

# uTerm10R

## JEDNOTKA IMPLEMETUJÍCÍ VZDÁLENÝ TERMINÁL TERM10

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.Popis objektu tTermT10R	6
4.1.    Proměnné	6
4.2.    Metody	7
4.2.1.    Init	7
4.2.2.    Tick	7
4.2.3.    SendDataToRemoteTerm	7
4.2.4.    ReceiveDataFromRemoteTerm	7
4.2.5.    FLLight	8
4.2.6.    DispContr	8
4.2.7.    WriteOut	8
4.2.8.    BellOn	8
4.2.9.    BellOff	8
4.2.10.    BeepKeyOn	9
4.2.11.    BeepKeyOff	9
4.2.12.    SendOutData	9
4.2.13.    SendSoundData	9
5.Funkce Cr_TermT10R	9



## 1. O dokumentu

---

### 1.1. Revize dokumentu

---

Verze dokumentu	Verze SW	Autor	Datum vydání	Popis změn
1.00	1.XX	Če		První vydání
1.10	2.XX	Tu	16.05.2003	Úprava dokumentu dle ISO9000

### 1.2. Účel dokumentu

---

Tento dokument slouží jako popis jednotky implementující vzdálený terminál Term10.

### 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, uCharBuf, uDispT10, uDspT10R, uATerm a uTermT10.

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

---

Jednotka **uTerm10R** implementuje funkce terminálu TERM10 ve vzdálené variantě. V této variantě není na systému přítomen fyzický hardware terminálu, ale pouze komunikační kanál. Kódy stisknutých tlačítek jsou přijímány z komunikačního kanálu a grafická data obrazovky nejsou zobrazována, ale vysílána v podobě ESC sekvencí na komunikační kanál.

Zděděné metody zde nejsou popsány, jejich popis je možno najít v dokumentaci k jednotkám **tATerm**, **tTermChr**, **uTermGr** a **uTermT10**.

### 4. Popis objektu tTermT10R

---

Objektový typ **tTermT10R** je potomkem typu **tTermT10**, který implementuje funkce terminálu TERM10 v lokální variantě. Typ **tTermT10R** implementuje funkce terminálu TERM10 ve vzdálené variantě. Na systému není přítomen fyzický hardware terminálu, ale pouze komunikační kanál. Kódy stisknutých tlačítek jsou přijímány z komunikačního kanálu. a grafická data obrazovky nejsou zobrazována, ale vysílána v podobě ESC sekvencí na komunikační kanál.

#### 4.1. Proměnné

---

```
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.

```
SendDataToRemoteTermRunning: Boolean;
```

Proměnná **SendDataToRemoteTermRunning** uchovává příznak běhu metody **SendDataToRemoteTerm**. Používá se interně k ochraně proti vícenásobnému vstupu do metody.

```
StBell: Boolean;
```

Proměnná **StBell** uchovává stav zvukové signalizace terminálu.

```
KeybFlStart: Boolean;
```

Proměnná **KeybFlStart** obsahuje příznak stisku tlačítka "START". Nahrazuje funkci proměnné **FlStart** v objektu **tKeybT10** v případě lokální varianty terminálu.

```
KeybFlStop: Boolean;
```

Proměnná **KeybFlStop** obsahuje příznak stisku tlačítka "STOP". Nahrazuje funkci proměnné **FlStop** v objektu **tKeybT10** v případě lokální varianty terminálu.

```
OutBuff: array [0..31] of char;
```

Proměnná **OutBuff** představuje buffer pro vysílání dat s popisem stavu optovýtupů.

```
SoundBuff: array [0..31] of char;
```

Proměnná **SoundBuff** představuje buffer pro vysílání dat s popisem stavu zvukové signalizace.

## 4.2. Metody

---

### 4.2.1. Init

```
constructor Init(NewDisp:pADisp;NewKeyb:pAKeyb;  
  NAddr:Word;NewChnTerm:pChnVirt;NewChnRecBuf:pointer);
```

Konstrutor **Init** inicializuje objekt terminálu **TERM10** a nastavuje proměnné **KeybFlStart**, **KeybFlStop** a **StBell** na false.

### 4.2.2. Tick

```
procedure Tick;virtual;
```

Metoda **Tick** je periodicky volána z kontextu procesu "Term" a zajišťuje činnost terminálu. Provádí tyto činnosti:

Přijímá znaky z komunikačního kanálu a předává je metodě **ReceiveDataFromRemoteTerm**.

Testuje příznaky stisku tlačítek "START" a "STOP" (**KeybFlStart** a **KeybFlStop**). Bylo-li některé z tlačítek stisknuto a je instalována příslušná uživatelská procedura (**UserStartProc**, **UserStopProc**), volá ji a nuluje příznak stisku příslušného tlačítka. Není-li uživatelská procedura instalována, není příznak nulován.

Odešle stav signalizačních LED voláním **LedSign(0,0)**, odešle stav zvukové signalizace voláním metody **SendSoundData** a je-li povolena obnova stavu optovýtupů (**FlRereshOut**), odešle stav optovýtupů voláním **SendOutData**.

Voláním metody objektu displeje (**Disp^.Tick**) zajišťuje odesílání dat pro zobrazení na displeji.

### 4.2.3. SendDataToRemoteTerm

```
procedure SendDataToRemoteTerm(SrcTrBufPtr:pBuf;  
  SrcTrBufSize:word);virtual;
```

Metoda **SendDataToRemoteTerm** zajišťuje odeslání dat po komunikačním kanálu do podřízené periferie. Data určená k odeslání se nachází v bufferu jehož adresa je předávána v parametru **SrcTrBufPtr**. Parametr **SrcTrBufSize** udává počet dat. Tato metoda je chráněna před vícenásobným vstupem z více procesů. Po dobu, kdy se některý z procesů nachází v chráněné sekci je nastavena proměnná **SendDataToRemoteTermRunning**. Pokud je v této době metoda zavolána z kontextu jiného procesu, vyčká tento proces ve smyčce s opakovaným voláním procedury **Wait** o.s. ReTOS na uvolnění chráněné sekce.

### 4.2.4. ReceiveDataFromRemoteTerm

```
procedure ReceiveDataFromRemoteTerm(RecBufPtr:pBuf;  
  RecBufSize:word);virtual;
```

Metoda **ReceiveDataFromRemoteTerm** provádí zpracování dat přijatých po komunikační lince. Parametrem **RecBufPtr** je předán ukazatel na buffer s přijatými daty a parametrem **RecBufSize** počet dat. Přijatá data představují kódy stisknutých kláves na podřízené periférii. Jsou vkládány voláním metody **Keyb^.InsertKey** do

bufferu klávesnice. Výjimku tvoří znaky **zStart** a **zStop** indikující stisk tlačítka "START" a "STOP". Tyto znaky nejsou do bufferu klávesnice vloženy, ale na základě jejich přijetí jsou nastaveny příznaky **KeybFlStart** a **KeybFlStop**. K obsluze tlačítek "START" a "STOP" se používají metody zděděné z objektu **tTermT10**.

#### 4.2.5. FlLight

```
procedure FlLight(B: Byte);virtual;
```

Metoda **FlLight** je prázdná. V objektu **tTermT10** slouží k nastavení jasu výbojky pro podsvětlení displeje. V této variantě je jas řízen lokálně v podřízené periférii (objekt **tTerm10S**).

#### 4.2.6. DispContr

```
procedure DispContr(B: Byte);virtual;
```

Metoda **DispContr** je prázdná. V objektu **tTermT10** slouží k nastavení kontrastu displeje. V této variantě je kontrast řízen lokálně v podřízené periférii (objekt **tTerm10S**).

#### 4.2.7. WriteOut

```
procedure WriteOut(On,Off:Byte);virtual;
```

Metodou **WriteOut** je možné ovládat optovýstupy. Nastavením jednotlivých bitů parametru **On** je definováno, které z bitů mají být nastaveny a nastavením jednotlivých bitů parametru **Off**, které z bitů mají být nulovány. Metoda pouze modifikuje uchovaný stav optovýstupů v proměnné **StOut**. Stav optovýstupů je pak možné odeslat v podobě ESC sekvence po komunikačním kanálu do podřízené periférie voláním metody **SendOutData**. Pokud je povolena obnova stavu optovýstupů (je nastaven zděděný příznak **FlRefreshOut**), je metoda **SendOutData** volána metodou **Tick**. Příznak **FlRefreshOut** lze ovládat zděděnými metodami **EnableRefreshOut** a **DisableRefreshOut**.

#### 4.2.8. BellOn

```
procedure BellOn;virtual;
```

Metodou **BellOn** je možné zapnout zvukovou signalizaci. Metoda pouze nastaví uchovávaný stav zvukové signalizace v proměnné **StBell**. Stav zvukové signalizace je pak možné odeslat v podobě ESC sekvence po komunikačním kanálu do podřízené periférie voláním metody **SendSoundData**. Tato metoda je volána metodou **Tick**.

#### 4.2.9. BellOff

```
procedure BellOff;virtual;
```

Metodou **BellOff** je možné vypnout zvukovou signalizaci. Metoda pouze nastaví uchovávaný stav zvukové signalizace v proměnné **StBell**. Stav zvukové signalizace je pak možné odeslat v podobě ESC sekvence po komunikačním kanálu



do podřízené periférie voláním metody **SendSoundData**. Tato metoda je volána metodou **Tick**.

#### 4.2.10. BeepKeyOn

```
procedure BeepKeyOn;virtual;
```

Metoda **BeepKeyOn** je prázdná. V objektu **tTermT10** slouží k zapnutí zvukové indikace stisku tlačítek klávesnice. V této variantě je zvuková indikace stisku tlačítek ovládána lokálně v podřízené periférii (objekt **tTerm10S**).

#### 4.2.11. BeepKeyOff

```
procedure BeepKeyOff;virtual;
```

Metoda **BeepKeyOff** je prázdná. V objektu **tTermT10** slouží k vypnutí zvukové indikace stisku tlačítek klávesnice. V této variantě je zvuková indikace stisku tlačítek ovládána lokálně v podřízené periférii (objekt **tTerm10S**).

#### 4.2.12. SendOutData

```
procedure SendOutData;virtual;
```

Metoda **SendOutData** sestaví v bufferu **OutBuff** na základě uchovávaného stavu optovýstupů ESC sekvenci a prostřednictvím metody **SendDataToRemoteTerm** ji odvysílá do podřízené periférie.

Syntaxe ESC sekvence:

<ESC> ' <On> <Off> <ESC> "

<On> a <Off> jsou znakové reprezentace čísel z intervalu 0 až 255. <On> definuje nastavením jednotlivých bitů, které výstupy mají být nastaveny a <Off> definuje nastavením jednotlivých bitů, které výstupy mají být nulovány.

Příklad vysílané ESC sekvence: <ESC> ' 13 242 <ESC> "

#### 4.2.13. SendSoundData

```
procedure SendSoundData;virtual;
```

Metoda **SendSoundData** sestaví v bufferu **SoundBuff** na základě uchovávaného stavu zvukové signalizace ESC sekvenci a prostřednictvím metody **SendDataToRemoteTerm** ji odvysílá do podřízené periférie.

Syntaxe ESC sekvence:

<ESC> ' ON|OFF <Off> <ESC> "

ON zapíná zvukovou signalizaci a OFF ji vypíná.

### 5. Funkce Cr\_TermT10R

---

```
function Cr_TermT10R(Colls,Rows:integer;Adr:Word;  
ChnTerm:pChnVirt;RecBuf:pointer):PTermT10R;
```

Funkce slouží **Cr\_TermT10R** vytvoří instance objektů terminálu klávesnice a displeje pro vzdálenou variantu terminálu TERM10. Jako parametry jsou předávány

rozměry znakového rastru displeje (**Colls** - sloupce, **Rows** - řádky), ukazatel na instanci komunikačního kanálu (**ChnTerm**) a ukazatel na buffer pro data přijatá komunikačním kanálem. Parametr **Adr** je předáván do vytvořených objektů jako adresa terminálu v IO prostoru. Funkce vrací ukazatel na vytvořenou instanci objektu terminálu.