

Knihovna InternetLib

**TXV 003 54.01
deváté vydání
květen 2013
změny vyhrazeny**

Historie změn

Datum	Vydání	Popis změn
Říjen 2009	1	První vydání
Únor 2010	2	Doplněny změny pro verzi knihovny InternetLib 1.2
Květen 2010	3	Doplněny změny pro verzi knihovny InternetLib 1.3
Říjen 2010	4	Opravena velikost zóny pro fbHttpRequest
Prosinec 2010	5	Přidán očekávaný kód odpovědi pro fbSendToFtp (InternetLib 1.7)
Únor 2012	6	Doplněny změny pro verzi knihovny InternetLib 2.1
Září 2012	7	Doplněny změny pro verzi knihovny InternetLib 2.3
Říjen 2013	8	Doplněny změny pro verzi knihovny InternetLib 2.4
Květen 2013	9	Nahrazeno v příkladech fbNsLookUp blokem fbNsLookUpEx Zvýrazněno nutné nastavení kanálů

OBSAH

1 Úvod.....	3
2 Datové typy.....	4
3 Konstanty.....	8
4 Překlad doménových jmen.....	8
4.1 Funkční blok fbNsLookUpEx.....	8
4.2 Funkční blok fbNsLookUp.....	11
4.3 Funkční blok fbNsLookUpByTable.....	14
5 Synchronizace času.....	17
5.1 Funkční blok fbSntp.....	17
6 Práce s elektronickou poštou.....	20
6.1 Funkční blok fbSntp.....	20
7 Komunikace HTTP protokolem.....	25
7.1 Funkční blok fbHttpRequest.....	25
7.2 Funkční blok fbHttpRequestL.....	32
8 Komunikace FTP protokolem.....	34
8.1 Funkční blok fbStoreToFtp.....	34

1 ÚVOD

Knihovna InternetLib obsahuje sadu funkcí pro práci se službami dostupnými v síti Internet. Knihovnu lze využít se systémy s centrální jednotkou řady K a L s firmware verze 4.9 a vyšší.

Obsažené funkční bloky realizují překlad doménových jmen na IP adresy, synchronizaci času s časovými servery, odesílání emailů SMTP protokolem a základní dotazy HTTP protokolu.

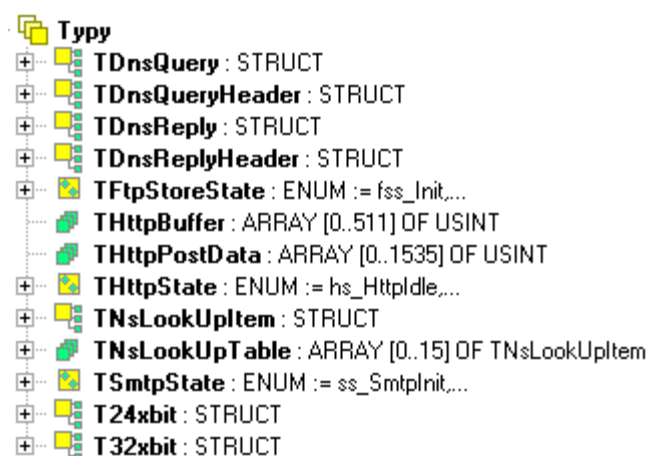
Knihovna využívá některé struktury, funkce a funkční bloky z knihoven FileLib (TXV 003 41) a ComLib (TXV 003 51). Pro správnou funkci musí být tyto knihovny zařazeny v projektu před knihovnou InternetLib.

Knihovna obsahuje následující funkční bloky

fbNsLookupEx	Překlad doménového jména s podporou firmware (od verze 7.1)
fbNsLookup	Překlad doménového jména
fbNsLookupByTable	Překlad více doménových jmen
fbSntp	Synchronizace času
fbSmtp	Odesílání emailů
fbHttpRequest	Komunikace protokolem HTTP
fbHttpRequestL	Komunikace protokolem HTTP s rozšířenou zónou pro metodu POST
fbStoreToFtp	Ukládání souborů na FTP server

2 DATOVÉ TYPY

V knihovně InternetLib jsou definovány následující datové typy:



Typ	Popis	Základní typ
TDnsQuery	Struktura dotazu na DNS server	STRUCT
TDnsQueryHeader	Struktura hlavičky dotazu na DNS server	STRUCT
TDnsReply	Struktura odpovědi DNS serveru	STRUCT
TDnsReplyHeader	Struktura hlavičky odpovědi DNS serveru	STRUCT
TFtpStoreState	Stavy komunikace FTP protokolem	ENUM
THttpBuffer	Pole pro přijatá data HTTP protokolem	ARRAY [0..511] OF USINT
THttpPostData	Pole pro data metody POST pro blok <i>fbHttpRequestL</i>	ARRAY [0..1535] OF USINT
THttpState	Stavy komunikace HTTP protokolem	ENUM
TnsLookUpItem	Dvojice IP adresa doménové jméno s příznaky	STRUCT
TnsLookUpTable	Pole dvojic IP adresa doménové jméno s příznaky	ARRAY [0..31] OF TnsLookUpItem
TSmtplibState	Stavy komunikace SMTP protokolem	ENUM
TStringStreamOrigin	Stavy pro <i>fbStringStream</i>	ENUM
T24xbit	Struktura 24 bitové slova po bytech	STRUCT
T32xbit	Struktura 32 bitové slova po bytech	STRUCT

Význam hodnot enumerací:

TSmtpState - Stavby komunikace SMTP protokolem		
0	ss_SmtpInit	Inicializace
1	ss_SmtpIdle	Není navázáno spojení, komunikace není aktivní
2	ss_SmtpSetIP	Nastavení IP adresy
3	ss_SmtpTxConnect	Navázání spojení se serverem
4	ss_SmtpRxConnect	Čekání na odezvu serveru číslo 220
5	ss_SmtpTxHelo	Odeslání příkazu HELO
6	ss_SmtpRxHelo	Čekání na odezvu serveru číslo 250
7	ss_SmtpTxAuthlogin	Odeslání příkazu AUTH (žádost o autorizované přihlášení)
8	ss_SmtpRxAuthlogin	Čekání na odezvu serveru číslo 334
9	ss_SmtpTxUserName	Odeslání uživatelského jména
10	ss_SmtpRxUserName	Čekání na odezvu serveru číslo 334
11	ss_SmtpTxPassword	Odeslání uživatelského hesla
12	ss_SmtpRxPassword	Čekání na odezvu serveru číslo 235
13	ss_SmtpTxMailFrom	Odeslání adresy odesílatele emailu (příkaz MAIL FROM)
14	ss_SmtpRxMailFrom	Čekání na odezvu serveru číslo 250
15	ss_SmtpTxRcptTo	Odeslání adres příjemců
16	ss_SmtpRxRcptTo	Čekání na odezvu serveru číslo 250 nebo 251
17	ss_SmtpTxData	Odeslání příkazu DATA
18	ss_SmtpRxData	Čekání na odezvu serveru číslo 354
19	ss_SmtpTxDataFrom	Odesílání těla zprávy - odesílatel
20	ss_SmtpTxDataTo	Odesílání těla zprávy - příjemce
21	ss_SmtpTxDataSubject	Odesílání těla zprávy - předmět
22	ss_SmtpTxMultipart	Odesílání těla zprávy - oddělovač částí
23	ss_SmtpTxDataText	Odesílání těla zprávy - text
24	ss_SmtpTxAttachement	Odesílání těla zprávy - oddělovač přílohy
25	ss_SmtpTxAttachementBody	Odesílání těla zprávy - příloha
26	ss_SmtpTxEndOfMail	Odesílání těla zprávy - konec emailu
27	ss_SmtpRxAck	Čekání na odezvu serveru číslo 250
28	ss_SmtpTxQuit	Odeslání příkazu QUIT pro ukončení spojení
29	ss_SmtpRxClose	Čekání na odezvu serveru číslo 221
30	ss_SmtpRxTimeout	Timeout komunikace vypršel
31	ss_SmtpRxError	Při komunikaci došlo k chybě

TFtpStoreState - Stavy komunikace FTP protokolem		
0	fss_Init	Inicializace
1	fss_Idle	Není navázáno spojení, komunikace není aktivní
2	fss_OpenFile	Otevření souboru určeného k přenosu
3	fss_IpCom	Nastavení IP adresy pro spojení pro vysílání příkazů
4	fss_Connect	Navazování spojení
5	fss_Rx220	Čekání na odpověď s kódem 220
6	fss_TxUser	Vysílání uživatelského jména
7	fss_RxUser	Čekání na potvrzení uživatelského jména
8	fss_TxPass	Vysílání hesla
9	fss_RxPass	Čekání na potvrzení přihlášení
10	fss_TxType	Požadavek na binární přenos
11	fss_RxType	Čekání na potvrzení módu přenosu
12	fss_TxPasv	Požadavek na pasivní režim
13	fss_RxPasv	Čekání na potvrzení pasivního režimu s IP adresou a portem
14	fss_TxStor	Příkaz pro uložení souboru na FTP
15	fss_IpDat	Nastavení datového spojení dle parametrů pasivního režimu
16	fss_WaitForOpen	Čekání na otevření spojení pro data
17	fss_TxData	Vysílání dat souboru
18	fss_RxComplete	Příjem potvrzení odvyšlých dat
19	fss_TxQuit	Požadavek na ukončení relace
20	fss_RxQuit	Čekání na potvrzení konce
21	fss_Close	Zavření spojení
22	fss_Timeout	Chybový stav při vypršení času pro odpověď (do verze 2.3)
22	fss_Error	Obecný chybový stav (od verze 2.3)
23	fss_UnexpectedReply	Chybový stav při neočekávané odpovědi serveru
24	fss_CannotOpen	Chybový stav při pokusu o otevření souboru (do verze 2.3)
24	fss_TxCreateDir	Příkaz pro vytvoření adresáře na FTP
25	fss_RxCreateDir	Čekání na potvrzení vytvoření adresáře na FTP
26	fss_ReadDir	Čtení struktury adresáře na paměťové kartě

THttpState - Stavy komunikace HTTP protokolem		
0	hs_HttpIdle	Není navázáno spojení, komunikace není aktivní
1	hs_HttpSetIP	Nastavení IP adresy
2	hs_HttpConnect	Čekání na navázání spojení
3	hs_HttpSend	Posílá se výzva serveru
4	hs_HttpReceivingData	Přijímají se data od serveru

3 KONSTANTY

V knihovně InternetLib nejsou definovány žádné veřejné konstanty.

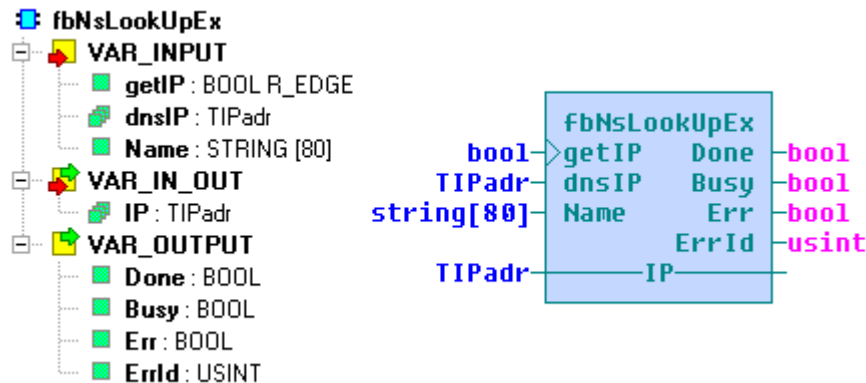
4 PŘEKLAD DOMÉNOVÝCH JMEN

Překlad doménových jmen využívá hierarchického systému doménových jmen DNS (Domain name system) pro získání IP adresy serverů s doménovým jménem.

IP adresy DNS serverů bývají v lokálních sítích stejné s adresou výchozí brány, routeru nebo proxy serveru. Kromě adres lokálních serverů lze použít i adresy přidělené poskytovatelem připojení nebo veřejných DNS serverů. V následujících příkladech se využívá veřejného DNS serveru poskytovaného společností [OpenDNS](#) zdarma. Dotazy jsou vysílány protokolem UDP na standardní port 53 (tento port nastavují bloky automaticky).

4.1 Funkční blok fbNsLookupEx

knihovna: *InternetLib*



Funkční blok *fbNsLookupEx* slouží pro získání IP adresy podle doménového jména. Blok využívá podporu firmware centrály. Podpora je obsažena v centrálách řady K a L od verze 7.1. Blok na rozdíl od *fbNsLookup* a *fbNsLookupByTable* nevyžaduje spojení na ethernet kanálu v režimu UNI.

Simultánně lze volat až sedm instancí funkčního bloku s různými doménovými jmény.

Žádost o IP adresu se vyvolá nastavení vstupu *getIP* na hodnotu TRUE. Žádosti všech instancí se řadí do fronty a jsou vyřizovány v pořadí v jakém byly vyvolány.

IP adresa DNS serveru se předává na vstupu *dnsIP*.

Doménové jméno, které chceme přeložit na IP adresu se zadává na vstupu *Name*.

Po dobu dotazu na DNS server je nastaven výstup *Busy*. V případě úspěšného dotazu je na jeden cyklus nastaven výstup *Done*. Pokud dotaz z nějakého důvodu selže, je nastaveny výstupy *Err* a *ErrId*. Hodnota *ErrId* určuje typ chyby, která nastala. Jednotlivé hodnoty jsou vysvětleny v popisu proměnných.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>getIP</i>	BOOL R_EDGE	Řídící proměnná. Náběžná hrana (přechod z hodnotu FALSE na hodnotu TRUE) zahájí žádost o získání IP adresy
	<i>dnsIP</i>	TIPadr	IP adresa DNS serveru
	<i>Name</i>	STRING	Doménové jméno
VAR_IN_OUT			
	<i>IP</i>	TIPadr	IP adresa získaná z DNS serveru
VAR_OUTPUT			
	<i>Done</i>	BOOL	Má hodnotu TRUE v okamžiku kdy je získána IP adresa Jinak vrací FALSE
	<i>Busy</i>	BOOL	Příznak průběhu získávání adresy
	<i>Err</i>	BOOL	Příznak chyby Pokud operace dopadla úspěšně má hodnotu FALSE, jinak TRUE.
	<i>ErrID</i>	USINT	Chybový kód: ErrID = 0 operace dopadla úspěšně ErrID = 1 vypršel čas pro odpověď serveru ErrID = 2 chybný formát – DNS server nebyl schopen interpretovat dotaz ErrID = 3 selhání serveru – DNS server nebyl schopen zpracovat dotaz, kvůli problémům serveru ErrID = 4 chybné jméno – jméno odkazované v dotazu neexistuje ErrID = 5 není implementováno – DNS server nepodporuje tento typ dotazu ErrID = 6 odmítnuto – DNS server odmítl zpracovat dotaz na základě svých pravidel ErrID = 7–16 jiná chyba – rezervováno pro budoucí užití ErrID = 17 neplatné jméno – doménové jméno je příliš dlouhé nebo prázdné ErrID = 18 DNS tabulka je plná – příliš mnoho dotazů ErrID = 254 nulová adresa DNS serveru

Příklad programu s voláním funkčního bloku *fbNsLookUp* :

Proměnná *GetNtpIp* vyvolává žádost o IP adresu časového serveru, jehož doménové jméno je dáno proměnnou *DomName*. V případě úspěšného obdržení adresy je nastaven bit *NtpIpReady* na hodnotu TRUE.

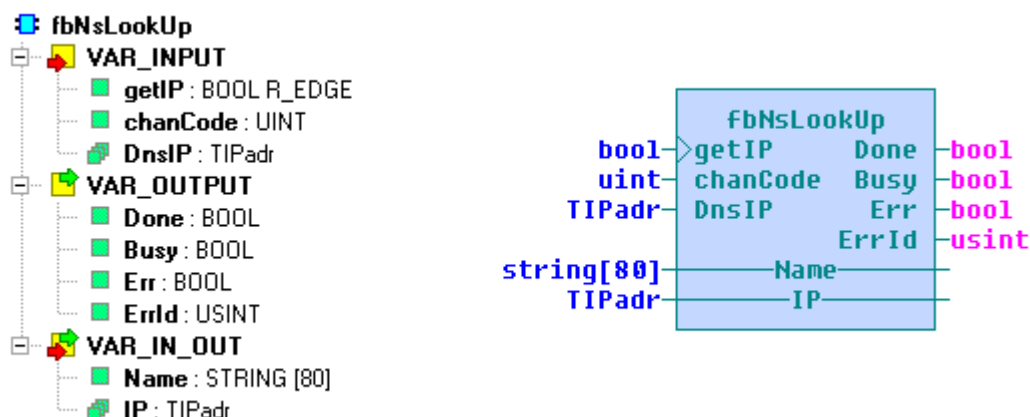
```
VAR_GLOBAL
  GetNtpIP      : BOOL;
  NtpIpReady    : BOOL;
END_VAR

PROGRAM prgExampleNsLookUpEx
  VAR
    NsLookUpEx : fbNsLookUpEx;
    ServerIP    : TIPadr;
    RSReady     : RS;
  END_VAR

  NsLookUpEx(getIP := GetNtpIP,
             DnsIP  := STRING_TO_IPADR('208.67.222.222'),
             Name   := 'cz.pool.ntp.org',
             IP     := ServerIP);

  RSReady(S := NsLookUpEx.Done, R1 := NsLookUpEx.Err, Q1 => NtpIpReady);
END_PROGRAM
```

4.2 Funkční blok fbNsLookup

knihovna: *InternetLib*

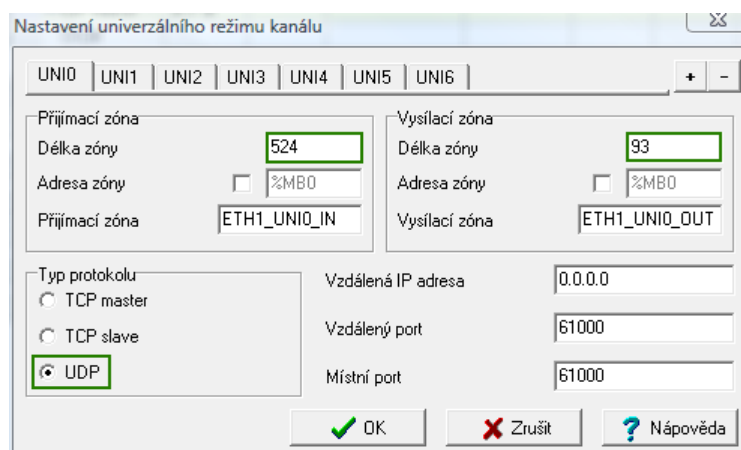
V nových projektech doporučujeme použít novější blok *fbNsLookupEx*.

Funkční blok *fbNsLookup* slouží pro získání IP adresy podle doménového jména. Žádost o IP adresu se vyvolá nastavení vstupu *getIP* na hodnotu TRUE. Žádost se provede přes spojení na ethernet kanálu v režimu UNI podle konstanty na vstupu *chanCode*. Spojení musí mít následujícími parametry: režim UDP, délka přijímací zóny 524 bytů, délka vysílací zóny 93 bytů. Pokud spojení není aktivní nebo nemá správné délky zón blok indikuje chybu na výstupech *Err* hodnotou TRUE a *ErrId* hodnotou 255.

IP adresa DNS serveru se předává na vstupu *DnsIP*, doménové jméno, které chceme přeložit na IP adresu je zadáno přes proměnou na vstupu *Name*.

Po dobu dotazu na DNS server je nastaven výstup *Busy*. V případě úspěšného dotazu je na jeden cyklus nastaven výstup *Done*. Pokud dotaz z nějakého důvodu selže, je nastaveny výstupy *Err* a *ErrId*. Hodnota *ErrId* určuje typ chyby, která nastala. Jednotlivé hodnoty jsou vysvětleny v popisu proměnných.

Pokud je v programu třeba získávat z DNS serveru více IP adres je vhodné využít blok *fbNsLookupByTable*.



Nastavení spojení na ethernetovém kanálu v režimu UNI pro funkční blok *fbNsLookup*

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>getIP</i>	BOOL R_EDGE	Řídící proměnná. Náběžná hrana (přechod z hodnotu FALSE na hodnotu TRUE) zahájí žádost o získání IP adresy
	<i>chanCode</i>	UINT	Kód spojení ETH1_uni0, ETH1_uni1,...
	<i>DnsIP</i>	TIPadr	IP adresa DNS serveru
VAR_IN_OUT			
	<i>Name</i>	STRING	Doménové jméno
	<i>IP</i>	TIPadr	IP adresa získaná z DNS serveru
VAR_OUTPUT			
	<i>Done</i>	BOOL	Má hodnotu TRUE v okamžiku kdy je získána IP adresa Jinak vrací FALSE
	<i>Busy</i>	BOOL	Příznak průběhu získávání adresy
	<i>Err</i>	BOOL	Příznak chyby Pokud operace dopadla úspěšně má hodnotu FALSE, jinak TRUE.
	<i>ErrId</i>	USINT	Chybový kód: <i>ErrId</i> = 0 operace dopadla úspěšně <i>ErrId</i> = 1 vypršel čas pro odpověď serveru <i>ErrId</i> = 2 chybný formát – DNS server nebyl schopen interpretovat dotaz <i>ErrId</i> = 3 selhání serveru – DNS server nebyl schopen zpracovat dotaz, kvůli problémům serveru <i>ErrId</i> = 4 chybné jméno – jméno odkazované v dotazu neexistuje <i>ErrId</i> = 5 není implementováno – DNS server nepodporuje tento typ dotazu <i>ErrId</i> = 6 odmítnuto – DNS server odmítl zpracovat dotaz na základě svých pravidel <i>ErrId</i> = 7–16 jiná chyba – rezervováno pro budoucí užití <i>ErrId</i> = 254 nulová adresa DNS serveru <i>ErrId</i> = 255 chybné nastavení spojení na ethernet kanálu

Příklad programu s voláním funkčního bloku *fbNsLookUp* :

Proměnná *GetNtpIp* vyvolává žádost o IP adresu časového serveru, jehož doménové jméno je dáno proměnnou *DomName*. V případě úspěšného obdržení adresy je nastaven bit *NtpIpReady* na hodnotu TRUE.

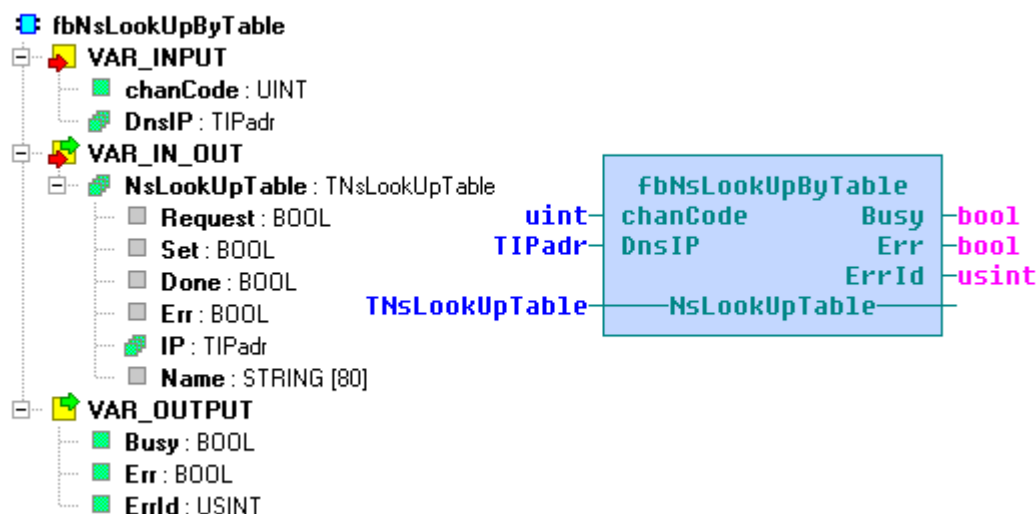
```
VAR_GLOBAL
  GetNtpIP      : BOOL;
  NtpIpReady    : BOOL;
END_VAR

PROGRAM prgExampleNsLookUp
  VAR
    NsLookUp : fbNsLookUp;
    DomName  : STRING := 'cz.pool.ntp.org';
    ServerIP : TIPadr;
    RSReady  : RS;
  END_VAR

  NsLookUp(getIP := GetNtpIP,
            chanCode := ETH1_uni0,
            DnsIP := STRING_TO_IPADR('208.67.222.222'),
            Name := DomName,
            IP := ServerIP);

  RSReady(S := NsLookUp.Done, R1 := NsLookUp.Err, Q1 => NtpIpReady);
END_PROGRAM
```

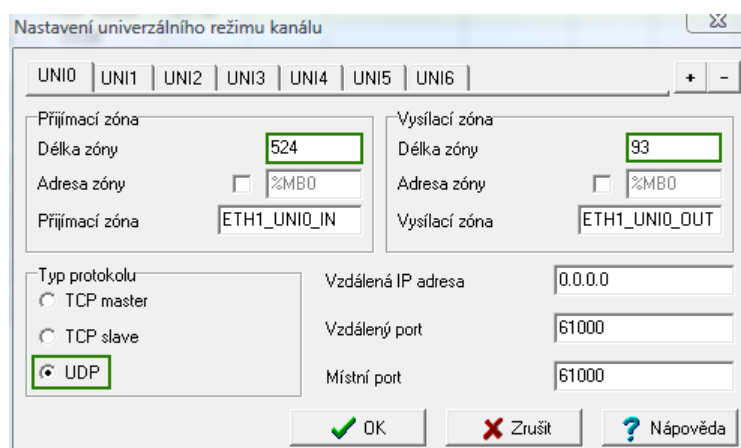
4.3 Funkční blok *fbNsLookUpByTable*

knihovna: *InterneLib*

Funkční blok *fbNsLookUpByTable* slouží k získání více IP adres podle doménových jmen přes jedno spojení. Blok vyžaduje spojení na ethernet kanálu v režimu UNI. Spojení musí mít následujícími parametry: režim UDP, délka přijímací zóny 524 bytů, délka vysílací zóny 93 bytů. Pokud spojení není aktivní nebo nemá správné délky zón blok indikuje chybu na výstupech *Err* hodnotou TRUE a *ErrId* hodnotou 255.

Blok má tři vstupy. Vstup *chanCode* určuje, přes které spojení bude blok pracovat, *DnsIP* adresu DNS serveru, od kterého budou informace získávány a *NsLookUpTable* odkazuje na strukturu s příznaky požadavků, doménovými jmény a IP adresami.

Struktura *NsLookUpTable* pojme až 32 páru doménové jméno, IP adresa. Každá tato dvojice je opatřena sadou bitových příznaků. Nastavením bitu *Request* dojde k zařazení požadavku na získání příslušné IP adresy. Tento bit je okamžitě po přijetí požadavku nulován. Ve chvíli, kdy je IP adresa získána, je nastaven bit *Done* a *Set*. Bit *Done* se nuluje v následujícím cyklu, bit *Set* až při dalšího požadavku. V případě chyby je nastavena proměnná *Err*. S proměnnou *Err* ve struktuře se nastavují společně i výstupy bloky *Err* a *ErrId* s upřesňujícím kódem chyby (Od verze knihovny 1.2 jsou v *ErrId* kódovány příznaky z „Response code“ uváděné RFC1035 pro lepší identifikaci problému. Kódování je provedeno dle klíče *ErrId* = „Response code“+1. Starší verze vrací *ErrId* = 2 při jakémkoli nenulovém „Response code“). Po dobu komunikace je nastaven výstup *Busy*.



Nastavení spojení na ethernetovém kanálu v režimu UNI pro funkční blok *fbNsLookUpByTable*

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>chanCode</i>	UINT	Kód spojení ETH1_uni0, ETH1_uni1,...
	<i>DnsIP</i>	TIPadr	IP adresa DNS serveru
VAR_IN_OUT			
	<i>NsLookUpTable</i>	<i>TNsLookUpTable</i>	Tabulka doménových jmen, příznaků a IP adres
	<i>Request</i>	BOOL	Bitový příznak žádosti o adresu
	<i>Set</i>	BOOL	Bitový příznak úspěšného získání IP adresy
	<i>Done</i>	BOOL	Náběžná hrana příznaku <i>Set</i>
	<i>Err</i>	BOOL	Bitový příznak chyby při získání IP adresy
	<i>IP</i>	TIPadr	IP adresa získaná z DNS serveru
	<i>Name</i>	STRING	Doménové jméno, ke kterému se hledá IP adresa
VAR_OUTPUT			
	<i>Busy</i>	BOOL	Má hodnotu TRUE po dobu komunikace s DNS serverem. Jinak vrací FALSE
	<i>Err</i>	BOOL	Příznak chyby Pokud poslední operace dopadla úspěšně má hodnotu FALSE, jinak TRUE.
	<i>ErrID</i>	USINT	Chybový kód: stejně jako <i>fbNsLookUp</i>

Příklad programu s voláním funkčního bloku fbNsLookupByTable:

V následujícím příkladu jsou získány tři IP adresy níže uvedených adres po startu systému (to je zajištěno inicializací bitů *Request*). Příklad neukazuje použití příznakových bitů *Done* a *Set*. Tyto příznaky mohou být použity kdekoli dále v programu. Příznak *Done* lze použít pro odstartování akce okamžitě po získání IP adresy. Příkaz *Set* lze využít pro kontrolu, zda-li byla IP adresa úspěšně získána a je možné jí použít pro další komunikace.

```
VAR_GLOBAL
  LookupTable : TNsLookupTable :=
    [(Request:= true, Name:= 'cz.pool.ntp.org'),
     (Request:= true, Name:= 'smtp.iol.cz'),
     (Request:= true, Name:= 'kamera.mukolin.cz')];
END_VAR

PROGRAM prgExampleNsLookupByTable
  VAR
    NsLookupByTable : fbNsLookupByTable;
  END_VAR

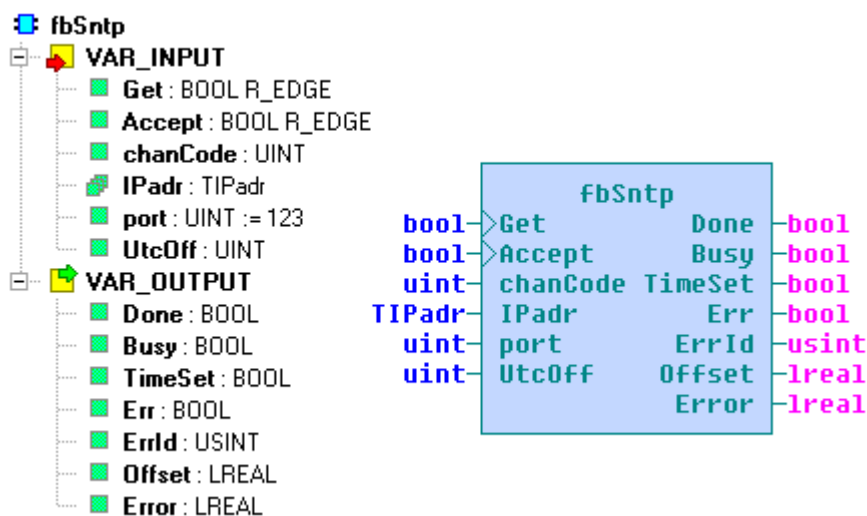
  NsLookupByTable(chanCode := ETH1_uni0,
                  DnsIP := STRING_TO_IPADR('208.67.222.222'),
                  NsLookupTable := LookupTable);
END_PROGRAM
```


5 SYNCHRONIZACE ČASU

Synchronizace času využívá SNTP (Simple Network Time Protocol) protokolu k získání časového rozdílu interních hodin proti času na časovém serveru. Tento rozdíl lze využít pro seřizování systémového času. Časový server může být provozován v lokální síti nebo lze využít veřejné servery. Seznam veřejných serverů lze najít na internetové adrese support.ntp.org. Dotazy jsou vysílány protokolem UDP na standardní port 123 (tento port nastavují bloky automaticky).

5.1 Funkční blok fbSntp

knihovna: *InterneLib*



Funkční blok *fbSntp* slouží k získání časové rozdílu mezi serverem a systémovým časem PLC. Žádost o získání časové difference se vyvolá nastavením vstupu *Get* na hodnotu TRUE. Žádost se provede přes spojení na ethernet kanálu v režimu UNI podle konstanty na vstupu *chanCode*. Spojení musí mít následujícími parametry: režim UDP, délka přijímací a vysílací zóny 60 bytů. Pokud spojení není aktivní nebo nemá správné délky zón, blok indikuje chybu na výstupech *Err* hodnotou TRUE a *ErrId* hodnotou 255.

Adresa časového serveru se předává na vstupu *IPadr* a port, na kterém server přijímá požadavky se nastavuje vstupem *port* (výchozí hodnota pro protokol SNTP je 123). Na vstupu *UtcOff* se očekává posun časové zóny proti GMT v minutách.

Během žádosti o časový rozdíl je nastaven výstup *Busy*. Při úspěšném dokončení operace se objeví na výstupu *Offset* získaný časový rozdíl, na výstupu *Error* maximální chyba získaného rozdílu a je nastaven na jeden cyklus výstup *Done*. V případě neúspěchu je nastaven výstup *Err* a *ErrId*, kde je upřesňující kód chyby.

Po úspěšném získání časového rozdílu, lze s jeho pomocí synchronizovat systémový čas PLC nastavením vstupu *Accept* na hodnotu TRUE. Pokud je vstup *Accept* nastaven na hodnotu TRUE je systémový čas nastaven okamžitě po úspěšném získání časového rozdílu. Pokud byl časový rozdíl již úspěšně získán je provedena korekce systémového času s náběžnou hranou na vstupu *Accept*. Úspěšné nastavení systémového času PLC dle získaného časového rozdílu je indikováno nastavením výstupu *TimeSet*.

Nastavení spojení na ethernetovém kanálu v režimu UNI pro funkční blok fbSntp

Popis proměnných:

	Proměnná	Typ	Význam
VAR_INPUT			
	Get	BOOL R_EDGE	Řídící proměnná. Náběžná hrana zahájí žádost o časový rozdíl
	Accept	BOOL R_EDGE	Nastavení času dle získaného offsetu.
	chanCode	UINT	Kód spojení ETH1_uni0, ETH1_uni1,...
	IPadr	TIPadr	IP adresa časového serveru
	port	UINT	Port časového serveru (výchozí hodnota pro protokol SNTP je 123)
	UtcOff	UINT	Posun časové zóny proti GMT v minutách
VAR_OUTPUT			
	Done	BOOL	Má hodnotu TRUE v okamžiku kdy je úspěšně získán časový rozdíl. Jinak vrací FALSE
	Busy	BOOL	Má hodnotu TRUE po dobu získávání časového rozdílu
	TimeSet	BOOL	Má hodnotu TRUE pokud byl poslední získaný časový rozdíl použitý pro nastavení systémového času
	Err	BOOL	Příznak chyby Pokud poslední operace dopadla úspěšně má hodnotu FALSE, jinak TRUE.
	ErrId	USINT	Chybový kód: ErrId = 0 operace dopadla úspěšně ErrId = 1 vypršel čas pro odpověď serveru ErrId = 2 z odpovědi serveru se nepodařilo určit časový rozdíl ErrId = 254 nulová adresa časového serveru ErrId = 255 chybné nastavení spojení na ethernet kanálu
	Offset	LREAL	Získaný časový rozdíl
	Error	LREAL	Maximální chyba získaného časového rozdílu

Následující příklad ukazuje použití funkčního bloku *fbSntp* pro získání přesného času. Program každý den pět minut před půlnocí zažádá o IP adresu časového serveru, podle kterého nastaví systémový čas. Příklad využívá funkci *GetTime* z knihovny SysLib.

```
VAR_GLOBAL
  NtpName : STRING := 'cz.pool.ntp.org';
  NtpIP : TIPadr;
END_VAR

PROGRAM prgExampleSntp
  VAR_INPUT
  END_VAR
  VAR
    NsLookUp : fbNsLookUpEx;
    Sntp      : fbSntp;
    now       : TIME;
  END_VAR
  VAR_OUTPUT
  END_VAR
  VAR_TEMP
  END_VAR

  now := GetTime();

  NsLookUp(getIP := now > T#23:55:00.0,
           DnsIP := STRING_TO_IPADR('208.67.222.222'),
           Name  := NtpName,
           IP    := NtpIP);

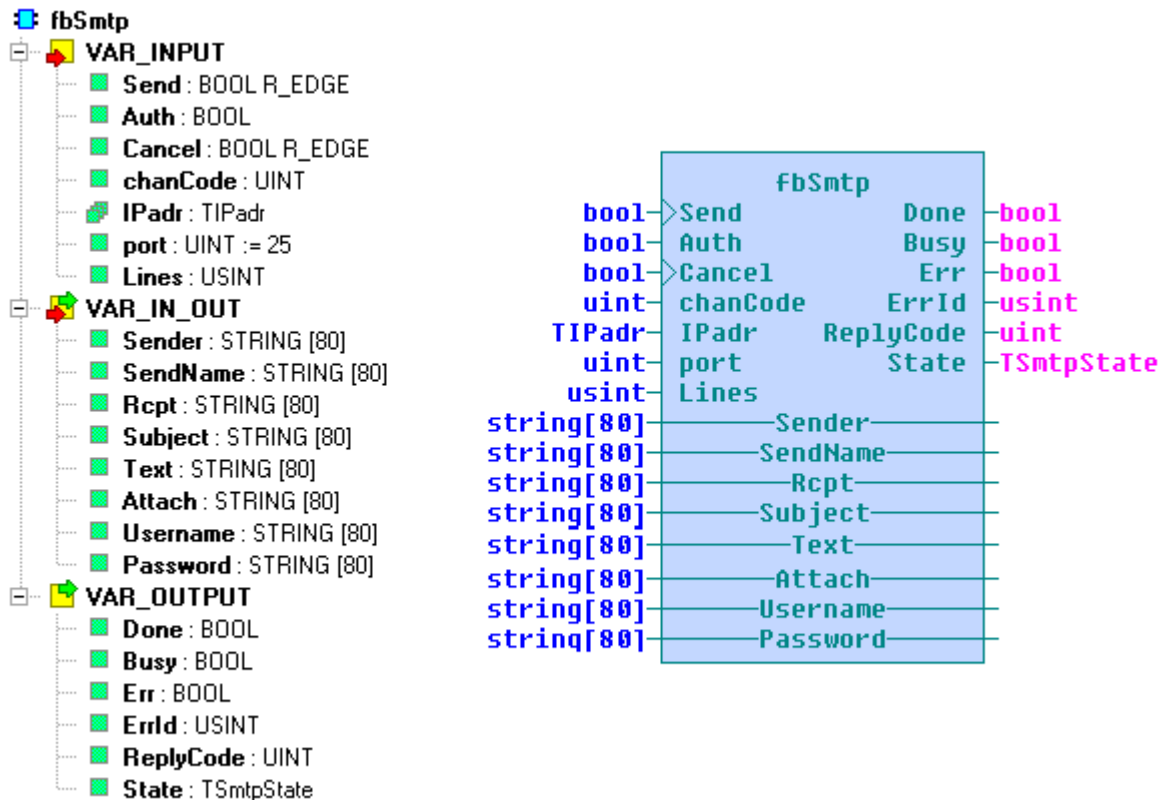
  Sntp(Get := NsLookUp.Done, Accept := Sntp.Done, chanCode := ETH1_uni1,
       IPadr := NtpIP, UtcOff := 60);
END_PROGRAM
```

6 PRÁCE S ELEKTRONICKOU POŠTOU

Knihovna nabízí blok pro odesílání elektronické pošty pomocí protokolu SMTP. Jména SMTP serverů zveřejňují poskytovatelé emailových služeb.

6.1 Funkční blok *fbSmtp*

knihovna: *InterneLib*



Funkční blok *fbSmtp* slouží k odesílání emailových zpráv SMTP protokolem. Odesílání zprávy se zahájí nastavení vstupu *Send* na hodnotu TRUE. Odeslání se provede přes spojení na ethernet kanálu v režimu UNI podle konstanty na vstupu *chanCode*. Spojení musí mít následujícími parametry: režim TCP master, délka přijímací a vysílací zóny 255 bytů. Pokud spojení není aktivní nebo nemá správné délky zón, blok indikuje chybu na výstupech *Err* hodnotou TRUE a *ErrId* hodnotou 255.

Adresa SMTP serveru se předává na vstupu *IPadr* a port, na kterém server přijímá požadavky se nastavuje vstupem *port* (výchozí hodnota pro protokol SMTP je 25).

Na vstupu *Sender* je očekávána proměnná s emailovou adresou odesílatele, na vstupu *SendName* proměnná se jménem odesílatele, které se má zobrazit příjemci a na vstupu *Rcpt* proměnná s adresami příjemců oddělené středníky.

Vlastní zpráva se předává přes proměnné na vstupu *Subject*, kde se očekává předmět zprávy, a vstupu *Text*. Tělo zprávy musí mít formu pole textových řetězců standardní délky (ARRAY [1..*n*] OF STRING), kde *n* je počet řádků zprávy. Na vstupu *Text* se předává první řádek těla zprávy. Počet řádek, které budou skutečně odeslány, udává vstup *Lines*. Hodnota vstupu *Lines* může být menší nebo rovna *n*.

K odesílané zprávě lze připojit soubor z paměťové karty PLC jako přílohu. Jméno souboru se předává proměnnou na vstupu *Attach*. Pro email bez přílohy je na vstup *Attach* potřeba předat proměnnou s prázdným řetězcem.

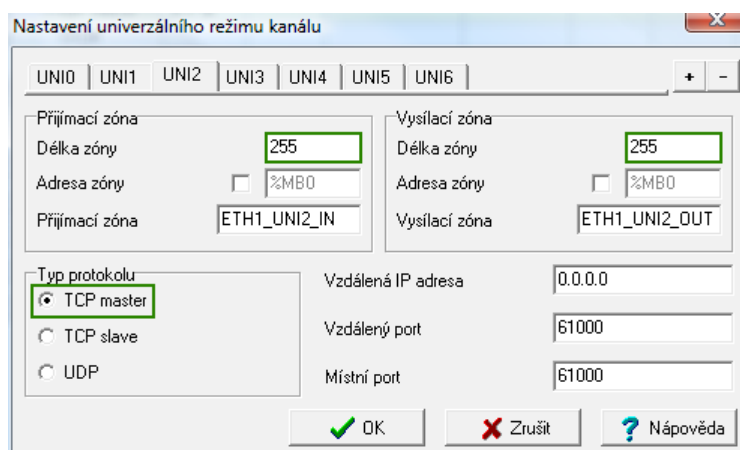
Pokud server vyžaduje ověření pomocí uživatelského jména a hesla je nutné nastavit vstup *Auth* na hodnotu TRUE a na vstupech *UserName* a *Password* předat proměnné s uživatelským jménem a heslem. Pokud server ověření nevyžaduje, může být předána proměnná s prázdným řetězcem.

Během odesílání zprávy je nastaven výstup *Busy* na hodnotu TRUE. Na výstupu *State* se aktualizuje stav komunikace ze serverem (viz enumerace *TSmtpState*). Kódy odpovědi serveru jsou vráceny na výstupu *ReplyCode*. Význam jednotlivých kódů je podrobně popsán a vysvětlen v [RFC 2821](#). Obecně platí, že první číslice odpovědi určuje její typ následujícím způsobem:

- 1yz – **Předběžná pozitivní odpověď** – Příkaz byl přijat, ale vykonání je odloženo. Tuto odpověď používají jen rozšířené příkazy SMTP, které funkční blok nepoužívá
- 2yz – **Pozitivní odpověď** – Příkaz byl přijat a vykonán. (Například spojení se serverem končí kódem 221)
- 3yz – **Pozitivní okamžitá odpověď** – Příkaz byl přijat, očekávají se další informace. (Do této skupiny patří například odpovědi při ověřování uživatele 334, nebo při odesílání těla zprávy 354)
- 4yz – **Dočasná negativní odpověď** – Příkaz nebyl přijat, důvod zamítnutí není trvalý, je možné příkaz zkusit znovu. (Odpovědi s tímto typem kódu jsou většinou příznakem zaneprázdnění nebo nedostatku prostředků na straně poštovního serveru)
- 5yz – **Trvale negativní odpověď** – Příkaz nebyl přijat, důvod je trvalý, nedoporučuje se opakovat žádost ze stejnými parametry. (Nejčastěji se tento kód v odpovědi objevuje, nedostal-li server ověření, které vyžadoval, nebo pokud ověření uživatele nezdaří.)

V případě úspěšného odeslání zprávy je nastaven na dobu jednoho cyklu výstup *Done*.

V případě chyby je nastaven výstup *Err* a *ErrId*, kde je upřesňující chybový kód.



Nastavení spojení na ethernetovém kanálu v režimu UNI pro funkční blok *fbSmtplib*

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>Send</i>	BOOL R_EDGE	Řídící proměnná. Náběžná hrana zahájí odesílání emailu.
	<i>Auth</i>	BOOL	Zapíná funkci ověřování uživatele jménem a heslem.
	<i>Cancel</i>	BOOL R_EDGE	Náběžná hrana předčasně ukončí probíhající odesílání.
	<i>chanCode</i>	UINT	Kód spojení ETH1_uni0, ETH1_uni1,...
	<i>IPadr</i>	TIPadr	IP adresa SMTP serveru
	<i>port</i>	UINT	Port časového serveru (výchozí hodnota pro protokol SMTP je 25)
	<i>Lines</i>	USINT	Počet řádků textu k odeslání
VAR_IN_OUT			
	<i>Sender</i>	STRING	Emailová adresa odesílatele
	<i>SendName</i>	STRING	Jméno odesílatele zobrazené příjemci (může obsahovat jen základní znaky bez diakritiky)
	<i>Rcpt</i>	STRING	Emailové adresy příjemců oddělené středníky
	<i>Subject</i>	STRING	Předmět zprávy
	<i>Text</i>	STRING	První řádek těla zprávy
	<i>Attach</i>	STRING	Jméno souboru pro připojení k emailové zprávě
	<i>Username</i>	STRING	Uživatelské jméno
	<i>Password</i>	STRING	Uživatelské heslo
VAR_OUTPUT			
	<i>Done</i>	BOOL	Má hodnotu TRUE v okamžiku kdy je email úspěšně odeslán. Jinak vrací FALSE
	<i>Busy</i>	BOOL	Má hodnotu TRUE po dobu odesílání emailu.
	<i>Err</i>	BOOL	Příznak chyby Pokud poslední operace dopadla úspěšně má hodnotu FALSE, jinak TRUE.
	<i>Errld</i>	USINT	Chybový kód: <i>Errld</i> = 0 operace dopadla úspěšně <i>Errld</i> = 1 vypršel čas pro odpověď serveru <i>Errld</i> = 2 neočekávaná odpověď serveru (více viz ReplyCode) <i>Errld</i> = 3 nelze otevřít soubor, email bude odeslán bez přílohy <i>Errld</i> = 254 nulová adresa SMTP serveru <i>Errld</i> = 255 chybné nastavení spojení na ethernet kanálu
	<i>ReplyCode</i>	LREAL	Kód odpovědi SMTP serveru
	<i>State</i>	TSmtpState	Stav komunikace se serverem (viz enumerace TSmtpState)

Následující příklad ukazuje použití funkčního bloku *fbSmtplib* pro odeslání emailové zprávy. Proměnná *HeatingIsOn* představuje stav topení (zapnuto/vypnuto), který se porovnává s posledním stavem ukládaným do lokální proměnné *LastHeatingState*. V případě změny stavu je proveden dotaz na DNS server na IP adresu SMTP serveru a sestavena zpráva. Základ zprávy je definován konstantou *BodyTemplate*, do které je doplněn aktuální datum a teploty ze vstupů PLC. K modifikaci těla zprávy jsou využity formátovací funkce z knihovny *ToStringLib*, aktuální datum a čas je získán funkcí *GetDateTime* z knihovny *SysLib*.

Po úspěšném dotazu na DNS server je zpráva odeslána.

```

VAR_GLOBAL
  SmtplibName      : STRING := 'smtp.seznam.cz';
  SmtplibIP        : TIpAddr;
  TempOutdoor      AT r0_p3_AI0.ENG : REAL;
  TempIndoor       AT r0_p3_AI1.ENG : REAL;
  TempHeating      AT r0_p3_AI2.ENG : REAL;
  HeatingIsOn      : BOOL;
END_VAR

VAR_GLOBAL CONSTANT
  NumberOfLines : USINT := 5;
END_VAR

TYPE
  TEmailBody : ARRAY [1..NumberOfLines] OF STRING;
END_TYPE

VAR_GLOBAL CONSTANT
  BodyTemplate : TEmailBody := ['Status report %TDD.MM.YYYY$A0hh:mm',
                                'Heating is switched ',
                                'Outdoor temperature is %5.1f°C',
                                'Indoor temperature is %5.1f°C',
                                'Heating temperature is %5.1f°C'];
END_VAR

PROGRAM prgExampleSmtplib
  VAR
    NsLookUp      : fbNsLookUpEx;
    Smtplib        : fbSmtplib;
    LastHeatingState : BOOL;
    Sender         : STRING := 'TestPLC@seznam.cz';
    SenderName     : STRING := 'Do not reply';
    UserName       : STRING := 'TestPLC@seznam.cz';
    Password       : STRING := '*****';

    Recipient      : STRING := 'notavailable@seznam.cz';
    Subject        : STRING := 'Heating status report';
    Attachment     : STRING;
    Body           : TEmailBody;

  END_VAR

  IF LastHeatingState <> HeatingIsOn THEN
    Body[1] := DT_TO_STRINGF(in := GetDateTime(),
                             format := BodyTemplate[1]);

    IF HeatingIsOn THEN
      Body[2] := CONCAT(BodyTemplate[2], 'on');
    ELSE
      Body[2] := CONCAT(BodyTemplate[2], 'off');
    END_IF;
  END_IF;

```

```
Body[3] := REAL_TO_STRINGF(in := TempOutdoor,  
                           format := BodyTemplate[3]);  
Body[4] := REAL_TO_STRINGF(in := TempIndoor,  
                           format := BodyTemplate[4]);  
Body[5] := REAL_TO_STRINGF(in := TempHeating,  
                           format := BodyTemplate[5]);  
END_IF;  
  
NsLookUp(getIP := LastHeatingState <> HeatingIsOn,  
         DnsIP :=  STRING_TO_IPADR('208.67.222.222'),  
         Name := SmtplibName,  
         IP := SmtplibIP);  
  
LastHeatingState := HeatingIsOn;  
  
Smtplib(Send := NsLookUp.Done, Auth := true,  
        chanCode := ETH1_uni2, IPadr := SmtplibIP,  
        Lines := NumberOfLines, Sender := Sender,  
        SendName := SenderName, Rcpt := Recipient,  
        Subject := Subject, Attach := Attachement,  
        Username := UserName, Password := Password,  
        Text := Body[1]);  
  
END_PROGRAM
```


7 KOMUNIKACE HTTP PROTOKOLEM

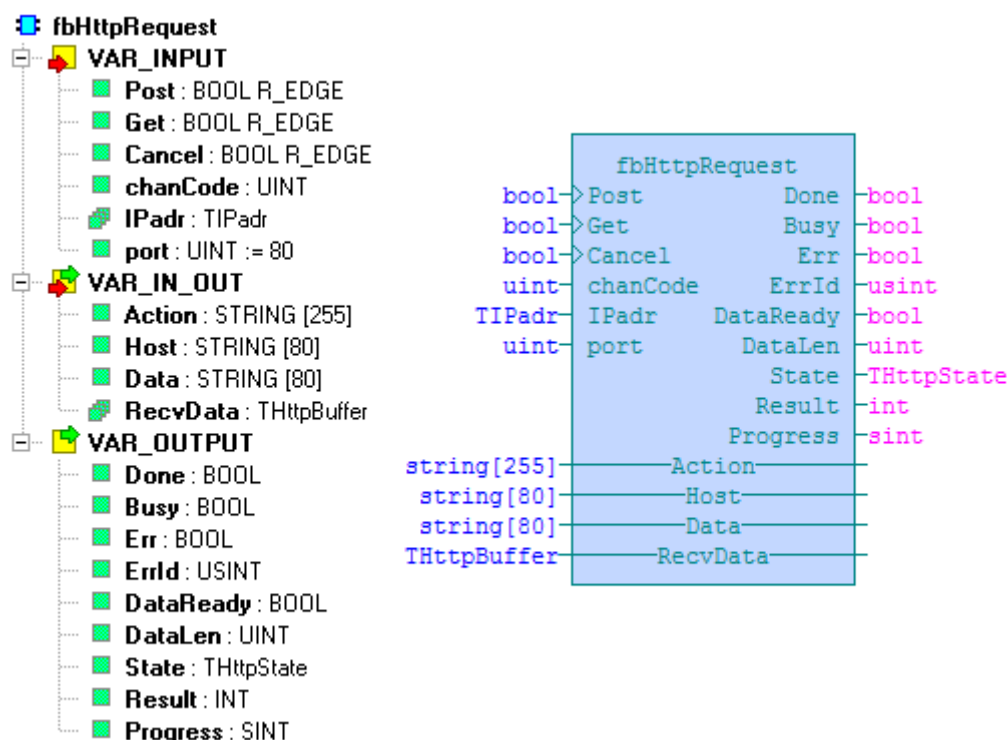
Knihovna nabízí funkční blok pro komunikaci s webovým serverem přes protokol HTTP. Blok implementuje ze souboru metod HTTP protokolu metody GET a POST.

Metoda GET slouží pro získání dat z webového serveru. Typicky lze využít k získání obrazu z IP kamery, stahování receptur ze serveru dispečinku nebo získávání dat z veřejných serverů (údaje o počasí apod.).

Metoda POST se využívá k odesílání dat na webový server. Typické použití je automatizovaný sběr dat vysíláním na centrální server.

7.1 Funkční blok *fbHttpRequest*

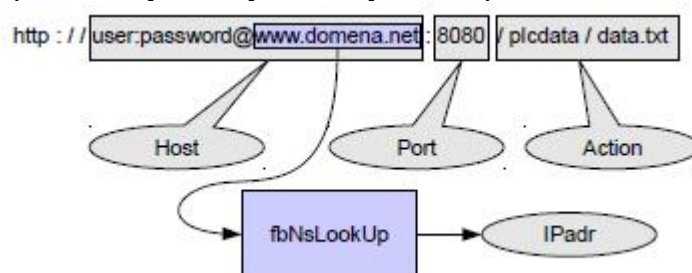
knihovna: *InterneLib*



Funkční blok *fbHttpRequest* slouží ke komunikaci s webovým serverem přes protokol HTTP 1.0. Blok implementuje ze souboru metod HTTP protokolu metody GET a POST. Komunikace probíhá přes spojení na ethernet kanálu v režimu UNI podle konstanty na vstupu *chanCode*. Spojení musí mít následujícími parametry: režim TCP master, délka přijímací a vysílací zóny 512 bytů. Pokud spojení není aktivní nebo nemá správné délky zón, blok indikuje chybu na výstupech *Err* hodnotou TRUE a *ErrId* hodnotou 255.

Adresa stahovaných dat se předává na čtyřech vstupech. Na vstupu *IPadr* se očekává adresa serveru (typicky získaná z doménového jména serveru blokem *fbNsLookUp* nebo *fbNsLookUpByTable* nebo *fbNsLookUpEx*), na vstupu *Port* se předává číslo portu, na kterém server naslouchá (výchozí hodnota pro HTTP protokol je 80). Na vstupu *Host* se očekává proměnná s doménovým jménem serveru. Od verze knihovny 2.4 je na vstupu *Host* možné také předat parametry základní autentizace jako součást doménového jména ve tvaru „jméno:heslo@“. Na vstupu *Action* se očekává proměnná s cestou k datům na serveru (cesta vždy začíná znakem lomítko!).

Na obrázku níže je naznačeno, jak souvisí data na adresním řádku webového prohlížeče s hodnotami předávanými na jednotlivých vstupech.



Převod dat z adresního řádku webového prohlížeče na vstupy funkčního bloku
 Port nemusí být uveden, v takovém případě má port výchozí hodnotu 80
 Parametry základní autentizace user:password@ jsou taktéž nepovinné

Komunikace se podle zvolené metody zahájí nastavením stavu *Get* nebo *Post*. Metoda *Post* očekává proti metodě *Get* navíc data v proměnné předané na vstupu *Data*. Pro snadné zpracování na straně serveru by měla mít proměnná na vstupu *Data* následující formát:

NázevHodnoty1	=	Hodnota1	&	NázevHodnoty2	=	Hodnota2	&	..	&	NázevHodnotyN	=	HodnotaN
---------------	---	----------	---	---------------	---	----------	---	----	---	---------------	---	----------

Například: temp1=20.4&state=1&error=0

Řetězce v proměnných na vstupu *Action* a *Data* musí být ve formátu URI (Uniform Resource Identifier) dle [RFC 2396](#). Obecně platí, že tyto řetězce mohou obsahovat jen číslice a písmena bez diakritiky, ostatní symboly včetně mezer by měly být kódovány ve tvaru % následované dvěma hexadecimálními číslicemi, které vyjadřují hodnotu znaku v ASCII tabulce (například „%20“ je zástupný kód pro mezeru).

Během komunikace je nastaven výstup *Busy* na hodnotu TRUE. V případě úspěšného ukončení je nastaven na jeden cyklus výstup *Done*. V případě neúspěchu je nastaven výstup *Err* a *ErrId*, který obsahuje upřesňující číslo chyby.

V případě chyby *ErrId* = 3 přesáhla velikost hlavičky protokolu velikost vysílacího bufferu. Tento stav nastane pokud jsou proměnné *Host*, *Action* a *Data* v součtu příliš dlouhé. V takovém případě je možné využít zkracovače jmen (např. službu bit.ly) nebo pro metodu post využít blok *fbHttpRequestL*, kde proměnná *Data* není součástí hlavičky.

Výstup *State* indikuje aktuální stav komunikace. Po načtení hlavičky zprávy ze serveru se nastaví výstup *Result* se stavovým kódem (viz tabulka níže) a pokud je dostupná délka následujících dat vrací se na výstupu *Progress* průběh stahování v procentech (0 až 100). Ve všech jiných stavech nebo pokud není délka známa vrací *Progress* hodnotu -1.

Význam nejčastějších stavových kódů na výstupu *Result*

Kód	Význam
200	OK – data nalezena
302	Found – data byla přesunuta
403	Forbidden – přístup odepřen
500	Internal Server Error – vnitřní chyba serveru
další kódy viz RFC 2616	

Data ze serveru přichází v po sobě následujících blocích. Každý cyklus může být vrácen jeden blok. Přítomnost nových dat je signalizována hodnotou TRUE na výstupu *DataReady*. Blok dat se vrací do proměnné na vstupu *RecvData* a jeho délku indikuje výstup *DataLen*.

Nastavení spojení na ethernetovém kanálu v režimu UNI pro funkční blok *fbHttpRequest*

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>Post</i>	BOOL R_EDGE	Náběžná hrana zahájí komunikaci metodou POST
	<i>Get</i>	BOOL R_EDGE	Náběžná hrana zahájí komunikaci metodou GET
	<i>Cancel</i>	BOOL R_EDGE	Náběžná hrana přeruší právě probíhající komunikaci
	<i>chanCode</i>	UINT	Kód spojení ETH1_uni0, ETH1_uni1,...
	<i>IPadr</i>	TIPadr	IP adresa webového serveru
	<i>port</i>	UINT	Port webového serveru (výchozí hodnota pro protokol HTTP je 80)
VAR_IN_OUT			
	<i>Action</i>	STRING	Cesta k datům na serveru (vždy začíná znakem /)
	<i>Host</i>	STRING	Doménové jméno serveru
	<i>Data</i>	STRING	Data pro metodu POST
	<i>RecvData</i>	THttpBuffer	Blok přijatých dat
VAR_OUTPUT			
	<i>Done</i>	BOOL	Má hodnotu TRUE v okamžiku kdy je úspěšně ukončena komunikace se serverem. Jinak vrací FALSE
	<i>Busy</i>	BOOL	Má hodnotu TRUE po komunikaci se serverem
	<i>Err</i>	BOOL	Příznak chyby Pokud poslední operace dopadla úspěšně má hodnotu FALSE, jinak TRUE.
	<i>ErrId</i>	USINT	Chybový kód: <i>ErrId</i> = 0 operace dopadla úspěšně <i>ErrId</i> = 1 vypršel čas pro odpověď serveru <i>ErrId</i> = 2 nepodařilo se získat všechna data ze serveru <i>ErrId</i> = 3 hlavička protokolu HTTP přesáhla 512 bytů <i>ErrId</i> = 254 nulová adresa webového serveru <i>ErrId</i> = 255 chybné nastavení spojení na ethernet kanálu
	<i>DataReady</i>	BOOL	Hodnota TRUE indikuje nový blok na vstupu <i>RecvData</i>
	<i>DataLen</i>	UINT	Délka přijatého bloku dat
	<i>State</i>	THttpState	Stav komunikace se serverem (viz enumerace THttpState)
	<i>Result</i>	INT	Stavový kód vrácený serverem
	<i>Progress</i>	SINT	Během stahování dat ze serveru indikuje průběh 0 až 100%. Jinak vrací -1

Následující příklad ukazuje použití funkčního bloku *fbHttpRequest* pro stažení obrázku z webové kamery. Příklad využívá funkční blok *WriteToFileSeq* z knihovny *FileLib* pro ukládání přijatých dat na paměťovou kartu. Stažení obrázku se iniciuje nastavením proměnné *GetPicture* a je podmíněné úspěšným vytvořením cesty pro ukládání souborů, což indikuje proměnná *PathOk*. Pro vytváření jmen souborů s inkrementujícím se inde-
xem je použita formátovací funkce z knihovny *ToStringLib*.

```

VAR_GLOBAL CONSTANT
  PathTemplate      : STRING := 'WWW/PICT/';
  FileNameTemplate : STRING := PathTemplate + 'PICT%04d.JPG';
END_VAR

VAR_GLOBAL
  HttpIP      : TIPadr;
  HttpName    : STRING := 'posta.mukolin.cz';
  Action      : STRING := '/axis-cgi/jpg/image.cgi?resolution=CIF';
  Path        : STRING := PathTemplate;
  FileName    : STRING;

  GetPicture   : BOOL;
  PathOk       : BOOL;
  PictIndx     : INT;
END_VAR

PROGRAM prgExampleHttpGet
  VAR
    NsLookUp    : fbNsLookUpEx;
    HttpRequest : fbHttpRequest;
    WriteToFile : WriteToFileSeq;
    CPath       : CreatePath;
    Empty       : STRING[1];
    Data        : THttpBuffer;
  END_VAR

  CPath(exec := NOT PathOk, fileName := Path, done => PathOk);

  NsLookUp(getIP := PathOk AND GetPicture,
    DnsIP := STRING_TO_IPADR('208.67.222.222'),
    Name  := HttpName,
    IP    := HttpIP);

  HttpRequest(Get := NsLookUp.Done, chanCode := ETH1_uni3,
    IPadr := HttpIP,
    Action := Action,
    Host   := HttpName,
    Data   := Empty, RecvData := Data);

  FileName := INT_TO_STRINGF(in := PictIndx,
    format := FileNameTemplate);

  WriteToFile(fileName := FileName,
    srcVar := void(Data),
    write  := HttpRequest.Result = 200 & HttpRequest.DataReady,
    close  := HttpRequest.Done OR HttpRequest.Err,
    size   := UINT_TO_UDINT(HttpRequest.DataLen));

  IF HttpRequest.Done THEN
    PictIndx := PictIndx + 1;
  END_IF;
END_PROGRAM

```

Druhý příklad ukazuje použití funkčního bloku *fbHttpRequest* pro odesílání dat do databáze na webovém serveru metodou POST. Proměnná *HeatIsOn* představuje stav topení (zapnuto/vypnuto), který se porovnává s posledním stavem ukládaným do lokální proměnné *LastHeatState*. V případě změny stavu je proveden dotaz na DNS server na IP adresu webového serveru a sestaven řetězec s daty. Po vyslání dat se v přijaté odpovědi hledá řetězec „OK“, který skript na serveru (uvedený níže) vrací v případě úspěšného uložení dat. Pro kopírování přijatých dat z buferu do řetězce je použita funkce *memcpy* z knihovny *SysLib*.

```

VAR_GLOBAL
TempOut      AT r0_p3_AI0.ENG : REAL;
TempIn       AT r0_p3_AI1.ENG : REAL;
TempHeat     AT r0_p3_AI2.ENG : REAL;
HeatIsOn     : BOOL;

HttpPostIP    : TIPadr;
HttpPostName   : STRING := 'foxtrot.howto.cz';
HttpPostAction : STRING := '/index.php';
END_VAR

PROGRAM prgExampleHttpPost
VAR
NsLookup      : fbNsLookupEx;
HttpRequest   : fbHttpRequest;
DataIn        : THttpBuffer;
DataInString  : STRING;
DataOut       : STRING;
LastHeatState : BOOL;
PostSuccesful : BOOL;
END_VAR

NsLookup.getIP := LastHeatState <> HeatIsOn;
LastHeatState := HeatIsOn;

IF NsLookup.getIP THEN
PostSuccesful := false;
END_IF;

NsLookup(DnsIP := STRING_TO_IPADR('208.67.222.222'),
Name := HttpPostName,
IP := HttpPostIP);
DataOut := 'Heat=' + BOOL_TO_STRING(HeatIsOn) +
'&TempOut=' + REAL_TO_STRING(TempOut) +
'&TempIn=' + REAL_TO_STRING(TempIn) +
'&TempHeat=' + REAL_TO_STRING(TempHeat);
HttpRequest(Post := NsLookup.Done, chanCode := ETH1_uni3,
IPadr := HttpPostIP,
Action := HttpPostAction,
Host := HttpPostName,
Data := DataOut, RecvData := DataIn);
IF HttpRequest.DataReady THEN
Memcpy(length := min(80, HttpRequest.DataLen),
source := VOID(DataIn),
dest := VOID(DataInString));
IF FIND(IN1 := DataInString, IN2 := 'OK') > 0 THEN
PostSuccesful := true;
END_IF;
END_IF;
END_PROGRAM

```

Následující PHP skript na straně serveru ukládá data vyslaná metodou POST do SQL databáze. Kromě této funkce navíc skript jako reakci na metodu GET generuje přehledovou tabulku se všemi zaznamenanými daty během aktuálního dne. Soubor je na serveru uložen v souboru *index.php*, na který se odkazuje příklad výše. Proměnné *\$db_server*, *\$db_name*, *\$db_user*, *\$db_pass* obsahují údaje pro připojení k serveru s SQL databází. K datům vyslaným z PLC je přístupováno pomocí globální proměnné *\$POST*, kde je jako index použitý název hodnoty.

```
<?php
$db_server = "mysql.ic.cz";
$db_name   = "ht_foxtrot";
$db_user   = "ht_foxtrot";
$db_pass   = "*****";

$link = mysql_connect($db_server, $db_user, $db_pass)
        or die("ERR - " . mysql_error());
mysql_select_db($db_name) or die("ERR - unable to select database");

if(!empty($_POST)) {
    header("Content-type: text/plain");
    $query = "INSERT INTO plc_data VALUES ('".date("Y-m-d-H:i:s").
        "', '". $_POST['Heat']."' , '". $_POST['TempOut']."' , '".
        $_POST['TempIn']."' , '". $_POST['TempHeat']."' );";
    $result = mysql_query($query) or die("ERR - " . mysql_error());
    echo "OK";
} else {

    echo "<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.01 Transitional//EN\">";
    echo "<html><head><title>PLC data</title></head><body><center>";

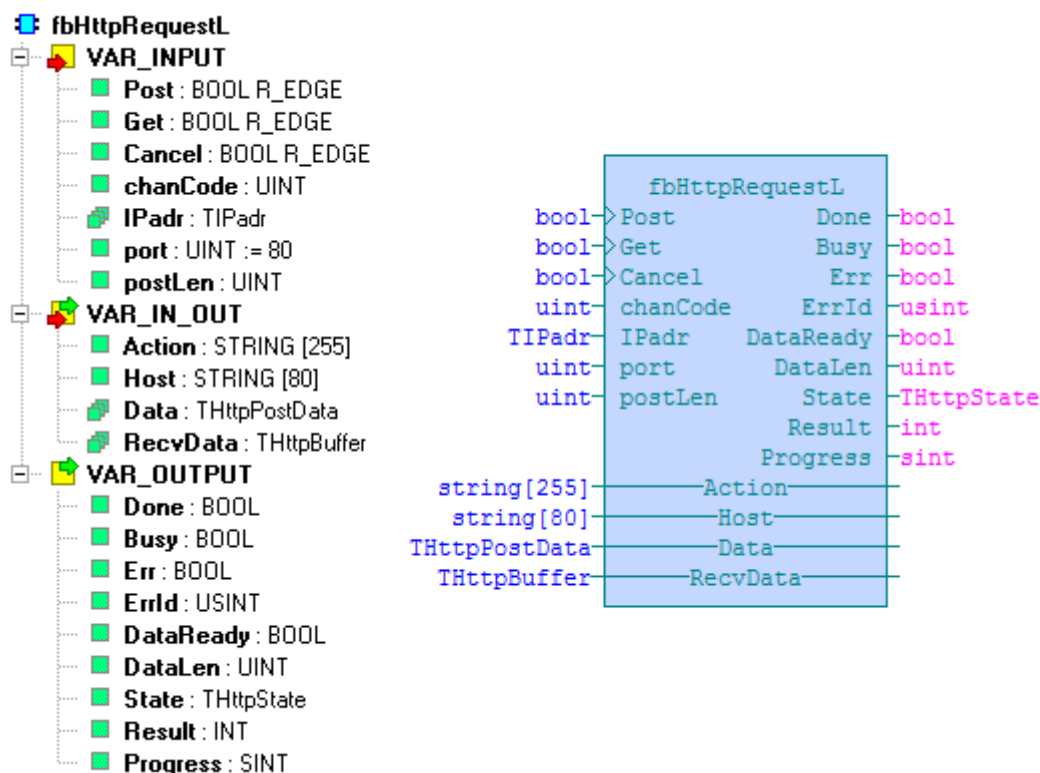
    $query = "SELECT * FROM plc_data WHERE datetime LIKE '".date("Y-m-d")."%';";
    $result = mysql_query($query) or die("ERR - " . mysql_error());

    print "<br><br><table border='1' cellpadding='2' cellspacing='1'>";
    print "<tr><th>Time stamp</th><th>State</th><th>Outdoor temperature</th>".
        "<th>Indoor temperature</th><th>Heating temperature</th></tr>";
    while ($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
        print "<tr><td>".$line['datetime']."</td><td>".$line['Heat'].
            "</td><td>".number_format($line['TempOut'],1)."</td><td>".
            number_format($line['TempIn'],1)."</td><td>".
            number_format($line['TempHeat'],1)."</td></tr>";
    }
    print "</table></center>";
    echo "</body></html>";
}
mysql_close($link);
?>
```

Tabulka v databázi využívaná PHP skriptem byla založena následujícím SQL příkazem:

```
CREATE TABLE `plc_data` (
  `datetime` timestamp NOT NULL default CURRENT_TIMESTAMP,
  `Heat` tinyint(1) NOT NULL, `TempOut` double NOT NULL,
  `TempIn` double NOT NULL, `TempHeat` double NOT NULL,
  UNIQUE KEY `datetime` (`datetime`)
);
```

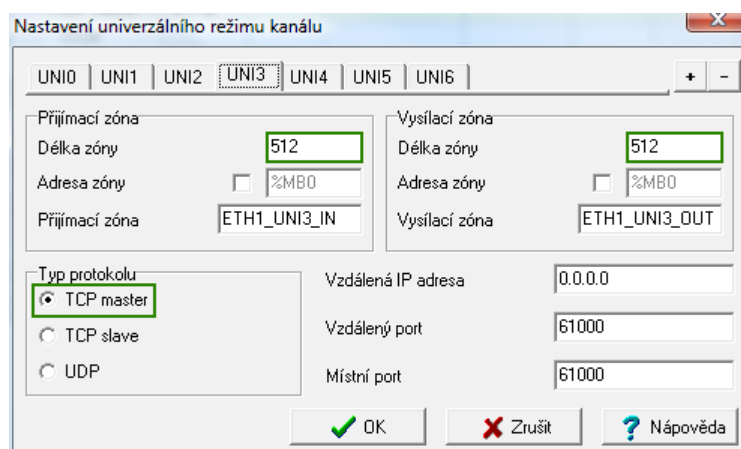
7.2 Funkční blok fbHttpRequestL

knihovna: *InterneLib*

Funkční blok *fbHttpRequestL* slouží ke komunikaci s webovým serverem přes protokol HTTP 1.0 stejně jako *fbHttpRequest*. *fbHttpRequestL* se liší od bloku *fbHttpRequest* možností odeslat metodou POST větší množství dat (až 1535 bytů).

Data pro metodu POST se nepředávají jako u *fbHttpRequest* přes typ STRING, ale přes obecnou VAR_IN_OUT proměnnou *Data*. Délka přenášených dat se předává na vstupu *postLen*. Tato délka musí být menší nebo rovna velikosti proměnná předané na vstupu *Data*. Všechny ostatní vstupy a výstupy mají stejný význam a funkci jako u bloku *fbHttpRequest*.

Nastavení spojení a chování je shodné s *fbHttpRequestL*.



Nastavení spojení na ethernetovém kanálu v režimu UNI pro funkční blok *fbHttpRequest*

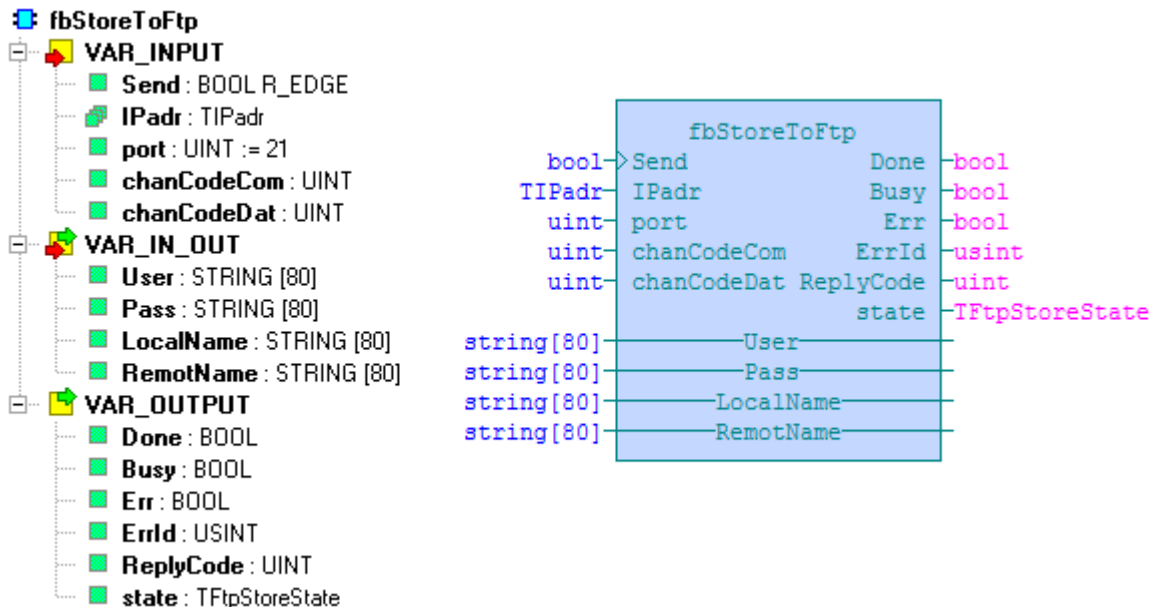
Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>Post</i>	BOOL R_EDGE	Náběžná hrana zahájí komunikaci metodou POST
	<i>Get</i>	BOOL R_EDGE	Náběžná hrana zahájí komunikaci metodou GET
	<i>Cancel</i>	BOOL R_EDGE	Náběžná hrana přeruší právě probíhající komunikaci
	<i>chanCode</i>	UINT	Kód spojení ETH1_uni0, ETH1_uni1,...
	<i>IPAdr</i>	TIPAdr	IP adresa webového serveru
	<i>port</i>	UINT	Port webového serveru (výchozí hodnota pro protokol HTTP je 80)
	<i>postLen</i>	UINT	Délka dat pro metodu POST
VAR_IN_OUT			
	<i>Action</i>	STRING	Cesta k datům na serveru (vždy začíná znakem /)
	<i>Host</i>	STRING	Doménové jméno serveru
	<i>Data</i>	THttpPost-Data	Data pro metodu POST
	<i>RecvData</i>	THttpBuffer	Blok přijatých dat
VAR_OUTPUT			
	<i>Done</i>	BOOL	Má hodnotu TRUE v okamžiku kdy je úspěšně ukončena komunikace se serverem. Jinak vrací FALSE
	<i>Busy</i>	BOOL	Má hodnotu TRUE po komunikaci se serverem
	<i>Err</i>	BOOL	Příznak chyby Pokud poslední operace dopadla úspěšně má hodnotu FALSE, jinak TRUE.
	<i>ErrId</i>	USINT	Chybový kód: <i>ErrId</i> = 0 operace dopadla úspěšně <i>ErrId</i> = 1 vypršel čas pro odpověď serveru <i>ErrId</i> = 2 nepodařilo se získat všechna data ze serveru <i>ErrId</i> = 3 hlavička protokolu HTTP přesáhla 512 bytů <i>ErrId</i> = 254 nulová adresa webového serveru <i>ErrId</i> = 255 chybné nastavení spojení na ethernet kanálu
	<i>DataReady</i>	BOOL	Hodnota TRUE indikuje nový blok na vstupu <i>RecvData</i>
	<i>DataLen</i>	UINT	Délka přijatého bloku dat
	<i>State</i>	THttpState	Stav komunikace se serverem (viz enumerace THttpState)
	<i>Result</i>	INT	Stavový kód vrácený serverem
	<i>Progress</i>	SINT	Během stahování dat ze serveru indikuje průběh 0 až 100%. Jinak vrací -1

8 KOMUNIKACE FTP PROTOKOLEM

Knihovna od verze 1.3 obsahuje funkční blok pro ukládání souborů na FTP serveru. Pro správnou funkci musí server umožňovat práci v pasivním režimu.

8.1 Funkční blok *fbStoreToFtp*



Funkční blok *fbStoreToFtp* slouží k uložení souboru z paměťové karty PLC na FTP server. Od verze knihovny 2.3 blok umožňuje vytvořit neexistující adresáře na FTP serveru a přenášet celé adresáře. Blok pracuje se serverem v pasivním režimu, což znamená, že se PLC aktivně připojuje pro přenos dat na port určený vzdáleným serverem.

Komunikace probíhá přes dvě spojení na ethernetovém kanálu v režimu UNI podle konstant na vstupech *chanCodeCom* a *chanCodeDat*. Spojení pro *chanCodeCom* musí mít následujícími parametry: režim TCP master, délka přijímací a vysílací zóny 255 bytů. Pro *chanCodeDat* – režim TCP master, délka vysílací zóny 255 bytů. Pokud spojení není aktivní nebo nemá správné délky zón, blok indikuje chybu na výstupech *Err* hodnotou TRUE a *ErrId* hodnotou 255 pro *chanCodeCom* a 254 pro *chanCodeDat*.

Adresa FTP serveru se předává na vstupu *IPadr* na vstupu *port* se očekává číslo portu, na kterém server naslouchá (výchozí hodnota pro FTP protokol je 21). Na vstupech *User* a *Pass* se očekávají přihlašovací údaje – uživatelské jméno a heslo. Proměnná se jménem a cestou k souboru nebo adresáři na paměťové kartě PLC se očekává na vstupu *LocalName*. Pokud soubor nebo adresář neexistuje funkční blok vyhlásí na výstupu *ErrId* chybu číslo 3. Proměnná na vstupu *RemotName* udává jméno a cestu, do které bude soubor uložen na serveru. Cesta na FTP serveru od verze knihovny 2.3 již nemusí existovat (ve starších verzích akce skončila v případě neexistující cesty neúspěchem – chyba číslo 2). Pokud na FTP serveru již soubor existuje, je, pokud tomu nebrání přístupová práva, přepsán.

Pro přenos celých adresářů je nutné, aby jak *LocalName* tak *RemotName* obsahoval cestu k adresáři ukončenou oddělovačem /. Pokud tato podmínka není splněna je vyhlášena chyba číslo 7.

Přenos souboru nebo adresáře se zahájí na náběžnou hranu na vstupu *Send*. Během komunikace je nastaven výstup *Busy* na hodnotu TRUE. V případě úspěšného ukon-

čení je nastaven na jeden cyklus výstup *Done*. V případě neúspěchu je nastaven výstup *Err* a *ErrId*, který obsahuje upřesňující číslo chyby.

Od verze knihovny 2.3 je během přenosu souboru na výstupu *CurrFile* jméno souboru na paměťové kartě a na výstupu *Progress* je indikován průběh odesílání souboru v procentech.

Přenos adresářů probíhá včetně podadresářů. Maximální počet vnoření podadresářů jsou 4. Pokud obsahuje adresář větší hloubku vnoření je vyhlášena chyba číslo 5.

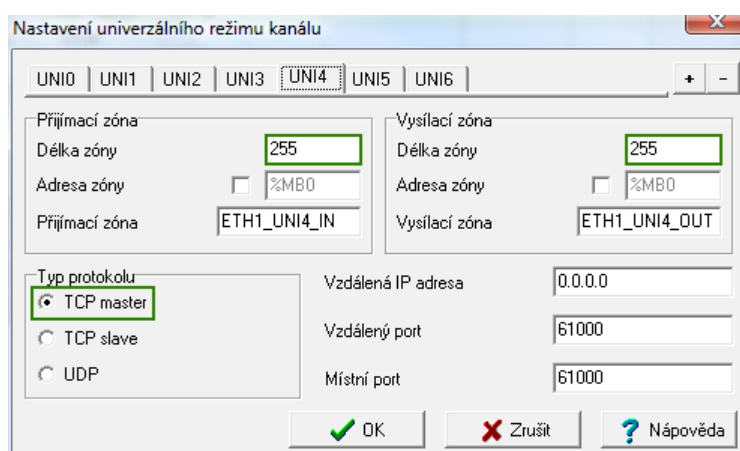
Výstup *State* indikuje aktuální stav komunikace. Na výstupu *ReplyCode* se objevují kódy odpovědí serveru, které lze využít k další diagnostice, pokud je vyhlášena na výstupu *ErrId* chyba číslo 2 – neočekávaná odpověď serveru.

Seznam očekávaných kódu odpovědí serveru pro jednotlivé stavy

Stav	Očekávaný kód odpovědi serveru
fss_Rx220	220
fss_RxUser	331
fss_RxPass	230
fss_RxType	200
fss_RxPasv	227
fss_WaitForOpen	150 nebo 125 (InternetLib 1.7) nebo 550 (InternetLib 2.3) – vyvolá pokus o vytvoření cesty na FTP
fss_RxCreateDir (InternetLib 2.3)	257 nebo 550 pokud adresář již existuje nebo nelze vytvořit
fss_RxComplete	226
fss_RxQuit	221

Bližší informace o významech kódů odpovědí viz [RFC 959](#).

Nastavení příkazového a datového spojení na ethernetovém kanálu v režimu UNI:



Nastavení spojení na ethernetovém kanálu v režimu UNI pro funkční blok *fbStoreToFTP* vstup *chanCodeCom* (příkazové spojení)

Nastavení spojení na ethernetovém kanálu v režimu UNI pro funkční blok fbStoreToFTP vstup chanCodeDat (datové spojení)

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	Send	BOOL R_EDGE	Náběžná hrana zahájí přenos souboru
	IPadr	TIPadr	IP adresa FTP serveru
	port	UINT	Port FTP serveru (výchozí hodnota pro protokol FTP je 21)
	chanCodeCom	UINT	Kód spojení pro příkazy ETH1_uni0, ETH1_uni1,...
	chanCodeDat	UINT	Kód spojení pro data ETH1_uni0, ETH1_uni1,...
VAR_IN_OUT			
	User	STRING	Uživatelské jméno pro přihlášení na server
	Pass	STRING	Uživatelské heslo pro přihlášení na server
	LocalName	STRING	Jméno souboru na paměťové kartě včetně cesty
	RemotName	STRING	Jméno souboru na FTP serveru včetně cesty
VAR_OUTPUT			
	Done	BOOL	Má hodnotu TRUE v okamžiku kdy je úspěšně ukončena komunikace se serverem. Jinak vrací FALSE
	Busy	BOOL	Má hodnotu TRUE po dobu komunikace se serverem
	Err	BOOL	Příznak chyby Pokud poslední operace dopadla úspěšně má hodnotu FALSE, jinak TRUE.

	Proměnná	Typ	Význam
	<i>Errld</i>	USINT	Chybový kód: <i>Errld</i> = 0 operace dopadla úspěšně <i>Errld</i> = 1 vypršel čas pro odpověď serveru <i>Errld</i> = 2 nepodařilo se získat všechna data ze serveru <i>Errld</i> = 3 nelze otevřít soubor na paměťové kartě <i>Errld</i> = 4 jméno souboru na FTP serveru je delší než 80 znaků <i>Errld</i> = 5 příliš mnoho vnoření podadresářů (více jak 4) <i>Errld</i> = 6 adresář pro odeslání neobsahuje žádné soubory <i>Errld</i> = 7 cíl pro adresář musí být opět adresář (LocalName i RemotName musí končit znakem /) <i>Errld</i> = 253 chybné nastavení spojení pro data <i>Errld</i> = 254 nulová adresa webového serveru <i>Errld</i> = 255 chybné nastavení spojení pro příkazy
	<i>ReplyCode</i>	UINT	Kód odpovědi serveru
	<i>DataLen</i>	UINT	Délka přijatého bloku dat
	<i>State</i>	TFtpStoreState	Stav komunikace se serverem (viz enumerace TFtpStoreState)
	<i>CurrFile</i>	STRING	Jméno aktuálně nahrávaného souboru
	<i>Progress</i>	USINT	Průběh nahrávání souboru v procentech

Následující příklad ukazuje použití funkčního bloku *fbStoreToFtp*. Při nastavení proměnné *SendFileToFtp* do logické jedničky dojde k odeslání souboru s obrázkem na FTP server.

```

VAR_GLOBAL
  SendFileToFtp : BOOL;
END_VAR

PROGRAM prgExampleStoreToFtp
  VAR
    NsLookUp      : fbNsLookUpEx;
    StoreToFtp    : fbStoreToFtp;
    FtpIP         : TIPadr;
    FtpName       : STRING := 'foxtrot.howto.cz';
    User          : STRING := 'foxtrot.howto.cz';
    Pass          : STRING := '*****';
    FileName      : STRING := '/WWW/IMAGES/Top.PNG';
    FtpFileName   : STRING := 'Top.PNG';
  END_VAR

  NsLookUp(getIP := SendFileToFtp,
    DnsIP := STRING_TO_IPADR('208.67.222.222'),
    Name := FtpName,
    IP := FtpIP);

  StoreToFtp(Send := NsLookUp.Done, IPadr := FtpIP,
    chanCodeCom := ETH1_uni4, chanCodeDat := ETH1_uni5,
    User := User, Pass := Pass,
    LocalName := FileName,
    RemotName := FtpFileName);
END_PROGRAM

```




teco

Objednávky a informace:

Teco a. s. Havlíčkova 260, 280 58 Kolín 4, tel. 321 737 611, fax 321 737 633

TXV 003 54.01

Výrobce si vyhrazuje právo na změny dokumentace. Poslední aktuální vydání je k dispozici na internetu www.tecomat.com