

Knihovna DebugComLib

TXV 003 57.01
druhé vydání
prosinec 2010
změny vyhrazeny

Historie změn

Datum	Vydání	Popis změn
Květen 2010	1	První vydání
Prosinec 2010	2	K verzi knihovny 1.4 (přidán blok <i>fbDebugComEx</i>)

OBSAH

1 Úvod.....	3
2 Datové typy.....	4
3 Konstanty.....	4
4 Ladění na kanálech a spojeních v režimu UNI.....	5
4.1 Funkční blok fbDebugCom.....	5
4.2 Funkční blok fbDebug2Com.....	8
4.3 Funkční blok fbDebugComEx.....	10

1 ÚVOD

Knihovna DebugComLib obsahuje funkční bloky pro záznam komunikace probíhající na kanálech a spojeních v režimu UNI do databoxu.

Knihovna využívá některé struktury, funkce a funkční bloky z knihoven SysLib (TXV 003 48), ComLib (TXV 003 51) a DataboxLib (TXV 003 56). Pro správnou funkci musí být tyto knihovny zařazeny v projektu před knihovnou DebugComLib.

Knihovna obsahuje následující funkční bloky

fbDebugCom	Ukládání komunikace jednoho kanálu/spojení
fbDebug2Com	Ukládání komunikace dvou kanálu/spojení

2 *DATOVÉ TYPY*

V knihovně DebugComLib nejsou definovány žádné datové typy.

3 *KONSTANTY*

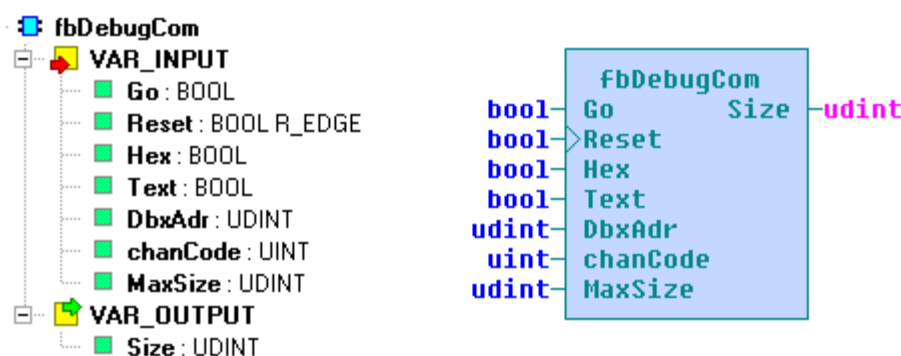
V knihovně DebugComLib jsou definovány pouze pomocné konstanty obsahující řetězce, které se uvnitř bloků opakují.

4 LADĚNÍ NA KANÁLECH A SPOJENÍCH V REŽIMU UNI

Ladící bloky slouží k zachycení všech vyslaných a přijatých komunikací na kanálech nebo spojeních v režimu UNI. Komunikace se ukládají do databoxu v textovém nebo hexadecimálním formátu.

4.1 Funkční blok fbDebugCom

knihovna: *DebugComLib*



Popis proměnných :

	<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
VAR_INPUT			
	<i>Go</i>	BOOL	Řídící proměnná. Pokud má hodnotu TRUE probíhá ukládání dat do databoxu
	<i>Reset</i>	BOOL R_EDGE	Náběžná hrana (přechod z hodnotu FALSE na hodnotu TRUE) vynuluje velikost, nová data se začnou plnit opět od začátku
	<i>Hex</i>	BOOL	Data se ukládají v hexadecimálním formátu
	<i>Text</i>	BOOL	Data se ukládají jako text
	<i>DbxAdr</i>	UDINT	Adresa začátku uložených dat v databoxu
	<i>chanCode</i>	UINT	Kód spojení ETH1_uni0, ETH1_uni1,... nebo kanálu CH1_uni, CH2_uni
	<i>MaxSize</i>	UDINT	Maximální velikost dat v databoxu v bytech
VAR_OUTPUT			
	<i>Size</i>	UDINT	Aktuální velikost uložených dat

Blok je možné volat, kdekoliv v programu. Pro správnou funkci je nutné dodržet, aby byl blok volán každou otočku PLC. Záznam je podmíněn vstupem *Go*. Náběžná hrana na vstupu *Reset* vynuluje obsazenou velikost, čímž umožňuje zachytávat novou sadu dat. Data jsou, pokud nejsou nastaveny vstupy *Hex* a *Text*, ukládána jako text, stejně jako při nastavení *Text* na TRUE. Při nastavení *Hex* na hodnotu TRUE se data ukládají hexadecimálně (jednotlivé byte oddělené mezerou). Při nastavení obou vstupů se použijí oba formáty ukládání.

Umístění dat v databoxu lze ovlivnit nastavením adresy na vstupu *DbxAdr*. Maximální velikost bloku všech zaznamenaných je omezena vstupem *MaxSize*. Pokud by velikost indikovaná výstupem *Size* přesáhla po uložení nových dat maximální velikosti ukládání se zastaví.

Formát dat v databoxu v textovém režimu je následující:

#08:35:20.670#TCP OPEN	Požadavek na navázání TCP spojení z uživatelského programu
#08:35:20.690#TCP ESTABLISHED	Navázání TCP spojení
#08:35:20.700#OUT len:249	Vysílání dat o délce 249 bytů
POST /plc/post.php HTTP/1.1 User-Agent: Foxtrot/5.5 (Tecomat; N; en-US) Cache-Control: max-age=0 Content-Type: application/x-www-form-urlencoded Accept: text/*;q=0.9, */*;q=0.1 Host: owned.own.cz Content-Length: 6 Connection: close val1=9	Vlastní odesílaná data
#08:35:21.570#TCP CLOSED	Uzavření spojení z protistrany
#08:35:21.570#IN len:345	Přijatá data o délce 345 bytů
HTTP/1.1 200 OK Date: Fri, 07 May 2010 06:35:17 GMT Server: Apache/2.2.3 (Debian) X-Powered-By: PHP/5.2.0-8+etch10 Content-Type: text/plain Vary: User-Agent,Accept-Encoding Connection: close Transfer-Encoding: chunked 1f val = 9 DT#10-05-07-08:35:19 0	Vlastní přijatá data
#08:35:21.570#TCP CLOSE	Požadavek na uzavření spojení z uživatelského programu

V hexadecimálním formátu vypadají data následovně:

#12:30:10.270#OUT len:32	Vysílání dat o délce 32 bytů
00 01 01 10 00 01 00 00 00 00 00 00 04 73 6D 74 70 06 73 65 7A 6E 61 6D 02 63 7A 00 00 01 00 01	Vlastní odesílaná data
#12:30:13.310#IN len:114	Přijatá data o délce 114 bytů
00 02 81 90 00 01 00 01 00 02 00 02 04 73 6D 74 70 06 73 65 7A 6E 61 6D 02 63 7A 00 00 01 00 01 C0 0C 00 01 00 01 00 00 00 63 00 04 4D 4B 4C 2B C0 11 00 02 00 01 00 00 00 45 00 05 02 6D 73 C0 11 C0 11 00 02 00 01 00 00 00 45 00 05 02 6E 73 C0 11 C0 3C 00 01 00 01 00 00 00 7D 00 04 4D 4B 4D 4D C0 4D 00 01 00 01 00 00 00 25 00 04 4D 4B 49 4D	Vlastní přijatá data

Příklad programu s voláním funkčního bloku *fbDebugCom* :

Proměnná *Debug* podmiňuje záznam dat z komunikace. Při ukončení záznamu (sestupná hrana proměnné *Debug*) se zachycená data uloží na paměťovou kartu do souboru DEBUGUNI.TXT, který lze stáhnout buď přes prostředí Mosaic nebo přes webové rozhraní. Příklad vyžaduje knihovnu *FileLib* (TXV 003 41).

```

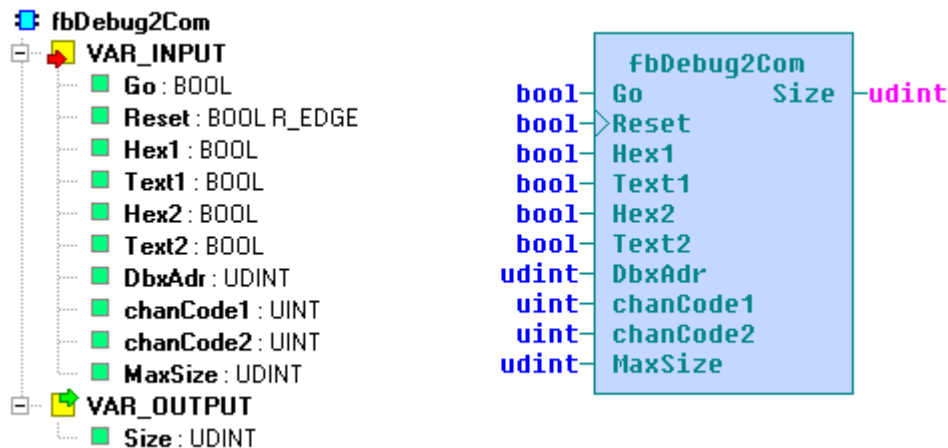
VAR_GLOBAL
  Debug : BOOL;
END_VAR

PROGRAM prgDebug1
  VAR
    DebugCom : fbDebugCom;
    iWriteDbxToFile : WriteDbxToFile;
    F1      : F_TRIG;
    FileName : STRING := 'WWW/DEBUGUNI.TXT';
  END_VAR

  DebugCom(Go := Debug, Hex := true, DbxAdr := 131072,
           chanCode := ETH1_uni0, MaxSize := 262144);
  F1(CLK := Debug);
  iWriteDbxToFile(fileName := FileName, exec := F1.Q,
                  srcAdr := DebugCom.DbxAdr, size := DebugCom.Size);
END_PROGRAM

```

4.2 Funkční blok fbDebug2Com

knihovna: *DebugComLib*

Popis proměnných :

	<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
VAR_INPUT			
	<i>Go</i>	BOOL	Řídící proměnná. Pokud má hodnotu TRUE probíhá ukládání dat do databoxu
	<i>Reset</i>	BOOL R_EDGE	Náběžná hrana (přechod z hodnotu FALSE na hodnotu TRUE) vynuluje velikost, nová data se začnou plnit opět od začátku
	<i>Hex1</i>	BOOL	Data z prvního kanálu/spojení se ukládají v hexadecimálním formátu
	<i>Text1</i>	BOOL	Data z prvního kanálu/spojení se ukládají jako text
	<i>Hex2</i>	BOOL	Data ze druhého kanálu/spojení se ukládají v hexadecimálním formátu
	<i>Text2</i>	BOOL	Data ze druhého kanálu/spojení se ukládají jako text
	<i>DbxAdr</i>	UDINT	Adresa začátku uložených dat v databoxu
	<i>chanCode1</i>	UINT	Kód prvního spojení ETH1_uni0, ETH1_uni1,... nebo kanálu CH1_uni, CH2_uni
	<i>chanCode2</i>	UINT	Kód druhého spojení ETH1_uni0, ETH1_uni1,... nebo kanálu CH1_uni, CH2_uni
	<i>MaxSize</i>	UDINT	Maximální velikost dat v databoxu v bytech
VAR_OUTPUT			
	<i>Size</i>	UDINT	Aktuální velikost uložených dat

Blok má stejné vlastnosti jako *fbDebugCom*. Navíc, ale umožňuje spojit záznam ze dvou kanálů nebo spojení. Vstupy *Hex*, *Text* a *chanCode* jsou doplněny indexem, který odpovídá prvnímu nebo druhému sledovanému kanálu nebo spojení.

Formát ukládaných dat se liší proti *fbDebugCom* tím, že je doplněn identifikátorem před časovou značkou (#CH1 pro první a #CH2 pro druhý kanál nebo spojení)

Příklad programu s voláním funkčního bloku *fbDebug2Com* :

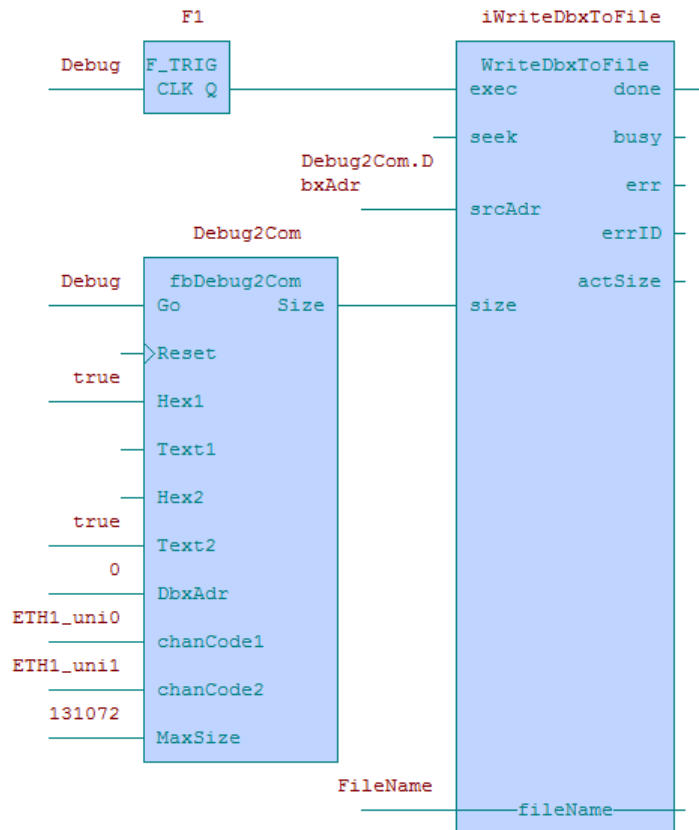
Proměnná *Debug* podmiňuje záznam dat z komunikace. Při ukončení záznamu (sestupná hrana proměnné *Debug*) se zachycená data uloží na paměťovou kartu do souboru DEBUGUNI.TXT, který lze stáhnout buď přes prostředí Mosaic nebo přes webové rozhraní. Příklad vyžaduje knihovnu *FileLib* (TXV 003 41).

```

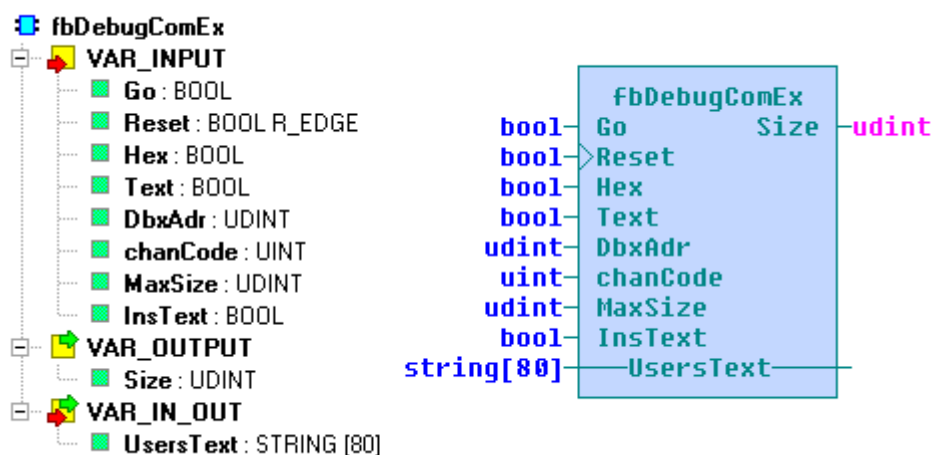
VAR_GLOBAL
  Debug : BOOL;
END_VAR

PROGRAM prgDebug2
  VAR
    Debug2Com : fbDebug2Com;
    iWriteDbxToFile : WriteDbxToFile;
    F1          : F_TRIG;
    FileName    : STRING := 'WWW/DEBUGUNI.TXT';
  END_VAR

```



4.3 Funkční blok fbDebugComEx

knihovna: *DebugComLib*

Popis proměnných :

	<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
VAR_INPUT			
	<i>Go</i>	BOOL	Řídící proměnná. Pokud má hodnotu TRUE probíhá ukládání dat do databoxu
	<i>Reset</i>	BOOL R_EDGE	Náběžná hrana (přechod z hodnotu FALSE na hodnotu TRUE) vynuluje velikost, nová data se začnou plnit opět od začátku
	<i>Hex</i>	BOOL	Data se ukládají v hexadecimálním formátu
	<i>Text</i>	BOOL	Data se ukládají jako text
	<i>DbxAdr</i>	UDINT	Adresa začátku uložených dat v databoxu
	<i>chanCode</i>	UINT	Kód spojení ETH1_uni0, ETH1_uni1,... nebo kanálu CH1_uni, CH2_uni
	<i>MaxSize</i>	UDINT	Maximální velikost dat v databoxu v bytech
	<i>InsText</i>	BOOL	Vloží do záznamu text na vstupu <i>UsersText</i>
VAR_IN_OUT			
	<i>UsersText</i>	STRING	Text, který se vloží do záznamu komunikace, když je <i>InsText</i> TRUE.
VAR_OUTPUT			
	<i>Size</i>	UDINT	Aktuální velikost uložených dat

Chování bloku je stejné jako u *fbDebugCom*. Blok je navíc rozšířen o možnost přidat k záznamu komunikace uživatelský text. Uživatelský text se předává přes vstup *UsersText* a je přidán do záznamu ve chvíli, kdy je blok zavolán se vstupem *InsText* s hodnotou TRUE. Text je opatřen stejnou časovou značkou jako komunikace a je doplněn o odřádkování.

Příklad programu s voláním funkčního bloku *fbDebugComEx* :

Proměnná *Debug* podmiňuje záznam dat z komunikace. Kromě komunikace je do databoxu uložen text z proměnné *ErrTxt* na náběžnou hranu proměnné *Err*. Při ukončení záznamu (sestupná hrana proměnné *Debug*) se zachycená data uloží na paměťovou kartu do souboru DEBUGUNI.TXT, který lze stáhnout buď přes prostředí Mosaic nebo přes webové rozhraní. Příklad vyžaduje knihovnu *FileLib* (TXV 003 41).

```
VAR_GLOBAL
  Debug   : BOOL;
  Err     : BOOL;
  ErrTxt  : STRING;
END_VAR

PROGRAM prgDebug3
  VAR
    DebugComEx : fbDebugComEx;
    iWriteDbxToFile : WriteDbxToFile;
    R1          : R_TRIG;
    F1          : F_TRIG;
    FileName   : STRING := 'WWW/DEBUGUNI.TXT';
  END_VAR

  R1(CLK := Err);
  DebugComEx(Go := Debug, Hex := true, Text := false, DbxAdr := 0,
             chanCode := ETH1_uni0, MaxSize := 131072,
             UsersText := ErrTxt, InsText := R1.Q);
  F1(CLK := Debug);
  iWriteDbxToFile(fileName := FileName, exec := F1.Q,
                 srcAdr := DebugComEx.DbxAdr, size := DebugComEx.Size);
END_PROGRAM
```



teco

Objednávky a informace:

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

TXV 003 57.01

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