

PESpro

**KOMPLEXNÍ VÝVOJOVÉ PROSTŘEDÍ
NA TVORBU A LADĚNÍ APLIKACÍ
PRO AUTOMATY SYSTÉMU PES**

02/2001

The screenshot displays the PESpro development environment. The main window shows a code editor with the following code:

```

if T0>30 then Delay := T0=0
D:\PESPRO\APP\TEST\TEST302\302-3.STP
POSITION:=0
DISPLAY=" TEST MPC302: "
POSITION:=40
if MenuPos=1 then DISPLAY="Modul Z-1.pozice" ; test modulu na 1.pozici
if MenuPos=2 then DISPLAY="Modul B-2.pozice" ; test modulu na 2.pozici
if MenuPos=3 then DISPLAY="Modul D-2.pozice"
if MenuPos=4 then DISPLAY="Modul C-2.pozice"
if MenuPos=5 then DISPLAY="Modul E-2.pozice"
if MenuPos=6 then DISPLAY="Modul F-2.pozice"
if MenuPos=7 then DISPLAY="Modul G-2.pozice"
if MenuPos=8 then DISPLAY="Modul A-3.pozice" ; test modulu na 3.pozici
if MenuPos=9 then DISPLAY="Modul B-3.pozice"
if MenuPos=10 then DISPLAY="Modul C-3.pozice"
if MenuPos=11 then DISPLAY="Modul D-3.pozice"
if MenuPos=12 then DISPLAY="Modul E-3.pozice"
if MenuPos=13 then DISPLAY="Modul G-3.pozice"
if MenuPos=14 then DISPLAY=" Akust.indikace "
if MenuPos=15 then DISPLAY=" Alarm "
if MenuPos=16 then DISPLAY=" Date-Time "
if MenuPos=17 then DISPLAY=" Jas "
if MenuPos=18 then DISPLAY=" Kontrast "
if MenuPos=19 then DISPLAY=" Klavesnice "

if KBDCODE=5 and (MenuPos>1) then MenuPos=MenuPos-1
if KBDCODE=6 and (MenuPos<19) then MenuPos=MenuPos+1
if KBDCODE=4 then ItemMenu
    
```

Below the code editor, there are several windows:

- Keyboard (Klávesnice):** A list of keyboard shortcuts and their corresponding actions, such as "Beh Ctrl-F9", "Zastav Alt-S", "Restart Ctrl-F2", "Program Shift-F9", "Zásobník", "Ústupy-Ústupy", "Další Skupiny", "Sledovač", "Pes Typ", "Pes Adresa", "Pes Jméno", and "Proměnná".
- Ustupy/vystupy (Outputs/Inputs):** A table with columns for item name, value, and action.

Item	Value	Action
V22	0	Sledovač
V23	0	
V24	0	itemmenu
V25	0	KBDCODE
V26	0	KBDELAY
V27	100	Hour
V28	13	Minute
V29	59	Speed
V30	70	
V31	0	
T0	0	
T1	0	
- Zásobník (Stack):** A list of memory addresses and their values.

Address	Value
0	00001
1	00001
2	00001
3	00000
4	05548
5	00001
6	00000
7	00380
8	21932
9	05115
10	00005
11	00002
12	00001

The status bar at the bottom shows "F1 Nápověda Alt-X Konec F10 Menu Zadne MICROPEL".

PESpro
Komplexní vývojové prostředí
na tvorbu a ladění aplikací pro automaty systému PES
popis verze 1.30 (a vyšší)

uživatelská příručka - edice 02.2001
3. verze dokumentu

Změny a doplňky

proti 2. verzi dokumentu (03.2000) :
doplněny vlastnosti poslední verze software -
viz "Zatažení dvou programů do automatu"
změny v kapitolách 2, 3 a 5

PESpro © MICROPEL 1996-2001
© TRIAMOND SW FEL-ČVUT 1996-2001

všechna práva vyhrazena

kopírování publikace dovoleno pouze bez změny textu a obsahu

<http://www.micropel.cz>

1. ÚVOD	5
1.1. Plná a omezená verze PESpro	5
1.2. Jaký počítač potřebujeme?	5
2. INSTALACE	6
2.1. Instalace z distribučního CD	6
2.2. Instalační program	6
2.3. Instalace z internetu	7
2.4. Instalace na síťový server	7
2.5. Spuštění integrovaného prostředí	8
2.6. Používání integrovaného prostředí pod MS Windows®	8
2.7. Propojení s automaty	8
3. POPIS A OVLÁDÁNÍ PROSTŘEDÍ	9
3.1. Organizace souborů na disku	9
3.2. Definiční soubory a jejich význam	9
3.3. Komunikace po sériové lince	9
3.4. Pracovní plocha	10
3.5. Ovládání pracovních oken	10
3.6. Ovládání dialogů	12
3.7. Práce s projekty	14
3.8. Přeložení a zatažení programu do automatu	14
3.9. Zatažení dvou programů do automatu	14
4. TYPY OKEN NA PRACOVNÍ PLOŠE	16
4.1. Okno editoru zdrojového textu	16
4.2. Okno sledovače, "Další skupiny", "Vstupy-výstupy"	18
4.3. Změna hodnot proměnných	19
4.4. Okno zásobníku	19
4.5. Přiřazení adresy automatu oknům sledovačů	20
4.6. Okno nápovědy	20
4.7. Výpis chyb a varování	20
5. POPIS NABÍDKY PŘÍKAZŮ (MENU)	21
5.1. Nabídka podpůrných funkcí	21
5.2. Nabídka Soubor	23
5.3. Nabídka Okna	25

5.4. Nabídka Editace	25
5.5. Nabídka Zobrazení	26
5.6. Nabídka Běh	26
5.7. Nabídka Volby	27
6. SETPES - NASTAVOVÁNÍ AUTOMATŮ	29
6.1. Parametry programu SETPES	30
7. SIMPLE 2 - POPIS PROGRAMOVÁNÍ	31
7.1. Syntaxe zápisů	31
7.2. Symbolické názvy	31
7.3. Proměnné	33
7.4. Instrukční soubor	33
7.5. Textové řetězce	35
7.6. Podprogramy	36
7.7. Konfigurační direktiva NetAddr	36
7.8. Omezení překladače	36
8. SPECIÁLNÍ FUNKCE A REGISTRY	37
8.1. RESET	37
8.2. Rychlost systému	37
8.3. Časovače	37
8.4. Reálný čas	38
8.5. A/D převodník	39
8.6. STACK (zásobník)	40
8.7. Obsluha displeje a klávesnice	41
8.8. Formátování tisku číselných hodnot	43
8.9. Formát pro tisk znaků	45
8.10. Definování vlastních grafických symbolů	46
8.11. Jak funguje zobrazování	47
8.12. Seznam speciálních funkčních proměnných	49
9. FUNKCE AUTOMATU	50
9.1. Spuštění automatu	50
9.2. Obsluha proměnných	50
9.3. Popis síťové komunikace	51
9.4. Tipy pro používání sítě	51

1. ÚVOD

Integrované vývojové prostředí PESpro slouží k vytváření a ladění řídicích algoritmů pro automaty PES a MPC300. Komfortní způsob ovládání jednotlivých funkcí prostředí vede ke zpříjemnění a urychlení práce na řešeném problému. Z hlediska základních funkčních bloků je možné integrované prostředí rozdělit na několik částí:

- Editor zdrojových textů vybavený funkcí automatického odsazování a funkcí barevného zvýraznění klíčových slov zdrojového textu.
- Blok zobrazování informací tvořený sekcí pro zobrazení stavu komunikační linky, sekcí zobrazení podrobných informací o automatech připojených na linku a sekcí zobrazení aktuálních hodnot proměnných v tom kterém automatu na komunikační lince.
- Blok podpůrných funkcí pro překlad a natažení ("download") přeloženého zdrojového textu, označování chybových řádků zdrojového textu a manažer projektů.
- Možnost spouštění dalších samostatných vývojových nástrojů z PESpro.

Pozn.: Celé prostředí PESpro a s ním související nástroje jsou aplikace DOS!

Uvedený seznam funkcí tvoří systém, pomocí něhož jsou sloučeny základní kroky (psaní zdrojového textu, překlad, "download" a sledování obsahu proměnných) vývoje aplikace pro všechny průmyslové automaty fy MICROPEL pod jednotné ovládání a do jednotného prostředí.

1.1. Plná a omezená verze PESpro

Integrované prostředí PESpro je možné používat v omezené verzi "START" naprosto zdarma. Jediné omezení verze "START" se vztahuje na velikost výsledného přeloženého kódu pro automat (maximálně 5 kB). Pokud uživatel zakoupí plnou verzi vývojového prostředí, dostává s programem i identifikační kód. Po zadání tohoto textu do okénka registrace programu (při instalaci programu) se prostředí nainstaluje v plné verzi. Díky tomu si může stávající uživatel stáhnout např. z internetu novou verzi programu a po zadání tohoto "hesla" instaluje automaticky plnou verzi.

1.2. Jaký počítač potřebujeme?

Minimální požadavky jsou tyto: počítač s procesorem třídy 486 nebo vyšším a pamětí RAM o velikosti alespoň 8MB s operačním systémem DOS nebo Windows 95 resp. 98. **Prostředí PESpro nelze provozovat pod MS Windows NT !** Přestože je prostředí plně ovladatelné z klávesnice, doporučujeme instalovat myš. Prostředí pro svou instalaci potřebuje alespoň 10MB volného prostoru na pevném disku (pokud se instalují kromě základního prostředí ještě další nástroje).

2. INSTALACE

2.1. Instalace z distribučního CD

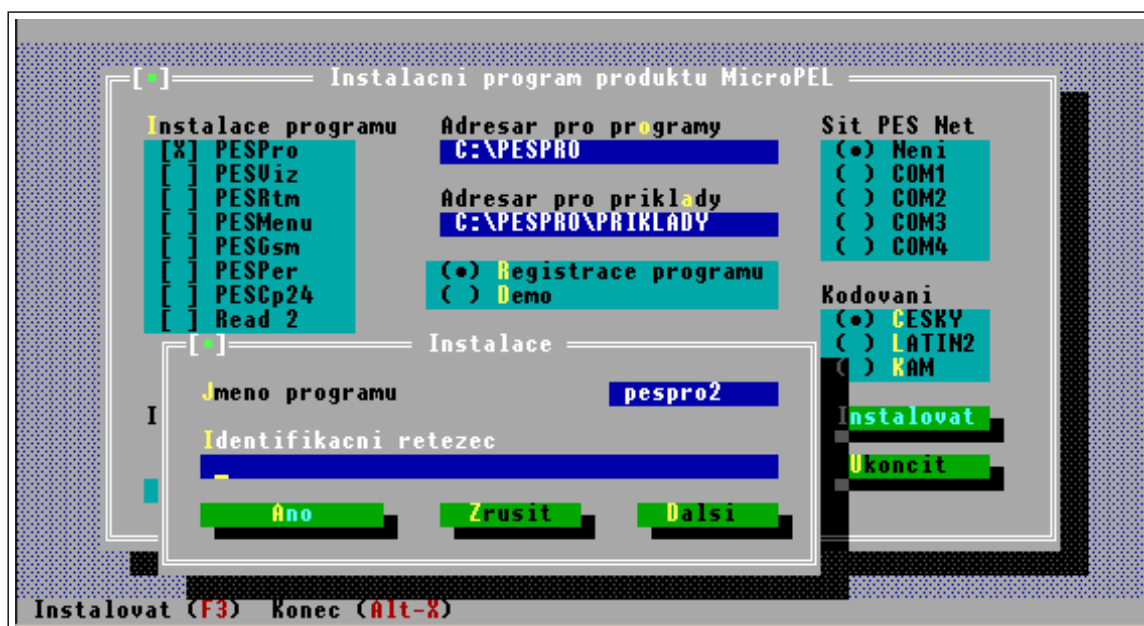
Na většině počítačů (pokud na nich běží Microsoft Windows[®] 95 nebo 98) se CD po zasunutí do mechaniky samo spustí a objeví se úvodní dialog. Nespustí-li se CD samo, je třeba ručně spustit ("otevřít") soubor **autorun.exe** na CD disku. Z nabídky úvodního dialogu zvolíme SOFTWARE a tím se spustí instalační program.

POZN.:

Instalační program je, stejně jako PESpro a související nástroje (kromě PESdde pro Windows), aplikace DOS. Při jeho spuštění pod Windows se tedy otevře okno DOS aplikace a v něm běží instalátor (stiskem **Alt-Enter** jej můžeme přepnout na celou obrazovku a naopak). Na počítačích, provozovaných výhradně pod systémem DOS, je třeba instalační program spustit ručně. Zpravidla je možno jej nalézt na této cestě: **DATA\SOFTWARE\INSTALL.EXE**

2.2. Instalační program

Před započítím vlastní instalace na pevný disk nabídne instalátor dialog s volbou možností instalace. Dialog je intuitivní, plně ovladatelný myší i klávesnicí.



Výčet dostupných programů k instalaci je uveden ve sloupci vlevo. U každé položky lze kliknutím myši zrušit/povolit instalování příslušného komponentu. Standardně jsou po spuštění instalátoru zaškrtnuty všechny položky.

Ve střední části jsou nabízeny adresáře, kam bude instalace jednotlivých souborů provedena. Názvy všech adresářů vč. určení disku lze změnit. Pokud adresář zadaného jména neexistuje, bude při instalaci vytvořen (zadaný disk existovat musí).

Adresář pro programy

Sem se nakopírují všechny programové a konfigurační soubory prostředí PESpro a všech vybraných podpůrných nástrojů.

Adresář pro příklady

Sem se nakopírují datové soubory, zdrojové texty apod. od všech dostupných příkladů pro prostředí PESpro i pro všechny vybrané nástroje. Standardně je tento adresář nabízen jako podadresář adresáře pro programy.

Cílový adresář PES DDE

Sem se kopírují soubory související s programy DDE rozhraní pro Windows - PESdde. Není-li zaškrtnuta instalace rozhraní PESdde, nemá tato položka význam.

Volba mezi plnou a START verzí

Uprostřed dialogu je přepínač s volbou mezi instalací verzí START a verzí plných. Tento přepínač má vliv pouze na instalaci těch programů, které existují ve verzi START (volná, bezplatná verze) a v plné verzi (placené). Pokud je přepínač v poloze "plná verze", bude instalátor po instalaci souborů na pevný disk vyžadovat vyplnění jména programu a identifikačního klíče (tyto údaje dostává uživatel při zakoupení plné verze).

Výběr COM portu a kódování diakritiky

Vpravo nahoře lze zaškrtnout jednu z možností COM1 až COM4 (nebo žádný). Jedná se o číslo sériového portu, na kterém bude připojen automat (síť automatů). Vpravo dole lze zvolit způsob kódování české diakritiky (závisí na typu podpory českých fontů pod DOS). Pro českou instalaci MS Windows je vhodné zvolit LATIN2, volba CESKY je vhodná pro systém bez podpory diakritiky (jen písmena bez háčeků a čárek).

Tyto volby se projeví pouze v podobě /COM parametru a parametrů pro diakritiku v BAT souborech, automaticky vygenerovaných instalátorem pro snadné spouštění nainstalovaných programů. Lze je tedy kdykoliv přepsat. Popis parametrů viz odst. "**Spuštění integrovaného prostředí**".

2.3. Instalace z internetu

Veškeré programové vybavení (v aktuálních verzích) je dostupné po internetu na adrese <http://www.micropel.cz>.

V oddílu software je možno stáhnout buď samostatné programy, nebo balíky programů. Stažený soubor je de-facto instalační program s příslušnými zabalenými komponenty. Po jeho spuštění se objeví dialog instalačního programu (viz výše).

2.4. Instalace na síťový server

Postup instalace je stejný, s tím rozdílem, že jako cílový disk pro instalaci zvolíme některý ze síťových disků. Vzhledem k tomu, že prostředí využívá svůj domovský adresář pouze pro čtení, je možné nastavit po skončení instalace pro tento adresář atribut Read Only pro všechny uživatele.

2.5. Spuštění integrovaného prostředí

Integrované prostředí spouštíme vždy z pracovního adresáře (tedy z adresáře, kde pracujeme na zdrojových textech aplikace). Příkazový řádek DOS může vypadat např. takto:

C:\PESPRO\PESPRO.EXE /LAT2 /COM3

Pokud spouštíme prostředí dávkovým souborem PES.BAT (který je vygenerován instalačním programem), je vždy jako pracovní nastaven podadresář PRIKLADY.

Příkazový řádek může obsahovat následující parametry s tímto významem :

- /?** výpis nápovědy k parametrům pro spuštění programu
- /COMx** nastavení sériového komunikačního portu, kde x je číslice od 1 do 4
- /US** prostředí nebude užívat znaky diakritiky tj. bude psát CESKY
- /LAT2** prostředí použije pro výstupy na obrazovku kódování podle tabulky LATIN2
- /KAM** prostředí kóduje výstupy na obrazovku podle tabulky bří Kamenických

Není-li specifikován parametr /COM, bude prostředí spuštěno bez možnosti komunikace se sítí automatů po sériové lince. Prostor bez komunikace lze použít například na psaní a překlad zdrojových textů, na první seznámení s programem apod.

2.6. Používání integrovaného prostředí pod MS Windows [®]

Po instalaci prostředí je vhodné vytvořit na pracovní ploše Windows zástupce. Ve vlastnostech zástupce je pak možno vypsát výše zmíněný příkazový řádek s příslušnými parametry a pracovní adresář. Můžeme vyrobit i několik zástupců s různými cestami do pracovních adresářů a různými parametry.

Prostředí a příp. související další nástroje je možné spustit pod Windows i ve více exemplářích najednou, je třeba však respektovat základní logické omezení :

Určitý komunikační port (COM) dostane přidělen jen ta aplikace, která o něj požádá jako první. Další aplikace vyžadující tento port již na něj přístup nemají!

Vyskytnou-li se potíže s komunikací po sériové lince, doporučujeme v nastavení systému Windows ve Správci zařízení zaškrtnout u dotyčného COM portu volbu "Vypnout v tomto hardwarovém profilu".

Prostředí PESpro nelze provozovat pod Windows NT !

2.7. Propojení s automaty

Aby bylo možné zatahovat programy do automatů, sledovat za běhu proměnné v automatech apod., je nutné propojit automaty na některý sériový COM port počítače přes převodník rozhraní RS485/RS232 - **PES-CA1**. Číslo použitého COM portu je třeba uvést na příkazovém řádku při spuštění programu (viz výše).

3. POPIS A OVLÁDÁNÍ PROSTŘEDÍ

Integrované prostředí PESpro podporuje a je postaveno pro práci na projektech. Pod pojmem projekt je zde míněno zpracování zadání algoritmu pro jeden či více automatů provozovaných na lince. Pro specifikovaný projekt ukládá prostředí vždy úplné nastavení volitelných parametrů, nastavení pracovní plochy včetně všech otevřených oken a k veškerému zpracování dat používá vždy adresář projektu.

3.1. Organizace souborů na disku

Prostředí vytváří v pracovním adresáři několik typů souborů. Jedná se o soubory zdrojových textů (obvykle s příponou .STP), soubory pro "download" (obvykle s příponou .DNL), soubor s nastavením barev a parametrů (PES.CFG), soubor s nastavením pracovní plochy (PES.DSK) a soubory s nastavením a definicí projektů (soubory s příponou .PRJ).

3.2. Definiční soubory a jejich význam

Definiční soubory (*.DEF) jsou určeny ke specifikaci výstupů a výstupů speciálních registrů a funkčních bloků daného automatu. Musí být uloženy v domovském adresáři prostředí. Specifikace těchto speciálních proměnných jsou v souborech uloženy v textové formě a je tedy možné je kdykoliv změnit. Tento zásah do definičních souborů však zásadně nedoporučujeme. V domovském adresáři prostředí musí být vždy instalován soubor GENERIC.DEF, který obsahuje definice speciálních proměnných základního hypotetického automatu. V tomto souboru hledá prostředí jména proměnných v okamžiku kdy není k dispozici definiční soubor pro daný typ automatu. Jména souborů odpovídají typům vyráběných automatů a podle těchto jmen se prostředí orientuje. Z tohoto důvodu tedy opět nedoporučujeme měnit jména těchto souborů. Je-li to nezbytně nutné pak je možné definiční soubor vytvořit či upravit. Před tímto zásahem doporučujeme konzultovat úpravy u výrobce automatů popř. vytvoření definičního souboru zadat.

3.3. Komunikace po sériové lince

Po spuštění prostředí se program pokusí navázat spojení s automaty na komunikační lince přes komunikační port specifikovaný parametrem příkazové řádky (např. /COM3). Pokud na příkazovém řádku nebyl zadán parametr pro nastavení určitého komunikačního portu, prostředí se spustí bez podpory komunikace a o žádnou komunikaci s automaty se nebude ani pokoušet. Tento režim používáme tehdy, když potřebujeme jen "na sucho" napsat a přeložit program anebo nemáme k počítači momentálně připojen žádný automat. Tento režim je na stavovém řádku nahoře vpravo indikován nápisem **Linka vyřazena...**

Pokud byl zadán na příkazovém řádku komunikační port, po spuštění programu se inicializuje komunikace se sítí automatů a prohledávají se všechny adresové pozice (max. 31 automatů). Tato fáze je indikována na stavovém řádku nahoře vpravo nápisem **Inicializace...** Po jejím skončení (trvá typicky několik sekund) se objeví buď stav **Linka připojena** (pokud je vše v pořádku a na lince se ozval alespoň jeden automat PES) nebo stav **Linka bez odezvy** (pokud na lince nebyl nalezen žádný automat).

Jsou-li potíže s komunikací a na počítači běží systém MS Windows, doporučujeme opatření popsané v kapitole "Používání integrovaného prostředí pod MS Windows".

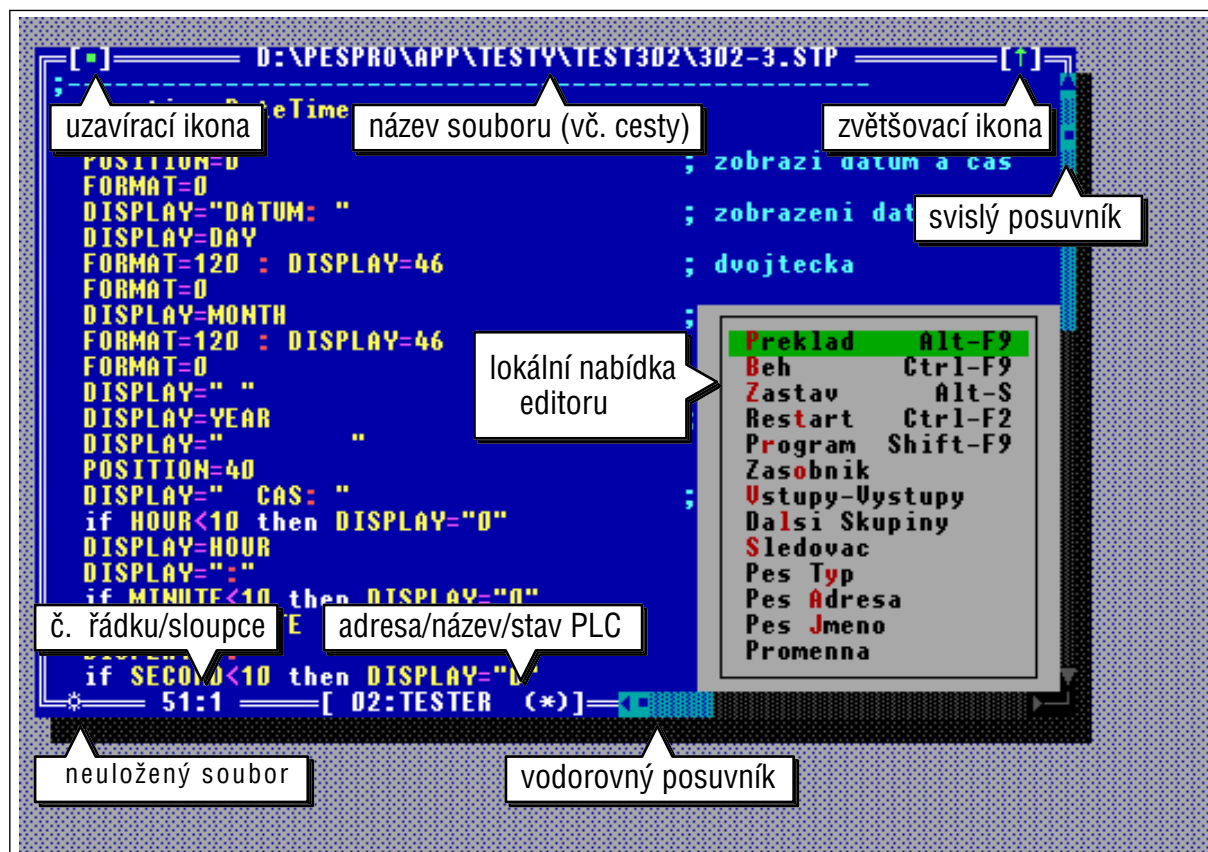
3.4. Pracovní plocha

Po inicializaci se objeví obrazovka pracovní plochy s nastavením barev a uspořádáním pracovní plochy ve tvaru v němž bylo naposledy prostředí v tomto adresáři opuštěno (pokud je ovšem v nastavení Volby/Prostředí povoleno ukládání těchto informací).

Pracovní plocha je tvořena horní lištou s nabídkou příkazů, spodní stavovou lištou združující vybrané příkazy pro ovládání prostředí a zbylou plochou obrazovky určenou pro zobrazení vyčítaných a zpracovávaných údajů, pro editování zdrojových textů atd.. Horní lišta nabídky je doplněna o indikaci stavu linky a výpis systémového času počítače. Spodní tj. stavová lišta obsahuje mimo vybrané příkazy výpis jména aktuálně otevřeného projektu a název vlastníka plné verze programu (pokud se nejedná o verzi START). Pomocí nastavení prostředí (viz dále) je možné zvolit jedno ze dvou rozlišení při výpisu na pracovní plochu - 25 nebo 50 řádků.

3.5. Ovládání pracovních oken

Na ploše mezi horní a dolní lištou (tzv. "desktop") je možno otevřít libovolné množství pracovních oken. V těchto oknech pak programátor realizuje veškerou vývojovou činnost (píše a překládá zde zdrojové texty programů pro automaty, v oknech sledovačů pak může vyčítat hodnoty proměnných v automatech za plného běhu). Každé okno je na svém obvodu vybaveno indikačními prvky a ovládacími prvky pro ovládání myší. Aby se tolik oken mohlo na plochu vejít, mají možnost se překrývat. Okno představuje obecně pouhý výřez z plochy zobrazovaných dat nebo psaného textu. V aktivním okně je možno rolovat horizontálním i vertikálním směrem jak myší tak klávesnicí. Při psaní textu roluje obsah v okně automaticky. Aktivní okno (tedy to se kterým momentálně programátor pracuje) je vždy jen jedno a je vždy úplně nahoře na ploše nad všemi ostatními okny. Manipulaci s okny (zavírání, otvírání, přemisťování, zvětšování, rolování v okně atd.) je možno provádět z klávesnice (kombinací funkčních kláves) i myší. Veškeré akce pro manipulaci s okny lze rovněž vyvolat také z menu **Okna** (viz dále). K oknu lze vyvolat též tzv. lokální nabídku, což je krátké menu funkcí a příkazů, použitelných v souvislosti s tímto oknem.



Aktivace okna (přesunutí na vrch pracovní plochy)

Myš Klikneme do okna (ukážeme tam myš a klepneme levým tlačítkem myši)

Klávesnicí Stiskem kláves **Alt-0** vyvoláme seznam oken a vybereme ze seznamu, nebo stiskem kláves **F6** resp. **Shift-F6** můžeme cyklicky procházet okny dopředu nebo dozadu.

Přemístění okna

Myš Uchopíme okno za horní lištu (tedy nastavíme tam ukazatel myši, stiskneme a držíme levé tlačítko), přetáhneme na žádané místo a pustíme (uvolníme tlačítko myši).

Klávesnicí Vybereme aktivní okno. Stiskneme klávesy **Ctrl-F5**, okno změní barvu, šipkami je dostaneme kam je třeba a stiskneme Enter.

Změna rozměru okna

Myš Uchopíme okno za pravý dolní roh a zvětšíme / zmenšíme jak je třeba.

Klávesnicí Vybereme aktivní okno. Stiskneme klávesy **Ctrl-F5**, okno změní barvu, současným stisknutím Shift + šipek nastavíme rozměr a stiskneme Enter.

Roztažení okna na celou plochu

Myš Klikneme myš na zvětšovací ikonu vpravo nahoře, další kliknutí na šipku vrátí okno do původního rozměru.

Klávesnicí Klávesou **F5**. Dalšíme stiskem **F5** vracíme okno do původního rozměru.

Zavření okna

Myší Klikneme myší na uzavírací ikonu vlevo nahoře.

Klávesnicí Stiskem kláves **Ctrl-F3**.

Pohyb v okně

Myší Vpravo dole a na pravém boku okna jsou posuvníky se šipkami na koncích. Tažením posuvníku rolujeme obsahem okna hrubě, tisknutím šipek jemně.

Klávesnicí Šipkami nebo klávesami **PageUp/PageDown**. Ctrl-PageUp resp. PageDown přecházejí na začátek resp. konec obsahu okna.

Vyvolání lokální nabídky

Myší Ukážeme myší kamkoli do aktivního okna a stiskneme **pravé** tlačítko myši.

Klávesnicí Kombinací kláves **Ctrl-Enter**.

3.6. Ovládání dialogů

Veškerá nastavení a změny parametrů, proměnných apod. se provádějí v tzv. dialogových oknech. Jsou to v podstatě jakési formuláře s širokou paletou možností pro zadávání a editování informací. Kromě editačních položek obsahuje dialog alespoň jedno výkonné tlačítko, kterým se na závěr provede nějaká akce (uložení informací, stornování změn atd.). Existují i dialogová okna (informační), kde nelze nic editovat, pouze stisknout potvrzovací tlačítko na zavření dialogu. Tato okna slouží jen jako hlášení pro uživatele.

Některé dialogy mají vpravo vedle okénka pro vstup textu ještě malou ikonu paměti položek (tvaru "šipka dolů") po jejímž stisku se zobrazí paměť již zadávaných textů.

Pokud jsou v dialogovém okně nějaké seznamy, ze kterých je možno vybrat položku (například seznam souborů v dialogu pro otevření souboru, nebo výše zmíněná paměť položek apod.), většinou mají u sebe posuvník, aby bylo možno seznamem rolovat (rolovat lze samozřejmě i z klávesnice pomocí šipek, nebo PageUp, PageDown).

Obsluha myší

Je jednoduchá. U přepínačů nebo zaškrťovacích seznamů stačí pro změnu stavu kliknout na příslušnou ikonu. U editorů hodnot je třeba kliknout do příslušného okénka a klávesnicí vypsát požadovaný text. U některých editorů hodnot je vpravo ještě ikona pro rozbalení seznamu. Po jejím stisknutí se rozbálí seznam s hodnotami použitelnými pro daný editor. Po vyplnění dialogu stačí stisknout některé z funkčních tlačítek.

Obsluha klávesnicí

Klávesou **TAB** se pohybujeme mezi jednotlivými objekty (přepínače, zaškrťovací seznamy, editory hodnot, tlačítka ...). Kromě **TAB** můžeme též použít kombi-

naci **Alt-písmeno** (nápisy příslušných objektů mají vždy zvýrazněné jedno písmeno, kterým lze objekt vybrat). Šipkami nahoru a dolů se lze pohybovat v rámci jednoho objektu (přepínače a zaškrtačací seznamy), položku v zaškrtačacím seznamu zaškrtneme klávesou **SPACE** (mezerník). Klávesou **Enter** stiskneme aktivní tlačítko (je zvýrazněné) a zároveň opustíme dialog, Aktivní tlačítko zvolíme rovněž klávesou **TAB** nebo kombinací **Alt-písmeno**. Nachází-li se kurzor v okénku u něhož je paměť položek, stiskem šipky dolů zobrazíme paměť položek.

Ukázky některých dialogů

Přidávání proměnné do sledovače

editor na vstup textu

paměť na položky

přepínač

další nabídka

uzavírací ikona

Nastavení parametrů prostředí

přepínač (lze aktivovat jen jednu položku)

zaškrtačací seznam (lze aktivovat každou položku)

Otevření souboru

editor na vstup textu

seznam položek

paměť na položky

funkční tlačítka

posuvník na procházení seznamem

Vytváření programů pro automaty

K vytváření nebo úpravě programu pro automaty (tzv. zdrojového textu) slouží editor zdrojových textů. Bližší popis okna editoru je v následující kapitole.

Soubor zdrojového textu je možno otevřít/uložit/kopírovat pomocí příkazů z nabídky menu "Soubor".

3.7. Práce s projekty

Při tvorbě více programů pro více automatů spojených např. v síti, nebo pro souběžnou práci na různých projektech, je užitečné použít tzv. "projekt". Ten lze opět otevřít/uložit/kopírovat pomocí příkazů z nabídky menu "Soubor". Projekt je speciální dialogové okno, obsahující seznam souborů k přeložení. Výhodou projektu je, že při překládání nebo zatahování programů do automatů porovnává aktuálnost jednotlivých zdrojových textů a zpracovává vždy všechny položky uvedené v seznamu. Kromě toho vždy při uložení nebo uzavření projektu a rovněž při ukončení práce s prostředím PESpro se k aktuálnímu projektu uloží i nastavení prostředí a všech oken na pracovní ploše, včetně soupisů proměnných v oknech sledovačů (viz dále).

Chceme-li se tedy vrátit k některému již existujícímu projektu, stačí tedy jen otevřít příslušný projekt. Celá pracovní plocha prostředí (tzv. desktop) včetně všech oken a soupisů proměnných ve sledovačích se obnoví do podoby, ve které byl tento projekt naposledy opuštěn.

Editace seznamu souborů projektu - viz kap. "POPIS NABÍDKY PŘÍKAZŮ", stať "Dialog Editace Projektu".

3.8. Přeložení a zatažení programu do automatu

Před zatažením programu do automatu je třeba jej nejprve přeložit (do tzv. strojového kódu, srozumitelného pro procesor v PLC). Překlad provedeme buď vyvoláním lokální nabídky v textovém editoru se zdrojovým textem (viz dále), nebo vybereme z menu Běh na horní liště položku Překlad (přeloží se vždy soubor v právě vybraném okně zdrojového textu). Po opravení všech případných chyb nahlášených překladačem je možno zvolit položku Program - tím se spustí zatažení programu do PLC po sériové lince (tzv. "download").

POZN.:

Prostředí před zatažením programu automaticky kontroluje časovou konzistenci souborů. Pokud byl změněn zdrojový text a neproběhl překlad, je volán nejprve překladač a pak se teprve provádí download.

3.9. Zatažení dvou programů do automatu

Do automatů s verzemi firmware, které tuto funkci podporují (např. u mini-automatu PES-M66 ji nelze použít vůbec), je možno zatáhnout dva programy za sebou.

Toto umožňují automaty PES-K a MPC300 se speciálními verzemi firmware. Tím se rozšíří prostor pro délku uživatelského programu ze standardních 30 kB na 60 kB.

Je jen třeba v prostředí PESpro označit v projektu zdrojové soubory, které budou spojeny a budou se zatahovat společně do jednoho automatu.

Programy spolu sdílí veškeré datové prostory, tedy všechny registry (I,O,D,W), všechny bity (X,Y,M,B) a celý zásobník. Je tedy třeba dát pozor na to, aby nedošlo ke kolizím. Prostředí PESpro před zatažením spojené dvojice programů kontroluje definované symboly a pokud se překrývají různé definice z obou programů, vydá varování. Protože závadnost takového překrytí může posoudit jedině programátor, jsou nalezené kolize vypsány a programátor musí rozhodnout, zda provést download.

Kontrolují se pouze symbolické názvy, a to takto:

Pokud je v obou programech stejně umístěná proměnná se stejným symbolickým názvem, není to považováno za chybu (jedná se patrně o proměnnou, která je v obou částech programu používána stejným způsobem).

Pokud je v obou programech stejně umístěná proměnná s různým symbolickým názvem, je to chyba (lze předpokládat, že se jedná o různé proměnné s různou funkcí a tudíž by měla být každá umístěna jinde).

Při tvorbě dvou spolupracujících programů pro automat je tedy třeba si jednak uvědomit časovou následnost běhu programů (programy jsou jakoby "přilepené" za sebou, přičemž hlavní soubor je jako první), jednak rozvrhnout rozdělit celý komplex na dvě samostatné části. Každý program může totiž využívat jen ty podprogramy, které jsou v něm definovány. Je-li třeba použít v obou programech stejný podprogram, je nutné jej zkopírovat do obou souborů (to je dáno tím, že každý soubor zdrojového textu je překládán zvlášť).

V automatu běží hlavní programová smyčka tak, že nejprve je spuštěn tzv. hlavní soubor, pak vedlejší soubor a pak teprve probíhá čtení/zápis vstupů a výstupů.

Spojení programů pro společné zatažení

Aby bylo možné zatáhnout do PLC dva programy, je nutné v PESpro založit projekt (spojení programů pro společný download se definuje v okně projektu). Nejprve je třeba vytvořit oba programy (zdrojové texty) pro automat (resp. de-facto dvě samostatné poloviny celkového programu). Oba soubory se přidají do projektu. V okně editace projektu se oba soubory označí (mezerníkem na klávesnici, nebo dvojklikem myši) a spojí se tlačítkem "Spojit". Potom je ještě nutné označit soubor který má být vždy spouštěn jako první v pořadí a nastavit jej jako hlavní tlačítkem "HlavníSoubor". To, který soubor je první, je vidět vlevo vedle soupisu v hranaté závorce číslováním typu A,B... (viz obrázek v odd. "Dialog Editace Projektu" v kap. "POPIS NABÍDKY PŘÍKAZŮ").

Zadáním příkazu "Program" z kteréhokoliv ze spojených zdrojových textů se provede korektní download obou souborů do automatu.

4. TYPY OKEN NA PRACOVNÍ PLOŠE

4.1. Okno editoru zdrojového textu

Slouží k vlastnímu "psaní programu" pro automaty.

Otevírá se příkazy z menu Soubor na horní liště (příkaz "Otevři soubor" - také zkráceně klávesou **F3** nebo příkaz "Nový")

Vestavěný editor zdrojového textu umožňuje editaci souboru o maximální délce 65535 znaků. Editor má standardní ovládání, kompatibilní s prostředím překladačů firmy Borland. Editor je doplněn dodatečnými funkcemi přístupnými přes lokální nabídku editoru a dále je vybaven vazbou na okno výpisu chyb a varování. Editor je schopen barevně rozlišovat zdrojový text podle syntaktických pravidel jazyka SIMPLE (viz popis menu Volby/Prostředí).

Veškeré blokové úpravy textu (přesuny, kopírování a mazání bloků textu) se provádějí následujícím způsobem: Nejprve označíme text buď myší (ukážeme myší na začátek bloku, stiskneme levé tlačítko a táhneme myš) nebo klávesnicí (stiskem kláves **Shift+šipky**). Potom použijeme pro požadovanou akci buď menu Editace na horní liště nebo rovnou příslušnou kombinaci kláves :

Ctrl-Ins	Kopíruj. Uloží blok do odkládacího editoru ("clipboard").
Shift-Ins	Vlož. Vloží na aktuální pozici kurzoru blok z odkládacího editoru.
Shift-Del	Vystřihni. Smaže text a zároveň jej uloží do odkládacího editoru.
Ctrl-Del	Smaž. Nevratně smaže blok textu.

Lokální nabídka editoru

Lokální nabídka editoru se vyvolá pomocí klávesnice stiskem kombinace kláves Ctrl-Enter nebo pomocí stisku pravého tlačítka myši za současného umístění kurzoru myši na okno editoru. Nabídka obsahuje tyto volby:

Překlad	spuštění překladu editovaného souboru
Běh	spuštění programu v automatu
Zastav	zastavení programu v automatu
Restart	restart automatu
Program	naprogramování přeloženého kódu do programové paměti automatu
Vstupy-Výstupy	otevření okna Sledovače se vstupy a výstupy automatu
Další Skupiny	otevření okna Sledovače s výběrem spec. funkčních proměnných
Sledovač	otevření okna Sledovače
Zásobník	otevření okna Zásobníku
Pes Typ	nastavení typu automatu např. A4, A3 apod.
Pes Adresa	nastavení adresy cílového automatu do něhož je určen zdrojový text
Pes Jméno	nastavení jména automatu (jen pro potřebu prostředí)
Proměnná	automaticky přidá do aktivního sledovače symbol na pozici kurzoru



Dále je editor zdrojového textu vybaven automatickým hledáním chybových řádků označených překladačem. Stiskem kláves **Alt-F8** je posunut ukazatel chyb na další chybový řádek. Předcházející chybový řádek je nalezen kombinací **Alt-F7**. Tyto funkce pracují ovšem pouze poté, když proběhl překlad.

Další související informace viz odstavec "Výpis chyb a varování".

Indikátor změny

Je v levém dolním rohu okna a pokud je zobrazen (hvězdička), znamená to, že soubor otevřený v okně ještě nebyl po provedení poslední změny uložen.

Indikátor adresy, jména a stavu automatu

Indikátor adresy, jména a stavu automatu je umístěn v levém dolním rohu každého okna zdrojového textu a každého okna pro výpis zásobníku a hodnot proměnných.



Výpis informací v indikátoru je proveden ve formálním tvaru : **[adresa:jméno:stav]** , kde položka adresa zobrazuje zvolenou adresu příslušejícího automatu na síti, položka jméno zobrazuje zadané jméno automatu a položka stav indikuje zda automat běží (*), stojí (-) nebo nebyl na síti nalezen (nevypíše se žádný znak).

Příklad:

[05:KOTELNA (*)]

Uvedený příklad výpisu označuje, že příslušné okno zobrazuje informace, proměnné, zásobník nebo zdrojový text pro automat umístěný na adrese 5. Automat má jméno KOTELNA a v daném okamžiku program aplikace v automatu běží.

Adresa automatu

Prostředí udržuje všechny nastavené parametry v projektu. Z tohoto důvodu doporučujeme používat direktivu NetAddr ve zdrojovém textu pro určení adresy automatu (viz dále). Pokud není zadána, je uživatel k jejímu zadání před programováním vyzván, příp. je možné adresu přesměrovat volbou Pes Adresa z lokální nabídky.

Jméno Automatu

Slouží pro snazší identifikaci automatu na síti. Vyčítá se z automatu, pokud bylo programem SETPES nastaveno (viz dále). Lze jej přepsat např. z lokální nabídky (Pes Jméno), avšak nepřenáší se do automatu (tedy po opětovném spuštění PESpro má automat opět původní název).

4.2. Okno sledovače, "Další skupiny", "Vstupy-výstupy"

Sledovač je uživatelsky definovatelné zobrazení proměnných programu automatu. Okno lze otevřít buď z hlavní nabídky na horní liště, nebo přes lokální nabídku okna zdrojového textu.



Prostředí umožňuje otevřít pro sledování skupin speciálních funkčních proměnných a vstupů/výstupů automatu ještě okna "**Další Skupiny**" a "**Vstupy-výstupy**". Tato okna jsou však stejná jako obecný sledovač, s tím rozdílem, že se do okna ještě před jeho otevřením automaticky nadefinuje celá skupina požadovaných proměnných.

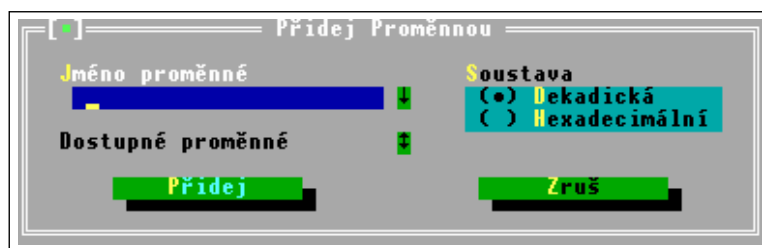
Editační a zobrazovací okno sledovače je vybaveno lokální nabídkou, která se vyvolá stiskem kláves **Ctrl-Enter** nebo stiskem pravého tlačítka myši při současném umístění kurzoru myši nad plochu okna zobrazení sledovače.

Lokální nabídka okna sledovače se týká vždy právě zvýrazněného řádku s proměnnou ve sledovači a obsahuje tyto příkazy:

- Změň** vyvolání dialogu změny hodnoty proměnné automatu
- Přidej** vyvolání dialogu pro výběr a přidání další proměnné sledovače
- Zruš** příkaz smaže řádek s zruší vyčítání proměnné na pozici kurzoru
- Hexadecimálně** nastavení zobrazování proměnné v hexadecimálním tvaru
- Decimálně** nastavení zobrazení proměnné v dekadickém tvaru

Zadávání nových proměnných do oken sledovačů

Provádí se vždy z některého aktivního okna typu sledovač buď přímo stiskem klávesy **Insert** anebo výběrem položky "Přidej" z lokální nabídky sledovače.



Dialogové okno zadání proměnné je tvořeno těmito objekty:

- Jméno Proměnné** Řádkový editor na zadání jména proměnné s pamětí položek.
- Dostupné Proměnné** Stiskem ikony "Dostupné proměnné" (nebo **Ctrl-Enter**) otevřeme seznam proměnných, použitých v odpovídajícím zdrojovém textu. Klávesou **Enter** položku rovnou dosadíme jako Jméno Proměnné.
- Soustava** Přepínač mezi dekadickou a hexadecimální soustavou.
- Přidej** Tlačítko potvrzení volby a přidání do výpisu.
- Zruš** Tlačítko pro zrušení akce.
- Dostupné Proměnné**

Výpis dostupných proměnných je tvořen seznamem jmen definovaných uživatelem ve zdrojovém textu. Seznam je tvořen pro každou adresu automatu z odpovídajícího zdrojového textu překladačem a je tedy aktualizován až po překladu.

V případě změn jmen proměnných ve zdrojovém textu může výpis obsahovat neaktuální jména. Tato nekonzistentnost dat je odstraněna po následném úspěšném překladu zdrojového textu.

Přímé vložení proměnné do sledovače

Novou proměnnou lze dokonce zadat přímo z okna zdrojového textu buď kombinací kláves **Ctrl-F7** nebo přes lokální nabídku - položka Proměnná. Podmínkou je, aby kurzor v textovém okně zrovna ukazoval na název některé proměnné. V tomto případě se vůbec nenabízí výše popsaný dialog a proměnná se založí do okna sledovače souvisejícího adresou se zdrojovým textem.

4.3. Změna hodnot proměnných

Okna sledovačů a zásobníku poskytují kromě vyčítání i možnost okamžité změny hodnot proměnných (tedy zápis z prostředí do proměnné v automatu). Vyvolat dialog pro změnu proměnné (nebo položky zásobníku) lze dvojným způsobem :

Myší a) Dvakrát za sebou klikneme myší na položku, kterou chceme změnit.
b) Vybereme položku (jedním kliknutím) a pravým tlačítkem myši vyvoláme lokální nabídku. Tam pak vybereme položku "Změň".

Klávesnicí a) Šípkami vybereme položku kterou chceme změnit a stiskneme **Enter**.
b) Na vybrané položce stiskem **Ctrl-Enter** vyvoláme lokální nabídku a z ní vybereme "Změň".

4.4. Okno zásobníku

Zobrazuje oblast zásobníku vybraného automatu.

Lze jej otevřít buď z hlavní nabídky na horní liště, nebo přes lokální nabídku okna zdrojového textu.

Editační a zobrazovací okno zásobníku je vybaveno lokální nabídkou, která se vyvolá stiskem kláves Ctrl-Enter nebo stiskem pravého tlačítka myši při současném umnístění kurzoru myši nad plochu okna zobrazení zásobníku.



Nabídka obsahuje příkazy:

- Změň** vyvolání dialogu změny hodnoty zásobníku na pozici kurzoru
- Hexadecimální** nastavení zobrazování hodnot v hexadecimálním tvaru
- Dekadická** nastavení zobrazení hodnot v dekadickém tvaru

POZN.: V zásobníku nelze přidávat ani mazat položky, jako např. v oknech sledovačů.

4.5. Přřazení adresy automatu oknům sledovačů

Pokud je okno zásobníku nebo sledovače otevřeno z lokální nabídky editoru zdrojového textu, resp. je při jeho otevření aktivní některé okno editoru zdrojového textu, pak se okno zásobníku váže k automatu jehož adresa je uvedena na okně editoru zdrojového textu. Změníme-li následně adresu pro toto "rodičovské" okno editoru zdrojového textu, změní se i adresy všech oken zásobníků a sledovačů "zavolaných" z tohoto okna.

Pokud při otevírání sledovacích oken není žádné okno editoru zdrojového textu právě aktivní (nebo třeba nemá přiřazenu adresu automatu), pak je při otevírání okna uživatel nejprve vyzván, aby ze soupisu vybral kýžený automat.

4.6. Okno nápovědy

Slouží jako stručný elektronický "on-line" manuál k vývojovému prostředí.

Okno se otevře buď kliknutím myši na položku Nápověda (dolní lišta vlevo) nebo stiskem klávesy **F1**. Nápověda se otevře v závislosti na právě aktivním okně pokud možno rovnou tam, kde je to třeba. Kliknutím myší na zvýrazněné položky lze v nápovědě přecházet mezi jednotlivými oddíly (zpět se vracíme stiskem kláves **Alt-F1**). Pokud ovládáme prostředí klávesnicí, nastavíme kurzor na zvýrazněnou položku a stiskneme **Enter**. V případě potřeby je možno najít na každé stránce položku "Soupis-index", která vede na hlavní stránku nápovědy se seznamem témat.

4.7. Výpis chyb a varování

Okno výpisu chyb a varování je otevíráno v okamžiku, kdy překlad zdrojového textu(ů) skončil neúspěšně z důvodu chyb. V okně se vypisují typy chyb z jednotlivých překládaných zdrojových textů. Pohybem kurzoru v okně jsou jednotlivé chyby zvýrazňovány v příslušném zdrojovém textu. Pomocí stisku klávesy "Mezera" nebo "Enter" je nalezen řádek s chybou na pozici kurzoru a současně je odpovídající editor zdrojového textu aktivizován a jeho kurzor je postaven na chybový řádek. V případě, že jsou chyby ve větším počtu zdrojových textů odlišuje se funkce klávesy "Mezera" od klávesy "Enter". Klávesa mezery označí v okně Chyb chyby příslušející zdrojovému textu jehož chyba byla označena. Pohyb kurzoru v okně chyb zůstává sice beze změny, ale vyhledávány jsou pouze chyby označeného souboru.

Okno chyb uzavřeme klávesami **Alt-F3** nebo příkazem Uzavření. Okno se též uzavře automaticky po úspěšném překladu zdrojových textů.

5. POPIS NABÍDKY PŘÍKAZŮ (MENU)

Hlavní nabídku tvoří příkazy příkazové lišty na horním řádku obrazovky. Položky menu aktivujeme buď kliknutím myši, nebo stiskem klávesy **F10** a následným výběrem pomocí šipek a ENTER, anebo zkratkou - kombinací kláves. Lištu hlavní nabídky tvoří příkazy:

- = příkaz pro vyvolání nabídky podpůrných programů (zkrácené volání provedeme stiskem **Alt-"mezera"**)
- Soubor** nabídka "soubor" je tvořena příkazy pro práci se soubory, projekty a zahrnuje některé příkazy systémového typu tj. např. změna adresáře, volání příkazové řádky DOSu a ukončení programu (zkrácené volání nabídky provedeme stiskem kláves **Alt-S**)
- Okna** nabídka "okna" sdružuje soubor příkazů pro práci s jednotlivými okny, umístěnými na pracovní ploše. Obsahuje příkazy pro přeuspořádání oken, příkazy volby aktivního okna a příkazy pro volbu jednoho ze dvou typů ("za sebou", "vedle sebe") automatického rozmístění oken na pracovní ploše. Zkrácené volání nabídky provedeme stiskem kláves **Alt-O**.
- Editace** nabídka "editace" sdružuje příkazy pro nejčastěji používané úkony v textovém editoru zdrojového textu. Jedná se o příkazy práce s blokem textu, hledání a záměny textových řetězců. Pro zkrácené volání nabídky je vyhrazena kombinace kláves **Alt-E**.
- Zobrazení** v nabídce zobrazení jsou seskupeny příkazy pro otevření oken výpisu obsahů proměnných, včetně standardního uspořádání proměnných a uspořádání specifických proměnných pro zadaný typ automatu. Pro zkrácené volání nabídky je vyhrazena kombinace kláves **Alt-Z**.
- Běh** nabídka "běh" je svojí strukturou určena k ovládání jednotlivých automatů dostupnými povely. Zkrácené volání provedeme pomocí kláves **Alt-B**.
- Volby** nabídka "volby" obsahuje příkazy pro uživatelské nastavení prostředí. Příkazy se týkají např. nastavení přípon zpracovávaných souborů, nastavení barevného zvýraznění zdrojového textu, volby režimu obrazovky apod. Je zde rovněž funkce pro nastavení rychlosti vyčítání dat ze sítě a funkce pro zobrazení stavu sítě a přehledu automatů. Zkrácené volání se v tomto případě provede stiskem kláves **Alt-V**.

5.1. Nabídka podpůrných funkcí

Sdružuje následující příkazy:

- Tabulka znaků** výpis kódů znaků ASCII tabulky včetně číselného vyjádření znaku v soustavě dekadické a hexadecimální

- Kalkulátor** jednoduchý kalkulátor základních početních operací vybavený převody mezi číselnými soustavami
- Videostop** implementace známé televizní hry ze stejnojmenného pořadu. Je určena pro chvíle naprostého vyčerpání programátora.
- Informace** výpis základních informací o nastavení prostředí a stavu komunikační linky
- O Programu** seznam autorů programu

Tabulka Znaků

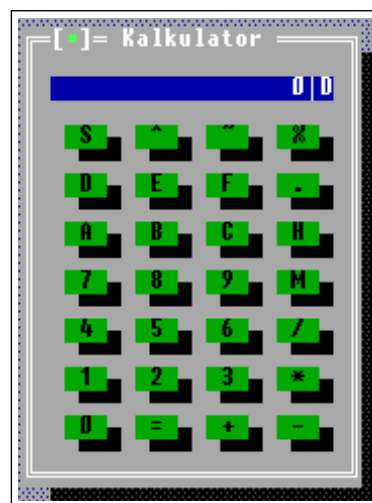
Tabulka znaků ASCII kódů je určena pro snadný převod znaku do číselného vyjádření. V horní polovině je všech 256 definovaných znaků ASCII tabulky. Pomocí kurzorových kláves je možné se pohybovat v tabulce. Na spodní liště je pak zobrazen vybraný znak a jeho číselná reprezentace v dekadické a hexadecimální soustavě. ASCII tabulku uzavřeme příkazem Uzavření (Alt-F3), nebo kliknutím myší na uzavírací ikonu.

Kalkulátor

Slouží pro pohodlný výpočet v rozsahu proměnných typu "word" tj. celých čísel v rozmezí 0-65535. Kalkulátor se ovládá buď myší a nebo stiskem znaků na klávesnici odpovídajících popisu tlačítek.

Význam jednotlivých kláves kalkulátoru je následující:

- S** mazání výpočtu
- ^** zobrazení čísla na dipleji v tzv. dvojkovém doplňku
- + -** změna znaménka zobrazovaného čísla
- %** převod čísla do binární soustavy
- .** převod čísla do dekadické soustavy
- H** převod čísla do hexadecimální soustavy
- M** zbytek po celočíselném dělení
- /** dělení
- *** násobení
- odčítání
- +** sčítání
- =** výpočet zadané operace
- 0-9** tlačítka pro zadávání čísel v dekadické soustavě
- A-F** tlačítka pro zadávání čísel v hexadecimální soustavě



Zobrazení výsledků je doplněno za oddělovací čarou informací o číselné soustavě v níž je výsledek zobrazen (D=dekadická, H=hexadecimální, B=binární soustava). Kalkulátor odstraníme příkazem Uzavření (Alt-F3), nebo kliknutím na uzavírací ikonu.

Videostop

Malé rozptýlení pro unaveného programátora.

5.2. Nabídka Soubor

Nabídka Soubor obsahuje příkazy související s prací nad soubory, projekty a obsahuje též příkaz pro ukončení programu a příkaz pro "odskok" do operačního systému (zde se rozumí MS-DOS a jeho příkazový řádek). Příkazy nabídky jsou:

Nový Soubor	příkaz umožňuje otevřít nový soubor zadaného jména
Otevři Soubor	příkaz pro otevření již existujícího souboru
Ulož Soubor	příkaz provede uložení souboru z aktivního editoru
Ulož Se Jménem	příkaz umožňující uložit obsah aktivního editoru do souboru se zvoleným jménem
Nový Projekt	příkazem je možné založit nový soubor projektu, tj. speciální soubor, v němž se uschovává nastavení prostředí, jména zdrojových textů souvisejících s projektem a pomocí něhož se udržuje aktuálnost editovaných souborů vůči spuštěným programům v automatech
Otevři Projekt	příkazem je možné zavést nastavení zvoleného projektu do vývojového prostředí
Ulož Projekt	uložení nastavení projektu do definičního souboru projektu
Uzavři Projekt	příkazem je možné uzavřít projekt v aktuálně rozpracované fázi pro potřeby dalšího zpracování v budoucnosti
Ulož Projekt jako Adresář	uložení aktuálně otevřeného projektu pod zvoleným jménem
DOS	příkazem je vyvolán dialog umožňující změnu aktuálně nastaveného pracovního adresáře
Konec	povel pro volání služeb operačního systému MSDOS - "odskok" do příkazového řádku MSDOS.
	ukončení práce s programem

Dialog Volby Souborů

Dialog volby souborů je složen z řádkového editoru jména souboru, z objektu výpisu souborů, informačního okna a tlačítek Otevři a Zruš. Toto dialogové okno bylo navrženo pro usnadnění volby editovaného souboru či projektu. Jméno souboru můžeme zadat buď přímo pomocí řádkového editoru nebo ho můžeme najít pomocí objektu výpisu souborů. V případě, že jsme s výběrem souboru spokojeni, stiskneme klávesu Enter nebo tlačítko Otevři. V případě, že chceme proceduru zadávání souboru ukončit aniž by naše volba byla provedena, stiskneme klávesu Esc nebo tlačítko Zruš. Volbu souboru provádíme vždy pomocí aktivního (vysvíceného objektu). Přepínání aktivity objektu je možné provést kliknutím myši na zvolený objekt nebo stiskem klávesy TAB. Dialog zadávání jména souboru je vybaven tzv. historií tj. soupisem jmen souborů zadaných v předchozích voláních. Soupis jmen souborů se vyvolá při aktivním řádkovém

editoru jména souboru stiskem kurzorové klávesy pro pohyb dolů. V soupisu jmen je možné listovat pomocí kurzorových kláves. V případě, že na vybraném jménu souboru z tohoto soupisu stiskneme klávesu Enter, bude jméno souboru dosazeno do řádkového editoru dialogového okna. V případě, že stiskneme klávesu Esc, dojde k uzavření okna historie a volba jména souboru není provedena.

Dialog Editace Projektu



Dialog editace projektu je přístupný při otevírání nebo zakládání projektu. Jinak je možno jej otevřít z menu **Volby** položkou **Editace Projektu**. Bližší informace k projektům viz kap. "Práce s projekty" a "Zatažení dvou programů do automatu".

Dialog obsahuje výpis souborů tvořících projekt a tlačítka Ano, Zruš, Přidej, Vypušt, Spojit, Rozpojit, Hlavní soubor. Význam tlačítek je následující:

- Přidej** výběr a přidání dalšího souboru do projektu (soubor musí existovat). Jméno souboru se volí pomocí dialogu pro výběr jmen souborů.
- Vypušt'** stiskem tlačítka je z projektu vypuštěn soubor na jehož jméno je posvíceno kurzorem
- Zruš** ukončení editace projektu bez akceptování provedených změn
- Ano** změny v sestavě jmen souborů budou akceptovány
- Spojit** spojení dvojice vybraných souborů pro společný download do automatu (výběr souboru mezerníkem nebo dvojklikem myši)
- Rozpojit** rozpojení vybrané dvojice souborů
- Hlavní soubor** vybraný soubor ze spojené dvojice bude nastaven jako hlavní (tedy jako první pro spuštění v hlavní smyčce programu)

Dialog Volba Pracovního Adresáře

Dialog "Volba Pracovního Adresáře" umožňuje změnit aktuální adresář. Dialog je vybaven řádkovým editorem pro Jméno adresáře s jehož pomocí je možné cestu na zvolený adresář napsat. Tento editor spolupracuje s výpisem adresáře, který je zpracován do formátu "stromu". Pomocí kurzorových kláves a klávesy Enter (případně myši) je možné pohybem kurzoru změnit aktuální adresář na nový. Volbu je možné buď potvrdit stiskem tlačítka Ano, stiskem tlačítka Zpět nastavit původní adresář nebo stiskem tlačítka Změň změnit adresář a pokračovat ve výběru. V případě, že nechceme změnit adresář, uzavřeme dialog pomocí stisku klávesy Esc nebo pomocí myši klikneme-li na uzavírací ikonu okna.

5.3. Nabídka Okna

V nabídce okna jsou k dispozici příkazy pro práci s aktuálně otevřenými okny výpisů a editorů na pracovní ploše prostředí. Příkazy nabídky jsou:

- Další** příkazem se předá ovládání z klávesnice a myši oknu, které následuje v řadě za právě aktivním oknem tj. zvolí se jako aktivní následující okno (takto lze postupně "listovat" po řadě všemi okny, která jsou právě na pracovní ploše).
- Předešlé** předání aktivity předešlému oknu v řadě oknen výpisů a editorů umístěných na pracovní ploše prostředí (tedy opačná funkce k funkci "Další").
- Zvětšení** zvětšení aktivního okna na maximální a nebo zpět na původní rozměr, pokud to ovšem charakter okna dovolí. Zvětšovat není možné dialogová okna.
- Uzavření** příkazem je možné odstranit okno z pracovní plochy prostředí.
- Rozměr** příkaz pro změnu aktivního stavu okna na stav, v němž je možné editovat rozměry a umístění okna na pracovní ploše prostředí (používá se pro úpravu rozměrů okna, zejména tehdy, nepracujeme-li s myší ale pouze s klávesnicí).
- Vedle Sebe** umístění oken na pracovní ploše tak, aby byl pokud možno vidět celý zobrazovaný obsah okna.
- Za Sebou** umístění oken po ploše do formátu podobného lístkům kartotéky.

5.4. Nabídka Editace

Tyto příkazy jsou zcela vyhrazeny pro práci s editory textu. Není tedy možné je použít např. pro okna výpisu proměnných a podobně. Jsou k dispozici tyto příkazy:

- Vrať** vrácení naposledy smazaného textu zpět do editovaného souboru.
- Vystříhni** smazání označeného bloku textu, přičemž se tento blok uloží do odkládacího editoru
- Kopíruj** vloží kopii obsahu označeného bloku textu do odkládacího editoru
- Vlož** vložení naposledy uloženého bloku textu z odkládacího editoru do editoru aktivního okna.
- Odkládací editor** zobrazení obsahu odkládacího editoru na pracovní ploše prostředí
- Smaž** smazání označeného bloku textu v editoru.
- Najdi** příkazem se vyvolá nastavení funkce pro hledání zadaného řetězce v editovaném souboru aktivního editovacího okna.
- Zaměň** příkazem se vyvolá nastavení funkce pro nahrazení zadaného řetězce v editovaném souboru aktivního editovacího okna.
- Znovu Najdi** příkazem se spustí funkce hledání řetězce ovšem s tím, že je vynecháno nastavování podmínek hledání řetězce a tudíž jsou po užity naposledy zadané symboly a jména.

5.5. Nabídka Zobrazení

Pomocí příkazů nabídky zobrazení se provádí umístění oken pro zobrazování proměnných a zásobníku na pracovní plochu prostředí. Jsou dostupné tyto příkazy:

- Sledovač** vyvolání okna určeného pro uživatelské uspořádání výpisu proměnných. Pomocí funkcí tohoto okna je možné obsah proměnných měnit.
- Zásobník** umístění okna s obsahem zásobníku zvoleného automatu na pracovní plochu prostředí
- Další Skupiny** umístění okna výpisu proměnných automatu jehož základní obsah je prefabrikován pomocí definice skupin v definičním souboru daného typu automatu (soubory typu *.DEF v domovském adresáři prostředí).
- Vstupy-Výstupy** zobrazení obsahu skupiny vstupů a výstupů zvoleného automatu. Počet vstupů a výstupů je definován jako jedna skupina v souborech typu *.def v domovském adresáři prostředí

5.6. Nabídka Běh

V této nabídce jsou umístěny povely pro volání překladače a "loaderu" přeložených souborů, dále povely přímo ovládající chod programu v připojeném automatu. Dostupné příkazy nabídky:

- Překlad** spustí překlad zdrojového textu v právě aktivním okně editoru překladačem jazyka SIMPLE.
- Přelož Projekt** vyvolá překlad všech souborů definovaných pomocí projektu u nichž došlo k nekonzistenci zdrojového textu a přeloženého programu (pokud je zdrojový text "mladší" než přeložený program).
- Běh** spustí program v zadaném automatu.
- Zastav** zastaví vykonávání programu v zadaném automatu.
- Restart** nastaví automat do počátečního stavu (včetně nastavení bitu RESET) a znovu-spuštění instalovaného programu v automatu.
- Program** zatáhne přeložený program do automatu resp. nejprve přeloží pokud byl zdrojový text změněn (zatahuje se vždy program z právě aktivního okna editoru).
- Nástroje** podmenu s nabídkou dalších podpůrných programů a nástrojů které mohou být spouštěny pod integrovaným prostředím.

Nástroje

Velmi užitečná funkce, která poskytuje přímo do prostředí PESpro nabídku dalších podpůrných programů. Prakticky všechny programy, dodávané k systému PES, mohou být začleněny jako nástroje (pokud jsou příslušné *.EXE soubory nainstalovány do domovského adresáře PESpro, jsou programy do tohoto menu automaticky začleněny. Všechny nástroje jsou samostatné programy a lze je samozřejmě spouštět i

samostatně bez prostředí PESpro. Výhoda spouštění prostřednictvím volby nástroje je v tom, že PESpro příslušný program už volá s nastavenými parametry (předá správné číslo COM-portu, nastaví správnou volbu české diakritiky a případně i předá název souboru nebo projektu). Volaný nástroj má potom díky těmto automatickým asociacím už otevřeny přesně ty soubory, které jsou potřeba.

5.7. Nabídka Volby

V nabídce volby jsou umístěny příkazy pomocí nichž je možné nastavit základní parametry prostředí. Je možné použít příkazy:

- Prostředí** dialogové okno pro nastavení základních vlastností prostředí, editoru a velikosti (rozlišení) obrazovky.
- Barvy** okno pro nastavení barev jednotlivých objektů.
- Editace Projektu** dialogové okno projektu - viz obr. výše a stať "Práce s projekty".
- Rychlost Vyčítání** vyvolání dialogového okna pro nastavení frekvence vyčítání dat z automatů čtecím a editovacím procesem proměnných v prostředí.
- Stav Sítě** příkazem je vyvolána dialogové okno se základními informacemi o stavu sítě, typech použitých automatů atd.
- Nápověda Index** vyvolá obsah integrované nápovědy k prostředí.

Prostředí

Pomocí dialogu pro Nastavení Prostředí provádíme uživatelskou konfiguraci základních parametrů zobrazování a chování prostředí. Jsou k dispozici tato nastavení a volby:

Rozměr Obrazovky

Volba umožňuje nastavit rozlišení obrazovky buď na 25 nebo 50 řádek.

Automatické Uložení

Je možné nastavit automatické uložení zdrojových textů a pracovní plochy při ukončení programu. V případě, že volby nejsou zapnuty, pak bude uživatel vyzván k potvrzení nebo odmítnutí uložení.

Textový editor

Auto Odsazení je volba která způsobí automatické zarovnávání (odsazení zleva) nových textových řádků (po stisknutí Enter) podle řádků předchozích.

Přepisování - vypne automatické odsouvání textu během vkládání nového.

Barevné zvýraznění

Zapíná tzv. "Syntax Highlight" funkci. V oknech textových editorů budou po zvolení této funkce barevně rozlišovány části textu s různým významem podle syntaxe jazyka SIMPLE (je možné odlišit klíčová slova, direktivy, komentáře atd.). Doporučujeme tuto funkci nastavit, neboť významnou měrou zpřehlední psaný text. Barvy jednotlivých syntaxních skupin je možno nastavit v menu Volby/Barvy (viz dále).

Extenze souborů

Umožňuje nastavit implicitní přípony souborů zdrojových textů a souborů přeloženého kódu (soubor aplikace) pro zatažení do automatů. Příponu DNL u souborů aplikací doporučujeme neměnit !

Nastavení Barev

Dialogové okno Nastavení Barev je určeno pro uživatelskou volbu barevného podání používaných programových objektů. Je možné nastavit např. barvu pracovní plochy, barvy rámečků oken atd. Ovládání okna pro nastavení barev se provádí pomocí výpisů Skupina a Položka. Obsah výpisu Položka se mění podle aktuální volby ve výpisu Skupina. Zvolíme-li objekt ve Skupině a příslušnou Položku, jsou nastaveny selektory barev pro popředí a pozadí podle aktuální volby a současně je zobrazen ukázkový text pod selektorem pozadí. Selektor pozadí též umožňuje nastavit blikání textu. Poslední volbou, kterou umožňuje dialog barev nastavit je znak zobrazovaný jako podklad pracovní plochy. Znak vybereme pomocí aktivního selektoru Vzorek podkladu tak, že stiskneme kurzorovou šipku dolů. Tím se otevře okno s výpisem ASCII znaků. Kurzorem vybereme znak a stiskneme klávesu Enter. Vybraný znak je po uzavření tabulky znaků zobrazen selektorem. V případě, že nové nastavení barev nechceme akceptovat uzavřeme dialog Nastavení barev stiskem klávesy Esc nebo stiskem tlačítka Zruš.

Rychlost Vyčítání

Nastavením parametru rychlosti vyčítání volíme zatížení komunikační linky procesem vyčítání hodnot proměnných z jednotlivých automatů. Čím je zatížení linky větší tím pomaleji probíhá komunikace mezi jednotlivými automaty na lince při současném zvýšení rychlosti obnovy dat na obrazovce počítače. Zatížení linky je možné volit v rozsahu 10%-90%. Volba se provádí buď v grafickém režimu pomocí vodorovné rolovací lišty nebo číselně pomocí řádkového editoru. Nastavení ukončíme potvrzením volby pomocí stisku tlačítka Změň. Požadujeme-li ponechat původní nastavení provedeme ukončení editace pomocí tlačítka Zruš nebo stiskem klávesy Esc.

Stav Sítě

Dialog zobrazení stavu sítě se používá pro zobrazení detailních informací o stavu jednotlivých automatů na komunikační lince. Prohlížení stavu sítě ukončíme stiskem tlačítka Ano nebo Ne. Dialogové okno je dále vybaveno výpisem nalezených automatů v síti. V případě, že byl automat nalezen, je vypsáno jméno instalované aplikace pokud bylo při instalaci zadáno. Dialog Stav Sítě je používán i při zadávání adresy automatu pro okna výpisu proměnných. V tomto případě je adresa automatu akceptována pouze při ukončení dialogového okna stiskem tlačítka Ano.

6. SETPES - NASTAVOVÁNÍ AUTOMATŮ

V hlavním adresáři se po instalaci nalézá ještě důležitý program SETPES.EXE. Slouží k nastavování parametrů automatů a zejména k přípravě automatů pro síťový provoz.

Některé funkce pro zápis důležitých parametrů do konfigurační paměti EEPROM v automatech PES jsou realizovány separátním programem SETPES a nejsou podporovány (prozatím) přímo v prostředí PESpro. Jedná se o tyto funkce :

- Přiřazení síťové adresy (0 ... 31)
- Nastavení komunikační rychlosti (2400, 9600, 19200 nebo 57600 Bd)
- Pojmenování automatu jménem (max. 8 znaků)
- Nastavení reálného času v automatu podle systémového času v PC.
- Výpis informací o automatu

Pokud neprovozujeme automaty v síti, zpravidla SETPES nepotřebujeme (až na nastavení systémového času, nebo jména automatu pro jeho snazší identifikaci).

Nastavení adresy

U automatu provozovaného samostatně je lhostejné, jakou bude mít adresu. Pokud plánujeme spojení automatů do sítě, musíme nejprve nastavit každému automatu separátní adresu (od 0 do 30).

Volba komunikační rychlosti

Musí být u všech automatů v síti stejná a lze si vybrat z hodnot 2400 Bd, 9600 Bd, 19200 Bd a 57600 Bd. Pokud celková délka kabeláže nepřesahuje asi 50 metrů, lze provozovat síť na rychlosti 57600 Bd, při délkách do 500 metrů doporučujeme použití rychlosti 19200 Bd, při silném rušení nebo větších délkách kabeláže je vhodné použít 9600 Bd nebo 2400 Bd.

Samozřejmě na komunikační rychlosti přímo závisí datová propustnost sítě a s tím související např. odezvy automatů na změny sdílených síťových proměnných. S nastavenou komunikační rychlostí automatů rovněž souvisí i rychlost obnovování informací v oknech výpisu proměnných v prostředí PESpro.

Doporučení

Je nanejvýš vhodné si již během nastavování adres do automatů poznačit viditelně na automaty nastavené adresy - předejde se tak pozdějším zmatkům při zapojování a ladění. Rovněž je užitečné nastavit programem SETPES každému automatu v síti nějaké příznačné jméno.

6.1. Parametry programu SETPES

Program **SETPES** vyžaduje připojení jen jediné stanice na lince. To lze provést např. tak, že postupně připojujeme jeden automat po druhém a provádíme nastavení, anebo zapojíme kompletně celou síť včetně instalace všech automatů a nastavování provádíme s postupným zapínáním a vypínáním jednotlivých automatů.

Nastavování všech parametrů se provádí pomocí direktiv na příkazovém řádku DOS napsaných za názvem programu. Parametry začínají vždy lomítkem a jsou odděleny mezerami. Jediný parametr, který je vždy nutné zadat je číslo COM portu, kde je připojen převodník PES-CA1 k automatu PES. Ostatní parametry přidáváme podle potřeby, podle toho co všechno je třeba na automatu nastavit.

Program se spouští z příkazové řádky například takto:

SETPES /COMx /ADDRx /BAUDx /TIME

- /?** vypíše na obrazovku stručnou nápovědu k parametrům
- /COMx** udává číslo portu, na kterém je připojen automat (může být /COM1 až /COM4) - musí být vždy uveden !!
- /ADDRx** přiřazuje automatu síťovou adresu - povolené hodnoty 0..30
- /BAUDx** nastavuje baudovou rychlost (musí být u všech automatů shodná) povolené hodnoty 2400, 9600, 19200 a 57600
- /TIME** přenese aktuální hodnotu reálného času z PC do automatu
- /NAME_xx** nastaví automatu jméno **xx** (maximálně 8 znaků)

Příklady:

SETPES /COM2 /ADDR15 /BAUD19200 /TIME

(automat na COM2, nastavíme mu adresu 15, rychlost 19200 a nastavíme reálný čas)

SETPES /COM3 /ADDR0 /NAME_KOTELNA

(automat na COM3, nastavíme mu adresu 0 a pojmenujeme ho "KOTELNA")

POZN.:

Nastavení reálného času zkopíruje stav reálného času v připojeném počítači do automatu.

Den v týdnu se nastaví tímto způsobem: Ne=1, Po=2, Út=3, St=4, Čt=5, Pá=6, So=7.

Naastavení reálného času lze samozřejmě provést i jinými způsoby přímým zápisem do registrů reálného času (z okna sledovače prostředí PESpro, prostřednictvím vizualizačního programu PESviz, programem v automatu, nebo vlastním softwarem na PC využívajícím knihovny PESlib nebo rozhraní PESdde).

7. SIMPLE 2 - POPIS PROGRAMOVÁNÍ

Programovací jazyk SIMPLE slouží k vyjádření jednoduchých řídicích algoritmů v průmyslových automatech systému PES.

Program je prováděn od začátku do příkazu "END", odkud se běh programu vrací zpět na začátek. Program tedy neustále probíhá v jediné smyčce kolem dokola a neumožňuje žádné skoky, pouze volání podprogramů. Tato vlastnost znemožňuje vznik fatálních chyb, především "bloudění" programu a "zacyklení" v nekonečné smyčce. Používání podprogramů dovolí zřehlednit strukturu programu a zefektivnit vlastní běh programu v automatu.

7.1. Syntaxe zápisů

SIMPLE je řádkový překladač, tedy každá instrukce i se souvisejícími parametry musí být zapsána na právě jednom textovém řádku a může být doplněna komentáři za středníkem. Je možný i zápis několika instrukcí na řádek - viz dále. Překladač jazyka SIMPLE nerozlišuje velká a malá písmena. Vícenásobné mezery v zápise nejsou na závadu.

Komentáře

Veškerý text, který je na řádku za středníkem, je považován za komentář a při překladu se neuplatní. Obdobně je při překladu ignorován i veškerý text za příkazem END. Toho lze využít například k zápisu globálních komentářů k celému programu.

7.2. Symbolické názvy

Každému textovému řetězci lze přiřadit symbolický název, který je při překladu nahrazován původním řetězcem. Toho lze použít především k přiřazování symbolických názvů proměnných. Jako symbolický název lze použít textový řetězec obsahující písmena a číslice a začínající vždy písmenem. Symbolický název nesmí obsahovat mezery, ani znaky aritmetických a logických operátorů. Přiřazení symbolické proměnné může být deklarováno kdekoliv ve zdrojovém textu s tím, že se uplatňuje od následujícího řádku. Přiřazení jednoho symbolického názvu musí být právě na jednom textovém řádku. Originální řetězec se napíše na začátek řádku, dále následuje operátor přiřazení "#" a za ním symbolický název. Originální řetězec nesmí obsahovat žádné jiné symbolické názvy, aby nemohlo dojít k rekurzivnímu přiřazení.

; DEFINICE SYMBOLU :

Y3 # Motor2 ; definice symbolického názvu pro výstup Y3
(I2>150) # Pretlak_Kotle1 ; zde je celý logický výraz (I2>150) definován jako symbol

; POUZIVANI SYMBOLU V PROGRAMU :

IF Pretlak_Kotle1 THEN Motor2' ; Při přetlaku kotle 1 vypni motor 2

Příklad jednoduchého programu v jazyce SIMPLE

```

; Příklad výstupu na žárovku, kde třemi tlačítky volíme tři stavy:
; Zhasnuto, Rozsvíceno, Blikání s periodou 1 sekunda
; DEFINICE SYMBOLICKÝCH NÁZVŮ
Y1 # ZAROVKA      ; výstup Y1
X0 # ZAPNI        ; vstup X0 - tlačítko
X1 # VYPNI        ; vstup X1 - tlačítko
X2 # BLIKEJ       ; vstup X2 - tlačítko
D1 # STAV         ; vnitřní proměnná - má např. tyto stavy :
                  ; 0..nesvítí, 1..svítí, 2..bliká

subroutine Blikani ; PODPROGRAM
  if T1 > 50 then ZAROVKA ; je-li T1>50 (tj. 500 ms) zapni žárovku
  if T1 > 100 then T1=0   ; je-li T1>100 (tj. 1sec.) vrací se T1 na nulu
return

; HLAVNÍ PROGRAMOVÁ SMYČKA
if RESET then T1=0 : TEN1 : STAV=0 ; úvodní inicializace po resetu automatu
                                   ; spustit časovač T1, nastavit STAV na "vypni"

if ZAPNI then STAV=1                ; nastavení stavu od jednotlivých vstupů
if BLIKEJ then STAV=2
if VYPNI then STAV=0                ; tlačítko "VYPNI" zpracujeme jako poslední (je-li více
                                   ; tlačítek stisknuto, pak má "VYPNI" poslední slovo)

ZAROVKA'                            ; vypnutí výstupu
                                   ; výstup můžeme bez obav nejprve vypnout a na
                                   ; dalších řádcích třeba zapnout, vstupy a výstupy
                                   ; se fyzicky přepisují až na konci smyčky (END)

if STAV=1 then ZAROVKA              ; zapnutí výstupu
if STAV=2 then Blikani              ; pokud je STAV=2, zavolá se podprogram
RESET'                              ; vynulovat RESET,
                                   ; aby se už neprováděly inicializace na začátku

END                                  ; KONEC SMYČKY

```

Program začíná instrukcemi v sekci **HLAVNÍ PROGRAMOVÁ SMYČKA** (v sekci **DEFINICE..** žádné výkonné instrukce nejsou a podprogramy jsou vyděleny z hlavní smyčky klíčovými slovy "subroutine" a "return"). Testování bitu **RESET** slouží k provedení úvodních inicializací po zapnutí automatu. V našem případě spouštíme časovač **T1** v základním nastavení (tedy dopředu a s krokem 10 ms) a nastavíme stavovou proměnnou **STAV** do definovaného stavu. Na konci smyčky se bit **RESET** nuluje, takže řádek s inicializacemi se provede jen jednou, po zapnutí automatu. Další činností je otestování vstupů a příp. přestavení proměnné **STAV**. Poté vypneme výstup **ZAROVKA** a následně ho různými logickými podmínkami můžeme zapnout. Tento postup nezpůsobí na výstupu žádné rušivé jevy, protože přepis proměnné **Y3** na fyzický elektrický výstup automatu se provádí až na konci smyčky (rozhodující je tedy hodnota **Y3** v bodě **END**). Pokud je **STAV=0**, zůstane výstup vypnutý. Pokud je **STAV=1**, výstup se zapne. Pro **STAV=2** se volá podprogram pro blikání. Ten prvních 500ms (50 kroků) nedělá nic, druhých 500ms zapíná výstup a pak vrátí **T1** opět na nulu.

7.3. Proměnné

SIMPLE pracuje s 320 binárními proměnnými. Z toho 64 proměnných reprezentuje jednotlivé bity vstupů/výstupů, 32 vstupních a 32 výstupních. Zbýlých 256 binárních proměnných reprezentuje "vnitřní relé", z nich je 128 obecně použitelných a 128 je rezervováno pro speciální funkce. Dále interpreter pracuje s 256 proměnnými typu WORD (16-bitové celočíselné proměnné). Z nich 32 reprezentuje analogové vstupy, 32 analogové výstupy, 64 je obecně použitelných a 127 zbývá pro registry se speciálními funkcemi.

Obsahy proměnných reprezentující vstupní úrovně jsou z fyzických vstupů vzorkovány vždy na začátku celého programového cyklu a ve stejném bodě jsou i obsahy proměnných reprezentující výstupní úrovně přepisovány na fyzické výstupy.

Horní polovina uživatelských proměnných (tedy horních 64 vnitřních relé - M64..M127 a horních 32 obecných proměnných - D32..D63) je sdílená mezi automaty ve společné síti. Pokud používáte automat samostatně, lze bez omezení používat všechny proměnné M i D, pokud používáte automaty v síti, doporučujeme pečlivě prostudovat kapitolu "Funkce síťové vrstvy".

Seznam dostupných proměnných

Název proměnné	Popis	Počet	Typ	Rozsah hodnot
X0 X31	Číslicové vstupy	32	bit	0 / 1
Y0 Y31	Číslicové výstupy	32	bit	0 / 1
M0 .. M127	Bity -tzv.vnitřní relé (M63-M127 síťové)	128	bit	0 / 1
B0 .. B127	Speciální funkční bity	128	bit	0 / 1
I0 I31	Analogové vstupy	32	word	0..65535
O0 O31	Analogové výstupy	32	word	0..65535
D0 D63	Obecné proměnné (D32-D63 síťové)	64	word	0..65535
W0 .. W127	Speciální funkční registry	128	word	0..65535

7.4. Instrukční soubor

V jazyce SIMPLE existují následující příkazy:

- výrazový příkaz
- podmíněný příkaz
- příkaz END

Výrazový příkaz

Může být buď nastavení, nulování nebo invertování binární proměnné a přiřazení do binární proměnné nebo proměnné typu WORD.

Příkaz pro nastavení binární proměnné je prostý zápis jména této proměnné, je-li za toto jméno připojen apostrof ('), jedná se o příkaz nulování binární proměnné. Lze samozřejmě i přiřadit hodnotu operátorem (=). Invertování binární proměnné se provádí operátorem (!). Operandem těchto instrukcí může být jakákoliv binární proměnná kromě binárních vstupů (X).

Y1 ; dva způsoby nastavení výstupu Y1
Y1=1
M12' ; dva způsoby nulování vnitřního relé M12
M12=0
!Y3 ; dva způsoby inverze výstupu Y3
Y3=Y3'

Přiřazení do proměnné se provádí pomocí operátoru "=". Přiřazení není možné provést do binárního ani analogového vstupu, tedy proměnné typu X a I. Přiřadit lze konstantu, proměnnou stejného typu a aritmetický resp. booleovský výraz složený z uvedených operandů. Aritmetický výraz může být složený a může používat tyto operátory:

+	sčítání
-	odčítání
*	celočíselné násobení
/	celočíselné dělení

O3 = (I1+I2)/2 ;na analog.výstup O3 přiřadí aritmetický průměr hodnot I1 a I2.
D21=D21+1 ;inkrementuje obsah proměnné D21

Booleovský výraz může být složený a může používat tyto operátory:

and	logický součin
or	logický součet
'	negace

Y3 = (X1 or X2)'and M2

Výrazový příkaz nesmí být delší než 256 znaků. Na jednom řádku může být více výrazových příkazů oddělených oddělovačem ":".

Y2': M3 : O1=D22+I3/4 : M5':Y5=X0 and X1

UPOZORNĚNÍ

Celý instrukční řádek může obsahovat maximálně 256 znaků včetně rozvinutí všech symbolů použitých na řádku.

Podmíněný příkaz

Slouží k podmíněnému provádění příkazů. Zápis podmíněného příkazu povinně obsahuje klíčová slova "IF" a "THEN". Klíčové slovo IF musí být umístěno jako první na textovém řádku, za ním následuje výraz podmínky. Podmínka je ukončena slovem THEN, které zároveň uvozuje vlastní příkaz. Tento příkaz je prováděn pouze tehdy, je-li podmínka splněna.

```
IF I2>100 THEN Y0'      ;je-li hodnota na analogovém vstupu I2
                        ;větší než 100, nuluje se výstup Y0.
```

V podmínkové části může být složená logická funkce elementárních binárních relací. Jako tyto elementární podmínky mohou být použity přímo obsahy binárních proměnných (X,Y,M,B) nebo aritmetická porovnání proměnné typu WORD s jinou proměnnou stejného typu, či s konstantou.

```
IF ((I1=1500) or ( D4>=I2)) and X0' THEN Y3
```

Jako operátory logických funkcí jsou přípustné "or", "and ", "xor" a operátor negace " ". Jako operátory aritmetického porovnání jsou přípustné "=", "<>", "<=", ">=". Při vyhodnocení aritmetických porovnání je možno využít i libovolných aritmetických výrazů.

```
IF I3 > (O5/2 + 100) THEN O5 = O5 + 1
```

V příkazové části je možný stejně jako u výrazového příkazu zápis několika výkonných příkazů oddělených oddělovačem ":". V případě, že je podmínka splněna, provedou se všechny tyto příkazy. Celý zápis instrukce musí být na jednom textovém řádku.

```
IF Y3 THEN Y1': D2 = 2 * I3 + I1 : O1 = O1 + 1
```

Příkaz END

Konec programu. Vrací běh programové smyčky na začátek.

7.5. Textové řetězce

Používání textových řetězců v programu je vyhrazeno pouze pro výstup textu na displej automatu a přiřazování textu je umožněno pouze do speciální funkční proměnné DISPLAY (viz dále). Nelze s nimi provádět žádné jiné operace. Textové řetězce se definují jako text uzavřený v uvozovkách a je možné je používat pouze jako konstantu (lze je tedy i definovat pod symbolickými názvy).

Ukázky možných způsobů použití textových řetězců :

```
"Dobry den " # TEXT1      ; definice symbolu TEXT1
DISPLAY=TEXT1             ; přiřazení textu TEXT1, tisk na displej
DISPLAY="Vítame Vas"      ; přímé přiřazení textového řetězce
```

7.6. Podprogramy

Jazyk SIMPLE umožňuje vytváření podprogramů a jejich volání z výrazového nebo podmínkového příkazu. Podprogramy jsou volány bez předávání formálních parametrů.

```
PID_REG           ; Volání podprogramu PID_REG
IF I0>123 THEN PRUMER ; Podmínkové volání podprogramu PRUMER
```

Definice podprogramu musí ve zdrojovém textu vždy předcházet jeho volání. Počet možných vnoření podprogramů je omezen na 5. Definice podprogramu je uvozena klíčovým slovem SUBROUTINE a názvem podprogramu. Tělo definice podprogramu je ukončeno klíčovým slovem RETURN.

```
SUBROUTINE PRUMER
    O1 = (I0 + I1)/2
RETURN
```

7.7. Konfigurační direktiva NetAddr

Nastavuje adresu stanice v síti. Může být v rozsahu 0 ... 30. U žádných automatů v jedné síti se nesmí vyskytovat shodné adresy. Automatů v síti tedy může být max. 31 !

```
NetAddr(20)      ; tento automat bude mít adresu 20
```

Tento parametr se předává do výsledného přeloženého souboru *.DNL a slouží jako informace pro následný přenos programu do automatu (prostřednictvím funkce "Program"). Je nanejvýš užitečné tuto direktivu používat, neboť prostředí při programování automatu již nastaví adresu automaticky a není ji tedy nutno při každém programování zadávat.

7.8. Omezení překladače

SIMPLE V2.10 má následující omezení :

Maximální velikost výstupního přeloženého kódu	30 kB
Maximální počet vnořených úrovní podprogramů	5
Maximální počet podprogramů	100
Maximální počet použitých textových řetězců	500

8. SPECIÁLNÍ FUNKCE A REGISTRY

Kromě běžných prostředků pro práci se vstupy, výstupy a vnitřními proměnnými má SIMPLE ještě tzv. speciální funkce. Jsou to vnitřní časovače, reálný čas, informace o rychlosti systému

Pro realizaci speciálních funkcí se využívají některé bity ze sady B0...B127 a některé registry ze sady W0...W127. Všechny proměnné, které mají speciální funkce, mají napevno přiřazeny symbolické názvy. Doporučujeme všude v programu používat tyto symboly - klesá potom možnost omylů.

I když většina bitů B a registrů W je zatím neobsazena, není vhodné je v programu využívat jako obecné proměnné (z důvodů kompatibility s dalšími verzemi překladače SIMPLE do budoucna).

8.1. RESET

Je speciální funkční bit, který je automaticky nastaven vždy po zapnutí automatu a též vždy po spuštění programu. Tento bit lze využít v programu na počáteční inicializaci a pak jej vynulovat, nebo se může použít i jinak, je však třeba počítat s tím, že po každém restartu automatu se tento bit nastaví.

8.2. Rychlost systému

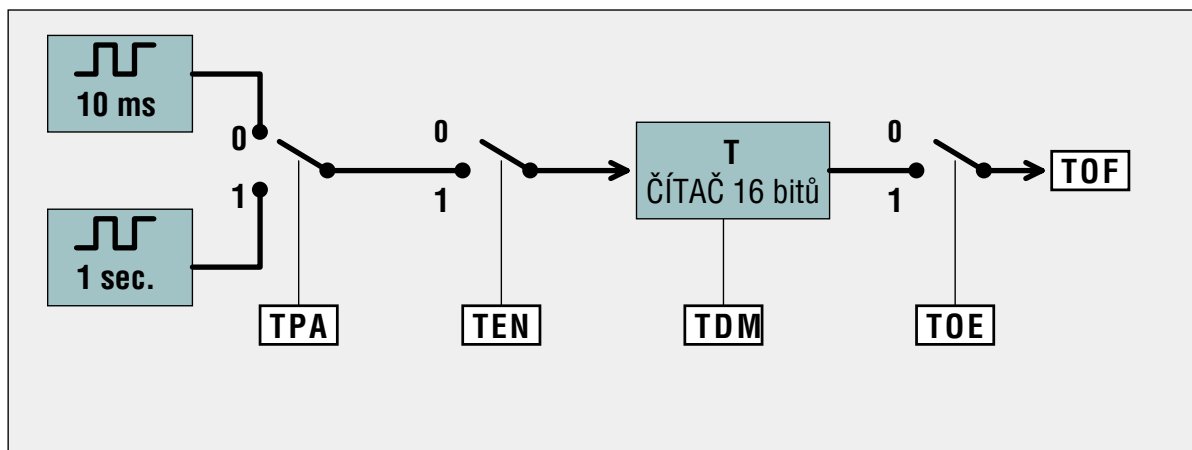
Při běhu programu v automatu je průběžně aktualizována proměnná **SPEED**, ve které je uveden počet průchodů celým programem (tedy de-facto počet obrátek hlavní smyčky) za sekundu. V aplikacích náročných na rychlost odezvy lze např. tento registr testovat a kontrolovat, zda se regulační program nedostává do časového skluzu.

8.3. Časovače

Časovače jsou reprezentovány speciálními funkčními registry (typu WORD) se symbolickými názvy T0 až T7. Všechny osm časovačů je zcela nezávislých a se shodnými funkcemi. Každý časovač lze kdykoliv přečíst, anebo do něj kdykoliv zapsat (se zapsanou hodnotou pak opět probíhá stejný proces čítání).

Činnost každého časovače souvisí se samostatnou šesticí funkčních bitů TEn, TPAn, TOEn, TOFn, TDMn, kde n = 0~7 specifikuje příslušný časovač. Tedy např. časovač T4 má svoji sadu řídicích bitů TEN4, TPA4, TOE4, TOF4, TDM4. Funkci těchto bitů naznačuje následující obrázek.

Časovač T a jeho řídicí bity :



- TEN** povolení čítání časovače : TEN=0 časovač stojí, TEN=1 časovač čítá
- TPA** zařazuje předdělič 1/100 : TPA=0 časovač čítá po 10ms, TPA=1 čítá po 1 s
- TOE** povoluje čítání přes přetečení
 TOE=0 časovač zastaví na 65535 (čítá-li nahoru), nebo na 0 (čítá-li dolů)
 TOE=1 časovač po dosažení meze nezastavuje, ale přetéká přes maximální rozsah čísla (při čítání nahoru 65535->0, při čítání dolů 0->65535)
- TOF** informace o přetečení časovače (musí být nulován programem)
- TDM** směr čítání časovače : TDM=0 časovač čítá nahoru, TDM=1 čítá dolů

8.4. Reálný čas

V jazyce SIMPLE mohou být programovány automaty systému PES, pokud mají osazen obvod reálného času (modifikace **R**). V tomto případě jsou uživateli přístupné ve speciálních funkčních registrech údaje o sekundách, minutách a hodinách reálného času. Kromě toho jsou k dispozici i registry s informací o dnech v měsíci, o měsících, letopočtu a dnech v týdnu. Hodinové údaje jsou v 24 hodinových cyklech. Kalendář nechtí přestupné roky. Dny v týdnu nejsou závislé na datu, čili se jedná o nezávislý čítač od 1 do 7. Je tedy na uživateli, kterým dnem bude začínat týden. Provede-li se však nastavení podle systémového času osobního počítače pomocí programu SETPES, potom týden začíná nedělí (Ne=1, Po=2, Út=3...). Funkční bit HOLD slouží k zachycení času. Po nastavení tohoto bitu se údaje v časových registrech nemění, čas však nadále běží na pozadí. Po vynulování tohoto bitu se opět v registrech objeví údaje odpovídající reálnému času. Nastavením bitu CLRSEC se zaokrouhluje reálný čas na celé minuty. Je vynulován obsah sekundového registru, a pokud je před nulováním obsah sekundového registru větší než 29, je zároveň inkrementován obsah minutového registru. Tento funkční bit je po zaokrouhlení času automaticky nulován.

Proměnné vyhrazené pro reálný čas:

SECOND	Sekundy reálného času	(0 ... 59)
MINUTE	Minuty reálného času	(0 ... 59)
HOURL	Hodiny reálného času	(0 ... 23)
DAY	Den v měsíci	(1 ... 31)
WEEK	Den v týdnu	(1 ... 7, Neděle=1, Pondělí=2....)
MONTH	Kalendářní měsíc	(1 ... 12)
YEAR	Roky ve století	(0 ... 99)

Funkční bity:

HOLD	Vzorkování reálného času
HOLD=0	registry jsou neustále aktualizovány
HOLD=1	registry "drží" poslední hodnoty
CLRSEC	Zaokrouhlení na minuty

UPOZORNĚNÍ :

Bity HOLD a CLRSEC nedoporučujeme v nových programech využívat, neboť v dalších verzích překladačů a operačních systémů v PLC již nemusí být podporovány.

Pozn.: Při čtení registrů reálného času na automatech, kde není obvod reálného času osazen, je třeba počítat s tím, že v registrech budou nesmyslné hodnoty.

Údaj letopočtu je pouze dvoumístný a obsáhne tedy pouze rozsah jednoho století. Je-li z nějakých důvodů třeba obsáhnout větší úsek, je nutné naprogramovat adekvátní softwarovou podporu pro počítání století.

Pro inspiraci je zde příklad :

```
D10 # TmpYear
D11 # CENTURY
subroutine PocitejStoleti          ; detekujeme pretečení roku pres nulu
  if (TmpYear > YEAR) then CENTURY = CENTURY+1
  TmpYear = YEAR
return
```

Pozn.: Proměnná CENTURY počítá století, ale protože se nikde neinicializuje (a ani to nelze, protože proces čtení století nesmí být narušen vypínáním a zapínáním automatu) je třeba ji jednorázově nastavit (nejlépe při nastavování data a roku).

8.5. A/D převodník

U automatů PES s analogovými vstupy se hodnoty těchto vstupů objevují v proměnných I0, I1 atd.

Rozsah těchto hodnot může být obecně 0...65535, jednotky v kterých je daná veličina měřena a její max. rozsah je dán typem analogového vstupu (tyto údaje jsou vždy uvedeny v technických údajích u jednotlivých typů automatů resp. modulů).

U standardních analogových vstupů je prováděna automatická digitální filtrace. Krok jednoho měření přes všechny vstupy trvá zhruba 700 ms. Analogové vstupy pro měření odporu mají tuto časovou konstantu ještě delší. Aktuální hodnoty jsou do proměnných IO..lx přepsány vždy na začátku programové smyčky a v celém průběhu programu smyčkou se nemění.

U základních analogových vstupů (první 4 u MPC300, prvních 6 u PES-K1 a PES-K10, 6 vstupů u M66) je implementována možnost kalibrace analogových vstupů. K tomuto účelu slouží speciální funkční proměnné CALIB0 až CALIB7. Po resetu programu je v těchto proměnných vždy hodnota 10000. Tato hodnota je chápána jako 1.0000 - a může být měněna uživatelským programem od 0.0 do 1.6000 (zápisem hodnot 0..16000 do proměnných CALIB0 - CALIB7).

Tato hodnota funguje jako násobná pro každý kanál, lze tedy chápat převod každého kanálu I_N takto :

$$I_N = (\text{nominální hodnota analog. vstupu}) \times (\text{CALIB}_N / 10000)$$

8.6. STACK (zásobník)

V zálohované paměti RAM je vyhrazen prostor pro uložení až 11776 šestnáctibitových hodnot (typu WORD) ve tvaru souvislého pole. Pouze typy s menší pamětí mají tento prostor menší (PES-M66, M66S). Lze sem ukládat všechny výsledky aritmetických operací a proměnné typu word (tedy proměnné I, O, D a W). Užitečné všude, kde je třeba po delší časový úsek hromadit měřené hodnoty, provádět statistická vyhodnocování, nebo si expandovat množství použitelných proměnných a registrů (pokud nestačí základní sada D0 - D63).

Zásobník se ovládá pomocí dvou speciálních funkčních registrů. Registr **POINTER** funguje jako adresa buňky v zásobníku (přípustný rozsah kontroluje BIOS automatu). Registr **STACK** zpřístupňuje vždy tu položku zásobníku, která je právě naadresována registrem POINTER.

Příklad uložení proměnné D5 na zásobník na adresu 100 :

POINTER = 100

STACK = D5

Příklad vyčtení zásobníku z adresy 0 do proměnné D1 :

POINTER = 0

D1 = STACK

Další příklady možných konstrukcí :

POINTER = 1500 + D15

if (STACK < 100) then STACK = STACK+1

D16 = STACK / 2

atd. atd.

8.7. Obsluha displeje a klávesnice

Kompaktní automaty PES-K a modulární automaty MPC302 mají shodnou filosofii ovládní displeje a klávesnice - sada funkčních registrů a funkčních bitů včetně jejich řídicích funkcí je vždy stejná.

Klávesnice

Vstupní informace z klávesnice je kódována do funkčního registru KBCODE tak, že každému tlačítku odpovídá určitý číselný kód. Pokud je stisknuto více kláves současně, je v proměnné KBCODE číselný kód jen jedné z nich. Tento kód se do KBCODE nastaví vždy na začátku programové smyčky a na jejím konci se automaticky vynuluje. Proměnná KBCODE je tedy po stisku klávesy aktivní právě po dobu jednoho průchodu programovou smyčkou. K opětovnému nastavení KBCODE je třeba tlačítko pustit a znovu stisknout (výjimkou je autorepeat - viz dále). Pokud je klávesnice v klidu, je KBCODE=0.

Číslice a desetinná tečka mají číselný kód zvolen tak, aby přesně odpovídal znakové tabulce ASCII, lze je tedy pod tímto kódem přímo vytisknout na displej.

klávesa	kód v KBCODE	ASCII znak
šipka vlevo	1	
šipka vpravo	2	
šipka nahoru	5	
šipka dolů	6	
ESC	3	
ENT	4	
+/-	7	
F1	8	
F2	9	
F3	10	
0	48	"0"
1	49	"1"
2	50	"2"
3	51	"3"
4	52	"4"
5	53	"5"
6	54	"6"
7	55	"7"
8	56	"8"
9	57	"9"
.	46	"."

POZN.:

V tabulce je uvedeno všech 21 kláves, které jsou na automatu PES-K10. Typy s menší klávesnicí (MPC302) mají jen prvních 6 kláves, kódování je však kompatibilní.

Autorepeat

Aby nebylo nutné pro určité akce (např. pro posuv kurzoru, rolování menu apod.) vícekrát mačkat tutéž klávesu, je možné nastavit "AUTOREPEAT". Pokud stiskneme některé tlačítko a držíme je stisknuté, potom po určité (nastavitelné) prodlevě bude do KBCODE znovu nastavován kód tlačítka (s nastavitelnou rychlostí opakování). Efekt je tentýž jako bychom neustále rychle tiskli tlačítko.

Funkce se zapíná nastavením bitu KBREPEN, počáteční prodleva se nastavuje registrem KBDELAY (v desítkách milisekund) a délka intervalu mezi jednotlivými "pseudo-stisky" registrem KBREPEAT (v desítkách milisekund).

Zapnutí funkce autorepeat s počáteční prodlevou 1 sec. a opakováním po 0.2 sec. :
 KBREPEN : KBDELAY = 100 : KBREPEAT = 20

Seznam funkčních proměnných, specifických pro klávesnici a displej:

NÁZEV	FUNKCE	POČ. HODNOTA	TYP/ UMÍSTĚNÍ
POSITION	pozice znaku na obrazovce	nedefinováno	word W34
FORMAT	formát pro tisk (vyjma textů)	nedefinováno	word W35
KBCODE	kód stisknuté klávesy	0	word W36
KBDELAY	autorepeat-startovací prodleva	100	word W37
KBREPEAT	autorepeat-opakovací interval	10	word W38
KBREPEN	zapnutí funkce autorepeat	0 (vypnuto)	bit B50
KBSOUND	zapnutí akustické indikace	1 (zapnuto)	bit B51

Akustická indikace

Stisk tlačítka (a případně i "pseudo-stisky" při zapnuté funkci AUTOREPEAT) je možné automaticky indikovat krátkým pípnutím vestavěného bzučáku. Funkce se zapíná nastavením bitu KBSOUND (je automaticky nastaven vždy po zapnutí).

Displej

Filosofie ovládání displeje je u všech automatů (obsahujících displej) stejná, i když mohou mít LCD s různým počtem řádků a různým počtem znaků. Texty, čísla nebo uživatelské symboly se vždy tisknou na virtuální obrazovku velikosti 4 řádky x 40 znaků. Pozice znaků na obrazovce se počítají od levého horního rohu počínaje nulou. Ve skutečnosti je potom zobrazeno jen tolik řádků a sloupců z virtuální obrazovky, kolik umožňuje použitý displej.

Díky tomuto způsobu ovládání jsou programy navzájem přenositelné. Vytvoříme-li program pro displej s 4x20 znaky, poběží úplně stejně i na displeji s 2x16 znaky, ovšem spodní 2 řádky a poslední 4 znaky již nebudou zobrazeny.

Pro tisk na displej je vyhrazena speciální proměnná DISPLAY, do které lze kromě konstant a proměnných přiřazovat i text (píše se do uvozovek). Pozici, odkud se bude vypisovat, lze volně nastavit pomocí funkční proměnné POSITION. Pokud chceme tisknout položky za sebou, není třeba počítat nové hodnoty do proměnné POSITION, neboť obsah této proměnné se automaticky po tisku každého znaku zvětší o 1.

Tisk textů je přímý, tj. to co je napsáno v uvozovkách a přiřazeno do proměnné DISPLAY je zobrazeno od pozice dané proměnnou POSITION.

Desetinná místa fungují tak, že ze zadaného čísla se poslední číslice (tolik, kolik je nastavených desetinných míst) chápou jako desetiny. Tedy např. číslo 27340 vytisknuté formátem 0 bude zobrazeno jako 27340, ale vytisknuté formátem 9 už bude zobrazeno jako 273.40

FORMÁTY PRO TISK ČÍSEL (0 ... 75)		FORMAT = TYP + DECI
TYP	0	zarovnání čísla vpravo, vstup je kladné číslo 0..65535
	20	zarovnání čísla vlevo, vstup je kladné číslo 0..65535
	40	zarovnání čísla vpravo, vstup je znaménkové číslo -32768..+32767
	60	zarovnání čísla vlevo, vstup je znaménkové číslo -32768..+32767
DECI	0	x 1 místo, bez desetinné tečky
	1	xx 2 místa, bez desetinné tečky
	2	xxx 3 místa, bez desetinné tečky
	3	xxxx 4 místa, bez desetinné tečky
	4	xxxxx 5 míst, bez desetinné tečky
	5	x.x 1 místo před, 1 místo za desetinnou tečkou
	6	xx.x 2 místa před, 1 místo za desetinnou tečkou
	7	xxx.x 3 místa před, 1 místo za desetinnou tečkou
	8	xxxx.x 4 místa před, 1 místo za desetinnou tečkou
	9	x.xx 1 místo před, 2 místa za desetinnou tečkou
	10	xx.xx 2 místa před, 2 místa za desetinnou tečkou
	11	xxx.xx 3 místa před, 2 místa za desetinnou tečkou
	12	x.xxx 1 místo před, 3 místa za desetinnou tečkou
	13	xx.xxx 2 místa před, 3 místa za desetinnou tečkou
	14	x.xxxx 1 místo před, 4 místa za desetinnou tečkou
15	x.xxxxx 1 místo před, 5 míst za desetinnou tečkou	
FORMÁT PRO TISK ZNAKŮ PODLE TABULKY ASCII		
FORMAT = 120	vytiskne 1 znak, odpovídající danému číslu podle tabulky ASCII	
FORMÁT PRO DEFINOVÁNÍ VLASTNÍCH ZNAKŮ A GRAF. SYMBOLŮ		
FORMAT = 121	definuje 1 grafický řádek v uživatelské sadě znaků - viz dále	

Interpretace záporných čísel (formáty 40 ... 75)

Vstupní čísla jsou chápána jako znaménková v dvojkovém doplňku, kde nejvyšší bit má funkci znaménka (numerický typ *integer*, resp. *int*). Až do hodnoty 32767 je interpretace těchto čísel stejná jako u neznaménkových. Odečtením většího čísla od menšího vznikne podtečení. Jazyk SIMPLE pracuje s neznaménkovými čísly - má rozsah hodnot 0...65535.

Pokud odečteme např. **5 - 7**, dojde k podtečení rozsahu a dostaneme zdánlivě nesmyslnou hodnotu **65534**. Pokud však tuto hodnotu budeme interpretovat ve dvojkovém doplňku se znaménkem na nejvyšším bitu a tedy znaménkově - dostáváme hodnotu **-2**. Pokud toto číslo budeme tisknout např. formátem 0, vytiskne se na displej **65534**, pokud jej budeme tisknout např. formátem 40, vytiskne se **-2**. Lze tedy i v jazyce SIMPLE tímto způsobem dostat jako výsledek záporné číslo.

Příklad : Automat MPC300 s modulem D se vstupy Pt100 lze nastavit tak, že udává hodnotu analogových vstupů přímo v desetinách Kelvina. V automatu přiřazujeme tento vstup např. do síťové proměnné D40 a na terminálu PES-K1 chceme tento údaj zobrazit ve stupních Celsia. Zkusíme to např. takto :

```
D1=D40-2732      ; odečteme 273.2K výsledek je v 0.1 0C
FORMAT = 67      ; formát pro záporná čísla
                  ; zarovnání vpravo, tvar čísla XXX.X
DISPLAY = D1     ; zobrazení
```

Pokud je tedy v D40 číslo 2627 (262.7 K resp. -10.5 °C), po odečtení dostaneme sice v jazyce SIMPLE zdánlivý nesmysl 65431, ale formáty pro znaménková čísla jej budou správně interpretovat jako -105. Volbou formátu s DECI=7 pak oddělíme správný počet desetinných míst, rezervujeme navíc celkovou délku čísla na 3 místa a dostáváme výsledných **-10.5** !

8.9. Formát pro tisk znaků

Formát 120 interpretuje vstupní hodnotu jako číslo znaku a vytiskne na displej 1 znak daný tímto číslem. Tento formát umožňuje tisknout i znaky, které nejsou běžně dostupné a nelze je tedy při psaní programu napsat z klávesnice do zadání textu pro textový výstup. Kód znaku musí být v rozsahu 0 ... 255.

KÓD	TYPY ZNAKŮ
0...7	vlastní předefinovatelné znaky, grafické symboly
32...122	abecední a číslicové znaky, kódované podle standardu ASCII
160...255	další užitečné znaky, specifické pro daný displej (např. japonská abeceda)

Např. znak C lze v jazyce SIMPLE vytisknout několika způsoby :

- a) DISPLAY = "C"
- b) FORMAT = 120 : DISPLAY = 67
- c) D1 = 67 : FORMAT = 120 : DISPLAY = D1

8.10. Definování vlastních grafických symbolů

Používané typy displejů umožňují vytvořit až 8 vlastních grafických symbolů. Znaků jsou kódovány pod čísly 0 až 7 a lze je tisknout prostřednictvím formátu 120 (viz výše). Tuto zajímavou funkci můžeme využít např. pro tvorbu a tisk znaků české abecedy.

Všechny znaky jsou tvořeny maticí z 8 grafických řádků po 5 bodech. U standardní pevné sady znaků se spodní řádek nevyužívá. U sady programovatelných znaků můžeme využít všech 8 řádek. Celkem tedy můžeme definovat 64 řádků, tvořících 8 kompletních znaků.

Definování znaků začíná nastavením hodnoty 121 do proměnné FORMAT. Tím se displej přepne do režimu programování znaků. Proměnné DISPLAY a POSITION mají nyní jinou funkci. Proměnná POSITION určuje, který grafický řádek editujeme a do proměnné DISPLAY zapisujeme data, určující obsah řádku. POSITION se i v tomto režimu po zápisu řádku automaticky zvětší o 1 a posune se tak na další řádek. Programování znaků ukončíme nastavením proměnné FORMAT na jinou hodnotu. Po návratu z programovacího režimu se do proměnné POSITION nevrací původní hodnota, je tedy nutné ji před dalším tiskem opět nastavit.

Umístění pozic grafických řádků :

znak 0	znak 1	znak 2	znak 3	znak 4	znak 5	znak 6	znak 7
0	8	16	24	32	40	48	56
1							
2							
3							
4							
5							
6							
7	15	23	31	39	47	55	63

Každý bod řádku má svoji numerickou váhu (odpovídá binárnímu kódování). Numerickou reprezentaci jednoho grafického řádku získáme sečtením hodnot odpovídajících jednotlivým bodům.

Číselné vyjádření bodů na grafickém řádku :


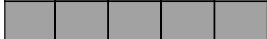
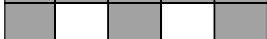



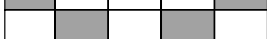

16	8	4	2	1
----	---	---	---	---

Příklad

Chceme zadefinovat grafický symbol (podle obr.) a zobrazit. Znak zadefinujeme například pod kódem 2. Nejprve spočítáme numerické vyjádření grafických řádků (viz obr.), potom napíšeme podprogram pro jazyk SIMPLE, který vykoná vše potřebné :

```

subroutine TISKNI_OBLUDU
  FORMAT=121          ; formát pro definování znaků
  POSITION=16           ; 0. grafický řádek znaku 2
  DISPLAY=14          ; zápis grafických řádků
  DISPLAY=31
  DISPLAY=21
  DISPLAY=27
  DISPLAY=31
  DISPLAY=17
  DISPLAY=10
  DISPLAY=14
  FORMAT=120          ; budeme tisknout znak podle kódu
  POSITION=0            ; na 1. pozici na 1. řádku
  DISPLAY=2           ; bude se tisknout znak číslo 2
return
    
```

GRAFICKÁ PODOBA	DATA
	14
	31
	21
	27
	31
	17
	10
	14

8.11. Jak funguje zobrazování

Tisk na virtuální obrazovku a vlastní fyzický zápis dat do obvodů displeje jsou dva zcela nezávislé děje, které běží v automatu paralelně. Odezva elektroniky displeje je totiž pomalá a kdyby měl uživatelský program čekat, až se skutečně vytiskne na displej nějaký delší text, došlo by v daném bodě k jeho nepřijatelnému zdržení. Celý tisk přes proměnnou DISPLAY se proto uloží do paměti RAM (zde je ta virtuální obrazovka) a zároveň s programem běží na pozadí automatický proces, který odtud stále čte znaky a posílá je na displej.

Tato metoda vůbec nezdržuje hlavní program, avšak skýtá jednu záludnost. Chceme-li například smazat řádek displeje a vytisknout na něj novou informaci, můžeme to provést např. takto :

```

subroutine NOVY_RADEK
  POSITION=40           ; budeme pracovat s 2. řádkem
  DISPLAY="           " ; tisk mezer přes celý řádek
  POSITION=40
  DISPLAY="NOVY TEXT" ; na začátek řádku nový text
return
    
```

Přepsání celého displeje trvá asi 200 ms. Pokud budeme tuto proceduru volat často (alespoň 3x až 4x za sekundu), může se stát, že zrovna po vytisknutí prázdného řádku se několik těchto znaků přenesou na displej a bude trvat asi 200 ms, než dojde k jejich přepisu novou hodnotou. A shodou náhod může být v tento okamžik obslužný program zase v bodě, kdy do inkriminované oblasti opět píše mezery... Obraz potom různě chaoticky problikává.

Řešením je buď omezit frekvenci volání takové procedury (max. 2x za sekundu), nebo tisknout na displej tak, aby nedocházelo bezprostředně po sobě k přepisu týchž pozic různou informací. Např. takto :

```
subroutine NOVY_RADEK
  POSITION=40           ; budeme pracovat s 2. řádkem
  DISPLAY="NOVY TEXT  " ; tisk textu a mezer až do konce
return
```

Další metodou, která tento problém zcela řeší, je důsledné používání formátů rezervujících určitý prostor pro tisk čísla (viz kap. Formátování tisku číselných hodnot). Tím je zajištěno, že vytištěné číslo zabere na displeji stále stejný počet znaků (tisková funkce jej doplní mezerami) bez ohledu na svou momentální číselnou velikost (pokud ovšem zvolíme dostatečnou rezervu). Pak můžeme jednotlivé textové a číselné elementy naskládat na displej těsně vedle sebe. Tím pádem k žádnému přepisování a mrkání nebude docházet ani když budeme zobrazování hodnot provádět v každém průchodu smyčky.

Například :

```
subroutine ZobrazKusy      ; podprogram tisku na displej
  POSITION=43
  FORMAT=2                ; rezervuji se 3 znaky pro cislo, zarovnani vpravo
  DISPLAY="Pocet:"        ; text
  DISPLAY=Kusy            ; cislo - promenna Kusy (predpoklad, ze je mensi nez 999)
  DISPLAY=" ks"          ; text
return
```

Tento podprogram je možno volat jakkoli rychle, protože všechny položky mají své pevné pořadí a místo, nic se nepřekrývá (tedy až do okamžiku, kdy proměnná Kusy bude větší než 999).

Je třeba na druhou stranu počítat s tím, že tisk na displej (přiřazování do proměnné DISPLAY) zabere poměrně dost strojového času a příliš mnoho tisků v každém průchodu programové smyčky výrazně zvýší dobu průchodu programu a sníží tak celkovou rychlost.

8.12. Seznam speciálních funkčních proměnných

Typ	Symbol	Umístění	Popis funkce
BIT	TEN0..TEN7	B0..B7	povolení čítání
	TDM0..TDM7	B8..B15	čítání dolů
	TPA0..TPA7	B16..B23	předdělič stem
	TOE0..TOE7	B24..B31	čítání přes přetečení
	TOF0..TOF7	B32..B39	příznak přetečení
	CLK0..CLK7	B40..B47	asynchronní impuls na časovač
	HOLD	B48	vzorkování reálného času.
	CLRSEC	B49	zaokrouhlení na minuty.
	KBREPEN	B50	zapnutí funkce autorepeat
	KBSOUND	B51	zapnutí akustické indikace
	RESET	B126	počáteční inicializace systému
WORD	T0..T7	W0..W7	šestnáctibitové časovače
	SECOND	W8	sekundy reálného času. (0~59)
	MINUTE	W9	minuty reálného času. (0~59)
	HOUR	W10	hodiny reálného času. (0~23)
	DAY	W11	dny v měsíci. (1~28..31)
	MONTH	W12	kalendářní měsíc. (1~12)
	YEAR	W13	roky století. (0~99)
	WEEK	W14	dny v týdnu (1~7)
	SPEED	W15	rychlost systému
	STACK	W16	aktuální položka zásobníku
	POINTER	W17	ukazatel do zásobníku
	CALIB0..CALIB7	W18..W25	kalibrace základních analogových vstupů
	ADCMODE	W26	nastavení módu A/D převodníku
	POSITION	W34	pozice znaku na obrazovce
	FORMAT	W35	formát pro tisk (vyjma textů)
	KBCODE	W36	kód stisknuté klávesy
	KBDELAY	W37	autorepeat-startovací prodleva
	KBREPEAT	W38	autorepeat-opakovací interval

9. FUNKCE AUTOMATU

9.1. Spuštění automatu

Po zapojení napájení automat nejprve čeká zhruba 1-2 sec. na případnou komunikaci po lince RS485 z počítače PC. Poté zahájí přípravy na spuštění programu.

- 1) Zjistí, zda je v paměti zatažen uživatelský program.
- 2) Zkontroluje, zda souhlasí kontrolní součet paměti, kde je uložen program.

Pokud jsou vše v pořádku, automat odstartuje uložený program.

9.2. Obsluha proměnných

Ač bylo uvedeno, že speciální funkční proměnné (typu B a W), je možno použít libovolně u jakéhokoliv typu PLC, je vhodné ještě upřesnit, že pokud použité funkční proměnné ovládají funkce, které daný typ automatu nemá, překladač sice práci s takovými proměnnými umožní, ale pochopitelně nebudou mít příslušnou funkci. Například nemá-li použitý automat reálný čas (např. PES-M66), překladač sice povolí práci s příslušnými registry, ale tyto akce budou poskytovat nedefinované výsledky.

Obsluha vstupů a výstupů

Při běhu programu automat pracuje s proměnnými uloženými v paměti a to i tehdy, jedná-li se o vstupy/výstupy. V tomto případě se vlastně pracuje nad záchytnými registry, kam jsou přepisovány úrovně (logické i analogové) z fyzických vstupů, resp. odkud jsou obsahy přepisovány na fyzické výstupy. Tento přepis probíhá vždy na začátku programové smyčky. To v praxi znamená, že se během jednoho běhu programové smyčky nemohou změnit obsahy vstupních proměnných **X** a **I** a zároveň se jakékoliv změny výstupních proměnných **Y** a **O** během běhu smyčky na skutečných výstupech neprojeví.

Tedy nastavíme-li například na jednom programovém řádku vypnutý výstup Y2 do 1 a na následujícím řádku ho čteme, přečteme samozřejmě 1. Ovšem přepis proměnné Y2 na fyzický výstup automatu se provede až na konci programové smyčky. Takže finální hodnota, která na výstupu nakonec bude, je ta která byla do Y2 v programové smyčce přiřazena jako poslední. To mimo jiné znamená i to, že je možné například na začátku smyčky výstup vynulovat a pokud je před koncem programu vždy znovu zapnut, pak na fyzickém výstupu bude nepřerušovaný stav stálého zapnutí.

Při psaní uživatelských programů je velmi důležité si tento princip uvědomit a jeho opomenutí může v určitých případech vést k problémům a zdánlivě špatné funkci programu. Jinak je tomu u časovačů - tam probíhá čítání neustále - tedy při testování obsahu čítače/časovače je třeba počítat s tím, že hodnoty T0-T7 se při běhu programu průběžně mění.

9.3. Popis síťové komunikace

Na automatech PES je používán protokol PESNET verze 3 (síťový protokol typu "multi-master", "token-passing"). Umožňuje komunikaci mezi automaty navzájem (podmínkou je pouze nastavení stejné komunikační rychlosti u všech automatů v síti a nastavení odlišných adres pro všechny automaty v síti). Stanic může být zapojeno ve společné síti max. 31. Stanice navzájem komunikují v logickém kruhu, jejich pořadí v kruhu je dáno adresou. Pořadí je vzestupné, za poslední adresou (30) následuje opět adresa 0.

Při startování, resp. po výpadku komunikace probíhá časování ochranné doby po jejímž uplynutí zahajují stanice vysílání. Tato doba je navíc závislá na adrese stanice (aby se omezily zbytečné kolize) - je tedy vhodné při instalaci sítě přiřazovat stanicím adresy vždy od spodních hodnot (nejlépe od nuly) - zkrátí se tak čekací doby při restartech sítě.

Protokol umožňuje začlenění libovolné další stanice do sítě, nebo naopak vypnutí kterékoliv stanice v síti v libovolný okamžik. K obnovení síťového provozu v takovýchto případech dochází automaticky a trvá zpravidla několik sekund. Při zapojení další stanice do již aktivní sítě, k přerušení provozu nedochází, začlenění stanice do logického kruhu v tomto případě trvá o něco déle, maximálně však 20 sekund.

Síťové proměnné

Aby se dalo jednoduše komunikovat mezi automaty pomocí prostředků jazyka SIMPLE, byl zvolen tento způsob :

Horní polovina uživatelských bitů **M** (tedy **M63** až **M127**) a horní polovina uživatelských proměnných **D** (tedy **D32** až **D63**) je sdílená všemi automaty v síti. Pokud tedy program v automatu zapíše hodnotu do některé této proměnné (nastaví nebo nuluje bit - např. `M64 : M65'` nebo zapíše hodnotu do proměnné např. `D32=3568 : D33=D11 / 120` apod.), automat při nejbližší možné příležitosti odvysílá do sítě adresu a hodnotu této proměnné. Ostatní automaty tuto zprávu zachytí a automaticky si do těchto proměnných zapíšou nové hodnoty.

9.4. Tipy pro používání sítě

Vysílání hodnot do sítě se děje vždy po zápisu do příslušné proměnné v automatu. Chceme-li tedy, aby některá důležitá hodnota byla stále k dispozici (např. po RESETU nastavíme v programu hodnotu takto:

```
if RESET then D35=2500 : RESET'
```

Pokud tuto hodnotu potřebujeme nutně v druhém automatu a tento automat zapneme až po startu prvního, už se tato hodnota přenášet nebude.

Je tedy vhodné u takovýchto důležitých proměnných provést čas od času "občerstvení" - vyhradíme si např. čítač T7 pro tyto potřeby a např. každé 2 sekundy provedeme nové přiřazení:

```
if RESET then TEN7 : TPA7 : RESET'  
if (T7 > 2) and M1 then M64  
if T7 > 2 then T7=0 : D35=OTACKY : D36=POLOHA
```

Samozřejmě můžeme přiřazovat do síťové proměnné neustále při každém průchodu programovou smyčkou, je však třeba počítat s tím, že datová propustnost sítě klesne. I tak je však zajištěno, že tabulky proměnných se prohledávají cyklicky a nemůže se tedy stát, že by některá často používaná síťová proměnná zablokovala přenos ostatních proměnných. Rovněž se nemůže stát, aby některý automat (který zapisuje do velkého počtu síťových proměnných) "kradl" vysílací čas na úkor automatů ostatních.

Pokud tedy pro chod celé aplikace není třeba zvlášť vysoké odezvy (řádově 0.1 sec), není nutné se problémem zatížení sítě nikterak zabývat.

Závěrem

Autoři integrovaného prostředí PESpro věří, že tento produkt Vám usnadní vývoj aplikací pro průmyslové automaty PES a stane se tak dobrým pomocníkem při Vaší práci.