

G240x128 a G128x64

JEDNOTKY IMPLEMENTUJÍCÍ GRAFICKÉ OPERACE VE VIDEOPAMĚTI

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.Popis objektu tBitmapInfoObj	9
5.1. Proměnné	9
5.2. Metody	9
5.2.1. AssignBMPAddr	9
6.Proměnné jednotky	9
6.1. User_FontArray	9
6.2. User_BkBmps	9
7.Popis procedur a funkcí	10
7.1.1. GetCFontMetrics	10
7.2. Procedury pro práci s videopamětí	10
7.2.1. mClear	10
7.2.2. mDrawPoint	10
7.2.3. mDrawLine	10
7.2.4. mDrawCircle	11
7.2.5. mDrawBox	11
7.2.6. mFillBox	11
7.2.7. mTextOutXY	11
7.3. Procedury pro skládání videopaměti	11
7.3.1. RopWord_Not	12
7.3.2. RopWord_Or	12
7.3.3. RopWord_Xor	12
7.4. Zobrazovací procedury pro fonty	12
7.4.1. pViewFont	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

1.2. Účel dokumentu

Tento dokument slouží jako popis knihoven implementujících grafické operace ve videopaměti G240x128 a G128x64.

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 uTermGr a uBitMap.

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

Jednotky **G240x128** a **G128x64** implementují typ videopaměti pro rozměr 240 × 128 resp. 128 × 64 pixelů a veškeré základní grafické operace nad touto videopamětí. Jednotky jsou využívány zejména objekty grafických displejů pro vykreslení grafického obsahu obrazovek. Implementace obou jednotek je shodná, liší se pouze v rozměru deklarované videopaměti.

Jednotky nahrazují starší jednotku **uShwFnt**.

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.

```
cDispPixelWidth = 240; { G240x128 }
cDispPixelWidth = 128; { G128x64 }
```

Konstanta **cDispPixelWidth** definuje horizontální rozměr videopaměti v pixelech.

```
cDispPixelHeight = 128; { G240x128 }
cDispPixelHeight = 64;  { G128x64 }
```

Konstanta **cDispPixelHeight** definuje vertikální rozměr videopaměti v pixelech.

```
cBytesOnLine      = cDispPixelWidth shr 3;
```

Konstanta **cBytesOnLine** definuje velikost paměti alokovanou pro jeden mikrořádek videopaměti.

```
cBytesOnArea      = cBytesOnLine*cDispPixelHeight;
```

Konstanta **cBytesOnArea** definuje velikost paměti alokovanou pro celou plochu videopaměti.

```
tVideoStr         = string[80];
```

Typ **tVideoStr** je typ řetězce pro výpis textů do videopaměti.

```
pVideoRWM = ^tVideoRWM;
tVideoRWM =
  record
    case byte of
      0: (B: array[0..cBytesOnArea-1] of byte);
      1: (W: array[0..(cBytesOnArea shr 1)-1] of word);
    end;
```

Typ **tVideoRWM** implementuje videopaměť jako bitovou mapu. Videopaměť je organizovaná po řádcích, přičemž jednomu bytu odpovídá 8 pixelů umístěných vedle sebe. Nejvýznamnější bit (MSB) přísluší pixelu umístěnému v osmici nejvíce vlevo. Nastavení bitu má význam bílé barvy, nulování má význam černé barvy.

Příklad:

```
var V : tVideoRWM
```

```
[0,0]
```

V.B[0]	...	V.B[cBytesOnLine-1]
V.B[cBytesOnLine]	...	V.B[2*cBytesOnLine-1]
...
V.B[cBytesOnArea-cBytesOnLine]	...	V.B[cBytesOnArea-1]

```
[cDispPixelWidth-1,cDispPixelHeight-1]
```

Na následujícím obrázku je zvýrazněno umístění osmice pixelů V.B[2*cBytesOnLine+1]

```
[0,0]
```

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	...	7	6	5	4	3	2	1	0
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	...	7	6	5	4	3	2	1	0
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	...	7	6	5	4	3	2	1	0
...														
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	...	7	6	5	4	3	2	1	0

```
[cDispPixelWidth-1,cDispPixelHeight-1]
```

```
tCFontMetrics = record
    PixW :word;
    PixH :word;
end;
```

Typ **tCFontMetrics** je používán pro získání metriky fontu. Význam položek záznamu:

PixW Šířka znaku v pixelech.

PixH Výška znaku v pixelech.

```
tViewProc=procedure( X,Y:Integer; Str:tVideoStr;
                    Rop:tVideoROP; pVideo:pVideoRWM; pFont:pFont2);
```

Typ **tViewProc** je typ pro proceduru používanou pro výpis fontu do videopaměti. V parametrech **X** a **Y** se předává umístění levého horního rohu vypisovaného textu v pixelech, v parametru **Str** řetězec pro výpis, v parametru **Rop** požadovaná grafická operace, která má být použita pro výpis, v parametru **pVideo** ukazatel na videopaměť, do které bude výpis prováděn, a v parametru **pFont** ukazatel na datovou strukturu s popisem fontu použitého pro výpis.

```
tCFontArrayRec = record
    P :tViewProc;
    U :pFont2;
end;
```

Typ **tCFontArrayRec** je typ záznamu v poli uživatelských fontů. V položce **P** uchovává ukazatel na proceduru pro výpis příslušného fontu a v položce **U** ukazatel na datovou strukturu s popisem fontu.

```
cImplFonts      = 0;
```

Konstanta **cImplFonts** obsahuje index implicitního fontu v poli uživatelských fontů.

```
cSizeVideoRWM   = SizeOf(tVideoRWM);
```

Konstanta **cSizeVideoRWM** obsahuje velikost paměti alokovanou pro videopaměť.

```
tCFontArrayIndex= cImplFonts..cNumFonts-1;
```

Typ **tCFontArrayIndex** je použit pro indexování pole uživatelských fontů. Konstanta **cNumFonts** je deklarována v jednotce **uTermGr** a obsahuje maximální počet uživatelských fontů.

```
tCFontArray      = array[tCFontArrayIndex] of tCFontArrayRec;
```

Typ **tCFontArray** je typ pole uživatelských fontů.

```
tBkBMPsIndex     = 0..BkBMPsCount-1;
```

Typ **tBkBmpsIndex** je typ pro indexování pole uživatelských bitmap. Konstanta **BkBmpsCount** je deklarována v jednotce **uTermGr** a obsahuje maximální počet uživatelských bitmap.

```
tABkBMPs         = array[tBkBMPsIndex] of tBitMapInfoObj;
```

Typ **tABkBmps** je typ pole uživatelských bitmap. Obsahuje ukazatele na instance objektu **tBitmapInfoObj** s popisem uživatelských bitmap.

```
const
    pViewFont08x08 : tViewProc = pViewFont;
    pViewFont12x16 : tViewProc = pViewFont;
    pViewFont16x32 : tViewProc = pViewFont;
    pViewFont24x64 : tViewProc = pViewFont;
```

Konstanty **pViewFontXXXX** jsou naplněny odkazem na implicitní výpisovou proceduru **pViewFont**. Jsou zavedeny pro zpětnou kompatibilitu.

5. Popis objektu tBitmapInfoObj

Objektový typ tBitmapInfoObj slouží k uchování popisu uživatelské bitmapy. Je používán v poli uživatelských bitmap.

5.1. Proměnné

BmpHeaderPtr : pBmpHeader;

Proměnná **BmpHeaderPtr** obsahuje ukazatel na hlavičku bitmapy. Hlavička uchovává rozměry bitmapy a typ jejího zakódování.

BmpImagePtr : pAByte;

Proměnná **BmpImagePtr** obsahuje ukazatel na binární obraz bitmapy.

BmpImageSize : word;

Proměnná **BmpImageSize** obsahuje velikost paměti alokované pro binární obraz bitmapy.

5.2. Metody

5.2.1. AssignBMPAddr

```
procedure AssignBMPAddr(p:pointer;var Lng:word);
```

Metoda **AssignBMPAddr** přiřadí proměnným objektu hodnoty podle umístění bitmapy v paměti. V parametru p je předán ukazatel na bitmapu a v parametru Lng velikost bitmapy. Metoda se používá pro instalaci uživatelské bitmapy.

6. Proměnné jednotky

6.1. User_FontArray

User_FontArray : tCFontArray;

Proměnná **User_FontArray** obsahuje pole uživatelských fontů. Každý z prvků pole obsahuje dvě položky: ukazatel na datovou strukturu s popisem fontu a ukazatel na proceduru, kterou je příslušný font zobrazován (zobrazovací procedura). Není-li instalována uživatelská zobrazovací procedura, použije se implicitní zobrazovací procedura **pViewFont**.

Proměnná je v inicializační části jednotky nulována. Před použitím fontů v aplikaci je třeba naplnit jednotlivé položky pole příslušnými ukazateli na datové struktury fontů a případně ukazateli na uživatelské zobrazovací procedury.

6.2. User_BkBmps

User_BkBMPs : tABkBMPs;

Proměnná **User_BkBmps** obsahuje pole uživatelských bitmap. Jednotlivé prvky pole jsou instancemi objektu **tBitmapInfoObj**, které uchovávají popis bitmapy (ukazatel na hlavičku bitmapy, ukazatel na binární obraz bitmapy a jeho velikost).

Proměnná je v inicializační části jednotky nulována. Před použitím uživatelských bitmap v aplikaci je třeba naplnit jednotlivé položky pole příslušnými ukazateli (např. voláním metody **tBitmapInfoObj.AssignBMPAddr**).

7. Popis procedur a funkcí

7.1.1. GetCFontMetrics

```
function GetCFontMetrics(FIndex:integer;  
                        var CFntMetrics:tCFontMetrics):Boolean;
```

Funkce **GetFontMetrics** slouží k získání metriky fontu (šířky a výšky znaku). Parametrem **FIndex** se předává index fontu v poli uživatelských fontů. Pokud je parametr **FIndex** mimo meze pole uživatelských fontů, použije se implicitní font (font číslo 0). Je-li zadáný font instalován (přiřazen v poli uživatelských fontů) vrací funkce jako návratovou hodnotu **true** a v parametru **CFntMetrics** metriku fontu. V opačném případě vrací **false** a hodnota v parametru **CFntMetrics** je neplatná.

7.2. Procedury pro práci s videopamětí

Následující procedury jsou určeny pro operace s videopamětí. Tyto procedury jsou používány objekty grafických displejů pro vytvoření binární podoby grafického obsahu obrazovky. Uživatel je také může použít pro přímé kreslení do videopaměti. (např. implementované objektem **tGraphicArea**)

7.2.1. mClear

```
procedure mClear(var vVideoRWM :tVideoRWM);
```

Procedura **mClear** smaže obsah videopaměti předané v parametru **vVideoRWM**.

7.2.2. mDrawPoint

```
procedure mDrawPoint( X,Y:Integer; Rop:tVideoROP;  
                    var vVideoRWM :tVideoRWM);
```

Procedura **mDrawPoint** vykreslí do videopaměti předané v parametru **vVideoRWM** grafickou operací **Rop** bod o souřadnicích **X** a **Y**.

7.2.3. mDrawLine

```
procedure mDrawLine( X1,Y1,X2,Y2:Integer;Rop:tVideoROP;  
                    var vVideoRWM :tVideoRWM);
```

Procedura **mDrawLine** vykreslí do videopaměti předané v parametru **vVideoRWM** grafickou operací **Rop** úsečku s koncovými body o souřadnicích **X1**, **Y1** a **X2**, **Y2**.

7.2.4. mDrawCircle

```
procedure mDrawCircle( Xr,Yr,R:Integer; Rop:tVideoROP;  
                      var vVideoRWM :tVideoRWM);
```

Procedura **mDrawCircle** vykreslí do videopaměti předané v parametru **vVideoRWM** grafickou operací **Rop** kružnici se středem o souřadnicích **Xr**, **Yr** a poloměrem **R**.

7.2.5. mDrawBox

```
procedure mDrawBox( X1,Y1,X2,Y2:Integer;Rop:tVideoROP;  
                  var vVideoRWM :tVideoRWM);
```

Procedura **mDrawBox** vykreslí do videopaměti předané v parametru **vVideoRWM** grafickou operací **Rop** obdélník. V parametrech **X1**, **Y1** a **X2**, **Y2** jsou předávány souřadnice dvou protilehlých rohů obdélníku.

7.2.6. mFillBox

```
procedure mFillBox( X1,Y1,X2,Y2:Integer;Rop:tVideoROP;  
                  var vVideoRWM :tVideoRWM);
```

Procedura **mFillBox** vykreslí do videopaměti předané v parametru **vVideoRWM** grafickou operací **Rop** vyplněný obdélník. V parametrech **X1**, **Y1** a **X2**, **Y2** jsou předávány souřadnice dvou protilehlých rohů obdélníku.

7.2.7. mTextOutXY










```
procedure mTextOutXY( X,Y:Integer;Str:tVideoStr; Rop:tVideoROP;  
                    var vVideoRWM:tVideoRWM;FIndex:Integer);
```

Procedura **mTextOutXY** vypíše do videopaměti předané v parametru **vVideoRWM** grafickou operací **Rop** text předaný parametrem **Str**. V parametrech **X**, **Y** se předávají souřadnice levého horního rohu výpisu textu a v parametru **FIndex** index fontu, který bude použit pro výpis, v poli uživatelských fontů.

7.3. Procedury pro skládání videopaměti

Následující procedury se používají pro skládání videopaměti zadanou grafickou operací. Procedury realizují příslušnou grafickou operaci nad obecným bufferem. Jsou používány objekty grafických displejů pro složení výsledné grafické obrazovky z dílčích grafických stránek.

Grafické operace:

grafická operace	zdroj před operací	cíl před operací	cíl po operaci
b_Or			
b_Not			
b_Xor			

7.3.1. RopWord_Not

```
procedure RopWord_Not(DestBuf,SrcBuf:pointer;
                     BufLng:word);
```

Procedura **RopWord_Not** provádí skládání videopaměti operací **b_Not**. Parametr **DestBuf** obsahuje ukazatel na cílový buffer a parametr **SrcBuf** ukazatel na zdrojový buffer. **BufLng** udává délku dat bufferů. Obsahy obou bufferů jsou složeny operací **b_Not** a výsledek je uložen do cílového bufferu.

7.3.2. RopWord_Or

```
procedure RopWord_Or (DestBuf,SrcBuf:pointer;
                     BufLng:word);
```

Procedura **RopWord_Or** provádí skládání videopaměti operací **b_Or**. Parametr **DestBuf** obsahuje ukazatel na cílový buffer a parametr **SrcBuf** ukazatel na zdrojový buffer. **BufLng** udává délku dat bufferů. Obsahy obou bufferů jsou složeny operací **b_Or** a výsledek je uložen do cílového bufferu.

7.3.3. RopWord_Xor

```
procedure RopWord_Xor(DestBuf,SrcBuf:pointer;
                     BufLng:word);
```

Procedura **RopWord_Xor** provádí skládání videopaměti operací **b_Xor**. Parametr **DestBuf** obsahuje ukazatel na cílový buffer a parametr **SrcBuf** ukazatel na zdrojový buffer. **BufLng** udává délku dat bufferů. Obsahy obou bufferů jsou složeny operací **b_Xor** a výsledek je uložen do cílového bufferu.

7.4. Zobrazovací procedury pro fonty

Tyto procedury jsou používány pro výpis textu do videopaměti. Musí používat vzdálený model volání (far) a mít hlavičku shodnou s typem tViewProc. V jednotce je implementována implicitní zobrazovací procedura **pViewFont**. Uživatel může implementovat vlastní zobrazovací proceduru, kterou pak může instalovat do pole uživatelských fontů. Takto je umožněno implementovat vlastní způsob výpisu fontu, např. transformaci souřadnic výpisu nebo tabulky znaků fontu.

7.4.1. pViewFont

```
procedure pViewFont( X,Y:Integer;Str:tVideoStr;  
    Rop:tVideoROP;pVideo:pVideoRWM;pFont:pFont2);far;
```

Procedura **pViewFont** je implicitní zobrazovací procedura, která se použije pro výpis fontu, není-li instalována uživatelská zobrazovací procedura. Procedura **pViewFont** zobrazuje fonty až do šířky 24 pixelů. Parametry **X,Y** udávají souřadnice levého horního rohu výpisu v pixelech, parametrem **Str** je předáván text pro výpis, parametrem **Rop** grafická operace, parametrem **pVideo** ukazatel na videopaměť a parametrem **pFont** ukazatel na datovou strukturu použitého fontu.