

uPCT01

SIMULÁTOR TERMINÁLU TERM01 NA POČÍTAČI PC

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: 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 konstant a typů	6
5.Globální proměnné jednotky	6
6.Objektový typ tPCTermT01	6
6.1. Proměnné objektu	7
6.2. Init	7
6.3. SetTerm	7
6.4. GetTerm	8
6.5. RunTerm	8
6.6. StopTerm	8
6.7. ReceiveKeyb	8
6.8. SendKeyb	8
6.9. ReceiveDisp	8
6.10. SendDisp	8
6.11. KbdTerm	8
6.12. DspTerm	9
6.13. WaitTerm	9
7.6.Příklad použití	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 simulátor terminálu Term01 na počítači PC.

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 a uATerm.

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

Terminál TERM01 je malý průmyslový obecně použitelný terminál s membránovou klávesnicí a alfanumerickým podsvíceným LCD displejem se čtyřmi řádky o šestnácti znacích připojitelný k řídicímu systému pomocí sériové komunikační linky RS-232 nebo RS-485.

Pro účely ladění je vhodné terminál simulovat na počítači PC. Simulátor tvoří samostatný proces v operačním systému ReTOS a může tedy být součástí laděného programu pro řídicí systém. V tomto případě lze ke komunikaci se simulátorem využít komunikační kanál **ChnMBox**, který komunikuje přes schránky v o.s. ReTOS a vyhnout se tak potřebě komunikačního kabelu i sériových portů. Simulátor zobrazuje do okénka na monitoru PC a jako klávesnice využívá klávesnici PC.

Tato jednotka implementuje objektový typ simulátoru TERM01 na počítači PC.

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.

```
Const
  cPcTermName   = 'PcTerm';   { Jméno procesu }
  cPcTermStk    = 2000;       { Velikost zásobníku procesu }
  cPcTermSPrio  = 100;       { Statická priorita procesu }
  cPcTermDPrio  = 254;       { Dynamická priorita procesu }
```

Tyto konstanty obsahují implicitní hodnoty parametrů procesu simulátoru TERM01.

```
type
  tAAtr2 = array[1..2] of Byte;
  tAAtr5 = array[1..5] of Byte;
```

Typy **tAAtr2** a **tAAtr5** jsou typy pro atributy simulátoru.

5. Globální proměnné jednotky

```
var
  PcTeMailBox : MailBox;
  TePcMailBox : MailBox;
```

Proměnné **PcTeMailBox** a **TePcMailBox** mohou být využity pro komunikaci kanálem **ChnMBox**.

6. Objektový typ tPCTermT01

```
type
  pPcTermT01 = ^tPcTermT01;
  tPcTermT01 = object(tObject);
```

Objektový typ **tPCTermT01** implementuje simulátor terminálu TERM01. Je dědicem kořenu objektové hierarchie **tObject**.

6.1. Proměnné objektu

```
PcTermName : IdentType;
PcTermStk  : Integer;
PcTermSPrio : Integer;
PcTermDPrio : Integer;
```

Tyto proměnné obsahují parametry procesu simulátoru terminálu.

```
OrTerRect : tRect;
```

Proměnná **OrTerRect** obsahuje znaky orámování okna na monitoru PC.

```
OrTerAtr : tAAtr2;
```

Proměnná **OrTerAtr** obsahuje atributy orámování okna na monitoru PC.

```
TerRect : tRect;
```

Proměnná **TerRect** obsahuje znaky orámování okénka simulátoru.

```
TerAtr : tAAtr5;
```

Proměnná **OrTerAtr** obsahuje atributy orámování okénka simulátoru.

```
TerWin : PCrtWin;
```

Proměnná **TerWin** obsahuje ukazatel na okénko simulátoru.

```
pFlEnd : ^Boolean;
```

Proměnná **pFlEnd** obsahuje ukazatel na proměnnou, která informuje o ukončení programu.

```
Char : Char;
TxStr : String;
ParamF : text;
TickDivider : Integer;
flNul : Boolean;
DspCtrl : Byte;
CursX : Byte;
CursY : Byte;
```

Toto jsou pomocné proměnné.

```
Chn : pChnVirt;
```

Proměnná **Chn** obsahuje ukazatel na objekt komunikačního kanálu.

6.2. Init

```
constructor Init(var FlEnd: Boolean; Channel: pChnVirt);
```

Konstruktor **Init** inicializuje proměnné objektu na implicitní hodnoty. Parametr **FlEnd** obsahuje proměnnou, jejíž nastavení informuje o ukončení aplikace a **Channel** ukazatel na objekt komunikačního kanálu.

6.3. SetTerm

```
procedure SetTerm(S: tParamStr);virtual;
```

Metoda **SetTerm** nastavuje simulátor. Parametr **S** obsahuje příkazovou řádku s hodnotami nastavení. Nastavuje se:

NAM=<string>	Jméno procesu.
STK=<zásobník>	Velikost zásobníku procesu.
SPR=<stat.priorita>	Statická priorita procesu.
DPR=<dyn.priorita>	Dynamická priorita procesu.

XY=<X> <Y>	Souřadnice levého horního rohu.
ATO=<at1> <at2>	Atributy okna na monitoru PC.
ATT=<at1> <at2> <at3> <at4> <at5>	Atributy okénka simulátoru.

6.4. GetTerm

```
function GetTerm: tParamStr;virtual;
```

Metoda **GetTerm** vrací nastavené parametry simulátoru.

6.5. RunTerm

```
procedure RunTerm;virtual;
```

Metoda **RunTerm** spouští proces simulátoru.

6.6. StopTerm

```
procedure StopTerm;virtual;
```

Metoda **StopTerm** ruší proces simulátoru.

6.7. ReceiveKeyb

```
procedure ReceiveKeyb(var S: String); virtual;
```

Metoda **ReceiveKeyb** přijímá znaky z klávesnice PC do proměnné **S**.

6.8. SendKeyb

```
procedure SendKeyb(S: String);virtual;
```

Metoda **SendKeyb** vyšle znaky z parametru **S** na komunikační kanál.

6.9. ReceiveDisp

```
procedure ReceiveDisp(var S: String); virtual;
```

Metoda **ReceiveDisp** přijímá z komunikačního kanálu znaky určené k zobrazení do proměnné **S**.

6.10. SendDisp

```
procedure SendDisp(S: String); virtual;
```

Metoda **SendDisp** zobrazí přijaté znaky z parametru **S** na monitor PC.

6.11. KbdTerm

```
procedure KbdTerm;virtual;
```

Metoda **KbdTerm** obsluhuje klávesnici simulátoru.

6.12. DspTerm

```
procedure DspTerm;virtual;
```

Metoda **DspTerm** obsluhuje displej simulátoru.

6.13. WaitTerm

```
procedure WaitTerm;virtual;
```

Metoda **WaitTerm** je volána na konci smyčky obsluhy terminálu.

7. 6.Příklad použití

Při použití této jednotky uživatel vytvoří a inicializuje instanci objektu simulátoru a komunikačního kanálu, nastaví parametry simulátoru a zavolá metodu **RunTerm**, která spustí proces simulátoru.

```
PcTerminal:=New(pPcTermT01, Init (FlEnd,  
                                New(pChnCom, ChInitParam (ParamStrTEs) ));  
with PcTerminal^ do  
begin  
  SetTerm('SPR=80 DPR=254 STK=6000');  
  RunTerm;  
end;
```