

Komunikační server PLCComS

Uživatelská příručka

22. vydání - duben 2020

Historie změn

| Datum | Verze | Popis změn |
|---------------|-------|---|
| Březen 2010 | 1 | První verze. |
| Duben 2010 | 2 | Zpracování proměnných typu <i>date</i> , <i>dt</i> , <i>time</i> a <i>tod</i> . Rozšíření příkladu o zpracování atributů souboru. |
| Srpen 2010 | 3 | Přidáno napojení na sdílený modul. Rozšíření a úprava konfiguračního souboru. Barevné zvýraznění příkladů v jazyce „C“. Přidána kapitola s příklady použití serveru a licencí. |
| Červen 2011 | 4 | Rozšíření konfiguračního souboru. |
| Srpen 2011 | 5 | Rozšířený popis kapitoly 3.2 Uživatelský public soubor . Opravena tabulka 5.1 Seznam zpráv . |
| Září 2011 | 6 | Upravena kapitola Chyba: zdroj odkazu nenalezen Chyba: zdroj odkazu nenalezen a rozšířena tabulka 8 Podporované operační systémy . |
| Říjen 2011 | 7 | Upraveny kapitoly 2.1 Příklad konfiguračního souboru , 4 Klientský protokol a tabulka 4.1 Seznam příkazů . |
| Březen 2013 | 8 | Upravena kapitola 4.2.2 Příkaz SET : |
| Srpen 2013 | 9 | Upravena kapitola 2.1 Příklad konfiguračního souboru a rozšířena tabulka 8 Podporované operační systémy . |
| Květen 2014 | 10 | Upravena kapitola 2.1 Příklad konfiguračního souboru . Rozšířena tabulka 4.1 Seznam příkazů a 5.1 Seznam zpráv . Přidána kapitola 4.2.8 Příkaz GETMEM : a aktualizována kapitola 4.2.13 Příkaz GETINFO : |
| Květen 2014 | 11 | Upravena kapitola 2.1 Příklad konfiguračního souboru . |
| Září 2014 | 12 | Upravena kapitola 2.1 Příklad konfiguračního souboru a kapitola 4.2.13 Příkaz GETINFO :. Rozšířena tabulka 4.1 Seznam příkazů a 5.1 Seznam zpráv . |
| Listopad 2014 | 13 | Upraveny kapitoly 1 Popis systému , 2.1 Příklad konfiguračního souboru , 3 Konfigurační public soubor , 3.2 Uživatelský public soubor , 4 Klientský protokol , 4.2.2 Příkaz SET :, 4.2.4 Příkaz EN :, 4.2.5 Příkaz DI : a 4.2.9 Příkaz GETFILE :. Upravena tabulka 4.1 Seznam příkazů a 5.1 Seznam zpráv . |
| Leden 2016 | 14 | Upraveny kapitoly 2.1 Příklad konfiguračního souboru , 3 Konfigurační public soubor a 4.2.13 Příkaz GETINFO :. Rozšířena tabulka 4.1 Seznam příkazů a 5.1 Seznam zpráv . |
| Únor 2016 | 15 | Upraveny kapitoly 4 Klientský protokol a 4.2.1 Příkaz LIST :. Rozšířena tabulka 4.1 Seznam příkazů . Přidány kapitoly 4.2.6 Příkaz HIDE : a 4.2.7 Příkaz UNHIDE :. |
| Říjen 2016 | 16 | Upraveny kapitoly 1 Popis systému , 4.2.13 Příkaz GETINFO : a 5.2 Příklady chybových zpráv . Přidány kapitoly 4.2.11 Příkaz WRITEFILE : a 4.2.12 Příkaz WRITEFILEINFO :. Rozšířena tabulka 4.1 Seznam příkazů a 5.1 Seznam zpráv . |

| Datum | Verze | Popis změn |
|---------------|--------------|--|
| Březen 2017 | 17 | Upraveny kapitoly 2.1 Příklad konfiguračního souboru a 4.2.11 Příkaz WRITEFILE:. |
| Duben 2018 | 18 | Upravena kapitola 2.1 Příklad konfiguračního souboru. |
| Červenec 2018 | 19 | Upravena kapitola 2.1 Příklad konfiguračního souboru a kapitola 4.2.3 Příkaz GET:. |
| Únor 2019 | 20 | Upraveny kapitoly 1 Popis systému , 4.2.1 Příkaz LIST: , 4.2.2 Příkaz SET: a 4.2.13 Příkaz GETINFO:. Opravena kapitola 2.1 Příklad konfiguračního souboru a tabulka 4.1 Seznam příkazů. |
| Říjen 2019 | 21 | Upraveny kapitoly 1 Popis systému a 2.1 Příklad konfiguračního souboru. |
| Duben 2020 | 22 | Upraveny kapitoly 4 Klientský protokol , 4.2.3 Příkaz GET: , 4.2.8 Příkaz GETMEM: , 4.2.9 Příkaz GETFILE: , 4.2.13 Příkaz GETINFO: , 5.1 Seznam zpráv , 5.2 Příklady chybových zpráv a 8 Podporované operační systémy. |

Obsah

| | | |
|----------|--|-----------|
| 1 | Popis systému..... | 5 |
| 2 | Konfigurační <i>ini</i> soubor..... | 6 |
| 2.1 | Příklad konfiguračního souboru..... | 6 |
| 3 | Konfigurační <i>public</i> soubor..... | 8 |
| 3.1 | Fixní <i>public</i> soubor..... | 8 |
| 3.1.1 | Seznam proměnných..... | 8 |
| 3.2 | Uživatelský <i>public</i> soubor..... | 8 |
| 4 | Klientský protokol..... | 9 |
| 4.1 | Seznam příkazů..... | 9 |
| 4.2 | Klientský protokol - příklady..... | 11 |
| 4.2.1 | Příkaz LIST:..... | 11 |
| 4.2.2 | Příkaz SET:..... | 11 |
| 4.2.3 | Příkaz GET:..... | 12 |
| 4.2.4 | Příkaz EN:..... | 13 |
| 4.2.5 | Příkaz DI:..... | 13 |
| 4.2.6 | Příkaz HIDE:..... | 14 |
| 4.2.7 | Příkaz UNHIDE:..... | 14 |
| 4.2.8 | Příkaz GETMEM:..... | 14 |
| 4.2.9 | Příkaz GETFILE:..... | 15 |
| 4.2.10 | Příkaz GETFILEINFO:..... | 15 |
| 4.2.10.1 | Seznam <i>atributů</i> | 15 |
| 4.2.11 | Příkaz WRITEFILE:..... | 16 |
| 4.2.12 | Příkaz WRITEFILEINFO:..... | 17 |
| 4.2.13 | Příkaz GETINFO:..... | 17 |
| 5 | Chybové zprávy..... | 18 |
| 5.1 | Seznam zpráv..... | 18 |
| 5.2 | Příklady chybových zpráv..... | 19 |
| 6 | Příklady použití..... | 20 |
| 6.1 | Příklad se simulovaným PLC v prostředí <i>Mosaic</i> | 20 |
| 6.2 | Příklad se skutečným PLC..... | 24 |
| 7 | Licence..... | 25 |
| 8 | Podporované operační systémy..... | 26 |

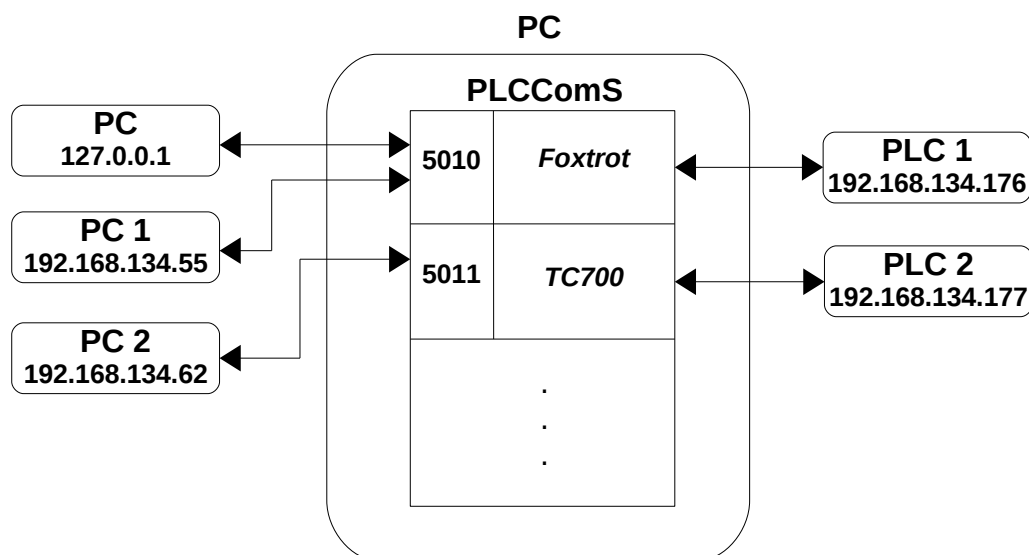
1 Popis systému

Komunikační server poskytuje TCP/IP spojení mezi klientským zařízením a PLC.

Komunikace serveru s klientem je řešena pomocí jednoduchého textově orientovaného protokolu typu DOTAZ/ODPOVĚĎ. Klient se tedy serveru dotazuje příkazy (Kap. 4.1) na hodnoty proměnných, jejichž jména jsou symbolická a jsou popsána v *public* souboru (Kap. 3). Je však možné použít i přímé adresování (Kap. 4.2.2). Server klientovi posílá pouze chybová hlášení a hodnoty proměnných jejichž hodnota se změnila. Tedy v případě, že jsou proměnné povoleny (Kap. 4.2.4 a 4.2.5).

Server s PLC komunikuje optimalizovaně pomocí protokolu EPSNET. Na hodnoty povolených proměnných se dotazuje absolutním adresováním a to periodicky v časovém rastru 100ms. Server umožňuje i nastavení nebo čtení zakázaných proměnných. To však není doporučeno, protože komunikace s těmito proměnnými není optimalizována a zatěžuje komunikační linku.

Konfigurace je řešena pomocí *ini* souboru.



Server je možné spustit s těmito parametry:

`PLCComS [-dtvh] [-c <konfigurační_soubor>] [-l <logovací_soubor>]`

| | |
|------------------------------|---|
| <code>-d, --daemon</code> | Server se spustí na pozadí. |
| <code>-t, --terminate</code> | Ukončit server, pokud byl konfigurační soubor změněn. |
| <code>-v, --verbose</code> | Upovídaný režim (Vícenásobné použití parametru <code>-v</code> zvýší upovídanost). |
| <code>-c, --config</code> | Konfigurační soubor. |
| <code>-l, --log</code> | Logovací soubor. |
| <code>-h, --help</code> | Zobrazí nápovědu. |

V případě spuštění s parametrem `-d` se server spustí na pozadí a veškerá hlášení se přesměrují do logovacího souboru. Není-li jeho jméno zadáno, nastaví se na „PLCComS.log“. Tato volba je dostupná pouze ve verzi pro operační systém Linux.

V případě, že nebyl zadán parametr `-c`, tak se jako jméno nastaví „PLCComS.ini“.

Není-li zadán parametr `-l`, tak jsou všechna hlášení serveru posílána do terminálu, ze kterého byl spuštěn.

2 Konfigurační *ini* soubor

Skládá se ze sekcí, kde sekce označená „[*]“ je povinná a obsahuje globální nastavení. Za touto sekcí následují ostatní, které popisují nastavení pro konkrétní PLC. Musí být minimálně jedna. Název takovéto sekce může být libovolný kromě „*“. Název se používá v logovacím souboru pro zřehlednění záznamů.

2.1 Příklad konfiguračního souboru

```
[*]

COMM_LOOP_DELAY    = 100    # Prodleva v hlavní smyčce ([1 - 1000]ms)
COMM_SCHED_PRI     = 10     # Real-time priorita plánovače (0 (zakázán),
                             1 (nízká) - 99 (vysoká))
NET_CONNECT_MAX    = 128    # Maximální počet klientských spojení (Maximum je
                             1024)
MEM_BLKSIZE        = 4096   # Velikost bloku v bytech pro přenos PLC paměti
                             (Maximum je 65536)
FFILE_BLKSIZE      = 16384  # Velikost bloku v bytech pro přenos souborů
                             # (Maximum je 65536)
FFILE_TIMEOUT      = 300    # Časový limit platnosti souboru (Maximum je 3600s)
FFILE_TIMEOUT_WR   = 5      # Časový limit mezi vysílanými datovými bloky
                             # příkazu WRITEFILE: (Maximum je 60s)
FFILE_MAXRECS      = 256    # Počet souborů uložených v paměti (Maximum je 1024)
END_LINE_CRLF      = Yes    # Koncový znak řádku (Yes = DOS [\r\n], No = UNIX
                             [\n])
PF_VAR_DISABLED    = Yes    # Výchozí stav proměnných
DIFF_VAR_ENABLED   = Yes    # Zakáže nebo povolí výpisy DIFF: zpráv
SYNC_VAR_ENABLED   = Yes    # Zakáže nebo povolí synchronizaci proměnných s PLC
                             během stahování souborů.
DATE_TIME_DECIMAL  = No     # Zobrazí nebo čte DATE and TIME jako desetinné
                             číslo
LIM_OF_DECIMALS    = No     # Omezí přesnost desetinných čísel pro generování
                             DIFF: zpráv (Yes = omezit)
NUM_OF_DECIMALS    = 10     # Počet desetinných číslic, která budou zobrazena
                             nebo omezena([0 - 6] REAL, [0 - 15] LREAL)
SCIENT_NOTATION    = No     # Vědecké zobrazení pro REAL a LREAL (Yes =
                             vědecké [-]d.ddd e[+/-]ddd, No = normální
                             [-]ddd.ddd)

#[Symbolický název PLC]

#IPADDR            = IP adresa PLC
#IPADDR_LOCAL      = IP adresa síťového rozhraní, ne kterém bude server s klientem
                             komunikovat (0.0.0.0 - všechny rozhraní).
#EPSADDR           = EPSNET zdrojová adresa (1 - 126). Není-li proměnná
                             nastavena, bude adresa vygenerována automaticky.
#EPSTOUT           = EPSNET timeout ([1 - 15000]ms). Není-li proměnná
                             nastavena, bude použita výchozí hodnota 1000ms.
#EPSTOUT_STEP      = EPSNET timeout přírůstek ([0 - 5000]ms). Není-li proměnná
                             nastavena, bude použita výchozí hodnota 500ms.
#EPSPORT           = EPSNET komunikační port. Užitečné pro službu TECOROUTE. Není-
                             li proměnná nastavena, bude použita výchozí hodnota 61682.
#SERVER_PORT       = Číslo portu, na kterém bude klient s PLC komunikovat.
#SERIAL_DEVICE     = Jméno sériového zařízení
#SERIAL_SPEED      = Komunikační rychlost sériové linky
#SERIAL_RTS        = Číslo GPIO pinu pro řízení datového toku. Není-li proměnná
                             nastavena, použije se hardwarové řízení.
```

#SHM_NAME = Název sdíleného modulu (knihovny *.dll or *.so)
#SHM_SOCKET = Číslo soketu pro komunikaci se sdíleným modulem
#SHM_PORT = Číslo portu pro komunikaci se sdíleným modulem
#PUBFILE_CRC = Zapne nebo vypne kontrolu CRC *public* souboru. [Yes/No]
#PUBFILE_WRITE = Zapne nebo vypne ukládání *public* souboru na lokální disk.
[Yes/No]
#PUBFILE = Jméno *public* souboru.
#PUBFILE_FIXED = Jméno fixního *public* souboru.

[Foxtrot]

IPADDR = 192.168.134.176
SERVER_PORT = 5010
PUBFILE_CRC = Yes
PUBFILE_WRITE = No
PUBFILE_FIXED = FIXED_Foxtrot.pub
PUBFILE = //www/webmaker.pub

[TC700]

IPADDR = 192.168.134.177
SERVER_PORT = 5011
PUBFILE_CRC = Yes
PUBFILE_WRITE = No
PUBFILE_FIXED = FIXED_TC700.pub
PUBFILE = //www/webmaker.pub

[TC700 RS232]

SERIAL_DEVICE = /dev/ttyS0
SERIAL_SPEED = 115200
SERVER_PORT = 5012
PUBFILE_CRC = Yes
PUBFILE_WRITE = No
PUBFILE_FIXED = FIXED_TC700.pub
PUBFILE = //www/webmaker.PUB

[SoftPLC]

SERVER_PORT = 5013
SHM_NAME = ShmSrv.dll
SHM_SOCKET = 0
SHM_PORT = 5
PUBFILE_CRC = Yes
PUBFILE_WRITE = No
PUBFILE_FIXED = FIXED_SoftPLC.pub
PUBFILE = //www/webmaker.pub

3 Konfigurační *public* soubor

Popisuje vzájemnou vazbu mezi názvem proměnné a fyzickou adresou registru v PLC. Používají se dva typy souborů, fixní a uživatelský. Fixní soubor je uložen v adresáři, ze kterého je server spuštěn. Uživatelský soubor může být uložen buď lokálně nebo v PLC. V případě lokálního uložení souboru je kořenový adresář závislý na nastavení proměnné `PUBFILE_WRITE`. Je-li tato proměnná nastavena na „No“ nebo v *ini* souboru chybí, pak je kořenovým adresářem adresář, ze kterého je server spuštěn. Je-li tato proměnná nastavena na „Yes“, pak je kořenovým adresářem podadresář, jehož jméno je schodné se jménem sekce v *ini* souboru a je umístěn v adresáři, ze kterého byl server spuštěn. Zadáním relativní cesty je možné přistupovat k lokálním souborům ležícím mimo tento kořenový adresář. Při otevření souboru má vždy přednost ten soubor, který je uložen lokálně. Není-li nalezen, zkusí se otevřít soubor na PLC. Začíná-li název souboru „/“ myslí se tím kořenový adresář.

3.1 Fixní *public* soubor

Je soubor potřebný pro vlastní chod serveru. Obsahuje proměnné, se kterými je možné pracovat stejným způsobem jako s proměnnými z uživatelského *public* souboru. S výjimkou příkazu „DI:“, který se pro tyto proměnné ignoruje.

3.1.1 Seznam proměnných

| Jméno | Datový typ | Význam |
|-----------|------------|---|
| __PLC_RUN | BOOL | Stav PLC. (1 = RUN, 0 = HALT) |
| __PF_CRC | DWORD | CRC hodnota uživatelského <i>public</i> souboru. Při její změně server automaticky načte nový <i>public</i> soubor. |

3.2 Uživatelský *public* soubor

Je soubor generovaný pomocí vývojového prostředí *Mosaic*. Proměnné z tohoto souboru server připojí k proměnným ze souboru *fixního*. V případě kolize jmen jsou proměnné z tohoto souboru ignorovány.

Soubor je opatřen hodnotou kontrolního součtu CRC, která je shodná s hodnotou proměnné `__PF_CRC`. Změna hodnoty této proměnné znamená změnu projektu v PLC. Server tedy tuto změnu hlídá a v případě, že k ní dojde, nahraje soubor nový a ohlásí tuto změnu klientovi zprávou „WARNING:250 Changed public file: 'Test.pub'“. Pokud se obsah nového souboru od aktuálního neliší, tak se nový soubor nestahuje a klientovi je to ohlášeno zprávou „WARNING:251 Public file not changed: 'Test.pub'“.

4 Klientský protokol

Textově orientovaný protokol, kde je každý příkaz ukončen znakem „:“ a každý konec řádku buď „\r\n“ (DOS) nebo „\n“ (UNIX). Typ zakončení řádku je volitelný z konfiguračního souboru nebo pomocí příkazu „*SETCONF*:“. Nastavení proměnných *crlf* a *diff* pomocí příkazu „*SETCONF*:“ má vliv pouze pro klienta, který nastavení provedl. Ostatní proměnné jako jsou *ipaddr*, *epsaddr*, ... ovlivní všechny připojené klienty k danému PLC. U jmen příkazů se nerozlišují malá a velká písmena a je tedy možná i jejich libovolná kombinace. Pro test spojení či diagnostiku je možné použít program *Telnet*, kde je možné pro ukončení spojení použít escape sekvenci „ctrl+d“.

4.1 Seznam příkazů

| Příkaz | Popis |
|--|--|
| <i>LIST</i> : <i>SET</i> :<jméno_proměnné,hodnota> <i>GET</i> :<jméno_proměnné> <i>EN</i> :<jméno_proměnné> [delta] <i>DI</i> :<jméno_proměnné> [delta] <i>HIDE</i> :<jméno_proměnné> <i>UNHIDE</i> :<jméno_proměnné> <i>GETMEM</i> :<jméno_proměnné velikost> <i>GETFILE</i> :<jméno_souboru> <i>GETFILEINFO</i> :<jméno_souboru> <i>WRITEFILE</i> :<jméno_souboru>[<velikost bloku>]=data <i>WRITEFILEINFO</i> :<jméno_souboru>[<velikost bloku>]=data <i>GETINFO</i> :<jméno> <i>version</i> <i>version_epsnet</i> <i>version_ini</i> <i>version_plc</i> <i>ipaddr</i> <i>epsaddr</i> <i>epsport</i> <i>serial_device</i> <i>pubfile</i> | Vypíše seznam všech proměnných z <i>public</i> souborů. Nastaví proměnnou v PLC na zadanou hodnotu. Získá hodnotu proměnné z PLC. Povolí proměnnou(né) z <i>public</i> souboru a nastaví jí změnu <i>delta</i> . Zakáže proměnnou(né) z <i>public</i> souboru a nastaví jí změnu <i>delta</i> . Skryje proměnnou(né) z <i>public</i> souboru. Odkryje proměnnou(né) z <i>public</i> souboru. Získá blok paměti z PLC. Získá soubor z PLC nebo PC. Získá informace o souboru uloženém v PLC nebo PC. Zapíše soubor do PLC nebo PC. Zapíše informace o souboru do PLC nebo PC. Vypíše informace o komunikačním serveru. Verze komunikačního serveru. Verze epsnetové knihovny. Verze parsovací <i>ini</i> knihovny. Verze PLC. IP adresa PLC. EPSNET zdrojová adresa. EPSNET komunikační port. Parametry sériové linky. Jména aktuálně použitých <i>public</i> souborů. |

| | |
|--|--|
| <i>network</i> | Seznam připojených klientů. |
| <i>SETCONF: <jméno_proměnné,hodnota>\n</i> | Změní hodnotu proměnné uvedené v konfiguračním <i>ini</i> souboru. |
| <i>jméno_proměnné:</i> | |
| <i>ipaddr</i> | IP adresa PLC. |
| <i>epsaddr</i> | EPSNET zdrojová adresa (1 - 126). |
| <i>epsport</i> | EPSNET komunikační port. |
| <i>serial_device</i> | Jméno sériové linky. |
| <i>serial_speed</i> | Komunikační rychlost sériové linky. |
| <i>pubfile</i> | Public soubor. |
| <i>crlf</i> | Koncový znak řádku (yes = DOS, no = UNIX). |
| <i>diff</i> | Potlačení výpisu „ <i>DIFF</i> :“ zpráv (yes / no). |
| <i>HELP:\n</i> | Vypíše nápovědu. |

Při zadávání jmen proměnných je možné použít hvězdičkovou konvenci. Znak „*“ ve jménu nahradí libovolný počet znaků. Počet znaků „*“ není omezen.

Za jménem příkazu je možné uvést i více proměnných jako zjednodušený zápis. Každá proměnná pak musí být oddělena znakem konce řádku (*Kap. 4*).

Odpověď serveru vždy začíná jménem příkazu, který ji vyvolal. Kromě příkazu „*SET*:“, u kterého odpověď začíná „*DIFF*:“. Tedy v případě, že tento výpis nebyl zakázán (*Kap. 2, 4*). V případě příkazu, kde v odpovědi může být výsledků více, je seznam odpovědí serveru ukončen samotným jménem příkazu.

Příklad výpisu proměnných jejichž jméno začíná řetězcem „test_“:

Dotaz:

*GET:test_**\n

Odpověď:

GET:test_1,123\n

GET:test_2,1.234500\n

GET:\n

Příklad dotazu na hodnotu konkrétní proměnné:

Dotaz:

GET:test_1\n

Odpověď:

GET:test_1,123\n

Příklad zjednodušeného dotazu na hodnoty konkrétních proměnných:

Dotaz:

GET:test_1\ntest_2\n

Odpověď:

GET:test_1,123\n

GET:test_2,1.234500\n

4.2 Klientský protokol - příklady

4.2.1 Příkaz LIST:

Dotaz:

LIST:\n

Odpověď:

LIST:test_1\n

LIST:test_2\n

LIST:test_string~\n

LIST:test_btn\n*

LIST:test_btn_up~\n*

LIST:test_btn_down~\n*

LIST:\n

Znak „*“, na konci jména proměnné, označuje zakázanou proměnnou (Kap. 4.2.5). Znak „~“ pak označuje skrytou proměnnou (Kap. 4.2.6).

4.2.2 Příkaz SET:

Dotaz:

SET:test_1,123\n

Odpověď:

DIFF:test_1,123\n

Dotaz:

SET:test_string,"Ahoj!"\n

Odpověď:

DIFF:test_string,"Ahoj!"\n

Odpověď začínající řetězcem „DIFF:“ je reakcí serveru na změnu některé proměnné v PLC. V kombinaci s příkazem „SET:“ ji lze použít jako kontrolu, že k nastavení proměnné skutečně došlo. Odpověď „DIFF:“ není generována v případě, že u proměnné nedošlo ke změně nebo změna nebyla větší než je hodnota odchylky *delta*. Tedy v případě, že u proměnné byla nastavena (Kap. 4.2.4, 4.2.5). Nebo pokud je proměnná zakázána (Kap. 4.2.4 a 4.2.5).

Hodnoty proměnných, které jsou typu *string* se vkládají do uvozovek. Hodnoty proměnných, které jsou typu *bool*, je možné nastavit pomocí výrazu *true* (1) nebo *false* (0). Odpovědí je vždy hodnota 1 nebo 0.

Pokud se místo jména proměnné použije přímá adresa, je formát následující: %<jméno registru><adresa registru>

Např.: %R100, %S6, ...

4.2.3 Příkaz GET:

Dotaz:

GET:test_1\n

Odpověď:

GET:test_1,123\n

Dotaz:

GET:test_string\n

Odpověď:

GET:test_string,"Ahoj!"\n

V případě dotazu na proměnnou typu *date* nebo *dt* se hodnota posílá jako 64-bitové celé neznaménkové číslo nebo jako číslo reálné jak definuje norma. Změnu formátu lze provést nastavením proměnné *DATE_TIME_DECIMAL* (Kap. 2.1).

Příklad v jazyce „C“:

```
time_t t;
struct tm *tm;
unsigned int ms;
unsigned long long int ldate;
double date;

...

sscanf (msg, "%llu", &ldate);

date = *(double *) &ldate;

ms = (unsigned int) ((date - (unsigned long long int) date) * 1000);

t = (time_t) date;
tm = gmtime (&t);

if (tm == NULL)
    return 1;

printf ("Date: %02d.%02d.%d", tm->tm_mday, tm->tm_mon + 1, tm->tm_year +
1900);
printf ("Time: %02d:%02d:%02d.%03d", tm->tm_hour, tm->tm_min, tm->tm_sec, ms);

...
```

Je-li proměnná typu *time* nebo *tod* posílá se hodnota jako 32-bitové neznaménkové celé číslo jak definuje norma.

Příklad v jazyce „C“:

```
struct tm tm;
unsigned long time;

...

sscanf (msg, "%lu", &time);

memset (&tm, 0, sizeof (struct tm));

tm.tm_hour = time / 3600000;
time -= (tm.tm_hour * 3600000);
tm.tm_min = time / 60000;
time -= (tm.tm_min * 60000);
tm.tm_sec = time / 1000;
tm.tm_isdst = time - (tm.tm_sec * 1000);

printf ("Time: %02d:%02d:%02d", tm.tm_hour, tm.tm_min, tm.tm_sec);

...
```

4.2.4 Příkaz EN:

Dotaz:

EN:\n

EN:teplota 0.5\n

4.2.5 Příkaz DI:

Dotaz:,

DI:\n

DI:osvetleni 10\n

Příkazy „EN:“ a „DI:“ povolují nebo zakazují proměnné v komunikační tabulce serveru s PLC. Jsou-li proměnné zakázány, server se na ně PLC neptá. Tím klesne objem přenášených dat mezi serverem a PLC. Zakázání nemá vliv na proměnné ve fixním *public* souboru. Za jménem proměnné může být zadána odchylka *delta*. Tato odchylka slouží k omezení počtu „DIFF:“ zpráv.

4.2.6 Příkaz HIDE:

Dotaz:

```
HIDE:teplota\n
HIDE:svetlo*\n
```

4.2.7 Příkaz UNHIDE:

Dotaz:

```
UNHIDE:teplota\n
UNHIDE:svetlo*\n
```

Příkazy „*HIDE*:“ a „*UNHIDE*:“ umožňují skrýt nebo odskrýt proměnné v komunikační tabulce serveru s PLC. Jsou-li proměnné skryty, znamená to, že jsou ignorovány příkazy, které pracují se všemi proměnnými komunikační tabulky. Například při zadání samotného znaku „*“ jako parametru v příkazech „*EN*:“, „*DI*:“ nebo „*GET*:“.

Skrytí proměnné neomezuje zasílání „*DIFF*:“ zpráv.

4.2.8 Příkaz GETMEM:

Dotaz:

```
GETMEM:test_string 5\n
```

Odpověď:

```
GETMEM:test_string[5]=Ahoj!\n
GETMEM:test_string[0]=\n
```

Obsah paměti se přenáší v blocích o velikosti dané proměnnou MEM_BLKSIZE (Kap. 2.1). Skutečná délka bloku je uvedena v hranatých závorkách na konci jména proměnné.

4.2.9 Příkaz GETFILE:

Dotaz:

GETFILE://www/TEST.TXT\n

Odpověď:

GETFILE://www/TEST.TXT[21]=Toto je pokusny text.\n

GETFILE://www/TEST.TXT[0]=\n

Obsah souboru se přenáší v blocích o velikosti dané proměnnou *FFILE_BLKSIZE* (Kap. 2.1). Skutečná délka bloku je uvedena v hranatých závorkách na konci jména souboru. Pokud končí jméno souboru znakem „/“, jedná se o adresář a přenáší se jeho obsah.

4.2.10 Příkaz GETFILEINFO:

Dotaz:

GETFILEINFO://www/TEST.TXT\n

Odpověď:

GETFILEINFO://www/TEST.TXT[35]=21 32 59391128503405 59391128503405\n

Délka odpovědi je uvedena v hranatých závorkách na konci jména souboru.

Struktura zprávy:

velikost atributy čas_vytvoření čas_změny

velikost - Velikost souboru v bytech.

atributy - 32 bitové číslo, které je interpretováno bitově.

čas_vytvoření - 64 bitové číslo, které je možné interpretovat pomocí struktury *ttida*.

čas_změny - 64 bitové číslo, které je možné interpretovat pomocí struktury *ttida*.

4.2.10.1 Seznam atributů

| Hexadecimální hodnota | Význam |
|-----------------------|-------------------|
| 0x00000001 | Pouze pro čtení |
| 0x00000002 | Skrytý soubor |
| 0x00000004 | Systémový soubor |
| 0x00000010 | Soubor je adresář |
| 0x00000020 | Archivní soubor |

struktura *ttida* - Velikost 8 bytu (rok, měsíc, den, hodina, minuta, sekunda, nic, nic).

Příklad v jazyce „C“:

```
#define FILE_ATTRIBUTE_READONLY  0x00000001
#define FILE_ATTRIBUTE_HIDDEN    0x00000002
#define FILE_ATTRIBUTE_SYSTEM    0x00000004
#define FILE_ATTRIBUTE_DIRECTORY 0x00000010
#define FILE_ATTRIBUTE_ARCHIVE   0x00000020

struct ttida {char year; char mon; char day; char hour; char min; char sec;
              char none1; char none2;} tida_c, tida_m;

...

long size, attr;

sscanf (msg, "%ld %ld %llu %llu", &size, &attr,
        (unsigned long long int *)&tida_c, (unsigned long long int *)&tida_m);

printf ("File size   : %ld bytes\n", size);

printf ("Attributes : %c%c%c%c%c\n", (attr&FILE_ATTRIBUTE_READONLY) ? 'R' : 'W',
        (attr&FILE_ATTRIBUTE_HIDDEN)   ? 'H' : '-',
        (attr&FILE_ATTRIBUTE_SYSTEM)   ? 'S' : '-',
        (attr&FILE_ATTRIBUTE_DIRECTORY)? 'D' : '-',
        (attr&FILE_ATTRIBUTE_ARCHIVE)  ? 'A' : '-');

printf ("Create time : %02d.%02d.%04d %02d:%02d:%02d\n", tida_c.day,
        tida_c.mon, tida_c.year + 1900, tida_c.hour, tida_c.min, tida_c.sec);

printf ("Modify time : %02d.%02d.%04d %02d:%02d:%02d\n", tida_m.day,
        tida_m.mon, tida_m.year + 1900, tida_m.hour, tida_m.min, tida_m.sec);

...
```

4.2.11 Příkaz WRITEFILE:

Dotaz:

WRITEFILE:test.txt[23]=Toto je pokusny text 1.\n

WRITEFILE:test.txt[23]=Toto je pokusny text 2.\n

WRITEFILE:test.txt[0]=\n

Odpověď:

WRITEFILE:test.txt OK\n

Obsah souboru se nejprve přenáší na server. Teprve poté, co se přenesení blok o nulové velikosti, je soubor zapsán do PLC nebo PC. Mezi jednotlivými bloky nesmí být prodleva větší než hodnota v proměnné FFILE_TIMEOUT_WR (Kap. 2.1), jinak budou přijatá data zahozena. Maximální doporučená velikosti bloku je 4096. Pokud jméno souboru začíná znakem „.“ (Např.: ./test.txt), bude soubor zapsán do PC.

4.2.12 Příkaz WRITEFILEINFO:

Dotaz:

WRITEFILEINFO:test.txt[34]=46 0 41854524393844 41854674928753\n

Odpověď:

WRITEFILEINFO:test.txt OK\n

Délka zprávy je uvedena v hranatých závorkách na konci jména souboru. Struktura zprávy je popsána v *Kap. 4.2.10*.

4.2.13 Příkaz GETINFO:

Dotaz:

GETINFO:\n

Odpověď:

GETINFO:VERSION,Ver 5.4 Apr 30 2020 09:14:05\n

GETINFO:VERSION_EPSNET,Ver 3.3 Apr 30 2020 09:14:03\n

GETINFO:VERSION_INI,Ver 3.2 Apr 30 2020 09:14:08\n

GETINFO:VERSION_PLC,CP2080I B 2.7 2.0\n

GETINFO:IPADDR,192.168.134.176\n

GETINFO:EPSADDR,1\n

GETINFO:EPSPORT,61682\n

GETINFO:PUBFILE,2/11 [FIXED_Foxtrot.pub,iFoxtrot.pub]\n

GETINFO:NETWORK,1/128 [127.0.0.1 2020-04-30 10:42:46]\n

GETINFO:NETWORK,2/128 [127.0.0.1 2020-04-30 10:45:10]\n

GETINFO:NETWORK,3/128 [127.0.0.1 2020-04-30 11:09:33]\n

GETINFO:\n

Dotaz:

GETINFO:version_plc\n

Odpověď:

GETINFO:VERSION_PLC,CP2080I B 2.7 2.0\n

5 Chybové zprávy

Server posílá dva typy zpráv. Chybové, které začínají řetězcem „*ERROR*:“ a výstražné, které začínají řetězcem „*WARNING*:“. Zprávy se dále dělí do skupin a to vždy po deseti.

5.1 Seznam zpráv

| Kód chyby | Text zprávy | Skupina | Typ |
|-----------|---------------------------------------|----------------------------|----------|
| 10 | Unable to connect to PLC. | Sít'ová komunikace | Chyba |
| 11 | Maximum connections reached. | Sít'ová komunikace | Chyba |
| 20 | Unable to get data from PLC. | Komunikace s PLC | Chyba |
| 21 | Unable to send data to PLC. | Komunikace s PLC | Chyba |
| 30 | Bad client request: | Klientské dotazy | Chyba |
| 31 | Incomplete client request. | Klientské dotazy | Chyba |
| 32 | Unknown command name: | Klientské dotazy | Chyba |
| 33 | Unknown register name: | Klientské dotazy | Chyba |
| 34 | Disabled register name: | Klientské dotazy | Chyba |
| 35 | Wrong parameter value: | Klientské dotazy | Chyba |
| 40 | Unable to get information about file: | Souborové operace | Chyba |
| 41 | Unable to get file: | Souborové operace | Chyba |
| 42 | Unable to write file: | Souborové operace | Chyba |
| 43 | Unable to write file information: | Souborové operace | Chyba |
| 44 | Unterminated write file data: | Souborové operace | Chyba |
| 45 | Timeout write file data: | Souborové operace | Chyba |
| 46 | Aborted get file: | Souborové operace | Chyba |
| 50 | Unknown name: | Příkaz „ <i>SETCONF</i> :“ | Chyba |
| 60 | Unknown name: | Příkaz „ <i>GETINFO</i> :“ | Chyba |
| 70 | Unable to connect to SHARED module. | Sdílený modul | Chyba |
| 80 | Unable to get data from address: | Příkaz „ <i>GETMEM</i> :“ | Chyba |
| 81 | Unable to set data in to address: | Příkaz „ <i>GETMEM</i> :“ | Chyba |
| 250 | Changed public file: | Příkaz „ <i>SETCONF</i> :“ | Varování |
| 251 | Public file not changed: | Příkaz „ <i>SETCONF</i> :“ | Varování |
| 1024 | Unknown error. | Nespecifikováno | Chyba |

5.2 Příklady chybových zpráv

Dotaz:

GET:test\n

Odpověď:

ERROR:33 Unknown register name: 'test'\n

Dotaz:

GET:%R70000\n

Odpověď:

ERROR:80 Unable to get data from address: '%R700000'\n

Dotaz:

EN:test_1 a\n

Odpověď:

ERROR:35 Wrong parameter value: 'a'\n

Dotaz:

SETCONF:pubfile,Test.pub\n

Odpověď:

WARNING:250 Changed public file: 'Test.pub'\n

Nebo v případě, že se soubory neliší:

WARNING:251 Public file not changed: 'Test.pub'\n

Dotaz:

WRITEFILE:test.txt[10]=Test\n

Odpověď:

ERROR:44 Unterminated write file data: ' test.txt'\n

Nebo v případě, že další blok dat nebude zapsán do doby nastavené proměnnou
FFILE_TIMEOUT_WR (Kap. 2.1):

ERROR:45 Timeout write file data: ' test.txt'\n

6 Příklady použití

Použití aplikace komunikačního serveru *PLCComS* lze demonstrovat na několika jednoduchých ukázkových příkladech.

6.1 Příklad se simulovaným PLC v prostředí *Mosaic*

Mějme jednoduchý program psaný v jazyce strukturovaného textu „ST“, který bude vyčítat informace o systémovém čase a datu a kupříkladu bude počítat průběžnou hodnotu funkce *sinus* a *cosinus* v závislosti na době běhu programu. Dále budeme mít proměnnou *positive* signalizující kladné půlplny funkce *sinus* a jednu proměnnou typu *STRING* obsahující textový řetězec.

Jako testovací PLC vybereme například *FOXTROT CP-1016*.

Příklad v jazyce „ST“:

```
VAR_GLOBAL // Globalni public promenne
  datum    {PUBLIC} : DATE;
  cas       {PUBLIC} : TIME;
  sinus     {PUBLIC} : LREAL;
  cosinus   {PUBLIC} : LREAL;
  c         {PUBLIC} : LREAL;
  positive  {PUBLIC} : BOOL;
  text      {PUBLIC} : STRING;
END_VAR

VAR_GLOBAL CONSTANT
  PI : LREAL := 3.14159265358979323846;
END_VAR

PROGRAM prgMain
  VAR
  END_VAR
  VAR_TEMP
  END_VAR

  datum := GetDate(); // Vrati aktualni datum
  cas    := GetTime(); // Vrati aktualni cas

  c := c + 0.001;      // Casovy prubeh pro vypocet sinus a cosinus

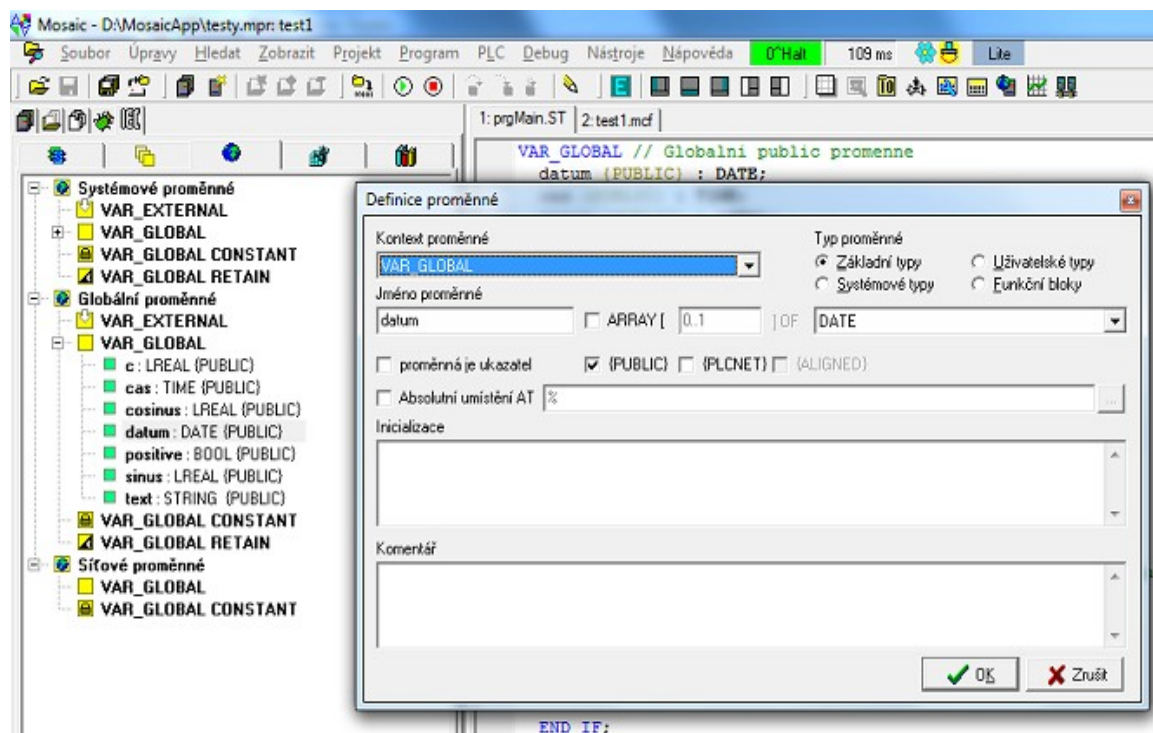
  sinus  := SIN(c);    // Vrati sinus hodnoty argumentu
  cosinus := COS(c);   // Vrati cosinus hodnoty argumentu

  IF sinus > 0.0 THEN
    positive := TRUE;
  ELSE
    positive := FALSE;
  END_IF;

  text := 'Have a nice day';

END_PROGRAM
```

Server umožňuje sledování pouze těch systémových proměnných, které jsou definované jako veřejné (PUBLIC). Tedy jako proměnné, ke kterým lze přistupovat zvenčí programu. Veřejnou proměnnou lze definovat přímo direktivou „{PUBLIC}” nebo pomocí dialogového okna „Definice proměnné“, kde je nutné zaškrtnout stejnojmennou volbu (Obr. 6.1.1).



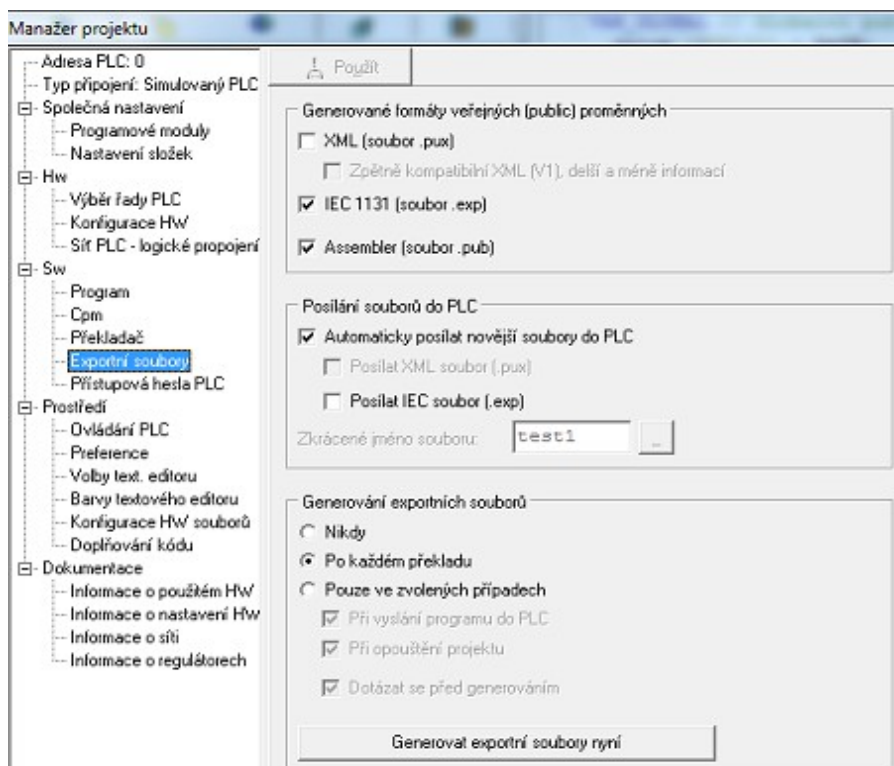
Obr. 6.1.1: Dialogové okno „Definice proměnné“

Pokud již máme definovány všechny veřejné proměnné, které chceme sledovat, je nutné vygenerovat *public* soubor (*.pub), tedy soubor obsahující informace o deklarovaných veřejných proměnných. Tento soubor nemusí být prostředím Mosaic automaticky generován. V tomto případě je nutné generování souboru nastavit v „Manažeru projektu“ ve větvi nastavení „Sw – Exportní soubory“, kde je nutné zaškrtnout volbu „IEC 1131“ (soubor .exp) a volbu „Assembler“ (soubor .pub) (Obr. 6.1.2).

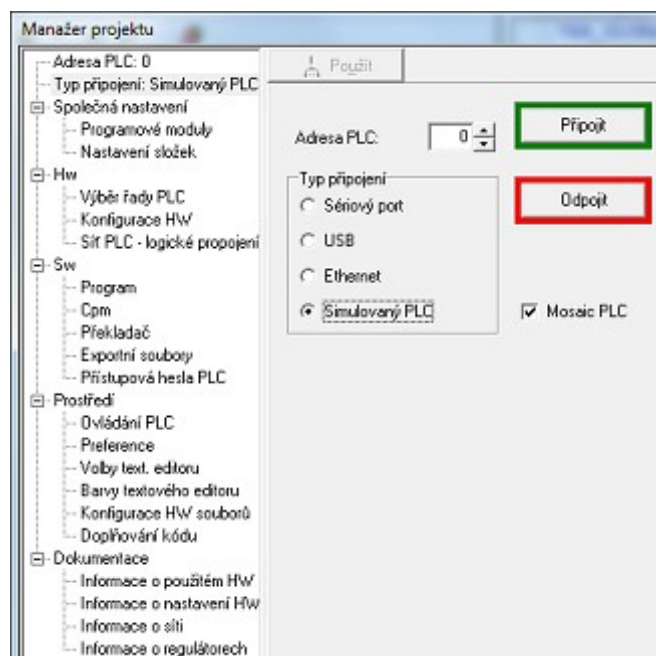
Vygenerovaný *public* soubor je po překladu programu uložen v projektovém adresáři pod jménem „název_projektu.pub“. Název tohoto souboru se pak použije jako konfigurační parametr *ini* souboru pro aplikaci serveru *PLCComS*.

V případě, že jméno souboru neobsahuje relativní cestu do projektového adresáře, je třeba, aby byl *public* soubor nakopírován do stejného adresáře odkud je server spouštěn!

Hotový testovací program uložíme. V „Manažeru projektu“ je nutné ve větvi „Typ připojení“ zvolit (pokud je PLC připojeno, musí se odpojit tlačítkem „Odpojit“) „Simulovaný PLC“ se zaškrtnutou volbou „Mosaic PLC“ a připojit toto PLC tlačítkem „Připojit“ (Obr. 6.1.3). Provedeme překlad a spustíme v režimu simulace.



Obr. 6.1.2: Nastavení generování exportních souborů



Obr. 6.1.3: Připojení simulovaného PLC

Server spustíme s parametry `-c` a `-l`. Bude tedy nutné zadat jméno konfiguračního (*.ini) a logovacího (*.log) souboru. V případě existence logovacího souboru jsou hlášení serveru připojeny na jeho konec. Konfigurační soubor je nutné nadefinovat. Pro námi zvolený příklad, kdy sledovaný PLC automat je simulován prostředím *Mosaic* na lokální stanici (*localhost* – *127.0.0.1*) bude obsah konfiguračního souboru vypadat následovně:

Obsah konfiguračního souboru:

```
# Configuration file for communication server

[*]

NET_CONNECT_MAX = 10          # Maximalni počet klientu (Max 32)
FFILE_BLKSIZE   = 1024        # Maximalni velikost bloku pro soubory (Max 65536)
FFILE_TIMEOUT   = 300         # Cas v sekundach pro dobu drzeni souboru
                                v pameti (Max 3600)
FFILE_MAXRECS   = 256         # pocet souboru ukladanych do pameti (Max 1024)
END_LINE_CRLF   = Yes         # Koncovy znak (Yes = DOS [\r\n], No = UNIX [\n])
PF_VAR_DISABLED = Yes         # Vychazi status pro promenne

[FOXTROT]                  # Alias pouziteho PLC

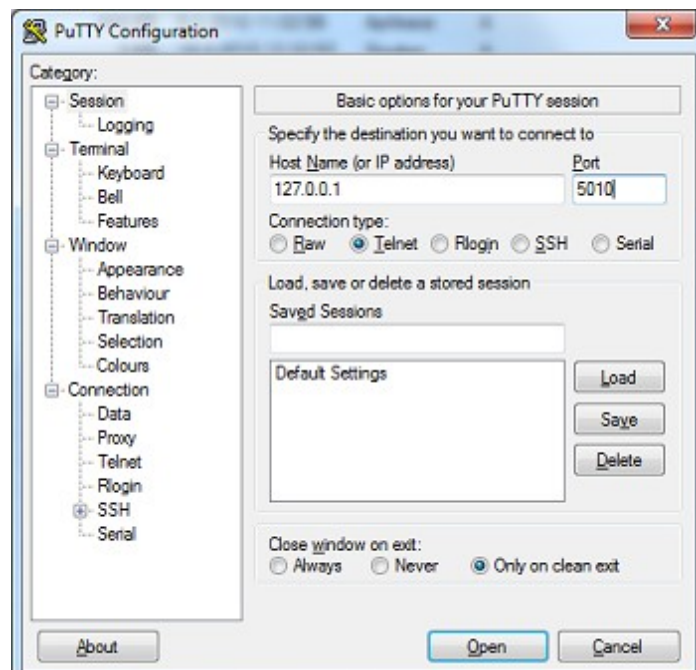
IPADDR          = 127.0.0.1    # IP adresa PLC
SERVER_PORT     = 5010         # Port serveru
PUBFILE_CRC     = No           # Neoverovat kontrolni soucet pro PUBFILE

PUBFILE_FIXED   = <název_fixedpubfile.pub> # Fixni PUBFILE
PUBFILE         = <název_pubfile.pub>      # PUBFILE
```

Máme-li tedy připraven konfigurační soubor, můžeme spustit server z příkazové řádky příkazem:
..>PLCComS.exe -c <jmeno_konfig> -l <jmeno_log>

Úspěšnost spuštění serveru můžeme sledovat v logovacím souboru nebo v případě vynechání volby *-l* v terminálovém okně.

Se spuštěným serverem *PLCComS* lze komunikovat programem *Telnet*. K tomuto účelu lze použít široce rozšířenou aplikaci *Putty*, pomocí které se lze k serveru pohodlně připojit (Obr. 6.1.4).



Obr. 6.1.4: Připojení k serveru *PLCComS* pomocí *Putty*

V programu *Putty* se pouze nastaví IP adresa připojovaného serveru (pokud server *PLCComS* běží na lokální stanici, tak se jedná o adresu *127.0.0.1*) a číslo portu, na kterém server poslouchá (v našem případě je to port *5010*). Stisknutím tlačítka „*Open*“ proběhne připojení.

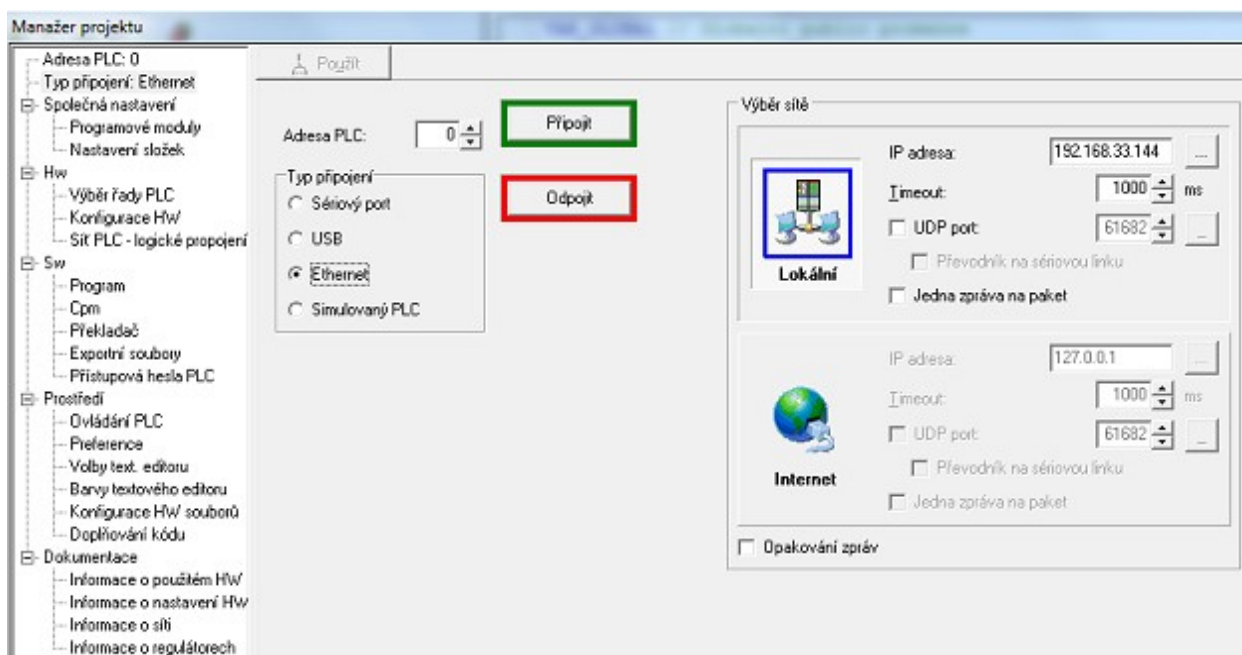
Nyní je již možné sledovat všechny veřejné proměnné, které jsme si v našem virtuálním PLC definovali. K těmto účelům je nutné používat příkazy definované aplikací *PLCComS* (například příkaz „*LIST*“ vrátí seznam všech sledovaných proměnných v PLC).

Činnost serveru lze ukončit escape sekvencí „*ctrl+c*“. Všechny důležité události z činnosti serveru jsou zapsány v logovacím souboru.

6.2 Příklad se skutečným PLC

Použijeme stejný program jako v předchozím případě. Pouze místo simulovaného běhu nyní použijeme skutečný automat, např. *FOXTROT CP-1004*.

Postup nastavení prostředí *Mosaic* (generování exportních souborů *PUBFILE*) a samotného programu (definice veřejných *PUBLIC* proměnných) je totožný s prvním příkladem. Jediné, v čem se postup liší, je typ připojení v „*Manažeru projektu*“, kde se již nezvolí volba „*Simulovaný PLC*“, ale nyní použijeme skutečné PLC zařízení umístěné na lokální síti ETHERNET definované svou IP adresou (Obr. 6.2.1). Zařízení připojíme tlačítkem „*Připojit*“, provedeme překlad a program uvedeme do režimu „*RUN*“.



Obr. 6.2.1: Připojení vzdáleného PLC

Konfigurační soubor aplikace *PLCComS* prodělá vůči prvnímu příkladu opět jen nepatrné změny. Jedinou položkou, kterou je nutné pozměnit, je *IPADDR* (IP adresa používaného PLC zařízení), kterou změníme z lokální adresy *127.0.0.1* (simulovaný režim) na IP adresu PLC zařízení v lokální síti (např. *192.168.33.144*).

Nyní je možné spustit server *PLCComS* se zadanými parametry konfiguračního a logovacího souboru příkazem z příkazové řádky (*Příklad 6.1*).

Ke komunikaci se serverem je opět doporučeno použít program *Putty* podporující protokol TELNET s odpovídajícím nastavením z předchozího příkladu (*Obr. 6.1.4*). Pozor na záměnu IP adresy hostitelské stanice, na které běží server *PLCComS*, s adresou PLC zařízení.

Přes program *Putty* se připojujeme k serveru, tedy např. k IP *127.0.0.1*, pokud běží server na lokální stanici.

7 Licence

Komunikační server *PLCComS* je softwarovým produktem (dále jen software) firmy Teco a.s. Kolín (dále jen výrobce). Je šířen bezplatně a jeho užívání je možné za následujících podmínek:

I. Jednání, které je zejména zakázáno:

- a) jakýmkoliv způsobem software analyzovat, měnit, překládat jej do jiných programovacích či národních jazyků nebo do zdrojového tvaru nebo do tvaru assembler, zahrnout jej do jiného software a šířit produkty takto vzniklé, odvozené od původního software a jakkoliv jinak zasahovat do vnitřní struktury software, s výjimkou případů uvedených v této smlouvě anebo výslovně povolených autorským zákonem,
- b) jakékoliv šíření software za úplatu.

II. Odpovědnost výrobce

Používání software je na bázi "tak, jak je", bez jakékoli záruky. Výrobce na sebe nebere jakoukoliv odpovědnost za ztrátu výnosů, zisků, dat nebo za speciální nepřímé následné nebo nahodilé škody. Výrobce neposkytuje žádnou záruku ani na činnost software, ani na jeho způsoblost k jakémukoli zvláštnímu upotřebení, používání nebo účelu. Výrobce zejména neručí za škody vzniklé nesprávným provozováním software v rozporu s podmínkami, uvedenými v uživatelské dokumentaci. Výrobce ani jeho smluvní partneři nenesou v žádném případě odpovědnost za ušlý zisk nebo jakoukoli jinou komerční ztrátu, včetně a bez omezení zvláštních, vedlejších, trestních a jiných škod, a to i v případě, kdy výrobce nebo jeho smluvní partneři jsou předem upozorněni na možnost takových škod. Veškeré riziko vyplývající z kvality a výkonu software je přeneseno na uživatele software. V případě, že software se ukáže jako vadný, uživatel software, a nikoli výrobce, přebírá veškeré výlohy spojené s jakýmkoli opravami a údržbou.

III. Doba platnosti ujednání

Toto ujednání zůstává v platnosti, pokud uživatel software neporuší podmínky tohoto ujednání. V případě, že uživatel software podmínky ujednání poruší, pozbývá toto ujednání automaticky platnost.

IV. Závěrečná ustanovení

Toto ujednání je v souvislosti s užíváním software kompletní a úplnou smlouvou mezi uživatelem software a výrobcem. Toto ujednání je náhradou za veškerou předcházející, současnou, mluvenou i psanou komunikaci mezi uživatelem software a výrobcem a má rozhodující význam v řešení sporných nebo dodatečně dohodnutých podmínek, nabídek, objednávek mezi stranami zahrnutými v tomto ujednání po dobu jeho platnosti. Jakékoliv úpravy tohoto ujednání jsou povoleny pouze v případě sepsání nové smlouvy nebo dodatku, který bude podepsán oběma zastoupenými stranami.

8 Podporované operační systémy

| OS | Architektura | Překladač |
|---------|--------------|---|
| Windows | x86 | i586-mingw32msvc-g++ |
| Linux | x86 | i586-linux-gnu-g++ |
| Linux | x86_64 | x86_64-linux-gnu-g++ |
| Linux | Armel | arm-linux-gnu-g++, arm-none-linux-gnueabi-g++ |
| Linux | Armhf | arm-linux-gnueabi-g++ |
| Linux | Arm64 | aarch64-linux-gnu-g++ |