

# ChnComBR

JEDNOTKA SÉRIOVÉ KOMUNIKACE  
RS232 / RS485 S OBVODEM I8250 S  
VYUŽITÍM BREAK INTERRUPTU

Příručka uživatele a programátora



**SofCon<sup>®</sup> spol. s r.o.**  
Střešovická 49  
162 00 Praha 6  
tel/fax: +420 220 180 454  
E-mail: [sofcon@sofcon.cz](mailto: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: 16.05.2003

Datum posledního uložení dokumentu: 16.05.2003

(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 jednoduché typy	6
4.1. Konstanty kódů pro metodu ChGetBinParam a ChSetBinParam	6
5.Objekty	7
5.1. tChnComBR	7
5.1.1. Položky	7
5.1.2. Metody	7
5.1.2.1. Init konstruktor	7
5.1.2.2. ChInitParam konstruktor	7
5.1.2.3. ChSetOneParam funkce	7
5.1.2.4. ChGetParam funkce	8
5.1.2.5. ChSetBinParam procedura	8
5.1.2.6. ChGetBinParam funkce	8
5.1.2.7. ChSendTick procedure	8
5.1.2.8. ChSendFlush procedura	8
5.1.2.9. ChReceiveChar funkce	9
5.2. tAddChnComBR	9
5.2.1. Metody	9
5.2.1.1. ChInit funkce	9
6.Příklad	9



## 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	08.04.2003	Úprava dokumentu dle ISO9000.
1.11	4.XX	We	16.05.2003	Doplněn popis fce ChSetBinParam, cmd_SetBrkDelay a upraveno ChGetBinParam.

### 1.2. Účel dokumentu

---

Tento dokument slouží jako popis jednotky sériové komunikace RS232 / RS485 s obvodem i8250 s využitím break interruptu.

### 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 rozhraní svých potomků, ChnCom a 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 ChnComBR definuje objekt **tChnComBR**, jehož instance vytváří fyzickou vrstvu k komunikačním kanálům tvořenou sériovým rozhraním RS232 nebo RS485 s obvodem i8250. Ke své činnosti využívá přerušovací systém počítače. Parametrem nastavovací metody **ChSetParam** lze určit vysílání Break Interruptu na začátku zprávy, což znamená, že před prvním vysílaným znakem se celá komunikační linka nastaví do logické úrovně 0 (reset linky) na stanovenou dobu. Při příjmu se tento stav zpětně dekóduje v podobě Break Detect. Těto vlastnosti využívají některé vyšší komunikační protokoly pro určení začátku zprávy. Pokud vysílání Break Interruptu není povoleno, chová se komunikační objekt tChnComBR stejně jako komunikační objekt tChnCom. Znaky přicházející po komunikační lince jsou v přerušovací proceduře ukládány do vstupního kruhového vyrovnávacího bufferu, z kterého jsou předávány metodami **ChReceive** a **ChReceiveChar** k dalšímu zpracování. Znaky určené k odeslání jsou zasílány z výstupního bufferu rovněž s využitím přerušovacího systému.

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

Protože je objekt **tChnComBR** dědicem rodičovského komunikačního objektu **tChnCom**, jsou v této příručce popsány jen odlišnosti a speciality pro tento druh sériové komunikace. Ostatní naleznete v příručce **ChnCom** a **ChnVirt**.

Některé použité konstanty a typy jsou předdefinované v jednotce **ChnTypes**.

### 4. Konstanty a jednoduché typy

---

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

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

```
cName = 'COMBR';
```

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

#### 4.1. Konstanty kódů pro metodu ChGetBinParam a ChSetBinParam

---

```
cmd_SetBrkDelay = $0201;
```

Kód pro číselné nastavení délky ve znacích pro Break Interrupt metodou SetBinParam.

```
cmd_GetBrkDelay = $0201;
```

Kód pro vrácení nastavené délky ve znacích pro Break Interrupt. Hodnota 0 znamená, že se Break Interrupt nevysílá.

```
cmd_GetBrkChar = $0202;
```

Kód pro vrácení statusu Break Detect přijatého znaku po volání metody ChReceiveChar.

---

## 5. Objekty

---

---

### 5.1. tChnComBR

---

#### 5.1.1. Položky

CH\_BrkChar : Boolean;

Položka **CH\_BrkChar** určuje status Break Detect posledního přijatého znaku metodou ChReceiveChar.

CH\_BrkDelay : Byte;

Položka **CH\_BrkDelay** definuje pauzu ve znacích pro Break Interrupt. Pokud její hodnota je rovna 0, nebude se Break Interrupt vysílat.

#### 5.1.2. Metody

##### 5.1.2.1. Init constructor

constructor Init;

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

```
inherited Init;
CH_Type      := cName;
CH_Name      := CH_Type;
CH_NumName   := ChNumName(CH_Type);
CH_NumNameParents := ChNumName(ChnCom.cName);
CH_BrkDelay  := 5;
CH_BrkChar   := False;
```

##### 5.1.2.2. ChInitParam constructor

constructor ChInitParam(const S: tParamStr);

Konstruktor **ChInitParam** slouží ke zkrácenému vytvoření instance komunikačního objektu s definovaným nastavením parametrů kanálu. Ve svém těle nejprve volá konstruktor **Init** a poté metodu **ChSetParam**.

##### 5.1.2.3. 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 tChnComBR dekoduje tyto parametry:

**BRK=aaa**

Parametr **BRK** "Length of Break Interrupt" určuje pauzu ve znacích pro Break Interrupt. aaa může nabývat hodnot 0 až 255. Pokud je hodnota aaa rovna 0, nebude se Break Interrupt vysílat.

Příklad:

Příklad ukazuje, jak je možné v jednotce COMBR nastavit parametry komunikace na sudou paritu, přenosovou rychlost 4800 Bd, velikost vstupního vyrovnávacího bufferu na 1000 položek a délku Break Interruptu na 2 znaky.

```
ChSetParam('NAM=COMBR PAR=E BD=4800 LRB=1000 BRK=2');
```

Pozn.: Všimněte si, že není volána metoda ChSetOneParam, ale metoda ChSetParam.

#### 5.1.2.4. ChGetParam funkce

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

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

#### 5.1.2.5. ChSetBinParam procedura

```
procedure ChSetBinParam(NumName: Word; Code: Word; Param: longint);
```

Metodou **ChSetBinParam** s parametrem NumName příslušným jménu tohoto komunikačního objektu, lze volbou parametru Code číselně nastavit tyto proměnné objektu:

```
Code = cmd_SetBrkDelay
```

Nastaví hodnotu pauzy ve znacích po Break Interrupt.

Příkazy s jiným Code přepouští svému rodičovskému objektu ChnCom.

#### 5.1.2.6. ChGetBinParam funkce

```
function ChGetBinParam(NumName: tChNumName; Code: Word): longint;
```

Metoda **ChGetBinParam** s parametrem NumName příslušným jménu tohoto komunikačního objektu, lze podle parametru Code vrátit tyto nastavení:

```
Code = cmd_GetBrkDelay
```

Vrátí nastavení pauzy ve znacích pro Break Interrupt.

```
Code = cmd_GetBrkChar
```

Vrátí příznak Break Detect posledního přijatého znaku metodou **ChReceiveChar**. Je-li vrácená hodnota = 0, potom nebyl přijat Break Detect. Je-li vrácená hodnota = 1, potom byl přijat Break Detect.

#### 5.1.2.7. ChSendTick procedure

```
procedure ChSendTick;
```

Metoda **ChSendTick** způsobí provedení kroků vysílacích automatů. Je nutné ji periodicky volat během vysílání. **ChSendTick** je rovněž automaticky volána v metodách **ChSendReady** a **ChSend**.

#### 5.1.2.8. ChSendFlush procedura

```
procedure ChSendFlush;
```

Pokud je kanál ve stavu **CHS\_Connect** způsobí metoda **ChSendFlush** ukončení vysílání a přechod automatu vysílače do stavu **CHS\_SendReady**.



### 5.1.2.9. ChReceiveChar funkce

```
function ChReceiveChar: Byte;
```

Pokud je kanál ve stavu **CHS\_Connect** a v přijímacím bufferu jsou přijata nějaká data, navrací metoda **ChReceiveChar** jeden přijatý znak z přijímacího bufferu. Pokud je pauza pro Break Interrupt větší než 0, dekóduje se status přijatého znaku na Break Detect. Podle tohoto testu se nastaví položka CH\_BrkChar a bit Break Detect se ze statusu odstraní. Výsledek operace přijímače se nastaví na výsledný status přijatého znaku. Metodu **ChReceiveChar** je možno volat pouze ve stavu automatu přijímače **CHS\_ReceiveReady**, proto před voláním této metody musí předcházet volání metody **ChReceiveReady** s testem na stav **CHS\_ReceiveReady**, jinak v případě nepřijetí žádného znaku skončí volání metody **ChReceiveChar** chybou.

## 5.2. tAddChnComBR

Typ **tAddChnComBR** je typem objektu, který slouží k definování prvku v seznamu správců komunikačních objektů (tzv. správce komunikačního objektu tChnComBR v seznamu správců). Objekt tAddChnComBR 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 tChnComBR a ukazatel na instanci tohoto objektu vrací jako svoji funkční hodnotu.

## 6. Příklad

Příklad ukazuje použití komunikační jednotky ChnComPB. Je vytvořen komunikační kanál definovaných vlastností, po kterém je zasílána zpráva a z kterého je poté očekáván příjem zpráv.

```
uses
  uString,
  ChnVirt,
  ChnCom,
  ChnComBR,
  ...
const
  ParamStr : tParamStr =
    'NAM=COMBR COM=1 IRQ=4 BD=1200 PAR=N BIT=8 STOP=2 '+
    'LRB=1000 BRK=5';
  LMess     = 40;

type
  tMess     = array [0..LMess+10] of Byte;

var
  Chn       : pChnVirt;
  SMess     : ^tMess;
  RMess     : ^tMess;
  LRMess    : Word;
  RecChar   : Byte;
```

```
begin
  ...
  New(SMess);
  New(RMess);
  ...
  { inicializace Chn }
  Chn:=ChnCollection^.ChNewInit(ChnComBR.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;
    { definování místa, kam se má přijatá zpráva uložit }
    ChReceiveBuffer(RMess,SizeOf(tMess));
    if ChReceiveResult<>res_Ok then WriteLn('Chyba');
    ChConnect;
    repeat
      if ChResult<>res_Ok then WriteLn('Chyba');
    until ChReady=CHS_Connect;
    ...
    { naplnění zprávy daty }
    ...
    { vyslání zprávy }
    if ChSendReady=CHS_SendReady then
    begin
      ChSend(SMess, LMess);
      repeat
        if ChSendResult<>res_Ok then WriteLn('Chyba');
      until ChSendReady=CHS_SendReady;
      if ChSendResult<>res_Ok then WriteLn('Chyba');
      ...
    end;
    ...
    { čekání na příjem zprávy }
    repeat
    until ChReceiveReady=CHS_ReceiveReady;
    { příjem zprávy }
    while ChReceiveReady=CHS_ReceiveReady do
    begin
      if ChReceiveResult<>res_Ok then WriteLn('Chyba')
      else
      begin
        RecChar:=ChReceiveChar;
        if ChGetBinParam(CH_NumName,cmd_GetBrkChar)<>0 then
          Write('[BrkDet]')
        else
          Write(RecChar);
      end;
    end;
    ...
    { ukončení }
    ChDisconnect;
    if ChResult<>res_Ok then WriteLn('Chyba');
    repeat
    until ChReady=CHS_DisConnect;
    ChClose;
    if ChResult<>res_Ok then WriteLn('Chyba');
    repeat
    until ChReady=CHS_Close;
  end;
  { zrušení instance objektu }
  Dispose(Chn,Done);
```

...  
end.