uDispT03

JEDNOTKA IMPLEMENTUJÍCÍ DISPLEJ TERMINÁLU TERM03

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 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 tVideoImageManager 6
5.1. Proměnné 7
5.2. Metody 7
5.2.1. Init 7
5.2.2. ClearImage 7
5.2.3. Clear 7
5.2.4. PaintImage 7
5.2.5. GetVideoImageManState 7
5.2.6. VideoImageROP 7
6.Popis objektu tVideoImageManText 8
6.1. Proměnné 8
6.2. Metody 8
6.2.1. Init 8
6.2.2. PaintImage 8
6.2.3. AcceptNewImageText 8
7.Popis objektu tVideoImageManGraphic 8
7.1. Metody 9
7.1.1. PaintImage 9
7.1.2. AcceptNewGraphicDscr 9
8.Popis objektu tBkGroundManager 9
8.1. Proměnné 9
8.2. Metody 9
8.2.1. Init 9
8.2.2. AssignBkGround 9
9.Popis objektu tDispT03 10
9.1. Proměnné 10
9.2. Metody 11
9.2.1. Init 11
9.2.2. Done 11
9.2.3. InitHwProc 11
9.2.4. Done 11
9.2.5. DoneHwProc 11
9.2.6. DispClrScr 11
9.2.7. ClearVideoRWM 11
9.2.8. DTickRefreshScr 12
9.2.9. MoveVideoRWM_ToHw 12
9.2.10. mDispContr 12
9.2.11. mFlLight 12
9.2.12. mPutRTS 12
9.2.13. mPutDTR 12

9.2.14. mGetDCD 12
9.2.15. mGetDSR 12
9.2.16. mGetCTS 12
9.2.17. mGetRI 13
9.2.18. mWriteEEPROM 13
9.2.19. mReadEEPROM 13

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. Přejmenované TbkGroundManager.AcceptBkGround → AssignBkGround a tDispT03.DtickRefreshScreen → DtickRefreshScr. Doplněné metody mFlLight, … mGetRI. Doplněná konstanta StWDPort.

1.2. Účel dokumentu

Tento dokument slouží jako popis jednotky implementující displej terminálu Term03.

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 G128x64.

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 objekt pro obsluhu grafického displej terminálu TERM03 a několik dalších pomocných objektů používaných objektem displeje.

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

```
const
  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
  StWDPort:byte=$FD;
```

Tato konstanta obsahuje stav řídicích signálů pro LCD, EEPROM, WatchDog a komunikace. Některé výstupy jsou multifunkční. Význam jednotlivých bytů (podrobněji v HW dokumentaci):

OUT

D7	D6	D5	D4	D3	D2	D1	D0
DTR	RTS	KT3	LED	KT2 WDI SKEE	KT1	CSEE	KT0 DIEE

IN

D7							D0
JP11	0	0	DOEE	RI	CTS	DSR	DCD

```
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. Tyto objekty jsou používány interně objektem displeje TERM03 **tDispT03** 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 vidopaměť obrazové roviny, kterou objekt spravuje.

```
vImageState:tVideoImageManState;
```

Proměnná **vImageState** uchovává stav objektu.

5.2. Metody

5.2.1. Init

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

Konstuktor **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;
```

Metoda **ClearImage** je statická metoda, která smaže videopaměť, kterou objekt spravuje.

5.2.3. Clear

```
procedure Clear;virtual;
```

Metoda **Clear** volá statickou metodu **ClearImage** pro smazání vidoopaměti.

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. AssignBkGround

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

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

9. Popis objektu tDispT03

```
type
  pDispT03=^tDispT03;
  tDispT03=object(tADisp);
```

Objektový typ **tDispT03** je potomek objektu abstraktního displeje **tADisp** a představuje displej terminálu TERM03. Typ **tDispT03** implementuje metody pro grafický displej s rozlišením 128 × 64 pixelů a definuje nové metody pro další funkce displeje terminálu TERM03.

9.1. Proměnné

```
vAddr:Word;
```

Proměnná **vAddr** obsahuje bázovou adresu displeje v I/O prostoru terminálu.

```
vAddrCS2CW:Word;
vAddrCS2SR:Word;
vAddrCS2DW:Word;
vAddrCS1CW:Word;
vAddrCS1SR:Word;
vAddrCS1DW:Word;
```

Tyto proměnné obsahují adresy jednotlivých registrů displeje v I/O prostoru terminálu.

```
vVideoRWM:tVideoRWM;
```

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

```
PomVideoRWM:tVideoRWM;
```

Proměnná **PomVideoRWM** představuje pomocnou videopaměť používanou interně.

```
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.

```
OldIntVec1C:procedure;
```

Proměnná **OldIntVec1C** uchovává adresu původní přerušovací rutiny vektoru 1Ch (v TERM03 obsluha obvodu WatchDog).

```
EEPromAdresBits:Byte;
```

Proměnná **EEPromAdresBits** uchovává počet adresových bitů osazené paměti EEPROM. Konstruktorem je nastavena na 7 (paměť 128 bytů).

9.2. Metody

9.2.1. Init

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

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, **NAdr** obsahuje bázovou adresu displeje v I/O prostoru terminálu. Nastavením **EnIniHwProc** na false lze potlačit implicitní volání metody **InitHwProc** v konstruktoru.

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. InitHwProc

```
procedure InitHwProc;virtual;
```

Metoda **InitHwProc** inicializuje hardware displeje a nahradí obsluhu přerušení 1Ch vlastní rutinou (zde obsluha obvodu WatchDog).

9.2.4. Done

```
destructor Done; virtual;
```

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

9.2.5. DoneHwProc

```
procedure DoneHwProc;
```

Metoda **DoneHwProc** ukončí práci hardware displeje.

9.2.6. DispClrScr

```
procedure DispClrScr;
```

Metoda **DispClrScr** smaže displej.

9.2.7. ClearVideoRWM

```
procedure ClearVideoRWM;
```

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

9.2.8. DTickRefreshScr

```
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 implementuje přesun obrazových dat do hardware displeje.

9.2.10. mDispContr

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

Metodou **mDispContr** se nastavuje kontrast displeje.

9.2.11. mFlLight

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

mFlLight je metoda pro nastavení jasu displeje.

9.2.12. mPutRTS

```
procedure mPutRTS(b :boolean);
```

Metoda **mPutRTS** nastavuje modemový signál Request To Send.

9.2.13. mPutDTR

```
procedure mPutDTR(b :boolean);
```

Metoda **mPutDTR** nastavuje modemový signál Data Terminal Ready.

9.2.14. mGetDCD

```
function mGetDCD:Boolean;
```

Metoda **mGetDCD** vrací nastavení modemového signálu Data Carrier Detect.

9.2.15. mGetDSR

```
function mGetDSR:Boolean;
```

Metoda **mGetDSR** vrací nastavení modemového signálu Data Send Ready.

9.2.16. mGetCTS

```
function mGetCTS:Boolean;
```

Metoda **mGetCTS** vrací nastavení modemového signálu Clear To Send.

9.2.17. mGetRI

```
function mGetRI:Boolean;
```

Metoda **mGetRI** vrací nastavení modemového signálu Ring Indicator.

9.2.18. mWriteEEPROM

```
procedure mWriteEEPROM(A:Word;D:byte); virtual;
```

Metoda **mWriteEEPROM** zapíše byte předaný parametrem **D** na adresu předanou parametrem **A** do sériové paměti EEPROM.

9.2.19. mReadEEPROM

```
function mReadEEPROM(A:Word):byte; virtual;
```

Metoda **mReadEEPROM** přečte byte z adresy předané parametrem **A** v sériové paměti EEPROM a vrátí ho jako návratovou hodnotu.