

# Chn

## JEDNOTKA OBECNÉHO SÉRIOVÉHO KOMUNIKAČNÍHO KANÁLU

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: 22.05.2003

Datum posledního uložení dokumentu: 22.05.2003

(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.Popis konstant a typů	6
5.Objekty	6
5.1. tChn	6
5.1.1. Metody	6
5.1.1.1. Init konstruktor	7
5.1.1.2. ChInitParam konstruktor	7
5.1.1.3. Done destruktork	7
5.1.1.4. ChGetParam funkce	7
5.2. tAddChn	7
5.2.1. Metody	7
5.2.1.1. ChInit funkce	7
6.Příklad	7



---

## 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.

### 1.2. Účel dokumentu

---

Tento dokument slouží jako popis jednotky obecného sériového komunikačního kanálu.

### 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.

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 Chn definuje objekt **tChn**, jehož instance vytváří vyšší vrstvu obecného sériového komunikačního kanálu. Téměř všechny metody tohoto objektu se odkazují na podřízený komunikační objekt, který byl zřetězen v komunikačním kanálu. Pokud instance tohoto podřízeného objektu není vytvořena, vracejí metody objektu tChn chybu. Význam a použití objektu tChn spočívá v tom, že aplikace volí přenosovou cestu složenou z komunikačních objektů nižších popřípadě i vyšších vrstev až za chodu programu, a to parametrem nastavovací metody ChSetParam. Pro toto použití by měla instance komunikačního objektu tChn tvořit nejvyšší vrstvu v komunikačním kanálu a aplikace by se měla odkazovat na tento komunikační objekt, jehož metody zajišťují volání patřičných metod nižší zřetěžené komunikační vrstvy.

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

Protože je objekt **tChn** dědicem rodičovského komunikačního objektu **tChnVirt**, jsou v této příručce popsány jen odlišnosti a speciality pro tento druh sériové komunikace. Ostatní naleznete v příručce popisující objekt **tChnVirt**.

---

### 4. Popis konstant a typů

---

```
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 = 'CHN';
```

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

---

### 5. Objekty

---

---

#### 5.1. tChn

---

##### 5.1.1. Metody

Většina metod je podobná metodám z rodičovského objektu tChnVirt s tím rozdílem, že metody objektu tChn volají metody podřízeného komunikačního objektu přímo přes ukazatel na instanci tohoto komunikačního objektu CH\_Chn. Pokud tento ukazatel není definován (instance podřízeného komunikačního objektu není vytvořena), vyhlásí tento stav jako chybný. Tato vlastnost se týká těchto metod: ChOpen, ChClose, ChConnect, ChDisconnect, ChState, ChReady, ChDestNode, ChSend, ChSendReady, ChSendFlush, ChReceiveBuffer, ChReceiveReady, ChReceive, ChReceiveChar, ChReceiveFlush, ChGetNode a ChTick.

Níže jsou popsány jen ty metody, které se od tohoto popisu odlišují.

### 5.1.1.1. Init konstruktor

```
constructor Init;
```

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

```
inherited Init;  
CH_Type      := cName;  
CH_Name      := CH_Type;  
CH_NumName   := ChNumName(CH_Type);
```

### 5.1.1.2. ChInitParam konstruktor

```
constructor ChInitParam(S : tParamStr);
```

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

### 5.1.1.3. Done destruktork

```
destructor Done;
```

Destruktor **Done** slouží ke zrušení instance komunikačního objektu. Ve svém těle volá pouze zděděný destruktork **Done** z objektu tChnVirt.

### 5.1.1.4. ChGetParam funkce

```
function ChGetParam(S : tParamStr);
```

Metoda **ChGetParam** navrácí nastavené hodnoty parametrů komunikačního objektu.

## 5.2. tAddChn

---

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

## 6. Příklad

---

Není zde uveden žádný konkrétní příklad, protože objekt tChn lze použít jako vyšší (zpravidla tu nejvyšší) vrstvu v komunikačním kanálu s libovolnou podřízenou vrstvou, která je tvořena instancí komunikačního objektu nižší (např. ChnCom) nebo i

vyšší (např. ChnPrt) vrstvy. Jak již bylo řečeno výše, metody objektu tChn zajistí předání řízení podřízené komunikační vrstvě stejně, jako kdyby se aplikace odkazovala na metody komunikačního objektu této vrstvy. To má za výhodu, že při vytváření instance na objekt nejvyšší komunikační vrstvy (a tudíž vlastně i na celý komunikační kanál) se aplikace ještě nemusí rozhodnout, jaký komunikační objekt pro tuto vrstvu zvolí. Jednoduše vytvoří instanci obecného komunikačního objektu tChn a komunikační objekty dalších vrstev zvolí až parametrem nastavovací metody ChSetParam.

Z tohoto popisu plynou tyto změny v aplikačním programu:

1. přilinkovat knihovnu Chn příkazem "uses Chn"
2. vytvořit instanci objektu tChn příkazem `"MyChn := ChnCollection^.ChNewInit (Chn.cName)"`

A o nic dalšího se aplikace starat nemusí (samozřejmě za předpokladu, že je správně volána metoda ChSetParam se správným řetězcem parametrů).