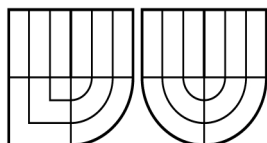


**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA STROJNÍHO INŽENÝRSTVÍ**  
**ÚSTAV AUTOMATIZACE A INFORMATIKY**



**FACULTY OF MECHANICAL ENGINEERING**  
**INSTITUTE OF AUTOMATION AND INFORMATION**

**GENEROVÁNÍ ČASOVÝCH PRŮBĚHŮ**  
**PROGRAMOVATELNÝM AUTOMATEM**  
**(BAKALÁŘSKÁ PRÁCE)**

**GENERATING TIME BEHAVIOUR BY THE HELP OF**  
**PROGRAMMABLE LOGIC CONTROLLER**  
**(BACHELOR'S THESIS)**

**AUTOR PRÁCE:**  
AUTHOR

**Michal Novák**

**VEDOUcí PRÁCE:**  
SUPERVISOR

**doc. Ing. Zdeněk Němec, CSc.**

Brno, říjen 2009

# Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav automatizace a informatiky  
Akademický rok: 2008/2009

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

student(ka): **Michal Novák**  
který/která studuje v **bakalářském studijním programu**  
obor: **Aplikovaná informatika a řízení (3902R001)**

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

### **Generování časových průběhů programovatelným automatem**

v anglickém jazyce:

#### **Generating time behaviour by the help of programmable logic controller**

Stručná charakteristika problematiky úkolu:

V některých úlohách řízení s programovatelným automatem je třeba generovat různé časové průběhy, například sinusového, trojúhelníkového, impulsního a obecně zadaného tvaru. Pro tento účel je žádoucí zpracovat programy obecnějšího použití, které by byly využitelné v aplikacích s automaty Simatic C7 a v programovém prostředí Step 7.

Cíle bakalářské práce:

1. Seznámit se s daným programovatelným automatem Simatic C7-635, jeho vývojovým prostředím a programovacím jazykem Step 7.
2. Navrhnout programy, které by generovaly časové průběhy alespoň sinusového, trojúhelníkového a impulsního tvaru.
3. Styk obsluhy s daným automatem řešit prostřednictvím tlačítek a obrazovky automatu. To se týká hlavně ovládání generátoru, volby druhu časového průběhu a nastavení velikosti amplitud a kmitočtu.
4. Osciloskopickým měřením ověřit generované časové průběhy, zjistit omezující podmínky a shrnout poznatky z aplikace automatu pro obdobné účely.

Seznam odborné literatury:

[1] Švarc, I.:Automatizace-Automatické řízení. Brno: CERM, 2005.

[2] Firemní dokumentace od fy Siemens k programovatelným automatům.

Vedoucí bakalářské práce: doc. Ing. Zdeněk Němec, CSc.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2008/2009.

V Brně, dne: 16.10.2009

L.S.

---

**doc. RNDr. Ing. Miloš Šeda, Ph.D.**  
**Ředitel ústavu**

---

**doc. RNDr. Miroslav Doupovec, CSc.**  
**Děkan fakulty**

# LICENČNÍ SMLOUVA POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

## 1. Pan / paní

Jméno a příjmení: Michal Novák  
Bytem: v Lesné 55, Želetava 675 26  
Narozen/a (datum a místo): 02.10.1983 v Třebíčí

(dále jen „autor“)

a

## 2. Vysoké učení technické v Brně

Fakulta strojního inženýrství  
se sídlem Technická 2896/2, 616 69 Brno  
jejímž jménem jedná na základě písemného pověření děkan fakulty:  
doc. RNDr. Miroslav Doupovec, CSc.

(dále jen „nabyvatel“)

## Článek 1 Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikace práce (VŠKP)\*:

- disertační práce
- diplomová práce
- bakalářská práce
- jiná práce, jejíž druh je specifikován jako.....

(dále jen VŠKP nebo dílo)

Název VŠKP: Generování časových průběhů programovatelným automatem  
Vedoucí / školitel VŠKP: doc. Ing. Zdeněk Němec, CSc.  
Ústav: Ústav automatizace a informatiky  
Datum obhajoby VŠKP: 04.11.2009

VŠKP odevzdal autor nabyvateli v\*:

- |   |   |                 |    |
|---|---|-----------------|----|
| <input type="checkbox"/> tištěné formě      | – | počet exemplářů | 2x |
| <input type="checkbox"/> elektronické formě | – | počet exemplářů | 2x |

---

\* hodící se zaškrtněte

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané specifikované. Autor dále prohlašuje, že při zpracování díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze je identická.

## **Článek 2**

### **Udělení licenčního oprávnění**

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnožení.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
  - ihned po uzavření této smlouvy
  - 1 rok po uzavření této smlouvy
  - 3 roky po uzavření této smlouvy
  - 5 let po uzavření této smlouvy
  - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č.111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

## **Článek 3**

### **Závěrečná ustanovení**

1. Smlouva je sepsána ve čtyřech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: \_\_\_\_\_

\_\_\_\_\_  
Nabyvatel

\_\_\_\_\_  
Autor

## **Anotace**

V této bakalářské práci je popsán laboratorní proces generování časových průběhů programovatelným automatem Simatic C7. Ovládání a zobrazování důležitých informací je realizováno pomocí operátorského panelu. Práce popisuje proces programování v programovém prostředí Step 7 a konfiguraci aplikace stylem krok za krokem.

## **Annotation**

In this bachelor thesis The laboratory process of generating time processing is described by a programmable Simatic C7. Maschine controlling and displaying of important information is implemented by an operator panel. This bachelor thesis describes the process of programming in the Step 7 programming environment and The Step by Step configuration of the applicaton.

## **Poděkování**

Na tomto místě bych rád poděkoval panu doc. Ing. Zdeňkovi Němcovi, CSc. za jeho odborné rady při realizaci této bakalářské práce.

## **Obsah**

<b>1. Úvod</b> .....	<b>9</b>
<b>2. Programovatelné automaty</b> .....	<b>10</b>
2.1 Bosch Rexroth .....	11
2.2 ABB .....	12
2.3 Tecomat .....	13
<b>3. Simatic C7 – 635</b> .....	<b>14</b>
3.1 Technické parametry .....	16
3.2 Komunikace .....	17
3.2.1 RS 232 .....	17
3.2.2 MPI .....	18
3.2.3 Profibus DP .....	19
<b>4. Realizace projektu</b> .....	<b>20</b>
4.1 Zapojení pracoviště .....	20
4.2 Použitý hardware .....	20
4.2.1 Simatic C7 – 635 .....	20
4.2.2 PC Adapter .....	21
4.2.3 LOGO! Power .....	21
4.3 Použitý software .....	22
4.3.1 Step 7 v 5.2 .....	22
4.3.2 ProTool Lite v 6.0 .....	23
4.4 Grafická část projektu .....	24
4.5 Řídící část projektu .....	32
4.5.1 Konfigurace hardware – HW Config .....	35
4.5.2 Konfigurace sítě MPI – NetPro .....	37
4.5.3 Problémy při konfiguraci .....	37
4.5.4 Řídící program programovatelného automatu .....	38
4.5.5 Popis jednotlivých částí programů .....	40
4.5.6 Výsledná měření .....	43
<b>5. Závěr</b> .....	<b>45</b>
<b>Seznam použité literatury</b> .....	<b>46</b>
<b>Přílohy</b> .....	<b>49</b>
<b>Obsah přiloženého DVD</b> .....	<b>57</b>

## 1. Úvod

V dnešní době plné automatizačních řídicích systému, které bývají obsaženy ve spoustě průmyslových strojů může dojít k problému, že bude za potřebí: generovat různé časové průběhy, například sinusového, impulsního nebo trojúhelníkového tvaru. V tomto případě přichází na řadu snadné, rychlé a levné řešení, v podání naprogramování již zabudovaného řídicího systému např: Simatic C7, který je schopen generování požadovaných průběhu, za určitých omezujících podmínek jako je například nízká frekvence. Vyhneme se tak instalaci dalších zařízení jako jsou generátory impulsů.

Cílem této bakalářské práce je laboratorní ukázka generování požadovaných průběhů s nastavitelnou amplitudou výstupního napětí a frekvence pomocí programovatelného automatu SIMATIC C7. Volba časových průběhů a nastavení požadovaných hodnot řešena přes funkční klávesy operátorského panelu OP 170B. Dalším úkolem je zjištění omezujících podmínek, za kterých je D/A převodník schopen generování průběhu a ověření správnosti pomocí měření osciloskopu.

Celá práce je rozdělena do čtyř částí. V první části je rozdělení a popis programovatelných automatů. Další část je zaměřena na programovatelný automat SIMATIC C7, včetně technických parametrů a na možnost komunikace s tímto PA. Třetí stěžejní kapitola se zabývá samotnou realizací aplikace. Sestavením použitých komponent, konfigurací aplikace a tvorbou řídicích a grafických programů. V poslední části je obsaženo zhodnocení celého projektu.



## 2. Programovatelné automaty

Programovatelné automaty jsou uživatelsky programovatelné řídicí systémy a v dnešní době nejčastěji využívanými prvky v současné automatizační technice. Důvodem jejich vzniku byla malá odolnost počítačů v průmyslovém prostředí. Jsou odolnější vůči vibracím, prachu, vlhkosti, výkyvům napětí a elektromagnetickému poli. Většina řídicích systémů v průmyslu se bez nich neobejde. Nejrozšířenějším označením programovatelných automatů je zkratka PLC<sup>1</sup>. V české literatuře se hojně používá zkratka PA<sup>2</sup>. Občas se můžeme setkat také s označením PC<sup>3</sup>, nebo v německy psané literatuře SPS<sup>4</sup>.

Původně byly navrženy k řešení logických úloh jako náhrada tehdejší pevné reléové logiky, kde funkce byla dána propojením relé, tranzistorů nebo integrovaných obvodů. Činnost PA<sup>2</sup> je dána cyklickým programem, který je naprogramován uživatelem pomocí různých programovacích jazyků a uložen do uživatelské paměti programovatelného automatu. Program se dá zaměňovat, kdežto u reléových schémat byla funkce určena zapojením jednotlivých komponent a skoro prakticky nezaměnitelná.

Jejich největší výhodou je, že mají stejný HW<sup>5</sup> pro spoustu jiných aplikací, což má obrovské výhody při jejich výrobě (sériová výroba) a tím samozřejmě nižší cenu. Každý programovatelný automat se skládá z procesoru, systémové paměti, uživatelské paměti a ze souboru vstupních a výstupních periférií (DI, DO, AI, AO)<sup>6</sup>, ke kterým lze připojit různá výrobní zařízení, stroje a soubor komunikačních jednotek pro komunikaci s nadřazenými řídicími systémy. Všechny tyto části programovatelného automatu jsou navzájem propojeny systémovou sběrnici.

Programovatelné automaty lze dělit podle flexibility vstupů, výstupů a speciálních funkcí:

- Kompaktní
- Modulární (stavebnicové)

Malé systémy bývají řešeny jako kompaktní s pevným počtem vstupů, výstupů a rozsahem operační paměti. Modulární velké systémy bývají označovány také jako stavebnicové. Vstupy a výstupy se sdružují do modulů, kde jejich počet bývá volitelný. Systém programování je u obou variant stejný liší se odlišnou konstrukcí a uživatelským rozhraním.

Mezi nejvýznamnějšími výrobce programovatelných automatů na trhu patří firmy:

- Siemens
- Bosch Rexroth
- ABB
- Mitsubishi
- Omron
- GE – Fanuc
- Teco Kolín (český výrobce)

---

PLC<sup>1</sup> – Programmable Logic Controller

PA<sup>2</sup> – Programovatelný Automat

PC<sup>3</sup> – Programmable Controller

SPS<sup>4</sup> – Speicher Programmierte Steuerung

HW<sup>5</sup> – Hardware

DI<sup>6</sup> – Digital Input (číslicový vstup)

DO<sup>6</sup> – Digital Output (číslicový výstup)

AI<sup>6</sup> – Analog Input (analogový vstup)

## 2.1 Bosch Rexroth

Bosch Rexroth nabízí kompletní řešení řídicích a automatizačních systémů jejich součástí jsou programovatelné automaty. Vyrábějí se s označením CL<sup>1</sup> v několika řadách. Odlišné jsou pouze výkonem a oblastí použití.

Hojně používaným základním modelem je programovatelný automat CL15\* který obsahuje 64kB paměť RAM<sup>2</sup>, 64kB Flash EPROM<sup>3</sup>, zálohovou baterii, 8 nebo 16 integrovaných vstupů, 8 integrovaných výstupů a 2x32bitový čítač. S použitím dalších karet lze automat připojit na průmyslovou sběrnici a nebo rozšířit o další vstupy a výstupy (digitální nebo analogové). CL15\* může obsluhovat až 64 programů nezávisle na sobě.

Výkonnějším programovatelným automatem je CL200 modulární konstrukce. Základem je deska s napájením a procesorová karta na kterou lze připojit další vstupy, výstupy nebo průmyslové sběrnice. CL200 obsahuje 128kB RAM, 128kB Flash EPROM. Velkou výhodou tohoto automatu je že k němu můžeme připojit karty pohonu, polohy pro krokové motory a lze ho pak využít k řízení různých manipulátorů. Dalšími výkonnějšími automaty od firmy Bosch Rexroth jsou CL350, -400, -500, 550.



Obr. 1 Programovatelný automat řady CL200

---

CL<sup>1</sup> – Controller Logic  
RAM<sup>2</sup> – Random Access Memory

## 2.2 ABB

Firma ABB nabízí na trhu širokou škálu programovatelných automatů například AC31, AC500 a jejich následné řady (AC31 07CR41, 07KR51...). Pro komunikaci mezi člověkem a strojem nabízí spoustu produktů od displejů pro zobrazování textu po dotykové panely s barevným displejem.

Konfigurace a programování těchto programovatelných automatů je rychlé a snadné díky stejnému software a stejným příkazům pro všechny přístroje. Propojení všech zařízení je realizováno jednoduše po RS232<sup>2</sup> a pro složitější aplikace je použit Ethernet, Modbus nebo Profibus – DP. Programovatelné automaty AC31 řady (40...50) obsahují v základním provedení 32kB paměť RAM, 32kB Flash EPROM, 8 integrovaných digitálních vstupů, 6 integrovaných digitálních výstupů a dále sériové porty RS232 a rozhraní RS485.



Obr. 2 Programovatelný automat AC31 série 40...50



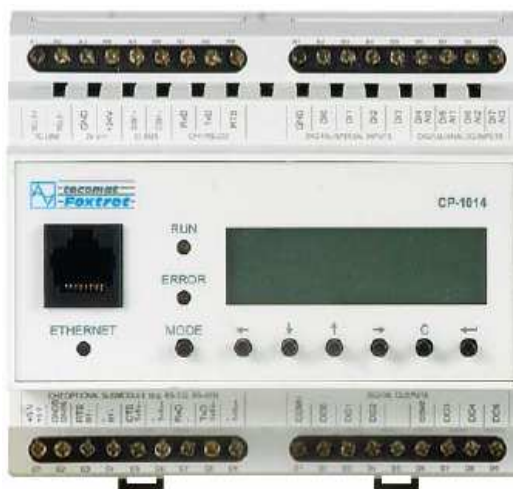
Obr. 3 Programovatelný automat AC500 - eCo

### 2.3 Tecomat

Programovatelný automat Tecomat nabízí na trhu česká firma TECO Kolín. Na trhu disponuje velkou řadou produktů od malých kompaktních systémů TC400 přes větší TC500, TC600, TC650 až po velké aplikace, pro které jsou určeny modulární systémy TC700.

Novinkou na trhu je malý modulární programovatelný automat řady Tecomat Foxtrot s rychlým Ethernetem a s integrovaným webserverem. Foxtrot disponuje až 134-mi digitálními vstupy a výstupy, 80-ti analogovými vstupy, 20-ti analogovými výstupy, 2-mi sériovými porty, Ethernetem a pamětí programu o velikosti 192 + 64kB.

Všechny dostupné Tecomat automaty se programují ve vývojovém prostředí Mosaic. Programovacím jazykem je firemní mnemokód typu IL<sup>1</sup>. Je tím zaručena kompatibilita programů a přenositelnost kódu. U nových programovatelných systémů Tecomat s 32 bitovými procesory je možno psát program ve formě ST<sup>2</sup>, což má za následek zpřehlednění programu a využití podpůrných nástrojů, které nebyly u starších typů podporovány.



Obr. 4 Programovatelný automat Tecomat Foxtrot

IL<sup>1</sup> – Instruction List

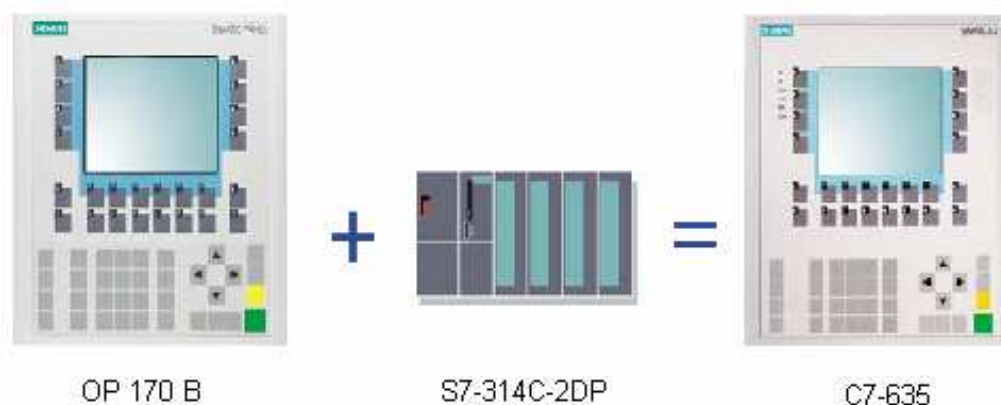
### 3. Simatic C7 – 635

Simatic C7 – 635 patří do kategorie kompletních přístrojů, které představují z technického i cenového hlediska ideální spojení řídicího systému a operátorského panelu. Obsahuje v sobě programovatelný automat řady 300 (konkrétně Simatic S7-314C-2DP) a operátorský panel OP170B. Řídicí systém se ovládá pomocí membránové klávesnice nebo dotykové obrazovky (podle typu).

Kompletní přístroj nalézá uplatnění hlavně ve strojích, kde je kladen důraz na co nejmenší prostor. Z toho vyplývají výhody integrace řídicího systému a operátorského panelu:

- výrazná úspora místa
- úspora času a zprovoznění a instalaci
- úspora nákladů ( až 20 % oproti klasickému uspořádání)
- výrazná flexibilita díky možnosti použití všech modulů S7 – 300

Klasickým vývojovým prostředím pro programovatelný automat řady 300 je Step7. Grafické rozhraní operátorského panelu OP170B je realizováno pomocí Protokol/Lite. Všechny tyto části v sobě obsahuje ústřední nástroj zvaný Simatic Manager z kterého lze jednotlivé prvky aplikace spouštět a zároveň slouží pro hardwarovou konfiguraci přístroje.



Obr. 5 Integrace OP170B a S7-314C-2DP do jednoho řídicího systému

Výkonnost jednotlivých typů je odstupňovaná. Tyto jednotky lze dále rozšířit o další v/v<sup>1</sup> moduly řady S7-300 na zadní stranu řídicí jednotky a to hned několika způsoby. Buď 2 moduly řady S7-300 na plochu nebo 4 moduly při uspořádání na hloubku podle (obr. 7 a 8). Přitom není za potřeby modul rozhraní (IM<sup>2</sup>) a jednotka si zachová minimální požadovanou velikost.

Dále je možno při větších nárocích na rozlišení použít modul rozhraní IM 360/361, který tvoří základ jedné řady (3 racku) a může obsahovat až 8 zásuvných modulů. Na jednotku C7 lze pak připojit 24 modulů typové řady S7-300.



Obr. 6 Zavěšení vstupně/výstupních modulů



Obr. 7 Uspořádání modulů na plochu



Obr. 8 Uspořádání modulů na hloubku

### 3.1 Technické parametry

Rozměry - jednotka (v x š x h) [mm] - panel (v x š) [mm]	274 x 260 x 79 257 x 231
Napájecí napětí - jmenovitá hodnota - rozsah	DC 24 V 20,4V až 28,8V
CPU	CPU 314C-2DP
Uživatelská paměť	64kB
Zaváděcí paměť - zásuvná, max.	8MB (MMC)
Bitové paměti	256 byte
Zálohování dat	Program a data v MMC
Doby provedení - bit operace - word operace	0,1 μs 1 μs
Integrované funkce - čítače - pulsní výstupy - měření frekvence - řízení polohy - řízení regulace	3 x inkrementální čidla 24V / 60kHz 4 x kanál PŠM (pulsní šířková modulace), 2,5kHz 4 x kanál, max. 60kHz SFB pro polohování jedné osy přes 2 DO, AO PID regulátor
Čítače (rozsah)	256 (0 až 999)
Časovače (rozsah)	256 (10ms až 9990ms)
Rozšiřovací moduly	max. 24
Adresovatelný prostor v / v	1024 / 1024 byte
Rozhraní	1 x RS232 1 x MPI 1 x Profibus DP (master nebo slave)
Digitální vstupy	24 x ; DC 24 V
Digitální výstupy	16 x ; DC 24 V; 0,5 A
Analogové vstupy	4 x ; ± 10 V; ± 20 mA; 0 až 600Ω
Analogové výstupy	2 x ; ± 10 V; ± 20 mA
Operátorský panel (OP)	OP170B (6") Tlačítek
Obraz procesu	128 / 128 byte

Displej - rozlišení - velikost	modrý jednobarevný 320 x 240 pixelů 5,7“ (14,5 cm)
Operační systém	Microsoft Windows CE 3.0
Počet programovatelných / funkčních tlačítek	14 / 10
Programování HMI	ProTool/Lite nebo vyšší WinCC flexible Compact nebo vyšší

Tab. 1 Technické údaje Simatic C7 – 635

### 3.2 Komunikace

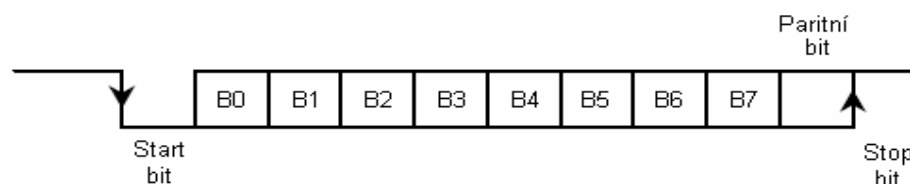
Dřívější označení pro sítě a síťové komponenty bylo SINEC (Siemens Network and Communication). V dnešní době se používá SIMATIC NET. Řídící systémy S7 – 300/400 mohou komunikovat prostřednictvím následujících sítí:

- RS 232
- MPI
- PROFIBUS
- Industrial Ethernet
- AS-Interface

Podrobněji se v této práci budu věnovat prvním třem komunikačním rozhraním v následujících podkapitolách.

#### 3.2.1 RS 232

Jedná se o sériové spojení kdy jsou data vysílána postupně za sebou. Někdy označované „bod-bod“ (PPI – Point Point Interface). Data jsou formována a vysílána v pořadí podle obrázku (obr. 9). Přenosová rychlost 50bitů/s až 19 200bitů/s. Pro přenos stačí minimálně 3 vodiče TXD, RXD, GND pro plný přenos je za potřebí 7event. více vodičů.



Obr. 9 Formát přenášených dat po RS 232



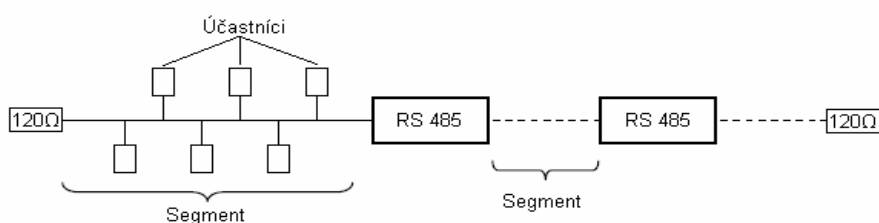
### 3.2.2 MPI

Každá centrální procesorová jednotka má integrované rozhraní pro vícebodové spojení MPI<sup>1</sup>, prostřednictvím kterého může být připojena do sítí bez nutnosti použití dalších jednotek. Do sítě MPI může být připojeno až 32 účastníků, jimiž mohou být:

- přístroje pro programování PG / PC<sup>2</sup>
- přístroje pro vizualizaci OP<sup>3</sup>
- další řídicí systémy S7-300 a S7-400

Přes globální data lze bez složitého programování vyměňovat data mezi uživatelskými programy v jednotlivých CPU<sup>4</sup> jednotkách.

Přenosová rychlost v síti MPI je pevně nastavena na 187,5kBaud. Délka vedení jednoho segmentu může dosahovat vzdálenosti až 50m. Ta se dá ještě prodloužit použitím dvou opakováčů RS 485 a na konci opatřenými zakončovacími odpory na 1000m.



Obr. 10 Schéma spojení účastníku v síti MPI pomocí RS 485

Každý účastník má pro identifikaci svoji specifickou MPI adresu. MPI adresa účastníků je při dodání předem přednastavena PG=0; OP=1; CPU=2.

V rámci jednoho segmentu může být až 32 účastníků a všichni musí mít různou adresu. Celkem může být v jedné síti MPI propojeno až 126 účastníků.

---

MPI<sup>1</sup> – Multi Point Interface  
 PG/PC<sup>2</sup> – Parameter Gate / Personál Computer  
 OP<sup>3</sup> – Operátorský panel

### 3.2.3 Profibus – DP

Profibus patří mezi nejrozšířenější sběrnice v automatizační technice na trhu zvláště v Evropě. Řídící jednotka C7-635 se dodává s integrovaným rozhraním Profibus-DP. Řídící jednotky pak lze konfigurovat jako master (řídící stanice) nebo slave (řízená stanice). Mezi největší výhody Profibusu patří rychlost, univerzálnost, spolehlivost a to vše za příznivou cenu.

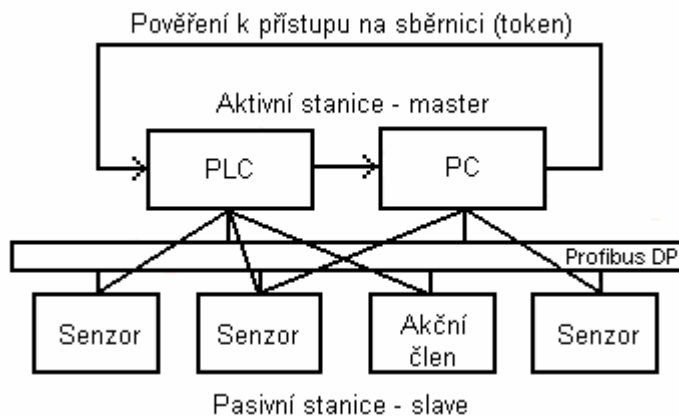
Pro spolehlivý a bezpečný chod sběrnice je zapotřebí funkcí pro diagnostiku. Diagnostické zprávy jsou posílány na sběrnici a ukládány v masterovi, kde jsou rozděleny do tří skupin:

- Diagnostika stanice – týká se operačních stavů stanice (teplota, přepětí,...)
- Diagnostika modulů – týká se I/O oblastí
- Diagnostika kanálů – týká se v/v periferií

Profibus-DP slouží pro rychlou cyklickou komunikaci mezi PLC a decentralizovanými vstupy a výstupy. Na jeden segment sběrnice je možno připojit až 32 zařízení a pro větší počet je potřeba použít opakovače. Profibus podporuje čtyři topologie, nepoužívanější sběrnici, strom, hvězdu a kruh.

Jelikož může na sběrnici vysílat v danou chvíli pouze jedno zařízení (následovala by kolize dat), musí být stanovena metoda řízení přístupu. Lze použít kombinaci dvou základních metod:

- Token-passing (předávání pověření v logickém kruhu)
- Master-slave (centrálně řízené dotazování)

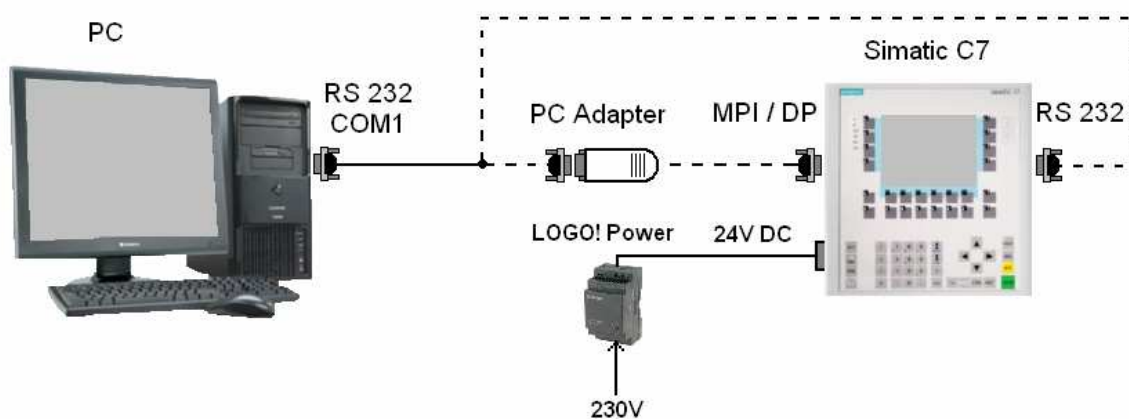


Obr. 11 Profibus-DP – princip přístupu k síti

## 4. Realizace projektu

### 4.1 Zapojení pracoviště

Základním krokem při realizaci projektu je nainstalování potřebného software Step7 a ProTool do PC. Dalším krokem bylo sestavení všech použitých komponentů podle obrázku Obr. 12. Simatic C7-635 je propojen s počítačem pomocí PC Adaptéru z portu MPI na port RS 232 a dále pomocí přímé sériové linky z portu RS 232 na port RS 232, který slouží pro naprogramování operátorského panelu. Jestliže máme PC pouze s jedním sériovým portem musíme vždy kabel přendávat ručně mezi vstupy RS 232 a MPI/DP Simaticu C7. Po správném zapojení všech zařízení můžeme zapnout zdroj napětí 24V LOGO! Power a je-li vše správně zapojeno zkontrolujeme pomocí indikační led diody DC5V (zelená) na operátorském panelu, která signalizuje 5V napájení pro CPU a sběrnici.



Obr. 12 Zapojení všech použitých komponentů

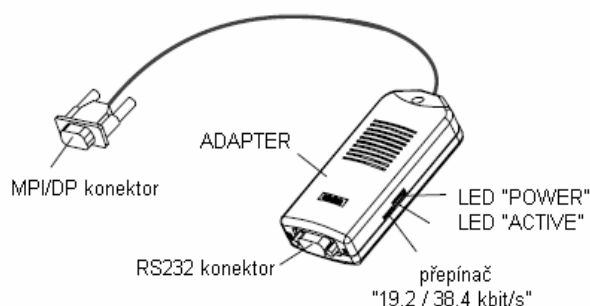
### 4.2 Použitý hardware

#### 4.2.1 Simatic C7 – 635

Simatic C7 – 635 je zařízení, které v sobě integruje programovatelný automat řady 300 (S7 – 314C – 2DP) a operátorský panel OP170B. Podrobnější informace včetně technických parametrů jsou obsaženy v samostatné kapitole 3.

### 4.2.2 PC Adapter

PC Adapter spojuje počítač přes sériové rozhraní RS232 s rozhraním MPI / DP u systémů S7 / M7 / C7 podle obrázku (obr. 12).



Obr. 13 PC Adapter

Podle jeho použití rozlišujeme dva druhy adapteru:

- PC Adapter – slouží pro spojení PLC s PC přes rozhraní MPI/DP s RS232
- TS Adapter – slouží pro dálkový servis přes libovolnou telefonní linku. Bývá ve spojení s modemy, ISDN<sup>1</sup> nebo GSM<sup>2</sup>.

### 4.2.3 LOGO! Power

Jako zdroj napájecího napětí je použit LOGO! Power též od firmy Siemens. Tento spínaný zdroj nám nabízí snadné použití v modulárních stavebnicových systémech.

Parametry:

Vstup – AC 230V

Výstup – DC 24V / 1.3A



Obr. 14 LOGO! Power – zdroj napětí

### 4.3 Použitý software

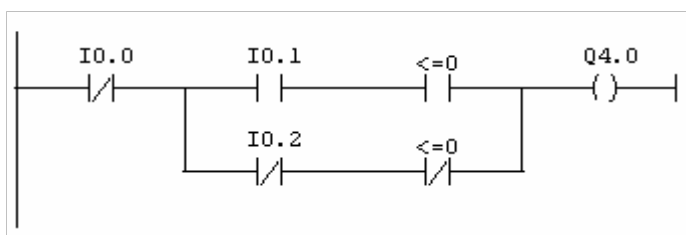
#### 4.3.1 Step 7 v 5.2

Step 7 je programovací a konfigurační software určený pro systémy SIMATIC především řady 300 a 400. Tento softwarový balíček v sobě zahrnuje:

- SIMATIC Manager – je integrovaná platforma pro všechny nástroje a konfiguraci projektu, ze kterého můžeme projekt spravovat.
- HW Config – pro konfiguraci hardwaru
- LAD/STL/FBD – editor pro tvorbu uživatelských programů
- Symbols – pro editaci a správu symbolických proměnných
- NetPro – pro konfiguraci síťových spojení MPI, PROFIBUS
- S7 – PDIAG – pro diagnostiku systému

Uživatelské programy ve STEP 7 se realizují ve LAD/STL/FBD editoru.

- LAD (Ladder Diagram) – neboli liniové (kontaktní) schéma je grafickou prezentací Booleovských výrazů kombinujících vstupní a výstupní kontakty. Využívá grafické symboly zobrazované podobně jako přičky u žebříku ohraničené po obou stranách levou a pravou vodící tyčí.



Obr. 15 Ukázka programování v LAD

- STL (Statement – Instruction List) – je univerzální programovací jazyk na úrovni assembleru. Vhodný pro složitější aplikace. Nevhodný pro velké množství strojově orientovaných instrukcí.

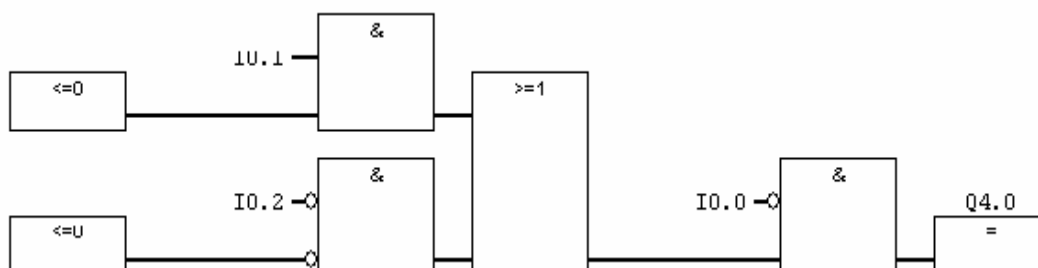
```

AN   I      0.0
A(
A    I      0.1
A    <=0
O
AN   I      0.2
AN   <=0
)
=    Q      4.0

```

Obr. 16 Ukázka programování v STL

- FBD (Function Block Diagram) – program se vytváří pomocí vzájemného propojení jednotlivých funkčních bloků. Výhodou je názornost díky obdobným schémátům s integrovanými obvody IO. Tento programovací jazyk může být nevhodný u složitých úloh.



Obr. 17 Ukázka programování ve FBD

Struktura programu může být rozdělena do několika částí:

- Organization Block (OBx) – organizační bloky – nejdůležitějším organizačním blokem je OB1, ve kterém je cyklicky vykonávaný hlavní program.
- Function Block (FBx) – funkční bloky – mohou uchovávat stavy vnitřních proměnných i do dalších cyklů. Nutnou součástí jsou ovšem DB.
- Function (FCx) – funkce – jedná se o klasické funkce
- Data Block (DBx) – datové bloky – slouží k ukládání dat
- Data Type (UDTx) – datové typy – předurčují vlastnosti dat. Datové typy se deklarují globálně v tabulce symbolů, nebo lokálně v deklarační části bloku. Označení datových typů bývá např.:BOOL,REAL, DINT...

#### 4.3.2 ProTool Lite v 6.0

ProTool Lite je program pro grafické naprogramování operátorských panelů. Programování probíhá po sériové lince RS232. Vývojové prostředí je jednoduché, z velké části probíhá pomocí přidávání jednotlivých komponent myší. Přímo v programu Pro ToolLite můžeme editovat jednotlivé obrazovky (Screens) i ovládací tlačítka operátorského panelu.

V současné době bývají všechny verze ProTool nahrazovány novým softwarem WinCC flexible, který umožňuje například servis a diagnostiku přes internet.

#### 4.4 Grafická část projektu

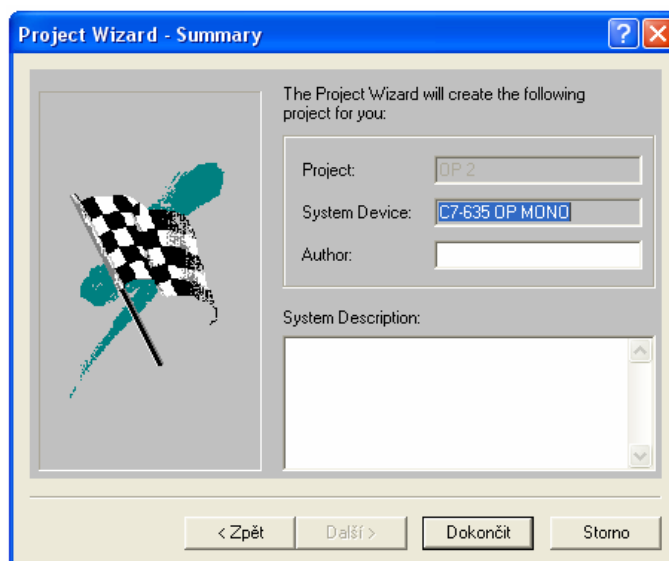
V této části bakalářské práce se budeme věnovat konfiguraci grafické části operátorského panelu OP-170B. Prvním krokem je vytvoření nové položky v SIMATIC Manageru, kde v hlavním menu „Insert“ vybereme „Station“ a zde položku „SIMATIC OP“. Vytvoří se nám odkaz „S7\_Pro1“. Po jeho otevření se nám spustí průvodce nastavení aplikace ProTool Lite. Zde vybereme zařízení „C7-635 OP MONO“ (obr. 18) a dáme „Další“, kde vybereme námi požadované zařízení „SIMATIC S7 300/400 V6.0“ (obr. 19). Po splnění všech popsanych úkonů klikneme na tlačítko „Dokončit“.



Obr. 18 Výběr operátorského panelu

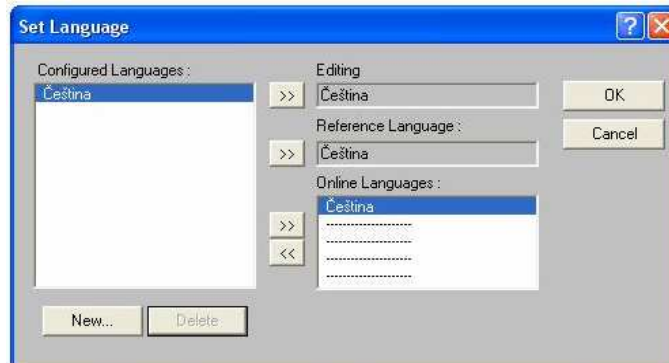


Obr. 19 Výběr typu protokolu



Obr. 20 Dokončení výběru typu

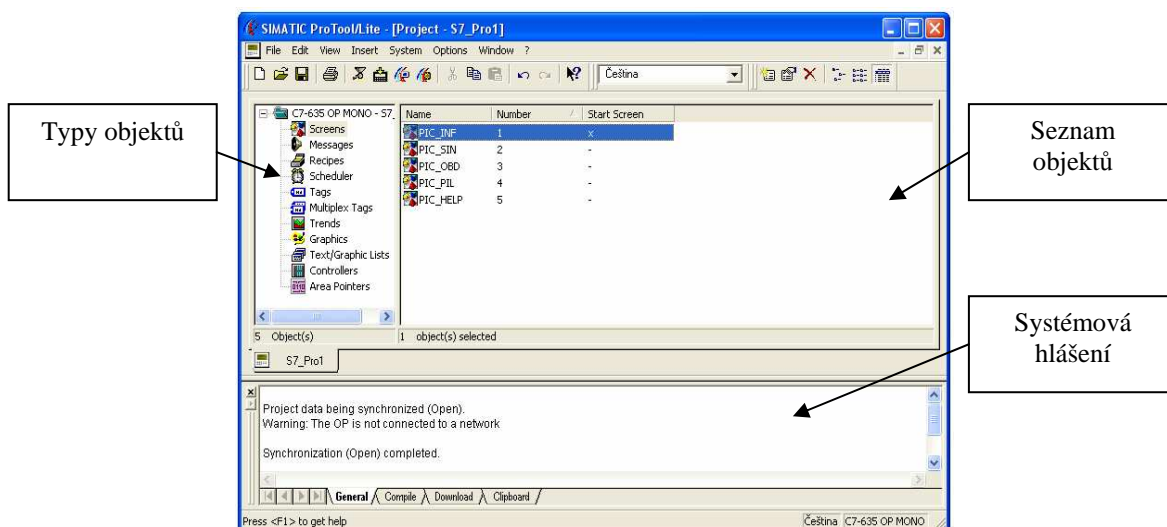
Dále ještě můžeme pro lepší a srozumitelnější orientaci nastavit českou diakritiku. V hlavním menu zvolíme z nabídky „Systém“ a zde „Language Assignment...“. Po zobrazení dialogového okna (obr. 20) klikneme na tlačítko „New...“, kde si vybereme češtinu.



Obr. 21 Výběr české diakritiky

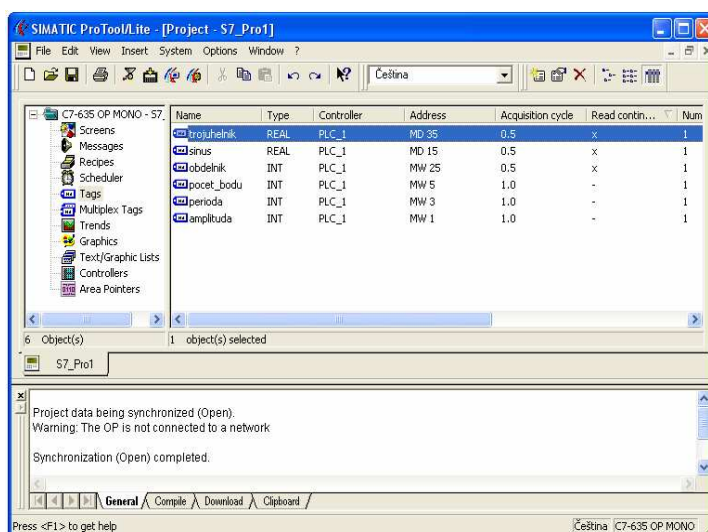
Ve vývojovém prostředí ProTool Lite (obr. 22) začneme s vytvořením nového projektu. Nejdříve si nadefinujeme obrazovky, které budeme chtít na operátorském panelu zobrazovat. V hlavním menu ve složce „Insert“ vybereme kolonku „Screens“, pomocí které se nám vloží nová položka „PIC\_1“, kterou můžeme následně pojmenovat podle našich potřeb. Stejným postupem vložíme všechny požadované obrazovky. Obrazovku, kterou chceme mít jako výchozí při spuštění systému nastavíme tak, že na požadovanou položku klikneme pravým tlačítkem myši. Dále klikneme na kolonku „Properties“, kde zatrheme volbu „Start Screen“.





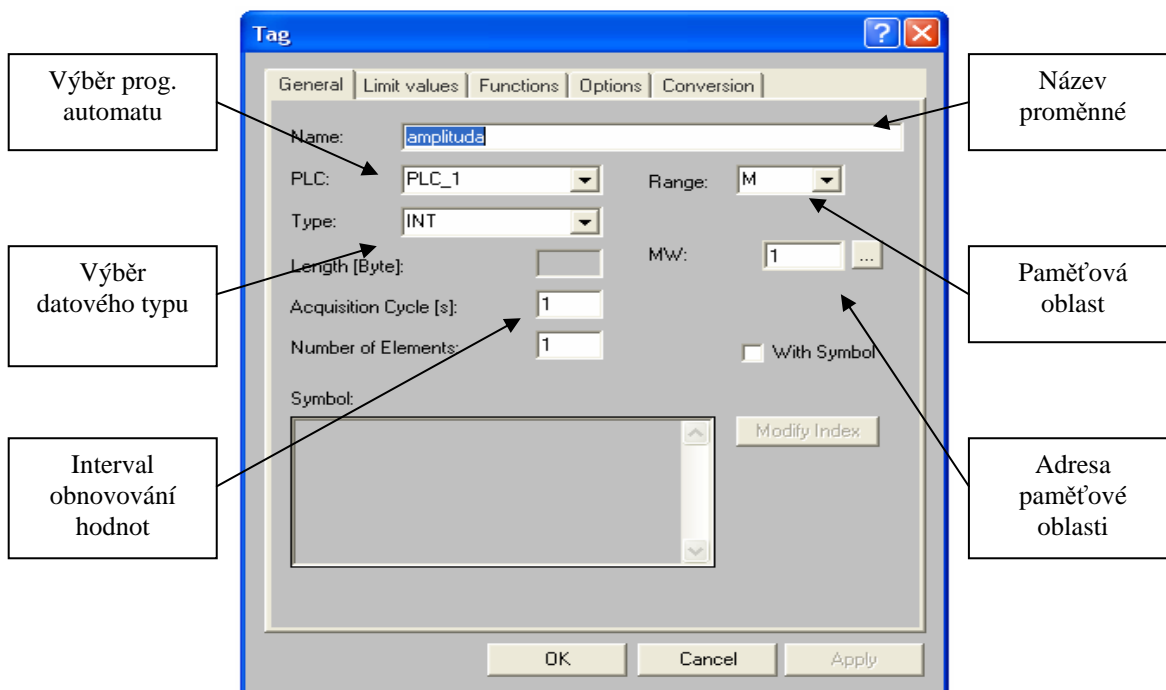
Obr. 22 Vývojové prostředí ProTool Lite

Abychom mohli na operátorském panelu zobrazovat data z programovatelného automatu, musíme si nadefinovat proměnné. Proměnné nadefinujeme pomocí tzv.tagů. Tagy vytvoříme dvojitým poklepnutím levého tlačítka myši na položku „Tag“ v kategorii objektů.



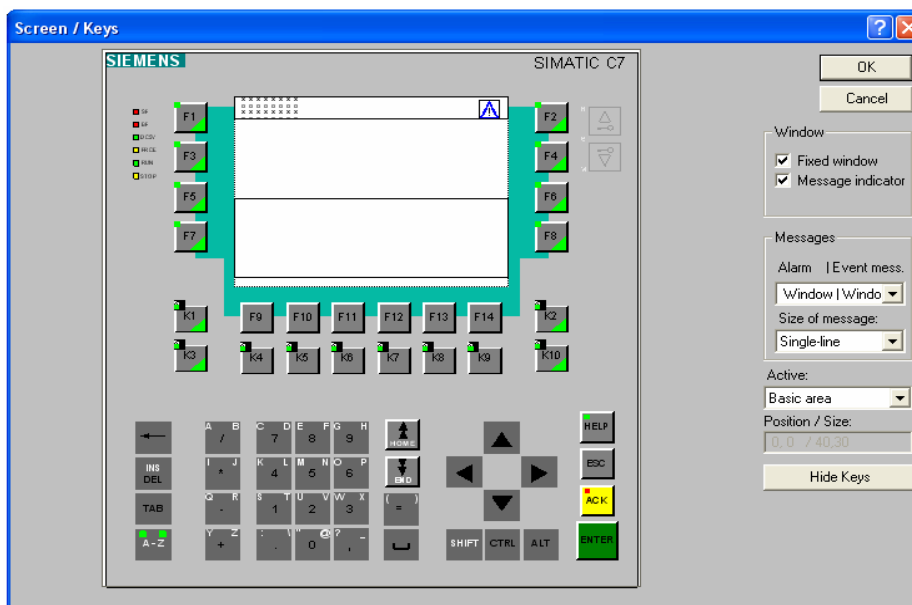
Obr. 23 Seznam definovaných proměnných

Parametry proměnných nadefinujeme tak, že si zvolíme požadovanou proměnnou a z hlavní nabídky menu vybereme „Edit“ následně „Properties“. Zobrazí se nám editované okno daného tagu podle obrázku (obr. 24). Zde můžeme proměnným přiřadit datový typ, paměťovou oblast, adresu atd.



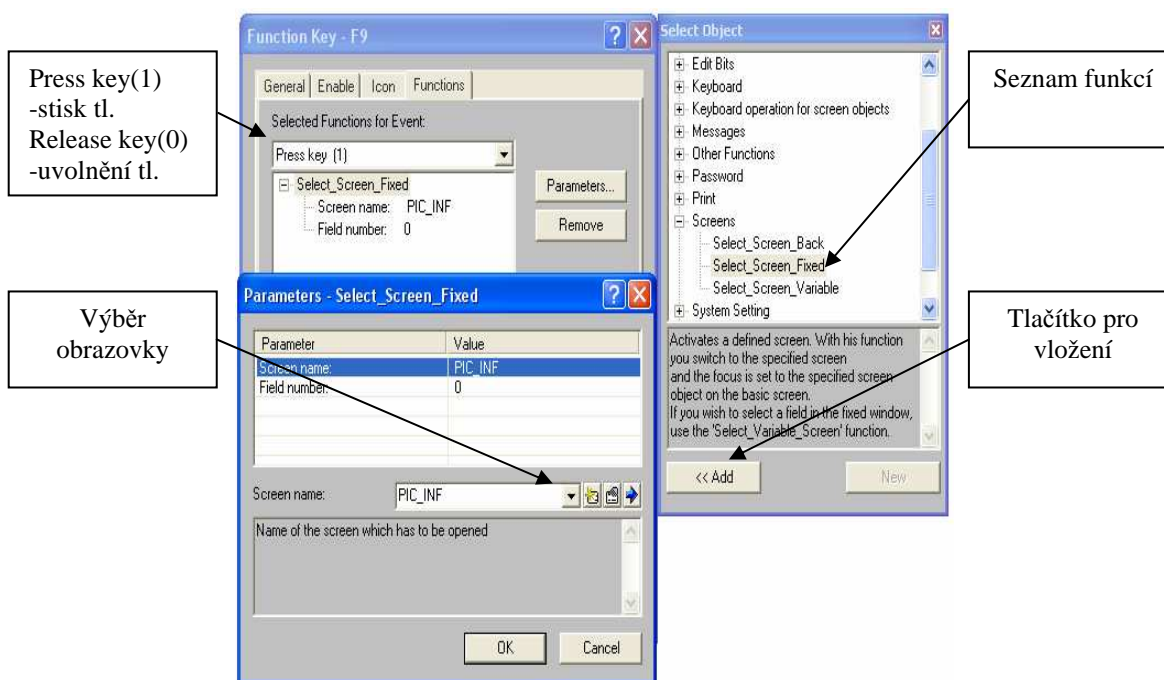
Obr. 24 Definice proměnných

Abychom mohli obrazovky mezi sebou přepínat a nastavovat hodnoty do vstupních textových polí pomocí operátorského panelu, musíme ještě nadefinovat funkční klávesy jednotlivých obrazovek. Toho se docílí tak, že v hlavním menu vybereme „Systém“ a zde „Screen/Keys...“.



Obr. 25 Okno pro výběr a nastavení funkčních kláves

Po zvolení funkčního tlačítka, které má sloužit pro výběr obrazovky (F9 ...F14) a které chceme konfigurovat, vybereme ze seznamu funkcí podle obrázku (obr. 26). „Screens“ a zde „Select\_Screen\_Fixed“. Pomocí tlačítka „Add“ danou funkci přiřadíme vybranému tlačítku. Následně pomocí „Parameters...“ zadáme tlačítku atributy.



Obr. 26 Konfigurace tlačítek pro přepínání obrazovek

Pro spuštění systému operátorského panelu slouží klávesa F14. Její funkci nastavíme stejným způsobem jako pro přepínání obrazovek, jen ze seznamu funkcí vybereme „Strt\_Control\_Panel“.

V neposlední řadě zbývá nastavení funkčních kláves pro nastavování požadovaných hodnot. Toho lze dosáhnout pomocí numerické klávesnice a potvrzení klávesou ENTER, nebo zvolenými klávesami, které mění požadovanou hodnotu skokově.

F1 a F3 \_\_\_\_\_ nastavení hodnoty amplituda \_\_\_\_\_ +/- 1  
 F2 a F4 \_\_\_\_\_ nastavení hodnoty amplituda \_\_\_\_\_ +/- 10  
 F5 a F7 \_\_\_\_\_ nastavení hodnoty perioda \_\_\_\_\_ +/- 1  
 F6 a F8 \_\_\_\_\_ nastavení hodnoty perioda \_\_\_\_\_ +/- 10  
 K1 a K3 \_\_\_\_\_ nastavení hodnoty počet bodů \_\_\_\_\_ +/- 1  
 K2 a K10 \_\_\_\_\_ nastavení hodnoty počet bodů \_\_\_\_\_ +/- 10

Nastavení funkce pro tyto klávesy se provede stejným způsobem jako pro přepínání obrazovek, nebo pro zobrazení systému operátorského panelu, ale místo funkce pro změnu obrazovky použijeme funkci pro zvýšení nebo snížení hodnoty. „Incerase\_value“ je funkce pro zvýšení hodnoty. „Decrease\_value“ je funkce pro snížení hodnoty. Tyto funkce se nachází pod složkou „Calculation“ v seznamu funkcí. Jako atributy nastavíme název proměnné a hodnotu požadované změny (1,10).

**Obrazovka – „INFO“**

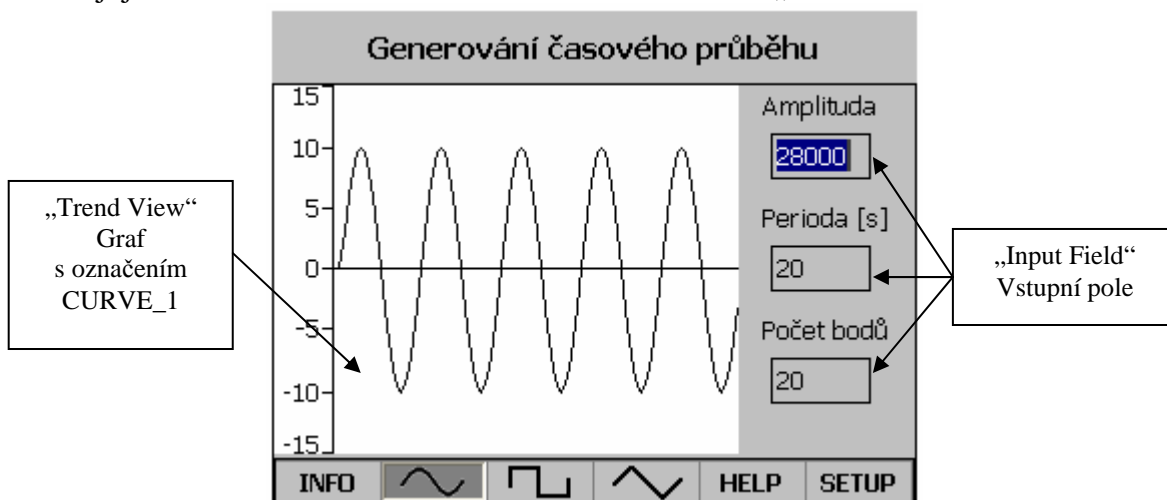
Tato obrazovka je nastavena jako výchozí, proto se zobrazuje jako první vždy po spuštění systému, nebo po restartu systému. Z dalších zobrazení se na ni přepneme funkční klávesou F9. Obsahuje informační údaje o autorovi.



Obr. 27 Základní obrazovka „INFO“

**Obrazovka – „SINUS“**

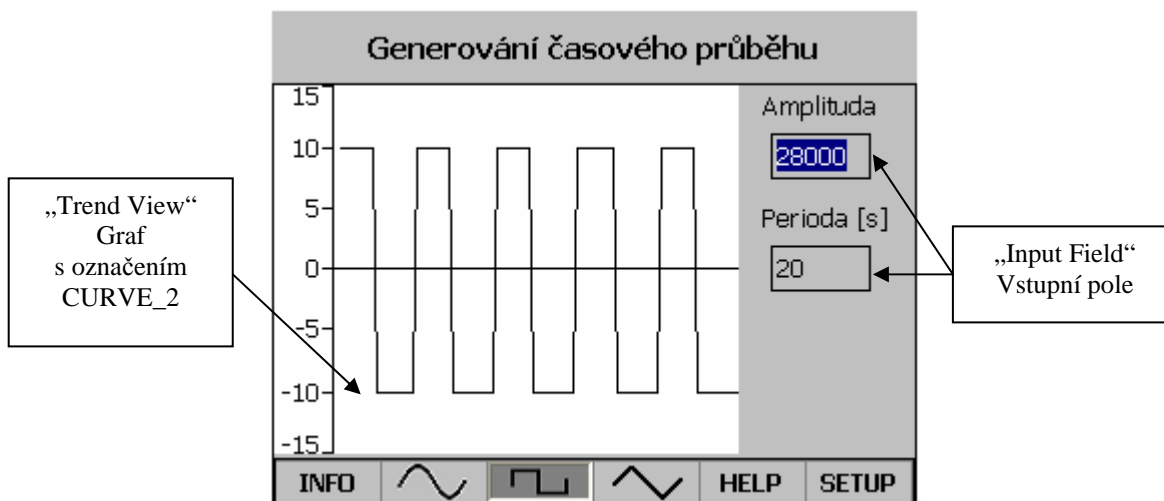
Na tuto obrazovku se přepneme díky tlačítku F10. Obsahuje graf, v kterém se zobrazuje generovaná funkce sinus. Osa X reprezentuje časovou osu, která je nastavená na 100sekund. Osa Y nám udává velikost napětí na analogovém výstupu v rozsahu +/- 15voltů. Dále tato obrazovka obsahuje 3 vstupní pole pro zadávání požadovaných veličin (Amplituda, Perioda, Počet bodů – udává kolika body bude zadaná funkce vygenerovaná za jednu periodu). Všechny tři vstupní pole jsou nastavená tak, aby minimální hodnota nemohla dosáhnout záporných hodnot. Tuto skutečnost nastavíme editací daného tagu, kde v záložce „Limit values“ nastavíme v editu „Lower Value“ hodnotu nula a zatrhneme z výběru „Const“. Jak a pomocí kterých funkčních kláves se hodnoty do těchto polí zadávají je zobrazeno v samostatné obrazovce s označením „HELP“.



Obr. 28 Obrazovka grafu funkce „SINUS“

### Obrazovka – „OBDÉLNÍK“

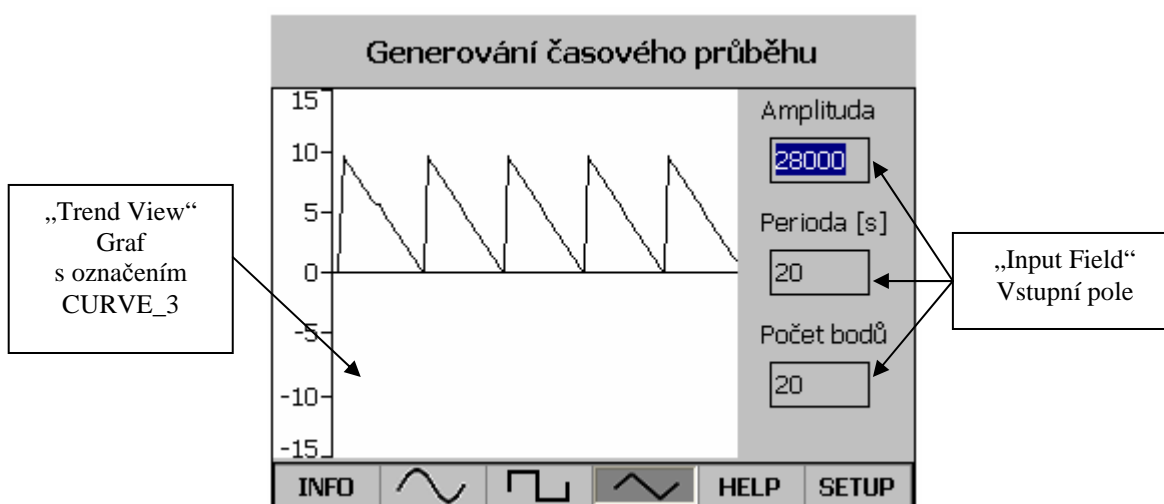
Stiskem funkční klávesy F11 se přepneme na další obrazovku zobrazující graf generované funkce obdélníka. Obsahuje stejné složky jako předchozí obrazovka s výjimkou toho, že neobsahuje vstupní pole pro zadání počtu bodů. V tomto případě tato hodnota není potřebná, protože obdélník nabývá pouze tří hodnot (nula, max., min.).



Obr. 29 Obrazovka grafu funkce „OBDÉLNÍK“

### Obrazovka – „TROJÚHELNÍK“

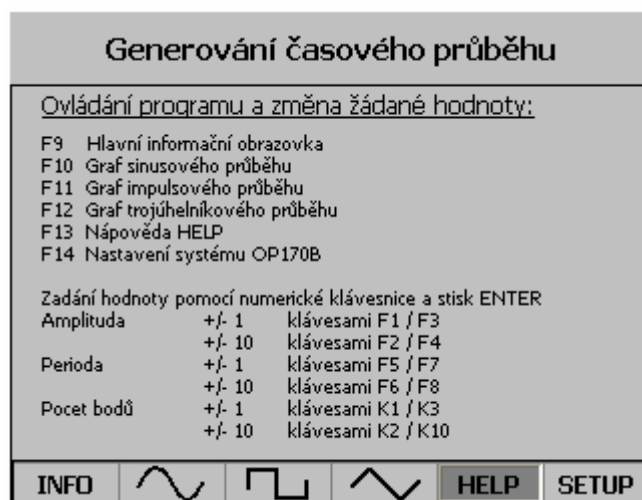
Volá se stiskem tlačítka F12 a obsahuje totožné prvky se zobrazením pro funkci sinus.





Obr. 30 Obrazovka grafu funkce „TROJÚHELNÍK“

**Obrazovka – „HELP“**

Tato nápověda se volá funkční klávesou F13 a obsahuje stručný popis pro ovládání operátorského panelu. Všechny zobrazené objekty jsou typu „Text Field“.



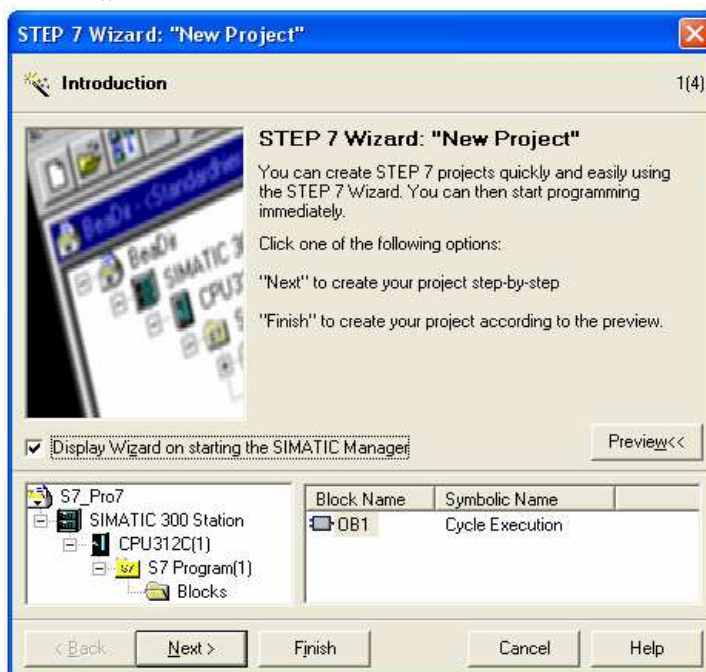
Obr. 31 Obrazovka grafu funkce „HELP“

Takto vytvořenou aplikaci v ProTool Lite můžeme nyní zkompilevat stiskem tohoto tlačítka  po jehož provedení se objeví hlášení pro uložení aplikace. A dále provedeme nahrání do operátorského panelu stiskem tlačítka „Download“  .

#### 4.5 Řídící část projektu

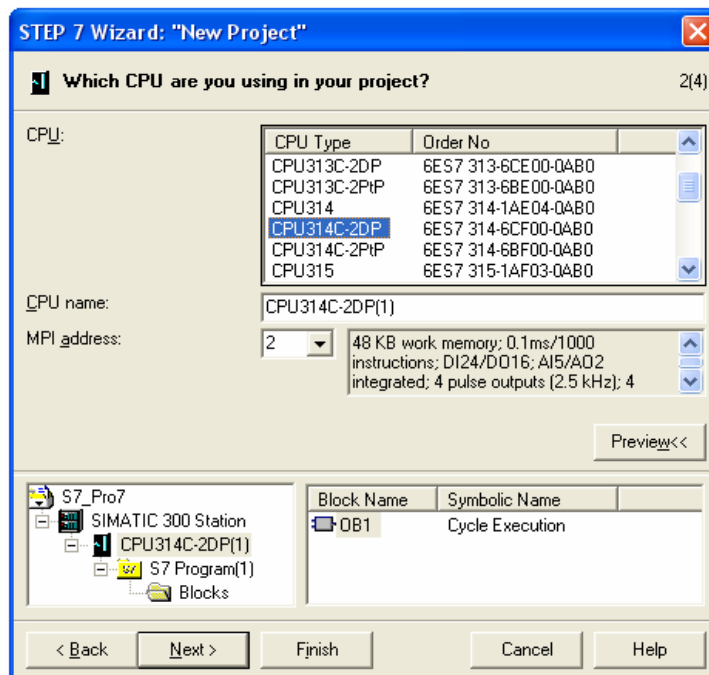
Řídící program programovatelných automatů řady 300/400, do které spadá i SIMATIC C7 se vytváří ve vývojovém software STEP 7. Tento software je umístěn v SIMATIC Manageru, který je ústředním nástrojem pro všechny dílčí programy STEP 7. Obsluha zodpovídá běžnému standardu ve WINDOWS.

Vytvoření nové aplikace začneme tak, že po spuštění SIMATIC Manageru v PC se nám otevře průvodce pro nové vytvoření aplikace STEP 7 Wizard: "New Project" (obr. 32), kde klikneme na tlačítko „Next“.

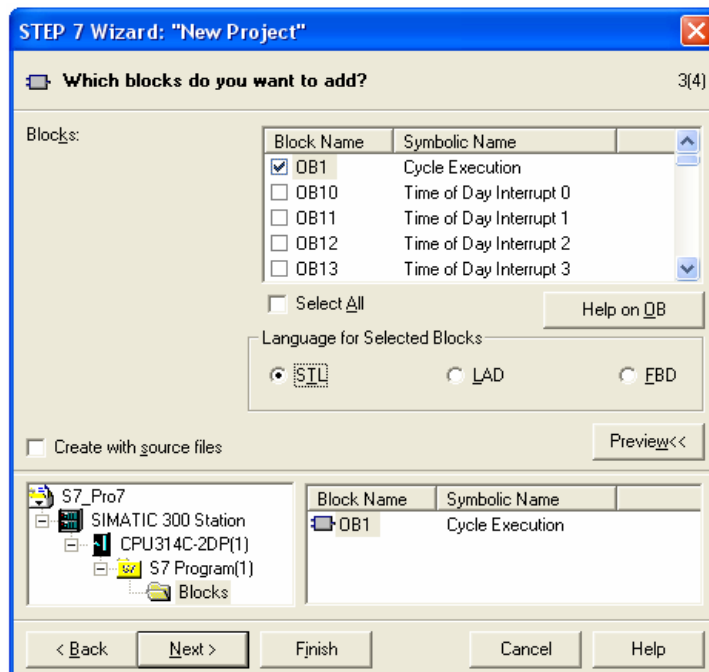


Obr. 32 Spuštění průvodce pro vytvoření nové aplikace

Dalším krokem průvodce je volba programovatelného automatu (obr. 33). Jelikož námi požadovaný PA (C7-635 OP) v hlavní nabídce není obsažen, vybereme z nabídky podobný (CPU 314C-2DP), který později v HW Cifigu nahradíme. Dále ještě zvolíme adresu MPI (2) a stiskneme tlačítko „Next“. Otevře se nám okno pro výběr organizačního bloku (OB1) a jazyka pro programování (STL) podle obrázku (obr. 34). Po splnění tohoto kroku průvodce a potvrzení klávesy „Next“ se zobrazí okno pro pojmenování aplikace (SIMATIC C7). Na konec potvrdíme dokončení tlačítkem „Finish“ (obr. 35).

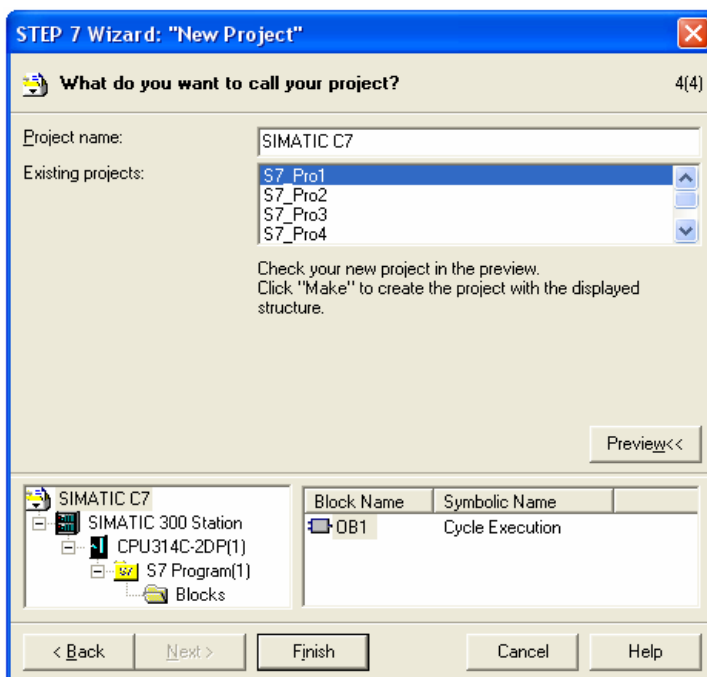


Obr. 33 Volba typu PA



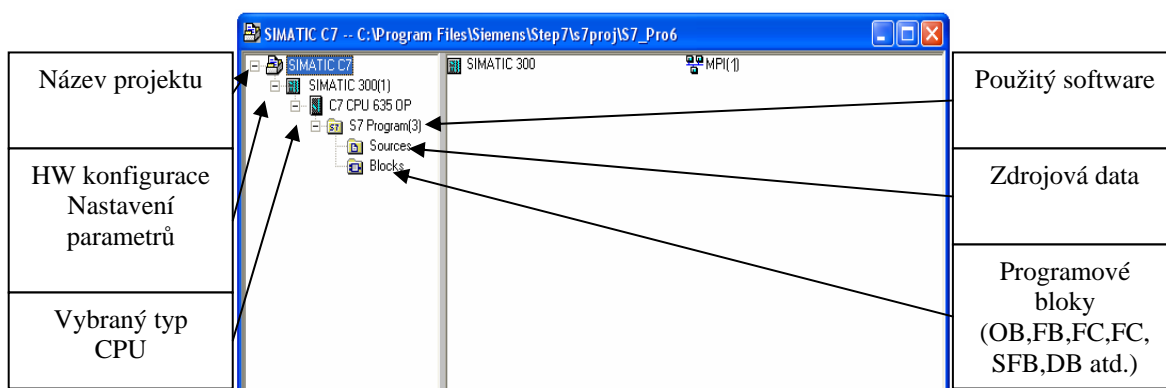
Obr. 34 Volba organizačního bloku a jazyka pro programování






Obr. 35 Okno pro pojmenování aplikace

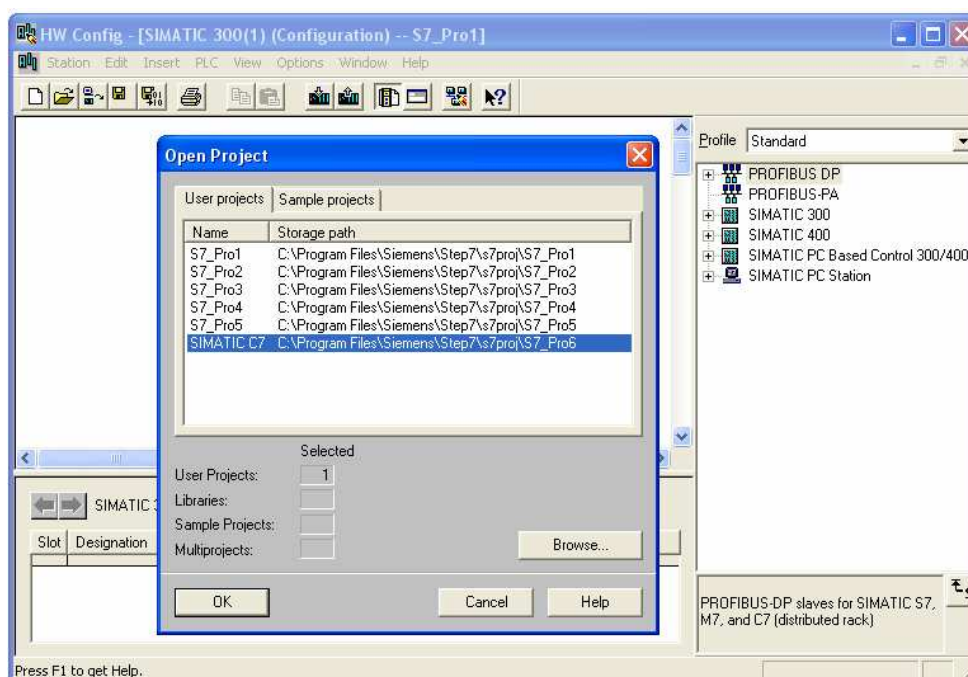
Po splnění všech kroků průvodce se nám otevře stromová struktura (obr. 36), která obsahuje všechny základní části nově vzniklé aplikace. Jednotka CPU je obsažena ve složce „SIMATIC 300“, rozhraní „MPI(1)“ a organizační blok (OB1), který se nachází pod složkou „Blocks“. Zbývajícím úkolem pro vytvoření aplikace je konfigurace hardware pomocí HW Config a NetPro.



Obr. 36 Vytvořený projekt průvodcem „New Project Wizard“

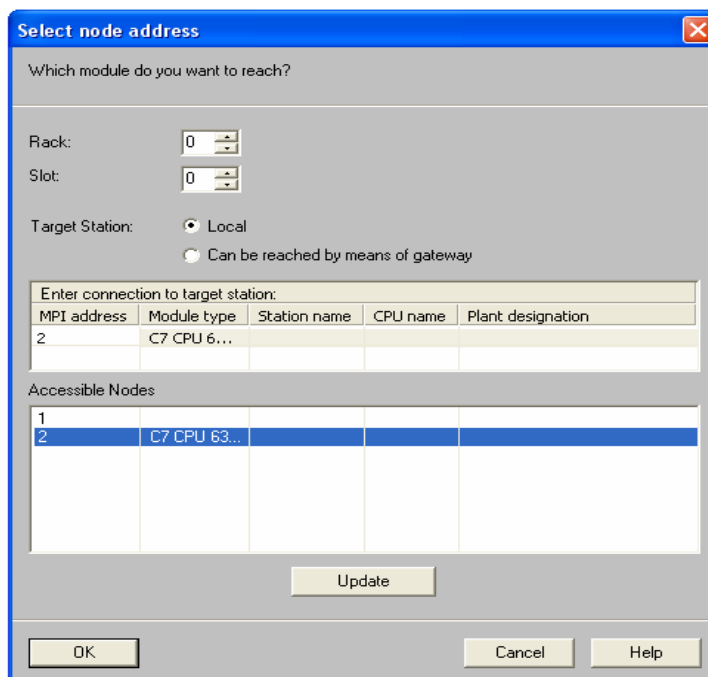
### 4.5.1 Konfigurace hardware – HW Config

Hardwarovou konfiguraci nastavíme tak, že v SIMATIC Manageru ve složce „SIMATIC 300“ otevřeme položku „Hardware“. Po jejím otevření se nám objeví hlavní okno programu HW Config obsahující zatím CPU314C-2DP. Jelikož průvodce pro vytvoření nové aplikace neobsahuje námi požadovaný C7-635 musíme jej nastavit nyní. Nejprve musíme CPU314C-2DP odstranit. Potom, máme-li navázané spojení, načteme konfiguraci přímo z PA. Prvním krokem je kliknutí na buton pro načtení zařízení „Upload to programming device“ s tím to symbolem . Otevře se nám okno, podle obrázku (obr. 37), s nabídkou, do jaké aplikace chceme nastavení zařadit.



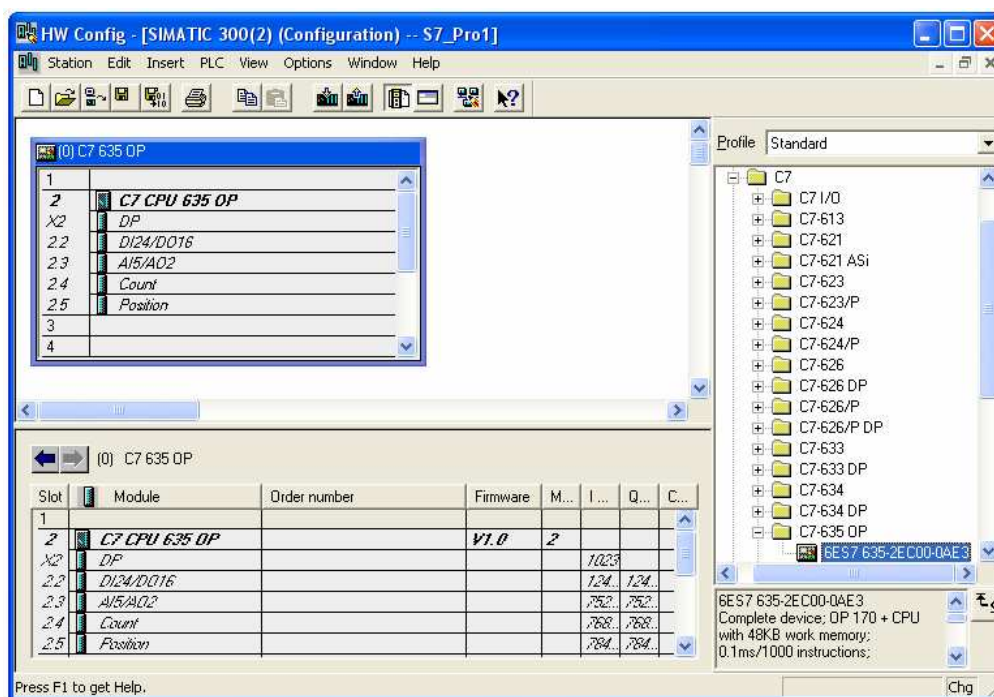
Obr. 37 Načtení PA a uložení do požadované aplikace

Po zařazení k požadované aplikaci a potvrzení tlačítka „OK“, se otevře okno pro zadání adres. Zde stačí kliknout na „View“. Po zobrazení načteného zařízení jej označíme podle obrázku (obr. 38), aktualizujeme stiskem tlačítka „Upload“ a potvrdíme „OK“. Tímto způsobem máme nastavenou hardwarovou konfiguraci. Stačí už jen uložit pomocí „Save“.



Obr. 38 Aktualizace nově načteným systémem

Dalším způsobem, kterým lze systém nakonfigurovat je ručním zadáním správného typu CPU C7-635, který je obsažen v pravém seznamu dostupných zařízení. Ve složce „SIMATIC 300“ vybereme „C7“ dále „C7-635 OP“ a zde „6ES7 635-2EC00-0AE3“.



Obr. 39 Nakonfigurovaný systém

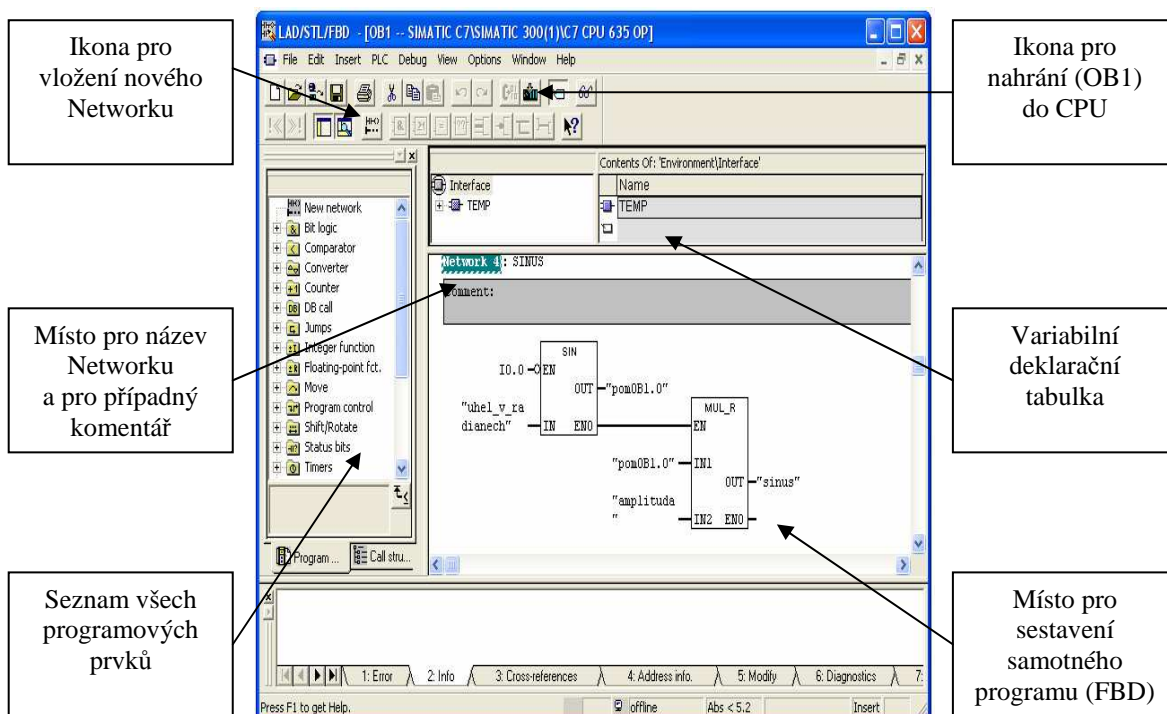


Mazání se provádí v SIMATIC Manageru přepnutím do online módu přes menu „View“ – „Online“. V online okně vybereme adresář „Block“, kde vybereme bloky které chceme odstranit a kliknutím na pravé tlačítko myši zvolíme „Delete“. Tímto způsobem se mohou vybrané bloky vymazat, zbývá jen potvrdit, že systémové funkce nemohou být vymazány.

Při přepnutí do online režimu se ovšem objevilo hlášení, že nefunguje spojení mezi CPU a MMC kartou. Tato skutečnost mne přivedla na myšlenku zda není poškozená karta. Po výměně karty dříve signalizované chyby zmizely a spojení šlo bez problému navázat.

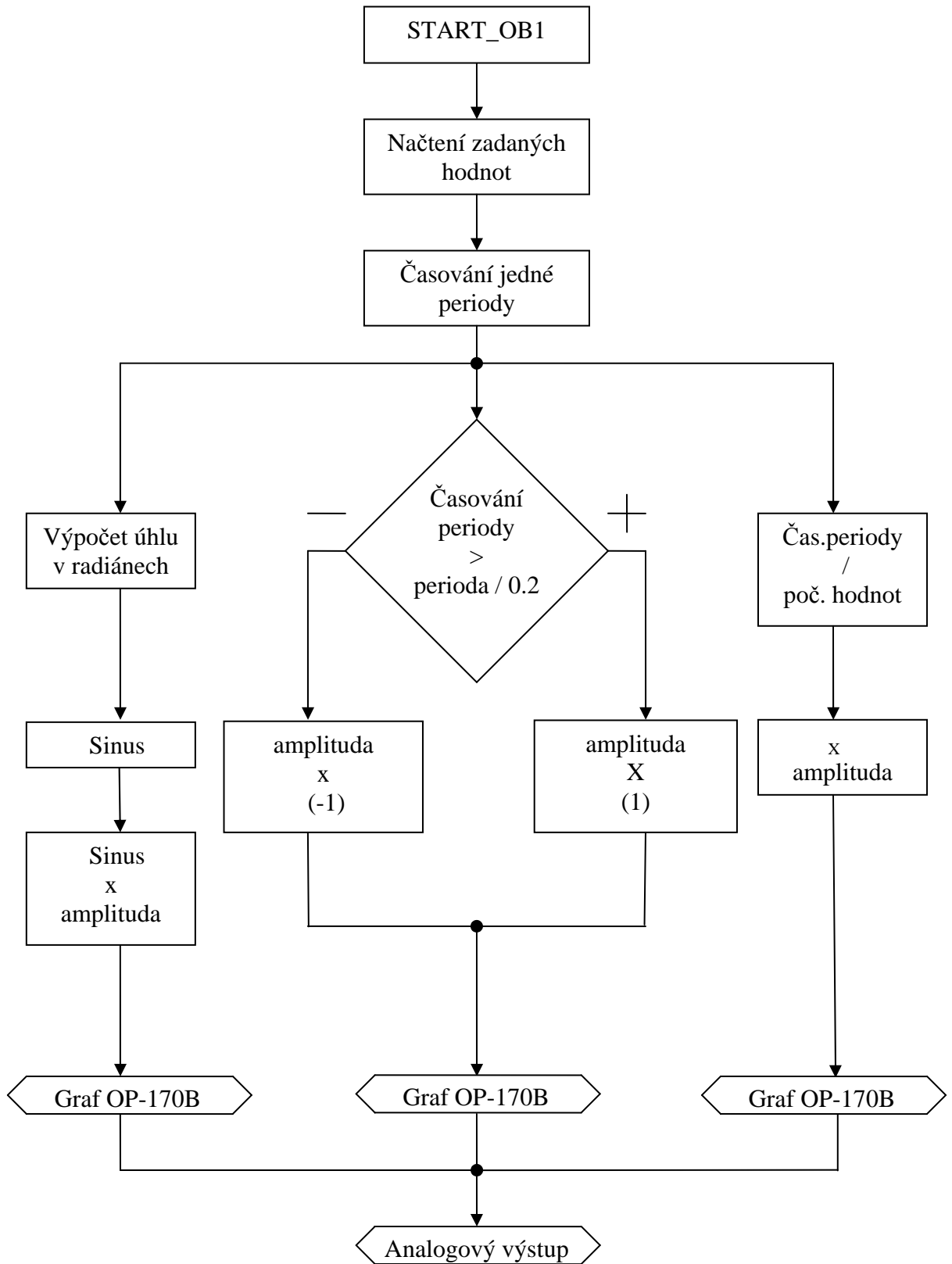
#### 4.5.4 Řídící program programovatelného automatu

Programové prostředí LAD/STL/FBD (obr. 41) pro tvorbu řídicích programů, nalezneme v SIMATIC Manageru otevřením organizačního bloku OB1 ve složce „Block“.



Obr. 41 Programové prostředí LAD/STL/FBD ve funkčním plánu FBD

Hlavní program OB1 názorně popisuje následující vývojový diagram.



#### 4.5.5 Popis jednotlivých částí programů

„**Network\_1**“ v tomto bloku je obsažena funkce FC1, která má za úkol načtení zadávaných proměnných (amplituda, perioda, počet bodů) a převod do námi používaných datových typů, abychom je nemuseli převádět vždy, když s ním musíme počítat.

„**Network\_2**“ obsahuje funkci FC2. Tato funkce v sobě zahrnuje 2 časovače, které se mezi sebou navzájem střídají. Jeden cyklus časovače nám udává jednu periodu generované funkce.

„**Network\_3**“ v sobě zahrnuje funkci FC3, která slouží pro výpočet úhlu, který zadáváme do výpočtu funkce sinus. Úhel musí být vypočítán v radiánech podle vzorce  $y=a/b*2*\pi$ . a...udává hodnotu časovače, b...udává počet bodů na jedné periodě

„**Network\_4**“ obsahuje přímo funkci sinus daného úhlu násobenou velikostí amplitudy.

„**Network\_5**“ v něm se nachází rozhodovací člen, který porovnává hodnoty na dvou vstupech. Na prvním vstupu se nachází aktuální hodnota časovače a na druhém celková hodnota časovače dělená 0.2. Podle toho zda je rozhodovací člen pravdivý či není se generuje kladná nebo záporná amplituda obdélníka.

„**Network\_6**“ řeší výpočet pro generování trojúhelníka podle vzorce  $y=a/b$  násobený nakonec velikostí amplitudy. Vysvětlení proměnných je stejné jako v network 3.

„**Network\_7**“ zajišťuje vyslání hodnot na analogový výstup. K tomu slouží funkce FC5, v které se musí přepínat jednotlivé průběhy, které chceme na analogovém výstupu zobrazovat. Důvodem je, že průběhy máme tři a analogové výstupy pouze dva.

„**Network\_8**“ tento poslední oddíl slouží k vypsání generovaných průběhů do grafu v operátorském panelu OP 170B pomocí naprogramované funkce FC6.

Vyslání hodnot na analogový výstup se provádí pomocí zadání příkazu a paměťové adresy (PQW xxx). Adresa analogových výstupů se zjistí v HW Configu. SIMATIC C7 disponuje dvěma analogovými výstupy označenými AO0 a AO1 s adresami 752 až 755. Každá z těchto adres zabírá 2 byte paměťové oblasti.

Abychom mohli libovolně měnit frekvenci, je nutné zadávat do časovačů hodnotu jako proměnnou. Protože hodnota do časovače se zadává ve formátu S5T#, je nutné zadanou hodnotu převést z INT na námi požadovaný formát. Jelikož pro převod používám v programu instrukční blok MOVE, který není schopen převést větší hodnotu jak 999, můžeme z časovačů dostat největší možnou frekvenci 9s990ms. Tato skutečnost je dána rozlišením časovačů a jejím rozsahem, který se navzájem mezi sebou překrývají.

Rozsah	Rozlišení[s]
10ms to 9s 990ms	0.01
100ms to 1m 39s 900ms	0.1
1s to 16m 39s	1
10s to 2h 46m 30s	10

Tab. 2 Rozlišení časovače limitováno podle rozsahu

## // ZAČÁTEK HLAVNÍ PROGRAM OB1 PRO SIMATIC C7

```

-----
NETWORK 1 //Převod zadávaných hodnot z INT na REAL
  AN I 0.0 //Negace vstupní adresy
  JNB _001 //Skočí na návěští _001, když RLO=0
  CALL "prevod_hodnot" //Zavolá funkci FC1, pro převod hodnot
_001: NOP 0 //Prázdná instrukce
-----
NETWORK 2 //Spuštění časovačů pro jednu periodu
  AN I 0.0 //Negace vstupní adresy
  JNB _002 //Skočí na návěští _002, když RLO=0
  CALL "casovani_period_1" //Zavolá funkci FC2, pro časování period
_002: NOP 0 //Prázdná instrukce
-----
NETWORK 3 //Výpočet úhlů v [rad]
  AN I 0.0 //Negace vstupní adresy
  JNB _003 //Skočí na návěští _003, když RLO=0
  CALL "vypocet_uhlu_v_radianech" //Zavolá funkci FC4, pro výpočet úhlu v [rad]
_003: NOP 0 //Prázdná instrukce
-----
NETWORK 4 //Výpočet funkce sinus
  A( //Začátek vnoření
  AN I 0.0 //Negace vstupní adresy
  JNB _004 //Skočí na návěští _004, když RLO=0
  L "uhel_v_radianech" //Načtení hodnoty úhlu v daném čase
  SIN //Funkce sinus
  T "pomOB1.0" //Výsledek uloží do pomocné "pomOB1.0"
  AN 0V //Přetečení
  SAVE //Uloží RLO do BR
  CLR //Vynulování RLO
_004: A BR //Nastavení BR do log."1"
  ) //Konec vnoření
  JNB _005 //Skočí na návěští _005, když RLO=0
  L "pomOB1.0" //Načte hodnotu pomocné "pomOB1.0"
  L "amplituda" //Načte velikost hodnoty "amplituda" z FC1
  *R //Podělí načtené hodnoty
  T "sinus" //Transfer do proměnné "sinus"
_005: NOP 0 //Prázdná instrukce
-----
NETWORK 5 //Výpočet obdélníka
  L "casovani_periody_1" //Načte aktuální hodnotu časovače
  L "pom_perioda" //Načte hodnotu periody dělenou 0.2
  >R //Porovná načtené hodnoty
  = L 20.0 //Zapíše do adresy 20.0 výsledek z RLO když
  // MCR=1, jinak do adresy 20.0 zapíše "0".
  A L 20.0 //Nastavení adresy 20.0 do log."1"
  JNB _006 //Skočí na návěští _006, když RLO=0

```



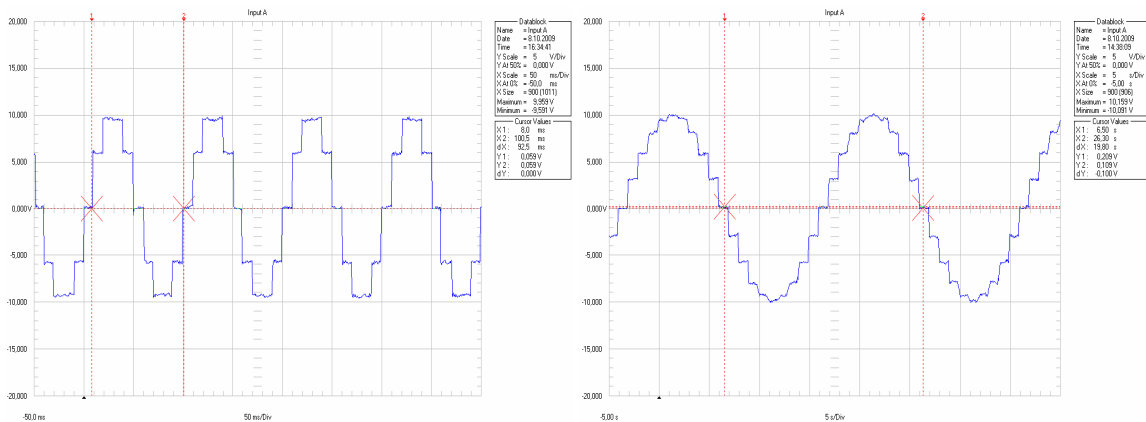
```

L    MW    1           //Načte hodnotu paměti MW1
L    1           //Načte1
*I           //Vynásobí načtené hodnoty
T    "obdelnik"     //Transfer do proměnné "obdelník"
_006: NOP 0           //Prázdná instrukce
AN   L      20.0     //Nastavení adresy 20.0 do log."0"
JNB  _007           //Skočí na návěští _007, když RLO=0
L    MW    1           //Načte hodnotu paměti MW1
L    -1           //Načte -1
*I           //Vynásobí načtené hodnoty
T    "obdelnik"     //Transfer do proměnné "obdelník"
_007: NOP 0           //Prázdná instrukce
-----
NETWORK 6           //Výpočet trojúhelníka
A(           //Začátek vnoření
AN   I      0.0     //Negace vstupní adresy
JNB  _008           //Skočí na návěští _008, když RLO=0
L    "casovani_periody_1" //Načte aktuální hodnotu časovače
L    "pocet_hodnot" //Načtení proměnné "počet_hodnot"
/R           //Podělení načtených hodnot
T    "pomOB1.1"     //Výsledek uloží do pomocné "pomOB1.1"
AN   OV           //Přetečení
SAVE           //Uloží RLO do BR
CLR           //Vynulování RLO
_008: A    BR           //Nastavení BR do log"1"
)           //Konec vnoření
JNB  _009           //Skočí na návěští _009, když RLO=0
L    "pomOB1.1"     //Načte hodnotu pomocné "pomOB1.1"
L    "amplituda"    //Načte velikost hodnoty "amplituda" z FC1
*R           //Podělí načtené hodnoty
T    "trojuhelnik" //Transfer do proměnné "trojuhelnik"
_009: NOP 0           //Prázdná instrukce
-----
NETWORK 7           //Zapsání všech průběhů na analogový výstup
AN   I      0.0     //Negace vstupní adresy
JNB  _00a           //Skočí na návěští _00a, když RLO=0
CALL "analogovy_vystup" //Zavolá funkci FC5, pro analogový výstup
_00a: NOP 0           //Prázdná instrukce
-----
NETWORK 8           //Zapsání všech průběhů do grafu OP 170B
AN   I      0.0     //Negace vstupní adresy
JNB  _00b           //Skočí na návěští _00b, když RLO=0
CALL "vystup_do-grafu" //Zavolá funkci FC6, pro výstup do grafu
_00b: NOP 0           //Prázdná instrukce
-----
// KONEC HLAVNÍ PROGRAM OB1 PRO SIMATIC C7

```

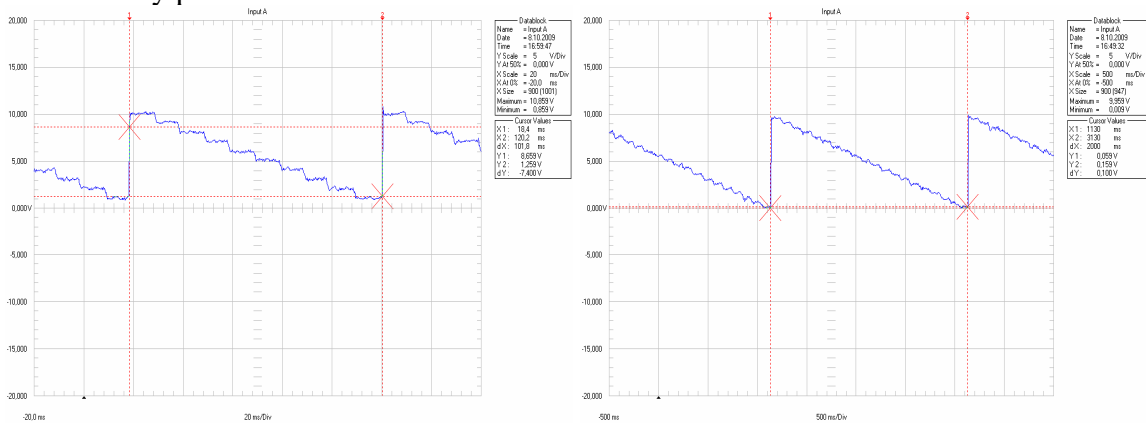
### 4.5.6 Výsledná měření

V této části bakalářské práce se budeme zabývat poznatky zjištěními při měření průběhů na analogových výstupech pomocí osciloskopu. Výstupní napětí D/A převodníku je +/- 10V, tomu odpovídá hodnota amplitudy 28000. Při této hodnotě byly prováděny všechny měření. Abychom dostali graf odpovídající sinusovému průběhu, musíme zadat na operátorském panelu velikost periody např.10s a tomu odpovídající počet bodů např.10. Tento poměr musí být zachován jinak dojde k deformaci průběhu. Tato skutečnost je dána algoritmem podle, podle kterého je tato funkce generována. Podle manuálu bychom mohli dosáhnout frekvence  $f=100\text{Hz}$ . My však z takto naprogramovaného PA dosáhneme nejvyšší frekvence  $f=10\text{Hz}$ . Je to dáno programem, protože využíváme dvou časovačů a ty se bohužel nestihnou pro vyšší frekvenci mezi sebou navzájem přepínat.



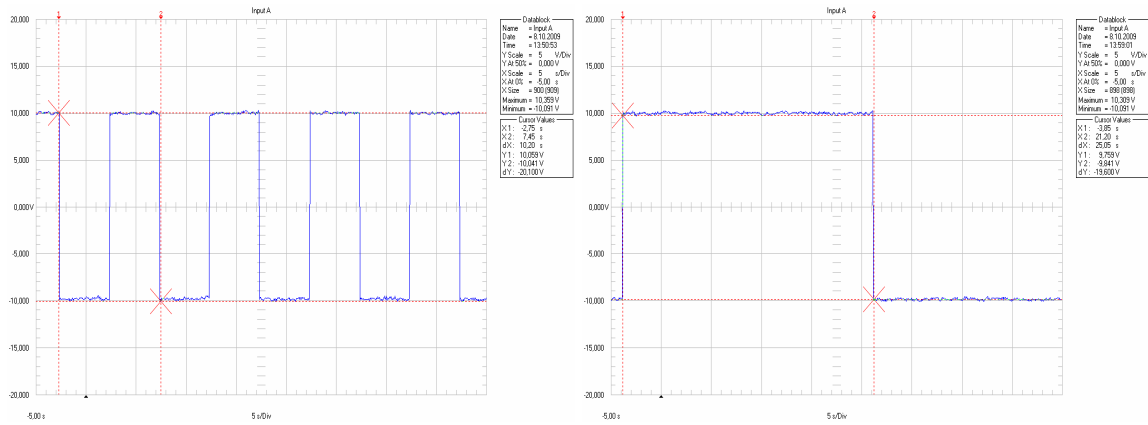
Graf 1 Grafy sinusového tvaru při periodě 0,1s-10bodů a 20s-20bodů

Graf trojúhelníkového tvaru, co se týče poměr počtu bodů a periody má podobné vlastnosti jako sinusový průběh.



Graf 42 Grafy trojúhelníkového tvaru při periodě 0,1s-10bodů a 20s-20bodů

Nejlépe vycházel průběh impulsního tvaru, který nabývá pouze dvou hodnot a to kladné nebo záporné amplitudy. Není omezován počtem bodů pro vygenerování jedné periody průběhu.



Graf 3 Grafy impulsního tvaru při periodě 10s a 50s

## 5. Závěr

Úkolem této bakalářské práce byla laboratorní ukázka generování časových průběhů sinusového, impulsního a trojúhelníkového tvaru programovatelným automatem SIMATIC C7 naprogramovaném ve vývojovém prostředí STEP 7. Obsluha a zadávání požadovaných hodnot byla realizována pomocí tlačítek na operátorském panelu. Při seznamování se s programovým prostředím STEP 7 jsem došel k názoru, že je do značné míry odlišný od prostředí STEP 7/MikroWin, který se používá pro programování 200 řad programovatelných automatů. Proto ne všechny příkazy a programové prvky se dají použít i ve STEP 7.

Při realizaci projektu byla odhalena nefunkční MMC karta, která vykazovala signalizaci systémové chyby. I přes snahu vytvořit tzv. „Image“ kartu se nepodařilo kartu opravit. Po její výměně konečně fungovalo všechno tak jak má.

Po naprogramování řídicího systému byla provedena pokusná měření pomocí osciloskopu, za účelem zjištění omezujících podmínek. Nejvyšší frekvence, které bylo možné dostat z D/A převodníku bylo  $f=10\text{Hz}$ . Podle manuálu by bylo možné za použití nejnižšího rozlišení časovače dosáhnout frekvence  $f=100\text{Hz}$ . Jelikož struktura řídicího programu obsahuje dva časovače, které se navzájem střídají, nebylo možné této frekvence dosáhnout, protože se nestíhali časovače mezi sebou navzájem přepínat. Pravděpodobně jsou i jiná optimálnější programová řešení, se kterými by šlo dosáhnout větších frekvencí generovaných signálů.

Takto generované průběhy lze použít jako pomocné vnitřní funkce automatu pro větší aplikace.

## Seznam použité literatury

- [1] KOPECKÝ, Dušan. Programovatelné logické automaty [online]. 18.9.2008, [cit 1.5.2009].  
Dostupný z: <<http://www.vscht.cz/ufmt/cs/pomucky/kopecky/docs/plc.pdf>>.
- [2] KAFKA, Jan., TESAŘ, Petr. Programování PLC Simatic S7 – 300 [online]. 2005, [cit.18.5.2009]. Dostupné z:  
<[http://www.spstrutnov.cz/o-skole/projekty/programovani-plc/programovani\\_plc.pdf](http://www.spstrutnov.cz/o-skole/projekty/programovani-plc/programovani_plc.pdf)>.
- [3] ZEZULKA, František. a kol. Programovatelné automaty [online]. 1.10.2003, [cit. 8.6.2009]. Dostupné z:  
<[http://www.advant.cz/Dokumentace/Programovatelné\\_automaty/Programovatelné\\_automaty-Skripta\\_FEKT\\_VUT\\_Brno.pdf](http://www.advant.cz/Dokumentace/Programovatelné_automaty/Programovatelné_automaty-Skripta_FEKT_VUT_Brno.pdf)>.
- [4] SIEMENS. Manuál pro SIMATIC konfigurátor. [PDF dokument]. 1.12.2004.  
Dostupné z:  
<[http://www1.siemens.cz/ad/current/index.php?ctxnh=ee5ad951ae&ctxp=doc\\_manualy&PHPSESSID=b22e4e1a33c26417e3a37c8c88ecf52c](http://www1.siemens.cz/ad/current/index.php?ctxnh=ee5ad951ae&ctxp=doc_manualy&PHPSESSID=b22e4e1a33c26417e3a37c8c88ecf52c)>.
- [5] SIEMENS. SIMATIC C7 Kompaktní řídicí systém s integrovaným operátorským rozhraním. [PDF dokument]. 11.3.2005, [cit. 9.6.2009]. Dostupné z:  
<[http://www.relko.cz/katalogy%2Fsiemens%2Fad5%2Fc7\\_prospekt\\_04\\_2005\\_cz.pdf](http://www.relko.cz/katalogy%2Fsiemens%2Fad5%2Fc7_prospekt_04_2005_cz.pdf)>.
- [6] SIEMENS. SIMATIC C7. [online]. [cit. 9.6.2009]. Dostupné z:  
<<http://www1.siemens.cz/ad/current/index.php?ctxnh=cbffe6312c&ctxp=home>>.
- [7] HANZÁLEK, Zdeněk., BERGET, Pavel. Průmyslová sběrnice Profibus [online]. 17.1.2004, [cit. 9.6.2009] Dostupné z:  
<<http://hw.cz/Rozhrani/ART1028-Prumyslova-sberrnice-Profibus.html>>.
- [8] TECOMAT FOXTROT. [PDF dokument]. 2008. září [cit. 9.6.2009]. Dostupné z:<  
<http://www.tecomat.cz/index.php?lang=cs&m1id=1&m2id=4&m3id=11&mid=12>>
- [9] BORČICKÝ, Tomáš. Aplikace sběrnice profibus. Brno, 2009. 8s.,28s., Diplomová práce na Fakultě strojního inženýrství-Ústav automatizace a informatiky, Vedoucí diplomové práce Ing. Zdeněk Němec, CSc.
- [10] ODBORNÉ ČASOPISY. Webová stránka projektu [online]. Dostupné z:  
<[http://www.odbornecasopisy.cz/index.php?id\\_document=28841](http://www.odbornecasopisy.cz/index.php?id_document=28841)>.
- [11] STYKAČE. Webová stránka projektu [online]. Dostupné z:  
<[http://stykace.cz/automatizace\\_cz.htm](http://stykace.cz/automatizace_cz.htm)>.

- [12] PROSYS. Webová stránka projektu [online]. Dostupné z: <[http://www.prosys.cz/news/News\\_04-2009-eCo.pdf](http://www.prosys.cz/news/News_04-2009-eCo.pdf)>.
- [13] BERGER, Hans., Automatizace se STEPem 7 – AWL. 28.3.200
- [14] Katedra řídicí techniky [databáze online]. Praha: ČVUT-FEL, 11.4.2002. Dostupné z URL <<http://dce.felk.cvut.cz/rs/plcs7315/dokumenty/>>.
- [15] DOČEKAL, Jan. PROFINET IO pro průmyslové aplikace. Praha, 2007. Bakalářská práce na Fakultě elektrotechniky-Katedra řídicí techniky, Vedoucí bakalářské práce Ing. Pavel Burget



**Přílohy**

// SEZNAM POUŽTÝCH SYMBOLŮ

// SYMBOL	ADRESA	DATOVÝ TYP	KOMENTÁŘ
Prevod_hodnot	FC 1	FC 1	
Casovani_period_1	FC 2	FC 2	
Vypocet_uhlu_v_radianech	FC 4	FC 4	
Analogovy_vystup	FC 5	FC 5	
Vystup_do_OP_graf	FC 6	FC 6	
sinus	MD 10	REAL	
trojuhelnik	MD 30	REAL	
pomFC1.0	MD 100	DINT	
pomFC1.1	MD 102	DINT	
pomFC2.14	MD 110	DINT	
pomFC2.24	MD 118	DINT	
pomFC4.0	MD 120	REAL	
pomFC4.1	MD 122	REAL	
pomOB1.0	MD 124	REAL	
pomOB1.1	MD 126	REAL	
pomFC1.2	MD 128	DINT	
pom_perioda	MD 130	REAL	
amplituda	MD 150	REAL	
pocet_hodnot	MD 160	REAL	
casovani_periody	MD 170	REAL	
uhel_v_radianech	MD 180	REAL	
sinus_analog_OUT	MD 200	DINT	
trojuhelnik_analog_OUT	MD 204	DINT	
perioda	MD 210	REAL	
obdelnik	MW 20	INT	
pomFC2.11	MW 104	WORD	
pomFC2.12	MW 106	INT	
pomFC2.13	MW 108	INT	
pomFC2.21	MW 112	WORD	
pomFC2.22	MW 114	INT	
pomFC2.23	MW 116	INT	
pomFC1.3	MW 230	INT	
pomFC1.4	MW 232	WORD	
CYCL_EXC	OB 1	OB 1	Cycle Execution
FILL	SFC 21	SFC 21	Initialize a Memory Area
VAT_1	VAT 1		



// FC1

```

-----
NETWORK 1 //Převod hodnoty "Amplituda" z INT na REAL
  A( //Začátek vnoření
  AN I 0.0 //Negace vstupní adresy
  JNB _001 //Skočí na návěští _001, když RLO=0
  L MW 1 //Načte hodnotu paměti MW1
  ITD //Konverze z 16-bit Integer na 32-bit Integer
  T "pomFC1.0" //Transfer do proměnné "pomFC1.0"
  SET //Nastavení RLO do log."1"
  SAVE //Uloží RLO do BR
  CLR //Vynulování RLO
_001: A BR //Nastavení BR do log"1"
  ) //Konec vnoření
  JNB _002 //Skočí na návěští _002, když RLO=0
  L "pomFC1.0" //Načtení proměnné "pomFC1.0"
  DTR //Konverze z Double Integer na Floatin Point
  T "amplituda" //Transfer do proměnné "amplituda"
_002: NOP 0 //Prázdná instrukce
-----

```

```

-----
NETWORK 2 //Převod hodnoty "Perioda" z INT na REAL
  A( //Začátek vnoření
  AN I 0.0 //Negace vstupní adresy
  JNB _003 //Skočí na návěští _003, když RLO=0
  L MW 3 //Načte hodnotu paměti MW1
  ITD //Konverze z 16-bit Integer na 32-bit Integer
  T "pomFC1.1" //Transfer do proměnné "pomFC1.1"
  SET //Nastavení RLO do log."1"
  SAVE //Uloží RLO do BR
  CLR //Vynulování RLO
_003: A BR //Nastavení BR do log"1"
  ) //Konec vnoření
  JNB _004 //Skočí na návěští _004, když RLO=0
  L "pomFC1.1" //Načtení proměnné "pomFC1.1"
  DTR //Konverze z Double Integer na Floatin Point
  T "perioda" //Transfer do proměnné "perioda"
_004: NOP 0 //Prázdná instrukce
-----

```

```

-----
NETWORK 3 //Převod hodnoty "Pocet hodnot" z INT na REAL
  A( //Začátek vnoření
  AN I 0.0 //Negace vstupní adresy
  JNB _005 //Skočí na návěští _005, když RLO=0
  L MW 5 //Načte hodnotu paměti MW5
  ITD //Konverze z 16-bit Integer na 32-bit Integer
  T "pomFC1.2" //Transfer do proměnné "pomFC1.2"
  SET //Nastavení RLO do log."1"
  SAVE //Uloží RLO do BR
-----

```

```

        CLR                                //Vynulování RLO
_005: A    BR                                //Nastavení BR do log"1"
    )                                        //Konec vnoření
        JNB _006                            //Skočí na návěští _006, když RLO=0
        L    "pomFC1.2"                    //Načtení proměnné "pomFC1.2"
        DTR                                //Konverze z Double Integer na Floatin Point
        T    "pocet_hodnot"                //Transfer do proměnné "pocet_hodnot"
_006: NOP  0                                //Prázdná instrukce
-----
NETWORK 4                                //Pomocná hodnota periody pro časování obdélníka
        AN    I    0.0                      //Negace vstupní adresy
        JNB _007                            //Skočí na návěští _007, když RLO=0
        L    "perioda"                    //Načtení proměnné "perioda"
        L    2.000000e-001                //Načte hodnotu 0.2
        /R                                  //Podělení načtených hodnot
        T    "pom_perioda"                //Transfer do proměnné "pom_perioda"
_007: NOP  0                                //Prázdná instrukce
-----
NETWORK 5                                //Převod hodnoty "Perioda" z INT na S5T#
        A(                                  //Začátek vnoření
        A(                                  //Začátek vnoření
        AN    I    0.0                      //Negace vstupní adresy
        JNB _008                            //Skočí na návěští _008, když RLO=0
        L    MW    3                      //Načte hodnotu paměti MW3
        L    100                          //Načte hodnotu 100
        *I                                  //Vynásobí načtené hodnoty
        T    "pomFC1.3"                    //Transfer do proměnné "pomFC1.3"
        AN    OV                            //Přetečení
        SAVE                                //Uloží RLO do BR
        CLR                                //Vynulování RLO
_008: A    BR                                //Nastavení BR do log"1"
    )                                        //Konec vnoření
        JNB _009                            //Skočí na návěští _009, když RLO=0
        L    "pomFC1.3"                    //Načtení proměnné "pomFC1.3"
        ITB                                //Konverze Integer do BCD
        T    "pomFC1.4"                    //Transfer do proměnné "pomFC1.4"
        AN    OV                            //Přetečení
        SAVE                                //Uloží RLO do BR
        CLR                                //Vynulování RLO
_009: A    BR                                //Nastavení BR do log"1"
    )                                        //Konec vnoření
        JNB _00a                            //Skočí na návěští _00a, když RLO=0
        L    "pomFC1.4"                    //Načtení proměnné "pomFC1.4"
        T    MW    220                      //Transfer do paměti MW220
_00a: NOP  0                                //Prázdná instrukce
-----
// KONEC FC1

```

// FC2

```

-----
NETWORK 1 //Časování period
  AN I 1.0 //Negace vstupní adresy
  L MW 220 //Načte hodnotu paměti MW220
  SP T 1 //Start adresy časovače, RLO z "0" do "1"
  NOP 0 //Prázdná instrukce
  NOP 0 //Prázdná instrukce
  LC T 1 //Načtení aktuální časové hodnoty z akumulátoru
  T "pomFC2.21" //Transfer do proměnné "pomFC2.21"
  A T 1 //Nastavení adresy časovače do log."1"
  = L 0.0 //Zapíše do adresy 0.0 výsledek z RLO když
           MCR=1, jinak do adresy 0.0 zapíše "0".

  AN L 0.0 //Nastavení adresy 0.0 do log."0"
  L MW 220 //Načte hodnotu paměti MW220
  SP T 2 //Start adresy časovače, RLO z "0" do "1"
  NOP 0 //Prázdná instrukce
  NOP 0 //Prázdná instrukce
  LC T 2 //Načtení aktuální časové hodnoty z akumulátoru
  T "pomFC2.11" //Transfer do proměnné "pomFC2.11"
  A T 2 //Nastavení adresy časovače do log."1"
  = L 0.1 //Zapíše do adresy 0.1 výsledek z RLO když
           MCR=1, jinak do adresy 0.1 zapíše "0".

  A L 0.1 //Nastavení adresy 0.1 do log."1"
  BLD 102 //Program displej
  = I 1.0 //Zapíše do adresy 1.0 výsledek z RLO když
           MCR=1, jinak do adresy 1.0 zapíše "0".

  A L 0.1 //Nastavení adresy 0.1 do log."1"
  JNB _001 //Skočí na návěští _001, když RLO=0
  L "pomFC2.11" //Načtení proměnné "pomFC2.11"
  BTI //Konverze z BCD na INT
  T "pomFC2.12" //Transfer do proměnné "pomFC2.12"
  SET //Nastavení RLO do log."1"
  SAVE //Uloží RLO do BR
  CLR //Vynulování RLO
_001: A BR //Nastavení BR do log"1"
     = L 0.2 //Zapíše do adresy 0.2 výsledek z RLO když
           MCR=1, jinak do adresy 0.2 zapíše "0".

  A( //Začátek vnoření
  A L 0.2 //Nastavení adresy 0.2 do log."1"
  JNB _002 //Skočí na návěští _002, když RLO=0
  L "pomFC2.13" //Načtení proměnné "pomFC2.13"
  ITD //Konverze Integer do BCD
  T "pomFC2.14" //Transfer do proměnné "pomFC2.13"
  SET //Nastavení RLO do log."1"
  SAVE //Uloží RLO do BR
  CLR //Vynulování RLO

```

```

_002: A    BR                //Nastavení BR do log"1"
      )                    //Konec vnoření
      JNB  _003              //Skočí na návěští _003, když RLO=0
      L    "pomFC2.14"      //Načtení proměnné "pomFC2.14"
      DTR                //Konverze z Double Integer na Floatin Point
      T    "casovani_periody" //Transfer do proměnné "casovani_periody"
_003: NOP  0                //Prázdná instrukce
      A    L    0.2          //Nastavení adresy 0.2 do log."1"
      JNB  _004              //Skočí na návěští _004, když RLO=0
      L    "pomFC2.12"      //Načtení proměnné "pomFC2.12"
      L    10                //Načte hodnotu 10
      /I                    //Podělení načtených hodnot
      T    "pomFC2.13"      //Transfer do proměnné "pomFC2.13"
_004: NOP  0                //Prázdná instrukce
      A    L    0.0          //Nastavení adresy 0.0 do log."1"
      JNB  _005              //Skočí na návěští _005, když RLO=0
      L    "pomFC2.21"      //Načtení proměnné "pomFC2.21"
      BTI                    //Konverze z BCD na INT
      T    "pomFC2.22"      //Transfer do proměnné "pomFC2.22"
      SET                    //Nastavení RLO do log."1"
      SAVE                  //Uloží RLO do BR
      CLR                    //Vynulování RLO
_005: A    BR                //Nastavení BR do log"1"
      =    L    0.1          //Zapíše do adresy 0.1 výsledek z RLO když
                               MCR=1, jinak do adresy 0.1 zapíše "0".
      A(                    //Začátek vnoření
      A    L    0.1          //Nastavení adresy 0.1 do log."1"
      JNB  _006              //Skočí na návěští _006, když RLO=0
      L    "pomFC2.23"      //Načtení proměnné "pomFC2.23"
      ITD                    //Konverze Integer do BCD
      T    "pomFC2.24"      //Transfer do proměnné "pomFC2.24"
      SET                    //Nastavení RLO do log."1"
      SAVE                  //Uloží RLO do BR
      CLR                    //Vynulování RLO
_006: A    BR                //Nastavení BR do log"1"
      )                    //Konec vnoření
      JNB  _007              //Skočí na návěští _007, když RLO=0
      L    "pomFC2.24"      //Načtení proměnné "pomFC2.24"
      DTR                //Konverze z Double Integer na Floatin Point
      T    "casovani_periody" //Transfer do proměnné"casovani_periody"
_007: NOP  0                //Prázdná instrukce
      A    L    0.1          //Nastavení adresy 0.1 do log."1"
      JNB  _008              //Skočí na návěští _008, když RLO=0
      L    "pomFC2.22"      //Načtení proměnné "pomFC2.22"
      L    10                //Načte hodnotu 10
      /I                    //Podělení načtených hodnot
      T    "pomFC2.23"      //Transfer do proměnné "pomFC2.23"
_008: NOP  0                //Prázdná instrukce

```

---

```
// KONEC FC2
```

---

// FC4

---

```

NETWORK 1 //Výpočet úhlu v radiánech
  A( //Začátek vnoření
  A( //Začátek vnoření
  AN I 0.0 //Negace vstupní adresy
  JNB _001 //Skočí na návěští _001, když RLO=0
  L "casovani_periody" //Načtení proměnné "casovani_periody"
  L "pocet_hodnot" //Načtení proměnné "pocet_hodnot"
  /R //Podělení načtených hodnot
  T "pomFC4.0" //Transfer do proměnné "pomFC4.0"
  AN OV //Přetečení
  SAVE //Uloží RLO do BR
  CLR //Vynulování RLO
_001: A BR //Nastavení BR do log"1"
) //Konec vnoření
  JNB _002 //Skočí na návěští _002, když RLO=0
  L "pomFC4.0" //Načtení proměnné "pomFC4.0"
  L 2.000000e+000 //Načte hodnotu 2
  *R //Podělení načtených hodnot
  T "pomFC4.1" //Transfer do proměnné "pomFC4.1"
  AN OV //Přetečení
  SAVE //Uloží RLO do BR
  CLR //Vynulování RLO
_002: A BR //Nastavení BR do log"1"
) //Konec vnoření
  JNB _003 //Skočí na návěští _003, když RLO=0
  L "pomFC4.1" //Načtení proměnné "pomFC4.1"
  L 3.140000e+000 //Načte hodnotu 3,14
  *R //Podělení načtených hodnot
  T "uhel_v_radianech" //Transfer do proměnné "uhel_v_radianech"
_003: NOP 0 //Prázdná instrukce

```

---

// KONEC FC4

---

---

```
// FC5
```

```
-----  
NETWORK 1 //Převod hodnoty "sinus" z REAL na DINT  
  A I 0.0 //Neaktivní vstupní adresa  
  JNB _001 //Skočí na návěští _001, když RLO=0  
  L "sinus" //Načtení proměnné "sinus"  
  TRUNC //Konverze z Floating Point na Integer  
  T "sinus_analog_OUT" //Transfer do proměnné "sinus_analog_OUT"  
_001: NOP 0 //Prázdná instrukce
```

```
-----  
NETWORK 2 //Výstup "sinus" na analogový výstup  
  A I 0.0 //Neaktivní vstupní adresa  
  JNB _002 //Skočí na návěští _002, když RLO=0  
  L "sinus_analog_OUT" //Načtení proměnné "sinus_analog_OUT"  
  T PQW 752 //Transfer na analogový výstup adresy 752  
_002: NOP 0 //Prázdná instrukce
```

```
-----  
NETWORK 3 //Výstup "obdelnik" na analogový výstup  
  A I 0.0 //Neaktivní vstupní adresa  
  JNB _003 //Skočí na návěští _003, když RLO=0  
  L "obdelnik" //Načtení proměnné "obdelnik"  
  T PQW 754 //Transfer na analogový výstup adresy 754  
_003: NOP 0 //Prázdná instrukce
```

```
-----  
NETWORK 4 //Převod hodnoty "trojuhelnik" z REAL na DINT  
  A I 0.0 //Neaktivní vstupní adresa  
  JNB _004 //Skočí na návěští _004, když RLO=0  
  L "trojuhelnik" //Načtení proměnné "trojuhelnik"  
  TRUNC //Konverze z Floating Point na Integer  
  T "trojuhelnik_analog_OUT"  
_004: NOP 0 //Prázdná instrukce
```

```
-----  
NETWORK 5 //Výstup "trojuhelnik" na analogový výstup  
  A I 0.0 //Neaktivní vstupní adresa  
  JNB _005 //Skočí na návěští _005, když RLO=0  
  L "trojuhelnik_analog_OUT" //Načtení proměnné "trojuhelnik_analog_OUT"  
  T PQW 754 //Transfer na analogový výstup adresy 754  
_005: NOP 0 //Prázdná instrukce
```

```
-----  
// KONEC FC5
```

---

// FC6

---

```
NETWORK 1 //Výstup hodnoty "sinus" do grafu OP 170B
  AN I 0.0 //Negace vstupní adresy
  JNB _001 //Skočí na návěští _001, když RLO=0
  L "sinus" //Načtení proměnné "sinus"
  L 2.800000e+003 //Načte hodnotu 2800
  /R //Podělení načtených hodnot
  T MD 15 //Transfer do adresy paměti MW15
_001: NOP 0 //Prázdná instrukce
```

---

```
NETWORK 2 //Výstup hodnoty "obdelnik" do grafu OP 170B
  AN I 0.0 //Negace vstupní adresy
  JNB _002 //Skočí na návěští _002, když RLO=0
  L "obdelnik" //Načtení proměnné "obdelnik"
  L 2800 //Načte hodnotu 2800
  /I //Podělení načtených hodnot
  T MW 25 //Transfer do adresy paměti MW25
_002: NOP 0 //Prázdná instrukce
```

---

```
NETWORK 3 //Výstup hodnoty "trojuhelnik" do grafu OP 170B
  AN I 0.0 //Negace vstupní adresy
  JNB _003 //Skočí na návěští _003, když RLO=0
  L "trojuhelnik" //Načtení proměnné "trojuhelnik"
  L 2.800000e+003 //Načte hodnotu 2800
  /R //Podělení načtených hodnot
  T MD 35 //Transfer do adresy paměti MW35
_003: NOP 0 //Prázdná instrukce
```

---

// KONEC FC6

### **Obsah přiloženého DVD**

- Tato bakalářská práce ve formátu Adobe Acrobat Reader [PDF]
- Zdrojový program pro řídicí systém SIMATIC C7-635
- Grafický program pro operátorský panel OP 170B