

Knihovna ComLib

TXV 003 51.01
osmé vydání
červenec 2013
změny vyhrazeny

Historie změn

Datum	Vydání	Popis změn
Březen 2009	1	První vydání
Září 2009	2	Doplněny informace pro ComLib_v13
Listopad 2009	3	Opraven popis <code>fbRecvFrom()</code> a <code>fbSendTo()</code> podle ComLib_v13 (vstupní proměnná <code>channel</code> byla přejmenovaná na <code>chanCode</code>)
Březen 2010	4	Doplněn popis funkce <code>GetChanStat()</code> a datového typu <code>Tuni_STAT</code> Doplněn popis funkcí <code>GetChanSettings()</code> a <code>SetChanSettings()</code> Popis odpovídá ComLib_v14
Srpen 2010	5	Opraven příklad použití konstant pro parametr řízení modemu u sériového komunikačního kanálu (<code>chanSettings.modemControl</code>) Popis odpovídá ComLib_v15
Březen 2011	6	Doplněn popis funkce <code>SetChanSettings()</code> a opraveno formátování tabulek pro help
Leden 2012	7	Doplněn popis funkcí <code>GetDNS_IP()</code> a <code>SetDNS_IP()</code> Popis odpovídá ComLib_v17
Červenec 2013	8	Doplněn popis funkce <code>GetWebServerAccess()</code> a datových typů <code>TWebServerAccess</code> a <code>TWebServerAccessTable</code> Doplněn popis funkčních bloků <code>fbKeepAliveTCP()</code> , <code>fbRecvFromTxt()</code> a <code>fbRecvFromChar()</code> Doplněn popis příznaku <code>link</code> ve struktuře <code>TEthStat</code> Popis odpovídá ComLib_v21

OBSAH

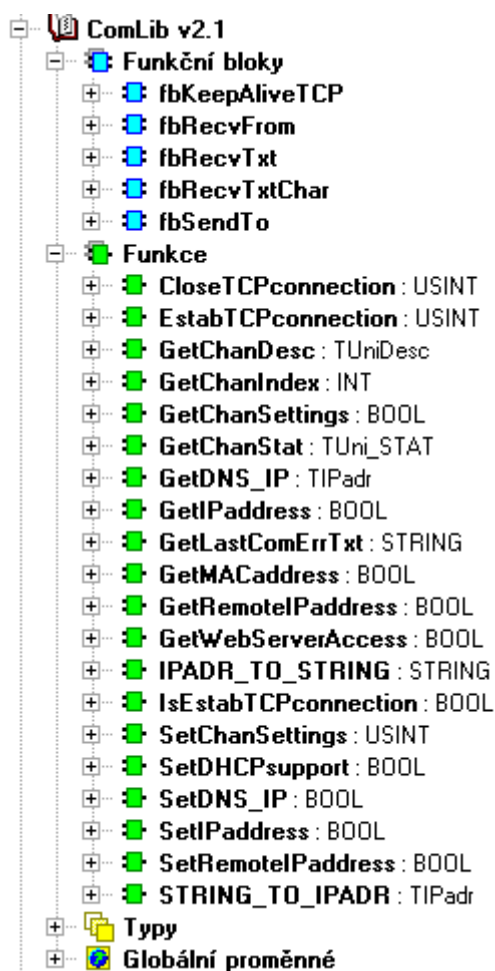
1 Úvod.....	5
2 Datové typy.....	6
2.1 Typ TEthStat.....	7
2.2 Typ TIPadr.....	8
2.3 Typ TLocalEthAdr.....	9
2.4 Typ TRemoteEthAdr.....	10
2.5 Typ TMacAdr.....	11
2.6 Typ TUniDesc.....	12
2.7 Typ TUni_STAT.....	13
2.8 Typ TChanSettings.....	14
2.9 Typ T_SESSION_STATE.....	15
2.10 Typ TWebServerAccess.....	16
2.11 Typ TWebServerAccessTable.....	17
3 Konstanty.....	18
4 Globální proměnné.....	23
5 Funkce	24
5.1 Funkce GetLastComErrTxt.....	26
5.2 Funkce EstabTCPconnection.....	27
5.3 Funkce CloseTCPconnection.....	28
5.4 Funkce IsEstabTCPconnection.....	29
5.5 Funkce SetRemoteIPaddress.....	30
5.6 Funkce GetRemoteIPaddress.....	32
5.7 Funkce SetIPaddress.....	33
5.8 Funkce GetIPaddress.....	35
5.9 Funkce GetMACaddress.....	37
5.10 Funkce SetDHCPsupport.....	38
5.11 Funkce STRING_TO_IPADR.....	40
5.12 Funkce IPADR_TO_STRING.....	41
5.13 Funkce GetChanStat.....	42
5.14 Funkce GetChanSettings.....	43
5.15 Funkce SetChanSettings.....	45
5.16 Funkce GetDNS_IP.....	47
5.17 Funkce SetDNS_IP.....	48
5.18 Funkce GetWebServerAccess.....	50
6 Funkční bloky.....	53

6.1 Funkční blok fbRecvFrom.....	54
6.2 Funkční blok fbSendTo.....	59
6.3 Funkční blok fbKeepAliveTCP.....	65
6.4 Funkční blok fbRecvFromTxt.....	66
6.5 Funkční blok fbRecvFromChar.....	68

1 ÚVOD

Knihovna ComLib je standardně dodávána jako součást programovacího prostředí Mosaic. Knihovna obsahuje funkce a funkční bloky umožňující přijímat a odesílat zprávy komunikačním kanálem PLC. Tím může být buď sériová linka nebo Ethernet.

Následující obrázek ukazuje strukturu knihovny ComLib v prostředí Mosaic



Pokud chceme funkce z knihovny ComLib použít v aplikačním programu PLC, je třeba nejprve přidat tuto knihovnu do projektu. Knihovna je dodávána jako součást instalace prostředí Mosaic od verze v2.0.15.0.

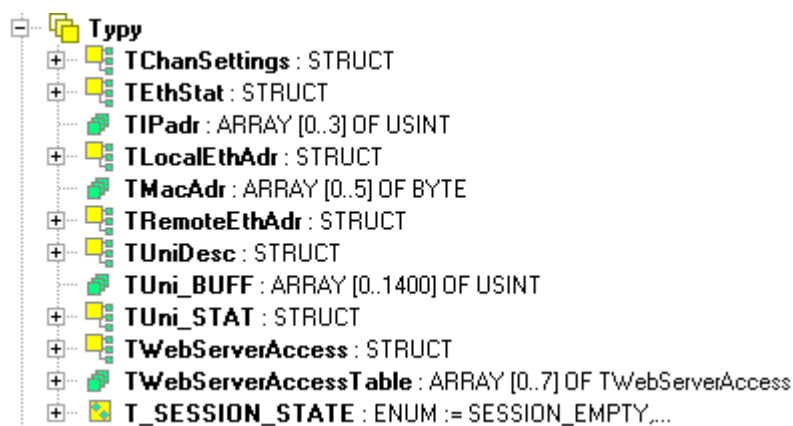
POZOR !!! Knihovna ComLib není podporovaná na systémech TC-650, u systému TC700 knihovnu ComLib nelze použít s procesorovými moduly CP-7002, CP-7003 a CP-7005.

Funkce a funkční bloky knihovny ComLib jsou podporovány v centrálních jednotkách řady K (TC700 CP-7000, CP-7004 a CP-7007, všechny varianty systému Foxtrot) od verze v4.4. Některé funkce vyžadují firmware centrálních jednotek v4.9 nebo vyšší (viz např. Funkce SetDHCPsupport nebo Funkce GetDNS_IP a Funkce SetDNS_IP).

Objednací číslo dokumentace ke knihovně ComLib je je TXV 003 51.

2 DATOVÉ TYPY

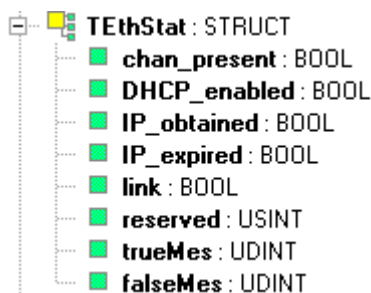
Datové typy v knihovně ComLib lze rozdělit na dvě skupiny. První skupinou datové typy používané interně v knihovně a v druhé skupině jsou datové typy určené pro použití v aplikačním programu.



Stručný popis datových typů udává následující tabulka:

<i>Identifikátor</i>	<i>Typ</i>	<i>Význam</i>
<i>TEthStat</i>	STRUCT	Stavové informace o rozhraní ethernet
<i>TIPadr</i>	ARRAY	IP adresa
<i>TLocalEthAdr</i>	STRUCT	Struktura obsahující IP adresu PLC, masku sítě a adresu brány
<i>TRemoteEthAdr</i>	STRUCT	Struktura obsahující remote IP adresu, remote port a local port
<i>TMacAdr</i>	ARRAY	Pole 6 bytů určené pro uložení MAC adresy
<i>TUniDesc</i>	STRUCT	Popis mapování komunikačního kanálu (používáno pouze interně v knihovně)
<i>TUni_BUFF</i>	ARRAY	Buffer pro vysílaná resp. přijímaná data (používáno pouze interně v knihovně)
<i>TUni_STAT</i>	STRUCT	Stavová zóna komunikačního kanálu
<i>TChanSettings</i>	STRUCT	Nastavení sériového komunikačního kanálu
<i>T_SESSION_STATE</i>	ENUM	Stav jednoho spojení prohlížeče s web serverem PLC
<i>TWebServerAccess</i>	STRUCT	Struktura obsahující informace o jednom uživateli přihlášeném k web serveru PLC
<i>TWebServerAccessTable</i>	ARRAY	Pole obsahující informace o všech uživateli, kteří jsou aktuálně přihlášení k web serveru PLC

2.1 Typ *TEthStat*

Knihovna : *ComLib*

Datový typ *TEthStat* je struktura obsahující informace o rozhraní ethernet. Tuto strukturu mají globální proměnné *ETH1_STAT* a *ETH2_STAT*, které obsahují informace o rozhraní ETH1 a ETH2.

Jednotlivé položky struktury *TEthStat* mají následující význam:

Identifikátor	Typ	Význam
<i>TEthStat</i>	STRUCT	Struktura obsahující informace o rozhraní ethernet
<i>.chan_present</i>	BOOL	Ethernet kanál je přítomný (je osazen)
<i>.DHCP_enabled</i>	BOOL	Povoleno automatické přidělování IP adresy DHCP serverem
<i>.IP_obtained</i>	BOOL	IP adresa byla přidělena DHCP serverem
<i>.IP_expired</i>	BOOL	Platnost automaticky přidělené IP adresy vypršela
<i>.link</i>	BOOL	Ethernet kabel je připojen (platné pouze pro rozhraní ETH1, potřebuje FW v7.6 a vyšší)
<i>.reserved</i>	USINT	Rezerva pro budoucí použití
<i>.trueMes</i>	UDINT	Celkový počet paketů, které byly systémem zpracovány
<i>.falseMes</i>	UDINT	Počet paketů, jejichž zpracování bylo odmítnuto (důvodem může být vadný paket nebo paket s protokolem, který není řídicím systémem podporován)

Viz také Globální proměnné

2.2 Typ *TIPadr*

 **TIPadr** : ARRAY [0..3] OF USINT

Knihovna : *ComLib*

Datový typ *TIPaddr* je pole 4 prvků typu USINT. Používá v případech, kdy je třeba definovat IP adresu nebo masku sítě.

Typ *TIPadr* je v knihovně deklarován následovně:

```
TYPE
  TIPaddr :  ARRAY [0..3] OF USINT;
END_TYPE
```

Prvek s indexem 0 obsahuje první číslo IP adresy, prvek s indexem 3 obsahuje poslední číslo IP adresy.

Příklad použití datového typu *TIPadr*:

```
VAR
  my_IP_addr    : TIPaddr := [192,168,001,010];  // 192.168.1.10
  my_net_mask   : TIPaddr := [255,255,255,000];  // 255.255.255.0
END_VAR
```

Viz také Typ *TLocalEthAdr*, Typ *TRemoteEthAdr*

2.3 Typ *TLocalEthAdr*

Knihovna : *ComLib*

Strukturu typu *TLocalEthAdr* používají funkce *GetIPAddress* a *SetIPAddress* pro předání IP adresy, masky a adresy brány. Všechny položky této struktury jsou typu *TIPAdr*, což je pole se čtyřmi prvky typu *USINT*.

Význam jednotlivých položek struktury *TLocalEthAdr* je následující:

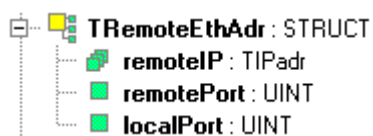
- *IP* IP adresa (např. 192. 168.1.10)
- *IM* IP maska (např. 255.255.255.0)
- *GW* adresa brány (např. 192. 168.1.100)

Příklad inicializace proměnné typu *TLocalEthAdr*:

```
VAR
  new_eth_adr : TLocalEthAdr := ( IP:= [192,168,001,010],
                                   IM:= [255,255,255,000],
                                   GW:= [192,168,001,100] );
END_VAR
```

Viz také Funkce *GetIPAddress*, Funkce *SetIPAddress*

2.4 Typ *TRemoteEthAdr*

Knihovna : *ComLib*

Strukturu typu *TRemoteEthAdr* používá funkce *SetRemoteIPAddress* pro nastavení cílové IP adresy, cílového portu a lokálního portu. Funkci lze použít pro ethernet kanály ETH1 a ETH2, které jsou nastaveny v univerzálním režimu se zapnutým protokolem TCP nebo UDP. Kanál ETH1 je zpravidla umístěn na procesorové jednotce PLC, kanál ETH2 najdeme na komunikační jednotce (např. SC-7102, atd.).

Význam jednotlivých položek struktury *TRemoteEthAdr* je následující:

- *remoteIP* cílová IP adresa (např. 192. 168.001.010)
- *remotePort* cílový port (tj. port, na který je zaslána zpráva TCP protokolem, např. 61000)
- *localPort* zdrojový port (tj. port, ze kterého je odeslána zpráva TCP protokolem, např. 61001)

Příklad inicializace proměnné typu *TRemoteEthAdr*:

```

VAR
  new_eth_adr  : TRemoteEthAdr := ( remoteIP   := [192,168,001,010],
                                     remotePort := 61000,
                                     localPort  := 61001 );
END_VAR

```

Viz také Funkce *GetRemoteIPAddress*, Funkce *SetRemoteIPAddress*

2.5 Typ *TMacAdr*

Knihovna : *ComLib* **TMacAdr** : ARRAY [0..5] OF BYTE

Pole typu *TMacAdr* používá funkce *GetMacAddress*, která zjišťuje MAC adresu ethernet rozhraní.

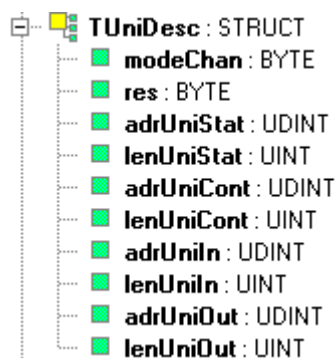
Příklad použití proměnné typu *TMacAdr*:

```
PROGRAM prgTestGetMAC
VAR
  mac_adr   : TMacAdr;
  tmp       : BOOL;
  message    : STRING;
END_VAR

tmp := GetMACaddress( EthChan := ETH1, MacAdr := my_mac_adr);
IF (mac_adr[0] = 0) AND (mac_adr[1] = 16#0A) AND (mac_adr[2] = 16#14) THEN
  message := 'This is Teco device';
END_IF;
END_PROGRAM
```

Viz také Funkce *GetMACaddress*

2.6 Typ *TUniDesc*

Knihovna : *ComLib*

Datový typ *TUniDesc* je struktura popisující mapování komunikačního kanálu, kterou vrací funkce *GetChanDesc()*. To je služební funkce využívaná pro vnitřní účely knihovny.

Typ *TUniDesc* je v knihovně deklarován následovně:

```

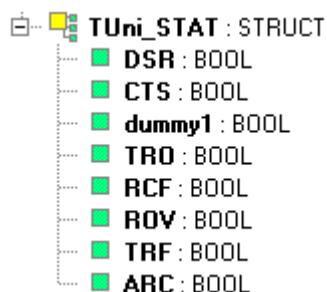
TYPE
  TUniDesc : STRUCT
    modeChan : byte;           // režim kanalu
    res      : byte;           // rezerva
    adrUniStat : uint;         // adresa stavove zony
    lenUniStat : uint;         // delka stavove zony
    adrUniCont : uint;         // adresa ridici zony
    lenUniCont : uint;         // delka ridici zony
    adrUniIn   : uint;         // adresa prijimaci zony
    lenUniIn   : uint;         // delka prijimaci zony
    adrUniOut  : uint;         // adresa vysilaci zony
    lenUniOut  : uint;         // delka vysilaci zony
  END_STRUCT;
END_TYPE

```

Kanál, který je nastaven v univerzálním režimu nastavuje položku *modeChan* ve struktuře *TUniDesc* na hodnotu 5.

Viz také Typ *TLocalEthAdr*, Typ *TRemoteEthAdr*

2.7 Typ *TUni_STAT*

Knihovna : *ComLib*

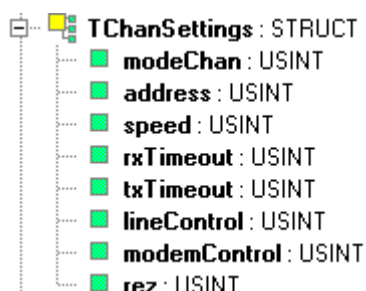
Datový typ *TUni_STAT* je struktura popisující aktuální stav komunikačního kanálu, kterou vrací funkce *GetChanStat()*.

Význam jednotlivých položek struktury *TUni_STAT* je následující:

<i>Identifikátor</i>	<i>Typ</i>	<i>Význam</i>
<i>TUni_STAT</i>	STRUCT	Struktura obsahující informace o aktuálním stavu komunikačního kanálu
<i>.DSR</i>	BOOL	stav signálu DSR
<i>.CTS</i>	BOOL	stav signálu CTS (připravenost k vysílání)
<i>.dummy1</i>	BOOL	rezerva
<i>.TRO</i>	BOOL	vysílací zásobníky jsou plné, zápis další zprávy bude neplatný (log.1)
<i>.RCF</i>	BOOL	přijímací zásobníky jsou zaplněny, dojde ke ztrátě již přijaté zprávy (log.1)
<i>.ROV</i>	BOOL	přetečení (log.1) - přijatá zpráva je delší, než vyhrazená přijímací zóna
<i>.TRF</i>	BOOL	probíhá vysílání, zápis další zprávy bude akceptován až po odvysílání (log.1)
<i>.ARC</i>	BOOL	alternace příjmu- při nově přijaté zprávě dojde ke změně bitu

Viz také Funkce *GetChanStat*

2.8 Typ *TChanSettings*

Knihovna : *ComLib*

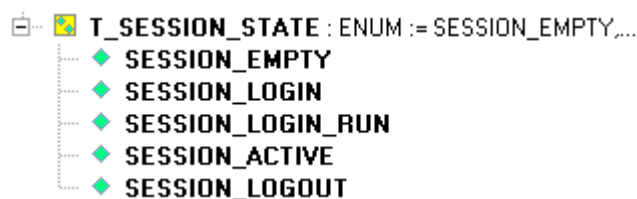
Datový typ *TChanSettings* je struktura popisující aktuální nastavení sériového komunikačního kanálu, kterou nastavuje funkce *GetChanSettings()*. Stejnou strukturu používá funkce *SetChanSettings()*, kterou lze použít ke změně nastavení sériového kanálu.

Význam jednotlivých položek struktury *TChanSettings* je následující:

Identifikátor	Typ	Význam
<i>TChanSettings</i>	STRUCT	Struktura obsahující informace o aktuálním nastavení sériového komunikačního kanálu
<i>.modeChan</i>	USINT	Režim komunikačního kanálu viz konstanty <i>MODE_OFF</i> , <i>MODE_PC</i> , <i>MODE_UNI</i> ,...
<i>.address</i>	USINT	Adresa pro komunikaci
<i>.speed</i>	USINT	Komunikační rychlost viz konstanty <i>BAUD_50</i> , <i>BAUD_100</i> ,...
<i>.rxTimeout</i>	USINT	Timeout při příjmu (minimální doba klidu na lince odpovídající počtu přijatých bytů, po které bude zahájen příjem další zprávy). Po uplynutí této doby je přijatá zpráva považována za celou.
<i>.txTimeout</i>	USINT	Timeout při vysílání (minimální zaručená prodleva mezi dvěma odesílanými zprávami odpovídající počtu vysílaných bytů). Tento parametr zabezpečí, že mezi dvěma vysílanými zprávami bude dodržen klid na lince minimálně této délky.
<i>.lineControl</i>	USINT	Parita, počet bitů, počet stop bitů viz konstanty <i>PARITY_ODD</i> , <i>PARITY_EVEN</i> , ...
<i>.modemControl</i>	USINT	Modemové signály (nastavení signálu RTS) viz konstanty <i>RTS_AUTO</i> , <i>HALF_DUPLEX</i> ,...
<i>.rez</i>	USINT	rezerva

Viz také Funkce *GetChanSettings*, Funkce *SetChanSettings*

2.9 Typ *T_SESSION_STATE*

Knihovna : *ComLib*

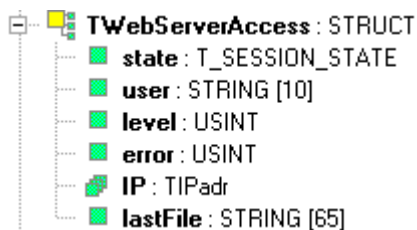
Výčtový typ *T_SESSION_STATE* udává stav spojení prohlížeče s web serverem v PLC. Tento typ je použit pro položku *state* v typu *TWebServerAccess*

Význam jednotlivých *stavů* je následující:

<i>Stav</i>	<i>Význam</i>
<i>SESSION_EMPTY</i>	Spojení je volné
<i>SESSION_LOGIN</i>	Zahájení přihlášení uživatele k web serveru
<i>SESSION_LOGIN_RUN</i>	Probíhá přihlašování uživatele k web serveru
<i>SESSION_ACTIVE</i>	Probíhá komunikace s úspěšně přihlášeným uživatelem Spojení je vyhrazeno pro konkrétního uživatele
<i>SESSION_LOGOUT</i>	Probíhá odlašování uživatele (ukončuje se spojení)

Viz také Typ *TWebServerAccess*, Typ *TWebServerAccessTable*

2.10 Typ *TWebServerAccess*

Knihovna : *ComLib*

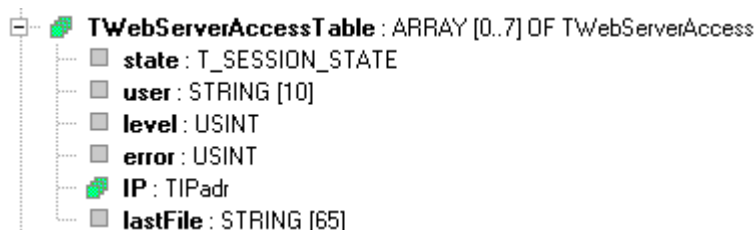
Datový typ *TWebServerAccess* je struktura obsahující informace o jednom uživateli přihlášeném k web serveru PLC prostřednictvím internetového prohlížeče. Tento typ je použit v typu *TWebServerAccessTable*.

Význam jednotlivých položek struktury *TWebServerAccess* je následující:

<i>Identifikátor</i>	<i>Typ</i>	<i>Význam</i>
<i>TWebServerAccess</i>		
<i>.state</i>	ENUM	Stav přihlášení (viz 2.9 Typ <i>T_SESSION_STATE</i>)
<i>.user</i>	STRING[10]	Jméno přihlášeného uživatele
<i>.level</i>	USINT	Úroveň přihlášení
<i>.error</i>	USINT	Kód chyby
<i>.IP</i>	TIPAdr	IP adresa přihlášeného uživatele
<i>.lastFile</i>	STRING[65]	Název posledního souboru, požadovaného prohlížečem

Viz také Typ *T_SESSION_STATE*, Typ *TWebServerAccessTable*

2.11 Typ *TWebServerAccessTable*

Knihovna : *ComLib*

Datový typ *TWebServerAccessTable* je pole struktur *TWebServerAccess* obsahující informace o všech uživateli, kteří jsou aktuálně přihlášení k web serveru PLC prostřednictvím internetového prohlížeče. Toto pole může obsahovat informace o max. 8 uživateli (což je maximální teoretický počet současně připojených uživatelů k web serveru v systému Foxtrot). Prvních 6 položek popisuje uživatele připojené přímo prostřednictvím prohlížeče (z lokální sítě nebo přes internet). Poslední 2 položky v poli *TWebServerAccessTable* obsahují informace o uživateli připojených přes službu TecoRoute.

Datový typ *TWebServerAccessTable* používá Funkce *GetWebServerAccess*

Viz také Typ *T_SESSION_STATE*, Typ *TWebServerAccess*

3 KONSTANTY

V knihovně ComLib jsou definovány následující konstanty:

```

VAR_GLOBAL CONSTANT
...
■ MODE_OFF : USINT := 16#00
■ MODE_PC : USINT := 16#02
■ MODE_UNI : USINT := 16#05
■ MODE_MPC : USINT := 16#06
■ MODE_MDB : USINT := 16#07
■ MODE_PFB : USINT := 16#08
■ BAUD_50 : USINT := 16#01
■ BAUD_100 : USINT := 16#02
■ BAUD_200 : USINT := 16#03
■ BAUD_300 : USINT := 16#04
■ BAUD_600 : USINT := 16#05
■ BAUD_1200 : USINT := 16#06
■ BAUD_2400 : USINT := 16#07
■ BAUD_4800 : USINT := 16#08
■ BAUD_9600 : USINT := 16#0A
■ BAUD_14400 : USINT := 16#0B
■ BAUD_19200 : USINT := 16#0C
■ BAUD_28800 : USINT := 16#0D
■ BAUD_38400 : USINT := 16#0E
■ BAUD_57600 : USINT := 16#10
■ BAUD_76800 : USINT := 16#12
■ BAUD_93750 : USINT := 16#13
■ BAUD_115200 : USINT := 16#14
■ NO_PARITY : USINT := 16#00
■ PARITY_ODD : USINT := 16#08
■ PARITY_EVEN : USINT := 16#18
■ PARITY_0 : USINT := 16#28
■ PARITY_1 : USINT := 16#38
■ SEVEN_BITS : USINT := 16#40
■ EIGHT_BITS : USINT := 16#00
■ ONE_STOP_BIT : USINT := 16#00
■ TWO_STOP_BITS : USINT := 16#80
■ RTS_0 : USINT := 16#00
■ RTS_1 : USINT := 16#02
■ RTS_MAN : USINT := 16#40
■ RTS_AUTO : USINT := 16#80
■ RTS_CTS_AUTO : USINT := 16#C0
■ HALF_DUPLEX : USINT := 16#08
■ ANY_IP : TIPadr := [0]
■ ETH1_uni0 : UINT := 16#07E1
■ ETH1_uni1 : UINT := 16#17E1
■ ETH1_uni2 : UINT := 16#27E1
■ ETH1_uni3 : UINT := 16#37E1
■ ETH1_uni4 : UINT := 16#47E1
■ ETH1_uni5 : UINT := 16#57E1
■ ETH1_uni6 : UINT := 16#67E1
■ ETH1_uni7 : UINT := 16#77E1
...
■ ETH2_uni0 : UINT := 16#07E2
■ ETH2_uni1 : UINT := 16#17E2
■ ETH2_uni2 : UINT := 16#27E2
■ ETH2_uni3 : UINT := 16#37E2
■ ETH2_uni4 : UINT := 16#47E2
■ ETH2_uni5 : UINT := 16#57E2
■ ETH2_uni6 : UINT := 16#67E2
■ ETH2_uni7 : UINT := 16#77E2
■ CH1_uni : UINT := 16#0101
■ CH2_uni : UINT := 16#0202
■ CH3_uni : UINT := 16#0103
■ CH4_uni : UINT := 16#0204
■ CH5_uni : UINT := 16#0105
■ CH6_uni : UINT := 16#0206
■ CH7_uni : UINT := 16#0107
■ CH8_uni : UINT := 16#0208
■ CH9_uni : UINT := 16#0109
■ CH10_uni : UINT := 16#020A
■ ETH1 : USINT := 16#E1
■ ETH2 : USINT := 16#E2
■ ETH3 : USINT := 16#E3
■ ETH4 : USINT := 16#E4
■ SCH1 : USINT := 16#01
■ SCH2 : USINT := 16#02
■ SCH3 : USINT := 16#03
■ SCH4 : USINT := 16#04
■ SCH5 : USINT := 16#05
■ SCH6 : USINT := 16#06
■ SCH7 : USINT := 16#07
■ SCH8 : USINT := 16#08
■ SCH9 : USINT := 16#09
■ SCH10 : USINT := 16#0A
■ COM_OK : USINT := 0
■ COM_ERR1 : USINT := 1
■ COM_ERR2 : USINT := 2
■ COM_ERR3 : USINT := 3
■ COM_ERR4 : USINT := 4
■ COM_ERR5 : USINT := 5
■ COM_ERR6 : USINT := 6
■ COM_ERR7 : USINT := 7
■ COM_ERR8 : USINT := 8
■ COM_ERR16 : USINT := 16#10
■ COM_ERR17 : USINT := 16#11
■ COM_ERR18 : USINT := 16#12
■ COM_ERR19 : USINT := 16#13
■ COM_ERR20 : USINT := 16#14
■ COM_ERR24 : USINT := 16#18
■ COM_ERR25 : USINT := 16#19
■ COM_ERR49 : USINT := 16#31
■ COM_ERR50 : USINT := 16#32
■ COM_ERR64 : USINT := 16#40

```

Konstanty *ETH1_uni0* až *ETH1_uni7* se používají pro specifikaci Ethernet kanálu ve funkcích *EtabTCPconnection*, *CloseTCPconnection*, *SetRemoteIPaddress* a ve funkčních blocích *fbSendTo* a *fbRecvFrom*. Stejný význam mají konstanty *ETH2_uni0* až *ETH2_uni7*, ..., *ETH4_uni0* až *ETH4_uni7*.

Identifikátor	Typ	Hodnota	Význam
<i>ETH1_uni0</i>	UINT	16#07E1	ethernet kanál ETH1, spojení uni0
<i>ETH1_uni1</i>	UINT	16#17E1	ethernet kanál ETH1, spojení uni1
<i>ETH1_uni2</i>	UINT	16#27E1	ethernet kanál ETH1, spojení uni2
<i>ETH1_uni3</i>	UINT	16#37E1	ethernet kanál ETH1, spojení uni3
<i>ETH1_uni4</i>	UINT	16#47E1	ethernet kanál ETH1, spojení uni4
<i>ETH1_uni5</i>	UINT	16#57E1	ethernet kanál ETH1, spojení uni5
<i>ETH1_uni6</i>	UINT	16#67E1	ethernet kanál ETH1, spojení uni6
<i>ETH1_uni7</i>	UINT	16#77E1	ethernet kanál ETH1, spojení uni7
<i>ETH2_uni0</i>	UINT	16#07E2	ethernet kanál ETH2, spojení uni0
<i>ETH2_uni1</i>	UINT	16#17E2	ethernet kanál ETH2, spojení uni1
<i>ETH2_uni2</i>	UINT	16#27E2	ethernet kanál ETH2, spojení uni2
<i>ETH2_uni3</i>	UINT	16#37E2	ethernet kanál ETH2, spojení uni3
<i>ETH2_uni4</i>	UINT	16#47E2	ethernet kanál ETH2, spojení uni4
<i>ETH2_uni5</i>	UINT	16#57E2	ethernet kanál ETH2, spojení uni5
...	...	...	...
<i>ETH4_uni7</i>	UINT	16#77E4	ethernet kanál ETH2, spojení uni7

Podobně konstanty *CH1_uni* až *CH10_uni* určují konkrétní sériový kanál při volání funkčních bloků *fbSendTo* a *fbRecvFrom*.

Identifikátor	Typ	Hodnota	Význam
<i>CH1_uni</i>	UINT	16#0101	sériový kanál CH1, režim uni
<i>CH2_uni</i>	UINT	16#0202	sériový kanál CH2, režim uni
<i>CH3_uni</i>	UINT	16#0103	sériový kanál CH3, režim uni
<i>CH4_uni</i>	UINT	16#0204	sériový kanál CH4, režim uni
<i>CH5_uni</i>	UINT	16#0105	sériový kanál CH5, režim uni
<i>CH6_uni</i>	UINT	16#0206	sériový kanál CH6, režim uni
<i>CH7_uni</i>	UINT	16#0107	sériový kanál CH7, režim uni
<i>CH8_uni</i>	UINT	16#0208	sériový kanál CH8, režim uni
<i>CH9_uni</i>	UINT	16#0109	sériový kanál CH9, režim uni
<i>CH10_uni</i>	UINT	16#020A	sériový kanál CH10, režim uni

Konstanty *ETH1* až *ETH4* slouží k výběru ethernet rozhraní ve funkcích *SetIPAddress*, *GetIPAddress* a *GetMACAddress*.

Identifikátor	Typ	Hodnota	Význam
<i>ETH1</i>	USINT	16#E1	ethernet rozhraní na centrální jednotce PLC
<i>ETH2</i>	USINT	16#E2	ethernet rozhraní na komunikačním modulu (pro systém TC700 např. SC-7102)
<i>ETH3</i>	USINT	16#E3	ethernet rozhraní na komunikačním modulu (pro systém TC700 např. SC-7104)
<i>ETH4</i>	USINT	16#E4	ethernet rozhraní na komunikačním modulu (pro systém TC700 např. SC-7104)

Podobně konstanty *SCH1* až *SCH10* určují konkrétní sériový kanál při volání funkcí *GetChanSettings* a *SetChanSettings*.

Identifikátor	Typ	Hodnota	Význam
<i>SCH1</i>	USINT	16#01	sériový kanál CH1
<i>SCH2</i>	USINT	16#02	sériový kanál CH2
<i>SCH3</i>	USINT	16#03	sériový kanál CH3
<i>SCH4</i>	USINT	16#04	sériový kanál CH4
<i>SCH5</i>	USINT	16#05	sériový kanál CH5
<i>SCH6</i>	USINT	16#06	sériový kanál CH6
<i>SCH7</i>	USINT	16#07	sériový kanál CH7
<i>SCH8</i>	USINT	16#08	sériový kanál CH8
<i>SCH9</i>	USINT	16#09	sériový kanál CH9
<i>SCH10</i>	USINT	16#0A	sériový kanál CH10

Konstanty *COM_OK*, *COM_ERR1* až *COM_ERR64* jsou návratové hodnoty, které vrací funkční bloky *fbSendTo* a *fbRecvFrom* ve výstupní proměnné *error*. Význam jednotlivých konstant je následující:

Identifikátor	Typ	Hodnota	Význam
<i>COM_OK</i>	USINT	0	komunikace bez chyby
<i>COM_ERR1</i>	USINT	1	požadovaný kanál není nastaven v univerzálním režimu
<i>COM_ERR2</i>	USINT	2	příliš mnoho vysílaných dat, resp. příliš malý buffer kanálu v uni režimu
<i>COM_ERR3</i>	USINT	3	příliš mnoho přijatých dat, resp. přijatá data se nevejdou do specifikované proměnné
<i>COM_ERR4</i>	USINT	4	chybné číslo komunikačního kanálu
<i>COM_ERR5</i>	USINT	5	předchozí zpráva nebyla ještě odeslána

			(vysílací zásobníky jsou plné)
<i>COM_ERR6</i>	USINT	6	nulová délka vysílaných dat
<i>COM_ERR16</i>	USINT	16	chybný počáteční znak
<i>COM_ERR17</i>	USINT	17	chyba parity
<i>COM_ERR18</i>	USINT	18	překročena maximální délka zprávy
<i>COM_ERR19</i>	USINT	19	chybný druhý byte potvrzení
<i>COM_ERR20</i>	USINT	20	chybný druhý byte koncového znaku
<i>COM_ERR24</i>	USINT	24	chyba kontrolního součtu
<i>COM_ERR25</i>	USINT	25	chybný koncový znak
<i>COM_ERR49</i>	USINT	49	chybná délka vysílaných dat
<i>COM_ERR50</i>	USINT	50	nulová délka vysílaných dat
<i>COM_ERR64</i>	USINT	64	nedodržen timeout
<i>COM_ERRc6</i>	USINT	198	sériový kanál není v požadovaném režimu

Konstanty *MODE_xx* slouží k určení režimu sériového komunikačního kanálu (viz *TchanSettings.modeChan*)

Identifikátor	Typ	Hodnota	Význam
<i>MODE_OFF</i>	USINT	16#01	Žádný režim, kanál není obsluhován
<i>MODE_PC</i>	USINT	16#02	Režim PC
<i>MODE_UNI</i>	USINT	16#05	Režim uni
<i>MODE_MPC</i>	USINT	16#06	Režim MPC
<i>MODE_MDB</i>	USINT	16#07	Režim Modbus (slave)
<i>MODE_PFB</i>	USINT	16#08	Režim Profibus DP (master)

Konstanty *BAUD_xx* slouží k určení rychlosti sériového komunikačního kanálu (viz *TchanSettings.speed*)

Identifikátor	Typ	Hodnota	Význam
<i>BAUD_50</i>	USINT	16#01	Rychlost přenosu 50 Baud
<i>BAUD_100</i>	USINT	16#02	Rychlost přenosu 100 Baud
<i>BAUD_200</i>	USINT	16#03	Rychlost přenosu 200 Baud
<i>BAUD_300</i>	USINT	16#04	Rychlost přenosu 300 Baud
<i>BAUD_600</i>	USINT	16#05	Rychlost přenosu 600 Baud
<i>BAUD_1200</i>	USINT	16#06	Rychlost přenosu 1200 Baud
<i>BAUD_2400</i>	USINT	16#07	Rychlost přenosu 2400 Baud
<i>BAUD_4800</i>	USINT	16#08	Rychlost přenosu 4800 Baud
<i>BAUD_9600</i>	USINT	16#0A	Rychlost přenosu 9600 Baud

<i>BAUD_14400</i>	USINT	16#0B	Rychlost přenosu 14400 Baud
<i>BAUD_19200</i>	USINT	16#0C	Rychlost přenosu 19200 Baud
<i>BAUD_28800</i>	USINT	16#0D	Rychlost přenosu 28800 Baud
<i>BAUD_38400</i>	USINT	16#0E	Rychlost přenosu 38400 Baud
<i>BAUD_57600</i>	USINT	16#10	Rychlost přenosu 57600 Baud
<i>BAUD_76800</i>	USINT	16#12	Rychlost přenosu 76800 Baud
<i>BAUD_93750</i>	USINT	16#13	Rychlost přenosu 93750 Baud
<i>BAUD_11500</i>	USINT	16#14	Rychlost přenosu 11500 Baud

Následující konstanty jsou určeny pro parametr řízení linky u sériového komunikačního kanálu (viz *TchanSettings.lineControl*). Výsledné nastavení se dosahuje logickým součtem požadovaných parametrů. Například *chanSettings.lineControl := EIGHT_BITS OR PARITY_EVEN OR ONE_STOP_BIT*.

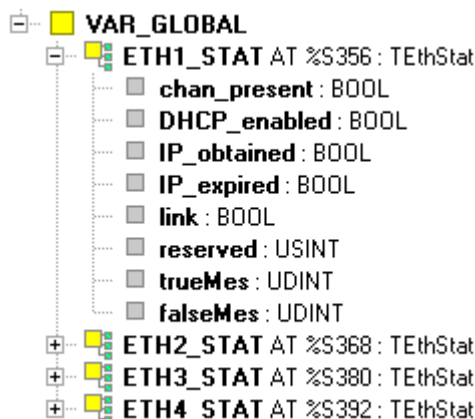
Identifikátor	Typ	Hodnota	Význam
<i>NO_PARITY</i>	USINT	16#00	Žádná parita
<i>PARITY_ODD</i>	USINT	16#08	Lichá parita
<i>PARITY_EVEN</i>	USINT	16#18	Sudá parita
<i>PARITY_0</i>	USINT	16#28	Parita trvale 0
<i>PARITY_1</i>	USINT	16#38	Parita trvale 1
<i>SEVEN_BITS</i>	USINT	16#40	Sedm bitů na znak
<i>EIGHT_BITS</i>	USINT	16#00	Osm bitů na znak
<i>ONE_STOP_BIT</i>	USINT	16#00	Jeden STOP bit
<i>TWO_STOP_BIT</i>	USINT	16#80	Dva STOP bity

Konstanty uvedené v následující tabulce jsou určeny pro parametr řízení modemu u sériového komunikačního kanálu (viz *TchanSettings.modemControl*). Výsledné nastavení se dosahuje logickým součtem požadovaných parametrů. Například *chanSettings.modemControl := RTS_AUTO OR HALF_DUPLEX*.

Identifikátor	Typ	Hodnota	Význam
<i>RTS_0</i>	USINT	16#00	Signál RTS trvale 0
<i>RTS_1</i>	USINT	16#02	Signál RTS trvale 1
<i>RTS_MAN</i>	USINT	16#40	Signál RTS ovládaný z programu PLC
<i>RTS_AUTO</i>	USINT	16#80	Signál RTS automaticky (podle vysílání)
<i>RTS_CTS_AUTO</i>	USINT	16#C0	Signály RTS a CTS automaticky
<i>HALF_DUPLEX</i>	USINT	16#08	Poloviční duplex (během vysílání je přijímač vypnutý)

4 GLOBÁLNÍ PROMĚNNÉ

V knihovně ComLib jsou definovány následující globální proměnné:



Globální proměnné *ETH1_STAT*, ..., *ETH4_STAT* obsahují informace o stavu ethernet rozhraní ETH1 až ETH4. Tyto proměnné jsou typu *TEthStat*, jež je rovněž definován v knihovně.

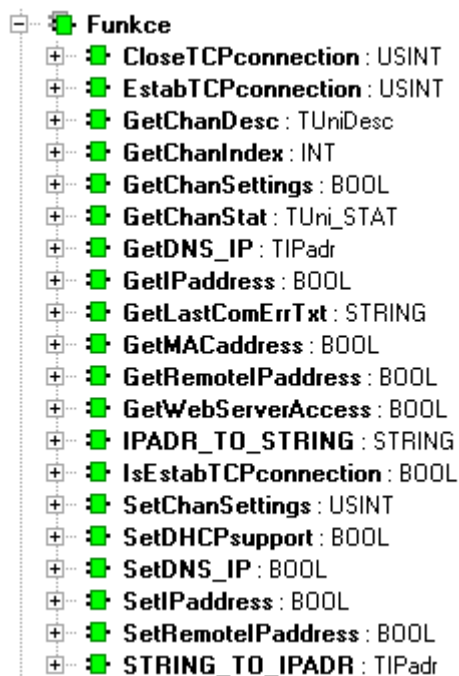
Jednotlivé položky struktury *TEthStat* mají následující význam:

Identifikátor	Typ	Význam
<i>TEthStat</i>	STRUCT	Struktura obsahující informace o rozhraní ethernet
<i>.chan_present</i>	BOOL	Ethernet kanál je přítomný (je osazen)
<i>.DHCP_enabled</i>	BOOL	Povoleno automatické přidělování IP adresy DHCP serverem
<i>.IP_obtained</i>	BOOL	IP adresa byla přidělena DHCP serverem
<i>.IP_expired</i>	BOOL	Platnost automaticky přidělené IP adresy vypršela
<i>.link</i>	BOOL	Ethernet kabel je připojen (platné pouze pro rozhraní ETH1, potřebuje FW v7.6 a vyšší)
<i>.reserved</i>	USINT	Rezerva pro budoucí použití
<i>.trueMes</i>	UDINT	Celkový počet paketů, které byly systémem zpracovány
<i>.falseMes</i>	UDINT	Počet paketů, jejichž zpracování bylo odmítnuto (důvodem může být vadný paket nebo paket s protokolem, který není řídicím systémem podporován)

Viz také Typ *TEthStat*

5 FUNKCE

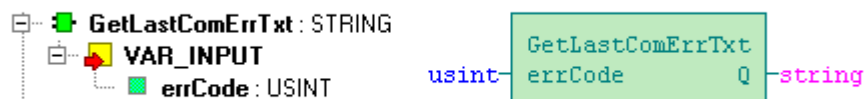
Knihovna ComLib obsahuje následující funkce:



<i>Funkce</i>	<i>Popis</i>
<i>GetChanDesc</i>	služební funkce, využívaná vnitřně v knihovně, která vrací popisovač komunikačního kanálu
<i>GetChanIndex</i>	služební funkce, využívaná vnitřně v knihovně, která vrací index komunikačního kanálu
<i>GetLastComErrTxt</i>	funkce vrátí text chyby vzniklé při komunikaci
<i>EstabTCPconnection</i>	funkce naváže TCP spojení
<i>CloseTCPconnection</i>	funkce ukončí TCP spojení
<i>IsEstabTCPconnection</i>	funkce otestuje, je-li navázáno TCP spojení
<i>SetRemoteIPaddress</i>	funkce nastaví cílovou IP adresu, remote port a local port pro daný ethernet kanál
<i>GetRemoteIPaddress</i>	funkce vrátí aktuální cílovou IP adresu, remote port a local port pro daný ethernet kanál
<i>GetIPAddress</i>	zjistí aktuální IP adresu, masku a adresu brány
<i>SetIPAddress</i>	nastaví novou IP adresu, masku a adresu brány
<i>GetMACaddress</i>	zjistí MAC adresu zadaného ethernet rozhraní
<i>SetDHCPsupport</i>	zapne podporu pro automatické přidělování IP adresy DHCP serverem
<i>STRING_TO_IPADR</i>	konverze IP adresy z řetězce do struktury IPadr
<i>IPADR_TO_STRING</i>	konverze IP adresy ze struktury IPadr na řetězec

<i>Funkce</i>	<i>Popis</i>
<i>GetChanStat</i>	funkce vrátí status komunikačního kanálu
<i>GetChanSettings</i>	funkce vrátí nastavení sériového komunikačního kanálu (komunikační rychlost, timeouty, atd.)
<i>SetChanSettings</i>	funkce nastaví parametry sériového komunikačního kanálu (komunikační rychlost, timeouty, atd.)
<i>GetDNS_IP</i>	Načte aktuálně nastavenou IP adresu DNS serveru
<i>SetDNS_IP</i>	Nastaví novou IP adresu DNS serveru
<i>GetWebServerAccess</i>	Vrátí informace o uživateli připojených k web serveru PLC

5.1 Funkce *GetLastComErrTxt*

Knihovna : *ComLib*

Funkce *GetLastComErrTxt* vrátí text popisující chybu vzniklou při komunikaci. Vstupním parametrem funkce je kód chyby, kterou vrací funkční bloky *fbSendTo* a *fbRecvFrom*.

Tato funkce je podporována na centrálních jednotkách řady K (TC700 CP-7004, Foxtrot) od verze v4.4.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>errCode</i>	USINT	Chybový kód hlášený funkčním blokem typu <i>fbSendTo</i> nebo <i>fbRecvFrom</i>
GetLastComErrTxt			
	Návratová hodnota	STRING	Popis chyby komunikace

Příklad programu s voláním funkce *GetLastComErrTxt* :

```

PROGRAM ExampleGetLastComErrTxt
VAR
    RecvFromCH1 : fbRecvFrom;
    rxBuf        : STRING[100];
    errMsg       : STRING;
END_VAR

// receiving
RecvFromCH1( rq      := TRUE, chanCode := CH1_uni,
              lenRx   := 100, data      := void(rxBuf) );

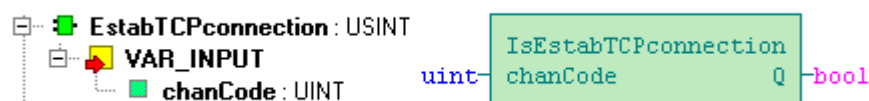
if RecvFromCH1.mesRec then
    // new message received
    if RecvFromCH1.error = 0 then
        // process new message
        // ...
    else
        // show error as a text
        errMsg := GetLastComErrTxt( RecvFromCH1.error );
    end if;
end if;
END_PROGRAM

```

Viz také Funkční blok *fbRecvFrom*, Funkční blok *fbSendTo*

5.2 Funkce EstabTCPconnection

Knihovna : ComLib



Funkce *EstabTCPconnection* nastartuje proces navazování TCP spojení pokud je předchozí spojení ukončeno. Vstupním parametrem funkce je kód komunikačního kanálu (*ETH1_uni0*, ..., *ETH2_uni7*). Funkce vrací chybový kód. Pokud je vše v pořádku, funkce vrátí *COM_OK* (0, žádná chyba). Funkce má smysl pouze v případě ethernet kanálu *ETH1* nebo *ETH2*, který musí být nastaven do režimu uni – master TCP. Navazování TCP spojení je proces, při kterém si klient (PLC) a server vymění synchronizační rámce TCP protokolem. Doba navazování spojení závisí na řadě podmínek (např. jestli se komunikuje v rámci lokální sítě, atd.). Informaci o tom, jestli se spojení podařilo navázat, lze získat pomocí funkce *IsEstabTCPconnection*.

Tato funkce je podporována na centrálních jednotkách řady K (TC700 CP-7004, Foxtrot) od verze v4.4.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	chanCode	USINT	Výběr komunikačního kanálu (<i>ETH1_uni0</i> , ..., <i>ETH1_uni7</i> , <i>ETH2_uni0</i> , ..., <i>ETH2_uni7</i>)
EstabTCPconnection			
	Návratová hodnota	USINT	0 pokud nedojde k chybě (<i>COM_OK</i>) kód chyb v ostatních případech (<i>COM_ERR1</i> , ..., <i>COM_ERR64</i>)

Příklad programu s voláním funkce *EstabTCPconnection* :

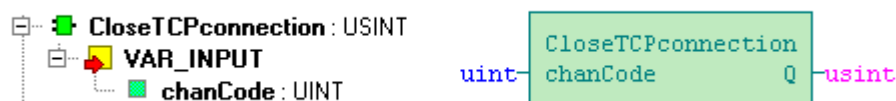
```
PROGRAM prgExampleEstabTCPcon
VAR
END_VAR

IF NOT IsEstabTCPconnection(chanCode := ETH1_UNI0) THEN
    EstabTCPconnection(chanCode := ETH1_UNI0);
END_IF;
END_PROGRAM
```

Viz také Funkce CloseTCPconnection, Funkce IsEstabTCPconnection

5.3 Funkce CloseTCPconnection

Knihovna : ComLib



Funkce *CloseTCPconnection* ukončí TCP spojení. Vstupním parametrem funkce je kód komunikačního kanálu (*ETH1_uni0*, ..., *ETH2_uni7*). Funkce vrací chybový kód. Pokud je vše v pořádku, funkce vrátí *COM_OK* (0, žádná chyba). Funkce má smysl pouze v případě ethernet kanálu ETH1 nebo ETH2, který musí být nastaven do režimu uni – master TCP. Informaci o tom, jestli se spojení podařilo ukončit, lze získat pomocí funkce *IsEstabTCPconnection*.

Tato funkce je podporována na centrálních jednotkách řady K (TC700 CP-7004, Foxtrot) od verze v4.4.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	chanCode	USINT	Výběr komunikačního kanálu (ETH1_uni0, ..., ETH1_uni7, ETH2_uni0, ..., ETH2_uni7)
CloseTCPconnection			
	Návratová hodnota	USINT	0 pokud nedojde k chybě (COM_OK) kód chyb v ostatních případech (COM_ERR1, ..., COM_ERR64)

Příklad programu s voláním funkce *CloseTCPconnection* :

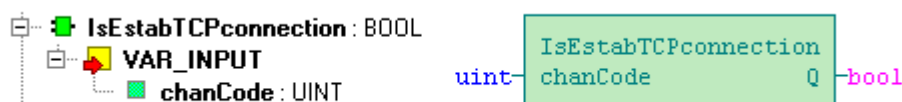
```
PROGRAM prgExampleIsEstabTCPcon
VAR
END_VAR

IF IsEstabTCPconnection(chanCode := ETH1_UNI0) THEN
  CloseTCPconnection(chanCode := ETH1_UNI0);
END_IF;
END_PROGRAM
```

Viz také Funkce EstabTCPconnection, Funkce IsEstabTCPconnection

5.4 Funkce IsEstabTCPconnection

Knihovna : ComLib



Funkce *IsEstabTCPconnection* otestuje, je-li navázáno TCP spojení. Vstupním parametrem funkce je kód komunikačního kanálu (*ETH1_uni0*, ..., *ETH2_uni7*). Funkce vrací hodnotu TRUE je-li spojení navázáno, je-li spojení ukončeno tak vrací hodnotu FALSE. Komunikační kanál musí být nastaven do režimu uni, typ protokolu TCP master nebo TCP slave.

Tato funkce je podporována na centrálních jednotkách řady K (TC700 CP-7004, Foxtrot) od verze v4.4.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	chanCode	USINT	Výběr komunikačního kanálu (ETH1_uni0, ..., ETH1_uni7, ETH2_uni0, ..., ETH2_uni7)
IsEstabTCPconnection			
	Návratová hodnota	BOOL	TRUE pokud je spojení navázáno FALSE v ostatních případech

Příklad programu s voláním funkce *IsEstabTCPconnection*:

```

PROGRAM prgExampleIsEstabTCPcon
VAR
END_VAR

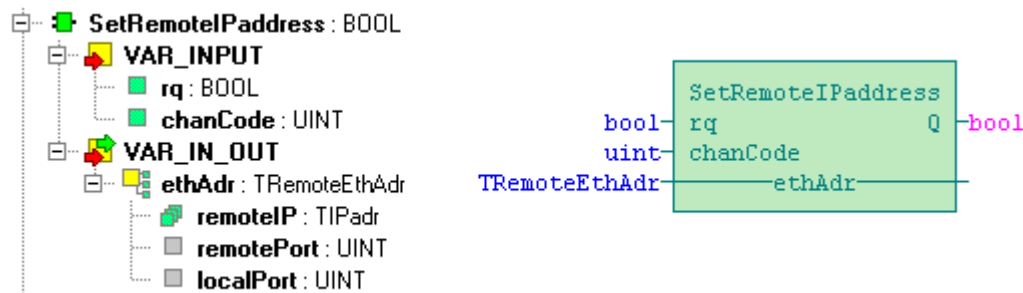
IF IsEstabTCPconnection(chanCode := ETH1_UNI0) THEN
    CloseTCPconnection(chanCode := ETH1_UNI0);
END_IF;
END_PROGRAM

```

Viz také Funkce CloseTCPconnection, Funkce EstabTCPconnection

5.5 Funkce SetRemoteIPAddress

Knihovna : ComLib



Funkce *SetRemoteIPAddress* nastaví vzdálenou IP adresu, číslo vzdáleného portu a číslo lokálního portu, pokud má vstupní parametr *rq* hodnotu TRUE. V parametru *chanCode* se funkci předává kód komunikačního kanálu (*ETH1_uni0.*, ..., *ETH1_uni7*). Funkce vrací hodnotu TRUE, pokud se podaří uplatnit požadavek na nové nastavení.

Parametr *remoteIP* udává IP adresu stanice, se kterou se bude komunikovat. Parametr *remotePort* udává číslo portu, na který budou zprávy odesílány. Parametr *localPort* udává, na kterém portu se budou zprávy přijímat.

Pokud je komunikační kanál nastaven na komunikaci UDP protokolem, nová IP adresa a porty se nastaví v nejbližší otočce cyklu PLC. Poté jsou všechny UDP zprávy vysílané komunikačním kanálem směrovány na novou IP adresu a nový vzdálený port. Mezi nastavením nové cílové IP adresy a vysláním zprávy na tuto adresu musí proběhnout alespoň jeden cykl PLC. Také příjem zpráv bude probíhat pouze od stanice, jejíž IP adresa odpovídá nově nastavené IP adrese a která ve zprávách používá odpovídající čísla portů.

Pokud je *remoteIP* adresa 0.0.0.0, kanál přijme zprávu od stanice s libovolnou IP adresou za podmínky, že port, na který je zpráva směrována odpovídá číslu zadanému v parametru *localPort*. V okamžiku příjmu zprávy se parametr *remoteIP* nastaví na IP adresu stanice, od které byla přijata zpráva. Taktéž parametr *remotePort* změní hodnotu podle čísla portu, ze kterého byla zpráva odeslána. Takže po příjmu UDP paketu je komunikační kanál nastaven tak, aby bylo možno odpovědět na přijatou zprávu.

Pokud je komunikační kanál nastaven na komunikaci TCP protokolem, novou IP adresu a porty lze nastavit pouze za podmínky, že je spojení uzavřeno. Nelze tedy změnit nastavení pokud je navázané TCP spojení. Při změně nastavení je třeba nejprve ukončit spojení. Další chování je stejné jako v případě UDP protokolu.

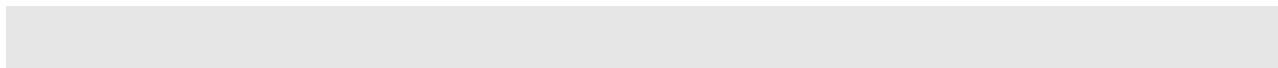
Vysílání zpráv na adresu 0.0.0.0 není přípustné.

Tato funkce je podporována na centrálních jednotkách řady K (TC700 CP-7004, Foxtrot) od verze v4.4. Funkce je podporována pouze pro rozhraní ETH1.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>rq</i>	BOOL	Žádost o nastavení nové IP adresy, se kterou se bude komunikovat. Náběžná hrana na tomto vstupu způsobí zahájení změny IP adresy (pokud je nová IP adresa jiná než aktuální IP adresa)
	<i>chanCode</i>	USINT	Výběr komunikačního kanálu (ETH1_uni0, ..., ETH1_uni7)
VAR_IN_OUT			
	<i>EthAdr</i>	TRemoteEthAdr	IP adresa vzdáleného zařízení a čísla portů, přes které probíhá komunikace
	<i>.remoteIP</i>	TIPadr	IP adresa zařízení, se kterým se komunikuje
	<i>.remotePort</i>	UINT	Číslo portu, na který budou odesílány zprávy resp. ze kterého se přijímají zprávy
	<i>.localPort</i>	UINT	Číslo portu, ze kterého se odesílají zprávy resp. na kterém jsou přijímány zprávy
SetRemoteIPAddress			
	<i>Návratová hodnota</i>	BOOL	TRUE pokud se podaří nastavit vzdálenou IP adresu. Jinak FALSE

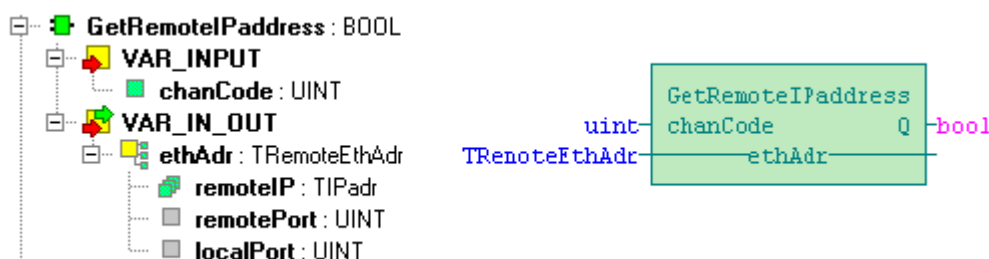
Příklad programu s voláním funkce *SetRemoteIPAddress* :



Viz také Funkce *GetRemoteIPAddress*

5.6 Funkce GetRemoteIPAddress

Knihovna : ComLib



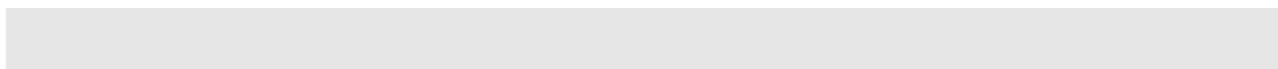
Funkce *GetRemoteIPAddress* vrátí aktuálně nastavenou vzdálenou IP adresu, číslo vzdáleného portu a číslo lokálního portu.

Tato funkce je podporována na centrálních jednotkách řady K (TC700 CP-7004, Foxtrot) od verze v4.4. Funkce je podporovaná pouze pro rozhraní ETH1.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>chanCode</i>	USINT	Výběr komunikačního kanálu (ETH1_uni0, ..., ETH1_uni7)
VAR_IN_OUT			
	<i>EthAdr</i>	TRemoteEthAdr	IP adresa vzdáleného zařízení a čísla portů, přes které probíhá komunikace
	<i>.remoteIP</i>	TIPadr	IP adresa zařízení, se kterým se komunikuje
	<i>.remotePort</i>	UINT	Číslo portu, na který budou odesílány zprávy resp. ze kterého se přijímají zprávy
	<i>.localPort</i>	UINT	Číslo portu, ze kterého se odesílají zprávy resp. na kterém jsou přijímány zprávy
GetRemoteIPAddress			
	Návratová hodnota	BOOL	TRUE pokud se podaří načíst vzdálenou IP adresu. Jinak FALSE

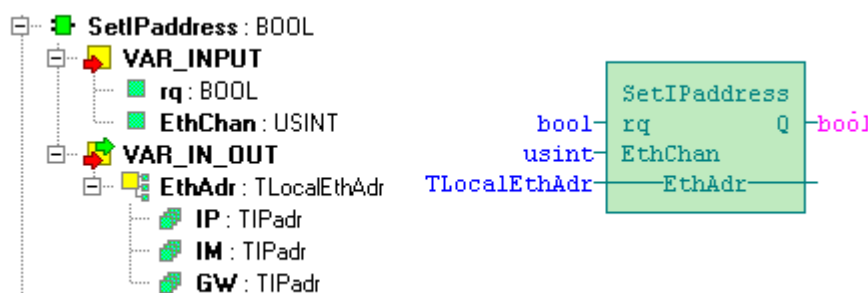
Příklad programu s voláním funkce *GetRemoteIPAddress* :



Viz také Funkce *SetRemoteIPAddress*

5.7 Funkce SetIPAddress

Knihovna : ComLib



Funkce *SetIPAddress* nastaví novou lokální IP adresu, IP masku a adresu brány. Nové hodnoty jsou očekávány v proměnné *EthAdr*, která je typu *TLocalEthAdr*. Nastavení nové adresy se provede na náběžnou hrany vstupní proměnné *rq*. Nastavení bude trvat několik cyklů PLC a vrátí hodnotu TRUE ve chvíli, kdy se nastaví nová IP adresa. Všechny komunikace s původní IP adresou budou zrušeny. Zároveň je zrušen požadavek na automatické získávání IP adresy od DHCP serveru (pokud byl zapnutý). Informaci o aktuálním stavu ethernet rozhraní lze kdykoliv získat v globální proměnné *ETH1_STAT* resp. *ETH2_STAT*.

Tato funkce je podporována na centrálních jednotkách řady K (TC700 CP-7004, Foxtrot) od verze v4.4.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>rq</i>	BOOL	Žádost o nastavení nové IP adresy Náběžná hrana na tomto vstupu způsobí zahájení změny IP adresy (pokud je nová IP adresa jiná než aktuální IP adresa)
	<i>EthChan</i>	USINT	Specifikace ethernet kanálu (ETH1 pro Ethernet na centrální jednotce, ETH2 pro Ethernet na komunikačním modulu)
VAR_IN_OUT			
	<i>EthAdr</i>	TLocalEthAdr	IP adresa, maska a brána
	<i>.IP</i>	TIPadr	IP adresa
	<i>.IM</i>	TIPadr	IP maska
	<i>.GW</i>	TIPadr	Adresa brány
SetIPAddress			
	Návratová hodnota	BOOL	TRUE pokud se podaří nastavit novou IP adresu. Jinak FALSE

Příklad programu s voláním funkce *SetIPAddress* :

```
PROGRAM prgTestSetIP
VAR
  old_eth_adr   : TLocalEthAdr;
  new_eth_adr   : TLocalEthAdr;
  set, res, tmp : BOOL;
END_VAR

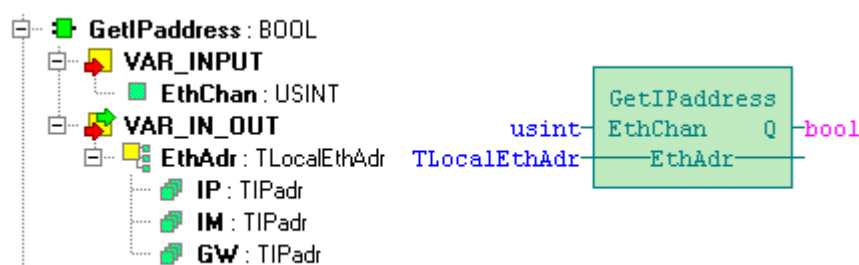
// read actual IP
tmp := GetIPAddress( EthChan := ETH1, EthAdr := old_eth_adr);
if new_eth_adr.IP[0] = 0 then
  new_eth_adr := old_eth_adr;           // use as init value for new adr
end_if;

// set new IP address if rq = TRUE
res := SetIPAddress( rq := set, EthChan := ETH1, EthAdr := new_eth_adr);
if res then set := FALSE; end_if;
END_PROGRAM
```

Viz také Funkce GetIPAddress, Typ TLocalEthAdr, Funkce SetDHCPsupport

5.8 Funkce GetIPAddress

Knihovna : ComLib



Funkce *GetIPAddress* vrátí aktuální IP adresu, IP masku a adresu brány pro zadaný ethernet kanál. Tyto hodnoty jsou uloženy do proměnné *EthAdr*, která má strukturu typu *TLocalEthAdr*. Deklarace této struktury je součástí knihovny ComLib.

Tato funkce je podporována na centrálních jednotkách řady K (TC700 CP-7004, Foxtrot) od verze v4.4.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>EthChan</i>	USINT	Specifikace ethernet kanálu (ETH1 pro Ethernet na centrální jednotce, ETH2 pro Ethernet na komunikačním modulu)
VAR_IN_OUT			
	<i>EthAdr</i>	TLocalEthAdr	Aktuální IP adresa, maska a brána
	<i>.IP</i>	TIPadr	IP adresa
	<i>.IM</i>	TIPadr	IP maska
	<i>.GW</i>	TIPadr	Adresa brány
GetIPAddress			
	Návratová hodnota	BOOL	TRUE pokud se podaří načíst IP adresu. Jinak FALSE

Příklad programu s voláním funkce *GetIPAddress* :

```

PROGRAM prgTestGetIP
VAR
  old_eth_adr   : TLocalEthAdr;
  new_eth_adr   : TLocalEthAdr;
  set, res, tmp : BOOL;
END_VAR

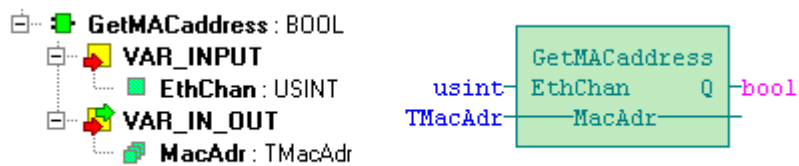
// read actual IP

```

```
tmp := GetIPAddress( EthChan := ETH1, EthAdr := old_eth_adr);  
if new_eth_adr.IP[0] = 0 then  
    new_eth_adr := old_eth_adr;           // use as init value for new adr  
end_if;  
  
// set new IP address if rq = TRUE  
res := SetIPAddress( rq := set, EthChan := ETH1, EthAdr := new_eth_adr);  
if res then set := FALSE; end_if;  
END_PROGRAM
```

Viz také Funkce SetIPAddress, Typ TLocalEthAdr

5.9 Funkce GetMACaddress

Knihovna : *ComLib*

Funkce *GetMACaddress* vrátí aktuální MAC adresu pro zadaný ethernet kanál. MAC adresa je uložena do proměnné *MacAdr*, která má strukturu typu *TMacAdr*. Deklarace této struktury je součástí knihovny ComLib. MAC adresa je jedinečné číslo na celém světě.

Tato funkce je podporována na centrálních jednotkách řady K (TC700 CP-7004, Foxtrot) od verze v4.9.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>EthChan</i>	USINT	Specifikace ethernet kanálu (ETH1 pro Ethernet na centrální jednotce, ETH2 pro Ethernet na komunikačním modulu)
VAR_IN_OUT			
	<i>MacAdr</i>	TMacAdr	Aktuální MAC adresa pro zadaný ethernet kanál
GetMACaddress			
	Návratová hodnota	BOOL	TRUE pokud se podaří načíst MAC adresu. Jinak FALSE

Příklad programu s voláním funkce *GetMACaddress* :

```

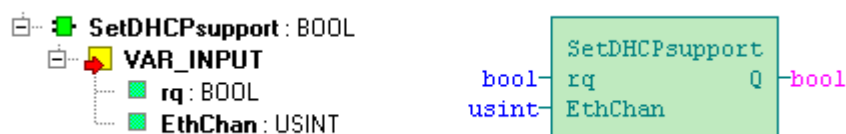
PROGRAM prgTestGetMAC
VAR
    mac_adr : TMacAdr;
    tmp     : BOOL;
    message : STRING;
END_VAR

tmp := GetMACaddress( EthChan := ETH1, MacAdr := mac_adr);
IF (mac_adr[0] = 0) AND (mac_adr[1] = 16#0A) AND (mac_adr[2] = 16#14) THEN
    message := 'This is Teco device';
END_IF;
END_PROGRAM
  
```

Viz také Typ TMacAdr

5.10 Funkce SetDHCPsupport

Knihovna : ComLib



Funkce *SetDHCPsupport* zapne v PLC funkci pro automatické získání IP adresy PLC od DHCP serveru. Zároveň ukončí všechny právě probíhající komunikace. Přechod na automatické získávání IP adresy bude trvat několik cyklů PLC a funkce *SetDHCPsupport* vrátí hodnotu TRUE ve chvíli, kdy je přechod dokončen. Poté PLC požádá DHCP server o přidělení IP adresy, která bude nastavena na daném ethernet rozhraní. Od chvíle, kdy je zahájen přechod na automatické získání IP adresy až do okamžiku jejího přidělení obsluhuje PLC pouze protokoly ARP, ICMP a DHCP. Pokud se během 4 sekund nepodaří získat IP adresu (např. DHCP server není na lokální síti přítomen), použije PLC jako náhradní konfiguraci IP adresu uloženou v konfiguraci ethernet rozhraní, která je uložena v paměti EEPROM. Informaci o aktuálním stavu ethernet rozhraní lze kdykoliv získat v globální proměnné *ETH1_STAT* resp. *ETH2_STAT*. Požadavek na automatické získávání IP adresy je zapamatován v PLC i během vypnutí napájení. Takže po zapnutí napájení bude PLC znovu žádat DHCP server o přidělení IP adresy. Požadavek na automatické získávání IP adresy lze zrušit zavoláním funkce *SetIPaddress*, která nastaví pevnou IP adresu pro dané ethernet rozhraní.

DHCP server přiděluje IP adresu pouze na omezenou dobu (typicky jeden den). Pokud je zapnuta podpora DHCP, PLC automaticky žádá o prodloužení doby, kterou byla IP adresa přidělena. Pokud se tuto dobu nepodaří prodloužit, PLC si ponechá naposledy přidělenou IP adresu a nastaví globální proměnnou *ETH1_STAT.IP_expired* na hodnotu TRUE.

Tato funkce je podporována na centrálních jednotkách řady K (TC700 CP-7004, Foxtrot) od verze v4.9. Funkce je podporována pouze na rozhraní ETH1. Do knihovny ComLib je funkce *SetDHCPsupport* zařazena od ComLib_v13.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>rq</i>	BOOL	Žádost o zapnutí podpory protokolu DHCP Náběžná hrana na tomto vstupu způsobí ukončení všech komunikací se stávající IP adresou a vyslání žádosti o přidělení nové IP adresy na DHCP server
	<i>EthChan</i>	USINT	Specifikace ethernet kanálu (ETH1 pro Ethernet na centrální jednotce)
SetDHCPsupport			
	<i>Návratová hodnota</i>	BOOL	TRUE když se podaří dokončit přechod na automatické získávání IP adresy PLC z DHCP serveru, jinak FALSE Hodnota TRUE znamená, že se PLC pokusí získat IP adresu od DHCP serveru. Informace o tom, jestli se adresu podařilo získat je uložena v globální proměnné <i>ETH1_STAT.IP_obtained</i>

Příklad programu s voláním funkce *SetDHCPsupport* :

```

PROGRAM prgTestDHCPsupport
VAR
  set_fix_IP : BOOL;           // request for fixed IP address
  set_aut_IP : BOOL;           // request for obtaining IP address from DHCP
  my_IP      : TLocalEthAdr;
  info       : STRING;
  fix_IP     : TLocalEthAdr := (IP := [192,168,1,1],
                                IM := [255,255,255,0],
                                GW := [192,168,1,200]);

END_VAR

IF set_fix_IP THEN
  IF SetIPAddress(rq:= set_fix_IP, EthChan:= ETH1, EthAdr:= fix_IP) THEN
    set_fix_IP := FALSE;
  END_IF;
ELSE
  IF set_aut_IP THEN
    IF SetDHCPsupport(rq := set_aut_IP, EthChan := ETH1) THEN
      set_aut_IP := FALSE;
    END_IF;
  END_IF;
END_IF;

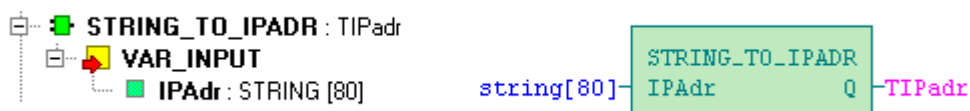
GetIPAddress(EthChan := ETH1, EthAdr := my_IP);
IF ETH1_STAT.DHCP_enabled AND ETH1_STAT.IP_obtained THEN
  info := 'DYNAMIC IP:' + IPADR_TO_STRING(IPAdr := my_IP.IP);
ELSE
  info := 'FIXED IP:' + IPADR_TO_STRING(IPAdr := my_IP.IP);
END_IF;
END_PROGRAM

```

Program nastaví pevnou IP adresu 192.168.1.1 na rozhraní ETH1 v případě, že zapíšeme TRUE do proměnné *set_fix_IP* (např. z WebMakeru). Pokud zapíšeme TRUE do proměnné *set_aut_IP*, pak se program pokusí získat IP adresu od DHCP serveru. Skutečně nastavenou IP adresu pak lze zkontrolovat v proměnné *info*.

Viz také Funkce SetIPAddress, Globální proměnné

5.11 Funkce *STRING_TO_IPADR*

Knihovna : *ComLib*

Funkce *STRING_TO_IPADR* provede konverzi IP adresy zapsané v proměnné typu *STRING* do struktury typu *TIPadr*. Funkce očekává IP adresu zapsanou v obvyklém tvaru např. '192.168.33.1', tedy jako dekadická čísla oddělená tečkou. Vedoucí nuly jsou přípustné např. '192.168.033.001'. Mezery v řetězci nejsou podporovány.

Do knihovny ComLib je funkce *STRING_TO_IPADR* zařazena od ComLib_v13.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>IPAdr</i>	STRING	IP adresa, řetězec znaků obsahující 4 dekadická čísla oddělená tečkami
STRING_TO_IPADR			
	<i>Návratová hodnota</i>	TIPadr	Vstupní řetězec převedený na pole se 4-mi prvky USINT Pokud se převod nepodaří, funkce vrací ANY_IP tj. 0.0.0.0

Příklad programu s voláním funkce *STRING_TO_IPADR* :

```

PROGRAM prgTest_STRING_TO_IPADR
VAR
  tst_IP      : TLocalEthAdr;
  IP_addr     : STRING[16] := '192.168.1.1';
  IP_mask     : STRING[16] := '192.168.1.13';
  GW_addr     : STRING[16] := '192.168.1.200';
END_VAR

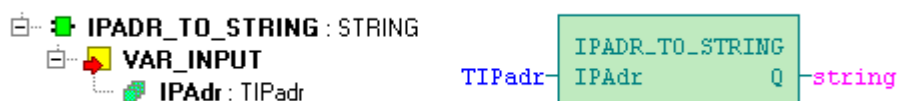
tst_IP.IP := STRING_TO_IPADR(IPAdr := IP_addr);
tst_IP.IM := STRING_TO_IPADR(IPAdr := IP_mask);
tst_IP.GW := STRING_TO_IPADR(IPAdr := GW_addr);
END_PROGRAM

```

Viz také Funkce *IPADR_TO_STRING*

5.12 Funkce IPADR_TO_STRING

Knihovna : ComLib



Funkce *IPADR_TO_STRING* provede konverzi IP adresy zapsané v proměnné typu TIPAdr do proměnné typu STRING.

Do knihovny ComLib je funkce *IPADR_TO_STRING* zařazena od ComLib_v13.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	IPAdr	TIPAdr	IP adresa, zapsaná jako pole se 4-mi prvky typu USINT
IPADR_TO_STRING			
	Návratová hodnota	STRING	Řetězec znaků obsahující 4 dekadická čísla oddělená tečkami

Příklad programu s voláním funkce *IPADR_TO_STRING* :

```

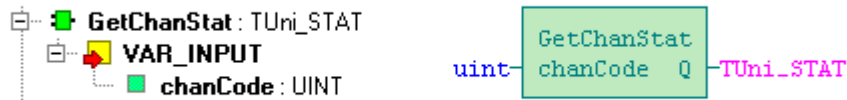
PROGRAM prgTest_IPADR_TO_STRING
VAR
  my_IP      : TLocalEthAdr;
  info       : STRING;
END_VAR

GetIPaddress(EthChan := ETH1, EthAdr := my_IP);
info := ' IP: ' + IPADR_TO_STRING(IPAdr := my_IP.IP) +
        ' IM: ' + IPADR_TO_STRING(IPAdr := my_IP.IM) +
        ' GW: ' + IPADR_TO_STRING(IPAdr := my_IP.GW);
END_PROGRAM
  
```

Viz také Funkce *STRING_TO_IPADR*

5.13 Funkce GetChanStat

Knihovna : ComLib



Funkce *GetChanStat* vrátí informace o aktuálním stavu komunikačního kanálu.

Do knihovny ComLib je funkce *GetChanStat* zařazena od ComLib_v14. Tato funkce je podporována na centrálních jednotkách řady K (TC700 CP-7004, Foxtrot) od verze v4.4.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	chanCode	UINT	Kód komunikačního kanálu <i>ETH1_uni0</i> ethernet kanál ETH1, spojení uni0 <i>ETH1_uni7</i> ethernet kanál ETH1, spojení uni7 <i>ETH2_uni0</i> ethernet kanál ETH2, spojení uni0 <i>ETH2_uni7</i> ethernet kanál ETH2, spojení uni7 <i>CH1_uni</i> sériový kanál CH1, režim uni <i>CH10_uni</i> sériový kanál CH10, režim uni
GetChanStat			
	Návratová hodnota	TUni_Stat	Informace o stavu komunikačního kanálu

Příklad programu s voláním funkce *GetChanStat* :

```

PROGRAM ExampleGetChanStat
VAR
  uniStat      : TUni_STAT;
  txBuf        : STRING[100];
  sendCnt      : UDINT;
END_VAR

uniStat := GetChanStat(chanCode := CH1_uni);
if NOT uniStat.TRO AND NOT uniStat.TRF then
  // channel CH1 is ready to send
  txBuf := 'Message number : ' + UDINT_TO_STRING( sendCnt);

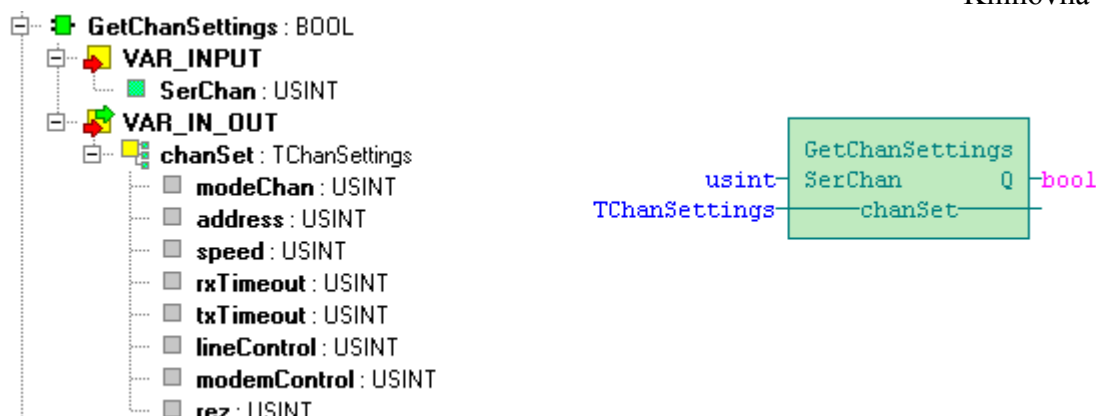
  // ...
end_if;
END_PROGRAM

```

Viz také Typ TUni_STAT

5.14 Funkce GetChanSettings

Knihovna : ComLib



Funkce *GetChanSettings* načte aktuální nastavení sériového komunikačního kanálu do proměnné *chanSet*. Do knihovny ComLib je funkce *GetChanSettings* zařazena od ComLib_v14. Potřebný FW centrální jednotky Foxtrot je v5.3 nebo vyšší.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>SerChan</i>	USINT	Kód sériového komunikačního kanálu <i>SCH1</i> až <i>SCH10</i> ... sériový kanál CH1 až CH10
VAR_IN_OUT			
	<i>chanSet</i>	TChanSet	Parametry sériového kanálu
	<i>.modeChan</i>	USINT	Aktuální režim sériového kanálu (MODE_OFF, MODE_PC, MODE_UNI, MODE_MPC, MODE_MDB)
	<i>.address</i>	USINT	Adresa pro komunikaci
	<i>.speed</i>	USINT	Komunikační rychlost (BAUD_50, ..., BAUD_115200)
	<i>.rxTimeout</i>	USINT	Timeout při příjmu (doba klidu na lince, po které bude zahájen příjem další zprávy)
	<i>.txTimeout</i>	USINT	Timeout při vysílání (minimální zaručená prodleva mezi dvěmi odesílanými zprávami)
	<i>.lineControl</i>	USINT	Parita, počet bitů, počet stop bitů viz konstanty PARITY_ODD, PARITY_EVEN, ...
	<i>.modemControl</i>	USINT	Modemové signály (nastavení signálu RTS) viz konstanty RTS_AUTO, HALF_DUPLEX,...
	<i>.rez</i>	USINT	Rezerva
GetChanSettings			
	Návratová hodnota	BOOL	TRUE pokud se podaří získat nastavení sériového kanálu

Příklad programu s voláním funkce *GetChanSettings* :

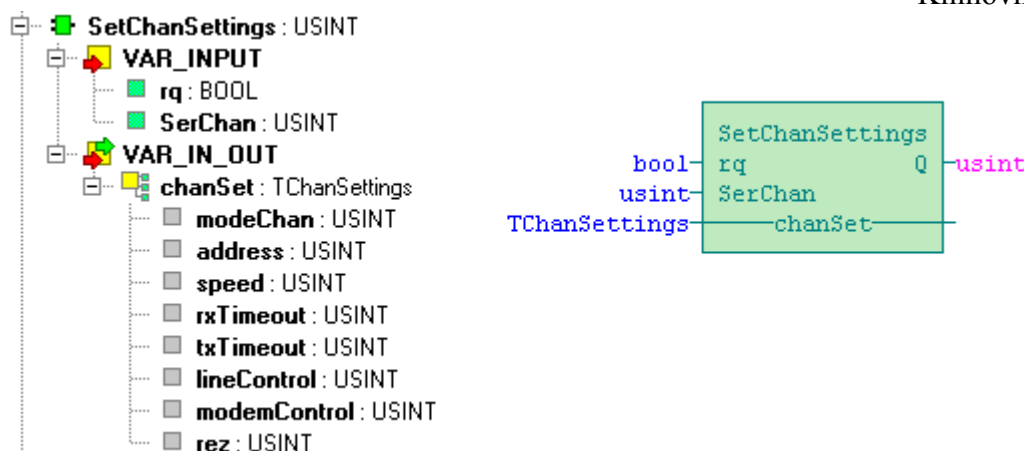
```
PROGRAM ExampleGetChanSettings
VAR
  chlSettings : TChanSettings;
  message     : STRING;
  tmp         : BOOL;
END_VAR

tmp := GetChanSettings( SerChan := SCH1, chanSet := chlSettings);
if tmp then
  case chlSettings.modeChan of
    MODE_OFF : message := 'Serial channel CH1 is switched off!';
    MODE_PC  : message := 'Mode of channel CH1 is PC (EPSNET slave)';
    MODE_UNI : message := 'Mode of channel CH1 is uni (universal)';
    MODE_MPC : message := 'Mode of channel CH1 is MPC (EPSNET master)';
    MODE_MDB : message := 'Mode of channel CH1 is MODBUS slave';
    MODE_PFB : message := 'Mode of channel CH1 is PROFIBUS DP master';
  else
    message := 'Mode of serial channel CH1 is unknown';
  end case;
end if;
END_PROGRAM
```

Viz také Typ TChanSettings

5.15 Funkce SetChanSettings

Knihovna : ComLib



Funkce *SetChanSettings* nastaví přenosové parametry sériového kanálu. Funkci lze použít pro kanály pracující v režimech PC, uni, MPC, MDB a PFB. Nové nastavení udává proměnná *chanSet*. Nastavovaný sériový kanál musí být definovaný v hardwarové konfiguraci PLC. Funkce *SetChanSettings* nedokáže změnit režim kanálu, naopak režim nastavený v hw konfiguraci musí odpovídat režimu v proměnné *chanSet.modeChan*. Parametry dané proměnnými *chanSet.lineControl* a *chanSet.modemControl* lze měnit pouze v režimu uni. Změna nastavení sériového kanálu platí do vypnutí PLC nebo do prvního restartu. Po zapnutí napájení (nebo po restartu PLC) se sériový kanál nastaví podle parametrů zadaných v hw konfiguraci systému. Pokud má být sériový kanál nastaven jinak, je třeba po zapnutí napájení zavolat funkci *SetChanSettings*, která přenastaví parametry podle potřeby.

Tato funkce je podporována pouze pro sériové kanály na centrální jednotce PLC.

Do knihovny ComLib je funkce *SetChanSettings* zařazena od ComLib_v14. Potřebný FW centrální jednotky Foxtrot je v5.3 nebo vyšší. Změna parametrů *lineControl* a *modemControl* vyžaduje FW v6.4 nebo vyšší.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>SerChan</i>	USINT	Kód sériového komunikačního kanálu <i>SCH1</i> sériový kanál CH1 <i>SCH2</i> sériový kanál CH2
VAR_IN_OUT			
	<i>chanSet</i>	TChanSet	Parametry sériového kanálu
	<i>.modeChan</i>	USINT	Režim sériového kanálu viz konstanty MODE_PC, MODE_UNI, MODE_MDB
	<i>.address</i>	USINT	Adresa pro komunikaci
	<i>.speed</i>	USINT	Komunikační rychlost viz konstanty BAUD_50, ..., BAUD_115200

	Proměnná	Typ	Význam
	<i>.rxTimeout</i>	USINT	Timeout při příjmu (doba klidu na lince, po které bude zahájen příjem další zprávy)
	<i>.txTimeout</i>	USINT	Timeout při vysílání (minimální zaručená prodleva mezi dvěmi odesílanými zprávami)
	<i>.lineControl</i>	USINT	Parita, počet bitů, počet stop bitů (pouze pro režim uni) viz konstanty PARITY_ODD, PARITY_EVEN, ...
	<i>.modemControl</i>	USINT	Modemové signály - nastavení signálu RTS atd. (pouze pro režim uni) viz konstanty RTS_AUTO, HALF_DUPLEX,...
	<i>.rez</i>	USINT	Rezerva
GetChanSettings			
	<i>Návratová hodnota</i>	USINT	0 pokud nedojde k chybě (COM_OK) kód chyb v ostatních případech viz konstanty COM_ERR1, ..., COM_ERR64

V následujícím příkladu se nejprve načte aktuální nastavení sériového kanálu CH1 a poté se zkontroluje, zda je nastavena komunikační rychlost 19200 Baud. Pokud je nastavena jiná komunikační rychlost, program nastaví 19200 Baud.

```

PROGRAM ExampleSetChanSettings
VAR
  chlSettings : TChanSettings;
  tmp         : BOOL;
  errCode     : USINT;
  errMsg      : STRING;
END_VAR

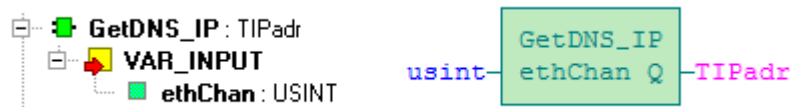
// change communication speed if necessary
tmp := GetChanSettings( SerChan := SCH1, chanSet := chlSettings);
IF chlSettings.speed <> BAUD_19200 THEN
  chlSettings.speed := BAUD_19200;
  errCode := SetChanSettings( rq      := true,
                             SerChan := SCH1,
                             chanSet := chlSettings);
  errMsg := GetLastComErrTxt( errCode);
END_IF;
END_PROGRAM

```

Viz také Typ TChanSettings

5.16 Funkce GetDNS_IP

Knihovna : ComLib



Funkce *GetDNS_IP* vrátí nastavenou IP adresu DNS serveru. Tato adresa je v PLC uložena společně s IP adresou, maskou a bránou sítě v energeticky nezávislé paměti (EEPROM) a PLC ji používá pro komunikaci s DNS serverem v případě, že není definována jiná konkrétní adresa (např. parametrem funkce pro komunikaci s DNS serverem).

Tato funkce je podporována na centrálních jednotkách řady K a L (TC700 CP-7004 a CP-7007, všechny centrální jednotky řady Foxtrot) od verze v7.1. Funkce je podporována pouze na rozhraní ETH1. Do knihovny ComLib je funkce *GetDNS_IP* zařazena od ComLib_v17.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>EthChan</i>	USINT	Specifikace ethernet kanálu (ETH1 pro Ethernet na centrální jednotce)
GetDNS_IP			
	Návratová hodnota	TIPadr	IP adresa DNS serveru

Následující program pomocí funkce *GetDNS_IP()* testuje, je-li nastavena IP adresa DNS serveru. Pokud je IP adresa DNS nulová (není nastavena), provede program pomocí funkce *SetDNS_IP()* nastavení adresy 208.67.222.222. Je-li IP adresa DNS serveru nenulová, pak program čte aktuálně nastavenou adresu a ukládá ji pro kontrolu do proměnné *IP_DNS*.

```

PROGRAM prgExampleGetSetDNS
VAR
    IP_DNS          : TIPadr;
END_VAR
VAR_CONSTANT
    NULL_DNS_IP : TIPadr := [0,0,0,0];
END_VAR

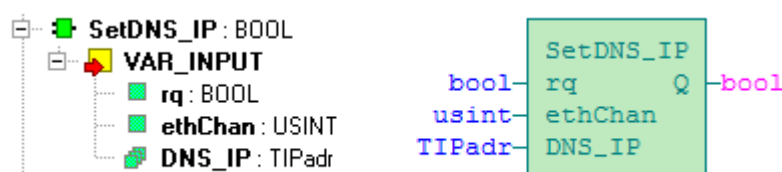
// není-li nastavena => nastavit adresu DNS serveru
IF NULL_DNS_IP = GetDNS_IP( ethChan := ETH1) THEN
    SetDNS_IP( rq          := true,
              ethChan      := ETH1,
              DNS_IP       := STRING_TO_IPADR('208.67.222.222'));
ELSE
    IP_DNS := GetDNS_IP(ethChan := ETH1);
END_IF;
END_PROGRAM

```

Viz také Funkce SetDNS_IP

5.17 Funkce SetDNS_IP

Knihovna : ComLib



Funkce *SetDNS_IP* nastaví novou IP adresu DNS serveru pokud má vstup *rq* hodnotu TRUE. Tato adresa je uložena společně s IP adresou, maskou a bránou sítě v energeticky nezávislé paměti (EEPROM). Funkce vrátí hodnotu TRUE pokud se podaří novou IP adresu DNS serveru nastavit. Tato adresa je používána pro komunikaci s DNS serverem v případě, že není definována jiná konkrétní adresa (např. parametrem funkce pro komunikaci s DNS serverem).

Pokud je zapnuto automatické získání IP adresy PLC od DHCP serveru, tak se od DHCP serveru převezme i adresa DNS serveru (adresa DNS serveru je automaticky nastavena DHCP serverem).

Tato funkce je podporována na centrálních jednotkách řady K a L (TC700 CP-7004 a CP-7007, všechny centrální jednotky řady Foxtrot) od verze v7.1. Funkce je podporována pouze na rozhraní ETH1. Do knihovny ComLib je funkce *SetDNS_IP* zařazena od ComLib_v17.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>rq</i>	BOOL	Žádost o nastavení IP adresy DNS serveru.
	<i>EthChan</i>	USINT	Specifikace ethernet kanálu (ETH1 pro Ethernet na centrální jednotce)
	<i>DNS_IP</i>	TIPadr	Nová IP adresa DNS serveru
SetDNS_IP			
	Návratová hodnota	BOOL	TRUE když se podaří IP adresu DNS serveru nastavit, jinak FALSE

Příklad programu s voláním funkce *SetDNS_IP* :

Následující program pomocí funkce *GetDNS_IP()* testuje, je-li nastavena IP adresa DNS serveru. Pokud je IP adresa DNS nulová (není nastavena), provede program pomocí funkce *SetDNS_IP()* nastavení adresy 208.67.222.222. Je-li IP adresa DNS serveru nenulová, pak program čte aktuálně nastavenou adresu a ukládá ji pro kontrolu do proměnné *info_DNS*.

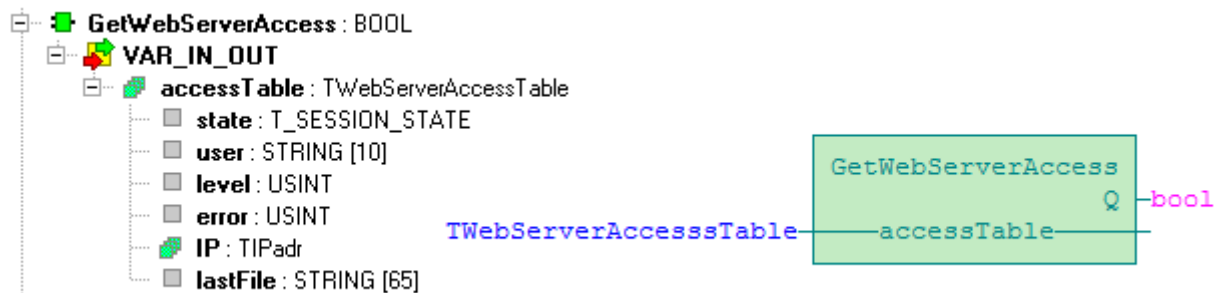
```
PROGRAM prgExampleGetSetDNS
VAR
    info_DNS      : STRING;
END_VAR
VAR_CONSTANT
    NULL_DNS_IP : TIPadr := [0,0,0,0];
END_VAR

// neni-li nastavena => nastavit adresu DNS serveru
IF NULL_DNS_IP = GetDNS_IP( ethChan := ETH1) THEN
    SetDNS_IP(rq := true,
              ethChan := ETH1,
              DNS_IP := STRING_TO_IPADR('208.67.222.222'));
ELSE
    info_DNS := 'DNS IP address : ' +
                IPADR_TO_STRING( GetDNS_IP(ethChan := ETH1));
END_IF;
END_PROGRAM
```

Viz také Funkce GetDNS_IP

5.18 Funkce GetWebServerAccess

Knihovna : ComLib



Funkce *GetWebServerAccess* slouží ke zjištění informací o uživateli, kteří jsou aktuálně připojeni k web serveru v PLC pomocí internetového prohlížeče. Tyto informace uloží funkce do proměnné typu *TWebServerAccessTable*, jejíž jméno musí být uvedeno jako parametr *accessTable*.

Tato funkce je podporována na centrálních jednotkách řady K a L (TC700 CP-7004 a CP-7007, všechny centrální jednotky řady Foxtrot) od verze v7.7. Do knihovny ComLib je funkce *GetWebServerAccess* zařazena od ComLib_v21.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_IN_OUT			
	<i>accessTable</i>	STRUCT	Proměnná, do které funkce <i>GetWebServerAccess</i> uloží získané informace o připojených uživateli
GetWebServerAccess			
	Návratová hodnota	BOOL	TRUE když se podaří získat informace o připojených uživateli

Viz také Typ *T_SESSION_STATE*, Typ *TWebServerAccess*,
Typ *TWebServerAccessTable*

Následující program volá funkci *GetWebServerAccess* a počítá, kolik uživatelů je aktuálně připojeno k web serveru PLC (viz proměnná *numUsers*). Dále je v programu vytvořena instance funkčního bloku *fbLogAccess*. Tento funkční blok ukládá informace o přístupech k web serveru na SD kartu do csv souboru. Pro tento blok je nutné přidat do projektu knihovny *FileLib* a *ToStringLib*. V PLC systému musí být osazena SD karta.

```

VAR_GLOBAL
  webAccessTable : TWebServerAccessTable;
END_VAR

PROGRAM prgTestWebAccess
  VAR
    LogWebAccess : fbLogWebAccess;
    i            : INT;
    numUsers     : USINT;
  
```

```

END_VAR

GetWebServerAccess(accessTable := webAccessTable);
LogWebAccess(accessTable := webAccessTable);
numUsers := 0;
FOR i := 0 TO 7 DO
  IF webAccessTable[i].state = SESSION_ACTIVE THEN
    numUsers := numUsers + 1;
  END_IF;
END_FOR;
END_PROGRAM

```

Funkční blok *fbLogWebAccess* je naprogramován následovně:

```

FUNCTION_BLOCK fbLogWebAccess
VAR_INPUT
  LogName : STRING := 'WWW/WEBLOG.CSV';
END_VAR
VAR_IN_OUT
  accessTable : TWebServerAccessTable;
END_VAR
VAR
  oldCrc : DWORD;
  oldAccessTable : TWebServerAccessTable;
  h : HANDLE := INVALID_HANDLE_VALUE;
  LogLine : STRING[255];
  CloseFileTim : TON;
END_VAR
VAR_TEMP
  i : UINT;
  NoChange : BOOL := true;
END_VAR

// change of program
IF CRC_OF_APLIC_PROGRAM <> oldCrc THEN
  oldAccessTable := accessTable; oldCrc := CRC_OF_APLIC_PROGRAM;
END_IF;

// change
IF oldAccessTable <> accessTable THEN
  IF h = INVALID_HANDLE_VALUE THEN
    h := FileOpen(mode := F_APPEND, fileName := LogName);
    IF FileSize(hFile := h) = 0 THEN
      LogLine := 'DATE;TIME;INDEX;ERROR;IP_ADR;FILE;USER;STATE$0D$0A';
      FileWrite(hFile := h, adrBuf := PTR_TO_UDINT(ADR(LogLine)),
        size := LEN(IN := LogLine));
    END_IF;
  END_IF;

  FOR i := 0 TO 7 DO
    IF oldAccessTable[i] <> accessTable[i] THEN
      CASE accessTable[i].state OF
        SESSION_EMPTY :
          IF oldAccessTable[i].user = '' THEN
            LogLine := ';Timed out ';
          ELSE
            LogLine := oldAccessTable[i].user + ';timed out ';
          END_IF;
        SESSION_LOGIN :
          LogLine := ';Login session started ';
        SESSION_LOGIN_RUN :
          LogLine := ';Login session runs ';
        SESSION_ACTIVE :

```

```

        LogLine := accessTable[i].user + ';logged in ';
    SESSION_LOGOUT :
        LogLine := oldAccessTable[i].user + ';logged out ';
    END_CASE;
    LogLine := DT_TO_STRINGF(in := GetDateTime(),
                            format := '%TDD.MM.YY;hh:mm:ss.zzz;') +
        UINT_TO_STRING(i) + ';' +
        USINT_TO_STRING(accessTable[i].error) + ';' +
        IPADR_TO_STRING(IPAdr := accessTable[i].IP) + ';' +
        accessTable[i].lastFile + ';' +
        LogLine + '$0D$0A';
    FileWrite(hFile := h, adrBuf := PTR_TO_UDINT(ADR(LogLine)),
        size := LEN(IN := LogLine));
    NoChange := false;
    END_IF;
    END_FOR;
    oldAccessTable := accessTable;
    END_IF;

//close if no changes
CloseFileTim(IN := NoChange AND h <> INVALID_HANDLE_VALUE, PT := T#2m30s);
IF CloseFileTim.Q THEN
    FileClose(hFile := h);
    h := INVALID_HANDLE_VALUE;
    END_IF;
END_FUNCTION_BLOCK

```

Informace o přístupech k web serveru jsou uloženy v souboru WEBLOG.CSV. Ten je na SD karte vytvořen v adresáři WWW. Data uložená v souboru mohou vypadat např. následovně:

```

DATE;TIME;INDEX;ERROR;IP_ADR;FILE;USER;STATE
23.07.13;10:06:35.949;0;0;192.168.33.4;;Login session runs
23.07.13;10:06:35.973;0;0;192.168.33.4;/syswww/login.xml;;Login session runs
23.07.13;10:06:44.144;0;0;192.168.33.4;/index.xml;user0;logged in
23.07.13;10:06:44.384;0;0;192.168.33.4;/PAGE1.XML;user0;logged in
23.07.13;10:06:48.435;0;0;192.168.33.4;/PAGE2.XML;user0;logged in
23.07.13;10:06:58.363;6;0;90.182.180.210;;Login session runs
23.07.13;10:06:58.455;6;0;90.182.180.210;/syswww/login.xml;;Login session runs
23.07.13;10:07:08.608;6;0;90.182.180.210;/index.xml;user5;logged in
23.07.13;10:07:12.513;6;0;90.182.180.210;/PAGE2.XML;user5;logged in
23.07.13;10:07:19.225;6;0;0.0.0.0;;user5;timed out
23.07.13;10:07:23.485;0;0;192.168.33.4;/syswww/logout.xml;user0;logged out

```

V tabulkovém kalkulátoru pak vidíme:

	A	B	C	D	E	F	G	H
1	DATE	TIME	INDEX	ERROR	IP_ADR	FILE	USER	STATE
2	23.7.2013	10:06:35.949		0	192.168.33.4			Login session runs
3	23.7.2013	10:06:35.973	0	0	192.168.33.4	/syswww/login.xml		Login session runs
4	23.7.2013	10:06:44.144	0	0	192.168.33.4	/index.xml	user0	logged in
5	23.7.2013	10:06:44.384	0	0	192.168.33.4	/PAGE1.XML	user0	logged in
6	23.7.2013	10:06:48.435	0	0	192.168.33.4	/PAGE2.XML	user0	logged in
7	23.7.2013	10:06:58.363	6	0	90.182.180.210			Login session runs
8	23.7.2013	10:06:58.455	6	0	90.182.180.210	/syswww/login.xml		Login session runs
9	23.7.2013	10:07:08.608	6	0	90.182.180.210	/index.xml	user5	logged in
10	23.7.2013	10:07:12.513	6	0	90.182.180.210	/PAGE2.XML	user5	logged in
11	23.7.2013	10:07:19.225	6	0	0.0.0.0		user5	timed out
12	23.7.2013	10:07:23.485	0	0	192.168.33.4	/syswww/logout.xml	user0	logged out.

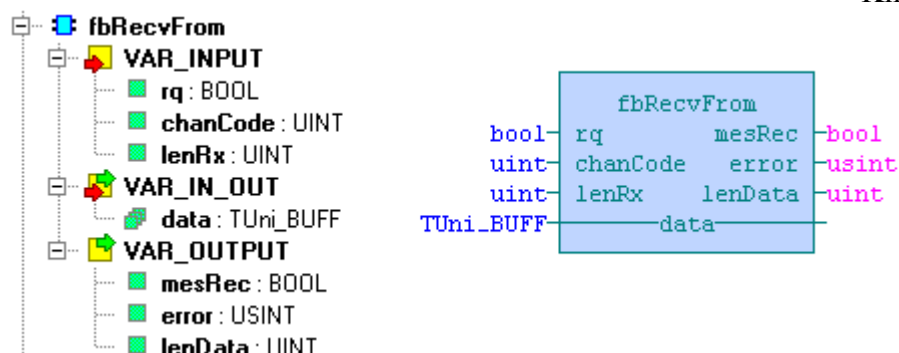
6 FUNKČNÍ BLOKY

Knihovna ComLib obsahuje následující funkční bloky:



<i>Funkční blok</i>	<i>Popis</i>
<i>fbKeepAlive</i>	funkční blok pro udržení TCP spojení
<i>fbRecvFrom</i>	funkční blok pro příjem dat z komunikačního kanálu
<i>fbRecvTxt</i>	funkční blok, který přijímá data z komunikačního kanálu zakončená znaky CR LF
<i>fbRecvTxtChar</i>	funkční blok, který přijímá data z komunikačního kanálu zakončená znakem uvedeným na vstupu <i>delimiter</i>
<i>fbSendTo</i>	funkční blok pro vysílání dat komunikačním kanálem

6.1 Funkční blok *fbRecvFrom*

Knihovna : *ComLib*

Funkční blok *fbRecvFrom* slouží k příjmu zprávy ze sériového kanálu nebo z ethernetu. Komunikační kanál musí být nastaven v režimu *uni*.

Tento funkční blok je podporován na centrálních jednotkách řady K a L (TC700 CP-7004, CP-7007, všechny verze CPU Foxtrot) od verze v4.4.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>rq</i>	BOOL	Řídicí proměnná. Je-li TRUE, funkční blok přijímá data z komunikačního kanálu, je-li FALSE příjem dat je zakázán
	<i>chanCode</i>	UINT	Kód komunikačního kanálu <i>ETH1_uni0</i> ethernet kanál ETH1, spojení uni0 <i>ETH1_uni7</i> ethernet kanál ETH1, spojení uni7 <i>ETH2_uni0</i> ethernet kanál ETH2, spojení uni0 <i>ETH4_uni7</i> ethernet kanál ETH4, spojení uni7 <i>CH1_uni</i> sériový kanál CH1, režim uni <i>CH10_uni</i> sériový kanál CH10, režim uni
	<i>lenRx</i>	UINT	Max. délka přijatých dat (velikost proměnné, do které bude uložena přijatá zpráva)
VAR_IN_OUT			
	<i>data</i>	TUniBuf	Proměnná, do které bude uložena přijatá zpráva
VAR_OUTPUT			
	<i>mesRec</i>	BOOL	Příznak přijaté zprávy Je-li TRUE byla přijatá nová zpráva.
	<i>error</i>	USINT	Chybový kód. Je-li 0 (COM_OK), příjem proběhl bez chyby V případě chyby vrací kódy COM_ERR1, ..., COM_ERR64
	<i>lenData</i>	UINT	Délka přijaté zprávy (počet bytů)

V následujícím příkladu je funkční blok *fbRecvFrom* použit pro příjem zpráv ze sériového kanálu CH1. Očekává se příjem textových zpráv s max.velikostí 100 znaků. Nastavení přenosových parametrů sériového kanálu je vidět z následujícího dialogu.

Nastavení univerzálního režimu kanálu

Přijímací zóna
Délka zóny: 101
Adresa zóny: 4
Přijímací zóna: CH2_ZoneIN

Vysílací zóna
Délka zóny: 101
Adresa zóny: 4
Vysílací zóna: CH2_ZoneOUT

Komunikační rychlost: 38 400
Formát dat: 8b, bez parity

Adresa stanice: 0
Adresa stanice: 0
Detekovat při příjmu: 0
Zápis při vysílání: 0

Počáteční znak
☐ Detekovat
☐ Vysílat
Kód znaku: 0

Koncový znak
☐ Detekovat
☐ Vysílat
☐ Dva znaky
Kód znaku: 0 0

Parita prvního bytu přijímané zprávy
☒ Stejná parita jako u ostatních
☐ Opačná parita než u ostatních

Parita prvního bytu vysílané zprávy
☒ Stejná parita jako u ostatních
☐ Opačná parita než u ostatních

Kontrolní součet
☐ Kontrola při příjmu
☐ Výpočet při vysílání
Poz. prvního znaku CHS: 0

Potvrzení zprávy bez dat
☐ Detekovat
☐ Vysílat
☐ Dva znaky
Kód znaku: 0 0

Délka zprávy
☐ Detekovat při příjmu
☐ Zápis při vysílání
Pozice délky zprávy: 0
Maximální délka: 0

Režim řízení modemových signálů
Řízení signálu RTS: automatická hodnota
Řízení signálu DTR: trvale hodnota 0
☐ Odpojení přijímače během vysílání

Min. doba klidu na lince mezi přijímanými zprávami (počet bytů): 10
Min. doba klidu na lince mezi vysílanými zprávami (počet bytů): 10

OK Zrušit nápověda

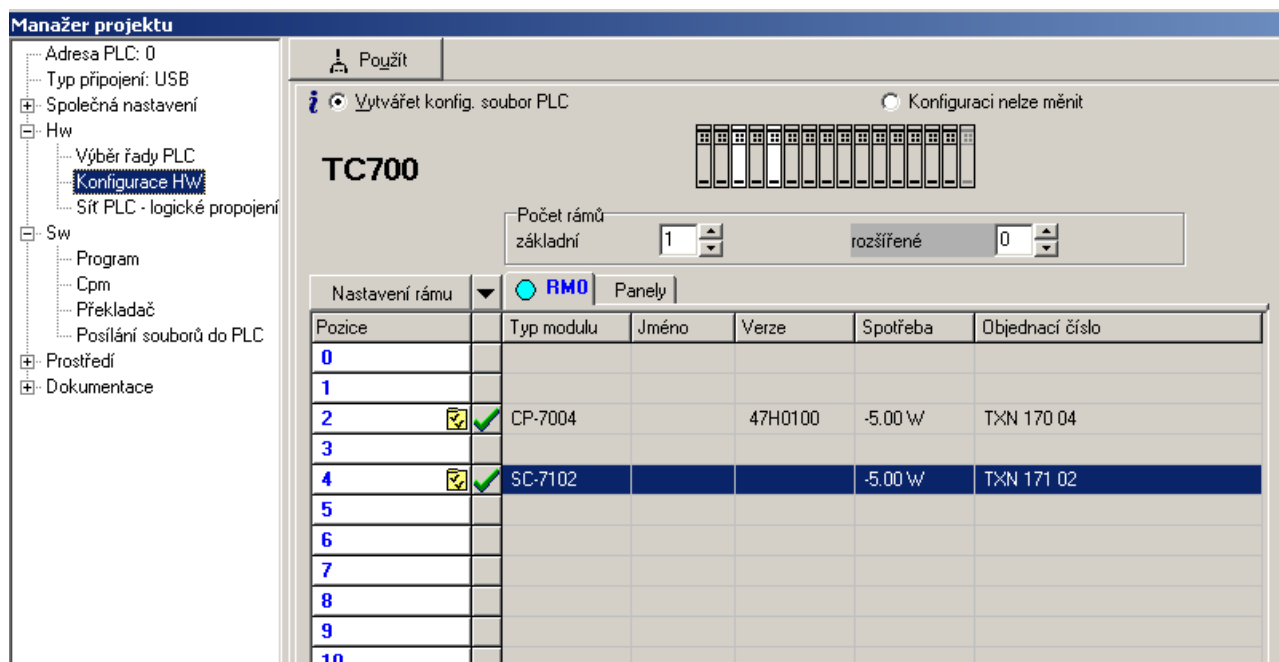
```

PROGRAM ExampleRecvFrom
VAR
  RecvFromCH1 : fbRecvFrom;
  rxBuf       : STRING[100];
  errMsg      : STRING;
END_VAR

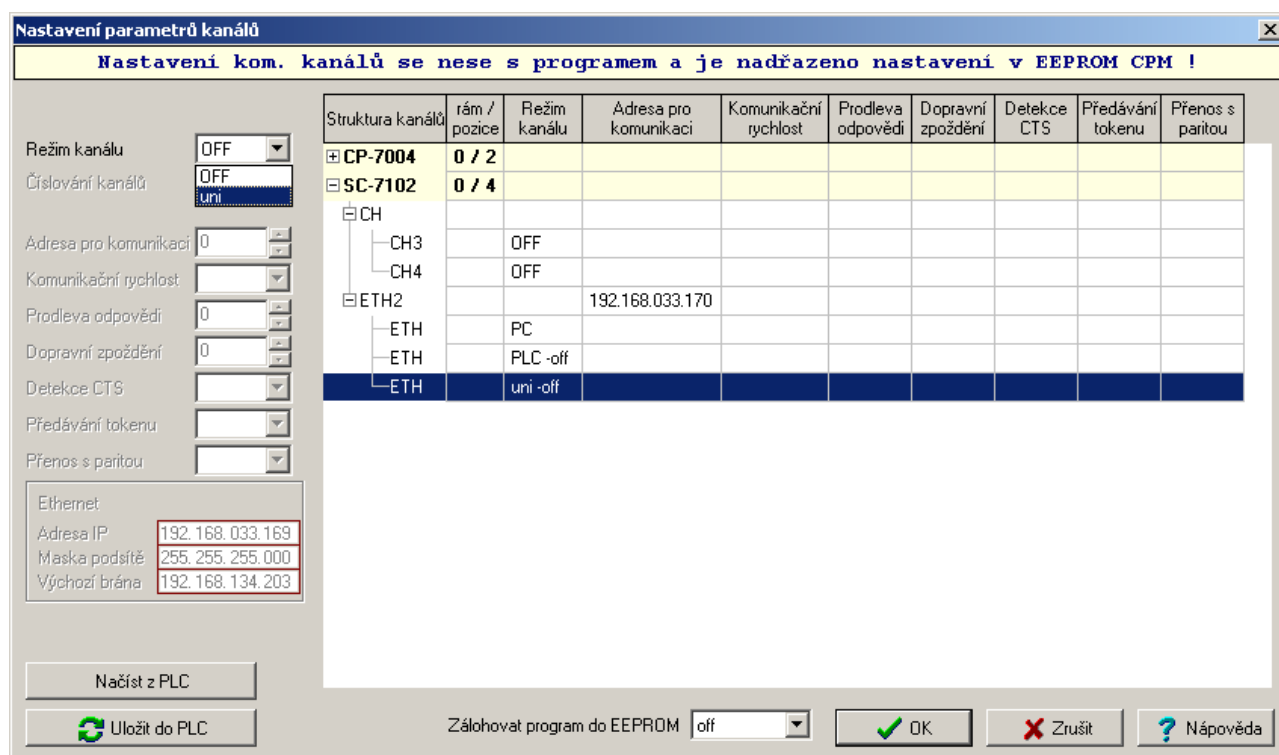
// receiving
RecvFromCH1( rq := TRUE, chanCode := CH1_uni,
             lenRx := 100, data := void(rxBuf));
if RecvFromCH1.mesRec then
  // new message received
  if RecvFromCH1.error = 0 then
    // process new message
    // ...
  else
    errMsg := GetLastComErrTxt( RecvFromCH1.error); // show error as a text
  end if;
end if;
END_PROGRAM

```

Další příklad ukazuje použití funkčního bloku *fbRecvFrom* pro příjem zpráv TCP protokolem. Nejprve je třeba nastavit režim uni pro rozhraní ethernet. Nastavení se provádí v prostředí Mosaic v Manažeru projektu v uzlu HW konfigurace. V tomto konkrétním případě bylo použito rozhraní ETH2 na modulu SC-7102.



Klinutím na žlutou ikonu u modulu SC-7102 (viz zvýrazněný řádek na předchozím obrázku) vyvoláme nastavení komunikací pro modul SC-7102. V tomto dialogu je třeba nejprve zapnout režim kanálu uni na rozhraní ethernet tak, že nejprve klikneme na řádek ETH uni-off (na následujícím obrázku je zvýrazněn) a poté přepneme režim kanálu v levém horním rohu z hodnoty OFF na uni.



Po nastavení režimu uni by dialog měl vypadat následovně

Nastavení parametrů kanálů

Nastavení kom. kanálů se nese s programem a je nadřazeno nastavení v EEPROM CPM !

Režim kanálu: **uni**

Číslování kanálů:

Adresa pro komunikaci: **0**

Komunikační rychlost:

Prodleva odpovědi: **0**

Dopravní zpoždění: **0**

Detekce CTS:

Předávání tokenů:

Přenos s paritou:

Ethernet

Adresa IP: **192.168.033.169**

Maska podsítě: **255.255.255.000**

Výchozí brána: **192.168.134.203**

Načíst z PLC

Uložit do PLC

Zálohovat program do EEPROM: **off**

OK Zrušit Nápověda

Struktura kanálů	rám / pozice	Režim kanálu	Adresa pro komunikaci	Komunikační rychlost	Prodleva odpovědi	Dopravní zpoždění	Detekce CTS	Předávání tokenů	Přenos s paritou
CP-7004	0 / 2								
SC-7102	0 / 4								
CH									
CH3		OFF							
CH4		OFF							
ETH2			192.168.033.170						
ETH		PC							
ETH		PLC -off							
ETH		uni							

Kliknutím na žlutou ikonu v řádku ETH uni (na předchozím obrázku je zvýrazněn) vyvoláme další dialog, ve kterém nastavíme max. velikosti vysílací a přijímací zóny (počet bytů), vybereme typ protokolu TCP slave (PLC nebude aktivně navazovat TCP spojení) a zadáme číslo portu, na kterém budeme očekávat data (místní port).

Nastavení univerzálního režimu kanálu

UNIO

Přijímací zóna

Délka zóny: **100**

Adresa zóny: ☐ %R0

Přijímací zóna: **ETH2_UNIO_IN**

Vysílací zóna

Délka zóny: **100**

Adresa zóny: ☐ %R0

Vysílací zóna: **ETH2_UNIO_OUT**

Typ protokolu

☐ TCP master

☒ TCP slave

☐ UDP

Vzdálená IP adresa: **0.0.0.0**

Vzdálený port: **61000**

Místní port: **61001**

OK Zrušit Nápověda

Následující program přijímá data zasílaná TCP protokolem na port 61001. Přijatá data jsou uložena v proměnné rxData. Navazování a ukončování TCP spojení řídí stanice, která vysílá data.

Touto stanicí může být např. PLC s programem `prgTestSendTCP`, který je uveden v příkladech následujících kapitol.

```

PROGRAM prgTestRecvTCP
VAR
    recvTim      : TON;
    recvFrom     : fbRecvFrom;
    rxData       : ARRAY[0..49] OF USINT;
    cntOK        : UDINT;
    cntERR       : UDINT;
    lastErr      : USINT;
    lastErrTxt   : STRING;
END_VAR

recvTim(IN := TRUE, PT := T#35s);
IF recvTim.Q THEN
    cntERR := cntERR + 1; recvTim( IN := false);
END_IF;
recvFrom( rq := true, chanCode := ETH2_uni0,
          lenRx := sizeof(rxData), data := void(rxData),
          error => lastErr);

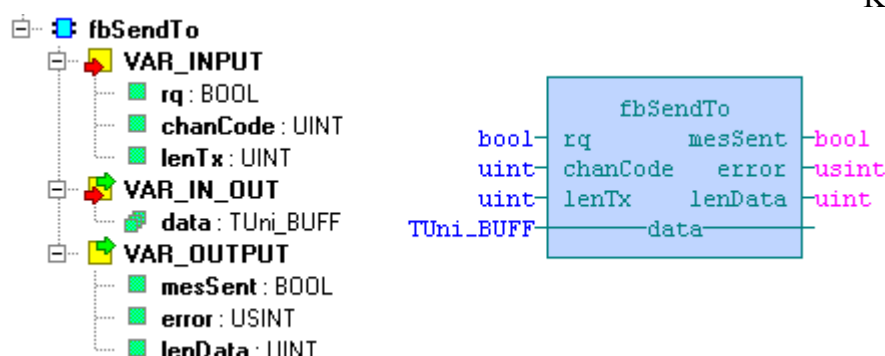
IF recvFrom.mesRec THEN
    recvTim( IN := false);
    IF recvFrom.error = COM_OK THEN
        cntOK := cntOK + 1;
        // process incoming data (see rxData[])
        // ...
    ELSE
        cntERR := cntERR + 1;
        lastErrTxt := GetLastComErrTxt(errCode := lastErr);
    END_IF;
END_IF;
END_PROGRAM

```

Viz také Funkční blok `fbSendTo`, Funkční blok `fbRecvFromTxt`, Funkční blok `fbRecvFromChar`

6.2 Funkční blok fbSendTo

Knihovna : ComLib



Funkční blok *fbSendTo* slouží k vysílání zprávy ze sériového kanálu nebo z ethernetu. Komunikační kanál musí být nastaven v režimu *uni*.

Tento funkční blok je podporován na centrálních jednotkách řady K a L (TC700 CP-7004, CP-7007, všechny verze CPU Foxtrot) od verze v4.4.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>rq</i>	BOOL	Řídící proměnná. Je-li TRUE, funkční blok vyšle data do komunikačního kanálu
	<i>chanCode</i>	UINT	Kód komunikačního kanálu <i>ETH1_uni0</i> ethernet kanál ETH1, spojení uni0 <i>ETH1_uni7</i> ethernet kanál ETH1, spojení uni7 <i>ETH2_uni0</i> ethernet kanál ETH2, spojení uni0 <i>ETH4_uni7</i> ethernet kanál ETH4, spojení uni7 <i>CH1_uni</i> sériový kanál CH1, režim uni <i>CH10_uni</i> sériový kanál CH10, režim uni
	<i>lenTx</i>	UINT	Délka vysílané zprávy (počet bytů)
VAR_IN_OUT			
	<i>data</i>	TUniBuf	Proměnná, ve které je připravena vysílaná zpráva
VAR_OUTPUT			
	<i>mesSent</i>	BOOL	Příznak odeslané zprávy Je-li TRUE vysílání zprávy bylo zahájeno
	<i>error</i>	USINT	Chybový kód. Je-li 0 (COM_OK), vše proběhlo bez chyby V případě chyby vrací kódy COM_ERR1, ..., COM_ERR64
	<i>lenData</i>	UINT	Délka vyslané zprávy (počet bytů)

V následujícím příkladu je funkční blok *fbSendTo* použit pro cyklické vysílání zprávy sériovým kanálem CH1. Jako zpráva se vysílá text „Message number : 1“. Číslo uvedené v řetězci se inkrementuje při každém odeslání zprávy takže udává počet odeslaných zpráv. Rytmus odesílání je řízen časovačem *sendTim*. Konkrétní nastavení sériového kanálu ukazuje následující obrázek.

Nastavení univerzálního režimu kanálu

Přijímací zóna
Délka zóny: 101
Adresa zóny: 4
Přijímací zóna: CH2_ZoneIN

Vysílací zóna
Délka zóny: 101
Adresa zóny: 4
Vysílací zóna: CH2_ZoneOUT

Komunikační rychlost: 38 400
Formát dat: 8b, bez parity

Adresa stanice
Adresa stanice: 0
Detekovat při příjmu: 0
Zápis při vysílání: 0

Počáteční znak
Detekovat: ☐
Vysílat: ☐
Kód znaku: 0

Koncový znak
Detekovat: ☐
Vysílat: ☐ Dva znaky
Kód znaku: 0 0

Parita prvního bytu přijímané zprávy
☒ Stejná parita jako u ostatních
☐ Opačná parita než u ostatních

Parita prvního bytu vysílané zprávy
☒ Stejná parita jako u ostatních
☐ Opačná parita než u ostatních

Kontrolní součet
Kontrola při příjmu: ☐
Výpočet při vysílání: ☐
Poz. prvního znaku CHS: 0

Potvrzení zprávy bez dat
Detekovat: ☐
Vysílat: ☐
Dva znaky: ☐
Kód znaku: 0 0

Délka zprávy
Detekovat při příjmu: ☐
Zápis při vysílání: ☐
Pozice délky zprávy: 0
Maximální délka: 0

Režim řízení modemových signálů
Řízení signálu RTS: automatická hodnota
Řízení signálu DTR: trvale hodnota 0
Odpojení přijímače během vysílání: ☐

Min. doba klidu na lince mezi přijímanými zprávami (počet bytů): 10
Min. doba klidu na lince mezi vysílanými zprávami (počet bytů): 10

OK Zrušit Nápověda

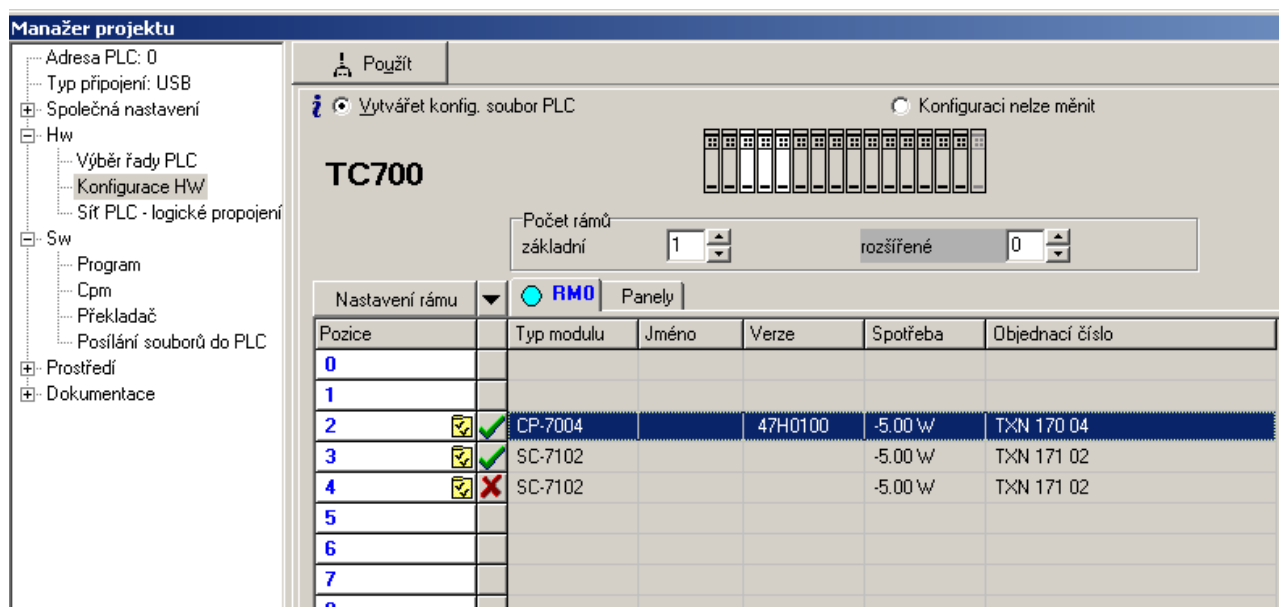
```

PROGRAM ExampleSendTo
VAR
    SendToCH1    : fbSendTo;
    txBuf        : STRING[100];
    sendTim      : TON;
    sendCnt      : UDINT;
    errMsg       : STRING;
END_VAR

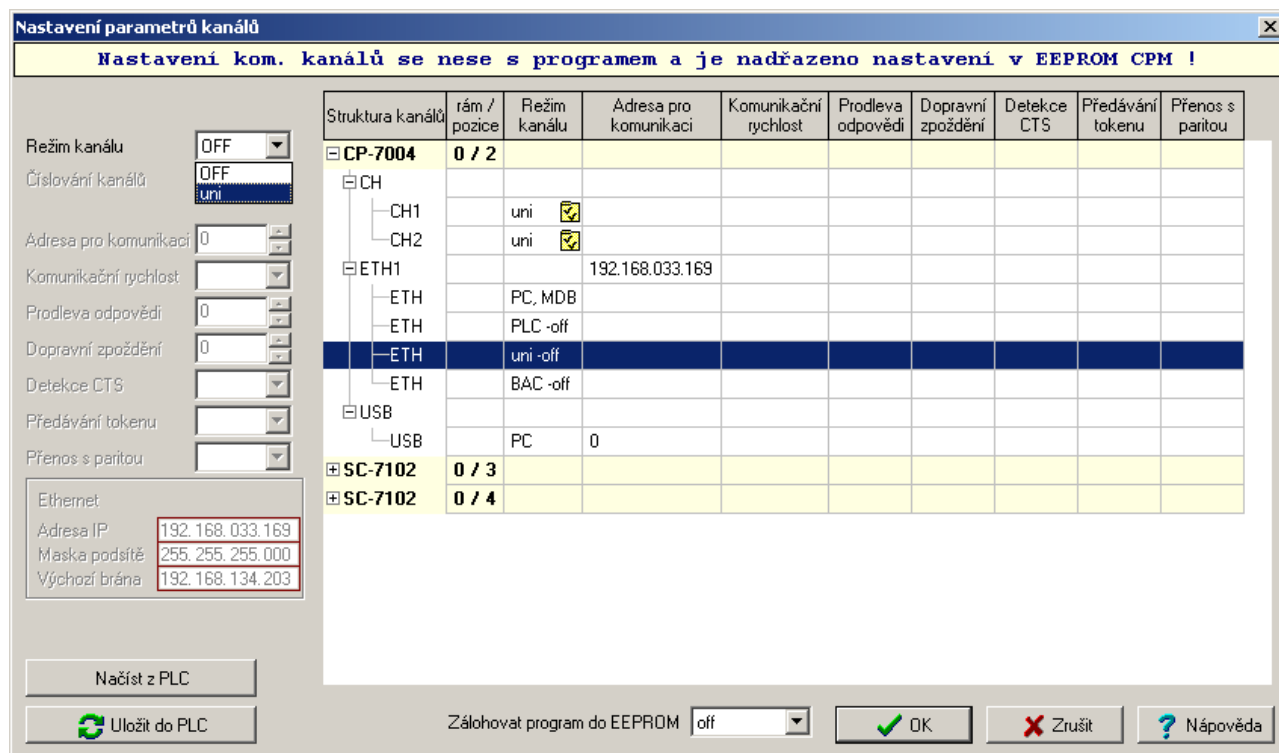
sendTim(IN := TRUE, PT := T#3s);
if sendTim.Q then                                     // send new message every 3 sec
    sendCnt := sendCnt + 1;                             // number of messages
    txBuf := 'Message number : ' + UDINT_TO_STRING( sendCnt);
    SendToCH1( rq := TRUE, chanCode := CH1_uni,
               lenTx := len(txBuf), data := void(txBuf));
    if SendToCH1.error = 0 then                         // no error
        if SendToCH1.mesSent then                     // message sent succesfully
            sendTim(IN := FALSE);                     // timer restart
        end_if;
    else
        errMsg := GetLastComErrTxt( SendToCH1.error); // show error as a text
    end_if;
end_if;
END_PROGRAM

```

Další příklad ukazuje použití funkčního bloku *fbSendTo* pro vysílání dat TCP protokolem přes rozhraní ETH1. Program každých 30 sec naváže spojení TCP protokolem, odvysílá data a spojení opět uzavře. Cílová IP adresa a čísla portů jsou v tomto případě pevně dány nastavením ethernet kanálu. Nastavení se provádí v prostředí Mosaic v Manažeru projektu v uzlu HW konfigurace.



Kliknutím na žlutou ikonu v řádku s centrální jednotkou (zde konkrétně CP-7004) otevřeme dialog pro nastavení ethernet rozhraní. V tomto dialogu je třeba nejprve zapnout režim kanálu uni na rozhraní ethernet tak, že nejprve klikneme na řádek ETH uni-off (na následujícím obrázku je zvýrazněn) a poté přepneme režim kanálu v levém horním rohu z hodnoty OFF na uni.



Po nastavení režimu uni by dialog měl vypadat následovně

Nastavení parametrů kanálů

Nastavení kom. kanálů se nese s programem a je nadřazeno nastavení v EEPROM CPM !

Režim kanálu: **uni**
Číslování kanálů: **0**
Adresa pro komunikaci: **0**
Komunikační rychlost: **0**
Prodleva odpovědi: **0**
Dopravní zpoždění: **0**
Detekce CTS: **0**
Předávání tokenů: **0**
Přenos s paritou: **0**

Ethernet
Adresa IP: **192.168.033.169**
Maska podsítě: **255.255.255.000**
Výchozí brána: **192.168.134.203**

Načíst z PLC
Uložit do PLC

Struktura kanálů	rám / pozice	Režim kanálu	Adresa pro komunikaci	Komunikační rychlost	Prodleva odpovědi	Dopravní zpoždění	Detekce CTS	Předávání tokenů	Přenos s paritou
CP-7004	0 / 2								
CH									
CH1		uni							
CH2		uni							
ETH1			192.168.033.169						
ETH		PC, MDB							
ETH		PLC -off							
ETH		uni							
ETH		BAC -off							
USB									
USB		PC	0						
SC-7102	0 / 3								
SC-7102	0 / 4								

Zálohovat program do EEPROM: **off**
OK Zrušit Nápověda

Kliknutím na žlutou ikonu v řádku ETH uni (na obrázku je zvýrazněn) vyvoláme další dialog, ve kterém nastavíme max. velikosti vysílací a přijímací zóny (počet bytů), vybereme typ protokolu TCP master (PLC bude aktivně navazovat TCP spojení) a zadáme IP adresu systému, kterému budou zasílána data. Vzdálený port je port na který budou data odesílána, místní port je naopak port ze kterého jsou data odesílána.

Nastavení univerzálního režimu kanálu

UNIO + -

Přijímací zóna
Délka zóny: **100**
Adresa zóny: ☐ %R0
Přijímací zóna: **ETH1_UNIO_IN**

Vysílací zóna
Délka zóny: **100**
Adresa zóny: ☐ %R0
Vysílací zóna: **ETH1_UNIO_OUT**

Typ protokolu
☒ TCP master
☐ TCP slave
☐ UDP

Vzdálená IP adresa: **192.168.33.171**
Vzdálený port: **61001**
Místní port: **61000**

OK Zrušit Nápověda

Program, který cyklicky každých 30 sec odesílá data TCP protokolem na adresu 192.168.33.171 a cílový port 61001 vypadá následovně.

```

TYPE
  states : (idle, estabCon, sendData, closeCon);
END_TYPE

PROGRAM prgTestSendTCP
  VAR
    sendTim      : TON;
    chkTim       : TON;
    state        : states := idle;
    txData       : ARRAY[0..49] OF USINT;
    sendTo       : fbSendTo;
    cntOK        : UDINT;
    cntERR       : UDINT;
    lastErr      : USINT;
    lastErrTxt   : STRING;
  END_VAR

  sendTim(IN := true, PT := T#30s);
  CASE state OF
    idle :
      IF IsEstabTCPconnection(chanCode := ETH1_UNI0) THEN
        CloseTCPconnection(chanCode := ETH1_UNI0);
      END_IF;
      IF sendTim.Q THEN
        state := estabCon; sendTim(IN := false); chkTim(IN := false);
      END_IF;

    estabCon :
      lastErr := EstabTCPconnection(chanCode := ETH1_UNI0);
      IF lastErr = COM_OK THEN
        state := sendData;
      END_IF;

    sendData :
      IF IsEstabTCPconnection(chanCode := ETH1_UNI0) THEN
        // prepare outgoing data (see txData)
        txData[0] := txData[0] + 1;
        sendTo( rq := true, chanCode := ETH1_UNI0,
              lenTx := sizeof(txData), data := void(txData),
              error => lastErr);
        IF sendTo.mesSent THEN
          cntOK := cntOK + 1;
        END_IF;
        state := closeCon;
      ELSE
        chkTim(IN := true, PT := T#20s);
        IF chkTim.Q THEN
          lastErr := COM_ERR64;
          state := closeCon; cntERR := cntERR + 1;
        END_IF;
      END_IF;

    closeCon :
      CloseTCPconnection(chanCode := ETH1_UNI0);
      state := idle;
  END_CASE;

  lastErrTxt := GetLastComErrTxt(errCode := lastErr);
END_PROGRAM

```

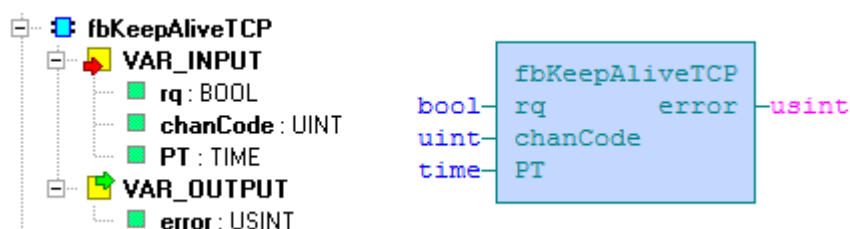
Poznámka.

V případě, že budou TCP protokolem vysílány větší objemy dat (např. soubory uložené na paměťové kartě), pak je třeba při vysílání kontrolovat výstupní proměnnou *error* funkčního bloku *sendTo*. Může se totiž stát, že se tok dat TCP protokolem na chvíli pozastaví (třeba proto, že server na který data odesíláme, je zaneprázdněn a na chvíli přestane potvrzovat přijaté pakety) následkem čehož může dojít k zaplnění vysílacích zásobníků komunikačního kanálu. V tomto případě funkční blok *sendTo* nastaví výstupní proměnnou *error* na hodnotu *COM_ERR5*, což znamená že jsou vysílací zásobníky plné. V této chvíli je třeba pozastavit předávání nových dat do funkčního bloku *sendTo* až do chvíle, kdy proměnná *error* bude mít hodnotu *COM_OK*. Pokud by se předávání nových dat do bloku *sendTo* nepozastavilo, došlo by ke ztrátě bloků dat předaných ve chvíli, kdy byla v proměnné *error* hodnota *COM_ERR5*.

Signalize přeplnění vysílacích zásobníků v režimu TCP uni master je podporovaná ve firmware systému Foxtrot od verze v5.3 a v procesorech CP-7004 od v5.3.

Viz také Funkční blok *fbRecvFrom*

6.3 Funkční blok *fbKeepAliveTCP*

Knihovna : *ComLib*

Funkční blok *fbKeepAliveTCP* slouží k udržování navázaného TCP spojení v případě, že aktuálně nemáme žádná data k vysílání a nechceme, aby došlo k uzavření TCP spojení. Funkční blok *fbKeepAliveTCP* odesílá pro udržení spojení ACK pakety s nulovou délkou. Interval odesílání ACK paketů je daný vstupem *PT*. Vstup *chanCode* určuje komunikační kanál, který musí být v režimu *uni* se zapnutým protokolem *TCP*.

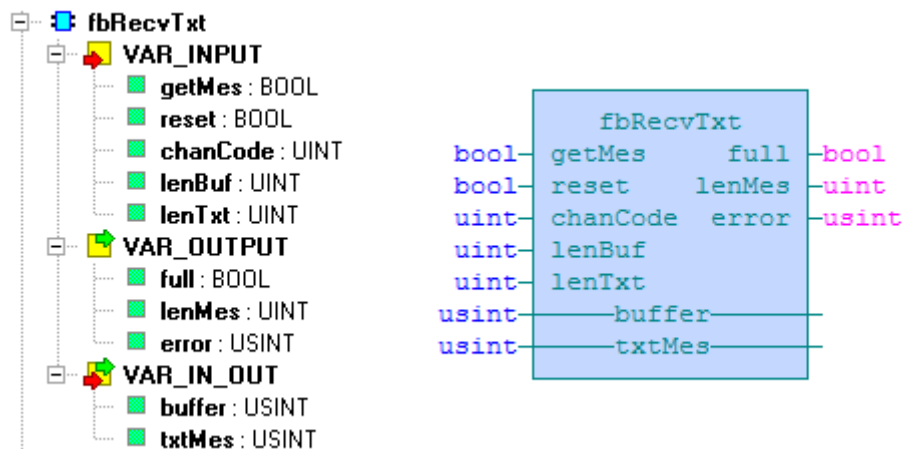
Tento funkční blok je podporován na centrálních jednotkách řady K a L (TC700 CP-7004, CP-7007, všechny verze CPU Foxtrot) od verze v7.7. Podporováno je pouze rozhraní ETH1. Do knihovny *ComLib* byl zařazen od v2.1.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>rq</i>	BOOL	Žádost o udržování spojení
	<i>chanCode</i>	UINT	Kód komunikačního kanálu <i>ETH1_uni0</i> ethernet kanál ETH1, spojení uni0 <i>ETH1_uni7</i> ethernet kanál ETH1, spojení uni7
	<i>PT</i>	TIME	Interval odesílání udržovacích paketů
VAR_OUTPUT			
	<i>error</i>	USINT	Chybový kód. Je-li 0 (COM_OK) tak nedošlo k chybě V případě chyby vrací kódy COM_ERR1, ..., COM_ERR64

Viz také Funkční blok *fbSendTo*

6.4 Funkční blok *fbRecvFromTxt*

Knihovna : *ComLib*

Funkční blok *fbRecvFromTxt* slouží k příjmu textových dat ze sériového kanálu nebo z ethernetu. Komunikační kanál musí být nastaven v režimu *uni*. Data jsou plynule ukládána do pracovního bufferu v pořadí, v jakém jsou přijímána. Při každém volání bloku s nastaveným vstupem *getMes* na hodnotu TRUE se zkontrolují všechna dosud přijatá data v pracovním bufferu a pokud obsahují znaky CR LF (16#0D, 16#0A), tak se přesune část přijatých dat (od začátku bufferu až do znaků CR LF včetně) do proměnné *txtMes*. Počet přesunutých znaků je pak nastaven do výstupní proměnné *lenMes*. Pokud není přijatá žádná věta (sekvence znaků ukončená CR LF) tak je výstupní proměnná *lenMes* = 0. Vstupní proměnné *lenBuf* a *lenTxt* udávají velikosti proměnných, které jsou použity pro pracovní buffer resp. pro jednu přijatou větu.

Jinými slovy funkční blok *fbRecvFromTxt* přijímá textová data a vrací je po jednotlivých větách (řádcích). Na každé volání bloku je vrácena jedna věta.

Tento funkční blok je podporován na centrálních jednotkách řady K a L (TC700 CP-7004, CP-7007, všechny verze CPU Foxtrot) od verze v4.4.

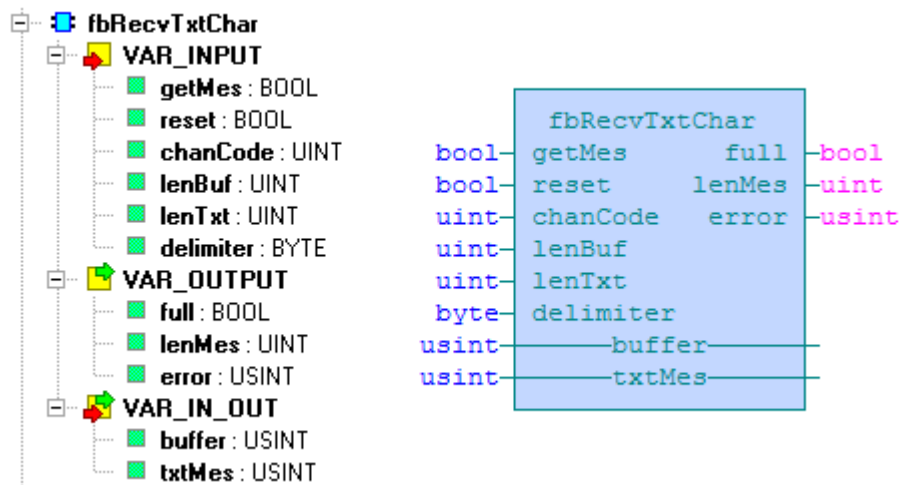
Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>getMes</i>	BOOL	Požadavek na jednu přijatou větu
	<i>reset</i>	BOOL	Vymazat pracovní buffer
	<i>chanCode</i>	UINT	Kód komunikačního kanálu <i>ETH1_uni0</i> ethernet kanál ETH1, spojení uni0 <i>ETH1_uni7</i> ethernet kanál ETH1, spojení uni7 <i>ETH2_uni0</i> ethernet kanál ETH2, spojení uni0 <i>ETH4_uni7</i> ethernet kanál ETH4, spojení uni7 <i>CH1_uni</i> sériový kanál CH1, režim uni

	Proměnná	Typ	Význam
			<i>CH10_uni</i> sériový kanál CH10, režim uni
	<i>lenBuf</i>	UINT	Délka pracovního bufferu (počet bytů)
	<i>lenTxt</i>	UINT	Délka proměnné <i>txtMes</i> (počet znaků)
VAR_IN_OUT			
	<i>buffer</i>	USINT	Odkaz na první byte pracovního bufferu
	<i>txtMes</i>	USINT	Odkaz na první byte proměnné, do které bude uložena přijatá věta
VAR_OUTPUT			
	<i>full</i>	BOOL	Pracovní buffer je přeplněn Zprávy nebyly odebírány dostatečně rychle
	<i>lenMes</i>	UINT	Délka přijaté věty (počet znaků včetně CR LF)
	<i>error</i>	USINT	Chybový kód. Je-li 0 (COM_OK), příjem proběhl bez chyby V případě chyby vrací kódy COM_ERR1, ..., COM_ERR64

Viz také Funkční blok `fbRecvFrom`, Funkční blok `fbRecvFromChar`

6.5 Funkční blok *fbRecvFromChar*

Knihovna : *ComLib*

Funkční blok *fbRecvFromChar* slouží k příjmu textových dat ze sériového kanálu nebo z ethernetu. Komunikační kanál musí být nastaven v režimu *uni*. Data jsou plynule ukládána do pracovního bufferu v pořadí, v jakém jsou přijímána. Při každém volání bloku s nastaveným vstupem *getMes* na hodnotu TRUE se zkontrolují všechna dosud přijatá data v pracovním bufferu a pokud obsahují znak definovaný vstupem *delimiter*, tak se přesune část přijatých dat (od začátku bufferu až do znaku *delimiter* včetně) do proměnné *txtMes*. Počet přesunutých znaků je pak nastaven do výstupní proměnné *lenMes*. Pokud není přijatá žádná věta (sekvence znaků ukončená znakem *delimiter*) tak je výstupní proměnná *lenMes* = 0. Vstupní proměnné *lenBuf* a *lenTxt* udávají velikosti proměnných, které jsou použity pro pracovní buffer resp. pro jednu přijatou větu.

Jinými slovy funkční blok *fbRecvFromChar* přijímá textová data a vrací je po jednotlivých větách (řádcích). Na každé volání bloku je vrácena jedna věta.

Tento funkční blok je podporován na centrálních jednotkách řady K a L (TC700 CP-7004, CP-7007, všechny verze CPU Foxtrot) od verze v4.4.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>getMes</i>	BOOL	Požadavek na jednu přijatou větu
	<i>reset</i>	BOOL	Vymazat pracovní buffer
	<i>chanCode</i>	UINT	Kód komunikačního kanálu <i>ETH1_uni0</i> ethernet kanál ETH1, spojení uni0 <i>ETH1_uni7</i> ethernet kanál ETH1, spojení uni7 <i>ETH2_uni0</i> ethernet kanál ETH2, spojení uni0 <i>ETH4_uni7</i> ethernet kanál ETH4, spojení uni7 <i>CH1_uni</i> sériový kanál CH1, režim uni

	Proměnná	Typ	Význam
			 <i>CH10_uni</i> sériový kanál CH10, režim uni
	<i>lenBuf</i>	UINT	Délka pracovního bufferu (počet bytů)
	<i>lenTxt</i>	UINT	Délka proměnné <i>txtMes</i> (počet znaků)
	<i>delimiter</i>	BYTE	Kód znaku, který ukončuje jednu větu
VAR_IN_OUT			
	<i>buffer</i>	USINT	Odkaz na první byte pracovního bufferu
	<i>txtMes</i>	USINT	Odkaz na první byte proměnné, do které bude uložena přijatá věta
VAR_OUTPUT			
	<i>full</i>	BOOL	Pracovní buffer je přeplněn Věty nebyly odebírány dostatečně rychle
	<i>lenMes</i>	UINT	Délka přijaté věty (počet znaků včetně znaku <i>delimiter</i>)
	<i>error</i>	USINT	Chybový kód. Je-li 0 (COM_OK), příjem proběhl bez chyby V případě chyby vrací kódy COM_ERR1, ..., COM_ERR64

Viz také Funkční blok fbRecvFrom, Funkční blok fbRecvFromTxt