

T10Drv

OVLADAČE TERMINÁLU TERM10 PRO VIZUALIZAČNÍ KNIHOVNY PRO JEDNOTKU KIT

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: 21.01.2004

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 T10Drv | 5 |
| 3.2. Terminál Term10 | 5 |
| 3.2.1. Ovladač klávesnice | 5 |
| 3.2.1.1. Podporované kódy kláves | 5 |
| 3.2.2. Ovladač displeje | 5 |
| 3.2.3. Použití ovladačů terminálu Term10 | 6 |
| 3.3. Nastavení terminálu Term10 | 6 |
| 4.Reference | 7 |
| 4.1. Třídy | 7 |
| 4.1.1. Třída TT10KeyDriver | 7 |
| 4.1.1.1. Položka TT10KeybDriver.IoBase | 7 |
| 4.1.1.2. Položka TT10KeybDriver.BeepState | 7 |
| 4.1.1.3. Konstruktor TT10KeybDriver.Init | 8 |
| 4.1.1.4. Metoda TT10KeybDriver.Initialize | 8 |
| 4.1.1.5. Metoda TT10KeybDriver.Tick | 8 |
| 4.1.1.6. Metoda TT10KeybDriver.GetEvent | 9 |
| 4.1.1.7. Metoda TT10KeybDriver.Flush | 9 |
| 4.1.1.8. Metoda TT10KeybDriver.Beep | 10 |
| 4.1.1.9. Metoda TT10KeybDriver.SetKeyBeep | 10 |
| 4.1.1.10. Metoda TT10KeybDriver.GetKeyBeep | 10 |
| 4.1.2. Třída TT10DispDriver | 11 |
| 4.1.2.1. Položka TT10DispDriver.IoBase | 11 |
| 4.1.2.2. Položka TT10DispDriver.Cache | 12 |
| 4.1.2.3. Položka TT10DispDriver.CacheSize | 12 |
| 4.1.2.4. Položka TT10DispDriver.LedState | 12 |
| 4.1.2.5. Položka TT10DispDriver.LcdCtrl | 12 |
| 4.1.2.6. Položka TT10DispDriver.Brightness | 12 |
| 4.1.2.7. Položka TT10DispDriver.Contrast | 12 |
| 4.1.2.8. Konstruktor TT10DispDriver.Init | 13 |
| 4.1.2.9. Destruktor TT10DispDriver.Done | 13 |
| 4.1.2.10. Metoda TT10DispDriver.Initialize | 13 |
| 4.1.2.11. Metoda TT10DispDriver.Finalize | 14 |
| 4.1.2.12. Metoda TT10DispDriver.Tick | 14 |
| 4.1.2.13. Metoda TT10DispDriver.Refresh | 15 |
| 4.1.2.14. Metoda TT10DispDriver.GetContrast | 15 |
| 4.1.2.15. Metoda TT10DispDriver.SetContrast | 15 |
| 4.1.2.16. Metoda TT10DispDriver.GetBrightness | 16 |
| 4.1.2.17. Metoda TT10DispDriver.SetBrightness | 16 |
| 4.1.2.18. Metoda TT10DispDriver.SetLedState | 17 |

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 T10Drv, 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 a Controls.
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 T10Drv

Knihovna T10Drv obsahuje ovladače klávesnice a displeje terminálu Term10 fy SofCon pro použití s vizualizačními knihovnami pro jednotku KIT.

3.2. Terminál Term10

Terminál Touch11 Mono je terminál s monochromatickým grafickým displejem s rozlišením 240x128 pixelů s 1-bitovou hloubkou barev. Terminál je dále vybaven membránovou klávesnicí.

3.2.1. Ovladač klávesnice

Knihovna T10Drv obsahuje třídu **TT10KeyDriver**, která implementuje ovladač alfanumerické klávesnice terminálu Term10. Ovladač vychází z abstraktní třídy **TKeyboardDriver** implementované v knihovně IoDrv.

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

Ovladač umožňuje zároveň řízení zvukové signalizace terminálu pomocí metod **Beep** a **SetKeyBeep**.

3.2.1.1. Podporované kódy kláves

Ovladač klávesnice **TT10KeybDriver** podporuje kódy speciálních kláves uvedené v následující tabulce. Jednotlivé konstanty s prefixem kb_ jsou popsány v dokumentaci ke knihovně IoDrv.

| | | |
|-------|--------------|-------------|
| kbF1 | kbEnter | kbInsert |
| kbF2 | kbShiftEnter | kbDelete |
| kbF3 | kbLeft | kbBackSpace |
| kbF4 | kbRight | kbTab |
| kbF5 | kbUp | kbEsc |
| kbF6 | kbDown | kbStart |
| kbF7 | kbShiftLeft | kbStop |
| kbF8 | kbShiftRight | kbAlt |
| kbF9 | kbShiftUp | |
| kbF10 | kbShiftDown | |

3.2.2. Ovladač displeje

Knihovna T10Drv obsahuje třídu **TT10DispDriver**, která implementuje ovladač grafického displeje terminálu. Ovladač vychází z abstraktní třídy **TCachedDisplayDriver** implementované v knihovně IoDrv.

Ovladač **TT10DispDriver** umožňuje řízení kontrast a jasu displeje a zároveň poskytuje metodu **SetLedState** pro ovládání indikačních led-diod umístěných na předním panelu terminálu.

3.2.3. Použití ovladačů terminálu Term10

Následující příklad ukazuje, jak vytvořit základní komponentu aplikace **TApplication** pro terminál Term10.

```
const
  ioTerm10 = $2300; { Bázová adresa terminálu v I/O prostoru }

var
  App : PApplication;

App :=
  New( PApplication, Init (
    New( PInputDriver, Init(
      New( PT10KeybDriver, Init( ioTerm10 )),
      nil
    )),
    New( PT10DispDriver, Init( ioTerm10 )),
    @g_AppSettings
  ));
```

Proměnná `g_AppSettings` obsahuje nastavení ovladače displeje a klávesnice terminálu. Tato proměnná musí být umístěna v zálohované paměti RAM.

3.3. Nastavení terminálu Term10

Terminál Term10 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
- Zvuková signalizace při stisku klávesy

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

Pro jednoduchou nastavení terminálu Term10 byla vytvořena knihovna T10Setup, obsahující speciální stránku, na které je možné tyto parametry terminálu modifikovat viz. následující obrázek:



4. Reference

4.1. Třídy

4.1.1. Třída TT10KeyDriver

Třída **TT10KeybDriver** implementuje ovladač klávesnice terminálu Term10. Tato třída vychází z básové třídy pro implementaci ovladačů klávesnic **TKeyboardDriver** (viz. dokumentace ke knihovně IoDrv)

```

PT10KeybDriver = ^TT10KeybDriver;
TT10KeybDriver = object( TKeyboardDriver )
public
  IoBase      : Word;
  BeepState   : Byte;

  constructor Init( AIoBase: Word );

  function Initialize: Boolean; virtual;
  procedure Tick; virtual;
  procedure GetEvent( var Event: TEvent ); virtual;

  procedure Flush;
  procedure Beep( AOn: Boolean );

  procedure SetKeyBeep( AValue: Boolean ); virtual;
  function GetKeyBeep: Boolean; virtual;
end;

```

4.1.1.1. Položka TT10KeybDriver.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 TT10KeybDriver.BeepState

Položka **BeepState** obsahuje kopii registru řadiče terminálu řídícího zvukovou signalizaci. Položka je interní a je určena pouze pro čtení. Její modifikaci provádějí

metody **Beep** (viz. kapitola 4.1.1.8) a **SetKeyBeep** (viz. kapitola 4.1.1.9).

```
BeepState : Byte;
```

4.1.1.3. Konstruktor TT10KeybDriver.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:

4.1.1.4. Metoda TT10KeybDriver.Initialize

Metoda **Initialize** provádí inicializaci hardware řadiče klávesnice terminálu Term10.

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

Parametry:

Metoda nemá žádné parametry.

Návratové hodnoty:

Metoda vrací hodnotu True v případě úspěšné inicializace řadiče klávesnice.

Poznámky:

Metoda **Initialize** předefinovává metodu **Initialize** bázové třídy **TKeyboardDriver** (viz. dokumentace ke knihovně IoDrv).

Metoda vyprázdní buffer klávesnice a nastaví zapne zvukové signalizaci při stisku klávesy.

4.1.1.5. Metoda TT10KeybDriver.Tick

Metoda **Tick** provádí krok automatu klávesnice implementovaného ovladačem klávesnice.

```
procedure Tick; virtual;
```


Parametry:

Metoda nemá žádné parametry.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

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

Metoda provádí periodický přepis položky BeepState na příslušný port řadiče klávesnice.

4.1.1.6. Metoda TT10KeybDriver.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 3.2.1.1.

4.1.1.7. Metoda TT10KeybDriver.Flush

Metoda **Flush** provádí vyprázdnění bufferu řadiče klávesnice.

```
procedure Flush;
```

Parametry:

Metoda nemá žádné parametry.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

4.1.1.8. Metoda TT10KeybDriver.Beep

Metoda **Beep** zapíná resp. vypíná zvukovou signalizaci terminálu.

```
procedure Beep( AOn: Boolean );
```

Parametry:

AOn Pokud je parametr AOn nastaven na hodnotu True, zvuková signalizace bude zapnuta. V opačném případě bude vypnuta.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

4.1.1.9. Metoda TT10KeybDriver.SetKeyBeep

Metoda **SetKeyBeep** zapíná resp. vypíná zvukovou signalizaci při stisku klávesy terminálu.

```
procedure SetKeyBeep( AValue: Boolean ); virtual;
```

Parametry:

AValue Pokud je parametr AValue nastaven na hodnotu True, zvuková signalizace bude zapnuta. V opačném případě bude vypnuta.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

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

4.1.1.10. Metoda TT10KeybDriver.GetKeyBeep

Metoda **GetKeyBeep** vrací aktuální stav zvukové signalizace při stisku klávesy

terminálu.

```
function GetKeyBeep: Boolean; virtual;
```

Parametry:

Metoda nemá žádné parametry.

Návratové hodnoty:

Metoda vrací hodnotu True, pokud je zvuková signalizace při stisku klávesy terminálu zapnuta. V opačném případě vrací hodnotu False.

Poznámky:

Metoda **GetKeyBeep** předefinovává metodu **GetKeyBeep** bázové třídy **TKeyboardDriver** (viz. dokumentace ke knihovně IoDrv).

4.1.2. Třída TT10DispDriver

Třída **TT10DispDriver** implementuje ovladač displeje terminálu Term10. Tato třída vychází z bázové třídy pro implementaci ovladačů displejů s vyrovnávací pamětí **TCachedDisplayDriver** (viz. dokumentace ke knihovně IoDrv)

```
TT10DispDriver = ^TT10DispDriver;  
TT10DispDriver = object( TCachedDisplayDriver )  
public  
    IoBase      : Word;  
    Cache       : Pointer;  
    CacheSize   : Word;  
    LedState    : Byte;  
    LcdCtrl     : Byte;  
    Brightness  : Integer;  
    Contrast    : Integer;  
  
    constructor Init( AIoBase: Word );  
    destructor Done; virtual;  
  
    function Initialize: Boolean; virtual;  
    procedure Finalize; virtual;  
    procedure Tick; virtual;  
    procedure Refresh( const R: TRect ); virtual;  
  
    function GetContrast: Integer; virtual;  
    procedure SetContrast( AValue: Integer ); virtual;  
    function GetBrightness: Integer; virtual;  
    procedure SetBrightness( AValue: Integer ); virtual;  
    procedure SetLedState( AState: Byte );  
end;
```

4.1.2.1. Položka TT10DispDriver.IoBase

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

```
IoBase      : Word;
```

4.1.2.2. Položka TT10DispDriver.Cache

Položka **Cache** obsahuje ukazatel na vyrovnávací paměť pro videodata zapisovaná do řadiče displeje alokovanou na hromadě. Tato položka je inicializovaná konstruktorem instance a je určena pouze pro čtení.

```
Cache       : Pointer;
```

4.1.2.3. Položka TT10DispDriver.CacheSize

Položka **CacheSize** obsahuje velikost vyrovnávací paměti pro videodata zapisovaná do řadiče displeje, tj. velikost bloku, na který se odkazuje položka Cache (viz. kapitola 4.1.2.2). Položka je inicializovaná konstruktorem instance a je určena pouze pro čtení.

```
CacheSize   : Word;
```

4.1.2.4. Položka TT10DispDriver.LedState

Položka **LedState** obsahuje aktuální stav led-diod terminálu Term10. Položka je určena pouze pro čtení. Pro její nastavení slouží metoda **SetLedState** (viz. kapitola 4.1.2.4).

```
LedState    : Byte;
```

Na terminálu Term10 je k dispozici celkem osm indikačních led-diod. Každá led-dioda má v této položce přiřazen jeden bit. Pokud je nastaven, led-dioda svítí. Pokud je vynulován led-dioda je zhasnutá.

4.1.2.5. Položka TT10DispDriver.LcdCtrl

Položka **LcdCtrl** obsahuje kopii registru řadiče displeje určeného pro řízení podsvícení a kontrastu displeje. Položka je interní a je určena pouze pro čtení.

```
LcdCtrl     : Byte;
```

4.1.2.6. Položka TT10DispDriver.Brightness

Položka **Brightness** obsahuje aktuální hodnotu jasu displeje. Položka je interní. Pro čtení a nastavení jasu displeje slouží metody **GetBrightness** a **SetBrightness** (viz. kapitoly 4.1.2.16 a 4.1.2.17).

```
Brightness  : Integer;
```

4.1.2.7. Položka TT10DispDriver.Contrast

Položka **Contrast** obsahuje aktuální hodnotu jasu displeje. Položka je interní. Pro čtení a nastavení kontrastu displeje slouží metody **GetContrast** a **SetContrast** (viz. kapitoly 4.1.2.14 a 4.1.2.15).

```
Contrast    : Integer;
```

4.1.2.8. Konstruktor TT10DispDriver.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 alokuje vyrovnávací paměť pro videodata zapisovaná do řadiče displeje. Vytvoří instanci textového kurzoru a kreslicího povrchu (viz. dokumentace ke knihovně IoDrv). Nastaví implicitní jas a kontrast displeje.

4.1.2.9. Destruktor TT10DispDriver.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 z paměti vyrovnávací paměť pro videodata a vytvořené instance textového kurzoru a kreslicího povrchu.

4.1.2.10. Metoda TT10DispDriver.Initialize

Metoda **Initialize** provádí inicializaci hardware řadiče displeje terminálu Term10.

```
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 reset řadiče displeje a nastaví u něj implicitní hodnoty jasu a kontrastu. Do paměti řadiče displeje přepíše aktuální stav vyrovnávací paměti (nuly).

4.1.2.11. Metoda TT10DispDriver.Finalize

Metoda **Finalize** provádí deinicializaci hardware řadiče displeje terminálu Term10.

```
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 a zhasnutí všech indikačních led-diod.

4.1.2.12. Metoda TT10DispDriver.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 **TCachedDisplayDriver** (viz. dokumentace ke knihovně IoDrv) a navíc přidává kód pro periodický přepis položek LedState a LcdCtrl do registrů řadiče displeje.

4.1.2.13. Metoda TT10DispDriver.Refresh

Metoda **Refresh** slouží k překopírování části kopie videopaměti do paměti řadiče displeje.

```
procedure Refresh( const R: TRect ); virtual;
```

Parametry:

R Obdélník ohraničující oblast videopaměti, kterou je potřeba přenést do paměti řadiče displeje.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Metoda **Refresh** předefinovává abstraktní metodu **Refresh** předka, tj. třídy **TCachedDisplayDriver** (viz. dokumentace ke knihovně IoDrv).

4.1.2.14. Metoda TT10DispDriver.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 0 až 15. Hodnota 0 odpovídá minimálnímu kontrastu. Hodnota 15 odpovídá maximálnímu kontrastu.

Poznámky:

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

4.1.2.15. Metoda TT10DispDriver.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 0 až 15. Hodnota 0 odpovídá minimálnímu kontrastu. Hodnota 15 odpovídá maximálnímu kontrastu.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Metoda **SetContrast** předefinováá metodu **SetContrast** bákové třídy **TDisplayDriver** (viz. dokumentace ke knihovně IoDrv).

4.1.2.16. Metoda TT10DispDriver.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 4. Hodnota 0 odpovídá vypnuté výbojce. Hodnota 4 odpovídá maximálnímu jasu displeje.

Poznámky:

Metoda **GetBrightness** předefinováá metodu **GetBrightness** bákové třídy **TDisplayDriver** (viz. dokumentace ke knihovně IoDrv).

4.1.2.17. Metoda TT10DispDriver.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 4. Hodnota je 0 odpovídá vypnuté výbojce. Hodnota 4 odpovídá maximálnímu jasu.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

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

4.1.2.18. Metoda TT10DispDriver.SetLedState

Metoda **SetLedState** slouží nastavení stavu indikačních led-diod terminálu Term10.

```
procedure SetLedState( AState: Byte );
```

Parametry:

| | |
|--------|---|
| AState | Požadovaná stavy indikačních led-diod. Na terminálu Term10 je k dispozici celkem osm indikačních led-diod. Každá led-dioda má v tomto parametru přiřazen právě jeden bit. Pokud je nastaven, led-dioda svítí. Pokud je vynulován led-dioda je zhasnutá. |
|--------|---|

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Aktuální stav led-diod je uložen v položce LedState (viz. kapitola 4.1.2.4).