SLOVENSKÁ TECHNICKÁ UNIVERZITA

Fakulta chemickej a potravinárskej technológie

Katedra informatizácie a riadenia procesov

Programovanie PLC SIMATIC 300

(Základná príručka)

Štefan Kožka, Michal Kvasnica

September 2001

Programovanie PLC SIMATIC 300

Tento manuál poskytuje základné informácie týkajúce sa programovania a konfigurácie priemyselného riadiaceho systému SIMATIC 300, ktorá je realizovaná v programe Step-7. Manuál obsahuje nasledujúce časti:

1. Typy premenných
2. Typy blokov
3. Vytvorenie projektu
4. Konfigurácia siete
5. Konfigurácia I/O modulov
6. Konfigurácia premenných
7. Programovanie blokov
8. Kopírovanie programu do pamäte procesora
9. Testovanie logických obvodov
10. Diagnostika CPU
11. Vizualizačný systém WinCC

1. Typy premenných

V Step-7 je možné používať nasledujúce dátové typy:

Typ	Veľkosť v bitoch	Formát voľby	Max-Min hodnota
BOOL	1	Boolean text	True False
BYTE	8	16	B16#0 B16#FF
WORD	16	2 16 BCD 10	2. 0 2#1111_1111_1111_1111 W#16#0 W#16#FFFF C#0 C#999 B#(0.0) B#(255.255)
DWORD	32	2 16 10	2#0 2#111_1111_1111_1111_1111_1111_1111_1111 DW#16#0000_0000 DW#16#FFFF_FFFF B#(0,0,0,0) B#(255,255,255,255)
INT	16	10	-32768 32767
DINT	32	10	L#2147483648

			L#2147483647
REAL	32	IEEE	±1.175495E-38 ±3.402823E+38
S5TIME	32	IEC [s] krok 1ms	T#24D_20H_31M_23S_648MS T#24D_20H_31M_23S_647MS
DATE	16	IEC krok 1 deň	D#1990-1-1 D#2168-12-31
TIME_OF_DAY	32	TIME krok 1ms	TOD#0:0:0.0 TOD#23:59:59.999
CHAR	8	ASCII	‘A‘,‘B‘

2. Typy blokov

V programe Step 7 je možno pracovať s nasledujúcimi blokmi:

OB – organizačné bloky. Udávajú štruktúru uživateľského programu. Blok **OB1** reprezentuje hlavný program, ktorý pracuje v cyklickom režime .

FB – funkčné bloky. Samostatne programovateľný blok. Obsahuje „pamäť“, ktorá umožňuje ukladanie vnútorných premenných do týchto blokov.

FC – funkcie. Obsahujú rutiny pre najčastejšie používané funkcie. Nemajú „pamäť“.

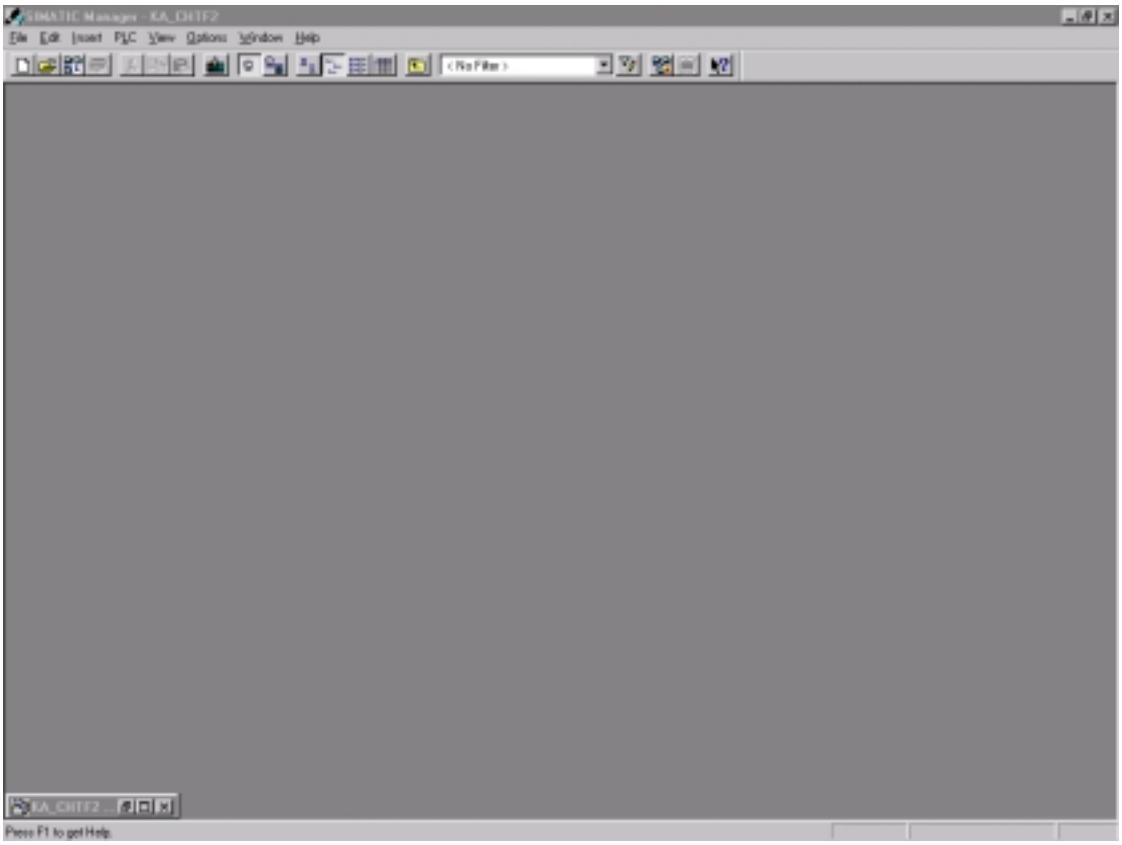
DB – dátové bloky. Využívajú sa uloženie užívateľských dát.

SFB a **SFC** – systémové funkčné bloky a systémové funkcie. Sú integrované priamo v S7 procesore (CPU) a umožňujú vstup do niektorých dôležitých systémových funkcií.

3. Vytvorenie projektu

Program Step 7 možno otvoriť kliknutím na ikonu  . Po jeho aktivácii sa otvorí nasledujúce okno, obr. 1.
Toto užívateľské rozhranie obsahuje tieto položky:

File – otváranie, organizácia a tlač projektu.
Edit, **Insert** – editovanie blokov a vkladanie programových komponentov.
PLC – nahrávanie programu do pamäte CPU a monitorovanie systému.
View, **Options**, **Window** – nastavenie okna, výber jazyka, nastavenie procesových dát.
Help – pomoc.



Obr. 1 Užívateľské rozhranie programu Step-7

Najskôr je potrebné vytvoriť nový projekt. Toto je možné realizovať pomocou ľavého kliku (LK) na nasledujúce položky **File –> 'New Project' Wizard…**. Otvorí sa okno **Step 7 Wizard: "New project"** ponuka **Introduction**. LK na **Next>**.
Objaví sa ponuka **Which CPU are you using in your project?**. V tejto ponuke je nutné vybrať typ používaného CPU. V našom prípade sa jedná o **CPU315-2DP**. Ďalej je potrebné priradiť **MPI** (Multi Point Interface) adresu. Zvyčajne je to **2**. LK na **Next>**.
Nachádzame sa teraz v ponuke **Which blocks do you want to add?**. K dispozícii máme zoznam všetkých **OB** blokov. Pre začiatok postačuje vybrať blok **OB1**.

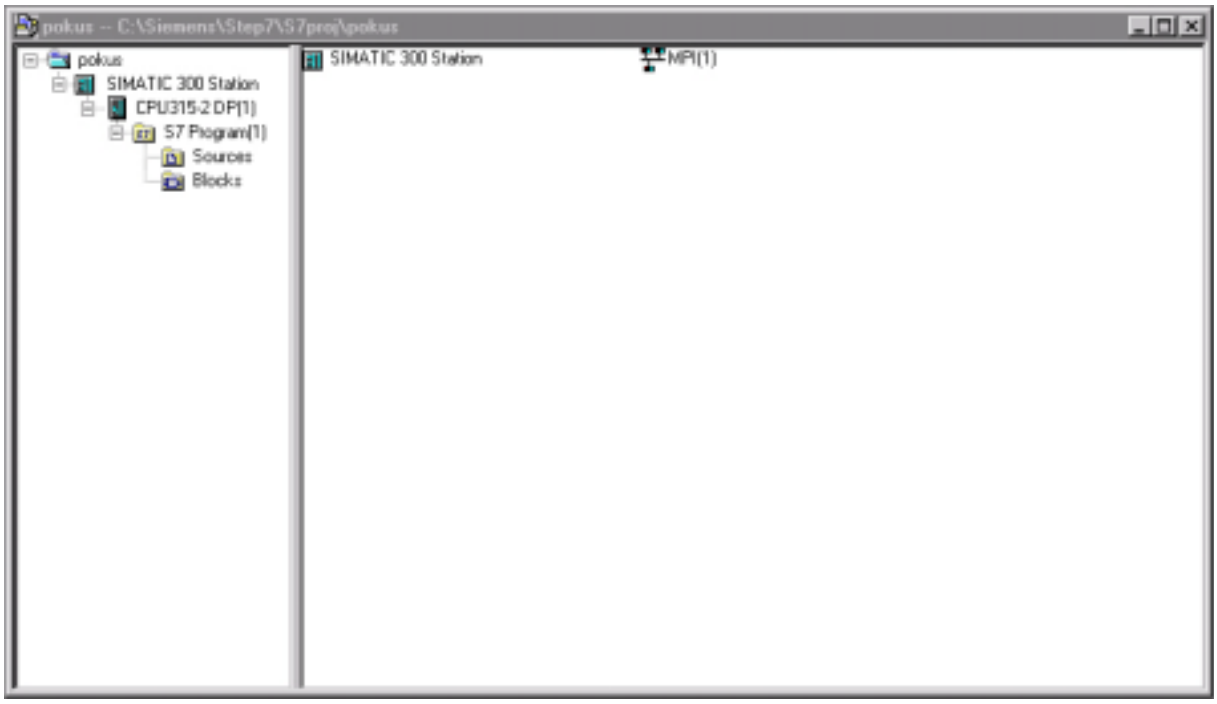
Poznámka Tieto bloky je možné pridávať aj počas programovania. Nie je nutné ich vybrať hneď na začiatku.

Ďalej si vyberieme jazyk v ktorom chceme príslušné bloky programovať **Language for Selected Blocks**.

STL – Statment list. Umožňuje programovanie pomocou príkazov.
LAD – Lader logic. Umožňuje programovanie pomocou schematického zapojenia.
FBD – Function block diagram. Umožňuje programovanie pomocou blokovej schémy.

Poznámka Všetky tri programovacie jazyky sú rovnocenné a možno ich zamieňať aj počas programovania. Napr. časť programu napíšeme pomocou STL a zvyšok pomocou FBD a pod.

LK na **Next>**.
V poslednej ponuke **What do you want to call your project?** zvolíme názov celého projektu. LK na **Finish**.
Po tomto kroku je automaticky vygenerované okno projektu (obr. 2).



Obr. 2 Okno vytvoreného projektu

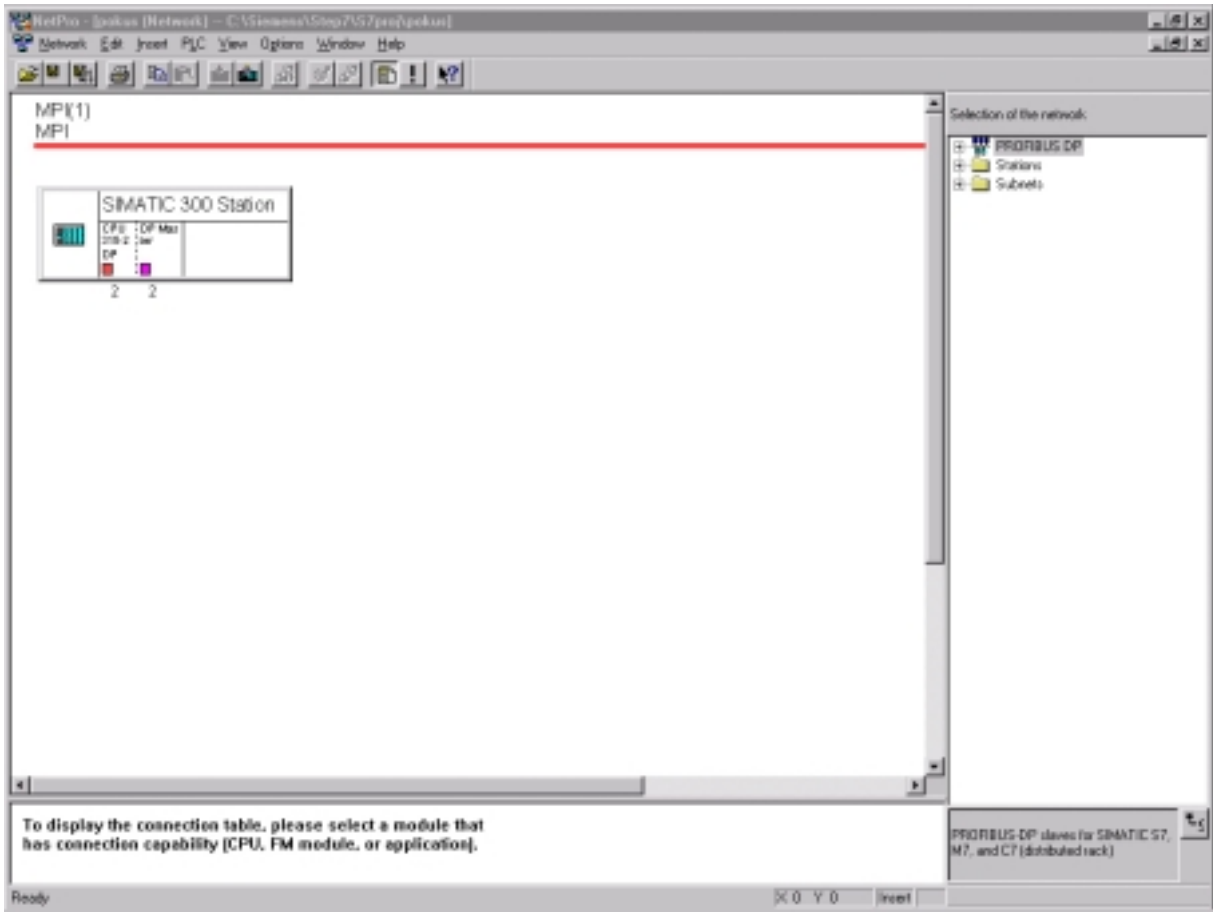
Na ľavom paneli tohoto projektu je zobrazená stromová štruktúra, pričom na najvyššej úrovni je projekt (POKUS), potom nasleduje pracovná stanica (SIMATIC 300), použitý typ CPU (CPU315-2DP) a jednotlivé programy. Na pravom paneli sa zobrazujú jednotlivé objekty patriace adresárom v ľavom paneli.

4. Konfigurácia siete

Ďalším krokom je vytvorenie komunikačnej siete medzi PC a príslušnými pracovnými stanicami.

V závislosti od použitej stanice a CPU je možné vytvoriť spojenie pomocou MPI, PROFIBUS-u, INDUSTRIAL ETHERNET-u resp. PTP. Taktiež je možné kombinovať jednotlivé spôsoby a tým vytvoriť sieť, ktorá umožní prenos dát medzi jednotlivými pracovnými stanicami a PC. Základné spojenie je tvorené pomocou MPI. V tomto manuáli je uvedený postup pre prípad vytvorenia komunikácie medzi PC a jednou pracovnou stanicou SIMATIC 300.

LK na názov projektu (najvyššia úroveň v stromovej štruktúre vytvoreného projektu). V pravom paneli sa objavia ikony pracovnej stanice SIMATIC 300 Station a použitej komunikácie MPI(1). 2x LK na MPI(1) umožní otvoriť okno **NetPro** v ktorom je možné konfigurovať danú komunikačnú sieť (obr. 3).



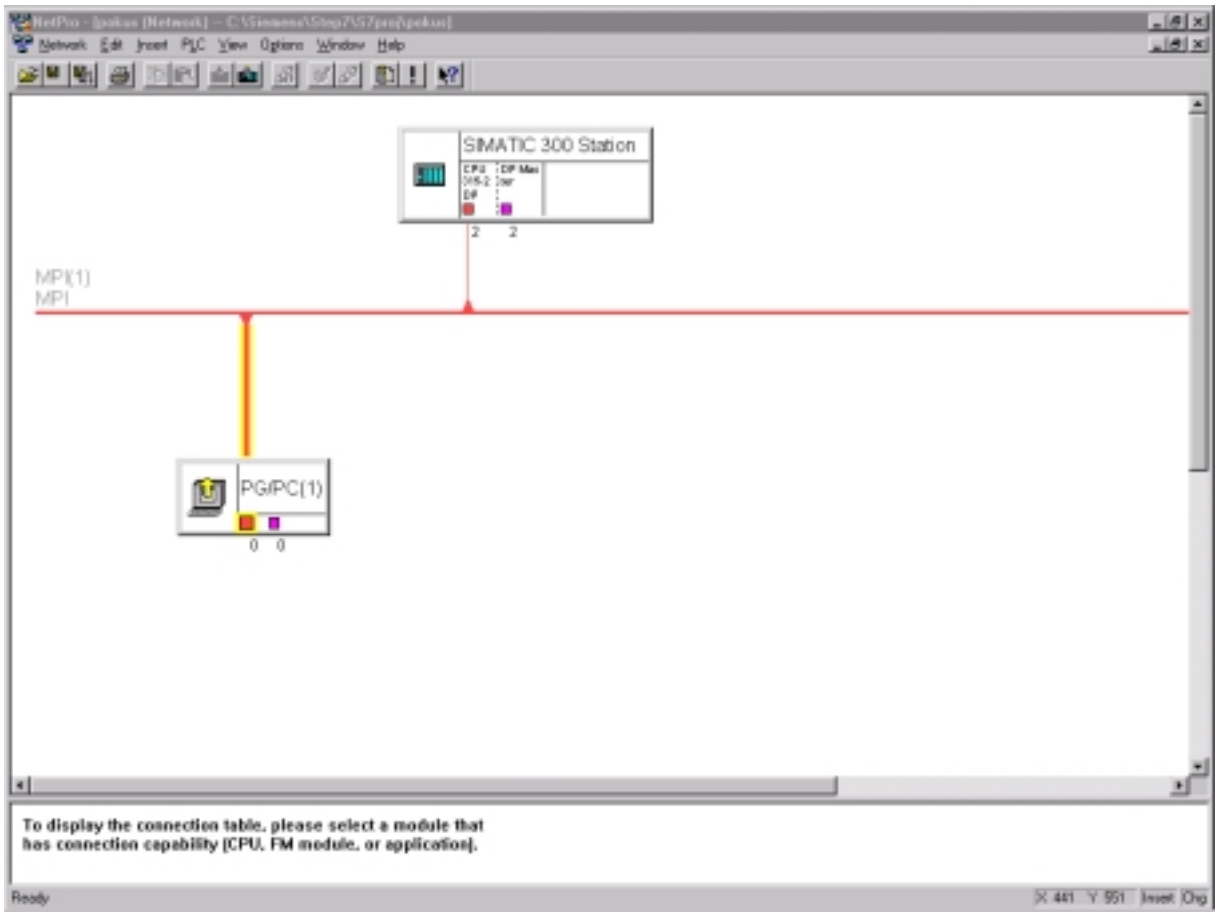
Obr. 3 Okno konfigurácie siete

Na pracovnej ploche je zobrazený MPI kábel spojenie (červená čiara) a blok pracovnej stanice. V pravej časť okna **Selection of the network** sa nachádzajú adresáre ďalších sietí, staníc a podsietí. **Selection of the network** je možné vypnúť alebo zapnúť pomocou ikony (vrchná časť okna **NetPro**). Pridržaním ľavého tlačítka (LT) myši je možné aj MPI kábel aj blok pracovnej stanice premiestniť na ľubovolné miesto pracovnej plochy. Keďže MPI kábel zabezpečuje spojenie medzi pracovnou stanicou a PC je potrebné umiestniť na pracovnú plochu aj blok PC. Zapnite (ak je to potrebné) zoznam **Selection of the network**. Otvorte adresár **Stations**. Pomocou LT presuňte ikonu PG/PC na pracovnú plochu.

Najskôr pripojíme blok pracovnej stanice na MPI kábel. Tento blok je rozdelený na tri časti. Prvá časť obsahuje názov pracovnej stanice **SIMATIC 300 Station**. Druhá časť **CPU315-2 DP** predstavuje rozhranie pre MPI kábel. Toto rozhranie je taktiež označené červeným štvorčekom. Tretia časť **DP Master** predstavuje rozhranie pre PROFIBUS kábel. Čísla pod uvedenými rozhraniami predstavujú tzv. **NOD**-y, adresy ktoré sú priradené pre každý objekt pripojený pomocou MPI alebo PROFIBUS spojenia.

Poznámka V rámci jedného typu spojenia je nevyhnutné aby mal každý objekt svoj vlastný **NOD**. Nesmie byť priradený dvom objektom rovnaký **NOD**.

Nasleduje krok spojenia bloku pracovnej stanice z MPI káblom. Nastavte sa na červený štvorček MPI rozhrania zvolenej pracovnej stanice. Stlačte LT a ťahajte myšou generovanú prerušovanú červenú čiaru smerom ku MPI káblu pokiaľ sa navzájom nespoja (prerušovaná červená čiara sa zmení na neprerušovanú červenú čiaru). Uvoľnite LT.
Nasleduje nastavenie rýchlosti prenosu dát. 2x LK na MPI rozhranie. Otvorí sa ponuka **Properties CPU 315-2 DP**. Vyberte záložku **General**. LK na tlačítko **Properties**. Otvorí sa ponuka **Properties – MPI interface CPU315-2 DP**. Vyberte záložku **Parameters**. LK na MPI v časti **Subnet**. Lk na tlačítko **Properties…**. Otvorí sa ponuka **Properties – MPI**. Vyberte záložku **Network settings**. Tu je možné zvoliť požadovanú rýchlosť prenosu dát **Transmission Rate**. Pre náš prípad postačuje rýchlosť 187.5 kbit/s. Potvrďte.
Po tejto operácii je zvolená pracovná stanica pripojená ku MPI káblu. Podobne je potrebné pripojiť aj blok **PG/PC** k MPI.
LK na blok **PG/PC**. Pravý klik (PK). Z ponúknutého menu vyberte **Assign PG/PC**. PC bude automaticky pripojený ku MPI káblu (objaví sa červeno-žltá čiara). **PG/PC** blok má podobne ako blok pracovnej stanice MPI a PROFIBUS rozhranie. 2x LK na MPI rozhranie **PG/PC** bloku (červený štvorček). Otvorí sa ponuka **Properties – MPI interface**. Zvoľte záložku **Parameters**. Nastavte **Address:** na 0. Týmto priradíte uvedenému bloku **NOD** 0. Rovnako nastavte aj **NOD** pre PROFIBUS rozhranie.
Nakonfigurovaná sieť je uvedená na obr. 4.



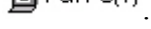
Obr. 4 Nakonfigurovaná sieť

Na záver je potrebné nakonfigurovanú sieť skontrolovať, nahrať na disk počítača a nahrať do pamäte CPU.

Kontrola – LK na menu **Network** (vrchná časť okna **NetPro**) **-> Check Consistency** po kontrole sa objaví informácia o zistených chybách. Ak je všetko v poriadku objaví sa hláška **No errors**. Zavrite okno s vygenerovanou hláškou.

Nahrávanie a kompilácia – LK na menu **Network -> Save and Compile…**. Objaví sa ponuka **Save and Compile**. Vyberte **Compile and Check Everything**. Opäť sa objaví informácia o nájdených chybách.

Nahrávanie do pamäte CPU – LK na blok pracovnej stanice. LK na menu **PLC -> Download -> Selected Stations**. Pri tomto kroku však treba mať už zapnutý aj PLC. Podrobnejšie informácie týkajúce sa nahrávania do pamäte CPU nájdete v kapitole 9.

Konfigurácia siete je ukončená. Zavrite okno **NetPro**. LK na názov projektu (najvyššia úroveň v stromovej štruktúre vytvoreného projektu). V pravom paneli sa objaví nová ikona 

.

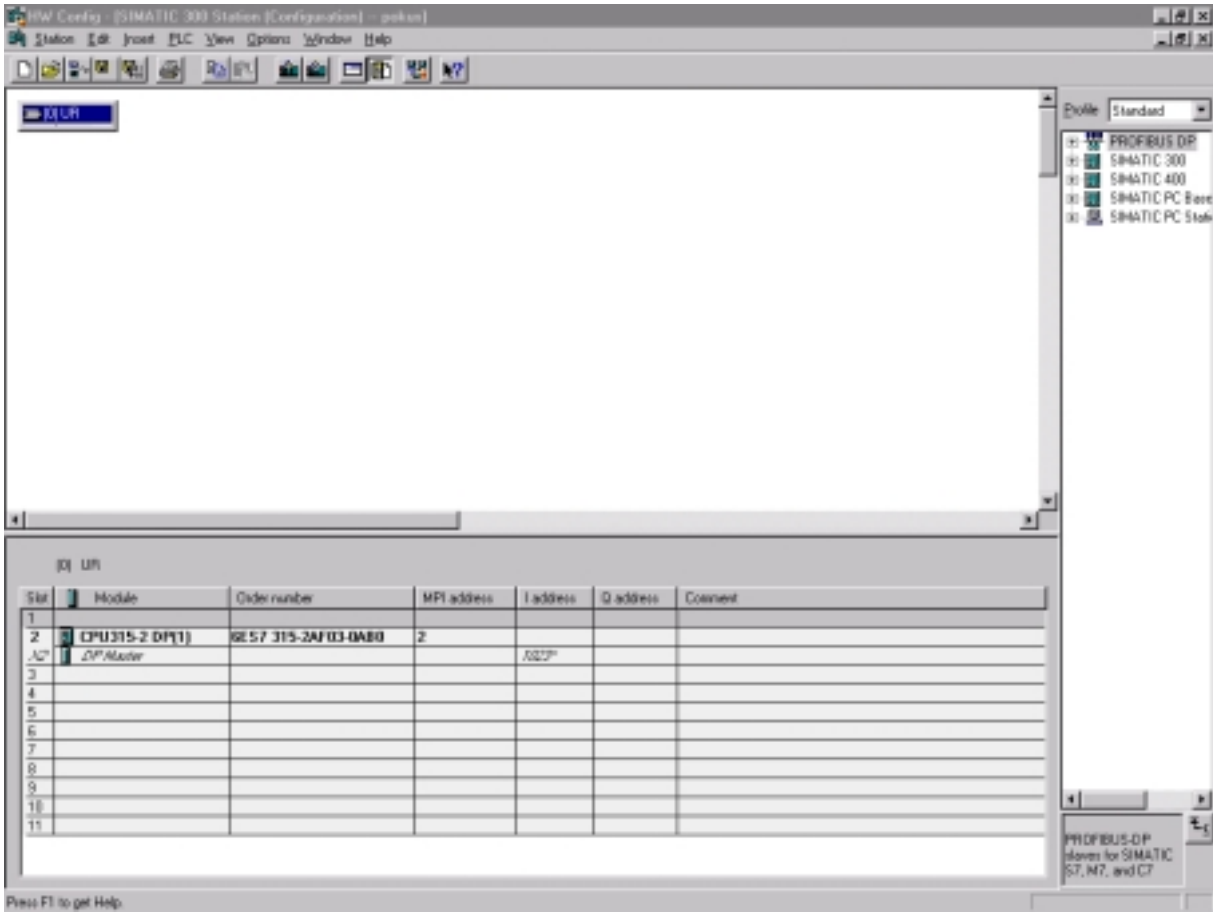
4.1 Niektoré pravidlá pri konfigurácii siete

1. Každý objekt musí mať rozdielny **NOD**.
2. CPU má rezervovaný **NOD** 2.
3. **PG/PC** má rezervovaný **NOD** 0.

5. Konfigurácia I/O modulov

Po úspešnej konfigurácii siete je možné pristúpiť ku konfigurácii použitých I/O (Input/Output) modulov.

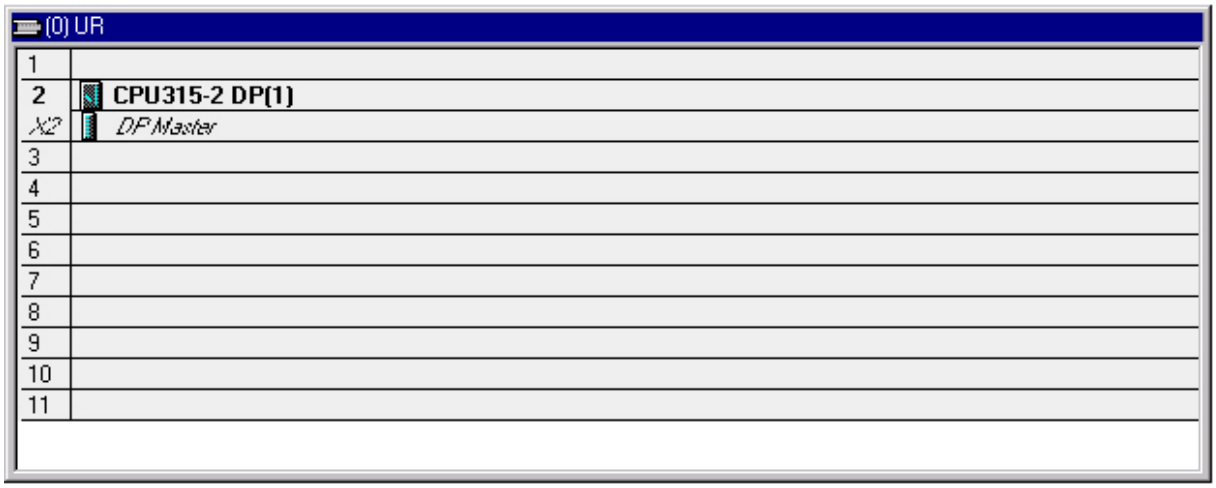
LK na názov pracovnej stanice (stromová štruktúra vytvoreného projektu). V pravom paneli sa objavia ikony hardwaru Hardware a použitého CPU CPU315-2 DP(1). 2x LK na Hardware umožní otvoriť okno **Hw Config** v ktorom je možné konfigurovať zapojené I/O moduly (obr. 5).



Obr. 5 Okno konfigurácie I/O modulov

Na pracovnej ploche uvedeného okna sa nachádza **Rail** (0) UR. **Rail** je možné rozvinúť stlačením LT napr. v pravom dolnom rohu s následným ťahaním myši po pracovnej ploche. Rozvinutý **Rail** je znázornený na nasledujúcom obrázku.

Jednotlivé riadky **Rail**-u predstavujú individuálne sloty do ktorých sú zapojené použité I/O moduly. Napr. v **Rail**-e 0 v slote 2 je umiestnený modul CPU (viď. Obr. 6).

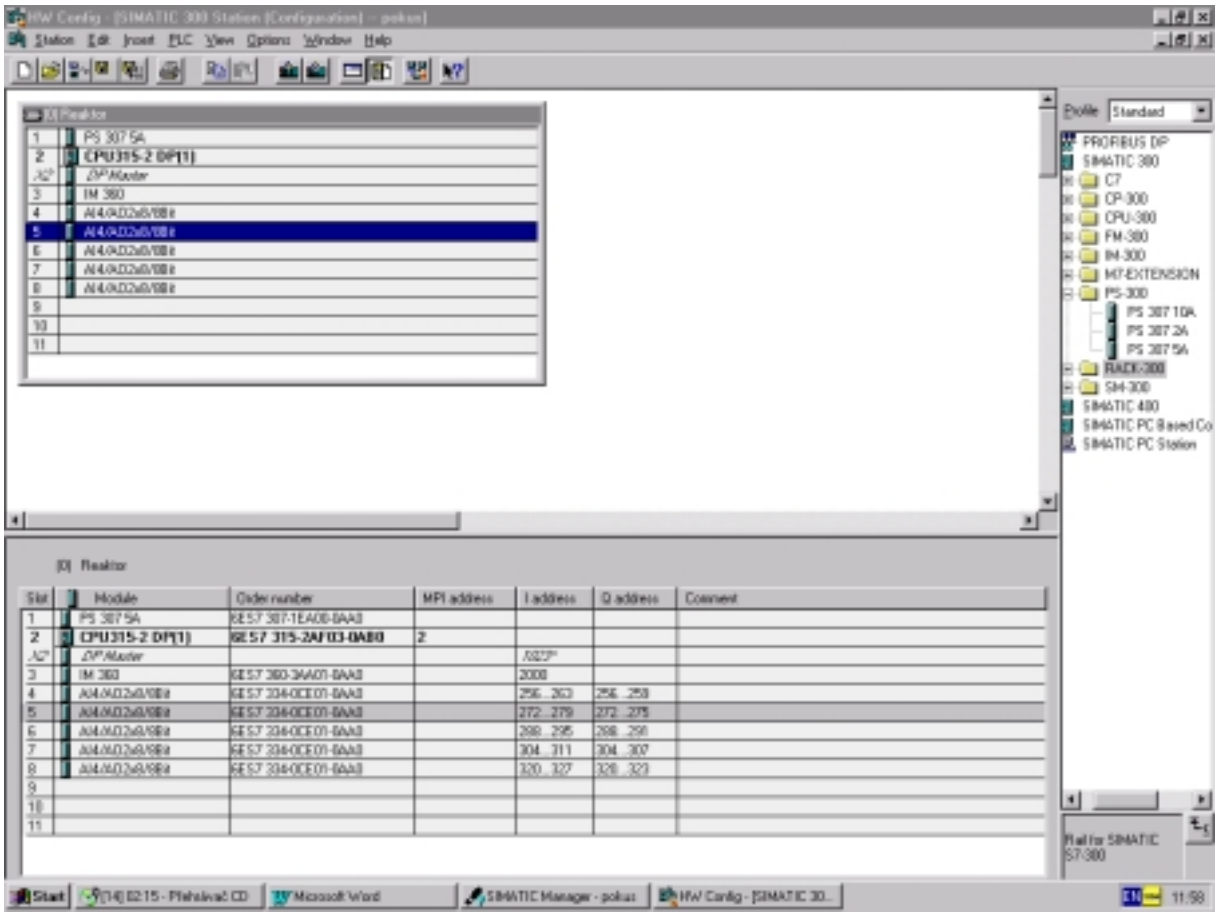


Obr. 6 Rozvinutý Rail.

Spodná časť okna **Hw Config** predstavuje konfiguračnú tabuľku, ktorá obsahuje MPI a I/O adresy. V pravej časti uvedeného okna sa nachádza katalóg použiteľných modulov. Možno ho zapnúť resp. vypnúť pomocou ikony v hornej časti okna **Hw Config**.
2x LK na názov rozvinutého **Rail**-u otvorí ponuku **Properties – UR – Rack 0**. V tejto ponuke je možné zadať nové meno **Rail**-u (napr. namiesto UR napíšeme Reaktor) a jeho poradové číslo.
V prípade, že je potrebné dodať ďalší **Rail**, potom je potrebné zapnúť katalóg použiteľných modulov. V tomto prípade otvoríme adresár **SIMATIC 300 -> RACK-300** v zapnutom katalógu. Pomocou stlačeného LT prenesieme ikonu Rail na pracovnú plochu okna **Hw Config**.
Konfigurácia I/O modulov spočíva v obsadzovaní jednotlivých voľných **Slot**-ov v danom **Rail**-e. Obsadzovanie je opäť realizované prostredníctvom stlačeného LT, pričom sú jednotlivé moduly prenášané s otvoreného katalógu do jednotlivých **Slot**-ov daného **Rail**-u. Napr. potrebujeme preniesť na prvý **Slot** napájací modul **PS 307 5A**. Najskôr otvoríme katalóg. Potom otvoríme adresár **SIMATIC 300 -> PS-300**. Z ponuky vyberieme ikonu modulu **PS 307 5A** a pomocou stlčeného LT ju prenesieme do 1 riadku rozvinutého **Rail**-u.
Podobne potom konfigurujeme aj zvyšné moduly. Nakonfigurovaný **Rail** je znázornený na obr. 7.
Po obsadení zvoleného **Slot**-u príslušným modulom sa tento modul objaví aj v kofiguračnej tabuľke (spodná časť okna **Hw Config**), kde jednotlivé stĺpce majú nasledujúci význam:

Slot – číslo daného slotu.
Module – typ modulu.
Order number – objednávacie číslo.
MPI address – MPI adresa ak je daný modul pripojený priamo na MPI kábel (napr. CPU).
I address – adresy vstupov do daného modulu ak ich obsahuje.
Q address – adresy výstupov z daného modulu ak ich obsahuje.
Comment – komentár napísaný užívateľom.

Poznámka Moduly uvedené v programe **Hw Config** sa musia presne zhodovať s reálne použitými modulmi.



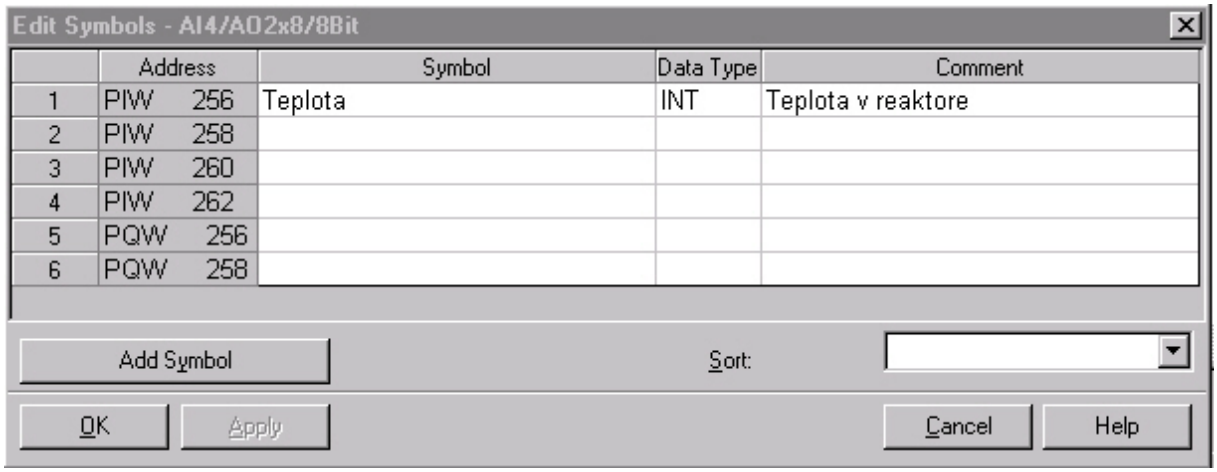
Obr. 7. Nakonfigurovaný rail.

Prehľad všetkých I/O adries je možné získať stlačením ikony  v hornej časti okna **Hw Config**. I/O adresy sú označované automaticky. Každý modul má začiatočnú adresu (adresa prvého kanálu). Adresy zvyšných kanálov v danom module potom závisia od začiatočnej adresy. Napr. obr. 7 má modul v slote 5 rezervované čísla pre vstupné kanály 272-279 a pre výstupné kanály taktiež od 272-279, pričom začiatočná adresa (adresa prvého kanálu) je v obidvoch prípadoch 272. Keďže sú jednotlivé adresy označené automaticky, potom môžeme pristúpiť k definovaniu jednotlivých symbolických premenných. Pomocou týchto premenných je možné zadefinovať vstupné resp. výstupné signály daného reálneho procesu na jednotlivé adresy použitých I/O modulov.
Napr. Na vstupný kanál číslo 256 chceme priviesť meranú teplotu v reaktore.
LK na **Slot** 4. PK a zo zobrazeného menu vyberte **Edit Symbolic Names…**. V otvorenom okne **Edit Symbols – AI4/AO2x8/8bit** je možné potom editovať príslušnú symbolickú premennú. Význam jednotlivých stĺpcov otvoreného okna je nasledovný:

Address – adresa (kanál) zvoleného modulu.
Symbol – názov zadefinovanej symbolickej premennej.
Data Type – dátový typ použitej symbolickej premennej (kapitola 1).
Comment – komentár.

Keďže chceme obsadiť kanál číslo 256 potom budeme definovať premennú do prvého riadku otvoreného okna **Address = PIW 256**. Názov symbolickej premennej nech je **Teplota** Dátový ty je možné v tomto prípade zvoliť buď **WORD alebo INT**. Zvoľme teda **INT**. Ako

Comment môžeme uviesť **Teplota v reaktore**. Editovanie je realizované 2x LK na príslušný riadok a stĺpec uvedeného okna.
Uvedené okno aj s naeditovanou symbolickou premennou je uvedené na obr. 8.



Obr. 8 Editovanie symbolických premenných

Na záver je potrebné nakonfigurovanú sieť skontrolovať, nahrať na disk počítača a nahrať do pamäte CPU.

Kontrola – LK na menu **Station** (v hornej časti okna **Hw Config**) **-> Consistency Check** po kontrole sa objaví informácia o zistených chybách. Ak je všetko v poriadku objaví sa hláška **No errors**.

Náhrávanie a kompilácia – LK na menu **Station -> Save and Compile…**.

Nahrávanie do pamäte CPU – LK na blok pracovnej stanice. LK na menu **PLC -> Download to Module…**. Pri tomto kroku však treba mať už zapnutý aj PLC. Podrobnejšie informácie týkajúce sa nahrávania do pamäte CPU nájdete v kapitole 9.

Konfigurácia I/O modulov je ukončená. Zavrite okno **Hw Config**.

5.1 Niektoré pravidlá pri konfigurácii I/O modulov

1. **Slot** 1 je rezervovaný pre PS moduly.
2. **Slot** 2 je rezervovaný pre CPU. Ak CPU v danom **Rail**-e nie je **Slot** 2 zostáva prázdny.
3. **Slot** 3 je rezervovaný pre IM (Interface Modul) moduly. Ak IM v danom **Rail**-e nie je **Slot** 3 zostáva prázdny.
4. **Slot** 4-11 rezervovaný pre I/O moduly.
5. Maximálny počet použitých **Rail**-ov je 4.

6. Konfigurácia premenných

Absolútna adresa Každý vstup a výstup má absolútnu adresu automaticky definovanú v **Hw Config** napr. I 1.5 (absolútna adresa digitálneho vstupu) alebo PIW 256 (absolútna adresa pre analógový vstup).

Symbolická premenná ktorou možno nahradiť absolútnu adresu. Každá absolútna adresa môže byť nahradená symbolickou premennou podľa vlastnej voľby.

V závere predchádzajúcej kapitoly sme uviedli spôsob definovania symbolických premenných priamo počas konfigurácie I/O modulov. Tieto premenné je možné definovať aj prostredníctvom symbolickej tabuľky.
LK na adresár **S7 Program** (stromová štruktúra vytvoreného projektu). V pravom paneli sa objavia ikony zdrojov Sources blokov Blocks a symbolov Symbols . 2x LK na Symbols umožní otvoriť okno **Symbol Editor** v ktorom je možné definovať alebo premenovávať už existujúce symbolické premenné. Význam jednotlivých stĺpcov v tomto okne je nasledujúci:

Symbol – názov symbolickej premennej.
Address – absolútna adresa (generovaná automaticky).
Data type – dátový typ (viď. kapitola 1).
Comment – komentár.

V okne **Symbol Editor**-u je možné priradiť symbolickú premennú nielen jednotlivým I/O adresám, ale aj použitým blokom napr. OB, FB, FC apod. V tomto prípade je v stĺpci **Address** ako absolútna adresa uvedený typ a číslo použitého bloku napr. **OB 1**. Rovnaké označenie je použité aj pre stĺpec **Data type**. Editovanie symbolickej tabuľky je zhodné s editovaním symbolických premenných v **Hw Config.**
Symbolickú tabuľku je možné dopĺňať aj počas programovania jednotlivých blokov. Nie je potrebné pred programovaním zadefinovať všetky symbolické premenné ktoré budú následne použité. Príklad symbolickej tabuľky je uvedený na obr. 9.

KA_CHTF2\SIMATIC 300 Station\CPU31...\Symbols

	Symbol	Address	Data type	Comment
1	DB_Konverzia_u	DB 5	FB 3	
2	DB_PID	DB 4	FB 41	Datovy blok pre PID
3	DB_poplach	DB 3	FB 2	Datovy blok pre poplachy
4	DI_STRNG	FC 5	FC 5	Double Integer to String
5	Kon_tep	FB 1	FB 1	Konverzia snimanych teplot
6	Konverzia_u	FB 3	FB 3	Kalibracna funkcia
7	PID	FB 41	FB 41	PID regulator
8	Poplach	FB 2	FB 2	Ohlasovanie poplachu
9	STRNG_I	FC 38	FC 38	String to Integer
10	Udaje	DB 2	DB 2	Vsetky udaje pre WinCC
11	vstup_1	PIW 256	INT	Teplota v reaktore
12	vstup_2	PIW 258	INT	Teplota chladiva na vstupe
13	vstup_3	PIW 260	INT	Teplota chladiva na vystupe
14	vystup_1	PQW 256	INT	Prietok chladiva
15				

Obr. 9 Symbolická tabuľka - **Symbol Editor**

Poznámka Ak boli k I/O adresám priradené symbolické premenné už v **Hw Config** tieto sú potom automaticky prenesené aj do symbolickej tabuľky (obr. 9).

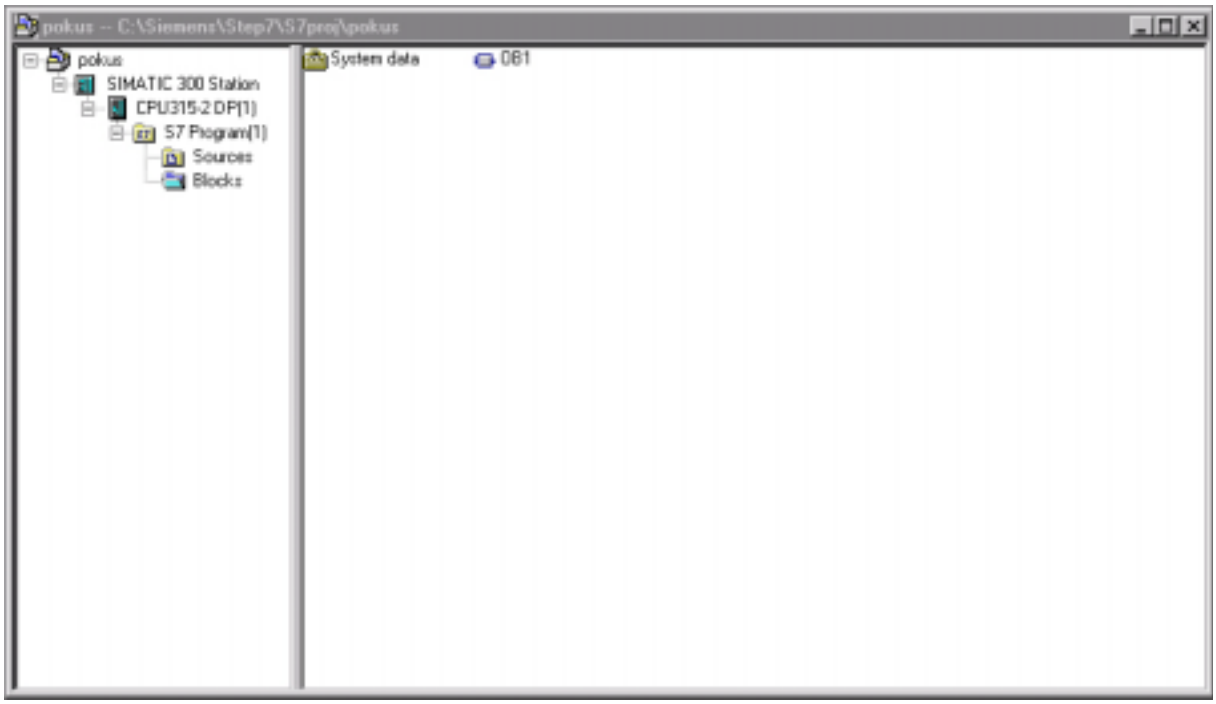
Po ukončení editácie zavrite okno **Symbol Editor**. Počas zatvárania potvrďte uloženie na disk počítača.

7. Programovanie blokov

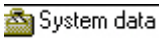
V tejto časti tohoto manuálu sa budeme zaoberať programovaním blokov uvedených v kapitole 2. Pomocou týchto blokov je možné naprogramovať všetky potrebné logické operácie ako aj riadenie daného reálneho procesu.

LK na adresár **S7 Program** (stromová štruktúra vytvoreného projektu). Ďalej 2x LK na 

. Po tomto kroku sa v pravom paneli otvoreného projektu objavia doposiaľ používané bloky (viď. obr.10)



Obr. 10 Okno vytvoreného projektu s informáciou o použitých blokoch

Ikona 

je vytvorená automaticky a obsahuje dáta vygenerované počas konfigurácie siete a I/O modulov. Blok 

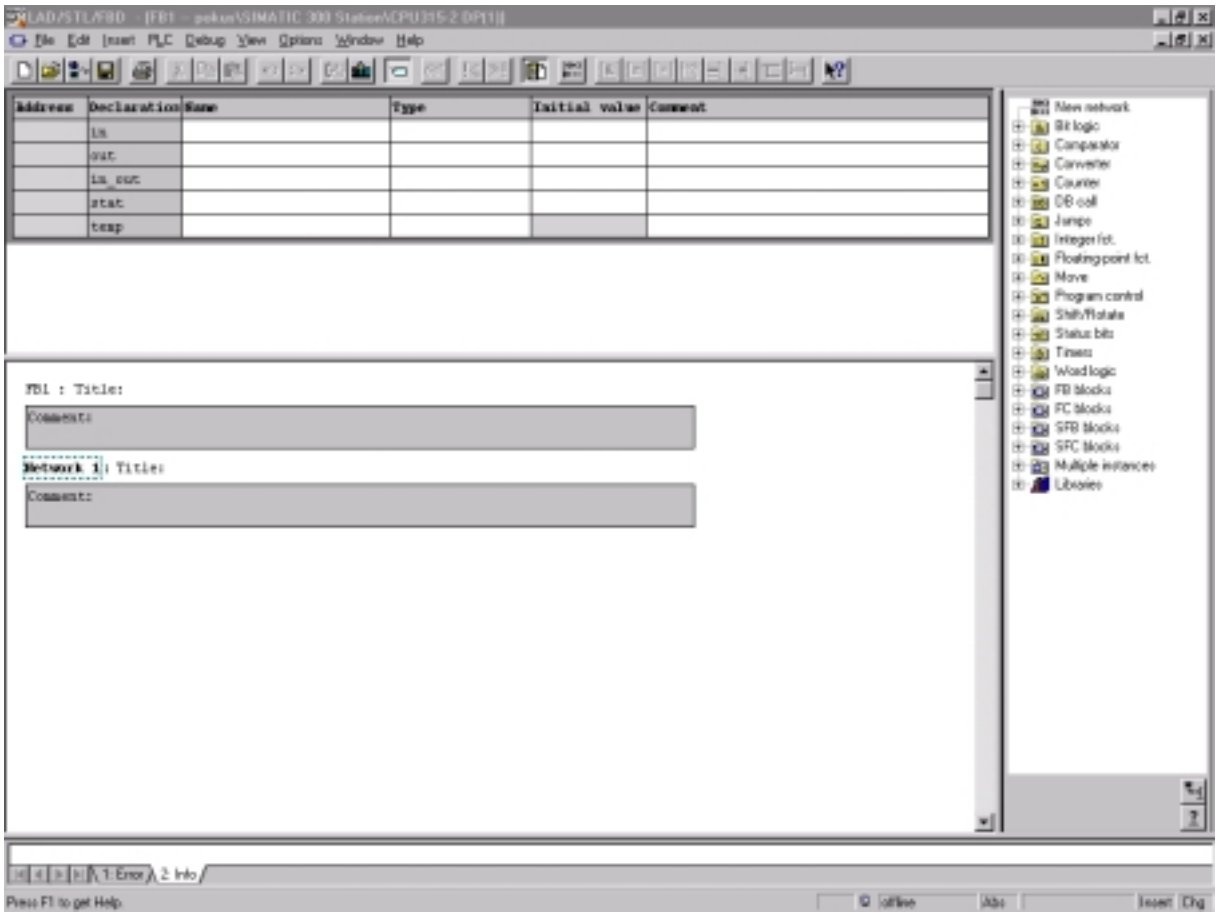
bol vygenerovaný počas vytvárania projektu. Ako už bolo povedané **OB1** predstavuje organizačný blok hlavného cyklu, ktorý spúšťa ostatné bloky.

7.1 Vkladanie ostatných blokov do projektu

PK na okno projektu. Z ponúknutého menu vyberte **Insert new object**. Zo zoznamu blokov potom zvoľte požadovaný typ bloku. (typy blokov viď. kapitola 2).

7.2 LAD/STL/FBD editor

2 x LK na ľubovoľný vytvorený blok je otvorené okno **LAD/STL/FBD** editoru. V tomto editore je možné programovať logické operácie ktoré má vykonávať daný blok (obr. 11).



Obr. 11 Okno LAD/STL/FBD editoru

Horná časť tohoto okna reprezentuje deklaračnú tabuľku, slúžiacu na deklarovanie lokálnych (vnútorných) premenných editovaného bloku.
Význam jednotlivých stĺpcov deklaračnej tabuľky je nasledujúci:

Address – adresa lokálnej premennej (generovaná automaticky)
Declaration – deklarovanie lokálnej premennej. Možno použiť nasledujúce premenné:
in vstupná premenná
out výstupná premenná
in_out vstupná/výstupná premenná
stat statická premenná
temp dočasná premenná
Type – typ premennej (viď. kapitola 1)
Initial value – začiatočná hodnota
Comment – komentár

Poznámka Pri premennej **temp** nie je možné zadať začiatočnú hodnotu. Ak je niektorá bunka deklaračnej tabuľky zadaná nesprávne, potom je uvedená hodnota zobrazená červenou farbou.

Po jednotlivých bunkách deklaračnej tabuľky je možné sa pohybovať pomocou myši resp. tabelátora.
Stredná časť okna **LAD/STL/FBD** editora predstavuje editovaciu plochu kde:

FB1 : Title: - umožňuje zadať názov daného bloku v tomto prípade je to blok **FB1**.

Comment: - komentár k danému bloku.
Network 1 : Title: - umožňuje zadať názov siete. Sieť slúži na rozdelenie komplexných algoritmov na viacero jednoduchších častí. Napr. je potrebné naprogramovať signalizáciu výšky hladiny v zásobníku kvapaliny. **Network 1:** bude obsahovať jednotlivé logické inštrukcie na zabezpečenie signalizácie vysokej hladiny a **Network 2** bude zase obsahovať inštrukcie na signalizáciu nízkej hladiny.
Comment: - komentár k danej sieti.

Plocha pod komentárom slúži na programovanie jednotlivých operácií.
Spodná časť **LAD/STL/FBD** poskytuje informáciu o vzniknutých chybách resp. informáciu o očakávaných typoch signálov.
Pravý panel **LAD/STL/FBD** editora predstavuje katalóg programových elementov. Tento katalóg je možné zapnúť resp. vypnúť pomocou ikony  .

7.3 Programovanie FB bloku
Uvažujme, že chceme naprogramovať súčin dvoch premenných typu INT. Nech je tento súčin realizovaný iba vtedy ak je splnená logická podmienka typu AND.

Vložte **FB** blok na pracovnú plochu projektu podľa kapitoly 7.1. Po vybraní bloku **Function block** zo zoznamu blokov je otvorené okno **Properties – Function block**. Zvoľte záložku **General – Part 1**. Položku **Name:** nemeňte. V položke **Symbolic name:** možno zadefinovať symbolickú premennú pod ktorou bude príslušný blok zapísaný v symbolickej tabuľke. V položke **Symbol comment:** je možné uviesť komentár k príslušnému bloku, ktorý bude taktiež automaticky zapísaný do symbolickej tabuľky (kapitola 6). V položke **Created in language** je možné zvoliť programovací jazyk v ktorom bude uvedený blok programovaný.
Zmeniť vlastnosti už vytvoreného bloku **FB1** je možné prostredníctvom nasledujúcich operácií. PK na blok 

. Zo zobrazeného menu zvoľte **Object Properties…**. Po tomto kroku je opäť otvorené okno **Properties – Function block**. Takže pre náš prípad môžeme písať:

Name: FB1
Symbolic name: Sucet
Symbol comment: Scitanie dvoch cisel
Created in language: FBD

LK na tlačítko **OK**. Po tomto kroku je zvolený blok umiestnený na pracovnú plochu (pravý panel) vytvoreného projektu.
Ak teraz otvoríme symbolickú tabuľku zistíme, že bola vygenerovaná nová symbolická premenná kde

Symbol Sucet
Address FB1
Data type FB1
Comment Scitanie dvoch cisel

2 x LK na blok **FB1**. Vo otvorenom **LAD/STL/FBD** (obr. 11) editore najskôr zadefinujeme lokálne premenné. Takže budeme potrebovať dve vstupné premenné typu BOOL na realizáciu logickej operácie AND, jednu vstupnú premennú typu INT a jednu statickú premennú typu INT na realizáciu súčinu (môže však byť rovnako použitá aj ďalšia vstupná premenná namiesto statickej). Ďalej budeme potrebovať jednu dočasnú premennú typu BOOL na

aktiváciu súčinu a na záver je potrebná ešte jedna výstupná premenná typu INT na zobrazenie výsledku súčinu. Vyplnená deklaračná tabuľka je znázornená na obr.12.

Address	Declaration	Name	Type	Initial value	Comment
0.0	in	signal_1	BOOL	FALSE	Signal c. 1 pre logicku operaciu
0.1	in	signal_2	BOOL	TRUE	Signal c. 2 pre logicku operaciu
2.0	in	vstup	INT	0	Signal ktory bude nasobeny konstantou
4.0	out	vysledok	INT	0	Vysledok sucinu
	in_out				
6.0	stat	konstanta	INT	325	Konstanta vyuzivana pri nasobeni
0.0	temp	log_vys	BOOL		Vysledok logickej operacie

Obr. 12 Deklaračná tabuľka

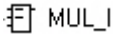
Úlohu rozvrhneme do dvoch sietí. V prevej sieti najskôr prebehne logická operácia AND. Výsledok tejto operácie bude prenesený do druhej siete kde bude realizovaný súčin.

FB1 : Sucin s logickou podmienkou
Comment:
Network 1 : Logicka operacia AND
Comment: Ak je podmienka AND splena potom je mozne realizovat sucin.

Programovanie budeme realizovať prostredníctvom jazyka **FBD**. Zapnite katalóg programových elementov pomocou  . V katalógu otvorte adresár **Bit logic**. Z ponuky vyberte blok  . Pomocou stlačeného LT ho preneste na plochu pod komentárom **Network**-u **1**. Uvedený blok je potom vyobrazený v nasledujúcej podobe

. Nasledujúcim krokom je nahradenie otáznikov za vopred deklarované premenné. LK na prvý otáznik. Do vyznačenej plochy napíšte **signal_1**. LK na druhý otáznik a píšte **signal_2**.
Symbol # znamená že sa jedná o lokálnu premennú ktorá je použiteľná iba v rámci programovaného bloku.
LK na blok operácie AND. Objaví sa zelené orámovanie uvedeného bloku. LK na ikonu  (vrchná lišta **LAD/STL/FBD** editora). Automaticky je k bloku AND pripojený blok =. Nahraďte otáznik za premennú **log_vys**. Týmto je programovanie logickej operácie ukončené. Nasleduje programovanie súčinu.
Najskôr vygenerujeme novú sieť. LK na ikonu  (vrchná lišta **LAD/STL/FBD** editora).

Networ 2: Operacia sucinu
Comment: Nasobenie konstantou 325.

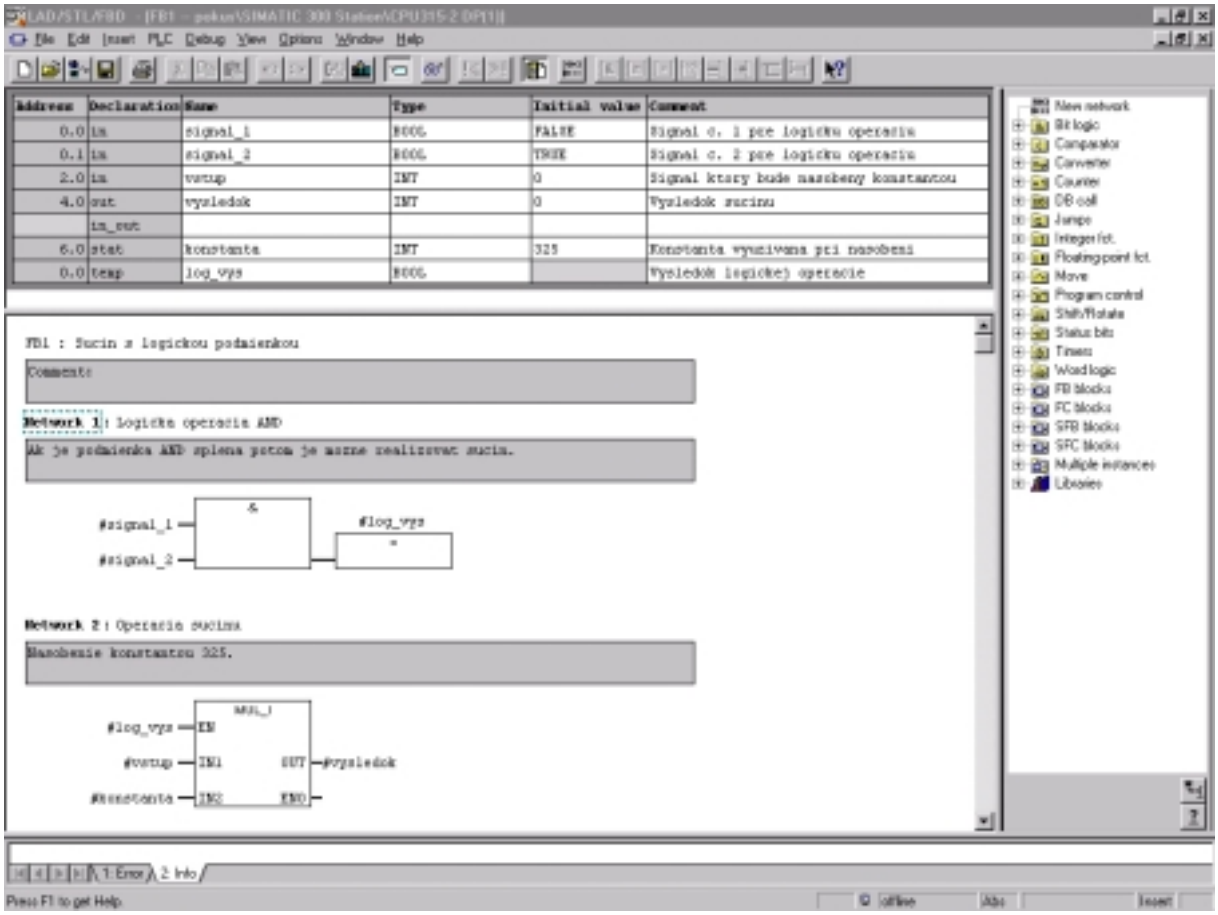
Z adresára **Integer fct.**, katalógu programových elementov, vyberte blok 

a preneste ho na pracovnú plochu pod komentár **Network**-u **2**.

EN – vstupná podmienka
ENO – výstupná podmienka

Tieto premenné sa používajú na definovanie podmienok za ktorých má byť daný blok volaný. Ak nie sú definované blok je vždy volaný.

Ako premennú **EN** definujte deklarovanú premennú **log_vys**. **IN1** je **vstup**. **IN2** je **konstanta** a **OUT** je **vysledok**. **ENO** je neobsadené. Týmto je programovanie bloku **FB1** ukončené. Naprogramovaný blok **FB1** je znázornený na obr. 13.



Obr. 13 Výsledný blok FB1

Nazáver uložte naprogramovaný **FB1** blok. **File -> Save**. Zavrite **LAD/STL/FBD** editor.

7.4 Programovanie DB bloku

V programe Step 7 je možné naprogramovať DB blok asociovaný s FB blokom a prípadový DB blok.

7.4.1 Asociovaný DB blok

Každý FB musí byť asociovaný s DB blokom. V našom prípade je potrebné asociovať **FB1** s DB. Vložte **DB1** blok na pracovnú plochu projektu podľa kapitoly 7.1. Po vybraní bloku **Data block** zo zoznamu blokov je otvorené okno **Properties – Data block**. Zvoľte záložku **General – Part 1**. Položku **Name:** nemeňte. V položke **Created in language** je možné zvoliť programovací jazyk v ktorom bude uvedený blok programovaný. Pre tento typ bloku je možné použiť iba jazyk **DB**.

2 x LK na vytvorený blok DB1. Automaticky je otvorené okno **New data Block**. Z položky **Create** vyberte **Data block referencing a function block**. Z ponuky **Assigment** potom vyberte príslušný FB. V tomto prípade to je **FB1 Sucet**. Po potvrdení je opäť otvorené okno **LAD/STL/FBD** editora v ktorom sa objaví deklaračná tabuľka zhodná s tabuľkou v definovanou v **FB1**.

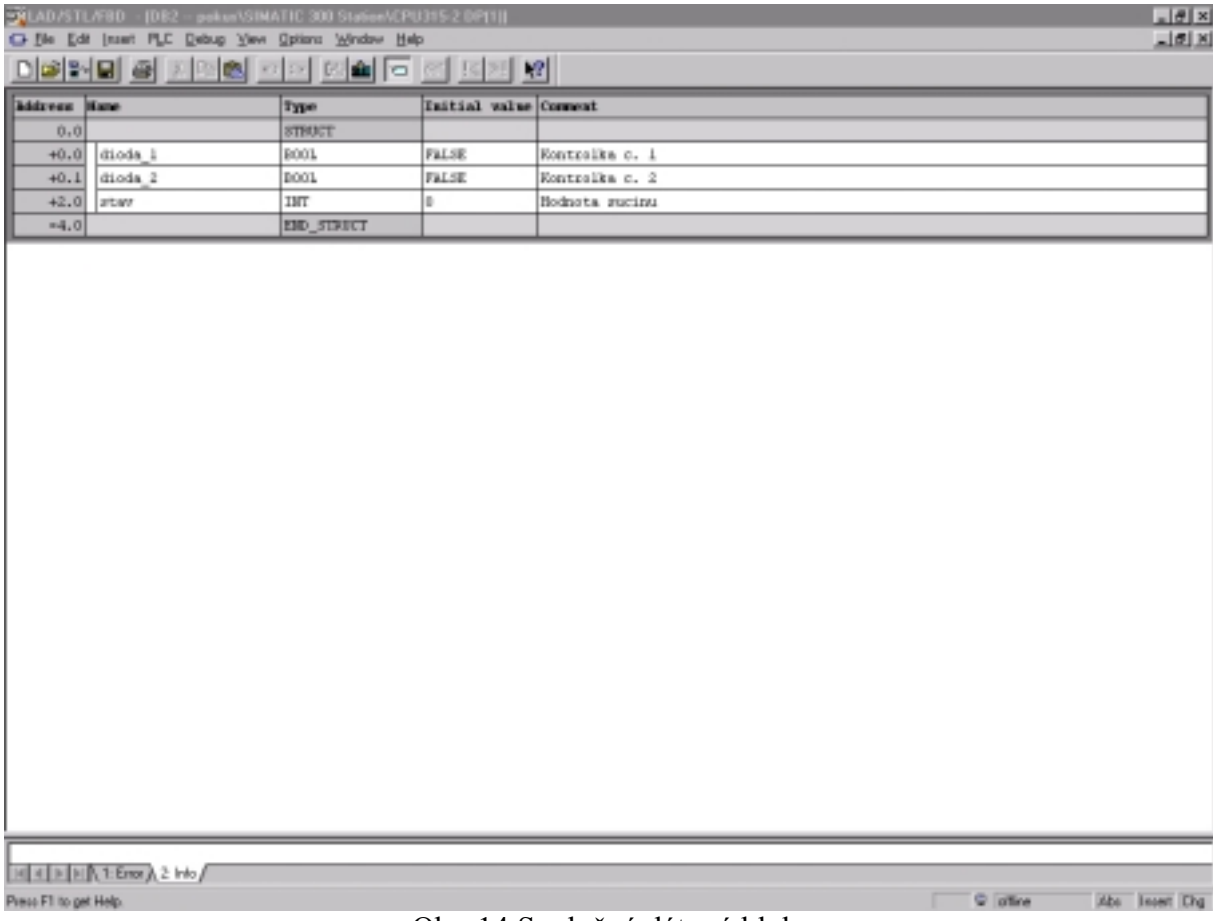
Uložte naprogramovaný **DB1** blok. **File -> Save**. Zavrite **LAD/STL/FBD** editor.

PK na blok DB1 Zo zobrazeného menu zvoľte **Object Properties…**. Po tomto kroku je opäť otvorené okno **Properties – Data block**. V tomto prípade môžete doplniť položky **Symbolic name:** a **Symbol comment:** Napr:

Symbolic name: Sucet_DB
Symbol comment: Datovy blok pre FB1

7.4.2 Spoločný DB blok

V prípade, že nie je dostatok CPU pamäte na nahranie všetkých dát, je možné tieto dáta ukladať v spoločných dátových blokoch. Údaje s týchto DB sú okamžite prístupné a použiteľné pre všetky FB. Naopak DB asociovaný s príslušným FB poskytuje údaje iba pre daný FB. Pri vytváraní spoločného DB postupujeme rovnako ako v kapitole 7.4.1, ale v okne **New data Block** vyberieme v položke **Create** možnosť **Data block**.V zobrazenej tabuľke potom môžeme editovať premenné ktoré budú použiteľné pre všetky FB bloky. V našom prípade si zadefinujeme dve premenné typu BOOL a jednu premennú typu INT. Výsledný spoločný dátový blok je znázornený na obr. 14.



Obr. 14 Spoločný dátový blok.

Editovanie tejto tabuľky je podobné ako editovanie deklaračnej tabuľky pri programovaní FB. Rozdiel je iba vtom, ze tu nie je k dispozícii stĺpec **Declaration**. Uložte naprogramovaný **DB2** blok. **File -> Save**. Zavrite **LAD/STL/FBD** editor.
Podobne ako pri vytváraní asociovaného DB zadefinujte položky **Symbolic name:** a **Symbol comment:** Napr:

Symbolic name: Spol_DB
Symbol comment: Spolocne data

7.5 Programovanie FC bloku

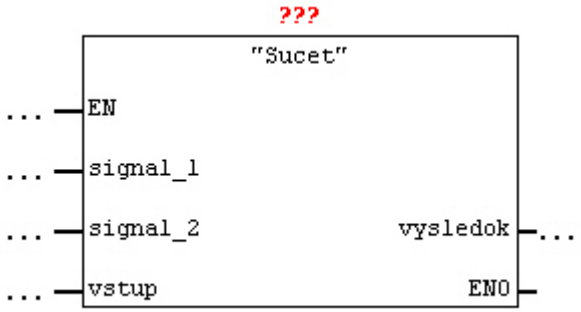
Programovanie funkcií, FC blokov, je realizované podobne ako programovanie FB blokov. Rozdiel je vtom, že v deklaračnej tabuľke nie je možne zadefinovať statické premenné. Taktiež po vytvorení FC bloku nie je potrebné definovať asociovaný DB pre naprogramovaný FC blok.

7.6 Programovanie OB blokov

Ak chceme aby naprogramovaný FB, FC, resp. DB bol skutočne využívaný počas riadenia reálneho procesu, je nevyhnutné aby bol „volaný“ prostredníctvom príslušného organizačného bloku OB. Hlavný cyklus, ktorý je periodicky opakovaný sa nachádza v bloku **OB1**.
PK na blok **OB1** aktivuje menu z ktorého vyberte **Object Properties…**. Podobne ako v predchádzajúcich prípadoch tak aj v prípade bloku **OB1** je otvorené okno **Properies – Organization block**. Opäť je v tomto okne možné definovať symbolickú premennú a príslušný komentár. Napr.

Symbolic name: Hlavny cyklus
Symbol comment: Hlavny program riadenia

2 x LK na blok **OB1** aktivuje už známy **LAD/STL/FBD** editor. Aj v tomto prípade je možné deklarovať vnútorné premenné bloku v deklaračnej tabuľke podobne ako tomu bolo v predchádzajúcich prípadoch. Rozdiel je iba v tom, že je možné použiť iba dočasné premenné. V našom prípade nie je potrebné definovať vnútorné premenné, ale je treba do otvoreného **OB1** bloku začleniť vopred pripravený blok **FB1**.
Keďže sa jedná o jednoduchú schému nebudeme uvádzať názov bloku ani siete resp. ich komentáre.
Zapnite katalóg programových elementov a z adresára **FB blocks** preneste pomocou stlačeného LT **FB1 Sucet** na plochu pod komentárom **Network**-u **1**. Uvedený blok je potom

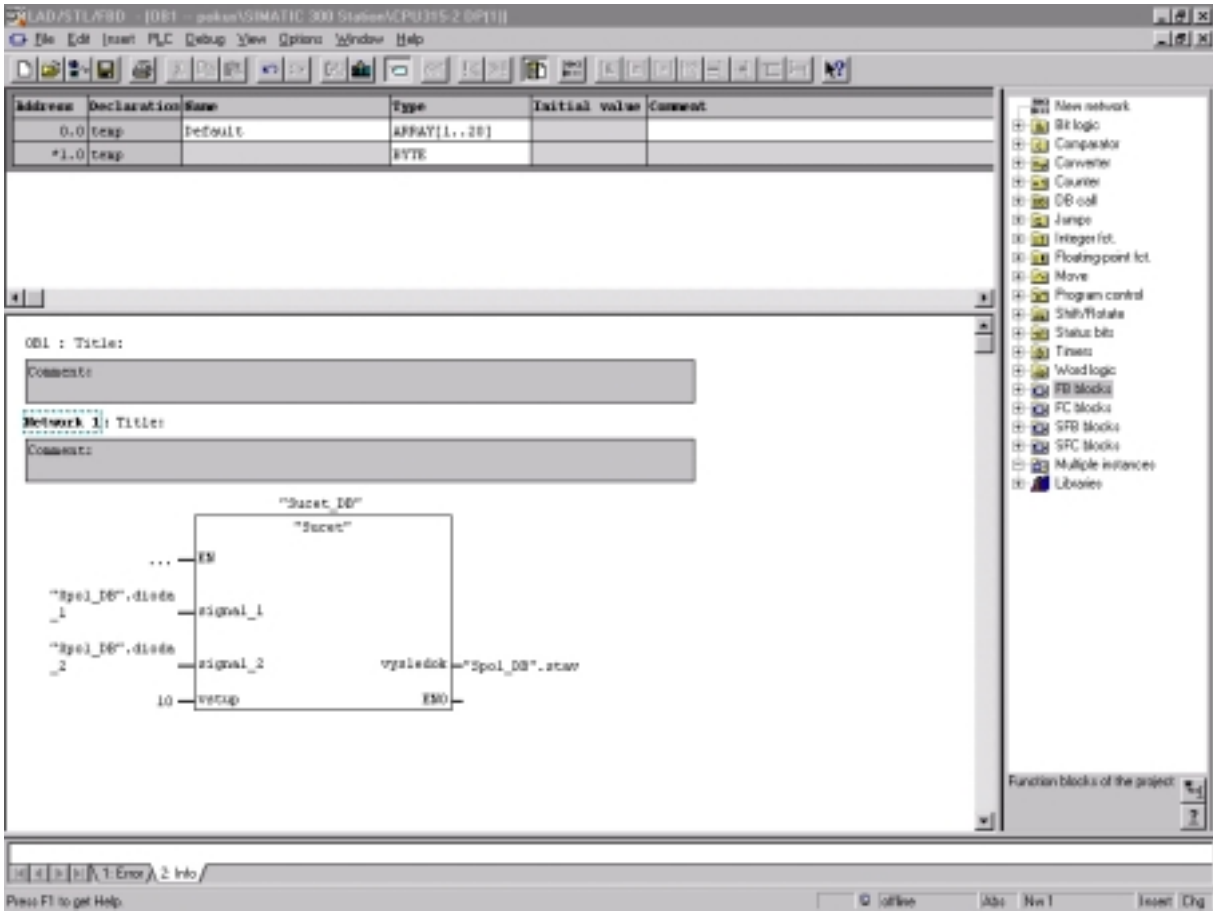


vyobrazený v nasledujúcej podobe . Vo vnútri bloku sú uvedené názvy vstupných a výstupných vnútorných premenných definovaných v bloku **FB1**. Namiesto **…** je potrebné priradiť vstupné resp výstupné signály. **???** je treba nahradiť symbolickou premennou DB bloku asociovaného s blokom **FB1**.
PK na **???**. Z ponúknutého menu vyberte **Insert Symbol**. Z otvorenej ponuky symbolických premenných vyberte **Sucet_DB**. Je to symbolická premenná bloku **DB1**, ktorý je asociovaný s blokom **FB1**.
K premenným **signal_1** a **signal_2**, ktoré sú typu BOOL, priradíme premenné definované v spoločnom dátovom bloku **DB2**.
LK na **…** (**signal_1**). Do prístupnej oblasti definujeme spomenutú premennú v tvare **“Spol_DB”.dioda_1**. Pričom výraz v úvodzovkách reprezentuje názov spoločného dátového

bloku **DB2** a výraz za bodkou je meno príslušnej premennej definovanej v **DB2**. Podobne zadefinujte aj premennú pre **signal_2**.

Poznámka Všetky premenné, ktoré sú uvedené v úvodzovkách reprezentujú **globálne** premenné a sú teda prístupné pre všetky bloky používané v danom projekte.

K poslednej vstupnej premennej (**vstup**) priraďte číslo typu INT napr. **10**. Hodnota premennej **vysledok** bude exportovaná opäť do spoločného dátového bloku **DB2**. Konkrétne do premennej **stav**. Týmto je programovanie bloku **OB1** ukončené. Výsledný organizačný blok je znázornený na obr. 15.



Obr. 15 Organizačný blok

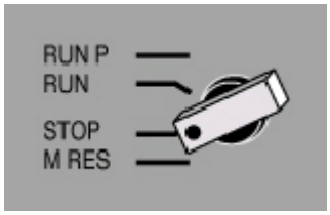
Uložte naprogramovaný **OB1** blok. **File -> Save**. Zavrite **LAD/STL/FBD** editor.

8. Kopírovanie programu do pamäte procesora

Skôr ne ž môžeme kopírovať vytvorený projekt do pamäte CPU je potrebné vytvoriť on-line spojenie medzi pracovnou stanicou SIMATIC 300 a PC na ktorom bol vytvorený spomenutý projekt. Predpokladajme, že PC a pracovná stanica sú navzájom prepojené prostredníctvom MPI kábla.

8.1 Vytvorenie on-line spojenia

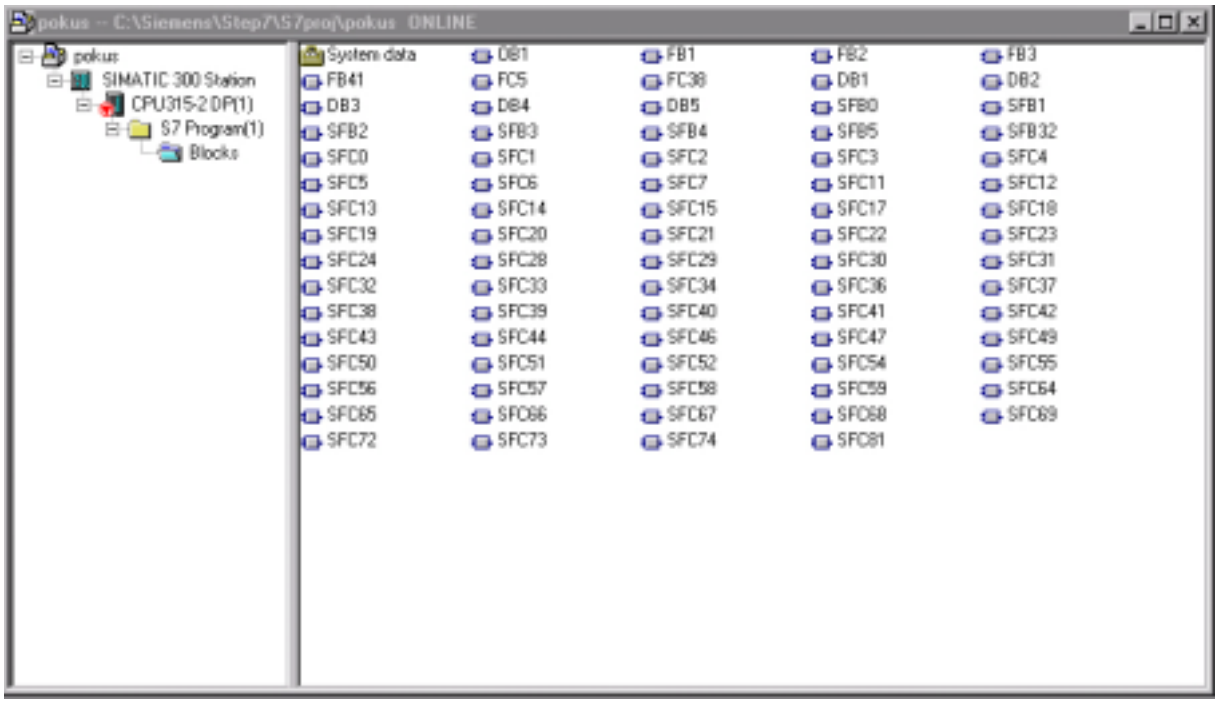
Zapnite zdroj pracovnej stanice. Poloha **ON**.



Otočte kľúč **CPU315-2DP** do polohy **STOP**.

Ak je všetko v poriadku tak svieti **zelená LED** zapnutého zdroja, ďalej **zelená LED** (DC5V) a **oranžová LED** (STOP) na module **CPU315-2DP**.

Otvorte projekt, ktorý chcete kopírovať.
V programe Step 7 LK na menu (horná lišta) **View -> Online**. Otvorí sa okno on-line spojenia (obr. 16).



Obr. 16 Okno on-line spojenia

8.2 Kopírovanie systémových dát a vytvorených blokov do pamäte CPU
V otvorenom okne (on-line spojenie) označte všetky bloky (OB, FB, FC, DB) ktoré chcete odstrániť. Stlačením klávesy **Delete** ich vymažte.

Poznámka Bloky SFB, SFC nie je možné z tohto okna odstrániť.

LK na okno užívateľom vytvoreného projektu (okno off-line spojenia). Označte všetky bloky, ktoré chcete prekopírovať. LK na menu (horná lišta) **PLC-> Download**. Na otázku **Do you want to load the systém data?** Zvoľte **Yes**. Na otázku **Do you want to delete the system data on the module [0/2/0] CPU315-2DP(1) completely and replace them with offline data?** Odpovedzte **Yes**. Tento krok umožní prekopírovať aj systémové dáta, ktoré obsahujú informácie o konfigurácii siete a modulov. To znamená, že nie je potrebné tieto informácie kopírovať osobitne.
Po tejto operácii sa objavia všetky bloky z off-line okna v on-line okne. Kopírovanie je ukončené.

8.3 Kopírovanie RAM do ROM
Všetky bloky a systémové dáta sú najskôr prekopírované do **RAM** pamäte. Po vypnutí zdroja a jeho opätovnom zapnutí sú tieto údaje z **RAM** pamäte CPU vymazané a treba ich znovu kopírovať. Ak chceme aby boli údaje prekopírované trvalo je potrebné prekopírovať z **RAM** do **ROM** pamäte.
LK na menu (horná lišta) **PLC-> Copy RAM to ROM…**. Po opätovnom zapnutí zdroja je obsah **ROM** automaticky kopírovaný do **RAM**.

8.4 Resetovanie CPU

RAM pamäť je možné resetovať. Po resetovaní je automaticky prekopírovaná **ROM** do **RAM**.

Resetovanie je možné realizovať pomocou nasledujúcich krokov:

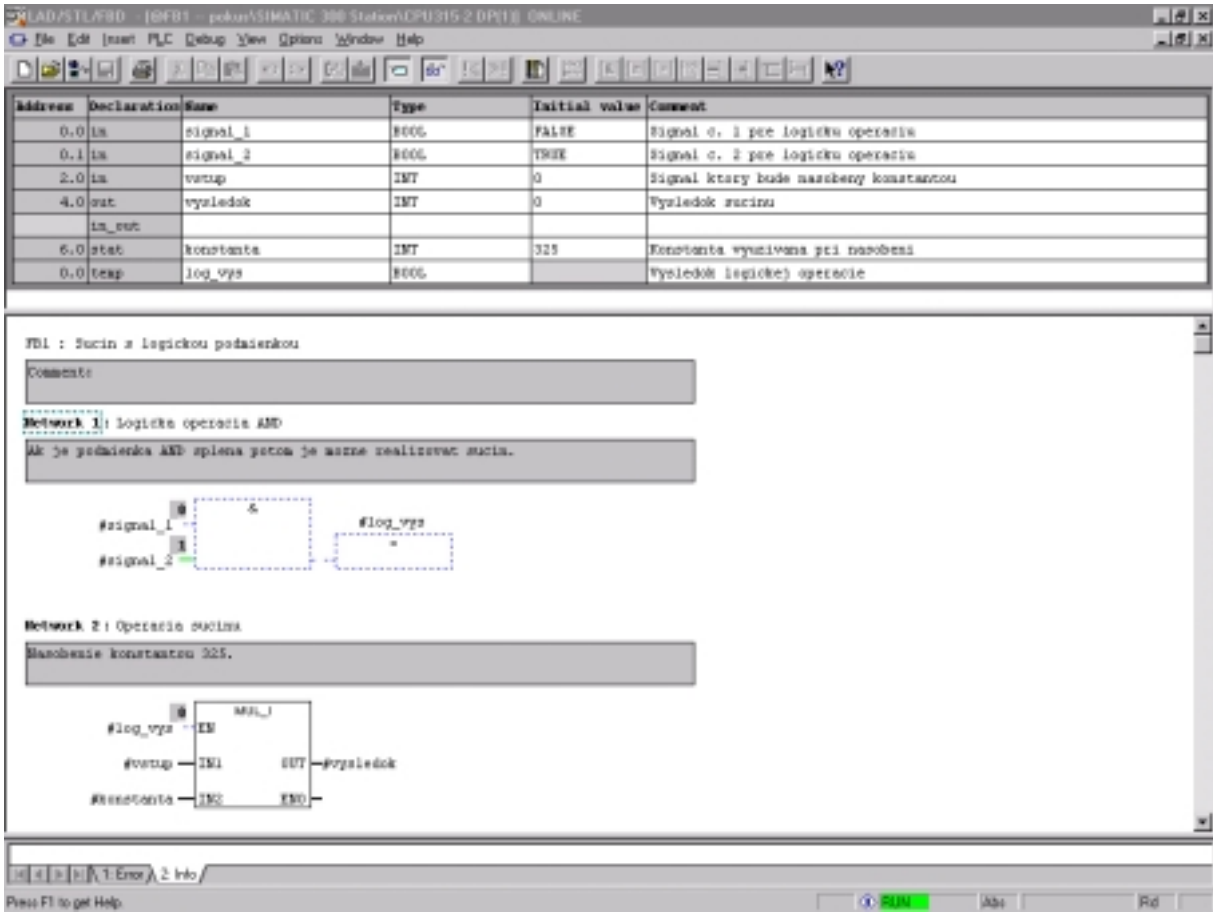
Otočte kľúč **CPU315-2DP** do pozície **MRES** a držte ho v tejto pozícii minimálne 3 s.

Pustite kľúč a max. do 3 s ho otočte opäť do pozície **MRES**.

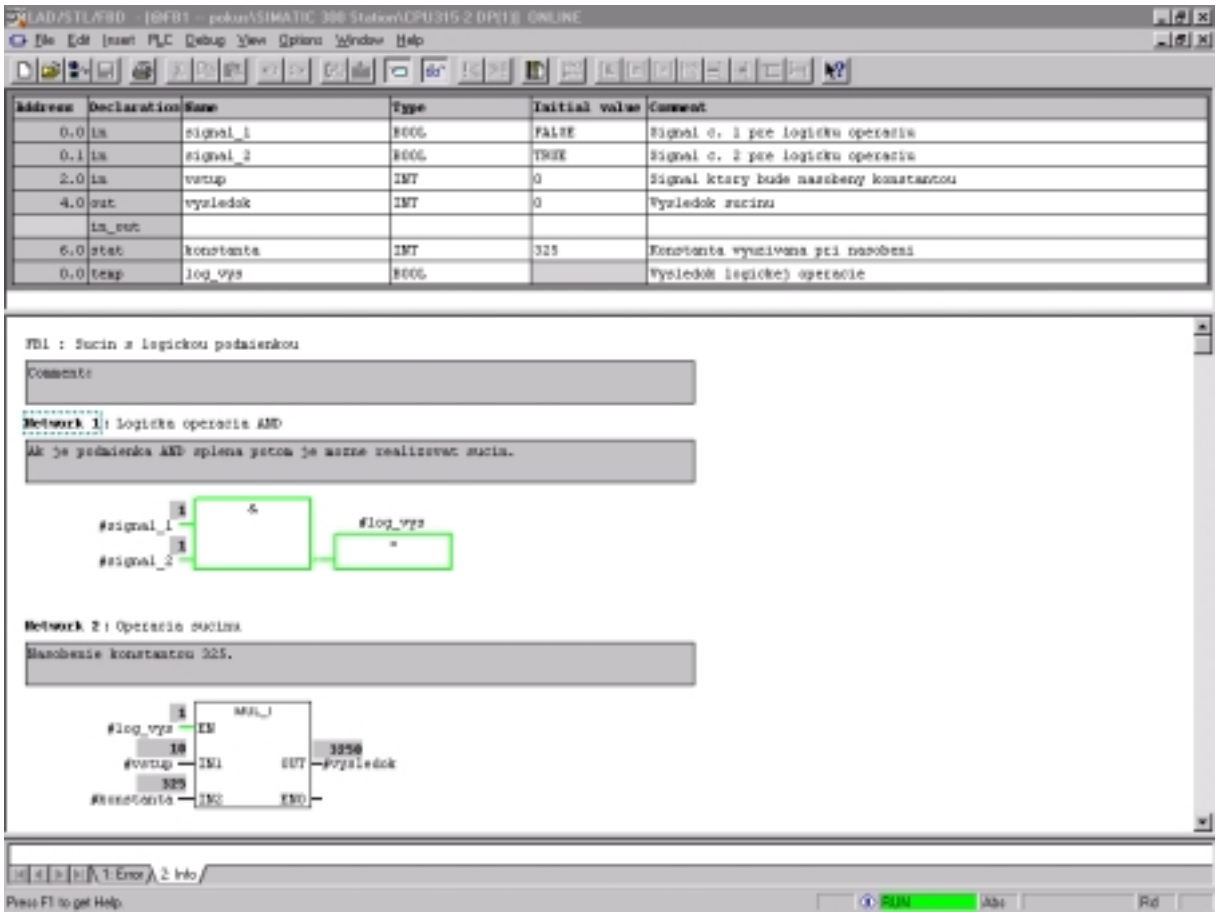
9. Testovanie logických obvodov

Pred použitím vytvorených logických a regulačných obvodov na reálny proces je možné tieto obvody testovať. Je nevyhnutné aby na tento účel bolo vytvorené on-line spojenie a daný projekt bol nakopírovaný do **Ram** pamäte CPU. Otočte kľúč **CPU315-2DP** do polohy **RUN** alebo **RUN-P**. Otvorte ľubovoľný blok (OB, FB, FC, DB) v okne on-line spojenia.

Napr. LK na blok **FB1** (okno on-line spojenia). LK na ikonu  . Pomocou tejto ikony sa aktivuje monitorovanie všetkých logických a riadiacich obvodov nachádzajúcich sa v príslušnom bloku. Hodnoty jednotlivých vstupných resp. výstupných signálov sú vyznačené v sivom rámiku priradenom ku každej premennej. (toto platí pre **FBD** programovací jazyk). Ak logická operácia neprebehla tak sú jednotlivé bloky patriace danej logickej operácii označené modrou prerušovanou čiarou (obr. 17). Ak daná logická operácia prebehla tak dané bloky sú označené zelenou plnou čiarou (obr. 18).



Obr. 17 Príklad neprebehnutej logickej operácie



Obr. 18 Príklad prebehnutej logickej operácie

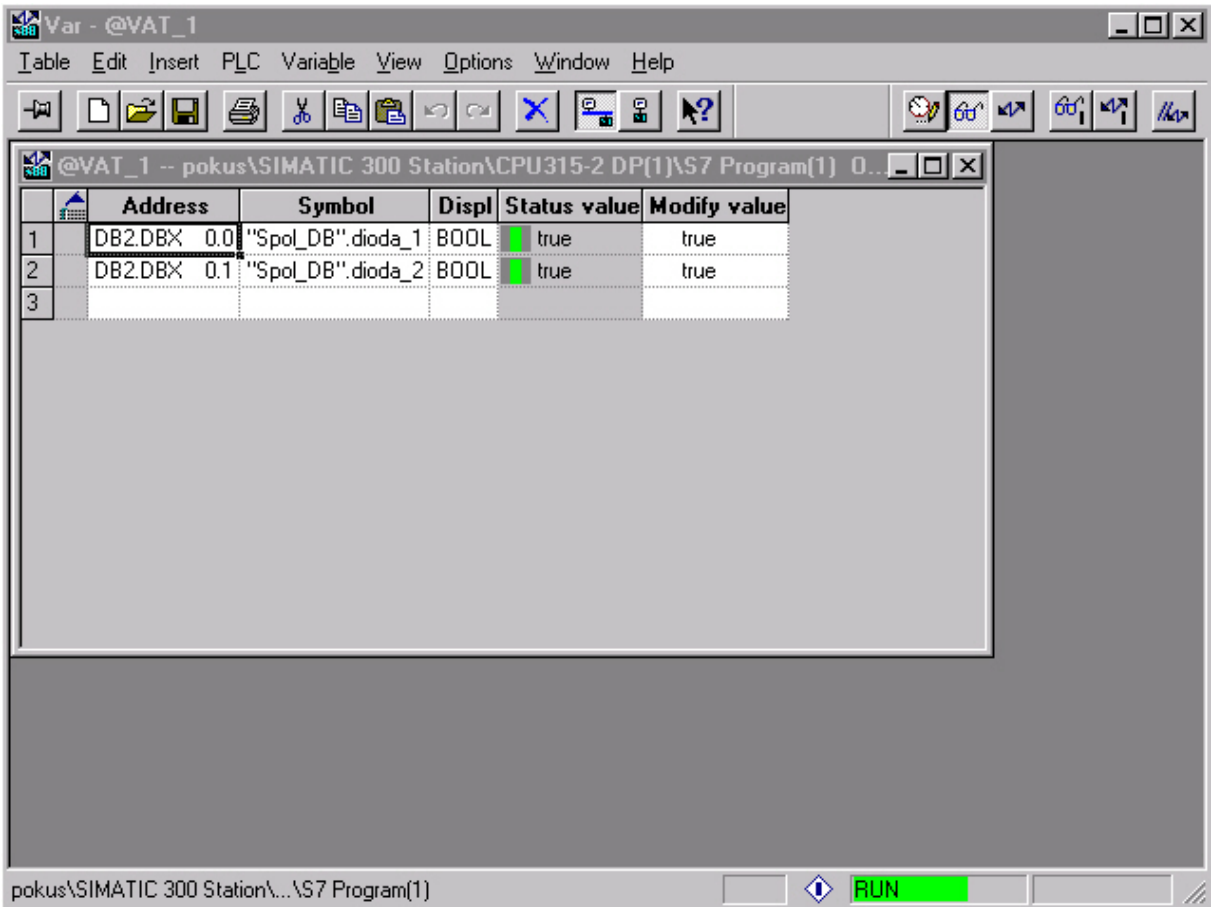
9.1 Testovanie programu pomocou tabuľky premenných

Umožňuje testovať premenné pomocou monitorovania a zmeny ich hodnôt. Je nevyhnutné aby na tento účel bolo vytvorené on-line spojenie a daný projekt bol nakopírovaný do Ram pamäte CPU. Najskôr vytvoríme tabuľku premenných, ktorá bude obsahovať všetky premenné, ktoré budeme chcieť meniť za chodu CPU.

PK na ploche vytvoreného projektu (okno off-line spojenia). Z ponúknutého menu vyberte Insert new object -> Variable table. Po otvorení okna Properties –Variable table vyplňte jednotlivé položky podobne jako při vytváraní napr FB bloku. 2 x LK na blok VAT_1. Vložte premenné, ktoré chcete počas chodu CPU meniť. V našom prípade to budú premenné dioda_1 a dioda_2, ktoré sa nachádzajú v spoločnom dátovom bloku. Preto do stĺpca Symbol napíšte "Spol_DB".dioda_1. Po potvrdení (ENTER) je vygenerovaný nový riadok, do ktorého napíšte "Spol_DB".dioda_2.

Otočte kľúč CPU315-2DP do polohy RUN-P. Poloha RUN-P povoľuje zmenu jednotlivých premenných za chodu procesora.

LK na ikonu (horná lišta v okne Var –VAT_1). Aktivujte monitorovanie pomocou ikony . Do stĺpca Modify value napíšte modifikovanú hodnotu zvolenej premennej a LK na ikonu . Po tomto úkone je hodnota danej premennej zmenená aj v RAM pamäti CPU. Pre veľký počet premenných je vhodné vytvoriť viac tabuliek. Príklad tabuľky premenných uvedený na obr. 19.

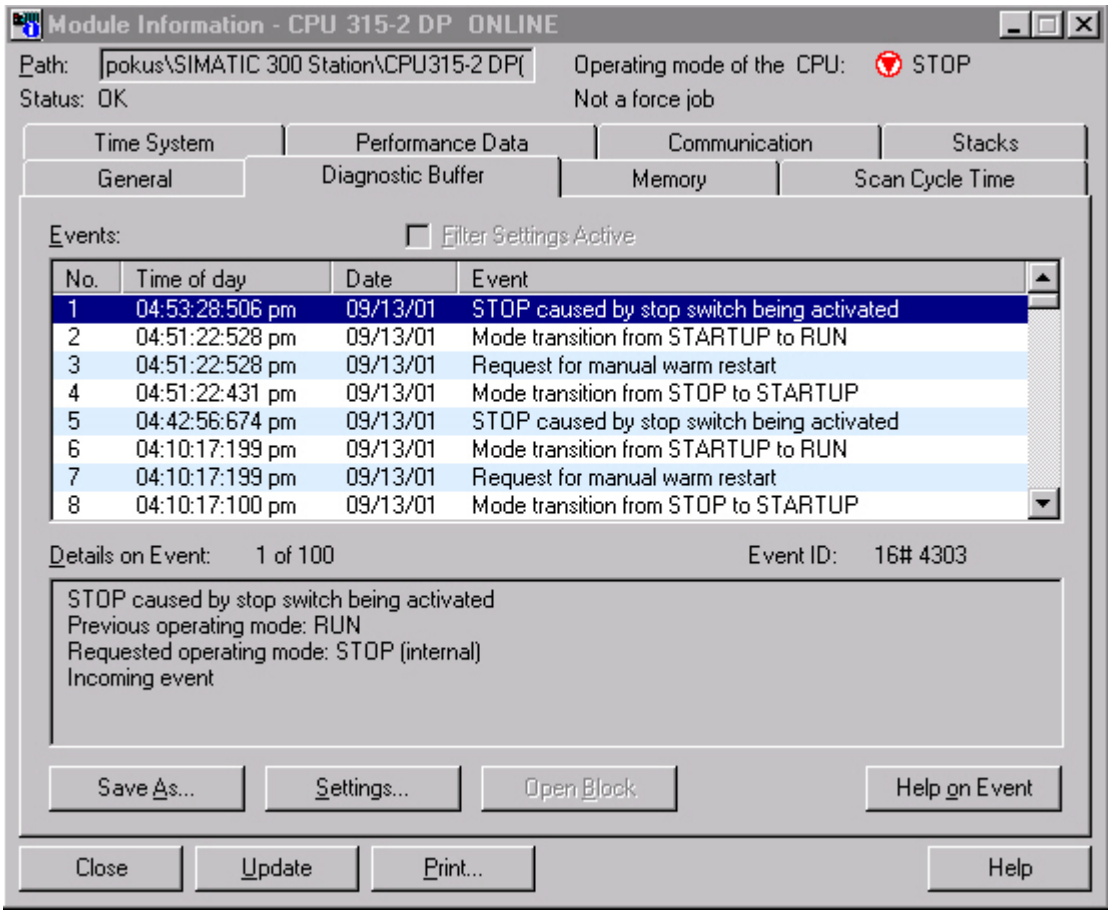


Obr. 19 Monitorovanie a modifikácia pomocou tabuľky premenných

Poznámka Premenné typu PQW nemôžu byť monitorované a premenné typu PIW nemôžu byť modifikované.

10. Diagnostika CPU

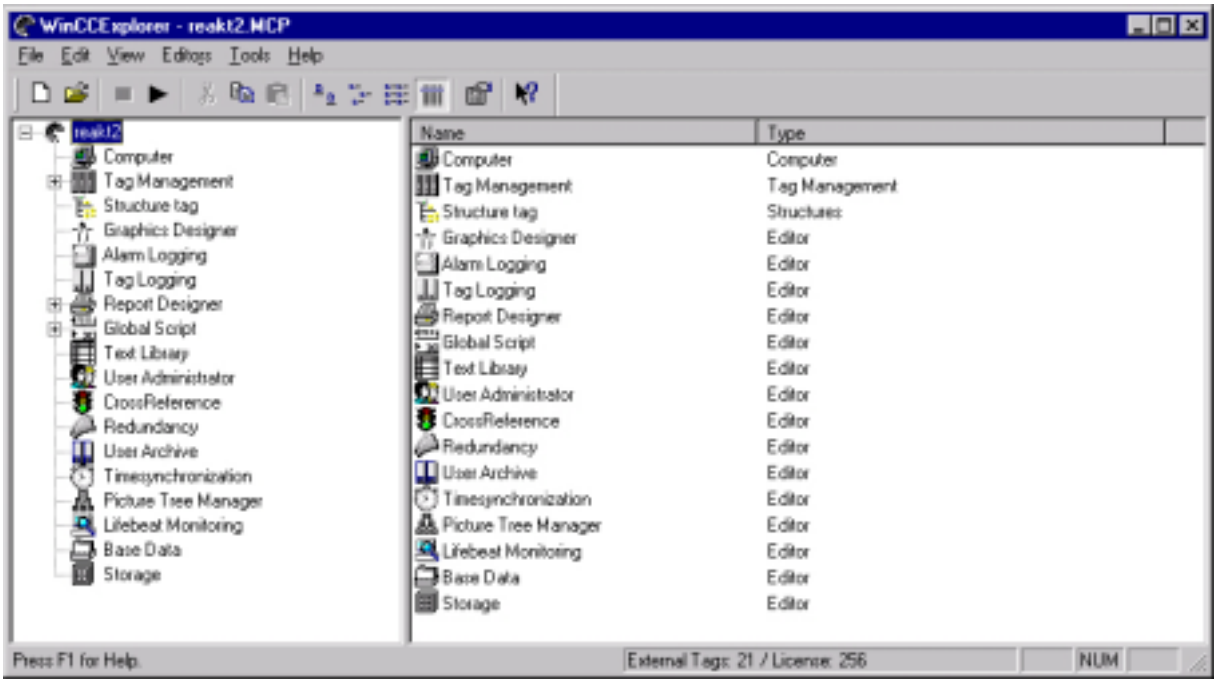
Ak v extrémnych prípadoch prejde CPU do STOP módu, alebo sa po nahraní programu nepodarí prejsť zo STOP do RUN alebo RUN-P (svieti oranžová LED (STOP) na module CPU315-2DP) znamená to, že je v systéme chyba, ktorú treba diagnostikovať pomocou diagnostického okna. Je nevyhnutné aby na tento účel bolo vytvorené on-line spojenie a daný projekt bol nakopírovaný do RAM pamäte CPU. Otočte kľúč CPU315-2DP do polohy STOP. PK a z ponúknutého menu vyberte PLC -> Module Information…. Následne je otvorené okno Module Information – CPU 315-2 DP ONLINE. Zvoľte záložku Diagnostic Buffer. Ak je chyba spôsobená chybným naprogramovaním niektorého bloku (OB, FB, FC) je možné tento blok otvoriť pomocou tlačítka Open Block. Daný blok bude otvorený a chybná sieť bude vysvietená. Diagnostické okno je znázornené na obr. 20.



Obr. 20 Diagnostické okno

11. Vizualizačný systém WinCC

Program *WinCC* sa spustí z menu *Štart – Simatic – WinCC – Windows Control Center 5.0*. Otvorí sa obrazovka ako na obr. 1.

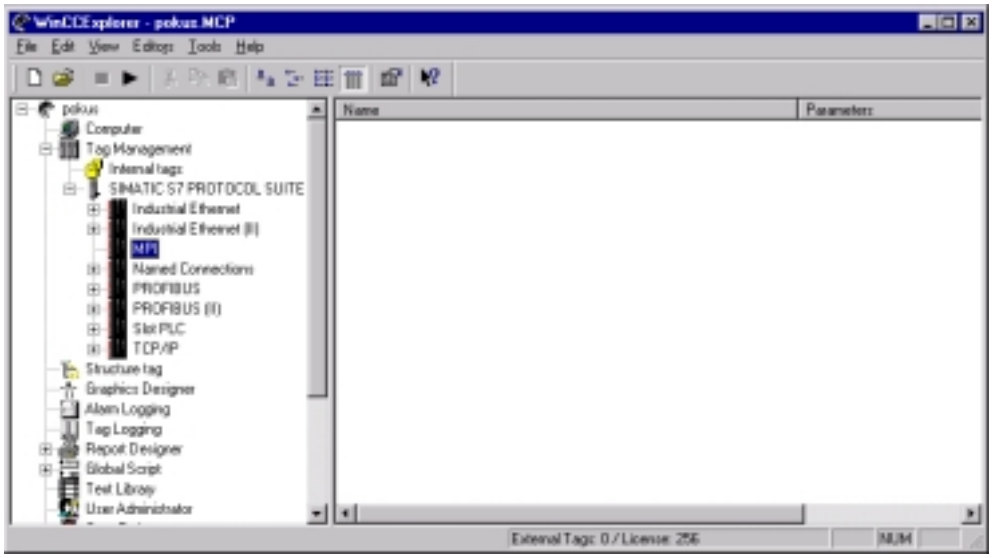


Obr. 1: WinCC s otvoreným neaktívnym projektom

Okno sa skladá z troch hlavných častí:
1. Menu a lišta nástrojov
2. Stromová štruktúra dostupných modulov
3. Vlastnosti jednotlivých modulov

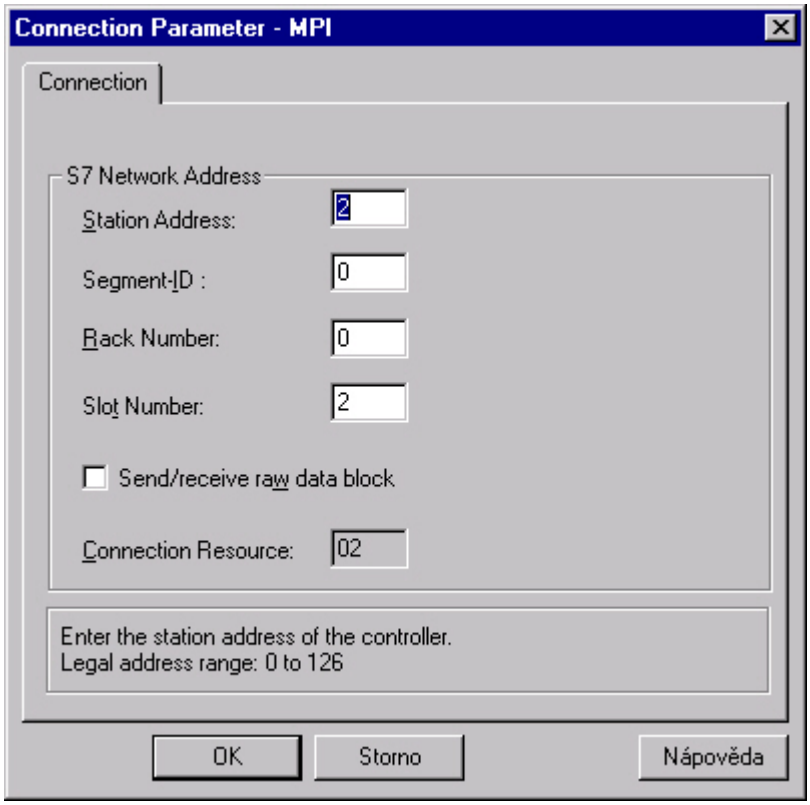
Vytvorenie nového projektu

Z ponuky *File – New* sa vyberie *Single-User Project* a do kolonky *Project Name* sa vloží názov nového projektu. Ako prvý úkon sa musí nadefinovať rozhranie, pomocou ktorého bude WinCC komunikovať s PLC. PT na *Tag Management* sa zvolí položka *Add New Driver* a z následnej ponuky sa zvolí súbor *Simatic S7 Protocol Suite.CHN*. V ponuke *Tag Management* sa objaví nový podstrom *SIMATIC S7 PROTOCOL SUITE*. (obr. 2).



Obr. 2: Definovanie nového pripojenia

Z tohto stromu sa PK na položke *MPI* vyberie položka *New Driver Connection*. V novootvorenom okne *Connection properties* sa do položky *Name* vloží názov pripojenia (napr. Simatic) a klikne sa na tlačidlo *Properties*. Tu sa nastavia parametre tak ako je na obr. 3.



Obr. 3: Nastavenie parametrov MPI pripojenia

Teraz je WinCC pripravený na definovanie tagov.

TAGy a práca s nimi

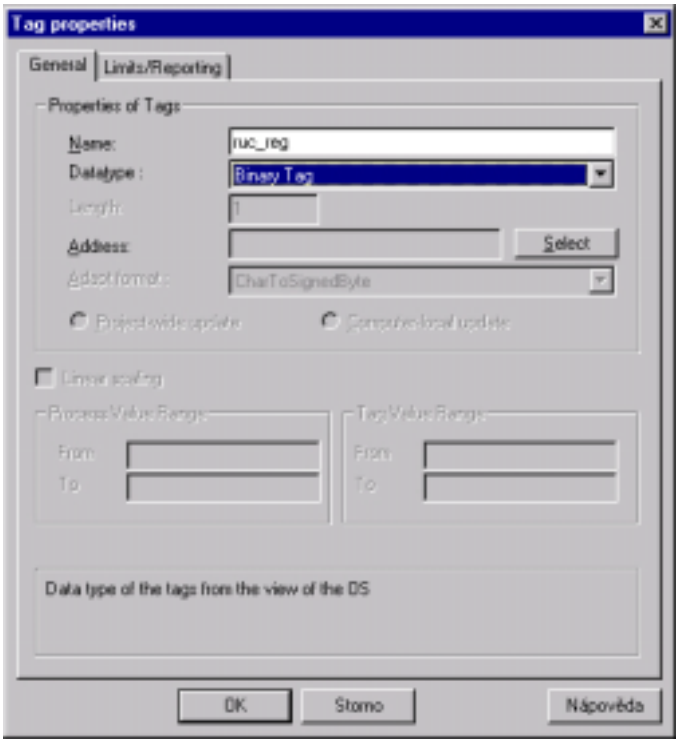
Tag je virtuálny dátový kanál, cez ktorý prechádzajú dáta. Jeden „koniec“ tagu je pripojený na určitú pamäťovú adresu (tá slúži ako zásobník dát) a druhý koniec tagu tieto dáta sprístupňuje užívateľovi. Tagy môžu byť dvojaké:
interné – slúžia na uchovávanie interných premenných (napr. viditeľnosť okna a pod.)
externé – slúžia na komunikáciu s PLC (teplota v reaktore, vznik alarmu a pod.)

Externé tagy

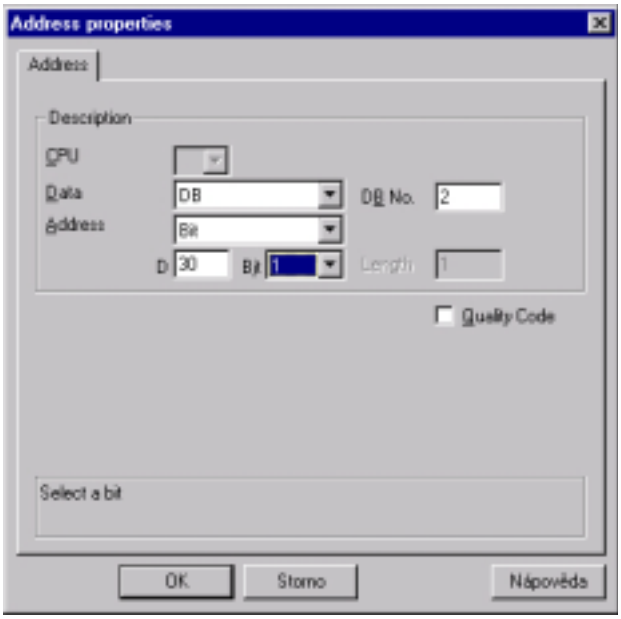
Externý tag sa definuje PK na okno vlastnotí modulu Tag Management – SIMATIC S7 – MPI – simatic. Objaví sa okno ako na obr. 4. Tu je potrebné do kolonky *Name* vložiť názov tagu (ruc_reg) a zvoliť jeho bitovú veľkosť. Keďže identifikátor zapnutia ručného riadenia nadobúda iba dve hodnoty (0 / 1), stačí zvoliť *Datatype – Binary Tag*. Ak by tag udával napr. teplotu, bolo by potrebné zvoliť iný typ, napr. 32-bitový *Floating Point*. Následne je potrebné kliknúť na tlačidlo *Select*. V otvorenom okne sa zvolí pamäťová adresa, z ktorej daný tag bude čerpať informácie (obr. 5). V kolonke *DB No* je potrebné vložiť číslo dátového bloku, kde je daná informácia uložená; Do kolonky *D* sa vloží bytová časť adresy a do *Bit* sa vloží bitová časť adresy. O tom, na akých pamäťových miestach sa dané premenné nachádzajú, je možné sa dozvedieť pomocou Simatic Managera a otvorení príslušného dátového bloku (obr. 6). V stĺpci *Adress* sa nachádzajú adresy premenných prislúchajúcich k príslušným riadkom stĺpca *Name*. Adresa je v tvare
[Bytová časť].[Bitová časť]

Interné tagy

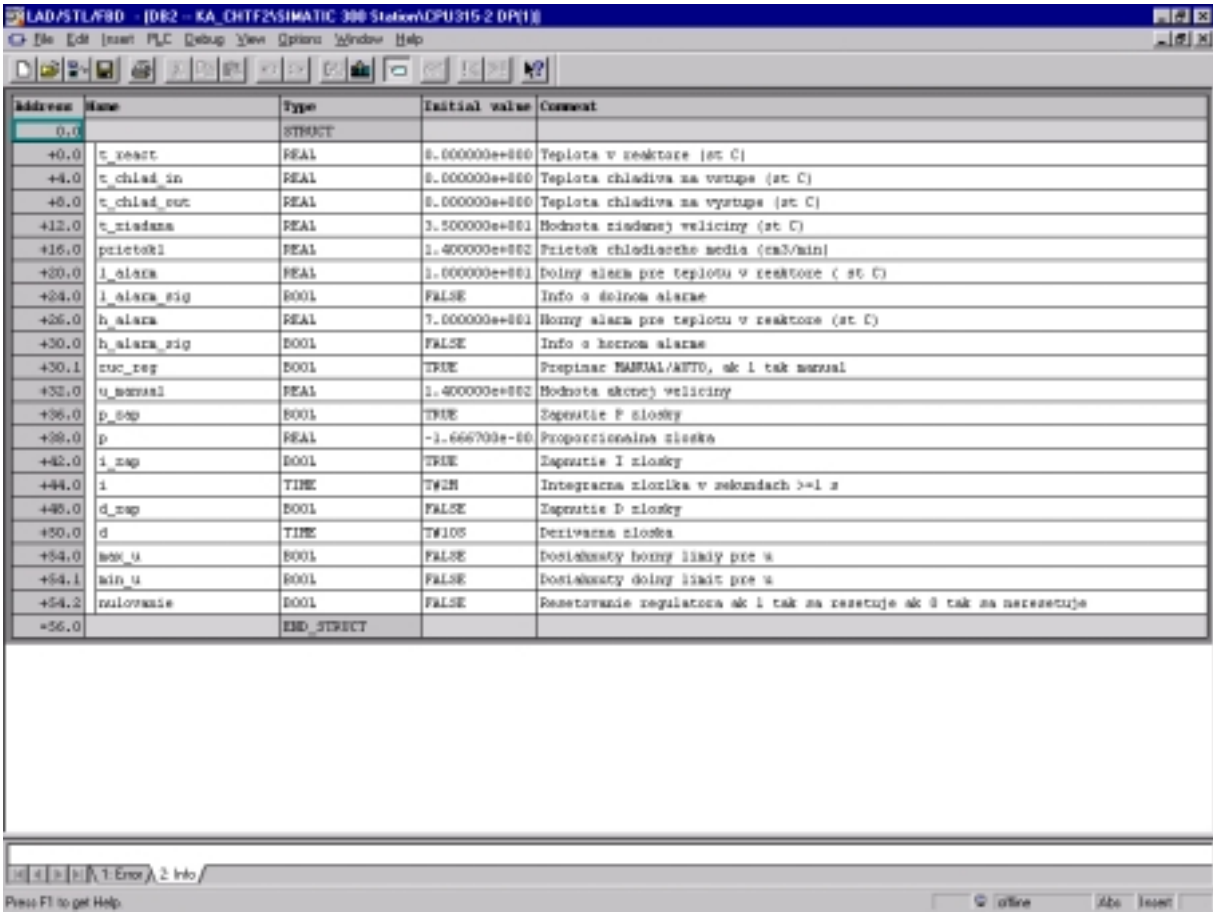
Definícia interných tagov prebieha zhodným postupom, ale vytvárajú sa v časti *Internal Tags*.



Obr. 4: Definícia nového tagu – krok č. 1



Obr. 5: Definícia nového tagu – krok č. 2



Obr. 6: Okno Simatic Manager s otvoreným dátovým blokom

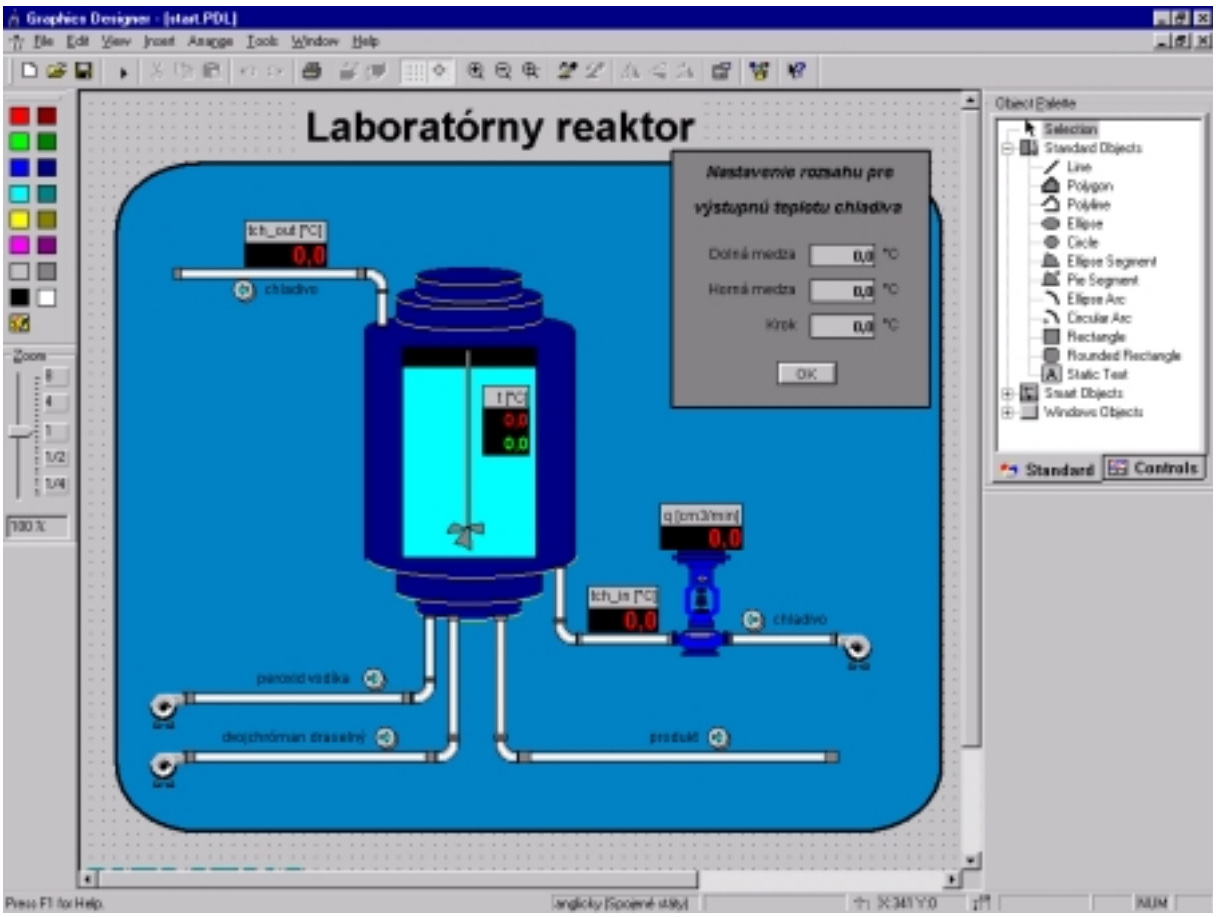
Graphics Designer

Vizuálne rozhranie sa vytvára v module *Graphics Designer*. Spustenie tohto modulu sa môže vykonať dvojako:
1. Otvorením už existujúceho obrázka – 2 x ĽK na názov súboru
2. Otvorením novej schémy – PK na plochu a zvolením *New Picture*

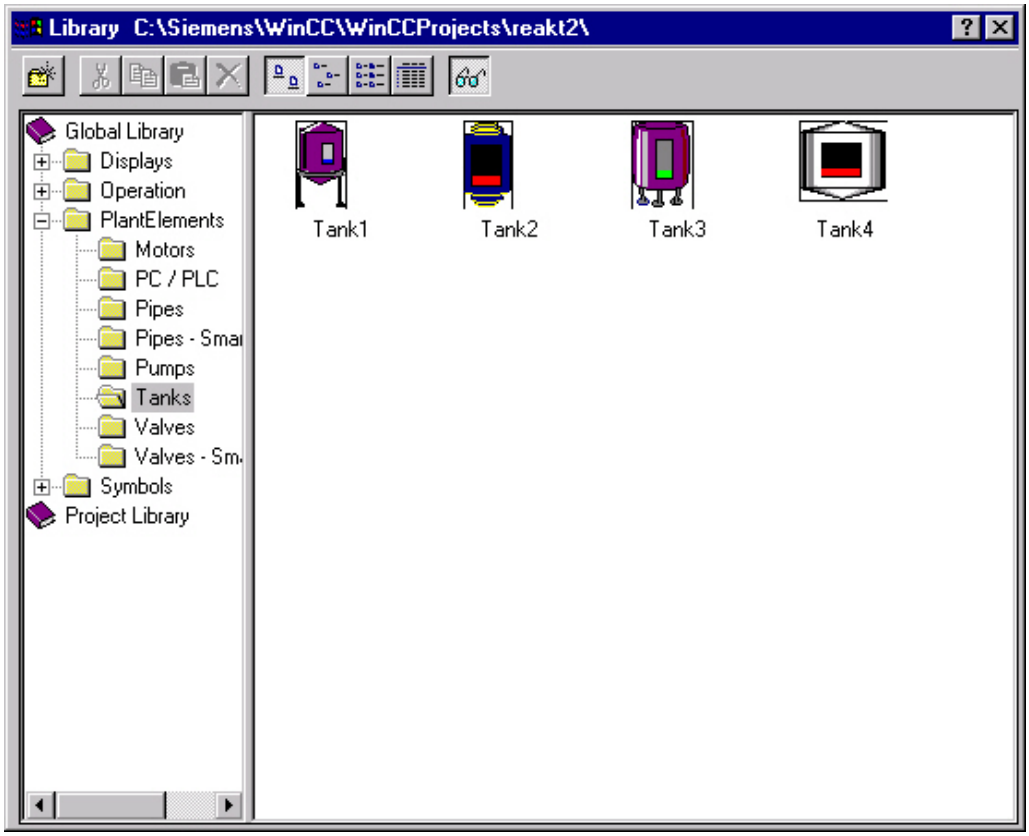
Okno grafického editora je na obr. 7. Skladá sa z troch hlavných častí:
1. Menu a lišta nástrojov
2. Vlastná kresliaca plocha
3. Okno s paletami nástrojov

Graphics Designer je vektorový editor, to znamená že každý objekt môžeme ľubovoľne zväčšovať, zmenšovať a presúvať. Objekty sa delia na dve hlavné skupiny:
1. Statické objekty
2. Dynamické objekty

Základné statické objekty sa nachádzajú na palete nástrojov v strome *Standard Objects*. Vyspelejšie grafické objekty (zásobníky, potrubia, ventily, atď) sa nachádzajú v tzv. knižnici. Tá sa dá otvoriť pomocou menu *View – Library*. Otvorí sa hierarchicky členená ponuka (obr. 8), z ktorej je možné objekty presunúť na hlavnú kresliacu plochu. Statické objekty tvoria "pozadie" každej schémy. Ide hlavne o texty, aparatúry a pod.



Obr. 7: Okno Graphics Designer



Obr. 8: Knižnica vytvorených objektov

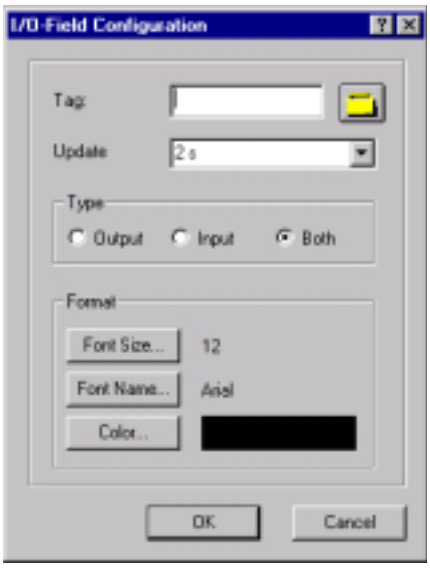
Dynamické objekty sa vyznačujú tým, že môžu ovplyvňovať a / alebo byť ovplyvňované vznikajúcimi udalosťami. Udalosťou sa chápe akákoľvek zmena, či už zmena hodnoty tagu (napr. zmena teploty v reaktore) alebo zásah užívateľa (kliknutie myšou a pod.).

Objekty ovplyvňované udalosťami:

Sú umiestnené na palete objektov v strome *Smart Objects*. Najjednoduhším príkladom je *I/O Field*. Ide o textové pole, ktorého hodnota je závislá na hodnote nejakého tagu. Vstupno-výstupné pole môže byť:

1. Vstupné (*Input*) – možeme do neho písať a tým vlastne zapíšeme vloženú hodnotu do daného tagu (napr. žiadaná teplota pre reaktor)
2. Výstupné (*Output*)– ich hodnota sa mení na základe aktuálnej hodnoty tagu (napr. aktuálna teplota v reaktore)
3. Kombinované (*Both*)

Hneď po premiestnení *I/O Field* na pracovnú plochu, alebo PK + *Configuration Dialog*, sa otvorí okno ako na obr. 9.



Obr. 9: Konfigurácia I/O Field

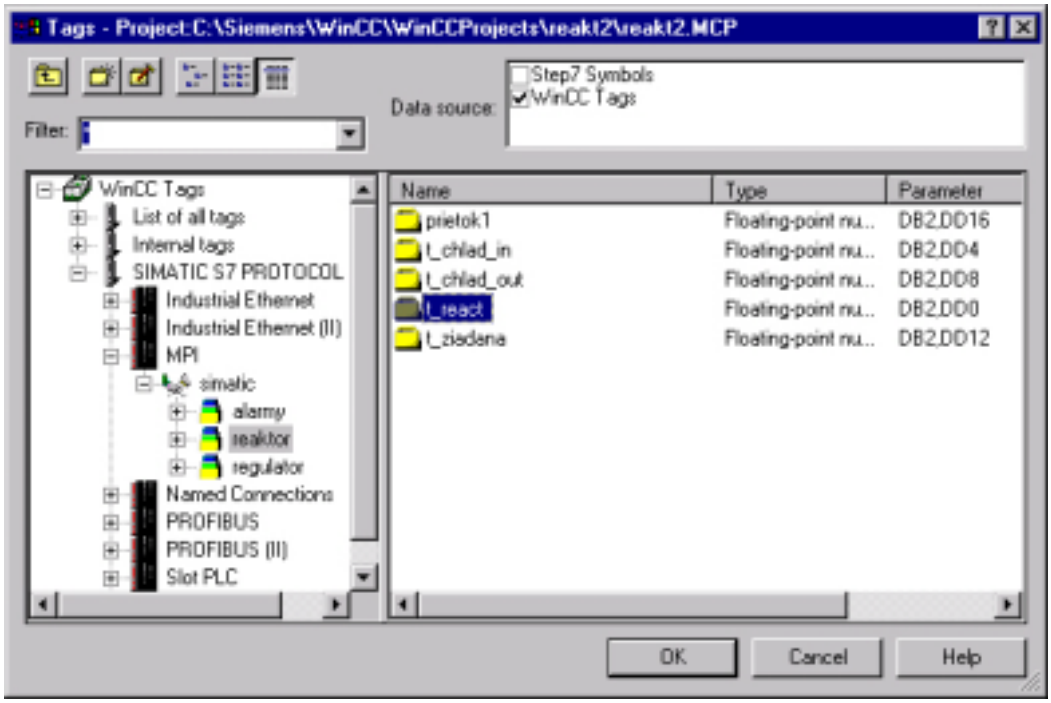
V kolonke *Tag* sa určí, na aký tag bude dané vstupno/výstupné pole napojené. Položka *Update* určuje, s akou frekvenciou budú dáta obnovované. Spodná časť okna slúžia na konfiguráciu vizuálnej stránky (veľkosť písma, font, farba písma). V prípade že užívateľ nepozná presný názov tagu, môže si ho vyhladať zo stromovej štruktúry stlačením  (obr. 10)

Po stlačení OK je daný objekt pripravený prijímať a / alebo zapisovať údaje do daného tagu. Na neskoršiu úpravu vlastností daného objektu slúži PK + *Properties*. Otvorí sa okno ako na obr. 11. V ľavej časti sa nachádza strom s kategóriami vlastností a v pravej jednotlivé vlastnosti a ich hodnoty. Najdôležitejšou je kategória *Output/Input*:

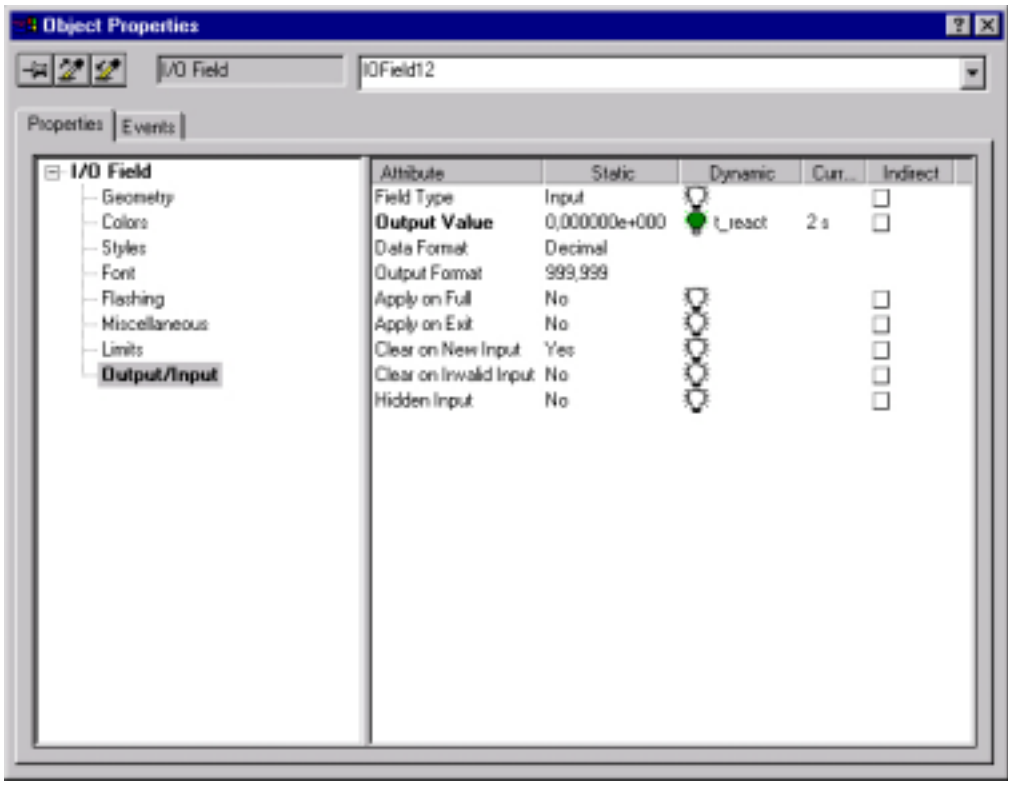
- *Field Type* – typ poľa (Input, Output, Both)
- *Output Value* – na ktorý tag je daný objekt napojený a z ktorého sa čerpajú dáta
- *Data Format* – formát výpisu (decimálne, hexadecimálne, reťazec)
- *Output Format* – pre decimálne čísla udáva počet desatinných miest

Podrobnejší popis vlastností najdôležitejších objektov sa nachádza v Tabuľke 1.

Každá vlastnosť môže byť buď statická alebo dynamická. Statická vlastnosť sa dá meniť iba v *Graphics Editore*, zatiaľ čo hodnotu dynamickej vlastnosti ovplyvňuje hodnota tagu. O tom, že je daná vlasnosť dynamická je užívateľ informovaný pomocou zelenej žiarovky pri príslušnej vlastnosti. Ak chceme danú vlastnosť pripojiť na nejaký tag, postupujeme nasledovne: PK na príslušnú nevysvietenú žiarovku + *Tag* . Z následnej stromovej ponuky sa vyberie požadovaný tag.



Obr. 10: Výber tagu zo stromovej štruktúry



Obr. 11: Vlastnoti zvoleného dynamického objektu

Objekty ovplyvňujúce udalosti

Sú umiestnené na palete *Standard – Windows Objects*. Ide o tlačidlá a iné ovládacie prvky. Bližšie si popíšeme činnosť jednoduchého tlačidla (Button). Po premiestnení symbolu tlačidla

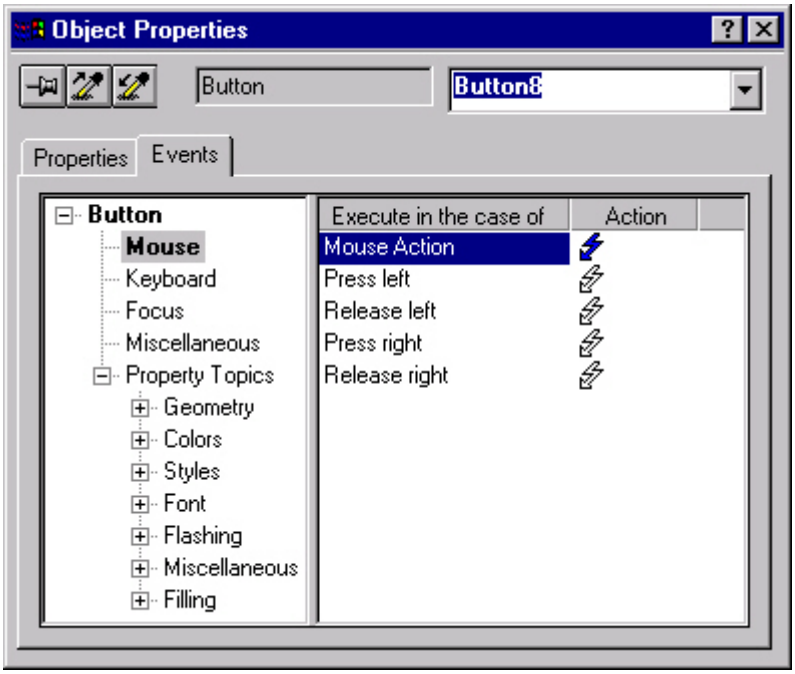
na pracovnú plochu sa otvorí okno ako na obr. 12. Pole Text obsahuje názov tlačidla, skupina Font určuje vizuálnu stránku. Do posledného okienka je možné vložiť (alebo vybrať zo zoznamu stlačením ) meno súboru s uloženým obrázkom, na ktorý sa po stlačení tohto tlačidla systém prepne.

Tabuľka 1:

Prvok	**Vlastnosť**	**Popis**
I/O Field	Display	Ak 1 tak je objekt viditeľný, ak 0 tak nie je
	Field Type	Typ poľa (vstupné, výstupné, both)
	Output Value	Sem treba pripojiť daný tag
	Data Format	Bin / hex / dec / string
Option Group	Number of Boxes	Počet prepínacích možností
	Display	Ak 1 tak je objekt viditeľný, ak 0 tak nie je
	Selected Box	Číslo určujúce, ktorá z možností bude pri štarte označená
Check Box	Number of Boxes	Počet prepínacích možností
	Display	Ak 1 tak je objekt viditeľný, ak 0 tak nie je
	Selected Box	Číslo určujúce, ktorá z možností bude pri štarte označená (binarne 11 znamená, že budú aktívne možnosti 1 a 2)
Static Text	Display	Ak 1 tak je objekt viditeľný, ak 0 tak nie je
	Flashing Background Active	Ak 1 tak bude textu blikať pozadie (použijú sa farby nastavené v Flashing Background Color On / Off)
	Flashing Text Active	Ak 1 tak bude blikať text (farby ako nastavené v Flashing Text Color On / Off)
Bar	Bar Direction	Orientácia ukazovateľa na výšku alebo na šírku
	Process Driver Connection	Sem sa napojí príslušný tag
	Maximum Value	Udáva maximálny rozsah ukazovateľa
	Zero Point Value	Udáva nulu na ukazovateli
	Minimum Value	Určuje minimálnu hodnotu ukazovateľa
Gauge Control	Display	Ak 1 tak je objekt viditeľný, ak 0 tak nie je
	Delta	Udáva krok medzi jednotlivými dielikami na ukazovateli
	Value	Sem sa napojí tag, ktorého hodnota sa má zobraziť
	ValueMax / ValueMin	Horný a dolný rozsah ukazovateľa
	Caption	Sem sa vloží text, ktorý sa zobrazí na ukazovateli
	WarningColor / DangerColor / NormalColor	Udáva akú farbu budú mať jednotlivé pásma na ukazovateli
	Warning	Číslo udávajúce začiatok vzniku tzv. warning situácie (normal – warning – danger)

Programovanie vo WinCC

Každý ovládací prvok môže reagovať na udalosti. Udalosťou je kliknutie myšou alebo stlačenie nejakej klávesy. PK + *Properties* sa otvorí okno ako na obr. 12.



Obr. 12: Definovanie obsluhy jednotlivých udalostí

Najskôr je potrebné sa prepnúť na záložku *Events*. Tu sa nachádzajú jednotlivé udalosti. Ak je značka blesku pri danej udalosti sfarbená, je ku tejto udalosti definovaná nejaká akcia. Akcie delíme podľa typu na:

1. Priame akcie – *Direct Connection* (modrý blesk)
2. Akcie jazyka C – *C action* (zelený blesk)

Programovanie pomocou Direct Connection

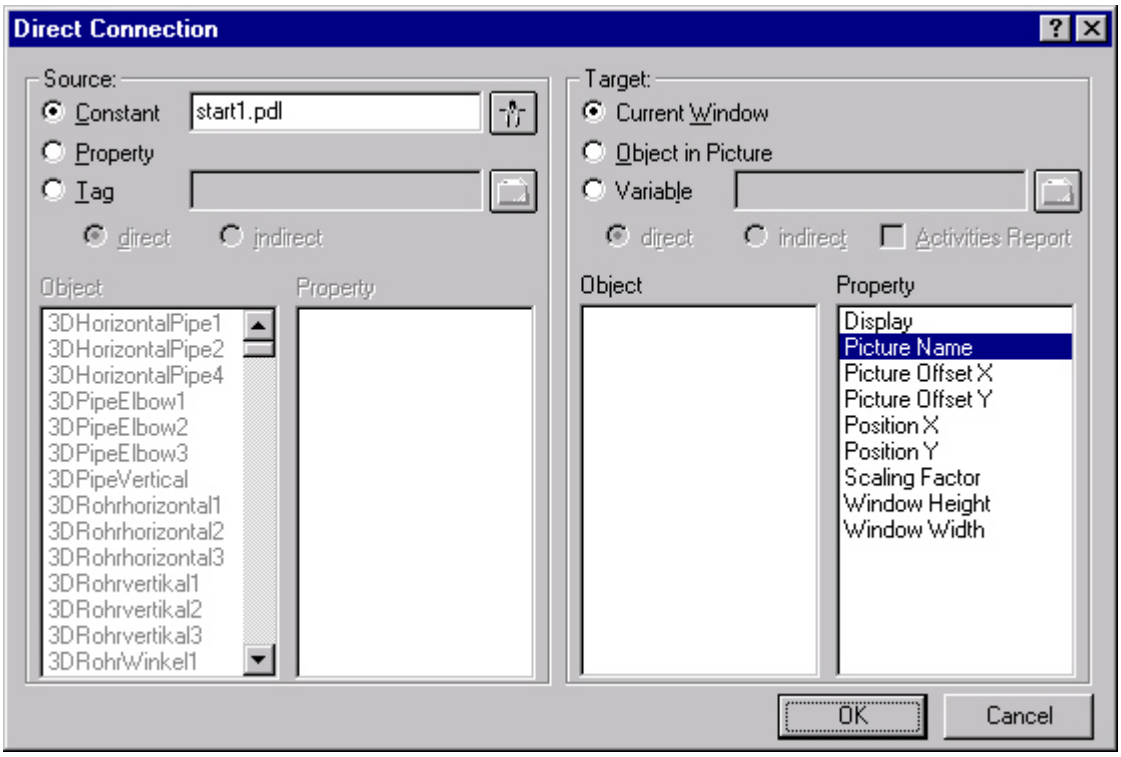
Najjednoduchším príkladom priamej akcie je použitie tlačidla na prepnutie sa do inej obrazovky (táto musí byť samozrejme dopredu vytvorená v *Graphics Editor*e a uložená na disku v súbore). PK na ikonku blesku pri príslušnej udalosti sa z príslušnej ponuky vyberie *Direct Connection* (obr. 13). Tu sa v bloku *Source* zvolí kolonka *Constant* a do príslušného riadku sa vloží názov súboru, ktorý obsahuje príslušnú obrazovku. V pravej časti okna sa z ponuky *Property* označí položka *Picture Name*.

Programovanie pomocou C action

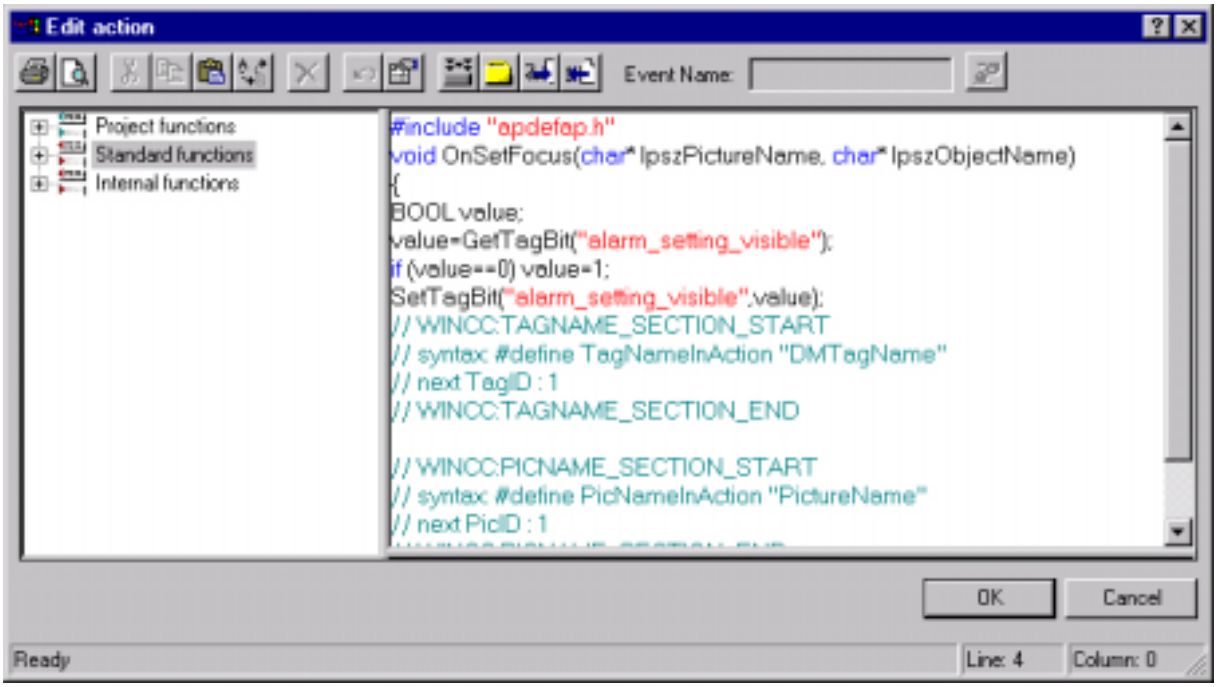
PK na ikonku blesku sa vyberie *C Action*. Otvorí sa okno ako na obr. 14. V pravej časti okna sa nachádza editovacie okno, kde sa píše kód podľa zvyklostí jazyka C. V ľavej časti sa nachádza stromová ponuka jednotlivých preddefinovaných funkcií. Dôležitá je najmä skupina *Internal functions / Tag*. Tu sa nachádza skupina funkcií, pomocou ktorých sa zisťovať a nastavovať hodnota tagov. Obr. 14 ukazuje príklad kódu, ktorým reaguje daný *Gauge Control* na kliknutie myšou. Zámerom bolo, aby sa pri kliknutí myšou zjavilo nejaké dialógové okno. Vlastnosť *Visibility* tohto okna bola napojená na interný bitový tag *alarm_setting_visible*. Pri kliknutí myšou sa vyvolá daná *C action*, ktorá najskôr zistí súčasnú hodnotu tohto tagu, a

v prípade že je nulová ju nastaví na 1. Tým sa vlastnosť dialogóvého okna *Visibility* zmení tiež na 1 a okno sa objaví.

Každá *C action* sa musí najskôr skompilovať. To sa vykoná kliknutím na ikonku  , prípadne stlačením tlačidla *OK*. V prípade, že kód neobsahuje chyby sa daná akcia skompiluje a je pripravená na použitie.



Obr. 13: Dialógové okno pre Direct Connection



Obr. 14: Okno editora akcií písaných v jazyku C

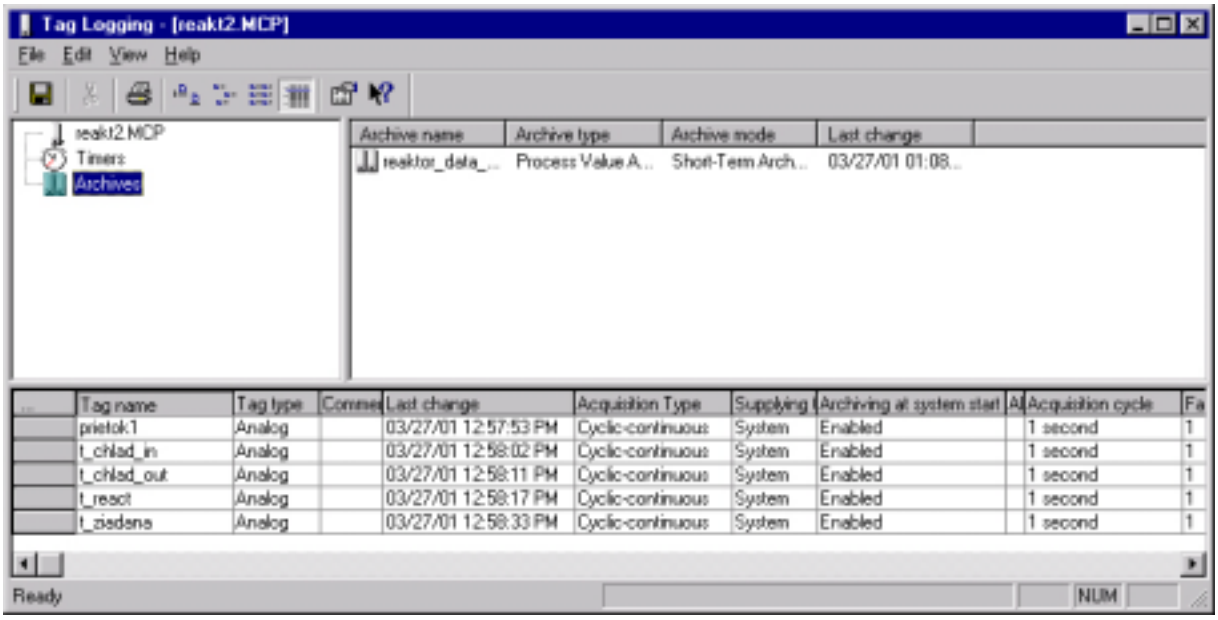
Tvorba trendov

Trendy slúžia na vizualizáciu minulých hodnôt procesných veličín. Ide vlastne o grafické zobrazenie údajov zaznamenaných za určitý časový limit. Práca s trendami sa dá rozdeliť na dve hlavné činnosti:

1. Vytvorenie a správa archívov
2. Vlastné nastavenie vizualizačného prvku

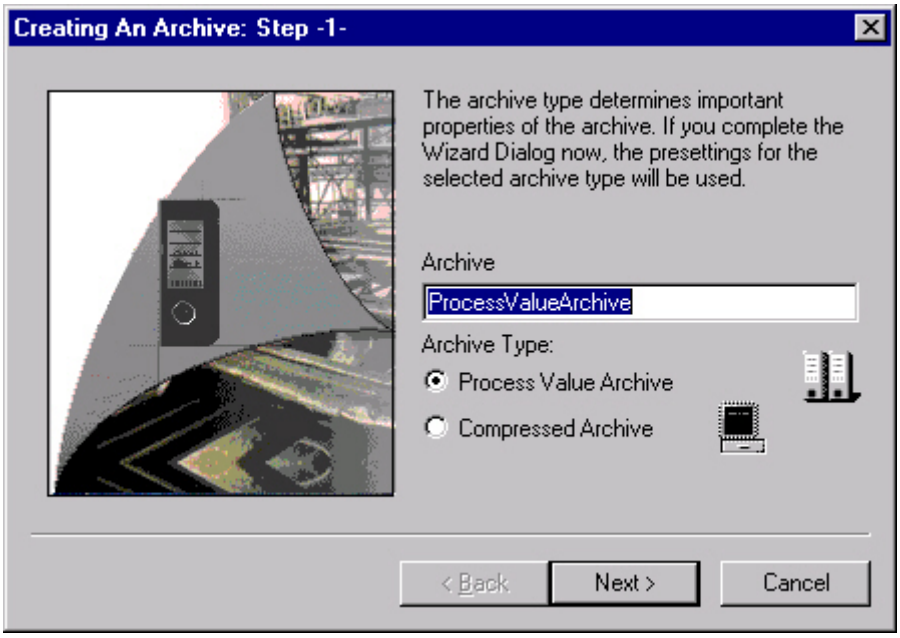
Vytvorenie a práca s archívom

Archív je súbor uložený na disku alebo v pamäti, do ktorého sa permanentne ukladá isté množstvo nameraných údajov. Na prácu s archívmi slúži modul Tag Logging (PK na Tag Logging vo WinCCExplorer + Open). Otvorí sa okno ako na obr. 15.



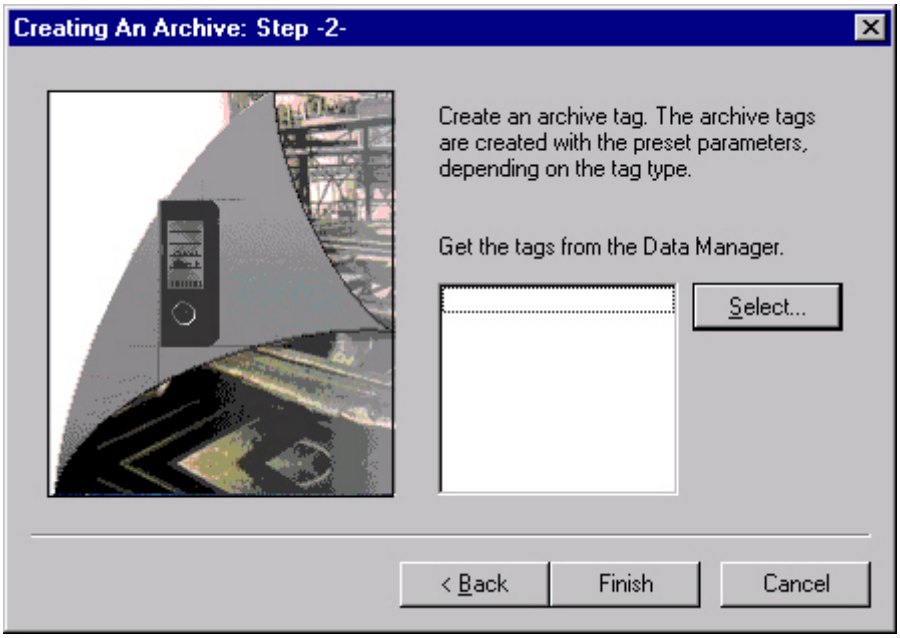
Obr. 15: Modul Tag Logging

Nový archív sa vytvorí PK na položku Archives + Archive Wizard (obr. 16)



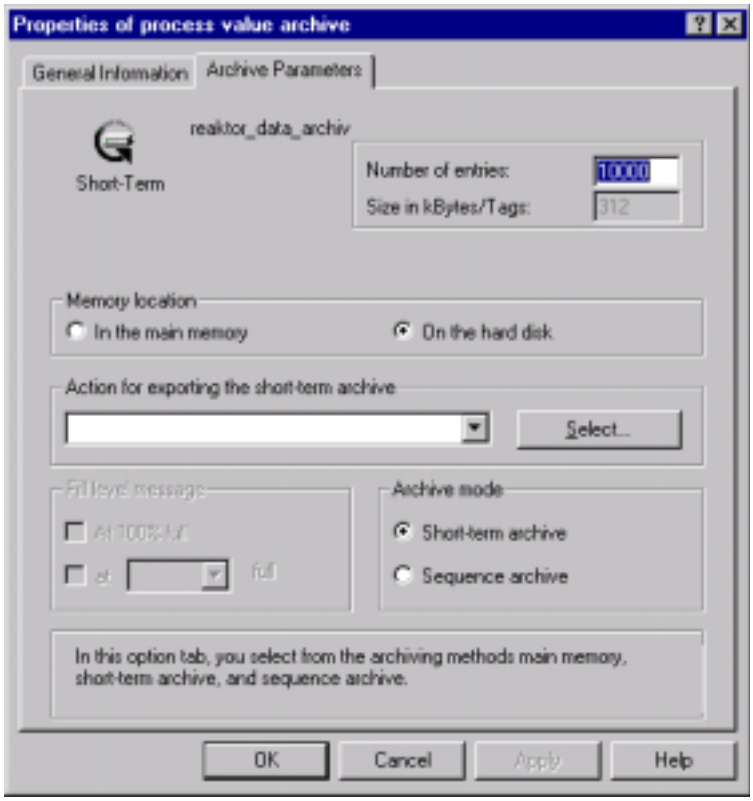
Obr. 16: Archive Wizard, krok č. 1

Do kolonky Archive sa vloží názov archívu a Archive Type bude Process Value Archive. V ďalšom kroku (obr. 17) sa pomocou tlačidla Select vyberú tagy, ktoré majú byť archivované.



Obr. 17: Archive Wizard, krok č. 2

Po stlačení tlačidla Finish sa v pravej časti okna vytvorí nový archív s požadovaným názvom. V dolnej časti okna sa nachádza zoznam tagov, ktoré sa budú ukladať v príslušnom archíve. PK na názov archívu + Properties sa otvorí okno, kde na záložke Archive Parameters je potrebné v kolonke Number of entries nastaviť počet meraní, ktoré sa do archívu uložia (obr. 18)

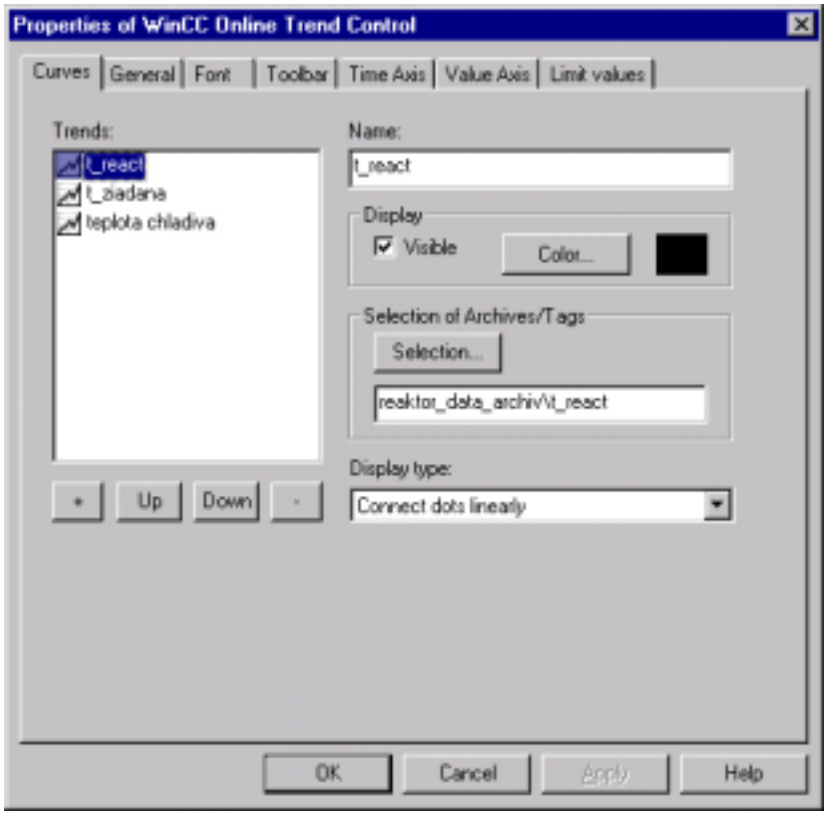


Obr. 18: Vlastnosti archívu

Dáta sa zbierajú v časovom inetrvale, ktorý je nastavený v dolnej časti okna (obr. 15) v stĺpci Acquisition Cycle. Do archívu sa dáta ukladajú v časovom horizonte určenom hodnotou stĺpca Archiving/Display Cycle.

Vizualizácia trendov

Ako vizualizačný prvok v prostredí Graphics Designera slúži WinCC Online Trend Control (paleta Controls). Po jeho umiestnení na pracovnú plochu je potrebné nastaviť jeho pripojenie na archívny súbor. 2 x ĽK na trendové okno sa otvorí okno vlastností (obr. 19)

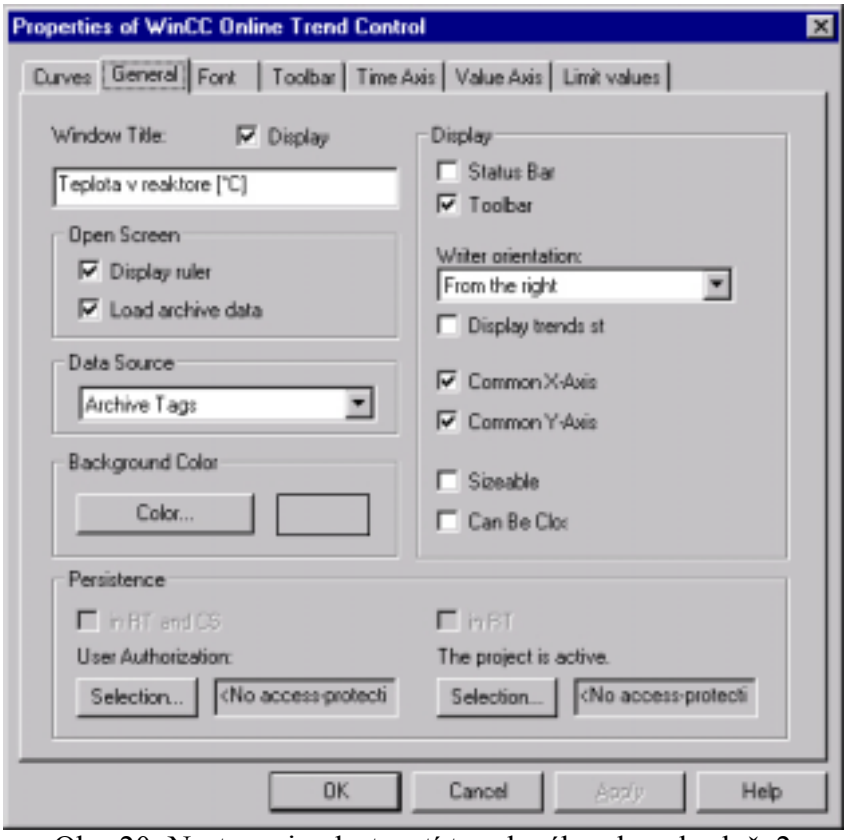


Obr. 19: Nastavenie vlastností trendového okna, krok č. 1

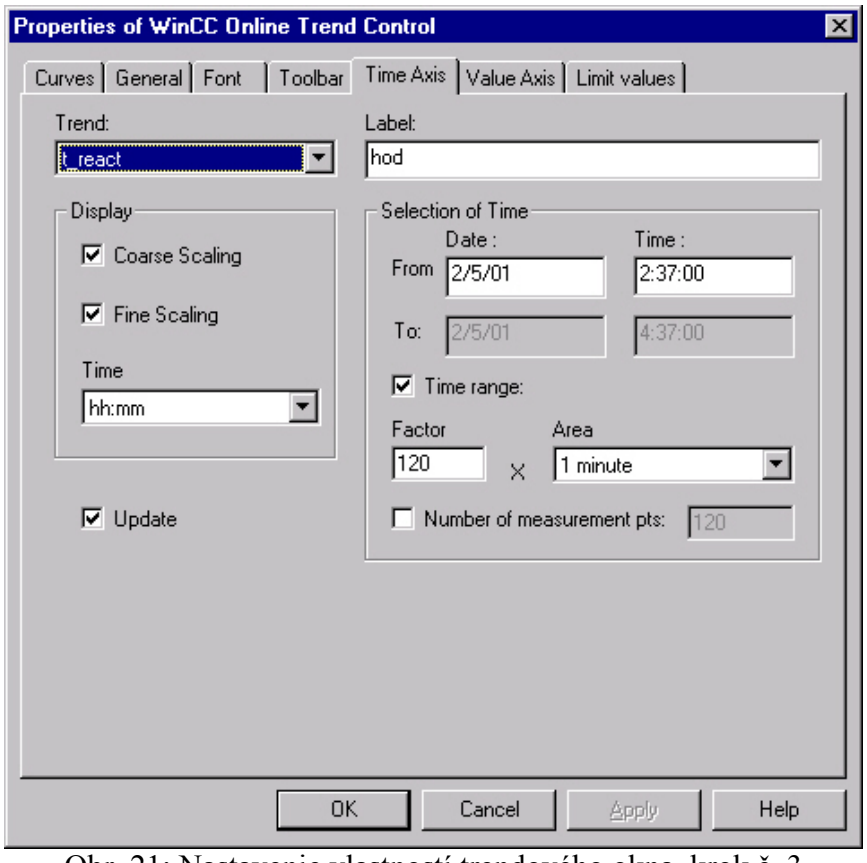
Postup pri nastavovaní vlastností trendového okna je nasledovný:

1. Pomocou tlačidla Selection je potrebné zvoliť príslušný archív a z neho tie tagy, ktorých časový priebeh chceme vidieť v príslušnom okne. V časti Trends sa zobrazia príslušné názvy tagov. Každý údaj v tomto zozname bude predstavovať na grafe samostatnú čiaru. Farba tejto čiary sa dá zmeniť pomocou tlačidla Color.
2. V časti General sa nastavia jednotlivé parametre tak, ako je to zrejmé z obr. 20. Dôležitá je najmä voľba Data Source, ktorá musí byť nastavená na Archive Tags.
3. Nastavenie záložky Time Axis je zrejmé z obr. 21. Kolonka Factor udáva celočíselný násobok časového intervalu určeného voľbou Area. V tomto prípade ide teda o časový interval 120 minút. Keďže je voľba Time Range aktivovaná, trendové okno bude zobrazovať údaje z posledných dvoch hodín.
4. Posledným krokom je nastavenie volieb na záložke Value Axis tak, ako je to zobrazené na obr. 22.

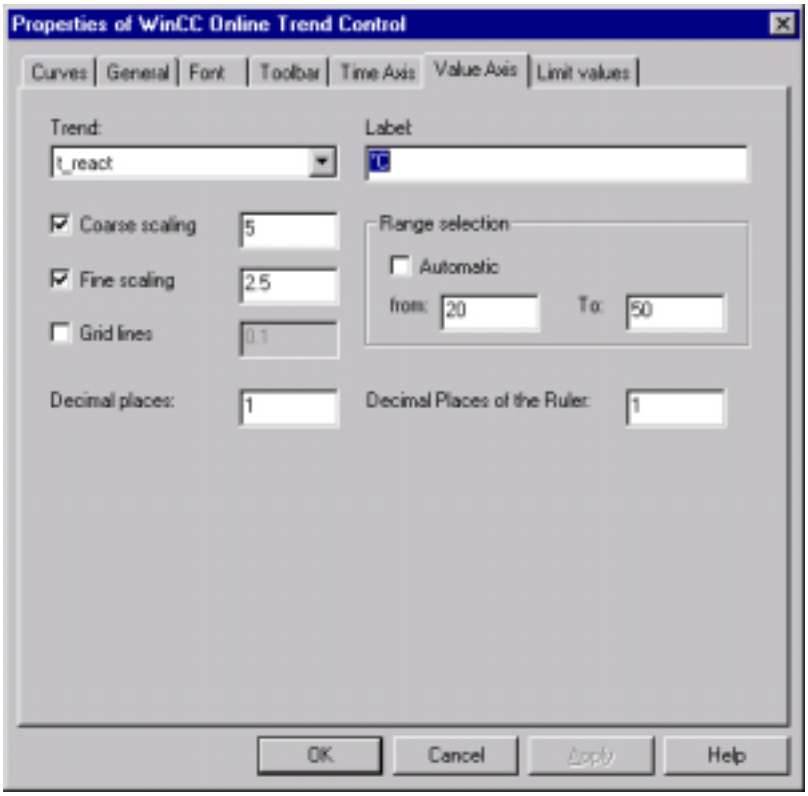
Vykonaním týchto úkonov je trendové okno pripravené zobrazovať snímané údaje.



Obr. 20: Nastavenie vlastností trendového okna, krok č. 2



Obr. 21: Nastavenie vlastností trendového okna, krok č. 3



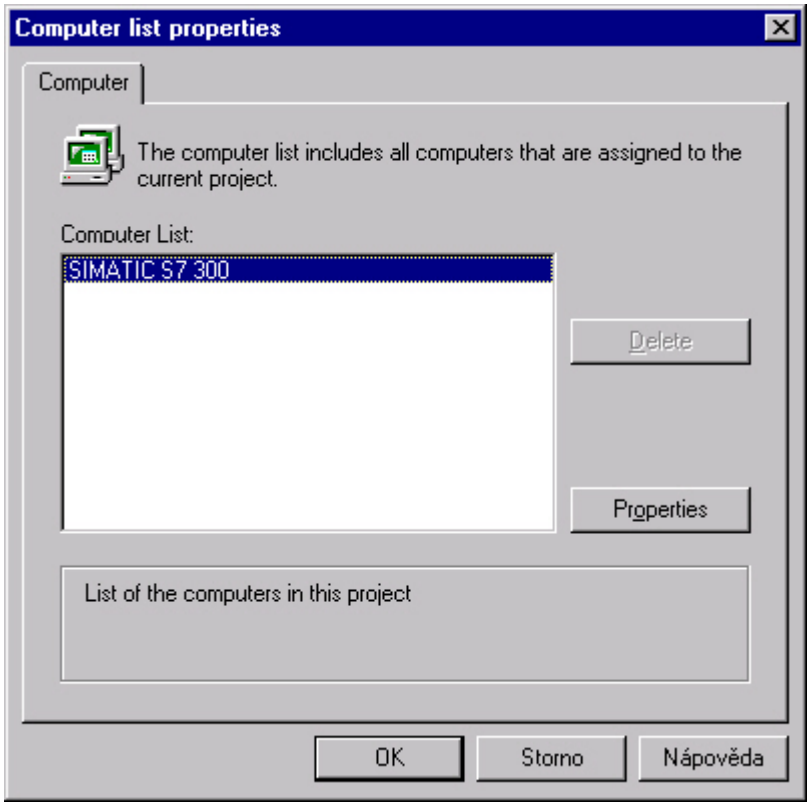
Obr. 22: Nastavenie parametrov trendového okna, krok č. 4

Aktivácia projektu

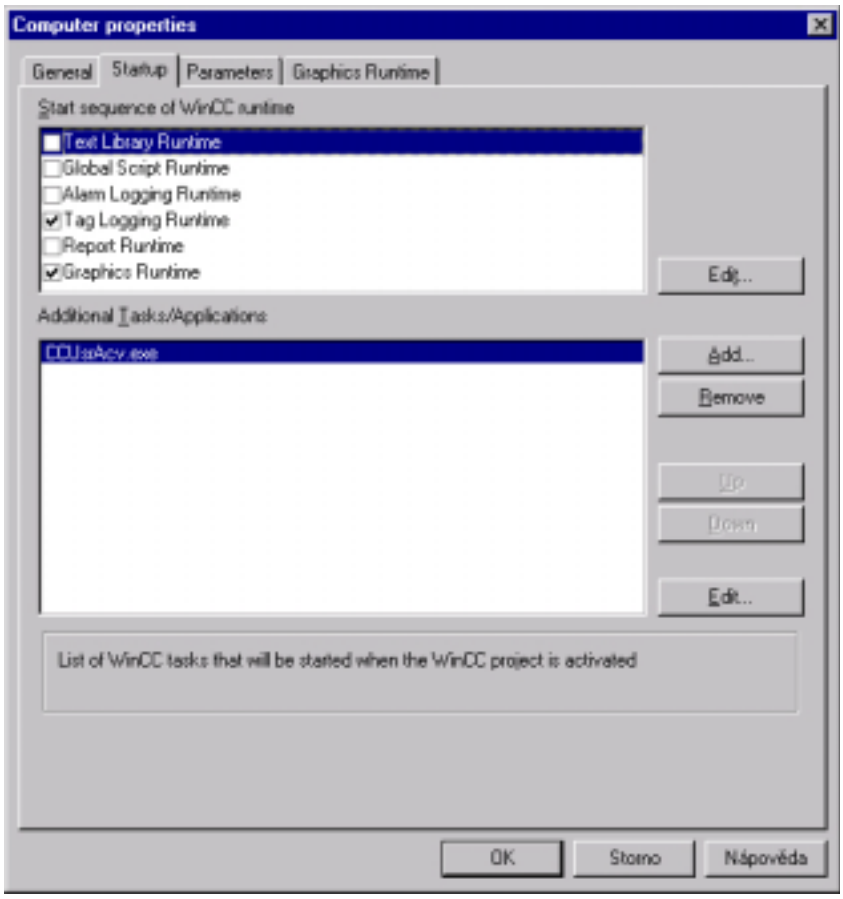
Všetky doteraz vykonané akcie slúžili na prípravu grafického rozhrania. *WinCC* však zatiaľ nekomunikuje s *PLC*. Aby program mohol prijímať signály z *PLC*, musí sa najskôr projekt skompilovať a aktivovať. Dôležitým faktorom je nastavenie parametrov výsledného modulu. To sa vykoná PK na vetvu *Computer* v hlavnom okne *WinCCExplorer*. Objaví sa okno ako na obr. 23. Kliknutím na *Properties* sa otvorí ďalšie okno, kde najdôležitejšou je záložka *Startup* (obr. 24). Tu sa nastavujú moduly, ktoré sa majú spustiť pri štarte projektu. Najzákladnejším je modul *Graphics Runtime*, ktorý zabezpečuje obsluhu grafického rozhrania. Taktiež je potrebné mať aktivovanú možnosť *Tag Logging Runtime*. Tento modul slúži na správu trendových archívov. Ostatné položky môžu ostať neaktívne. Ak sa však užívateľ rozhodne, že bude používať napr. správu alarmov, musí aktivovať príslušný *Runtime* modul. Na záložke *Graphics Runtime* (obr. 25) je najdôležitejším parametrom názov grafickej schémy, ktorá sa ako prvá ukáže po štarte projektu. Toto je zabezpečené vpísaním mena súboru do kolonky *Start Picture*. Ak sú všetky nastavenia hotové, projekt sa môže aktivovať. To sa vykoná stlačením tlačidla  na lište nástrojov.

Upozornenie: Ak je projekt aktivovaný, nedajú sa robiť zmeny v nastavení Tagov !!!!.

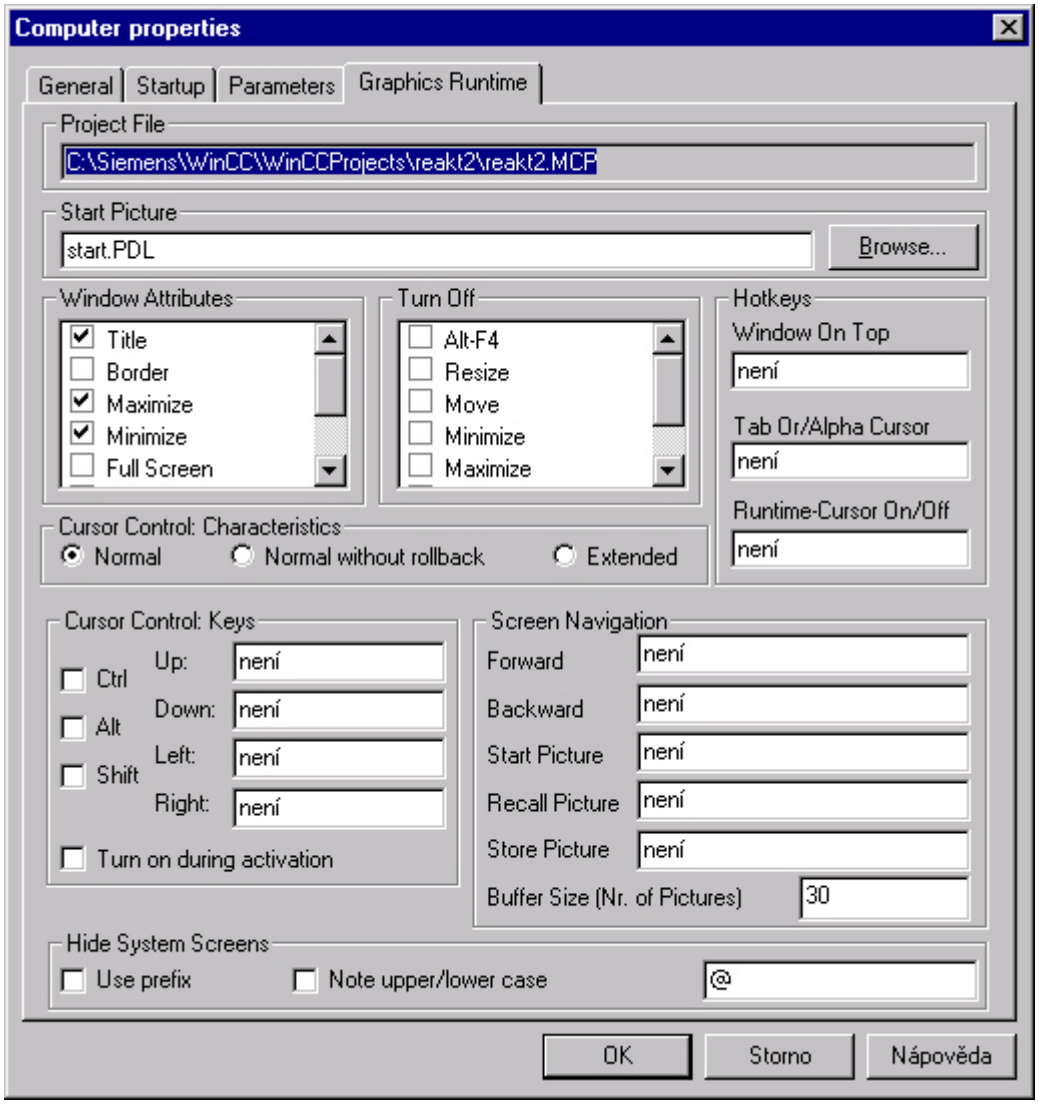
V prípade aktívneho projektu sa dá meniť grafické rozhranie pomocou *Graphics Designera*. Po vykonaní úprav v schéme je možné túto aktualizovať a zaradiť do bežiaceho projektu stlačením tlačidla  na lište nástrojov v okne *Graphics Designer*.



Obr. 23: Nastavenie parametrov výsledného projektu



Obr. 24: Nastavenie runtime modulov spúšťaných pri štarte projektu



Obr. 25: Nastavenie parametrov grafického runtime modulu