

T11CDrv a T11TfDrv

OVLADAČE TERMINÁLU TOUCH11
COLOR A COLOR TFT PRO
VIZUALIZAČNÍ KNIHOVNY PRO ŘS 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: 27.09.2005

Datum posledního uložení dokumentu: 27.09.2005

(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 knihoven T11CDrv a T11TfDrv	5
3.2.	Terminály Touch11 Color a Touch11 Color TFT	5
3.2.1.	Ovladač displeje	5
3.2.2.	Použití ovladačů terminálu Touch11 Color a Color TFT	5
3.3.	Nastavení terminálu Touch11 Color a Color TFT	6
4.	Reference	7
4.1.	Konstanty	7
4.1.1.	Konstanty cl_	7
4.2.	Třídy	7
4.2.1.	Třída TT11ColorDispDriver	7
4.2.1.1.	Položka TT11ColorDispDriver.IoBase	8
4.2.1.2.	Položka TT11ColorDispDriver.VideoBase	8
4.2.1.3.	Položka TT11ColorDispDriver.TempState	8
4.2.1.4.	Položka TT11ColorDispDriver.TempTimer	8
4.2.1.5.	Položka TT11ColorDispDriver.ContrastOfs	8
4.2.1.6.	Položka TT11ColorDispDriver.Contrast	9
4.2.1.7.	Položka TT11ColorDispDriver.Brightness	9
4.2.1.8.	Konstruktor TT11ColorDispDriver.Init	9
4.2.1.9.	Destruktor TT11ColorDispDriver.Done	9
4.2.1.10.	Metoda TT11ColorDispDriver.Initialize	9
4.2.1.11.	Metoda TT11ColorDispDriver.Finalize	10
4.2.1.12.	Metoda TT11ColorDispDriver.Tick	10
4.2.1.13.	Metoda TT11ColorDispDriver.GetContrast	11
4.2.1.14.	Metoda TT11ColorDispDriver.SetContrast	11
4.2.1.15.	Metoda TT11ColorDispDriver.GetBrightness	11
4.2.1.16.	Metoda TT11ColorDispDriver.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
1.10	1.00	Wil	27.09.2005	Doplnění o T11TfDrv

1.2. Účel dokumentu

Tento dokument slouží jako popis knihoven T11CDrv a T11TfDrv, které jsou součástí balíku vizualizačních knihoven pro řídicí 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 knihoven T11CDrv a T11TfDrv

Knihovna T11CDrv obsahuje ovladače displeje terminálu Touch11 Color fy SofCon pro použití s vizualizačními knihovnami pro řídicí jednotku KIT.

Knihovna T11TfDrv obsahuje ovladače displeje terminálu Touch11 Color TFT fy SofCon pro použití s vizualizačními knihovnami pro řídicí jednotku KIT.

3.2. Terminály Touch11 Color a Touch11 Color TFT

Terminál Touch11 Color je terminál s barevným grafickým displejem s rozlišením 320x240 pixelů s 8-bitovou hloubkou barev.

Terminál Touch11 Color TFT je terminál s barevným grafickým displejem HITACHI s rozlišením 320x240 pixelů s 8-bitovou hloubkou barev.

Terminály jsou dále vybaveny dotykovými panely. Ovladač dotykového panelu je implementován v knihovně TPDrv.

3.2.1. Ovladač displeje

Knihovna T11CDrv i T11TfDrv obsahuje třídu **TT11ColorDispDriver**, která implementuje ovladač grafického displeje terminálu. Ovladač vychází z abstraktní třídy **TDisplayDriver** implementované v knihovně IoDrv.

Ovladač TT11ColorDispDriver pro **Touch11 Color** 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.

Ovladač TT11ColorDispDriver pro **Touch11 Color TFT** umožňuje řízení jasu displeje. Ovladač neměří teplotu displeje, jelikož případná kolísající teplota nemá vliv na nastavení kontrastu displeje.

3.2.2. Použití ovladačů terminálu Touch11 Color a Color TFT

Následující příklad ukazuje, jak vytvořit základní komponentu aplikace **TApplication** pro terminál Touch11 Color a Touch11 Color TFT.

```
const
  ioTouch11 = $2300; { Bázová adresa terminálu v I/O prostoru }
  mmTouch11 = $C000; { Bázová adresa videopaměti }
var
  App : PApplication;
begin
  App :=
    New( PApplication, Init (
      New( PInputDriver, Init(
        nil,
        New( PTouchPanelDriver, Init( ioTouch11 )
      )),
      New( PT11ColorDispDriver, Init( ioTouch11, mmTouch11 ) ),
      @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 Touch11 Color a Color TFT

Terminál Touch11 Color a Touch11 Color TFT jako celek umožňuje nastavovat následující parametry:

- Úroveň kontrastu displeje (jen pro Touch11 Color, ne pro TFT)
- 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 Touch11 byla vytvořena knihovna T11Setup, obsahující tři speciální stránky, na kterých je možné tyto parametry terminálu modifikovat viz dokument T11Setup.

4. Reference

Poznámka: Nebude-li řečeno jinak, bude popis v této kapitole a podkapitolách platit jak pro Touch11 Color tak i pro Touch11 Color TFT.

4.1. Konstanty

4.1.1. Konstanty cl_

Konstanty cl_ definují prvních šestnáct indexů do standardní palety barev pro terminál. Paleta barev je definována v jednotce StdPal.

Identifikátor	Hodnota	Popis
clBlack	0	Černá
clBlue	1	Modrá
clGreen	2	Zelená
clCyan	3	Modrozelená
clRed	4	Červená
clMagenta	5	Fuchsinová
clBrown	6	Hnědá
clLightGray	7	Světle šedá
clDarkGray	8	Tmavě šedá
clLightBlue	9	Světle modrá
clLightGreen	10	Světle zelená
clLightCyan	11	Světle modrozelená
clLightRed	12	Světle červená
clLightMagent	13	Světle fuchsinová
a		
clYellow	14	Žlutá
clWhite	15	Bílá

4.2. Třídy

4.2.1. Třída TT11ColorDispDriver

Třída **TT11ColorDispDriver** implementuje ovladač displeje terminálu. Tato třída vychází z abstraktní třídy pro implementaci ovladačů displejů **TDisplayDriver** (viz dokumentace ke knihovně IoDrv)

```

PT11ColorDispDriver = ^TT11ColorDispDriver;
TT11ColorDispDriver = object( TDisplayDriver )
public
  IoBase      : Word;
  VideoBase   : Word;

  TempState   : Byte;      {ne pro TFT}
  TempTimer   : TTimerEx; {ne pro TFT}

```

```

ContrastOfs : Integer; {ne pro TFT}
Contrast    : Integer; {ne pro TFT}
Brightness  : Byte;

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

function Initialize: Boolean; virtual;
procedure Finalize; virtual;
procedure Tick; virtual;                                {ne pro TFT}

function GetContrast: Integer; virtual;                {ne pro TFT}
procedure SetContrast( AValue: Integer ); virtual; {ne pro TFT}
function GetBrightness: Integer; virtual;
procedure SetBrightness( AValue: Integer ); virtual;
end;

```

4.2.1.1. Položka TT11ColorDispDriver.IoBase

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

```
IoBase      : Word;
```

4.2.1.2. Položka TT11ColorDispDriver.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;
```

4.2.1.3. Položka TT11ColorDispDriver.TempState

Položka **TempState** obsahuje stav automatu pro měření teploty displeje. Položka je interní a je určena pouze pro čtení a pouze pro Touch11 Color (ne TFT).

```
TempState  : Byte;
```

4.2.1.4. Položka TT11ColorDispDriver.TempTimer

Položka **TempTimer** obsahuje strukturu časovače určenou pro účely měření teploty displeje. Položka je interní a je určena pouze pro čtení a pouze pro Touch11 Color (ne TFT).

```
TempTimer  : TTimerEx;
```

4.2.1.5. Položka TT11ColorDispDriver.ContrastOfs

Položka **ContrastOfs** obsahuje optimální hodnotu kontrastu displeje, vypočítanou podle změřené teploty displeje. Hodnota položky je aktualizována v pravidelných intervalech automatem řadiče displeje a je určena pouze pro čtení a pouze pro Touch11 Color (ne TFT).

```
ContrastOfs : Integer;
```


4.2.1.6. Položka TT11ColorDispDriver.Contrast

Položka **Contrast** obsahuje aktuální hodnotu jasu displeje. Položka je interní. Pro čtení a nastavení kontrastu displeje slouží metody **GetContrast** a **SetContrast** (viz kapitoly 4.2.1.13 a 4.2.1.14). Položka je určena pouze pro Touch11 Color (ne TFT).

```
Contrast : Integer;
```

4.2.1.7. Položka TT11ColorDispDriver.Brightness

Položka **Brightness** obsahuje aktuální hodnotu jasu displeje. Položka je interní. Pro čtení a nastavení jasu displeje slouží metody **GetBrightness** a **SetBrightness** (viz. kapitoly 4.2.1.15 a 4.2.1.16).

```
Brightness : Byte;
```

4.2.1.8. Konstruktor TT11ColorDispDriver.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.

4.2.1.9. Destruktor TT11ColorDispDriver.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.

4.2.1.10. Metoda T11ColorDispDriver.Initialize

Metoda **Initialize** provádí inicializaci hardware řadiče displeje terminálu Touch11 Color.

```
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 u něj standardní barevnou paletu danou jednotkou StdPal (viz. dokumentace k této knihovně). Dále nastaví implicitní hodnoty jasu a kontrastu.

4.2.1.11. Metoda TT11ColorDispDriver.Finalize

Metoda **Finalize** provádí deinicializaci hardware řadiče displeje terminálu Touch11 Color.

```
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.

4.2.1.12. Metoda TT11ColorDispDriver.Tick

Metoda **Tick** provádí jeden krok automatu ovladače displeje Touch11 Color (ne TFT).

```
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.

4.2.1.13. Metoda TT11ColorDispDriver.GetContrast

Metoda **GetContrast** vrací aktuální hodnotu kontrastu displeje Touch11 Color (ne TFT).

```
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).

4.2.1.14. Metoda TT11ColorDispDriver.SetContrast

Metoda **SetContrast** slouží pro nastavení aktuální hodnoty kontrastu displeje Touch11 Color (ne TFT).

```
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).

4.2.1.15. Metoda TT11ColorDispDriver.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** bazové třídy **TDisplayDriver** (viz. dokumentace ke knihovně IoDrv).

4.2.1.16. Metoda TT11ColorDispDriver.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** bazové třídy **TDisplayDriver** (viz. dokumentace ke knihovně IoDrv).