SÉRIOVÁ KOMUNIKACE PROGRAMOVATELNÝCH AUTOMATŮ TECOMAT FOXTROT 2

SÉRIOVÁ KOMUNIKACE PROGRAMOVATELNÝCH AUTOMATŮ TECOMAT FOXTROT 2

TXV 004 69

3. vydání - duben 2021

OBSAH

1. PŘEHLED KOMUNIKAČNÍCH MOŽNOSTÍ SYSTÉMŮ TECOMAT 4
1.1. Přehled režimů komunikací 4
1.2. Komunikační možnosti jednotlivých typů PLC 5
1.2.1. TECOMAT FOXTROT 2 5

2. KOMUNIKAČNÍ REŽIMY SÉRIOVÝCH KANÁLŮ 7
2.1. Režim UNI - obecný uživatelský kanál 7
2.1.1. Konfigurace 7
2.1.2. Poskytovaná data 12
2.1.3. Vysílání a příjem dat 16
2.1.4. Ovládání modemových signálů 17
2.1.5. Klid na lince 18
2.1.6. Chybová hlášení 19
2.1.7. Monitorování provozu sériového kanálu 20
2.2. Režim PC - komunikace s nadřízeným systémem 22
2.2.1. Konfigurace 22
2.2.2. Poskytovaná data 25
2.2.3. Chybová hlášení 26
2.3. Režim CSJ - připojení sběrnice CAN 27
2.3.1. Konfigurace 27
2.3.2. Poskytovaná data 33
2.3.2.1. Struktura zpráv 37
2.3.3. Výměna dat 39
2.3.4. Režimy řadiče a chybové stavy 41

3. KOMUNIKAČNÍ REŽIMY ROZHRANÍ ETHERNET, WLAN A LTE 43
3.1. Režim UNI - obecný uživatelský kanál 45
3.1.1. Konfigurace 45
3.1.2. Poskytovaná data 49
3.1.3. Vysílání a příjem dat 52
3.1.4. Nastavení parametrů spojení a ovládání rozhraní 53
3.1.5. Chybová hlášení 54
3.2. Režim PC - komunikace s nadřízeným systémem 55
3.3. Režim PLC - síť se sdílením dat 56
3.3.1. Režim PLC a PLCx - siť se sdílením dat 57
3.3.2. Režim PLD - siť s rozšířeným sdílením dat 58
3.3.2.1. Založení sítě v režimu PLD 58
3.3.2.2. Přidání dalšího účastníka do sítě v režimu PLD 64
3.3.2.3. Provoz sítě v režimu PLD 67
3.3.2.4. Síť v režimu PLD v simulátoru PLC 73

4. KOMUNIKAČNÍ REŽIMY ROZHRANÍ USB 75
4.1. Režim PC - komunikace s nadřízeným systémem 75

5. PROTOKOL EPSNET 76
5.1. Obecná struktura protokolu EPSNET 76
5.2. Podmínky komunikace 77
5.3. Odpovědi a chybová hlášení 78
5.4. Struktura protokolu EPSNET UDP / TCP 80
5.5. Komunikační služby 81
5.6. Popis komunikačních služeb 81
5.6.1. Zahájení komunikace 82
CONNECT - navázání spojení 82
IDENT - identifikace systému 83
5.6.2. Provozní služby 84
SETTID - nastavení času 84
SETCW - nastavení řídícího slova 85
GETSW - přečtení stavového slova 86
GETERR - přečtení chybového zásobníku 87
MASKCW - nastavení jednotlivých bitů řídícího slova 88
5.6.3. Datové služby 89
READN - čtení z datové paměti 89
WRITEN - zápis do datové paměti 90
WANDRN - zápis do a čtení dat z datové paměti 91
READB - čtení bitů z datové paměti 92
WRITEB - zápis do bitů datové paměti 93
READBD - destruktivní čtení bitů z datové paměti 94
READND - destruktivní čtení z datové paměti 95
WANDRND - zápis do a destruktivní čtení dat z datové paměti 96

1. PŘEHLED KOMUNIKAČNÍCH MOŽNOSTÍ SYSTÉMŮ TECOMAT

1.1. PŘEHLED REŽIMŮ KOMUNIKACÍ

Určení příručky pro jednotlivé systémy

Tato příručka je v plném rozsahu platná pouze pro programovatelné automaty TECOMAT FOXTROT 2.

Možnosti sériové komunikace programovatelných automatů TECOMAT se dále liší podle typu systému a typu použité centrální jednotky.

Komunikační kanály

Komunikační kanály PLC TECOMAT FOXTROT 2 lze rozdělit do následujících skupin:

- sériové kanály (CHx)
- rozhraní Ethernet (ETHx), WiFi (WLANx), LTE (LTEx)
- rozhraní USB (USBx)

Sériové kanály

Sériové kanály se označují CH1 až CH10. Číslo si volí uživatel v rámci konfigurace systému v projektu uživatelského programu.

Sériové kanály mohou pracovat v následujících režimech:

režim **UNI** - obecný uživatelský kanál pro univerzální použití (připojení frekvenčních měničů, operátorských panelů, čteček čárového kódu, inteligentních čidel, apod. - kap.2.1.)

režim **PC** - připojení nadřízeného systému, zpravidla počítače PC, pomocí protokolu EPSNET, k dispozici jsou veřejné i systémové služby (kap.2.2.)

režim **CSJ** - připojení sběrnice CAN (kap.2.3.)

Dostupnost výše uvedených režimů pro jednotlivé sériové kanály je závislá na typu osazeného submodulu nebo připojeného komunikačního modulu.

Rozhraní Ethernet, WiFi, LTE

PLC obsahují dvě nezávislá rozhraní Ethernet označená ETH1 a ETH2, dále volitelně rozhraní WiFi označené WLAN1 a volitelně rozhraní LTE označené LTE1. Další rozhraní WLAN2 lze realizovat připojením externího WiFi adapteru přes rozhraní USB host.

Tato rozhraní mohou pracovat v následujících režimech:

režim **UNI** - vysílání a příjem libovolných dat protokoly UDP a TCP (kap.3.1.)

režim **PC** - připojení nadřízeného systému, zpravidla počítače PC, pomocí protokolu EPSNET UDP/TCP, k dispozici jsou veřejné i systémové služby (kap.3.2.)

režim **PLC** - propojení více systémů TECOMAT pomocí protokolu EPSNET UDP za účelem rychlého sdílení dat (kap.3.3.) - kompatibilní s režimem **PLC** v systémech FOXTROT (CP-1xxx) a TC700

režim **PLD** - sdílení dat mezi PLC systémy pomocí rozšířeného protokolu EPSNET UDP (kap.3.4.) - určeno pouze pro systémy FOXTROT 2

Rozhraní LTE nepodporuje režimy **PLC** a **PLD**.

Každé rozhraní může pracovat v několika režimech současně. Režim **PC** je trvale aktivní, ostatní jsou volitelné. Rozhraní lze tedy využít i k programování a ladění pomocí vývojového prostředí Mosaic.

Rozhraní USB

Rozhraní USB device je určeno pro výměnu dat protokolem EPSNET (režim **PC** - kap.4.1.), k dispozici jsou veřejné i systémové služby, takže rozhraní lze použít k programování a ladění pomocí vývojového prostředí Mosaic.

1.2. KOMUNIKAČNÍ MOŽNOSTI JEDNOTLIVÝCH TYPŮ PLC

1.2.1. TECOMAT FOXTROT 2

TECOMAT FOXTROT 2 je kompaktní programovatelný automat s binárními a analogovými vstupy a výstupy a s možností rozšíření. Obsahuje centrální jednotku řady I.

Základní modul lze osadit až 4 sériovými kanály pomocí výměnných submodulů MR-0130 - MR-0134. Jedná se o standardní sériové kanály s rozhraním RS-232 nebo RS-485.

Dalších 6 sériových kanálů pak lze přidat pomocí samostatných komunikačních modulů SC-11xx na každou linku sběrnice TCL2. Tyto moduly obsahují jak standardní sériové kanály, tak připojení ke sběrnici CAN nebo do bezdrátové sítě.

Základní modul CP-2000 má navíc 1 interní sériový kanál, nicméně celkový maximální počet obsluhovaných sériových kanálů zůstává 10.

Základní moduly PLC obsahují dvě nezávislá rozhraní Ethernet označená ETH1 a ETH2, dále volitelně rozhraní WiFi označené WLAN1 a volitelně rozhraní LTE označené LTE1. Další rozhraní WLAN2 lze realizovat připojením externího WiFi adapteru přes rozhraní USB1.

Rozhraní USB2 je určeno pro výměnu dat protokolem EPSNET, k dispozici jsou veřejné i systémové služby, takže rozhraní USB2 lze použít k programování a ladění pomocí vývojového prostředí Mosaic.

Podrobnosti jsou uvedeny v příručce Programovatelné automaty TECOMAT FOXTROT, obj. č. TXV 004 50.01.

Komunikační možnosti jednotlivých centrálních jednotek jsou přehledně uvedeny v tab.1.1.

Tab.1.1 Přehled komunikačních možností PLC TECOMAT FOXTROT 2

Typ modulu	**CP-2000**	**CP-2005 CP-2007 CP-2080 CP-2090**
Sériové kanály (CH1 – CH10):		
- na základním modulu	1	-
- dostupné režimy	UNI, PC	
- přidané pomocí submodulů MR-013x	4	4
- dostupné režimy	UNI, PC	UNI, PC
- přidané pomocí komunikačních modulů SC-11xx na sběrnici TCL2	max. 6 na jednu linku	max. 6 na jednu linku
- dostupné režimy (SC-1101, SC-1111, SC-1112)	UNI	UNI
- dostupné režimy (SC-1102)	CSJ	CSJ
Rozhraní USB device:		
- na základním modulu (USB2)	1	
- dostupné režimy	PC	
Rozhraní Ethernet:		
- na základním modulu	2	
- dostupné režimy	UNI, PC, PLC, PLD	
Rozhraní WLAN:		
- rozhraní WLAN na základním modulu (WLAN1)	1 (podle varianty)	
- rozhraní WLAN přes adapter na USB1 (WLAN2)	1	
- dostupné režimy	UNI, PC, PLC, PLD	
Rozhraní LTE:		
- rozhraní LTE na základním modulu (LTE1)	1 (podle varianty)	
- dostupné režimy	UNI, PC	

2. KOMUNIKAČNÍ REŽIMY SÉRIOVÝCH KANÁLŮ

Sériové kanály jsou realizovány pomocí submodulů vložených do základního modulu PLC, nebo pomocí externích samostatných modulů. Tím je dáno i fyzické rozhraní (RS-232, RS-485, bezdrátová síť, apod.). Některé moduly nejsou prostým převodníkem sériové linky, ale obsahují řadič, který umožňuje připojení k příslušné sběrnici (např. CAN). Nicméně pokud mluvíme souhrnně o sériových kanálech, zahrnujeme mezi ně i komunikaci přes tyto moduly.

V následujících kapitolách jsou popsány jednotlivé komunikační režimy použitelné na sériových kanálech (tab.2.1).

Tab.2.1 Přehled komunikačních režimů sériových kanálů

Komunikační režim	Funkce	Popis
UNI	obecný uživatelský kanál	kap.2.1.
PC	komunikace s nadřízeným systémem	kap.2.2.
CSJ	připojení sběrnice CAN	kap.2.3.

2.1. REŽIM UNI - OBECNÝ UŽIVATELSKÝ KANÁL

Režim **UNI** je určen pro univerzální použití. Pokud je komunikační kanál v tomto režimu, pak lze tímto kanálem z uživatelského programu odesílat a přijímat libovolná data. K obsluze vysílání a příjmu jsou určeny bloky *fbSendTo* a *fbRecvFrom* z knihovny *ComLib*.

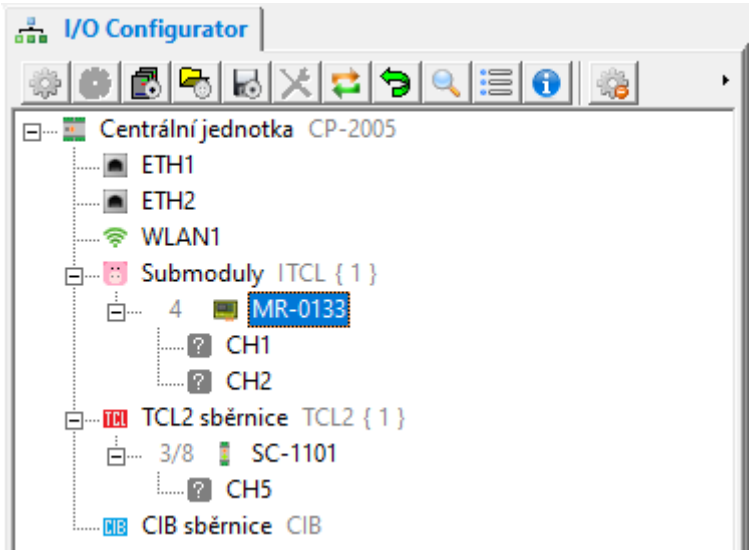
Sériové kanály v tomto režimu umožňují přenosovou rychlost do 345,6 kBd a přenos až 512 bytů v jedné zprávě. Čím vyšší přenosová rychlost, tím kratší dobu trvá přenos dat. Sériové kanály na submodulech nebo na komunikačních modulech sice nezatěžují centrální jednotku během vlastní komunikace na sériovém kanálu, avšak prodlužují dobu cyklu PLC při předávání dat mezi komunikačním modulem a centrální jednotkou po systémové sběrnici. U submodulů MR-013x, které pro komunikaci s centrální jednotkou používají výkonnou interní sběrnici, je toto prodloužení řádově ve stovkách μs. U komunikačních modulů SC-11xx připojených na sběrnici TCL2 je toto prodloužení řádově v jednotkách až desítkách ms.

2.1.1. Konfigurace

Konfigurace komunikačních kanálů stejně jako konfigurace všech částí PLC TECOMAT FOXTROT 2 se v prostředí Mosaic provádí pomocí nástroje *I/O Configurator*.

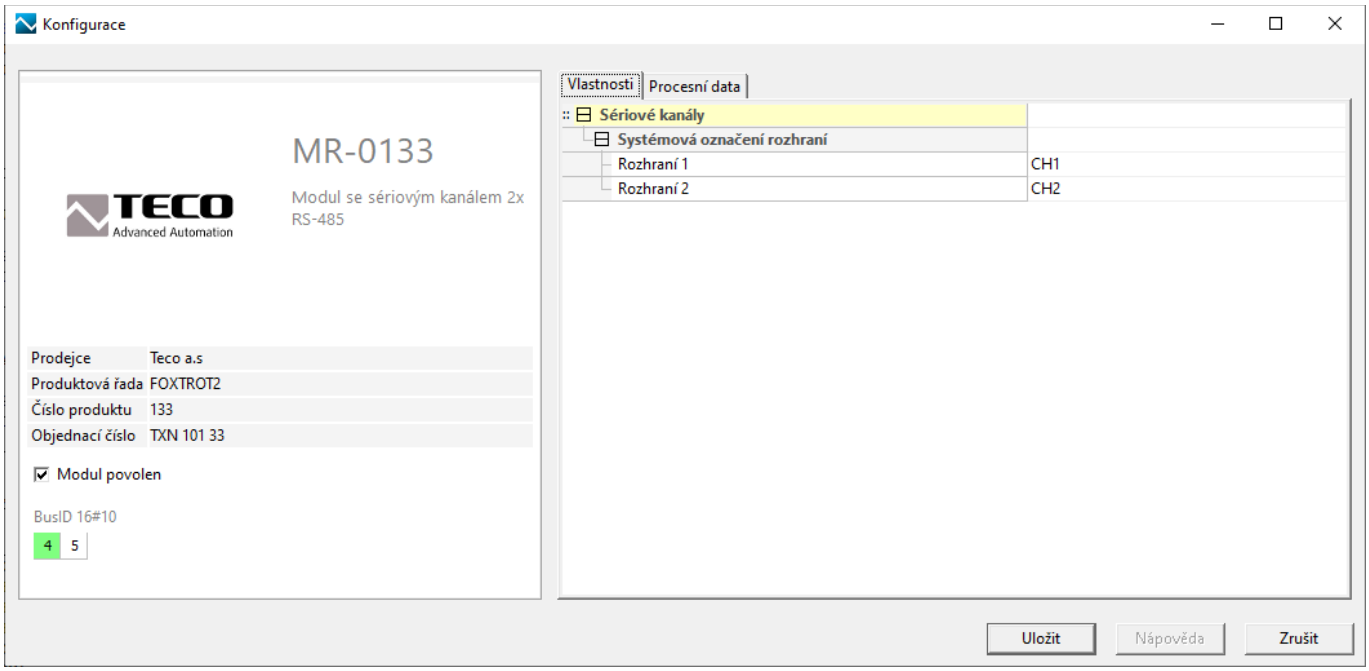
Volba čísla kanálu

Při vkládání submodulu nebo komunikačního modulu do konfigurace, případně při načtení konfigurace připojeného PLC nástroj *I/O Configurator* přidělí jednotlivým komunikačním kanálům označení. Označení sériových kanálů, které jsou realizovány pomocí submodulů a samostatných komunikačních modulů, může uživatel změnit.

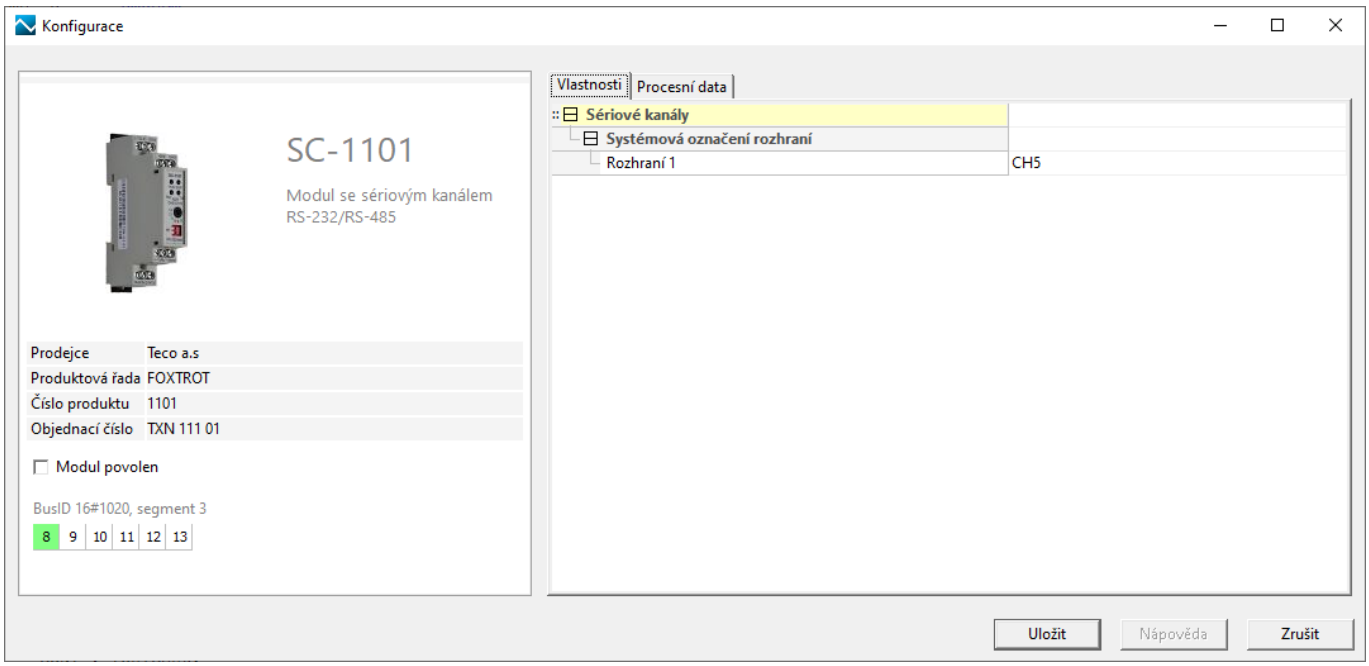


Obr.2.1 Příklad konfigurace se sériovými kanály bez nastavených režimů

Poklepáním levým tlačítkem myši na názvu submodulu nebo komunikačního modulu se otevře konfigurační okno, ve kterém můžeme nastavit označení sériového kanálu v rozsahu CH1 až CH10 (obr.2.2, obr.2.3). Každý kanál musí mít jiné označení.



Obr.2.2 Konfigurace submodulu MR-0133 se dvěma sériovými kanály

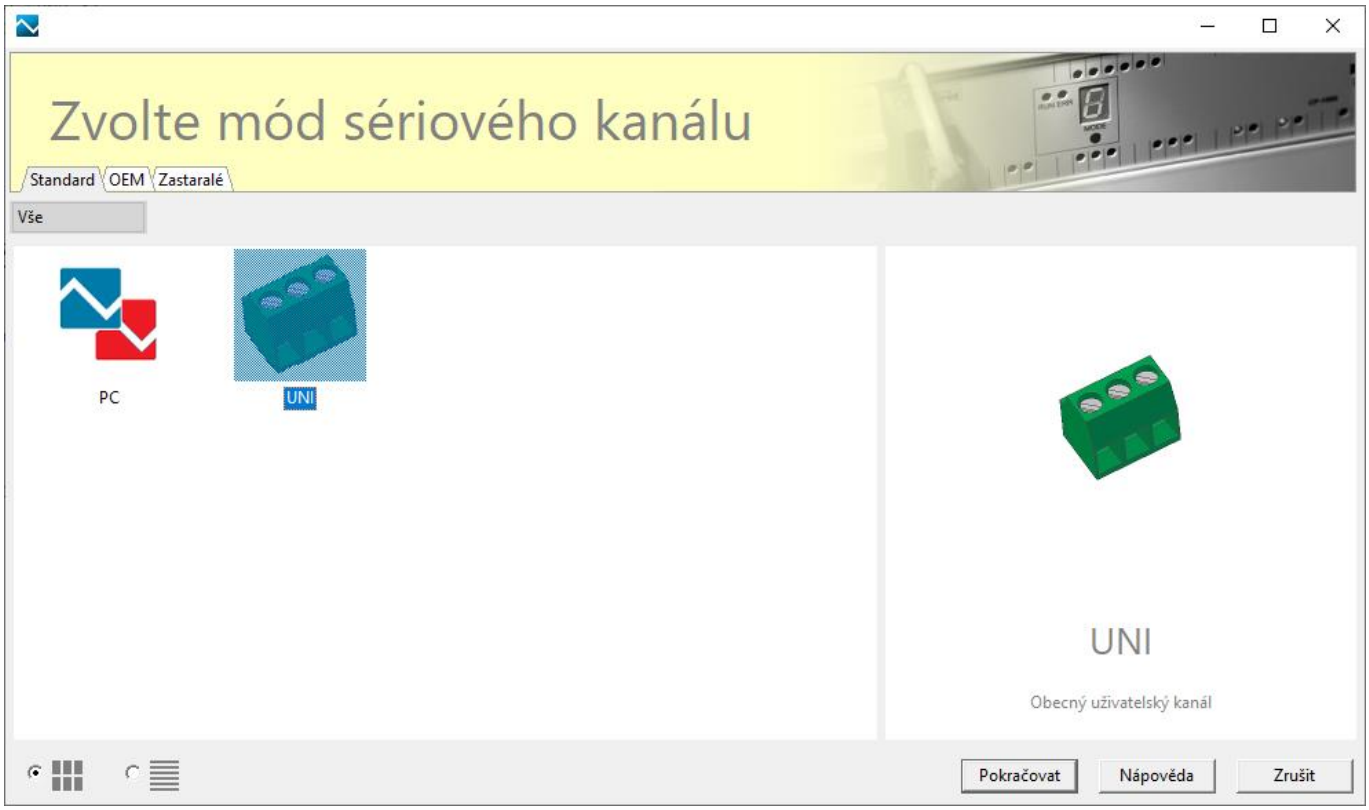


Obr.2.3 Konfigurace modulu SC-1101 s jedním sériovým kanálem

Nastavení režimu kanálu

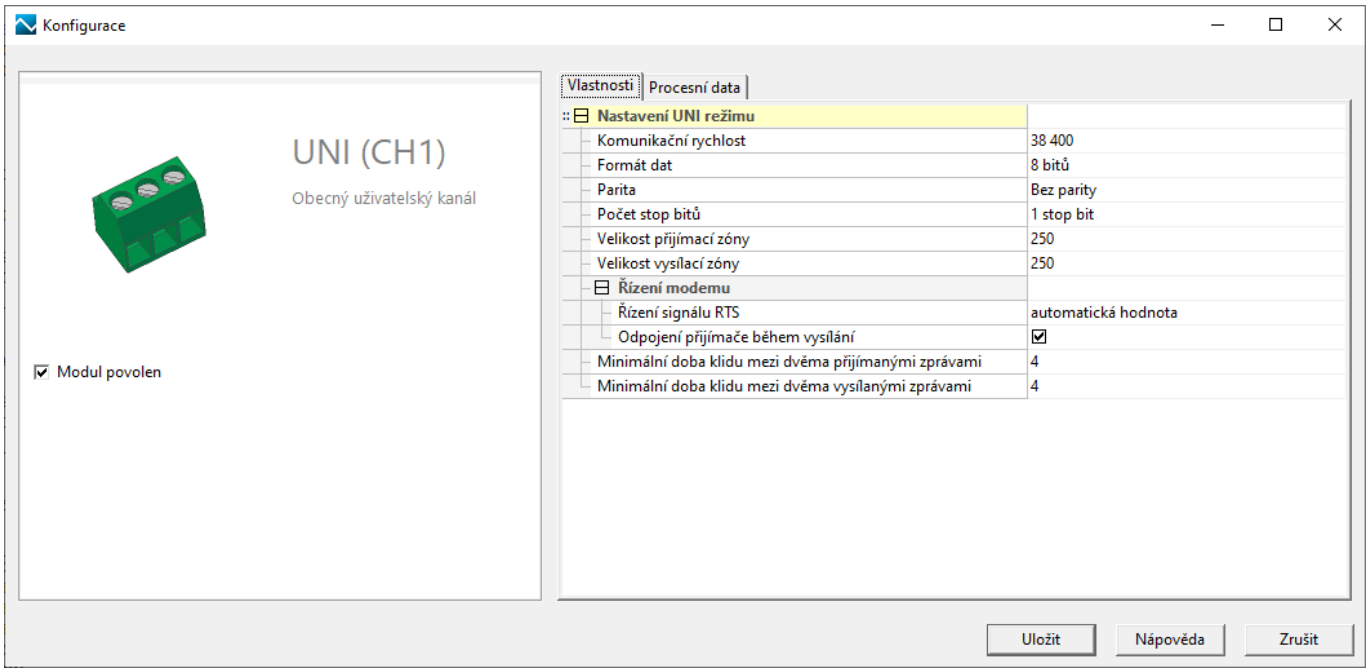
Režim kanálu nastavíme tak, že poklepeme levým tlačítkem myši na označení příslušného kanálu (např. CH1). Otevře se okno výběru režimu sériového kanálu (obr.2.4).

Sériové kanály na základním modulu (interní) a sériové kanály na submodulech nabízejí režimy **PC** a **UNI**, zatímco sériové kanály na samostatných modulech SC-11xx nabízejí pouze režim **UNI**.



Obr.2.4 Výběr režimu sériového kanálu

Poklepáním vybereme režim **UNI** a otevře se panel konfigurace (obr.2.5).



Obr.2.5 Konfigurace režimu UNI pro sériový kanál

V levé části panelu *Konfigurace* se nachází položka *Modul povolen*. Pokud je zaškrtnuta, provádí se obsluha kanálu podle nastavení dostupných v pravé části panelu. Pokud položka nebude zaškrtnutá, je obsluha kanálu vypnuta beze ztráty aktuálního nastavení. To lze využít při ladění uživatelského programu.

V pravé části panelu se v záložce *Vlastnosti* nachází všechny položky konfigurace daného kanálu. V záložce *Procesní data* pak nalezneme seznam všech proměnných, které kanál poskytuje, s možností jejich vlastního pojmenování (viz kap.2.1.2).

Význam a možnosti nastavení jednotlivých položek v pravé části panelu jsou následující:

Komunikační rychlost

Volba přenosové rychlosti v Bd (bity za sekundu).

Formát dat

Počet přenášených bitů v jednom bytu dat (8 nebo 7 bitů).

Parita

Nastavení parity:

bez parity - paritní bit není přenášen (pouze pro formát dat 8 bitů)
lichá parita - hodnota paritního bitu je taková, aby počet jedniček datových bitů a paritního bitu v jednom bytu byl lichý
sudá parita - hodnota paritního bitu je taková, aby počet jedniček datových bitů a paritního bitu v jednom bytu byl sudý
parita trvale 0 - hodnota paritního bitu je trvale log.0
parita trvale 1 - hodnota paritního bitu je trvale log.1

Není možné nastavit formát 7 bitů bez parity. Paritu lze vypnout jen pro formát 8 bitů.

Počet stop bitů

Počet stop bitů (1 nebo 2 bity).

Velikost přijímací zóny

Velikost přijímací zóny v bytech, max. 512 B.

Velikost vysílací zóny

Velikost vysílací zóny v bytech, max. 512 B.

Řízení signálu RTS

Nastavení režimu řízení modemového signálu RTS:

automatická hodnota - signál je ovládán vysílačem sériového kanálu, po dobu vysílání má hodnotu log.0

trvale hodnota 0 - signál má trvalou hodnotu log.0

trvale hodnota 1 - signál má trvalou hodnotu log.1

podle SIGN.1 - signál je ovládán z uživatelského programu PLC příslušným bitem

automatická hodnota s detekcí CTS - signál je ovládán vysílačem sériového kanálu, po dobu vysílání má hodnotu log.0, vysílání je podmíněno nastavením signálu CTS na log.0

Odpojení přijímače během vysílání

Při zaškrtnutí políčka je po dobu vysílání vypnut přijímač. Pokud je připojeno nějaké zařízení, které způsobuje echování vysílaných znaků, zamezí se tak jejich nežádoucímu příjmu.

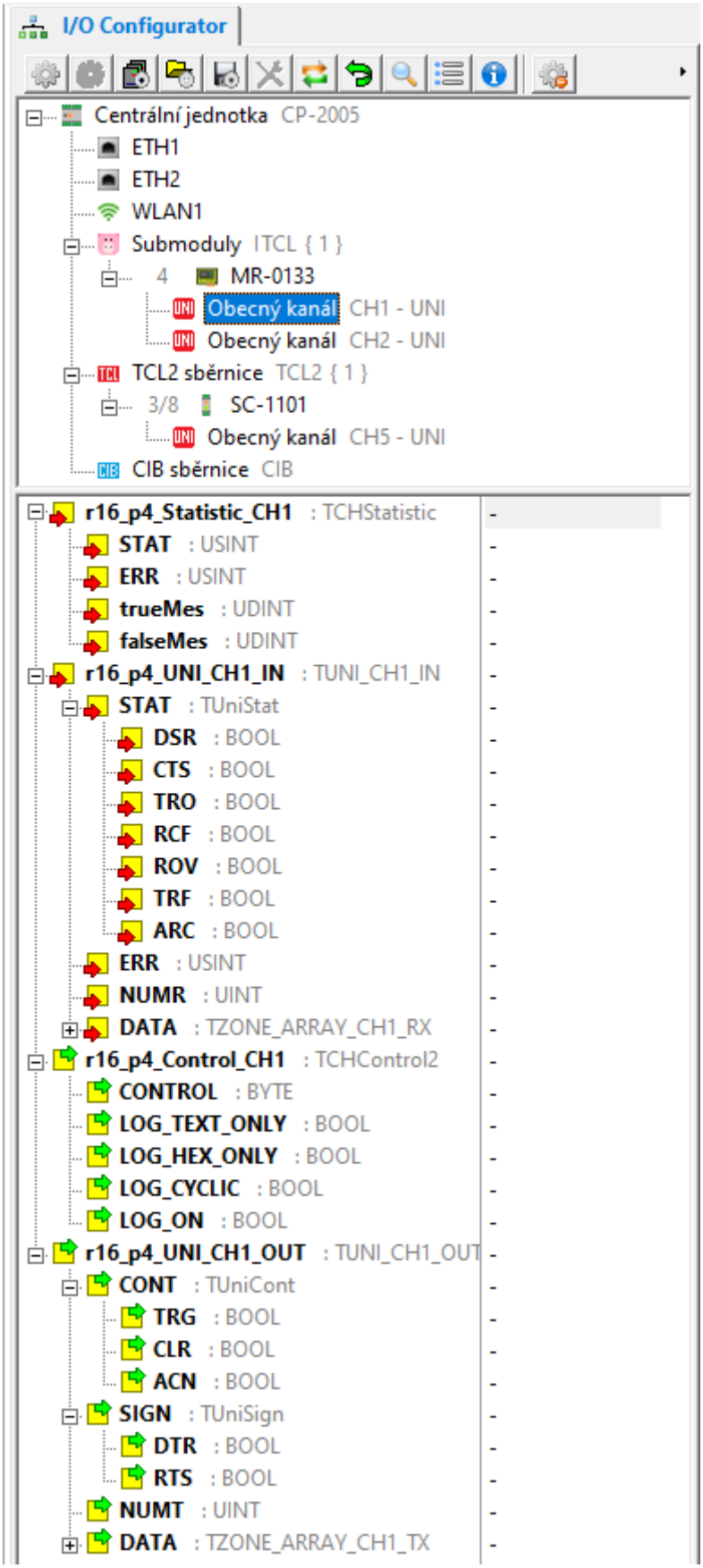
Minimální doba klidu mezi přijímanými zprávami

Minimální doba klidu na lince odpovídající počtu přijatých bytů, která je definována mezi dvěma přijatými zprávami. Po uplynutí této doby je přijatá zpráva považována za celou. Tento parametr umožňuje příjem zpráv různé délky. Pokud je tento parametr 0, přijaté znaky se neskládají do ucelených zpráv, ale předávají se do zápisníku PLC bezprostředně v každém cyklu programu.

Minimální doba klidu mezi vysílanými zprávami

Minimální doba klidu na lince odpovídající počtu vysílaných bytů, která je definována mezi dvěma vysílanými zprávami. Tento parametr zabezpečí, že mezi dvěma vysílanými zprávami bude dodržen klid na lince minimálně této délky.

Po stisknutí tlačítka *Uložit* se ve stromečku konfigurace na řádku příslušného kanálu objeví označení aktuálně nastaveného režimu (obr.2.6).



Obr.2.6 Příklad konfigurace se sériovými kanály s nastavenými režimy UNI

2.1.2. Poskytovaná data

Obsluha komunikačního kanálu v režimu **UNI** probíhá prostřednictvím komunikačních zón v zápisníkové paměti. Struktura dat je uvedena v tab.2.2.

K obsluze komunikačního kanálu v uživatelském programu slouží knihovna *ComLib*. Doporučujeme její použití z důvodu vyvarování se problémů s obsluhou příznaků kanálu v reálném čase. Následující informace jsou určeny především pro ty, kteří knihovnu nepoužijí.

Položky struktury mají přidělena symbolická jména, která začínají vždy stejnou kombinací znaků určujících adresu a označení kanálu v rámci systému. Ve sloupci *Úplný zápis* je uvedeno vždy konkrétní symbolické jméno pro danou položku. Pokud chceme data použít v uživatelském programu, použijeme buď toto symbolické jméno, nebo ve sloupci *Alias* zapíšeme svoje symbolické jméno, které pak můžeme používat. V žádném případě nepoužíváme absolutní operandy, protože se mohou po novém překladu uživatelského programu změnit.

Alias vytvoříme v nástroji *I/O Configurator* následovně. Poklepáním na jméno modulu ve stromečku konfigurace otevřeme panel *Konfigurace*. V záložce *Procesní data* zapíšeme požadované jméno v příslušném řádku do sloupce *Alias*.

Uspořádání diagnostických dat a komunikačních zón

Komunikační kanál v režimu **UNI** zveřejňuje diagnostická data stavu linky, přijímaná a vysílaná data a stav komunikace. Tyto údaje se ukládají do zápisníku a jsou snadno přístupné v nástroji *I/O Configurator*.

Do vysílací zóny zapisuje uživatel data určená k odeslání. V přijímací zóně jsou data přijatá. Před jejich zpracováním je třeba zkontrolovat hodnotu statusu.

Tab.2.2 Struktura dat (n je číslo sériového kanálu)

Struktura dat	Úplný zápis	Popis objektu
Statistic_CHn	rx_py_Statistic_CHn	diagnostická zóna kanálu
STAT: usint	rx_py_Statistic_CHn.STAT	
ERR: usint	rx_py_Statistic_CHn.ERR	
trueMess: udint	rx_py_Statistic_CHn.trueMess	
falseMess: udint	rx_py_Statistic_CHn.falseMess	
UNI_CHn_IN	rx_py_UNI_CHn_IN	přijímací zóna
STAT	rx_py_UNI_CHn_IN.STAT	
DSR: bool	rx_py_UNI_CHn_IN.STAT.DSR	
CTS: bool	rx_py_UNI_CHn_IN.STAT.CTS	
TRO: bool	rx_py_UNI_CHn_IN.STAT.TRO	
RCF: bool	rx_py_UNI_CHn_IN.STAT.RCF	
ROV: bool	rx_py_UNI_CHn_IN.STAT.ROV	
TRF: bool	rx_py_UNI_CHn_IN.STAT.TRF	
ARC: bool	rx_py_UNI_CHn_IN.STAT.ARC	
ERR: usint	rx_py_UNI_CHn_IN.ERR	
NUMR: uint	rx_py_UNI_CHn_IN.NUMR	
DATA[...]: usint	rx_py_UNI_CHn_IN.DATA	
Control_CHn	rx_py_Control_CHn	řídicí zóna kanálu
CONTROL: byte	rx_py_Control_CHn.CONTROL	
LOG_TEXT_ONLY: bool	rx_py_Control_CHn.LOG_TEXT_ONLY	
LOG_HEX_ONLY: bool	rx_py_Control_CHn.LOG_HEX_ONLY	
LOG_CYCLIC: bool	rx_py_Control_CHn.LOG_CYCLIC	
LOG_ON: bool	rx_py_Control_CHn.LOG_ON	
UNI_CHn_OUT	rx_py_UNI_CHn_OUT	vysílací zóna
CONT	rx_py_UNI_CHn_OUT.CONT	
TRG: bool	rx_py_UNI_CHn_OUT.CONT.TRG	
CLR: bool	rx_py_UNI_CHn_OUT.CONT.CLR	
ACN: bool	rx_py_UNI_CHn_OUT.CONT.ACN	
SIGN	rx_py_UNI_CHn_OUT.SIGN	
DTR: bool	rx_py_UNI_CHn_OUT.SIGN.DTR	
RTS: bool	rx_py_UNI_CHn_OUT.SIGN.RTS	
NUMT: uint	rx_py_UNI_CHn_OUT.NUMT	
DATA[...]: usint	rx_py_UNI_CHn_OUT.DATA	

Vstupní data diagnostiky kanálu (rx_py_Statistic_CHn)

STAT - status kanálu

	X	X	X	X	X	X	X	X
bit	.7	.6	.5	.4	.3	.2	.1	.0

nepoužito

ERR - chyba komunikace (viz kap.2.1.6.)

trueMess - počet platných komunikací

falseMess - počet chybných komunikací

Vstupní data sériového kanálu (rx_py_UNI_CHn_IN)

STAT - status vysílání a příjmu

	ARC	TRF	ROV	RCF	TRO	X	CTS	X
bit	.7	.6	.5	.4	.3	.2	.1	.0

ARC - alternace příjmu
- při nově přijaté zprávě dojde ke změně hodnoty bitu

TRF - 0 - neprobíhá žádné vysílání
- 1 - probíhá vysílání, další zpráva bude převzata až po odvysílání

ROV - 0 - příjem zprávy v pořádku
- 1 - přetečení, přijatá zpráva je delší, než vyhrazená přijímací zóna

RCF - 0 - příjem zpráv bez omezení
1 - přijímací zásobníky jsou plné, poslední přijatá zpráva bude ztracena při dalším příjmu

TRO - 0 - vysílání zpráv bez omezení
1 - vysílací zásobníky jsou plné, zápis další zprávy bude neplatný

CTS - stav signálu CTS (připravenost k vysílání)

ERR - chyba příjmu (viz kap.2.1.6.)

NUMR - počet přijatých bytů

DATA[x] - přijatá zpráva

Výstupní data řízení kanálu (rx_py_Control_CHn)

CONTROL - řízení kanálu

	0	0	0	0	0	0	0	0
bit	.7	.6	.5	.4	.3	.2	.1	.0

nepoužito

LOG_TEXT_ONLY
- 0 - ukládat do souboru .LOG data v číselné i textové formě
- 1 - ukládat do souboru .LOG data pouze v textové formě

LOG_HEX_ONLY
- 0 - ukládat do souboru .LOG data v číselné i textové formě
- 1 - ukládat do souboru .LOG data pouze v číselné formě (formát hex.)

LOG_CYCLIC - 0 - ukládat do souboru .LOG data pouze do jeho zaplnění
- 1 - ukládat do souboru .LOG data cyklicky, soubor tak obsahuje poslední data

LOG_ON - 0 - ukládání do souboru .LOG vypnuto
- 1 - ukládání do souboru .LOG zapnuto

Tyto proměnné slouží k ovládání monitorování provozu sériového kanálu (viz kap.2.1.7.).

Výstupní data sériového kanálu (rx_py_UNI_CHn_OUT)

CONT - řízení vysílání a příjmu

	ACN	CLR	TRG	X	X	X	X	X
bit	.7	.6	.5	.4	.3	.2	.1	.0

ACN - alternace řízení
- při změně bitu dojde k akceptování hodnot ostatních bitů CONT

CLR - 1 - vymazání vysílacích a přijímacích zásobníků

TRG - 1 - spuštění vysílání zprávy

SIGN - řízení modemových signálů

	X	X	X	X	X	X	RTS	X
bit	.7	.6	.5	.4	.3	.2	.1	.0

RTS - řízení signálu RTS

NUMT - počet vysílaných bytů

DATA[x] - vysílaná zpráva

2.1.3. Vysílání a příjem dat

Předávání zpráv určených k odvysílání z uživatelského programu do vnitřního vysílacího zásobníku komunikačního kanálu a přebírání přijatých zpráv z vnitřního přijímacího zásobníku komunikačního kanálu do uživatelského programu je prováděno výhradně v otočce cyklu se zajištěnou časovou konzistencí dat (tj. shodné „stáří“ všech bytů zprávy).

Sériový kanál má přijímací zásobník a vysílací zásobník, každý o velikosti 512 bytů dat.

K obsluze vysílání a příjmu dat jsou určeny bloky *fbSendTo* a *fbRecvFrom* z knihovny *ComLib*. Doporučujeme její použití z důvodu vyvarování se problémů s obsluhou příznaků kanálu v reálném čase. Následující informace jsou určeny především pro ty, kteří knihovnu nepoužijí.

Postup při vysílání zprávy

Do vysílací zóny komunikačního kanálu zapíšeme zprávu, nastavíme bit *TRG* v proměnné *CONT* na hodnotu 1 a změníme hodnotu bitu alternace *ACN*. Vysílání další zprávy aktivujeme opět změnou hodnoty bitu alternace *ACN* (bit *TRG* má stále hodnotu 1). Pokud budeme chtít pouze smazat zásobníky a nevysílat, musíme před alternací bitu *ACN* bit *TRG* vynulovat.

Po dobu vysílání zprávy sériovým kanálem je bit *TRF* v proměnné *STAT* nastaven na hodnotu 1.

Vysílání více zpráv za sebou na sériovém kanálu

Před realizací obsluhy sériového kanálu v uživatelském programu je třeba si uvědomit, jak dlouho bude trvat odvysílání jedné zprávy a jak dlouho trvá jeden cyklus uživatelského programu. Ke kontrole slouží bit *TRF* v proměnné *STAT*. Pokud má hodnotu log.0, je vysílací zásobník prázdný a vysílání bude zahájeno v otočce cyklu po zápisu nové zprávy (viz předchozí text).

Plně duplexní komunikace může způsobit zahlcení sériového kanálu

Pokud je komunikace plně duplexní (současný nezávislý provoz přijímače i vysílače), je třeba zajistit vhodným časováním, aby se zprávy do sériového kanálu neposílaly častěji, než je schopen je vysílat. Zaplnění vysílacího zásobníku oznamuje bit *TRF* v proměnné *STAT* hodnotou 1. Pokud během vysílání zapíšeme další zprávu určenou k odvysílání, tato bude převzata ze zápisníku do vyrovnávacího zásobníku vysílače, kde vyčká na odvysílání předchozí zprávy. Tento stav indikuje bit *TRO* v proměnné *STAT*, který je nastaven na hodnotu 1 a je vynulován v okamžiku překopírování čekající zprávy do uvolněného vysílacího zásobníku.

Po dobu, kdy bit *TRO* je nastaven na hodnotu 1, nesmíme vyslat další zprávu k odvysílání. Zaplněný vyrovnávací zásobník vysílače ji nepřijme a zpráva bude ztracena.

Zakázání příjmu během vysílání na sériovém kanálu

Pokud je komunikační protokol typu dotaz - odpověď, tento problém odpadá, protože vysílání další zprávy čeká na příjem odpovědi na zprávu předchozí. Případné hazardy při okrajových

stavech komunikace můžeme vyloučit zakázáním příjmu během vysílání zaškrtnutím volby *Odpojení přijímače během vysílání*.

Postup při příjmu zprávy

Přijme-li komunikační kanál zprávu, předá ji v otočce cyklu do přijímací zóny v zápisníku PLC. Bit *ARC* v proměnné *STAT* změní hodnotu (alternuje).

Pokud je přijatá zpráva delší, než je velikost přijímací zóny v zápisníku, je bit *ROV* v proměnné *STAT* nastaven na hodnotu 1.

Příjem více zpráv za sebou

Před realizací obsluhy sériového kanálu v uživatelském programu je třeba si uvědomit, jak dlouho bude trvat příjem jedné zprávy a jak dlouho trvá jeden cyklus uživatelského programu. Přijímací zásobník má vyrovnávací funkci, takže lze přijímat další zprávu dříve, než je předána do zápisníku zpráva předchozí. Je-li zásobník zaplněn, má bit *RCF* v proměnné *STAT* hodnotu 1. Sériový kanál pak bude další zprávu přijímat přes zprávu předchozí, která bude nenávratně ztracena. Po předání jedné zprávy do systému je zásobník uvolněn pro další příjem a bit *RCF* se změní na hodnotu 0.

Vymazání vysílacích a přijímacích zásobníků

Nastavením bitu *CLR* v proměnné *CONT* na log.1 provedeme vymazání vysílacích a přijímacích zásobníků. Všechny zprávy, které zde byly uloženy, budou ztraceny. Nastavení kanálu se nemění. Bit je třeba v následujícím cyklu opět vynulovat.

Tento příkaz lze kombinovat s příkazem vysílání, vymazání zásobníků se provede vždy jako první.

Alternace bitu *ACN* v proměnné *CONT* je podmínkou pro akceptování tohoto příkazu.

2.1.4. Ovládání modemových signálů

Sériový kanál umožňuje řídit signál RTS a monitorovat signál CTS v různých režimech. Tato nastavení mají samozřejmě význam pouze v případě vyvedení obou signálů na rozhraní RS-232 (submodul MR-0130).

Trvalé nastavení signálu RTS na pevnou hodnotu

Pokud chceme, aby signál RTS měl stále stejnou předem definovanou úroveň, nastavíme v konfiguračním panelu položku *Řízení signálu RTS* do režimu *trvale hodnota 0* nebo *trvale hodnota 1*.

Ovládání signálu RTS z uživatelského programu PLC

Pokud chceme, aby signál RTS bylo možné ovládat přímo z uživatelského programu PLC, nastavíme v konfiguračním panelu položku *Řízení signálu RTS* do režimu *podle SIGN.1*.

Od okamžiku startu uživatelského programu je úroveň signálu určena hodnotou bitu *RTS* v proměnné *SIGN* ve výstupních datech sériového kanálu.

Automatická změna signálu RTS při vysílání

Pokud chceme, aby signál RTS se automaticky změnil po dobu vysílání zprávy, nastavíme v konfiguračním panelu položku *Řízení signálu RTS* do režimu *automatická hodnota*.

Úroveň signálu RTS bude v klidu a během příjmu log.1. Bezprostředně před zahájením vysílání se změní na log.0 a zpět se vrátí bezprostředně po odvysílání posledního znaku zprávy.

Změna signálu RTS při vysílání, vysílání podmíněno signálem CTS

Pokud chceme, aby se signál RTS automaticky změnil po dobu vysílání zprávy a vysílání zprávy chceme podmínit signálem CTS, např. z důvodu připojení modemu, nastavíme v konfiguračním panelu položku *Řízení signálu RTS* do režimu *automatická hodnota s detekcí CTS*.

Pokud chceme připojit k sériovému kanálu modem, obvykle potřebujeme zahájit vysílání až po potvrzení signálu RTS signálem CTS, který oznámí, že modem je připraven vysílat. K tomu použijeme tuto volbu.

Úroveň RTS, bude v klidu a během příjmu log.1. Bezprostředně před zahájením vysílání se změní na log.0, ale vysílání je zahájeno až poté, kdy je tatáž úroveň zjištěna na signálu CTS. Zpět se úroveň RTS vrátí bezprostředně po odvysílání posledního znaku zprávy.

2.1.5. Klid na lince

Definice klidu na lince slouží především k bezpečnému zachycení a předání celé zprávy najednou. Snahou je odstranit problémy s roztržením přijaté zprávy, nebo se spojením po sobě následujících zpráv.

Detekce klidu na lince

Pokud je na lince klid delší dobu, než jaká je zadána, ukončí se příjem předchozí zprávy a zpráva se předá do systému.

Služba je aktivována při nenulové hodnotě položky *Min. doba klidu na lince mezi přijímanými zprávami* v inicializačním panelu.

Nenulová hodnota položky *Min. doba klidu na lince mezi přijímanými zprávami* udává minimální dobu klidu na lince přepočtenou na počet přijatých bytů. Z toho vyplývá, že výsledná doba určená touto hodnotou bude záviset na přenosové rychlosti. Ke kontrole klidu na lince slouží vnitřní časovač sériového kanálu s taktem 100 µs. Výpočet výsledné minimální doby klidu na lince v časových jednotkách se provádí podle následujících vzorců:

$$T = 100 \cdot k_T \quad [\mu s]$$

$$k_T = \left(\frac{110000 \cdot TOR}{BD} + 1 \right) \bmod 1$$

kde BD je přenosová rychlost v Bd
TOR je počet bytů odpovídajících minimální době klidu na lince
k_T je počet taktů vnitřního časovače zaokrouhlený dolů na celé číslo
T je teoretická minimální doba klidu na lince

Vzhledem k tomu, že vnitřní časovač s taktem 100 µs běží neustále, musíme mít na zřeteli chybu jednoho taktu zaviněnou skutečností, že první takt časovače bude nikoli 100 µs, ale bude se pohybovat v intervalu <1, 100> µs v závislosti na rozdílu okamžiku příjmu posledního bytu a okamžiku posledního taktu vnitřního časovače. Skutečná minimální doba klidu na lince T_I se bude pohybovat v intervalu <(T–99), T>.

Dále je třeba počítat s tím, že minimální detekovatelná hodnota doby klidu na lince je 300 µs, což se projeví u vysokých komunikačních rychlostí. Např. pro rychlost 345,6 kBd tato doba odpovídá přenosu 9 bytů.

Hodnotu doby klidu je třeba zvolit vyšší, než je maximální možná prodleva mezi jednotlivými byty zprávy. Jinak by mohlo dojít k přetržení zprávy.

Maximální nastavitelná hodnota doby klidu je 255 bytů.

Příklad:

Přenosová rychlost 19,2 kBd, minimální klid na lince 3 byty.
(Při rychlosti 19,2 kBd trvá příjem 3 bytů 1719 µs.)
inicializační tabulka - Komunikační rychlost = 19 200
- Min. doba klidu na lince mezi přijímanými zprávami = 3

$$k_T = \left(\frac{110000 \cdot 3}{19200} + 1 \right) \bmod 1 = 18{,}1875 \bmod 1 = 18$$

$T = 100 \cdot 18 = 1800$ [µs]
$T_l = \langle 1701, 1800 \rangle$ [µs]

Režim volného kanálu

Pokud je v konfiguračním panelu hodnota položky *Minimální doba klidu mezi přijímanými zprávami* 0, je zapnut na příjmu režim volného kanálu. Kanál plní funkci pouhého příjmu znaků ze sériové linky. V každém cyklu PLC jsou předány do systému všechny byty, které byly během předchozího cyklu přijaty. Tento režim je pro většinu aplikací nevýhodný především v tom, že v závislosti na rychlosti komunikace a na době cyklu trhá přijímanou zprávu na více částí, které pak uživatelský program musí spojovat dohromady. Tento režim také nedokáže od sebe oddělit dvě zprávy, které přišly za sebou během jednoho cyklu PLC.

Klid na lince mezi dvěma vysílanými zprávami

Při nenulové hodnotě položky *Minimální doba klidu mezi vysílanými zprávami* v konfiguračním panelu je zapnuto generování klidu na lince mezi dvěma vysílanými zprávami. V případě, že je do vyrovnávacího zásobníku vysílače předána ze systému zpráva k odvysílání ještě během vysílání předchozí zprávy, začne se další zpráva vysílat až po uplynutí minimální doby klidu na lince určené touto hodnotou zadanou v počtu vysílaných bytů. Pro výpočet doby klidu na lince v časových jednotkách platí vztahy uvedené v předchozích odstavcích.

Maximální nastavitelná hodnota doby klidu je 255.

Pokud je hodnota položky *Minimální doba klidu mezi vysílanými zprávami* 0, mezi vysílanými zprávami není generován žádný klid na lince. Lze tak vysílat kontinuálně.

2.1.6. Chybová hlášení

Provozní chybová hlášení sériového kanálu se zobrazují v přijímací zóně v proměnné *ERR* a v diagnostické zóně v proměnné *ERR*.

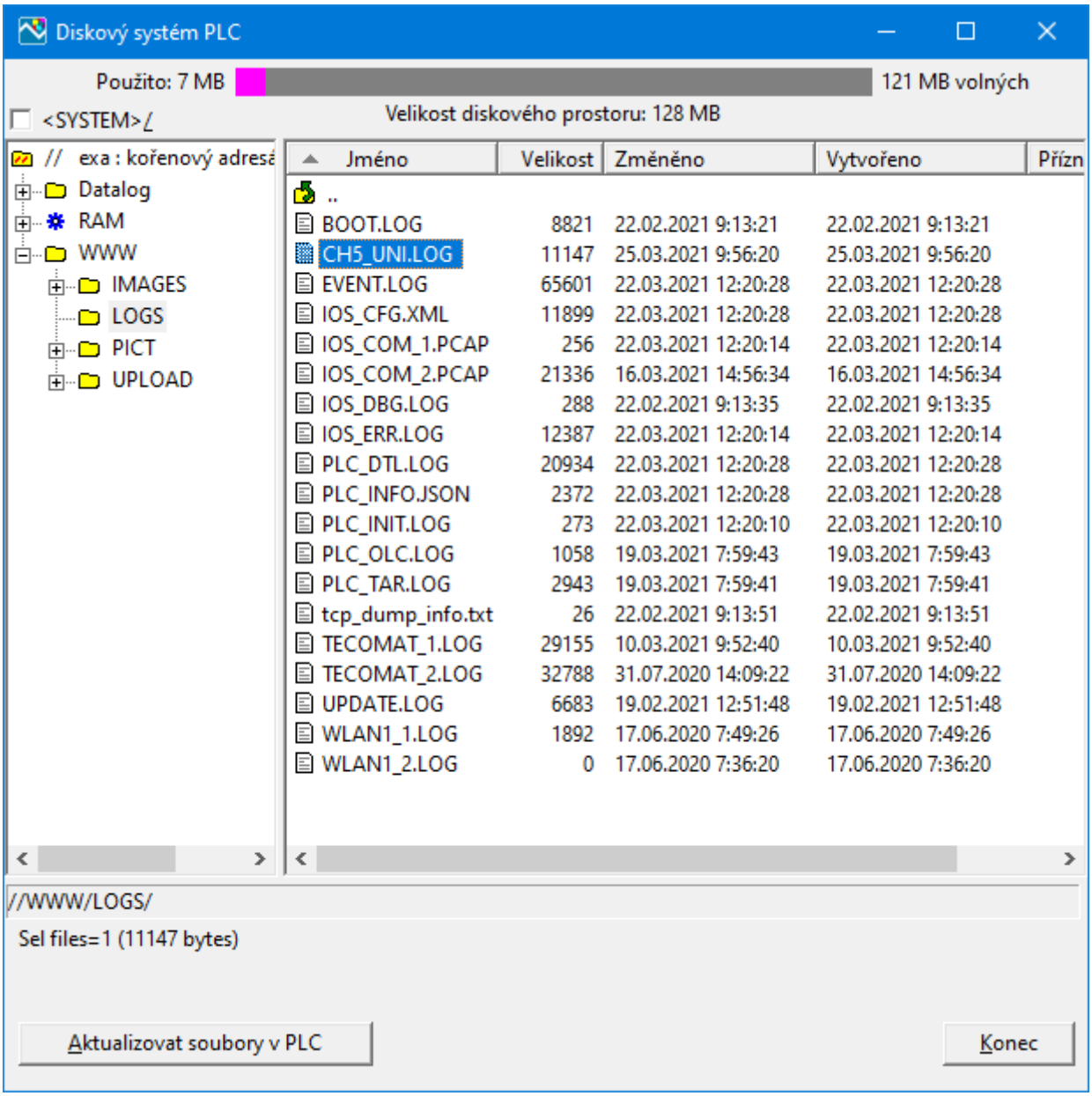
$11 chyba parity
Aspoň jeden byte zprávy měl chybnou paritu nebo stopbit.

$31 chybná délka vysílaných dat
Hodnota délky vysílaných dat překračuje velikost vysílací zóny.

$32 nulová délka vysílaných dat
Hodnota délky vysílaných dat musí být nenulová.

2.1.7. Monitorování provozu sériového kanálu

Sériový kanál umožňuje ukládat provoz na sériovém kanálu do souboru .LOG se jménem odvozeným od čísla a režimu kanálu, např. CH5_UNI.LOG. Tato funkce je užitečná pro případy ladění komunikace, nebo řešení provozních problémů s komunikací. Vzniklý soubor je uložen v souborovém systému PLC v adresáři //WWW/LOGS/. Souborový systém PLC je dostupný v prostředí Mosaic pomocí ikony v hlavní nástrojové liště.



Obr.2.7 Umístění souboru CH5_UNI.LOG v souborovém systému PLC

Soubor .LOG obsahuje informace o nastavení kanálu a výpis přijímaných a vysílaných dat včetně časových značek. Data přenášená kanálem jsou zobrazena v číselné formě v hexadecimálním formátu a v textové formě ve formátu ASCII.

```
SIO UNI CHANNEL (/CH5_UNI.LOG), F2x CP2005I v2.1.043d (Mar 22 2021 10:34:05),
start 2021-03-25 09:56:11.653343
================================================================================
2021-03-25 09:56:11.653579  ---- LOG start ---- F2x CP2005I v2.1.043d
(Mar 22 2021 10:34:04)
2021-03-25 09:56:11.653701  "CH5": "mode UNI, 8-N-1, 38400 Baud, HALF DUPLEX,
RTS AUTO, rxTimeout 4, txTimeout 4"
2021-03-25 09:56:11.653775  SEND 8
 01 03 00 00 00 04 44 09                          ......D.
2021-03-25 09:56:11.753669  RECV 13
 01 03 08 00 00 00 00 00 00 00 95 D7              ............
2021-03-25 09:56:11.833008  SEND 8
 01 03 00 04 00 06 84 09                          ........
2021-03-25 09:56:11.939460  RECV 17
 01 03 0C 00 00 00 00 00 00 00 00 00 00 00 93     ................
 70                                               p
2021-03-25 09:56:12.005187  SEND 8
 01 01 00 09 00 05 2C 0B                          ......,.
2021-03-25 09:56:12.109787  RECV 6
 01 01 01 00 51 88                                ....Q.
2021-03-25 09:56:12.176315  SEND 8
 01 03 00 00 00 04 44 09                          ......D.
2021-03-25 09:56:12.281673  RECV 13
 01 03 08 00 00 00 00 00 00 00 95 D7              ............
2021-03-25 09:56:12.360841  SEND 8
 01 03 00 04 00 06 84 09                          ........
2021-03-25 09:56:12.461697  RECV 17
 01 03 0C 00 00 00 00 00 00 00 00 00 00 00 93     ................
 70                                               p
```

Generování souboru .LOG lze ovládat pomocí proměnných ve struktuře *rx_py_Control_CHn* (např. *r16_p4_Control_CH1*).

Proměnná *LOG_ON* ovládá generování souboru. Zápisem hodnoty 1 do proměnné vytvoříme soubor a je zahájeno ukládání dat. Zápisem hodnoty 0 se ukládání ukončí.

Pokud má proměnná *LOG_CYCLIC* hodnotu 0, pak se soubor po dosažení velikosti 64 KB sám uzavře a ukládání dat se ukončí. Zápisem hodnoty 1 aktivujeme cyklický režim zápisu. Data se po dosažení velikosti souboru 64 KB začnou zapisovat opět na začátek souboru a přepisovat tak původně uložená data. V souboru .LOG tak máme k dispozici vždy nejnovějších 64 KB dat. Při prohlížení souboru je třeba mít na paměti, že data se přepisují cyklicky, to znamená, že předěl mezi nejnovějšími a nejstaršími daty „putuje“ uvnitř souboru. Po uzavření souboru nulováním proměnné *LOG_ON* nedojde k přerovnání souboru od nejstarších k nejnovějším datům. Předěl je v souboru viditelně označen řádkem složeným ze znaků ~ (tilda), které se nikde jinde nevyskytují, takže jej lze snadno vyhledat.

```
2021-03-25 09:56:20.117882  SEND 8
 01 03 00 04 00 06 84 09                          ........
 ~~~~~~~~~~~~~~~~~~~~~~~~
2021-03-25 09:56:11.653775  SEND 8
 01 03 00 00 00 04 44 09                          ......D.
```

Množství zapisovaných dat můžeme snížit pomocí proměnných *LOG_HEX_ONLY* a *LOG_TEXT_ONLY*. Pokud proměnnou *LOG_HEX_ONLY* nastavíme na hodnotu 1, v souboru bude vynechána textová forma dat. Pokud proměnnou *LOG_TEXT_ONLY* nastavíme na hodnotu 1, v souboru bude vynechána číselná forma dat.

2.2. REŽIM PC - KOMUNIKACE S NADŘÍZENÝM SYSTÉMEM

Režim **PC** slouží k připojení nadřízeného systému k PLC. Komunikace probíhá protokolem EPSNET, který obsahuje i systémové služby umožňující programování a ladění systému z prostředí Mosaic. V jednom časovém okamžiku může využívat systémové služby jen jeden nadřízený systém na jednom spojení. Naopak služby pro čtení a zápis dat z paměti proměnných mohou být použity současně na více kanálech, resp. spojeních najednou.

Komunikace jsou vyvolávány nadřízeným systémem na principu dotaz - odpověď. PLC TECOMAT v režimu **PC** se chová jako pasivní podřízený účastník (server).

Veřejné a systémové služby sítě EPSNET

V režimu **PC** je na všech komunikačních kanálech dostupný kompletní soubor služeb sítě EPSNET, jehož součástí jsou jak systémové služby, určené k programování a ladění systému, tak i veřejné služby, určené k výměně dat. Omezující podmínkou je, že systémové služby podporuje v daném časovém okamžiku pouze jeden komunikační kanál.

Síť EPSNET a její veřejné služby jsou popsány v kap.5.

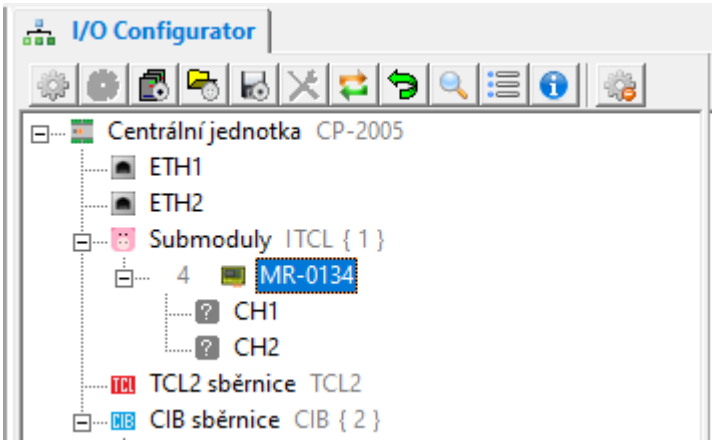
Upozornění: Vývojové prostředí Mosaic využívá systémové služby sítě EPSNET a z tohoto důvodu je lze provozovat pouze přes jeden komunikační kanál v režimu **PC**. Oprávnění využívat systémové služby dostane ten komunikační kanál, který přijde s požadavkem jako první. Oprávnění zaniká 5 sekund po ukončení komunikace.

2.2.1. Konfigurace

Konfigurace komunikačních kanálů stejně jako konfigurace všech částí PLC TECOMAT FOXTROT 2 se v prostředí Mosaic provádí pomocí nástroje *I/O Configurator*.

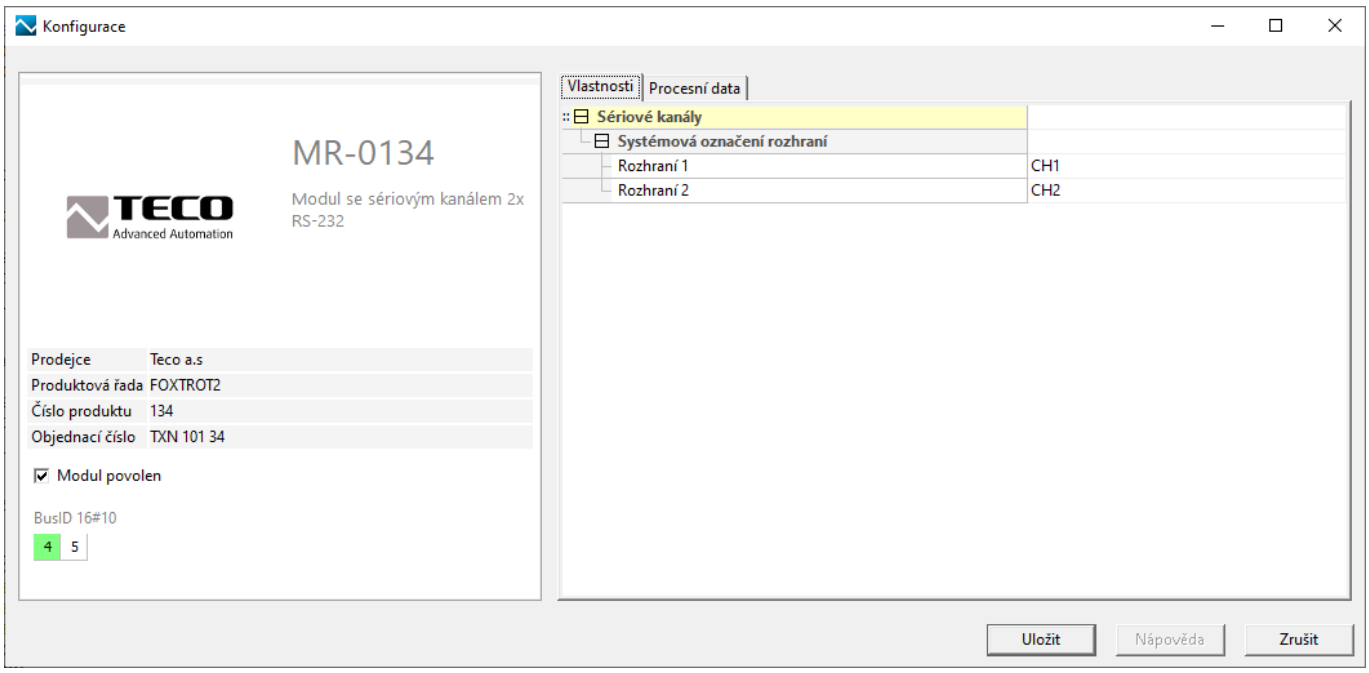
Volba čísla kanálu

Při vkládání submodulu do konfigurace, případně při načtení konfigurace připojeného PLC nástroj *I/O Configurator* přidělí jednotlivým komunikačním kanálům označení. Označení sériových kanálů, které jsou realizovány pomocí submodulů, může uživatel změnit.



Obr.2.8 Příklad konfigurace se sériovými kanály bez nastavených režimů

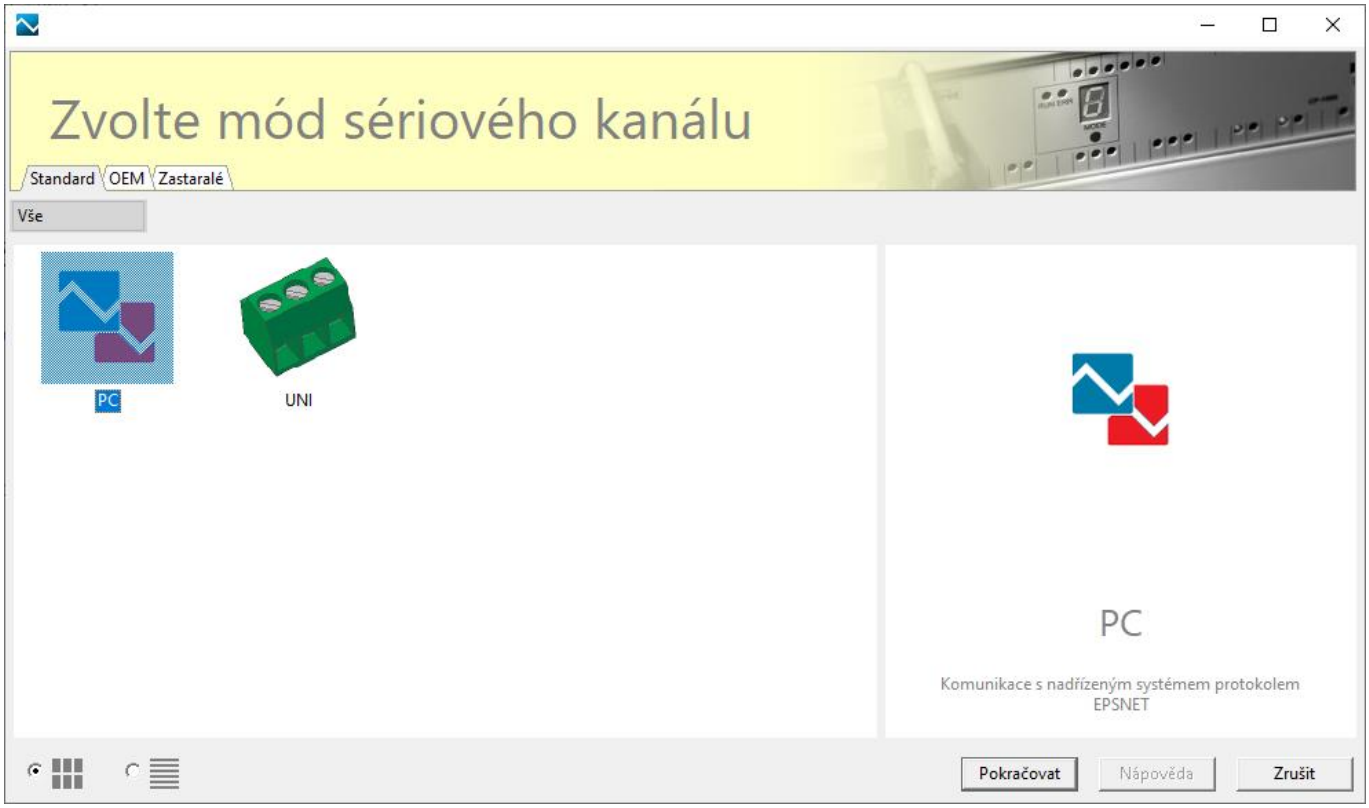
Poklepáním levým tlačítkem myši na názvu submodulu se otevře konfigurační okno, ve kterém můžeme nastavit označení sériového kanálu v rozsahu CH1 až CH10 (obr.2.9). Každý kanál musí mít jiné označení.



Obr.2.9 Konfigurace submodulu MR-0134 se dvěma sériovými kanály

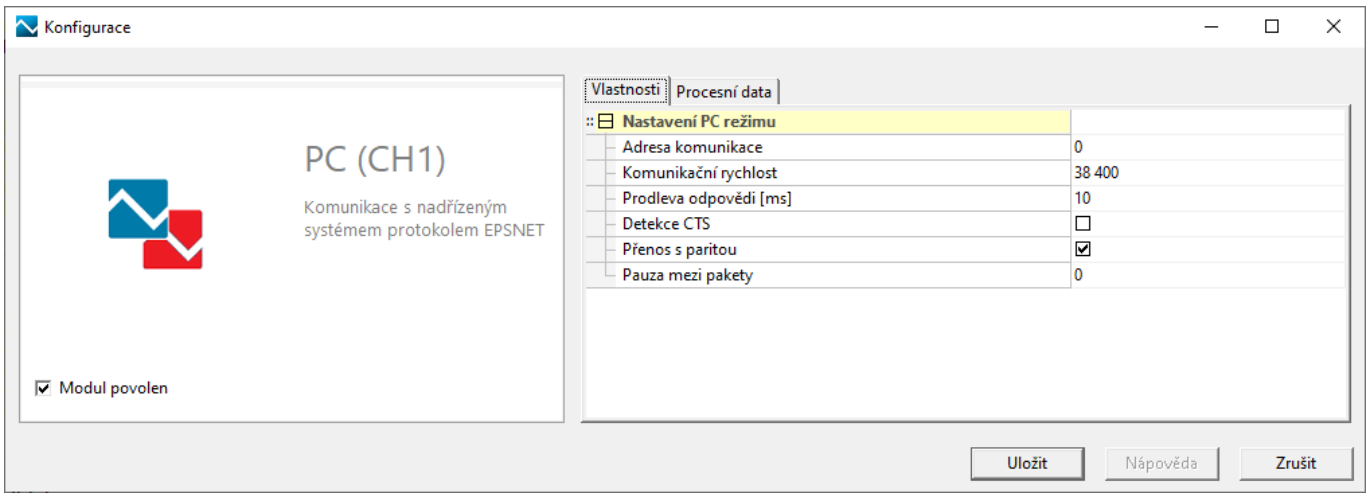
Nastavení režimu kanálu

Režim kanálu nastavíme tak, že poklepeme levým tlačítkem myši na označení příslušného kanálu (např. CH1). Otevře se okno výběru režimu sériového kanálu (obr.2.10).



Obr.2.10 Výběr režimu sériového kanálu

Poklepáním vybereme režim **PC** a otevře se panel konfigurace (obr.2.11).



Obr.2.11 Konfigurace režimu PC pro sériový kanál

V levé části panelu *Konfigurace* se nachází položka *Modul povolen*. Pokud je zaškrtnuta, provádí se obsluha kanálu podle nastavení dostupných v pravé části panelu. Pokud položka nebude zaškrtnutá, je obsluha kanálu vypnuta beze ztráty aktuálního nastavení. To lze využít při ladění uživatelského programu.

V pravé části panelu se v záložce *Vlastnosti* nachází všechny položky konfigurace daného kanálu. V záložce *Procesní data* pak nalezneme seznam všech proměnných, které kanál poskytuje, s možností jejich vlastního pojmenování (viz kap.2.2.2).

Význam a možnosti nastavení jednotlivých položek v pravé části panelu jsou následující:

Adresa komunikace

Volba adresy účastníka sítě EPSNET v rozsahu 0 až 126 (viz kap.5.). Volba adresy umožňuje připojení více systémů TECOMAT k jednomu nadřízenému systému (v tomto případě je nutností rozhraní RS-485). Zde je třeba zajistit, aby připojené PLC i nadřízený systém měly každý jinou adresu. Adresy nemusí tvořit spojitou řadu.

Komunikační rychlost

Volba přenosové rychlosti v Bd (bity za sekundu).

Prodleva odpovědi

Tento parametr je v systémech TECOMAT FOXTROT 2 nastaven implicitně.

Detekce CTS

Tento parametr v systémech TECOMAT FOXTROT 2 nemá význam (signál CTS není vyveden).

Přenos s paritou

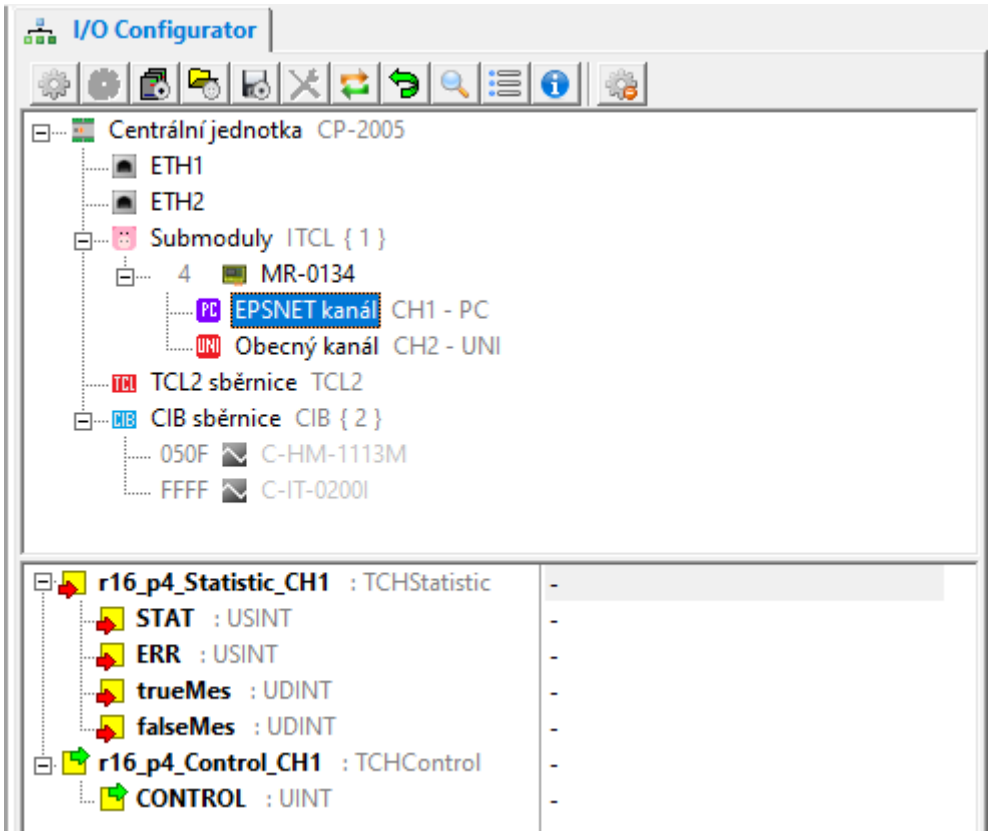
Protokol sítě EPSNET používá sudou paritu. Některé modemy však neumožňují paritu přenášet. V případě jejich použití lze paritu vypnout zrušením zaškrtnutí této položky (v tom případě musí přenos bez parity podporovat i nadřízený systém).

Pozor! Parita se výrazně podílí na bezpečnosti přenosu dat. Jejím vypnutím se zvyšuje riziko nezjištěného přenosu chybných dat.

Pauza mezi pakety

Tento parametr je v systémech TECOMAT FOXTROT 2 nastaven implicitně.

Po stisknutí tlačítka *Uložit* se ve stromečku konfigurace na řádku příslušného kanálu objeví označení aktuálně nastaveného režimu (obr.2.12).



Obr.2.12 Příklad konfigurace se sériovým kanálem s nastaveným režimem PC

2.2.2. Poskytovaná data

Obsluha komunikačního kanálu v režimu **PC** probíhá prostřednictvím systémového ovladače bez nutnosti zásahu uživatelského programu. V zápisníkové paměti je k dispozici pouze diagnostická zóna. Struktura dat je uvedena v tab.2.3.

Položky struktury mají přidělena symbolická jména, která začínají vždy stejnou kombinací znaků určujících adresu a označení kanálu v rámci systému. Ve sloupci *Úplný zápis* je uvedeno vždy konkrétní symbolické jméno pro danou položku. Pokud chceme data použít v uživatelském programu, použijeme buď toto symbolické jméno, nebo ve sloupci *Alias* zapíšeme svoje symbolické jméno, které pak můžeme používat. V žádném případě nepoužíváme absolutní operandy, protože se mohou po novém překladu uživatelského programu změnit.

Alias vytvoříme v nástroji *I/O Configurator* následovně. Poklepáním na jméno modulu ve stromečku konfigurace otevřeme panel *Konfigurace*. V záložce *Procesní data* zapíšeme požadované jméno v příslušném řádku do sloupce *Alias*.

Uspořádání diagnostických dat

Komunikační kanál v režimu **PC** zveřejňuje diagnostická data stavu linky a stav komunikace. Tyto údaje se ukládají do zápisníku a jsou snadno přístupné v nástroji *I/O Configurator*.

Tab.2.3 Struktura dat (n je číslo sériového kanálu)

Struktura dat	Úplný zápis	Popis objektu
Statistic_CHn	rx_py_Statistic_CHn	diagnostická zóna kanálu
STAT: usint	rx_py_Statistic_CHn.STAT	
ERR: usint	rx_py_Statistic_CHn.ERR	
trueMess: udint	rx_py_Statistic_CHn.trueMess	
falseMess: udint	rx_py_Statistic_CHn.falseMess	
Control_CHn	rx_py_Control_CHn	řídicí zóna kanálu
CONTROL: uint	rx_py_Control_CHn.CONTROL	

Vstupní data diagnostiky kanálu (rx_py_Statistic_CHn)

STAT - status kanálu

	X	X	X	X	X	X	X	X
bit	.7	.6	.5	.4	.3	.2	.1	.0

nepoužito

ERR - chyba komunikace (viz kap.2.2.3.)

trueMess - počet platných komunikací

falseMess - počet chybných komunikací

Výstupní data řízení kanálu (rx_py_Control_CHn)

CONTROL - řízení kanálu

	0	0	0	0	0	0	0	0
bit	.7	.6	.5	.4	.3	.2	.1	.0
	0	0	0	0	0	0	0	0
bit	.15	.14	.13	.12	.11	.10	.9	.8

nepoužito

2.2.3. Chybová hlášení

Provozní chybová hlášení sériového kanálu se zobrazují v přijímací zóně v proměnné *ERR* a v diagnostické zóně v proměnné *ERR*.

$11 chyba parity

Aspoň jeden byte zprávy měl chybnou paritu nebo stopbit.

$31 chybná délka vysílaných dat

Hodnota délky vysílaných dat překračuje velikost vysílací zóny.

$32 nulová délka vysílaných dat

Hodnota délky vysílaných dat musí být nenulová.

2.3. REŽIM CSJ - PŘIPOJENÍ SBĚRNICE CAN

Funkce režimu CSJ

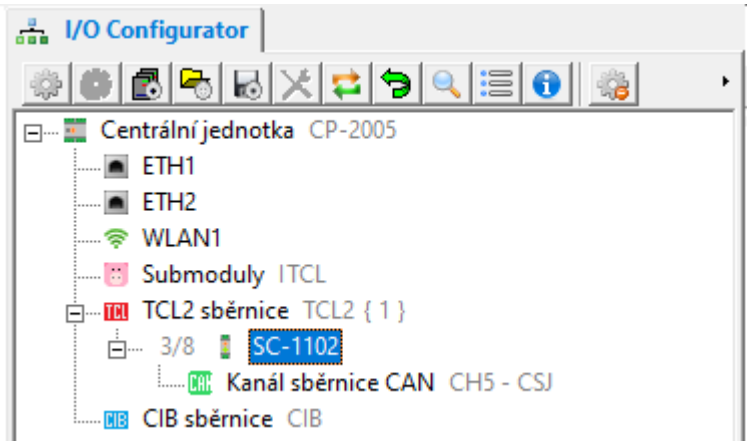
Režim **CSJ** umožňuje obecnou obsluhu sběrnice CAN připojené pomocí komunikačního modulu SC-1102. Přenosová rychlost je volitelná 1000, 500, 250, 125, 50 nebo 20 kBd.

2.3.1. Konfigurace

Konfigurace komunikačních kanálů stejně jako konfigurace všech částí PLC TECOMAT FOXTROT 2 se v prostředí Mosaic provádí pomocí nástroje *I/O Configurator*.

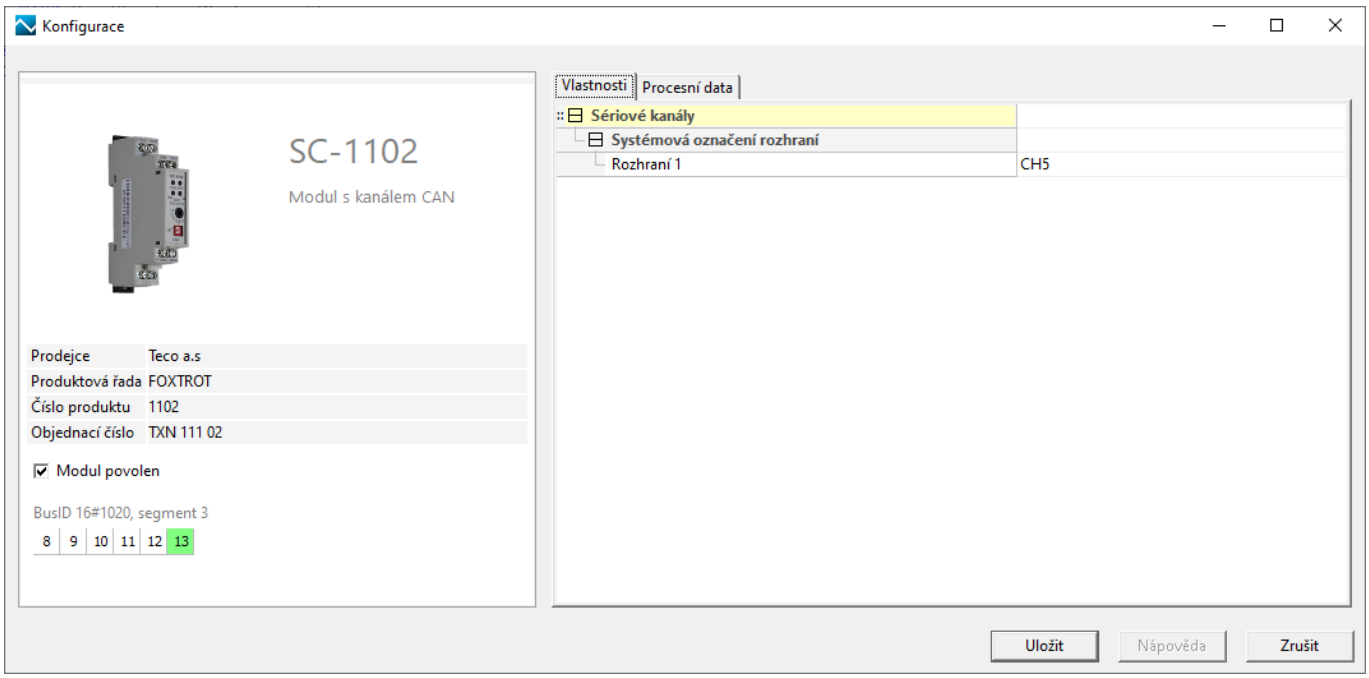
Volba čísla kanálu

Při vkládání submodulu nebo komunikačního modulu do konfigurace, případně při načtení konfigurace připojeného PLC nástroj *I/O Configurator* přidělí jednotlivým komunikačním kanálům označení. Označení sériových kanálů, které jsou realizovány pomocí submodulů a samostatných komunikačních modulů, může uživatel změnit.



Obr.2.13 Příklad konfigurace s modulem SC-1102 bez nastaveného režimu

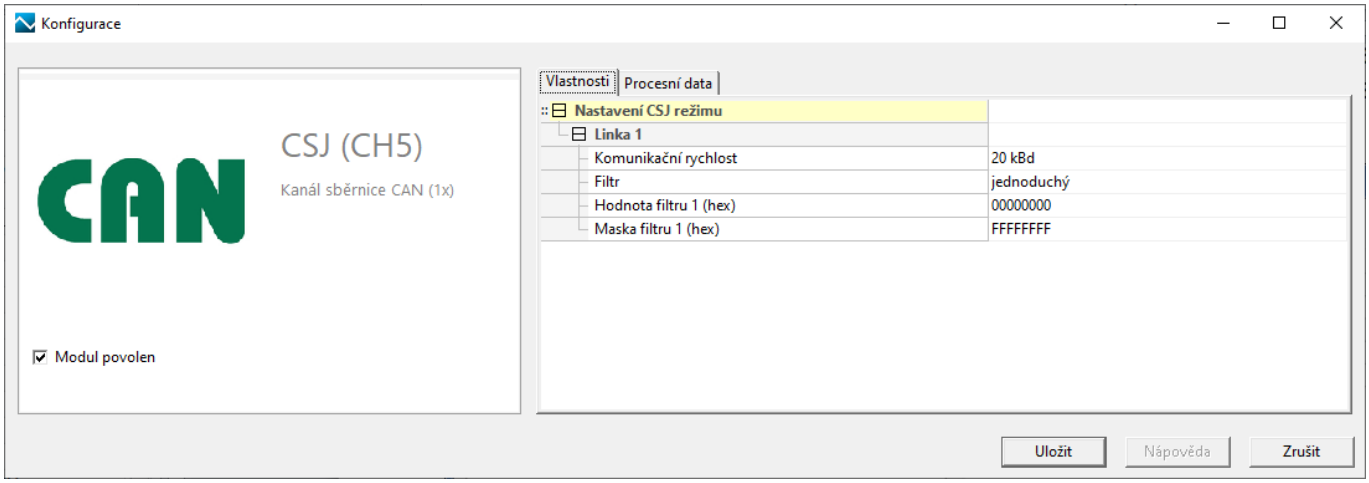
Poklepáním levým tlačítkem myši na názvu komunikačního modulu se otevře konfigurační okno, ve kterém můžeme nastavit označení sériového kanálu v rozsahu CH1 až CH10 (obr.2.14). Každý kanál musí mít jiné označení.



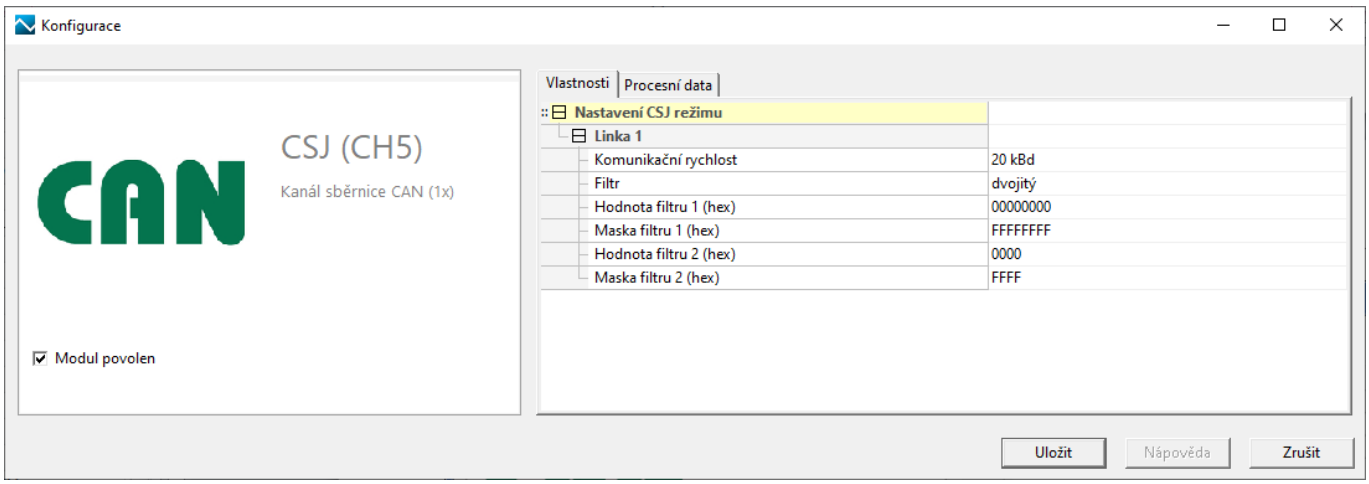
Obr.2.14 Konfigurace modulu SC-1102

Nastavení režimu kanálu

Vzhledem k tomu, že modul SC-1102 lze provozovat pouze v režimu CSJ, je tento režim již přednastaven a není třeba jej potvrzovat. Konfiguraci režimu provedeme tak, že poklepeme levým tlačítkem myši na řádek označení režimu kanálu (např. *Kanál sběrnice CAN CH5 - CSJ*). Otevře se panel konfigurace (obr.2.15, obr.2.16).



Obr.2.15 Konfigurace režimu CSJ s jednoduchým filtrem



Obr.2.16 Konfigurace režimu CSJ s dvojitým filtrem

V levé části panelu *Konfigurace* se nachází položka *Modul povolen*. Pokud je zaškrtnuta, provádí se obsluha kanálu podle nastavení dostupných v pravé části panelu. Pokud položka nebude zaškrtnutá, je obsluha kanálu vypnuta beze ztráty aktuálního nastavení. To lze využít při ladění uživatelského programu.

V pravé části panelu se v záložce *Vlastnosti* nachází všechny položky konfigurace daného kanálu. V záložce *Procesní data* pak nalezneme seznam všech proměnných, které kanál poskytuje, s možností jejich vlastního pojmenování (viz kap.2.3.2).

Význam a možnosti nastavení jednotlivých položek v pravé části panelu jsou následující:

Komunikační rychlost

Volba přenosové rychlosti sítě CAN v Bd (bity za sekundu). Všechny stanice v síti CAN musí mít tuto rychlost shodnou.

Filtr

Volba režimu přijímacího filtru:
jednoduchý filtr - jeden filtr testující čtyři byty identifikátoru přijaté zprávy
dvojitý filtr - dva filtry testující čtyři, resp. dva byty identifikátoru přijaté zprávy

Hodnota filtru 1

Hodnota, kterou první filtr porovnává se začátkem přijaté zprávy. Hodnota se zadává v hexadecimální soustavě.

Maska filtru 1

Maska určující, které bity hodnoty prvního filtru jsou porovnávány a které jsou ignorovány. Hodnota se zadává v hexadecimální soustavě.

Hodnota filtru 2

Hodnota, kterou druhý filtr porovnává se začátkem přijaté zprávy. Hodnota se zadává v hexadecimální soustavě.

Maska filtru 2

Maska určující, které bity hodnoty druhého filtru jsou porovnávány a které jsou ignorovány. Hodnota se zadává v hexadecimální soustavě.

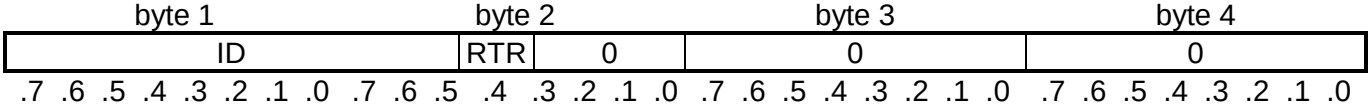
V následujících odstavcích si popíšeme použití přijímacích filtrů.

Jednoduchý přijímací filtr

V tomto režimu lze definovat jeden dlouhý filtr (4 byty). Význam odpovídajících bitů filtru (hodnoty a masky filtru) a začátku přijaté zprávy závisí na právě přijímaném formátu zprávy (šířce identifikátoru) a na typu řadiče CAN.

Jednoduchý filtr - standardní rámec zprávy

Pokud je přijata zpráva se standardním rámcem, tj. identifikátor má šířku 11 bitů, celý identifikátor a příznak žádosti o data jsou testovány vstupním filtrem.



Obr.2.17 Struktura začátku přijaté zprávy se standardním rámcem pro jednoduchý filtr
ID - identifikátor zprávy
RTR - příznak žádosti o data

Pro platný příjem musí být všechny bity začátku přijaté zprávy shodné s hodnotou filtru. Pokud některé bity nechceme testovat, nastavíme masku filtru na pozici těchto bitů na hodnotu 1. Nejnižší 4 bity druhého bytu, celý třetí a čtvrtý byte (na obr.2.17 označené 0) nejsou testovány. Z důvodu kompatibility s budoucími verzemi doporučuje výrobce řadiče nastavit příslušné bity masky filtru vždy na hodnotu 1 jako nepoužité.

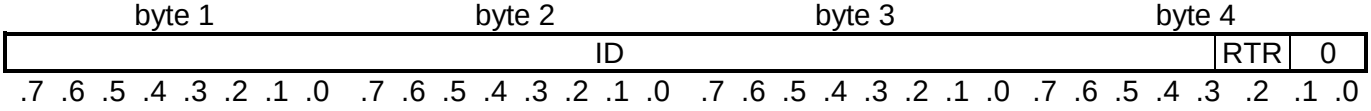
Pokud chceme přijímat všechny zprávy ze sběrnice CAN, nastavíme masku filtru na hodnotu FFFFFFFF (hex).

Pokud chceme přijímat jen zprávy s jedním konkrétním identifikátorem, např. 0340 (hex), pak nastavíme masku filtru na 001FFFFF (hex) a hodnotu filtru nastavíme na 68000000 (hex).

Pokud chceme přijímat jen zprávy s několika identifikátory např. 0340, 0341, 0342 a 0343 (hex), pak nastavíme masku filtru na 007FFFFF (hex) a hodnotu filtru nastavíme na 68000000 (hex).

Jednoduchý filtr - rozšířený rámec zprávy

Pokud je přijata zpráva s rozšířeným rámcem, tj. identifikátor má šířku 29 bitů, celý identifikátor a příznak žádosti o data jsou testovány vstupním filtrem.



Obr.2.18 Struktura začátku přijaté zprávy s rozšířeným rámcem pro jednoduchý filtr
ID - identifikátor zprávy
RTR - příznak žádosti o data

Pro platný příjem musí být všechny bity začátku přijaté zprávy shodné s hodnotou filtru. Pokud některé bity nechceme testovat, nastavíme masku filtru na pozici těchto bitů na hodnotu 1. Nejnižší 2 bity čtvrtého bytu (na obr.2.18 označené 0) nejsou testovány. Z důvodu kompatibility s budoucími verzemi doporučuje výrobce řadiče nastavit příslušné bity masky filtru vždy na hodnotu 1 jako nepoužité.

Pokud chceme přijímat všechny zprávy ze sběrnice CAN, nastavíme masku filtru na hodnotu FFFFFFFF (hex).

Pokud chceme přijímat jen zprávy s jedním konkrétním identifikátorem, např. 10000210 (hex), pak nastavíme masku filtru na 00000007 (hex) a hodnotu filtru nastavíme na 80001080 (hex).

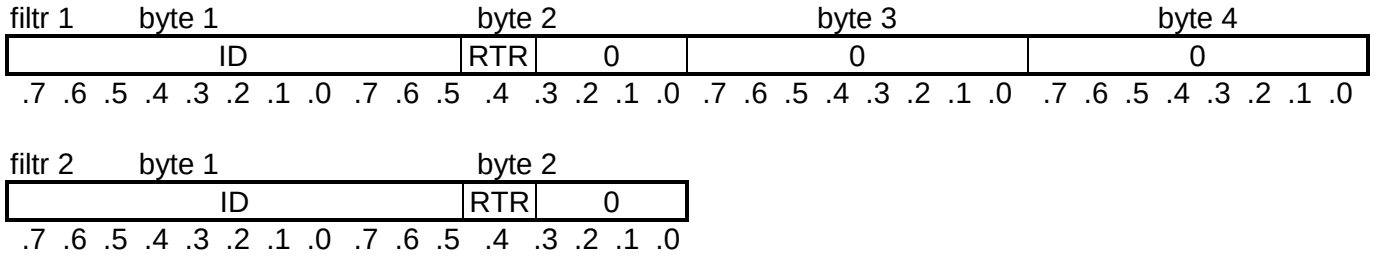
Pokud chceme přijímat jen zprávy s několika identifikátory např. 10000210, 10000211, 10000212 a 10000213 (hex), pak nastavíme masku filtru na 0000001F (hex) a hodnotu filtru nastavíme na 80001080 (hex).

Dvojitý přijímací filtr

V tomto režimu lze definovat dva zkrácené filtry. Jejich velikost a význam odpovídajících bitů filtru (hodnoty a masky filtru) a začátku přijaté zprávy závisí na právě přijímaném formátu zprávy (šířce identifikátoru) a na typu řadiče CAN.

Dvojitý filtr - standardní rámec zprávy

Pokud je přijata zpráva se standardním rámcem, tj. identifikátor má šířku 11 bitů, celý identifikátor a příznak žádosti o data jsou testovány vstupními filtry.



Obr.2.19 Struktura začátku přijaté zprávy se standardním rámcem pro dvojitý filtr
ID - identifikátor zprávy
RTR - příznak žádosti o data

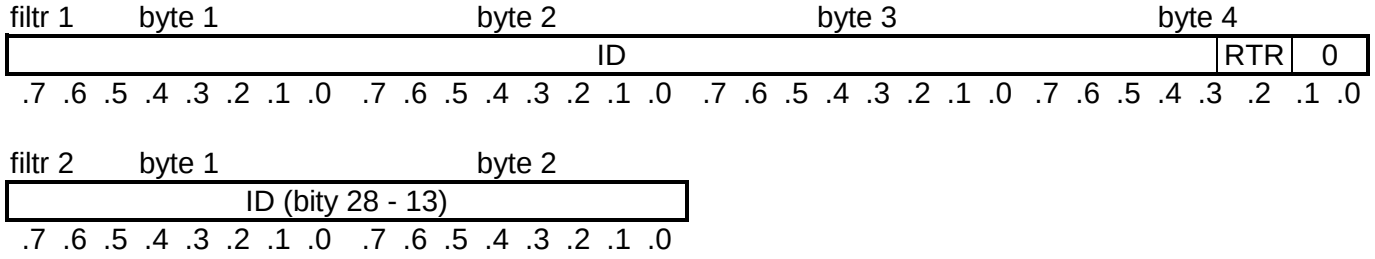
Pro platný příjem musí být všechny bity začátku přijaté zprávy shodné s hodnotou aspoň jednoho filtru. Pokud některé bity nechceme testovat, nastavíme masku filtru na pozici těchto bitů na hodnotu 1. Nejnižší 4 bity druhého bytu, celý třetí a čtvrtý byte (na obr.2.19 označené 0) nejsou testovány. Z důvodu kompatibility s budoucími verzemi doporučuje výrobce řadiče nastavit příslušné bity masky filtru vždy na hodnotu 1 jako nepoužité.

Pokud chceme přijímat všechny zprávy ze sběrnice CAN, nastavíme masku prvního filtru na hodnotu FFFFFFFF (hex) a masku druhého filtru na FFFF (hex).

Pokud chceme přijímat jen zprávy s dvěma konkrétními identifikátory, např. 0340 a 0412 (hex), pak nastavíme masku prvního filtru na 001FFFFF (hex) a hodnotu na 68000000 (hex), masku druhého filtru na 001F (hex) a hodnotu na 8240 (hex).

Dvojitý filtr - rozšířený rámec zprávy

Pokud je přijata zpráva s rozšířeným rámcem, tj. identifikátor má šířku 29 bitů, celý identifikátor a příznak žádosti o data jsou testovány vstupními filtry.



Obr.2.20 Struktura začátku přijaté zprávy s rozšířeným rámcem pro dvojitý filtr
ID - identifikátor zprávy (ve filtru 2 jen nejvyšších 16 bitů)
X - nepoužito

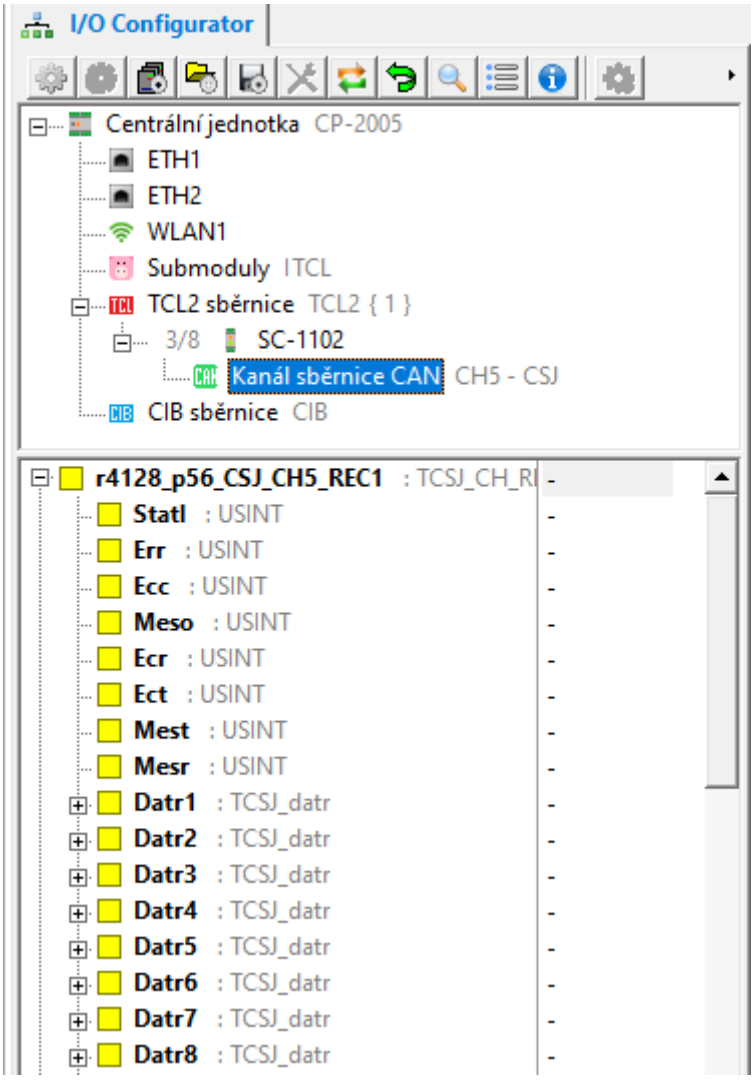
Pro platný příjem musí být všechny bity začátku přijaté zprávy shodné s hodnotou aspoň jednoho filtru. Pokud některé bity nechceme testovat, nastavíme masku filtru na pozici těchto

bitů na hodnotu 1. Nejnižší 2 bity čtvrtého bytu (na obr.2.20 označené 0) nejsou testovány. Z důvodu kompatibility s budoucími verzemi doporučuje výrobce řadiče nastavit příslušné bity masky filtru vždy na hodnotu 1 jako nepoužité.

Pokud chceme přijímat všechny zprávy ze sběrnice CAN, nastavíme masku prvního filtru na hodnotu FFFFFFFF (hex) a masku druhého filtru na FFFF (hex).

Pokud chceme přijímat jen zprávy s dvěma skupinami identifikátorů, např. 10002000 - 10003FFF a 10028000 - 10029FFF (hex), pak nastavíme masku prvního filtru na 0000FFFF (hex) a hodnotu na 80010000 (hex), masku druhého filtru na 0000 (hex) a hodnotu na 8014 (hex).

Po stisknutí tlačítka *Uložit* se ve stromečku konfigurace na řádku příslušného kanálu objeví označení aktuálně nastaveného režimu (obr.2.21).



Obr.2.21 Příklad konfigurace se sériovým kanálem s nastaveným režimem CSJ

2.3.2. Poskytovaná data

Obsluha sítě probíhá na pozadí uživatelského programu asynchronně s dobou cyklu uživatelského programu. Nicméně výměna aktuálních dat se zápisníkem probíhá vždy v otočce cyklu uživatelského programu. Struktura dat je uvedena v tab.2.4.

Položky struktury mají přidělena symbolická jména, která začínají vždy stejnou kombinací znaků určujících adresu a označení kanálu v rámci systému. Ve sloupci *Úplný zápis* je uvedeno vždy konkrétní symbolické jméno pro danou položku. Pokud chceme data použít v uživatelském programu, použijeme buď toto symbolické jméno, nebo ve sloupci *Alias* zapíšeme svoje symbolické jméno, které pak můžeme používat. V žádném případě nepoužíváme absolutní operandy, protože se mohou po novém překladu uživatelského programu změnit.

Alias vytvoříme v nástroji *I/O Configurator* následovně. Poklepáním na jméno modulu ve stromečku konfigurace otevřeme panel *Konfigurace*. V záložce *Procesní data* zapíšeme požadované jméno v příslušném řádku do sloupce *Alias*.

Uspořádání diagnostických dat a komunikačních zón

Komunikační kanál v režimu **CSJ** zveřejňuje diagnostická data stavu řadiče CAN, data přijímaná a data vysílaná. Tyto údaje se ukládají do zápisníku a jsou snadno přístupné v nástroji *I/O Configurator*.

Do vysílací zóny zapisuje uživatel data určená k odeslání. V přijímací zóně jsou data přijatá. Před jejich zpracováním je třeba zkontrolovat hodnotu statusu.

Tab.2.4 Struktura dat (n je číslo sériového kanálu)

Struktura dat	Úplný zápis	Popis objektu
CSJ_CHn_REC1	rx_py_CSJ_CHn_REC1	přijímací zóna
Statl: usint	rx_py_CSJ_CHn_REC1.Statl	
Err: usint	rx_py_CSJ_CHn_REC1.Err	
Ecc: usint	rx_py_CSJ_CHn_REC1.Ecc	
Meso: usint	rx_py_CSJ_CHn_REC1.Meso	
Ecr: usint	rx_py_CSJ_CHn_REC1.Ecr	
Ect: usint	rx_py_CSJ_CHn_REC1.Ect	
Mest: usint	rx_py_CSJ_CHn_REC1.Mest	
Mesr: usint	rx_py_CSJ_CHn_REC1.Mesr	
Datr1	rx_py_CSJ_CHn_REC1.Datr1	
Stat: usint	rx_py_CSJ_CHn_REC1.Datr1.Stat	
Frm: usint	rx_py_CSJ_CHn_REC1.Datr1.Frm	
Data[0...11]: usint	rx_py_CSJ_CHn_REC1.Datr1.Data	
Datr2	rx_py_CSJ_CHn_REC1.Datr2	
:	:	
Datr32	rx_py_CSJ_CHn_REC1.Datr32	
CSJ_CHn_SEND1	rx_py_CSJ_CHn_SEND1	vysílací zóna
Contl: usint	rx_py_CSJ_CHn_SEND1.Contl	
Mess: usint	rx_py_CSJ_CHn_SEND1.Mess	
Dats1	rx_py_CSJ_CHn_SEND1.Dats1	
Stat: usint	rx_py_CSJ_CHn_SEND1.Dats1.Stat	
Frm: usint	rx_py_CSJ_CHn_SEND1.Dats1.Frm	
Data[0...11]: usint	rx_py_CSJ_CHn_SEND1.Dats1.Data	
Dats2	rx_py_CSJ_CHn_SEND1.Dats2	
:	:	
Dats32	rx_py_CSJ_CHn_SEND1.Dats32	
Statistic_CHn	rx_py_Statistic_CHn	diagnostická zóna kanálu
STAT: usint	rx_py_Statistic_CHn.STAT	
ERR: usint	rx_py_Statistic_CHn.ERR	
trueMess: udint	rx_py_Statistic_CHn.trueMess	
falseMess: udint	rx_py_Statistic_CHn.falseMess	
Control_CHn	rx_py_Control_CHn	řídicí zóna kanálu
STOP: bool	rx_py_Control_CHn.STOP	
REST: bool	rx_py_Control_CHn.REST	

Vstupní data kanálu (rx_py_CSJ_CHn_REC1)

Statl - stav řadiče

	RESM	STT	STOP	X	X	X	X	X
bit	.7	.6	.5	.4	.3	.2	.1	.0

RESM - režim řadiče
0 - řadič je v operačním režimu
1 - řadič je v režimu reset
STT - přechod z režimu reset
0 - stav řadiče je stabilní
1 - řadič přechází ze stavu reset do operačního režimu

STOP - přechod do režimu reset
0 - stav řadiče je stabilní
1 - řadič přechází do stavu reset

Err - chyba hlášená řadičem

	BS	EW	X	X	X	BE	DOS	X
bit	.7	.6	.5	.4	.3	.2	.1	.0

BS - stav sběrnice
0 - sběrnice CAN bez závad
1 - stav bus-off
Sběrnice byla odpojena z důvodu vysokého počtu chyb při vysílání, čítač *Ect* dosáhl hodnoty 255. Řadič automaticky přechází do režimu reset. Ovladač komunikačního kanálu se jej následně pokusí restartovat a opět připojit ke sběrnici.

EW - zvýšený počet chyb
0 - počet chyb na sběrnici nepřekročil varovnou mez
1 - varování z důvodu zvýšeného počtu chyb, některý z čítačů *Ecr* nebo *Ect* dosáhl hodnoty 96, nebo ji překročil

BE - chyba sběrnice
0 - provoz na sběrnici CAN bez závad
1 - na sběrnici CAN došlo k chybě (podle specifikace protokolu CAN 2.0B), podrobný údaj o chybě je uložen v proměnné *Ecc*

DOS - přeplnění přijímacího zásobníku v řadiči
0 - přijímací zásobník řadiče pracuje normálně
1 - přijímací zásobník řadiče je přeplněn
Řadič obsahuje vnitřní vyrovnávací zásobník pro příjem dat. Pokud ovladač komunikačního kanálu nestíhá přijaté zprávy vyzvedávat, zásobník se přeplní.

Ecc - chyba sběrnice

	EC1	EC0	DIR	SEG4	SEG3	SEG2	SEG1	SEG0
bit	.7	.6	.5	.4	.3	.2	.1	.0

EC1 - EC0 - typ chyby
00 - chyba bitu (bit error)
Nepovedlo se vyslat bit do sběrnice (na sběrnici je jiná úroveň, než odpovídá vyslanému bitu).
01 - chyba hodnoty (form error)
Bit má jinou hodnotu, než by měl podle protokolu mít.
10 - chyba úrovně (stuff error)
Bylo přijato více než 5 po sobě jdoucích bitů stejné úrovně, což je uvnitř zprávy nepřípustné.
11 - jiná chyba
Další chyby v protokolu.

DIR - směr přenosu
0 - chyba vznikla při vysílání
1 - chyba vznikla při příjmu

SEG4 - SEG0 - segment zprávy, kde došlo k chybě
00010 - bity 28 až 21 rozšířeného, resp. bity 10 až 3 standardního identifikátoru
00011 - začátek rámce zprávy
00100 - bit SRTR
00101 - bit IDE

00110 - bity 20 až 18 rozšířeného, resp. bity 2 až 0 standardního identifikátoru
00111 - bity 17 až 13 rozšířeného identifikátoru
01000 - CRC sekvence
01001 - rezervní bit 0
01010 - datové pole
01011 - délka dat
01100 - bit RTR
01101 - rezervní bit 1
01110 - bity 4 až 0 rozšířeného identifikátoru
01111 - bity 12 až 5 rozšířeného identifikátoru
10001 - příznak aktivní chyby
10010 - přestávka mezi zprávami
10011 - tolerance dominantních bitů
10110 - příznak pasivní chyby
10111 - chybový znak
11000 - CRC znak
11001 - potvrzovací slot
11010 - konec rámce
11011 - potvrzovací znak
11100 - příznak přetížení

Meso - počet ztracených zpráv
Počet zpráv, které nebyly přijaty z důvodu zaplnění vnitřního zásobníku komunikačního kanálu

Ecr - čítač přijatých chybných zpráv
Počet chyb detekovaných řadičem během příjmu. Pokud tento čítač dosáhne nebo překročí hodnotu 96, je v proměnné *Err* nastaven bit *EW*. Příjem zprávy bez chyb sníží hodnotu čítače o 1.

Ect - čítač vyslaných chybných zpráv
Počet chyb detekovaných řadičem během vysílání. Pokud tento čítač dosáhne nebo překročí hodnotu 96, je v proměnné *Err* nastaven bit *EW*. Odeslání zprávy bez chyb sníží hodnotu čítače o 1.

Mest - počet neodvysílaných zpráv
Počet zpráv, které se ještě nestihly odvysílat. Čítač nezahrnuje nové požadavky z předchozího cyklu uživatelského programu.

Mesr - počet přijatých zpráv
Počet zpráv, které byly přijaty během posledního cyklu uživatelského programu a jsou uloženy v následujících polích za sebou tak, jak přišly.

Datr1 - přijatá zpráva 1 (struktura zprávy *TCSJ_DATR* viz kap.2.3.2.1.)
:
Datr32 - přijatá zpráva 32 (struktura zprávy *TCSJ_DATR* viz kap.2.3.2.1.)

Výstupní data sériového kanálu (rx_py_UNI_CHn_OUT)

Contl - řízení řadiče

	RESM	X	X	X	X	X	X	X
bit	.7	.6	.5	.4	.3	.2	.1	.0

RESM - režim řadiče
0 - operační režim
1 - režim reset
Při změně bitu *RESM* z hodnoty 1 na 0 dojde k reinicializaci řadiče podle inicializační tabulky (stejně jako při restartu uživatelského programu).

Mess - počet zpráv ve vysílacím zásobníku
Počet zpráv, které jsou zapsány do vysílacího zásobníku.

Dats1 - zpráva 1 k odvysílání (struktura zprávy *TCSJ_DATS* viz kap.2.3.2.1.)
:
Dats32 - zpráva 32 k odvysílání (struktura zprávy *TCSJ_DATS* viz kap.2.3.2.1.)

Vstupní data diagnostiky kanálu (rx_py_Statistic_CHn)

STAT - status kanálu

	X	X	X	X	X	X	X	X
bit	.7	.6	.5	.4	.3	.2	.1	.0

nepoužito

ERR - chyba komunikace
nepoužito

trueMess - počet platných komunikací

falseMess - počet chybných komunikací

Výstupní data řízení kanálu (rx_py_Control_CHn)

CONT - řízení kanálu

	X	X	X	X	X	X	X	X
bit	.7	.6	.5	.4	.3	.2	.1	.0

nepoužito

2.3.2.1. Struktura zpráv

Přijímané zprávy

Položky *Datr1* až *Datr32* obsahující přijaté zprávy mají každá následující strukturu TCSJ_DATR:

Stat - status zprávy
nepoužito

Frm - formát zprávy

	FF	RTR	X	X	DLC3	DLC2	DLC1	DLC0
bit	.7	.6	.5	.4	.3	.2	.1	.0

FF - formát rámce zprávy
0 - standardní rámec (identifikátor šířky 11 bitů)
1 - rozšířený rámec (identifikátor šířky 29 bitů)
RTR - žádost o data
0 - běžná zpráva s daty
1 - některý uzel sběrnice CAN požaduje vyslání dat s identifikátorem neseným ve zprávě
DLC3 - DLC0 - délka dat
Délka dat nesených ve zprávě v rozmezí 0 až 8.

Data [x] - identifikátor a data zprávy
Struktura závisí na formátu rámce.
Standardní rámec:

byte 0	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3
byte 1	ID2	ID1	ID0	RTR	0	0	0	0
byte 2	DATA1							
:	:							
byte 9	DATA8							
bit	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*

Rozšířený rámec:

byte 0	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21
byte 1	ID20	ID19	ID18	ID17	ID16	ID15	ID14	ID13
byte 2	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5
byte 3	ID4	ID3	ID2	ID1	ID0	RTR	0	0
byte 4	DATA1							
:	:							
byte 11	DATA8							
bit	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*

ID - identifikátor zprávy
RTR - žádost o data
0 - běžná zpráva s daty
1 - některý uzel sběrnice CAN požaduje vyslání dat s identifikátorem neseným ve zprávě
DATA1 - DATA8 - data zprávy
0 až 8 bytů podle hodnoty *DLC* v proměnné *Frm*

Vysílané zprávy

Položky *Dats1* až *Dats32* obsahující vysílané zprávy mají každá následující strukturu TCSJ_DATS:

Cont - řízení zprávy

	0	0	0	0	0	0	0	SEND
bit	.7	.6	.5	.4	.3	.2	.1	.0

SEND - vyslání dat
0 - zpráva nebude vyslána

1 - vyslat zprávu (bit je po zařazení zprávy do vysílací fronty automaticky vynulován)

Frm - formát zprávy

	FF	RTR	0	0	DLC3	DLC2	DLC1	DLC0
bit	.7	.6	.5	.4	.3	.2	.1	.0

FF - formát rámce zprávy
0 - standardní rámec (identifikátor šířky 11 bitů)
1 - rozšířený rámec (identifikátor šířky 29 bitů)

RTR - žádost o data
0 - běžná zpráva s daty
1 - žádost o data od jiného uzlu s identifikátorem neseným ve zprávě

DLC3 - DLC0 - délka dat
Délka dat nesených ve zprávě v rozmezí 0 až 8.

Data [x] - identifikátor a data zprávy
Struktura závisí na formátu rámce.
Standardní rámec:

byte 0	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3
byte 1	ID2	ID1	ID0	RTR	0	0	0	0
byte 2	DATA1							
:	:							
byte 9	DATA8							
bit	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*

Rozšířený rámec:

byte 0	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21
byte 1	ID20	ID19	ID18	ID17	ID16	ID15	ID14	ID13
byte 2	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5
byte 3	ID4	ID3	ID2	ID1	ID0	RTR	0	0
byte 4	DATA1							
:	:							
byte 11	DATA8							
bit	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*

ID - identifikátor zprávy

RTR - žádost o data
Hodnota tohoto bitu není podstatná, nicméně výrobce řadiče doporučuje, aby se jeho hodnota shodovala s hodnotou bitu *RTR* v proměnné *Frm*.

DATA1 - DATA8 - data zprávy
0 až 8 bytů podle hodnoty *DLC* v proměnné *Frm*

2.3.3. Výměna dat

Příjem dat vysílaných jiným uzlem

Do přijímacího zásobníku kanálu se ukládají všechny přijaté zprávy, které propustí přijímací filtr (nastavení viz kap.2.3.1.), v pořadí, jak byly přijaty. Přijímací zásobník umožňuje přijmout až 256 zpráv. Přijaté zprávy jsou během otočky cyklu přesunuty do zápisníku a zveřejněny pro uživatelský program v položkách přijímací zóny *Datr1* až *Datr32*. Pokud se v přijímacím

zásobníku nachází více zpráv než 32, je do přijímací zóny v zápisníku předáno 32 nejstarších zpráv a zbývající čekají na další cyklus PLC.

Počet přijatých zpráv obsahuje proměnná *Mesr*. Bude-li například její hodnota 5, znamená to, že bylo přijato 5 zpráv a jsou uloženy v položkách *Datr1* až *Datr5* v pořadí, jak byly přijaty. Maximální hodnota proměnné *Mesr* je 32. Odpovídá tedy počtu zpráv aktuálně dostupných v přijímací zóně v zápisníku.

Pokud se přijímací zásobník kanálu zcela zaplní, pak další přijaté zprávy budou ztraceny. Počet takto ztracených zpráv obsahuje proměnná *Meso*. Pokud k tomuto jevu dochází, musíme zkrátit dobu cyklu PLC, aby došlo k rychlejšímu odebírání zpráv, nebo omezit intenzitu provozu na sběrnici CAN, nebo změnit nastavení přijímacího filtru tak, aby jím procházelo méně zpráv.

Žádost o data od jiného uzlu

Všechny zprávy, které nesou data, mají bit *RTR* nastaven na hodnotu 0. Pokud přijde zpráva s bitem *RTR* nastaveným na hodnotu 1, pak nenese žádná data, ale naopak požaduje zaslání dat příslušných k identifikátoru, se kterým tato výzva přišla. Uživatelský program by měl reagovat vysláním zprávy s těmito daty.

Například:

přijatá zpráva: ID = 0401 (hex), RTR = 1
odeslaná zpráva: ID = 0401 (hex), RTR = 0, DATA

Směrodatná hodnota bitu *RTR* je vždy v proměnné *Frm*.

Vysílání dat do jiného uzlu

Do vysílací zóny v zápisníku do položek *Dats1* až *Dats32* zapisujeme zprávy, které chceme vyslat.

Počet platných zpráv k odeslání obsahuje proměnná *Mess*. Bude-li například její hodnota 5, znamená to, že je připraveno k odeslání 5 zpráv v položkách *Dats1* až *Dats5*, které budou odeslány v tomto pořadí.

Vyslání každé zprávy je ještě podmíněno nastavením bitu *SEND* na hodnotu 1 v proměnné *Cont* ve struktuře zprávy. V otočce cyklu je v okamžiku zařazení zprávy do vysílací fronty komunikačního kanálu tento bit smazán. Pokud je řadič v režimu reset, pak zpráva není do vysílací fronty zařazena a bit *SEND* zůstává nastaven na hodnotu 1.

Počet neodvysílaných zpráv ve vysílací frontě obsahuje proměnná *Mest*. Jedná se o hodnotu zjištěnou při vstupu do otočky cyklu, tedy před přidáním nových zpráv k odvysílání. Je to tedy počet minulých zpráv, které ještě čekají ve frontě na odvysílání. Pokud se tato hodnota stále zvyšuje, znamená to, že řadič nestíhá zprávy vysílat. Vnitřní vysílací fronta pojme 255 zpráv. Je řešena pomocí kruhového zásobníku, tzn. že pokud dojde k jeho zaplnění, pak zápisem 256. zprávy dojde k jeho smazání a ztrátě všech zpráv včetně právě zapsané zprávy. Následující zpráva bude opět normálně vyslána.

Díky podmínkovému bitu *SEND* můžeme k vysílací zóně přistupovat dvěma způsoby. Buď položky *Dats1* až *Dats32* používáme jako klasický zásobník, tzn. že v každém cyklu do nich zapíšeme zprávy k odeslání, všem nastavíme příslušné bity *SEND* na hodnotu 1 a jejich počet zapíšeme do proměnné *Mess*.

Můžeme ale také položky *Dats1* až *Dats32* využívat staticky, tzn. že si každou položku vyhradíme pro zprávy s konkrétním identifikátorem. V tomto případě máme zprávy trvale připravené v položkách *Dats1* až *Dats32*, v každém cyklu jim aktualizujeme data a příslušné bity *SEND* nastavujeme na hodnotu 1 periodicky (např. každých 200 ms a každou v jiném cyklu). Do proměnné *Mess* zapíšeme počet všech takto obsluhovaných položek, nikoli počet aktuálně vysílaných zpráv. Tento postup se hodí především k vysílání periodických datových objektů.

Oba uvedené postupy můžeme kombinovat, např. první čtyři položky *Dats1* až *Dats4* vyhradíme pro periodické vysílání a položky *Dats5* až *Dats32 používáme jako* klasický zásobník zpráv. Do proměnné *Mess* pak zapisujeme počet všech zpráv uložených do vysílací zóny bez

ohledu na stav příslušných bitů *SEND*. Je to tedy hodnota odpovídající nejvyššímu indexu použité položky *Dats* (např. pokud poslední zprávu uložíme do položky *Dats18*, zapíšeme do proměnné *Mess* hodnotu 18).

Načtení dat z jiného uzlu

Pokud chceme přečíst data z jiného uzlu (stanice), který tato data aktivně nevysílá, musíme vyslat žádost, kterou představuje zpráva s příslušným identifikátorem, ve které je bit *RTR* nastaven na hodnotu 1. Tato zpráva nenese data, nicméně délka dat *DLC* se zpravidla nastavuje na hodnotu délky očekávaných dat. Důvodem je, aby nedošlo na sběrnici k chybě, pokud dva řadiče *CAN* začnou vysílat žádost o data se stejným identifikátorem současně.

Pro vysílání těchto zpráv platí úplně stejné podmínky jako pro datové zprávy (viz předchozí odstavce). Směrodatná hodnota bitu *RTR* je vždy v proměnné *Frm*.

2.3.4. Režimy řadiče a chybové stavy

Režimy řadiče

Řadič CAN pracuje ve dvou základních režimech - v režimu reset a v operačním režimu. V režimu reset je řadič odpojen od sběrnice CAN a umožňuje nastavení komunikačních parametrů. V operačním režimu je řadič připojen ke sběrnici CAN a provádí komunikaci podle nastavených parametrů, které v tomto režimu nelze měnit.

Do režimu reset se řadič dostane třemi způsoby:

- při restartu uživatelského programu
- příkazem z uživatelského programu
- při vysokém výskytu chyb na sběrnici CAN

Do operačního režimu se řadič dostane třemi způsoby:

- po restartu uživatelského programu
- příkazem z uživatelského programu
- po automatické reinicializaci vyvolané vysokým výskytem chyb na sběrnici CAN

Stav řadiče indikují bity v proměnné *Statl* v přijímací zóně. Bit *RESM* má v režimu reset hodnotu 1. Přechodové stavy mezi oběma režimy jsou indikovány bity *START* a *STOP*.

Režim reset vyvolaný uživatelským programem

Během restartu uživatelského programu je řadič CAN uveden do režimu reset. V tomto režimu je řadič odpojen od sběrnice a umožňuje nastavení komunikačních parametrů. Po inicializaci řadiče je spuštěn operační režim řadiče.

Do režimu reset lze řadič uvést kdykoli pomocí bitu *RESM* v proměnné *Contl* ve vysílací zóně. Dokud tento bit bude mít hodnotu 1, řadič setrvává v režimu reset. Jakmile se hodnota bitu *RESM* v proměnné *Contl* změní na hodnotu 0, řadič je opět inicializován na hodnoty dané uživatelským programem a opouští režim reset.

Při přechodu do operačního režimu řadič vždy vyčká na klid na sběrnici (linka má klidovou úroveň po dobu odpovídající min. 11 bitů).

Chyby na sběrnici

Řadič obsahuje čítače chyb sběrnice detekovaných při příjmu a vysílání. Jejich hodnoty jsou přístupné v proměnných *Ecr* a *Ect*. Vlastní chyba sběrnice je hlášena bitem *BE* v proměnné *Err* a v proměnné *Ecc* je uvedena podrobná diagnostika chyby.

Pokud některý z obou čítačů dosáhne hodnotu 96, je v proměnné *Err* nastaven bit *EW* na log.1. Po dosažení hodnoty 255 na čítači chyb při vysílání je v proměnné *Err* nastaven bit *BS* na

log.1, řadič se odpojí od sběrnice a samostatně přechází do stavu reset. V následující otočce cyklu je řadič automaticky reinicializován a opět převeden do operačního režimu.

Na rozdíl od opouštění režimu reset vyvolaného řízenou žádostí (restart uživatelského programu, restart řadiče z uživatelského programu) při přechodu do operačního režimu po chybě na sběrnici řadič vždy vyčká na 128 výskytů klidu na sběrnici (linka má klidovou úroveň po dobu odpovídající min. 11 bitů). Počet výskytů klidu na sběrnici indikuje čítač *Ect* odpočítáváním z hodnoty 127 do 0. Čítač *Ecr* je automaticky vynulován.

3. KOMUNIKAČNÍ REŽIMY ROZHRANÍ ETHERNET, WLAN A LTE

Kompaktní PLC TECOMAT FOXTROT 2 jsou standardně vybaveny dvěma nezávislými rozhraními Ethernet 10/100baseT podle IEEE 802.3. To umožňuje komunikovat přes standardní LAN sítě s provozovaným TCP / IP transportním protokolem rychlostí 10 nebo 100 Mb/s. Komunikace zahrnuje služby potřebné pro programování PLC, služby pro sdílení dat mezi PLC a také služby využívané SCADA / HMI systémy.

Připojení do sítě Ethernet je zajištěno přímo prostřednictvím konektorů RJ45 a kabelů UTP podle mezinárodního standardu IEEE 802.3. IP adresy pro obě rozhraní lze nastavit pomocí tlačítek na čelním panelu PLC nebo z prohlížeče (na výchozí adrese 192.168.134.176:8080).

Volitelně mohou PLC TECOMAT FOXTROT 2 obsahovat také rozhraní WLAN pro WiFi komunikaci a rozhraní LTE pro komunikaci prostřednictvím GSM sítě. Jejich chování je z hlediska přenosu dat shodné se sítí Ethernet, proto je v následujícím textu zahrnujeme pod kanály Ethernet.

Adresování Ethernetu

Každý kanál Ethernet má svoji IP adresu a IP masku. Jejich hodnoty závisí na nastavení protějšího účastníka komunikace. Obecně platí zásada, že IP adresy obou účastníků komunikace musí být shodné v těch místech, kde má IP maska nenulovou hodnotu. IP maska by měla být pro oba účastníky shodná, není to však podmínkou.

Příklad:	PC		PLC	
	IP adresa:	192.168.1.1	IP adresa:	192.168.1.2
	IP maska:	255.255.255.0	IP maska:	255.255.255.0
	nebo:			
	PC		PLC	
	IP adresa:	192.168.12.1	IP adresa:	192.168.25.8
	IP maska:	255.255.0.0	IP maska:	255.255.0.0

Pozor: První číselný údaj hodnoty IP adresy **nesmí být 0**, poslední číselný údaj hodnoty IP adresy **nesmí být 0 nebo 255**! Poslední číselný údaj hodnoty IP masky **nesmí být 255**! V případě zadání chybných hodnot IP adresy a IP masky nastaví centrální jednotka tyto parametry na implicitní hodnoty dané výrobcem.

IP adresa a IP maska platí pro všechny provozované režimy na všech spojeních na jednom kanálu Ethernet.

Výchozí nastavení ETH1 a ETH2

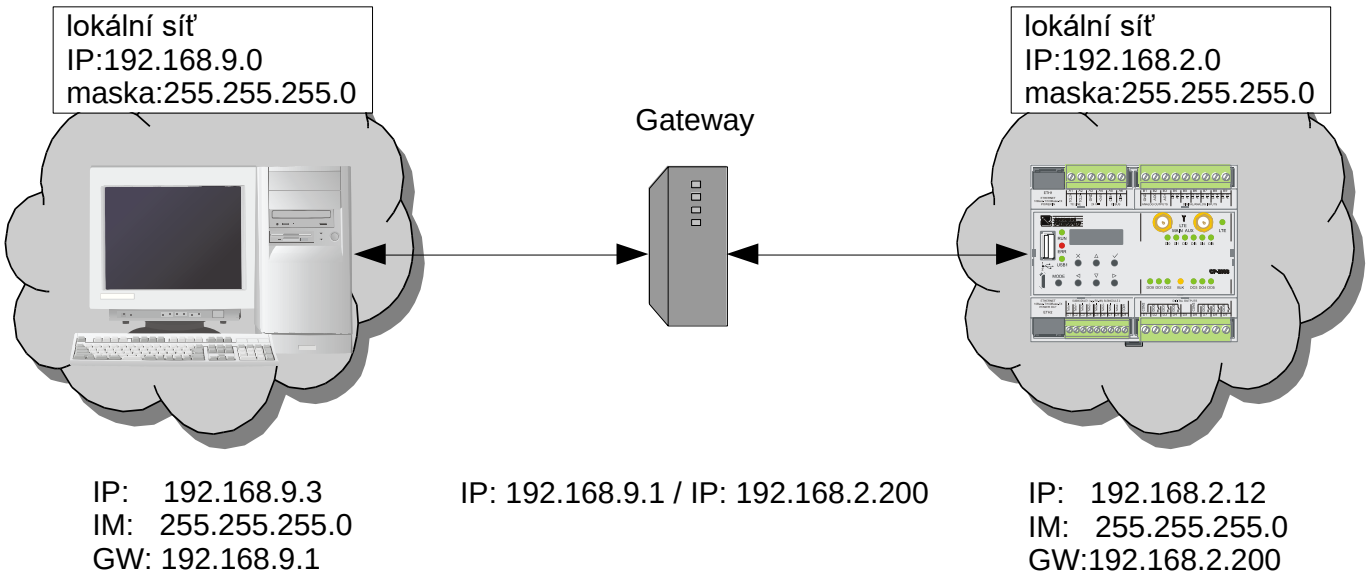
Výrobce dodává systémy FOXTROT 2 s následujícím nastavením:

ETH1	IP adresa	192.168.134.176 (fixní)
	maska sítě	255.255.255.0
	brána	nenastavena (0.0.0.0)
	DNS server	nenastaven (0.0.0.0)
ETH2	zapnuto přidělování adres DHCP serverem	

Toto nastavení lze kdykoli uživatelsky změnit.

Komunikace mezi sítěmi pomocí brány

Pokud potřebujeme připojit počítač, který se nachází v jiné síti, než PLC TECOMAT, obě sítě propojíme pomocí tzv. brány (gateway). Brána má v každé z obou sítí přidělenou IP adresu (obr.3.1). Protože zprávy mezi sítěmi jsou směrovány přes bránu, musí se v PLC nastavit její IP adresa.



Obr.3.1 Příklad propojení dvou sítí pomocí brány (gateway)

Režimy komunikace

Přes kanály Ethernet lze komunikovat protokolem EPSNET UDP v režimech **PC, PLC a PLD.** V režimu **UNI** lze posílat a přijímat libovolná data v UDP nebo TCP paketu.

Režim **PC** je trvale aktivní a umožňuje i programování PLC. Komunikovat lze současně přes 10 až 32 spojení (počet spojení je nastavitelný uživatelem). Systémové služby ale může v daném okamžiku využívat jen jeden účastník. Toto omezení platí pro všechny komunikační kanály systému jako celku nezávisle na typu a umístění kanálu (tedy včetně kanálu na rozhraní USB).

Dále je možné zapnout režim **PLC** nebo **PLD** pro vzájemnou výměnu dat mezi PLC (rozhraní LTE tyto dva režimy nepodporuje).

A konečně režim **UNI** slouží pro vysílání a příjem dat pomocí protokolu UDP nebo TCP. Nástroj *I/O Configurator* umožňuje nastavit na jednom Ethernet kanálu až 8 spojení v režimu **UNI**. Další spojení v režimu **UNI** lze otevřít pomocí funkce *OpenUniSocket()* z knihovny *ComLib*. Funkcí *OpenUniSocket()* lze současně otevřít maximálně 128 spojení (ať už na stejném Ethernet kanálu nebo na různých Ethernet kanálech - celkové maximum otevřených UNI kanálů je 128).

Tab.3.1 Seznam režimů na kanálech Ethernet a jejich možnosti

Komunikační režim	Vstupní port	Počet spojení	Protokol	Aktivace
PC	61682	10 - 32	EPSNET UDP / TCP	trvale
PLC	61681	1	EPSNET UDP	uživatelským programem
PLD	volitelný	1	EPSNET UDP	uživatelským programem
UNI	volitelný	8 + 128	obecný UDP, obecný TCP	uživatelským programem

Zapnuté režimy **PLC**, **PLD** a **UNI** nijak neovlivňují funkci a možnosti komunikací v režimu **PC**, ani sebe navzájem.

V následujících kapitolách jsou popsány jednotlivé komunikační režimy použitelné na kanálech Ethernet (tab.3.2).

Tab.3.2 Přehled komunikačních režimů kanálů Ethernet

Komunikační režim	Funkce	Popis
UNI	obecný uživatelský kanál	kap.3.1.
PC	komunikace s nadřízeným systémem	kap.3.2.
PLC	síť se sdílením dat (kompatibilní s PLC FOXTROT a TC700)	kap.3.3.
PLD	síť se sdílením dat (pouze pro PLC FOXTROT 2, možnost šifrování)	kap.3.4.

3.1. REŽIM UNI - OBECNÝ UŽIVATELSKÝ KANÁL

Režim **UNI** je určen pro univerzální použití. Pokud je komunikační kanál v tomto režimu, pak lze tímto kanálem z uživatelského programu odesílat a přijímat libovolná data. K obsluze vysílání a příjmu jsou určeny bloky *fbSendTo* a *fbRecvFrom* z knihovny *ComLib*.

Kanály Ethernet v tomto režimu umožňují výměnu dat s obecným zařízením pomocí protokolu UDP nebo TCP. Na každém kanálu lze pomocí nástroje *I/O Configurator* deklarovat až 8 nezávislých spojení. Další spojení lze v případě potřeby otevřít funkcí *OpenUniSocket.* Každé spojení umožňuje přenos až 1350 bytů v jedné zprávě obecným protokolem UDP nebo TCP. V případě protokolu TCP je nutné určit, jestli má PLC navázat spojení s protější stanicí (nastavení *TCP Client* pro běžnou komunikaci nebo *TLS/SSL Client* pro kryptovanou komunikaci), nebo čekat, až spojení naváže protější stanice (nastavení *TCP Server*).

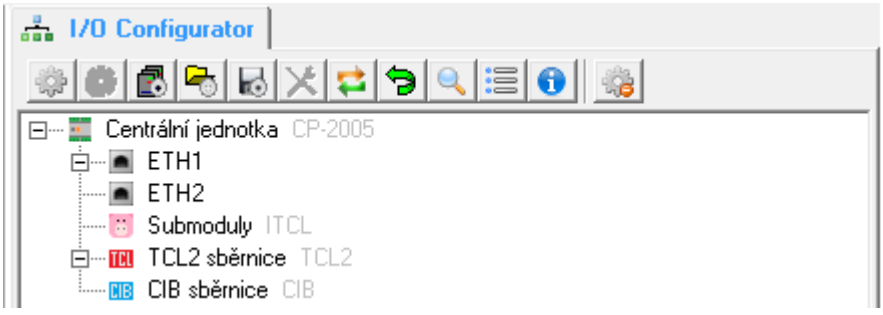
3.1.1. Konfigurace

Konfigurace komunikačních kanálů stejně jako konfigurace všech částí PLC TECOMAT FOXTROT 2 se v prostředí Mosaic provádí pomocí nástroje *I/O Configurator*.

Rozhraní obsluhovaná přímo centrální jednotkou jsou označena ETH1, ETH2, WLAN1, WLAN2 a LTE1.

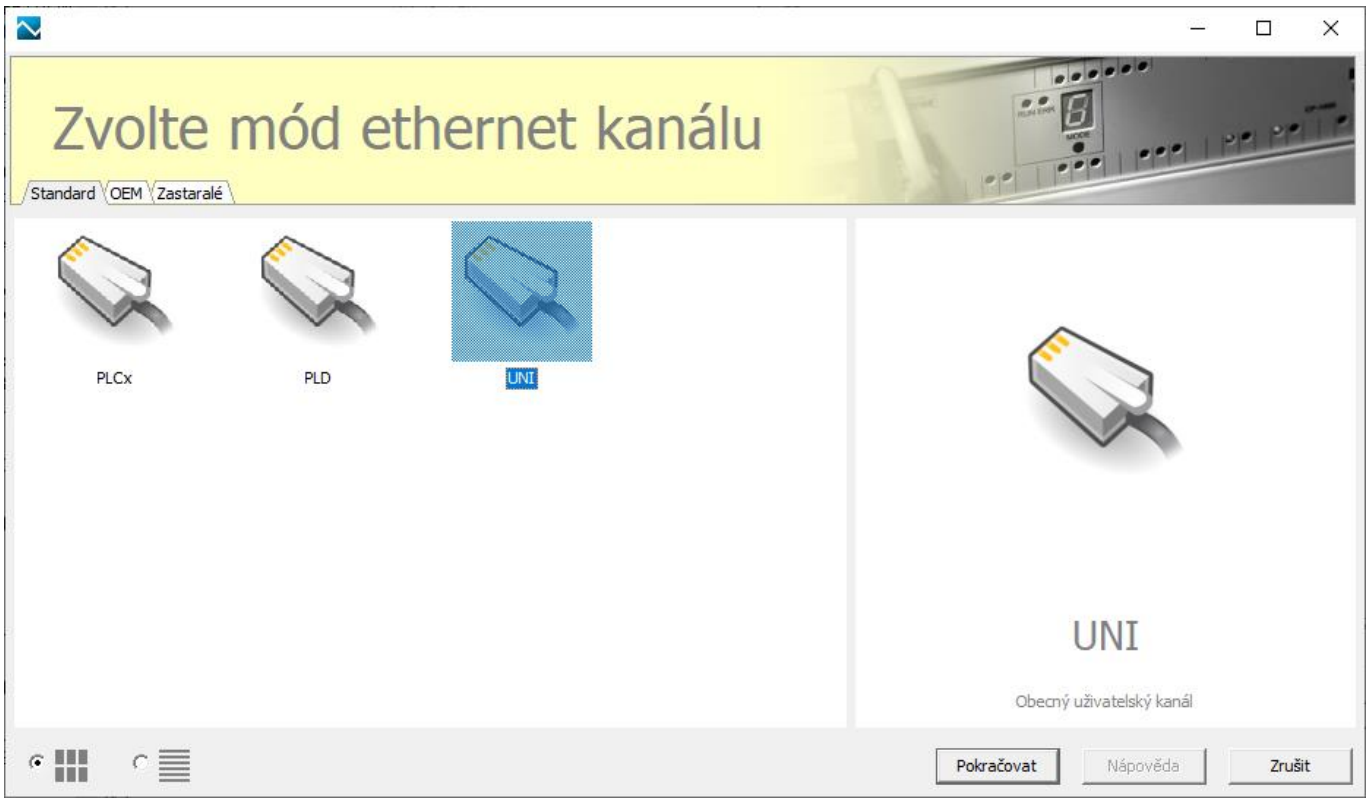
Nastavení režimu kanálu

Režim kanálu nastavíme tak, že v okně nástroje *I/O Configurator* poklepeme levým tlačítkem myši na označení příslušného rozhraní (např. ETH1).



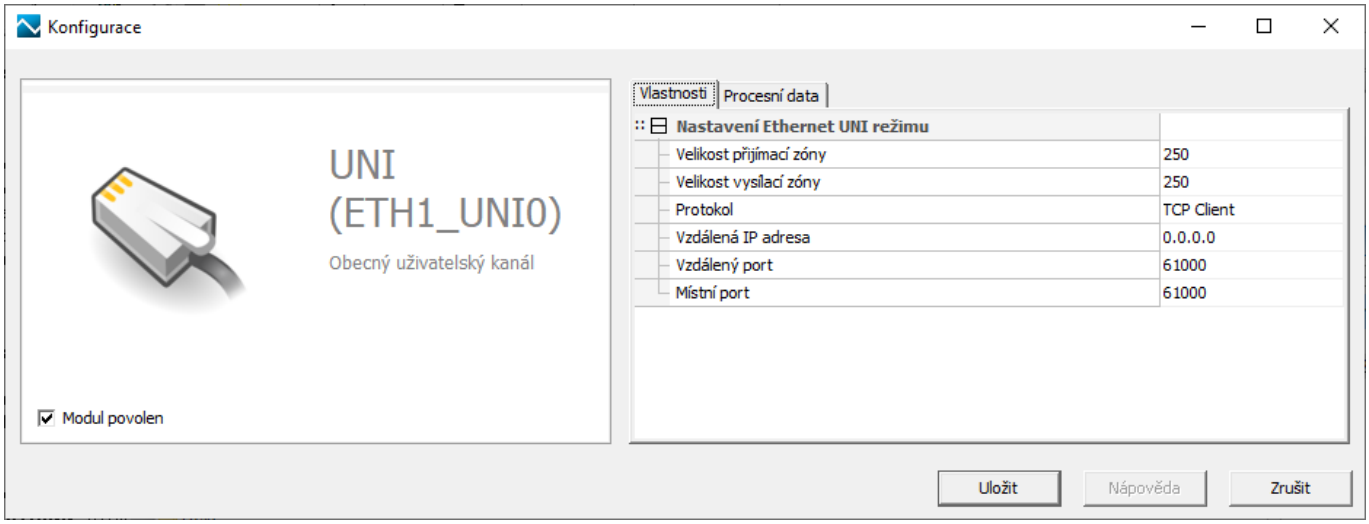
Obr.3.2 Výběr rozhraní Ethernet

Otevře se okno výběru režimu kanálu rozhraní Ethernet (obr.3.3).



Obr.3.3 Výběr režimu kanálu na rozhraní Ethernet

Poklepáním vybereme režim **UNI** a otevře se panel konfigurace (obr.3.4).



Obr.3.4 Konfigurace režimu UNI na jednom spojení na rozhraní Ethernet

V levé části panelu *Konfigurace* se nachází položka *Modul povolen*. Pokud je zaškrtnuta, provádí se obsluha kanálu podle nastavení dostupných v pravé části panelu. Pokud položka nebude zaškrtnutá, je obsluha kanálu vypnuta beze ztráty aktuálního nastavení. To lze využít při ladění uživatelského programu.

V pravé části panelu se v záložce *Vlastnosti* nachází všechny položky konfigurace daného kanálu. V záložce *Procesní data* pak nalezneme seznam všech proměnných, které kanál poskytuje.

Význam a možnosti nastavení jednotlivých položek v pravé části panelu jsou následující:

Velikost přijímací zóny

Velikost přijímací zóny v bytech, max. 1350 B.

Velikost vysílací zóny

Velikost vysílací zóny v bytech, max. 1350 B.

Typ protokolu

TCP Client - Přenos dat protokolem TCP. PLC aktivně vytváří a udržuje spojení s protější stanicí. Protější stanice by měla být pasivní (TCP Server) a čekat na navázání spojení.

TCP Server - Přenos dat protokolem TCP. PLC nevytváří spojení s protější stanicí, ale čeká, až tato stanice provede spojení sama. Protější stanice by tedy měla být TCP Client.

TLS/SSL Client - Přenos dat protokolem TCP s šifrováním. PLC aktivně vytváří a udržuje spojení s protější stanicí. Protější stanice by měla být pasivní (TCP Server) a čekat na navázání spojení.

UDP - Přenos dat protokolem UDP.

Vzdálená IP adresa

IP adresa protější stanice, se kterou se provádí spojení.

V případě TCP Server se tento údaj nevyplňuje. PLC čeká na výzvu od stanice TCP Client bez ohledu na její IP adresu (viz kap.3.1.4.).

Vzdálený port (remote port)

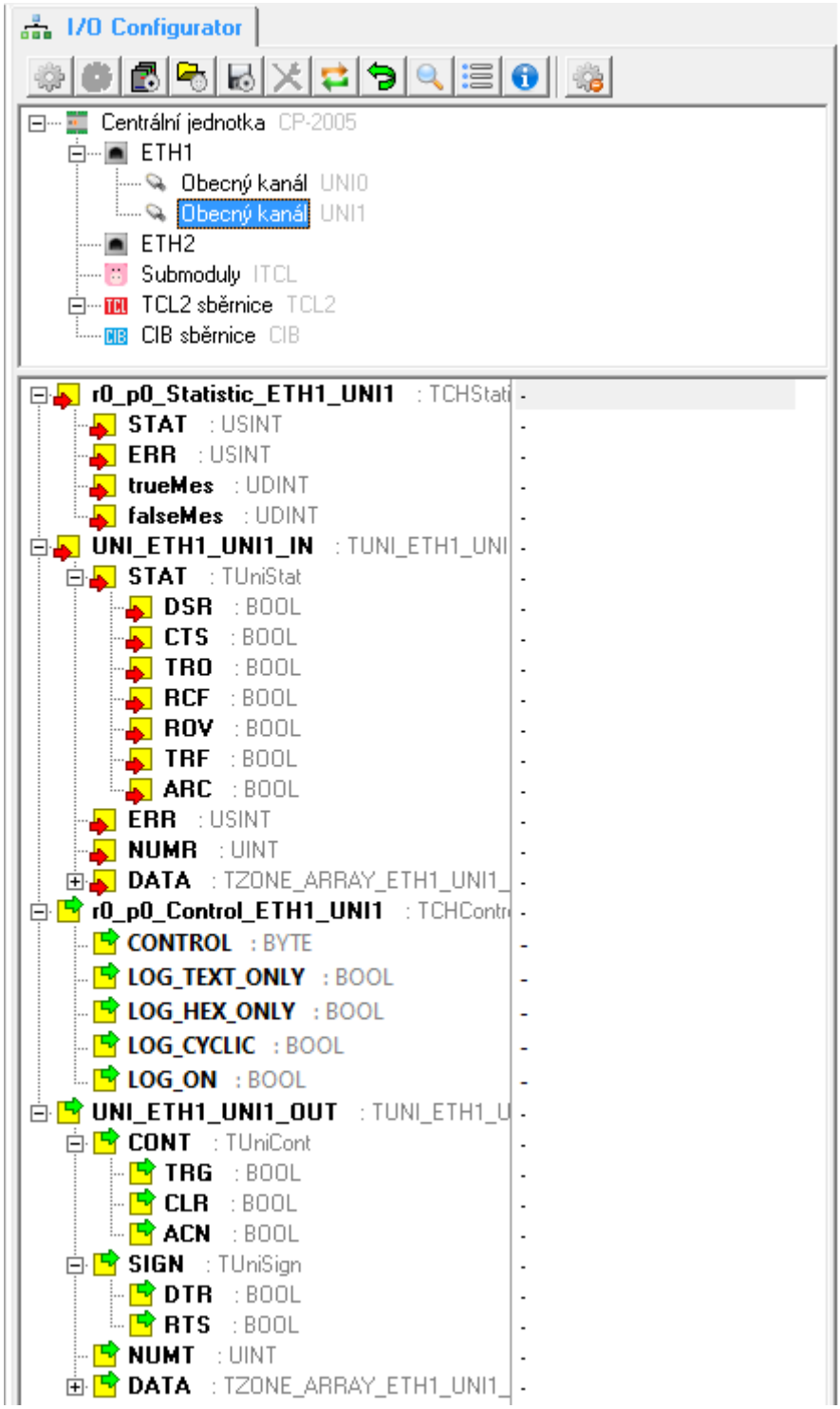
Vstupní port protější stanice, se kterou se provádí spojení.

V případě TCP Server se tento údaj nevyplňuje. PLC čeká na výzvu od stanice TCP Client bez ohledu na její port (viz kap.3.1.4.).

Místní port (local port)

Vstupní port PLC - číslo portu, na kterém PLC přijímá zprávy od protější stanice. Lze zadat jakékoli číslo vyšší než 1024, kromě 61680 - 61699, které jsou vyhrazeny pro systémové použití.

Po stisknutí tlačítka *Uložit* se ve stromečku konfigurace pod řádkem příslušného rozhraní objeví řádek aktuálně otevřeného spojení UNI0 - UNI7 (obr.3.5).



Obr.3.5 Příklad konfigurace se dvěma spojeními v režimu UNI na rozhraní ETH1

Pokud chceme změnit konfiguraci konkrétního již vytvořeného spojení, poklepeme levým tlačítkem myši na řádek tohoto spojení.

Pokud chceme přidat nové spojení, poklepeme levým tlačítkem myši na řádek příslušného rozhraní (např. ETH1) a postupujeme opět od začátku výběrem režimu kanálu.

Pokud chceme stávající spojení ubrat, označíme levým tlačítkem myši řádek tohoto spojení a stiskneme klávesu *Del*, nebo stiskneme pravé tlačítko myši a vybereme položku *Vymazat*.

Na jednom rozhraní může být maximálně 8 takto vytvořených spojení v režimu **UNI**.

3.1.2. Poskytovaná data

Obsluha komunikačního kanálu v režimu **UNI** probíhá prostřednictvím komunikačních zón v zápisníkové paměti. Struktura dat je uvedena v tab.3.3.

K obsluze komunikačního kanálu v uživatelském programu slouží knihovna *ComLib*. Doporučujeme její použití z důvodu vyvarování se problémů s obsluhou příznaků kanálu v reálném čase. Knihovna umožňuje kromě obsluhy 8 základních spojení otevřít ještě další spojení pomocí funkce *OpenUniSocket*. Tato spojení lze obsluhovat jedině pomocí knihovny.

Následující informace jsou určeny především pro ty, kteří knihovnu nepoužijí a týkají se pouze základních 8 spojení.

Položky struktury mají přidělena symbolická jména, která začínají vždy stejnou kombinací znaků určujících adresu a označení kanálu v rámci systému. Ve sloupci *Úplný zápis* je uvedeno vždy konkrétní symbolické jméno pro danou položku. Pokud chceme data použít v uživatelském programu, použijeme buď toto symbolické jméno, nebo ve sloupci *Alias* zapíšeme svoje symbolické jméno, které pak můžeme používat. V žádném případě nepoužíváme absolutní operandy, protože se mohou po novém překladu uživatelského programu změnit.

Alias vytvoříme v nástroji *I/O Configurator* následovně. Poklepáním na jméno modulu ve stromečku konfigurace otevřeme panel *Konfigurace*. V záložce *Procesní data* zapíšeme požadované jméno v příslušném řádku do sloupce *Alias*.

Uspořádání diagnostických dat a komunikačních zón

Komunikační kanál v režimu **UNI** zveřejňuje diagnostická data stavu linky, přijímaná a vysílaná data a stav komunikace. Tyto údaje se ukládají do zápisníku a jsou snadno přístupné v nástroji *I/O Configurator*.

Do vysílací zóny zapisuje uživatel data určená k odeslání. V přijímací zóně jsou data přijatá. Před jejich zpracováním je třeba zkontrolovat hodnotu statusu.

Tab.3.3 Struktura dat (ccc je označení kanálu, např. ETH1, n je číslo spojení)

Struktura dat	Úplný zápis	Popis objektu
Statistic_ccc_UNIn	rx_py_Statistic_ccc_UNIn	diagnostická zóna kanálu
STAT: usint	rx_py_Statistic_ccc_UNIn.STAT	
ERR: usint	rx_py_Statistic_ccc_UNIn.ERR	
trueMess: udint	rx_py_Statistic_ccc_UNIn.trueMess	
falseMess: udint	rx_py_Statistic_ccc_UNIn.falseMess	
UNI_ccc_UNIn_IN	UNI_ccc_UNIn_IN	přijímací zóna
STAT	UNI_ccc_UNIn_IN.STAT	
DSR: bool	UNI_ccc_UNIn_IN.STAT.DSR	
CTS: bool	UNI_ccc_UNIn_IN.STAT.CTS	
TRO: bool	UNI_ccc_UNIn_IN.STAT.TRO	
RCF: bool	UNI_ccc_UNIn_IN.STAT.RCF	
ROV: bool	UNI_ccc_UNIn_IN.STAT.ROV	
TRF: bool	UNI_ccc_UNIn_IN.STAT.TRF	
ARC: bool	UNI_ccc_UNIn_IN.STAT.ARC	
ERR: usint	UNI_ccc_UNIn_IN.ERR	
NUMR: uint	UNI_ccc_UNIn_IN.NUMR	
DATA[...]: usint	UNI_ccc_UNIn_IN.DATA	
Control_ccc_UNIn	rx_py_Control_ccc_UNIn	řídicí zóna kanálu
CONTROL: uint	rx_py_Control_ccc_UNIn.CONTROL	
LOG_TEXT_ONLY: bool	rx_py_Control_ccc_UNIn.LOG_TEXT_ONLY	
LOG_HEX_ONLY: bool	rx_py_Control_ccc_UNIn.LOG_HEX_ONLY	
LOG_CYCLIC: bool	rx_py_Control_ccc_UNIn.LOG_CYCLIC	
LOG_ON: bool	rx_py_Control_ccc_UNIn.LOG_ON	
UNI_CHn_OUT	UNI_ccc_UNIn_OUT	vysílací zóna
CONT	UNI_ccc_UNIn_OUT.CONT	
TRG: bool	UNI_ccc_UNIn_OUT.CONT.TRG	
CLR: bool	UNI_ccc_UNIn_OUT.CONT.CLR	
ACN: bool	UNI_ccc_UNIn_OUT.CONT.ACN	
SIGN	UNI_ccc_UNIn_OUT.SIGN	
DTR: bool	UNI_ccc_UNIn_OUT.SIGN.DTR	
RTS: bool	UNI_ccc_UNIn_OUT.SIGN.RTS	
NUMT: uint	UNI_ccc_UNIn_OUT.NUMT	
DATA[...]: usint	UNI_ccc_UNIn_OUT.DATA	

Vstupní data diagnostiky kanálu (rx_py_Statistic_ccc_UNIn)

STAT - status kanálu

	X	X	X	X	X	X	X	CON
bit	.7	.6	.5	.4	.3	.2	.1	.0

CON - stav spojení TCP
- 0 - rozpojeno
- 1 - spojeno

ERR - chyba komunikace (viz kap.3.1.5.)

trueMess - počet platných komunikací

falseMess - počet chybných komunikací

Vstupní data komunikačního kanálu (UNI_ccc_UNIn_IN)

STAT - status vysílání a příjmu

	ARC	TRF	ROV	RCF	TRO	X	X	X
bit	.7	.6	.5	.4	.3	.2	.1	.0

ARC - alternace příjmu
- při nově přijaté zprávě dojde ke změně hodnoty bitu

TRF - 0 - neprobíhá žádné vysílání
- 1 - probíhá vysílání, další zpráva bude převzata až po odvysílání

ROV - 0 - příjem zprávy v pořádku
- 1 - přetečení, přijatá zpráva je delší, než vyhrazená přijímací zóna

RCF - 0 - příjem zpráv bez omezení
1 - přijímací zásobníky jsou plné, poslední přijatá zpráva bude ztracena při dalším příjmu

TRO - 0 - vysílání zpráv bez omezení
1 - vysílací zásobníky jsou plné, zápis další zprávy bude neplatný

ERR - chyba příjmu (viz kap.3.1.5.)

NUMR - počet přijatých bytů

DATA[x] - přijatá zpráva

Výstupní data řízení kanálu (rx_py_Control_ccc_UNIn)

CONTROL - řízení kanálu

	0	0	0	0	0	0	0	RES
bit	.7	.6	.5	.4	.3	.2	.1	.0

RES - reset spojení TCP
- 0 - beze změny
- 1 - reset spojení

LOG_TEXT_ONLY
- 0 - ukládat do souboru .LOG data v číselné i textové formě
- 1 - ukládat do souboru .LOG data pouze v textové formě

LOG_HEX_ONLY
- 0 - ukládat do souboru .LOG data v číselné i textové formě
- 1 - ukládat do souboru .LOG data pouze v číselné formě (formát hex.)

LOG_CYCLIC - 0 - ukládat do souboru .LOG data pouze do jeho zaplnění
- 1 - ukládat do souboru .LOG data cyklicky, soubor tak obsahuje poslední data

LOG_ON - 0 - ukládání do souboru .LOG vypnuto
- 1 - ukládání do souboru .LOG zapnuto

Tyto proměnné slouží k ovládání monitorování provozu kanálu, chování je identické se sériovým kanálem (viz kap.2.1.7.).

Výstupní data komunikačního kanálu (UNI_ccc_UNIn_OUT)

CONT - řízení vysílání a příjmu

	ACN	CLR	TRG	X	X	X	X	X
bit	.7	.6	.5	.4	.3	.2	.1	.0

ACN - alternace řízení
- při změně bitu dojde k akceptování hodnot ostatních bitů CONT

CLR - 1 - vymazání vysílacích a přijímacích zásobníků

TRG - 1 - spuštění vysílání zprávy

SIGN - řízení modemových signálů

	X	X	X	X	X	X	X	X
bit	.7	.6	.5	.4	.3	.2	.1	.0
	nepoužito							

NUMT - počet vysílaných bytů

DATA[x] - vysílaná zpráva

3.1.3. Vysílání a příjem dat

Předávání zpráv určených k odvysílání z uživatelského programu do vnitřního vysílacího zásobníku komunikačního kanálu a přebírání přijatých zpráv z vnitřního přijímacího zásobníku komunikačního kanálu do uživatelského programu je prováděno výhradně v otočce cyklu se zajištěnou časovou konzistencí dat (tj. shodné „stáří“ všech bytů zprávy).

Kanály Ethernet umožňují vysílat i přijímat data o délce až 1350 bytů.

K obsluze vysílání a příjmu dat jsou určeny bloky *fbSendTo* a *fbRecvFrom* z knihovny *ComLib*. Doporučujeme její použití z důvodu vyvarování se problémů s obsluhou příznaků kanálu v reálném čase. Následující informace jsou určeny především pro ty, kteří knihovnu nepoužijí, a týkají se jen 8 základních spojení.

Postup při vysílání zprávy

Vysílání zpráv přes kanály Ethernet se liší v závislosti na použitém protokolu.

V případě protokolu UDP se v otočce cyklu přesune zpráva z vysílací zóny komunikačního kanálu do zásobníku UDP soketu, odkud je bezprostředně odeslána. V následující otočce lze bez omezení připravit další zprávu do vysílací zóny komunikačního kanálu a změnou bitu *ACN* jí odvysílat. UDP protokolem je tedy možné vysílat v každém cyklu PLC novou zprávu. Aby se nepřerušovalo spojení (nedošlo k uzavření UDP soketu), je třeba komunikovat s periodou kratší než 2 s.

V případě protokolu TCP je nezbytné nejprve navázat spojení a teprve poté je možné zahájit vlastní vysílání zpráv. PLC se automaticky pokusí navázat TCP spojení v případě, že je komunikační kanál v režimu TCP Client a bit *RES* v proměnné *CONTROL* má hodnotu 0.

Pokud je komunikační kanál v režimu TCP Server, očekává se navázání spojení od protistrany. Při navázaném spojení má bit *CON* v proměnné *STAT* hodnotu 1. Před přípravou nové zprávy do vysílací zóny komunikačního kanálu je nutné zkontrolovat, jestli není nastaven bit *TRO* nebo *TRF* v proměnné *STAT* v přijímací zóně kanálu na hodnotu 1. Pokud jsou tyto bity nulové, je možné zapsat novou zprávu do vysílací zóny komunikačního kanálu, nastavit bit *TRG* na hodnotu 1 a změnou bitu *ACN* zahájit vysílání zprávy. Pokud má bit *TRO* nebo *TRF* hodnotu 1 znamená to, že ještě nedošlo k odvysílání předchozí zprávy (zásobník TCP soketu je plný, důvodem může být např. opakování zpráv v navázaném TCP spojení z důvodu ztráty paketu na lince). V tomto případě **nesmí** dojít ke změně vysílací zóny komunikačního kanálu, bit *TRG* **musí** být nastaven na hodnotu 1 a bit *ACN* se **nesmí** změnit. Zpráva je pak přesunuta z vysílací zóny komunikačního kanálu do zásobníku TCP soketu, jakmile je v zásobníku soketu dostatečné místo (poté co se podaří odvysílat předchozí zprávy nahromaděné v TCP soketu). Poté spadnou bity *TRO* a *TRF* na hodnotu 0 a je možné připravit další zprávu do vysílací zóny komunikačního kanálu.

V každém případě nelze přes kanály Ethernet odvysílat více než jednu zprávu za jeden cyklus PLC.

Postup při příjmu zprávy

Přijme-li komunikační kanál zprávu, předá ji v otočce cyklu do přijímací zóny v zápisníku PLC. Bit *ARC* v proměnné *STAT* změní hodnotu (alternuje).

Pokud je přijatá zpráva delší, než je velikost přijímací zóny v zápisníku, je bit *ROV* v proměnné *STAT* nastaven na hodnotu 1.

Příjem více zpráv za sebou

Kanály Ethernet jsou vybaveny přijímacím zásobníkem s dostatečnou kapacitou, ze kterého se v každé otočce cyklu přenese celá zpráva, pokud je přijata. V následující otočce se přenese další přijatá zpráva, atd. Bit *RCF* se nenastavuje.

Vymazání vysílacích a přijímacích zásobníků

Nastavením bitu *CLR* v proměnné *CONT* na log.1 provedeme vymazání vysílacích a přijímacích zásobníků. Všechny zprávy, které zde byly uloženy, budou ztraceny. Nastavení kanálu se nemění. Bit je třeba v následujícím cyklu opět vynulovat.

Tento příkaz lze kombinovat s příkazem vysílání, vymazání zásobníků se provede vždy jako první.

Alternace bitu *ACN* v proměnné *CONT* je podmínkou pro akceptování tohoto příkazu.

3.1.4. Nastavení parametrů spojení a ovládání rozhraní

Protokol UDP

V případě protokolu UDP vytvoříme spojení tak, že oběma stanicím nadefinujeme parametry podle následujícího příkladu:

	PLC	vzdálená stanice
protokol	UDP	UDP
vlastní IP adresa	192.168.33.160	192.168.33.166
místní port	61000	61001
vzdálená IP adresa	192.168.33.166	192.168.33.160
vzdálený port	61001	61000

Při tomto nastavení si obě stanice mohou vzájemně posílat data protokolem UDP. Místní i vzdálený port mohou mít stejné hodnoty.

Pokud je číslo místního portu v PLC nastaveno na 0, pak je číslo portu při každém jeho otevření přiděleno automaticky.

Pokud je požadavek přijímat data z více vzdálených stanic, pak nadefinujeme parametry následovně:

	PLC	vzdálená stanice 1	vzdálená stanice 2
protokol	UDP	UDP	UDP
vlastní IP adresa	192.168.33.160	192.168.33.166	192.168.33.169
místní port	61000	61001	61002
vzdálená IP adresa	0.0.0.0	192.168.33.160	192.168.33.160
vzdálený port	0	61000	61000

Při tomto nastavení PLC čeká, až některá vzdálená stanice naváže spojení. Poté jsou posílána data protokolem UDP. Pokud vzdálená stanice přestane komunikovat, po cca 2 sekundách je spojení uvolněno a spojení může navazovat jiná vzdálená stanice. Místní i vzdálené porty mohou mít stejné hodnoty.

Aby se nepřerušovalo spojení, je třeba komunikovat s periodou kratší než 2 s.

Protokol TCP

V případě protokolu TCP musíme rozhodnout, která stanice bude aktivně navazovat a udržovat spojení. Pokud chceme, aby spojení navazoval PLC (TCP Client), nadefinujeme parametry podle následujícího příkladu:

	PLC	vzdálená stanice
protokol	TCP Client	TCP Server
vlastní IP adresa	192.168.33.160	192.168.33.166
místní port	61000	61001
vzdálená IP adresa	192.168.33.166	0.0.0.0
vzdálený port	61001	0

Při tomto nastavení začne PLC po přechodu do režimu RUN navazovat spojení se vzdálenou stanicí. Ta musí čekat, až se spojení naváže. Po navázání spojení jsou posílána data protokolem TCP. Místní i vzdálené porty mohou mít stejné hodnoty.

Pokud je číslo místního portu v PLC v režimu TCP Client nastaveno na 0, pak je číslo portu při každém jeho otevření přiděleno automaticky. V režimu TCP Server musí být číslo místního portu jednoznačně určeno.

Pokud chceme, aby spojení navazovala vzdálená stanice, pak je PLC v režimu TCP Server a parametry nadefinujeme následovně:

	PLC	vzdálená stanice
protokol	TCP Server	TCP Client
vlastní IP adresa	192.168.33.160	192.168.33.166
místní port	61000	61001
vzdálená IP adresa	0.0.0.0	192.168.33.160
vzdálený port	0	61000

Pokud nastavíme bit *RES* v proměnné *CONTROL* v řízení kanálu, bude proveden reset spojení, který spočívá v uzavření spojení. Spojení bude uzavřeno po celou dobu, kdy má bit *RES* hodnotu 1. Když se hodnota bitu *RES* změní zpět na 0 a PLC je nastaven jako Client, provede vzápětí nové otevření spojení. Pro navázání spojení lze použít funkci *EstabTcpConnection*, pro ukončení spojení je určena funkce *CloseTcpConnection*, obě funkce jsou z knihovny *ComLib*.

Pokud je PLC nastaven jako Server, očekává navázání nového spojení vzdálenou stanicí.

3.1.5. Chybová hlášení

Provozní chybová hlášení sériového kanálu se zobrazují v přijímací zóně v proměnné *ERR* a v diagnostické zóně v proměnné *ERR*.

$31 chybná délka vysílaných dat

Hodnota délky vysílaných dat překračuje velikost vysílací zóny.

$32 nulová délka vysílaných dat

Hodnota délky vysílaných dat musí být nenulová.

Pokud jsou pro vysílání a příjem použity funkční bloky *fbRecvFrom* a *fbSendTo* z knihovny *ComLib* pak jsou chybová hlášení k dispozici na výstupech *error* u těchto bloků.

3.2. REŽIM PC - KOMUNIKACE S NADŘÍZENÝM SYSTÉMEM

Režim **PC** slouží k připojení nadřízeného systému k PLC. Komunikace probíhá protokolem EPSNET UDP/TCP, který obsahuje i systémové služby umožňující programování a ladění systému z prostředí Mosaic. V jednom časovém okamžiku může využívat systémové služby jen jeden nadřízený systém na jednom spojení. Naopak služby pro čtení a zápis dat z paměti proměnných mohou být použity současně na více kanálech, resp. spojeních najednou.

Komunikace jsou vyvolávány nadřízeným systémem na principu dotaz - odpověď. PLC TECOMAT v režimu **PC** se chová jako pasivní podřízený účastník (server).

Veřejné a systémové služby sítě EPSNET

V režimu **PC** je na všech komunikačních kanálech dostupný kompletní soubor služeb sítě EPSNET, jehož součástí jsou jak systémové služby, určené k programování a ladění systému, tak i veřejné služby, určené k výměně dat. Omezující podmínkou je, že systémové služby podporuje v daném časovém okamžiku pouze jeden komunikační kanál.

Síť EPSNET a její veřejné služby jsou popsány v kap.5.

Upozornění: Vývojové prostředí Mosaic využívá systémové služby sítě EPSNET a z tohoto důvodu je lze provozovat pouze přes jeden komunikační kanál v režimu **PC**. Oprávnění využívat systémové služby dostane ten komunikační kanál, který přijde s požadavkem jako první. Oprávnění zaniká 5 sekund po ukončení komunikace.

3.3. SÍTĚ SE SDÍLENÍM DAT

Pro sdílení dat lze v systémech FOXTROT 2 použít režimy **PLC, PLCx** nebo **PLD**. V jednom systému FOXTROT 2 lze nakonfigurovat maximálně 4 sítě pro sdílení dat s těmito režimy.

Režim PLC, PLCx

Režimy **PLC** a **PLCx** jsou plně kompatibilní s režimem **PLC** ve starších systémech FOXTROT a TC700. Používají se tedy především v situacích, kdy je třeba sdílet data mezi různými generacemi systémů TECOMAT.

Charakteristické vlastnosti sdílení dat v režimu PLC, PLCx

- všechny PLC systémy musí být umístěny ve stejném segmentu lokální sítě Ethernet
- maximální počet PLC v síti je 32
- jeden PLC systém může zveřejnit (sdílet) max. 238 bytů dat
- data jsou do sítě vysílána nešifrovaně (v broadcast paketu)
- frekvence sdílení dat je společná pro všechny PLC systémy (společná perioda vysílání dat)
- ve všech systémech jsou sdílená data uložena na stejných absolutních adresách
- na každém rozhraní ETH1, ETH2 lze založit pouze jednu síť v režimu **PLCx**

Režimy **PLC** a **PLCx** jsou funkčně shodné a liší se pouze způsobem konfigurace sítě pro sdílení dat.

V případě režimu **PLC** se konfigurace provádí v prostředí Mosaic přes Manažer projektu (nástroj *Select PLC)*. Jedná se tedy o případ, kdy je síť se sdílením dat vytvořená některou z předcházejících verzí prostředí Mosaic s nástrojem *Select PLC* a je třeba přidat do této sítě nový systém FOXTROT 2.

Režim **PLCx** se použije v případě, kdy jsou všechny projekty pro PLC, mezi kterými chceme sdílet data, vytvořeny v prostředí Mosaic s nástrojem *I/O Configurator* a některý z PLC systémů v síti je typu FOXTROT (základní moduly CP-1xxx). Konfigurace sítě se sdílením dat se provádí v nástroji *I/O Configurator*.

Režim PLD

Pro sdílení dat mezi systémy FOXTROT 2 je určen režim **PLD**. Tento režim je rozšířením původního režimu **PLC** a **nelze** jej použít pro starší systémy FOXTROT a TC700.

Charakteristické vlastnosti sdílení dat v režimu PLD

- všechny PLC systémy musí být umístěny ve stejném segmentu lokální sítě Ethernet
- maximální počet PLC v síti je 64
- jeden PLC systém může zveřejnit (sdílet) max. 1190 bytů dat
- data je možné do sítě vysílat šifrovaně (výchozí stav)
- perioda vysílání dat je individuálně nastavitelná pro každý systém
- sdílená data jsou ve všech systémech uložena ve shodně pojmenovaných proměnných, které nemusí být nutně umístěny na stejných absolutních adresách
- všechna sdílená data jsou doplněna informací o tom, z jaké IP adresy byla data odeslána
- na každém rozhraní ETH1, ETH2 lze založit pouze jednu síť v režimu **PLD**

Síť se sdílením dat v režimu **PLD** lze konfigurovat pouze nástrojem *I/O configurator*.

3.3.1. Režim PLC a PLCx - síť se sdílením dat

Režim **PLC**, resp. **PLCx**, je určen pro situace, kdy je třeba sdílet data mezi různými generacemi systémů TECOMAT. Jinými slovy pokud je třeba sdílet data systémů FOXTROT 2 se systémy FOXTROT nebo TC700. Vlastnosti sdílení dat jsou v tomto případě určovány schopnostmi starších systémů FOXTROT a TC700. Podrobný popis je v dokumentaci Sériová komunikace PLC TECOMAT - model 32 bitů (TXV 004 10.01), kap.2.3. Režim PLC - síť se sdílením dat.

Pro sdílení dat jsou ve všech systémech v síti vyhrazeny předávací zóny pro každého účastníka sítě, které jsou vždy na stejných adresách v paměti proměnných. Předávací zóna dat z jednoho systému je vždy umístěna na stejných adresách ve všech ostatních systémech propojených do sítě. Výhodou tohoto způsobu sdílení je vysoká propustnost sítě, protože stejné rozmístění předávacích zón ve všech systémech dovoluje přenos jednoho bloku dat vždy do všech účastníků sítě zároveň. Tím dochází ke značné časové úspoře a nižším nárokům na strojový čas procesorů centrálních jednotek.

Konfigurace předávacích zón sítě je součástí uživatelských programů propojených systémů a **musí být pro všechny systémy shodná** (pokud jsou jednotlivé systémy součástí stejné skupiny projektů, tak tento požadavek zajistí vývojové prostředí Mosaic).

Režim **PLC** je na síti realizován tak, že každý účastník vysílá svá data do sítě zprávou typu broadcast (všeobecný oběžník – IP adresa 255.255.255.255, MAC adresa FF:FF:FF:FF:FF:FF) v časovém taktu nastavitelném v rozmezí 10 - 2550 ms. Současně přijímá tyto zprávy od ostatních účastníků. Výhodou tohoto typu komunikace je vysoká propustnost sítě. Je však třeba mít na paměti, že zprávy typu broadcast přijímají všichni účastníci v LAN, i když je nezpracovávají. Navíc při krátkém časovém taktu je těmito komunikacemi silně zatížená linka a může docházet ke kolizím paketů.

Z těchto důvodů nedoporučujeme režim **PLC** určený pro výměnu dat v reálném čase provozovat přes síť, která slouží i dalším s technologií nesouvisejícím účastníkům (standardní podniková síť). Pro výměnu dat mezi PLC je vhodnější vyhradit samostatnou síť.

Režim **PLC** je nástrojem *I/O Configurator* podporován pouze z důvodu zpětné kompatibility a umožňuje pouze naimportovat konfiguraci sítě vytvořenou nástrojem *Select PLC*. V systémech FOXTROT 2 by se tento režim měl používat pouze v případech, kdy je třeba sdílet data se staršími systémy FOXTROT a TC700. Podrobný popis nastavení sítě je uveden v dokumentaci Nástroj I/O Configurator a v dokumentaci Sériová komunikace PLC TECOMAT - model 32 bitů (TXV 004 10.01).

Sdílení dat v režimu **PLCx** lze plně zkonfigurovat nástrojem *I/O Configurator*. Podmínkou je, aby všechny projekty pro PLC, mezi kterými budeme sdílet data, byly ve stejné projektové skupině a všechny byly vytvořeny s nástrojem *I/O Configurator*.

3.3.2. Režim PLD - síť s rozšířeným sdílením dat

Režim **PLD** je určen pro sdílení dat mezi systémy FOXTROT 2.

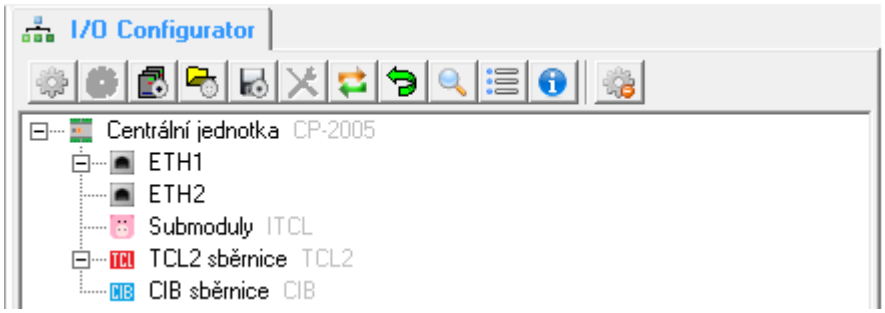
Pro sdílení dat jsou ve všech systémech vyhrazeny předávací zóny pro každého účastníka sítě. Tyto zóny mají ve všech systémech shodné názvy a velikosti a nemusí být umístěny na stejných adresách v paměti proměnných. Konfigurace předávacích zón sítě je součástí uživatelských programů propojených systémů. Jednotlivé projekty sdílejí v prostředí Mosaic shodnou konfiguraci sítě.

Režim **PLD** je na síti realizován tak, že každý účastník vysílá svá data do sítě zprávou typu broadcast (všeobecný oběžník – IP adresa 255.255.255.255, MAC adresa FF:FF:FF:FF:FF:FF) v časovém taktu nastavitelném v rozmezí 10 - 2550 ms. Každý účastník sítě může zveřejňovat data v jiném časovém taktu. Vysílaná data mohou být šifrována (doporučeno) a mohou být doplněna časovou značkou, která omezuje platnost dat na krátký časový úsek. Současně každý účastník přijímá zprávy od ostatních účastníků, dešifruje přijatá data a pokud jsou platná, tak uloží přijatá data do příslušné předávací zóny v paměti proměnných. Zápis i čtení předávacích zón probíhají mezi jednotlivými cykly výpočtu PLC programu, takže konzistence dat je zajištěna.

Výhodou tohoto typu komunikace je vysoká propustnost sítě. Je však třeba mít na paměti, že zprávy typu broadcast přijímají všichni účastníci v LAN, i když je nezpracovávají. Navíc při krátkém časovém taktu je těmito komunikacemi silně zatížená linka a může docházet ke kolizím paketů. Z těchto důvodů nedoporučujeme režim **PLD** určený pro výměnu dat v reálném čase provozovat přes síť, která slouží i dalším účastníkům kromě PLC (standardní podniková síť). Pro takovou výměnu dat mezi PLC je vhodnější vyhradit samostatnou síť. Naopak pokud PLC systém zveřejňuje 100 bytů dat jednou za sekundu, pak to pro běžnou síť nebude problém.

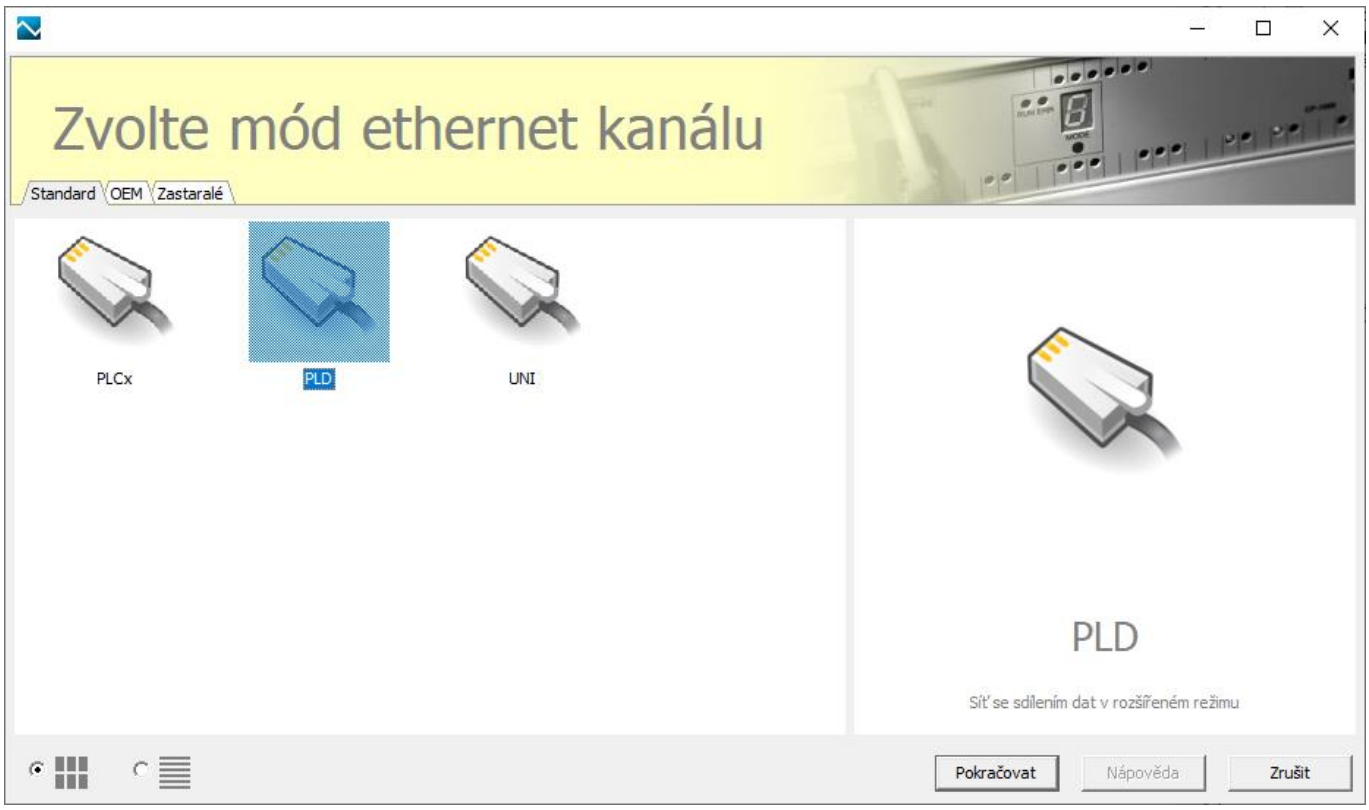
3.3.2.1. Založení sítě v režimu PLD

Předpokládejme, že máme založený projekt pro FOXTROT CP-2005. Úvodním krokem konfigurace sítě **PLD** je nastavení komunikačního kanálu do režimu **PLD** (např. na rozhraní ETH1).



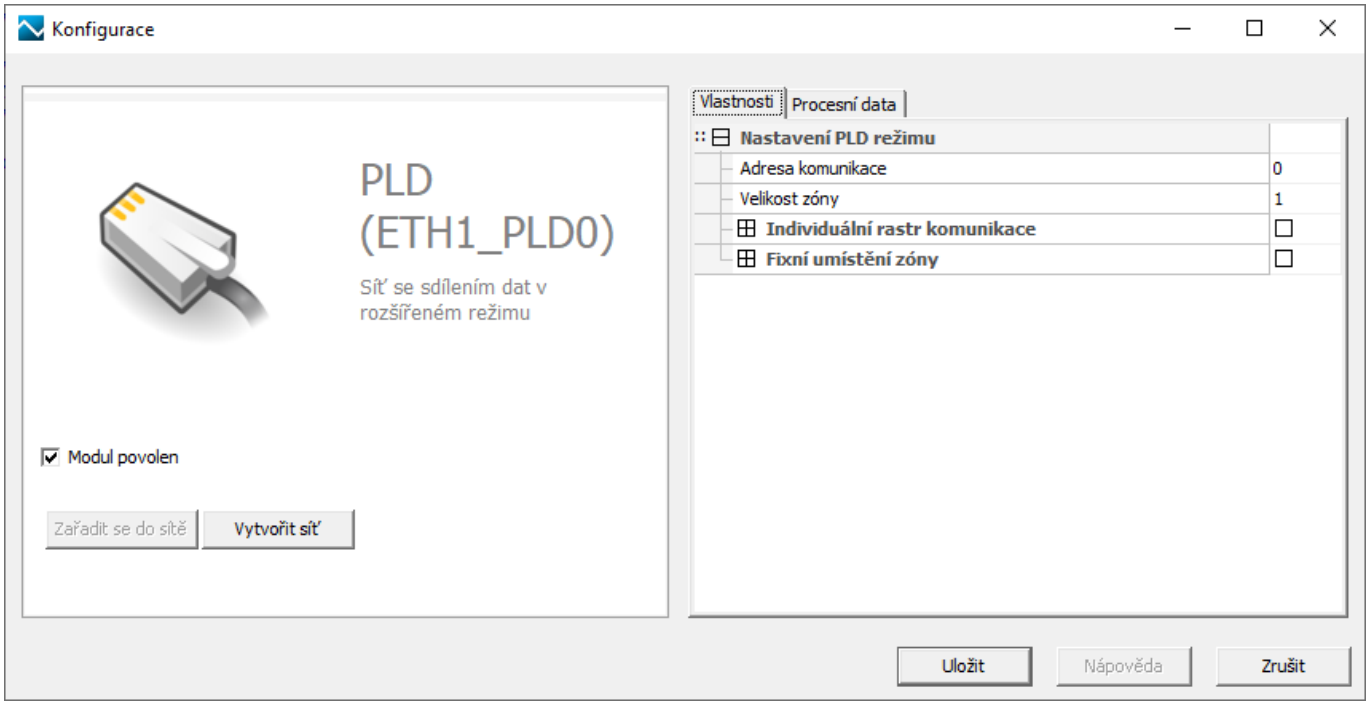
Obr.3.6 Výběr komunikačního kanálu

Založení komunikačního kanálu na rozhraní Ethernet lze po výběru příslušného uzlu ve stromu konfigurace provést dvojitým kliknutím levého tlačítka myši, stiskem klávesy ENTER nebo volbou *Přidat* z kontextové nabídky. Následuje zobrazení výběrového dialogu režimu komunikačního kanálu.



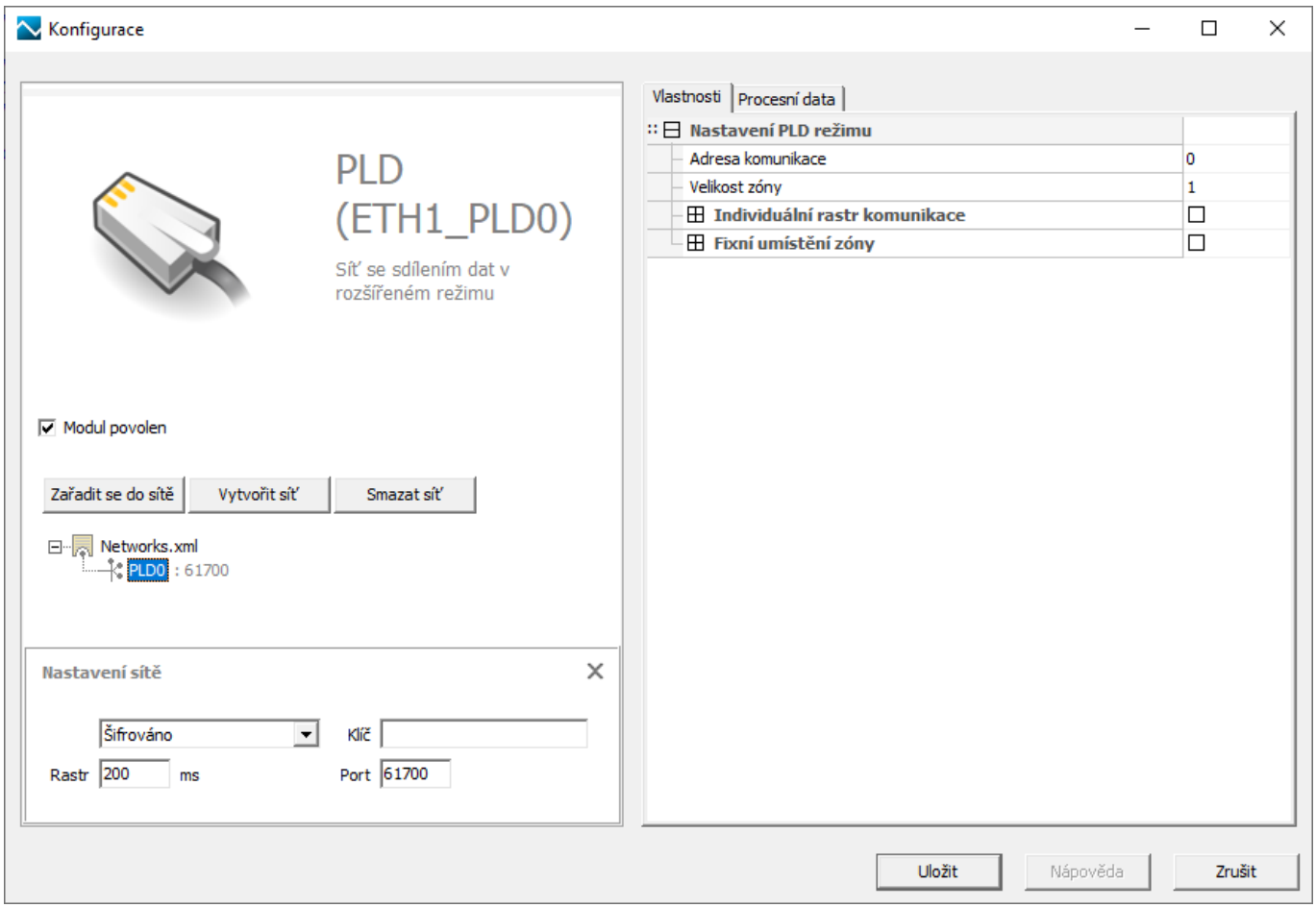
Obr.3.7 Volba režimu komunikace

Dvojitým kliknutím na ikonu pro režim **PLD** vyvoláme dialog pro nastavení kanálu.



*Obr.3.8 Dialog nastavení kanálu v režimu **PLD***

V tomto dialogu je třeba nejprve kliknout na tlačítko *Vytvořit síť*. Pokud zakládáme první síť v aktuální projektové skupině, vyskočí okno s dotazem, zda se má založit soubor *Networks.xml*. Dotaz je třeba potvrdit stisknutím tlačítka *Ano*. Poté se v dialogu objeví uzel nově založené sítě s výchozím názvem *PLD0*. Šifrování dat je zapnuto, rastr vysílání dat je 200 ms a komunikační port je 61700 (viz levá polovina dialogu na obr.3.9).



*Obr.3.9 Dialog vytvoření sítě v režimu **PLD***

Nyní je třeba nahradit výchozí název sítě *PLD0* vlastním názvem sítě.

Další volby v části *Nastavení sítě* mají následující význam:

Režim komunikace

Komunikace, kterými se sdílí data, mohou být

- nešifrované
- šifrované (výchozí volba)
- šifrované s časovým razítkem (pouze pro systémy, jejichž čas je synchronizován)

Rastr

Základní rastr pro vysílání dat z každé stanice, pokud není zvolen *Individuální rastr komunikace* (viz dále)

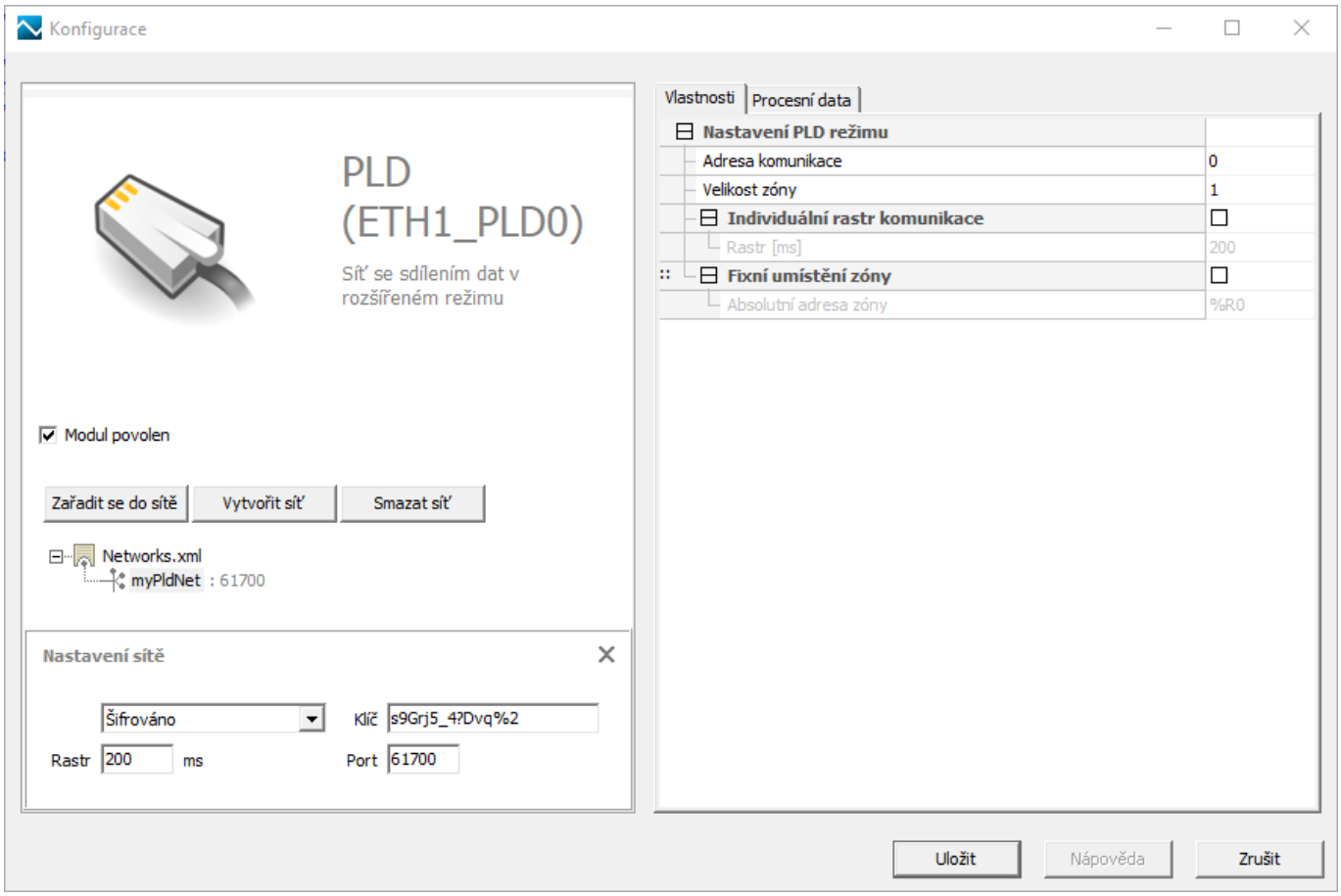
Klíč

Klíč, který se použije pro šifrování dat (libovolný text, nesmí obsahovat národní znaky)

Port

Výchozí port pro komunikaci - pokud chceme změnit výchozí hodnotu, je třeba použít číslo větší než 1024 a vynechat rozsah 61680 až 61699

Pokud tedy nazveme nově založenou síť např. *myPldNet* a zadáme klíč pro šifrování dat, tak bude dialog vypadat následovně:



*Obr.3.10 Příklad nastavení sítě v režimu **PLD***

Dalším krokem je nastavení parametrů vysílaných dat na záložce *Vlastnosti*. Jednotlivé položky mají následující význam:

Adresa komunikace

Každý PLC systém má přidělenou logickou adresu pro komunikaci (v rámci sítě musí být jedinečná) v rozsahu 0 až 255.

Velikost zóny

Velikost vysílaných dat (počet bytů) v rozsahu 1 až 1190 bytů.

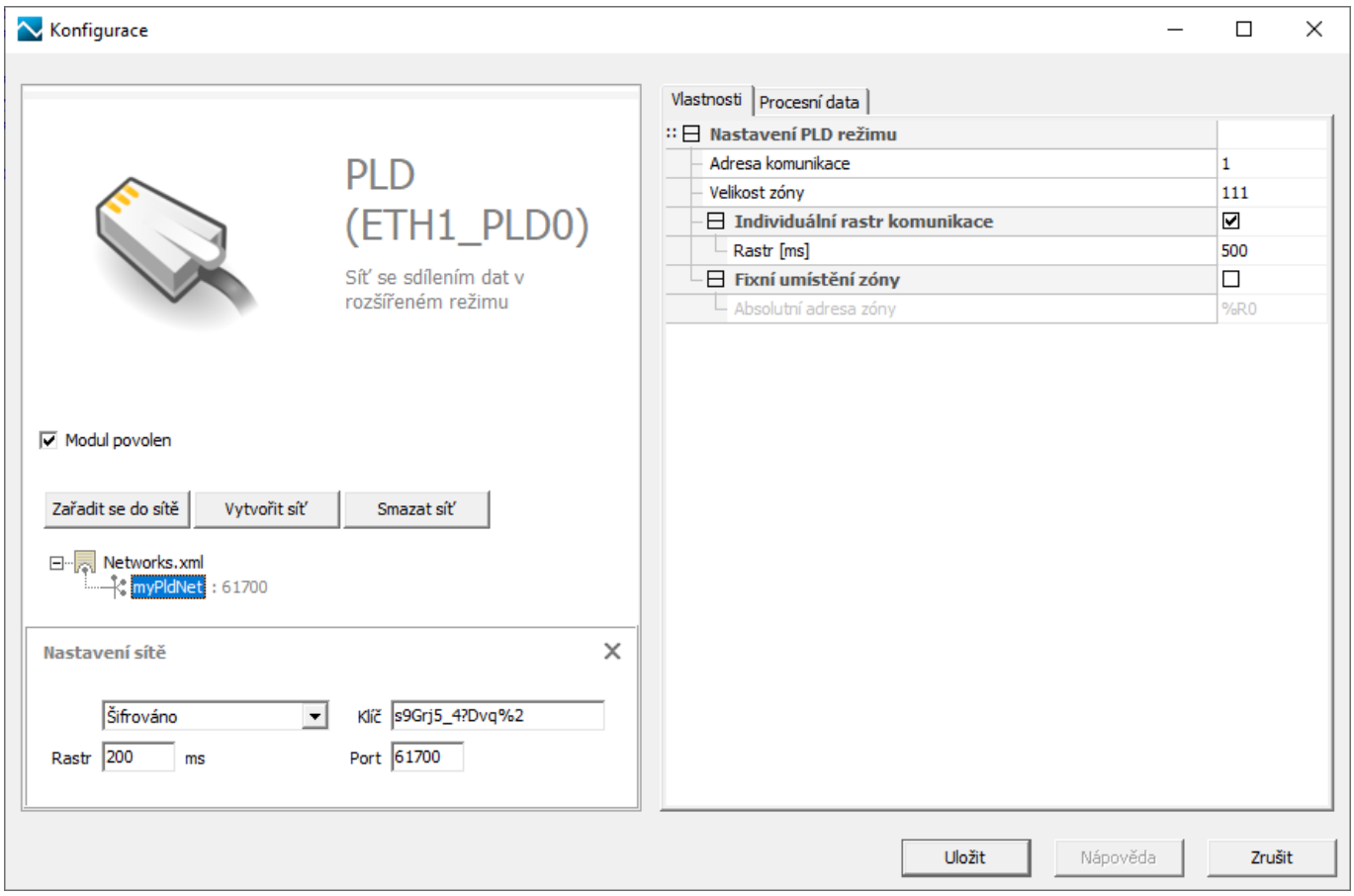
Individuální rastr komunikace

Umožňuje nastavit vlastní rastr pro vysílání dat v rozsahu 10 až 2550 ms, není-li zatrženo, použije se rastr komunikace z nastavení sítě.

Fixní umístění zóny

Umožňuje zafixovat uložení zóny sdílených dat ve všech PLC systémech na konkrétní adresu v paměti proměnných.

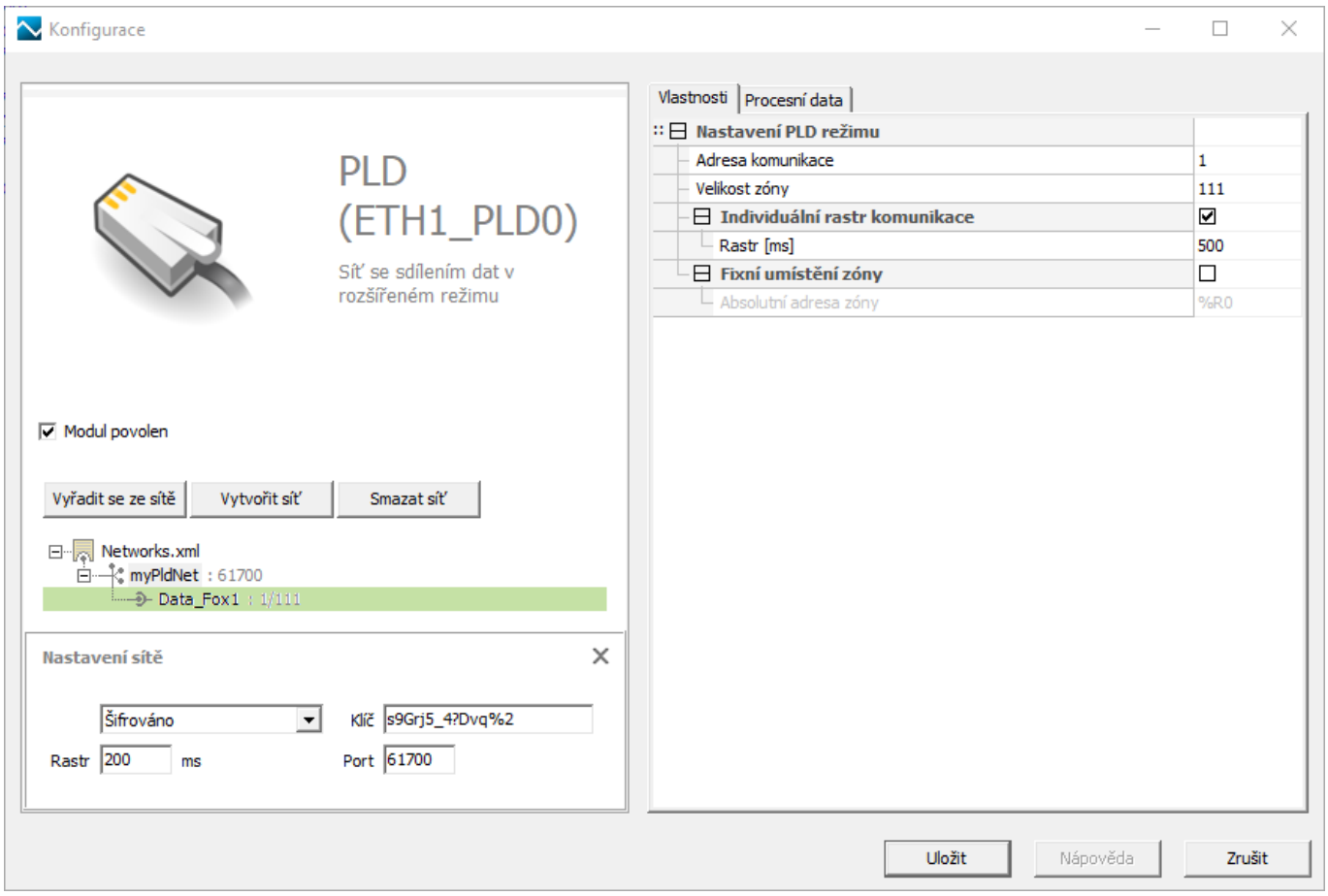
Pokud nastavíme adresu komunikace 1, velikost zóny na 111 bytů a individuální rastr komunikace na 500 ms, tak bude dialog vypadat následovně:



*Obr.3.11 Příklad nastavení vlastností sdílených dat sítě v režimu **PLD***

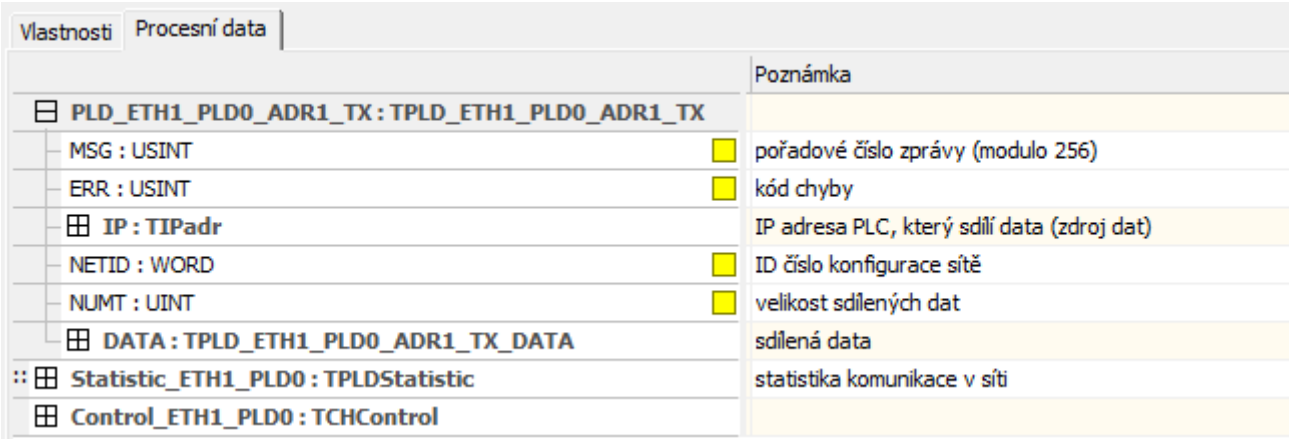
Posledním krokem je zařazení sdílených dat do sítě, což se provede kliknutím na název sítě (*myPldNet*) a stiskem tlačítka *Zařadit se do sítě*. Poté je třeba zadat název proměnné, ve které budou uložena sdílená data ve všech systémech v síti (např *Data_Fox1*). Konfigurace se dokončí stiskem tlačítka *Uložit*.

Síť *myPldNet* má v tomto okamžiku jednoho účastníka s logickou adresou 1, který sdílí každých 500 ms data v proměnné *Data_Fox1* (velikost 111 bytů). Data jsou v komunikaci zašifrována klíčem *s9Grj5_4?Dvq%2*, komunikace bude probíhat broadcast komunikacemi UDP protokolem na port 61700 (obr.3.12).



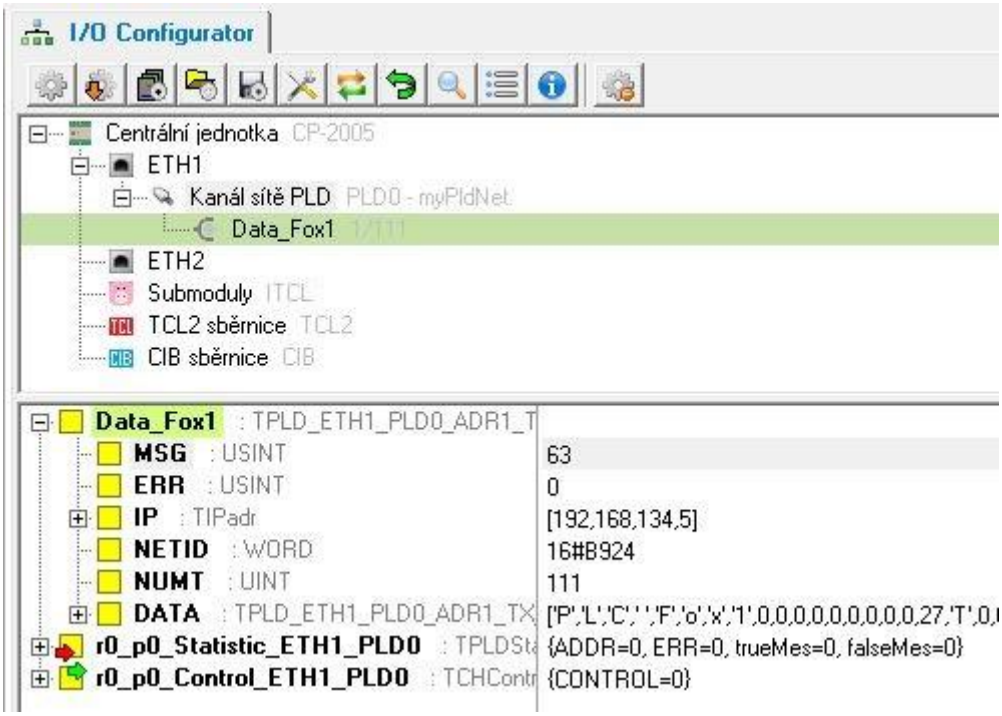
Obr.3.12 Dialog zařazení do sítě

Zeleným pruhem je v dialogu zvýrazněn systém z aktuálně otevřeného projektu. Struktura jeho procesních dat v síti **PLD** bude následující:



*Obr.3.13 Struktura procesních dat v režimu **PLD***

V okně *I/O Configurator* budou data zobrazena následovně:



Obr.3.14 Zobrazení procesních dat v okně I/O Configurator

Význam jednotlivých položek je následující:

MSG - čítač zpráv
Počet odeslaných zpráv. Každá odeslaná zpráva zvýší tento čítač o 1.

ERR - kód chyby při komunikaci
0 - bezchybný stav

IP - IP adresa zdroje sdílených dat

NETID - identifikační číslo, které charakterizuje konfiguraci celé sítě
Identifikační číslo se přenáší společně se sdílenými daty. Při změně jakéhokoliv parametru sítě se toto číslo změní. Umožňuje tak zjistit, jestli všechny PLC systémy v síti používají stejnou konfiguraci sítě.

NUMT - počet bytů vysílaných dat
Udává velikost proměnné DATA v bytech.

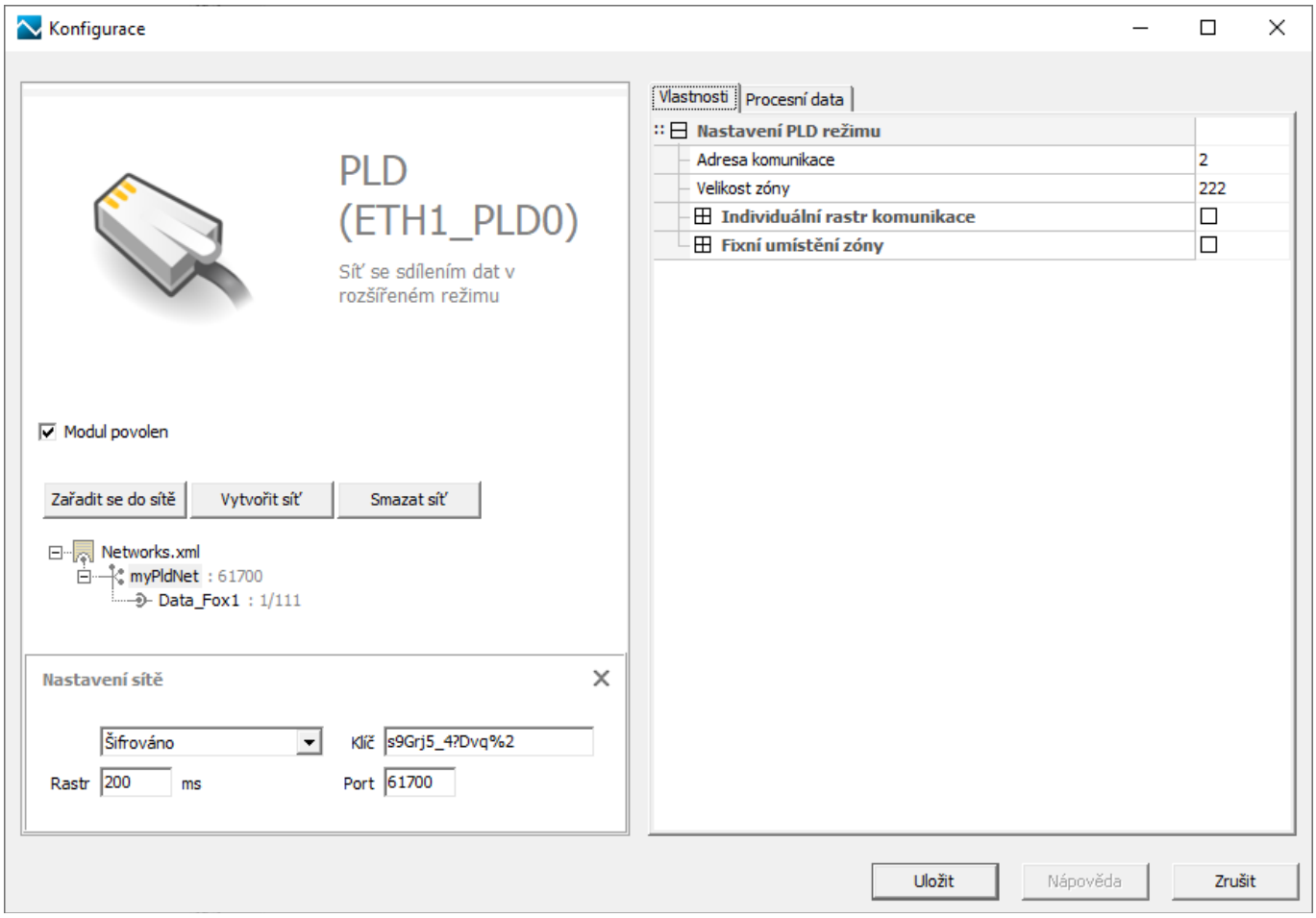
DATA[x] - sdílená data

3.3.2.2. Přidání dalšího účastníka do sítě v režimu PLD

Předpokládejme, že máme založený projekt pro Foxtrot CP-2005 ve kterém jsme založili síť se sdílením dat v režimu **PLD** (viz kap.3.3.2.1.). Nyní potřebujeme přidat další PLC systém, např. CP-2080, který by v rámci sdílení dat dostával data ze zmíněného CP-2005 a navíc mu nějaká data také posílal.

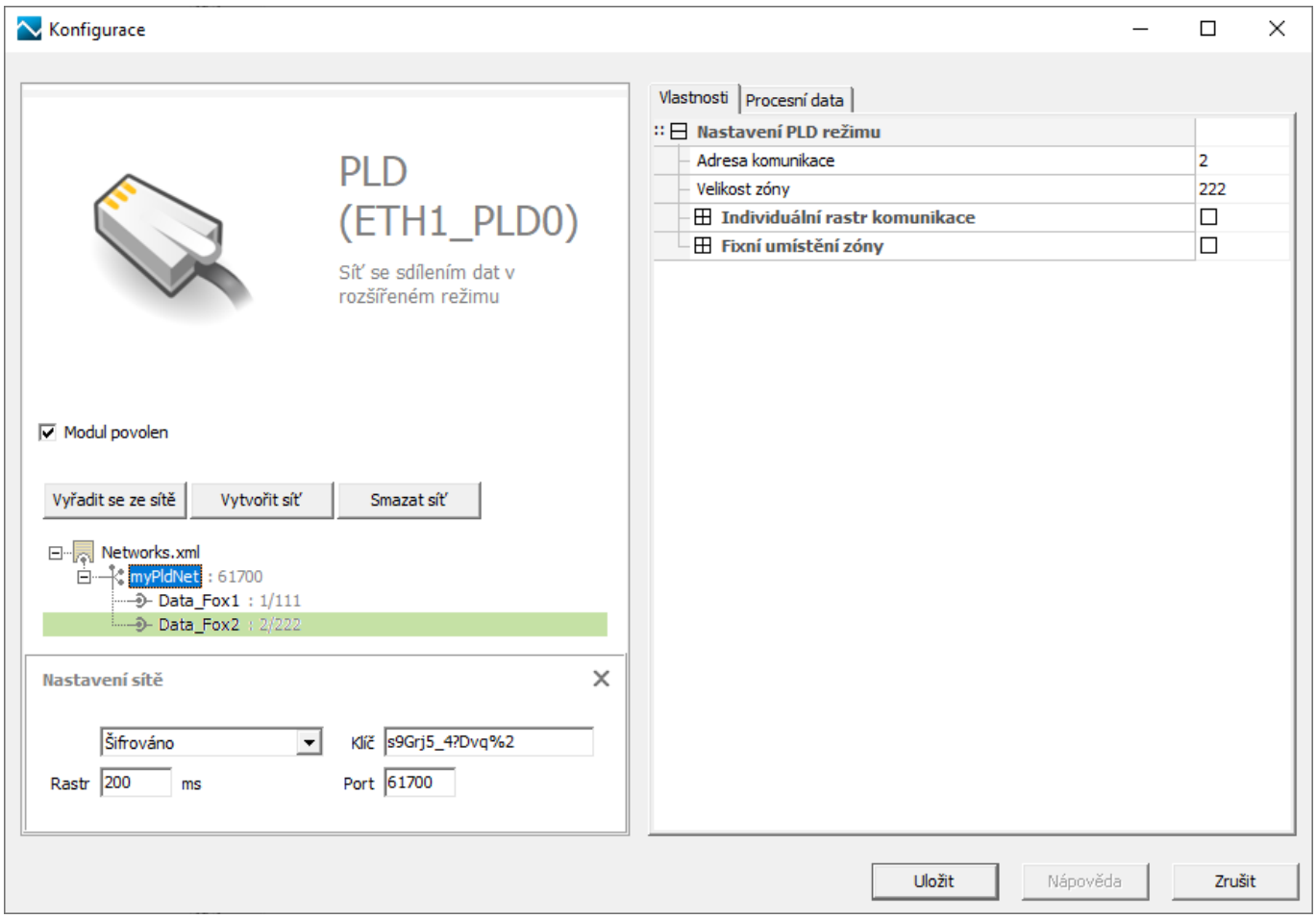
Projekt pro další FOXTROT CP-2080 založíme ve stejné projektové skupině jako je předchozí CP-2005. Nejprve je třeba zvolit rozhraní, přes které budou PLC systémy sdílet data. Zvolené rozhraní systému CP-2080 musí mít IP adresu ve stejném segmentu sítě jako rozhraní s režimem **PLD** u systému CP-2005.

Dvojitým kliknutím na zvolené rozhraní v okně nástroje *I/O Configurator* (např. ETH1) zobrazíme dialog volby režimu kanálu a zvolíme režim **PLD**. Zobrazí se okno konfigurace a v něm se objeví již nakonfigurovaná síť *myPldNet* z předchozí kap.3.3.2.1. V tomto dialogu nastavíme na záložce *Vlastnosti* adresu komunikace na 2 a velikost zóny sdílených dat na 222.



Obr.3.15 Dialog nastavení vlastností sdílených dat pro další systém

Kliknutím na název sítě *myPldNet* označíme síť, do které chceme nově přidávaný systém zařadit. A poté klikneme na tlačítko *Zařadit se do sítě* a vyplníme název proměnné, ve které budou uložena sdílená data právě přidávaného PLC systému (např *Data_Fox2*). Následující obrázek ukazuje, jak bude dialogové okno v tomto okamžiku vypadat.



*Obr.3.16 Dialog přidání dalšího systému do sítě v režimu **PLD***

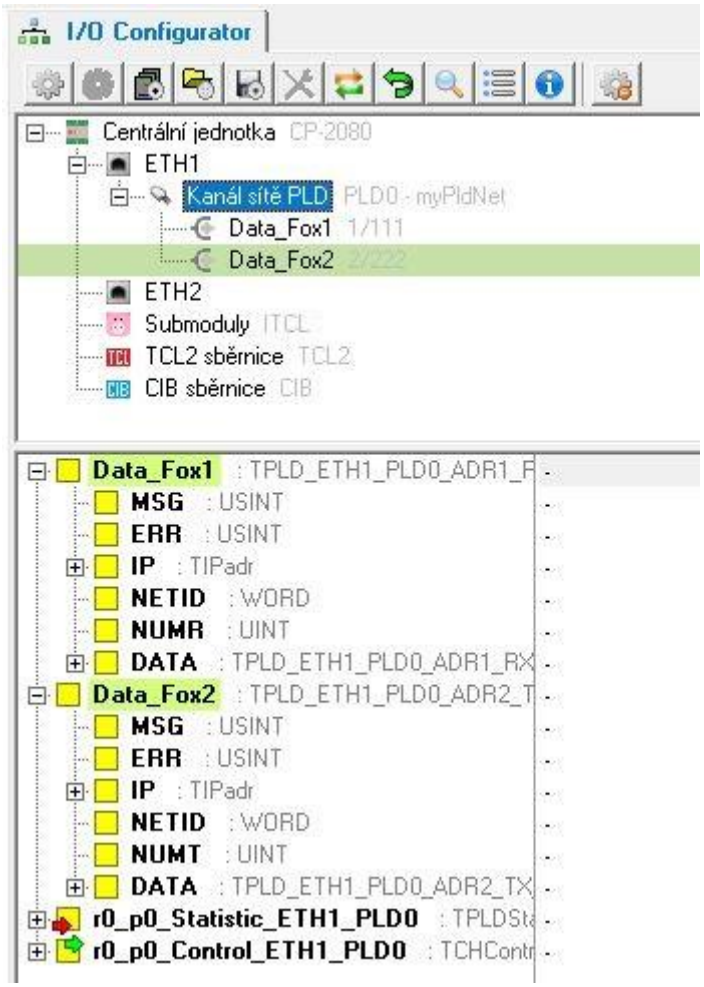
Zeleným pruhem je v dialogu zvýrazněn systém z aktuálně otevřeného projektu. Struktura procesních dat v síti v režimu **PLD** bude z pohledu systému CP-2080 následující:

Vlastnosti | Procesní data

	Alias	{PUBLIC}	{PUBLIC_API}	Poznámka
:: PLD_ETH1_PLD0_ADR1_RX : TPLD_ETI	Data_Fox1	☐	☐	Data_Fox1
MSG : USINT				pořadové číslo zprávy (modulo 256)
ERR : USINT				kód chyby při komunikaci
IP : TIPadr				IP adresa PLC, který sdílí data (IP zdroje dat)
NETID : WORD				ID číslo konfigurace sítě
NUMR : UINT				počet přijatých bytů
DATA : TPLD_ETH1_PLD0_ADR1_R)				přijatá data
PLD_ETH1_PLD0_ADR2_TX : TPLD_ETI	Data_Fox2	☐	☐	Data_Fox2
MSG : USINT				pořadové číslo zprávy (modulo 256)
ERR : USINT				kód chyby při komunikaci
IP : TIPadr				IP adresa PLC, který sdílí data (IP zdroje dat)
NETID : WORD				ID číslo konfigurace sítě
NUMT : UINT				počet vysílaných bytů
DATA : TPLD_ETH1_PLD0_ADR2_T)				vysílaná data
Statistic_ETH1_PLD0 : TPLDStatistic				statistika komunikace v PLD síti
Control_ETH1_PLD0 : TCHControl				

Obr.3.17 Struktura procesních dat z pohledu dalšího systému

Po uzavření okna *Konfigurace* bude okno *I/O Configurator* vypadat následovně:



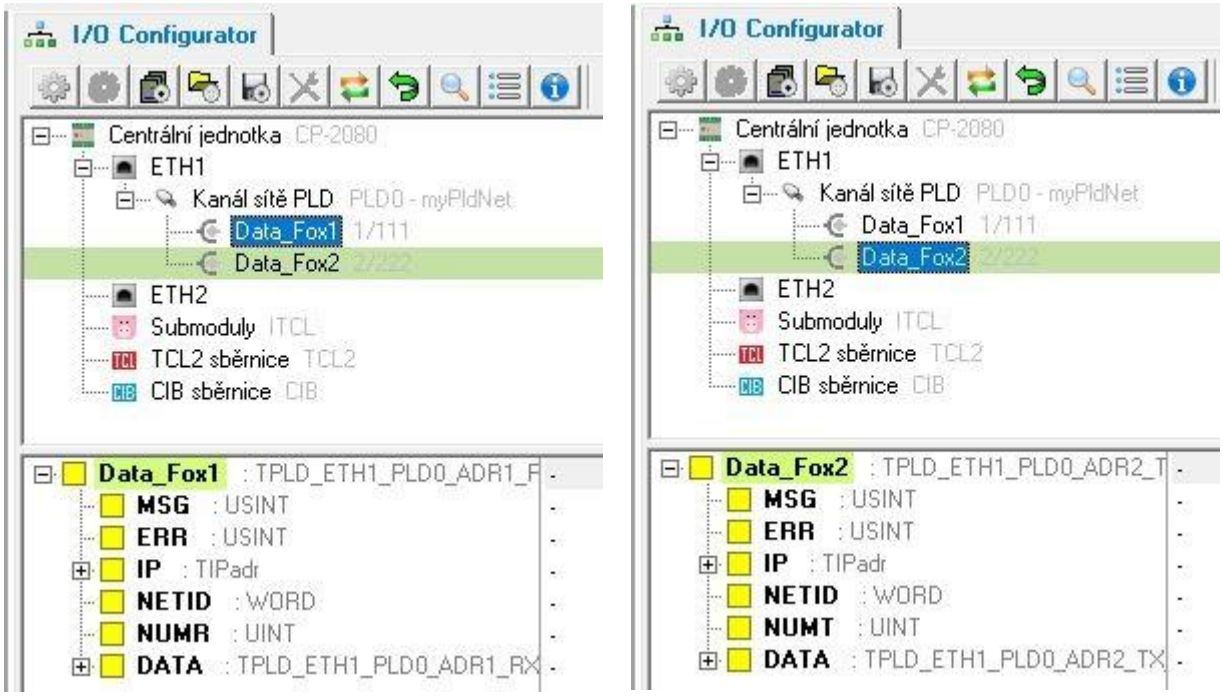
*Obr.3.18 Náhled na celou síť v režimu **PLD** v okně I/O Configurator*

3.3.2.3. Provoz sítě v režimu PLD

Obsluha sítě probíhá na pozadí uživatelského programu asynchronně s dobou cyklu uživatelského programu. Veškerá aktuální data sítě jsou uložena ve vnitřní paměti systému odkud jsou předávána do paměti proměnných vždy v otočce cyklu uživatelského programu.

Výměna dat v síti PLD

Struktura dat sdílených jednotlivými PLC systémy v režimu **PLD** je zobrazena na obr.3.19.



*Obr.3.19 Náhledy na jednotlivé účastníky v síti v režimu **PLD** v okně I/O Configurator*

Procesní data obsahují položky s následujícím významem:

MSG - čítač zpráv
Každá odeslaná zpráva zvýší tento čítač o 1. Hodnota se přenáší společně se sdílenými daty. Umožňuje tak kontrolovat, jestli se data cyklicky obnovují (čítač se musí měnit) a jestli se neztrácí nějaké pakety (změna čítače o hodnotu větší než 1 znamená ztrátu jednoho nebo více paketů)

ERR - kód chyby při komunikaci
Vyhodnocuje se při příjmu.
0 - bezchybný stav
80 - timeout při příjmu dat (od příjmu posledního paketu od daného PLC systému uplynula doba větší než je pětinásobek jeho vysílacího rastru)

IP - IP adresa zdroje sdílených dat
Získává se z UDP protokolu při příjmu dat a umožňuje kontrolu zdroje dat.

NETID - identifikační číslo, které charakterizuje konfiguraci celé sítě
Identifikační číslo se přenáší společně se sdílenými daty. Při změně jakéhokoliv parametru sítě se toto číslo změní. Umožňuje tak zjistit, jestli všechny PLC systémy v síti používají stejnou konfiguraci sítě.

NUMT - počet bytů vysílaných dat
Udává velikost proměnné DATA v bytech a přenáší se společně se sdílenými daty.

NUMR - počet bytů přijímaných dat
Udává velikost proměnné DATA v bytech a získává se z přijatého paketu, který obsahuje sdílená data.

DATA[x] - sdílená data

Proměnné, které chce PLC systém zveřejnit pro ostatní systémy, je třeba zapsat v PLC programu do položky *DATA*. Ostatní položky v procesních datech jsou automaticky vyplněny PLC systémem při vysílání resp. při příjmu dat.

V našem případě (viz předchozí kapitoly) máme v síti dva systémy:

Fox1

- logická adresa 1
- zveřejňuje strukturu *Data_Fox1*
- velikost sdílených dat *Data_Fox1.DATA* je 111 bytů

Fox2

- logická adresa 2
- zveřejňuje strukturu *Data_Fox2*
- velikost sdílených dat *Data_Fox2.DATA* je 222 bytů

V obou případech jsou položky *DATA* definovány jako pole prvků typu USINT. Do těchto polí je třeba zapsat informace, které budeme sdílet s ostatními PLC systémy v síti. Ve většině případů narazíme na to, že pole prvků USINT není úplně vhodné pro to, co potřebujeme sdílet. Předpokládejme, že z každého PLC systému chceme pro začátek zveřejnit jeho název (popis), kód typu PLC a čítač cyklů. Sdílené informace popíšeme např. následujícím datovým typem:

```
TYPE
  T_Shared_Data_Fox1 : STRUCT
    name     : STRING[16];
    cpuCode  : UINT;
    cycleCnt : UDINT;
  END_STRUCT

  T_Shared_Data_Fox2 : STRUCT
    name     : STRING[16];
    cpuCode  : UINT;
    cycleCnt : UDINT;
  END_STRUCT
END_TYPE
```

Uvedenou deklaraci datových typů sdílených dat je vhodné zapsat do samostatného souboru, který bude společný pro oba projekty PLC v síti. Pokud bude někdy později nezbytné udělat nějakou změnu struktury sdílených dat, tak stačí změnit společný soubor s datovými typy a pak přeložit dotčené projekty a nahrát je do PLC. V každém případě je výhodné pouze přidávat nové položky na konec již definovaných struktur a jinak strukturu neměnit. Podmínkou pro to je, aby byla ve sdílených zónách nějaká rezerva (velikost zón nastavená v konfiguračním dialogu musí být větší než aktuální velikost typu definovaného pro sdílení dat). Pokud pouze přidáme nějaké položky na konec struktury sdílených dat, pak lze přehrávat programy v jednotlivých PLC postupně bez nebezpečí chybné interpretace přijatých dat. Naopak pokud je nezbytné změnit zásadním způsobem strukturu sdílených dat nebo velikost sdílených zón, pak je třeba nejprve zastavit všechna PLC v síti (nastavit jim režim HALT) a poté postupně nahrát nové programy a spustit PLC do režimu RUN.

Dále vytvoříme globální proměnné s použitím typů *T_Shared_Data_Fox?*, které namapujeme na proměnné *Data_Fox1.DATA* resp *Data_Fox2.DATA*. To se zařídí následující deklarací:

```
VAR_GLOBAL
  myData  AT Data_Fox1.DATA : T_Shared_Data_Fox1;
  hisData AT Data_Fox2.DATA : T_Shared_Data_Fox2;
END_VAR
```

Tím jsme získali přístup k proměnné *Data_Fox1.DATA* přes strukturu *T_Shared_Data_Fox1.* Zápis do sdílených dat *Data_Fox1.DATA* v systému Foxtrot1 bude vypadat následovně:

```
myData.name     := 'PLC Fox1';
myData.cpuCode  := CPU_CODE;
myData.cycleCnt := myData.cycleCnt + 1;
```

Otestování dat sousedního systému, která jsou sdílená v proměnné *Data_Fox2.DATA* bude vypadat podobně:

```
IF hisData.cpuCode = 2005 THEN
  // data processing from other PLCs
END_IF;
```

Položky *MSG*, *ERR*, *NETID* a *NUMR* v proměnné *Data_Fox2* umožňují zkontrolovat platnost přijatých dat. Před zpracováním dat přijatých od ostatních systémů lze zkontrolovat následující:

MSG - změna této položky znamená nově přijatá data
Pokud se nová hodnota liší od předcházející hodnoty více jak o 1, znamená to, že došlo ke ztrátě jednoho nebo více paketů od daného PLC.
ERR - kód chyby při komunikaci musí být 0
NETID - tyto položky se musí shodovat pro všechny PLC v síti
NUMR - velikost sdílených dat musí odpovídat velikosti proměnné *Data_Fox2.DATA*

Pro PLC Fox1 může celý program vypadat např. následovně:

```
VAR_GLOBAL
  myData  AT Data_Fox1.DATA : T_Shared_Data_Fox1;
  hisData AT Data_Fox2.DATA : T_Shared_Data_Fox2;
END_VAR

PROGRAM prgDataSharingFox1
  VAR
    init     : BOOL;
    start    : BOOL;
    errorPLD : STRING;
    hisName  : STRING;
    oldMSG2  : USINT;
    recvTim2 : TON;
    misMSG2  : UDINT;
  END_VAR

  errorPLD := '';

  // PLC Fox1 publikuje data pro ostatní PLC
  IF sizeof( T_Shared_Data_Fox1) <= sizeof( Data_Fox1.DATA) THEN
    IF NOT init THEN
      myData.name := 'PLC Fox1';
      init := TRUE;
    END_IF;
    myData.cpuCode  := CPU_CODE;
    myData.cycleCnt := myData.cycleCnt + 1;
  ELSE
    errorPLD := 'Wrong size of T_Shared_Data_Fox1!';
  END_IF;

  // kontrola site PLD
  IF Data_Fox2.ERR <> 0 THEN
    // chyba pri prijmu dat od PLC Fox2
    errorPLD := 'Receiving error: ' + USINT_TO_STRING(Data_Fox2.ERR);
    IF Data_Fox2.ERR = 80 THEN
      errorPLD := errorPLD + ' (timeout)';
    END_IF;

  ELSIF Data_Fox1.NETID <> Data_Fox2.NETID THEN
    // NETID musi byt stejne u vsech ucastniku site
    errorPLD := 'Some NETID is wrong!';

  ELSIF sizeof( T_Shared_Data_Fox2) > sizeof( Data_Fox2.DATA) THEN
    // struktura T_Shared_Data_Fox2 se musi vejit do Data_Fox2.DATA
    errorPLD := 'Wrong size of T_Shared_Data_Fox2!';
```

```
  ELSIF Data_Fox2.MSG <> oldMSG2 THEN
    IF (Data_Fox2.MSG - oldMSG2) > 1 THEN
      IF start THEN
        misMSG2 := misMSG2 + USINT_TO_UDINT(Data_Fox2.MSG - oldMSG2);
      ELSE
        start := TRUE;
      END_IF;
    END_IF;

  ELSIF recvTim2.Q THEN
    errorPLD := 'Shared data are not refreshed for a long time!';

  ELSIF Data_Fox2.NUMR <> sizeof(Data_Fox2.DATA) THEN
    errorPLD := 'Wrong value of Data_Fox2.NUMR';
  END_IF;

  recvTim2(IN := Data_Fox2.MSG = oldMSG2, PT := T#5s);
  oldMSG2 := Data_Fox2.MSG;

  // zpracovani dat od ostatnich PLC
  IF errorPLD = '' THEN
    hisName := hisData.name;
    //
    // ...
    //
  END_IF;
END_PROGRAM
```

Podobně může vypadat i program pro PLC Fox2:

```
VAR_GLOBAL
  myData  AT Data_Fox2.DATA : T_Shared_Data_Fox2;
  hisData AT Data_Fox1.DATA : T_Shared_Data_Fox1;
END_VAR

PROGRAM prgDataSharingFox2
  VAR
    init     : BOOL;
    start    : BOOL;
    errorPLD : STRING;
    hisName  : STRING;
    oldMSG1  : USINT;
    recvTim1 : TON;
    misMSG1  : UDINT;
  END_VAR

  errorPLD := '';

  // PLC Fox2 publikuje data pro ostatní PLC
  IF sizeof( T_Shared_Data_Fox2) <= sizeof( Data_Fox2.DATA) THEN
    IF NOT init THEN
      myData.name := 'PLC Fox2';
      init := TRUE;
    END_IF;
    myData.cpuCode  := CPU_CODE;
    myData.cycleCnt := myData.cycleCnt + 1;
  ELSE
    errorPLD := 'Wrong size of T_Shared_Data_Fox2!';
  END_IF;

  // kontrola site PLD
  IF Data_Fox1.ERR <> 0 THEN
    // chyba pri prijmu dat od PLC Fox1
```

```
    errorPLD := 'Receiving error: ' + USINT_TO_STRING(Data_Fox1.ERR);
    IF Data_Fox1.ERR = 80 THEN
      errorPLD := errorPLD + ' (timeout)';
    END_IF;

  ELSIF Data_Fox1.NETID <> Data_Fox2.NETID THEN
    // NETID musi byt stejne u vsech ucastniku site
    errorPLD := 'Some NETID is wrong!';

  ELSIF sizeof( T_Shared_Data_Fox1) > sizeof( Data_Fox1.DATA) THEN
    // struktura T_Shared_Data_Fox1 se musi vejit do Data_Fox1.DATA
    errorPLD := 'Wrong size of T_Shared_Data_Fox1!';

  ELSIF Data_Fox1.MSG <> oldMSG1 THEN
    IF (Data_Fox1.MSG - oldMSG1) > 1 THEN
      IF start THEN
        misMSG1 := misMSG1 + USINT_TO_UDINT(Data_Fox1.MSG - oldMSG1);
      ELSE
        start := TRUE;
      END_IF;
    END_IF;

  ELSIF recvTim1.Q THEN
    errorPLD := 'Shared data are not refreshed for a long time!';
    RETURN;
  END_IF;

  recvTim1(IN := Data_Fox1.MSG = oldMSG1, PT := T#5s);
  oldMSG1 :=Data_Fox1.MSG;

  // zpracovani dat od ostatnich PLC
  IF errorPLD = '' THEN
    hisName := hisData.name;
    //
    // ...
    //
  END_IF;
END_PROGRAM
```

Diagnostika provozu sítě PLD

Součástí procesních dat v každé sítě v režimu **PLD** je i struktura s diagnostickými informacemi o síti.

Vlastnosti | Procesní data

	Alias	{PUBLIC}	{PUBLIC_API}	Poznámka
PLD_ETH1_PLD0_ADR1_TX : TPLD_	Data_Fox1	☐	☐	Data_Fox1
PLD_ETH1_PLD0_ADR2_RX : TPLD_	Data_Fox2	☐	☐	Data_Fox2
Statistic_ETH1_PLD0 : TPLDStatist				statistika komunikace v síti
ADDR : USINT				adresa PLC uvedená v posledním chybném paketu
ERR : USINT				kód poslední vzniklé chyby
trueMes : UDINT				počet úspěšně přijatých zpráv
falseMes : UDINT				počet zpráv při jejichž zpracování došlo k chybě
Control_ETH1_PLD0 : TCHControl				rezerva

*Obr.3.20 Struktura diagnostických dat sítě v režimu **PLD***

Význam jednotlivých položek je následující:

ADDR - adresa stanice, od které přišla poslední chybná zpráva (kód chyby viz položka ERR)

ERR - kód poslední vzniklé chyby
Vyhodnocuje se při příjmu.
0 - komunikace bez chyb
84 - chybný rámec příchozí zprávy
85 - neznámý režim výměny dat
86 - chybná IP adresa nebo maska stanice
87 - časové razítko příchozí zprávy je neplatné
88 - chybná délka příchozího paketu

trueMes - počet platných komunikací sítě

falseMes - počet chybných komunikací v síti

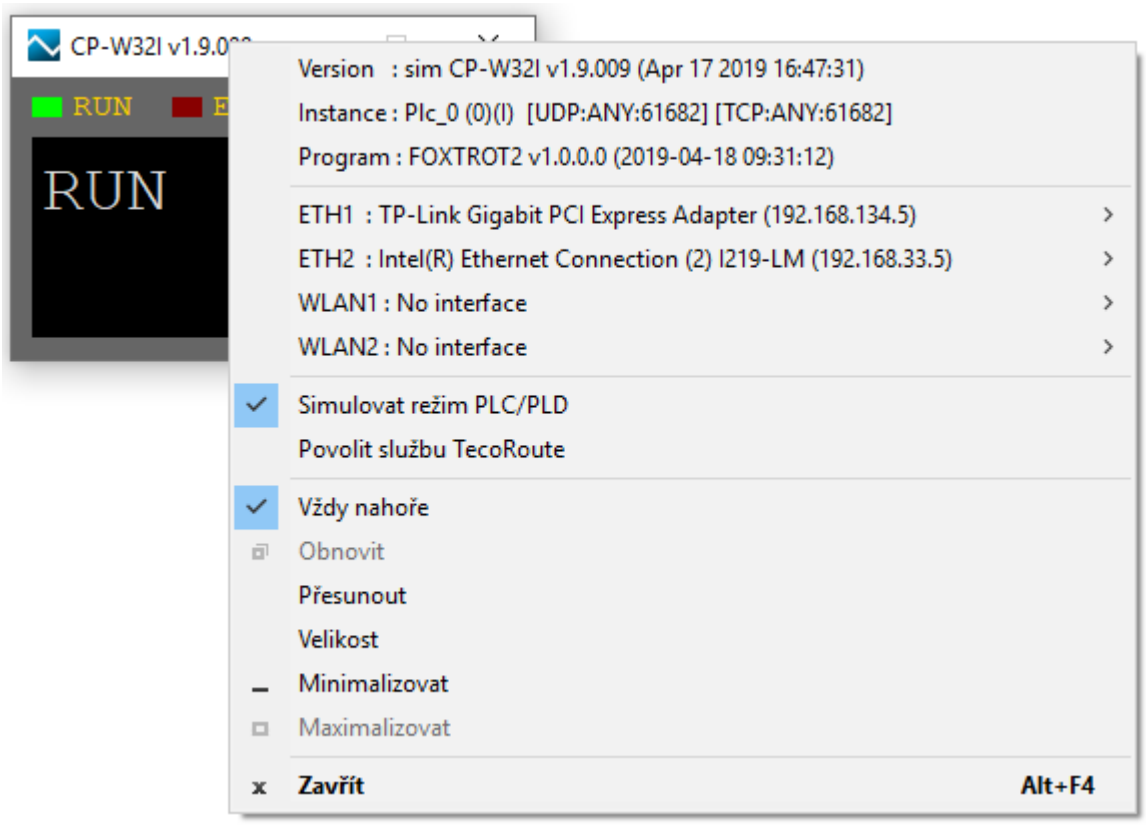
3.3.2.4. Síť v režimu PLD v simulátoru PLC

V simulátoru PLC FOXTROT 2 (v prostředí Mosaic) lze také simulovat sdílení dat v sítích s režimem PLD. Tato vlastnost je ve výchozím stavu simulátoru vypnutá. Pokud tedy chceme při ladění použít místo reálného PLC jeho simulátor v prostředí Mosaic, je nejprve potřeba v simulátoru aktivovat vlastnost *Simulovat režim PLC/PLD* a zároveň je nezbytné přiřadit síťová rozhraní počítače k rozhraním simulátoru PLC. K tomu slouží lokální nabídka okna simulátoru, která se zobrazí po kliknutí na ikonu v levém horním rohu okna simulátoru.



Obr.3.21 Okno simulátoru PLC v prostředí Mosaic

Lokální nabídka simulátoru PLC vypadá následovně:



Obr.3.22 Lokální nabídka simulátoru PLC v prostředí Mosaic

Kliknutím na řádek s názvem síťového rozhraní (ETH1, ETH2, WLAN1, WLAN2) je možné přiřadit k danému rozhraní požadovaný adaptér počítače. IP adresa takto zvoleného adaptéru musí být ve stejném segmentu sítě jako jsou adaptéry PLC systémů v síti. Kliknutím na řádek *Simulovat režim PLC/PLD* se zapne resp. vypne simulace režimu **PLD**. Zapnutá simulace je signalizována zobrazením zatržítka v lokální nabídce. Simulaci má smysl zapnout pouze v případě, že v síti chybí příslušný PLC systém. Jinak budou stejná data sdílená jak reálným PLC systémem tak simulátorem současně, což je nežádoucí situace, která může vést k problémům při zpracování takto sdílených dat v okolních systémech.

4. KOMUNIKAČNÍ REŽIMY ROZHRANÍ USB

Přes rozhraní USB lze komunikovat v režimu **PC**. Režim **PC** je trvale aktivní a umožňuje i programování PLC, tzn. podporuje i systémové služby (pokud je v daném okamžiku nevyužívá jiný komunikační kanál).

Logická adresa stanice (adresa v protokolu EPSNET) je 0.

V následujících kapitolách jsou popsány jednotlivé komunikační režimy použitelné na rozhraní USB (tab.4.1).

Tab.4.1 Přehled komunikačních režimů rozhraní USB

Komunikační režim	Funkce	Popis
PC	komunikace s nadřízeným systémem	kap.4.1.

4.1. REŽIM PC - KOMUNIKACE S NADŘÍZENÝM SYSTÉMEM

Režim **PC** slouží k připojení nadřízeného systému k PLC.

Komunikace jsou vyvolávány nadřízeným systémem na principu dotaz - odpověď. PLC TECOMAT v režimu **PC** se chová jako pasivní podřízený účastník (server).

Veřejné a systémové služby sítě EPSNET

V režimu **PC** je na všech komunikačních kanálech dostupný kompletní soubor služeb sítě EPSNET, jehož součástí jsou jak systémové služby, určené k programování a ladění systému, tak i veřejné služby, určené k výměně dat. Omezující podmínkou je, že systémové služby podporuje v daném časovém okamžiku pouze jeden komunikační kanál.

Síť EPSNET a její veřejné služby jsou popsány v kap.5.

Upozornění: Vývojové prostředí Mosaic využívá systémové služby sítě EPSNET a z tohoto důvodu je lze provozovat pouze přes jeden komunikační kanál v režimu **PC**. Oprávnění využívat systémové služby dostane ten komunikační kanál, který přijde s požadavkem jako první. Oprávnění zaniká 5 sekund po ukončení komunikace.

5. PROTOKOL EPSNET

Komunikace jsou vyvolávány nadřízenou stanicí na principu dotaz - odpověď. Nadřízenou stanicí je zpravidla počítač PC s vizualizačním nebo dispečerským softwarem vybavený ovladačem pro komunikaci se systémy TECOMAT. Podřízenými stanicemi jsou systémy TECOMAT s komunikačním kanálem v režimu **PC**.

5.1. OBECNÁ STRUKTURA PROTOKOLU EPSNET

Komunikace ve směru nadřízená stanice → podřízená stanice (zpráva, dotaz)

a) zpráva bez datového pole

SD1	DA	SA	FC	FCS	ED

b) zpráva s datovým polem

SD2	LE	LER	SD2R	DA	SA	FC	DATA.....	FCS	ED

c) zpráva token - předání řízení sítě (bez odpovědi, odpovědí je převzetí řízení)

SD4	DA	SA

Komunikace ve směru podřízená stanice → nadřízená stanice (potvrzení, odpověď)

a) odpověď bez datového pole
- krátké pozitivní potvrzení

SAC

- jiný typ potvrzení

SD1	DA	SA	FC	FCS	ED

b) odpověď s datovým polem

SD2	LE	LER	SD2R	DA	SA	FC	DATA.....	FCS	ED

SD1 - úvodní znak 1 (start delimiter 1)
- pevná hodnota $10

SD2 - úvodní znak 2 (start delimiter 2)
- pevná hodnota $68

SD4 - úvodní znak 4 (start delimiter 4)
- pevná hodnota $DC

LE - délka dat (length)
- počet bytů položek DA+SA+FC+DATA, tj. 3 ... 249

LER - opakovaná délka dat (length repeat)
- stejná hodnota jako LE

SD2R - opakovaný úvodní znak 2 (start delimiter 2 repeat)
- pevná hodnota $68

DA - cílová adresa (destination address)
- hodnota pro podřízenou stanici 0 ... 126 (dotaz)
- hodnota pro nadřízenou stanici 0 ... 126 (odpověď)

SA - zdrojová adresa (source address)
- hodnota pro nadřízenou stanici 0 ... 126 (dotaz)
- hodnota pro podřízenou stanici 0 ... 126 (odpověď)

FC - řídící byte rámce (frame control byte)
- konkrétní hodnoty jsou uvedeny u jednotlivých zpráv (kap.5.7.)

DATA - vlastní datové tělo zprávy
- maximálně 246 bytů

FCS - kontrolní součet (frame check sum)
- bytový součet všech bytů položek DA, SA, FC a DATA se zanedbáním vyšších řádů vzniklých přenosem

pro SD1 $\sum_{DA}^{FC} \bmod 256$

pro SD2 $\sum_{DA}^{FCS-1} \bmod 256$

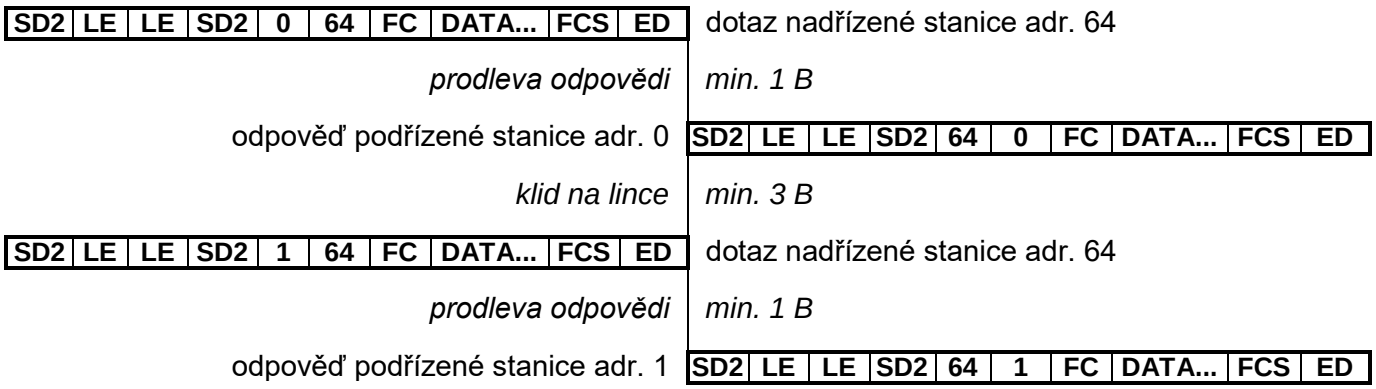
ED - koncový znak (end delimiter)
- pevná hodnota $16

SAC - krátké potvrzení (short acknowledge)
- pevná hodnota $E5

5.2. PODMÍNKY KOMUNIKACE

Pro komunikaci na sériových linkách a rozhraní USB platí následující podmínky:

- mezi jednotlivými byty vysílanými z nadřízené stanice do stanice podřízené musí být **kratší** prodleva (klid na lince) než trojnásobek doby potřebné pro vyslání jednoho bytu
- mezi přijatou odpovědí a vyslanou další zprávou musí být klid na lince **delší** než trojnásobek doby potřebné pro vyslání jednoho bytu (obr.5.1), tímto klidem na lince se synchronizují přijímače všech stanic před zahájením nového vysílání



Obr.5.1. Časové poměry na lince

Dále je třeba mít na paměti následující zásady:

- na každou zprávu PLC odpoví, pokud ne, jde o závažnou chybu komunikace (chybně sestavená zpráva, neprůchodná linka, nedodržení výše uvedených podmínek)
- mezi posledním bytem vyslané zprávy a prvním bytem přijaté odpovědi je prodleva minimálně stejná jako doba potřebná pro vyslání jednoho bytu (viz výše), maximální dobu (tzv. timeout) si stanoví programátor nadřízené stanice s ohledem na přenosovou rychlost, dopravní cestu (např. při použití modemů nebo opakovačů dochází ke zvýšení dopravního

zpoždění komunikace) a doby cyklu (čtení a zápis dat jsou prováděny pouze v otočce cyklu, aby nedošlo k časovému hazardu dat během vykonávání uživatelského programu)
- adresa nadřízené stanice musí být odlišná od adres ostatních stanic

Zprávy protokolu EPSNET mají pevný formát 1 start bit, 8 datových bitů, sudá parita, 1 stop bit.

Data ve zprávě jsou zabezpečena sudou paritou, kontrolním součtem FCS a správnou sekvencí hodnot rámce protokolu. Jakákoli chyba zjištěná těmito prostředky má za následek prohlášení zprávy za chybnou a její zahození. Tím je zajištěno, že data, která jsou propuštěna do systému, jsou správná.

5.3. ODPOVĚDI A CHYBOVÁ HLÁŠENÍ

Pozitivní odpověď

Jak už bylo uvedeno, na každý dotaz přijde odpověď (pokud proběhne bez závad vlastní přenos zprávy). Ta může být buď pozitivní nebo negativní. Pozitivní odpověď znamená správné provedení služby a platnost přenášených dat. Liší se podle konkrétní služby.

SAC

SAC = $E5 - krátké pozitivní potvrzení bez dat

SD1	DA	SA	FC	FCS	ED

FC = 0 - pozitivní potvrzení bez dat

SD2	LE	LER	SD2R	DA	SA	FC	DATA.....	FCS	ED

FC = 8 - pozitivní potvrzení, zpráva nese platná data

Negativní odpověď

Negativní odpověď může být několika druhů. Buď centrální jednotka tuto komunikační službu nezná (FC = 2), nebo tato komunikační služba není v tomto okamžiku aktivována (FC = 3), nebo je tato komunikační služba blokována heslem (FC = 4), nebo nejsou požadovaná data ještě k dispozici, ale lze očekávat, že v dohledné době budou (FC = 9), nebo komunikační služba má chybné parametry, které nelze správně interpretovat (FC = $0C).

SD1	DA	SA	FC	FCS	ED

FC = 2 - neznámá komunikační služba
FC = 3 - komunikační služba není v tomto okamžiku aktivována
FC = 4 - komunikační služba blokovaná heslem
FC = 9 - data zatím nedostupná

SD2	LE	LER	SD2R	DA	SA	FC	ER1 ER2	FCS	ED

LE = 5
LER = 5
FC = $0C (12) - chybné parametry komunikační služby
ER1 = $30 (48) - chyba při zpracování zprávy
ER2 - specifikace chyby
$0B (11) - chybné parametry komunikační služby READN / READND
$0C (12) - v komunikační službě READN / READND se čte tabulka z neexistujícího uživatelského programu
$0D (13) - v komunikační službě READN / READND se čte neexistující tabulka z uživatelského programu

$0E (14) - bloky čtené v komunikační službě READN / READND se nevejdou do zprávy
$0F (15) - chybná adresa v komunikační službě WRITEN
$10 (16) - nulová délka bloku v komunikační službě WRITEN
$11 (17) - chybná adresa v komunikační službě READB
$12 (18) - chybná adresa v komunikační službě WRITEB
$13 (19) - chybné parametry komunikační služby WANDRN / WANDRND v části zápisu
$14 (20) - chybné parametry komunikační služby WANDRN / WANDRND v části čtení
$17 (23) - neznámá komunikační služba

Žádná odpověď

Pokud žádná odpověď nepřijde (vyloučíme-li problémy hardwaru jako přerušená linka, závada na napájení, apod., nebo chybně nastavenou přenosovou rychlost), znamená to, že vyslaná zpráva má chybně sestavený rámec (chybné délky LE, LER, chybná hodnota SD, SDR, ED, chybná adresa DA, chybný kontrolní součet FCS, apod.) nebo nebyly dodrženy zásady uvedené v kap.5.2. (příliš dlouhá prodleva mezi byty zprávy nebo příliš krátká prodleva mezi předchozí odpovědí a následující zprávou, případně chybný datový formát).

Chybová hlášení

Provozní chybová hlášení komunikačního kanálu se zobrazují v proměnné *Err* v diagnostické zóně nebo v přijímací zóně příslušné stanice (podle režimu komunikace). Přijatá zpráva, při které se vyskytla některá z těchto chyb, je ignorována a je zahrnuta do počtu chybných komunikací (proměnná *falseMes*). Příčinou chyb může být vysoká úroveň rušení, chybně zapojená komunikační linka, chybné nastavení parametrů nebo chyba na straně nadřízeného systému.

$10 chybný počáteční znak SD
Hodnota počátečního znaku zprávy neodpovídá definované hodnotě.

$11 chyba parity
Některý byte zprávy měl chybnou paritu.

$12 chyba opakované délky dat LER
Hodnota opakované délky dat LER je jiná než předchozí hodnota LE.

$13 chyba opakovaného počátečního znaku
Hodnota opakovaného počátečního znaku SDR není $68.

$14 chybná adresa zdroje SA
Odpověď pochází od jiné stanice, než která byla dotazována.

$15 chyba příznaku směru přenosu v řídicím bytu FC
Příznak směru přenosu v FC neodpovídá skutečnosti.

$17 chybná hodnota řídícího bytu FC
Hodnota řídícího bytu FC neodpovídá žádné funkci.

$18 chyba kontrolního součtu FCS
Přijatá hodnota kontrolního součtu neodpovídá spočtené hodnotě.

$19 chybný koncový znak ED
Hodnota koncového znaku není $16.

$20 chybná služba

Požadovaná služba není implementována.

$50 podřízená stanice neodpovídá

Nepřišla odpověď od podřízené stanice do uplynutí stanovené doby (0,5 s + dopravní zpoždění).

$54 podřízená stanice odpověděla chybně

Přišla chybná odpověď od podřízené stanice (chyba protokolu).

5.4. STRUKTURA PROTOKOLU EPSNET UDP / TCP

Na kanálech Ethernet se používá protokol EPSNET UDP, který, jak už název napovídá, je založen na protokolu UDP. Možné je použít též variantu EPSNET TCP založenou na protokolu TCP. Číslo vstupního portu na straně PLC TECOMAT je v režimu **PC** vždy 61682. Režim **PLC** používá vstupní port 61681.

Data v UDP paketu zasílaném do PLC začínají záhlavím délky 6 bytů a následuje vlastní zpráva, která má shodnou strukturu jako standardní protokol EPSNET (kap.5.1.). Navíc v jednom UDP paketu lze za jedním záhlavím poslat až 5 zpráv EPSNET a tím komunikaci zefektivnit (obr.5.2). PLC pak tyto vícenásobné zprávy zpracovává v tom pořadí, v jakém leží za sebou.

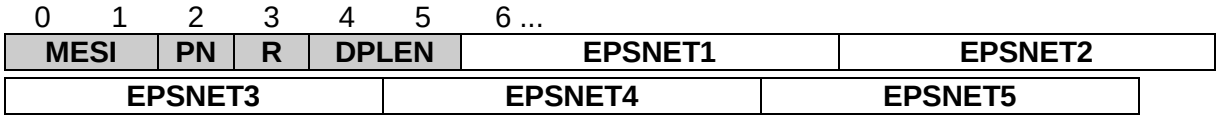
Odpověď od PLC má stejný formát. Za záhlavím délky 6 bytů následuje příslušný počet odpovědí v protokolu EPSNET.

Logická adresa stanice (adresa v protokolu EPSNET) je 0.

Struktura dat v TCP paketu v protokolu EPSNET TCP je shodná jako u varianty EPSNET UDP, proto budeme nadále popisovat obě varianty dohromady jako EPSNET UDP / TCP.

Struktura záhlaví

Každá zpráva protokolu EPSNET UDP / TCP (v libovolném směru) obsahuje na začátku datové zóny záhlaví, které tvoří 6 bytů (obr.5.2).



Obr.5.2 Struktura protokolu EPSNET UDP / TCP

Jejich význam je následující:

byte 0 a 1 **MESI** - číslo relace
Slouží k identifikaci jednotlivých zpráv. PLC v odpovědi vrací totéž číslo relace, jaké nesl dotaz vyslaný z nadřízeného systému. Tak lze jednoznačně určit, ke kterému dotazu patří přijatá odpověď.

byte 2 **PN** - kód režimu komunikace
Určuje, v jakém režimu se mají následující data zpracovat.
2 - režim **PC**
3 - režim **PLC**

byte 3 **R** - rezerva

byte 4 a 5 **DPLEN** - délka následujících dat
Určuje délku následujících dat (všechny zprávy EPSNET v tomto UDP / TCP paketu bez záhlaví). Byte 4 obsahuje vyšší byte délky, byte 5 obsahuje nižší byte délky (konvence big endian).

Poznámka: Celková délka dat v UDP / TCP paketu musí být povinně sudá. Pokud při komunikaci vychází lichá délka dat, pak je nutno na konec dat přidat jeden nulový byte a zvýšit celkovou délku dat v UDP / TCP paketu. Uvedená úprava délky je potřebná pouze pro vysílání UDP / TCP paketu, položka DPLEN v záhlaví se neupravuje (může být i lichá).

5.5. KOMUNIKAČNÍ SLUŽBY

Systémové a veřejné komunikační služby

Protokol EPSNET obsahuje dvě skupiny služeb pro komunikaci se systémy TECOMAT. Veřejné komunikační služby umožňují výměnu dat, systémové komunikační služby slouží k programování a ladění těchto systémů.

Dostupnost systémových komunikačních služeb

Systémové komunikační služby, které využívá vývojové prostředí Mosaic, jsou dostupné na všech komunikačních kanálech v režimu **PC**. Ovšem v daném okamžiku je možné používat systémové služby jen na jednom z těchto kanálů. Oprávnění využívat systémové služby dostane ten komunikační kanál, který přijde s požadavkem jako první. Oprávnění zaniká 5 sekund po ukončení komunikace.

Dostupnost veřejných komunikačních služeb

Veřejné komunikační služby jsou dostupné na všech komunikačních kanálech v režimu **PC** a to i souběžně. Přehled veřejných komunikačních služeb uvádí tab.5.1.

Tab.5.1 Přehled veřejných komunikačních služeb protokolu EPSNET

kód služby		název	funkce
$08	8	SETTID	nastavení času
$0A	10	GETSW	přečtení stavového slova
$0B	11	READN	čtení z paměti proměnných
$0C	12	WRITEN	zápis do paměti proměnných
$0D	13	WANDRN	zápis do a čtení z paměti proměnných
$0E	14	GETERR	přečtení chybového zásobníku
$0F	15	READB	čtení bitů z paměti proměnných
$10	16	WRITEB	zápis do bitů paměti proměnných
$90	144	READBD	destruktivní čtení bitů z paměti proměnných
$91	145	READND	destruktivní čtení z paměti proměnných
$93	147	WANDRND	zápis do a destruktivní čtení z paměti proměnných

5.6. POPIS KOMUNIKAČNÍCH SLUŽEB

Následující text popisuje formáty veřejných komunikačních služeb. První skupinu tvoří služby potřebné pro zahájení komunikace. Používají se k navázání spojení. Druhou skupinu tvoří služby provozní, které umožňují zjistit stav dotazovaného systému. Třetí skupinu tvoří služby datové, které provádějí vlastní přenos dat.

5.6.1. Zahájení komunikace

CONNECT - navázání spojení

Funkce:

Zahájení komunikace spojené s inicializací komunikačních struktur.

Syntaxe:

zpráva

SD1	DA	SA	FC	FCS	ED

FC = $49 nebo $69

pozitivní odpověď

SD1	DA	SA	FC	FCS	ED

FC = 0

Příklad:

Navázání komunikace se stanicí s nastavenou adresou 0. Nadřízené stanici určíme adresu jinou, např. 126.

zpráva - $10 $00 $7E $69 $E7 $16
odpověď - $10 $7E $00 $00 $7E $16

IDENT - identifikace systému

Funkce:

Zjištění informací o typu připojeného systému.

Syntaxe:

zpráva

SD1	DA	SA	FC	FCS	ED

FC = $4E nebo $6E

pozitivní odpověď

SD2	LE	LER	SD2R	DA	SA	FC	DATA.....	FCS	ED

FC = 0

DATA - byte 0 - LIS - délka identifikačního řetězce (obvykle 7)
byte 1 - LIP - délka typu implementace protokolu (vždy 1)
byte 2 - LST - délka řetězce určujícího verzi struktur (vždy 3)
byte 3 - LSW - délka řetězce určujícího verzi sw (vždy 3)
od bytu 4 - identifikační řetězec centrální jednotky (např. CP2005I)
byte 4+LIS - znak implementace protokolu (B)
od bytu 4+LIS+LIP - řetězec verze struktur (např. 2.5)
od bytu 4+LIS+LIP+LST - řetězec verze softwaru (např. 1.2)

Příklad:

Přečtení identifikace systému s nastavenou adresou 0. Nadřízená stanice má adresu 126.

zpráva - $10 $00 $7E $6E $EC $16
odpověď - $68 $15 $15 $68 $7E $00 $00 | $08 $01 $03 $03 | $43 $50 $32 $30 $30 $35 $49 $20 | $42 | $32 $2E $35 | $31 $2E $32 | $B4 $16
datové pole v ASCII CP2005I B2.51.2

5.6.2. Provozní služby

SETTID - nastavení času

Funkce:

Zápis časového údaje do časového obvodu RTC. Od tohoto okamžiku systém pracuje s novým časem.

Pozn.: Pomocí instrukce RDT lze číst synchronizovaný čas v milisekundách. Po příjmu služby SETTID je nastaven čas a milisekundy jsou vynulovány (to neplatí pro čas v registrech S5 až S12, kde se milisekundy nenulují - čas je průběžný z důvodu kontinuity časově závislých funkcí systému).

Syntaxe:

zpráva

SD2	LE	LER	SD2R	DA	SA	FC	08	TIME	FCS	ED

LE = $0B
LER = $0B
FC = $43 nebo $63
TIME - časový údaj délky 7 bytů
1. byte - rok (poslední dvojčíslí 0 - 99, $00 - $63)
2. byte - měsíc (1 - 12, $01 - $0C)
3. byte - den (1 - 31, $01 - $1F)
4. byte - hodina (0 - 23, $00 - $17)
5. byte - minuta (0 - 59, $00 - $3B)
6. byte - sekunda (0 - 59, $00 - $3B)
7. byte - den v týdnu (1 - pondělí, 2 - úterý, 3 - středa, 4 - čtvrtek, 5 - pátek, 6 - sobota, 7 - neděle)

pozitivní odpověď

SAC

Příklad:

Zápis data a času 20.1.1996 6:55:00 pátek do podřízené stanice s adresou 3.

zpráva - $68 $0B $0B $68 $03 $7E $63 | $08 $60 $01 $14 $06 $37 $00 $05 | $A3 $16
odpověď - $E5

SETCW - nastavení řídícího slova

Funkce:

Zápis do řídícího slova.

Pozor! Tato služba je systémová, je tedy přístupná pouze na komunikačních kanálech podporujících systémové služby.

Syntaxe:

zpráva

SD2	LE	LER	SD2R	DA	SA	FC	09	CW_L	CW_H	FCS	ED

LE = 6
LER = 6
FC = $43 nebo $63
CW_L - dolní byte řídícího slova

0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0

CW_H - horní byte řídícího slova

MOD	BLO	CLO	TPR	RES	0	0	CLE
7	6	5	4	3	2	1	0

CLE - požadavek nulování chyby (aktivní v 1)
RES - požadavek na restart systému (aktivní v 1)
TPR - typ restartu
0 - teplý restart
1 - studený restart
CLO - požadavek nulování výstupů (aktivní v 1)
BLO - požadavek blokování výstupů (aktivní v 1)
MOD - požadovaný režim systému
0 - režim HALT
1 - režim RUN
0 - povinný zápis hodnoty 0

pozitivní odpověď

SAC

Příklad:

Zablokování výstupů podřízené stanice s adresou 3.

zpráva - $68 $06 $06 $68 $03 $7E $63 | $09 $00 $40 | $2D $16
odpověď - $E5

GETSW - přečtení stavového slova

Funkce:

Čtení stavového slova.

Syntaxe:

zpráva

SD2	LE	LER	SD2R	DA	SA	FC	$0A	FCS	ED

LE = 4
LER = 4
FC = $4C nebo $6C

pozitivní odpověď

SD2	LE	LER	SD2R	DA	SA	FC	SW_L	SW_H	FCS	ED

LE = 5
LER = 5
FC = 8
SW_L - dolní byte stavového slova

X	X	X	X	X	X	X	X
7	6	5	4	3	2	1	0

SW_H - horní byte stavového slova

MOD	BLO	X	X	ERS	X	X	ERH
7	6	5	4	3	2	1	0

ERH - indikace závažné chyby (aktivní v 1)
ERS - indikace ostatních chyb (aktivní v 1)
BLO - indikace blokování výstupů (aktivní v 1)
MOD - indikace režimu systému
0 - režim HALT
1 - režim RUN
X - rezerva

Příklad:

Přečtení stavového slova podřízené stanice s adresou 3.

zpráva - $68 $04 $04 $68 $03 $7E $6C | $0A | $F7 $16
odpověď - $68 $05 $05 $68 $7E $03 $08 | $00 $80 | $09 $16

Systém je v režimu RUN s odblokovanými výstupy bez chyb.

GETERR - přečtení chybového zásobníku

Funkce:

Čtení hlavního chybového zásobníku centrální jednotky.

Syntaxe:

zpráva

SD2	LE	LER	SD2R	DA	SA	FC	$0E	FCS	ED

LE = 4
LER = 4
FC = $4C nebo $6C

pozitivní odpověď

SD2	LE	LER	SD2R	DA	SA	FC	DATAE	FCS	ED

LE = $23
LER = $23
FC = 8
DATAE - chybový zásobník obsahující celkem 8 chybových hlášení v pořadí od nejstaršího k nejnovějšímu, každé o délce 4 byty (viz příručku příslušného systému), celková délka pole je 32 bytů

Příklad:

Přečtení hlavního chybového zásobníku podřízené stanice s adresou 3.

zpráva - $68 $04 $04 $68 $03 $7E $6C | $0E | $FB $16
odpověď - $68 $23 $23 $68 $7E $03 $08 | $00 $00 $00 $00 | $00 $00 $00 $00 | $00 $00 $00 $00 | $00 $00 $00 $00 | $00 $00 $00 $00 | $00 $00 $00 $00 | $08 $00 $00 $00 | $80 $30 $11 $24 | $76 $16

Systém hlásí chybu překročení doby cyklu (80 30 pcpc), které předcházelo varování (08 00 00 00).

MASKCW - nastavení jednotlivých bitů řídícího slova

Funkce:

Nastavení jednotlivých bitů řídícího slova.

Pozor! Tato služba je systémová, je tedy přístupná pouze na komunikačních kanálech podporujících systémové služby.

Syntaxe:

zpráva

SD2	LE	LER	SD2R	DA	SA	FC	$11	$C0_L$	$C0_H$	$C1_L$	$C1_H$	FCS	ED

LE = 8
LER = 8
FC = $43 nebo $63
$C0_L$, $C0_H$ - dolní a horní byte nulové masky řídícího slova
všechny bity řídícího slova, na jejichž pozicích je 0, budou vynulovány, ostatní zůstanou zachovány (funkce AND)
$C1_L$, $C1_H$ - dolní a horní byte jedničkové masky řídícího slova
všechny bity řídícího slova, na jejichž pozicích je 1, budou nastaveny, ostatní zůstanou zachovány (funkce OR)

pozitivní odpověď

SAC

Struktura řídícího slova je následující:

CW_L - dolní byte řídícího slova

0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0

CW_H - horní byte řídícího slova

MOD	BLO	CLO	TPR	RES	0	0	CLE
7	6	5	4	3	2	1	0

CLE - požadavek nulování chyby (aktivní v 1)
RES - požadavek na restart systému (aktivní v 1)
TPR - typ restartu
0 - teplý restart
1 - studený restart
CLO - požadavek nulování výstupů (aktivní v 1)
BLO - požadavek blokování výstupů (aktivní v 1)
MOD - požadovaný režim systému
0 - režim HALT
1 - režim RUN
0 - systémová rezerva, nenastavovat!

Příklad:

Zablokování výstupů podřízené stanice s adresou 3 s ponecháním stávajícího režimu bez nutnosti jeho zjišťování.

zpráva - $68 $08 $08 $68 $03 $7E $63 | $11 | $FF $FF | $00 $40 | $33 $16
odpověď - $E5

5.6.3. Datové služby

READN - čtení z paměti proměnných

Funkce:

Čtení obsahu paměti proměnných a obsahu přídavné paměti dat DataBox.

Syntaxe:

zpráva

SD2	LE	LER	SD2R	DA	SA	FC	$0B	TR1	$IR1_L$	$IR1_H$	LR1	...

...	TRn	IRn_L	IRn_H	LRn	FCS	ED

LE = 4 + 4n (n je počet čtených bloků dat)
LER = 4 + 4n
FC = $4C nebo $6C
TRn - oblast paměti proměnných, ze které se čte
0 - registry %X
1 - registry %Y
2 - registry %S
3 - registry %R (%R0 - %R65535 .. index $0000 - $FFFF)
4 - registry %R (%R65536 - %R131071 .. index $0000 - $FFFF)
5 - registry %R (%R131072 - %R196607 .. index $0000 - $FFFF)
6 - registry %R (%R196608 - %R262143 .. index $0000 - $FFFF)
7 - registry %R (%R262144 - %R327679 .. index $0000 - $FFFF)
- oblast přídavné paměti dat DataBox, ze které se čte
$80 - adresy $00000 - $0FFFF
$81 - adresy $10000 - $1FFFF
$82 - adresy $20000 - $2FFFF
atd. až do maximálního rozsahu této paměti (dáno typem DataBoxu)
IRn_L, IRn_H - dolní a horní byte indexu prvního čteného registru
LRn - počet čtených registrů

pozitivní odpověď

SD2	LE	LER	SD2R	DA	SA	FC	DATAR1	...	DATARn	FCS	ED

LE = LR1 + ... + LRn + 3
LER = LR1 + ... + LRn + 3
FC = 8
DATARn - blok čtených registrů

Příklad:

Přečtení obsahu registrů %R30 až %R35 a %X0 a %X1 podřízené stanice s adresou 4.

zpráva - $68 $0C $0C $68 $04 $7E $6C | $0B | $03 $1E $00 $06 | $00 $00 $00 $02 | $22 $16
odpověď - $68 $0B $0B $68 $7E $04 $08 | $01 $02 $03 $04 $05 $06 | $01 $02 | $A2 $16

WRITEN - zápis do paměti proměnných

Funkce:

Zápis do paměti proměnných a do přídavné paměti dat DataBox.

Syntaxe:

zpráva

SD2	LE	LER	SD2R	DA	SA	FC	$0C	TW1	$IW1_L$	$IW1_H$	LW1	DATAW1	...

...	TWn	IWn_L	IWn_H	LWn	DATAWn	FCS	ED

LE = 4 + 4n + LW1 + ... + LWn (n je počet zapisovaných bloků dat)
LER = 4 + 4n + LW1 + ... + LWn
FC = $43 nebo $63
TWn - oblast paměti proměnných, do které se zapisuje
0 - registry %X
1 - registry %Y
2 - registry %S
3 - registry %R (%R0 - %R65535 .. index $0000 - $FFFF)
4 - registry %R (%R65536 - %R131071 .. index $0000 - $FFFF)
5 - registry %R (%R131072 - %R196607 .. index $0000 - $FFFF)
6 - registry %R (%R196608 - %R262143 .. index $0000 - $FFFF)
7 - registry %R (%R262144 - %R327679 .. index $0000 - $FFFF)
- oblast přídavné paměti dat DataBox, do které se zapisuje
$80 - adresy $00000 - $0FFFF
$81 - adresy $10000 - $1FFFF
$82 - adresy $20000 - $2FFFF
atd. až do maximálního rozsahu této paměti (dáno typem DataBoxu)
IWn_L, IWn_H - dolní a horní byte indexu prvního zapisovaného registru
LWn - počet zapisovaných registrů
DATAWn - blok zapisovaných registrů

pozitivní odpověď

SAC

Příklad:

Zápis do registrů %R30 až %R35 a %Y0 a %Y1 podřízené stanice s adresou 4.

zpráva - $68 $14 $14 $68 $04 $7E $63 | $0C | $03 $1E $00 $06 $01 $02 $03 $04 $05 $06 | $01 $00 $00 $02 $01 $02 | $33 $16
odpověď - $E5

WANDRN - zápis do a čtení z paměti proměnných

Funkce:

Zápis do a čtení z paměti proměnných nebo přídavné paměti dat DataBox.

Syntaxe:

zpráva

SD2	LE	LER	SD2R	DA	SA	FC	$0D	TR	IR_L	IR_H	LR	TW	IW_L	IW_H	LW
DATAW				FCS	ED										

LE = $0C + LW
LER = $0C + LW
FC = $4C nebo $6C
TR - oblast paměti proměnných, ze které se čte
0 - registry %X
1 - registry %Y
2 - registry %S
3 - registry %R (%R0 - %R65535 .. index $0000 - $FFFF)
4 - registry %R (%R65536 - %R131071 .. index $0000 - $FFFF)
5 - registry %R (%R131072 - %R196607 .. index $0000 - $FFFF)
6 - registry %R (%R196608 - %R262143 .. index $0000 - $FFFF)
7 - registry %R (%R262144 - %R327679 .. index $0000 - $FFFF)
- oblast přídavné paměti dat DataBox, ze které se čte
$80 - adresy $00000 - $0FFFF
$81 - adresy $10000 - $1FFFF
$82 - adresy $20000 - $2FFFF
atd. až do maximálního rozsahu této paměti (dáno typem DataBoxu)
IR_L, IR_H - dolní a horní byte indexu prvního čteného registru
LR - počet čtených registrů
TW - oblast paměti proměnných, do které se zapisuje
(kódování viz TR)
- oblast přídavné paměti dat DataBox, do které se zapisuje
(kódování viz TR)
IW_L, IW_H - dolní a horní byte indexu prvního zapisovaného registru
LW - počet zapisovaných registrů
DATAW - blok zapisovaných registrů

pozitivní odpověď

SD2	LE	LER	SD2R	DA	SA	FC	DATAR	FCS	ED

LE = LR + 3
LER = LR + 3
FC = 8
DATAR - blok čtených registrů

Příklad:

Zápis do registrů %R30 až %R35 a přečtení registrů %X0 a %X1 podřízené stanice s adresou 4.

zpráva - $68 $12 $12 $68 $04 $7E $6C | $0D | $00 $00 $00 $02 | $03 $1E $00 $06 $01 $02 $03 $04 $05 $06 | $39 $16
odpověď - $68 $05 $05 $68 $7E $04 $08 | $01 $02 | $8D $16

READB - čtení bitů z paměti proměnných

Funkce:

Čtení obsahu bitů paměti proměnných.

Syntaxe:

zpráva

SD2	LE	LER	SD2R	DA	SA	FC	$0F	TR1	$IR1_L$	$IR1_H$	BR1	...

...	TRn	IRn_L	IRn_H	BRn	FCS	ED

LE = 4 + 4n (n je počet čtených bitů)
LER = 4 + 4n
FC = $4C nebo $6C
TRn - oblast paměti proměnných, ze které se čte
0 - registry %X
1 - registry %Y
2 - registry %S
3 - registry %R (%R0 - %R65535 .. index $0000 - $FFFF)
4 - registry %R (%R65536 - %R131071 .. index $0000 - $FFFF)
5 - registry %R (%R131072 - %R196607 .. index $0000 - $FFFF)
6 - registry %R (%R196608 - %R262143 .. index $0000 - $FFFF)
7 - registry %R (%R262144 - %R327679 .. index $0000 - $FFFF)
IRn_L, IRn_H - dolní a horní byte indexu čteného registru
BRn - index čteného bitu (0 až 7)

pozitivní odpověď

SD2	LE	LER	SD2R	DA	SA	FC	BITR1	...	...	BITRn	FCS	ED

LE = 3 + n
LER = 3 + n
FC = 8
BITRn - hodnota přečteného bitu roztažená na byte (0 → 00, 1 → $FF)

Příklad:

Přečtení obsahu bitů %R34.2 a %R34.5 podřízené stanice s adresou 4.

zpráva - $68 $0C $0C $68 $04 $7E $6C | $0F | $03 $22 $00 $02 | $03 $22 $00 $05 | $4E $16
odpověď - $68 $05 $05 $68 $7E $04 $08 | $00 | $FF | $89 $16

Bit %R34.2 obsahuje hodnotu 0, bit %R34.5 obsahuje hodnotu 1.

WRITEB - zápis do bitů paměti proměnných

Funkce:

Zápis do bitů v paměti proměnných.

Syntaxe:

zpráva

SD2	LE	LER	SD2R	DA	SA	FC	$10	TW1	$IW1_L$	$IW1_H$	BW1	...

...	TWn	IWn_L	IWn_H	BWn	FCS	ED

LE = 4 + 4n (n je počet zapisovaných bitů)
LER = 4 + 4n
FC = $43 nebo $63
TWn - oblast paměti proměnných, do které se zapisuje
0 - registry %X
1 - registry %Y
2 - registry %S
3 - registry %R (%R0 - %R65535 .. index $0000 - $FFFF)
4 - registry %R (%R65536 - %R131071 .. index $0000 - $FFFF)
5 - registry %R (%R131072 - %R196607 .. index $0000 - $FFFF)
6 - registry %R (%R196608 - %R262143 .. index $0000 - $FFFF)
7 - registry %R (%R262144 - %R327679 .. index $0000 - $FFFF)
IWn_L, IWn_H - dolní a horní byte indexu zapisovaného registru
BWn - index zapisovaného bitu, na nejvyšším bitu zapisovaná hodnota (zápis 0 - $00 až $07, zápis 1 - $80 až $87)

pozitivní odpověď

SAC

Příklad:

Nastavení bitu %R15.6 a nulování bitu %R17.1 podřízené stanice s adresou 4.

zpráva - $68 $0C $0C $68 $04 $7E $63 | $10 | $03 $0F $00 $86 | $03 $11 $00 $01 | $A2 $16
odpověď - $E5

READBD - destruktivní čtení bitů z paměti proměnných

Funkce:

Přečtení obsahu bitů paměti proměnných a jejich následné vynulování.

Syntaxe:

zpráva

SD2	LE	LER	SD2R	DA	SA	FC	$0F	TR1	$IR1_L$	$IR1_H$	BR1	...

...	TRn	IRn_L	IRn_H	BRn	FCS	ED

LE = 4 + 4n (n je počet čtených bitů)
LER = 4 + 4n
FC = $4C nebo $6C
TRn - oblast paměti proměnných, ze které se čte
0 - registry %X
1 - registry %Y
2 - registry %S
3 - registry %R (%R0 - %R65535 .. index $0000 - $FFFF)
4 - registry %R (%R65536 - %R131071 .. index $0000 - $FFFF)
5 - registry %R (%R131072 - %R196607 .. index $0000 - $FFFF)
6 - registry %R (%R196608 - %R262143 .. index $0000 - $FFFF)
7 - registry %R (%R262144 - %R327679 .. index $0000 - $FFFF)
IRn_L, IRn_H - a horní byte indexu čteného registru
BRn - index čteného bitu (0 až 7)

pozitivní odpověď

SD2	LE	LER	SD2R	DA	SA	FC	BITR1	...	...	BITRn	FCS	ED

LE = 3 + n
LER = 3 + n
FC = 8
BITRn - hodnota přečteného bitu roztažená na byte (0 → 00, 1 → $FF)

Příklad:

Přečtení obsahu bitů %R34.2 a %R34.5 podřízené stanice s adresou 4.

zpráva - $68 $0C $0C $68 $04 $7E $6C | $90 | $03 $22 $00 $02 | $03 $22 $00 $05 | $CF $16
odpověď - $68 $05 $05 $68 $7E $04 $08 | $00 | $FF | $89 $16

Bit %R34.2 obsahoval hodnotu 0, bit %R34.5 obsahoval hodnotu 1. Po přečtení byly oba bity vynulovány.

READND - destruktivní čtení z paměti proměnných

Funkce:

Čtení obsahu paměti proměnných, resp. obsahu přídavné paměti dat DataBox, a jejich vynulování.

Syntaxe:

zpráva

SD2	LE	LER	SD2R	DA	SA	FC	$0B	TR1	$IR1_L$	$IR1_H$	LR1	...

...	TRn	IRn_L	IRn_H	LRn	FCS	ED

LE = 4 + 4n (n je počet čtených bloků dat)
LER = 4 + 4n
FC = $4C nebo $6C
TRn - oblast paměti proměnných, ze které se čte
0 - registry %X
1 - registry %Y
2 - registry %S
3 - registry %R (%R0 - %R65535 .. index $0000 - $FFFF)
4 - registry %R (%R65536 - %R131071 .. index $0000 - $FFFF)
5 - registry %R (%R131072 - %R196607 .. index $0000 - $FFFF)
6 - registry %R (%R196608 - %R262143 .. index $0000 - $FFFF)
7 - registry %R (%R262144 - %R327679 .. index $0000 - $FFFF)
- oblast přídavné paměti dat DataBox, ze které se čte
$80 - adresy $00000 - $0FFFF
$81 - adresy $10000 - $1FFFF
$82 - adresy $20000 - $2FFFF
atd. až do maximálního rozsahu této paměti (dáno typem DataBoxu)
IRn_L, IRn_H - dolní a horní byte indexu prvního čteného registru
LRn - počet čtených registrů

pozitivní odpověď

SD2	LE	LER	SD2R	DA	SA	FC	DATAR1	...	DATARn	FCS	ED

LE = LR1 + ... + LRn + 3
LER = LR1 + ... + LRn + 3
FC = 8
DATARn - blok čtených registrů

Příklad:

Přečtení obsahu registrů %R30 až %R35 a %X0 a %X1 podřízené stanice s adresou 4.

zpráva - $68 $0C $0C $68 $04 $7E $6C | $91 | $03 $1E $00 $06 | $00 $00 $00 $02 | $A8 $16
odpověď - $68 $0B $0B $68 $7E $04 $08 | $01 $02 $03 $04 $05 $06 | $01 $02 | $A2 $16

Po přečtení obsahu byly registry vynulovány.

WANDRND - zápis do a destruktivní čtení z paměti proměnných

Funkce:

Zápis do registrů a čtení obsahu registrů paměti proměnných nebo přídavné paměti dat DataBox s jejich následným vynulováním.

Syntaxe:

zpráva

SD2	LE	LER	SD2R	DA	SA	FC	$0D	TR	IR_L	IR_H	LR	TW	IW_L	IW_H	LW
DATAW				FCS	ED										

LE = $C + LW
LER = $C + LW
FC = $4C nebo $6C
TR - oblast paměti proměnných, ze které se čte
0 - registry %X
1 - registry %Y
2 - registry %S
3 - registry %R (%R0 - %R65535 .. index $0000 - $FFFF)
4 - registry %R (%R65536 - %R131071 .. index $0000 - $FFFF)
5 - registry %R (%R131072 - %R196607 .. index $0000 - $FFFF)
6 - registry %R (%R196608 - %R262143 .. index $0000 - $FFFF)
7 - registry %R (%R262144 - %R327679 .. index $0000 - $FFFF)
- oblast přídavné paměti dat DataBox, ze které se čte
$80 - adresy $00000 - $0FFFF
$81 - adresy $10000 - $1FFFF
$82 - adresy $20000 - $2FFFF
atd. až do maximálního rozsahu této paměti (dáno typem DataBoxu)
IR_L, IR_H - dolní a horní byte indexu prvního čteného registru
LR - počet čtených registrů
TW - oblast paměti proměnných, do které se zapisuje
(kódování viz TR)
- oblast přídavné paměti dat DataBox, do které se zapisuje
(kódování viz TR)
atd. až do maximálního rozsahu této paměti (dáno typem DataBoxu)
IW_L, IW_H - dolní a horní byte indexu prvního zapisovaného registru
LW - počet zapisovaných registrů
DATAW - blok zapisovaných registrů

pozitivní odpověď

SD2	LE	LER	SD2R	DA	SA	FC	DATAR	FCS	ED

LE = LR + 3
LER = LR + 3
FC = 8
DATAR - blok čtených registrů

Příklad:

Zápis do registrů %R30 až %R35 a přečtení registrů %X0 a %X1 podřízené stanice s adresou 4.

zpráva - $68 $12 $12 $68 $04 $7E $6C | $93 | $00 $00 $00 $02 | $03 $1E $00 $06 $01 $02 $03 $04 $05 $06 | $8F $16

odpověď - $68 $05 $05 $68 $7E $04 $08 | $01 $02 | $8D $16

Po přečtení obsahu byly registry vynulovány.

Teco, a.s., Průmyslová zóna Šťáralka 984, 280 02 Kolín, tel. 321 401 111

TXV 004 69.01

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