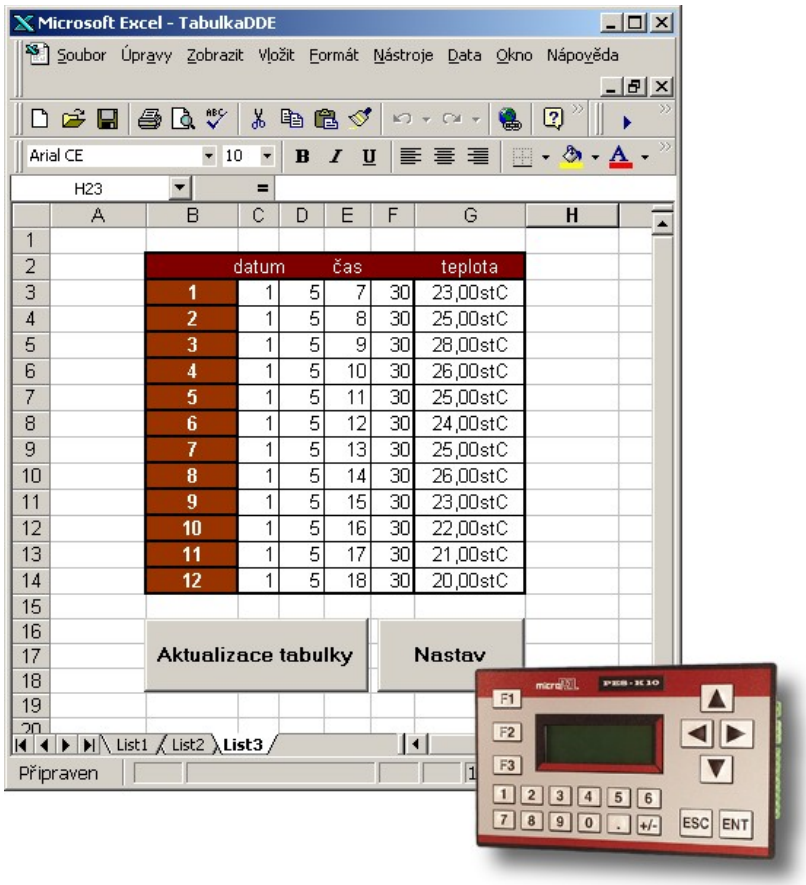
ExcelVBAmodul

Podpora pro výměnu dat automatů MICROPEL s aplikací Microsoft Excel



ExcelVBAmodul

Podpora pro výměnu dat automatů MICROPEL s aplikací Microsoft Excel

2. verze dokumentu
02/2009
MICROPEL s.r.o.

Tomáš Navrátil
navratil@micropel.cz

Úvod

Program EXCEL, z balíku programů Microsoft Office, umožňuje zobrazovat a nastavovat hodnoty v síti automatů MICROPEL. Tato schopnost je dána tím, že náš softwarový server DataServer (nebo starší PesDDE) poskytuje DDE rozhraní, které podporuje také program EXCEL. Program EXCEL umožňuje v zásadě dvě možnosti, jak přistupovat k datům v síti automatů MICROPEL. Jednak přímým zápisem formule pro rozhraní DDE do buňky EXCELu a jednak použitím vestavěného programovacího jazyku VBA (Visual Basic for Application). Obě možnosti mají své výhody i nevýhody. Pro použití VBA jsme připravili univerzální funkce pro mapovaní dat z tabulky EXCELu do automatu a zpět.

Přílohou je příklad dokumentu excelu VizDDE.xls.

A. Přímý zápis požadavku do buňky

Obecně lze tuto metodu popsat tak, že uživatel zapíše do buňky EXCELu přímo textový příkaz (protokolu komunikace po DDE) pro čtení nebo zápis, program EXCEL tento formuli provede a do buňky zobrazí buďto data (v případě čtení ze sítě) a nebo výsledek operace (v případě zápisu dat do sítě).
Textový formát požadavku zapsaného do buňky je (například čtení stacku z automatu s adresou 30, index stacku je 0, počet čtených dat je 1:

=pesdde|var!'rs[30][0][1]'

Kde:

=	uvození vzorce (vlastnost excelu)
pesdde	název služby DDE, kterou poskytuje aplikace DataServer nebo PesDDE
var	název tématu výše uvedené služby
'rs[30][0][1]'	vlastní text požadavku na provedení příkazů

Pozn.: podrobnější popis možností čtení a zápisu tohoto protokolu je popsán v dokumentaci k aplikacím DataServer a PesDDE.
Nevýhodou přímého zápisu formule DDE do buňky je jednak formát dat v případě že se vyčítá a nebo zapisuje více než jedna hodnota a jednak to, že se standardně provede pouze v okamžiku vložení do buňky.

B. Použití maker pro VBA (Visual Basic for Application)

Tato metoda umožňuje v EXCELu mapovat tabulky dat do datových oblastí jednotlivých automatů (a obráceně). Tato výměna dat je inicializovaná uživatelem (nejčastěji aktivací tlačítka) nebo vyvoláním události (například od časovače). Výhodou je jednoduché použití (napojením na naše funkce lze získat již hotové mapování) a za výhodu lze také považovat umístění kódu VBA mimo uživatelskou oblast koncového uživatele a uložením tohoto kódu pod přístupové heslo. Následující část dokumentu se věnuje použití maker VBA.

Prostředky pro datové propojení Excelu a PLC

Instalace MICROPEL DataServer (nebo PesDDE)

Podmínkou napojení programu EXCEL na síť automatů MICROPEL je server, aplikace MICROPEL DataServer. Ten program pomocí portu COM, USB ,Ethernet komunikuje s automaty a data poskytuje pomocí unifikovaného rozhraní DDE do operačního systému.

Pozn.: Tento program je třeba před spustit před spuštěním programu EXCEL, jehož dokument obsahuje makra pro komunikaci se serverem PesDDE.

Instalace maker VBA potřebné podpory

Balíček souborů pro podporu komunikace Excelu obsahuje:

iVBAmodul.exe
VBAmodul.dl_
DDEClientXLS.bas

Tato část má dva kroky. Prvním krokem je **instalace vlastních funkcí pro VBA**. Ta jsou uložena v otevřeném kódu VB v souboru DDEClientXLS.bas a je třeba je importovat do každého dokumentu Excelu, který má komunikovat s automaty. Instalace je taková, že v programu EXCEL vytvoříme nový dokument (nebo otevřeme již existující), zapneme vestavěný editor VBA a v nabídce Soubor (při vybraném projektu, který nese název otevřeného dokumentu Excelu) vybereme Import soubor. Zobrazí se dialog pro nalezení souboru a po vybrání DDEClientXLS.bas v seznamu projektu (levé okno) se zobrazí stejnojmenný modul.

Druhým krokem je **instalace knihovny VBAmodul.dll**. (ta poskytuje funkcím posuvník, pro zobrazování časově náročných akcí). Instalace spočívá ve spuštění programu iVBAmodul.exe, který požadovanou knihovnu uloží do systému a také ji zaregistruje. Pozn.: tento instalační program stačí spustit jen jednou.

Popis funkcí modulu DDEClientXLS.bas

Instalovaný modul DDEClientXLS (viz výše) poskytuje tyto výkonné funkce
WriteTableToPLC (…)
ReadTableFromPLC (…)
ProcessDDE(…)
a také inicializační funkce
InitVBAmodul()
CloseVBAmodul()
CloseVBAmodul2(…)

Úkolem těchto funkcí je přenášet vybranou oblast buněk do a z automatu. Výkonné funkce mají stejný seznam parametrů a stejnou návratovou hodnotu pro vyjádření výsledku operace. Liší se jen směrem přenosu dat. Inicializační funkce se volají před a po ukončení operce.

WriteTableToPLC

Tato funkce čte zadanou tabulku a hodnoty z tabulky zapisuje do paměti automatu. Parametry funkce popisují zdroj dat (tabulku v EXCELu) a cíl (automat).

Tabulka nemusí být jen dvojrozměrná, ale může být ve formě řádku a nebo sloupce případně tabulkou můžeme roumět také jednu samostatnou buňku. V každém případě je oblast tabulky určena levým horním a pravým dolním bodem, tedy buňkou EXCELu. Také je třeba určit jak se tabulka má vyčítat. Zda-li po sloupcích (parametr COLUMN) a nebo řádcích (parametr LINE). Tento parametr je třeba uvést i v případě, že se jedná o jednorozměrnou tabulku (řádek nebo sloupec) a směr je vlastně dán.

Cílem dat je automat. Tam je třeba určit adresu automatu v síti, typ paměti (stack, bity, wordy) a adresu v daném typu paměti, od které se budou data zapisovat.

Nepovinným parametrem je parametr perLen. Ten používáme v případě, že na jednu akci uživatele voláme více funkcí WriteTableToPLC a nebo ReadTableFromPLC za sebou. Nenulová hodnota tohoto parametru sděluje funkci , jaká procentuální část ukazatele průběhu mu připadá. Tato hodnota by měla vyjadřovat procentuální množství dat, které volaná funkce zapisuje z celkového množství dat, zapisovaných (nebo čtených) v daném makru. Pokud se tato hodnota vynechá, pak si funkce přisoudí celý ukazatel průběhu. Tento stav odpovídá situaci, že máme jen jeden příkaz na čtení a nebo zápis v uživatelském makru.

Deklarace funkce :

```
Function WriteTableToPLC( ByVal direction As String, _
                          ByVal startLine, _
                          ByVal startCol, _
                          ByVal endLine, _
                          ByVal endCol As Integer, _
                          ByVal typeMem As String, _
                          ByVal addrPLC As Integer, _
                          ByVal addrMem As Integer, _
                          Optional ByVal perLen = 100) As Boolean
```

Popis parametrů:

direction – textová hodnota “LINE“ nebo “COLUMN“ určuje, jak bude tabulka vyčítána jestli po řádcích nebo sloupcích
startLine & startCol - číselné definice levé horní buňky tabulky
endLine & endCol - číselné definice pravé dolní buňky tabulky
typeMem - textová proměnná určuje druh paměti, kam se data v automatu budou zapisovat, může nabývat hodnot "stack" , "bit", "word"
addrPLC - adresa automatu v siti PESNET
addrMem - adresa v paměti, kam se mají nahrát data z tabulky
POZOR! Rozumí se absolutní adresa – viz. dokumentace k aplikaci DataServer

perLen - (nepovinné) hodnota od 0 do 100, představuje takovou procentuální část proužkového ukazatele, který má tato funkce k dispozici pro zobrazování průběhu operace

Návratová hodnota:
Návratová hodnota je typu BOOL (pravda / nepravda) a vyjadřuje požadavek uživatele na ukončení operace. Na ukazateli průběhu operace je tlačítko STORNO. Pokud uživatel zmáčkne toto tlačítko, tak návratová hodnota funkce WriteTableToPLC je FALSE (nepravda). Pokud uživatel tlačítko nezmáčkne je návratová hodnota funkce WriteTableToPLC TRUE. Z toho plyne konstrukce použití funkce ukázána v následujícím příkladu (viz Použití funkcí).

ReadTableFromPLC

Tato funkce má stejný seznam parametrů a stejnou návratovou hodnotu pro vyjádření výsledku operace jako funkce WriteTableToPLC. Liší se jen směrem přenosu dat. Tato funkce tedy naopak čte z paměťové oblasti automatu hodnoty a zapisuje je do zadané tabulky EXCELu.

ProcessDDE

Tato funkce slouží k vykonání požadavku čtení nebo zápisu přímo z kódu VBA bez napojení na nějakou buňku listu. Předchozí funkce návratovou hodnotu uložili (nebo četli) z vybrané oblasti buněk. Pokud ale potřebujeme přímo v kódu VBA znát hodnotu, například aby se na základě hodnoty v automatu rozhodlo jaká oblast se má číst, lze použít tuto funkci.

Deklarace funkce:

```
Function ProcessDDE(ByVal itemToDDE As String) As String
```

Jako parametr **itemToDDE** je třeba vložit textový řetězec ve formátu komunikace DDE (viz dokumentace DataServer)

Například resetovaní programu:

```
'Public Sub Reset_PLC()
        result = ProcessDDE("sb[9][318]1")     'pozn.: RESET je index 318
        If result = "OK" Then MsgBox "Program resetován"
        If result = "ERROR" Then MsgBox "Chyba operace!"
'End Sub
```

InitVBAmodul

```
Sub InitVBAmodul()
```

Tato procedura inicializuje globální hodnoty potřebné pro práci výše uvedených funkcí a je tedy nutné ji volat před voláním funkcí ReadTableFromPLC a WriteTableToPLC.

CloseVBAmodul

```
Sub CloseVBAmodul()
```

Tato procedura uzavírá globální objekty a v případě, že během komunikace došlo k chybám, vypíše seznam adres automatů, s nimiž nenavázal spojení.

CloseVBAmodul2

```
Sub CloseVBAmodul2(msg As Boolean)
```

Tato procedura také uzavírá globální objekty jako předchozí s tím, že případný dialog zobrazí pouze pokud byl nastaven předaný parametr msg = 1. Pokud je nastaven na 0, pak se provede „tiché“ ukončení operace.

Použití funkcí

Pro správnou funkci je třeba dodržet také pravidla pro použití výše uvedených funkcí. Následující postup je vhodné dodržet v každém samostatném bloku zpracovávajícího data jako výsledek jedné události. Typicky jako blok funkcí, spuštěných jako reakce na stisk tlačítka.

1. Nejdříve je nutné zavolat inicializační funkci InitVBAmodul

2. Pote jé možné volat funkce ReadTableFromPLC a WriteTableToPLC a to i na různé adresy atomatů.
3. Na závěr zavolat funkci CloseVBAmodul (a to buď se zobrazením chyb nebo bez zobrazení)

Funkce ProcessDDE je možné volat bez volání funkcí InitVBAmodul a CloseVBAmodul.

Protože návratová hodnota funkcí Read… a Write… odráží případný požadavek uživatele na stornování akce (ukazatel průběhu má tlačítko STORNO), je doporučená konstrukce použití funkcí následující:

```
Sub Reakce_na_stisk_tlacitka()

        ‘1. inicializace komunikace
        Call InitVBAmodul

        ‘2. Vlastni vymena dat. Pokud je zaznamenan pozadavek na STORNO je mozne volat konec
makra
        If  WriteTableToPLC("LINE", 10, 3, 20, 15, "stack", 30, 10, 50) = False Then GoTo ENDmakro
        If  WriteTableToPLC("COLUMN", 22, 3, 32, 15, "bit", 1, 20, 50) = False Then GoTo ENDmakro

        ‘3. konec komunikace
ENDmakro:
        Call CloseVBAmodul
End Sub
```

Příklady

Příklad:

```
Private Sub Tlačítko_Click()

        Call InitVBAmodul        ‘inicializace globálních objektů

        If  WriteTableToPLC("LINE", 10, 3, 20, 15, "stack", 30, 10, 50) = False Then GoTo ENDmakro

‘ řádkový zápis dat z tabulky, jejíž levý horní roh je buňka 10,3, pravý dolní roh je buňka 20,15,
zapisovat se bude do stacku automatu s adresou 30 a do místa ve stacku s adresou 10. Toto makro
má k dispozici 50% ukazatele průběhu

        If  WriteTableToPLC("COLUMN", 22, 3, 32, 15, "bit", 1, 20, 50) = False Then GoTo ENDmakro

‘sloupcový  zápis dat z tabulky, jejíž levý horní roh je buňka 22,3, pravý dolní roh je buňka 32,15,
zapisovat se bude do automatu s adresou 01 a do bitových proměnných s adresou s adresou 20. Toto
makro má k dispozici 50% ukazatele průběhu

ENDmakro:                                                                ‘návěšťí konce programu

        Call CloseVBAmodul ‘ ukončení globálních objektů

End Sub
```

Příklad řádkové tabulky, kde každý řádek odpovídá jinému automatu:

```
Private Sub Tlačítko2_Click()
```

```
Call InitVBAmodul                                                    ‘inicializace globálních
objektů

If  WriteTableToPLC("LINE", 10, 3, 10, 15, "stack", 30, 10, 100 /3) = False    Then GoTo ENDmakro
If  WriteTableToPLC("LINE", 11, 3, 11, 15, "stack", 1, 20, 100/ 3)   = False    Then GoTo ENDmakro
If  WriteTableToPLC("LINE", 12, 3, 12, 15, "stack", 1, 20, 100/3)    = False    Then GoTo ENDmakro

ENDmakro:                                                            ‘návěšťí konce programu

Call CloseVBAmodul ‘ ukončení globálních objektů

End Sub
```

Chybové stavy, hlášení

Pokud některá funkce (WriteTableToPLC, ReadTableFromPLC) nemůže být provedena (tady bývá ta chyba, že cílový automat neodpovídá), zapíše si vzniklou chybu do svého vnitřního objektu, ale její návratová hodnota je TRUE, tedy následující funkce mohou bít vykonány. Vnitřní chybový objekt je zobrazen až po ukončení komunikace a zavolání funkce CloseVBAmodul. To znamená, že pokud je v uživatelském makru více funkcí (WriteTableToPLC, ReadTableFromPLC) provedou se všechny i v případě lokálních chyb a chybné adresy se zobrazí až na konci provádění makra.

Návody na práci s VBA

Napojení maker na Tlačítko

Máme-li nějakou oblast dat v Excelu kterou chceme ukládat a nebo číst, je třeba tuto operaci odvodit od akce uživatele. Nejjednodušším způsobem je napojením na tlačítko. Pro gram Excel umožňuje vložit tlačítko dvěma způsoby:

1) Z menu Zobrazit (View) vybrat Panely nástrojů (Tollbars) a zaškrtnout položku Visual Basic. Zobrazí se panel s mininabídkou. Kliknutím na tlačítko Ovládací prvky se zobrazí panel s prvky. Vybereme tlačítko a a na požadovaném místě ho vytvoříme pomocí kurzoru myši. Pravým tlačítkem myši dostaneme místní nabídku. Přes položku Vlastnosti můžeme měnit nápis na tlačítku (Caption), velikost písma (Font). Položka (Name) určuje, jaké makro se bude připojovat (Name_Click). Přes položku Zobraz kód se dostaneme do předdefinovaného makra, které je prázdné a je připojené k tlačítku a jmenuje se podle názvu tlačítka a přípony _Click. V tomto makru je pak třeba volat potřebné funkce InitVBAmodul, WriteTableToPLC, ReadTableFromPLC, CloseVBAmodul.
   Po ukončení editace se klikne na tlačítko Ukončit režim návrhu v nabídce panelu nástrojů Visual Basic a tím se možnosti editace tlačítka zamknou.

2) Z menu Zobrazit (View) vybrat Panely nástrojů (Tollbars) a zaškrtnout položku Formuláře (Forms). Zobrazí se panel, nabízející různé prvky. Vybereme prvek Tlačítko a na požadovaném místě ho vytvoříme pomocí kurzoru myši. K tomuto prvku přiřadíme makro, které musí již existovat a nebo které v editoru VBA musíme sami vytvořit. Tato metoda ale umožňuje editovat tlačítko koncovému uživateli jako každý jiný prvek a navíc neautomatizuje napojení na makro v editoru VBA na rozdíl od předcházející metody.

Uzamknutí kódu

Projekt ve VBA editoru lze uzamknout proti nežádoucímu zásahu. Ve VBA editoru v levém okně (prohlížeč projektů) vybereme náš projekt s modulem DDEClientXLS a s makry pro tlačítka, vybereme přes místní nabídku Vlastnosti projektu a vybereme záložku Zámek. Tam zaškrtneme zamknutí a nastavíme heslo. Po příštím otevření VBA editoru bude požadováno heslo před prohlížením projektu VBA.

Opakované vyčítání dat

Excel umožňuje volání libovolné funkce v předem určenou dobu. K tomu slouží funkce objektu aplikace **OnTime (čas volání, jméno volané funkce)** .Tato funkce má jako 1. parametr čas volání stanovené funkce a 2. parametr je právě jméno funkce, která se bude ve stanovený čas volat

Řešení periodického vyčítání dat je ve dvojici navzájem se volajících funkcí, kde jedna funkce nastavuje čas volání druhé funkce a tato druhá funkce provádí vyčítání dat plus nastavení příštího volání na sama sebe.

První funkce (v příkladu nazvaná VyctiData) provede vyčtení dat (nebo cokoliv jiného) a na závěr zavolá funkci (SetNextCall) pro nastavení příštího času volání funkce sama sebe, tedy funkce VyctiData.

Ve funkci SetNextCall se volá funkce OnTime a jako parametr času se předá aktuální čas (Now) plus 5 veřin (TimeValue("00:00:05")) a název volané funkce je Vyčti data.

Příklad:

```
Sub VyctiData ()
        'zde je kód pro vyčtení dat, např.:
        Range („A1“) = “ =pesdde|var!’rs[1][0]’”

        'zde je inicializace dalšího volaní fce VyctiData
        Call SetNextCall()
End Sub

Sub SetNextCall ()
   Application.OnTime  Now + TimeValue("00:00:05"), "VyctiData"
End Sub
```

Upozornění: rozsah platnosti funkce jazyka VBA

Pokud se funkce VBA deklarují v Listech a nikoliv v modulu, který je globální, je třeba ve funkci OnTime určit plné jméno funkce včetně názvu listu. Např.:

```
Sub SetNextCall ()
   Application.OnTime  Now + TimeValue("00:00:05"), "List1.VyctiData"
End Sub
```