

Střední průmyslová škola a Střední odborné učiliště Trutnov, Školní 101








# **Programování PLC**

## **Simatic S7-300**

Zpracovali: Jan Kafka  
Petr Tesař

2005

## Obsah:

		Strana
1.	<b>Úvodní slovo</b> .....	4
	Používaný hardware.....	5
	Používaný software.....	9
2.	 <b>Poznámky k programování SIMATIC S7-300 pomocí STEP 7</b> .....	11
2.1	Automatizační systém SIMATIC S7-300 .....	11
2.2	Programovací software STEP 7 .....	11
3.	 <b>Instalace softwaru STEP 7</b> .....	12
4.	<b>Nastavení programovacího rozhraní (PC- adapter)</b> .....	13
5.	 <b>Co je to PLC a k čemu se používá?</b> .....	17
5.1	Co znamená pojem PLC?.....	17
5.2	Jak řídí PLC proces? .....	17
5.3	Odkud dostává PLC informace o průběhu procesu? .....	17
5.4	V čem je rozdíl mezi vypínacími a spínacími kontakty? .....	18
5.5	Jak komunikuje PLC s jednotlivými vstupními a výstupními signály? .....	18
5.6	Jak se zpracovává program v PLC? .....	19
5.7	Jak vypadají logické operace v PLC-programu? .....	20
5.7.1	Operace AND.....	20
5.7.2	Operace OR.....	22
5.7.3	Operace NOT.....	23
5.8	Jak se sestavuje PLC-program? Jak se dostane do paměti PLC? .....	24
6.	<b>Konstrukce a obsluha SIMATIC S7-300</b> .....	24
7.	 <b>Vzorová úloha</b> .....	27
8.	 <b>Složení STEP 7- projektu</b> .....	28
	Bloky programu .....	30
9.	<b>Napsání STEP 7- programu ve funkčním plánu FBD</b> .....	37
10.	<b>Testování STEP 7- programu v CPU</b> .....	50
11.	<b>Programování CPU - úvodní slovo</b> .....	51
12.	<b>Údaje pro použití CPU 313C-2DP</b> .....	52
12.1	Obsluha CPU .....	52
12.2	Oblasti paměti CPU .....	54
13.	<b>Sestavení konfigurace pro CPU</b> .....	55
14.	<b>Napsání STEP 7-programu</b> .....	64
15.	<b>Testování STEP 7-programu</b> .....	66

<b>16.</b>	<b>Archivace, dokumentace programu - úvod</b> .....	67
16.1	Symbolické zadání adresy.....	68
16.2	Program v EPROM-Memory Card - ukládání na CPU .....	69
16.3	Ukládání a vybírání projektů z archivu .....	71
<b>17.</b>	<b>Testovací a online funkce – úvodní slovo</b> .....	74
17.1	Diagnostika systému.....	78

V tomto textu jsou použity následovné symboly:



Informace



Instalace



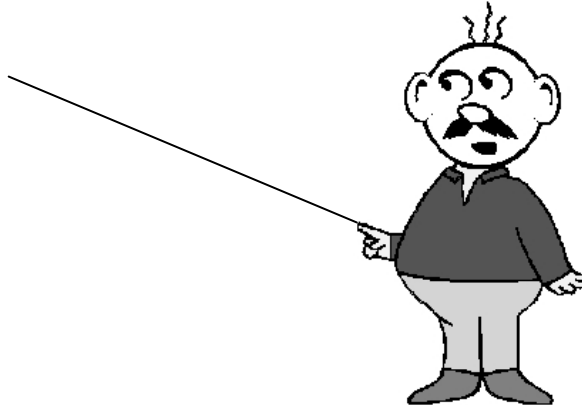
Programování



Vzorová úloha



Poznámka



## 1. Úvodní slovo

### Učební cíl:

Cílem je naučit se programovat programovatelný automat (PLC) Siemens Simatic s pomocí programovacího nástroje STEP 7. Budou zde probrány základní pravidla práce s tímto programem a na základě podrobného příkladu i způsob práce v následujících krocích:

- Instalování softwaru a nastavení programového rozhraní
- Vysvětlení co je to CPU a jak pracuje
- Struktura a obsluha CPU SIMATIC S7-300
- Vypracování vzorového příkladu
- Uložení a testování vzorového příkladu

### Předpoklady:

Pro úspěšné zvládnutí tohoto modulu se předpokládá následovní znalost:

- Zvládnutí systému Windows

## Používaný hardware

- 1 PC, operační systém Windows 95/98/2000/ME/NT4.0/XP, min. 600MHz, 256MB RAM, volná paměť na HDD min 300 MB
- 2 PLC Simatic S7-300, typ 313 C – 2DP ( k dispozici zatím 6ks)



### Základní technická data:

- integrováno 16 digitálních vstupů 24V DC
- integrováno 16 digitálních výstupů 24V DC, 0,5A
- možnost připojení dalších až 32 jednotek
- možnost propojení do sítě Profibus či MPI
- zálohování programu pomocí paměťové karty MMC
- programování pomocí programu STEP7 z PC
- integrované funkce – čítač 24V, 30kHz
  - pulsní výstup – 3 kanály pulsně šířkové modulace max. 2,5kHz
  - měření frekvence – 3 kanály max. 30kHz

### 3 MPI rozhraní pro PC (např. PC – adapter USB, popř. PCI karta CP 5611 )



Na obrázku PC adapter s propojovacími kabely a softwarem pro konfiguraci adaptéru (dodáváno jako komplet).

MPI rozhraní slouží k propojení PC a Simatic S7-300 (CPU) přes USB port PC. Přenosová rychlost 19,2kB/s.

MPI rozhraní lze nahradit PCI kartou typ CP5611. Karta má oproti tomuto rozhraní vyšší přenosovou rychlost

#### 4 Analogový modul SM334



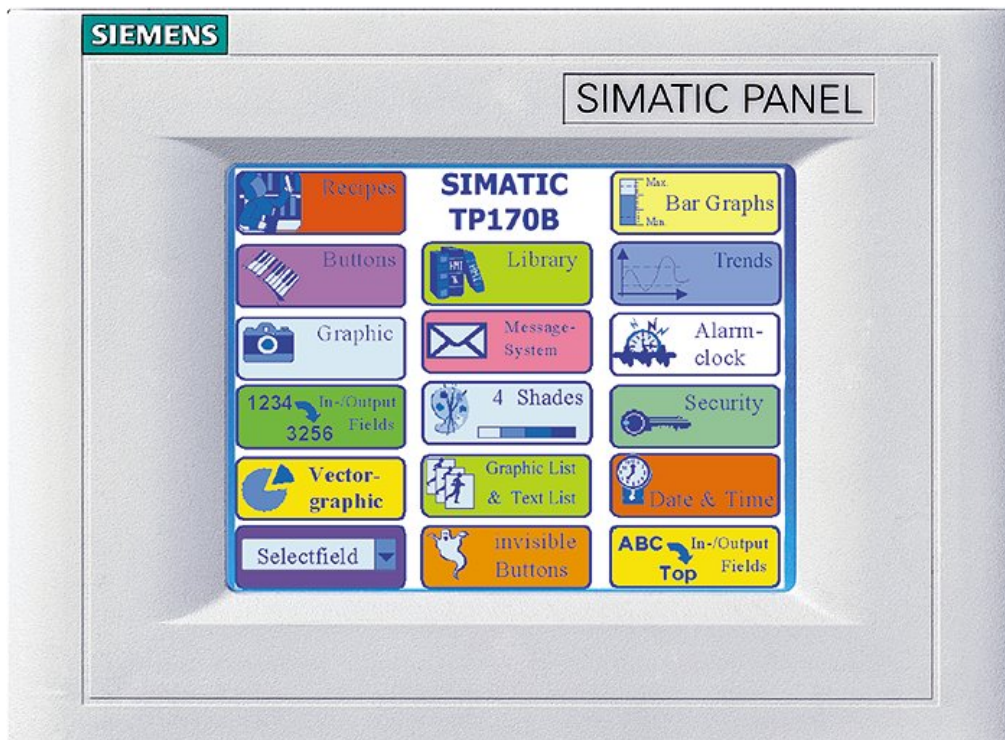
Základní charakteristika:

- 4 analogové vstupy (0-20mA, 0-10V)
- 2 analogové výstupy (0-10V)
- v modulu analogového vstupu integrován A/D převodník

## 5 Hardware pro vizualizaci – ovládací panel TP170B



Úvodní menu TP 170B (viz obr):



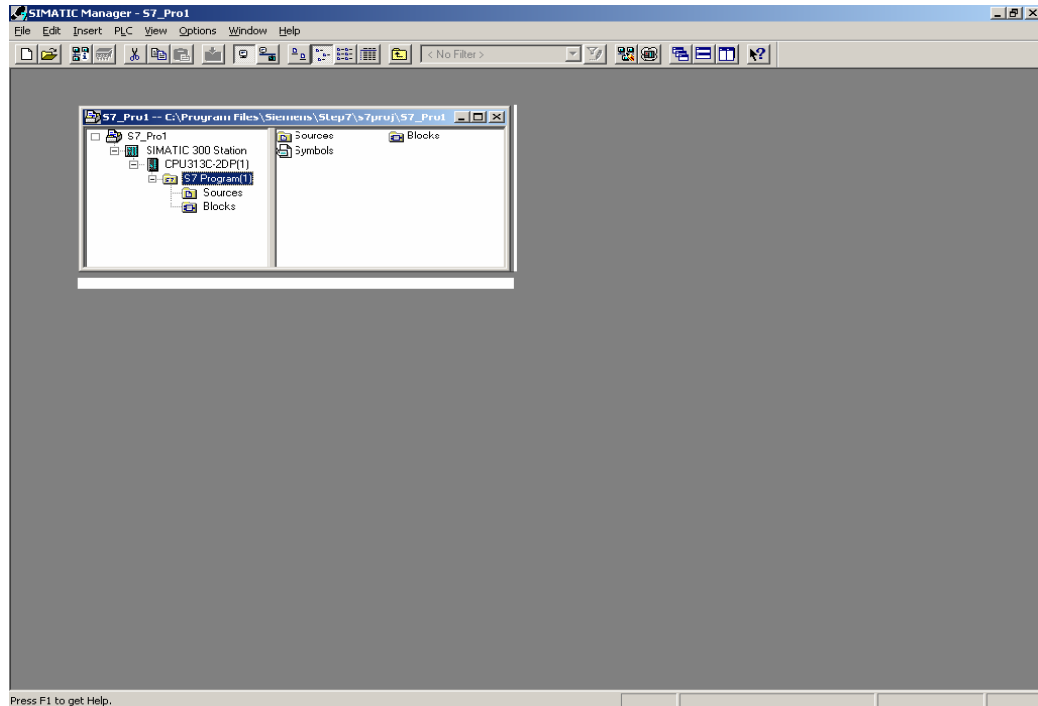
Základní technická data:

- LCD displej (16 barev)
- dotyková obrazovka 5,7", rozlišení 320x240 bodů
- slot pro paměťovou kartu typu CF
- rozhraní – sériové, Profibus, MPI
- konfigurační software WinCC Flexible

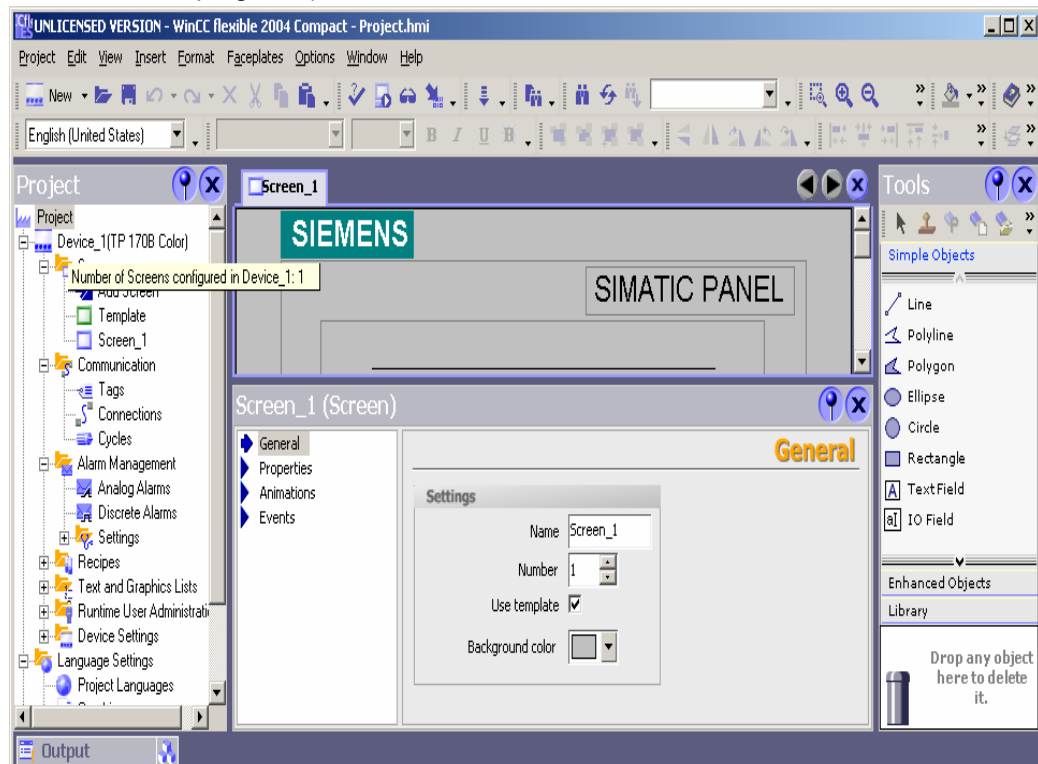


## Používaný software

- 1 Program STEP7 – pro vytváření programů a přenášení programů do CPU Simatic (na obrázku úvodní obrazovka programu)

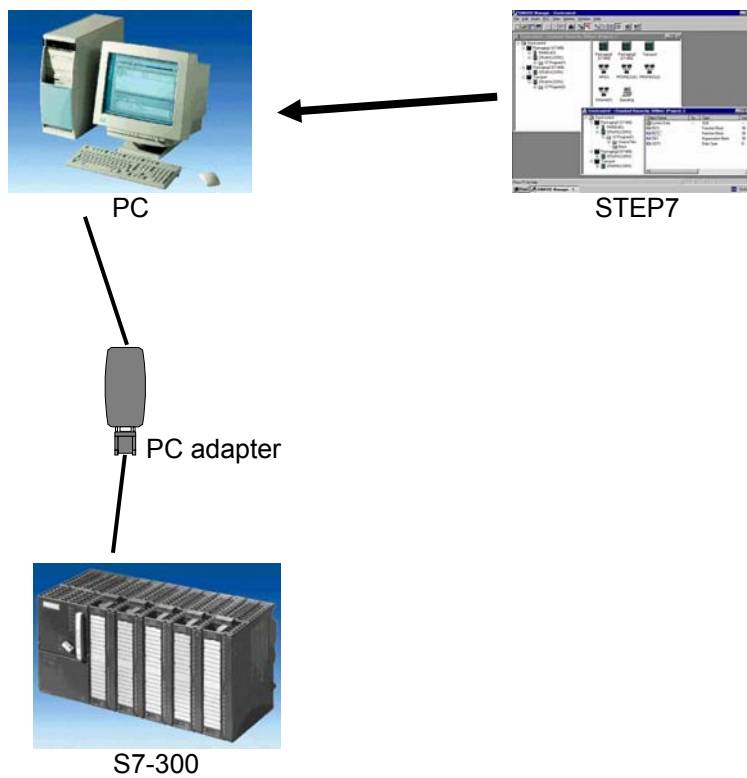


- 2 WinCC flexible – konfigurace operátorských panelů, vizualizace procesů (na obrázku úvodní obrazovka programu)



**Použitá hardwarová konfigurace pro práci v těchto skriptech:**

- Zdroj napájení: externí, 24V, 1A
- PC Athlon 2,4 GHz
- Simatic S7-300, CPU 313 C – 2DP
- Analogová karta SM334
- PC adaptér



## 2. Poznámky k programování SIMATIC S7-300 pomocí STEP 7

### 2.1 Automatizační systém SIMATIC S7-300



Automatizační systém SIMATIC S7-300 je, modulární PLC řídicí systém pro nízké a střední rozsahy výkonů.

K dispozici jsou různé typy CPU. Existuje rovněž široké spektrum modulů s možností rozšíření až na 32 modulů pro optimální přizpůsobení pro řešení automatizace výrobního procesu různé složitosti.

Nejčastěji používané moduly jsou:

- digitální vstupní moduly 24V DC
- digitální výstupní moduly 24V DC
- analogové vstupní moduly – napětí, proud, odpor, termočlánek
- analogové výstupní moduly – napětí, proud
- interfaceové moduly – slouží k propojení sběrnic (řad)
- funkční moduly – zajišťují funkce jako „počítání“, pozicování, regulace PID

Možnost propojení do sítě Multipoint interface (MPI), Profibus, Ethernet. Centrální připojení PC s přístupem do všech modulů.

Simatic S7-300 se skládá ze zdroje proudu, z CPU (procesové jednotky) a vstupních a výstupních modulů. Eventuálně je možné zařadit i komunikační procesory a funkční moduly pro speciální úlohy jako např. řízení krokových motorů.

Řízení programovatelnou pamětí (SPS) dohlíží na stroj nebo proces pomocí S7-programu. Přitom v S7-programu zareagují vstupní a výstupní celky přes vstupní a výstupní adresy.

Systém je naprogramován pomocí softwaru STEP 7.

### 2.2 Programovací software STEP 7



Software STEP 7 je nástroj programování pro automatizační systémy:

- SIMATIC S7-300
- SIMATIC S7-400

STEP 7 je programovací a konfigurační software určený pro profesionální použití spolu s řídicími systémy Simatic. Pomocí STEP 7 je možné použít následující funkce pro automatizaci zařízení:

- Konfigurace a parametrování hardwaru
- Stanovení komunikace
- Programování
- Test, uvedení do provozu a servis
- Dokumentace, archivace
- Provozní a diagnostické funkce

### 3. Instalace softwaru STEP 7



STEP 7 existuje ve dvou variantách:

- **STEP 7 Professional** se skládá z následujících částí:
  - STEP 7 Basic včetně osvědčených programovacích jazyků STL, LAD, FBD
  - S7-GRAPH pro grafické programování sekvenčních řízení
  - S7-SCL vyšší programovací jazyk pro realizaci komplexnějších úloh
  - S7-PLCSIM pro off-line simulaci automatizačních řešení

Škola má k dispozici balíček STEP7 Pro Training a 12-ti multilicencemi. Balíček obsahuje též 48 ks studentských verzí STEP 7

- **STEP 7 Student Version** obsahuje přídatný soubor S7- PLCSIM. Tento software se musí autorizovat, od autorizace je platný 120 dní.

STEP 7 se dodává na CD-ROM, přičemž je k STEP 7 Professional přiložena disketa, obsahující License Key (autorizace), která se musí přenést do PC a teprve pak je umožněné používání STEP 7. Abychom mohli použít autorizaci i na jiném PC, dá se znovu stáhnout na disketu a vložit na jiný PC. Ve verzi STEP 7 Professional V5.3 je možné tuto autorizaci spravovat i přes síť.

Při instalaci STEP 7 postupujte následovně:

1. Vložte CD STEP 7 do CD-ROM mechaniky.
2. Setup-program se automaticky spustí. Jestliže ne, spustěte ho dvojitým kliknutím na ,→ **setup.exe**'.  
Setup-program Vás doprovází celou instalaci STEP 7.
3. Pro použití STEP 7 Professional se ve Vašem PC musí nacházet License Key (autorizace), t.z. uživatelské právo. Tuto licenci musíte přenést z autorizační diskety do Vašeho počítače.

V dialogu se Setup-programem budete tázáni, jestli chcete uskutečnit autorizaci. Když zvolíte ,**Yes**' ,**Ano**' , musíte vložit disketu a oprávnění pro autorizace se přenesou na Váš počítač.

#### 4. Nastavení programovacího rozhraní (PC-adapter)



Aby bylo možné PLC Simatic programovat z PC nebo PG (speciální přenosné programovací zařízení) pomocí STEP7, je potřebné MPI spojení.

MPI je zkratka pro **M**ulti **P**oint **I**nterface (vícebodové rozhraní) a jedná se o komunikační rozhraní až pro 32 účastníků, které se používá k programování, na obsluhu & pozorování s HMI a pro výměnu dat mezi CPU několika SIMATIC S7.

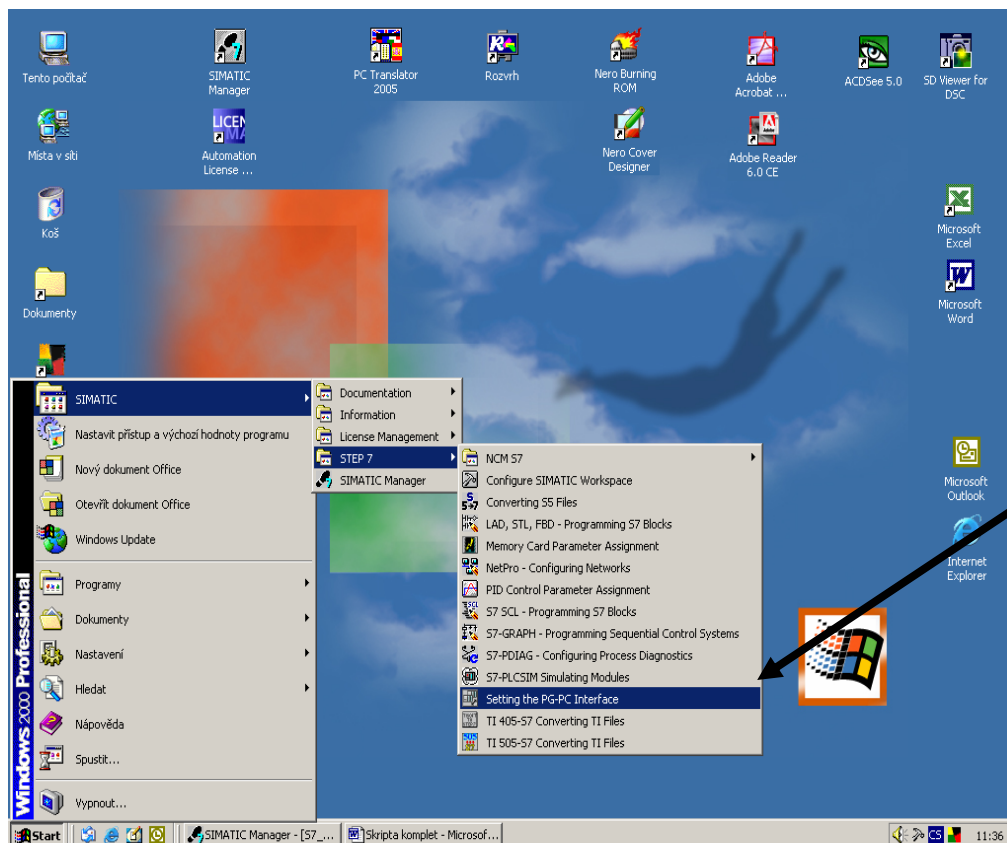
Každá CPU ze SIMATIC S7-300 vlastní takovéto integrované rozhraní.

Pro připojení PC, PG nebo laptopu na MPI existuje několik možností:

- Integrovaný ISA - komunikační procesory pro PG
- ISA - komunikační procesory pro PC (např. MPI-ISA Card)
- PCI - komunikační procesory pro PC (např. CP5611)
- PCMCIA - komunikační procesory pro laptop (např. CP5511)
- Adapter pro komunikaci přes sériové nebo USB rozhraní počítače nebo laptopu (např. PC-adapter)

V následujících krocích je popsáno nastavení a parametrizace PC-adaptéru pro počítač.

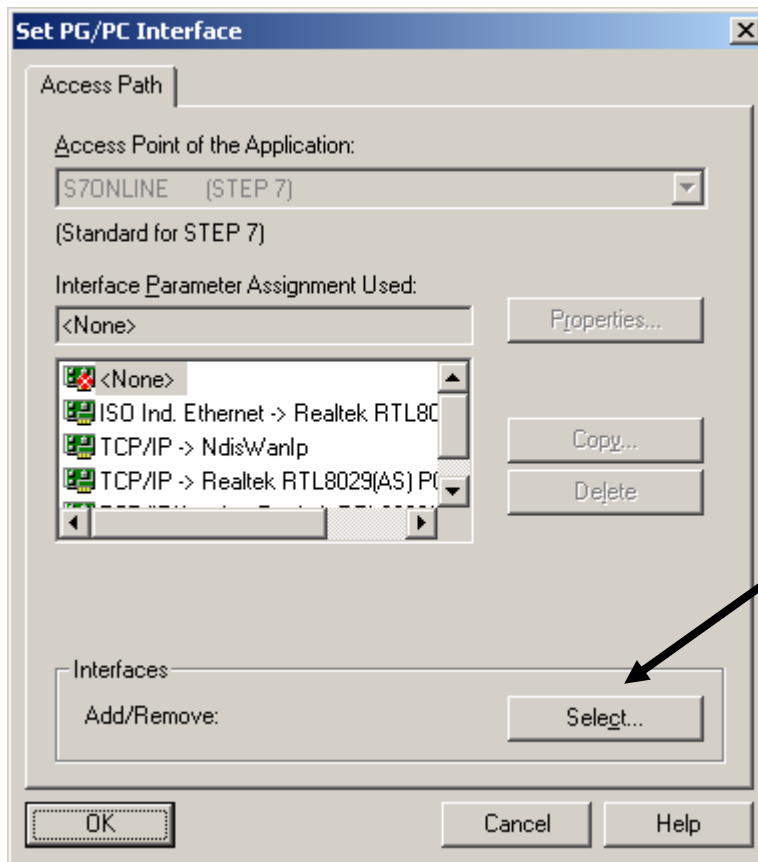
##### 4.1 Vyvolat 'Setting To PG-PC Interface' (nastavit rozhraní). ( → Start → SIMATIC → STEP7 → Setting To PG-PC Interface)



Klikněte na  
'Setting To  
PG-PC  
Interface'  
„Nastavit PG-  
PC-rozhraní“

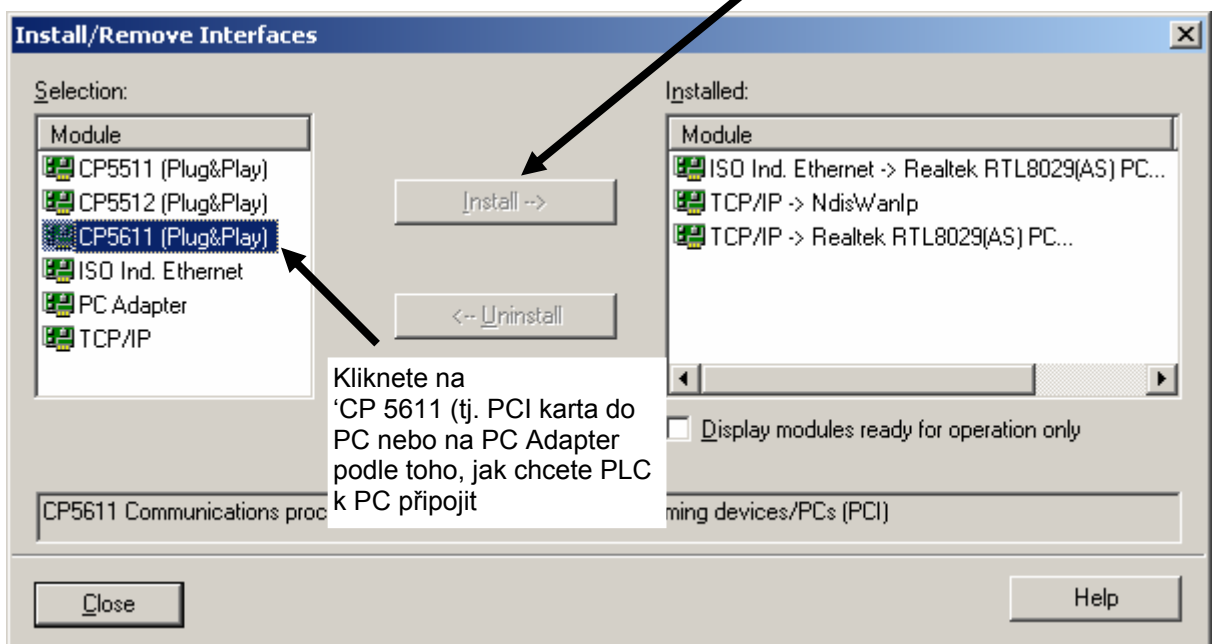


4.2 Zvolte tlačítko **„Select“ (zvolte)** , který je k dispozici jako MPI-rozhraní. ( → Select)



Klikněte na 'Select' -  
'Zvolit'

4.3 V levém okně zvolte žádaný celek **„Karta CP5611 nebo PC-Adapter“** a **„Install“**- „instalovat“ (→ PC-Adapter → Install).



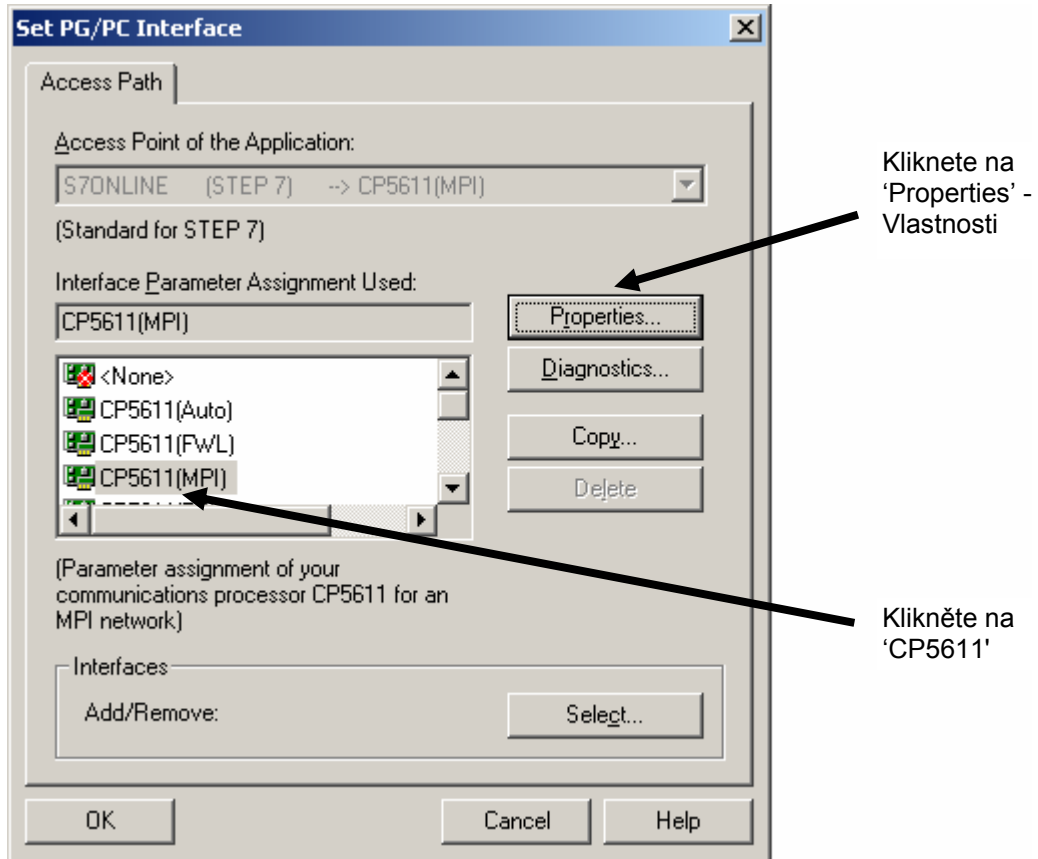
Kliknete na  
'Install'

Kliknete na  
'CP 5611 (tj. PCI karta do  
PC nebo na PC Adapter  
podle toho, jak chcete PLC  
k PC připojit

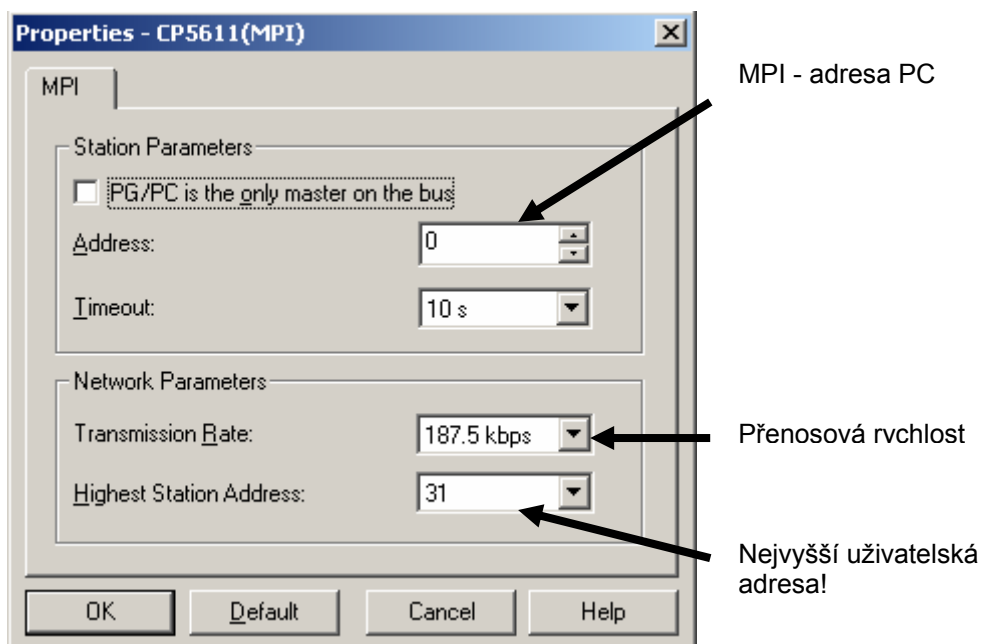


4.4 V dalším okně potvrďte s **Yes** - „Ano“, jestliže nově zvolené rozhraní ze STEP7 se má užívat jako standardní rozhraní pro vstup v **Online**. Pak zavřete okno pro volbu rozhraní tlačítkem **Close**. (→ Yes → Close )

4.5 Zvolte **Properties** „Vlastnosti“ **Karty CP5611**



4.6 Stanovte **Adress** a **Transmission Rate** - Rychlost přenosu





**Poznámka:** Doporučuje se převzít už nastavené hodnoty potvrzením tlačítka OK! Rychlost přenosu dat u PC adaptéru max. 19,2 kB/s




4.7 Pro otestování správného nastavení, vyvolejte **„SIMATIC Manager“** dvojitým kliknutím na následující tlačítko na ploše: ( → SIMATIC Manager)



SIMATIC Manager

4.8 Propojíme MPI-rozhraní počítače na MPI-rozhraní CPU a zapne se zdroj napájecího napětí pro CPU (24V). MPI-rozhraní najdete na CPU ve formě 9-pólové zásuvky za přední krytkou CPU. Pokud jsou na CPU tyto konektory dva, použijte ten levý. Pravý konektor slouží pro připojení Profibusu.

4.9 Pokud nyní klikneme na tlačítko  – **„Accesible nodes“** - ‚Dosažitelní účastníci‘, objeví se, když byly všechny parametry správně zvolené, následující obraz se zakladačem pro dosažitelné MPI-účastníky. V okně najdeme MPI-adresu připojené CPU, která má základní nastavení 2.

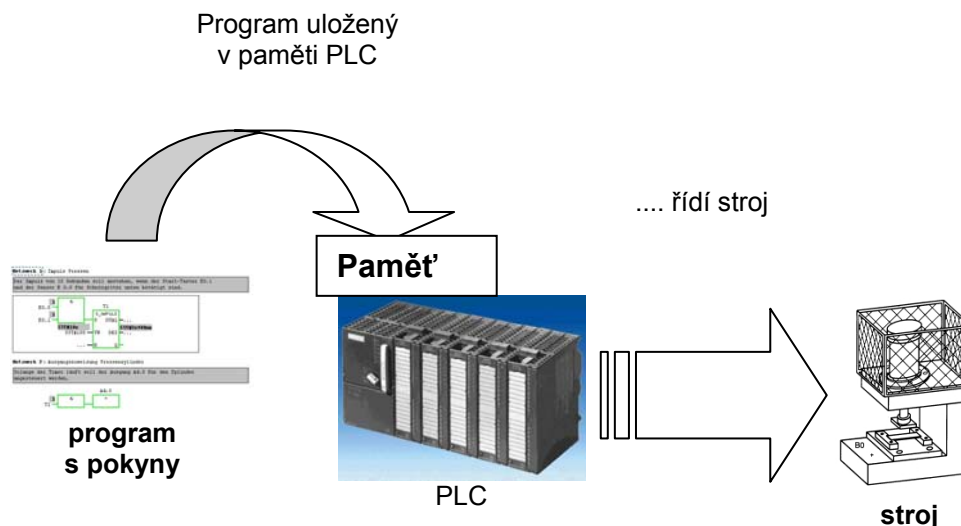


## 5. Co je to PLC a kde se PLC používá?

### 5.1 Co znamená pojem PLC?



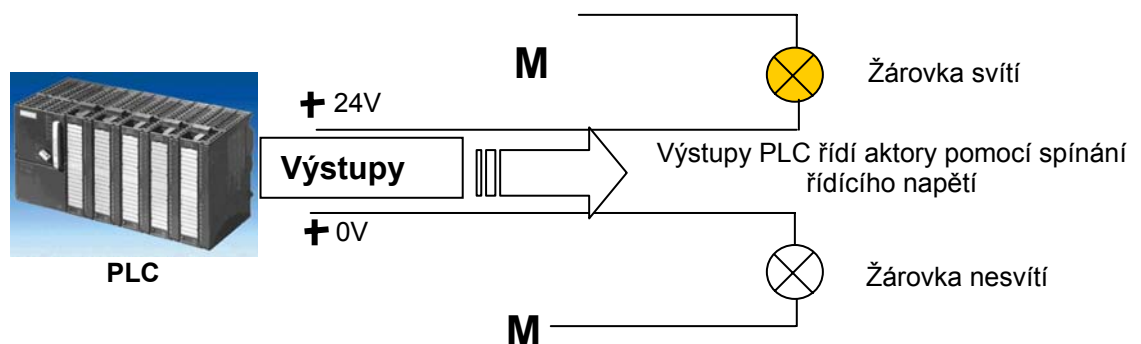
**PLC** je zkratka pro **Programmable Logic Controller**, neboli programovatelný automat. V německé literatuře se setkáváme s pojmem **SPS - Speicherprogrammierbare Steuerung**. Tento pojem představuje přístroj, který řídí jistý proces (např. tiskací stroj pro tisk novin, plnicí stroj pro cement, lis pro umělohmotných výlisky, atd... ).  
To se děje na základě pokynů programu, který se nachází v paměti přístroje.



### 5.2 Jak PLC řídí nějaký proces?



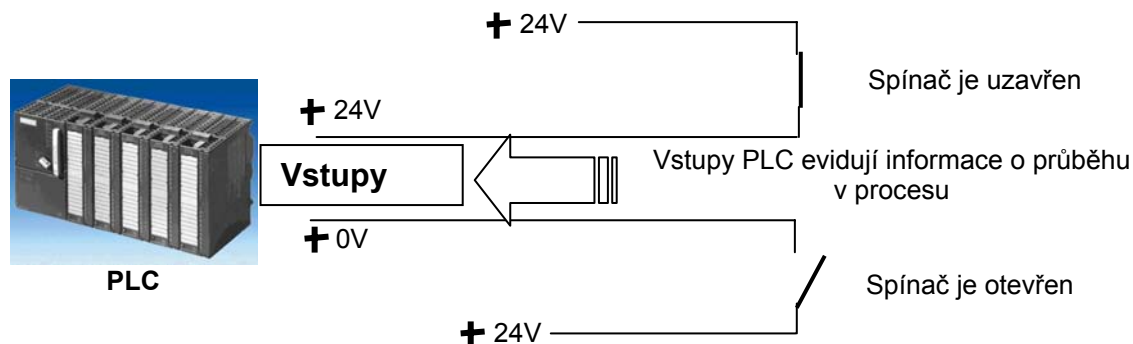
PLC řídí proces tím, že jeho **výstupy** jsou napojeny na řídicí napětí např. 24V, které na výstupu buď je, nebo není. Tím je možné spustit a vypnout **aktory** - motory, ventily zavřít a otevřít, nebo lampy rozsvítit a zhasnout.



### 5.3 Odkud dostává PLC informace o stavu procesu?



Informace o procesu dostává PLC od tzv. **signálních čidel**, které jsou spojeny se **vstupy** PLC. Tyto čidla mohou být např. senzory, které zaregistrují, jestli se obrobek nachází na určené pozici. Mohou to být ale i jednoduché spínače nebo tlačítka (např. koncový spínač), které mohou být otevřené nebo zavřené. Přitom se rozlišuje ještě mezi **rozpínacími kontakty**, které jsou v klidu zavřeny, a **spínacími kontakty**, které jsou v klidu otevřené (viz další obr.).

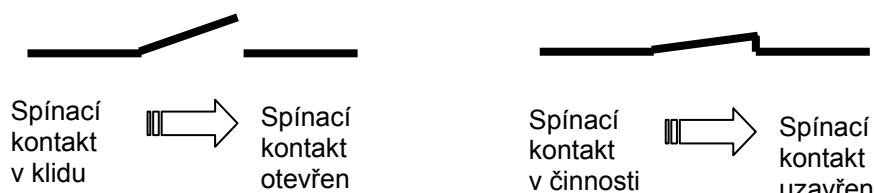


#### 5.4 V čem spočívá rozdíl mezi vypínacími a spínacími kontakty?

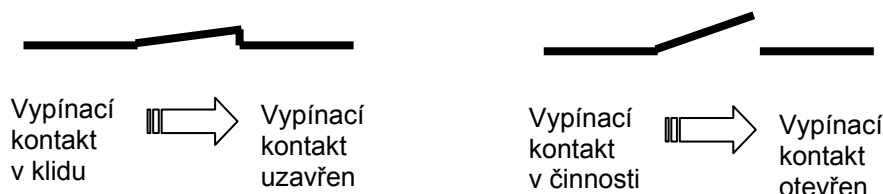
U signálových čidlech se rozlišuje mezi **rozpínacími kontakty** a **spínacími kontakty**.



Zde znázorněný spínač je **spínací kontakt**, tzn. že je právě tehdy sepnutý, pokud je v činnosti.



Zde znázorněný spínač je **rozpínací kontakt**, tzn. že je právě tehdy sepnutý, pokud je v klidu.



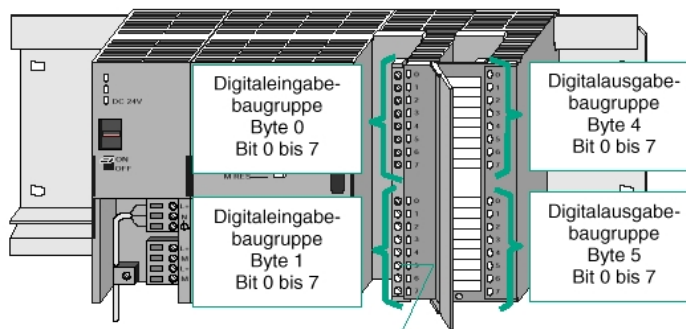
#### 5.5 Jak komunikuje PLC s jednotlivými vstupními a výstupními signály?



Údaj jistého vstupu nebo výstupu uvnitř programu se označuje jako zadání adresy.

Vstupy a výstupy PLC jsou většinou pojmuty v 8 početních skupinách na digitálních vstupních nebo digitálních výstupních konstrukčních celcích. Tyto 8-početné jednotky se označuje jako **byte**. Každá tato skupina obdrží číslo, takzvanou **byte adresu**.

Aby bylo možné oslovit vstup nebo výstup uvnitř jednoho byte-u, je každý byte rozdělený na 8 jednotlivých **bitů**. Tyhle jsou očíslovány od bit 0 do bit 7. Tak získáme **bit adresu**. Zde znázorněné PLC má vstupní byte 0 a 1 jako i výstupní byte 4 a 5.



Když např. chceme oslovit pátý vstup shora, zadáme následující adresu:

I 0.4

I označuje typ adresy jako vstup, 0 je byte-adresa a 4 je bit-adresa. Byte-adresa a bit-adresa jsou vždy odděleny tečkou.



**Poznámka:** Pro bit-adresu stojí zde 4 pro pátý vstup, protože počítat se začíná od 0. První vstup nahore má tedy adresu I 0.0!!

Jestliže chceme např. oslovit nejspodnější výstup, zadáme následující adresu:

Q 5.7

Q označuje typ adresy jako výstup, 5 je byte-adresa a 7 je bit-adresa. Byte-adresa a bit-adresa jsou vždy odděleny tečkou.



**Poznámka:** Pro bit-adresu zde stojí 7 pro nejspodnější (osmý) výstup, protože počítat se začíná shora od 0. První výstup má tedy adresu Q 5.0!!

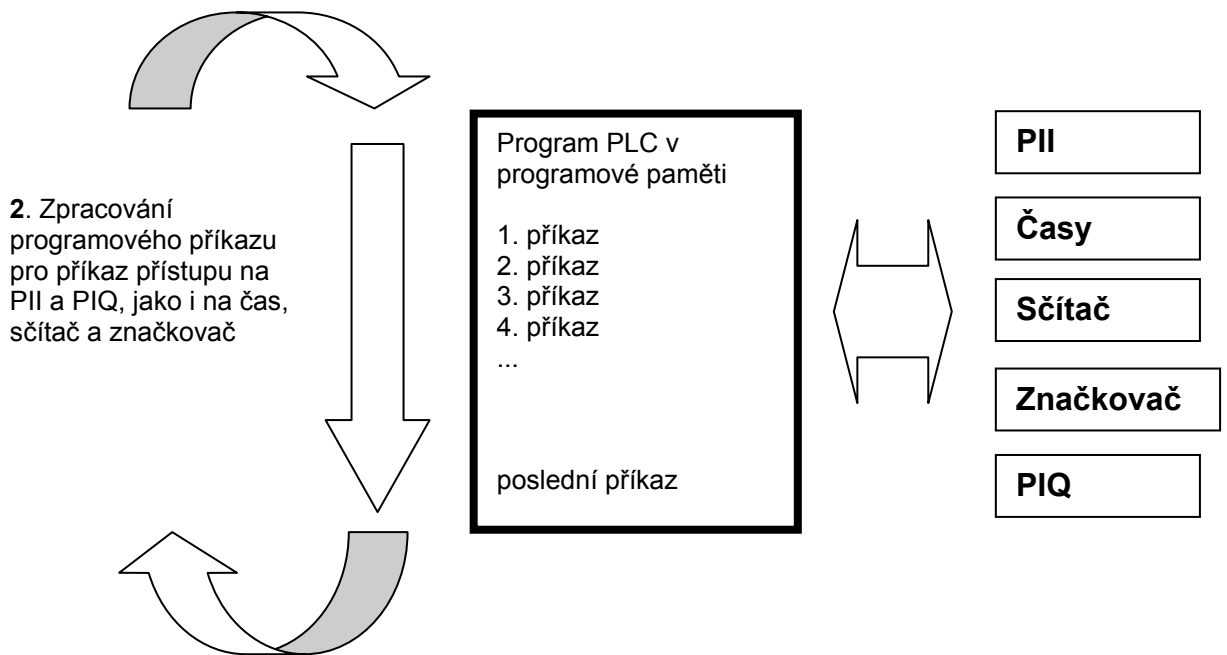
## 5.6 Jak se v PLC zpracovává program?



Zpracování programu v PLC probíhá cyklicky v následovném průběhu:

- Po té co se PLC uvede do provozu, ptá se **procesor**, který je mozkiem PLC, jestli na jednotlivých vstupech je napětí či ne. Tento status pro vstupy se uloží do paměti jako zobrazení procesu vstupů (**PII**). je zde uložen signální stav všech vstupů. Přitom se vstupům, které vedou napětí zadá informace 1 nebo „High“, a pro vstupy, které nevedou napětí informaci 0 nebo „Low“.
- Procesor pak zpracuje program, který byl uložen do programové paměti. Tento se skládá ze seznamu logických vazeb a příkazů, které se zpracovávají jeden po druhém. Přitom se pro potřebné vstupní informace sáhne na již načítané PII a výsledky vazeb se zapíše do takzvaného zobrazení procesu výstupů (**PIQ**). Z procesoru se sáhne během zpracování programu, jestli je nutno, i na jiné oblasti paměti pro sčítač, čas a značkovač. V třetím kroku, po zpracování uživatelského programu se přenesou status z PIQ na výstupy a tyto se zapnou nebo vypnou. Pak se postupuje jako v bodu 1.

1. Status vstupů uložit do PII (= tabulka vstupů). Nachází se v paměťové oblasti CPU.



2. Zpracování programového příkazu pro příkaz přístupu na PII a PIQ, jako i na čas, sčítač a značkovač

3. Status z PIQ (= tabulky výstupů) přenést na výstupy. Tabulka obsahuje hodnoty výstupů, vyplývající z průběhu programu



**Poznámka:** Čas, který potřebuje procesor pro tento průběh (tj. od prvního k poslednímu příkazu) se nazývá cyklus procesoru. tento čas je odvislý od počtu příkazů.

## 5.7 Jak vypadají logické vazby v PLC-programu?



Logické vazby se používají na to, aby bylo možné stanovit podmínky pro spínání vstupu. V PLC - programu mohou tyto být sestavené v programovacích jazycích: kontaktní plán (**LAD**), funkční plán (**FBD**) nebo seznam příkazů (**STL**).

Kvůli názornosti provedeme vysvětlení v **LAD**.

Existuje velký počet různých logických vazeb, které mohou být použity v PLC programech.

Operace **AND** jako i **OR** a **NOT** pro kombinace vstupů se však používají nejčastěji a proto jsou zde objasněny na základě příkladů.

**Poznámka:** Informace pro jiné logické vazby je možné rychle a přehledně získat přes Online-pomoc (funkce Help v programu).

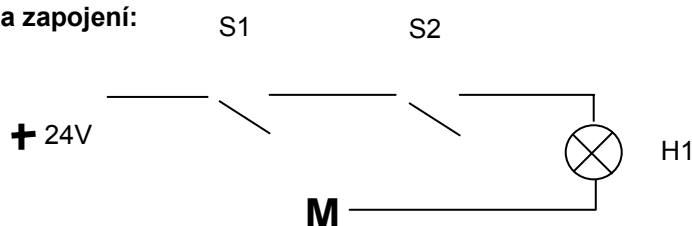
### 5.7.1 Operace AND



**Příklad operace AND:**

Žárovka má svítit, když jsou současně dva spínače jako spínací kontakt v činnosti.

**Schéma zapojení:**



### Vysvětlení:

Žárovka svítí, když jsou oba spínače v činnosti.

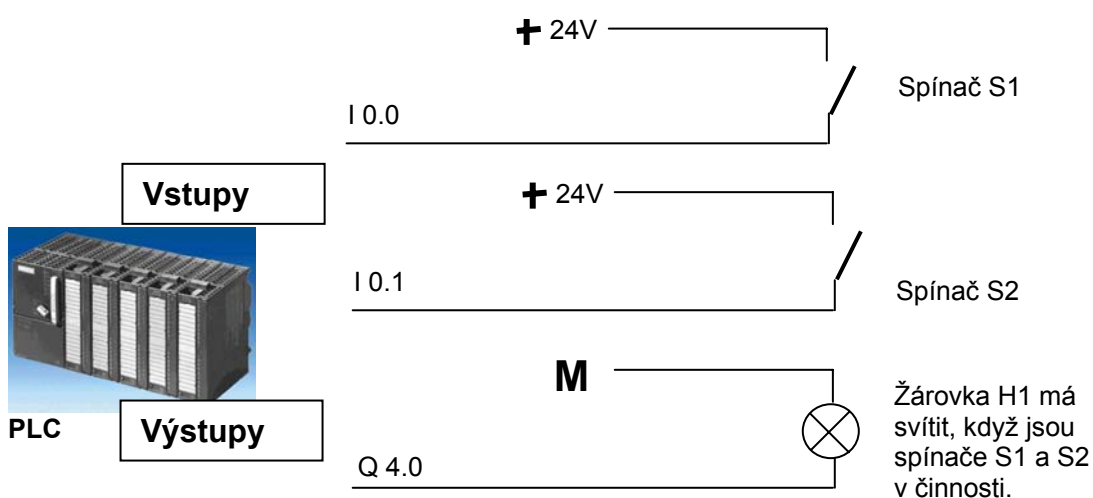
Když jsou tedy S1 a S2 v činnosti, svítí žárovka H1.



### Propojení v PLC:

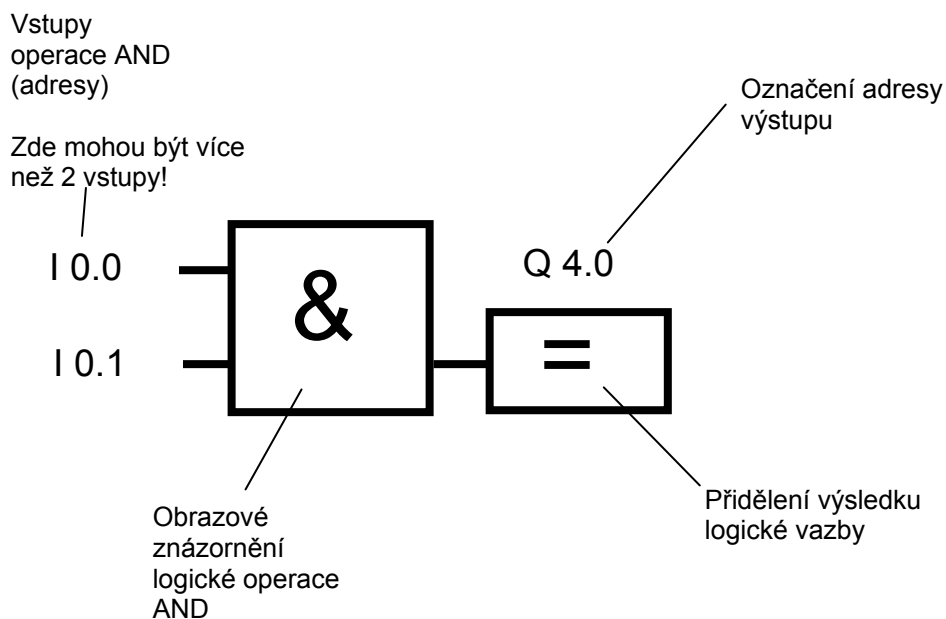
Abychom mohli realizovat logiku v PLC programu, musí být samozřejmě oba spínače připojeny na vstupy PLC. Spínač S1 je přepojen na vstup I 0.0 a S2 na vstup I 0.1.

Kromě toho je žárovka H1 připojena na výstup ,např. Q 4.0.



### Operace AND ve funkčním plánu FBD:

Ve funkčním plánu FBD se programuje operace AND pomocí obrazového znázornění a vypadá následovně:



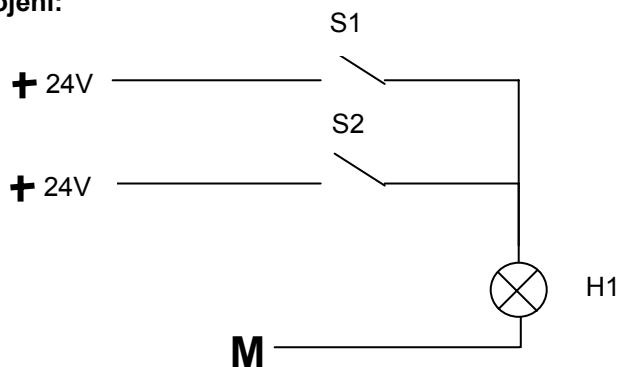
## 5.7.2 Operace OR



### Příklad operace OR (funkce NEBO):

Žárovka má svítit, když jeden nebo oba ze dvou spínačů jsou v činnosti jako spínací kontakt.

#### Schéma zapojení:



#### Vysvětlení:

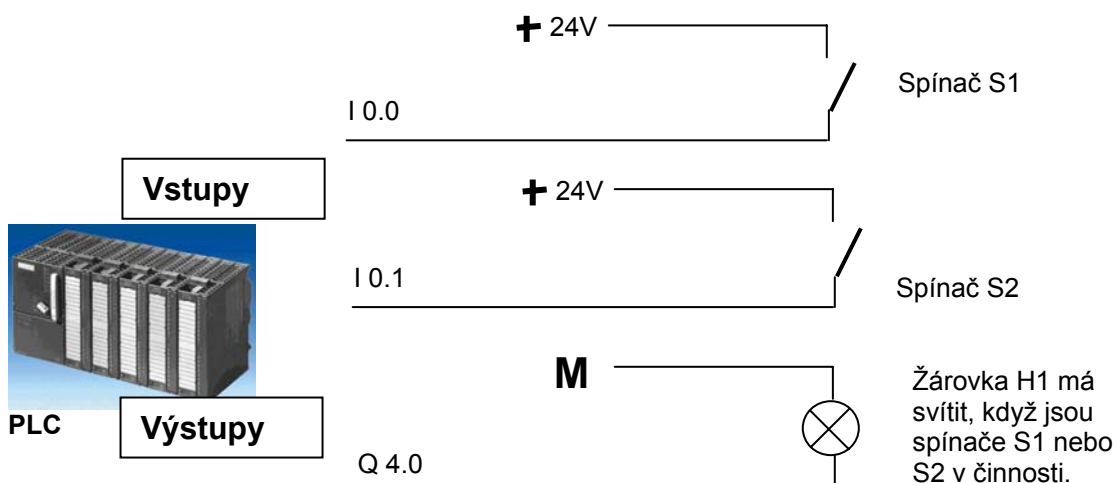
Žárovka svítí právě tehdy, když je jeden nebo oba spínače v činnosti.

Když je spínač S1 **nebo** S2 v činnosti svítí žárovka H1.

#### Propojení PLC:

Abychom mohli realizovat logiku v PLC-programu, musí být samozřejmě oba spínače připojeny na vstupy PLC. Zde je spínač S1 přepojen na vstup I 0.0 a spínač S2 na vstup I 0.1.

Kromě toho musí být žárovka H1 připojena na výstup, např. Q 4.0.



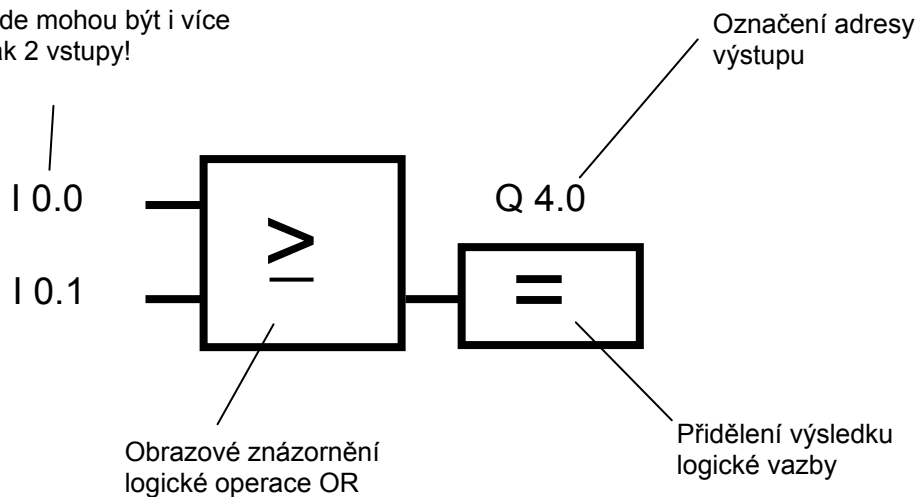


### Operace OR ve funkčním plánu FBD:

Ve funkčním plánu FBD se operace OR programuje pomocí obrazového znázornění a výsledek vypadá následovně:

Vstupy operace OR.

Zde mohou být i více jak 2 vstupy!



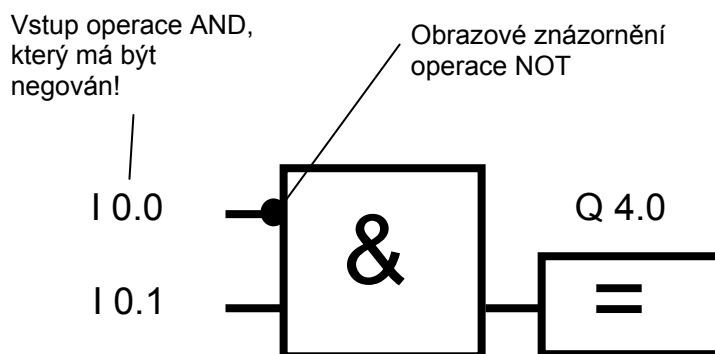
### 5.7.3 Operace NOT



V logických vazbách je často nutné se tázat, jestli některý ze **spínacích** nebo **vypínacích kontaktů není v činnosti** a tedy na daném vstupu se nenachází napětí.

To se dá zjistit programováním **operace NOT** na vstupu operace AND příp. OR.

Ve funkčním plánu FBD se naprogramuje operace NOT na vstupu operace AND pomocí následovního obrazového znázornění:

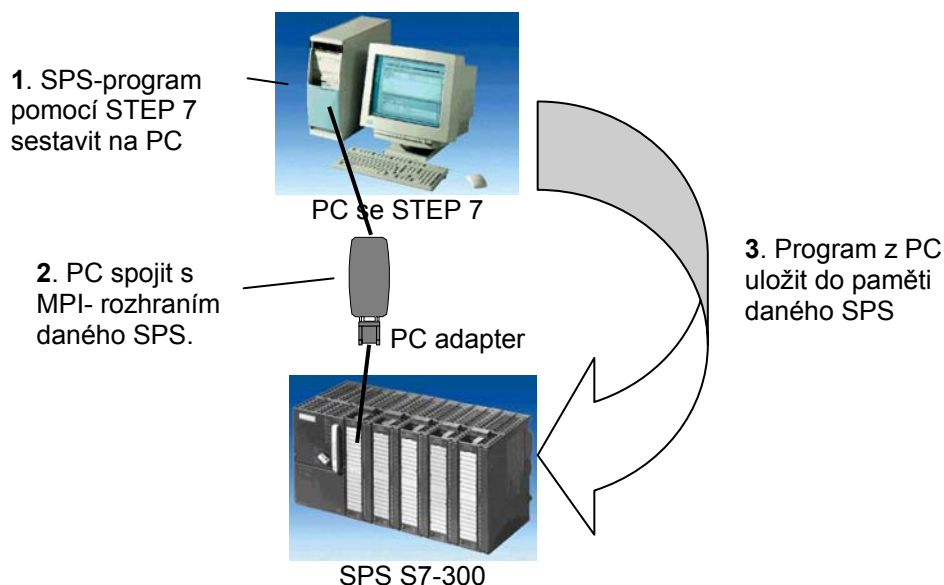


Na výstupu Q 4.0 se nachází napětí právě tehdy, když I 0.0 nespíná a I 0.1 spíná.

## 5.8 Jak se sestaví PLC program? Jak se dostane do paměti PLC?



PLC program se sestaví na PC s pomocí softwaru STEP 7 a tam se dočasně uloží. Jakmile je PC spojeno s MPI rozhraním daného PLC, je možné program pomocí zaváděcí funkce uložit do paměti PLC.



**Poznámka:** Přesný postup je uvedený krok za krokem v kapitolách 8 až 10.

## 6. Konstrukce a obsluha SIMATIC S7-300



### Spektrum konstrukčních celků:

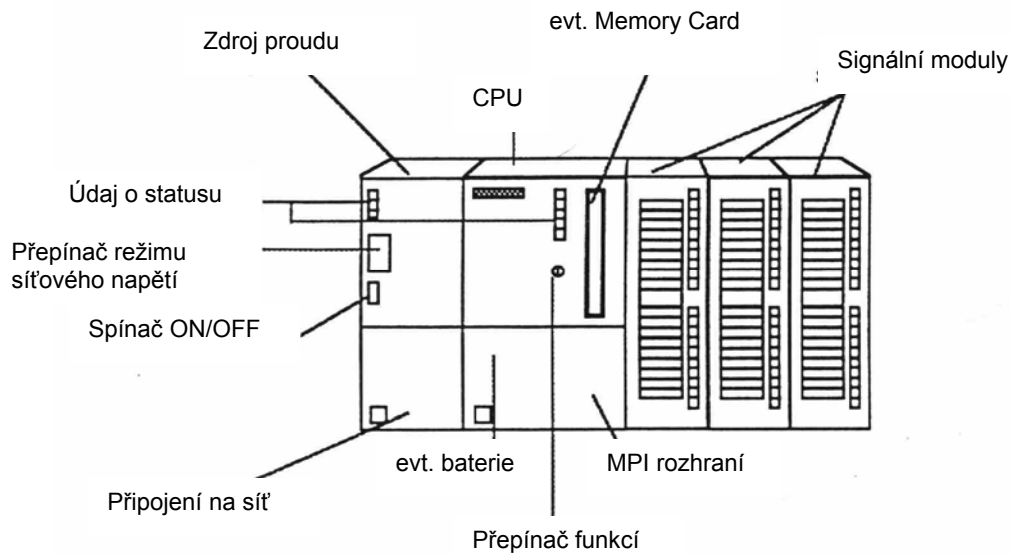
SIMATIC S7-300 je modulární automatizační systém, který nabízí následovné spektrum konstrukčních celků:

- Centrální konstrukční celky (CPU) rozdílných výkonů, částečně s integrovanými vstupy a výstupy (např. CPU 313C, 314C), některé s integrovaným rozhraním PROFIBUS (např. CPU 313C-2DP, 315C-2DP)
- Napájecí zdroj 24V s 2A, 5A nebo 10A
- Konstrukční celky (interfaceové moduly) IM pro rozšíření řídicího systému v případě víceřadé sestavy SIMATIC S7-300. vytvářejí propojení od jedné řady k následující
- Signální celky (moduly) SM s digitálními a analogovými vstupy a výstupy
- Funkčné celky (moduly) FM pro speciální funkce (např. řízení krokových motorů, počítání)
- Komunikační procesory CP pro přepojení na síť pomocí:
  - propojení typu Point-to-point
  - Profibus
  - Průmyslový Ethernet





## Základní sestava PLC:

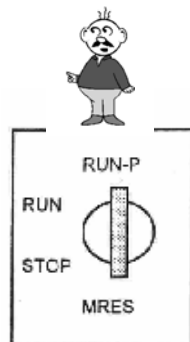


### MPI-rozhraní:

Na každé CPU se nachází MPI-rozhraní pro připojení programovacího kabelu od PC (např. přes PC-adapter, či z MPI karty CP5611 umístěné přímo v PC). MPI rozhraní se nachází za krytkou na přední straně CPU.

### Přepínač funkcí:

Každá CPU vlastní přepínač funkcí pro přepojení podle druhu provozu. Tento je většinou proveden ve formě přepínače. Možné jsou následovné druhy provozu/funkcí:



- RUN-P: Pracuje se na programu, všechny funkce programování CPU jsou povoleny
- RUN: Běh nahreného programu, možné jsou jen načtené funkce, CPU nelze programovat
- STOP: Program se neprovádí, všechny funkce programování CPU jsou povoleny
- MRES: V této poloze je možné provést totální vymazání dat, jako je dále popsáno



**Poznámka:** U jednotek CPU série CPU31xC je tento spínač ve formě kolébkového přepínače. Druh provozu RUN se souborem dat jen pro čtení, zde chybí. Zde jsou povoleny v RUN i funkce pro psaní.



### Totální vymazání dat:

Totální vymazání dat (funkce MRES) vymaže všechna provozní data na centrální jednotce CPU, avšak ne na paměťové kartě (Micro Memory Card). Totální vymazání dat by se mělo provést jednorázově před programováním.

To se uskuteční ve 3 krocích:

Krok	Provedění	Výsledek
1	Nastavte provozní spínač do polohy <b>STOP</b> .	Údaj STOP svítí.
2	Nastavte provozní spínač do polohy <b>MRES</b> a držte ho v této poloze (ca. 3 sekundy) až se rozsvítí údaj STOP.	Údaj STOP zhasne a po ca. 3 sekundách se opět rozsvítí. U nových CPU čkejte až se STOP znovu rozsvítí. <b>Důležité:</b> Mezi krokem 2 a 3 smí proběhnout max. 3 sekundy.
3	Nastavte provozní spínač zpátky do polohy <b>STOP</b> a následně v průběhu 2 sekund opět do polohy <b>MRES</b> .	Údaj STOP bliká ca. 3 sekundy a pak se opět rozsvítí: všechno je o.k.; <b>CPU je vymazána.</b>

## 7. Vzorová úloha



Pro náš první STEP 7-program máme vyřešit jednoduchou úlohu.

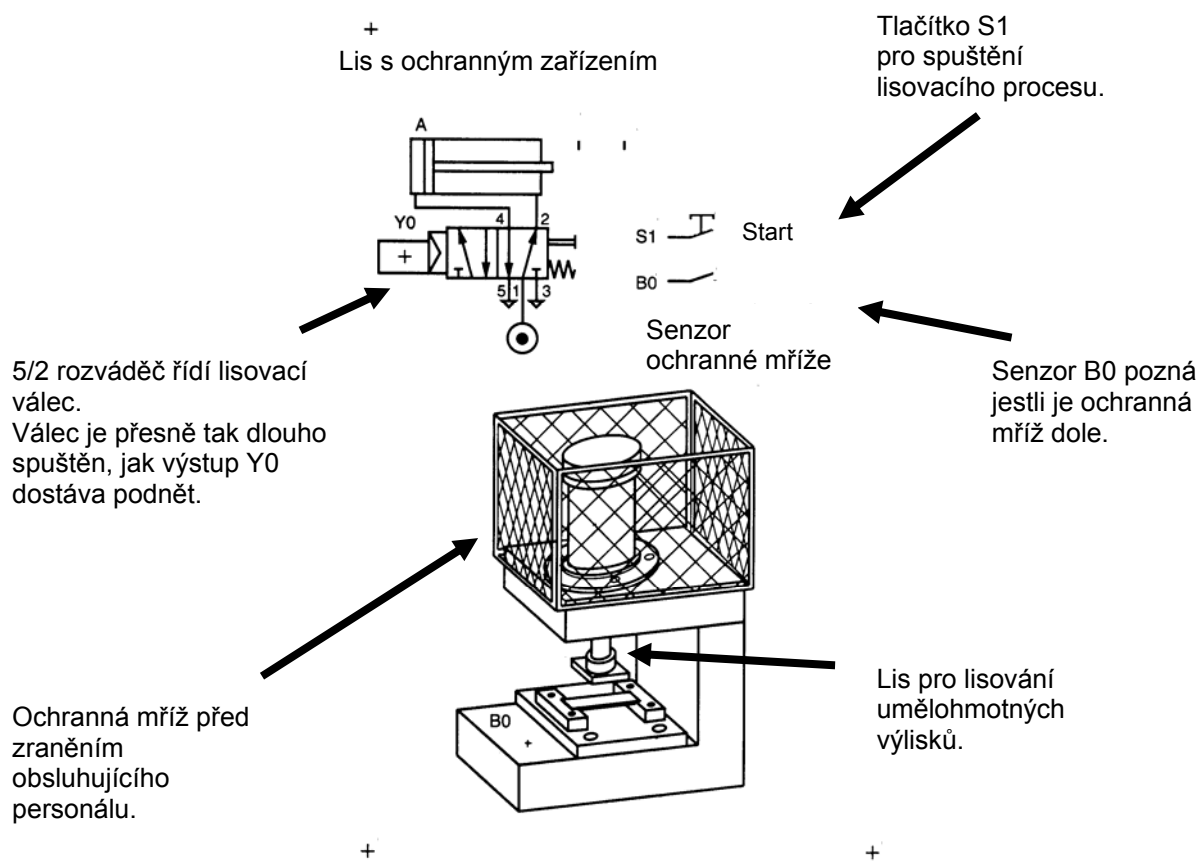
Lis s ochranným zařízením se dá spustit pomocí tlačítka START S1 jenom poté, když je ochranné zařízení uzavřeno. Tento stav je kontrolován přes senzor BO na ochranné mříži.

Jestli je mříž uzavřena, 5/2 rozváděč Y0 pro lisovací válec dostane podnět na 10 sekund, aby mohla být vylisována umělohmotná forma.

Z bezpečnostních důvodů se musí lis opět pohnout nahoru, i když startovní tlačítko S1 není ovládáno nebo senzor B0 ochranné mříže nereaguje.

### Seznam přiřadění:

Adresa	Symbol	Pojmenování
I 0.0	B0	Senzor ochranní mříže
I 0.1	S1	Start- tlačítko
Q 4.0	Y0	5/2 rozváděč pro lisovací válec



## 8. Složení STEP 7- projektu



'SIMATIC Manager' spravuje data v programu STEP 7. Zde mohou být kopírovány např. programové celky nebo kliknutím na myš vyvolány jiné nástroje pro další zpracování. Obsluha zodpovídá běžnému standardu ve WINDOWS. (Tak můžeme obdržet např. možnost výběru každé složky z menu, kliknutím na pravou klávesnici myši).  
V registrech ‚SIMATIC 300 Station‘ a ‚CPU‘ je znázorněná struktura hardwaru PLC. Proto je takový projekt vždy možné vidět v daném hardwaru.

Ve STEP 7 má tento projekt pevně stanovenou strukturu. Programy jsou uloženy v následujících seznámech:

### Projekt:

Zde je zapsán název projektu. Každý projekt představuje databázi, v níž jsou uložena všechna data, příslušící danému projektu

### Stanice SIMATIC 300:

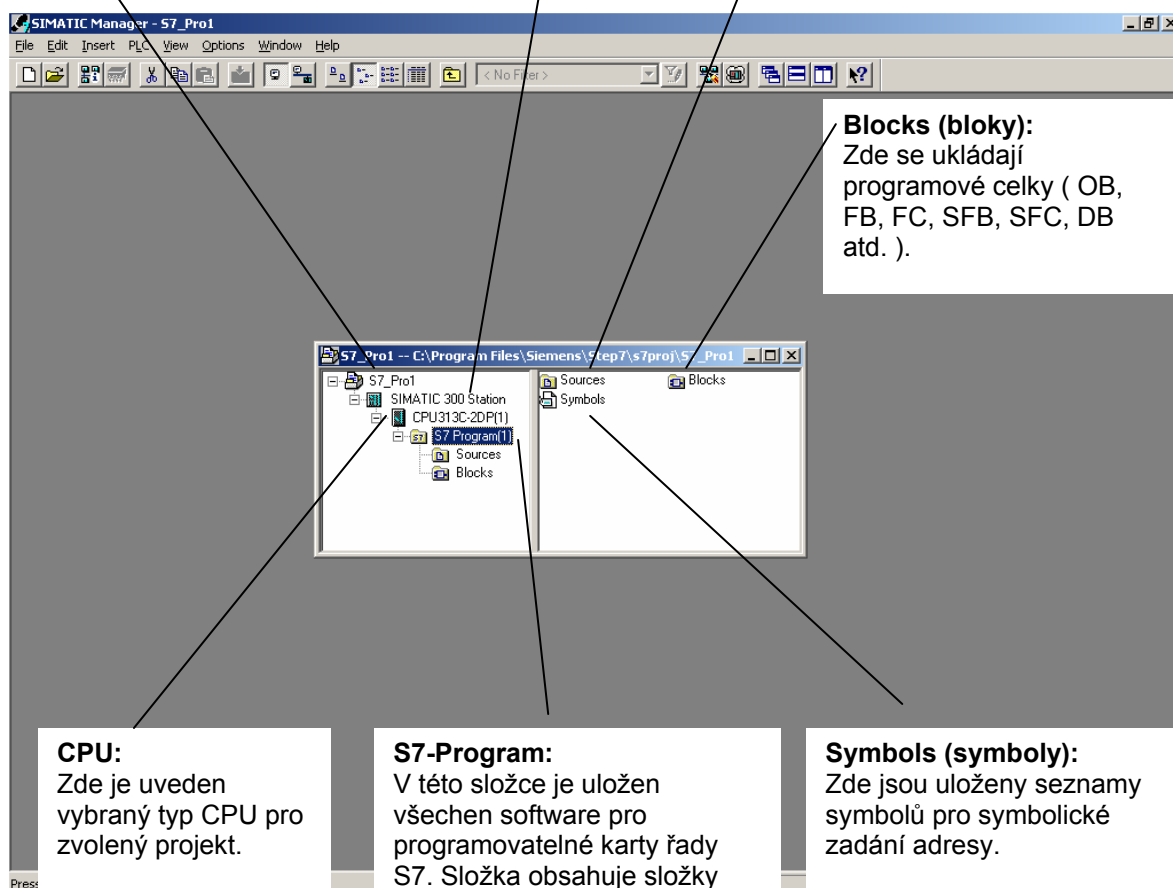
Zde jsou uloženy příslušné informace o údajích hardwarové konfigurace a o parametrech nastavení modulů

### Sources (zdroje):

Zde se ukládají zdrojové soubory programu.

### Blocks (bloky):

Zde se ukládají programové celky ( OB, FB, FC, SFB, SFC, DB atd. ).



### CPU:

Zde je uveden vybraný typ CPU pro zvolený projekt.

### S7-Program:

V této složce je uložen všechny software pro programovatelné karty řady S7. Složka obsahuje složky pro bloky a zdrojové soubory programu

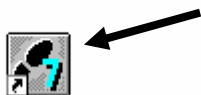
### Symbols (symboly):

Zde jsou uloženy seznamy symbolů pro symbolické zadání adresy.



Uživatel musí provést následující kroky, aby mohl sestavit projekt, ve kterém pak může napsat program, jako řešení konkrétního úkolu.

1. Centrálním nástrojem ve STEP 7 je **'SIMATIC Manager'**, který vyvoláme dvojitým kliknutím.  
( → SIMATIC Manager)

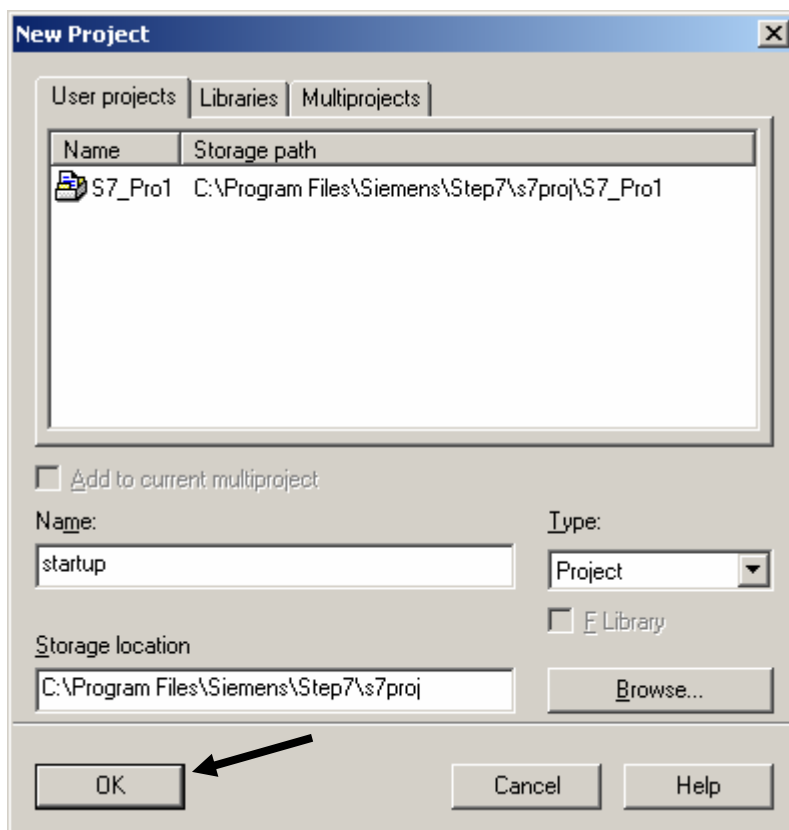


SIMATIC Manager

2. STEP 7- programy se spravují v projektech. Nyní založíme nový projekt. V roletovém menu vybereme nabídku → File → New



3. Objeví se následující okno. Projektu se zadáme jméno do okna **'Name:'**. Projekt jsme pojmenovali např. **'startup'**. Poté potvrdíme tlačítkem OK. ( → startup → OK)



4. V tomto projektu **„startup“** provedeme vložení programu do tohoto projektu. Z nabídky Insert vybereme položku Program a dále položku **„S7-Program“**. ( → Insert → Program → S7-Program)

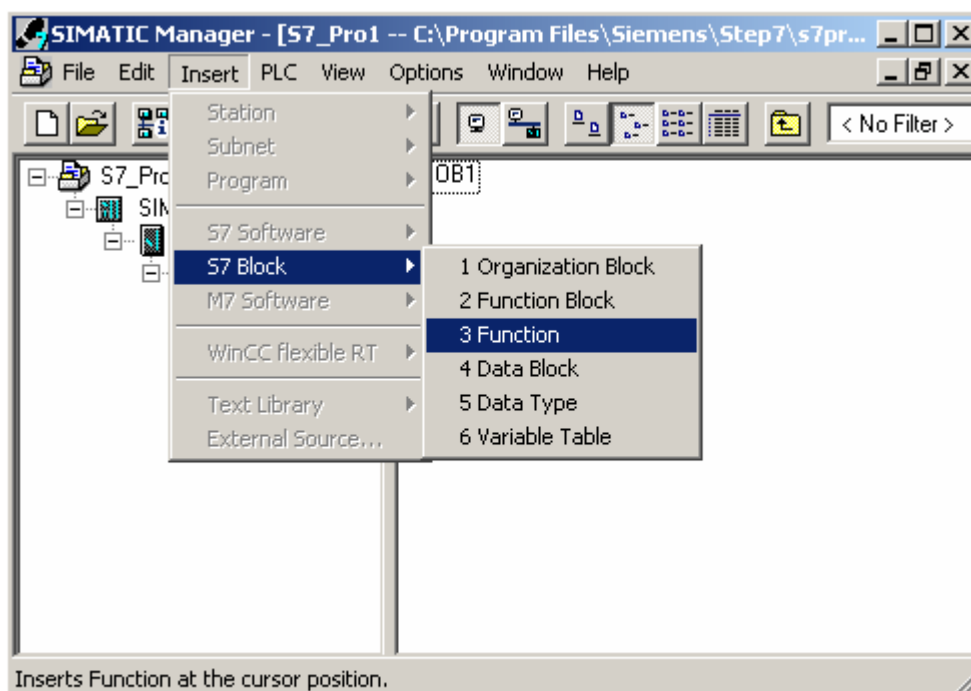
## Bloky programu



Průběh programu je ve STEP 7 psaný v takzvaných blocích. Seznam jednotlivých typů bloků zobrazíte přes nabídku Insert – S7Block. Existují tři základní typy bloků:

- Organizační bloky (OB)
- Funkce (FC) a funkční bloky (FB)
- Datové bloky (DB)

Nový blok vložíme do programu z nabídky **„Insert“** – **„S7 Block“**. Zde máme k dispozici možnost vložit Organizační blok (OB), Funkční blok (FB), Funkci (FC), Datový blok (DB)...



## Organizační bloky

Organizační bloky (OB) jsou rozhraním mezi operačním systémem CPU a uživatelským programem. OB1 je základním organizačním blokem, který je obsažen v každém programu. OB1 obsahuje buď samotný cyklický program, nebo jej organizuje (voláním jiných bloků). Organizační bloky nemohou být volány jinými bloky. Jsou volány operačním systémem v reakci na jisté události, např.:

- při spuštění CPU
- v určenou dobu dne
- v pevných intervalech
- při uplynutí nastavené časové periody
- při výskytu chyby
- při výskytu hardwarového přerušení

OB jsou prováděny v pořadí priorit, které jim jsou přiřazeny. Když operační systém volá jiný OB, cyklické provádění programu je přerušeno, protože OB1 má nejnižší prioritu. Kterýkoli jiný OB tedy může přerušit hlavní program a provést svůj vlastní program. Poté pokračuje provádění OB1 v bodě přerušení.

Pokud je volán OB s vyšší prioritou než ten, který je právě prováděn, dojde k přerušení prováděného bloku hned po dokončení aktuálního kroku (Networku). Operační systém uloží celý zásobník registru pro přerušovaný blok. Informace registru jsou obnoveny v tom okamžiku, kdy operační systém obnoví provádění přerušovaného bloku.

**Spuštění** – spouštěcí program je proveden před cyklickým provedením programu po obnoveném spuštění nebo po změně operačního režimu (přepínačem režimů na CPU). Spuštění se děje pomocí OB100 až OB102. (Simatic S7-300 má vždy pouze OB100, ostatní dva jsou využívány Simaticem S7-400). V bloku OB100 můžeme např. předvolit daná komunikační spojení.

**Cyklické provedení programu** – program, který se má souvisle provádět je uložen v organizačním bloku OB1. Po tom, co je program zcela proveden v OB1, nový cyklus začne aktualizací tabulky vstupů a výstupů a provedením prvního kroku (Networku) v OB1. Čas snímání cyklu a čas odezvy systému je výsledkem těchto operací. Čas odezvy je součet prováděcího času operačního systému CPU a času potřebného pro provedení celého uživatelského programu. Čas odezvy, tedy jak rychle může být výstup přepnut v závislosti na vstupním signálu, se rovná dvojnásobku času snímání cyklu.

**Periodické provedení programu** – s periodickým prováděním programu můžete přerušit cyklické provádění programu v pevných intervalech. U cyklických přerušování jsou organizační bloky OB30 až OB37 prováděny po vypršení předem nastaveného časovacího kódu, např. každých 100ms. Pravidelně po uplynutí 100ms se potom bude provádět např. OB30. U přerušování typu „time-of-day“ je OB proveden v konkrétní čas, např. každý den v 17:00, aby uložil data.

**Událostmi řízené provedení programu** – hardwarové přerušování může být použito k rychlé reakci na událost v procesu. Po výskytu této události je cyklus okamžitě přerušeno a je proveden přerušovací program. Přerušování „time-delay“ (časová prodleva) reaguje na událost v procesu až po periodě časové prodlevy. Pomocí chybových OB můžeme určit, jak se má systém chovat, např. při selhání záložní baterie.

## **Spouštěcí OB**

**Start** – OB jsou spouštěny výhradně operačním systémem. Existuje několik spouštěcích událostí, které vedou ke spuštění asociovaných OB v odpovídající třídě priority. OB obsahují normální řídicí program a také deklarační tabulku.

**Vlastnosti** – každé provedení programu OB může být přerušeno událostí vyšší priority na rozmezí příkazu. Priority jsou odstupňovány od 0 do 27, kde 0 má prioritu nejnižší a 27 nejvyšší. OB se stejnou prioritou se nemohou navzájem přerušovat, ale jsou spouštěny jeden po druhém v pořadí, v jakém jsou rozlišeny.

**Spouštění** – Simatic S7-300 má spouštění typu „kompletní restart“. ten smaže tabulky vstupů a výstupů, nezálohovací časovače a bitové paměti. Provedení programu v OB1 začne u prvního výrazu. Simatic S7-400 má i spouštění typu „restart“. Všechna data (bitové paměti, časovače, čítače, tabulky vstupů a výstupů) jsou zachována. provádění programu pokračuje od bodu, kde se vyskytlo přerušování. Při kompletní restartu je proveden OB100, u restartu je proveden OB101.

## **Přehled OB pro S7-300**

**OB1** (priorita 1) – provádí cyklický program

**OB10** (priorita 2) – přerušování „time-of-day“ – užívá se pro provádění jistého programu, volaného buď jen jednou v určitém okamžiku, nebo periodicky (jednou za minutu, jednou za hodinu...). Konfiguruje se pomocí nástroje „HW Config“, postupným výběrem menu „CPU“ – „Object Properties“ – „Time-of-Day Interrupts“.

**OB20** (priorita 3) – přerušování „time-delay“ (časová prodleva) – přerušování reaguje na událost v procesu až po periodě časové prodlevy.

**OB35** (priorita 12) – cyklické přerušování – cyklická přerušování se užívají pro provádění bloků v určených pevných intervalech. Výchozí interval pro OB35 je 100ms. Můžeme jej změnit na hodnotu v povoleném rozsahu od 1ms do 1s. Je nutné dbát na to, aby námi určený interval byl delší, než je doba potřebná pro provedení OB.

**OB40** (priorita 16) – hardwarové přerušení – program v tomto OB se provede, jakmile se vyskytne jistá událost. Např. pro modul analogového vstupu jsou nakonfigurovány vhodné limitní hodnoty. Pokud měřená hodnota překročí limit, je volán OB40.

**OB82** (priorita 26/28) – diagnostické přerušení – je dáno chybou ve funkčnosti PLC. Pokud je zjištěno selhání ve stavu RUN a pokud byl naprogramován patřičný chybový OB, pak je tento volán a je proveden v něm obsažený program. Takový program může být např.

- instrukce pro zapnutí sirény
- instrukce pro zálohování dat, následované instrukcí STOP
- program pro zaznamenání frekvence výskytu selhání, aniž by CPU přešlo do režimu STOP

Pokud OB pro konkrétní selhání neexistuje, CPU se automaticky přepne do režimu STOP. Přerušovací OB82 je volán například v následujících situacích:

- přerušení vodiče na diagnostikovaném modulu
- selhání přívodu energie na modul analogového vstupu
- měřící rozsah analogového vstupu byl překročen

## Funkce a funkční bloky

Na rozdíl od funkcí (FC), mají funkční bloky (FB) paměť (připomínací). To znamená, že blok lokálních dat je přiřazen k funkčnímu bloku. Je to tzv. instanční datový blok. Voláte-li FB, musíte také specifikovat číslo instančního datového bloku (DB), který je automaticky otevřen. Instanční DB se užívá k ukládání statických proměnných. Při volání FB jsou hodnoty konkrétního parametru uloženy v instančním DB. Nový instanční blok lze generovat dvěma způsoby:

- když voláte FB, určíte, s kterým instančním DB má FB pracovat. Objeví se hlášení : „Instance data block DB x does not exist. Do you want to generate it?“ (Instanční datový blok DB x neexistuje. Chcete jej vytvořit?)
- když vytvoříte nový datový blok DB, vyberete možnost „Data block referencing a function block“. (datový blok odkazující na funkční blok).

Jeden instanční DB může odkazovat pouze na jeden FB. Jeden FB však může při každém volání přijímat odkazy od jiného instančního DB. U každého FB můžete specifikovat různé konkrétní parametry. Když je FB ukončen, data v něm zůstanou uchována.

Výhody FB:

- pokud píšete program pro FC, musíte hledat prázdné adresové slovo nebo datové oblasti v bitové paměti a musíte je sami udržovat. Statické proměnné FB jsou oproti tomu udržované softwarem STEP7.
- Užitím statických proměnných předejdete riziku, že přiřadíte adresové nebo datové oblasti v bitové paměti dvakrát
- volání bloku FB je jednodušší

## Datové bloky

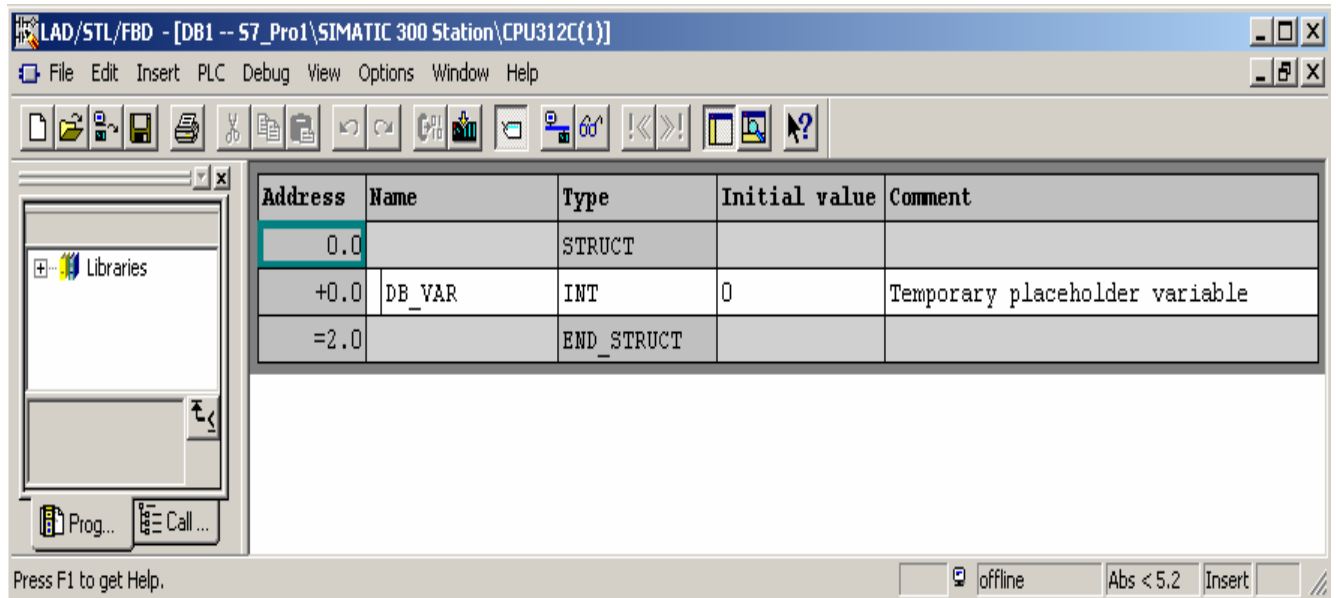
Kromě programových bloků obsahuje uživatelský program také data s informacemi o stavech procesu, signálech atd., které jsou pak zpracovávány podle instrukcí v uživatelském programu. Stejně jako logické bloky (FC, FB, OB) zabírají datové bloky prostor v uživatelské paměti. Datové bloky obsahují proměnná data (např. číselné hodnoty), které používá uživatelský program. Uživatelský program získává přístup k datům v datovém bloku přes bitové operace, bytové operace, slova nebo dvojslova. Datové bloky můžeme používat různými způsoby podle jejich obsahu.



Dělá se rozdíl mezi:

- globální datové bloky – obsahují informace, ke kterým mají přístup všechny logické bloky uživatelského programu
- instanční datové bloky – ty jsou vždy přiřazeny konkrétnímu FB

Jednotlivé datové prvky vkládáme v příslušném DB do tabulky:



Sloupce v tabulce mají následující významy:

*Address* – je zadána Editorem programů. Je to první bytová adresa, kterou zabírá proměnná v datovém bloku

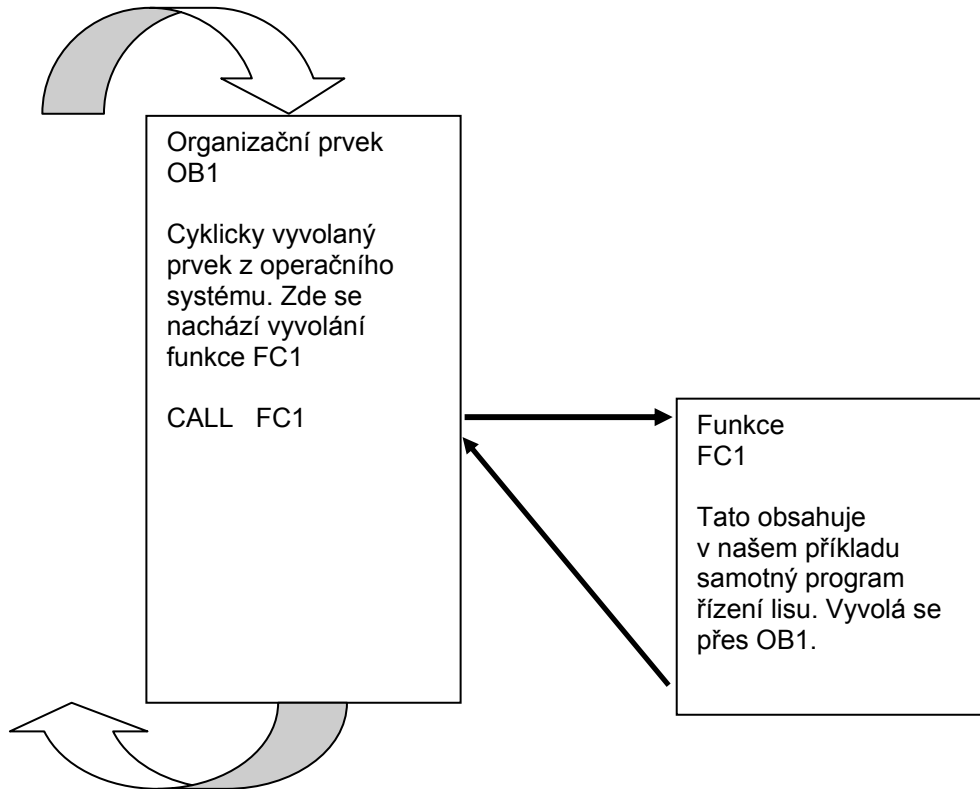
*Name* – symbolický název prvku

*Type* – typ dat (lze jej vybrat kliknutím pravého tlačítka myši)

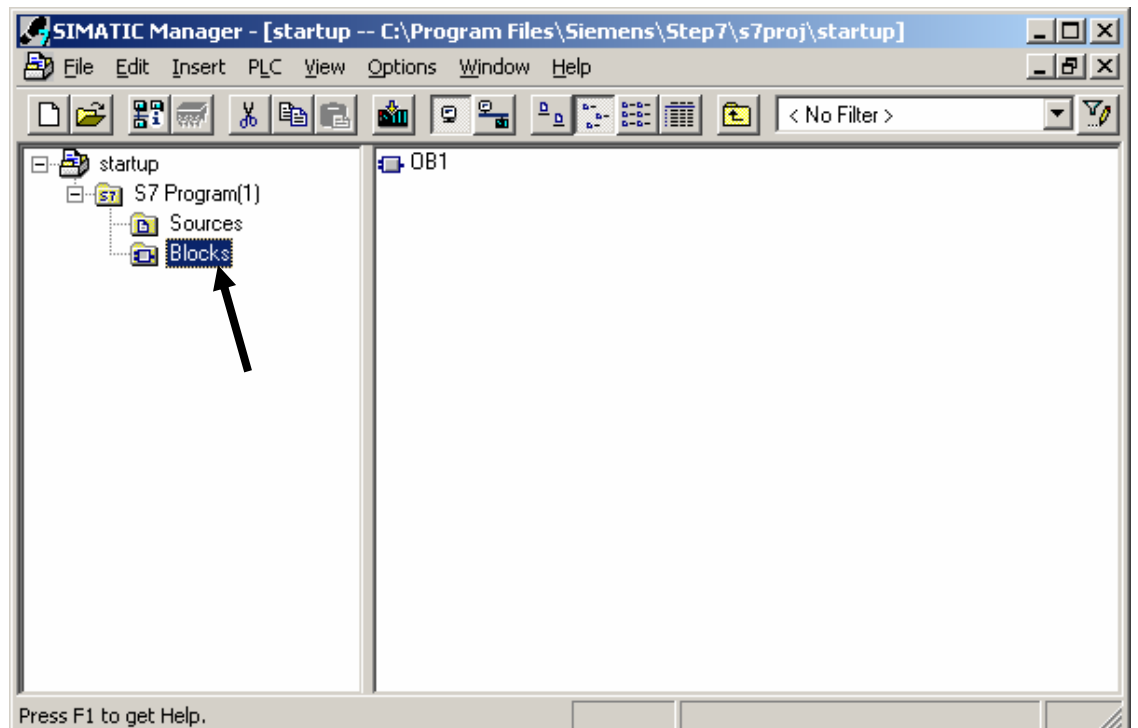
*Initial Value* – nastavení výchozí hodnoty prvku. Pokud nebude zadáno nic, výchozí hodnota je nula.

*Comment* – komentář pro dokumentaci datového prvku (nepovinné)

5. Struktura programu příkladu:

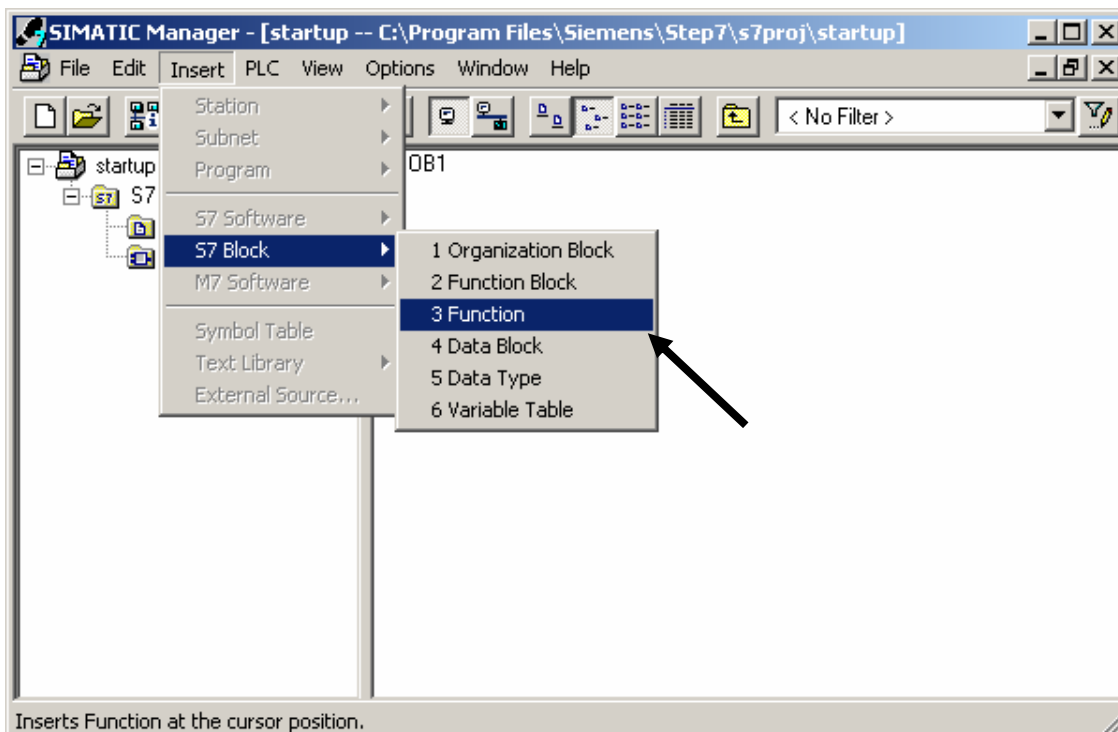


6. Abychom mohli vložit do projektu i další prvek (např. funkci FC1, musíme označit složku 'Blocks'.

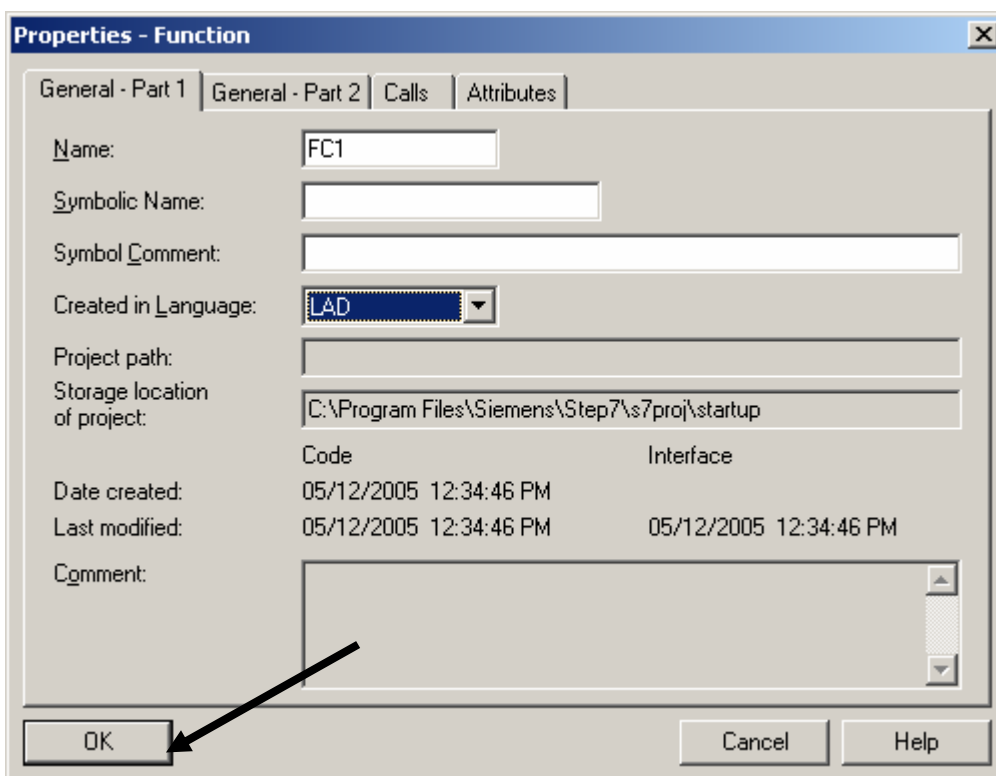




- Nyní vybereme v nabídce Insert položku **'S7 Block'** a dále položku **'Funktion'**. Blok FC1 se nyní přidá do složky Blocks

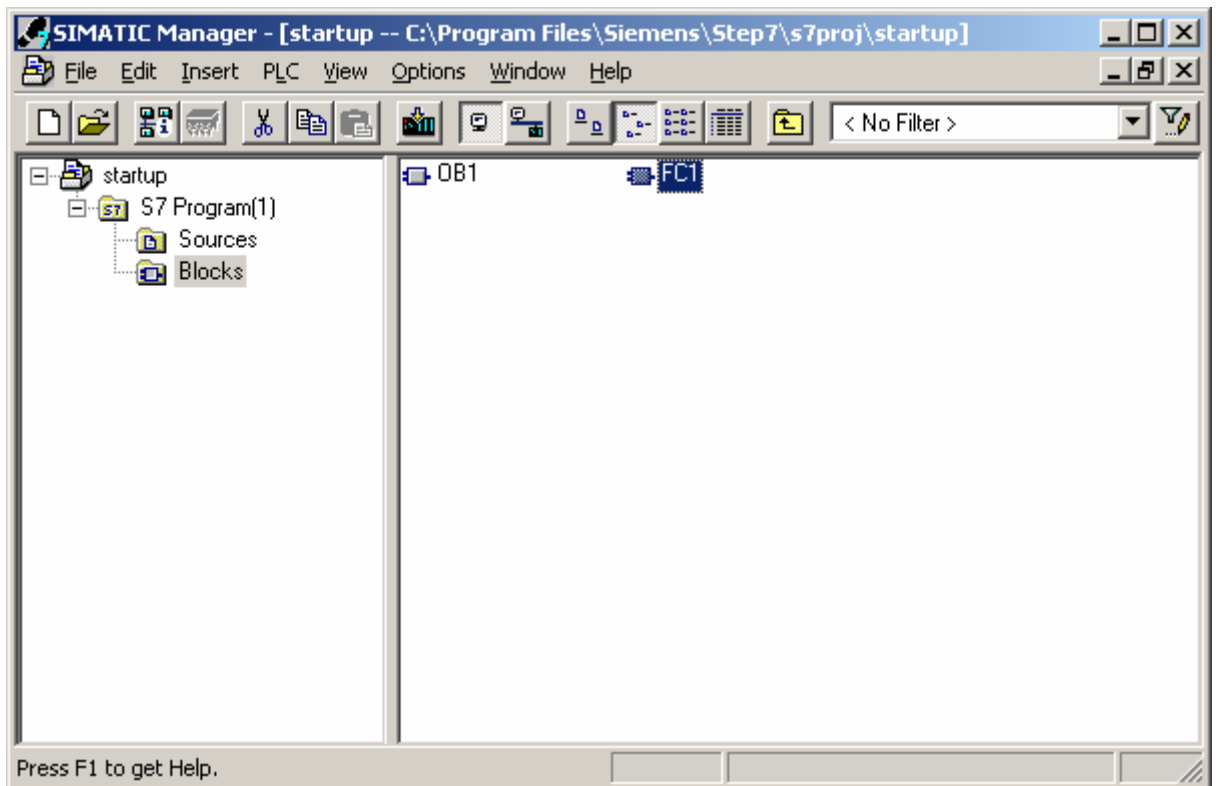


- Nyní se může zvolit jméno pro funkci (zde FC1) a uskutečnit další zadání (např. vybrat způsob zobrazení programu (zde LAD) pro dokumentaci prvků. Nakonec potvrdíme vše tlačítkem OK. ( → FC1 → OK)





8. Ve složce Blocks jsou teď oba prvky OB1 a FC1 k dispozici pro další programování.

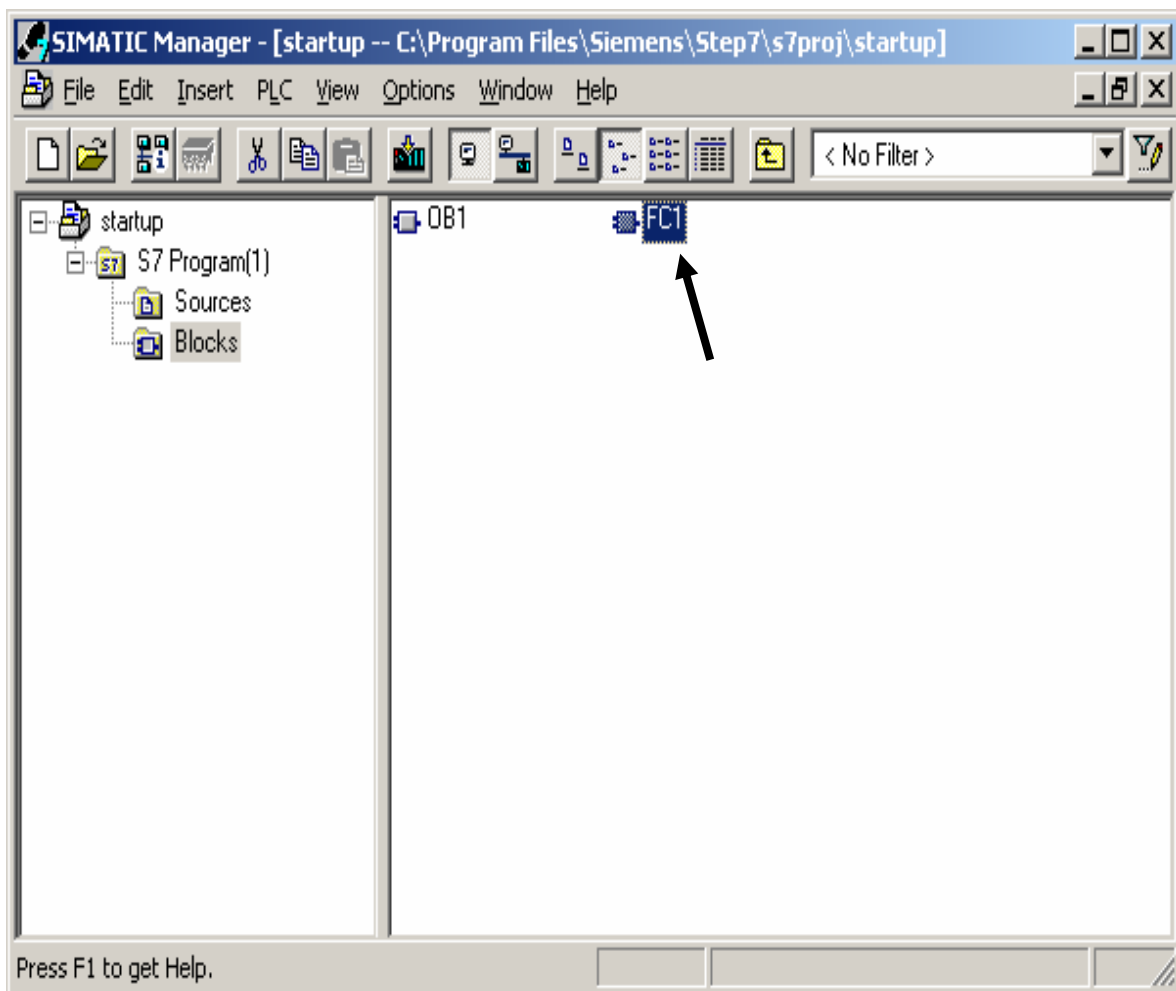


## 9. Napsání STEP 7- programu ve funkčním plánu FBD

Jedna z možností sestavení STEP 7 programu je funkční plán FBD. Je to obrazové znázornění úlohy řízení pomocí symbolů a označení funkcí. Na levé straně symbolu jsou seřazeny vstupy a na pravé straně výstupy.

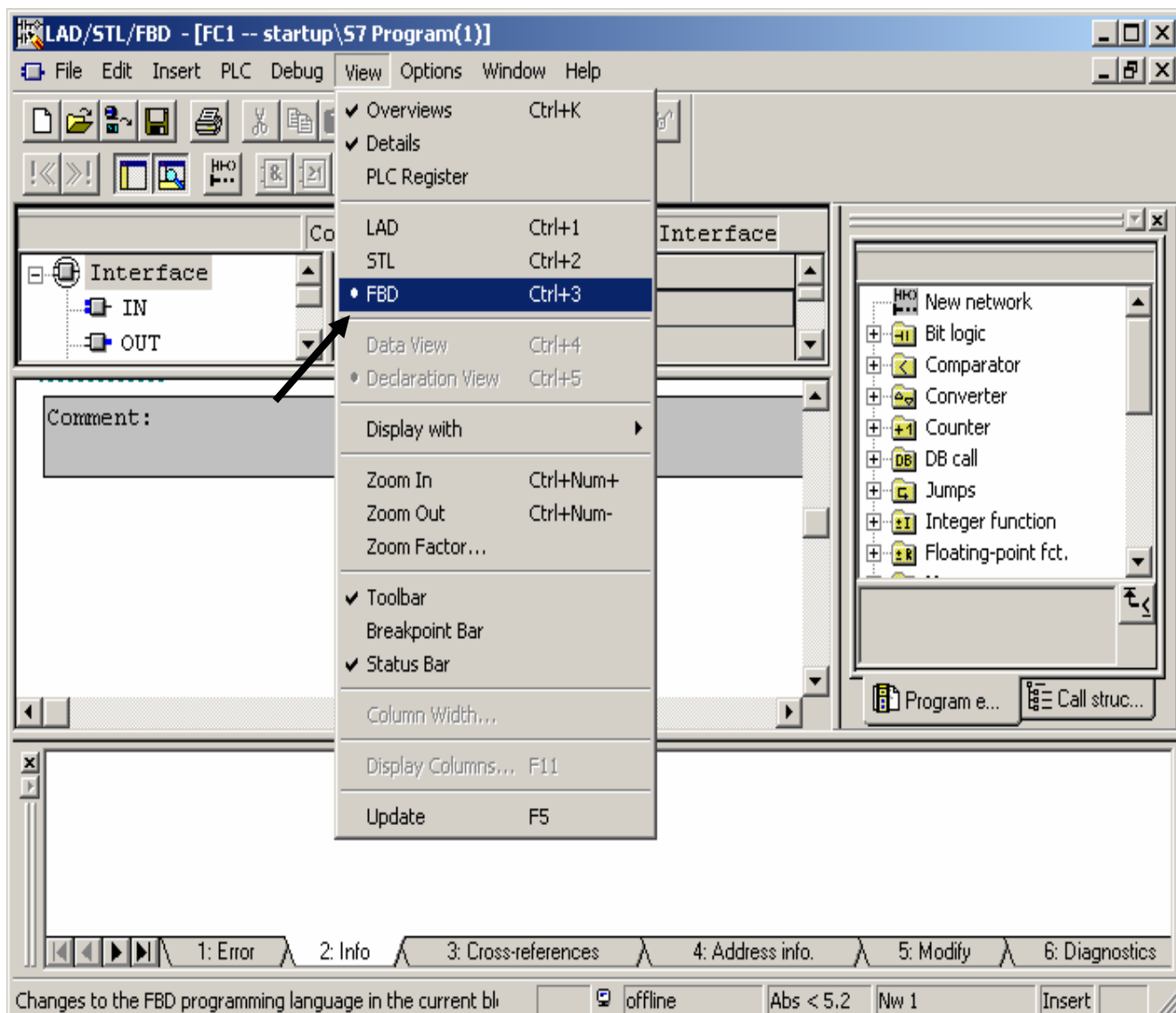


1. Jako první se zpracovává prvek funkce ,FC1'. Tento se v ,SIMATIC Manager' otevře dvojitým kliknutím ( → FC1)





2. V tomto, nyní otevřeném editoru ‚LAD/STL/FBD‘ se přesuneme na ‚View‘ (Náhled) a vybereme pro programovací jazyk znázornění pomocí funkčního plánu ‚FBD‘.





3. Programovací plocha pro programování ve funkčním plánu (FBD) pak vypadá následovně:

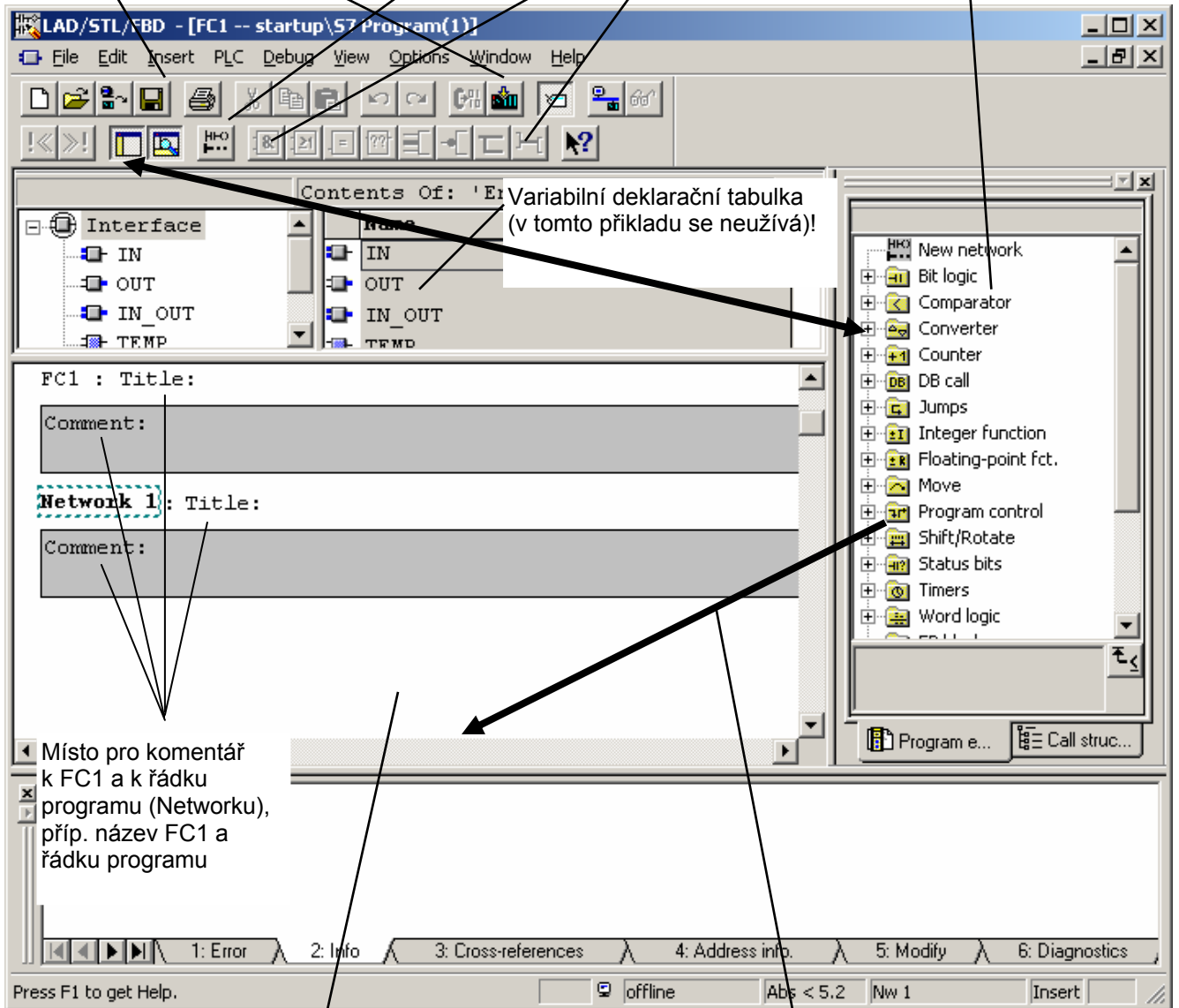
Často používané příkazy jako AND-Box, OR-Box, přidělení, binární vstup negovat, T-odbočka...

Prvek (FC1) uložit do paměti PC

Prvek (FC1) nahrát do CPU

Vložit nový řádek programu (Network)

Katalog všech programových prvků



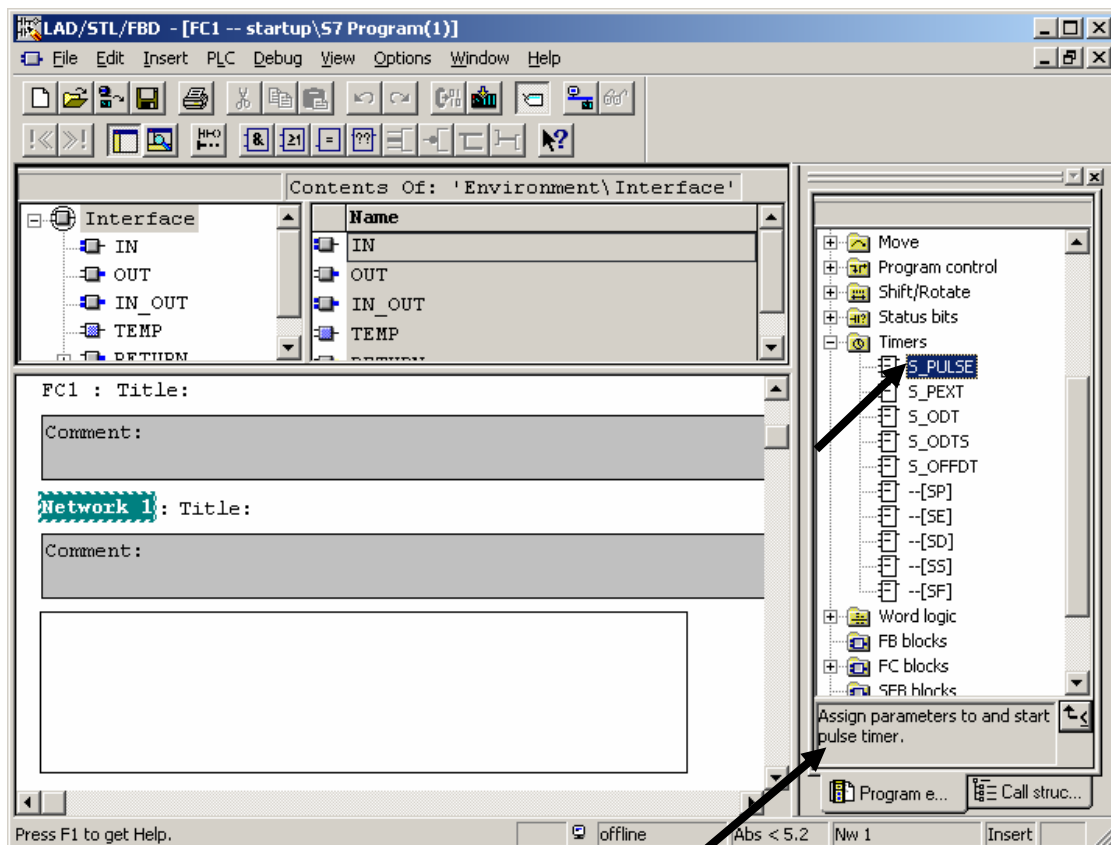
Místo pro komentář k FC1 a k řádku programu (Networku), příp. název FC1 a řádku programu

Zde je místo k sestavení samotného programu (po řádcích)

Programové prvky se dají posunout do programovacího pole způsobem 'DRAG and DROP'.



4. Pro náš příklad použijeme časovač (timer) jako zdroj impulsu. V tabulce prvků ve STEP 7 jej najdeme pod označením **S\_PULSE** ve složce **Timers**.



**Poznámka:** Pod nabídkou funkcí se zobrazí stručný popis vybraného obvodu (viz spodní šipka).





5. Přesný popis každé funkce získáme po stisknutí funkčního tlačítka ‚F1‘ (Online-Help). Popis je velmi podrobný a funkce je též vysvětlena na základě podrobného příkladu.

**S\_PULSE : Assign Pulse Timer Parameters and Start**

**Symbol**

**English**

T no.

S	BI
TV	BCD
R	Q

**German**

T-Nr

S	DUAL
TW	DEZ
R	Q

Parameter English	Parameter German	Data Type	Memory Area	Description
no.	Nr.	TIMER	T	Timer identification number. The range depends on the CPU.
S	S	BOOL	I, Q, M, D, L, T, C	Start input
TV	TW	S5TIME	I, Q, M, D, L or constant	Preset time value (range 0-9999)
R	R	BOOL	I, Q, M, D, L, T, C	Reset input
BI	DUAL	WORD	I, Q, M, D, L	Time remaining (value in integer format)
BCD	DEZ	WORD	I, Q, M, D, L	Time remaining (value in BCD format)
Q	Q	BOOL	I, Q, M, D, L	Status of the timer



**Poznámka:**

Časovač ‚S\_PULSE‘, který je zde použitý, vydá na výstupu ‚Q‘ jednu ‚log.1‘, tak dlouho, pokud na vstupu ‚S‘ je ‚log.1‘.

Když na ‚TV‘ udaný čas uplyne nebo úroveň signálu na vstupu ‚S‘ je ‚0‘, pak je na výstupu ‚Q‘ opět ‚log. 0‘.



6. Nyní vložíme funkci ,S\_PULSE' do obvodu tím, že ji z pravého okna myší přetáhneme do pracovního okna Networku.

The screenshot displays the SIMATIC Manager LAD editor. The main window is titled 'LAD/STL/FBD - [FC1 -- startup\S7 Program(1)]'. The interface includes a menu bar (File, Edit, Insert, PLC, Debug, View, Options, Window, Help) and a toolbar. On the left, a tree view shows the 'Interface' structure with components like IN, OUT, IN\_OUT, TEMP, and RETURN. The central workspace shows 'Network 1' with a ladder logic diagram. A red box highlights the 'S\_PULSE' function block, which has inputs for 'S' (setpoint), 'TV' (time delay), and 'R' (reset), and outputs for 'BI', 'BCD', and 'Q'. A black arrow points from the 'S\_PULSE' block in the right-hand 'Contents Of: Environment\Interface' library to the 'S\_PULSE' block in the network diagram. The library also lists other timer functions like S\_PEXT, S\_ODT, S\_ODTS, S\_OFFDT, and various status bits. The status bar at the bottom indicates 'offline' and 'Abs < 5.2 Nw 1 Insert Chg'.





8. Teď se musí ještě pojmenovat timer jako **T1** a zadat časovou hodnotu na vstupu TV na 10 s ve formátu **S5T#10s**. Kromě toho se zadají vstupy **I 0.0** a **I 0.1** na AND hradle.

The screenshot displays the SIMATIC Manager software interface. The main window shows a ladder logic network (Network 1) with an AND gate connected to inputs I0.0 and I0.1. The output of the AND gate is connected to the S input of a timer block T1 (S\_PULSE). The TV input of the timer is set to S5T#10s. The software interface includes a menu bar (File, Edit, Insert, PLC, Debug, View, Options, Window, Help), a toolbar, and a project tree on the left. A right-hand pane shows a library of PLC blocks, with 'Timers' expanded to show 'S\_PULSE' selected. A tooltip at the bottom right of the library pane reads: 'Assign parameters to and start pulse timer.'




**Poznámka:** Abychom mohli časovači zadat čas, musí se dodržet následující způsob zadávání:

### **S5T# 10s**

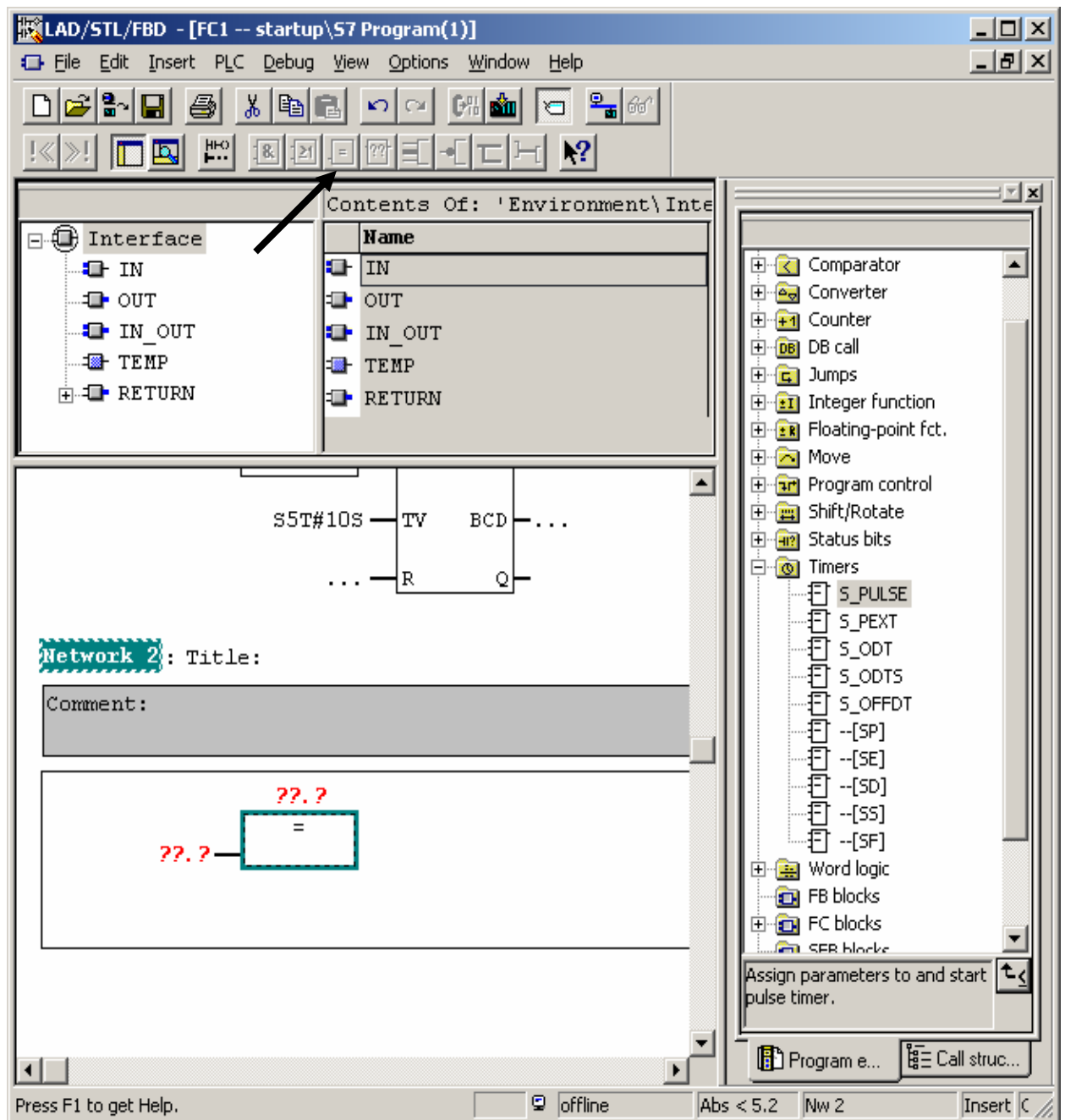
Přitom **S5T#** uvádí formát a údaj za křížkem udává čas (zde **10 sekund**). Zadát se mohou i milisekundy (MS), minuty (M) a hodiny (H). Tyhle časové jednotky je možné zadat i společně (např. S5T#3M\_3S).



Dále otevřeme další krok programu stisknutím tlačítka . Otevře se okno pro Network 2.



Press F1 to get Help. offline Abs < 5.2 Nw 2 Insert C

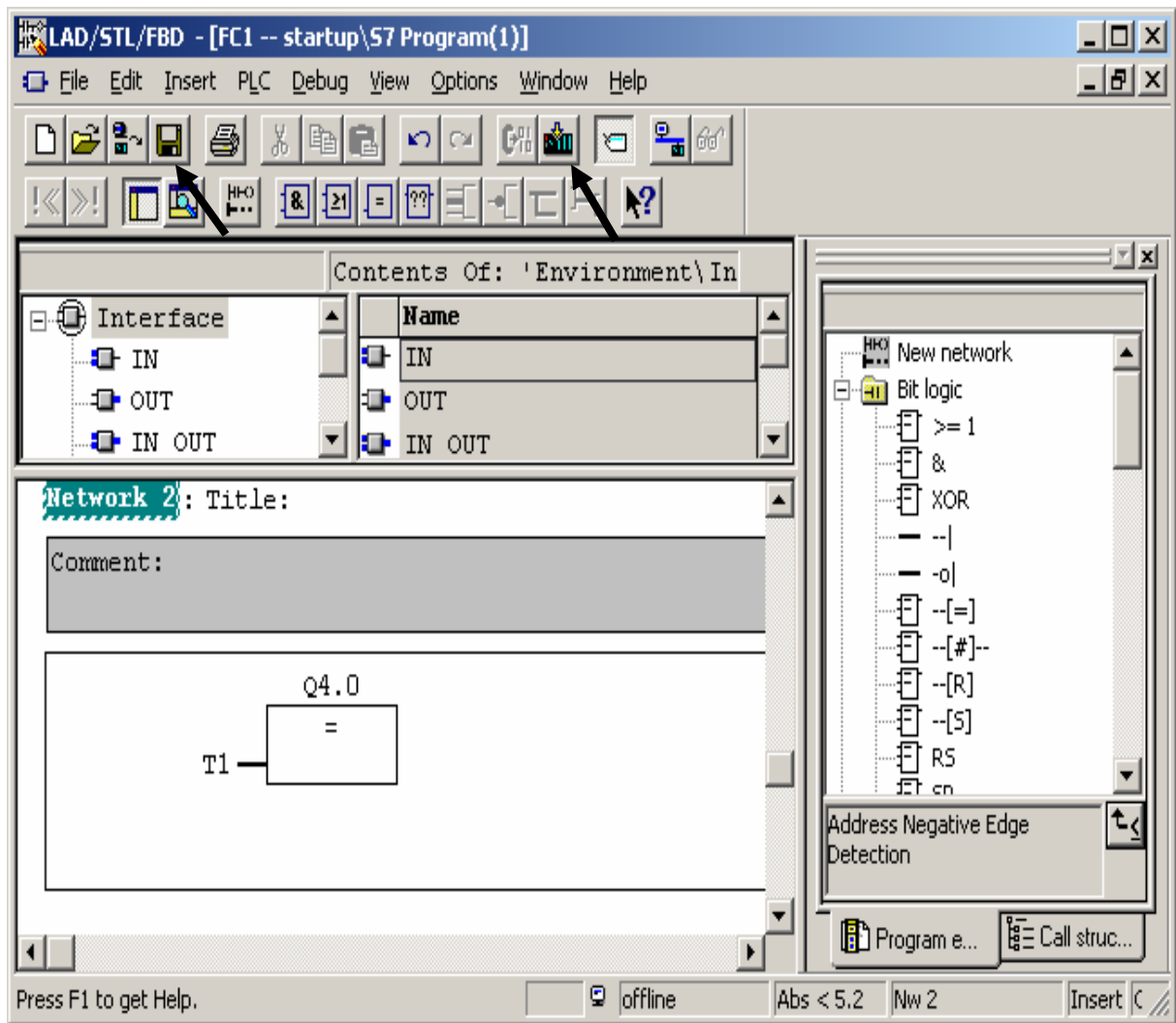
10. Potom kliknutím myši na symbol , vložíme příkaz „Assignment“



The screenshot shows the SIMATIC Manager software interface. The main window is titled "LAD/STL/FBD - [FC1 -- startup\S7 Program(1)]". The menu bar includes File, Edit, Insert, PLC, Debug, View, Options, Window, and Help. The toolbar contains various icons for file operations and editing. The main editor area shows a ladder logic network with a timer T1 and a coil Q. A red box highlights the assignment symbol (=) in the main editor, and a red arrow points to it from the "Contents Of" window. The "Contents Of" window is titled "Contents Of: 'Environment\Inte" and shows a list of symbols including IN, OUT, IN\_OUT, TEMP, and RETURN. The right sidebar shows a library of symbols, including Timers, Word logic, and FB blocks. The status bar at the bottom indicates "Press F1 to get Help.", "offline", "Abs < 5.2", "Nw 2", and "Insert C".





11. Tento příkaz zde platí pro výstup ,Q 4.0' a musí proběhnout pokud timer ,T1' má signál „log.1“. Oba tyto operandy se musí teď vložit, dříve než se FC1 uloží do paměti PC  a než se program přenesou do CPU .

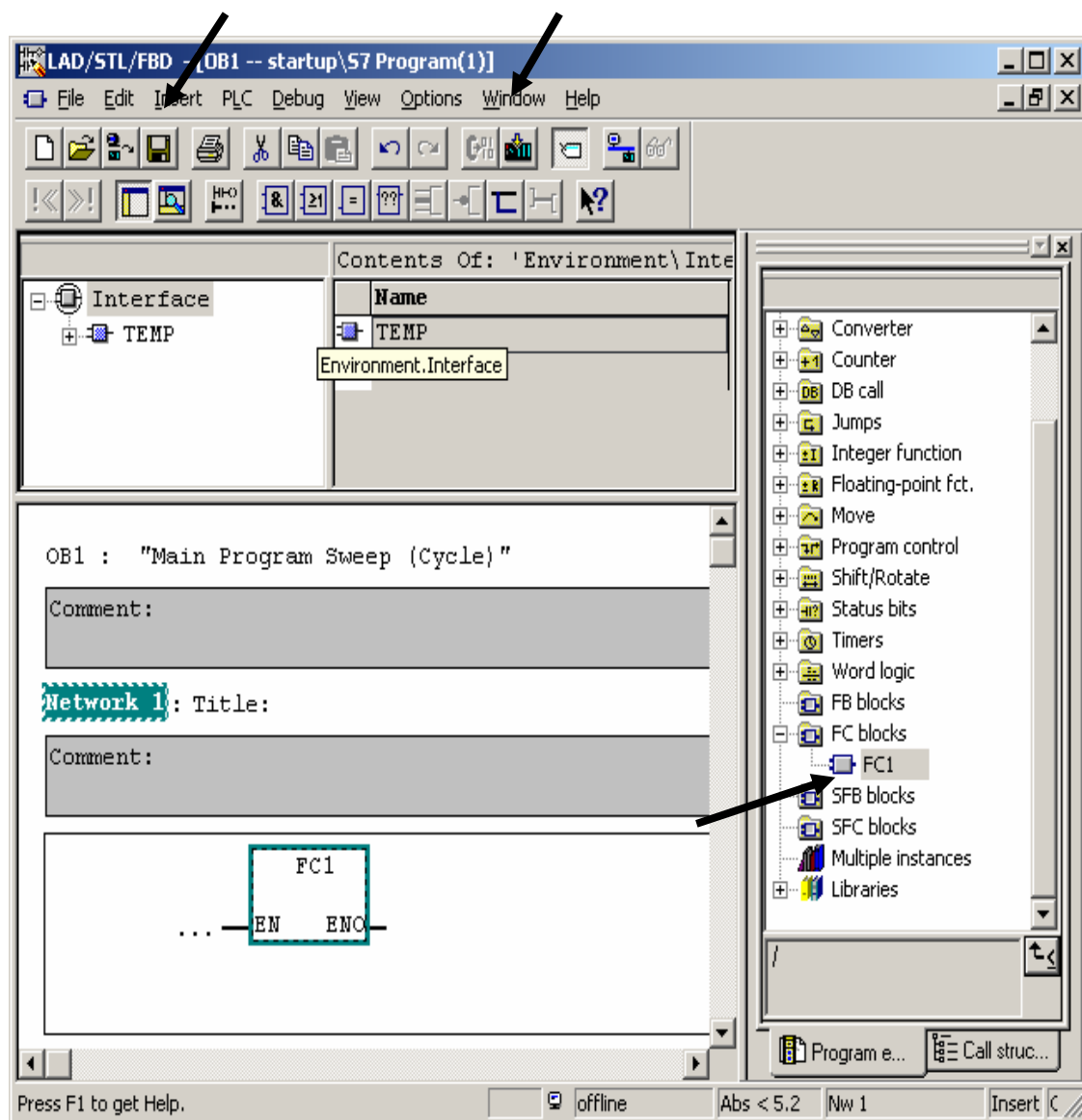


**Poznámka:** Nyní nebudeme zavírat editor „STL/LAD/FBD“ a přes funkci File – Open otevřeme OB1.

12. V OB1 musíme naprogramovat vyvolání FC1. ,OB1' otevřeme dvojitým kliknutím ve ,SIMATIC Manager'
13. Vlastnosti OB1 se ponechají a převezmou s ,OK' ( → OK)
14. V editoru ,LAD/STL/FBD' u OB1 přes nabídku ,View' (Náhled) nastavíme programovací jazyk pro OB1 do tvaru funkčního plánu ,FBD'.



15. Z nabídky funkcí vybereme ‚FC Blocks‘ dále pak ‚FC1‘ a tento blok přetáhneme myší do Networku 1 bloku OB1. Pak se OB1 uloží do paměti PC  a nahraje do CPU .



Tím je programování příkladu ukončeno, program je kompletně sestaven a připraven pro užívání.

## 16. Zpracování analogových hodnot

Ve výrobním procesu existuje řada spojitě se měnících fyzických veličin (tlak, teplota, rychlost, úhlová rychlost atd.), které mají být v PLC zpracovány pro potřeby automatizace. než mohou být tyto analogové hodnoty zpracovány v CPU, musí být konvertovány do digitální podoby. To se děje přes ADC (Analog-to-Digital Converter) ve vstupním analogovém modulu. Typ a rozsah měření se nastavuje přímo na vstupním modulu pomocí kódovacích klíčů. Speciální moduly bez kódovacích klíčů mají různé koncovky pro měření napětí a proudu. Moduly rozsahu měření s kódovacími klíči jsou umístěny na levé straně analogového vstupního modulu. Před instalací modulu je nutné je správně nastavit.



Simatic S7-300 má speciální adresovou oblast pro analogové vstupy a výstupy. Tato oblast leží na bytech 256 až 767. Adresy 0 až 255 slouží pro digitální vstupy a výstupy. Analogové adresy jsou vždy sudá čísla, protože jedna adresa zabírá vždy 2 byte. Vlastnosti analogové karty nastavujeme v hardwarové konfiguraci („HW Config“) kliknutím na tuto kartu.

Analogový modul může zjistit následující diagnostické události:

- chyba v konfiguraci (přidělení)
- přerušení vedení (pokud je aktivován „With Wire Break Check“)
- nad rozsahem měření
- pod rozsahem měření

Pokud hodnota vstupu překročí rozsah nastavený položkami „Upper Limit Value“ a „Lower Limit Value“, modul spustí hardwarové přerušení.

Jednotlivé vstupy:

- *Type of Measurement* – zatrhneme-li tuto položku, zobrazí se možné typy měření (napětí, proud..). Pro nepoužité kanály nebo skupiny kanálů vyberte možnost „Deactivated“. tyto kanály můžete spojit s podvozkovým základem (chassis ground) modulu.
- *Measuring Range* – zatrhneme-li tuto položku, zobrazí se možné měřicí rozsahy pro vybrané typy měření
- *Coding Key Setting* – pokud vyberete typ a rozsah měření, musí být nastaveny na kódovacím klíči velmi konkrétní údaje. Ty jsou zobrazeny zde.



Námi používaný analogový modul SM334 má 4 analogové vstupy a 2 analogové výstupy. Každý ze čtyř vstupních kanálů může mít jiné nastavení. Nepoužité výstupní kanály musejí zůstat otevřené, nikoliv propojené obvodem jako analogové vstupy a musejí být deaktivovány v software.

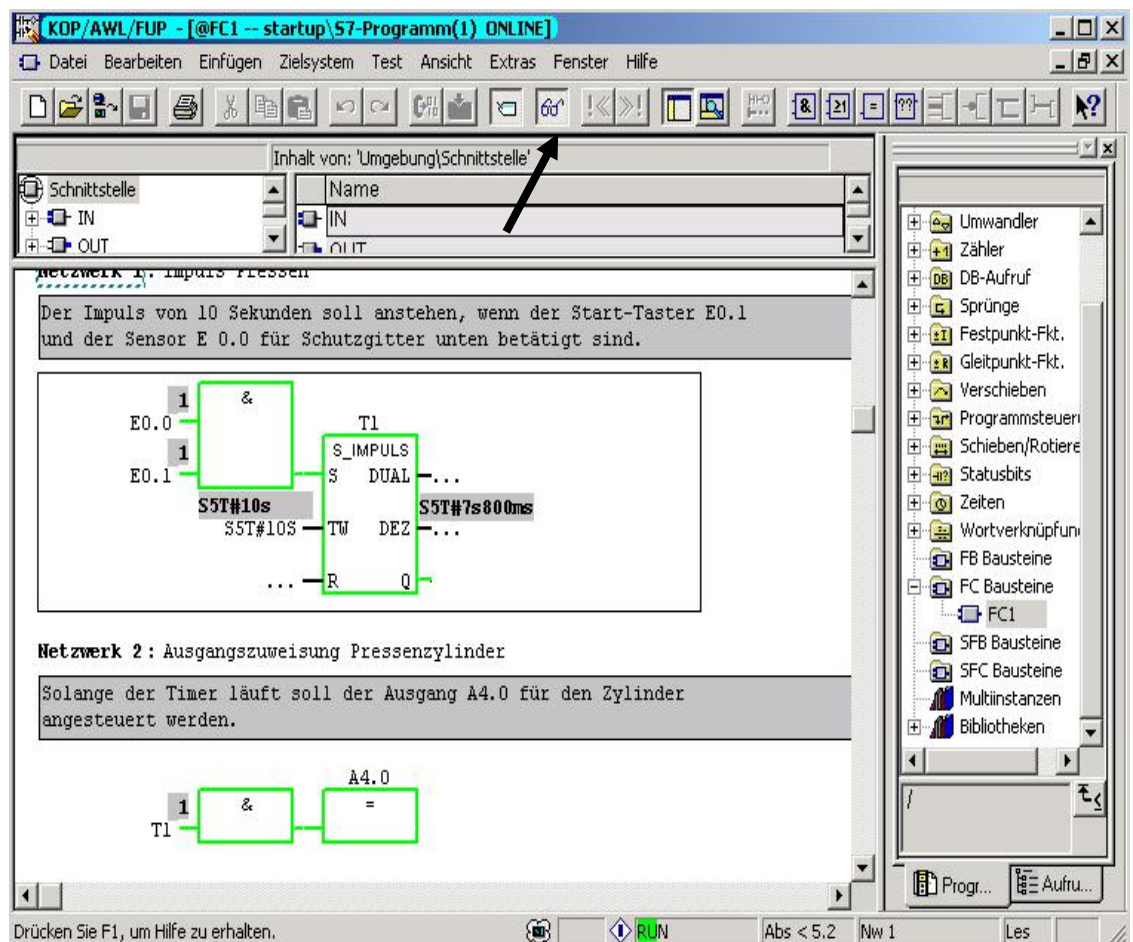
Uživatelský program počítá analogové hodnoty v rozsahu procent (např. 0 až 100.0%). Pokud má být tato hodnota poslána přes modul analogového výstupu, musíme ji převést z reálného čísla 0 až 100.0% na 16-ti bitové celé číslo mezi 0 a 27 648. K tomu využíváme speciální blok FC106. tento blok najdeme v softwaru STEP7 v knihovně „StandardLibrary“ v programu S7 „TI-S7 Converting Blocks“.

## 10. Testování STEP 7- programu v CPU

1. Abychom mohli v FC1 pozorovat program, musíme v editoru 'LAD/STL/FBD' přepnout v nabídce **'Window'** na prvek **'FC1'**.



2. Kliknutím na symbol , můžeme program ve FC1 pozorovat. Taktéž se znázorní průběh timeru a stav signálů pro vstupy a výstupy. ( →  ). Zelenomodré probarvení nápisu v horním rámečku značí, že sledujeme běh programu přímo v CPU Simatic. Znázorněno je to též zeleným označením okna RUN na spodní liště obrazovky.



The screenshot shows the SIMATIC Manager software interface. The window title is "KOP/AWL/FUP - [@FC1 -- startup\S7-Programm(1) ONLINE]". The main area displays two networks:

**Netzwerk 1: Impuls Pressen**  
Der Impuls von 10 Sekunden soll anstehen, wenn der Start-Taster E0.1 und der Sensor E 0.0 für Schutzgitter unten betätigt sind.

The logic diagram for Network 1 shows inputs E0.0 and E0.1 connected to an AND gate (&). The output of the AND gate is connected to a timer T1 (SIMPULS) with a delay of S5T#10s. The timer T1 is connected to a coil (Q) with a delay of S5T#7s800ms.

**Netzwerk 2: Ausgangszuweisung Pressenzylinder**  
Solange der Timer läuft soll der Ausgang A4.0 für den Zylinder angesteuert werden.

The logic diagram for Network 2 shows the timer T1 connected to an AND gate (&), which is connected to an output coil (=) labeled A4.0.

The status bar at the bottom shows "RUN" in green, indicating the program is running in the CPU.

## 11. Programování CPU - úvodní slovo

### Učební cíl:

V této části se naučíme sestavit konfiguraci hardwaru pro CPU 31x-2DP a psát a testovat STEP 7-program. Modul nabízí na základě podrobného příkladu způsob provedení v následujících krocích:

- Zřízení STEP 7-projektu
- Sestavení konfigurace hardwaru pro CPU 313-2DP
- Psaní STEP 7-programu
- Testování programu

### Předpoklady:

Pro úspěšné zvládnutí následující kapitoly se předpokládá znalost:

- Schopnost zacházení s OS Windows
- Základy programování PLC pomocí STEP 7

### Potřebný hardware a software

PC, operační systém Windows 95/98/2000/ME/NT4.0/XP s

- 600MHz s 256MB RAM, volná disková paměť cca. 300 - 600 MB

2 Software STEP7 V 5.x

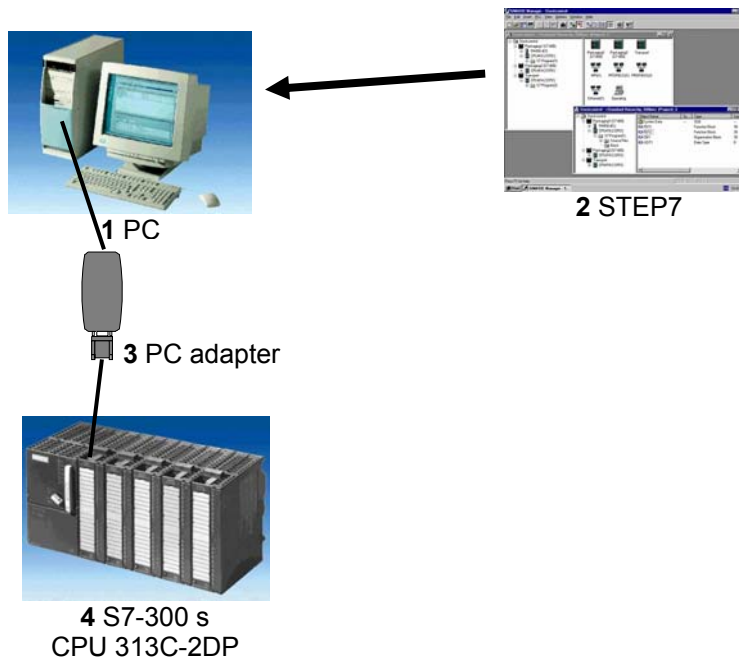
3 MPI-rozhraní pro PC (např. PC-adapter)

4 PLC SIMATIC S7-300 s CPU 313C-2DP

Příklad konfigurace:

Zdroj napájení: PS 307 2A (popř. jiný externí zdroj 24V=, 2A)

- CPU: CPU 313-2DP



## 12. Údaje pro použití CPU 313C-2DP



CPU 313C-2DP je centrální procesor, který se dodává s integrovaným rozhraním PROFIBUS DP jako i s integrovanými vstupy a výstupy.

Pro CPU 313C-2DP jsou k dispozici následující PROFIBUS - profily protokolu:

- DP-rozhraní jako Master podle normy EN 50170
- DP-rozhraní jako Slave podle normy EN 50170
- přenosová rychlost 12 Mbit/s
- počet slave jednotek na stanici – max. 32

PROFIBUS-DP (**D**ecentrální **P**eriferie) je profil protokolu pro připojení přístrojů s velmi rychlou reakční dobou.

Vlastnosti CPU 313C-2DP s uvedenými daty se obzvláště dobře hodí pro vzdělávací účely:

- 32KByte pracovní paměť, paměť pro zavádění dat - MicroMemoryCard (MMC), od 64KByte do 8 MByte (u našich CPU používáme paměť 2MB)
- čas zpracování příkazu 0,1 ms / 1kB příkazů
- Bitová paměť 256Byte
- 16 DI (digitální vstupy) 24V DC integrované,
- 16DQ (digitální výstupy) 24V DC, 0,5A
- 3 kanály pulsně-širokové modulace (max. 2,5kHz)
- 3 kanály měření frekvence (max. do 30kHz)

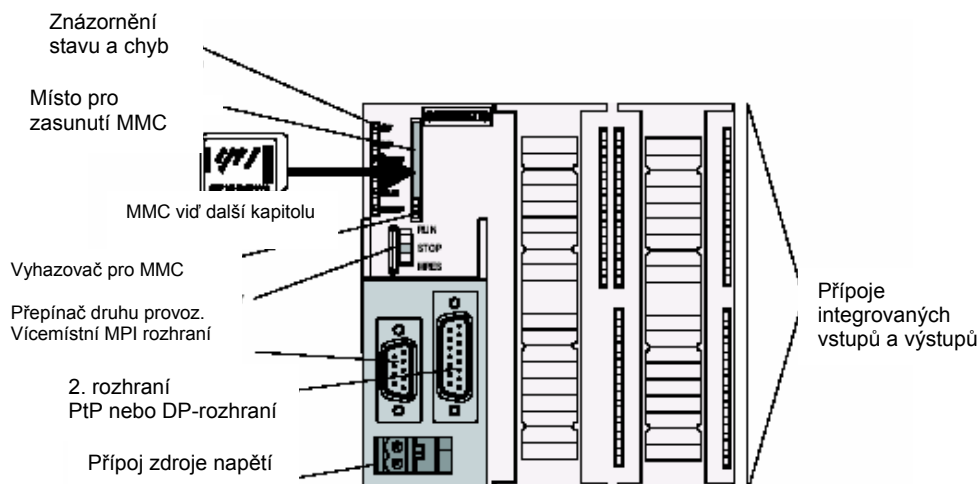
### 12.1 Obsluha CPU 313C-2DP



#### Prvky ovládání a zobrazení

Následující obrázek představuje prvky ovládání a zobrazení CPU 313C.

Uspořádání a počet prvků je odlišné u některých CPU, záleží na výrobní šarži





### Znázornění stavu a chyb

CPU je vybavená následujícími signály, pomocí LED-diod:

#### Znázornění na CPU:

SF	(červená)	skupinová chyba, vnitřní závada CPU nebo modulu
BF	(červená)	závada v baterii (vybitá nebo nepřítomná)
DC5V	(zelená)	5V-napájení pro CPU a sběrnici je OK
FRCE	(žlutá)	indikuje, že alespoň 1 vstup či výstup je trvale ovlivněn (forsován)
RUN	(zelená)	svítí je-li CPU v pozici RUN; bliká při spouštění CP s $f = 2\text{Hz}$ ; v pozici STOP bliká s $f = 0,5\text{Hz}$
STOP	(žlutá)	svítí při režimu STOP pomalu bliká při požadavku resetování paměti ( $f = 0,5\text{Hz}$ ) rychle bliká v průběhu resetu paměti ( $f = 2\text{Hz}$ ) pomalu bliká je-li nezbytné resetovat paměť z důvodu vsunutí paměťové karty

#### Místo pro zasunutí SIMATIC Micro Memory Card (MMC)

Jako modul paměti pro CPU 313C se používá SIMATIC Micro Memory Card (MMC). MMC má funkci pomocné paměti a mobilního nosiče dat. Pro funkci CPU **se musí** MMC zasunout do určeného místa, protože CPU 313C-2DP nemá integrovanou vlastní pomocnou paměť.

#### Přepínač funkcí

Pomocí přepínače funkcí se dá nastavit druh provozu CPU.  
Tento je ve formě kolébkového přepínače s 4 polohami spínání.

#### Polohy přepínače funkcí

Polohy přepínače jsou vysvětleny v pořadí, v jakém se nacházejí na CPU.

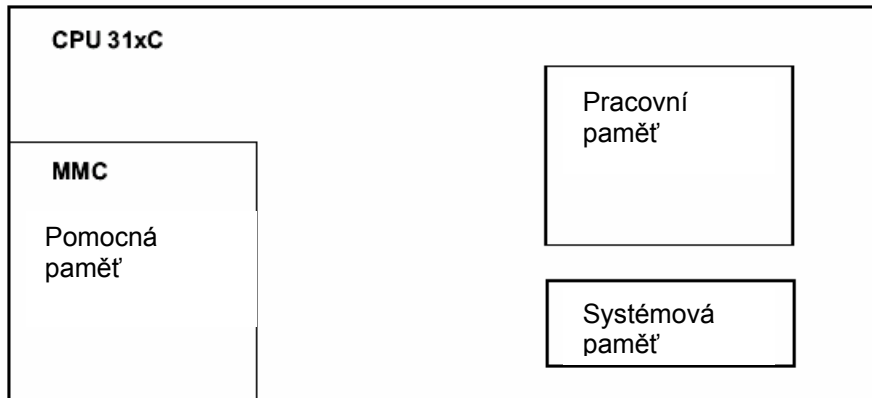
Poloha	Znamená	Vysvětlení
RUN	Vykonávání programu	CPU zpracovává uživatelský program, přístup z PC pouze pro čtení
STOP	Druh provozu STOP	CPU nezpracovává uživatelský program
MRES	Smazání dat	Poloha klávesnice přepínače pro vymazání dat. Vymazání pomocí tohoto přepínače vyžaduje řadu speciálních obslužných operací
RUN-P	Vykonávání programu	CPU zpracovává uživatelský program, přístup z PC pro čtení i zápis

## 12.2 Oblasti paměti CPU 313C



### Úvod

Paměť CPU 313C se rozděluje do 3 oblastí:



### Poznámka

Ukládání uživatelských programů a tím i provoz CPU 313C je možný **jenom se zasunutou MMC**.

### Pomocná paměť

Pomocní paměť se nachází na SIMATIC Micro Memory Card (MMC).

Její velikost odpovídá přesně MMC. Slouží pro snímání kódů a dat prvků jako i dat systémů (konfigurace, spojení, parametry konstrukčních celků, atd.).

Prvky, označené jako relevantní pro provoz, se ukládají výhradně v této paměti.

Také je možno na kartu MMC uložit i kompletní data projektu.

### Pracovní paměť

Pracovní paměť je integrovaná na CPU a nedá se rozšířit. Slouží pro zpracování kódu i dat uživatelského programu. Zpracování programu probíhá výlučně v oblasti pracovní a systémové paměti.

### Systémová paměť

Systémová paměť je integrována na CPU a nedá se rozšířit.

Obsahuje:

- oblast operandů snímače, časů a čítače
- zobrazení procesů vstupu a výstupu
- lokální data



### Remanence

CPU 313C obsahuje remanentní paměť. Remanence se realizuje na MMC a na CPU.

Díky remanenci zůstává obsah paměti MMC zachován i když je napájení CPU vypnuto, rovněž při novém startu CPU je obsah MMC zachován.

### Pomocná paměť

Program v pracovní paměti (MMC) zůstane vždy remanentní (zachován). Je zabezpečený proti výpadku proudu již při ukládání dat a na MMC je uložen bezpečně proti vymazání.

## Pracovní paměť

Vaše data v pracovní paměti jsou rovněž zajištěna při výpadku proudu na MMC.

## Systémová paměť

Pro snímače, časy a sčítač určité Vy, v projektování (vlastnosti CPU, remanence registru), které části mají být remanentní a které se musí při novém startu instalovat s "0".

Diagnostický bufer, MPI-adresy jako i čítač provozních hodin jsou všeobecně uloženy v remanentní oblasti na CPU. Remanencí MPI-adres a výkonu přenosu dat je zabezpečené, že Vaše CPU, v případě výpadku proudu, po vymazání dat nebo po ztrátě komunikačního parametrování (vytažením karty MMC nebo vymazáním parametrů) je ještě stále schopná komunikace.

## 13. Sestavení konfigurace hardwaru pro CPU 313C-2DP



Spravování dat ve STEP 7 se provádí v 'SIMATIC Manager'-u. Zde se dají např. prvky programu kopírovat, měnit zpracovávat. Obsluha zodpovídá běžnému standardu pro WINDOWS 95/98/NT/XP. K obsluze programu lze využívat i pravé klávesy myši.

Ve STEP 7 má každý projekt pevně určenou strukturu. Programy jsou uloženy v paměti v následujících blocích:

**SIMATIC 300 Station:**  
Zde jsou uloženy data určená pro konfiguraci hardwaru

**Projekt:**  
Tento blok obsahuje název projektu a další propojení s okolím (např. MPI a PROFIBUS).

**Sources:**  
Zde se ukládají zdroje (např. SCL- zdroje), které se překladem do jiných progr. jazyků mohou proměnit na funkční programy.

**Blocks:**  
Zde se ukládají programové prvky (OB, FB, FC, SFB, SFC, DB atd.)

**Symbols:**  
Zde se ukládají seznamy prvků se symbolicky zadanými adresami

**CPU:**  
Zde je uložen údaj o vybrané konkrétní CPU.

**S7 Program:**  
Zde se spravují uživatelské programy



**Uživatel musí udělat následující kroky, aby mohl sestavit projekt a konfiguraci hardwaru:**

1. Centrálním nástrojem ve STEP 7 je **'SIMATIC Manager'**, který vyvoláme dvojitým kliknutím na následující ikonu na pracovní ploše.

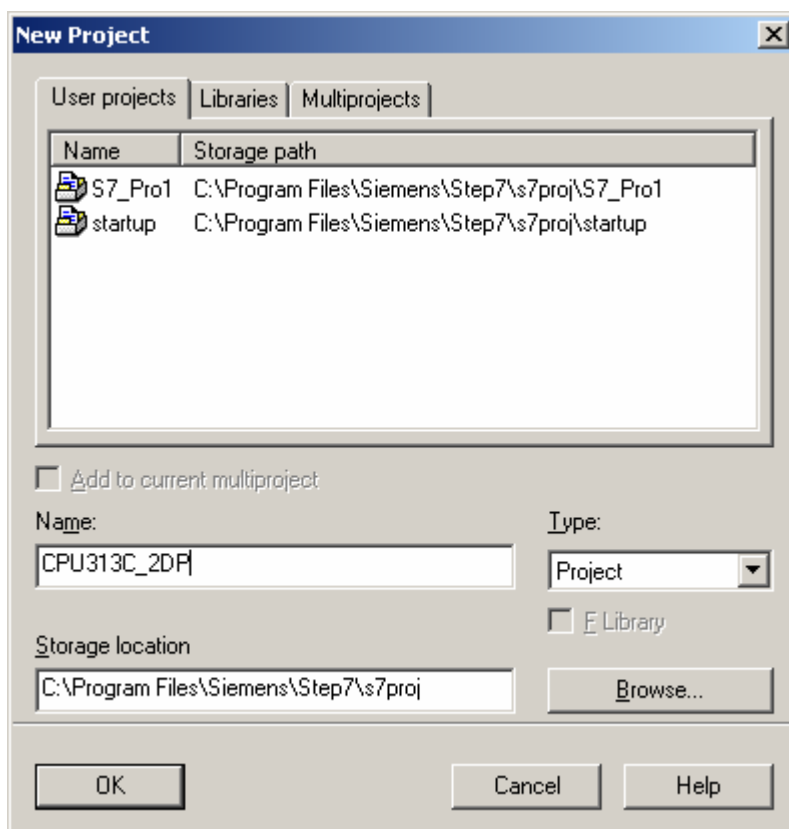


SIMATIC Manager

2. STEP 7-programy se spravují v projektech . Založíme nový projekt a to tak, že v roletovém menu v nabídce „File“ klikneme na položku „New“.

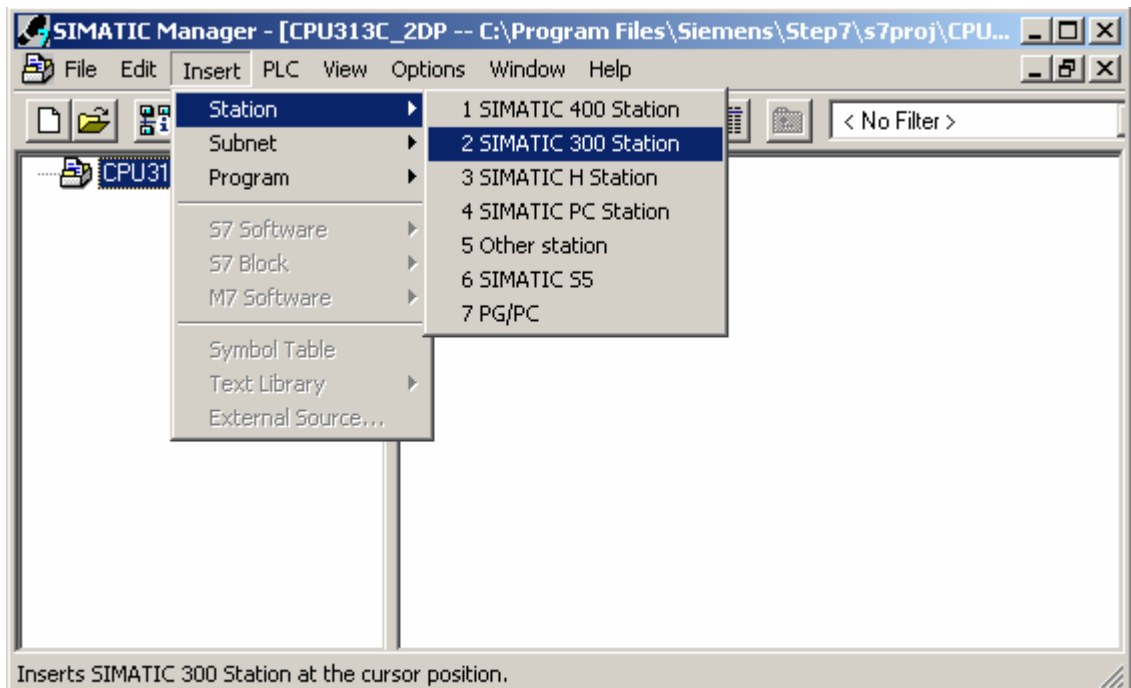


3. Projektu se zadá jméno **'CPU313C\_2DP'** do pole **'Name'** a potvrdíme tlačítkem **'OK'**.

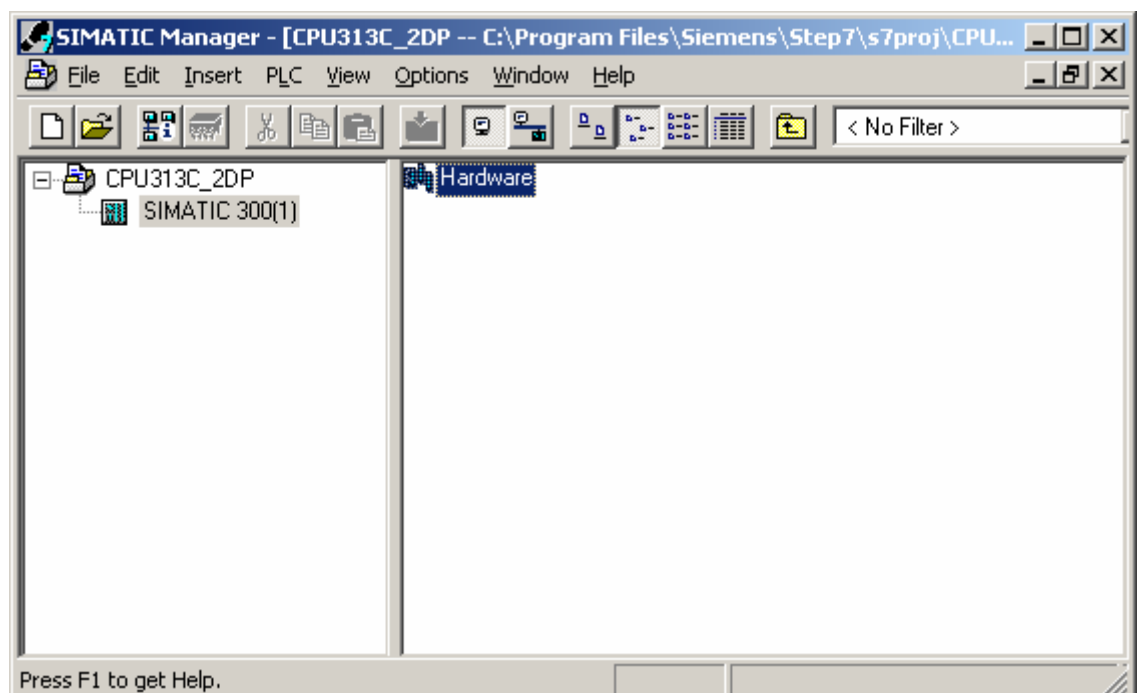





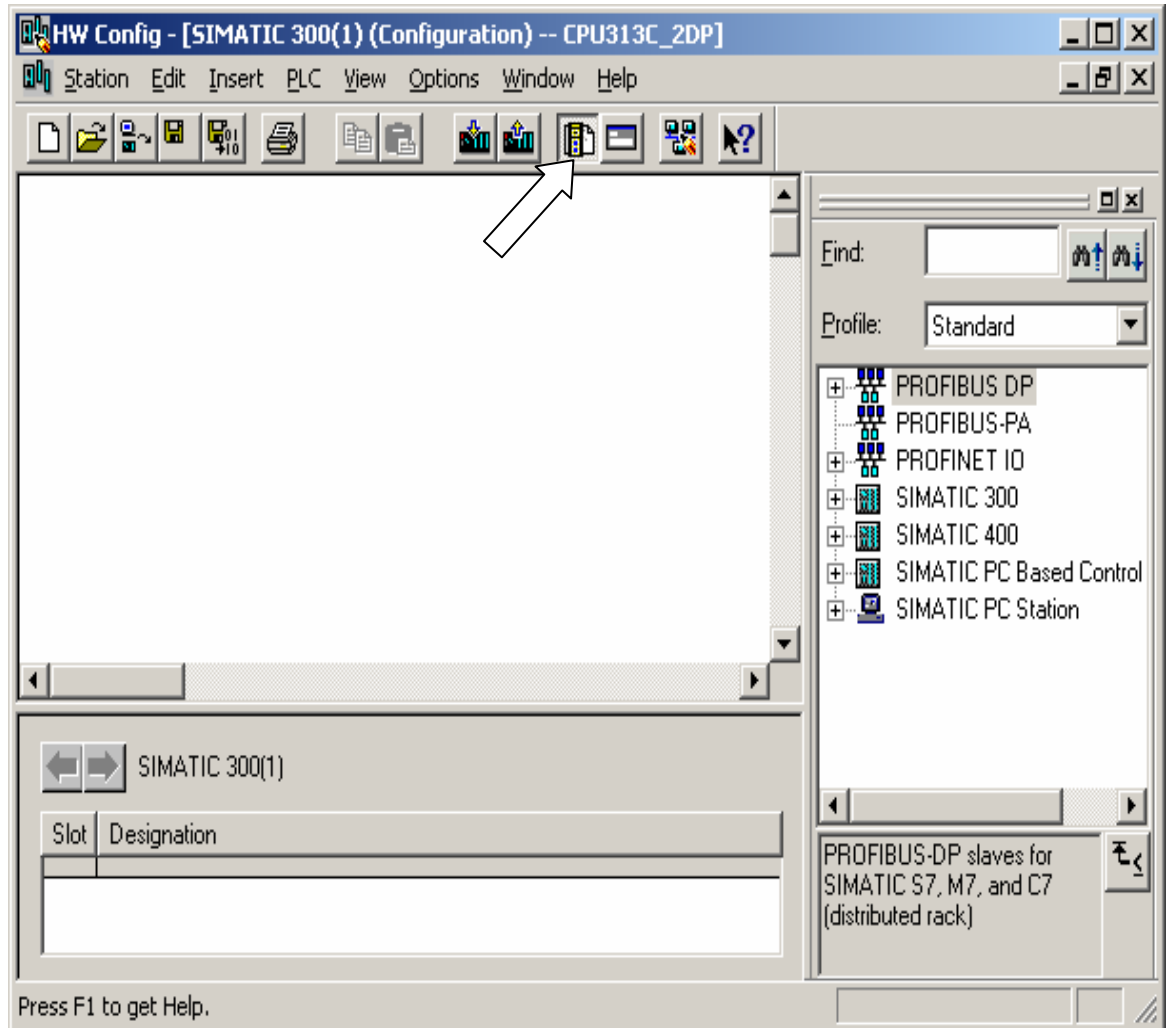
4. Dalším krokem je přiřazení typu systému. Protože používáme systém S7-300, vybereme z nabídky ‚**SIMATIC 300-Station**‘. K nabídce se dostaneme výběrem položky ‚Insert‘ v roletovém menu. Dále postupně vybereme položky ‚Station‘ a ‚Simatic 300 Station‘ (viz následující obrázek).



5. Nástroj konfigurace hardware otevřeme dvojitým kliknutím myši na položku ‚**Hardware**‘.

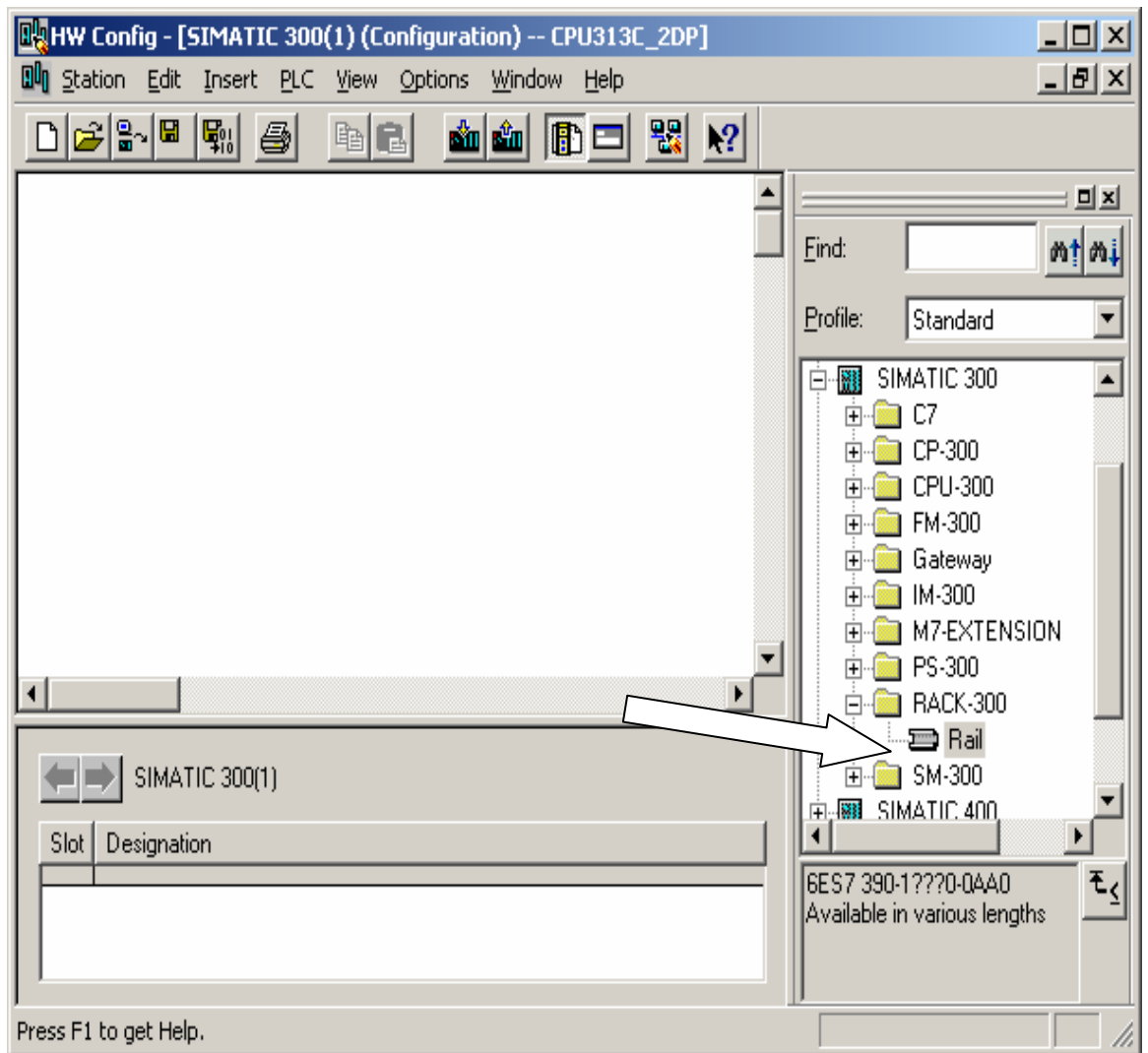


6. Otevře se další okno, ve kterém kliknutím na symbolu  otevřeme katalog hardwaru (okno vpravo). Zde jsou k dispozici všechny nositele celků, moduly rozhraní pro projektování Vaší sestavy hardwaru, které jsou rozděleny do seznamů:
- PROFIBUS-DP/PA, PROFINET IO, SIMATIC 300, SIMATIC 400 a SIMATIC PC Based Control, SIMATIC PC Station.





7. V pravém okně vybereme položku **'Rail'** -, Profilová lišta ( → SIMATIC 300 → RACK-300 → Rail ).



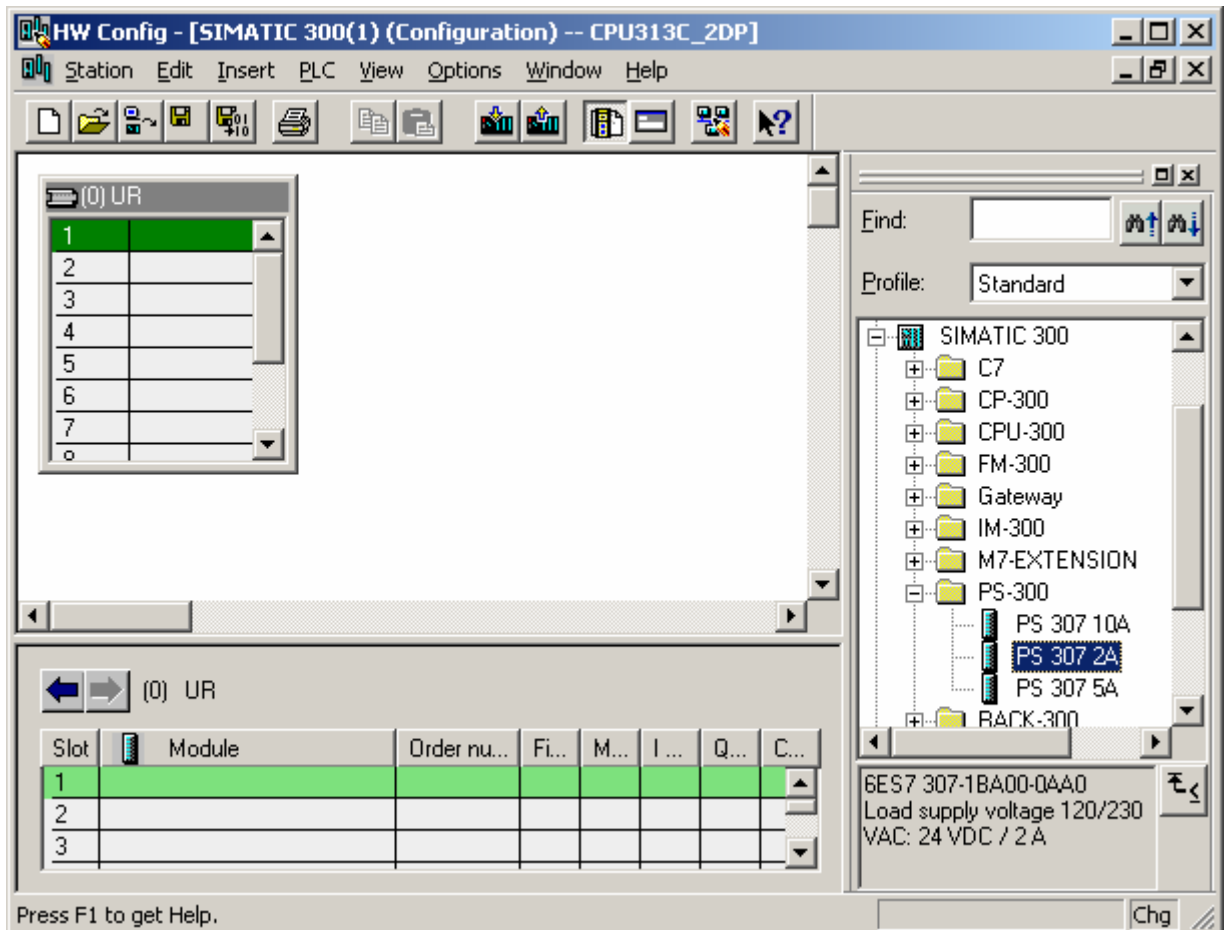
Poté se automaticky zobrazí tabulka konfigurace pro strukturu prvního závěsného rámu (Rack 0).



8. Z katalogu hardwaru v pravém okně se teď dají zvolit všechny konstrukční celky a zadají se do tabulky konfigurace. To jsou pak prvky, které použijete ve Vašem reálním závěsném rámu (Rack).

Aby se tak stalo, musíte kliknout na označení daného celku, klávesnicí myši podržet a přenést daný konstrukční celek přetažením myši do řádku v tabulce, nadepsané „(0) UR“.

Začněte nejprve se zdrojem ,PS 307 2A'. (→ SIMATIC 300 → PS-300 → PS 307 2A)

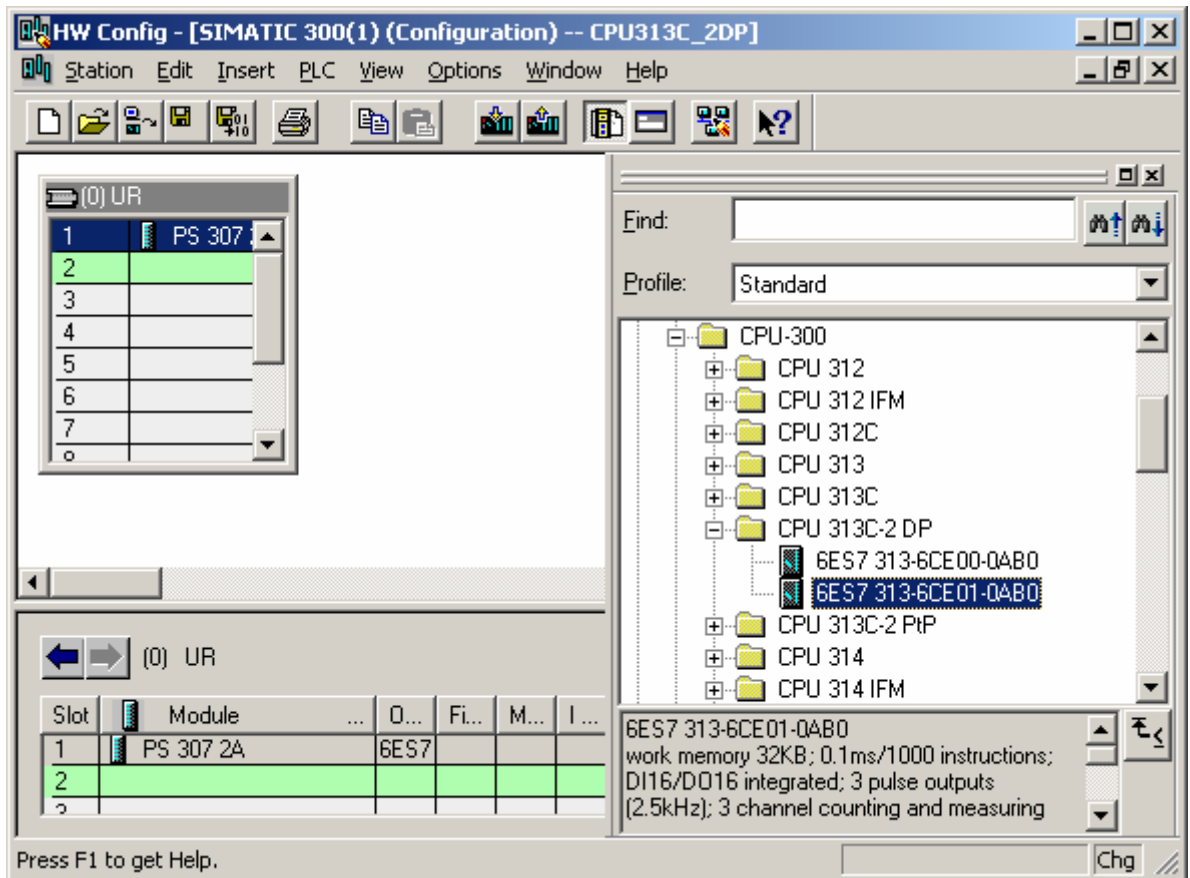


**Poznámka:** Jestliže se Váš hardware odlišuje od zde znázorněného, musíte Vaše dané konstrukční celky zvolit z katalogu a zavést do závěsného rámu.

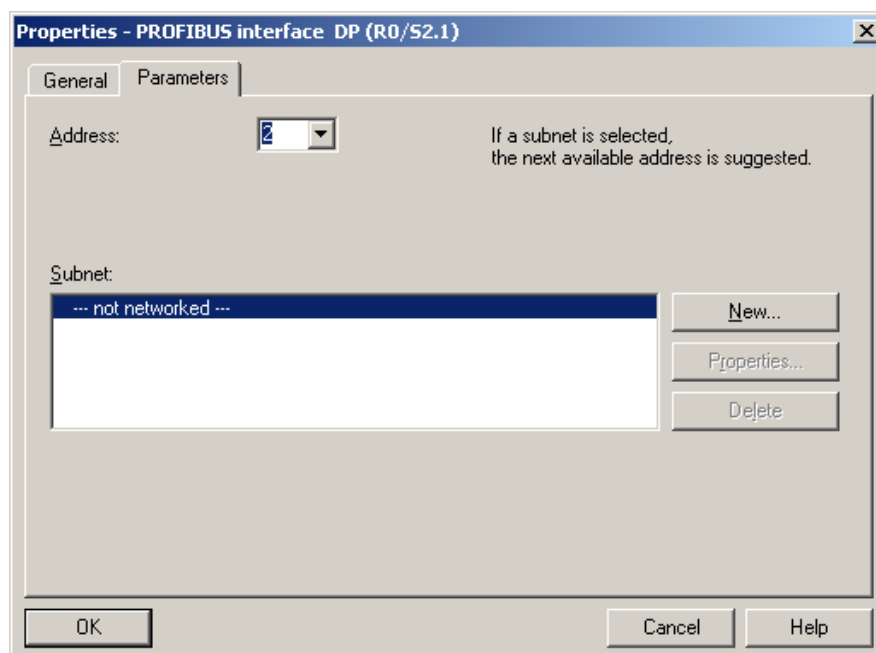
Základní údaje pro zvolený konstrukční díl najdete aktuálně v rámečku pod hardwarovým katalogem. Najdete zde rovněž objednávací číslo tohoto celku. Toto objednávací číslo se nachází i přímo na komponentech.



9. V dalším kroku přeneseme CPU 313C-2DP na druhou pozici v sestavě. Na první pozici máme již umístěn zdroj z předchozího kroku. Musíme vybrat přesně tu CPU, kterou podle objednáčích čísla skutečně máme. Objednáčí číslo se nám znázorňuje v hardwarovém katalogu.



10. V dalším dialogu musíme nastavit integrované rozhraní PROFIBUS. Doporučuje se převzít nabízené nastavení stiskem tlačítka „OK“.





**Poznámka:** Pozice č.3 v tabulce se neosazuje, je rezervována jako logická adresa pro modul interfacu (pro spojení s dalším rámem – „Rackem“). Další konstrukční celky můžeme dle vlastního výběru umístit od pozice (slotu) 4 dále. Přetažením myši můžete vkládané hardwarové moduly vložit na kterékoli místo v tabulce.



- U některých konstrukčních celků můžeme změnit jejich vlastnosti. Klikneme-li např. na CPU 313C-2DP v tabulce, objeví se nám okno se záložkami, ve kterých lze nastavit nejrůznější parametry CPU. Na obrázku je znázorněna záložka, ve které můžeme nastavit monitorovací čas snímání cyklu (zde 150ms). Pokud je tento čas překročen, přejde CPU do stavu STOP. nejčastější příčinou překročení času je chyba v programu. V záložce „General“ najdeme přednastavenou MPI adresu.

The screenshot shows the 'HW Config' window for a SIMATIC 300(1) system. The main window displays a rack configuration table with the following modules:

Slot	Module	O...	Fi...	M...
1	PS 307 2A	6ES7		
2	CPU 313C-2 DP	6ES7	V2.0 2	
X2	DP			
2.2	DI16/DO16			124.. 124..
2.4	Count			768.. 768..
3				

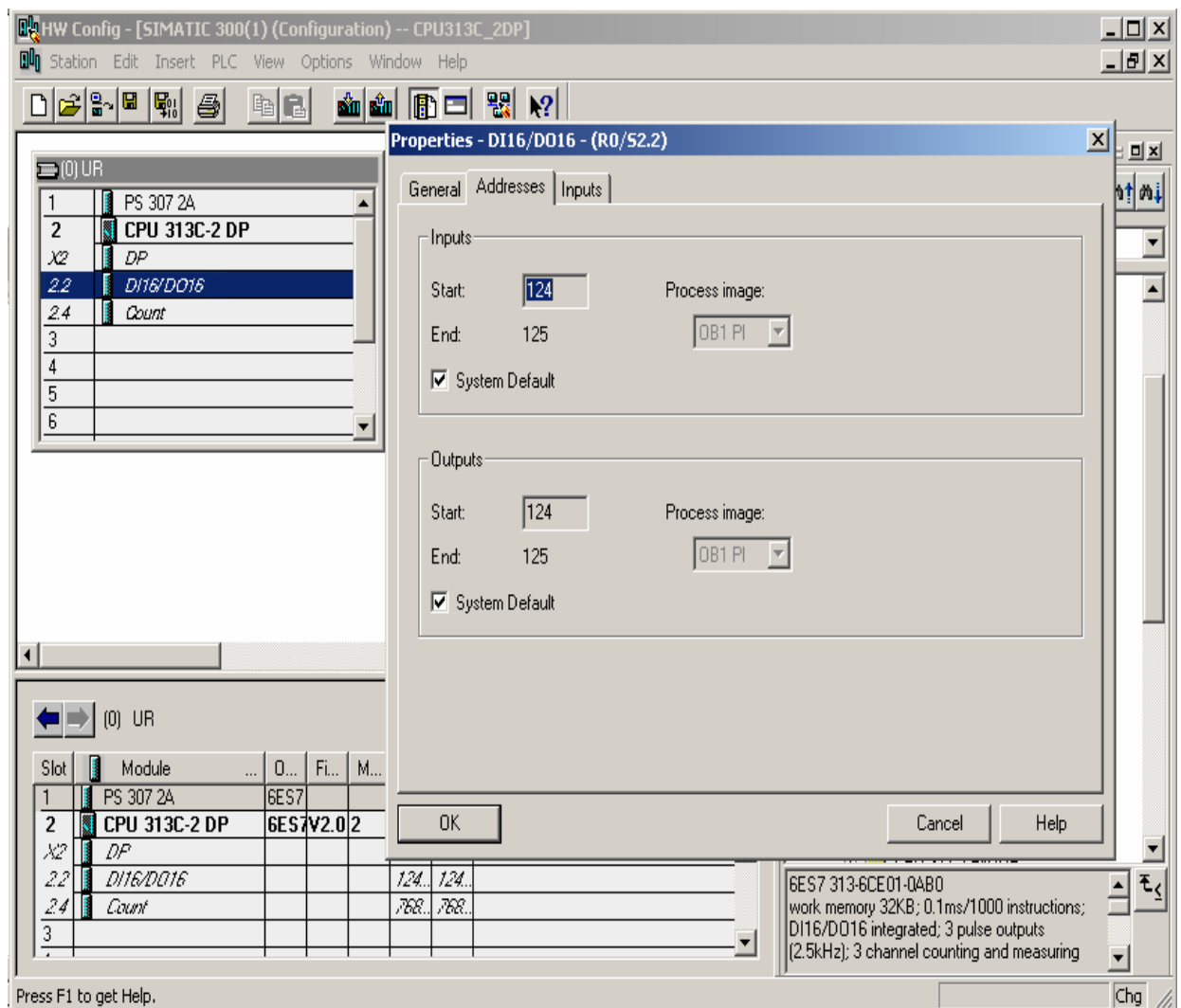
The 'Properties - CPU 313C-2 DP - (R0/S2)' dialog box is open, showing the 'Cycle' tab. The 'Scan Cycle Monitoring Time [ms]' is set to 150. Other settings include 'Update OB1 process image cyclically' (checked), 'Minimum Scan Cycle Time [ms]' (0), 'Scan Cycle Load from Communication [%]' (20), and 'OB85 - Call Up at I/O Access Error' (No OB85 call up). The 'Clock Memory' section is also visible, with 'Clock memory' unchecked and 'Memory Byte' set to 0.



At the bottom right of the dialog, the following specifications are listed:  
6ES7 313-6CE01-0AB0  
work memory 32KB; 0.1ms/1000 instructions;  
DI16/DO16 integrated; 3 pulse outputs  
(2.5kHz); 3 channel counting and measuring



12. Adresy vstupů a výstupů konstrukčních celků se dají změnit jen u jednotek S7-300 CPU s integrovaným rozhraním PROFIBUS.

To je možné provést tím, že dvakrát klikneme na příslušný konstrukční celek v malé tabulce a adresy nastavíme v okně, které se otevře (viz obr.). Pokud tak neučiníme, proběhne automatické zadání adresy závisle od pozice umístění (slotu) na Racku.



13. Tabulka konfigurace se poté nejprve uloží do paměti PC a přeloží kliknutím na symbol  (Save a compile) a pak se nahraje do paměti CPU kliknutím na symbol . Přitom musí být přepínač na čelní straně CPU v pozici Stop!

## 14. Napsání STEP 7-programu



Program, který se zde testuje, se dá napsat ve formě seznamu pokynů (STL) a obsahuje pouze dva řádky.

A sice, frekvence ovládacího bytu taktu MB100, který je aktivován v hardwaru, se zde vyvedou na jeden výstupní byte.

### Seznam přiřazení:

MB100	Takt	ovládací byte taktu
QB0	AB	údaj

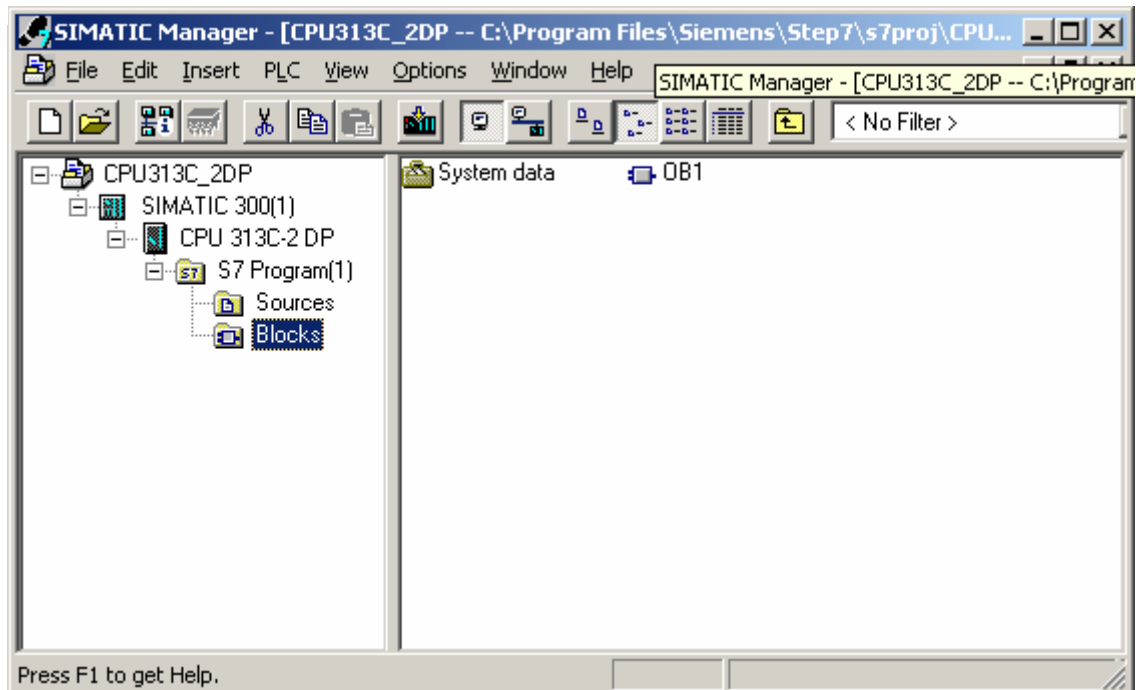


Každému bitu ovládacího bytu taktu je zadána doba periody/frekvence. Je stanovena takto:

Bit:	7	6	5	4	3	2	1	0
Doba periody (s):	2	1,6	1	0,8	0,5	0,4	0,2	0,1
Frekvence (Hz):	0,5	0,625	1	1,25	2	2,5	5	10



1. Označte v ‚SIMATIC Manager‘ registr ‚Blocks‘ - bloky.( → SIMATIC Manager → Blocks)







2. V SIMATIC Manageru pak pokračujeme dvojitým kliknutím na blok **OB1** v pravém okně. Otevře se okno s vlastnostmi bloku OB1. Zde vybereme položku „Created in Language“ a změním na STL (viz následující obr.).

**Properties - Organization Block**

General - Part 1 | General - Part 2 | Calls | Attributes

Name:

Symbolic Name:

Symbol Comment:

Created in Language:

Project path:


Storage location of project:

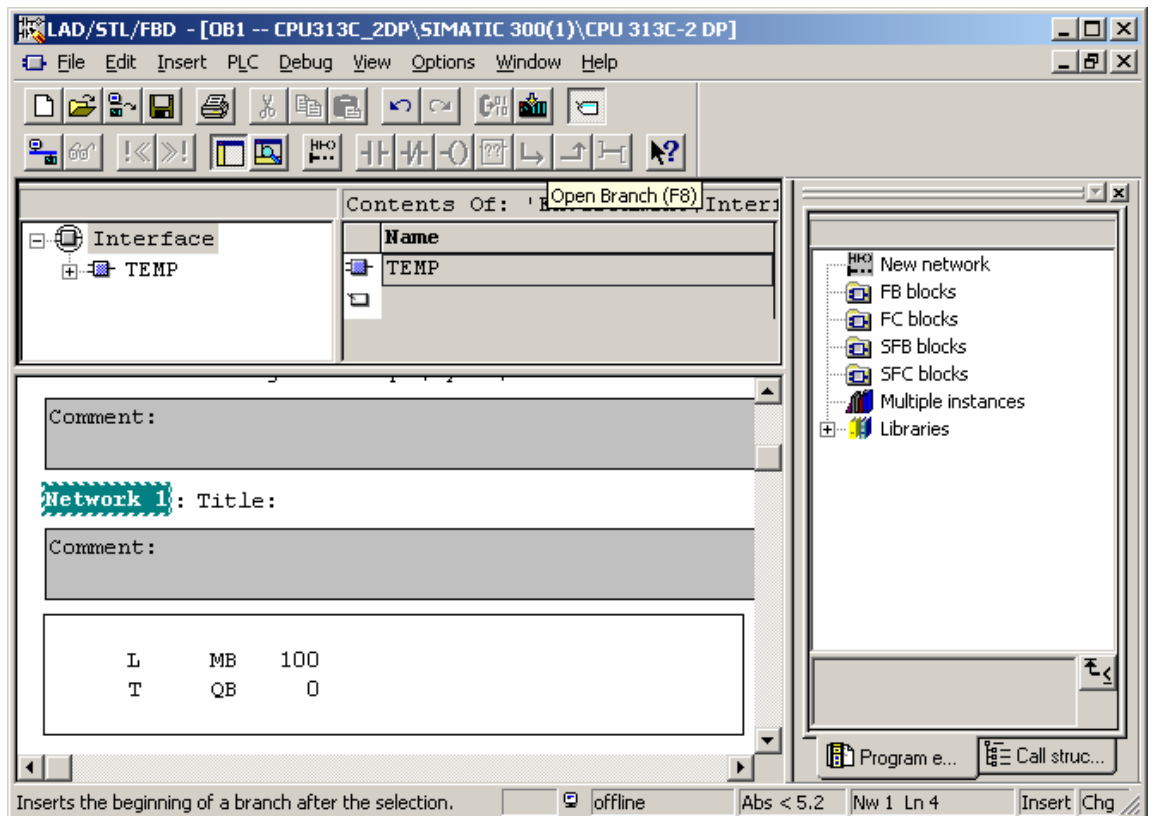
	Code	Interface
Date created:	07/12/2005 08:20:18 AM	
Last modified:	07/02/2001 03:03:43 PM	15/02/1996 04:51:12 PM

Comment:

OK Cancel Help



3. V následujícím okně „LAD, STL, FBD“ můžeme začít psát STEP 7-program. Je otevřen organizační blok OB1 s prvním otevřeným prázdným krokem (Networkem). Abychom mohli začít do tohoto prázdného kroku (Networku) zapisovat, musíme jej označit. Jednotlivé programy jsou ve STEP 7 obvykle rozděleny na kroky. Kliknutím na symbol obvodu , otevřete nový krok (Network). Zápis provedeme ve formátu STL. Textové komentáře a poznámky k programu se zde od programových příkazů oddělují ve formátu znaménkem „//“.



```
L    MB 100
T    QB 0,
```



Tímto zápisem jsme tedy naprogramovali, že MB100 (tj. Byte zápisníkové paměti č. 100 se přenesou na byte výstupů s číslem 0.

V důsledku toho bliká 8 bitů výstupního Bytu v různých frekvencích ovládacích bitů taktu.


## 15. Testování STEP 7-programu

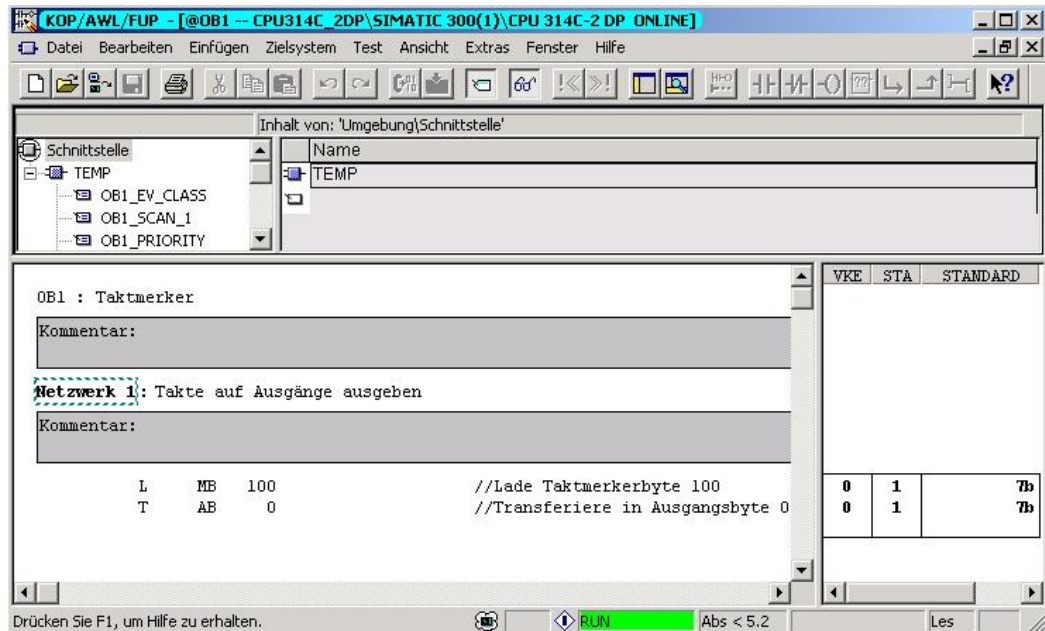


STEP 7-program, který máme otestovat, se teď uložíme do CPU PLC. V našem případě je to jenom organizační blok OB1, jiné prvky náš program neobsahuje.

17. Organizační blok OB1 uložit nejprve do paměti PC tlačítkem  a potom jej zavedeme do paměti CPU tlačítkem . Přitom je klíčový přepínač na čelní stěně CPU v poloze STOP!



1. Přepnutím přepínače na CPU do polohy RUN se spustí program a kliknutím na symbol  je možné program v ,OB1' pozorovat. Zelená barva v horním rámečku hlásí, že sledujeme běh programu On-line, tedy přímo v CPU.



## 16. Archivace, dokumentace projektu - úvodní slovo

### Učební cíl:

V této kapitole se seznámíme s nástroji pro programovou dokumentaci a pro ukládání programů STEP 7 do paměti:

- symbolické zadání adres
- ukládání programů na EPROM-Memory Card v SIMATIC S7-300
- STEP 7-programy archivovat a dearchivovat
- přehled referenčních dat pro STEP 7-program

### Předpoklady:

Pro úspěšné zvládnutí této kapitoly se předpokládá následovní znalost:

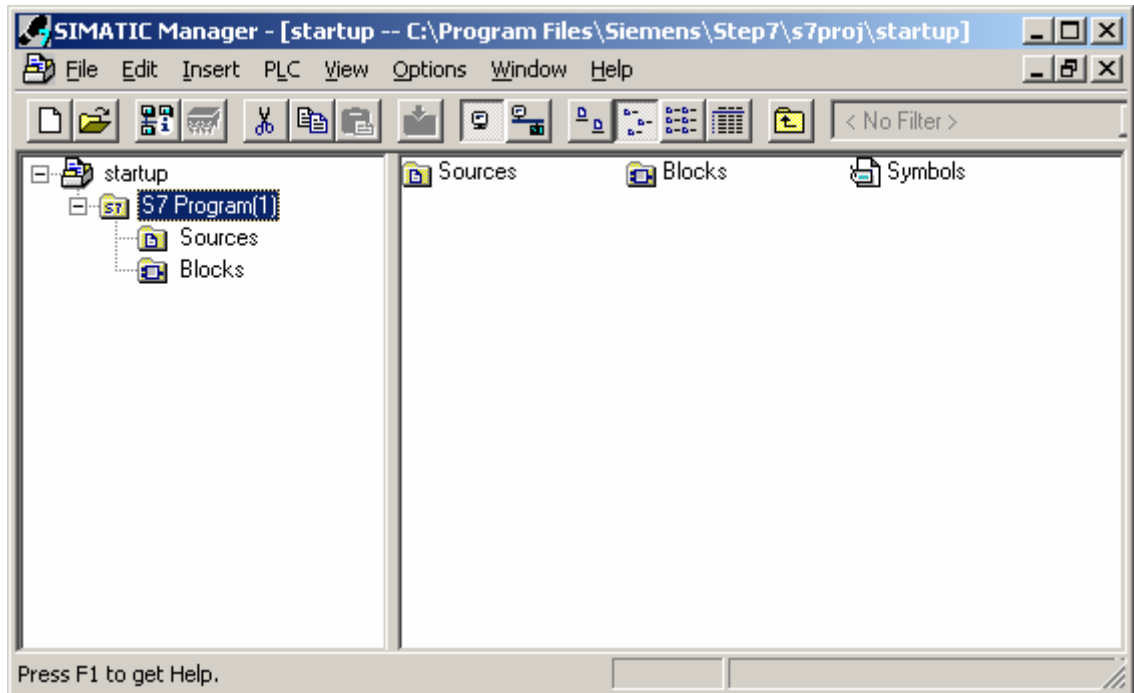
- zvládnutí OS Windows
- znalost základů programování PLC pomocí STEP 7

## 16.1 Symbolické zadání adresy

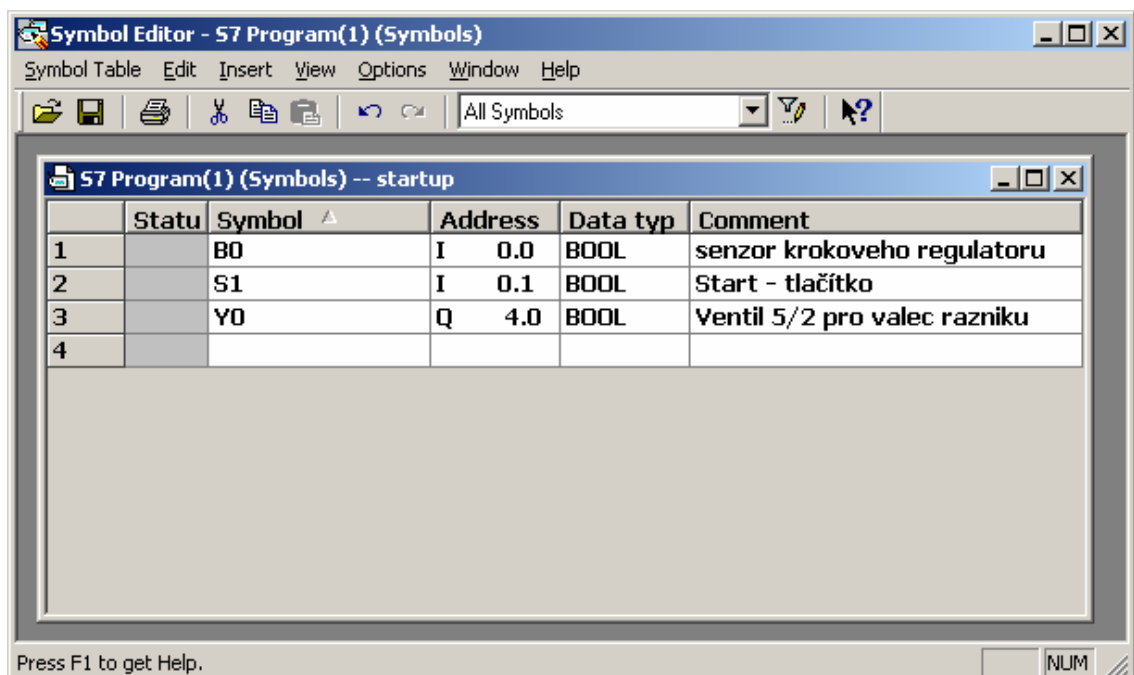


Symbolické zadání adresy umožňuje pracovat místo adres se symboly, např. 'MOTOR\_ON'. Symboly pro vstupy, výstupy, čítač, značkovač, časy a prvky jsou uvedeny ve seznamu symbolů. Tabulka symbolů pro projekt ,**startup**' se sestaví následovně:

1. Otevřete tabulku symbolů „Symbols“. Ikonu pro její otevření najdeme ve složce S7 program(1)

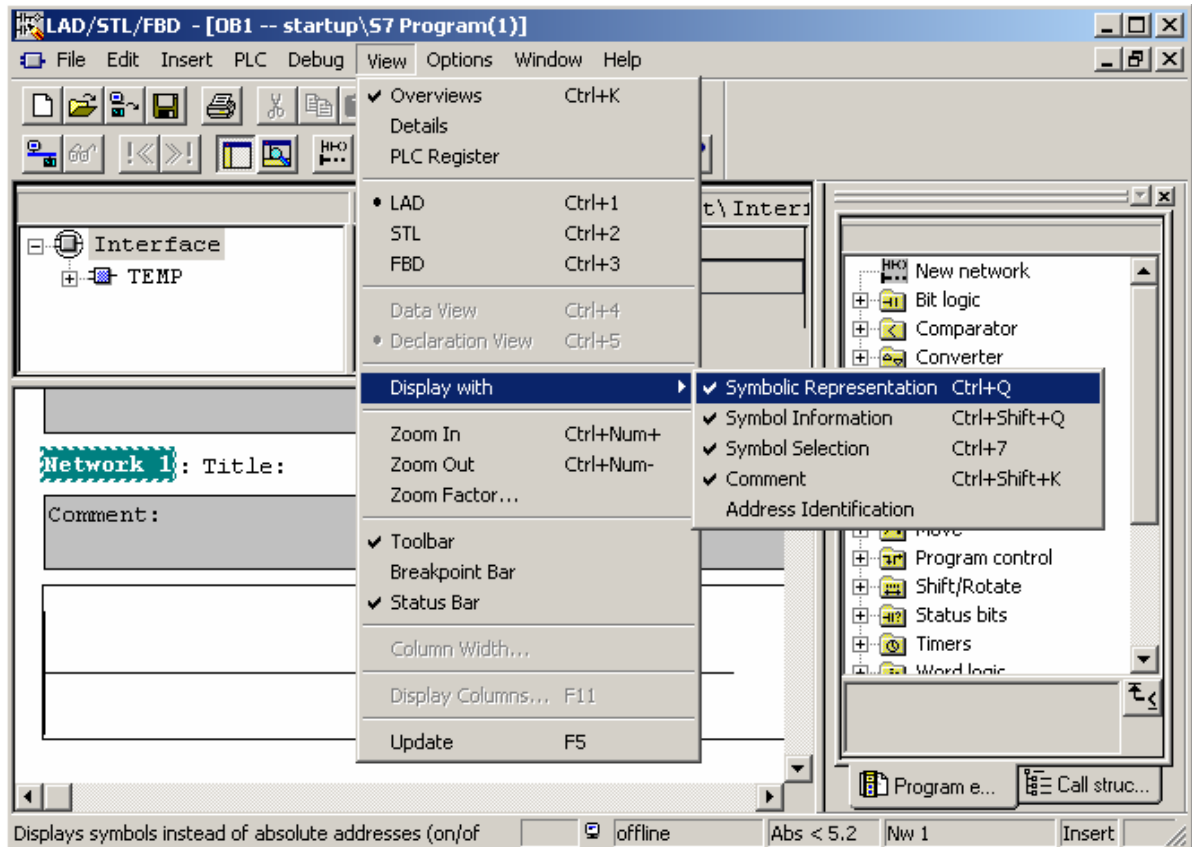


2. Zapište symboly, adresy i komentář (Comment) do tabulky symbolů (viz následující obrázek) a uložte tabulku do paměti tak, že v roletovém menu vyberete položku „Symbol table“ a v nabídce klepnete na „Save“.





3. V okně 'LAD/STL/FBD' můžeme vybrat bloky (např. OB1), se znázorněním symbolické adresy. Symbolické adresy se nám objeví, zatrhneme-li nabídku „Symbolic Representation“ v nabídce roletového menu „View“ v položce „Display with“ (viz obr.).



Znázorníme-li si potom blok OB1 ve formátu LAD (kontaktní plán), nebo FBD (funkční plán), bude u příslušných prvků viditelný popis, který jsme zadali do tabulky (viz bod 2.2).



**Poznámka:** Jestliže v okně „LAD,STL,FBD“ ukážeme myší, bez toho, že na ni klikneme, na některý z prvků, objeví se u myši informační text s příslušnými symbolickými informacemi.

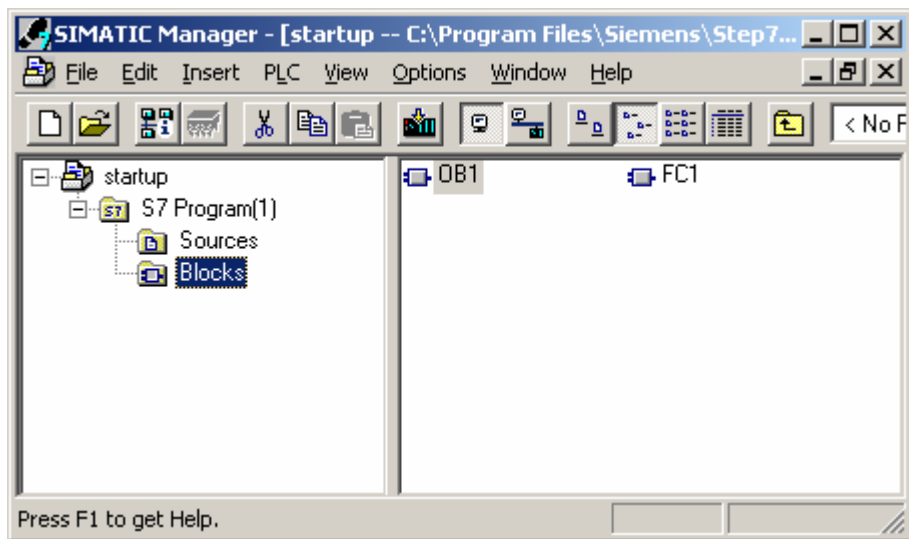
## 16.2 Uložení programu do EPROM-Memory Card v CPU



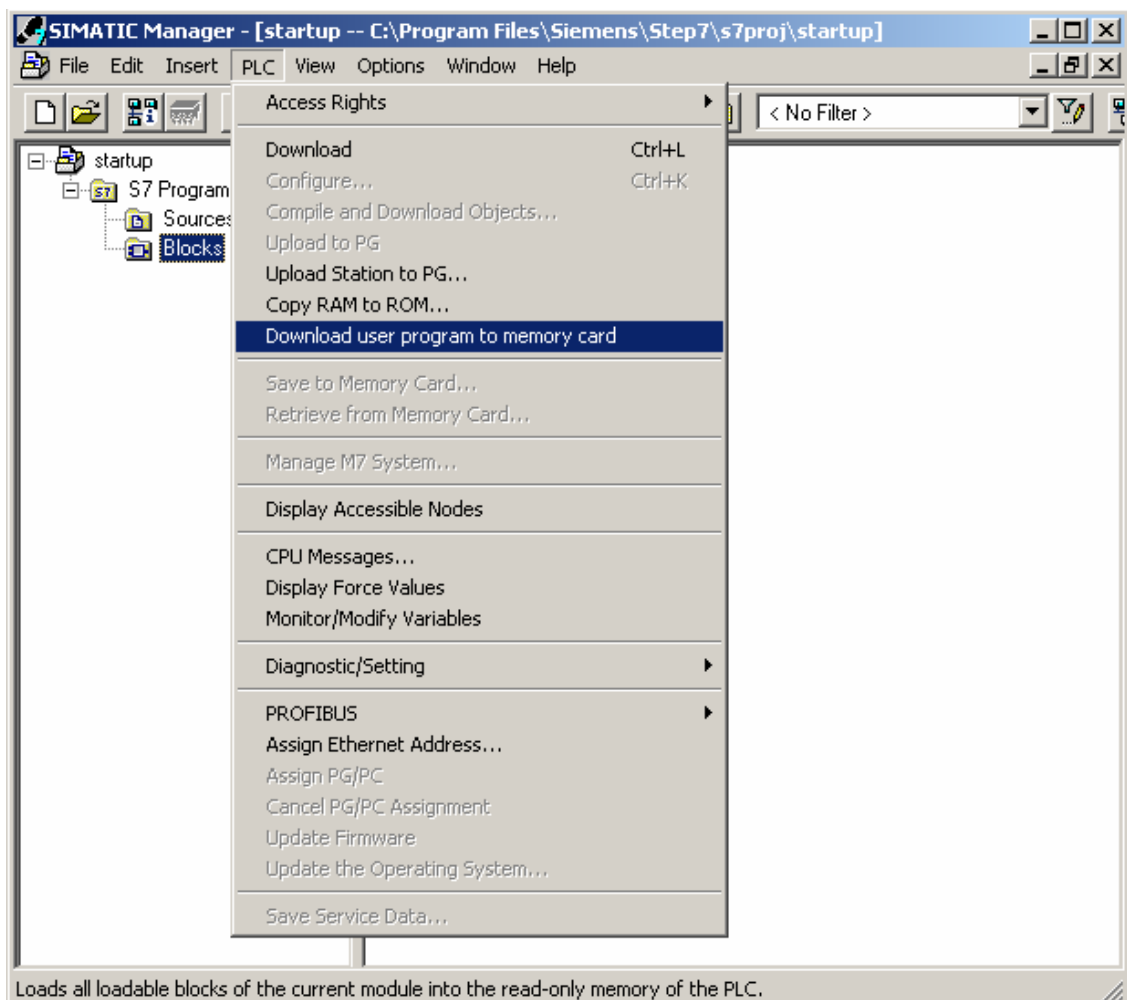
Pro SIMATIC S7-300 existuje pro všechny druhy CPU možnost přenést program na takzvaný Flash-EPROM nebo jej z něho stáhnout. Výjimku představují CPU312 IFM a CPU314 IFM, které je EPROM již integrován.

Předpokladem je, že programovací přístroj je spojený s CPU přes MPI a klíčový přepínač je na CPU v poloze STOP. Nyní postupujte takto:

1. Označte v SIMATIC Manageru složku **Blocks**.



2. Zvolte v roletovém menu položku „PLC“ a vyberte nabídku „Download user program to memory card“. Uživatelský program se uloží na paměťovou kartu (Memory Card). Viz následující obrázek:





3. Tím přeneseme všechny prvky z projektu do CPU do zde se nacházející EPROM-Memory Card .
4. Pokud chceme, aby program běžel na jiném PLC Simatic (CPU), musí se tato další CPU vypnout, zasunout paměťová karta EPROM s programem.

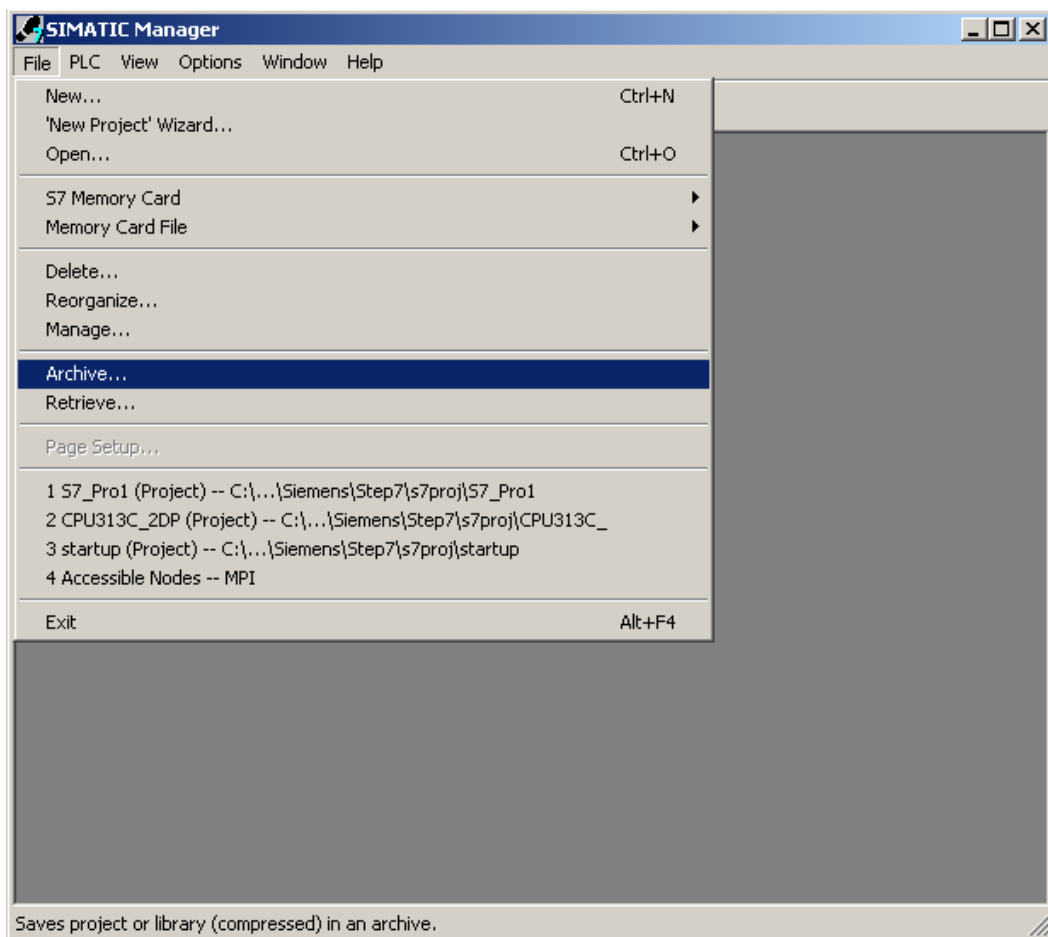
### 16.3 Ukládání a vybírání projektů z archivu



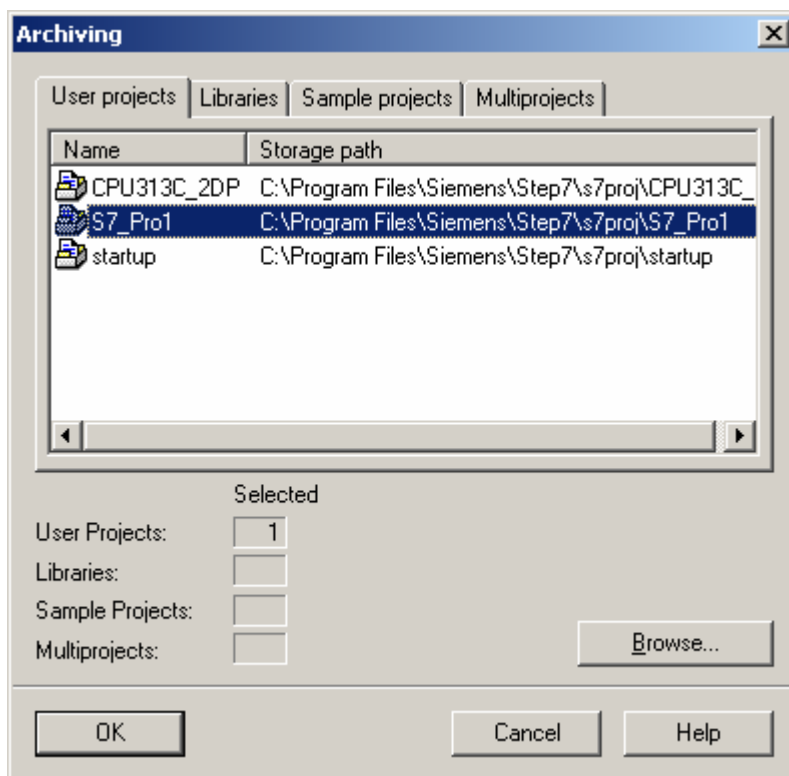
Protože některé projekty v STEP7 jsou velmi rozsáhlé a je třeba je nemazat, ale uchovat, nabízí STEP7 integrovanou funkci pro archivaci dat.

Je nutné provést následující kroky:

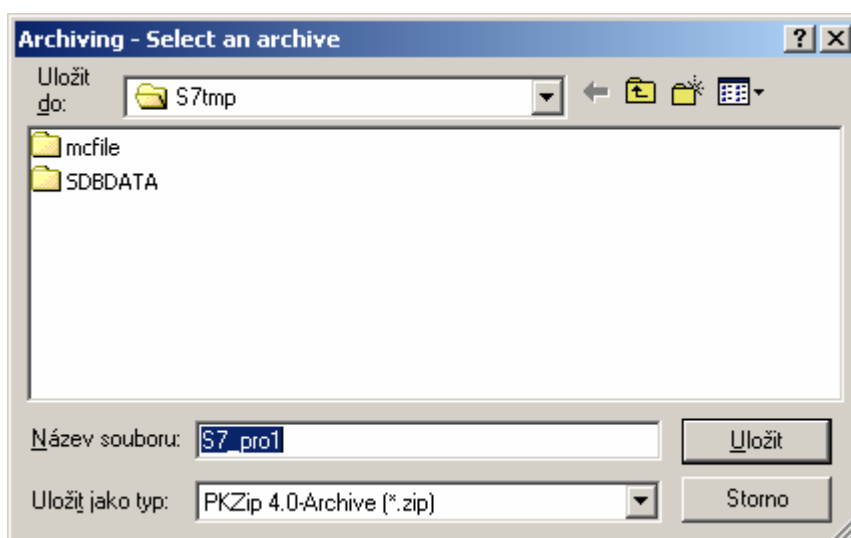
1. Hotový sestavený projekt uzavřít a v **'SIMATIC Manager'**-u zvolit funkci pro archivaci (→ SIMATIC Manager → File → Archive).



2. Objeví se okno, kde je třeba zvolit projekt (název), který se má archivovat a volbu potvrdit tlačítkem **,OK'**.



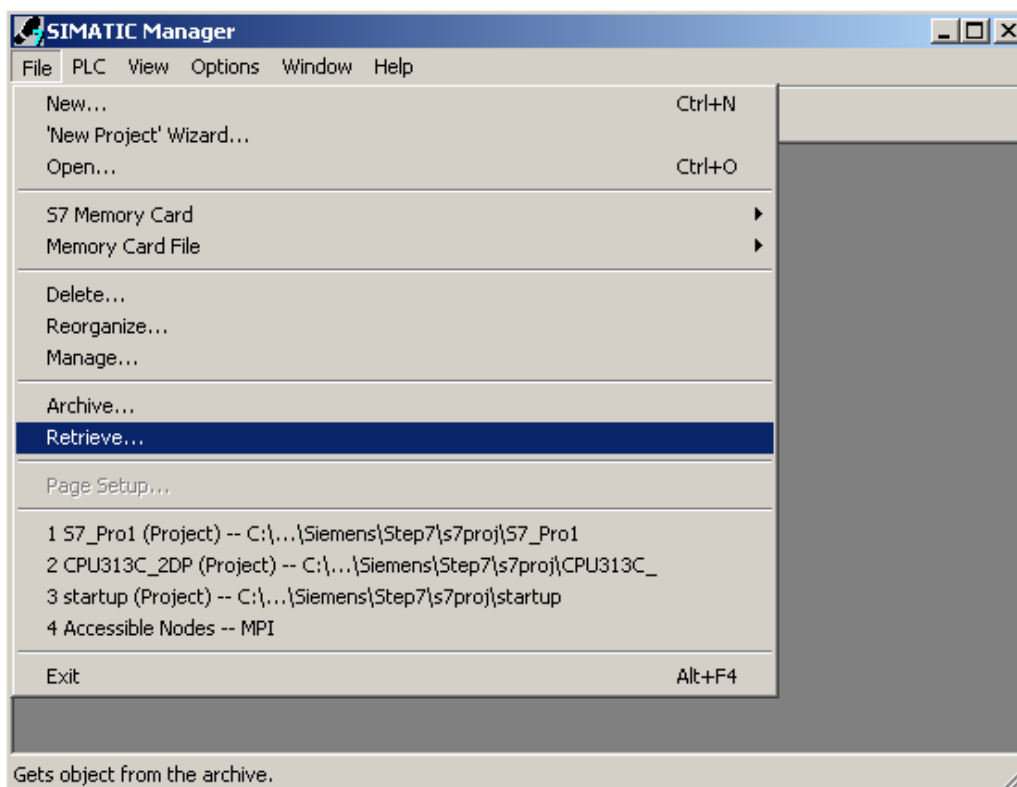
3. Dále je třeba v dalším okně zvolit název cílového archivu a pak uložit program do vybraného archivu stiskem tlačítka „Uložit“.



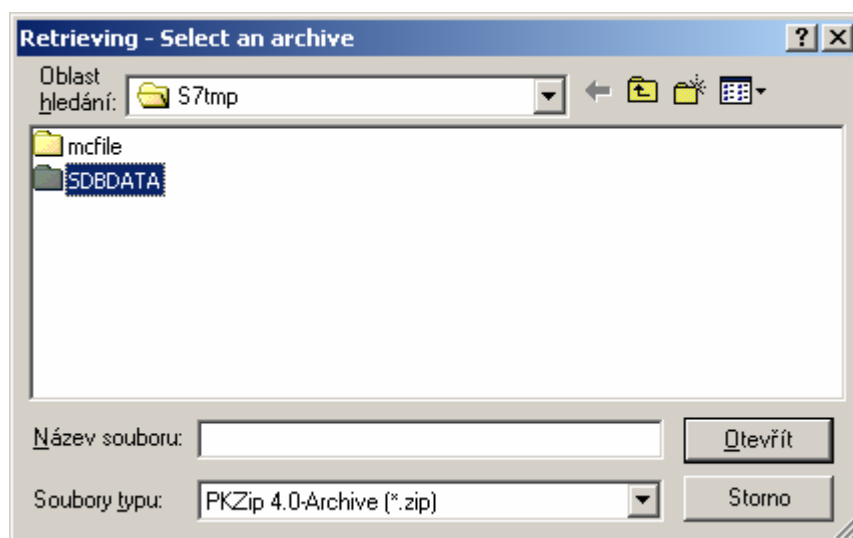




4. ‚Dearchive‘ probíhá v ‚SIMATIC Manager‘ následovně. Z nabídky roletového menu vybereme položku „File“ a z nabídky klikneme na řádku „Retrieve“.

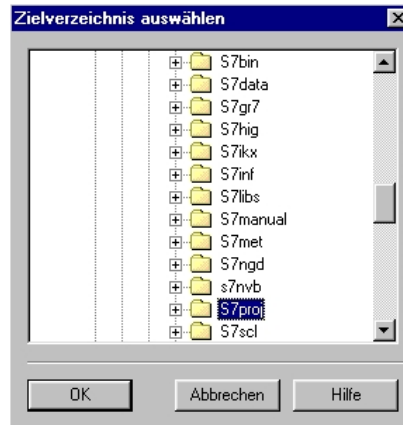


5. Zde je nutné archiv, který se má opět obnovit, zvolit v nabídkovém okně a pak otevřít kliknutím na tlačítko „Otevřít“.





6. Po otevření archivu v něm vybereme program, který chceme načíst zpět z archivu k používání (dearchivovat) a volbu potvrdíme tlačítkem ,OK'.



7. Dearchivovaný projekt (program) je možné ihned otevřít pro další zpracování.

## 17. Testovací a online funkce - úvodní slovo

### Učební cíl:

Cílem této kapitoly je naučit se ovládat nástroje, které jsou nápomocné při hledání chyb:

- testovací funkce
- online funkce

### Předpoklady:

Pro úspěšné zvládnutí této kapitoly se předpokládají následující znalosti:

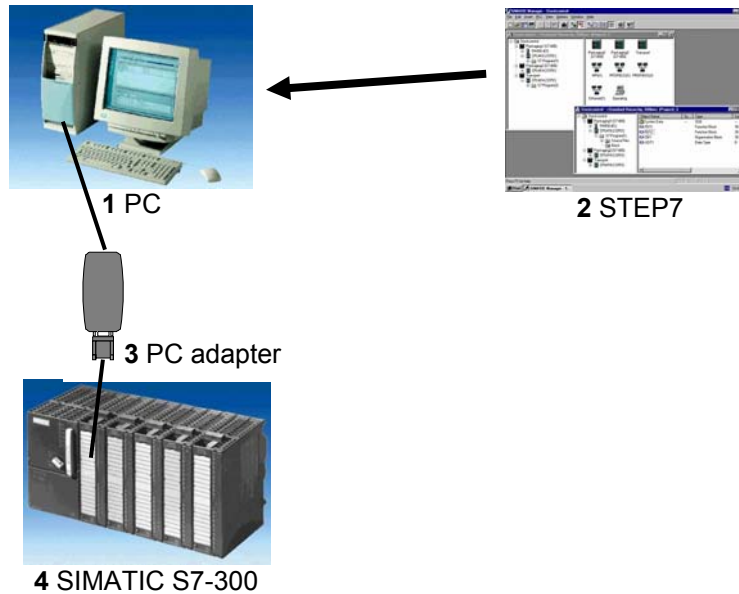
- schopnost práce s OS Windows
- základy programování SPS pomocí STEP 7

### Potřebný hardware a software

- 1 PC, operační systém Windows 95/98/2000/ME/NT4.0/XP s
  - 600MHz a 256MB RAM, volná disková paměť ca. 300 - 600 MB
- 2 Software STEP 7 V 5.x
- 3 MPI- rozhraní pro PC (např. PC-adapter, MPI karta CP5611)
- 4 PLC SIMATIC S7-300

Příklad konfigurace:

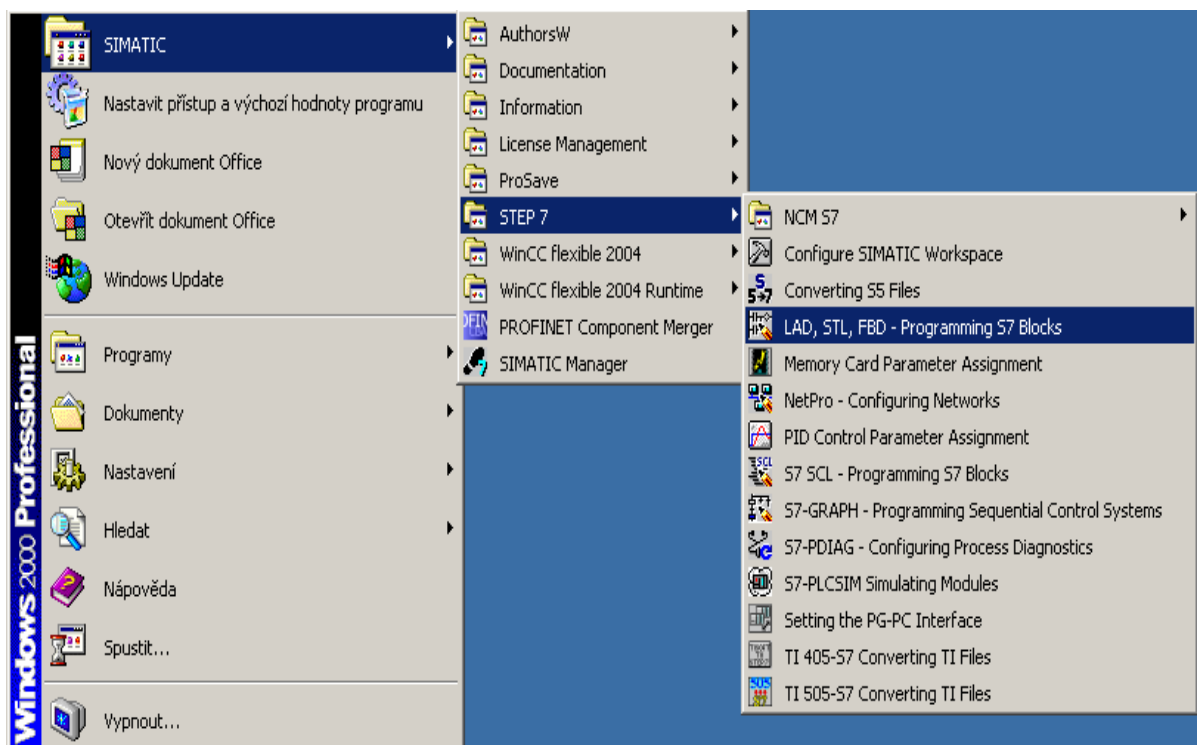
- Zdroj proudu: PS 307 2A
- CPU: CPU 31x
- Digitální vstupy: DI 16x DC24V
- Digitální výstupy: DO 16x DC24V / 0,5 A



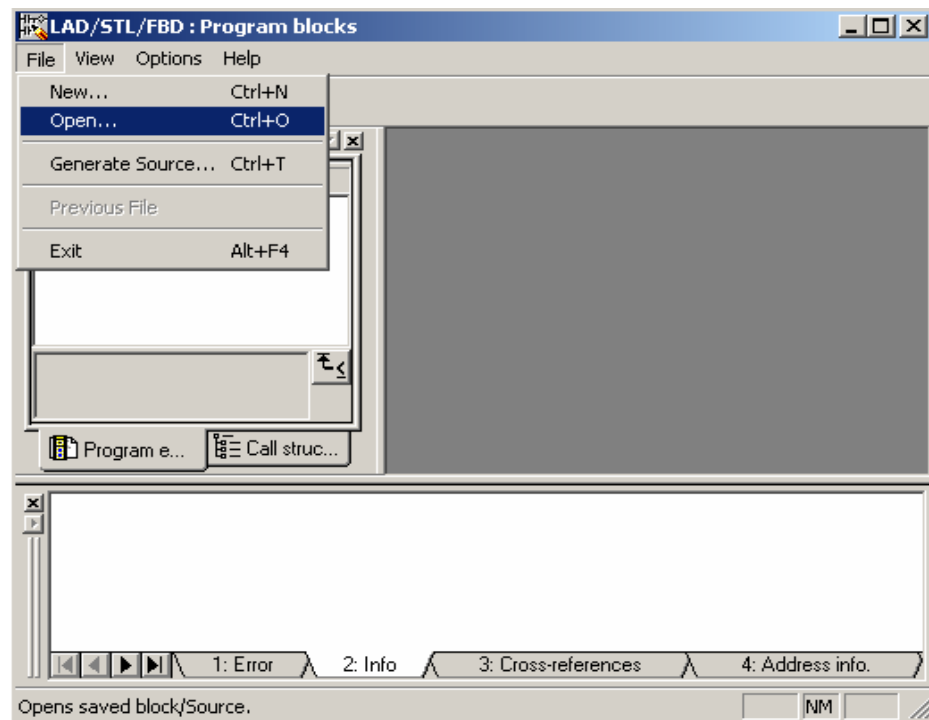
V této kapitole představujeme testovací a online funkce, které můžete testovat např. pomocí STEP 7- Projekt ‚Startup‘ z Modulu A3 - ‚Startup‘ SPS- programování pomocí STEP 7.

V programu STEP 7 máte k dispozici různé testovací a diagnostické funkce. Aby jste je mohli použít, je nutné uskutečnit následující kroky:

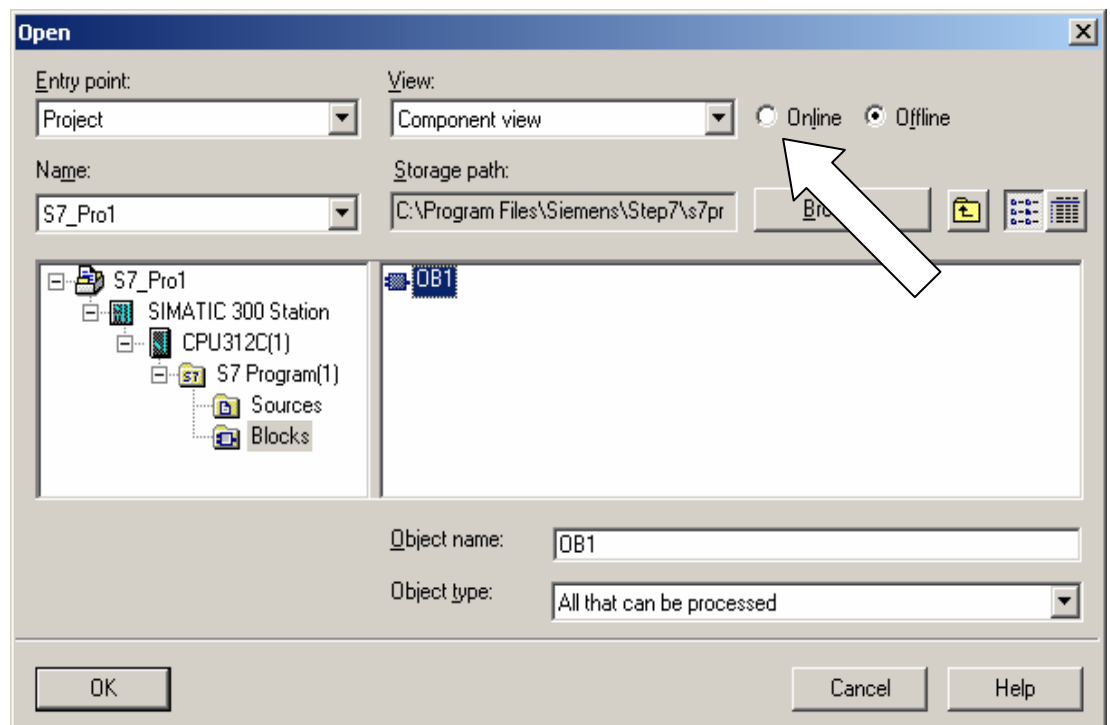
1. Jako první se otevře nástroj ‚LAD/STL/FBD‘ v nabídce hlavního panelu.



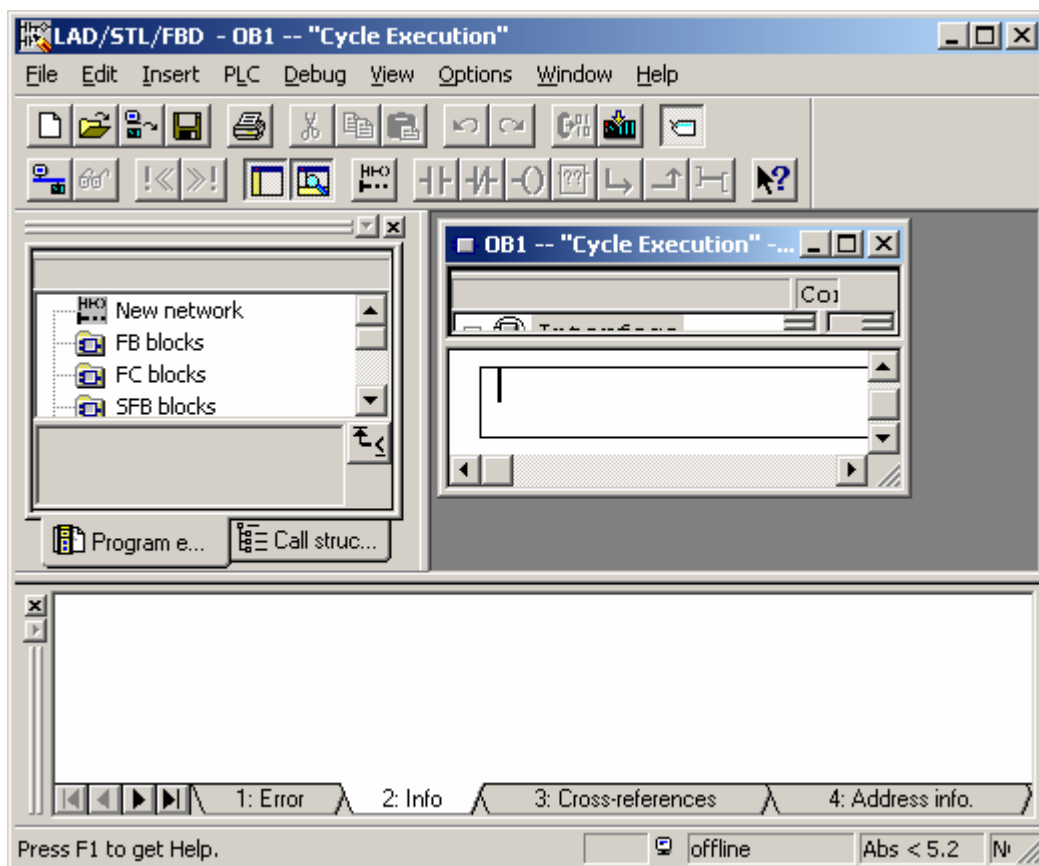
2. V následujícím okně zvolíme nabídku roletového menu **'File'** a vybereme položku **'Open'**.




3. V tabulce, která se nám otevře, označíme položku **'Online'** (viz šipka). V pravém okně vybereme blok nebo funkci, kterou chceme sledovat (zde nemáme jinou možnost než OB1).



4. Otevře se nám okno s programem v OB1 (viz obr.).



5. Monitorovací test aktuálně otevřeného bloku můžeme spustit kliknutím na ikonu , nebo přes menu „Debug“ výběrem položky „Monitor“.



**Poznámka:** Během doby, kdy je spuštěn monitorovací testovací režim, nelze v programu provádět žádné změny! Nelze ani měnit způsob zobrazení bloku (FBD, STL, LAD). Stav může být zobrazen pouze tehdy, jsou-li prováděny instrukce. Stav není zobrazen, když CPU je v režimu STOP, nebo když je blok volán.

6. Testovací režim má dva režimy provozu, které se liší svými následky na čas snímaného cyklu uživatelského programu:

- Process operation (prováděcí provoz)
- Test operation (testovací provoz)

*Prováděcí provoz* – v tomto provozu jsou testovací funkce omezeny tak, že Vámi zadané povolené navýšení snímacího času není překročeno. Testovací funkce „Breakpoint“ („Bod přerušení“) a „Krokování programu“ (Single-step program execution) nelze provést.

*Testovací provoz* – všechny testovací funkce jsou proveditelné bez omezení. Čas snímacího cyklu může být výrazně zvýšen díky aktualizaci testovací funkce „Program status“ („Stav programu“) při každém provedení smyčky.

## 17.1 Diagnostika systému

Diagnostika jsou obecné detekční a nahrávací funkce CPU. Oblast do níž se zaznamenává informace o chybě se nazývá diagnostický buffer (zásobník). Velikost zásobníku závisí na typu CPU (např. CPU 313 = 100 hlášení).

Když se vyskytne chyba nebo událost, například změna operačního režimu, stane se následující:

- do diagnostického zásobníku je vloženo hlášení s datem a časem. poslední hlášení je uloženo na začátek zásobníku. Pokud je paměť plná, starší hlášení jsou smazána.
- událost je uložena do seznamu stavu systému (System status list)
- pokud je to nezbytné, událost aktivuje patřičný chybový OB

S pomocí diagnostiky CPU mohou být identifikovány následující chyby:

- systémová chyba CPU nebo chyba modulu
- programová chyba CPU

Při řešení problémů se dělá rozdíl mezi následujícími třídami chyb:

- chyby, které způsobují zastavení CPU
- logické chyby, tj. CPU provede program, ale jeho funkce není splněna
- sporadické chyby, které se vyskytují pouze při jistých stavech systému. Ty mohou způsobit buď přechod CPU do režimu STOP, nebo se vyskytnou jako logická chyba