

# T33MDrv

OVLADAČE TERMINÁLU TOUCH33  
MONO PRO VIZUALIZAČNÍ KNIHOVNY  
PRO JEDNOTKU KIT

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 T33MDrv	5
3.2. Terminál Touch33 Mono	5
3.2.1. Ovladač displeje	5
3.2.2. Použití ovladačů terminálu Touch33 Mono	5
3.3. Nastavení terminálu Touch33 Mono	6
4.Reference	7
4.1. Třídy	7
4.1.1. Třída TT33MonoDispDriver	7
4.1.1.1. Položka TT33MonoDispDriver.ioBase	7
4.1.1.2. Položka TT33MonoDispDriver.VideoBase	7
4.1.1.3. Položka TT33MonoDispDriver.TempState	7
4.1.1.4. Položka TT33MonoDispDriver.TempTimer	8
4.1.1.5. Položka TT33MonoDispDriver.ContrastOfs	8
4.1.1.6. Položka TT33MonoDispDriver.Contrast	8
4.1.1.7. Položka TT33MonoDispDriver.Brightness	8
4.1.1.8. Konstruktor TT33MonoDispDriver.Init	8
4.1.1.9. Destruktor TT33MonoDispDriver.Done	9
4.1.1.10. Metoda TT33MonoDispDriver.Initialize	9
4.1.1.11. Metoda TT33MonoDispDriver.Finalize	9
4.1.1.12. Metoda TT33MonoDispDriver.Tick	10
4.1.1.13. Metoda TT33MonoDispDriver.GetContrast	10
4.1.1.14. Metoda TT33MonoDispDriver.SetContrast	11
4.1.1.15. Metoda TT33MonoDispDriver.GetBrightness	11
4.1.1.16. Metoda TT33MonoDispDriver.SetBrightness	12

---

## 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 T33MDrv, která je součástí 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 T33MDrv

---

Knihovna T33MDrv obsahuje ovladače displeje terminálu Touch33 Mono fy SofCon pro použití s vizualizačními knihovnami pro jednotku KIT.

### 3.2. Terminál Touch33 Mono

---

Terminál Touch33 Mono je terminál s monochromatickým grafickým displejem s rozlišením 640x480 pixelů s 1-bitovou hloubkou barev. Terminál je dále vybaven dotykovým panelem. Ovladač dotykového panelu je implementován v knihovně TPDrv.

#### 3.2.1. Ovladač displeje

Knihovna T33MDrv obsahuje třídu **TT33MonoDispDriver**, která implementuje ovladač grafického displeje terminálu. Ovladač vychází ze třídy **TDisplayDriver** implementované v knihovně IoDrv.

Ovladač **TT33MonoDispDriver** umožňuje řízení kontrastu a jasu displeje. Ovladač měří v nastavených periodách teplotu displeje a podle naměřených vzorků automaticky upravuje kontrast, tak aby nedocházelo k jeho kolísání.

#### 3.2.2. Použití ovladačů terminálu Touch33 Mono

Následující příklad ukazuje, jak vytvořit základní komponentu aplikace **TApplication** pro terminál Touch33 Mono.

```
const
  ioTouch33 = $2300; { Bázová adresa terminálu v I/O prostoru }
  mmTouch33 = $C000; { Segment bázov0 adresy videopameti }
var
  App : PApplication;

App :=
  New( PApplication, Init (
    New( PInputDriver, Init(
      nil,
      New( PTouchPanelDriver, Init( ioTouch33 )
    )),
    New( PT33MonoDispDriver, Init( ioTouch33, mmTouch33 ) ),
    @g_AppSettings
  ));
```

Pozn.:

- Ovladač TTouchPanelDriver je definován v knihovně TPDrv.
- Proměnná g\_AppSettings obsahuje nastavení ovladače displeje a klávesnice terminálu. Tato proměnná musí být umístěna v zálohované paměti RAM.

### 3.3. Nastavení terminálu Touch33 Mono



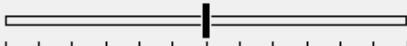
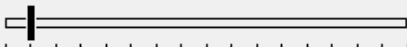
Terminál Touch33 Mono jako celek umožňuje nastavovat následující parametry.

- Úroveň kontrastu displeje
- Úroveň jasu displeje (úroveň podsvícení)
- Čas pro automatické snížení jasu při nečinnosti uživatele
- Úroveň jasu při nečinnosti uživatele
- Kalibrační konstanty dotykového panelu
- Parametry řadiče dotykového panelu

Všechny tyto parametry lze nastavit pomocí k tomu určených metod ovladače displeje a dotykového panelu.

Pro jednoduchou nastavení terminálu Touch33 byla vytvořena knihovna T33Setup, obsahující tři speciální stránky, na kterých je možné tyto parametry terminálu modifikovat viz. následující obrázek:

Stránka pro nastavení parametrů displeje:

Nastavení displeje terminálu Touch 33			
Kontrast displeje:			
Jas výbojky:			1
Automaticke vypnutí:			60
Jas ve vypnutém stavu:			1
Verze:	N/A		
Teplota:	N/A		
Zpět	Panel	Kalibrace	

## 4. Reference

---

### 4.1. Třídy

---

#### 4.1.1. Třída TT33MonoDispDriver

Třída **TT33MonoDispDriver** implementuje ovladač displeje terminálu Touch33 Mono. Tato třída vychází z básové třídy pro implementaci ovladačů displejů **TDisplayDriver** (viz. dokumentace ke knihovně IoDrv)

```

PT33MonoDispDriver = ^TT33MonoDispDriver;
TT33MonoDispDriver = object( TDisplayDriver )
public
  IoBase      : Word;
  VideoBase   : Word;
  TempState   : Byte;
  TempTimer   : TTimerEx;
  ContrastOfs : Integer;
  Contrast    : Integer;
  Brightness  : Byte;

  constructor Init( AIoBase: Word; AVideoBase: Word );
  destructor Done; virtual;

  function Initialize: Boolean; virtual;
  procedure Finalize; virtual;
  procedure Tick; virtual;

  function GetContrast: Integer; virtual;
  procedure SetContrast( AValue: Integer ); virtual;
  function GetBrightness: Integer; virtual;
  procedure SetBrightness( AValue: Integer ); virtual;
end;

```

##### 4.1.1.1. Položka TT33MonoDispDriver.IoBase

Položka **IoBase** obsahuje básovou adresu řadiče terminálu v I/O prostoru. Položka je inicializovaná konstruktorem a je určena pouze pro čtení.

```
IoBase      : Word;
```

##### 4.1.1.2. Položka TT33MonoDispDriver.VideoBase

Položka **VideoBase** obsahuje segment adresy videopaměti řadiče displeje v paměťovém adresovém prostoru. Položka je inicializovaná konstruktorem a je určena pouze pro čtení.

```
VideoBase : Word;
```

##### 4.1.1.3. Položka TT33MonoDispDriver.TempState

Položka **TempState** obsahuje stav automatu pro měření teplotu displeje. Položka je

interní a je určena pouze pro čtení.

```
TempState : Byte;
```

#### 4.1.1.4. Položka TT33MonoDispDriver.TempTimer

Položka **TempTimer** obsahuje strukturu časovače určenou pro účely měření teploty displeje. Položka je interní a je určena pouze pro čtení.

```
TempTimer : TTimerEx;
```

#### 4.1.1.5. Položka TT33MonoDispDriver.ContrastOfs

Položka **ContrastOfs** obsahuje optimální hodnotu kontrastu displeje, vypočítanou podle změřené teploty displeje. Hodnota položky je aktualizována v pravidelných intervalech automaticky řadiče displeje a je určena pouze pro čtení.

```
ContrastOfs : Integer;
```

#### 4.1.1.6. Položka TT33MonoDispDriver.Contrast

Položka **Contrast** obsahuje aktuální hodnotu jasu displeje. Položka je interní. Pro čtení a nastavení kontrastu displeje slouží metody **GetContrast** a **SetContrast** (viz. kapitoly 4.1.1.13 a 4.1.1.14).

```
Contrast : Integer;
```

#### 4.1.1.7. Položka TT33MonoDispDriver.Brightness

Položka **Brightness** obsahuje aktuální hodnotu jasu displeje. Položka je interní. Pro čtení a nastavení jasu displeje slouží metody **GetBrightness** a **SetBrightness** (viz. kapitoly 4.1.1.15 a 4.1.1.16).

```
Brightness : Byte;
```

#### 4.1.1.8. Konstruktor TT33MonoDispDriver.Init

Konstruktor **Init** provádí inicializaci instance třídy.

```
constructor Init( AIOBase: Word );
```

##### **Parametry:**

AIOBase                    Bázová adresa řadiče terminálu v I/O prostoru.

##### **Návratové hodnoty:**

Konstruktor nevrací žádnou hodnotu.

##### **Poznámky:**

Konstruktor vytvoří instanci textového kurzoru a kreslicího povrchu (viz. dokumentace ke knihovně IoDrv). Nastaví implicitní jas a kontrast displeje.



#### 4.1.1.9. Destruktor TT33MonoDispDriver.Done

Destruktor **Done** provádí uvolnění prostředků alokovaných konstruktorem.

```
destructor Done; virtual;
```

##### **Parametry:**

Destruktor nemá žádné parametry.

##### **Návratové hodnoty:**

Destruktor nevrací žádnou hodnotu.

##### **Poznámky:**

Destruktor uvolní vytvořené instance textového kurzoru a kreslicího povrchu.

#### 4.1.1.10. Metoda TT33MonoDispDriver.Initialize

Metoda **Initialize** provádí inicializaci hardware řadiče displeje terminálu Touch33 Mono.

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

##### **Parametry:**

Metoda nemá žádné parametry.

##### **Návratové hodnoty:**

Metoda vrací hodnotu True v případě úspěšné inicializace řadiče displeje.

##### **Poznámky:**

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

Metoda provede nastavení řadiče displeje do správného pracovního režimu a nastaví implicitní hodnoty jasu a kontrastu.

#### 4.1.1.11. Metoda TT33MonoDispDriver.Finalize

Metoda **Finalize** provádí deinicializaci hardware řadiče displeje terminálu Touch33 Mono.

```
procedure Finalize; virtual;
```

##### **Parametry:**

Metoda nemá žádné parametry.

**Návratové hodnoty:**

Metoda nevrací žádnou hodnotu

**Poznámky:**

Metoda **Finalize** předefinovává metodu **Finalize** báze třídy **TDisplayDriver** (viz. dokumentace ke knihovně IoDrv).

Metoda provede zhasnutí výbojky terminálu.

#### 4.1.1.12. Metoda TT33MonoDispDriver.Tick

Metoda **Tick** provádí jeden krok automatu ovladače displeje.

```
procedure Tick; virtual;
```

**Parametry:**

Metoda nemá žádné parametry.

**Návratové hodnoty:**

Metoda nevrací žádnou hodnotu.

**Poznámky:**

Metoda **Tick** volá metodu **Tick** svého předka, tj. třídy **TDisplayDriver** (viz. dokumentace ke knihovně IoDrv). a navíc přidává automat pro pravidelné měření teploty displeje pro účely korekce kontrastu.

#### 4.1.1.13. Metoda TT33MonoDispDriver.GetContrast

Metoda **GetContrast** vrací aktuální hodnotu kontrastu displeje.

```
function GetContrast: Integer; virtual;
```

**Parametry:**

Metoda nemá žádné parametry.

**Návratové hodnoty:**

Metoda vrací aktuální hodnotu kontrastu displeje. Hodnota může být v rozsahu -255 až 255. Hodnota -255 odpovídá minimálnímu kontrastu. Hodnota 255 odpovídá maximálnímu kontrastu. Hodnota 0 odpovídá optimálnímu kontrastu zjištěného měřením teploty displeje.

**Poznámky:**

Metoda **GetContrast** předefinovává metodu **GetContrast** báze třídy **TDisplayDriver** (viz. dokumentace ke knihovně IoDrv).

#### 4.1.1.14. Metoda TT33MonoDispDriver.SetContrast

Metoda **SetContrast** slouží pro nastavení aktuální hodnoty kontrastu displeje.

```
procedure SetContrast( AValue: Integer ); virtual;
```

##### Parametry:

AValue	Požadovaná hodnota kontrastu. Hodnota může být v rozsahu -255 až 255. Hodnota -255 odpovídá minimálnímu kontrastu. Hodnota 255 odpovídá maximálnímu kontrastu. Hodnota 0 odpovídá optimálnímu kontrastu zjištěného měřením teploty displeje.
--------	--

##### Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

##### Poznámky:

Metoda **SetContrast** předefinovává metodu **SetContrast** báze třídy **TDisplayDriver** (viz. dokumentace ke knihovně IoDrv).

#### 4.1.1.15. Metoda TT33MonoDispDriver.GetBrightness

Metoda **GetBrightness** vrací aktuální hodnotu jasu (úroveň podsvícení) displeje.

```
function GetBrightness: Integer; virtual;
```

##### Parametry:

Metoda nemá žádné parametry.

##### Návratové hodnoty:

Metoda vrací aktuální hodnotu jasu displeje. Vracená hodnota je v rozsahu 0 až 16. Hodnota 0 odpovídá vypnuté výbojce. Hodnota 16 odpovídá maximálnímu jasu displeje.

##### Poznámky:

Metoda **GetBrightness** předefinovává metodu **GetBrightness** báze třídy **TDisplayDriver** (viz. dokumentace ke knihovně IoDrv).

#### 4.1.1.16. Metoda TT33MonoDispDriver.SetBrightness

Metoda **SetBrightness** slouží pro nastavení aktuální hodnoty jasu (úrovně podsvícení) displeje.

```
procedure SetBrightness( AValue: Integer ); virtual;
```

##### **Parametry:**

AValue	Požadovaná hodnota jasu. Hodnota se může pohybovat v rozsahu 0 až 16. Hodnota je 0 odpovídá vypnuté výbojce. Hodnota 16 odpovídá maximálnímu jasu.
--------	--

##### **Návratové hodnoty:**

Metoda nevrací žádnou hodnotu.

##### **Poznámky:**

Metoda **SetBrightness** předefinovává metodu **SetBrightness** báze třídy **TDisplayDriver** (viz. dokumentace ke knihovně IoDrv).