PROGRAM DDEAP

edice 04.2021
verze 1.7

Návod k použití programu DDEAP

http:/www.micropel.cz

Obsah

DdeAP

DdeAP je program vytvořený firmou MICROPEL s.r.o. spustitelný pod operačním systémem Microsoft Windows. Program slouží ke zprostředkování datového spojení mezi sítí automatů MICROPEL a dalšími programy s komunikací DDE (Dynamic Data Exchange). Program DDEAP tak z hlediska možností přístupu k automatům kanálem DDE nahrazuje starší program MICROPEL DataServer, jehož podpora již byla ukončena. Kromě funkce DDE serveru může program, stejně jako DataServer, posloužit i jako TCP/IP server překlápějící požadavky několika současně běžících vizualizací, komunikujících protokolem EPNP, do sítě automatů.

Typicky se program DdeAP použije společně s vývojovými prostředky MICROPEL StudioG nebo StudioWin k nahrání vytvořených programů do automatů MICROPEL a k ladění těchto programů za pomoci sledovačů. DdeAP též poslouží jako přístupový bod (AP) do sítě automatů MICROPEL pro libovolný vizualizační software s komunikačním rozhraním DDE a možností nastavit odesílání dotazů a zpracování odpovědí se syntaxí definovanou dále v dokumentu.

DdeAP neumožňuje provádět nastavení automatů MICROPEL, k provádění nastavení nebo aktualizace firmware je třeba použít program MICROPEL PlcConfig.

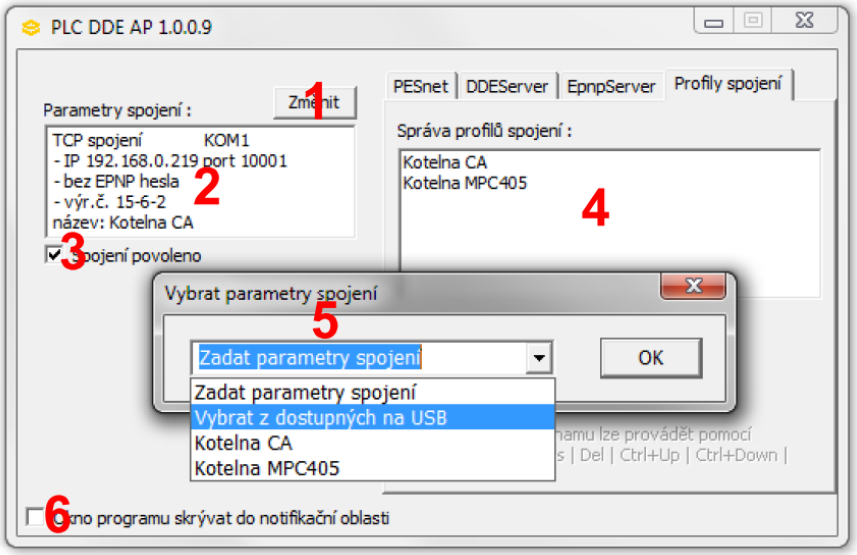
Propojení na komunikační převodníky CA4, CA3 či CA21 již není programem podporováno!

Propojení na síť automatů MICROPEL

Program DdeAP umožňuje sestavit spojení s automatem/převodníkem řady 400. Připojený automat/převodník pak bude zajišťovat propojení i na dalším automaty, které nalezne v síti PESnet, případně i EXbus - automat/převodník musí mít pro linku RS485 nastaven příslušný ovladač, PESnet resp. EXbus-Master. Zde je třeba zmínit, že program DdeAP automaticky provede vyhledání dostupných automatů v síti PESnet, ale síť EXbus nijak neprohledá a ani nezobrazuje její stav.
Spojení lze navázat na lince USB při přímém propojením automatu/převodníku s PC vhodným USB kabelem. V případě, kdy je automat/převodník vybaven modulem ETHERNET či modulem GSM s platnou SIM kartou a povolenými datovými přenosy GPRS, lze s ním navázat spojení též protokolem TCP/IP.

Postup nastavení spojení

K popisu postupu nastavení spojení poslouží následující obrázek okna programu:



Obr. 1 - Okno programu DdeAP s vyvolaným selektorem výběrem spojení

Postup je pak následující:

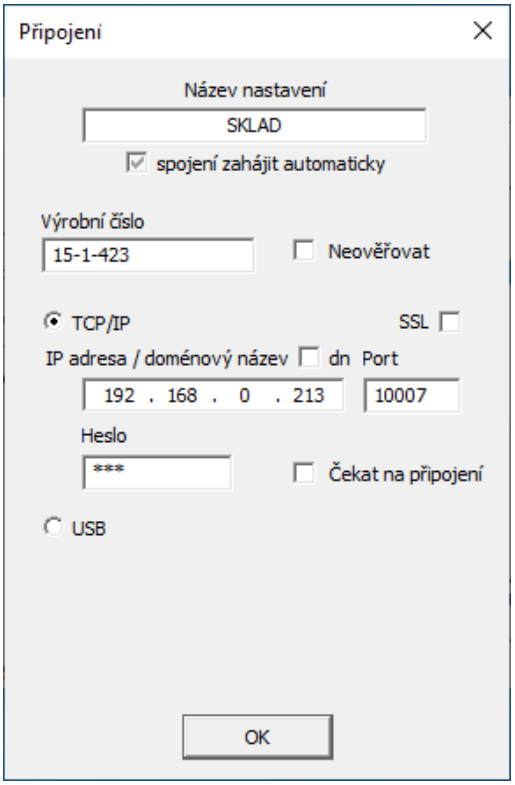
1. Klikneme na tlačítko (1), zobrazí se okno selektoru (5).
2. V selektoru (5) zvolíme požadovaný typ spojení a klikneme na OK. Nabídka v selektoru bude obsahovat i dříve přednastavené parametry spojení ze seznamu profilů spojení (4).
3. Pokud je zaškrtávacím políčkem (3) spojování povoleno, bude spojování s automatem/převodníkem ihned zahájeno.
    - Při výběru prvního nabízeného řádku v selektoru (5) se nejdříve zobrazí dialogové okno pro zadání požadovaných parametrů spojení.
    - Při výběru druhého nabízeného řádku z nabídky dojde k vyhledání všech dostupných automatů připojených na USB. S nalezením jednoho či více připojených automatů dojde k zobrazení volby, jestli, a který automat chceme připojit.

Zaškrtávacím políčkem (3) lze spojení s automatem/převodníkem ukončit nebo naopak spojování zahájit. V případě, kdy program není na automat/převodník propojen, bude informační okno (2) ke spojení „zašedlé“ (neaktivní).
Přepínač (6) povoluje skrytí okna programu do notifikační oblasti při jeho minimalizaci, jinak bude okno po minimalizaci dostupné též klasicky na hlavním panelu (mezi právě spuštěnými programy).

Dialog nastavení parametrů spojení

Dialog se objeví při provádění změny parametrů spojení s automatem / komunikátorem, poté, co vybereme ***Zadat parametry spojení,*** nebo se též otevře při vytváření nové a editaci existující položky seznamu profilů spojení.

Text editačního okna ***Název nastavení*** bude zobrazen v informačním okně ke spojení (v prvku (2) na Obr. 1). A zároveň má při editaci profilu spojení význam jména profilu. Třístavový přepínač ***automatické zahájení spojení*** určuje režim zahájení spojování po výběru tohoto nastavení spojení a pak též i po každém spuštění DdeAP. Tři různé stavy znamenají: (zaškrtnuto) zahajovat spojení ihned / (třetí, zašedlý stav) řídit se posledním stavem tlačítka povolení spojení / (odškrtnuto) spojení nezahajovat, počkat vždy na aktivaci tlačítkem povolení spojení. Do editačního okna ***Výrobní číslo*** lze opsat výrobní číslo automatu / komunikátoru, aby bylo znemožněno, že se DdeAP neočekávaně automaticky spojí s jiným dostupným automatem / komunikátorem řady 400. Ověřování výrobního čísla při spojování se vynechá, pokud zatrhneme ***Neověřovat***. Pokud nastavujeme propojení typu USB a výrobní číslo nezadáme, DdeAP pak při nalezení automatů / komunikátorů řady 400 na USB zobrazí nabídku, s jakým z nalezených zařízení se má propojit (i pokud byl nalezen jen jeden automat / komunikátor).



U připojení TCP/IP, které vybereme pro spojení s automatem / komunikátorem vybaveným modulem ETHERNET nebo GSM-GPRS, nastavíme IP adresu a komunikační port, na nichž má DdeAP automat / komunikátor očekávat. Zaškrtnutím přepínače ***dn*** je možno alternativně namísto IP adresy uvést název, typicky internetovou adresu, na níž se cíl nachází. V případě, kdy je automat / komunikátor nastaven do režimu aktivního připojování (tzv. na DataServer) – např. u GPRS spojení, kdy IP adresa přidělená mobilním operátorem není předem známa - vybereme ***Čekat na spojení***. V tom případě se vyplněná hodnota IP adresy nijak neuplatní. Do pole ***Heslo*** se vyplňuje číselné heslo zabezpečení EPNP komunikace, stejné heslo musí být nastaveno i v automatu / komunikátoru. V případě heslem nezabezpečené komunikace lze ponechat pole prázdné, nebo doplnit nulu. Zaškrtnutý přepínače ***SSL*** se použije pro programování nebo ladění automatu / komunikátoru připojeného k vizualizačnímu serveru MICROPEL.

Pozn. U TCP/IP spojení, kdy má DdeAP čekat na připojení, je třeba zajistit, aby byl program DdeAP povolen v bráně Windows Firewall.

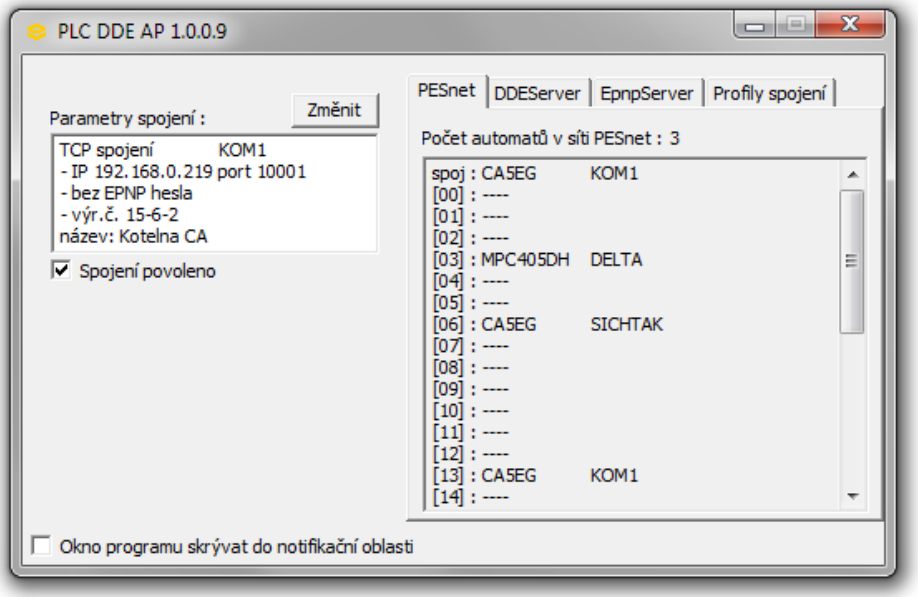
Pozn. TCP/IP adresa ani port se nemusí nutně shodovat s hodnotami, které má nastaveny (přiděleny) automat / převodník. Např. v případě, kdy je převodník přístupný pomocí sítě Internet a je od Internetu oddělen směrovačem (routerem), bude potřeba do parametrů spojení zadat internetovou adresu směrovače a vyplnit port takový, jaký je směrovačem vyhrazen pro propojení na převodník.

Profily spojení

Na kartě „Profily spojení“ programu DdeAP lze vytvářet a upravovat seznam přednastavení parametrů spojení na automat/převodník. Jednotlivým přednastavením se přiřadí názvy, veškerá přednastavení pak budou pod svými názvy nabízena k použití při provádění změny spojení (viz „Postup nastavení spojení“).

Dostupné automaty v PESnet

Dostupné automaty v síti PESnet lze pozorovat na kartě „PESnet“ v okně programu. S navázáním spojení s automatem/převodníkem dojde k vyhledání všech dostupných automatů na lince PESnet automatu/převodníku (jestliže má spuštěn ovladač PESnet linky RS485). Seznam je pak průběžně aktualizován. U každé aktivní PESnet adresy se v seznamu zobrazí typ zařízení, jaké se na této adrese hlásí, a uživatelský název zařízení, jestliže jej zařízení poskytuje. Převodník/automat bude uveden v zobrazeném seznamu figurovat 2x, jednak na prvním řádku seznamu, pak též na řádku odpovídajícímu jeho adrese v síti PESnet. Okno DdeAp se seznamem dostupných automatů ukazuje Obr. 3. Viditelná je vždy jen část seznamu, zobrazení seznamu lze posouvat posuvníkem napravo.



Obr. 3 - Okno programu DdeAP se zobrazeným seznamem dostupných automatů na PESnet

Parametry programu

Programu DdeAP lze při spuštění předat z příkazové řádky několik volitelných parametrů. Seznam akceptovaných parametrů je následující:

-h nebo **--hidden**

- Spouštěná instance programu bude po startu minimalizována do notifikační oblasti.

-mi nebo **--multi-inst**

- Dovoluje následné spuštění dalších instancí programu DdeAP. Pokud spustíme instanci programu bez tohoto parametru, nebude již nadále možno spustit další instanci.
- Od DdeAP verze 1.0.2.1 lze použít konstrukci parametru -mi=Text, která dovolí spustit novou instanci programu, jestliže již není spuštěna instance volaná se stejným parametrem za rovnítkem. Hodnota parametru (zde Text) se zároveň použije jako název instance programu.

-cf= nebo **--config-file=**

- Nastavuje cestu ke konfiguračnímu souboru, který má spouštěná instance programu použít. Cestu je třeba specifikovat za rovnítkem, nejlépe v uvozovkách. Bez použití tohoto parametru se jinak bude pracovat s konfiguračním souborem „ddeap.cfg“ v adresáři spuštění programu. Parametr se typicky využije při současném spouštění několika instancí programu.

-cpd= nebo **--connect-profiles-dir=**

- Nastavuje adresář umístění souboru „connections.cfg“ s uloženými profily spojení, který má spouštěná instance programu použít. Adresář je třeba specifikovat za rovnítkem, raději v uvozovkách. Bez použití tohoto parametru se jinak bude pracovat se souborem umístěným v podadresáři „MICROPEL“ v adresáři vyhrazeném pro data aplikací uživatele přihlášeného ve Windows. Parametr se typicky využije při současném spouštění více instancí programu.

-n= nebo **--name=**

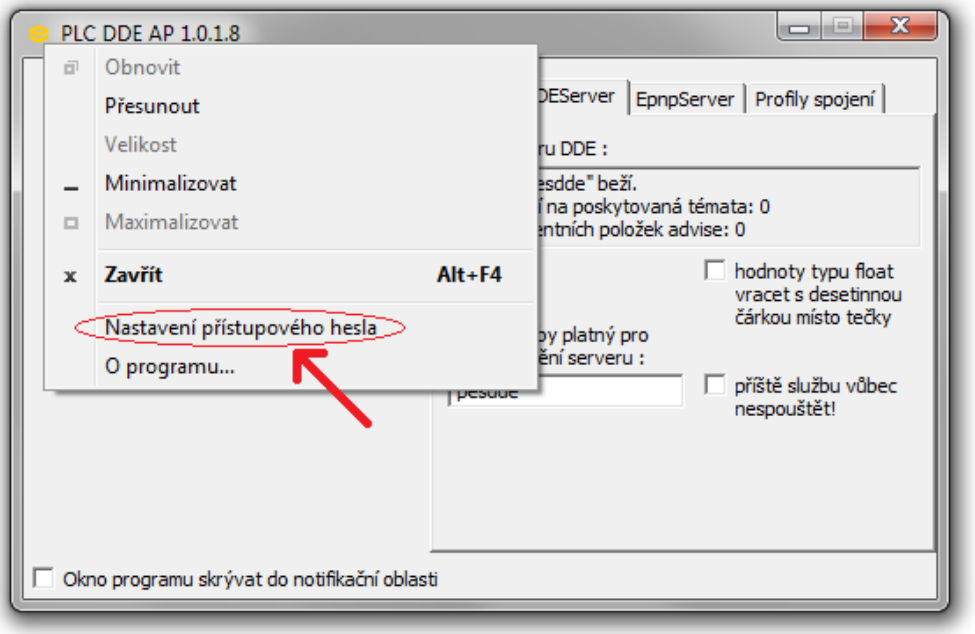
- Nastavuje zadaný název pro spouštěnou instanci programu. Název je třeba specifikovat za rovnítkem, nejlépe v uvozovkách.
- Parametr je zaveden od verze programu 1.0.1.4.

-nscf

- Provedené změny v nastavení programu, např. změna spojení na automat, se nebudou ukládat.
- Parametr je zaveden od verze programu 1.0.1.4.

Zabezpečení přístupu heslem

Program DdeAP lze nastavit tak, aby požadoval zadání bezpečnostního hesla. Heslem pak bude zakódován používaný soubor nastavení programu a zadání hesla bude vyžadováno před každým zobrazením okna programu či pokusem o ukončení programu.
Změna hesla je dostupná ze systémového menu programu. Heslované nastavení programu bude automaticky načteno se spuštěním programu. Automatické načtení ale může selhat, a to v případě, kdy došlo ke změně absolutní cesty k souboru nastavení programu.



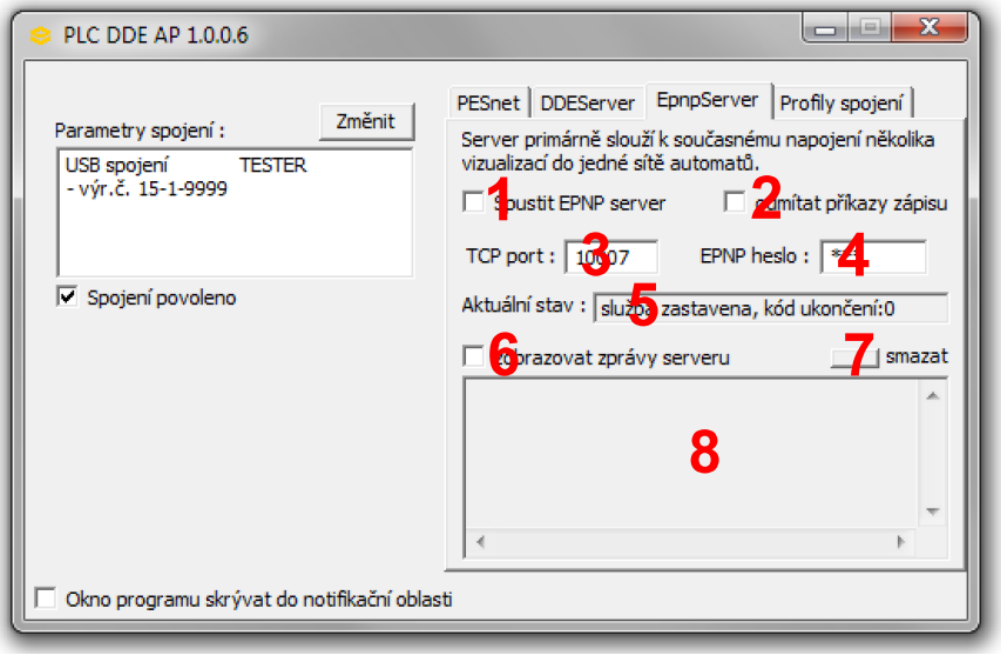
Obr. 4 - Okno programu DdeAP se zobrazeným systémovým menu

Server EPNP

Tato služba poskytuje data typicky pro webové vizualizace, které jsou produktem aplikace MICROPEL StudioWEB, tj. které využívají webovou vizualizaci obsluhovanou jazykem JavaScript, nebo Java strojkem CA4vis.jar.
Principem je, že DdeAP otevře TCP/IP port na PC, na němž od klientů (vizualizací) přijímá HTTP nebo EPNP požadavky. Požadavky dále předává (ve formátu EPNP) do připojené sítě automatů, a následně klientům vrací získané odpovědi. Při přímém propojení vizualizace na automat/převodník by se již do sítě automatů nepřipojila žádná další vizualizace, tento nedostatek řeší právě DdeAP EPNP server.
Spuštění služby je volitelné a otevření TCP/IP portu programem DdeAP je potřeba povolit v bráně Windows Firewall.

Nastavení serveru EPNP

Nastavení služby je v programu DdeAP dostupné na kartě „EpnpServer“:



Obr. 5 - Okno programu DdeAP se zobrazenou kartou nastavení pro EPNP server

Přepínač (1) provádí spuštění nebo zastavení služby. Přepínač (2) zakazuje vykonání veškerých příkazů zápisu do paměti protokolu EPNP. Editační pole (3) a (4) nastavují poslouchací TCP/IP port serveru a číselné heslo zabezpečení EPNP komunikace, aktuální obsah polí server přebírá a kontroluje v okamžiku svého spuštění. V případě nezabezpečené komunikace lze pole (4) ponechat pole prázdné, nebo doplnit nulu. Pole (5) zobrazuje stav služby - zda je aktuálně spuštěna nebo zastavena. Přepínačem (6) lze povolit vypisování informačních zpráv od spuštěné služby do pole (8), klinutí na tlačítko (7) způsobí smazání obsahu pole 8.

Výhody oproti přímému spojení vizualizace na automat/převodník:

- Lze připojit více vizualizací najednou ve stejném okamžiku.
- Automat/převodník nemusí podporovat síťový protokol TCP/IP, může být k PC připojen po USB.

Omezení komunikačního kanálu EPNP

EPNP je textový protokol vyvinutý firmou MICROPEL, který je určen pro komunikaci s automaty MICROPEL. EPNP server v programu DdeAP podporuje pouze novější, zjednodušenou verzi protokolu implementovanou v automatech řady 400 (např. i v převodníku CA5) - původní program MICROPEL DataServer podporoval naopak pouze starší verzi EPNP protokolu kompatibilní s převodníkem CA4. Množina příkazů, které server v současné době podporuje, je následující:

- **ReadRAM**
- **WriteRAM**
- **GetPlcList**
- **GetServerInfo**

Při zatržené volbě přepínače (2) v nastavení služby nebudou žádné příkazy zápisu předávány do sítě automatů - v současném stavu omezení postihne pouze příkaz **WriteRAM**.
Pozn. Sada aktuálně podporovaných příkazů též neumožňuje přístup k automatům dostupným prostřednictvím sítě EXbus z připojeného automatu/převodníku.

Formát EPNP rámců

EPNP je textový protokol. Popisovaný formát platí pro novější, zjednodušenou verzi EPNP, jíž rozumí automaty a převodníky řady 400. Položky typu longword či word se v rámcích ukládají ve formátu big-endian, tj. postupně od nejvyššího bajtu hodnoty.

Obecná struktura rámce

Nečíslovaný EPNP požadavek:

@	ADR	*	CMD	N x Data	#	SUM	<CR>

a odpověď na něj:

@	ADR	*	CMD	M x Data	#	SUM	<CR>

nebo v případě chyby zpracování požadavku odpověď:

@	ADR	!	CMD	ERR	#	SUM	<CR>

Číslovaný EPNP požadavek:

@	ADR	+	CMD	SID	N x Data	#	SUM	<CR>

a odpověď na něj:

@	ADR	-	CMD	SID	M x Data	#	SUM	<CR>

nebo v případě chyby zpracování požadavku odpověď:

@	ADR	?	CMD	SID	ERR	#	SUM	<CR>

V následujících vysvětlivkách platí, že bajtovou hodnotu v EPNP rámci reprezentují 2 hexadecimální znaky! Např. dekadická hodnota 8 je reprezentována znaky „08“, hodnota 253 pak znaky „FD“.
ADR … 1 bajt, adresa cílového automatu v připojené síti PESnet, platný rozsah je 00 až 1F (0 až 31 dekadicky), hodnota 1F (31 dekadicky) určuje, že cílem je automat, se kterým je provedeno spojení
CMD … 1 bajt, kód příkazu (typ EPNP požadavku) zadaný vysílací stranou
ERR … 1 bajt, kód chyby zpracování požadavku
N x Data … N bajtů, datový obsah požadavku, závislý na typu požadavku
M x Data … M bajtů, datový obsah odpovědi, závislý na typu požadavku
SID … 1 bajt, libovolný číselný identifikátor (00 až FF) přidělený EPNP požadavku vysílací stranou
SUM … 1 bajt, suma číselných hodnot všech znaků rámce umístěných před znakem #
<CR> … ukončující znak s hodnotu 0D (13 dekadicky)

Pozn. První tři znaky (@ a ADR) specifikují cílovou PESnet adresu, ale mohou být v požadavku vynechány. Pak platí za cíl požadavku PESnet adresa, která byla uvedena v předešlém EPNP požadavku.

Požadavek ReadRAM

Příkaz pro čtení dat z paměťového prostoru automatu specifikovaného pomocí dlouhé adresy.
CMD = „2E“
Data[0 až 3] = dlouhá adresa paměti, např. adresa „00001802“ ukazuje na StackB[2], tedy třetí bajt zásobníku
Data[4] = hodnota DCTRL, udává datový typ a počet požadovaných položek, pro získání hodnoty bitu je třeba přečíst celou bajtovou hodnotu obsahující požadovaný bit

Odpověď na platný požadavek pak bude obsahovat datovou část:
Data[0 až 4] = kopie odeslaných Data[0 až 4]
Data[5 až (4+K*SZ)] = příslušný počet vyžádaných dat (K je počet položek, SZ velikost jedné položky)

Příklad požadavku a odpovědi čtení jednoho longwordu z adresy 0x604, cílovým registrem je tak síťový longword NetLW[1], přidělené číslo požadavku SID=0x5A:

```
@02+2E5A00000604C1#B8
@02-2E5A00000604C1000003E8#5A
```

Oba rámce jsou ukončeny nezobrazitelným znakem s hodnotou 13. Vyčtena byla dekadická hodnota 1000 (000003E8 hexadecimálně).

Význam bajtové hodnoty DCTRL v požadavku ReadRAM nebo WriteRAM:

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Zápis bitu	0	0	x	x	hodnota	index bitu		
Čtení/zápis bajtů	0	1	počet bajtů (hodnota 0 znamená počet=64)					
Čtení/zápis word	1	0	počet wordů (hodnota 0 znamená počet=64)					
Čtení/zápis longword	1	1	počet longwordů (hodnota 0 znamená počet=64)					

Požadavek WriteRAM

Příkaz pro zápis dat do paměťového prostoru automatu specifikovaného pomocí dlouhé adresy.
CMD = „2F“
Data[0 až 3] = dlouhá adresa paměti, např. adresa „00001802“ ukazuje na StackB[2], tedy třetí bajt zásobníku
Data[4] = hodnota DCTRL, udává datový typ a počet zapisovaných položek
Data[5 až (4+K*SZ)] = příslušný počet zapisovaných dat (K je počet položek, SZ velikost jedné položky), jen požadavek zápisu bitu tuto část neobsahuje, zapisovaná hodnota je totiž přítomna v DCTRL

Odpověď na platný požadavek pak bude obsahovat datovou část:
Data[0 až 4] = kopie odeslaných Data[0 až 4]

Příklad požadavku a odpovědi nastavení bit[1] bajtu na adrese 0x208, cílovým bitem je tak M[1]:

```
@02*2F0000020809#37
@02*2F0000020809#37
```

Oba rámce jsou ukončeny nezobrazitelným znakem s hodnotou 13. Požadavek i odpověď jsou v tomto případě stejné.

Ostatní příkazy

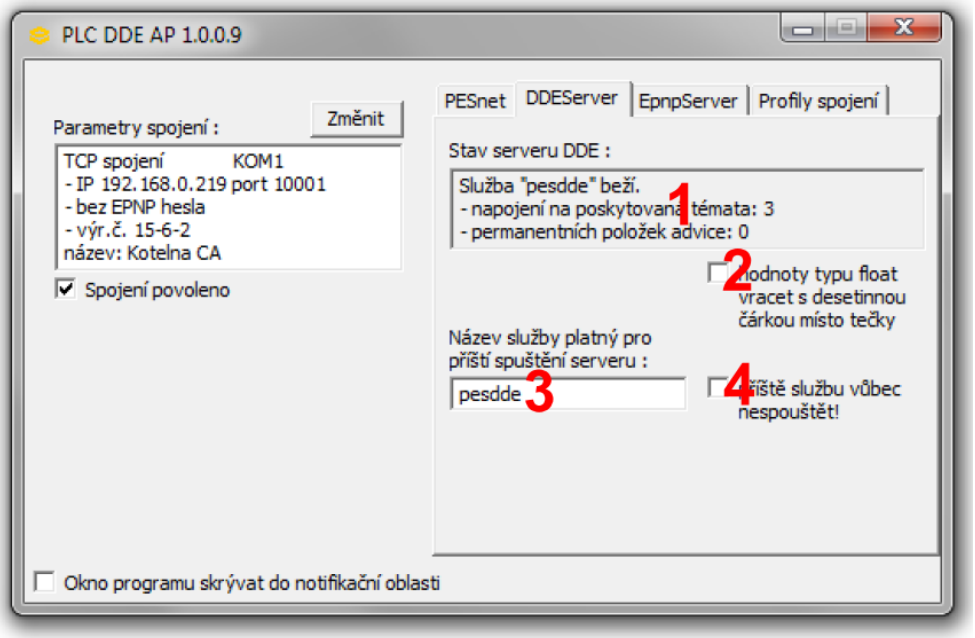
Popis ostatních podporovaných a výše nepopsaných příkazů lze nalézt v samostatném dokumentu věnovaném popisu zjednodušeného protokolu EPNP.

Server DDE

DDE server v programu DdeAP slouží především k poskytování dat z automatů MICROPEL programům podporujícím komunikaci DDE v režimu klient. DDE klienty mohou být programy StudioWin či StudioG firmy MICROPEL, DdeAP těmto programům umožňuje programovat automaty i vyčítat hodnoty sledovačů z nich. DDE klienty ale mohou být i např. vizualizační programy od jiných firem, komunikaci DDE protokolem podporuje i tabulkový editor Microsoft Excel.

Nastavení serveru DDE

Nastavení služby je v programu DdeAP dostupné na kartě „DDEServer“:



Obr. 6 - Okno programu DdeAP se zobrazenou kartou nastavení pro DDE server

Pole (1) na kartě „DDEServer“ zobrazuje stav DDE serveru. Editační pole (3) definuje název poskytované služby DDE - klienti vyhledávají DDE server právě podle tohoto názvu. Při zatržení přepínače (4) bude spuštění DDE serveru při příštím spuštění programu DdeAP vynecháno. Přepínač (2) určuje znak desetinné čárky, který server doplní do hodnoty typu float předávané klientovi (hodnota se přenáší v textové podobě). Když je přepínač zatržený, bude server tisknout desetinnou čárku, v opačném případě desetinnou tečku. Od klienta server akceptuje hodnoty float v obou možných variantách.
Pozn. Program MICROPEL StudioWin např. k propojení vyžaduje název služby **pesdde**. A v protokolu DDE není velikost písmen rozlišována, takže název **PESDDE** je též správný.

Komunikace se serverem DDE

Komunikace se severem DDE v DdeAP je vždy textová (formátu CF_TEXT). Při sestavování žádosti o DDE propojení ze strany klienta je třeba specifikovat tři položky spojení DDE:

název služby (service)	Identifikuje DDE službu spuštěné aplikace, standardní jméno je **pesdde**.
název tématu (topic)	Určuje jednotlivé podslužby služby DDE. DataServer poskytuje dvě podslužby: **var** a **mem** (viz dále).

název položky (item)	Definuje vlastní požadavek na akci. Typicky popis paměťového místa, registru automatu, ze kterého se má číst nebo do kterého se má zapisovat.

Témata „var“ a „mem“

K dispozici jsou dvě základní služby (témata), **var** a **mem**, k nim přísluší textové komunikační protokoly VAR a MEM. V současném stavu lze, z později zmíněných důvodů, v tématu **var** použít i protokol MEM.
Téma **var** tedy podporuje původní protokol VAR, jehož koncepce vycházela z historických požadavků (lze přistupovat pouze k pevně deklarovaným registrům automatu, jako jsou vstupy, výstupy, zásobník, síťové proměnné).
Téma **mem** umožňuje přístup i k uživatelsky definovaným proměnným protokolem MEM (těm, které se deklarují jako globální v programu jazyka SIMPLE4) a několika dalším speciálním paměťovým prostorům. Navíc tato služba předpokládá využití požadavků REQUEST resp. POKE pro čtení resp. zápis. Protokol MEM tak v popisu položky (paměťového místa) nepotřebuje udávat směr akce, jako tomu je v případě položek protokolu VAR. Téma **mem** navíc podporuje trvalé propojení na data – DDE službou ADVISE. Při tomto způsobu propojení nejdříve klient u serveru zaregistruje položku čtení paměťového místa, server poté sám zasílá data klientovi, opakovaně jednou za čas, nebo ihned poté, co zjistí změnu hodnoty.

Společné vlastnosti protokolů VAR a MEM

Formální vyjádření hodnot

Číselné vyjádření indexů, bitů, adres automatů, adres v paměti

Formální vyjádření hodnot indexů, čísla bitů, adres automatů, adres v paměti je následující. Veškerá tato čísla mohou být předávána jak v dekadické formě, tak ve formě hexadecimální (hexadecimální číslo je uvozeno buď 0x nebo jen x). A to bez znaménka, záporná hodnota je chápána jako chyba.
Příklad významově stejných položek, kanálů:

```
user;longword[0xC]?0x0A;0x03;0x40
user;longword[xC]?xA;x3;x40
user;longword[12]?10;3;64
user;longword[12]?10;3;0x40
```

Číselné vyjádření dat pro zápis

Data určená pro zápis mohou být předávána buď v dekadické kladné hodnotě, nebo v dekadické záporné hodnotě u znaménkových typů, nebo v hexadecimální formě (hexadecimální číslo je uvozeno buď 0x nebo jen x). U typu float může být hodnota předána ve formátu s desetinným separátorem (tečka nebo čárka) a exponentem (e nebo E).

Data vrácená kanálem při čtení

Vrácená data (při čtení) jsou ve formátu dekadickém (se znaménkem u znaménkových typů) nebo u typu float ve formátu s desetinným separátorem (tečka nebo čárka, podle nastavení v konfiguračním souboru) a exponentem (E). Hexadecimální tvar čtených dat nelze nastavit.

Adresa automatu

V každém požadavku přístupu do paměti automatu je třeba uvést síťovou adresu cílového automatu. Pro cílový automat v síti PESnet je třeba jako adresu použít hodnotu PESnet adresy automatu (v rozsahu 0 až 30). Pokud je cílem požadavku automat/převodník zprostředkovávající spojení, stačí použít adresu 31. Pokud se cílový automat nachází v síti EXbus jako Slave, je možno použít adresu o hodnotě (EXbusAdr * 32) + 31.

Maximální počet hodnot požadavku

Většina dostupných paměťových prostorů popsaných dále v dokumentu umožňuje čtení nebo zápis více položek pole v jednom požadavku, program DdeAP podporuje přenos max. **512** hodnot.

Formát předávaných hodnot položek

Zapisované či vyčtené hodnoty jsou vždy posledním (mohou být též jediným) parametrem v textovém řetězci požadavku resp. odpovědi. Pokud se přenáší jedna hodnota, bude jako poslední parametrem v textu uvedena jednoduše tato hodnota `číslo`. Při přenosu několika hodnot pole bude mít poslední parametr v textu podobu `##číslo#číslo# ...#číslo##`. Namísto textu „číslo“ bude vytisknuta příslušná hodnota.

Síťové proměnné PESnet

Síťové proměnné jsou takové, do nichž zapsaná hodnota v jednom automatu je automaticky odeslána ostatním automatům v síti PESnet. U automatů řady 400 se lze spolehnout, že pokud do jeho síťové proměnné zapíšeme hodnotu i prostřednictvím DDE, rozeslání hodnoty zafunguje. U vývojově starších automatů se ale o rozeslání musí postarat převodník. Jistější je tedy kanálem DDE zapisovat hodnoty síťových proměnných do převodníku/automatu zajišťujícího spojení se serverem. A to buď prostřednictvím varianty zápisu v protokolu MEM nevyžadující zadání adresy automatu, nebo je jinak propojující automat/převodník dostupný pod PESnet adresou 31.

Komunikace v tématu „var“

Pro ustanovení DDE propojení tématem je třeba v datovém propojení stanovit následující hodnoty:

název služby (service)	**pesdde**
název tématu (topic)	**var**
název položky (item)	popis paměťového místa protokolem VAR

Název položky je již vlastní popis paměťového místa. Při komunikaci v tématu **var** je třeba užít speciálního textového protokolu. Tento protokol bude dále nazýván protokolem VAR.

Téma **var** v serveru DDE používá následující typy požadavků DDE:

REQUEST	Jednorázové čtení položky. Hodnota je přečtena z paměti automatu a volání služby se vrátí až po ukončení čtení a zaslání hodnoty.
POKE	Jednorázový zápis položky. Hodnota je zapsána do paměti automatu a volání se vrátí až po vykonání zápisu.

Protože některé klientské aplikace používají pouze službu REQUEST (požadavek na čtení), je protokol VAR uzpůsoben tak, aby přesto bylo možné provést touto službou i zápis, proto je v položce definující registr automatu také určen směr operace (viz popis protokolu VAR).

V tématu **var** lze pomocí speciálních značek použít i textový protokol MEM (popsán dále v dokumentu). To má smysl v případech, kdy klient není schopen řízeně použít službu POKE pro zápis. To se týká např. klienta DDE programu Microsoft Excel.

Popis protokolu VAR

Protokol VAR zpřístupňuje tyto registry automatu:

vstupy, výstupy	pole I, O, X, Y

síťové proměnné	bity M, wordy D, longwordy NetLW (též jako floaty NetF)
vnitřní registry	bity M, wordy D
speciální funkční registry	W, B
zásobník	STACK

Syntaxe čtení paměťového místa

Při žádosti o čtení je třeba název položky specifikovat ve formátu:

`rX[Automat][Adresa_počátku_bloku][Počet_položek]`

přičemž místo *X* uvedeme znak:

- **b** - čtení bitů (proměnné X, Y, M, B)
- **w** - čtení wordů (proměnné I, O, D, W)
- **s** - čtení wordů ze zásobníku (STACK)
- **L** - čtení síťových longwordů (L neboli NetLW)
- **F** - čtení síťových floatů (NetF) ve formátu: 3.402821+/-e38 (zaokrouhleno vždy na 6 míst za desetinnou tečkou)

Žádost slouží k vyčtení jedné položky nebo souvislého bloku položek. Chybí-li část `[Počet_položek]`, čte se jedna položka.

Automat … Adresa automatu v síti.

Adresa_počátku_bloku … Adresa první položky (**b**: rozsah 0 až 319; **w**: rozsah 0 až 255; **s**: rozsah 0 až 11775; **L**: rozsah 0 až 255; **F**: rozsah 0 až 255). Pro přístup do jednotlivých bitových polí X, Y, M, B v prostoru **b** a wordových polí I, O, D, W v prostoru **w** je potřeba zadávat absolutní adresy (viz níže).

Počet_položek … Počet položek (rozsah 1 až 255).

Návratová hodnota

DDE server po úspěšném provedení vrátí textový řetězec ve formátu `##číslo#číslo# ...#číslo##`. Pokud jde o jedinou položku, řetězec bude obsahovat samostatnou hodnotu (bez znaků #).

U čtení z datového kanálu typu float je možné jako desetinný oddělovač zvolit tečku nebo čárku (tlačítkovým přepínačem v programu DdeAP na kartě DDE serveru).

Absolutní adresy proměnných v SIMLPE2

Typ	Proměnná v SIMPLE2	Absolutní adresa	Popis
BIT	X0 … X31	0 … 31	digitální vstupy
	Y0 … Y31	32 … 63	digitální výstupy
	M0 … M127	64 … 191	uživatelské bity
	B0 … B127	192 … 319	speciální funkční bity
WORD	I0 … I31	0 … 31	analogové vstupy
	O0 … O31	32 … 63	analogové výstupy
	D0 … D63	64 … 127	uživatelské registry
	W0 … W127	128 … 255	speciální funkční registry

Syntaxe zápisu paměťového místa

Při žádosti o zápis je třeba název položky specifikovat ve formátu:

`sX[Automat][Adresa_počátku_bloku]číslo`

nebo v případě blokového zápisu:

`sX[Automat][Adresa_počátku_bloku]##číslo#číslo#...#číslo##`

přičemž místo *X* uvedeme znak:

- **b** - zápis bitů (proměnné X, Y, M, B)
- **w** - zápis wordů (proměnné I, O, D, W)
- **s** - zápis wordů na zásobník (STACK)
- **L** - zápis síťových longwordů (L neboli NetLW)
- **F** - zápis síťových floatů (NetF) ve formátu: 3.4027+/-e38

Žádost slouží k zápisu jedné položky nebo souvislého bloku položek. Zapisuje-li se jediná položka, řetězec *číslo* se uvede bez pomocných znaků #.
Automat … Adresa automatu v síti.
Adresa_počátku_bloku … Adresa první položky (**b**: rozsah 0 až 319; **w**: rozsah 0 až 255; **s**: rozsah 0 až 11775; **L**: rozsah 0 až 255; **F**: rozsah 0 až 255). Pro přístup do jednotlivých bitových polí X, Y, M, B v prostoru **b** a wordových polí I, O, D, W v prostoru **w** je potřeba zadávat absolutní adresy (viz výše).

Návratová hodnota

DDE server po úspěšném provedení vrátí textový řetězec `OK`.

Pozn. U zápisu do datového kanálu typu float je možné použít jako oddělovač desetinných míst tečku nebo čárku. A to bez ohledu na místní nastavení PC nebo na nastavení desetinného oddělovače pro návratové hodnoty typu float (v odpovědích na požadavky čtení).

Použití protokolu VAR v programu Microsoft Excel

Do libovolné buňky sešitu Microsoft Excel lze napsat textový řetězec, který bude program Excel chápat jako žádost o DDE relaci, provede ji, a umístí vrácená data zpět do této buňky. Aby zápis do buňky byl chápán jako požadavek DDE, musí mít tento zápis formát:

```
=pesdde|var!'Název_položky'
```

kde *Název_položky* je zápisu požadavku protokolem VAR.

Příklad

Čtení

Ukázka zadání příkazu do buňky v Microsoft Excel, který vyčte hodnoty 4 bitů z automatu s adresou 30 počínaje bitem na adrese 66 (tedy bity M2 až M5):

```
=pesdde|var!'rb[30][66][4]'
```

Návratová hodnota (například):

```
##0#0#1#0##
```

Zápis

Zápis hodnoty 123 na zásobník do StackW[4] automatu s adresou 5:

```
=pesdde|var!'ss[5][4]123'
```

Návratová hodnota:

```
OK
```

Pozn. Pokud je takto provedený zápis dat ukončen úspěšně, bude v té samé buňce zobrazena hodnota `OK`. V případě chyby vyřízení zápisu bude v buňce zobrazeno hlášení přímo od ovladače Excelu.
Pozn. Program Microsoft Excel po zadání DDE příkazu zkusí zaslat daný řetězec nejprve jako žádost o trvalé propojení (DDE požadavek typu ADVISE) a pak jako žádost o jednorázové čtení (požadavek typu REQUEST). To je důvod, proč se při komunikaci s Excelem zobrazí v poli hlášení o průběhu informace o chybné syntaxi, protože syntaxe jednorázových a trvalých příkazů se liší (viz výše). Komunikace DDE se přesto nakonec ustanoví správně, protože Excel zkouší různé druhy požadavků, až se „trefí“.

Komunikace v tématu „mem“

Pro ustanovení DDE propojení tématem je třeba v datovém propojení stanovit následující hodnoty:

název služby (service)	**pesdde**
název tématu (topic)	**mem**
název položky (item)	popis paměťového místa protokolem MEM

Název položky je již vlastní popis paměťového místa. Při komunikaci v tématu **mem** je třeba užít speciálního textového protokolu. Tento protokol bude dále nazýván protokolem MEM.

Téma **mem** v serveru DDE používá následující typy požadavků DDE:

REQUEST	Jednorázové čtení položky. Hodnota je přečtena z paměti automatu a volání služby se vrátí až po ukončení čtení a zaslání hodnoty.
POKE	Jednorázový zápis položky. Hodnota je zapsána do paměti automatu a volání se vrátí až po vykonání zápisu.
ADVISE	Trvalé propojení, tj. opakované vyčítání datové položky. Volání této služby se na rozdíl od služby REQUEST vrací ihned. Následně server DDE sám zjišťuje požadovanou hodnotu a zasílá novou hodnotu po její změně, pro jistotu zasílá jednou za čas i nezměněnou hodnotu.

Popis protokolu MEM

Protokol MEM zpřístupňuje tyto registry automatu:

v zásadě celou paměť (RAM) automatu	vstupy, výstupy, uživatelskou RAM, zásobník, uzly EXbus

Protokolem MEM lze v některých předdefinovaných paměťových prostorech přistupovat i k položkám různých datových typů (bit, word, int, longword, longint, float), podle potřeby.

Obecná podoba datového požadavku (název položky DDE)

Požadavek je textový řetězec složený ze sady až 5 parametrů oddělených středníkem:

```
parametr1;parametr2;parametr3;parametr4;parametr5
```

Jednotlivé parametry mohou být pro přehlednost odděleny mezerami. Celkový počet parametrů závisí na cílovém paměťovém prostoru definovaném prvním parametrem i směru (čtení nebo zápis). U požadavku zápisu budou vždy posledním parametrem zapisovaná data. U požadavku čtení může být jako poslední parametr uveden specifikátor počtu požadovaných položek, pokud není přítomen, bude se číst jen jedna položka.

Dostupné paměťové oblasti

Prvním parametrem v položce požadavku protokolem MEM je klíčové slovo definující cílovou paměťovou oblast a též určující celkový počet a význam dalších parametrů v požadavku protokolu MEM. Klíčová slova sémanticky odpovídají vybrané cílové oblasti (viz souhrnnou tabulku).

Tabulka dostupných paměťových oblastí

Následující tabulka je shrnutím seznamu dostupných paměťových oblastí protokolem MEM a seznam a významem jednotlivých parametrů. Jednotlivé oblasti mohou ještě nést určité upřesňující výjimky, které nejsou v této tabulce uvedeny a jsou popsány dále v dokumentu, v popisu jednotlivých oblastí.

	parametr			
	1.	**2.**	**3.**	**4.**
popis paměťové oblasti	***typ paměti***	***typ datové proměnné***	***adresa automatu***	***adresa proměnné v příslušné oblasti***
uživatelská paměť	**user**	bit, byte, word, int, longword, longint, float	0-31 (příp. EXbus)	0 až N-1 (rozsah dle typu automatu)
zásobník	**stack**	bit, byte, word, int, longword, longint, float	0-31 (příp. EXbus)	--
obecná paměť	**abs**	bit, byte, word, int, longword, longint, float	0-31 (příp. EXbus)	0 až N-1 (rozsah dle typu automatu)
prostor vnitřních a vnějších (EXbus) IO uzlů	**sys_ebio**	bit, byte, word, int, longword, longint, float	0-31 (příp. EXbus)	0 až N-1 (rozsah dle typu automatu)
seznam aktivních IO uzlů	**sys_eblist**	bit, byte, word, longword	0-31 (příp. EXbus)	--
digitální vstupy	**sys_X**	bit	0-31	--
digitální výstupy	**sys_Y**	bit	0-31	--
uživatelské bity M	**sys_M**	bit	0-31 (příp. EXbus)	--
uživatelské síťové bity M	**sys_netM**	bit	--	--
speciální funkční bity	**sys_B**	bit	0-31 (příp. EXbus)	--
analogové vstupy	**sys_I**	word,int	0-31	--
analogové výstypy	**sys_O**	word,int	0-31	--
uživatelské registry D	**sys_D**	word,int	0-31 (příp. EXbus)	--
uživatelské síťové registry D	**sys_netD**	word,int	--	--
speciální funkční registry	**sys_W**	word,int	0-31 (příp. EXbus)	--
uživatelské proměnné L, taktéž NetLW, NetLI, NetF	**sys_L**	longword, longint, float	0-31 (příp. EXbus)	--
uživatelské síťové proměnné L, taktéž NetLW, NetLI, NetF	**sys_netL**	longword, longint, float	--	--
registry periferie EX	**perex**	word, int	0-31 (příp. EXbus)	--

Typ datové proměnné

Druhý parametr požadavku určuje typ datové proměnné. Určuje datovou velikost proměnné (registru), jaká bude použita při čtení nebo zápisu. Typ je rozlišen klíčovými slovy: **bit**, **byte**, **word**, **int**, **longword**, **longint**, **float**. Význam jednotlivých klíčových slov je patrný z tabulky následující níže. Současně také typ datové proměnné určuje formát přenášených dat komunikačním kanálem. Formátem se myslí znaménková nebo neznaménková reprezentace, nebo, v případě typu float, tisk / očekávání dat v exponenciálním tvaru. Zároveň je typem datové proměnné vymezena minimální a maximální velikost přenášené hodnoty.
Jednotlivé paměťové oblasti, uvedené dále v textu, mají různé skupiny povolených datových typů. Například do oblasti síťových wordů D lze přistupovat pouze typem word či int (znaménkový word).
Parametr typu datové proměnné lze rozšířit o operátor přístupu do pole nebo o operátor bitového přístupu.

Tabulka datových typů a jejich vlastností

typ	***délka paměti zabraná proměnnou (v bajtech)***	***znaménková konvence***	***přístup po bitech typ?x***	***přístup do pole typu typ[index]***	***min hodnota***	***max hodnota***
bit	1	ne	ne	ano	0	1
byte	1	ne	ano	ano	0	255

word	2	ne	ano	ano	0	65535
int	2	ano	ne	ano	-32758	32767
longword	4	ne	ano	ano	0	4294967295
longint	4	ano	ne	ano	-2147483648	2147483647
float	4	ano	ne	ano	1.175494351e–38	3.402823466e+38

Pozn. Hodnota v kanálu typu float bude zaokrouhlena vždy na 6 míst za desetinnou tečkou.

Přístup do pole stejného typu

Součástí definice typu datové proměnné může být syntaktické rozšíření o operátor přístupu do pole. Myslí se tím do pole prvků stejného typu jako je zvolený typ datové proměnné. Syntaxe použitého operátoru je stejná jako v jazyce Simple4. Tedy, index do pole se udává v hranatých závorkách. Formální předpis je pak tento:

```
typ_datové_proměnné[index]
```

Příklad

Máme-li v jazyce Simple4 deklarovanou proměnnou typu pole `var word mojePoleWord[10]`, která začíná na adrese 0x20 v automatu 3, a chceme-li založit položku směrující např. na pátý word z tohoto pole, pak deklarace položky může být následující:

`user; word[5]; 3; 0x20` → kanál směřuje na 5tou položku pole wordů, které začíná na adrese 0x20

nebo také tato, pokud nevyužijeme možnost rozšíření o operátor přístupu do pole:

`user; word; 3; 0x2A` → kanál směřuje na 5tou položku pole wordů, které začíná na adrese 0x20

Pozn. Jedná se o 5-tý word (word = 2 bajty), tedy, nevyužijeme-li přístupu do pole, musíme přepočítat adresu wordu, tj. 5x2=10=0x0A (0x0A je hexadecimální zápis hodnoty 10).

Pozn. První hodnota indexu je 0, takže zápis typu datové proměnné word a word[0] je rovnocenný z hlediska výsledné adresy.

Pozn. Operátoru přístupu do pole stejného typu využívá celá řada paměťových oblastí jako jediného způsobu adresace. Například oblast zásobníku nevyužívá adresu v paměti, ale vystačí si pouze s tímto operátorem, přístup do zásobníku je tak shodný s tím, jak jej známe buď z jazyka Simple4, nebo ze sledovačů ve StudioWin.

Pozn. Ne u každého typu je možné použít tento operátor, viz tabulku výše.

Přístup do proměnné po bitech

Součástí definice typu datové proměnné může být také operátor přístupu na konkrétní jeden bit zvolené datové proměnné. Syntaxe použitého operátoru bitového přístupu je stejná, jako v jazyce Sipmle4, tedy, index bitu ve zvoleném datovém typu se píše za otazník.

Příklad

Máme-li v jazyce Simple4 deklarovanou proměnnou typu longword `var longword mujLong`, která začíná na adrese 0x40 v automatu 3, a chceme-li založit položky směrující například na pátý a desátý bit této proměnné, pak jsou deklarace položek následující:

`user; longword?4 ;3 ; 0x40` → kanál směřuje na 5. bit proměnné longword ležící na adrese 0x40

`user; longword?9 ;3 ; 0x40` → kanál směřuje na 10. bit proměnné longword ležící na adrese 0x40

Pozor! Indexace bitů začíná od 0.

Pozn. Celkem logicky se při použití operátoru bitového přístupu změní typ předávané hodnot při vyhodnocení požadavku na typ bit.

Pozn. Ne u každého typu je možné použít tento operátor, viz tabulku výše.

Kombinace operátoru přístupu do pole a k bitu

Celkem snadno může vzniknout požadavek na přístup k jednotlivým bitům proměnných, které jsou seřazeny v poli. Syntaxe je stejná jako v jazyce Simple4, tedy nejdříve se použije operátor přístupu do pole a poté operátor přístupu k bitu.

Příklad

Máme-li v jazyce Simple4 deklarované pole wordů `var word mojePoleWord[10]`, které začíná na adrese 0x20 v automatu 3, a chceme-li založit položku směrující například na devátý bit pátého wordu z tohoto pole, pak deklarace položky může být následující:

`user; word[4]?8; 3; 0x20` → kanál směřuje na 5. položku pole wordů, které začíná na adrese 0x20

Pozor! Indexace položek pole i bitů začíná od 0.

Příklady syntaxe protokolu MEM

Příklad zápisu položek protokolu MEM spočívá v ukázce deklarace proměnných v jazyce Simple4, v jejich fixaci pomocí klíčového slova fix v aplikaci MICROPEL StudioWin (v aplikaci lze využít též nástroje automatického vytvoření seznamu fixací).

Deklarace a fixace v Simple4

Fixace deklarované proměnné v paměti automatu má následující syntaxi:

```
fix nazevPromenne = ( adresa_v_pameti_PLC, delka_umisteni_v_pameti_PLC)
```

vysvětlivky

fix	klíčové slovo jazyka Simple4
názevPromenne	název proměnné uvedený v deklaraci proměnné
adresa_v_pameti_PLC	adresa v paměti PLC - nejlépe adresa, kam proměnnou umístil překladač při prvním překladu, tedy v době, kdy proměnná ještě nebyla fixovaná
delka_umisteni_v_pameti_PLC	délka v bajtech, jakou zabírá fixovaná proměnná v paměti

Tabulka příkladů deklarací a fixací

deklarace proměnných v Simple4	fixace proměnných v Simple4

var word mujWord;
var word mujWord2;
var byte mujByte;
var byte mujByte2;
var byte mujByte3;
var bit mujBit;
var bit mujBit2;
var longword mujLong;
var bit[7] mujBitPole7;
var bit[9] mujBitPole9;
var float mujFloat;
var byte[2048] mujSuperBajt;
var int mujInt;
var longint mujLongint;
var word[10] mujWord10;
var longword[10] mujLongWord10;
	fix mujWord = (0x0, 2);
	fix mujWord2 = (0x2, 2);
	fix mujByte = (0x4, 1);
	fix mujByte2 = (0x5, 1);
	fix mujByte3 = (0x6, 1);
	fix mujBit = (0x7, 1);
	fix mujBit2 = (0x8, 1);
	fix mujLong = (0x9, 4);
	fix mujBitPole7 = (0xd, 1);
	fix mujBitPole9 = (0xe, 2);
	fix mujFloat = (0x10, 4);
	fix mujSuperBajt = (0x14, 2048);
	fix mujInt = (0x814, 2);
	fix mujLongint = (0x816, 4);
	fix mujWord10 = (0x81a, 20);
	fix mujLongWord10 = (0x82e, 40);

Příklady protokolu MEM

Příklady protokolu MEM se vztahují k výše uvedené tabulce deklarací a fixací. Ve všech příkladech má cílový automat adresu 3.

Přístup k základním typům

deklarace	přístup k proměnné
`user; bit; 3;0x7`	mujBit
`user; byte; 3; 0x4`	mujByte
`user; byte; 3; 0x5`	mujByte2
`user; word; 3; 0x0`	mujWord
`user; longword; 3; 0x9`	mujLong
`user; longint; 3; 0x816`	mujLongint
`user; float; 3; 0x10`	mujFloat

Přístup do pole základních typů

deklarace	přístup do pole proměnné
`user; bit[4]; 3;0x0d`	mujButPole7[4] - přistupuje k 4 bitu z pole mujButPole7
`user; word[9]; 3;0x81a`	mujWord10[9]
`user; longword[9]; 3;0x82e`	mujLongWord10[9]

Bitový přístup

deklarace	přístup do pole proměnné
`user; longword?30; 3; 0x9`	přistupuje k 30 bitu z proměnné mujLong
`user; word[9]?4; 3;0x81a`	přistupuje k 4 bitu z proměnné mujWord10[9]
`user; byte?0; 3;0x04`	přistupuje k 0 bitu z proměnné mujByte

Jednotlivé dostupné paměťové oblasti podrobněji

Uživatelská programová paměť (user)

Uživatelskou oblastí paměti se myslí ta oblast, kam jsou umisťovány uživatelské deklarace proměnných v jazyce Simple4. Tedy deklarujeme-li v jazyce Simple4 proměnnou „mujWord“ například takto: `var word mujWord`, bude překladačem umístěna právě do této oblasti paměti. Tato paměťová oblast je určena v protokolu MEM klíčovým slovem **user** jakožto prvním parametrem.

	parametr			
	1.	**2.**	**3.**	**4.**
paměťová oblast	***typ paměti***	***typ datové proměnné***	***adresa automatu***	***adresa proměnné v příslušné oblasti***
uživatelská paměť	**user**	bit, byte, word, int, longword, longint, float	0-31 (příp. EXbus)	0 až N-1(rozsah v dle typu automatu)

Popis parametrů „adresa automatu“ a „typ datové proměnné“ lze nalézt ve shodně nazvaných podkapitolách uvedených dříve v textu.

Adresa proměnné v příslušné oblasti

Adresa datové proměnné je hodnota adresy v uživatelské paměti, na kterou umístil překladač jazyka Simple4 požadovanou proměnnou. Proměnné se v rámci uživatelské oblasti umisťují od adresy 0.

Přepočet adresy datové proměnné

Základní nativní datový typ v uživatelské paměti je bajt. Typ má význam při přepočtu adresy, pokud použijeme například operátor přístupu do pole.

Fixace adresy v jazyce Simple4

Proměnnou buď překladač umisťuje podle svého uvážení, nebo podle požadavku programátora a to sice příkazem fix (viz dále). Adresu proměnné z již přeloženého kódu jazyka Simple4 lze zjistit pomocí Správce proměnných v aplikaci StudioWIN (lze vyvolat z editoru jazyka Simple4).

Je doporučeno přistupovat pouze na ty proměnné, které jsou takzvaně fixovány již během překladu programu automatu. Fixace proměnné je proces překladače Simple4, kdy v textu programu je nejen vlastní deklarace proměnné (například deklarace: `var word mojePoleWord[10]`), ale také příkaz fixace. Fixace je požadavek na pevné umístění proměnné v paměti automatu (klíčové slovo Fix) (příklad fixace výše uvedené deklarace:

`fix mojePoleWord = (0x10,20)` - pole proměnných začíná na adrese 0x10 a má délku 20 bajtů.

Je také doporučeno použít pro fixaci nástroj v editoru aplikace StudioWin, který sám optimálně rozvrhne umístění proměnných v paměti automatu. Fixací proměnné jinak nastavíme požadovanou adresu proměnné v paměti automatu.

Pozn. Parametr délky, jakou proměnná zabírá v paměti automatu pro potřeby protokolu MEM v zásadě nepotřebujeme. Tato délka bude určena zvoleným typem datové proměnné.

Příklad syntaxe pro přístup do uživatelské paměti lze nalézt v části Příklady syntaxe protokolu MEM.

Zásobník (stack)

Zásobník je předdefinovaná oblast paměti automatu, která je vyhrazena pro použití za běhu uživatelského programu. Tato paměťová oblast je určena v protokolu MEM klíčovým slovem **stack** jakožto prvním parametrem.

	parametr			
	1.	**2.**	**3.**	**4.**
paměťová oblast	***typ paměti***	***typ datové proměnné***	***adresa automatu***	***adresa proměnné v příslušné oblasti***
zásobník	**stack**	bit, byte, word, int, longword, longint, float	0-31 (příp. EXbus)	--

Popis parametrů „adresa automatu“ a „typ datové proměnné“ lze nalézt ve shodně nazvaných podkapitolách uvedených dříve v textu.

Adresace položek v zásobníku

Adresace v rámci této oblasti paměti probíhá operátorem přístupu do pole, který se používá spolu s typem datové proměnné.

Základní nativní datový typ zásobníku je word. Velikost paměti typu zásobník je ve většině automatů 11776 wordů. Adresace první a poslední položky zásobníku v automatu s adresou 3 jsou tak následující:

`stack; word[0]; 3` kanál směřuje na první word ze zásobníku automatu s adresou 3

`stack; word[11775]; 3` kanál směřuje na poslední word ze zásobníku automatu s adresou 3

Pokud použijeme v přístupu jiný datový typ než základní, hodnota horní hranice maximálního indexu se samozřejmě změní. Například při bajtovém přístupu takto:

`stack; byte[0]; 3` kanál směřuje na **první bajt** ze zásobníku automatu s adresou 3

`stack; byte[23551]; 3` kanál směřuje na **poslední bajt** ze zásobníku automatu s adresou 3

Rozsahy indexů pro jednotlivé datové typy:

typ	*dolní index*	*horní index*
bit	0	23551
byte	0	23551
word	0	11775
int	0	11775
longword	0	5887
longint	0	5887
float	0	5887

Obecná paměť (abs)

Přístup do obecné paměti automatu je vyhrazena pro služby aplikací MICROPEL a není proto doporučen pro ostatní uživatele.

Datová oblast IO uzlů (sys_ebio)

Každý aktivní EXbus-Slave uzel s adresou na lince EXbus má v paměti Mastera dostupný svůj datový stav o velikosti 40 bajtů. Uzly 0 až 16 jsou lokálními uzly automatu, jde např. o uzly osazených IO modulů automatu či uzly příkazové brány ovládání Ethernet komunikace z programu automatu. Lokální uzly mohou být alokovány, i pokud automat nemá spuštěn ovladač EXbus-Master. Význam datových hodnot uzlu je závislý na typu uzlu, strukturu dat IO uzlu daného typu lze většinou dohledat v programovací příručce pro PLC řady 400. Zápisy do této oblasti se z hlediska bezpečnosti chodu systému nedoporučují.

	parametr			
	1.	**2.**	**3.**	**4.**
paměťová oblast	***typ paměti***	***typ datové proměnné***	***adresa automatu***	***adresa proměnné v příslušné oblasti***
Prostor vnitřních a vnějších IO uzlů	**sys_ebio**	bit, byte, word, int, longword, longint, float	0-31 (příp. EXbus)	0 až N-1 (rozsah dle typu automatu)

Popis parametrů „adresa automatu“ a „typ datové proměnné“ lze nalézt ve shodně nazvaných podkapitolách uvedených dříve v textu.

Adresa proměnné v příslušné oblasti

Adresa datové proměnné je bajtový index do paměti s datovými stavy IO uzlů. Každý uzel zabírá v této oblasti 40 bajtů. Na adresách 0 až (16*40-1) jsou přítomny datové struktury vnitřních uzlů automatu, na vyšších adresách potom datové struktury uzlů zařízení EXbus-Slave zapojených na lince EXbus – např. druhý IO modul zařízení s EXbus adresou 30 na lince má počáteční adresu (30+1)*40 v této paměťové oblasti.

Pro správnou interpretaci hodnot z adres uvnitř této adresní oblasti je potřeba přesně vědět, jakou konkrétní strukturu a funkci má uzel, na nějž adresa ukazuje!

Příklad

V programu PLC s PESnet adresou 4 a spuštěným ovladačem EXbus-Master přečteme hodnotu analogového vstupu na 3. modulu (modulu B) periferie MEX401 s adresou 40 kódem `vstI1 = iob[42].i[1]`.

Pro přístup ke stejné hodnotě protokolem MEM použijeme syntaxi

```
sys_ebio; word[1]; 4; 0x690
```

Seznam aktivních IO uzlů (sys_eblist)

Pole 2048 bitů v paměti automatu určené pouze ke čtení. Hodnota každého bitu pole udává, zda je uzel s (EXbus) adresou odpovídající indexu bitu alokován či nikoli.

	parametr			
	1.	**2.**	**3.**	**4.**
paměťová oblast	***typ paměti***	***typ datové proměnné***	***adresa automatu***	***adresa proměnné v příslušné oblasti***
seznam aktivních IO uzlů	**sys_eblist**	bit, byte, word, longword	0-31 (příp. EXbus)	--

Popis parametrů „adresa automatu“ a „typ datové proměnné“ lze nalézt ve shodně nazvaných podkapitolách uvedených dříve v textu.

Digitální vstupy/výstupy (sys_X, sys_Y)

Pole digitálních vstupů a výstupů automatů vývojově starších než řady 400 je v jazyce Simple označováno jako bitové pole X a Y. Přístupem do těchto oblastí má uživatel plný přístup na reálné digitální vstupy/výstupy automatu (pokud jsou tyto pozice osazeny).

	parametr			
	1.	**2.**	**3.**	**4.**
paměťová oblast	***typ paměti***	***typ datové proměnné***	***adresa automatu***	***adresa proměnné v příslušné oblasti***
digitální vstupy	**sys_X**	bit	0-31	--
digitální výstupy	**sys_Y**	bit	0-31	--

Popis parametrů „adresa automatu“ a „typ datové proměnné“ lze nalézt ve shodně nazvaných podkapitolách uvedených dříve v textu. Přístup k datové proměnné je v této paměťové oblasti omezen na typ bit.

Adresace vstupů a výstupů

Adresace v rámci této oblasti paměti probíhá operátorem přístupu do pole, který se používá spolu s typem datové proměnné.

Základní a jediný nativní datový typ těchto oblastí je bit. Velikost obou popisovaných oblastí je 32 bitů. Adresace prvního digitálního vstupu a posledního výstupu v automatu s adresou 3 jsou tak následující:

`sys_X; bit[0]; 3` kanál směřuje na první digitální vstup X0 automatu s adresou 3

`sys_Y; bit[31]; 3` kanál směřuje na poslední digitální výstup Y31 automatu s adresou 3

Rozsahy indexů pro jednotlivé oblasti:

typ	***dolní index***	***horní index***
sys_X	0	31
sys_Y	0	31

Uživatelské a síťové bity M (sys_M, sys_netM)

Uživatelské bity jsou označovány v jazyce Simple jako pole bitů M (bity M0 až M127). Uživatelské znamená, že jsou volně k použití, a že jejich obsah je dán konkrétním uživatelským programem v automatu (pokud je pole M

programem vůbec použito). Do tohoto označení bitového pole M v jazyce Simple spadá také pole bitů, které jsou sdíleny mezi všemi automaty v síti. Tyto sdílené bity se nacházejí v rozsahu bitů M64 až M127.
Paměťová oblast **sys_M** pokrývá celou oblast pole bitů M, tedy M0 až M127 a součástí popisu paměťové oblasti **sys_M** je adresa automatu.
Paměťová oblast **sys_netM** pokrývá pouze síťové bity, tedy podmnožinu uživatelských bitů, v rozsahu M64 až M127. Adresa automatu není součástí popisu položky z této paměťové oblasti.

	parametr			
	1.	**2.**	**3.**	**4.**
paměťová oblast	***typ paměti***	***typ datové proměnné***	***adresa automatu***	***adresa proměnné v příslušné oblasti***
uživatelské bity M	**sys_M**	bit	0-31 (příp. EXbus)	--
síťové bity M	**sys_netM**	bit	--	--

Popis parametrů „adresa automatu“ a „typ datové proměnné“ lze nalézt ve shodně nazvaných podkapitolách uvedených dříve v textu. Přístup k datové proměnné je v této paměťové oblasti omezen na typ bit.

Adresace bitů

Adresace v rámci této oblasti paměti probíhá operátorem přístupu do pole, který se používá spolu s typem datové proměnné.
Základní a jediný nativní datový typ těchto oblastí je bit. Velikost **sys_M** oblasti je 128 bitů, **sys_netM** oblast je podmnožinou **sys_M** o velikosti 64 bitů. Adresace prvních a posledních bitů obou oblastí jsou následující:

`sys_M; bit[0]; 3` kanál směřuje na první uživatelský bit M0 automatu s adresou 3

`sys_M; bit[127]; 3` kanál směřuje na poslední uživatelský bit M127 automatu s adresou 3

`sys_netM; bit[64]` kanál směřuje na první síťový bit M64 převodníku

`sys_netM; bit[127]` kanál směřuje na poslední síťový bit M127 převodníku

Rozsahy indexů pro jednotlivé oblasti:

typ	***dolní index***	***horní index***
sys_M	0	127
sys_netM	64	127

Pozn. Při zápisu síťových bitů M64 až M127 se o zaslání hodnot těchto bitů i do sítě PESnet postará cílový automat, pokud není vývojově starší než řady 400 (dříve rozeslání zajišťoval převodník CA21/CA3/CA4).
Pozn. Při přístupu do oblasti **sys_netM** se automaticky přistupuje do paměti automatu/převodníku, s nímž je vytvořeno přímé spojení. Proto není třeba v popisu kanálu specifikovat adresu automatu.

Speciální funkční bity B (sys_B)

Speciální funkční bity je pole bitů označované v jazyce Simple jako pole bitů B (B0 až B127). Význam těchto bitů naleznete v dokumentaci k automatu.

	parametr			
	1.	**2.**	**3.**	**4.**
paměťová oblast	***typ paměti***	***typ datové proměnné***	***adresa automatu***	***adresa proměnné v příslušné oblasti***
speciální funkční bity	**sys_B**	bit	0-31 (příp. EXbus)	--

Popis parametrů „adresa automatu“ a „typ datové proměnné“ lze nalézt ve shodně nazvaných podkapitolách uvedených dříve v textu.

Adresace bitů

Adresace v rámci této oblasti paměti probíhá operátorem přístupu do pole, který se používá spolu s typem datové proměnné.

Základní a jediný nativní datový typ této oblasti je bit, velikost oblasti je 128 bitů. Adresace prvního a posledního bitů v automatu s adresou 3 jsou následující:

`sys_B; bit[0]; 3` kanál směřuje na první uživatelský bit M0 automatu s adresou 3

`sys_B; bit[127]; 3` kanál směřuje na poslední uživatelský bit M127 automatu s adresou 3

Rozsah indexů v oblasti je tedy od 0 do 127.

Analogové I/O (sys_I, sys_O)

Pole analogových vstupů a výstupů automatů vývojově starších než řady 400 je v jazyce Simple označováno jako wordové pole I a O. Přístupem do těchto oblastí má uživatel plný přístup na reálné analogové vstupy/výstupy automatu (pokud jsou tyto pozice osazeny).

	parametr			
	1.	**2.**	**3.**	**4.**
paměťová oblast	***typ paměti***	***typ datové proměnné***	***adresa automatu***	***adresa proměnné v příslušné oblasti***
analogové vstupy	**sys_I**	word, int	0-31	--
analogové výstupy	**sys_O**	word, int	0-31	--

Popis parametrů „adresa automatu“ a „typ datové proměnné“ lze nalézt ve shodně nazvaných podkapitolách uvedených dříve v textu. Přístup k datové proměnné je v této paměťové oblasti omezen na typy word nebo int.

Adresace vstupů a výstupů

Adresace v rámci této oblasti paměti probíhá operátorem přístupu do pole, který se používá spolu s typem datové proměnné.

Nativním a jediným datovým typem je word, nebo int jakožto znaménkový word. Velikost obou popisovaných oblastí je 32 wordů. Adresace prvního analogového vstupu a posledního výstupu v automatu s adresou 3 mohou být následující:

`sys_I; word[0]; 3` kanál směřuje na první analogový vstup I0 automatu s adresou 3

`sys_O; int[31]; 3` kanál směřuje na poslední analogový výstup O31 automatu s adresou 3

Rozsahy indexů pro jednotlivé oblasti:

typ	***dolní index***	***horní index***
sys_I	0	31
sys_O	0	31

Uživatelské a síťové wordy D (sys_D, sys_netD)

Uživatelské wordy jsou označovány v jazyce Simple jako pole wordů D (bity D0 až D63). Uživatelské znamená, že jsou volně k použití, a že jejich obsah je dán konkrétním uživatelským programem v automatu (pokud je pole D programem vůbec použito). Do tohoto označení pole D v jazyce Simple spadá také pole wordů, které jsou sdíleny mezi všemi automaty na síti. Tyto sdílené wordy se nacházejí v rozsahu wordů D32 až D63.

Paměťová oblast **sys_D** pokrývá celou oblast pole wordů D, tedy D0 až D63. Součástí popisu paměťové oblasti **sys_D** je adresa automatu.
Paměťová oblast **sys_netD** pokrývá pouze síťové wordy, tedy podmnožinu uživatelských wordů, a to v rozsahu D31 až D63. Adresa automatu není součástí popisu položky z této paměťové oblasti.

	parametr			
	1.	**2.**	**3.**	**4.**
paměťová oblast	***typ paměti***	***typ datové proměnné***	***adresa automatu***	***adresa proměnné v příslušné oblasti***
registry D	**sys_D**	word, int	0-31 (příp. EXbus)	--
síťové registry D	**sys_netD**	word, int	--	--

Popis parametrů „adresa automatu“ a „typ datové proměnné“ lze nalézt ve shodně nazvaných podkapitolách uvedených dříve v textu. Přístup k datové proměnné je v této paměťové oblasti omezen na typy word nebo int.

Adresace registrů

Adresace v rámci této oblasti paměti probíhá operátorem přístupu do pole, který se používá spolu s typem datové proměnné.
Nativním a jediným datovým typem je word, nebo int jakožto znaménkový word. Velikost **sys_D** oblasti je 64 wordů, **sys_netD** oblast je podmnožinou **sys_D** o velikosti 32 wordů. Adresace prvních a posledních registrů obou oblastí mohou být následující:

`sys_D; word[0]; 3`	kanál směřuje na první uživatelský word D0 automatu s adresou 3
`sys_D; int[63]; 3`	kanál směřuje na poslední uživatelský word D63 automatu s adresou 3
`sys_netD; int[32]`	kanál směřuje na první síťový word D32 převodníku
`sys_netD; word[63]`	kanál směřuje na poslední síťový word D63 převodníku

Rozsahy indexů pro jednotlivé oblasti:

typ	***dolní index***	***horní index***
sys_D	0	63
sys_netD	32	63

Pozn. Při zápisu síťových wordů D32 až D63 se o zaslání hodnot těchto wordů i do sítě PESnet postará cílový automat, pokud není vývojově starší než řady 400 (dříve rozeslání zajišťoval převodník CA21/CA3/CA4).
Pozn. Při přístupu do oblasti **sys_netD** se automaticky přistupuje do paměti automatu/převodníku, s nímž je vytvořeno přímé spojení. Proto není třeba v popisu kanálu specifikovat adresu automatu.

Speciální funkční registry W

Speciální funkční registry W je pole wordů označované v jazyce Simple jako pole wordů W (W0 až W127). Význam těchto bitů naleznete v katalogu k automatu (jsou mezi nimi je například registry reálného času).

	parametr			
	1.	**2.**	**3.**	**4.**
paměťová oblast	***typ paměti***	***typ datové proměnné***	***adresa automatu***	***adresa proměnné v příslušné oblasti***
speciální funkční registry	**sys_W**	word, int	0-31 (příp. EXbus)	--

Popis parametrů „adresa automatu“ a „typ datové proměnné“ lze nalézt ve shodně nazvaných podkapitolách uvedených dříve v textu. Přístup k datové proměnné je v této paměťové oblasti omezen na typy word nebo int.

Adresace registrů

Adresace v rámci této oblasti paměti probíhá operátorem přístupu do pole, který se používá spolu s typem datové proměnné.

Nativním a jediným datovým typem je word, nebo int jakožto znaménkový word, velikost oblasti je 128 wordů. Adresace prvního a posledního registru oblasti v automatu s adresou 3 mohou být následující:

`sys_W; int[0]; 3`	kanál směřuje na první uživatelský word W0 automatu s adresou 3
`sys_W; word[127]; 3`	kanál směřuje na poslední uživatelský word W127 automatu s adresou 3

Rozsah indexů v oblasti je tedy od 0 do 127.

Uživatelské a síťové proměnné L (sys_L, sys_netL)

Uživatelské proměnné L je pole obecně použitelných 256 síťových 4bajtových proměnných typu longword. V jazyce Simple4 má pole L též alternativní názvy: NetLW (pole typu longword), NetLI (pole typu longint) a NetF (pole typu float). Zápis jakékoli hodnoty položky pole automatu bude automatem odvysílán do sítě PESnet, pokud není automat vývojově starší než řady 400 (v opačném případě je třeba je pro zajištění odvysílání hodnoty do sítě zapsat příslušnou hodnotu do pole L převodníku či automatu řady 400).

	parametr			
	1.	**2.**	**3.**	**4.**
paměťová oblast	***typ paměti***	***typ datové proměnné***	***adresa automatu***	***adresa proměnné v příslušné oblasti***
uživatelské síťové proměnné L	**sys_L**	bit, byte, word, int, longword, longint, float	0-31 (příp. EXbus)	--
uživatelské síťové proměnné L	**sys_netL**	bit, byte, word, int, longword, longint, float	--	--

Popis parametrů „adresa automatu“ a „typ datové proměnné“ lze nalézt ve shodně nazvaných podkapitolách uvedených dříve v textu. Přístup k datové proměnné není této paměťové oblasti omezen typem.

Adresace registrů

Adresace v rámci této oblasti paměti probíhá operátorem přístupu do pole, který se používá spolu s typem datové proměnné.

Nativním datovým typem v této oblasti je longword, k přístupu ale lze využít libovolný datový typ. Velikost oblasti je 256 longwordů. Adresace prvního a posledního registru oblastí mohou být následující:

`sys_L; longword[0]; 3`	kanál směřuje na první registr L0 automatu s adresou 3
`sys_L; longint[255]; 3`	kanál směřuje na poslední registr L255 automatu s adresou 3
`sys_netL; longint[0]`	kanál směřuje na první síťový registr L0 převodníku
`sys_netL; longword[255]`	kanál směřuje na poslední síťový registr L255 převodníku

Rozdíl mezi prostory sys_L a sys_netL

Oba prostory pokrývají stejný adresní prostor. U prostoru **sys_netL** se ale neuvádí adresa cílového automatu, požadavek čtení nebo zápisu pak automaticky směřuje do paměti automatu/převodníku, s nímž je vytvořeno přímé spojení. Převodník již zajistí rozeslání hodnot do sítě PESnet.

Registry periferie EX (perex)

Tato oblast paměti je svázána s typem zařízení, které se nachází na uvedené adrese automatu. Zařízením je předpokládána periferie typu EX (EX01, EX02 aj.).
Význam jednotlivých registrů je určen typem periferie a slouží k nastavení chování dané periferie.

	parametr			
	1.	**2.**	**3.**	**4.**
paměťová oblast	***typ paměti***	***typ datové proměnné***	***adresa automatu***	***adresa proměnné v příslušné oblasti***
registry periferie EX	**perex**	word, int	0-31 (příp. EXbus)	--

Popis parametrů „adresa automatu“ a „typ datové proměnné“ lze nalézt ve shodně nazvaných podkapitolách uvedených dříve v textu. Přístup k datové proměnné je v této paměťové oblasti omezen na typy word nebo int.

Adresace registrů

Adresace v rámci této oblasti paměti probíhá operátorem přístupu do pole, který se používá spolu s typem datové proměnné.
Nativním a jediným datovým typem je word, nebo int jakožto znaménkový word. Velikost oblasti je 15 wordů, indexace registrů začíná od 1. Adresace prvního a posledního registru oblasti v automatu s adresou 3 mohou být následující:

`perex; int[1]; 3` kanál směřuje na první registr periferie EX s adresou 3

`perex; word[15]; 3` kanál směřuje na poslední registr periferie EX s adresou 3

Rozsah indexů v oblasti je tedy od 1 do 15.

Ukázka přístupu k periferii EX

deklarace	***typ dotčeného registru***
perex; word[1]; 3	konfigurační registr ADDR (W[65])
perex; word[2]; 3	konfigurační registr BAUD (W[66])
perex; word[15]; 3	datový registr MAP7 (W[79])

Použití protokolu MEM v programu Microsoft Excel

Zavedením služby ADVICE (trvalého propojení) v tématu MEM se výrazně zjednodušilo využití Excelu jako zobrazovače dat automatu MICROPEL. Zjednodušení spočívá především v tom, že již není třeba externím makrem řešit čtení (aktualizaci dat) v pevném časovém intervalu. Aktualizaci dat zajišťuje sám program DdeAP a prostřednictvím služby ADVICE zasílá změny dat.

Příkaz trvalého propojení DDE v buňce

Požadavek na datové propojení DDE lze zapsat přímo do buňky Excelu. Formát je následující:

znak DDE propojení	***jméno DDE serveru***	***oddělovač tématu***	***jméno tématu***	***oddělovač položky***	***obsah položky***
=	**pesdde**	**|**	**mem**	**!**	**'protokol MEM'**

Protože Excel jako první typ propojení DDE zkouší propojení typu ADVICE, tak stačí do buňky Excelu napsat příkaz propojení DDE a žádost o trvalé propojení je na světě. DdeAP se sám stará o pravidelné vyčítání hodnot a pokud došlo od posledního čtení ke změně, pošle novou hodnotu klientovi, tedy v našem případě do buňky Excelu.

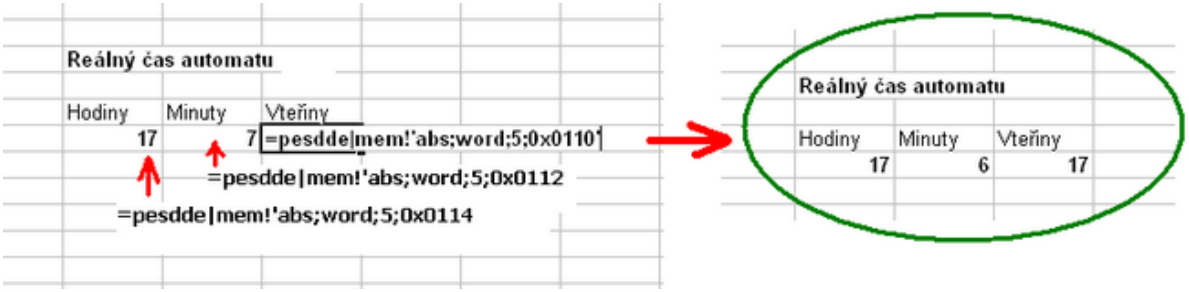
Příklad vizualizace reálného času automatu

Příklad: propojení na reálný času automatu s adresou v síti 5

vteřiny **=pesdde|mem!'abs;word;05;0x0110'**

minuty **=pesdde|mem!'abs;word;05;0x0112'**

hodiny **=pesdde|mem!'abs;word;05;0x0114'**



Problém se zápisem z programu Excel

Žádost o trvalé propojení, tedy o trvalé čtení, je jednoduchá. Problém nastane, pokud chceme naopak zapsat hodnotu z Excelu do automatu. K tomu je třeba použít službu DDE, tzv. službu POKE. To již není tak jednoduché vyvolat, a není způsob, jak to jednoduše zapsat přímo do buňky Excelu. Při zápisu do buňky se Excel pokusí provést trvalé propojení, a pokud je neúspěšný, zkouší službu REQUEST. Jsou v zásadě dva způsoby zápisu.

Visual Basic for Application

Vestavěný makro jazyk - VBA obsahuje metodu *DDEPoke(Channel, Item, Data)*. Bylo by možné tuto funkci volat pro zápis (například jako reakce na stisk tlačítka ve formuláři nebo na ploše), ale má to zádrhel. Zřejmě chybou v jazyce VBA není možné předat jako položku *Item* programově vyplněný textový řetězec, ale pouze odkaz na jinou buňku Excelu, kde je možné umístit požadovaný řetězec (v našem případě položka protokolu MEM). Tato chyba použití makra jazyka VBA značně diskvalifikuje.

Využití tématu „var“

Aby přesto bylo možné zapsat hodnotu formátem protokolu MEM, byla zvolena následující obezlička. Využilo se tématu **var**, který nepodporuje službu ADVICE a kde lze i zápis protokolem VAR provádět službou REQUEST. Protokol tématu **var** byl rozšířen o úvodní klíčové slovo **mem**, kterým se říká, že následuje speciální zápis požadavku v protokolu MEM. Zápis následuje klíčovým slovem určujícím směr operace (**write**, **read**). Dále již následuje „čistý“ protokol MEM.

Tohoto způsobu lze využít jednak při přímém zápisu do buňky, nebo při použití VBA metodou *DDERequest(Channel, Item)*, která pracuje správně (na rozdíl od metody *DDEPoke*). A způsob lze využít i u jiných klientů, kteří v zásadě umí pouze číst, tedy používat službu REQUEST (což je případ přímého zápisu DDE formule v Excelu).

Položka rozšířeného protokol tématu VAR	
čtení	**mem;read;“*deklarace paměťového místa protokolem MEM*“**
zápis	**mem;write; “*deklarace paměťového místa protokolem MEM*“;data**

Formulka propojení DDE do buňky Excelu	
čtení	**=pesdde|var!'mem;read;"*deklarace paměťového místa protokolem MEM*"**
zápis	**=pesdde|var!'mem;write;"*deklarace paměťového místa protokolem MEM*";data**

Příklad: zápis hodnoty 12 do registru vteřin reálného času

vteřiny **=pesdde|var!'mem;write; abs;word;5;0x0110;12'**

Příklad: čtení hodnoty registru vteřin reálného času

vteřiny **=pesdde|var!'mem;read; abs;word;5;0x0110'**