

ChnFESTO

JEDNOTKA DEFINUJÍCÍ KOMUNIKAČNÍ PROTOKOL S PLC FESTO

Příručka uživatele a programátora



SofCon[®] spol. s r.o.
Střešovická 49
162 00 Praha 6
tel/fax: +420 220 180 454
E-mail: sofcon@sofcon.cz
www: <http://www.sofcon.cz>

Informace v tomto dokumentu byly pečlivě zkontrolovány a SofCon věří, že jsou spolehlivé, přesto SofCon nenese odpovědnost za případné nepřesnosti nebo nesprávnosti zde uvedených informací.

SofCon negarantuje bezchybnost tohoto dokumentu ani programového vybavení, které je v tomto dokumentu popsáno. Uživatel přebírá informace z tohoto dokumentu a odpovídající programové vybavení ve stavu, jak byly vytvořeny a sám je povinen provést validaci bezchybnosti produktu, který s použitím zde popsaného programového vybavení vytvořil.

SofCon si vyhrazuje právo změny obsahu tohoto dokumentu bez předchozího oznámení a nenese žádnou odpovědnost za důsledky, které z toho mohou vyplynout pro uživatele.

Datum vydání: 05.04.2004

Datum posledního uložení dokumentu: 05.04.2004

(Datum vydání a posledního uložení dokumentu musí být stejné)

Upozornění:

V dokumentu použité názvy výrobků, firem apod. mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

Obsah :

1.	O dokumentu	5
1.1.	Revize dokumentu	5
1.2.	Účel dokumentu	5
1.3.	Rozsah platnosti	5
1.4.	Související dokumenty	5
2.	Termíny a definice	5
3.	Úvod	6
4.	Konstanty a typy	6
4.1.	Řídící znaky a řetězce protokolu	6
4.2.	Typy příkazů protokolu	7
4.3.	Typy čtených a zapisovaných parametrů	7
4.4.	Typy způsobů zpráv pro zápis parametrů	8
4.5.	Konstanty výsledků přijímacího automatu	8
4.6.	Struktury přijímacích a vysílacích bufferů	8
5.	Objekty	10
5.1.	tChnFesto	10
5.1.1.	Položky	10
5.1.2.	Metody	11
5.1.2.1.	Init konstruktor	11
5.1.2.2.	ChInitParam konstruktor	11
5.1.2.3.	Done destruktork	11
5.1.2.4.	ChSetOneParam funkce	11
5.1.2.5.	ChGetParam funkce	12
5.1.2.6.	ChConnect procedura	12
5.1.2.7.	ChDisConnect procedura	12
5.1.2.8.	ChSend procedura	12
5.1.2.9.	ChReceiveReady funkce	12
5.1.2.10.	ChReceive procedura	13
5.1.2.11.	ChReceiveFlush procedura	13
5.1.2.12.	ChReceiveTick procedura	13
5.2.	tAddChnFesto	13
5.2.1.	Metody	13
5.2.1.1.	ChInit funkce	13
6.	Struktura zpráv protokolu FESTO	13
6.1.	Struktura resetovací zprávy „Break Signal“	14
6.2.	Struktura zprávy „Run“ pro spuštění programu	14
6.3.	Struktura zprávy „Stop“ pro zastavení programu	14
6.4.	Struktura zpráv pro čtení parametru	15
6.4.1.	Struktura zprávy pro čtení jednoho digitálního vstupu	15
6.4.2.	Struktura zprávy pro čtení jednoho digitálního výstupu	15
6.4.3.	Struktura zprávy pro čtení jednoho příznaku	16
6.4.4.	Struktura zprávy pro čtení slova (16-ti) digitálních vstupů	16
6.4.5.	Struktura zprávy pro čtení slova (16-ti) digitálních výstupů	16
6.4.6.	Struktura zprávy pro čtení slova (16-ti) příznaků	17
6.4.7.	Struktura zprávy pro čtení příznaku spuštění časovače	17
6.4.8.	Struktura zprávy pro čtení předvolby časovače	18
6.4.9.	Struktura zprávy pro čtení aktuální hodnoty časovače	18

6.4.10.	Struktura zprávy pro čtení příznaku spuštění čítače	18
6.4.11.	Struktura zprávy pro čtení předvolby čítače	19
6.4.12.	Struktura zprávy pro čtení aktuální hodnoty čítače	19
6.4.13.	Struktura zprávy pro čtení registru	20
6.4.14.	Struktura zprávy pro čtení chybového registru	20
6.4.15.	Struktura zprávy pro čtení příznaku spuštění procesu	20
6.5.	Struktura zpráv pro zápis parametru	21
6.5.1.	Struktura zprávy pro zápis jednoho digitálního výstupu	21
6.5.2.	Struktura zprávy pro zápis jednoho digitálního vstupu, slova (16-ti) digitálních vstupů a chybového registru	22
6.5.3.	Struktura zprávy pro zápis jednoho příznaku	23
6.5.4.	Struktura zprávy pro zápis slova (16-ti) digitálních výstupů	23
6.5.5.	Struktura zprávy pro zápis slova (16-ti) příznaků	24
6.5.6.	Struktura zprávy pro zápis příznaku spuštění časovače	24
6.5.7.	Struktura zprávy pro zápis předvolby časovače	25
6.5.8.	Struktura zprávy pro zápis aktuální hodnoty časovače	25
6.5.9.	Struktura zprávy pro zápis příznaku spuštění čítače	26
6.5.10.	Struktura zprávy pro zápis předvolby čítače	26
6.5.11.	Struktura zprávy pro zápis aktuální hodnoty čítače	27
6.5.12.	Struktura zprávy pro zápis hodnoty registru	27
6.5.13.	Struktura zprávy pro zápis příznaku spuštění procesu	28
6.6.	Struktura zprávy „Acces Error“ od Slave stanice	28
7.	Příklad	28

1. O dokumentu

1.1. Revize dokumentu

Verze dokumentu	Verze SW	Autor	Datum vydání	Popis změn
1.00	1.XX	Wi		První vydání.
1.10	4.XX	Tu	22.05.2003	Úprava dokumentu dle ISO9000.
1.20	4.XX	Wil	05.04.2004	Přidání konstant res_ErrFrame, res_ErrLen, res_ErrUnknownCode..

1.2. Účel dokumentu

Tento dokument slouží jako popis jednotky definující komunikační protokol s PLC FESTO.

1.3. Rozsah platnosti

Určen pro programátory a uživatele programového vybavení SofCon.

1.4. Související dokumenty

Pro čtení tohoto dokumentu je potřeba seznámit se s manuálem „ChnVirt“ popisujícím základní rodičovský prvek pro tvorbu komunikačních objektů a s manuálem „ChnTypes“.

Popis formátu verze knihovny a souvisejících funkcí je popsán v manuálu „LibVer“.

2. Termíny a definice

Používané termíny a definice jsou popsány v samostatném dokumentu „Termíny a definice“.

3. Úvod

Knihovna definuje formát komunikačního protokolu používaného při komunikaci se zařízeními PLC firmy FESTO. Obstarává zabezpečení dat, vkládání a vyjímání nadbytečností, tak jak to tento protokol předepisuje. Fyzický přenos dat je zajištěn prostřednictvím nižší komunikační vrstvy.

Knihovna ChnFesto definuje komunikační objekt **tChnFesto**, který je dědicem od rodičovského komunikačního objektu tChnVirt. Instance objektu tChnFesto reprezentuje vyšší komunikační vrstvu v komunikačním kanálu. Transformuje předávaná data mezi komunikačními objekty nižších vrstev, které provádějí fyzický přenos, a aplikací nebo případně další vyšší komunikační vrstvou. Určení, přes jakou fyzickou komunikační vrstvu bude komunikace probíhat, je voleno až parametry nastavovací metody ChSetParam. Komunikační protokol Festo má ale jednu specifickou vlastnost a tou je hardwarové vyslání Break signálu při zprávě „Break Signal“, kterou se obnovuje (resetuje) komunikace s připojeným zařízením PLC FESTO (na začátku komunikace, při zatuhnutí komunikace apod.). Jelikož knihovna ChnFesto definuje pouze transparentní formát protokolu, **musí se o hardwarové vyslání signálu Break postarat aplikace**. To se provede například prostřednictvím komunikační knihovny ChnComBR, jak je také ukázáno dále v příkladu.

Knihovna rovněž definuje objekt **tAddChnFesto**, který je dědicem od rodičovského objektu tAddChnVirt. Objekt tAddChnFesto zajistí, aby daný komunikační objekt (objekt tChnFesto) byl k aplikaci připojen a popřípadě zajistí vytvoření instance tohoto objektu. Po přilinkování této jednotky do aplikace (příkazem "uses ChnFesto"), se jméno objektu tChnFesto automaticky vloží do seznamu správců komunikačních objektů pro případné použití.

Protože je objekt **tChnFesto** dědicem rodičovského komunikačního objektu **tChnVirt**, jsou v této příručce popsány jen odlišnosti a speciality pro tento druh sériové komunikace. Ostatní naleznete v příručce **ChnVirt**. Některé použité konstanty a typy jsou předdefinované v jednotce **ChnTypes**.

4. Konstanty a typy

```
cVerNo = např. $0251; { BCD formát }  
cVer   = např. '02.51,07.08.2003';
```

Číslo verze jednotky v BCD tvaru a v textové podobě včetně datumu změny.

```
cName = 'FESTO';
```

Konstanta **cName** definuje jméno komunikačního objektu **tChnFesto**.

4.1. Řídící znaky a řetězce protokolu

```
zBr = #$00;  
zEr = #$07;  
zBS = #$08;  
zLF = #$0A;  
zCR = #$0D;  
zEM = #$11;
```

Konstanta **zBr** definuje kód znaku pro vyslání současně se hardwarovým signálem Break komunikační linky.

Konstanta **zEr** definuje kód znaku pro označení, že zpráva obsahuje řetězec „ACCESS ERROR“, který udává, že stanice nemohla zprávu správně dekodovat.

Konstanta **zBS** definuje kód znaku „BackSpace“ pro vymazání naposledy přijatého (předešlého) znaku.

Konstanta **zLF** definuje kód znaku „LineFeed“ používaného za znakem **zCR**.

Konstanta **zCR** definuje kód znaku „CarriageReturn“ pro označení konce datové části zprávy.

Konstanta **zEM** definuje kód znaku „End of Message“ pro označení konce celé zprávy od Slave stanice.

```
cAcsErrStr = 'ACCESS ERROR';
```

Konstanta **cAcsErrStr** definuje textový řetězec, který udává, že stanice nemohla zprávu správně dekodovat.

4.2. Typy příkazů protokolu

```
tCode =
  (tcBreak,
   tcDisp,
   tcModify,
   tcRun,
   tcStop,
   tcAcsErr);
```

Výčtový typ **tCode** definuje základní typy zpráv v protokolu FESTO.

Položka **tcBreak** je pro resetovací zprávu „Break Signal“, kterou se obnovuje komunikace s připojeným PLC FESTO (tzn. vysílá ji pouze Master).

Položka **tcDisp** je pro zprávu na čtení parametrů a položka **tcModify** je pro zprávu na zápis parametrů.

Položka **tcRun** je pro zprávu na spuštění programu v PLC FESTO a položka **tcStop** je pro zprávu na zastavení programu.

Položka **tcAcsErr** je pro zprávu „Access Error“, kterou Slave stanice (PLC FESTO) oznamuje, že naposledy přijatou zprávu nemohla správně dekodovat.

4.3. Typy čtených a zapisovaných parametrů

```
tParam =
  (tpDIn,
   tpDOut,
   tpFlag,
   tpDIns,
   tpDOuts,
   tpFlags,
   tpTimer,
   tpTimerPre,
   tpTimerVal,
   tpCounter,
   tpCounterPre,
   tpCounterVal,
   tpReg,
   tpErrReg,
   tpStatus);
```

Výčtový typ **tParam** definuje různé typy parametrů, které je možno číst nebo zapisovat pomocí příslušných zpráv (**tcDisp** a **tcModify**).

Položka **tpDIn** je pro čtení/zápis jednoho digitálního vstupu (bit).

Položka **tpDOut** je pro čtení/zápis jednoho digitálního výstupu (bit).

Položka **tpFlag** je pro čtení/zápis jednoho příznaku (bit).
 Položka **tpDIns** je pro čtení/zápis slova (16-ti) digitálních vstupů (word).
 Položka **tpDOuts** je pro čtení/zápis slova (16-ti) digitálních výstupů (word).
 Položka **tpFlags** je pro čtení/zápis slova (16-ti) příznaků (word).
 Položka **tpTimer** je pro čtení/zápis příznaku spuštění daného časovače (bit).
 Položka **tpTimerPre** je pro čtení/zápis předvolby daného časovače (byte).
 Položka **tpTimerVal** je pro čtení/zápis aktuální hodnoty daného časovače (byte).
 Položka **tpCounter** je pro čtení/zápis příznaku spuštění daného čítače (bit).
 Položka **tpCounterPre** je pro čtení/zápis předvolby daného čítače (byte).
 Položka **tpCounterVal** je pro čtení/zápis aktuální hodnoty daného čítače (byte).
 Položka **tpReg** je pro čtení/zápis hodnoty registru (byte).
 Položka **tpErrReg** je pro čtení/zápis chybového registru (3 byty).
 Položka **tpStatus** je pro čtení/zápis příznaku statusu daného procesu (bit).

4.4. Typy způsobů zpráv pro zápis parametrů

```
tModifyType =
  (tmWhole,
   tmOnlyCmd,
   tmOnlyVal);
```

Výčtový typ **tModifyType** definuje tři způsoby (druhy zpráv), jak lze zapsat daný parametr.

Položka **tmWhole** říká, že zpráva od Master i Slave stanice obsahuje příkaz pro zápis dat i zapisovaná data. Jedná se o nejčastější způsob.

Položka **tmOnlyCmd** říká, že zpráva od Master stanice obsahuje pouze příkaz pro zápis dat a Slave stanice na tuto zprávu odpovídá aktuální hodnotou daného parametru a čeká na zápis nové hodnoty (viz další zpráva s **tmOnlyVal**).

Položka **tmOnlyVal** říká, že zpráva od Master i Slave stanice obsahuje pouze nová zapisovaná data. Tato zpráva se smí posílat pouze po zprávě s **tmOnlyCmd**.

4.5. Konstanty výsledků přijímacího automatu

V této kapitole jsou popsány definice chybových kódů, které může vrátet metoda **ChReceiveResult** v průběhu přijímacího automatu.

```
res_ErrFrame = $20;
```

Výsledek **res_ErrFrame** indikuje chybu formátu přijaté zprávy.

```
res_ErrLen = $22;
```

Výsledek **res_ErrLen** indikuje chybu délky přijaté zprávy.

```
res_ErrUnknownCode = $25;
```

Výsledek **res_ErrUnknownCode** indikuje neznámý kód při vysílání zprávy.

4.6. Struktury přijímacích a vysílacích bufferů

```
pSendRecord = ^tSendRecord;
tSendRecord = record
  case Code          : tCode of
    tcBreak:
      (ID_Str       : string[30]);
    tcRun,
```



```

tcStop:
  ();
tcAcsErr:
  ();
tcDisp,
tcModify:
  (Index          : byte;
   ModifyType     : tModifyType;
   case Param     : tParam of
     tpDIn        ,
     tpDOut       ,
     tpFlag       :
       (IOF_Indx  : 0..15;
        IOF_Val   : boolean);
     tpDIns       ,
     tpDOuts      ,
     tpFlags      :
       (IOF_ValW  : word);
     tpTimer      :
       (Tm_OnOff  : boolean);
     tpTimerPre   :
       (Tm_Pre    : byte);
     tpTimerVal   :
       (Tm_Val    : byte);
     tpCounter    :
       (Ct_OnOff  : boolean);
     tpCounterPre :
       (Ct_Pre    : byte);
     tpCounterVal :
       (Ct_Val    : byte);
     tpReg        :
       (Reg_Val   : byte);
     tpErrReg     :
       (Error1,
        Error2,
        Error3   : byte);
     tpStatus     :
       (Sts_OnOff: boolean);
  );
end;

```

TSendRecord je typ variantního záznamu, který svou strukturou odpovídá datům protokolu FESTO posílaným Master i Slave stanicí ven z jednotky, přičemž je zbaven nadbytečností, které jsou při vysílání doplněny. Položka **Code** definuje základní typ vysílané zprávy.

V případě resetovací zprávy „Break Signal“ používá Slave stanice pro vysílání položku **ID_Str**, do které vloží identifikační řetězec daného zařízení. Master stanice používá tuto položku v případě přijaté odpovědi. Aby mohla Master stanice odpovědět na zprávu „Break Signal“ správně dekodovat, musí položka **ID_Str** obsahovat řetězec „FESTO“.

V případě zpráv pro čtení či zápis parametrů se používají dále tyto položky: položka **Index**, která obsahuje číslo čteného/zapisovaného parametru (v případě čtení/zápisu chybového registru se nepoužívá), položka **ModifyType**, která definuje typ zápisové zprávy v případě zprávy pro zápis parametrů (v případě zprávy pro čtení parametrů se nevyužívá) a položka **Param**, která definuje typ čteného či zapisovaného parametru. Podle typu čteného či zapisovaného parametru se používají dále některé další položky. Pro čtení/zápis jednoho příznaku nebo digitálního vstupu či výstupu se používá položka **IOF_Idx**, která udává bitový index parametru, a položka **IOF_Val**, která obsahuje hodnotu čteného/zapisovaného parametru (tuto položku nastavuje v případě čtení Slave stanice a v případě zápisu Master stanice a podobně je tomu u položek obsahujících hodnotu dalších čtených či zapisovaných parametrů). Pro čtení/zápis slova (16-ti) příznaků nebo digitálních vstupů či výstupů se používá položka **IOF_ValW**, která obsahuje hodnotu čteného/zapisovaného parametru. Pro čtení/zápis příznaku spuštění daného časovače se používá položka **Tm_OnOff**, pro čtení/zápis předvolby daného časovače se používá položka **Tm_Pre** a pro čtení/zápis aktuální hodnoty daného časovače se používá položka **Tm_Val**. Pro čtení/zápis příznaku spuštění daného čítače se používá položka **Ct_OnOff**, pro čtení/zápis předvolby daného čítače se používá položka **Ct_Pre** a pro čtení/zápis aktuální hodnoty daného čítače se používá položka **Ct_Val**. Pro čtení/zápis hodnoty registru se používá položka **Reg_Val**. Pro čtení/zápis hodnoty chybového registru se používají položky **Error1**, **Error2** a **Error3**. Pro čtení/zápis příznaku spuštění daného procesu se používá položka **Sts_OnOff**.

```
pRecRecord = ^tRecRecord;  
tRecRecord = tSendRecord;
```

TRecRecord je typ variantního záznamu, který svou strukturou odpovídá datům protokolu FESTO přijímaným Master i Slave stanicí, přičemž je zbaven nadbytečností, které jsou při příjmu odstraněny. Svou vnitřní strukturou je shodný s typem **TSendRecord**.

5. Objekty

5.1. tChnFesto

5.1.1. Položky

CH_RTICK : Boolean;

Položka **CH_RTICK** označuje, že je vykonávána činnost přijímacího automatu. Tato položka se používá pro ladění.

CH_SBuff : Pointer;

Položka **CH_SBuff** definuje ukazatel na vysílací buffer.

CH_MSBuff : Word;

Položka **CH_MSBuff** definuje délku vysílacího bufferu.

CH_LRMess : Word;

Položka **CH_LRMess** definuje délku přijímané zprávy.

CH_Master : Boolean;

Položka **CH_Master** definuje, je-li stanice zapojena v síti jako nadřazená jednotka (Master) nebo jako podřízená jednotka (Slave).

CH_SendLF : Boolean;

Položka **CH_SendLF** určuje, zda-li se má za řídicím znakem **zCR** vysílat i řídicí znak **zLF**.

5.1.2. Metody

5.1.2.1. Init konstruktor

constructor Init;

Konstruktor **Init** slouží k vytvoření a inicializaci instance komunikačního objektu. Ve svém těle zavolá zděděný konstruktor **Init** (inherited Init) od rodičovského objektu tChnVirt a inicializuje položky objektu. Tělo konstruktoru vypadá následovně:

```
inherited Init;
CH_Type      := cName;
CH_Name      := CH_Type;
CH_NumName   := ChNumName(CH_Type);
CH_RTick     := false;
CH_SBuff     := nil;
CH_MSBuff    := 0;
CH_LRMess    := 0;
CH_Master    := true;
CH_SendLF    := false;
CH_ss        := '';
CH_ii        := 1;
```

5.1.2.2. ChInitParam konstruktor

constructor ChInitParam(const S: tParamStr);

Konstruktor **ChInitParam** je sloučením konstruktoru **Init** a metody **ChSetParam**. Slouží ke zkrácenému vytvoření instance komunikačního objektu s nastavením parametrů komunikace.

5.1.2.3. Done destruktork

destructor Done;

Destruktork **Done** slouží ke zrušení instance komunikačního objektu. Pokud je alokovan vysílací buffer, je odstraněn z paměti. Na konci destruktork je volána zděděná metoda **Done** od přímého rodičovského objektu pro uzavření podřízené komunikační vrstvy.

5.1.2.4. ChSetOneParam funkce

```
function ChSetOneParam(const S: tWordString; var CmdL: tCmd)
: tChResult;
```

Metoda **ChSetOneParam** slouží k dekodování a nastavení jednoho konkrétního parametru, který je zadán v parametru S. Tato metoda se volá v aplikaci prostřednictvím metody **ChSetParam**. Metoda **ChSetOneParam** komunikačního objektu tChnFesto dekoduje tyto parametry:

MAS=MASTER / SLAVE

Parametrem **MAS** ("Master or Slave") se určuje, zda je jednotka v komunikační síti jako Master (nadržena) nebo jako Slave (podřizena).

LSB=Size

Parametrem **LSB** ("Length of Send Buffer") je alokován nový vysílací buffer **CH_MSBuff** dané velikosti Size.

SLF=ON/OFF

Parametrem **SLF** ("Sending LF") se určí příznak **CH_SendLF**, který udává, že se bude za znakem zCR vysílat i znak zLF.

5.1.2.5. ChGetParam funkce

```
function ChGetParam(const S: TParamStr): TParamStr;
```

Metoda **ChGetParam** navrácí nastavené hodnoty parametrů komunikačního objektu. Nejprve vrátí nastavení parametrů rodičovského komunikačního objektu tChnVirt a poté k nim připojí seznam svých parametrů. Seznam parametrů je uveden výše u popisu metody **ChSetOneParam**.

5.1.2.6. ChConnect procedura

```
procedure ChConnect;
```

Metoda **ChConnect** zavolá zděděnou metodu **ChConnect** od přímého rodičovského objektu a pokud nenastala žádná chyba, nastaví automat přijímače **CH_RCtrl** do počátečního stavu pro příjem zprávy v protokolu FESTO.

5.1.2.7. ChDisconnect procedura

```
procedure ChDisconnect;
```

Metoda **ChDisconnect** zavolá zděděnou metodu **ChDisconnect** od přímého rodičovského objektu a pokud nenastala žádná chyba, nastaví automat přijímače **CH_RCtrl** do neaktivního stavu, aby se nepřijímaly žádné zprávy.

5.1.2.8. ChSend procedura

```
procedure ChSend(Buff : Pointer; Len : Word);
```

Metoda **ChSend** způsobí započetí vysílání zprávy podle protokolu FESTO na podkladu záznamu typu tSendRecord, na který ukazuje parametr Buff. Parametr Len udává délku vysílacího bufferu pro vysílání. Tento parametr není nutno správně zadat, jelikož protokol si umí délku vysílané zprávy spočítat sám na základě nastavených položek záznamu.

5.1.2.9. ChReceiveReady funkce

```
function ChReceiveReady: tChState;
```

Metoda **ChReceiveReady** způsobí provedení kroku přijímacího automatu na základě volání metody **ChReceiveTick**. Jako svoji funkční hodnotu vrací aktuální stav automatu přijímače komunikačního kanálu, který je uložen v položce **CH_RCtrl**. Zpravidla se provádí test pouze na stabilní stav **CHS_ReceiveReady** (který znamená, že byla přijata nějaká zpráva), protože ostatní stavy jsou stavy probíhajícího příjmu.

5.1.2.10. ChReceive procedura

```
procedure ChReceive(var Len: Word);
```

Metoda **ChReceive** provede přijetí celé zprávy a její uložení do přijímacího bufferu, který svou vnitřní strukturou odpovídá datům záznamu typu `tRecRecord` a byl definován metodou **ChReceiveBuffer**. Metoda naplní buffer pouze patřičnými položkami typu `tRecRecord`, úvodní a zakončovací řídicí znaky ze zprávy metoda vyhodnotí a pro uživatele odstraní. V parametru `Len` vrátí velikost přijaté zprávy. Odpověď na zprávu, ať došla v pořádku nebo porušená, generuje uživatel sám pomocí metody **ChSend**.

5.1.2.11. ChReceiveFlush procedura

```
procedure ChReceiveFlush;
```

Metoda **ChReceiveFlush** způsobí vyprázdnění přijímacích bufferů a nastavení stavu automatu přijímače na počátek příjmu zpráv v protokolu FESTO.

5.1.2.12. ChReceiveTick procedura

```
procedure ChReceiveTick;
```

Metoda **ChReceiveTick** způsobí provedení jednoho či více kroků automatu přijímače. Je nutné ji periodicky volat při přijímání. Metoda **ChReceiveTick** je rovněž automaticky volána v metodě **ChReceiveReady**.

5.2. tAddChnFesto

Typ **tAddChnFesto** je typem objektu, který slouží k definování prvku v seznamu správců komunikačních objektů (tzv. správce komunikačního objektu `tChnFesto` v seznamu správců). Objekt `tAddChnFesto` je dědicem od rodičovského objektu **tAddChnVirt**.

5.2.1. Metody

5.2.1.1. ChInit funkce

```
function ChInit: pChnVirt;
```

Metoda **ChInit** slouží k vytvoření instance komunikačního objektu `tChnFesto` a ukazatel na instanci tohoto objektu vrací jako svoji funkční hodnotu.

6. Struktura zpráv protokolu FESTO

Jelikož je protokol FESTO převážně textový (nikoliv binární), kódují se do něj datové typy parametrů následovně:

- Boolean - kóduje se na znak '0' (False) nebo '1' (True)
- Byte - kóduje se na řetězec dekadických číslic (např. '125')
- Word - kóduje se na řetězec dekadických číslic, ale položky typu `word` se přetypovávají na `integer` a naopak:
např. \$FFFF se konvertuje na '-1', \$8000 na '-32768', \$0100 na '256' apod.

6.1. Struktura resetovací zprávy „Break Signal“

Tuto zprávu vysílá pouze Master stanice a jak již bylo řečeno v úvodu tohoto dokumentu, vyslání celé zprávy spočívá v hardwarovém vyslání Break signálu po komunikační lince. Slave stanice na tento signál odpovídá zprávou:

zCR,(zLF)	ID_Str	zCR,(zLF)	'>'	zEM
-----------	--------	-----------	-----	-----

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Slave stanice:

```
Code = tcBreak;
ID_Str = 'FESTO ...';
```

6.2. Struktura zprávy „Run“ pro spuštění programu

Master stanice vysílá zprávu:

'R'	zCR,(zLF)
-----	-----------

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Master stanice:

```
Code = tcRun;
```

Slave stanice odpovídá zprávou:

'R'	zCR,(zLF)	'>'	zEM
-----	-----------	-----	-----

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Slave stanice:

```
Code = tcRun;
```

6.3. Struktura zprávy „Stop“ pro zastavení programu

Master stanice vysílá zprávu:

'S'	zCR,(zLF)
-----	-----------

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Master stanice:

```
Code = tcStop;
```

Slave stanice odpovídá zprávou:

'S'	zCR,(zLF)	'>'	zEM
-----	-----------	-----	-----

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Slave stanice:

```
Code = tcStop;
```

6.4. Struktura zpráv pro čtení parametru

6.4.1. Struktura zprávy pro čtení jednoho digitálního vstupu

Master stanice vysílá zprávu:

'D'	'E'	Index	'.'	IOF_Indx	zCR,(zLF)
-----	-----	-------	-----	----------	-----------

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Master stanice:

```
Code      = tcDisp;
Index     = např. 0;
Param     = tpDIn;
IOF_Indx  = např. 0;
```

Slave stanice odpovídá zprávou:

'D'	'E'	Index	'.'	IOF_Indx	'='	IOF_Val	zCR,(zLF)	'>'	zEM
-----	-----	-------	-----	----------	-----	---------	-----------	-----	-----

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Slave stanice:

```
Code      = tcDisp;
Index     = 0;
Param     = tpDIn;
IOF_Indx  = 0;
IOF_Val   = např. True;
```

6.4.2. Struktura zprávy pro čtení jednoho digitálního výstupu

Master stanice vysílá zprávu:

'D'	'A'	Index	'.'	IOF_Indx	zCR,(zLF)
-----	-----	-------	-----	----------	-----------

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Master stanice:

```
Code      = tcDisp;
Index     = např. 0;
Param     = tpDOut;
IOF_Indx  = např. 0;
```

Slave stanice odpovídá zprávou:

'D'	'A'	Index	'.'	IOF_Indx	'='	IOF_Val	zCR,(zLF)	'>'	zEM
-----	-----	-------	-----	----------	-----	---------	-----------	-----	-----

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Slave stanice:

```
Code      = tcDisp;
Index     = 0;
Param     = tpDOut;
IOF_Indx  = 0;
IOF_Val   = např. True;
```

6.4.3. Struktura zprávy pro čtení jednoho příznaku

Master stanice vysílá zprávu:

'D'	'M'	Index	'.'	IOF_Indx	zCR,(zLF)
-----	-----	-------	-----	----------	-----------

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Master stanice:

```
Code      = tcDisp;
Index     = např. 0;
Param     = tpFlag;
IOF_Indx = např. 0;
```

Slave stanice odpovídá zprávou:

'D'	'M'	Index	'.'	IOF_Indx	'='	IOF_Val	zCR,(zLF)	'>'	zEM
-----	-----	-------	-----	----------	-----	---------	-----------	-----	-----

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Slave stanice:

```
Code      = tcDisp;
Index     = 0;
Param     = tpFlag;
IOF_Indx = 0;
IOF_Val   = např. True;
```

6.4.4. Struktura zprávy pro čtení slova (16-ti) digitálních vstupů

Master stanice vysílá zprávu:

'D'	'EW'	Index	zCR,(zLF)
-----	------	-------	-----------

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Master stanice:

```
Code      = tcDisp;
Index     = např. 0;
Param     = tpDIns;
```

Slave stanice odpovídá zprávou:

'D'	'EW'	Index	'='	IOF_ValW	zCR,(zLF)	'>'	zEM
-----	------	-------	-----	----------	-----------	-----	-----

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Slave stanice:

```
Code      = tcDisp;
Index     = 0;
Param     = tpDIn;
IOF_ValW = např. $8000;
```

6.4.5. Struktura zprávy pro čtení slova (16-ti) digitálních výstupů

Master stanice vysílá zprávu:

'D'	'AW'	Index	zCR,(zLF)
-----	------	-------	-----------

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Master stanice:

```
Code      = tcDisp;
```


Index = např. 0;
Param = tpDOuts;

Slave stanice odpovídá zprávou:

'D'	'AW'	Index	'='	IOF_ValW	zCR,(zLF)	'>'	zEM
-----	------	-------	-----	----------	-----------	-----	-----

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Slave stanice:

Code = tcDisp;
Index = 0;
Param = tpDOuts;
IOF_ValW = např. \$8000;

6.4.6. Struktura zprávy pro čtení slova (16-ti) příznaků

Master stanice vysílá zprávu:

'D'	'MW'	Index	zCR,(zLF)
-----	------	-------	-----------

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Master stanice:

Code = tcDisp;
Index = např. 0;
Param = tpFlags;

Slave stanice odpovídá zprávou:

'D'	'MW'	Index	'='	IOF_ValW	zCR,(zLF)	'>'	zEM
-----	------	-------	-----	----------	-----------	-----	-----

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Slave stanice:

Code = tcDisp;
Index = 0;
Param = tpFlags;
IOF_ValW = např. \$8000;

6.4.7. Struktura zprávy pro čtení příznaku spuštění časovače

Master stanice vysílá zprávu:

'D'	'T'	Index	zCR,(zLF)
-----	-----	-------	-----------

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Master stanice:

Code = tcDisp;
Index = např. 0;
Param = tpTimer;

Slave stanice odpovídá zprávou:

'D'	'T'	Index	'='	Tm_OnOff	zCR,(zLF)	'>'	zEM
-----	-----	-------	-----	----------	-----------	-----	-----

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Slave stanice:

Code = tcDisp;
Index = 0;

```
Param    = tpTimer;
Tm_OnOff = např. True;
```

6.4.8. Struktura zprávy pro čtení předvolby časovače

Master stanice vysílá zprávu:

'D'	'TV'	Index	zCR,(zLF)
-----	------	-------	-----------

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Master stanice:

```
Code     = tcDisp;
Index    = např. 0;
Param    = tpTimerPre;
```

Slave stanice odpovídá zprávou:

'D'	'TV'	Index	'='	Tm_Pre	zCR,(zLF)	'>'	zEM
-----	------	-------	-----	--------	-----------	-----	-----

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Slave stanice:

```
Code     = tcDisp;
Index    = 0;
Param    = tpTimerPre;
Tm_Pre   = např. 200;
```

6.4.9. Struktura zprávy pro čtení aktuální hodnoty časovače

Master stanice vysílá zprávu:

'D'	'TW'	Index	zCR,(zLF)
-----	------	-------	-----------

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Master stanice:

```
Code     = tcDisp;
Index    = např. 0;
Param    = tpTimerVal;
```

Slave stanice odpovídá zprávou:

'D'	'TW'	Index	'='	Tm_Val	zCR,(zLF)	'>'	zEM
-----	------	-------	-----	--------	-----------	-----	-----

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Slave stanice:

```
Code     = tcDisp;
Index    = 0;
Param    = tpTimerVal;
Tm_Val   = např. 123;
```

6.4.10. Struktura zprávy pro čtení příznaku spuštění čítače

Master stanice vysílá zprávu:

'D'	'Z'	Index	zCR,(zLF)
-----	-----	-------	-----------

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Master stanice:

```
Code    = tcDisp;
Index   = např. 0;
Param   = tpCounter;
```

Slave stanice odpovídá zprávou:

'D'	'Z'	Index	'='	Ct_OnOff	zCR,(zLF)	'>'	zEM
-----	-----	-------	-----	----------	-----------	-----	-----

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Slave stanice:

```
Code    = tcDisp;
Index   = 0;
Param   = tpCounter;
Ct_OnOff = např. True;
```

6.4.11. Struktura zprávy pro čtení předvolby čítače

Master stanice vysílá zprávu:

'D'	'ZV'	Index	zCR,(zLF)
-----	------	-------	-----------

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Master stanice:

```
Code    = tcDisp;
Index   = např. 0;
Param   = tpCounterPre;
```

Slave stanice odpovídá zprávou:

'D'	'ZV'	Index	'='	Ct_Pre	zCR,(zLF)	'>'	zEM
-----	------	-------	-----	--------	-----------	-----	-----

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Slave stanice:

```
Code    = tcDisp;
Index   = 0;
Param   = tpCounterPre;
Ct_Pre  = např. 200;
```

6.4.12. Struktura zprávy pro čtení aktuální hodnoty čítače

Master stanice vysílá zprávu:

'D'	'ZW'	Index	zCR,(zLF)
-----	------	-------	-----------

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Master stanice:

```
Code    = tcDisp;
Index   = např. 0;
Param   = tpCounterVal;
```

Slave stanice odpovídá zprávou:

'D'	'ZW'	Index	'='	Ct_Val	zCR,(zLF)	'>'	zEM
-----	------	-------	-----	--------	-----------	-----	-----

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Slave stanice:

```
Code    = tcDisp;
```

```

Index    = 0;
Param    = tpCounterVal;
Ct_Val   = např. 123;

```

6.4.13. Struktura zprávy pro čtení registru

Master stanice vysílá zprávu:

'D'	'R'	Index	zCR,(zLF)
-----	-----	-------	-----------

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Master stanice:

```

Code     = tcDisp;
Index    = např. 0;
Param    = tpReg;

```

Slave stanice odpovídá zprávou:

'D'	'R'	Index	'='	Reg_Val	zCR,(zLF)	'>'	zEM
-----	-----	-------	-----	---------	-----------	-----	-----

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Slave stanice:

```

Code     = tcDisp;
Index    = 0;
Param    = tpReg;
Reg_Val  = např. 100;

```

6.4.14. Struktura zprávy pro čtení chybového registru

Master stanice vysílá zprávu:

'D'	'F'	zCR,(zLF)
-----	-----	-----------

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Master stanice:

```

Code     = tcDisp;
Index    = např. 0;
Param    = tpErrReg;

```

Slave stanice odpovídá zprávou:

'D'	'F'	'='	Error1	','	Error2	','	Error3	zCR,(zLF)	'>'	zEM
-----	-----	-----	--------	-----	--------	-----	--------	-----------	-----	-----

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Slave stanice:

```

Code     = tcDisp;
Index    = 0;
Param    = tpErrReg;
Error1   = např. 0;
Error2   = např. 0;
Error3   = např. 0;

```

6.4.15. Struktura zprávy pro čtení příznaku spuštění procesu

Master stanice vysílá zprávu:

'D'	'S'	Index	zCR,(zLF)
-----	-----	-------	-----------

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Master stanice:

```
Code      = tcDisp;
Index     = např. 0;
Param     = tpStatus;
```

Slave stanice odpovídá zprávou:

'D'	'S'	Index	'='	Sts_OnOff	zCR,(zLF)	'>'	zEM
-----	-----	-------	-----	-----------	-----------	-----	-----

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Slave stanice:

```
Code      = tcDisp;
Index     = 0;
Param     = tpStatus;
Sts_OnOff= např. False;
```

6.5. Struktura zpráv pro zápis parametru

Jak již bylo řečeno výše existují dva způsoby, jak zapsat daný parametr. Nejběžnějším způsobem je spolu s příkazem pro zápis poslat i zapisovaná data (tomu odpovídá typ **tmWhole**). Druhým způsobem je, že Master stanice pošle nejdříve pouze příkaz na zápis parametru, Slave stanice vrátí aktuální hodnotu parametru a Master stanice poté pošle pouze nová data (tomu odpovídají typy **tmOnlyCmd** a **tmOnlyVal**). Druhý (méně častý) způsob si ukážeme pouze na zápisu jednoho digitálního výstupu, jelikož u zápisu ostatních parametrů je obdobný.

6.5.1. Struktura zprávy pro zápis jednoho digitálního výstupu

Vyslání zapisovaných dat společně s příkazem

Master stanice vysílá zprávu s příkazem i daty:

'M'	'A'	Index	'.'	IOF_Indx	'='	IOF_Val	zCR,(zLF)
-----	-----	-------	-----	----------	-----	---------	-----------

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Master stanice:

```
Code      = tcModify;
ModifyType = tmWhole;
Index     = např. 0;
Param     = tpDOut;
IOF_Indx  = např. 0;
IOF_Val   = např. True;
```

Slave stanice odpovídá zprávou:

'M'	'A'	Index	'.'	IOF_Indx	'='	IOF_Val	zCR,(zLF)	'>'	zEM
-----	-----	-------	-----	----------	-----	---------	-----------	-----	-----

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Slave stanice:

```
Code      = tcModify;
ModifyType = tmWhole;
Index     = 0;
Param     = tpDOut;
IOF_Indx  = 0;
IOF_Val   = True;
```

Vyslání nejdříve příkazu a potom zapisovaných dat

Master stanice vysílá zprávu pouze s příkazem:

'M'	'A'	Index	'.'	IOF_Indx	zCR,(zLF)
-----	-----	-------	-----	----------	-----------

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Master stanice:

```
Code      = tcModify;
ModifyType = tmOnlyCmd;
Index     = např. 0;
Param    = tpDOut;
IOF_Indx = např. 0;
```

Slave stanice odpovídá zprávou:

'M'	'A'	Index	'.'	IOF_Indx	'='	IOF_Val	'.'	zEM
-----	-----	-------	-----	----------	-----	---------	-----	-----

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Slave stanice:

```
Code      = tcModify;
ModifyType = tmOnlyCmd;
Index     = 0;
Param    = tpDOut;
IOF_Indx = 0;
IOF_Val  = např. False;
```

Master stanice potom musí vyslat zprávu pouze s daty:

IOF_Val	zCR,(zLF)
---------	-----------

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Master stanice:

```
Code      = tcModify;
ModifyType = tmOnlyVal;
Param    = tpDOut;
IOF_Val  = např. True;
```

Slave stanice odpovídá zprávou:

IOF_Val	'>'	zEM
---------	-----	-----

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Slave stanice:

```
Code      = tcModify;
ModifyType = tmOnlyVal;
Param    = tpDOut;
IOF_Val  = True;
```

6.5.2. Struktura zprávy pro zápis jednoho digitálního vstupu, slova (16-ti) digitálních vstupů a chybového registru

Tyto zprávu asi nemá smysl nikdy použít, nicméně protokol to dovoluje a zařízení PLC vrátí platnou odpověď. Strukturu této zprávy zde však uvádět nebudeme, jelikož je hodně podobná zprávě pro zápis jednoho digitálního výstupu.

6.5.3. Struktura zprávy pro zápis jednoho příznaku

Master stanice vysílá zprávu s příkazem i daty:

'M'	'M'	Index	'.'	IOF_Indx	'='	IOF_Val	zCR,(zLF)
-----	-----	-------	-----	----------	-----	---------	-----------

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Master stanice:

```
Code      = tcModify;
ModifyType = tmWhole;
Index     = např. 0;
Param     = tpFlag;
IOF_Indx  = např. 0;
IOF_Val   = např. True;
```

Slave stanice odpovídá zprávou:

'M'	'M'	Index	'.'	IOF_Indx	'='	IOF_Val	zCR,(zLF)	'>'	zEM
-----	-----	-------	-----	----------	-----	---------	-----------	-----	-----

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Slave stanice:

```
Code      = tcModify;
ModifyType = tmWhole;
Index     = 0;
Param     = tpFlag;
IOF_Indx  = 0;
IOF_Val   = True;
```

6.5.4. Struktura zprávy pro zápis slova (16-ti) digitálních výstupů

Master stanice vysílá zprávu s příkazem i daty:

'M'	'AW'	Index	'='	IOF_ValW	zCR,(zLF)
-----	------	-------	-----	----------	-----------

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Master stanice:

```
Code      = tcModify;
ModifyType = tmWhole;
Index     = např. 0;
Param     = tpDOuts;
IOF_ValW  = např. $0100;
```

Slave stanice odpovídá zprávou:

'M'	'AW'	Index	'='	IOF_ValW	zCR,(zLF)	'>'	zEM
-----	------	-------	-----	----------	-----------	-----	-----

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Slave stanice:

```
Code      = tcModify;
ModifyType = tmWhole;
Index     = 0;
Param     = tpDOuts;
IOF_ValW  = $0100;
```

6.5.5. Struktura zprávy pro zápis slova (16-ti) příznaků

Master stanice vysílá zprávu s příkazem i daty:

'M'	'MW'	Index	'='	IOF_ValW	zCR,(zLF)
-----	------	-------	-----	----------	-----------

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Master stanice:

```
Code      = tcModify;
ModifyType = tmWhole;
Index     = např. 0;
Param     = tpFlags;
IOF_ValW  = např. $AAAA;
```

Slave stanice odpovídá zprávou:

'M'	'MW'	Index	'='	IOF_ValW	zCR,(zLF)	'>'	zEM
-----	------	-------	-----	----------	-----------	-----	-----

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Slave stanice:

```
Code      = tcModify;
ModifyType = tmWhole;
Index     = 0;
Param     = tpFlags;
IOF_ValW  = $AAAA;
```

6.5.6. Struktura zprávy pro zápis příznaku spuštění časovače

Master stanice vysílá zprávu s příkazem i daty:

'M'	'T'	Index	'='	Tm_OnOff	zCR,(zLF)
-----	-----	-------	-----	----------	-----------

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Master stanice:

```
Code      = tcModify;
ModifyType = tmWhole;
Index     = např. 0;
Param     = tpTimer;
Tm_OnOff  = např. True;
```

Slave stanice odpovídá zprávou:

'M'	'T'	Index	'='	Tm_OnOff	zCR,(zLF)	'>'	zEM
-----	-----	-------	-----	----------	-----------	-----	-----

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Slave stanice:

```
Code      = tcModify;
ModifyType = tmWhole;
Index     = 0;
Param     = tpTimer;
Tm_OnOff  = True;
```


6.5.7. Struktura zprávy pro zápis předvolby časovače

Master stanice vysílá zprávu s příkazem i daty:

'M'	'TV'	Index	'='	Tm_Pre	zCR,(zLF)
-----	------	-------	-----	--------	-----------

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Master stanice:

```
Code      = tcModify;
ModifyType = tmWhole;
Index     = např. 0;
Param     = tpTimerPre;
Tm_Pre    = např. 200;
```

Slave stanice odpovídá zprávou:

'M'	'TV'	Index	'='	Tm_Pre	zCR,(zLF)	'>'	zEM
-----	------	-------	-----	--------	-----------	-----	-----

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Slave stanice:

```
Code      = tcModify;
ModifyType = tmWhole;
Index     = 0;
Param     = tpTimerPre;
Tm_Pre    = 200;
```

6.5.8. Struktura zprávy pro zápis aktuální hodnoty časovače

Master stanice vysílá zprávu s příkazem i daty:

'M'	'TW'	Index	'='	Tm_Val	zCR,(zLF)
-----	------	-------	-----	--------	-----------

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Master stanice:

```
Code      = tcModify;
ModifyType = tmWhole;
Index     = např. 0;
Param     = tpTimerVal;
Tm_Val    = např. 10;
```

Slave stanice odpovídá zprávou:

'M'	'TW'	Index	'='	Tm_Val	zCR,(zLF)	'>'	zEM
-----	------	-------	-----	--------	-----------	-----	-----

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Slave stanice:

```
Code      = tcModify;
ModifyType = tmWhole;
Index     = 0;
Param     = tpTimerVal;
Tm_Val    = 10;
```

6.5.9. Struktura zprávy pro zápis příznaku spuštění čítače

Master stanice vysílá zprávu s příkazem i daty:

'M'	'Z'	Index	'='	Ct_OnOff	zCR,(zLF)
-----	-----	-------	-----	----------	-----------

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Master stanice:

```
Code      = tcModify;
ModifyType = tmWhole;
Index     = např. 0;
Param     = tpCounter;
Ct_OnOff  = např. True;
```

Slave stanice odpovídá zprávou:

'M'	'Z'	Index	'='	Ct_OnOff	zCR,(zLF)	'>'	zEM
-----	-----	-------	-----	----------	-----------	-----	-----

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Slave stanice:

```
Code      = tcModify;
ModifyType = tmWhole;
Index     = 0;
Param     = tpcounter;
Ct_OnOff  = True;
```

6.5.10. Struktura zprávy pro zápis předvolby čítače

Master stanice vysílá zprávu s příkazem i daty:

'M'	'ZV'	Index	'='	Ct_Pre	zCR,(zLF)
-----	------	-------	-----	--------	-----------

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Master stanice:

```
Code      = tcModify;
ModifyType = tmWhole;
Index     = např. 0;
Param     = tpCounterPre;
Ct_Pre    = např. 200;
```

Slave stanice odpovídá zprávou:

'M'	'ZV'	Index	'='	Ct_Pre	zCR,(zLF)	'>'	zEM
-----	------	-------	-----	--------	-----------	-----	-----

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Slave stanice:

```
Code      = tcModify;
ModifyType = tmWhole;
Index     = 0;
Param     = tpCounterPre;
Ct_Pre    = 200;
```

6.5.11. Struktura zprávy pro zápis aktuální hodnoty čítače

Master stanice vysílá zprávu s příkazem i daty:

'M'	'ZW'	Index	'='	Ct_Val	zCR,(zLF)
-----	------	-------	-----	--------	-----------

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Master stanice:

```
Code      = tcModify;
ModifyType = tmWhole;
Index     = např. 0;
Param     = tpCounterVal;
Ct_Val    = např. 10;
```

Slave stanice odpovídá zprávou:

'M'	'ZW'	Index	'='	Ct_Val	zCR,(zLF)	'>'	zEM
-----	------	-------	-----	--------	-----------	-----	-----

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Slave stanice:

```
Code      = tcModify;
ModifyType = tmWhole;
Index     = 0;
Param     = tpCounterVal;
Ct_Val    = 10;
```

6.5.12. Struktura zprávy pro zápis hodnoty registru

Master stanice vysílá zprávu s příkazem i daty:

'M'	'R'	Index	'='	Reg_Val	zCR,(zLF)
-----	-----	-------	-----	---------	-----------

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Master stanice:

```
Code      = tcModify;
ModifyType = tmWhole;
Index     = např. 0;
Param     = tpReg;
Reg_Val   = např. 123;
```

Slave stanice odpovídá zprávou:

'M'	'R'	Index	'='	Reg_Val	zCR,(zLF)	'>'	zEM
-----	-----	-------	-----	---------	-----------	-----	-----

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Slave stanice:

```
Code      = tcModify;
ModifyType = tmWhole;
Index     = 0;
Param     = tpReg;
Reg_Val   = 123;
```

6.5.13. Struktura zprávy pro zápis příznaku spuštění procesu

Master stanice vysílá zprávu s příkazem i daty:

'M'	'S'	Index	'='	Sts_OnOff	zCR,(zLF)
-----	-----	-------	-----	-----------	-----------

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Master stanice:

```
Code      = tcModify;
ModifyType = tmWhole;
Index     = např. 0;
Param     = tpStatus;
Sts_OnOff = např. False;
```

Slave stanice odpovídá zprávou:

'M'	'S'	Index	'='	Sts_OnOff	zCR,(zLF)	'>'	zEM
-----	-----	-------	-----	-----------	-----------	-----	-----

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Slave stanice:

```
Code      = tcModify;
ModifyType = tmWhole;
Index     = 0;
Param     = tpStatus;
Sts_OnOff = False;
```

6.6. Struktura zprávy „Acces Error“ od Slave stanice

Pokud Slave stanice (zařízení PLC FESTO) nebylo schopno zprávu od Master stanice správně dekodovat, odpovídá právě touto zprávou.

Master stanice posílá nějakou nesmyslnou zprávu:

nesmysly	zCR,(zLF)
----------	-----------

Slave stanice odpovídá zprávou:

nesmysly	zEr	zCR,(zLF)	'ACCESS ERROR'	zCR,(zLF)	'>'	zEM
----------	-----	-----------	----------------	-----------	-----	-----

Tomu odpovídají tyto nastavené položky ve struktuře vysílacího bufferu Slave stanice:

```
Code      = tcAcsErr;
```

7. Příklad

Následující příklad ukazuje způsob vyslání Break signálu a poté zprávy na čtení digitálních vstupů do PLC. Fyzický přenos se realizuje prostřednictvím knihovny ChnComBR.

```
uses
  NumToStr,
  uString,
  ChnVirt,
  ChnComBR,
  ChnFesto;
```

```

const
  ParamStr : tParamStr =
    'NAM=FESTO LSB=500 SLF=OFF ' +
    'NAM=COMBR COM=1 IRQ=4 BD=9600 BIT=8 STOP=1 ' +
    'PAR=E LRB=1000 BRK=0';

var
  Chn      : pChnVirt;
  SMess    : pSendRecord;
  RMess    : pRecRecord;
  LSMess   : word;
  LRMess   : word;

begin
  ...
  New(SMess);
  New(RMess);
  ...
  { vytvoření instance Chn }
  Chn:=ChnCollection^.ChNewInit(ChnFesto.cName);
  with Chn^ do
  begin
    { nastavení parametrů komunikace }
    ChSetParam(ParamStr);
    if ChResult<>res_Ok then WriteLn('Chyba');
    ChOpen;
    repeat
      if ChResult<>res_Ok then WriteLn('Chyba');
    until ChReady=CHS_Open;
    if ChResult<>res_Ok then WriteLn('Chyba');
    { definování místa, kam se má přijatá zpráva uložit }
    ChReceiveBuffer(RMess,SizeOf(RMess^));
    if ChReceiveResult<>res_Ok then WriteLn('Chyba');
    ChConnect;
    repeat
      if ChResult<>res_Ok then WriteLn('Chyba');
    until ChReady=CHS_Connect;
    if ChResult<>res_Ok then WriteLn('Chyba');
    ...
    { vyslání signálu Break pomocí knihovny ChnComBR }
    ChSetParam('| BRK=5');
    if ChResult<>res_Ok then WriteLn('Chyba');
    Smess^.Code := tcBreak;
    if ChSendReady=CHS_SendReady then
    begin
      ChSend(SMess, 0);
      { čekání na odvysílání zprávy }
      repeat
        if ChSendResult<>res_Ok then WriteLn('Chyba');
      until ChSendReady=CHS_SendReady;
      if ChSendResult<>res_Ok then WriteLn('Chyba');
      ...
    end;
    ChSetParam('| BRK=0');
    if ChResult<>res_Ok then WriteLn('Chyba');
    ...
    { čekání na příjem odpovědi na signál Break }
    while not ChReceiveReady=CHS_ReceiveReady do
    begin
      if ChReceiveResult<>res_Ok then WriteLn('Chyba');
    end;
    { příjem zprávy }
    ChReceive(LRMess);
    if ChReceiveResult<>res_Ok then WriteLn('Chyba')
    else
      { dekodování správné odpovědi }
      with RMess^ do

```

```

begin
  if Code = tcBreak then
    writeln('Ok ID_Str=',ID_Str)
  else
    writeln('Chyba');
end;
...
{ příklad naplnění zprávy daty (např. pro čtení dig. vstupů)}
with SMess^ do
begin
  Code := tcDisp;
  Index := 0;
  Param := tpDIns;
end;
{ vyslání zprávy }
if ChSendReady=CHS_SendReady then
begin
  ChSend(SMess, 0);
  { čekání na odvysílání zprávy }
  repeat
    if ChSendResult<>res_Ok then WriteLn('Chyba');
  until ChSendReady=CHS_SendReady;
  if ChSendResult<>res_Ok then WriteLn('Chyba');
  ...
end;
...
{ čekání na příjem zprávy }
while not ChReceiveReady=CHS_ReceiveReady do
begin
  if ChReceiveResult<>res_Ok then WriteLn('Chyba');
end;
{ příjem zprávy }
ChReceive(LRMess);
if ChReceiveResult<>res_Ok then WriteLn('Chyba')
else
  { dekódování správné odpovědi (např. odpověď na ReadData)}
  with RMess^ do
  begin
    if (Code = tcDisp) and
      (Index = 0) and
      (Param = tpDIns) then
      writeln('Ok dig.vstupy = $',WordStrHex(IOF_ValW))
    else
      writeln('Chyba');
  end;
  ...
  { ukončení }
  ChDisconnect;
  repeat
    if ChResult<>res_Ok then WriteLn('Chyba');
  until ChReady=CHS_DisConnect;
  if ChResult<>res_Ok then WriteLn('Chyba');
  ChClose;
  repeat
    if ChResult<>res_Ok then WriteLn('Chyba');
  until ChReady=CHS_Close;
  if ChResult<>res_Ok then WriteLn('Chyba');
end;
{ zrušení instance Chn }
Dispose(Chn,Done);
...
end.

```