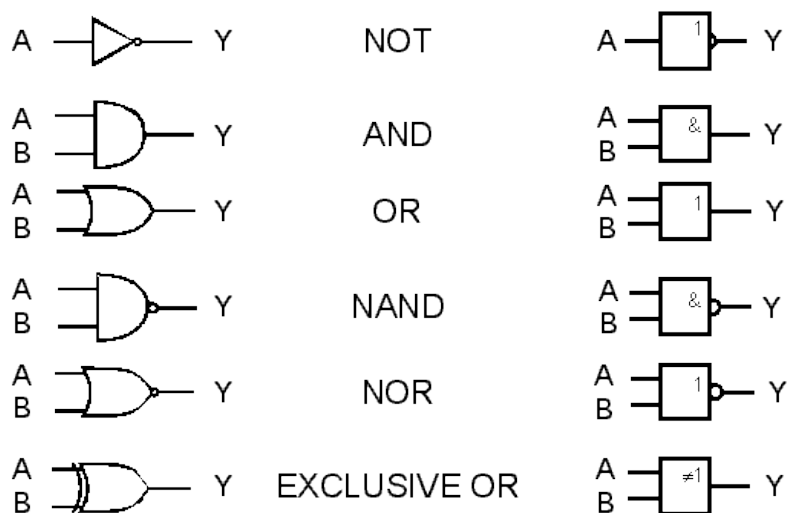
Výsledkem je tedy $f = bc + ac + ab$

Následně sestavíme schéma dané, logické funkce f , v programu logisim, za použití logických hradel.



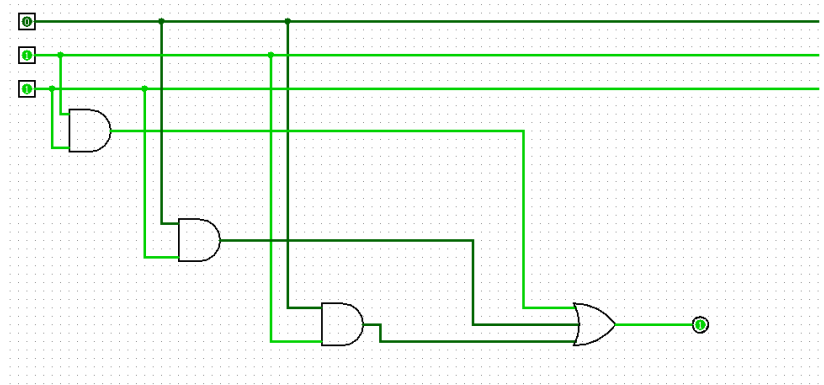
Obrázek 7: Schéma funkce f před minimalizací

Na obrázku 8 jsou znázorněny schématické značky pro logická hradla.



Obrázek 8: Schématické značky logických hradel [6]

Na obrázku 9 je potom vidět, že schéma funkce f po minimalizaci má stejný výstup, při stejné kombinaci vstupů, jako funkce f , před minimalizací. Dokazuje to, že funkce si zachovává po minimalizaci svou logiku, je pouze jednodušší a úspornější. Samozřejmě by tyto dvě funkce (respektive jedna a tatáž!) měly stejné výstupy při jakýchkoliv, stejných kombinacích vstupních parametrů. Závěrem této podkapitoly je zajisté tvrzení, že cílem minimalizace logické funkce je stav, kdy již funkce nemůže být dále zjednodušena. Mimo minimalizaci za pomoci booleovi algebry je možné funkce minimalizovat za pomoci metody McCluskey nebo za použití Karnaughovi mapy.



Obrázek 9: Schéma funkce f po minimalizaci

2.3 Programovací jazyky PLC

„K programování nabízejí PLC systémy specializované jazyky, původně navržené pro snadnou, názornou a účinnou realizaci logických funkcí. Jazyky systémů různých výrobců jsou podobné, nikoliv však stejné. Přímá přenositelnost programů mezi PLC různých výrobců není možná, daří se to obvykle jen mezi systémy téhož výrobce. Mezinárodní norma IEC 1131-3 sjednocuje programovací jazyky pro PLC. Kodifikuje čtyři typy jazyků“ [1, str. 49]. Těmito jazyky jsou jazyk mnemokódů, LD (ladder diagram), FBD (function block diagram), STL (structured text language).

2.3.1 Jazyk mnemokódů

Tento jazyk je ve své podstatě obdobou jazyka assembler. Je tedy velmi strojově orientovaný. Je to seznam příkazů, pevně svázaných s instrukcemi PLC. Nabízí, stejně jako jazyk assembler, symbolická pojmenování pro různé instrukce, návěští, uživatelské registry a jiné. Je to tedy jazyk, který nejlépe pronikne k HW PLC, a díky kterému dokáže profesionální programátor nejlépe využít instrukční sady daného PLC [1, str. 49].

```

Label: LD      in2
ANA          (* result := ana (in2) *)
MUL          2      (* result := 2*ana (in2) *)
ST          temp   (* temp := result *)
LD          in1
ANA          (* result := 2*ana (in2) + ana (in1) *)
ADD          temp   (* result := 2*ana (in2) + ana (in1) *)
MUL          2      (* 4*ana (in2) + 2*ana (in1) *)
ST          temp   (* temp := result *)
LD          in0
ANA          (* result := 4*ana (in2) + 2*ana (in1) + ana(in0) *)
ADD          temp   (* result := 4*ana (in2) + 2*ana (in1) + ana(in0) *)
ST          SUBPRO (* return the current result *)
              (* to the calling program *)

```

Obrázek 10: Příklad kódu v jazyce mnemokódů [7]

2.3.2 Ladder Diagram

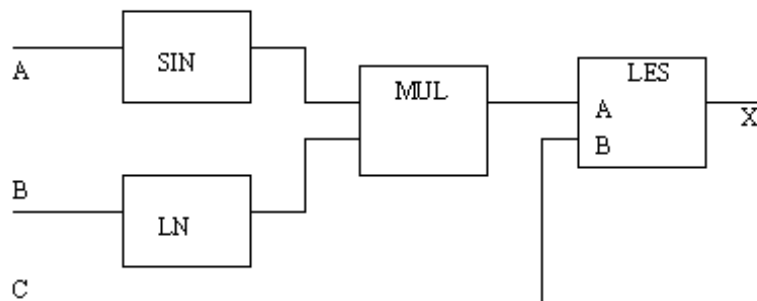
Ladder Diagram, nebo také německy Kontaktplan, je jazykem, který již nabízí vývojáři jistý programátorský komfort, a to ve formě grafického znázornění. Jazyk LD je totiž interpretovaný schémata reléové logiky, se symboly spínacích prvků a s bloky, jako komparace, bitové a nebitové operace a další, které jsou prostředky programovacích jazyků. Tyto bloky jsou reprezentovány jako obdélníky se vstupy a výstupy. Výhodou jazyka Ladder Diagram bývá grafické znázornění vodivé cesty, kdy v simulaci snadno poznáme porouchaný logický prvek a podobně. Jazyk LD lze navrhnout pro použití, jako jednoduchý diagnostický nástroj.



Obrázek 11: Příklad kódu v jazyce LD [8]

2.3.3 Function Block Diagram

Jazyk FBD je opět jazykem grafickým. FBD používá pro znázornění bloky o různé velikosti, podle počtu vstupů a výstupů. Své bloky mají jak logické instrukce, tak i funkční bloky, jako čítače či časovače. Jako příklad zde práce uvede jazyk Merkur pro regulátory Tecoreg nebo PLC Tecomat [1, str. 50].



Obrázek 12: Příklad kódu v jazyce FBD [9]

2.3.4 Structured Text Language

Jazyk STL je jakousi obdobou vyšších programovacích jazyků jako např. jazyk C nebo Pascal. „Umožňuje úsporný a názorný zápis algoritmů. Je oblíbený zejména u mladých absolventů odborných škol, pro které je nejpřirozenější“ [1, str. 51].

```
0002 VAR
0003     fbTimer: TON;
0004     bToggle AT %Q*: BOOL;
0005 END_VAR
0006
0001 fbTimer(PT:=t#250ms, in:= NOT fbTimer.Q);
0002 IF fbTimer.Q THEN
0003     bToggle:=NOT bToggle;
0004 END_IF
0005
0006
0007
```

Obrázek 13: Příklad kódu v jazyce STL [10]

2.4 Vykonávání programu v PLC

Program v PLC je jakousi posloupností příkazů a operací. Jeho vykonávání je cyklické, ovšem programátor, ani nikdo jiný se nemusí starat o navrácení programu na začátek, jak tomu může být kdokoliv zvyklý z programů pro MCU (micro controller unit). Tedy u programu pro logické automaty je smyčka uzavírána systémově, samotným PLC.

„Naopak každé dlouhodobé setrvání programu v programové smyčce je „fatální chybou“ a systém jej hlásí jako „překročení doby cyklu“. Vždy po vykonání poslední instrukce uživatelského programu (např. E0) je předáno řízení systémovému programu, který provede tzv. otočku cyklu. V ní nejprve aktualizuje hodnoty výstupů a vstupů: hodnoty uložené dosud v paměti jako obrazy výstupů přepíše do registrů výstupních periferních modulů a hodnoty ze vstupních modulů okopíruje do paměťových obrazů vstupů. Dále aktualizuje časové údaje pro časovače a systémové registry, ošetří komunikaci a provede ještě řadu realizačních úkonů. Po otočce cyklu je opět předáno řízení první instrukci uživatelského programu“ [1, str. 47].

Program v PLC tedy nepracuje ani tak se samotnými daty, jako spíše s jejich obrazy v registrech PLC. Jak bylo vysvětleno výše, aktualizace jejich hodnot je provedena při otočce cyklu. Touto implementací práce s daty jsou ošetřeny hazardní situace vyplývající z možné nekonzistence dat. Takto se například PLC nemůže odvolávat na hodnoty, které neexistují a

podobně. Některé bloky se v PLC „otočí“ pouze jednou, a to na začátku chodu PLC, jiné se „otáčejí“ neustále dokola, podle systémového programu, který je již zakomponován v PLC.

3 Výukové materiály – Praktický úvod do programování

3.1 Vývojové prostředí Step7

V následujícím textu pojedná práce o hlavním vývojovém nástroji pro tvorbu programů pro PLC Siemens Simatic, kterým je Step7, konkrétně verze 5.5 pro Windows XP. Zde si již předpokládá, že je SW Step7 spolu se simulátorem PLCSIM1 nainstalován na PC se systémem Windows XP. Výukový materiál uvádí příklady za použití HW CPU IM151-8 PN/DP.

Tabulka 9: Specifikace CPU IM151-8 PN/DP

CPU	Napájení	Vnitřní paměť (program a data)	
IM151-8 PN/DP	24 V DC	128 kbyte	
CPU time	Max. počet bloků	DB (počet/paměť)	FB (počet/paměť)
0.1 μ s	1024	511 / 64 kbyte	1024 / 64 kbyte
FC (počet/paměť)	OB (počet/paměť)	Oblast I/O adres	Rozhraní
1024 / 64 kbyte	64 kbyte	2048 byte	PROFINET (ethernet)

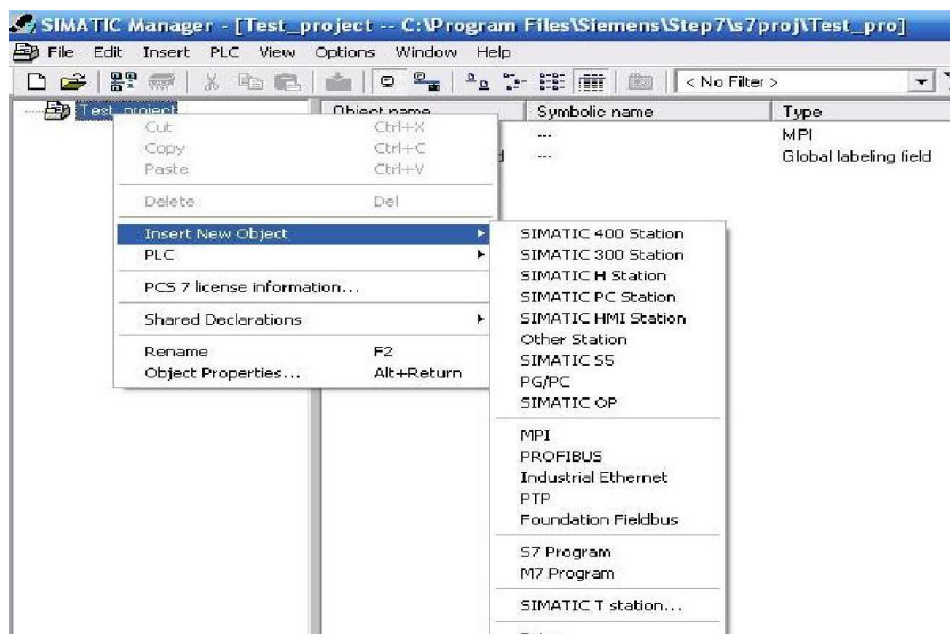
Všechny části programu a veškeré systémové soubory jsou v Step7 uloženy v jednom, vlastním projektu. Jako ve většině vývojových prostředí (Eclipse, NetBeans, Microsoft Visual Studio, AVR Studio) i v SW Step7 začíná veškerá činnost vytvořením nového projektu.

Postup vytvoření nového projektu:

- 1) Kombinací kláves Ctrl + N vytvoříme nový projekt. Zvolíme vlastní pojmenování a umístění projektu.

3.1.1 HW Config

HW Config je nástroj pro HW konfiguraci celého S7 projektu. To zahrnuje zejména: jaká výpočetní jednotka bude použita či jaké moduly budou v projektu využívány. Komponenty se do projektu přidávají pomocí tak zvaného order number. Je to jedinečné označení komponentů v rámci HW Siemens, pomocí kterého se komponenty nakupují v průmyslové praxi. Naše testovací PLC je typu modulární PLC a bude obsahovat následující komponenty: CPU IM151-8 PN/DP (typ Simatic 300), napájecí modul PM-E DC24V, modul digitálních vstupů 4DI DC24V ST (4 digitální vstupy), modul digitálních vstupů 8DI DC24V (8 digitálních vstupů) a modul digitálních výstupů 4DO DC24V/0.5A ST (4 digitální výstupy). Digitální moduly vstupní i výstupní obsahují adresy pro své proměnné. Adresy vstupních modulů nesou označení IX.Y, adresy výstupních modulů nesou potom označení QX.Y. V průvodci si ukážeme, jak nastavit vnitřní časovač CPU na konkrétní byte. Dále jak nastavit konkrétní adresy vstupních a výstupních modulů a také jakým způsobem do projektu přidat Siemens HMI Station operátorský panel.

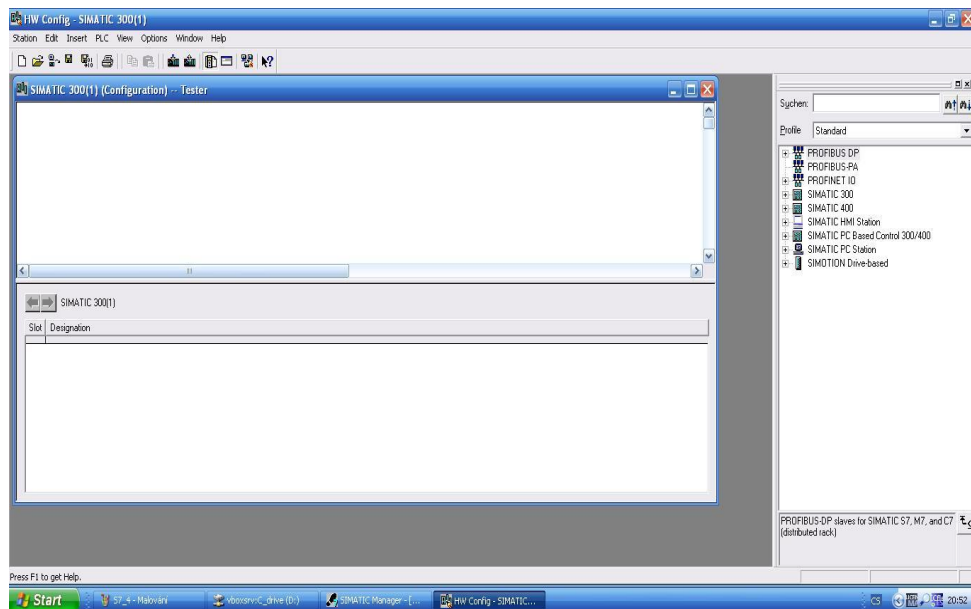


Obrázek 14: Vkládání objektů v Step7

Postup HW konfigurace v HW Config:

- 1) Klikněme v levé části prostředí na ikonu s názvem projektu pravým tlačítkem myši a zvolme položku Insert New Object

- 2) Kliknutím zvolíme stanici Simatic 300 Station pro náš projekt.
- 3) Levým tlačítkem myši klikneme na Simatic 300(1), které se vytvořilo v levé části prostředí Step7 a v pracovní ploše dvojklikem zvolíme položku Hardware.
- 4) V pravé části ve vyhledávacím okně vložíme CPU IM151-8 PN/DP podle order number 6ES7 151-8AB00-0AB0. IP adresu nastavíme prozatím libovolnou.
- 5) Dvojklikem na přidané CPU se dostaneme do nastavení. V záložce Cycle/Clock memory nastavíme časovač CPU na byte 255, zaškrtnutím políčka Clock memory. Clock memory nastavíme na 255.

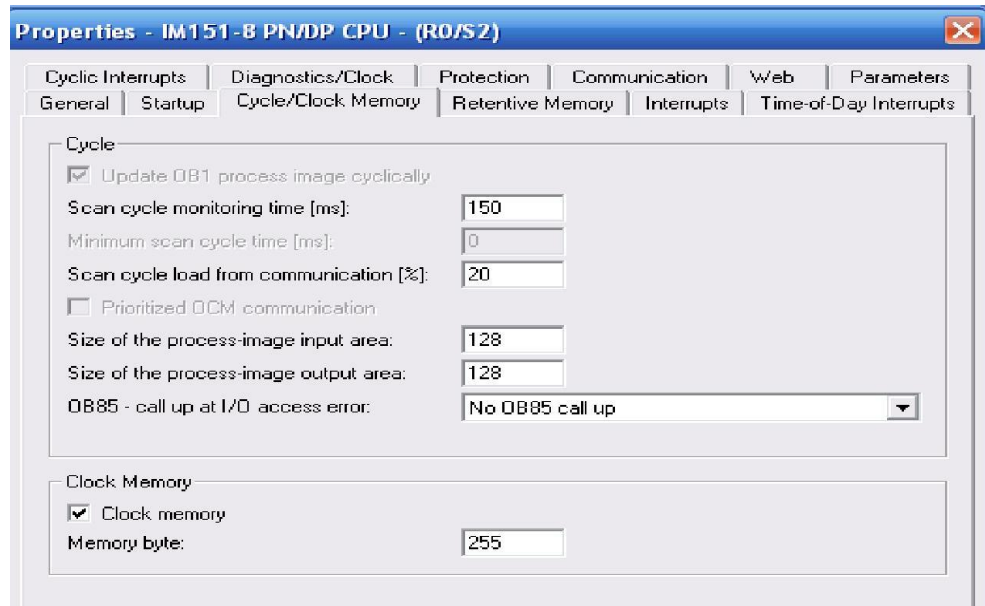


Obrázek 15: HW Config - Simatic 300(1) v Step7

- 6) V pravé části ve vyhledávacím okně vložíme napájecí modul, podle katalogového čísla 6ES7 138-4CA01-0AA0. Jedná se o modul PM-E DC24V.
- 7) V pravé části ve vyhledávacím okně vložíme modul digitálních vstupů, podle order number 6ES7 131-4BD01. Jedná se o modul 4DI DC24V ST.
- 8) Klikneme dvakrát na modul 4DI DC24V ST a vybereme záložku Addresses. Ponecháme vstupním proměnným modulu následující adresy: I0.0, I0.1, I0.2, I0.3.
- 9) V pravé části ve vyhledávacím okně vložíme modul digitálních výstupů podle katalogového čísla 6ES7 132-4BD01-0AA0. Je to modul 4DO DC24V/0.5A ST.

10) Klikneme dvakrát na modul 4DO DC24V/0.5A ST a přesuneme se do záložky Addresses. Zde ponecháme adresy výstupních proměnných tohoto modulu jako: Q0.0, Q0.1, Q0.2, Q0.3.

11) V pravé části ve vyhledávacím okně vložíme modul digitálních výstupů podle katalogového čísla 6ES7 131-4BF00-0AA0. Je to modul 8DI DC24V



Obrázek 16: Nastavení časovače CPU na byte 255

12) Klikneme dvakrát na modul 8DI DC24V a přesuneme se do záložky Addresses. Zde ponecháme adresy vstupních proměnných tohoto modulu jako: I2.0 – I2.7.

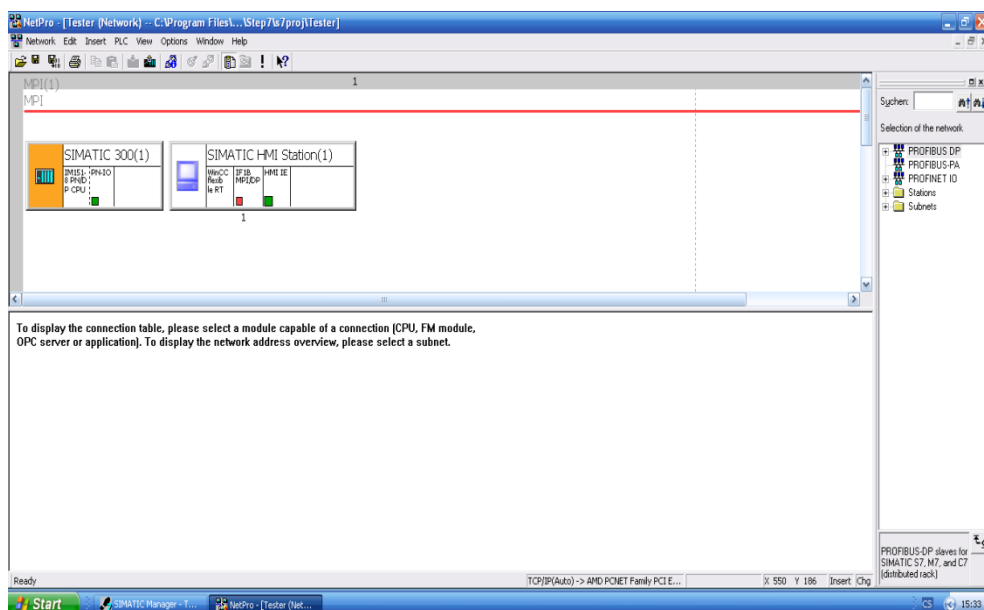
13) Klikneme pravým tlačítkem na název projektu a vybereme možnost Insert New Object. Vybereme možnost SIMATIC HMI station. Vložíme panel s označením TP 177B 4“ color PN/DP.

3.1.2 NetPro

NetPro je nástroj pro správu komunikace mezi jednotlivými částmi PLC systému. Jednotlivé komponenty systému jsou schopny komunikovat pomocí některých z komunikačních technologií, jako MPI nebo ethernet. V tomto případě je schopno CPU IM151-8 PN/DP komunikovat pomocí standardu ethernet, zatímco daný operátorský panel je možno spojit pomocí technologie ethernet nebo MPI. Naše finální řešení bude tedy komunikovat přes rozhraní industrial ethernet.

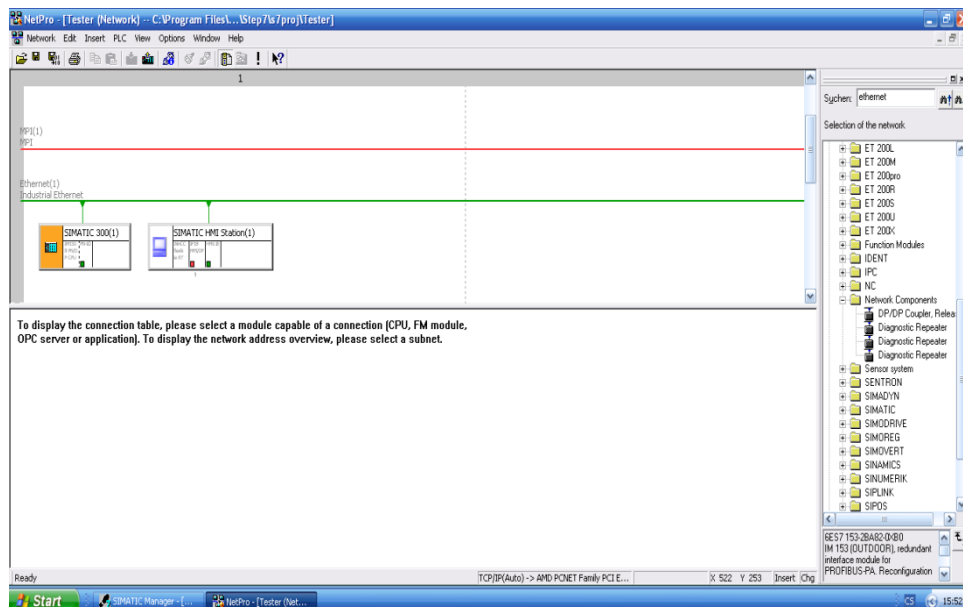
Postup nastavení sítě v NetPro:

- 1) Dvakrát klikneme na Simatic 300(1) v levé části Step7
- 2) Otevřeme NetPro kliknutím na connections
- 3) Dvakrát klikneme na zelený čtverec v rámci bloku Simatic 300(1).



Obrázek 17: Prostředí NetPro

- 4) Zvolíme chtěnou IP adresu CPU a klikneme na New subnet.
- 5) Změníme IP adresu operátorského panelu, tak aby byl v komunikačním dosahu CPU, a aby nenastala kolize IP adres (nesmí být stejné).



Obrázek 18: Prostředí NetPro s nastavenou sítí industrial ethernet (zelená)

3.1.3 Programování

Hlavní struktury, využívané v prostředí Step7 v rámci tvorby řídicího SW jsou následující: tabulka symbolů, bloky, funkce, tabulka proměnných.

Základním logickým prvkem pro manipulaci s daty je bitová proměnná. Ta se v PLC Siemens značí MX.Y. Při použití CPU IM151-8 PN/DP má programátor k dispozici 255 bytů těchto bitových proměnných. To činí 2048 bitů typu MX.Y, pro práci s proměnnými typu boolean. Adresování v PLC je dvojího typu. Přímé a symbolické. Přímé adresování probíhá tak, že k proměnné přistupujeme přímo, a to skrze její fyzickou adresu. Například M20.1 je přímá adresa bitu v CPU. Druhým typem je symbolické adresování, kde programátor vytvoří symbol v tabulce symbolů, a proměnné přiřadí symbolický název. Lze tak zařadit například proměnnou M0.0 do tabulky symbolů a pojmenovat ji naseMalaPromenna. Dále lze bit v CPU s adresou M0.0 adresovat jako naseMalaPromenna, místo přímého adresování pomocí M0.0.

V PLC Siemens rozlišujeme některé základní typy bloků. Bloky jsou schránky na data, ve kterých se uchovávají proměnné nebo části programu. Pro obsluhu inicializační fáze slouží v PLC Siemens tzv. organizační bloky. Blok OB100 se „zavolá“ jednou, a to na začátku chodu PLC. Blok OB1 je určen pro neustálé „otáčení“ cyklu. Tedy na začátku každého cyklu se vykoná obsah organizačního bloku OB1.

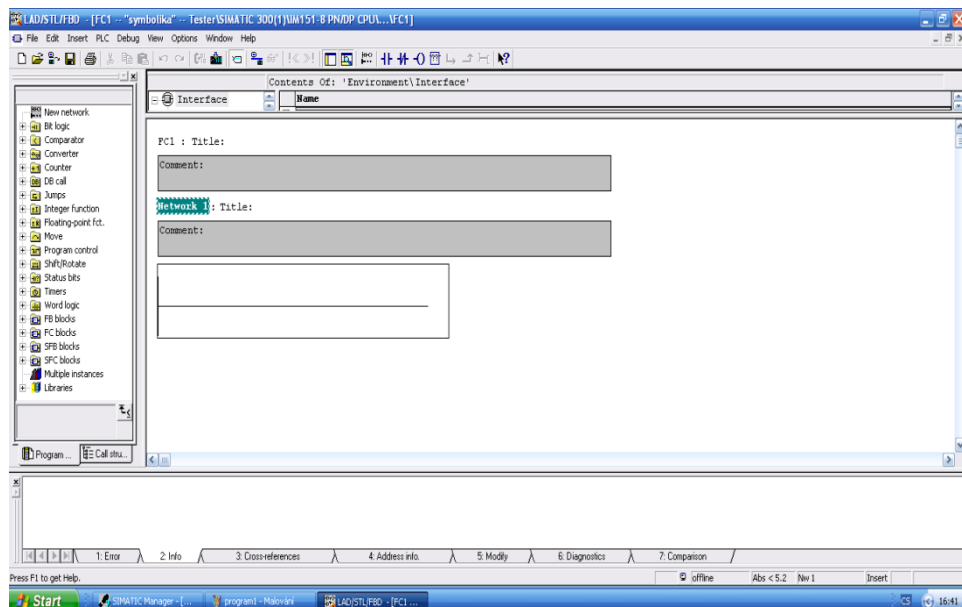
Pro program samotný slouží struktury nesoucí název funkce. Funkce jsou rozděleny na části zvané Networks. Každá tato síť slouží k pojetí několika logických úkonů, které tvoří logiku celého programu. K dispozici (jazyk LD) má programátor mnoho grafických prvků pro tvorbu řídicího SW. Jsou jimi bitové operace typu: aritmetické operace, komaprátory, konvertory, skoky, operace s proměnnými typu int, float, podmínky typu if - else, a tak dále.

Postup tvorby symbolického názvu proměnné v tabulce symbolů:

- 1) Klikneme na záložku Simatic 300, vybereme IM151-8 PN/DP, vybereme záložku S7 Program(1) a klikneme dvakrát na položku Symbols.
- 2) Na prázdné místo vložíme do položky Symbol: naseMalaPromenna, do položky Address: M0.0 a do položky Data type: BOOL. Klikneme na uložit.

Postup tvorby bloku či funkce (organizační, datový atd.):

- 1) Klikneme na záložku Simatic 300, vybereme IM151-8 PN/DP, vybereme záložku S7 Program(1) a klikneme pravým tlačítkem na Blocks.
- 2) Vybereme možnost Insert New Object. Vložíme konkrétní blok (organizační, datový atd.) či funkci.



Obrázek 19: Rozhraní pro programování ve funkci FC1

3.1.4 Datové bloky

Jednou z možností pro Insert New Object je Data Block, je to místo v paměti, kde uchováváme proměnné, jejichž obsah chceme zachovat i po vypnutí PLC. Je to tedy struktura obsahující data různých typů, například integer, která jsou definována adresou, názvem, typem a původní hodnotou. Údaj o původní hodnotě, nebo také initial value je hodnota, kterou proměnná představuje na začátku své existence. Tedy pokud dojde k poruše automatu nebo odstavení jeho provozu, proměnné ze struktury Data Block si udrží svoji aktuální hodnotu až do dalšího uvedení systému do provozu. Tyto proměnné se potom adresují opět přímo nebo symbolicky.

Postup vytvoření perzistentní proměnné typu integer:

- 1) Klikneme na záložku Simatic 300, vybereme IM151-8 PN/DP, vybereme záložku S7 Program(1) a klikneme pravým tlačítkem na Blocks.
- 2) Vybereme možnost Insert New Object. Vložíme Data Block.
- 3) Dvakrát klikneme na Data Block. Na implicitní adresu +0.0 klikneme pravým tlačítkem myši. Zvolíme Declaration Line after Selection.
- 4) Do nového pole vložíme do pole Name: naseVetsiPromenna, do pole Type: INT, do pole Initial value: 50.
- 5) Klikneme na uložit.

Tuto novou proměnnou lze adresovat přímo jako DB1.DBW2 nebo symbolicky jako datablok1.naseVetsiPromenna (kde datablok1 je symbol datového bloku v tabulce symbolů).

3.1.5 Tabulka proměnných

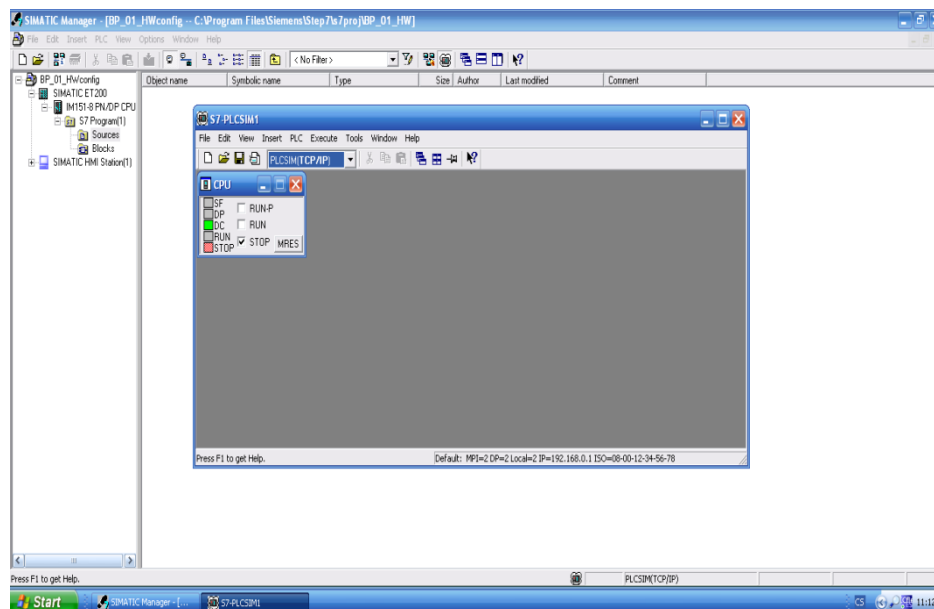
Další možností tlačítka Insert New Object v záložce Blocks je tabulka proměnných, také variable table. Tato tabulka je velmi důležitým monitorovacím nástrojem. Jakékoliv proměnné v PLC lze sledovat za provozu PLC v této tabulce. V Step7 lze přepínat mezi režimem online a offline. Online znamená přímo za běhu programu, offline znamená mimo běh programu (i bez PLC).

Postup vložení proměnné do tabulky proměnných:

- 1) Klikneme na záložku Simatic 300, vybereme IM151-8 PN/DP, vybereme záložku S7 Program(1) a klikneme pravým tlačítkem na Blocks.
- 2) Vybereme možnost Insert New Object. Vložíme Variable table.
- 3) Vyplníme přímou adresu proměnné, kterou chceme sledovat, tabulátorem vyplníme automaticky zbylé údaje.
- 4) Klikneme na uložit.

3.1.6 Simulátor S7-PLCSIM1

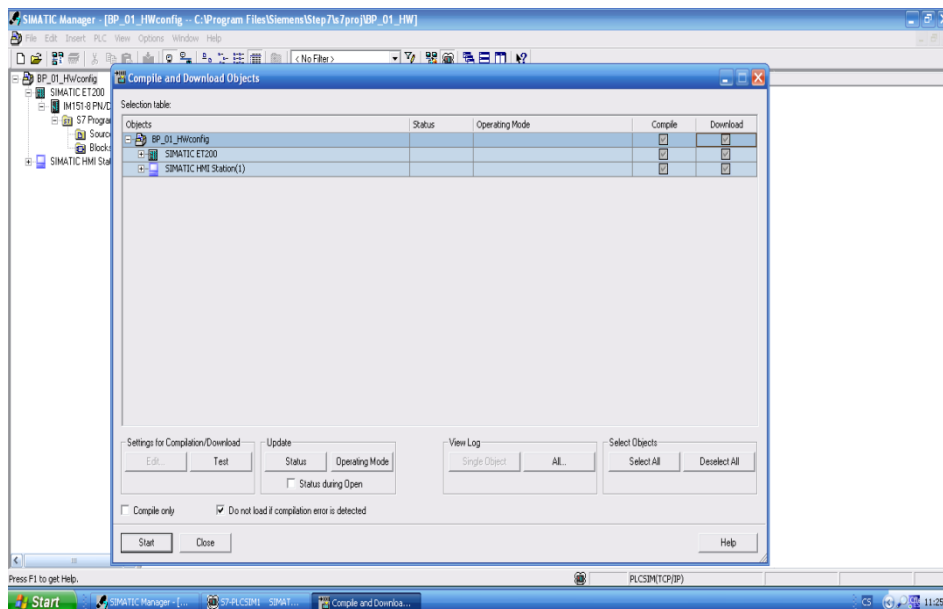
Velmi často se stává, že potřebujeme otestovat program (nebo monitorovat proměnné) ale nemáme k dispozici reálné PLC, Step7 nám v tomto případě nabídne simulované PLC s názvem S7-PLCSIM1. Simulátor nabízí možnosti simulace v režimu RUN či RUN-P (režim RUN-P dovoluje měnit za běhu části programu). Pro správnou funkčnost je nutné celý program zkompilovat a stáhnout do simulátoru PLC, stejně jako by to bylo reálné PLC.



Obrázek 20: Simulátor S7-PLCSIM1

Postup pro kompilaci a stažení programu do simulátoru S7-PLCSIM1:

- 1) V prostředí Step7 vybereme možnost Options a vybereme záložku Simulate Modules
- 2) V simulátoru zvolíme protokol TCP/IP.
- 3) Pravým tlačítkem klikneme v levém sloupci Step7 na název projektu a vybereme možnost PLC, dále vybereme možnost Compile and Download Objects.
- 4) V pravé části zvolíme Select All a poté možnost Start.
- 5) V simulátoru zvolíme možnost RUN (nebo RUN-P).



Obrázek 21: Compile and Download Objects v Step7

Postup pro monitorování proměnné v tabulce proměnných:

- 1) Otevřeme tabulku proměnných. Vložíme proměnnou na adrese M0.0 (naseMalaPromenna) do tabulky proměnných.
- 2) Klikneme na ikonu „brýlí“ zastupující funkci Monitor variable.
- 3) Sledujeme stav proměnné (jak se mění za běhu programu).
- 4) Klikneme pravým tlačítkem na proměnnou a zvolíme možnost Modify Address to 0 respektive Modify Address to 1. (tím změníme záměrně hodnotu proměnné).

	Address	Symbol	Display format	Status value	Modify value
1	M 0.0	"m_off"	BOOL	false	
2	M 0.1	"m_on"	BOOL	true	
3	M 2.0	"w_button"	BOOL	false	
4	M 3.0	"C_button"	BOOL	false	
5	M 4.0	"T_button"	BOOL	false	
6	M 2.1	"W_NULL"	BOOL	false	
7	M 3.1	"C_NULL"	BOOL	false	
8	M 4.1	"T_NULL"	BOOL	false	
9	M 255.0		BOOL	true	
10	M 255.1		BOOL	false	
11	M 255.2		BOOL	true	
12	M 255.3		BOOL	true	
13	M 255.4		BOOL	true	
14	M 255.5		BOOL	false	
15	M 255.6		BOOL	false	
16	M 255.7	"CasovacONLINE"	BOOL	true	
17					

Obrázek 22: Monitorování proměnných v režimu online

Veškeré výše popsané metody a možnosti prostředí Step7 jsou nástroje potřebné k programování PLC Siemens v prostředí Step7. Při programování samotném je již čistě na programátorovi, jak daný SW vytvoří, na co všechno při vývoji pomyslí a co zahrne do kódu. Následující část se bude věnovat prostředí WinCC Flexible 2008, které je nástrojem pro programování Simatic HMI Station. Tedy nástrojem pro vizualizaci celého projektu pomocí dotykových panelů Siemens.

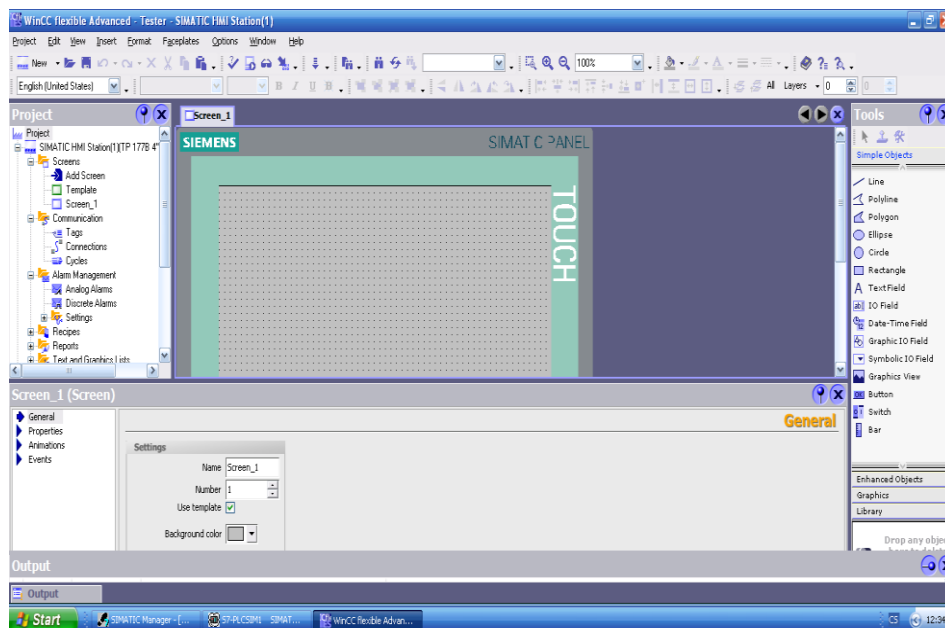
3.2 Vývojové prostředí WinCC Flexible 2008

Prostředí WinCC Flexible 2008 slouží k programování vizualizace projektu na operátorských panelech společnosti Siemens. Samotné programování je spíše záležitostí pečlivosti, než logického myšlení. Probíhá na grafické bázi, stylem, který připomíná přetahování prvků a spojování prvků s proměnnými v projektu v Step7. V rámci vizualizace má programátor k dispozici prvky jako: základní geometrické útvary, ovládací a informační prvky, jako jsou textová pole, vstupně – výstupní pole, tlačítka, přepínače nebo lišty s datem a časem. Důležitou podmínkou k tomu, aby s proměnnou mohlo být ve vizualizaci manipulováno je její obsažení v tabulce symbolů. Ve Vizualizaci se tedy adresuje pouze symbolicky, nikoliv přímo. Pro vizualizaci zde práce nabídne modelový příklad, který bude obsahovat: tlačítko pro ovládání stavu proměnné M0.0 (naseMalaPromenna). Dále obdélník, který poslouží jako signalizace stavu této proměnné. Dále obdélník, který znázorní přepínání časovače.

Postup pro otevření WinCC Flexible 2008:

- 1) V Step7 klikneme v levé části na záložku Simatic HMI Station(1).
- 2) Vybereme záložku WinCC Flexible RT, poté Screens.
- 3) Klikneme dvakrát na Screen_1.

Nově otevřené okno nese název WinCC Flexible Advanced – názevProjektu – SIMATIC HMI Station(1). Již zmiňované prvky se nachází vpravo na liště zvané Tools.

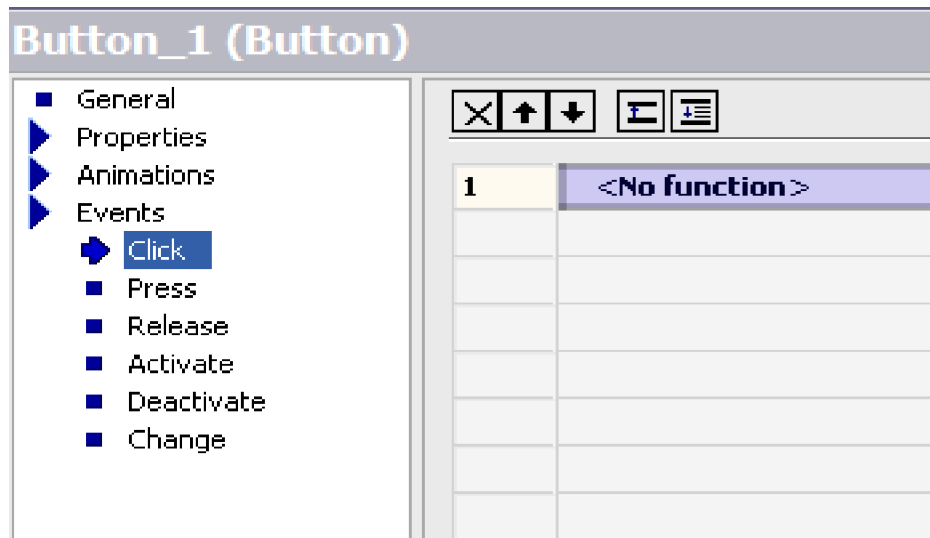


Obrázek 23: Prostředí WinCC Flexible 2008

Postup pro vložení a naprogramování tlačítka:

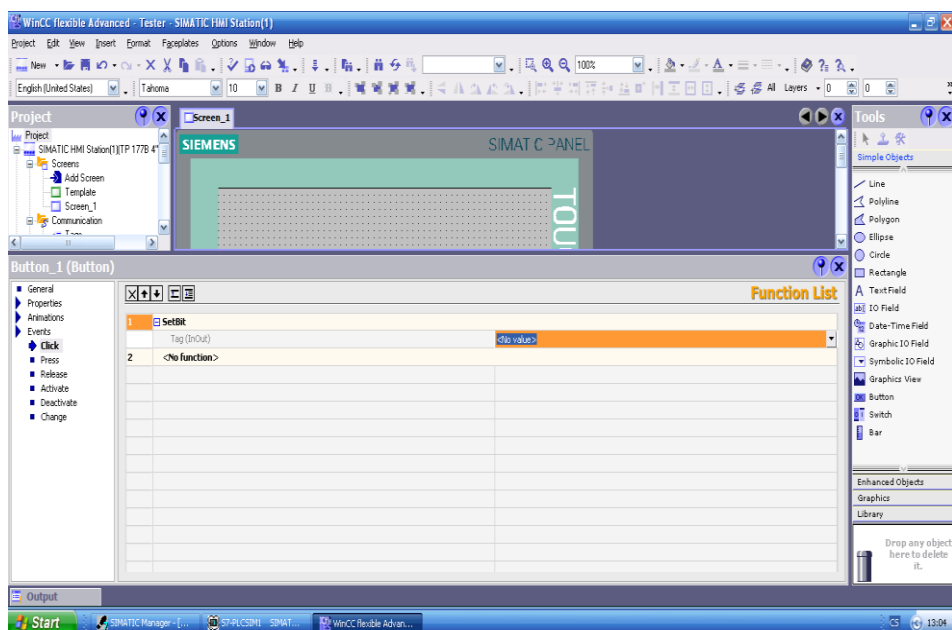
- 1) z lišty Tools přetáhneme do prostoru Screen_1 prvek s názvem button.
- 2) Klikneme na prvek Button_1 levým tlačítkem a přesuneme se do záložky Events.
- 3) Vybereme možnost Click.
- 4) Uprostřed nastavení tlačítka Button_1 zvolíme pro tlačítko funkci Edit bits, poté možnost SetBit.
- 5) V záložce Tag zvolíme proměnnou v projektu Step7 s názvem naseMalaPromenna.

Pro prostředek tlačítko lze nastavovat funkčnost v následujících položkách: General, Properties, Animations, Events. V záložkách Animation a Events lze naprogramovat chování a vzhled tlačítka v konkrétní situaci.



Obrázek 24: Nabídka Events v prostředí button

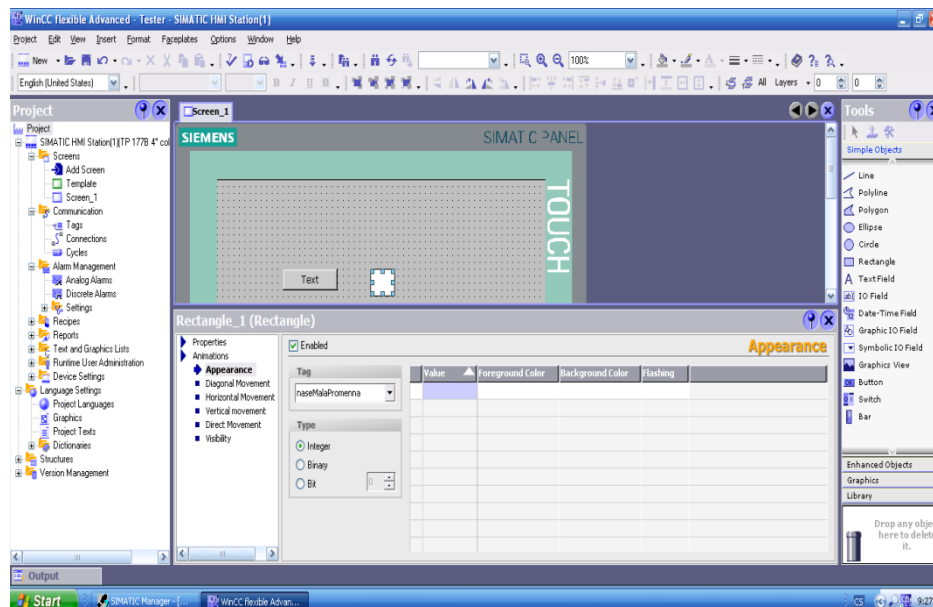
Tlačítko Button_1 tedy provede konkrétní akci, a to na základě spouštěcího mechanismu Click, Press, Release, Activate, Deactivate nebo Change. Veškeré mechanismy jsou ekvivalentní s českými překlady. Tedy klik, podržení, uvolnění, aktivace, deaktivace, změna. V našem případě je spouštěcím mechanismem kliknutí a akcí je editace bitu na hodnotu 1.



Obrázek 25: SetBit ve WinCC Flexible 2008

Postup nastavení prvku obdélník jako signalizace stavu:

- 1) Z lišty Tools přetáhneme do prostoru stanice prvek Rectangle.
- 2) Levým tlačítkem klikneme na obdélník a přejdeme do záložky Animations, poté do záložky Appearance.
- 3) V záložce Tag vybereme proměnnou s názvem naseMalaPromenna.
- 4) Na první řádek vložíme do políčka Value hodnotu 0. Barvu pro Background color a Foreground color zvolíme červenou.
- 5) Na druhý řádek vložíme do políčka Value hodnotu 1. Barvu pro Background color a Foreground color zvolíme zelenou.



Obrázek 26: Nastavení záložky Appearance v rámci prvku Rectangle

Nyní máme nastaveno tlačítko i signalizaci. Kdybychom měli udělat malé shrnutí vizualizace. Kliknutím na tlačítko nastavíme proměnnou naseMalaPromenna na hodnotu jedna. Na displeji bude svítit obdélník jako stav proměnné naseMalaPromenna. Ve stavu logická 0 bude signalizace červená, ve stavu logická 1 bude signalizace zelená.

Do projektu vizualizace přidáme další prvek, kterým bude onen časovač, nastavený v rámci HW konfigurace. Byl nastaven v rámci HW Config na bit 255.0 – 255.7.

Postup nastavení obdélníku jako signalizace časovače:

- 1) Z lišty Tools přetáhneme do prostoru stanice prvek Rectangle.
- 2) V prostředí Step7 vložíme do tabulky symbolů bit 255.4 se symbolickým názvem casovacVisual (nutnost pro práci ve vizualizaci!).
- 3) Levým tlačítkem klikneme na obdélník a přejdeme do záložky Animations, poté do záložky Appearance.
- 4) V záložce Tag vybereme proměnnou s názvem casovacVisual.
- 5) Na první řádek vložíme do políčka Value hodnotu 0. Barvu pro Background color a Foreground color zvolíme červenou.
- 6) Na druhý řádek vložíme do políčka Value hodnotu 1. Barvu pro Background color a Foreground color zvolíme zelenou.

Co se týká vizualizace, je zde jeden problém, na který dříve či později narazíte. Jedná se o obnovovací frekvenci stavu proměnné na operátorském panelu. Pro některé proměnné je nutné nastavit dostatečně vysokou obnovovací frekvenci, aby byla na panelu zobrazena každá změna proměnné. Implicitně je nastavena tato frekvence na hodnotu 1s, pokud bychom v tuto chvíli simulovali vizualizaci, zjistili bychom, že proměnná časovače se nestihá řádně obnovovat. Vypadalo by to tedy, že se přepíná v nepravidelném intervalu.

Postup nastavení obnovovací frekvence:

- 1) Ve WinCC Flexible zvolíme v levém sloupci záložku Communication, dále vybereme možnost Tags.
- 2) Pro proměnnou casovacVisual zvolíme Acquisition cycle jako 100 ms.

Acquisition cycle ovšem nastavujte střízlivě, při velkém množství proměnných a málo výkonném HW by mohlo nastat k prodloužení celkové odezvy operátorského panelu, který by se tímto stal nefunkční. Proto nastavujte vysokou obnovovací frekvenci pouze tam, kde je to opravdu nutné.

Postup pro simulaci vizualizace:

- 1) Otevřeme simulátor S7-PLC1M1.
- 2) Nahrajeme celý projekt do simulátoru.
- 3) spustíme simulátor zvolením možnosti RUN (nebo RUN-P).
- 4) Ve WinCC Flexible 2008 zvolíme Project, poté Compiler, poté možnost Rebuild All.
- 5) Zvolíme Project, poté Compiler, poté Start Runtime

Díky kapitole 3.1 a 3.2 je možné si osvojit základní dovednosti týkající se programování PLC Siemens Simatic ve vývojovém prostředí Step7 a WinCC Flexible 2008. V následující části se nachází zadání samostatných úkolů, tedy projektů v rámci logických automatů společnosti Siemens. Slouží jako cvičení k výukovému materiálu.

4 Zadání samostatných projektů

4.1 Zadání projektu nápojového automatu

Zadání:

Cílem samostatného projektu je pomocí prostředí Step7 vytvořit program pro obsluhu nápojového automatu. Součástí projektu bude i vizualizace, pomocí které budeme automat ovládat. Celé řešení bude odzkoušeno na vybavení laboratoře automatizace KŘP na FEI.

Nutné podmínky:

Nápojový automat musí obsahovat: minimálně 3 položky k prodeji, indikátor vložených peněz, informace o počtu položek, informace o ceně položek, systém vracení peněz, systém doplňování zboží.

Nutné přílohy projektu:

Vývojové diagramy hlavních částí programu. Popis funkce nápojového automatu a sestavených algoritmů jednotlivých částí projektu.

V projektu použijte organizační bloky, funkce, datové bloky, tabulku symbolů, tabulku proměnných, libovolné další, potřebné prvky. Programovací jazyk zvolte libovolný.

4.2 Zadání projektu ovládání vrat

Zadání:

Cílem samostatného projektu je pomocí prostředí Step7 vytvořit program pro obsluhu garážových vrat. Součástí projektu bude i vizualizace, pomocí které budeme automat ovládat. Celé řešení bude odzkoušeno na vybavení laboratoře automatizace KŘP na FEI.

Nutné podmínky:

Ovládání vrat musí obsahovat: systém zvedání a spouštění, bezpečnostní vypínač.

Nutné přílohy projektu:

Vývojové diagramy hlavních částí programu. Popis funkce garážových vrat a sestavených algoritmů jednotlivých částí projektu.

V projektu použijte organizační bloky, funkce, tabulku symbolů, tabulku proměnných, libovolné další, potřebné prvky. Programovací jazyk zvolte libovolný.

5 Závěr

V celé práci, v rámci teoretické části, byly shrnuty základy teorie řízení a práce s logickými operátory. Teoretický úvod práce je nezbytným minimem, potřebným pro osvojení programování logických automatů. Dále jsme v praktické části nastínili základní funkce a možnosti práce v prostředích Step7 a WinCC Flexible 2008. Hlavní kroky v programování PLC pomocí prostředí Step7 jsou tedy HW konfigurace pomocí nástroje HW Config, dále nastavení síťové komunikace pomocí nástroje NetPro. Byl také probrán simulator S7-PLCSIM1. Díky bakalářské práci je tedy možné realizovat základní řídicí programy pro PLC automaty Simatic společnosti Siemens a navrhnout celý PLC systém od návrhu komponentů, po jejich zakoupení podle katalogového čísla. Je také možné podle práci vytvořit SW pro operátorské panely HMI Station, taktéž společnosti Siemens.

Práci na toto téma jsem si zvolil čistě z profesního zájmu. Do budoucna vynaložím úsilí k zvládnutí problematiky programování PLC. Logické automaty jsou velmi efektivním nástrojem pro automatizaci jakýchkoliv průmyslových procesů. Počínaje chemickým, potravinářským průmyslem a strojírenskou výrobou konče. Protože jsou PLC systémy velmi rozšířeným prostředkem automatického řízení, byla tato práce zpracována, jako podpůrný studijní materiál. S PLC automaty by se studenti automatizace měli dle mého názoru setkat již na bakalářském stupni studia, proto tato práce vznikla také jako krok směrem k ovládnutí problematiky. Již v rané fázi studia na VŠ.

Použitá literatura

- [1] ŠMEJKAL, Ladislav a Marie MARTINÁSKOVÁ. *PLC a automatizace*. 1. vyd. Praha: BEN - technická literatura, 1999, 223 s. ISBN 80-860-5658-9.
- [2] DOSTÁL, Petr a František GAZDOŠ. *Učební texty vysokých škol: Řízení technologických procesů*. Zlín. [cit. 2013-04-26]. Dostupné z: <http://www.uloz.to/xsKFeoc/rizeni-technologickych-procesu-scripta-utb-pdf>
- [3] *Mechatronické systémy - EL. přístroje: Programovatelné automaty*. [cit. 2013-04-26]. Dostupné z: http://homen.vsb.cz/~hav278/MS/Studijni_opory/Programovatelne_automaty.pdf
- [4] SIEMENS. *Industry Mall* [online]. [cit. 2013-04-26]. Dostupné z: <https://eb.automation.siemens.com/mall/en/WW/Catalog/Products/5000015?tree=CatalogTree>
- [5] MICROPEL: *Instalační rámečky pro PLC PCM300*. [online]. [cit. 2013-04-26]. Dostupné z: http://www.micropel.cz/index.php?option=com_content&view=article&id=135&Itemid=12&lang=cs&karta=3
- [6] *Kombinační logické systémy* [online]. [cit. 2013-04-26] Dostupné z: http://lucy.troja.mff.cuni.cz/~tichy/elektronika/kap6/6_3.html
- [7] URBAN, Luboš. *Programování PLC podle normy IEC EN 61131-3 – víc než jednotné jazyky*. In: www.odbornecasopisy.cz [online]. FCC Public. [cit. 2013-04-26]. Dostupné z: http://www.odbornecasopisy.cz/index.php?id_document=30310
- [8] *Logic to Ladder Diagram* [online]. [cit. 2013-04-26]. Dostupné z: <http://program-plc.blogspot.cz/2010/02/logic-ladder-diagram.html>
- [9] *Engineer On A Disk: 20FUNCTION BLOCK PROGRAMMING* [online]. [cit. 2013-04-26]. Dostupné z: http://engineeronadisk.com/V2/book_PLC/engineeronadisk-148.html
- [10] *Beckhoff Information System* [online]. [cit. 2013-04-26]. Dostupné z: http://infosys.beckhoff.com/english.php?content=../content/1033/cx8090/html/cx8000_firststeps.htm&id=819

Přílohy

Příloha A Vzorové řešení projektu nápojového automatu

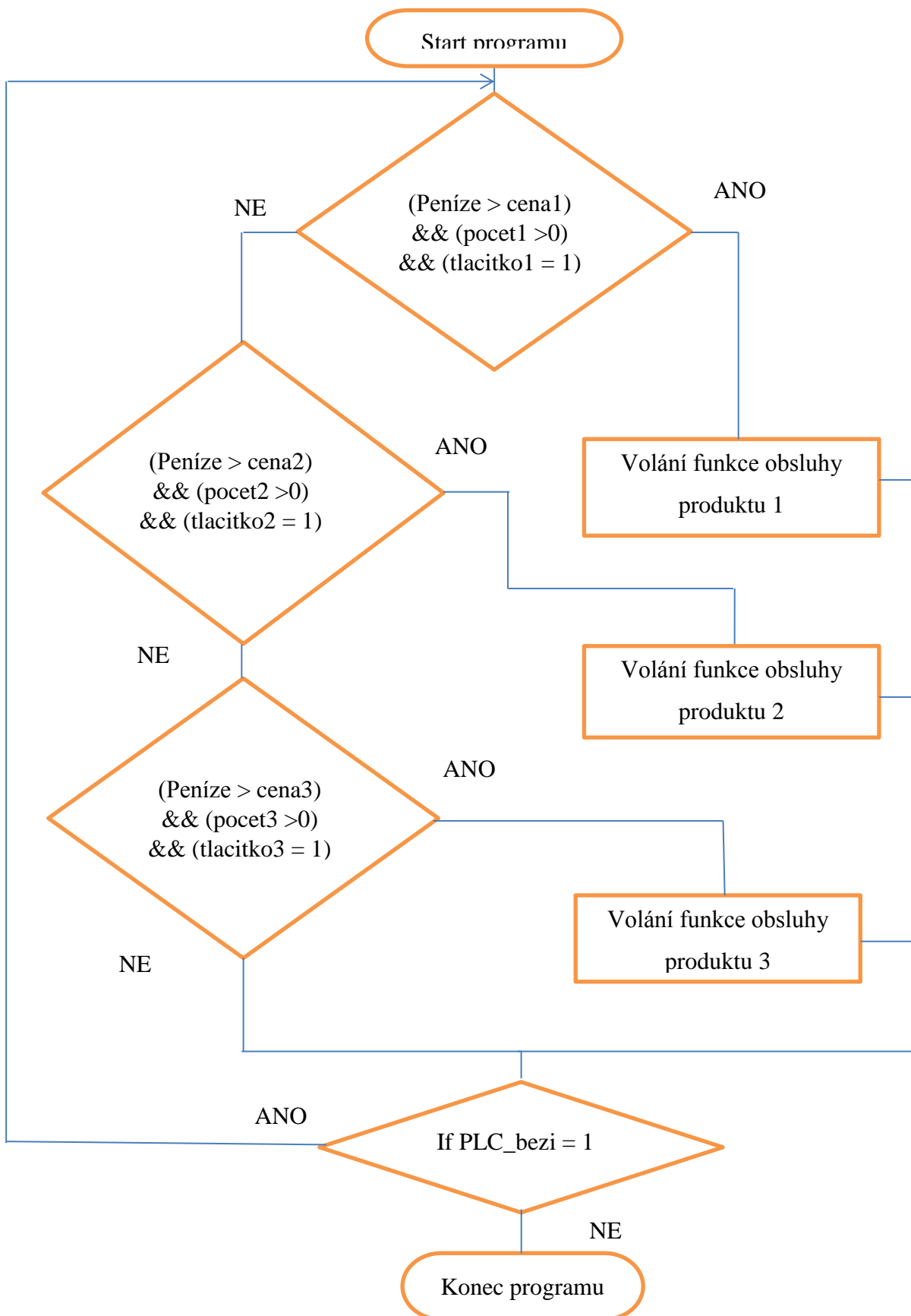
K řešení projektu byl pro programování použit jazyk LD. V prostředí Step7 byly k naprogramování zadaného projektu použity následující prvky: OB100, OB1, FC1, FC2, FC3, FC4, DB1, VAT1, Symbols (tabulka symbolů).

OB100 – blok OB100 se zavolá pouze jednou, a to na začátku práce PLC automatu. Obsahuje kontrolní proměnné M0.0 a M0.1, které hlídají, zda je automat aktivní.

OB1 – blok OB1 je využit pro neustálé periodické volání funkce FC1.

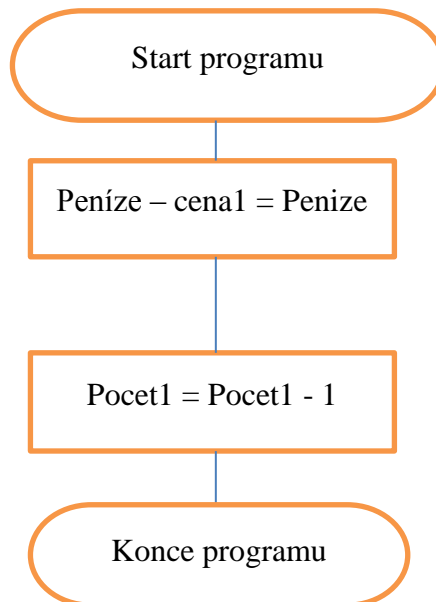
FC1 – Funkce FC1 obsahuje určité podmínky, při jejichž splnění dojde ke skoku do jedné z následujících funkcí: FC2, FC3, FC4.

OB1 – Cyklický blok. Obsahuje FC1, což je cyklická funkce programu, které hlídá spouštěče v automatu. Spouštěče jsou stavy, které přimějí automat vykonat vydání určitého produktu. Je to tedy soubor podmínek, které musí platit.



Obrázek 27: Zjednodušené schéma algoritmu řízení

FC2, FC3, FC4 – Tyto tři funkční bloky řídí samotné vykonání vydání položky a provádí matematickou režiji. Tedy výpočty vložených peněz, vrácených peněz a ceny zboží.

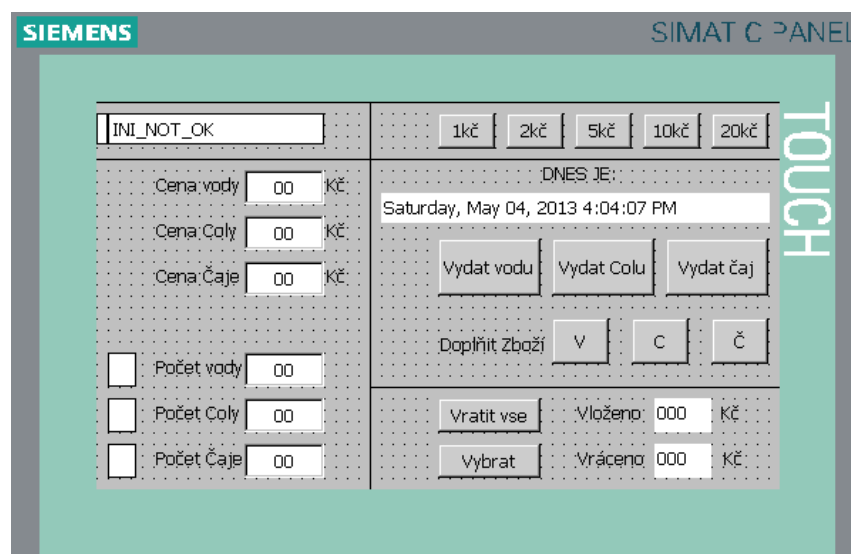


Obrázek 28: Zjednodušené schéma algoritmu řízení funkce FC2

DB1 – Datový blok 1 je blok obsahující proměnné typu integer, který uchovává informaci o počtu položek a jejich ceně, dále o penězích vložených a vrácených.

VAT1 – Je tabulka obsahující proměnné, určené k monitorování.

Zbylé ovládací prvky jsou řešeny pomocí operátorského panelu. Prvky jsou rozmístěny podle obrázku 29.



Obrázek 29: Podoba vizualizace ve vzorovém řešení