uDispT10

JEDNOTKA IMPLEMENTUJÍCÍ DISPLEJE TERMINÁLU TERM10

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

Verze dokumentu 1.10

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 popsané. 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)

Obsah :

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. Doplněná metoda tADispT10.DispClrScr.

1.2. Účel dokumentu

Tento dokument slouží jako popis jednotky implementující displej terminálu 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 uATerm, uTermGr, uBitmap a G240x128.

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 tři objekty pro obsluhu grafických displejů terminálu TERM10 a několik dalších pomocných objektů používaných objekty displeje. Objekt **tADispT10** je obecný displej terminálu TERM10. Objekty **tDispT10** a **tDispT10A** implementují funkce pro konkrétní grafické displeje používané v terminálu TERM10 v návaznosti na jejich hardware.

Veškeré funkce displeje jsou uživateli dostupné prostřednictvím objektu terminálu. Uživatele tak bude nejvíce zajímat pouze vytvoření instance objektu displeje, jejíž odkaz se předává objektu terminálu.

Zděděné metody jsou popsány v dokumentaci k jednotce **uATerm**.

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.

```
type
  tVideoImageManState=(vims_ClearImage,
                       vims_PaintedImage,
                       vims_RePaintImageRq);
```

Typ **tVideoImageManState** je používán pro definici stavu objektu **tVideoImageManager** a jeho potomků **tVideoImageManText** a **tVideoImageManGraphic**.

5. Popis objektu tVideoImageManager

```
type
  pVideoImageManager=^tVideoImageManager;
  tVideoImageManager=object(tObject);
```

Objektový typ **tVideoImageManager** je abstraktní typ správce videopaměti. Od tohoto typu jsou odvozeny další dva objektové typy pro správu obrazové roviny textu a pro správu obrazové roviny vektorové grafiky. Tyty objekty jsou používány interně objektem abstraktního displeje TERM10 **tADispT10** pro vykreslení jednotlivých obrazových rovin a jejich skládání.

5.1. Proměnné

```
pImageGrDataMan:pGraphicDataMan;
```

Proměnná **pImageGrDataMan** uchovává odkaz na objekt spravující textové popisy obsahu obrazovky.

```
vImage:tVidoRWM;
```

Proměnná **vImage** představuje videopaměť obrazové roviny, kterou objekt spravuje.

```
vImageState:tVideoImageManState;
```

Proměnná **vImageState** uchovává stav objektu.

5.2. Metody

5.2.1. Init

```
constructor Init(ImgGrDataMan:pGraphicDataMan);
```

Konstruktor **Init** inicializuje objekt a smaže videopaměť, kterou spravuje. Parametrem **ImgGrDataMan** je mu předáván odkaz na objekt spravující textové popisy obsahu obrazovky.

5.2.2. ClearImage

```
procedure ClearImge(Fl: Boolean);
```

Metoda **ClearImage** je statická metoda, která smaže videopaměť, kterou objekt spravuje. Pokud je nastaven příznak Fl, bude provedeno fyzické nulování (naplnění paměti hodnotou $FF).

5.2.3. Clear

```
procedure Clear(Fl: Boolean);
```

Metoda **Clear** volá statickou metodu **ClearImage** pro smazání videopaměti. Pokud je nastaven příznak Fl, bude provedeno fyzické nulování (naplnění paměti hodnotou $FF).

5.2.4. PaintImage

```
procedure PaintImage;virtual;
```

Metoda **PaitImage** je abstraktní metoda pro vykreslení konkrétního obsahu videopaměti. Vlastní vykreslení je implementováno v potomcích objektu.

5.2.5. GetVideoImageManState

```
function GetVideoImageManState:tVideoImageManState;
```

Metoda **GetVideoImageManState** vrací stav objektu.

5.2.6. VideoImageROP

```
procedure VideoImageROP(var DestVidoRWM:tVideoRWM;
                        Rop:tVideoROP);
```

Metoda **VideoImageROP** zjistí stav videopaměti a je-li nastavena žádost o překreslení (stav vims_RePaintImageRq) voláním **PaitImage** vykreslí obrazovou rovinu do videopaměti. Tu pak složí s videopamětí zadanou parametrem **DestVideoRWM** operací **Rop** a výsledek vloží do videopaměti **DestVideoRWM**.

6. Popis objektu tVideoImageManText

```
type
  pVideoImageManText=^tVideoImageManText;
  tVideoImageMantext=object(tVideoImageManager);
```

Objektový typ **tVideoImageManText** je potomkem abstraktního objektového typu **tVideoImageManager**. Implementuje metody pro práci s obrazovou rovinou textu.

6.1. Proměnné

```
aDisplayOwner:pADisp;
```

Proměnná **aDisplayOwner** uchovává odkaz na vlastníka objektu, objekt displeje.

6.2. Metody

6.2.1. Init

```
constructor Init(Owner:pADisp;ImgGrDataMan:pGraphicDataMan);
```

Konstuktor **Init** inicializuje objekt a smaže videopaměť, kterou spravuje. Parametrem **ImgGrDataMan** je mu předáván odkaz na objekt spravující textové popisy obsahu obrazovky a parametrem **Owner** odkaz na vlastníka objektu, objekt displeje.

6.2.2. PaintImage

```
procedure PaintImage;virtual;
```

Metoda **PaitImage** vykreslí obsah obrazové roviny textu.

6.2.3. AcceptNewImageText

```
procedure AcceptNewImageText;
```

Metoda **AcceptNewImageText** zjistí zda došlo ke změně v obrazových datech popisujících textovou rovinu obrazovky a případně nastaví stav objektu na žádost o překreslení.

7. Popis objektu tVideoImageManGraphic

```
type
  pVideoImageManGraphic=^tVideoImageManGraphic;
  tVideoImageManGraphic=object(tVideoImageManager);
```

Objektový typ **tVideoImageManGraphic** je potomkem abstraktního objektového typu **tVideoImageManager**. Implementuje metody pro práci s obrazovou rovinou vektorové grafiky.

7.1. Metody

7.1.1. PaintImage

```
procedure PaintImage;virtual;
```

Metoda **PaitImage** vykreslí obsah obrazové roviny vektorové grafiky.

7.1.2. AcceptNewGraphicDscr

```
procedure AcceptNewGraphicDscr;
```

Metoda **AcceptNewGraphicDscr** zjistí zda došlo ke změně v obrazových datech popisujících obrazovou rovinu vektorové grafiky a případně nastaví stav objektu na žádost o překreslení.

8. Popis objektu tBkGroundManager

```
type
  pBkGroundManager=^tBkGroundManager;
  tBkGroundManager=object(tObject);
```

Objektový typ **tBkGroundManager** slouží ke správě obrazové roviny pozadí obrazovky.

8.1. Proměnné

```
OwnerGrDataMan:pGraphicDataMan;
```

Proměnná **OwnerGrDataMan** obsahuje odkaz na objekt spravující grafická data obrazovky.

8.2. Metody

8.2.1. Init

```
constructor Init(Owner:pGraphicDataMan);
```

Konstruktor **Init** inicializuje objekt. Parametrem **Owner** je mu předáván odkaz na objekt pro správu textových popisů grafického obsahu obrazovky.

8.2.2. AcceptBkGround

```
procedure AcceptBkGround(var BkImage:tVideoRWM);
```

Metoda **AcceptBkGround** vykreslí do videopaměti zadané parametrem BkImage obsah grafické roviny pozadí obrazovky.

9. Popis objektu tADispT10

```
type
  pADispT10=^tADispT10;
```

```
tADispT10=object(tADisp);
```

Objektový typ **tADispT10** je potomek objektu abstraktního displeje **tADisp** a představuje abstraktní displej terminálu TERM10. Typ **tADispT10** implementuje metody pro obecný grafický displej s rozlišením 240 × 128 pixelů a definuje nové abstraktní metody pro funkce displeje terminálu TERM10.

9.1. Proměnné

```
vVideoRWM:tVideoRWM;
```

Proměnná **vVideoRWM** představuje videopaměť, ve které je skládán výsledný obraz předávaný hardware displeje.

```
DispGrDataMan:pGraphicDataMan;
```

Proměnná **DispGrDataMan** obsahuje odkaz na instanci objektu pro správu textových popisů grafické obrazovky displeje.

```
VideoImageManText:pVideoImageManText;
```

Proměnná **VideoImageManText** obsahuje odkaz na instanci objektu pro správu obrazové roviny textu.

```
VideoImageManGraphic:pVideoImageManGraphic;
```

Proměnná **VideoImageManGraphic** obsahuje odkaz na instanci objektu pro správu obrazové roviny vektorové grafiky.

```
BkGroundManager:pBkGroundManager;
```

Proměnná **BkGroundManager** obsahuje odkaz na instanci objektu pro správu obrazové roviny pozadí obrazovky.

```
StLedSign:Byte;
```

Proměnná **StLedSign** uchovává stav signalizačních LED na terminálu.

9.2. Metody

9.2.1. Init

```
constructor Init(TermOwner:pATerm; CharColls,CharRows:Byte);
```

Konstruktor **Init** slouží k inicializaci objektu. Vytvoří instanci objektu pro správu textových popisů grafické obrazovky a instance objektů pro správu jednotlivých obrazových rovin. Parametr **TermOwner** obsahuje odkaz na vlastníka objektu, objekt terminálu. Parametry **CharCols** a **CharRows** obsahují rozměry znakového rastru displeje.

9.2.2. Done

```
destructor Done;virtual;
```

Destruktor **Done** zruší instanci objektu pro správu textových popisů grafické obrazovky a instance objektů pro správu jednotlivých obrazových rovin.

9.2.3. mFlLight

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

Metoda **mFlLight** je abstraktní metoda pro nastavení jasu displeje.

9.2.4. mLedSign

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

Metoda **mLedSign** je abstraktní metoda pro ovládání signalizačních LED na terminálu.

9.2.5. mDispContr

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

Metoda **mDispContr** je abstraktní metoda pro nastavení kontrastu displeje.

9.2.6. mDispClrScr

```
procedure DispClrScr;
```

Metoda **DispClrScr** smaže displej.

9.2.7. ClearVideoRWM

```
procedure ClearVideoRWM;
```

Metoda **ClearVideoRWM** smaže videopaměť pro skládání výsledného obrazu displeje.

9.2.8. DTickRefreshScreen

```
procedure DTickRefreshScreen;virtual;
```

Metoda **DTickRefreshScreen** zabezpečuje periodickou obnovu obrazu na displeji. Doplňuje metodu **tADisp.DTickRefreshScreen** o vykreslení jednotlivých obrazových rovin a jejich složení.

9.2.9. MoveVideoRWM_ToHw

```
procedure MoveVideoRWM_ToHw;virtual;
```

Metoda **MoveVideoRWM_ToHw** je abstraktní metoda, která předává výsledný obraz hardware displeje.

10. Popis objektu tDispT10

```
type
  pDispT10=^tDispT10;
  tDispT10=object(tADispT10);
```

Objektový typ **tDispT10** je potomek objektu abstraktního displeje terminálu TERM10 **tADispT10.** Implementuje konkrétní podobu abstraktních metod svého předka v návaznosti na hardware displeje. Je určen pro displej Seiko.

10.1. Proměnné

```
vAddr:Word;
```

Proměnná **vAddr** uchovává bázovou adresu displeje v I/O prostoru procesoru.

```
vAdrData:Word;
```

Proměnná **vAdrData** uchovává adresu datového registru displeje v I/O prostoru procesoru.

```
vAdrCmd:Word;
```

Proměnná **vAdrCmd** uchovává adresu příkazového a stavového registru displeje v I/O prostoru procesoru.

```
StDispContr:Byte;
```

Proměnná **StDispContr** uchovává nastavený kontrast displeje.

10.2. Metody

10.2.1. Init

```
constructor Init(TermOwner:pATerm; CharColls,CharRows:Byte;
                 Adresa:Word;EnIniHwProc:Boolean);
```

Konstruktor **Init** slouží k inicializaci objektu. Inicializuje předka objektu a nastavuje proměnné objektu. Parametr **TermOwner** obsahuje odkaz na vlastníka objektu, objekt terminálu. Parametry **CharCols** a **CharRows** obsahují rozměry znakového rastru displeje a parametr **Adresa** bázovou adresu displeje v I/O prostoru procesoru. Parametr **EnIniHwProc** definuje, zda se bude volat metoda **InitHwProc**. Jeho nastavením na false lze volání **InitHwProc** potlačit a tuto metodu volat explicitně později např. po inicializaci dalších proměnných v konstruktoru dědice tohoto objektu.

10.2.2. InitHwProc

```
procedure InitHwProc;virtual;
```

Metoda **InitHwProc** inicializuje hardware displeje.

10.2.3. InitDisplay

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

Metoda **InitDisplay** provádí některá inicializační nastavení hardware displeje. Je určena pro vnitřní potřebu objektu.

10.2.4. Done

```
destructor Done; virtual;
```

Destruktor **Done** volá metodu **DoneHwProc** pro ukončení práce hardware displeje a zruší objekt.

10.2.5. DoneHwProc

```
procedure DoneHwProc;
```

Metoda **DoneHwProc** ukončí práci hardware displeje.

10.2.6. mFlLight

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

Metoda **mFlLight** nastavuje jas displeje. Parametr **B** udává požadovaný jas v rozsahu 0 (tma) až 4 (nejvyšší jas).

10.2.7. 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).

10.2.8. mDispContr

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

Metoda **mDispContr** nastavuje kontrast displeje. Parametr B udává nastavovaný kontrast v rozsahu od 0 (nejnižší kontrast) do 6 (nejvyšší kontrast).

10.2.9. MoveVideoRWM_ToHw

```
procedure MoveVideoRWM_ToHw;virtual;
```

Metoda **MoveVideoRWM_ToHw** je implementuje přesun obrazových dat do hardware displeje.

11. Popis objektu tDispT10A

```
type
  pDispT10A=^tDispT10A;
  tDispT10A=object(tADispT10);
```

Objektový typ **tDispT10A** je potomek objektu abstraktního displeje terminálu TERM10 **tADispT10.** Implementuje konkrétní podobu abstraktních metod svého předka v návaznosti na hardware displeje. Je určen pro displej Toshiba.

11.1. Proměnné

```
vAddr:Word;
```

Proměnná **vAddr** uchovává bázovou adresu displeje v I/O prostoru procesoru.

```
StLightContr:Byte;
```

Proměnná **StLightContr** uchovává nastavený jas a kontrast displeje.

11.2. Metody

11.2.1. Init

```
constructor Init(TermOwner:pATerm;
                 CharColls,CharRows:Byte;
                 Adresa:Word;EnIniHwProc:Boolean);
```

Konstruktor **Init** slouží k inicializaci objektu. Inicializuje předka objektu a nastavuje proměnné objektu. Parametr **TermOwner** obsahuje odkaz na vlastníka objektu, objekt terminálu. Parametry **CharCols** a **CharRows** obsahují rozměry znakového rastru displeje a parametr **Adresa** bázovou adresu displeje v I/O prostoru procesoru. Parametr **EnIniHwProc** definuje, zda se bude volat metoda **InitHwProc**. Jeho nastavením na false lze volání **InitHwProc** potlačit a tuto metodu volat expilicitně později např. po inializaci dalších proměnných v konstruktoru dědice tohoto objektu.

11.2.2. InitHwProc

```
procedure InitHwProc;virtual;
```

Metoda **InitHwProc** inicializuje hardware displeje.

11.2.3. Done

```
destructor Done; virtual;
```

Destruktor **Done** volá metodu **DoneHwProc** pro ukončení práce hardware displeje a zruší objekt.

11.2.4. DoneHwProc

```
procedure DoneHwProc;
```

Metoda **DoneHwProc** ukončí práci hardware displeje.

11.2.5. mFlLight

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

Metoda **mFlLight** nastavuje jas displeje. Parametr **B** udává požadovaný jas v rozsahu 0 (tma) až 4 (nejvyšší jas).

11.2.6. 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).

11.2.7. mDispContr

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

Metoda **mDispContr** nastavuje kontrast displeje. Parametr B udává nastavovaný kontrast v rozsahu od 0 (nejnižší kontrast) do 15 (nejvyšší kontrast).

11.2.8. MoveVideoRWM_ToHw

```
procedure MoveVideoRWM_ToHw;virtual;
```

Metoda **MoveVideoRWM_ToHw** je implementuje přesun obrazových dat do hardware displeje.