

# GraphWin

JEDNOTKA PRO PRÁCI S  
OBRAZOVKOU A KLÁVESNICÍ V  
GRAFICKÉM REŽIMU

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: 22.05.2003

Datum posledního uložení dokumentu: 22.05.2003

(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 TGraphWin	6
5.1. Pole	6
5.2. Metody	7
5.2.1. Init	7
5.2.2. Done	7
5.2.3. LockWin	7
5.2.4. UnLockWin	7
5.2.5. Save	8
5.2.6. Restore	8
6.Popis proměnných	8
7.Popis procedur	8
7.1. GraphInit	8
8.Příklad použití GraphWin	8



## 1. O dokumentu

---

### 1.1. Revize dokumentu

---

Verze dokumentu	Verze SW	Autor	Datum vydání	Popis změn
1.00	1.XX	Ku		První vydání.
1.10	2.XX	Tu	22.05.2003	Úprava dokumentu dle ISO9000.

### 1.2. Účel dokumentu

---

Tento dokument slouží jako popis jednotky pro práci s obrazovkou a klávesnicí v grafickém režimu.

### 1.3. Rozsah platnosti

---

Určen pro programátory a uživatele programového vybavení SofCon.

### 1.4. Související dokumenty

---

Pro čtení tohoto dokumentu není potřeba číst žádný další manuál, ale je potřeba se orientovat v používání programového vybavení SofCon.

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 GraphWin byla vytvořena pro zjednodušení práce uživatele při přístupu paralelních procesů k obrazovce a klávesnici počítače třídy PC. Programátor při práci s obrazovkou může využívat standardní služby pro práci v grafickém režimu poskytované překladačem. Před jejich použitím je však nutné, aby proces, voláním služby jednotky GraphWin, získal výhradní právo přístupu k obrazovce a klávesnici. Jednotka zároveň poskytuje prostředky pro vytváření pracovních ploch, tzv. oken, které práci s obrazovkou značně usnadní. Jednotka se doporučuje používat v kombinaci s jednotkou UMGaph.

### 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.

### 5. Popis objektu TGraphWin

---

```
PGraphWin = ^TGraphWin;
TGraphWin = object
```

Objekt umožňuje vytvářet na obrazovce pracovní plochy, tzv. okna. Při výpisu na takto vytvořená okna je nutno dodržet následující postup. Nejprve je třeba voláním metody **LockWin** získat výhradní právo přístupu k obrazovým službám a k danému oknu. Poté lze provést výstup na obrazovku, změnu parametrů pro výstup na obrazovku nebo vstup z klávesnice. (Služby jednotek Graph a UMGaph). Na konec je třeba voláním metody **UnLockWin** zrušit přístup k obrazovým službám a k danému oknu. Metody **LockWin** a **UnLockWin** je možné do sebe vnořovat. Zároveň je možné, aby vnořené metody byly od různých instancí (oken).

#### 5.1. Pole

---

```
wRect : tGraphRect;
```

**wRect** obsahuje údaje o umístění, velikosti a ořezávání vytvořeného okna. Každé okno je umístěno absolutně vzhledem k základní ploše **MainWin**. Zadává se X,Y,W,H: byte; X,Y=souřadnice levého horního rohu, W=šířka, H=výška a Clip který určuje, zda se má kolem okna ořezávat ClipOn nebo ne ClipOff. Proměnná je inicializována parametrem Rect konstruktoru Init.

```
wX,wY : Integer;
```

**wX,wY** jsou souřadnice umístění kurzoru v daném okně. Po vytvoření okna je kurzor nastaven na pozici 1, 1. Proměnné jsou aktualizovány při volání metody **UnLockWin**.

```
wBkColor : Integer;
```

**wBkColor** je údaj udávající barvu pozadí grafického okna. Při nastavování lze použít konstant nadefinovaných v jednotce Graph. Přiřazení barev je následující:

0 černá

1 modrá

3 blankytně modrá

- 4 červená
- 5 fialová
- 6 hnědá
- 7 světle šedá
- 8 tmavě šedá
- 9 modrá přisvětlená
- 10 zelená přisvětlená
- 11 blankytně modrá přisvětlená
- 12 červená přisvětlená
- 13 fialová přisvětlená
- 14 žlutá
- 15 bílá

Po vytvoření okna je proměnná inicializována hodnotou **AW** konstruktoru **Init** a je aktualizována při volání metody **UnLockWin**.

wColor : Integer;

**wColor** je údaj udávající barvu rámečku a aktuálního pera kreslicího v grafickém okně. Přiřazení barev je ekvivalentní parametru wBkColor.

## 5.2. Metody

---

### 5.2.1. Init

constructor Init(Rect: tGraphRect; CFrame: Byte);

Voláním konstruktoru **Init** vytvoříme nové okno dané velikosti s daným umístěním na obrazovce. **Rect** určuje umístění a velikost vytvářeného okna. **CFrame** je barva rámečku okna. Za barvu pozadí a popředí se dosadí naposledy nastavené hodnoty.

### 5.2.2. Done

destructor Done;

Destruktor **Done** slouží pro zrušení objektu daného okna.

### 5.2.3. LockWin

procedure LockWin;

Voláním metody je získáno výhradní právo pro přístup k zobrazovacím službám. Současně se dané okno stane oknem aktuálním. Uvnitř metody **LockWin** je volající proces pozastaven do chvíle než získá výhradní přístup k obrazovým službám. Při následném volání procedur **Výpisu na obrazovku** bude výpis proveden právě do okna určeného danou instancí objektu **tGraphWin**. Při čekání na znak z klávesnice, při volání procedur **Čtení z klávesnice**, je při čekání na znak uvolněn přístup k obrazovým službám pro jiné procesy. Po stisku klávesy je tento přístup opět získán. Uživatel musí zajistit, aby klávesnicové služby volal zároveň jen jeden proces.

### 5.2.4. UnLockWin

procedure UnLockWin;

Metoda obnoví okno, které bylo aktivní před voláním odpovídající metody **LockWin** a případně uvolní přístup k zobrazovacím službám. Metody **LockWin** a **UnLockWin** je možné vnořovat do sebe. Lze vnořovat metody i od různých instancí objektu (oken).

### 5.2.5. Save

```
procedure Save;
```

Metoda slouží jen pro vnitřní použití.

### 5.2.6. Restore

```
procedure Restore;
```

Metoda slouží jen pro vnitřní použití.

## 6. Popis proměnných

---

```
MainWin : PGraphWin=nil;
```

**MainWin** je ukazatel na instanci objektu **tGraphWin** vzniklou při volání procedury **GraphInit**. Popisuje hlavní okno na obrazovce.

```
ActWin : PGraphWin=nil;
```

**ActWin** je ukazatel na aktuální okno na obrazovce.

```
WaitKey : Word=2;
```

**WaitKey** je počet wait stavů jádra Retos při čekání na stisk klávesy.

## 7. Popis procedur

---

### 7.1. GraphInit

---

```
procedure GraphInit
```

Procedura slouží k inicializaci jednotky GraphWin a k vytvoření hlavní pracovní plochy o velikosti odpovídající maximální velikosti obrazovky aktuálního obrazovému režimu. Na tuto plochu je možné se odkazovat prostřednictvím instance MainWin. Proceduru **GraphInit** je možné volat jen při spuštění jádra operačního systému ReTOS. Tuto proceduru je nutné zavolat před vytvořením instancí objektu **tGraphWin**.

## 8. Příklad použití GraphWin

---

Ukázkový program vytváří tři různé typy procesů. Proces P1 vypisuje opakovaně na obrazovku čas v sekundách. Proces P2 čte řetězec z klávesnice a čeká na řetězec "quit". Proces P3 kreslí na obrazovce graf sinusovky. Hlavní program vytvoří instance všech procesů. Celkem jsou tedy spuštěny tři procesy, které se střídají při přístupu k obrazovým službám.



```

uses
  Kernel,
  Graph,
  UMLGraph,
  GraphWin;

const
  Tm=18;
  FlEnd:Boolean=False;
  NbWindow=3;
  ARect:array[1..NbWindow]of tGraphRect=
    ((X: 2;Y: 2;W:500;H:300;Clip:ClipOn),
     (X:510;Y: 2;W:120;H:300;Clip:ClipOn),
     (X: 2;Y:310;W:630;H: 30;Clip:ClipOn));

procedure P1(Rect: tGraphRect; FColors:Integer; X,Y:Integer;
  BColor,FColor:Integer); far;
var
  Win : PGraphWin;
  i : Longint;
begin
  Start('Proc1',8000,200,255);Exit;
  Win:=New(PGraphWin,Init(Rect,FColors));
  Win^.LockWin;
  DispValS(X+30,Y,'Time [s]',8,BColor,FColor);
  Win^.UnlockWin;
  i:=1;
  repeat
    Win^.LockWin;
    DispValR(X,Y,i,3,0,BColor,FColor);
    Win^.UnlockWin;
    Wait(Tm);
    {$F-}
    Inc(i);
    {$F+}
  until FlEnd;
end;

procedure P2(Rect: tGraphRect; FColors:Integer; X,Y:Integer;
  BColor,FColor:Integer); far;
var
  Win : PGraphWin;
  S : String;
  ReadEnd : Boolean;
begin
  Start('Proc2',8000,200,255);Exit;
  Win:=New(PGraphWin,Init(Rect,FColors));
  Win^.LockWin;
  DispValS(X,Y-10,'quit ... write quit',11,BColor,FColor);
  Win^.UnlockWin;
  repeat
    S:='';
    ReadEnd:=False;
    repeat
      Win^.LockWin;
      BuildString(X,Y,10,BColor,FColor,S,ReadEnd);
      Win^.UnlockWin;
      Wait(1){musi byt, jinak blbne}
    until ReadEnd;
    Win^.LockWin;
    DispValS(X,Y,'',10,BColor,FColor);
    Win^.UnlockWin;
    FlEnd:=(S='quit')or(S='QUIT');
  until FlEnd;
end{P2};
{$F+}
procedure P3(Rect: tGraphRect; FColors:Integer);

```

```

{$F-}
const
  YMin:TARealP=(-1.2,-1.2,-1.2,-1.2,-1.2,-1.2,-1.2,-1.2,-1.2,-1.2);
  YMax:TARealP=(1.2,1.2,1.2,1.2,1.2,1.2,1.2,1.2,1.2,1.2);
  PCol:TAWord=(15,14,13,12,11,10,9,8,7,6);
  PopisX:tAxisTextStr='Ó[°]   ';
  PopisY:tAxisTextStr='y';
  max=60;
var
  j:TNumberOfY;
  k:0..360;
  Win : PGraphWin;
begin
  Start('Procl',8000,200,255);Exit;
  repeat
    Win:=New(PGraphWin,Init(Rect,FColors));
    Win^.LockWin;
    UMSetViewPort(-45,380,YMin,YMax,PCol);
    UMPutXAxis(FColors,0,0,3,60,FColors,TopText,4,0,PopisX);
    UMPutYAxis(FColors,0,0,3,0.2,FColors,RightText,2,1,PopisY);
    Win^.UnLockWin;
    for j:=1 to 10 do
      begin
        for k:=0 to max do
          begin
            Win^.LockWin;
            if k=0 then
              UMMoveTo(360/max*k,sin(2*Pi*(j/10+k/max)),j)
            else
              UmLineTo(360/max*k,sin(2*Pi*(j/10+k/max)),j);
            if (k/(max/6))=(k div (max div 6)) then UMMark(3,j);
            Win^.UnLockWin;
            Wait(1);
          end;
        end;
        Win^.LockWin;
        ClearViewPort;
        Win^.UnLockWin;
        dispose(Win);
      until FlEnd;
    end{P3};

  begin{Main}
    StartMain(200,255);           { init Kernelu }
    InitInterruptStack(1,254);
    StartTimeSlicing(8);

    GraphInit;

    P3(ARect[1],3);              { graf }
    P1(ARect[2],2, 10,140, 0, 2); { citac }
    P2(ARect[3],4,210,15, 0, 4);  { komunikace }

    SetPriority(10,255);

    repeat
      until FlEnd;

    Abort('*');
    TerminateMain;

  end{Main}.

```