

# uSimT10

## JEDNOTKA IMPLEMENTUJÍCÍ OBJEKTY PRO SIMULACI TERMINÁLU TERM10 NA PC

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 konstant a typů	6
5.Popis objektu tSimDispT10	6
5.1. Proměnné	6
5.2. Metody	7
5.2.1. Init	7
5.2.2. InitHwProc	7
5.2.3. DoneHwProc	8
5.2.4. MoveVideoRWM_ToHw	8
5.2.5. mFILight	8
5.2.6. mLedSign	8
5.2.7. mDispContr	8
6.Popis objektu tSimKeybT10	8
6.1. Proměnné	9
6.2. Metody	9
6.2.1. Init	9
6.2.2. KTick	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 objekty pro simulaci terminálu Term10 na 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 uATerm, uKeybT10, uDispT10 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 implementuje objekty pro simulaci klávesnice a displeje terminálu TERM10 na PC **tSimKeybT10** a **tSimDispT10**. Objekty jsou potomky objektových typů **tKeybT10** a **tADispT10**. Pro simulaci terminálu na PC stačí vytvořit jejich instance a předat je objektu terminálu **tTermT10** místo instancí objektů **tKeybT10** a **tDispT10**.

Klávesnice terminálu je nahrazena klávesnicí počítače PC. Jednotlivá tlačítka terminálu jsou nahrazena příslušnými tlačítky klávesnice PC až na tyto výjimky:

terminál	klávesnice PC
START	Alt - A
STOP	Alt - S
Shift – Enter	Ctrl - Enter

Dále je zavedena kombinace tlačítek Alt - X, která je určena k ukončení programu.

Displej terminálu je nahrazen okénkem na monitoru PC. V okénku jsou pod okénkem vlastního displeje simulovány signalizační LED a nad displejem je zobrazen nastavený jas a kontrast displeje.

Zděděné metody jsou popsány v dokumentacích k jednotkám **uATerm**, **uDispT10** a **uKeybT10**.

### 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
  zAltXPc      =#$2d;
  zAltAPc      =#$1E;
  zAltSPc      =#$1F;
  zCtrlEnterPc =#$0A;
  zDelPc       =#$53;
```

Tyto konstanty definují kódy některých kláves na klávesnici PC. Jsou používány interně.

### 5. Popis objektu tSimDispT10

```
type
  pSimDispT10 = ^tSimDispT10;
  tSimDispT10 = object(tADispT10);
```

Objektový typ **tSimDispT10** je potomkem objektového typu abstraktního displeje terminálu TERM10 **tADispT10**. Slouží k simulaci displeje na monitoru PC.

#### 5.1. Proměnné

```
vSimT10GraphRect: tGraphRect;
```

Proměnná **vSimT10GraphRect** obsahuje rámeček okénka displeje na monitoru PC.

```
vOfsX:Word;  
vOfsY:Word;
```

Proměnné **vOfsX** a **vOfsY** obsahují posunutí zobrazení displeje od počátku souřadného systému obrazovky v pixelech.

```
vDoubleSize:Boolean;
```

Proměnná **vDoubleSize** obsahuje příznak zobrazení displeje v dvojnásobné velikosti.

```
SimTERM10Win:pGraphWin;
```

Proměnná **SimTerm10Win** obsahuje odkaz na okénko, ve kterém je displej zobrazen.

```
FlReqShowLed:Boolean;
```

Proměnná **FlReqShowLed** obsahuje příznak požadavku pro vykreslení LED na monitor.

```
VisibleLight:Byte;  
VisibleContr:Byte;
```

Proměnné **VisibleLight** a **VisibleContr** obsahují zobrazený jas a kontrast displeje.

## 5.2. Metody

---

### 5.2.1. Init

```
constructor Init(TermOwner:pATerm;  
                 CharColls,CharRows:Byte;  
                 Adr:Word;EnIniHwProc:Boolean;  
                 OfsX,OfsY:Word;  
                 OrgX,OrgY,OrgW,OrgH:Integer;  
                 QDoubleSize:Boolean);
```

Konstruktor **Init** inicializuje objekt, nastavuje jeho proměnné. Parametr **TermOwner** obsahuje odkaz na vlastníka objektu, objekt terminálu, parametry **CharColls** a **CharRows** rozměry znakového rastru displeje, parametr **Adr** obsahuje adresu displeje v I/O prostoru a je zaveden pro zpětnou kompatibilitu, parametr **EnIniHwProc** definuje, zda se má volat metoda **InitHwProc**. Jeho nastavení na false umožňuje toto volání potlačit a metodu **InitHwProc** volat dodatečně např. po inicializaci dalších proměnných v konstruktoru potomka tohoto objektového typu. Parametry **OfsX** a **OfsY** udávají posunutí zobrazení displeje vůči počátku souřadného systému monitoru (levý horní roh) v pixelech. Parametry **OrgX**, **OrgY**, **OrgW**, **OrgH** definují umístění a rozměry okénka pro zobrazení displeje v pixelech a parametr **QDoubleSize** definuje požadavek na zobrazení displeje v dvojnásobné velikosti.

### 5.2.2. InitHwProc

```
procedure InitHwProc;virtual;
```

Metoda **InitHwProc** slouží k inicializaci hardware displeje. V tomto objektu vytvoří a vykreslí grafické okénko pro zobrazení displeje na monitoru PC.

### 5.2.3. DoneHwProc

```
procedure DoneHwProc;virtual;
```

Metoda **DoneHwProc** slouží k ukončení činnosti hardware displeje. V tomto objektu zruší grafické okénko pro zobrazení displeje na monitoru PC.

### 5.2.4. MoveVideoRWM\_ToHw

```
procedure MoveVideoRWM_ToHw;virtual;
```

Metoda **MoveVideoRWM\_ToHw** slouží k přesunu dat pro zobrazení na hardware displeje. V tomto objektu vykreslí data na monitor PC.

### 5.2.5. mFlLight

```
procedure mFlLight(b:byte);virtual;
```

Metoda **mFlLight** nastavuje jas displeje. Nastavený jas je vypsán nad zobrazením displeje na monitoru. Hodnota zobrazeného jasu je omezena na rozsah 0 až 99.

### 5.2.6. mLedSign

```
procedure mLedSign(On,Off:byte); virtual;
```

Metoda **mLedSign** ovládá signalizační LED terminálu. Parametr **On** definuje, které LED se mají rozsvítit a parametr **Off**, které se mají zhasnout. Jednotlivé bity v parametrech **On** a **Off** přísluší jednotlivým LED a to tak, že levé LED přísluší bit s nejnižší vahou (LSB) pravé LED (v tlačítku Start) přísluší bit s nejvyšší vahou (MSB).

V tomto objektu metoda pouze nastaví proměnnou uchovávající stav LED a požadavek na vykreslení. Vlastní vykreslení na monitor PC je prováděno v metodě **MoveVideoRWM\_ToHw**. Vzhledem k tomu, že je tímto způsobem z metody vypuštěno zdlouhavé vykreslení na monitor, je možné metodu volat z přerušení stejně jako při použití skutečného displeje TERM10. Je však nutné si uvědomit, že je vykreslení na monitor zpožděno a proto nebudou rychlé změny svitu LED na monitoru PC patrné.

### 5.2.7. mDispContr

```
procedure mDispContr(B:byte); virtual;
```

Metoda **mDispContr** nastavuje kontrast displeje. Nastavený kontrast je vypsán nad zobrazením displeje na monitoru. Hodnota zobrazeného kontrastu je omezena na rozsah 0 až 99.

## 6. Popis objektu tSimKeybT10

---

```
type
```

```
  pSimKeybT10=^tSimKeybT10;  
  tSimKeybT10=object(tKeybT10);
```

Objektový typ **tSimKeybT10** je potomkem objektového typu klávesnice TERM10 **tKeybT10**. Slouží k simulaci klávesnice terminálu TERM10 klávesnicí PC.



## 6.1. Proměnné

---

`FlNull:Boolean;`

Proměnná **FlNull** je využívána interně pro dekodování kláves stisknutých na klávesnici PC.

`FlgEndPtr:^Boolean;`

Proměnná **FlgEndPtr** je ukazatel na proměnnou pro ukončení programu.

## 6.2. Metody

---

### 6.2.1. Init

```
constructor Init(TermOwner:pATerm;Len:Word;  
                ADr:Word;var AFlgEnd:Boolean);
```

Konstruktor **Init** inicializuje objekt a nastavuje jeho proměnné. Parametr **TermOwner** obsahuje odkaz na vlastníka objektu, objekt terminálu, parametr **Len** délku alokovaného vyrovnávacího bufferu klávesnice a **Adr** bázovou adresu hardware klávesnice v I/O prostoru procesoru a **AFlgEnd** je proměnná pro ukončení programu. Tato proměnná bude nastavena na true po stisku kombinace tlačítek Alt - X.

### 6.2.2. KTick

```
procedure KTick;virtual;
```

Metoda **KTick** zabezpečuje cyklické čtení stisknutých kláves z hardware klávesnice. V tomto objektu čte kódy stisknutých kláves na klávesnici PC.