

# TEDrv

## OVLADAČE KLÁVESNICE, MYŠI A DISPLEJE EMULÁTORU TERMINÁLU PRO MS WINDOWS

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: 23.01.2004

Datum posledního uložení dokumentu: 23.01.2004

(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	4
1.1. Revize dokumentu	4
1.2. Účel dokumentu	4
1.3. Rozsah platnosti	4
1.4. Související dokumenty	4
2.Termíny a definice	4
3.Úvod	5
3.1. Účel knihovny TEDrv	5
3.1.1. Ovladač klávesnice	5
3.1.1.1. Podporované kódy kláves	5
3.1.2. Ovladač myši	6
3.1.3. Ovladače displeje	6
3.1.4. Použití ovladačů emulátoru terminálu	6
4.Reference	7
4.1. Třídy	7
4.1.1. Třída TTEMouseDriver	7
4.1.1.1. Položka TTEMouseDriver.Buttons	7
4.1.1.2. Položka TTEMouseDriver.Position	7
4.1.1.3. Konstruktor TTEMouseDriver.Init	7
4.1.1.4. Destruktor TTEMouseDriver.Done	8
4.1.1.5. Metoda TTEMouseDriver.Initialize	8
4.1.1.6. Metoda TTEMouseDriver.GetEvent	8
4.1.1.7. Metoda TTEMouseDriver.SetDbClickDelay	9
4.1.1.8. Metoda TTEMouseDriver.GetDbClickDelay	9
4.1.1.9. Metoda TTEMouseDriver.SetDbClickArea	10
4.1.1.10. Metoda TTEMouseDriver.GetDbClickArea	10
4.1.1.11. Metoda TTEMouseDriver.SetRepeatDelay	10
4.1.1.12. Metoda TTEMouseDriver.GetRepeatDelay	11
4.1.1.13. Metoda TTEMouseDriver.SetRepeatRate	11
4.1.1.14. Metoda TTEMouseDriver.GetRepeatRate	11
4.1.2. Třída TTEKeybDriver	12
4.1.2.1. Metoda TTEKeybDriver.GetEvent	12
4.1.3. Třída TTEDisplayDriver	12
4.1.3.1. Konstruktor TTEDispDriver.Init	13
4.1.3.2. Metoda TTEDispDriver.Initialize	13
4.1.3.3. Metoda TTEDispDriver.Finalize	13

---

## 1. O dokumentu

---

### 1.1. Revize dokumentu

---

Verze dokumentu	Verze SW	Autor	Datum vydání	Popis změn
1.00	1.00	Cr	21.01.2004	První vydání

### 1.2. Účel dokumentu

---

Tento dokument slouží jako popis knihovny TEDrv, která je součástí balíku vizualizačních knihoven pro jednotku KIT.

### 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 IoDrv.

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

### 3.1. Účel knihovny TEDrv

Knihovna PCDrv slouží ke spuštění aplikace určené pro terminál fy SofCon v emulačním prostředí na PC. Knihovna obsahuje ovladače klávesnice (**TTEKeybDriver**) a myši (**TTEMouseDriver**) a dva ovladače videokarty v monochromatické VGA režimu o rozlišení 640x480 pixelů (**TVGAMonoDriver**) a ovladač VGA karty v některém 8bpp barevných VESA režimů (**TVESADriver**).

Aplikace používající tuto knihovnu musí být spuštěna pod operačním systémem Microsoft Windows 2000/XP nebo vyšším. Tato knihovna zároveň vyžaduje nainstalovaný ovladač virtuálního zařízení (VDD) termemu.dll, který je součástí aplikace TermEmu (Emulátor terminálu).

#### 3.1.1. Ovladač klávesnice

Knihovna TEDrv obsahuje třídu **TTEKeybDriver**, která implementuje ovladač standardní klávesnice připojené k PC. Ovladač vychází z abstraktní třídy **TKeyboardDriver** implementované v knihovně IoDrv. Ovladač **TTEKeybDriver** je napojen na okno emulátoru terminálu (tj. aplikaci TermEmu).

Ovladač implementuje metodu **GetEvent**, která vrací událost evKeyDown s kódem klávesy umístěné na začátku fronty řadiče klávesnice.

##### 3.1.1.1. Podporované kódy kláves

Ovladač klávesnice **TTEKeybDriver** podporuje kódy speciálních kláves uvedené v následující tabulce. Jednotlivé konstanty s prefixem kb\_ jsou popsány v dokumentaci ke knihovně IoDrv.

kbF1	kbCtrlF1	kbAltA	kbAltU	kbInsert
kbF2	kbCtrlF2	kbAltB	kbAltV	kbDelete
kbF3	kbCtrlF3	kbAltC	kbAltW	kbBackSpace
kbF4	kbCtrlF4	kbAltD	kbAltX	kbTab
kbF5	kbCtrlF5	kbAltE	kbAltY	kbEsc
kbF6	kbCtrlF6	kbAltF	kbAltZ	kbShiftTab
kbF7	kbCtrlF7	kbAltG	kbAlt0	kbEnter
kbF8	kbCtrlF8	kbAltH	kbAlt1	kbShiftEnter
kbF9	kbCtrlF9	kbAltI	kbAlt2	kbLeft
kbF10	kbCtrlF10	kbAltJ	kbAlt3	kbRight
kbShiftF1	kbAltF1	kbAltK	kbAlt4	kbUp
kbShiftF2	kbAltF2	kbAltL	kbAlt5	kbDown
kbShiftF3	kbAltF3	kbAltM	kbAlt6	kbHome
kbShiftF4	kbAltF4	kbAltN	kbAlt7	kbEnd
kbShiftF5	kbAltF5	kbAltO	kbAlt8	kbPageUp
kbShiftF6	kbAltF6	kbAltP	kbAlt9	kbPageDown
kbShiftF7	kbAltF7	kbAltQ		kbShiftLeft

kbShiftF8	kbAltF8	kbAltR		kbShiftRight
kbShiftF9	kbAltF9	kbAltS		kbShiftUp
kbShiftF10	kbAltF10	kbAltT		kbShiftDown

### 3.1.2. Ovladač myši

Knihovna TEDrv obsahuje třídu **TTEMouseDriver** (viz. kapitola 4.1.1), která implementuje ovladač myši emulátoru terminálu. Ovladač vychází z abstraktní třídy **TMouseDriver** implementované v knihovně IoDrv. Tento ovladač nahrazuje v simulačním prostředí na PC ovladač dotykového panelu terminálu. Ovladač myši je napojen na okno emulátoru terminálu (tj. aplikaci TermEmu) a generuje události pouze tehdy, jestliže uživatel pohybuje s myši v okně této aplikace.

### 3.1.3. Ovladače displeje

Knihovna TEDrv obsahuje ovladač displeje **TTEDispDriver** (viz. kapitola 4.1.3), který vychází z abstraktní třídy **TDisplayDriver** implementované v knihovně IoDrv. Tento ovladač je napojen na okno emulátoru terminálu (tj. aplikace TermEmu) spuštěné ve Windows.

### 3.1.4. Použití ovladačů emulátoru terminálu

Následující příklad ukazuje, jak vytvořit základní komponentu aplikace **TApplication** v simulačním prostředí na PC.

```
var
  App : PApplication;

App :=
  New( PApplication, Init (
    New( PInputDriver, Init(
      New( PTEKeybDriver, Init ),
      New( PTEMouseDriver, Init ),
    )),
    New( PTEDispDriver, Init( 320, 240 ) ),
    @g_AppSettings
  ));
```

Proměnná `g_AppSettings` obsahuje nastavení ovladače displeje a klávesnice terminálu.

---

## 4. Reference

---

---

### 4.1. Tříd

---

#### 4.1.1. Třída TTEMouseDriver

Třída **TTEMouseDriver** implementuje ovladač myši emulátoru terminálu. Tato třída vychází z bázové třídy pro implementaci ovladačů myši **TMouseDriver** (viz. dokumentace ke knihovně IoDrv)

```
PTEMouseDriver = ^TTEMouseDriver;
TTEMouseDriver = object( TMouseDriver )
public
  Buttons      : Byte;
  Position     : TPoint;

  constructor Init;
  destructor Done; virtual;
  function Initialize: Boolean; virtual;
  procedure Finalize; virtual;
  procedure Tick; virtual;
  procedure GetEvent( var AEvent: TEvent ); virtual;

  procedure SetDbClickDelay( AValue: Integer ); virtual;
  function GetDbClickDelay: Integer; virtual;
  procedure SetDbClickArea( AValue: Integer ); virtual;
  function GetDbClickArea: Integer; virtual;
  procedure SetRepeatDelay( AValue: Integer ); virtual;
  function GetRepeatDelay: Integer; virtual;
  procedure SetRepeatRate( AValue: Integer ); virtual;
  function GetRepeatRate: Integer; virtual;
end;
```

##### 4.1.1.1. Položka TTEMouseDriver.Buttons

Položka **Buttons** obsahuje aktuální stav tlačítek myši. Položka je aktualizována automaticky a je určena pouze pro čtení.

```
Buttons      : Byte;
```

Položka obsahuje kombinaci příznaků mbLeft, mbRight a mbCenter pro levé, pravé a prostřední tlačítko myši (viz. dokumentace ke knihovně IoDrv)

##### 4.1.1.2. Položka TTEMouseDriver.Position

Položka **Position** obsahuje aktuální souřadnice myši. Položka je aktualizována automaticky a je určena pouze pro čtení.

```
Position     : TPoint;
```

##### 4.1.1.3. Konstruktor TTEMouseDriver.Init

Konstruktor **Init** provádí inicializaci instance třídy.

```
constructor Init;
```

**Parametry:**

Konstruktor nemá žádné parametry.

**Návratové hodnoty:**

Konstruktor nevrací žádnou hodnotu.

**Poznámky:**

Konstruktor nastaví parametry ovladače na implicitní hodnoty, tj.

Oblast dvojkliku	2 pixely
Zpoždění detekce dvojkliku	300 ms
Zpoždění před opakováním	500 ms
Perioda opakování	150 ms

#### 4.1.1.4. Destruktor TTEMouseDriver.Done

#### 4.1.1.5. Metoda TTEMouseDriver.Initialize

Metoda **Initialize** provádí detekci emulátoru terminálu a vyprázdnění fronty událostí myši.

```
function Initialize: Boolean; virtual;
```

**Parametry:**

Metoda nemá žádné parametry.

**Návratové hodnoty:**

Metoda vrací hodnotu True v případě úspěšné inicializace hardware myši.

**Poznámky:**

Metoda **Initialize** předefinovává metodu **Initialize** báze třídy **TMouseDriver** (viz. dokumentace ke knihovně IoDrv).

#### 4.1.1.6. Metoda TTEMouseDriver.GetEvent

Metoda **GetEvent** předá nejstarší událost a odstraní ji z fronty událostí.

```
procedure GetEvent( var AEvent: TEvent ); virtual;
```

**Parametry:**



**AEvent** Po provedení metody je do parametru AEvent uložena událost typu evMouseXXX a jsou vyplněny položky Buttons a Pos. V případě, že ve fronta událostí ovladače myši je prázdná, pak je vyplněna pouze položka Code hodnotou evNothing.

#### Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

#### Poznámky:

Ovladač myši generuje následující uvedené v tabulce níže. U všech typů událostí jsou vyplněny položky Pos a Buttons struktury **TEvent**.

<b>Událost</b>	<b>Popis události</b>
evMouseDown	Stisk tlačítka myši
evMouseUp	Uvolnění tlačítka myši
evMouseMove	Pohyb ukazatele myši (tlačítka mohou být stisknuta i uvolněna)
evMouseDbI	Dvojklik tlačítka myši
evMouseRep	Automatické opakování stisku tlačítka

#### 4.1.1.7. Metoda TTEMouseDriver.SetDbIClickDelay

Metoda **SetDbIClickDelay** slouží pro nastavení maximálního časového intervalu mezi dvěma kliknutími detekovanými jako dvojklik.

#### Parametry:

**AValue** Délka časového intervalu v milisekundách.

#### Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

#### Poznámky:

#### 4.1.1.8. Metoda TTEMouseDriver.GetDbIClickDelay

Metoda **GetDbIClickDelay** vrací hodnotu maximálního časového intervalu mezi dvěma kliknutími detekovanými jako dvojklik.

#### Parametry:

Metoda nemá žádné parametry.

**Návratové hodnoty:**

Metoda vrací délku časového intervalu v milisekundách.

**Poznámky:****4.1.1.9. Metoda TTEMouseDriver.SetDbClickArea**

Metoda **SetDbClickArea** slouží k nastavení velikosti oblasti dvojkliku, tj. maximální vzdálenosti mezi pozicema dvou kliknutí, detekovanými jako dvojklik.

**Parametry:**

AValue            Počet pixelů.

**Návratové hodnoty:**

Metoda nevrací žádnou hodnotu.

**Poznámky:**

Optimální hodnota velikosti oblasti dvojkliku závisí na rozměrech bodu displeje. Obvykle se pohybuje mezi 4 až 8 pixely.

**4.1.1.10. Metoda TTEMouseDriver.GetDbClickArea**

Metoda **GetDbClickArea** vrací velikost oblasti dvojkliku, tj. maximální vzdálenosti mezi pozicema dvou kliknutí, detekovanými jako dvojklik

**Parametry:**

Metoda nemá žádné parametry.

**Návratové hodnoty:**

Metoda vrací počet pixelů..

**Poznámky:****4.1.1.11. Metoda TTEMouseDriver.SetRepeatDelay**

Metoda **SetRepeatDelay** slouží k nastavení zpoždění generování události evMouseRep při držení tlačítka myši na jednom místě po delší dobu.

**Parametry:**

AValue            Délka časového intervalu v milisekundách.

**Návratové hodnoty:**

Metoda nevrací žádnou hodnotu.

**Poznámky:**

#### 4.1.1.12. Metoda TTEMouseDriver.GetRepeatDelay

Metoda **GetRepeatDelay** vrací nastavené zpoždění generování události evMouseRep při držení tlačítka myši na jednom místě po delší dobu.

**Parametry:**

Metoda nemá žádné parametry.

**Návratové hodnoty:**

Metoda vrací délku časového intervalu v milisekundách.

**Poznámky:**

#### 4.1.1.13. Metoda TTEMouseDriver.SetRepeatRate

Metoda **SetRepeatRate** slouží k nastavení periody generování události evMouseRep při držení tlačítka myši na jednom místě po delší dobu.

**Parametry:**

AValue                      Délka časového intervalu v milisekundách.

**Návratové hodnoty:**

Metoda nevrací žádnou hodnotu.

**Poznámky:**

#### 4.1.1.14. Metoda TTEMouseDriver.GetRepeatRate

Metoda **GetRepeatRate** vrací nastavenou periodu generování události evMouseRep při držení tlačítka myši na jednom místě po delší dobu.

**Parametry:**

Metoda nemá žádné parametry.

**Návratové hodnoty:**

Metoda vrací délku časového intervalu v milisekundách.

**Poznámky:**

### 4.1.2. Třída TTEKeybDriver

Třída **TTEKeybDriver** implementuje ovladač klávesnice emulátoru terminálu. Tato třída vychází z báze třídy pro implementaci ovladačů klávesnic **TKeyboardDriver** (viz. dokumentace ke knihovně IoDrv)

```
PTEKeybDriver = ^TTEKeybDriver;  
TTEKeybDriver = object( TKeyboardDriver )  
public  
    procedure GetEvent( var AEvent: TEvent ); virtual;  
end;
```

#### 4.1.2.1. Metoda TTEKeybDriver.GetEvent

Metoda **GetEvent** předá nejstarší událost a odstraní ji z fronty událostí ovladače klávesnice.

```
procedure GetEvent( var AEvent: TEvent ); virtual;
```

#### Parametry:

AEvent	Po provedení metody je do parametru AEvent uložena událost typu evKeyDown a jsou vyplněny položky KeyCode, CharCode. Položka VirtKey je nastavena na 0. V případě, že nebyla stisknuta žádná klávesa je vyplněna pouze položka Code hodnotou evNothing.
--------	---

#### Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

#### Poznámky:

Metoda **GetEvent** předefinovává metodu **GetEvent** báze třídy **TKeyboardDriver** (viz. dokumentace ke knihovně IoDrv).

Všechny kódy kláves, které ovladač **TT10KeybDriver** může vrátit jsou popsány v kapitole 3.1.1.1.

### 4.1.3. Třída TTEDisplayDriver

Třída **TTEDispDriver** implementuje ovladač displeje emulátoru terminálu. Tato třída vychází z báze třídy pro implementaci ovladačů displejů **TDispDriver** (viz. dokumentace ke knihovně IoDrv)

```
PTEDispDriver = ^TTEDispDriver;  
TTEDispDriver = object( TDisplayDriver )  
public  
    constructor Init( AWidth, AHeight: Integer );
```

```
function Initialize: Boolean; virtual;  
procedure Finalize; virtbual;  
end;
```

#### 4.1.3.1. Konstruktor TTEDispDriver.Init

Konstruktor **Init** provádí inicializaci instance třídy.

```
constructor Init( AWidth, AHeight: Integer );
```

##### Parametry:

AWidth	Počet pixelů na šířku displeje.
AHeight	Počet pixelů na výšku displeje.

##### Návratové hodnoty:

Konstruktor nevrací žádnou hodnotu.

##### Poznámky:

Hodnoty AWidth a AHeight mohou být naprosto libovolné.

#### 4.1.3.2. Metoda TTEDispDriver.Initialize

Metoda Initialize provádí test funkčnosti emulátoru terminálu a inicializaci jeho displeje, tj. nastavení správného rozlišení.

```
function Initialize: Boolean; virtual;
```

##### Parametry:

Metoda nemá žádné parametry.

##### Návratové hodnoty:

V případě úspěšné inicializace metoda vrací hodnotu True. V opačném případě vrací hodnotu False.

##### Poznámky:

Pokud v MS Windows, na kterém běží tato aplikace není nainstalován ovladač virtuálního zařízení (VDD) termemu.dll, vrátí tato metoda vždy hodnotu False.

#### 4.1.3.3. Metoda TTEDispDriver.Finalize

Metoda Finalize provádí ukončení práce s displejem emulátoru terminálu.

```
procedure Finalize; virtual;
```

##### Parametry:

Metoda nemá žádné parametry.

**Návratové hodnoty:**

Metoda nevrací žádnou hodnotu.

**Poznámky:**