

TPDrv

OVLADAČE DOTYKOVÉHO PANELU 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: 21.01.2004

(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	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
3.1. Účel knihovny TPDrv	6
3.2. Ovladač sériové linky dotykového panelu	6
3.3. Ovladač dotykového panelu	6
4.Reference	6
4.1. Typy	6
4.1.1. Struktura TTCDS settings	6
4.1.2. Struktura TTCID	8
4.1.3. Typ TTCRequest	8
4.1.4. Typ TTCRequests	9
4.1.4.1. Struktura TTCEvent	9
4.2. Konstanty	9
4.2.1. Konstanta TRQ_READ_SETTINGS	9
4.2.2. Konstanta TRQ_WRITE_SETTINGS	10
4.3. Třídy	10
4.3.1. Třída TTouchCommDriver	10
4.3.1.1. Položka TTouchCommDriver.ioBase	11
4.3.1.2. Položka TTouchCommDriver.Position	11
4.3.1.3. Položka TTouchCommDriver.Pressed	11
4.3.1.4. Položka TTouchCommDriver.Temp	11
4.3.1.5. Položka TTouchCommDriver.Id	12
4.3.1.6. Položka TTouchCommDriver.Settings	12
4.3.1.7. Položka TTouchCommDriver.Request	12
4.3.1.8. Položka TTouchCommDriver.ResetRoutine	13
4.3.1.9. Konstruktor TTouchCommDriver.Init	14
4.3.1.10. Destruktor TTouchCommDriver.Done	14
4.3.1.11. Metoda TTouchCommDriver.GetEvent	14
4.3.1.12. Metoda TTouchCommDriver.Tick	14
4.3.1.13. Metoda TTouchCommDriver.InterruptProc	15
4.3.1.14. Metoda TTouchCommDriver.ResetController	15
4.3.2. Třída TTouchPanelDriver	16
4.3.2.1. Položka TTouchPanelDriver.ioBase	16
4.3.2.2. Konstruktor TTouchPanelDriver.Init	17
4.3.2.3. Metoda TTouchPanelDriver.Initialize	17
4.3.2.4. Metoda TTouchPanelDriver.Finalize	17
4.3.2.5. Metoda TTouchPanelDriver.Tick	18
4.3.2.6. Metoda TTouchPanelDriver.GetEvent	18
4.3.2.7. Metoda TTouchPanelDriver.SetMode	19
4.3.2.8. Metoda TTouchPanelDriver.Calibrate	19
4.3.2.9. Metoda TTouchPanelDriver.SetPressDelay	20
4.3.2.10. Metoda TTouchPanelDriver.GetPressDelay	21
4.3.2.11. Metoda TTouchPanelDriver.SetReleaseDelay	21

4.3.2.12.	Metoda TTouchPanelDriver.GetReleaseDelay	21
4.3.2.13.	Metoda TTouchPanelDriver.SetDeglitch	22
4.3.2.14.	Metoda TTouchPanelDriver.GetDeglitch	22
4.3.2.15.	Metoda TTouchPanelDriver.SetChargeWS	22
4.3.2.16.	Metoda TTouchPanelDriver.GetChargeWS	23
4.3.2.17.	Metoda TTouchPanelDriver.SetFilterWeight	23
4.3.2.18.	Metoda TTouchPanelDriver.GetFilterWeight	23
4.3.2.19.	Metoda TTouchPanelDriver.SetStabilization	24
4.3.2.20.	Metoda TTouchPanelDriver.GetStabilization	24
4.3.2.21.	Metoda TTouchPanelDriver.SetDbClickDelay	24
4.3.2.22.	Metoda TTouchPanelDriver.GetDbClickDelay	25
4.3.2.23.	Metoda TTouchPanelDriver.SetDbClickArea	25
4.3.2.24.	Metoda TTouchPanelDriver.GetDbClickArea	26
4.3.2.25.	Metoda TTouchPanelDriver.SetRepeatDelay	26
4.3.2.26.	Metoda TTouchPanelDriver.GetRepeatDelay	26
4.3.2.27.	Metoda TTouchPanelDriver.SetRepeatRate	27
4.3.2.28.	Metoda TTouchPanelDriver.GetRepeatRate	27
4.3.2.29.	Metoda TTouchPanelDriver.GetCalibration	27
4.3.2.30.	Metoda TTouchPanelDriver.SetCalibration	28
4.4.	Procedury	28
4.4.1.	Funkce TCDInitialize	28
4.4.2.	Procedura TCDRelease	29
4.4.3.	Procedure TCDInterruptProc	29
4.5.	Globální proměnné	30
4.5.1.	Proměnná TCDriver	30

1. O dokumentu

1.1. Revize dokumentu

Verze dokumentu	Verze SW	Autor	Datum vydání	Popis změn
1.00	1.00	Cr	21.01.2004	Úprava dokumentu dle ISO9000

1.2. Účel dokumentu

Tento dokument slouží jako popis knihovny TPDrv, 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 TPDrv

Knihovna TPDrv implementuje ovladače pro řízení řadiče dotykových panelů fy SofCon.

3.2. Ovladač sériové linky dotykového panelu

Knihovna TPDrv obsahuje třídu **TTouchCommDriver** (viz. kapitola 4.3.1), která implementuje ovladač sériové linky určené pro komunikaci s řadičem dotykového panelu. Tento ovladač umožňuje vyčítat a nastavovat parametry řadiče, implementuje a implementuje frontu událostí, do které jsou vkládány informace o stisku a uvolnění dotykového panelu.

Tento ovladač využívá ovladač dotykového panelu (**TTouchPanelDriver**) a některé ovladače displejů pro měření teploty displeje a následnou korekci kontrastu.

3.3. Ovladač dotykového panelu

Knihovna TPDrv obsahuje třídu **TTouchPanelDriver** (viz. kapitola 4.3.2) implementující ovladač dotykového panelu vycházející ze třídy **TMouseDriver** (viz. dokumentace ke knihovně IoDrv). Ovladač dotykového panelu plně emuluje myš s jedním tlačítkem a dále obsahuje jednoduchou podporu pro kalibraci.

4. Reference

4.1. Typy

4.1.1. Struktura TTCDSSettings

Struktura **TTCDSSettings** obsahuje nastavení řadiče dotykového panelu. Jednotlivé položky této struktury lze pomocí sériové linky vyčíst nebo zapsat do řadiče dotykového panelu (viz. položka Settings třídy **TTouchCommDriver** v kapitole 4.3.1.6).

```
TTCDSSettings = record
  PressDelay      : Byte;
  ReleaseDelay    : Byte;
  Deglitch        : Byte;
  ChargeWS        : Byte;
  Stabilization   : Byte;
  FilterWeight    : Byte;
  Brightness      : Byte;
end;
```

Význam jednotlivých položek je uveden v následující tabulce:

Položka	Rozsah	Popis
PressDelay	1-20	Tento parametr ovlivňuje citlivost na přechodové jevy při stisku obrazovky. Hodnota odpovídá času, po který musí být dotyková obrazovka stisknuta, aby byl tento stisk vyhodnocen. Pokud během této doby dojde k uvolnění a opětovnému stisku začne se čas měřit od začátku.
ReleaseDelay	1-20	Tento parametr ovlivňuje citlivost na přechodové jevy při uvolňování obrazovky a odstraňuje krátké pulsy (uvolnění obrazovky) při pohybu prstu (předmětu), který stlačuje obrazovku. Hodnota odpovídá času, za který se posílá stisk případně uvolnění dotykové obrazovky.
Deglitch	1-20	Parametr, který slouží k detekci a následnému odstranění vzorků, jejichž odměr proběhl při nedostatečně stisknutém panelu. Tímto způsobem se eliminují nestability v zasílaných souřadnicích způsobených nesouvislým přitlakem při pohybu prstu po obrazovce. Zvýšením hodnoty se zvýší nejenom citlivost řadiče na dotyk, ale také pravděpodobnost zasílání nestabilních hodnot. Naopak velké snížení této hodnoty vede při rychlých pohybech k „přerušování“ souvislého pohybu kurzoru (pokud rychlost pohybu překročí stanovenou mez, řadič vygeneruje událost Released). Proto doporučujeme ponechat přednastavenou hodnotu.
ChargeWS	4-16	Počet prodlev (WaitState) o délce 125 μ s vkládaných mezi odpojení budicího napětí ze snímací elektrody a vlastní měření. Tato prodleva je určena k odeznění vzniklých přechodových jevů.
Stabilization	0-255	Tento parametr slouží k omezení pohybu (skákání) kurzoru na obrazovce. Tato situace nastává, jeli stisk obrazovky nerovnoměrný a souřadnice kolísají v malém rozmezí zpravidla ± 1 . Hodnota parametru je počet bodů v ose X a Y, které určují oblast zanedbání změny souřadnic od posledního stisku.
FilterWeight	1-32	Parametr udávající váhu filtru, který můžeme popsat následujícím způsobem: $Y_n = \left(1 - \frac{K}{64}\right)U_n + \frac{K}{64}Y_{n-1}$ <i>U_n</i> – změřená hodnota <i>K</i> – váha filtru <i>Y_{n-1}</i> – předchozí hodnota <i>Y_n</i> – nová hodnota
Brightness	0-255	Jas displeje (úroveň podsvícení)

4.1.2. Struktura TTCDId

Struktura **TTCDId** obsahuje identifikaci desky řadiče dotykového panelu. Tuto strukturu lze vyčíst pomocí sériové linky z řadiče dotykového panelu (viz. položka Id třídy **TTouchCommDriver** v kapitole 4.3.1.5).

```
TTCDId = packed record
    BoardVersion : Byte;
    MajorVersion : Byte;
    MinorVersion : Byte;
end;
```

Význam jednotlivých položek je uveden v následující tabulce:

Položka	Popis
BoardVersion	Verze desky
MajorVersion	Verze firmware
MinorVersion	Verze firmware

4.1.3. Typ TTCDRequest

Jednotlivé varianty výčtového typu **TTCDRequest** označují požadavky na čtení nebo zápis parametru z/do řadiče dotykového panelu (viz. položka Request třídy **TTouchCommDriver** v kapitole 4.3.1.7).

```
TTCDRequest = (
    TRQ_READ_ID,
    TRQ_READ_TEMP,
    TRQ_READ_PRESS_DELAY,
    TRQ_READ_RELEASE_DELAY,
    TRQ_READ_DEGLITCH,
    TRQ_READ_CHARGEWS,
    TRQ_READ_STABILIZATION,
    TRQ_READ_FILTERWEIGHT,
    TRQ_READ_BRIGHTNESS,
    TRQ_WRITE_PRESS_DELAY,
    TRQ_WRITE_RELEASE_DELAY,
    TRQ_WRITE_DEGLITCH,
    TRQ_WRITE_CHARGEWS,
    TRQ_WRITE_STABILIZATION,
    TRQ_WRITE_FILTERWEIGHT,
    TRQ_WRITE_BRIGHTNESS
);
```

Význam jednotlivých identifikátorů je uveden v následující tabulce:

Položka	Popis
TRQ_READ_ID	Požadavek na čtení identifikace desky řadiče.
TRQ_READ_TEMP	Požadavek na čtení teploty displeje.
TRQ_READ_PRESS_DELAY	Požadavek na čtení parametru PressDelay
TRQ_READ_RELEASE_DELAY	Požadavek na čtení parametru ReleaseDelay
TRQ_READ_DEGLITCH	Požadavek na čtení parametru Deglitch
TRQ_READ_CHARGEWS	Požadavek na čtení parametru ChargeWS

TRQ_READ_STABILIZATION	Požadavek na čtení parametru Stabilization
TRQ_READ_FILTERWEIGHT	Požadavek na čtení parametru FilterWeight
TRQ_READ_BRIGHTNESS	Požadavek na čtení parametru Brightness
TRQ_WRITE_PRESS_DELAY	Požadavek na zápis parametru PressDelay
TRQ_WRITE_RELEASE_DELAY	Požadavek na zápis parametru ReleaseDelay
TRQ_WRITE_DEGLITCH	Požadavek na zápis parametru Deglitch
TRQ_WRITE_CHARGEWS	Požadavek na zápis parametru ChargeWS
TRQ_WRITE_STABILIZATION	Požadavek na zápis parametru Stabilization
TRQ_WRITE_FILTERWEIGHT	Požadavek na zápis parametru FilterWeight
TRQ_WRITE_BRIGHTNESS	Požadavek na zápis parametru Brightness

4.1.4. Typ TTCDDRequests

Typ **TTCDDRequests** definuje množinu požadavků na čtení nebo zápis parametru z/do řadiče dotykového panelu (viz. položka Request třídy **TTouchCommDriver** v kapitole 4.3.1.7).

```
TTCDDRequests = set of TTCDDRequest;
```

4.1.4.1. Struktura TTCDEvent

Struktura **TTCDEvent** popisuje událost dotykového panelu. Jedná se o jednu položku fronty událostí třídy **TTouchCommDriver**.

```
TTCDEvent = record
  Position : TPoint;
  Pressed  : Boolean;
  Time     : Longint;
end;
```

Význam jednotlivých položek je uveden v následující tabulce:

Položka	Popis
Position	Souřadnice stisku nebo uvolnění
Pressed	Obsahuje True, pokud jde o stisk a False, pokud jde o uvolnění.
Time	Čas příchodu zprávy v tících přerušeni IRQ0 (viz. metoda GetTickCount v jednotce IoDrv).

4.2. Konstanty

4.2.1. Konstanta TRQ_READ_SETTINGS

Konstanta **TRQ_READ_SETTINGS** je množina všech požadavků na vyčtení nastavení z řadiče dotykového panelu.

```
TRQ_READ_SETTINGS = [
  TRQ_READ_ID
```

```
TRQ_READ_PRESS_DELAY,  
TRQ_READ_RELEASE_DELAY,  
TRQ_READ_DEGLITCH,  
TRQ_READ_CHARGEWS,  
TRQ_READ_STABILIZATION,  
TRQ_READ_FILTERWEIGHT,  
TRQ_READ_BRIGHTNESS ];
```

4.2.2. Konstanta TRQ_WRITE_SETTINGS

Konstanta TRQ_WRITE_SETTINGS je množina všech požadavků na zápis nastavení do řadiče dotykového panelu.

```
TRQ_WRITE_SETTINGS = [  
    TRQ_READ_ID,  
    TRQ_WRITE_PRESS_DELAY,  
    TRQ_WRITE_RELEASE_DELAY,  
    TRQ_WRITE_DEGLITCH,  
    TRQ_WRITE_CHARGEWS,  
    TRQ_WRITE_STABILIZATION,  
    TRQ_WRITE_FILTERWEIGHT,  
    TRQ_WRITE_BRIGHTNESS ];
```

4.3. Třídy

4.3.1. Třída TTouchCommDriver

Třída **TTouchCommDriver** implementuje ovladač sériové linky pro komunikaci s řadičem dotykového panelu. Tento ovladač umožňuje:

- Nastavování a vyčítání parametrů řadiče
- Vyčtení verze firmware a identifikace desky řadiče.
- Vyčítání aktuální teploty displeje
- Přístup do fronty události, obsahující události typu stisk a uvolnění dotykového panelu

Ovladač komunikuje s řadičem dotykového panelu pomocí sériové linky. Protože se při komunikaci s řadičem sériové linky nevyužívá žádný přerušovací signál, je potřeba periodicky v min. 10ms intervalech volat rutinu, která zajistí přepis dat z/do řadiče sériové linky do přijímacího resp. vysílacího bufferu (viz. metoda **InterruptProc** v kapitole 4.3.1.13).

Ovladač řadiče sériové linky využívá ovladač dotykového panelu, tj. třída **TTouchPanelDriver** (viz. kapitola 4.3.2) a některé typy ovladačů displejů, pro účely měření teploty displeje a následné úpravy kontrastu (např. **T11MonoDriver**, **T11ColorDriver** apod.). Problém sdílení instance třídy **TTouchCommDriver** řeší funkce **TCDInitialize** a **TCDRelease** (viz. kapitoly 4.4.1 a 4.4.2).

```
PTouchCommDriver = ^TTouchCommDriver;  
TTouchCommDriver = object( TObject )  
public  
    IoBase      : Word;  
    Position    : TPoint;
```

```

Pressed      : Boolean;
Temp        : Integer;
Id          : TTCDDid;
Settings    : TTCDDSettings;
Request     : TTCDDRequests;
ResetRoutine : TTCDDResetRoutine;

constructor Init( AIOBase: Word );

function GetEvent( var AEvent: TTCDEvent ): Boolean;
procedure Tick; virtual;

procedure InterruptProc;
procedure ResetController; virtual;

end;

```

4.3.1.1. Položka TTouchCommDriver.IOBase

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

```
IOBase      : Word;
```

4.3.1.2. Položka TTouchCommDriver.Position

Položka **Position** obsahuje poslední pozici odeslanou řadičem dotykového panelu. Položka je určena pouze pro čtení.

```
Position    : TPoint;
```

4.3.1.3. Položka TTouchCommDriver.Pressed

Položka **Pressed** obsahuje aktuální stav stisku dotykového panelu. Pokud je tato položka nastavena na hodnotu True, dotykový panel je stisknut. Položka je určena pouze pro čtení.

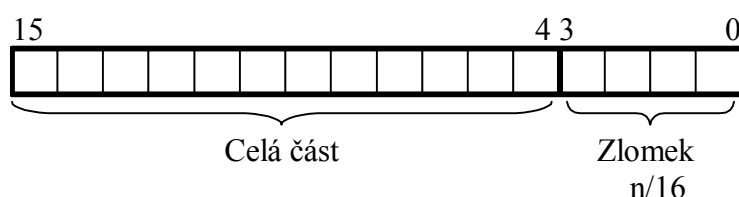
```
Pressed     : Boolean;
```

4.3.1.4. Položka TTouchCommDriver.Temp

Položka **Temp** obsahuje naposledy zjištěnou teplotu displeje. Položka je určena pouze pro čtení.

```
Temp        : Integer;
```

Teplota je uložena v pevné řadové čárce v doplňkovém kódu v šestnáctinách stupně Celsia. Viz. následující obrázek:



Teplotu lze vyčíst nastavením příznaku TRQ_READ_TEMP v položce Request (viz. kapitola 4.3.1.7).

4.3.1.5. Položka TTouchCommDriver.Id

Položka **Id** obsahuje naposledy vyčtenou strukturu **TTCDId** (viz. kapitola 4.1.2), která obsahuje identifikaci desky řadiče dotykového panelu. Položka je určena pouze pro čtení.

```
Id          : TTCDId;
```

Teplotu lze vyčíst nastavením příznaku TRQ_READ_ID v položce Request (viz. kapitola 4.3.1.7).

4.3.1.6. Položka TTouchCommDriver.Settings

Položka **Settings** obsahuje nastavení řadiče dotykového panelu, tj. strukturu **TTCDSettings** (viz. kapitola 4.1.1). Položka je určena jak pro čtení, tak pro zápis.

```
Settings    : TTCDSettings;
```

Jednotlivé položky struktury Settings lze z řadiče dotykového panelu vyčíst nastavením příznaků v položce Request (viz. kapitola 4.3.1.7).

```
TRQ_READ_PRESS_DELAY  
TRQ_READ_RELEASE_DELAY  
TRQ_READ_DEGLITCH  
TRQ_READ_CHARGEWS  
TRQ_READ_STABILIZATION  
TRQ_READ_FILTERWEIGHT
```

Pro vyčtení celé struktury Settings lze použít konstantu TRQ_READ_SETTINGS obsahující všechny uvedené příznaky.

Pro uložení jednotlivých položek struktury Settings do řadiče dotykového panelu je potřeba nastavit v položce Request alespoň jeden z příznaků:

```
TRQ_WRITE_PRESS_DELAY  
TRQ_WRITE_RELEASE_DELAY  
TRQ_WRITE_DEGLITCH  
TRQ_WRITE_CHARGEWS  
TRQ_WRITE_STABILIZATION  
TRQ_WRITE_FILTERWEIGHT
```

Pro uložení celé struktury Settings do registrů řadiče displeje struktury lze použít konstantu TRQ_WRITE_SETTINGS obsahující všechny výše uvedené příznaky.

4.3.1.7. Položka TTouchCommDriver.Request

Položka **Request** je určena k řízení automatu pro zápis a čtení parametrů do/z řadiče

dotykového panelu. Položka je určena jak ke čtení, tak k zápisu.

```
Request      : TTCDRequests;
```

Jednotlivé příznaky položky Request, tj. konstanty s prefixem TRQ_READ_ nebo TRQ_WRITE_ (viz. kapitola 4.1.3) určují požadavky na čtení nebo zápis parametru z/do řadiče dotykového panelu.

Nastavením jednoho nebo více příznaků v této položce se spustí automat pro čtení nebo zápis parametru. Příslušný příznak zůstává nastaven do té doby než je požadavek proveden.

Příklad: Fragment automatu pro vyčtení teploty pomocí ovladače řadiče dotykového panelu.

```
case State of
  ST_...:
  begin
  end;

  ST_READ_TEMP:
  begin
    { Požadavek na čtení teploty }
    Include( Driver^.Request, TRQ_READ_TEMP );
    State := ST_WAIT_READ_TEMP;
  end;

  ST_WAIT_READ_TEMP:
  begin
    { Čekáme na dokončení požadavku }
    if TRQ_READ_TEMP in Driver^.Request then
    begin
      { Teplota byla přečtena }
      WriteLn( Driver^.Temp / 16 );
      State := ..;
    end;
  end;

  ST_...:
  begin
  end;
end;
```

4.3.1.8. Položka TTouchCommDriver.ResetRoutine

Položka **ResetRoutine** obsahuje odkaz na proceduru, která provádí reset řadiče dotykového panelu. Tento odkaz může obsahovat hodnotu nil. Pak je reset řadiče prováděn pouze pomocí zprávy odeslané po sériové lince.

```
ResetRoutine : TTCDResetRoutine;
```

Typ **TTCDResetRoutine** je definován následovně:

```
TTCDResetRoutine = procedure( ATCD: PTouchCommDriver );
```

4.3.1.9. Konstruktor `TTouchCommDriver.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 provede inicializaci řadiče sériové linky.

4.3.1.10. Destruktor `TTouchCommDriver.Done`

smazat z programu

4.3.1.11. Metoda `TTouchCommDriver.GetEvent`

Metoda **GetEvent** slouží k získání poslední události z fronty událostí dotykového panelu.

```
function GetEvent( var AEvent: TTCDEvent ): Boolean;
```

Parametry:

AEvent Odkaz na proměnnou, do které bude uložena struktura události dotykového panelu..

Návratové hodnoty:

V případě úspěchu vrací metoda vrací hodnotu `True`. V případě, že je fronta prázdná metoda vrací hodnotu `False` a parametr `AEvent` je nedefinovaný.

Poznámky:

Třída **TTouchCommDriver** implementuje frontu událostí o maximální délce 16 událostí. Do této fronty jsou ukládány přijaté informace o stisku a uvolnění dotykového panelu. Každá událost, tj. struktura **TTCDEvent** (viz. kapitola 4.1.4.1) obsahuje informace zda jde o stisk nebo uvolnění, souřadnice X a Y a čas, kdy byla zpráva přijata.

4.3.1.12. Metoda `TTouchCommDriver.Tick`

Metoda **Tick** provádí krok automatu ovladače komunikační linky dotykového panelu.

```
procedure Tick; virtual;
```

Parametry:

Metoda nemá žádné parametry.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

4.3.1.13. Metoda TTouchCommDriver.InterruptProc

Metoda **InterruptProc** je nízkoúrovňová rutina pro příjem a vysílání znaků do/z přijímacích front.

```
procedure Tick; virtual;
```

Parametry:

Metoda nemá žádné parametry.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Při komunikaci s řadičem sériové linky dotykového panelu se nevyužívá přerušení. Obvyklá obsluha, která se nachází v přerušení je nahrazena periodickým voláním této metody. Metoda **InterruptProc** musí být volána alespoň jednou za 10ms.

Metoda **IntterruptProc** se nevolá nikdy přímo, ale vždy skrze proceduru **TCDInterruptProc** (viz. kapitola 4.4.3).

4.3.1.14. Metoda TTouchCommDriver.ResetController

Metoda **ResetController** provádí hardwarový reset řadiče dotykového panelu.

```
procedure ResetController; virtual;
```

Parametry:

Metoda nemá žádné parametry.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Po resetu řadiče jsou automaticky nastaveny všechny parametry uložené v položce Settings.

4.3.2. Třída TTouchPanelDriver

Třída **TTouchPanelDriver** implementuje ovladač dotykového panelu pro různé typy terminálu (Touch11, Touch33 apod.). Tato třída vychází z báze třídy pro implementaci ovladačů myši **TMouseDriver** (viz. dokumentace ke knihovně IoDrv)

```
PTouchPanelDriver = ^TTouchPanelDriver;
TTouchPanelDriver = object( TMouseDriver )
public
    IOBase          : Word;

    constructor Init( AIOBase: Word );

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

    procedure Tick; virtual;
    procedure GetEvent( var AEvent: TEvent ); virtual;
    procedure SetMode( ACalibration: Boolean ); virtual;
    function Calibrate( DispA, MouseA, DispB, MouseB,
        DispC, MouseC : TPoint ): Boolean; virtual;

    procedure SetPressDelay( AValue: Byte ); virtual;
    function GetPressDelay: Byte; virtual;
    procedure SetReleaseDelay( AValue: Byte ); virtual;
    function GetReleaseDelay: Byte; virtual;
    procedure SetDeglitch( AValue: Byte ); virtual;
    function GetDeglitch: Byte; virtual;
    procedure SetChargeWS( AValue: Byte ); virtual;
    function GetChargeWS: Byte; virtual;
    procedure SetFilterWeight( AValue: Byte ); virtual;
    function GetFilterWeight: Byte; virtual;
    procedure SetStabilization( AValue: Byte ); virtual;
    function GetStabilization: Byte; 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;
    procedure GetCalibration( var ACalibration:
        TMouseCalibration ); virtual;
    procedure SetCalibration( const ACalibration:
        TMouseCalibration ); virtual;

end;
```

4.3.2.1. Položka TTouchPanelDriver.IOBase

Položka **IOBase** obsahuje báze adresu terminálu v I/O prostoru. Položka je inicializovaná konstruktorem a je určena pouze pro čtení


```
IoBase      : Word;
```

4.3.2.2. Konstruktor TTouchPanelDriver.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:

4.3.2.3. Metoda TTouchPanelDriver.Initialize

Metoda **Initialize** provádí inicializaci hardware dotykového panelu.

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

Parametry:

Metoda nemá žádné parametry.

Návratové hodnoty:

Metoda vrací hodnotu True v případě úspěšné inicializace řadiče dotykového panelu.

Poznámky:

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

Metoda provede inicializaci řadiče komunikační linky pomocí metody **TCDInitialize**.

4.3.2.4. Metoda TTouchPanelDriver.Finalize

Metoda **Finalize** provádí deinicializaci hardware dotykového panelu.

```
procedure Finalize; virtual;
```

Parametry:

Metoda nemá žádné parametry.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu

Poznámky:

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

Metoda provede uvolnění řadiče komunikační linky dotykového panelu pomocí procedury **TCDRelease**.

4.3.2.5. Metoda TTouchPanelDriver.Tick

Metoda **Tick** provádí jeden krok automatu ovladače dotykového panelu.

```
procedure Tick; virtual;
```

Parametry:

Metoda nemá žádné parametry.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Metoda **Tick** předefinovává metodu **Tick** svého předka, tj. třídy **TMouseDriver** (viz. dokumentace ke knihovně IoDrv).

4.3.2.6. Metoda TTouchPanelDriver.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**.

Událost	Popis události
evMouseDown	Stisk tlačítka myši (stisk dotykového panelu)
evMouseUp	Uvolnění tlačítka myši (uvolnění dotykového panelu)
evMouseMove	Pohyb ukazatele myši (tlačítka mohou být stisknuta i uvolněna)
evMouseDb1	Dvojklik tlačítka myši („dvojstisk“ dotykového panelu)
evMouseRep	Automatické opakování stisku tlačítka

4.3.2.7. Metoda TTouchPanelDriver.SetMode

Metoda **SetMode** slouží k přepnutí ovladače dotykového panelu do kalibračního režimu a zpět.

```
procedure SetMode( ACalibration: Boolean ); virtual;
```

Parametry:

ACalibration Při přepnutí do kalibračního režimu obsahuje parametr hodnotu True. Při přepnutí do pracovního režimu obsahuje hodnotu False.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

V kalibračním režimu ovladač neprovádí konverzi souřadnic podle nastavených kalibračních konstant a vrací přímo souřadnice přečtené z řadiče dotykového panelu. (Týká se položky Pos této třídy a položky Pos struktury **TEvent** vrácené metodou **GetEvent**).

4.3.2.8. Metoda TTouchPanelDriver.Calibrate

Metoda **Calibrate** provádí nastavení převodních konstant ze souřadnic dotykového panelu na souřadnice displeje.

```
function Calibrate( DispA, MouseA, DispB, MouseB,  
DispC, MouseC : TPoint ): Boolean; virtual;
```

Parametry:

DispA Souřadnice displeje bodu A.
MouseA Souřadnice polohovacího zařízení bodu A.
DispB Souřadnice displeje bodu B.
MouseB Souřadnice polohovacího zařízení bodu B.

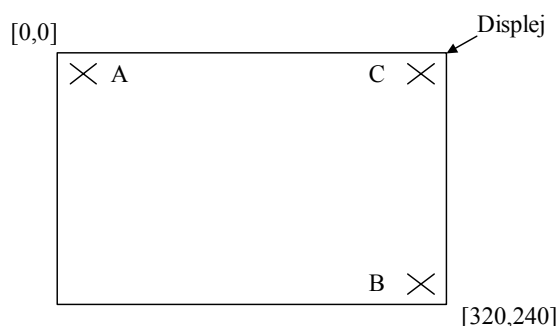
DispC	Souřadnice displeje bodu C.
MouseC	Souřadnice polohovacího zařízení bodu C.

Návratové hodnoty:

Metoda vrací hodnotu True, pokud byla kalibrace úspěšná. Pokud metoda vrátí hodnotu False, kalibrace neproběhla a nastavení kalibračních konstant nebylo změněno.

Poznámky:

Metoda **Calibrate** umožňuje tříbodovou kalibraci, tj. kalibraci pomocí třech bodů umístěných obvykle v rozích displeje. Viz. následující obrázek:



Uživatel nastaví v kalibračním režimu postupně označí body A, B a C. Poté se zavolá metoda **Calibrate** s odměřenými souřadnicemi (parametry MouseA, MouseB a MouseC) a s požadovanými souřadnicemi (DispA, DispB, DispC). Metoda nastaví kalibrační konstanty (viz. struktura **TMouseCalibration** v dokumentaci ke knihovně IoDrv). Pro přímé nastavení a čtení aktuálních kalibračních konstant lze použít metodu **SetCalibration** a **GetCalibration**.

4.3.2.9. Metoda TTouchPanelDriver.SetPressDelay

Metoda **SetPressDelay** slouží pro nastavení položky PressDelay struktury **TTCDSettings** (viz. kapitola 4.1.1).

```
procedure SetPressDelay( AValue: Byte ); virtual;
```

Parametry:

AValue	Hodnota parametru.
--------	--------------------

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Význam parametru a jeho meze jsou uvedeny v kapitole 4.1.1.

4.3.2.10. Metoda TTouchPanelDriver.GetPressDelay

Metoda **GetPressDelay** vrací aktuální nastavení položky PressDelay struktury **TTCDSettings** (viz. kapitola 2.1.1).

```
function GetPressDelay: Byte; virtual;
```

Parametry:

Metoda nemá žádné parametry.

Návratové hodnoty:

Metoda vrací aktuální hodnotu parametru. Rozsah návratových hodnot je uveden k příslušné položce v kapitole 4.1.1.

4.3.2.11. Metoda TTouchPanelDriver.SetReleaseDelay

Metoda **SetReleaseDelay** slouží pro nastavení položky PressDelay struktury **TTCDSettings** (viz. kapitola 4.1.1).

```
procedure SetReleaseDelay( AValue: Byte ); virtual;
```

Parametry:

AValue Hodnota parametru.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Význam parametru a jeho meze jsou uvedeny v kapitole 4.1.1.

4.3.2.12. Metoda TTouchPanelDriver.GetReleaseDelay

Metoda **GetReleaseDelay** vrací aktuální nastavení položky ReleaseDelay struktury **TTCDSettings** (viz. kapitola 2.1.1).

```
function GetReleaseDelay: Byte; virtual;
```

Parametry:

Metoda nemá žádné parametry.

Návratové hodnoty:

Metoda vrací aktuální hodnotu parametru. Rozsah návratových hodnot je uveden

k příslušné položce v kapitole 4.1.1.

4.3.2.13. Metoda TTouchPanelDriver.SetDeglitch

Metoda **SetDeglitch** slouží pro nastavení položky Deglitch struktury **TTCDSettings** (viz. kapitola 4.1.1).

```
procedure SetDeglitch( AValue: Byte ); virtual;
```

Parametry:

AValue Hodnota parametru.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Význam parametru a jeho meze jsou uvedeny v kapitole 4.1.1.

4.3.2.14. Metoda TTouchPanelDriver.GetDeglitch

Metoda **GetDeglitch** vrací aktuální nastavení položky Deglitch struktury **TTCDSettings** (viz. kapitola 2.1.1).

```
function GetDeglitch: Byte; virtual;
```

Parametry:

Metoda nemá žádné parametry.

Návratové hodnoty:

Metoda vrací aktuální hodnotu parametru. Rozsah návratových hodnot je uveden k příslušné položce v kapitole 4.1.1.

4.3.2.15. Metoda TTouchPanelDriver.SetChargeWS

Metoda **SetChargeWS** slouží pro nastavení položky ChargeWS struktury **TTCDSettings** (viz. kapitola 4.1.1).

```
procedure SetChargeWS ( AValue: Byte ); virtual;
```

Parametry:

AValue Hodnota parametru.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Význam parametru a jeho meze jsou uvedeny v kapitole 4.1.1.

4.3.2.16. Metoda TTouchPanelDriver.GetChargeWS

Metoda **GetChargeWS** vrací aktuální nastavení položky ChargeWS struktury **TTCDSettings** (viz. kapitola 2.1.1).

```
function GetChargeWS: Byte; virtual;
```

Parametry:

Metoda nemá žádné parametry.

Návratové hodnoty:

Metoda vrací aktuální hodnotu parametru. Rozsah návratových hodnot je uveden k příslušné položce v kapitole 4.1.1.

4.3.2.17. Metoda TTouchPanelDriver.SetFilterWeight

Metoda **SetFilterWeight** slouží pro nastavení položky FilterWeight struktury **TTCDSettings** (viz. kapitola 4.1.1).

```
procedure SetFilterWeight( AValue: Byte ); virtual;
```

Parametry:

AValue Hodnota parametru.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Význam parametru a jeho meze jsou uvedeny v kapitole 4.1.1.

4.3.2.18. Metoda TTouchPanelDriver.GetFilterWeight

Metoda **GetFilterWeight** vrací aktuální nastavení položky FilterWeight struktury **TTCDSettings** (viz. kapitola 2.1.1).

```
function GetFilterWeight: Byte; virtual;
```

Parametry:

Metoda nemá žádné parametry.

Návratové hodnoty:

Metoda vrací aktuální hodnotu parametru. Rozsah návratových hodnot je uveden k příslušné položce v kapitole 4.1.1.

4.3.2.19. Metoda TTouchPanelDriver.SetStabilization

Metoda **SetStabilization** slouží pro nastavení položky Stabilization struktury **TTCDSettings** (viz. kapitola 4.1.1).

```
procedure SetStabilization( AValue: Byte ); virtual;
```

Parametry:

AValue Hodnota parametru.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

Význam parametru a jeho meze jsou uvedeny v kapitole 4.1.1.

4.3.2.20. Metoda TTouchPanelDriver.GetStabilization

Metoda **GetStabilization** vrací aktuální nastavení položky Stabilization struktury **TTCDSettings** (viz. kapitola 2.1.1).

```
function GetStabilization: Byte; virtual;
```

Parametry:

Metoda nemá žádné parametry.

Návratové hodnoty:

Metoda vrací aktuální hodnotu parametru. Rozsah návratových hodnot je uveden k příslušné položce v kapitole 4.1.1.

4.3.2.21. Metoda TTouchPanelDriver.SetDbClickDelay

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

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


Parametry:

AValue Délka časového intervalu v milisekundách.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

4.3.2.22. Metoda TTouchPanelDriver.GetDbClickDelay

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

```
function GetDbClickDelay: Integer; virtual;
```

Parametry:

Metoda nemá žádné parametry.

Návratové hodnoty:

Metoda vrací délku časového intervalu v milisekundách.

Poznámky:

4.3.2.23. Metoda TTouchPanelDriver.SetDbClickArea

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

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

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.

4.3.2.24. Metoda TTouchPanelDriver.GetDbClickArea

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

```
function GetDbClickArea: Integer; virtual;
```

Parametry:

Metoda nemá žádné parametry.

Návratové hodnoty:

Metoda vrací počet pixelů..

Poznámky:

4.3.2.25. Metoda TTouchPanelDriver.SetRepeatDelay

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

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

Parametry:

AValue Délka časového intervalu v milisekundách.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

4.3.2.26. Metoda TTouchPanelDriver.GetRepeatDelay

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

```
function GetRepeatDelay: Integer; virtual;
```

Parametry:

Metoda nemá žádné parametry.

Návratové hodnoty:

Metoda vrací délku časového intervalu v milisekundách.

Poznámky:

4.3.2.27. Metoda TTouchPanelDriver.SetRepeatRate

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

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

Parametry:

AValue Délka časového intervalu v milisekundách.

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

4.3.2.28. Metoda TTouchPanelDriver.GetRepeatRate

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

```
function GetRepeatRate: Integer; virtual;
```

Parametry:

Metoda nemá žádné parametry.

Návratové hodnoty:

Metoda vrací délku časového intervalu v milisekundách.

Poznámky:

4.3.2.29. Metoda TTouchPanelDriver.GetCalibration

Metoda **GetCalibration** provádí čtení kalibračních konstant pro převod souřadnic polohovacího zařízení na souřadnice displeje.

```
procedure GetCalibration( var ACalibration:  
                          TMouseCalibration ); virtual;
```

Parametry:

ACalibration Parametr bude po návratu metody naplněn kalibračníma konstantama (viz. kapitola **Chyba! Nenalezen zdroj odkazů.**).

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

4.3.2.30. Metoda TTouchPanelDriver.SetCalibration

Metoda **SetCalibration** provádí nastavení kalibračních konstant pro převod souřadnic polohovacího zařízení na souřadnice displeje.

```
procedure SetCalibration( const ACalibration:
                        TMouseCalibration ); virtual;
```

Parametry:

ACalibration	Parametr odkazující se na strukturu obsahující kalibrační konstanty (viz. dokumentace ke knihovně IODrv).
--------------	---

Návratové hodnoty:

Metoda nevrací žádnou hodnotu.

Poznámky:

4.4. Procedury

4.4.1. Funkce TCDInitialize

Funkce **TCDInitialize** provádí inicializaci ovladače komunikační linky řadiče dotykového panelu.

```
function TCDInitialize( AIOBase: Word; AResetSignal: Boolean ):
Boolean;
```

Parametry:

AIOBase	Bázová adresa řadiče sériové linky dotykového panelu.
AResetSignal	Je nastaven na True, pokud je připojen standardní je v I/O prostoru zpřístupněn reset signál řadiče dotykového panelu (pouze Touch11 a Touch33)

Návratové hodnoty:

Funkce vrací hodnotu True, pokud se podařilo úspěšně ovladač inicializovat. Odkaz na instanci ovladače je uložen v globální proměnné TCDriver (viz. kapitola 4.5.1).

Poznámky:

Ovladač komunikační linky řadiče dotykového panelu je sdílený prostředek. Využívá jej jak ovladač dotykového panelu, tak ovladač displeje terminálu pro účely měření teploty displeje.

První volání funkce **TCDInitialize** vytvoří instanci třídy **TTouchCommDriver**. Každé další volání pouze zvýší interní čítač, který zaznamenává počet odkazů na tuto instanci. Ta část programu, která zavolala funkce **TCDInitialize** musí po ukončení používání tohoto ovladače zavolat proceduru **TCDRelease**, která sníží interní čítač odkazů. Pokud tento čítač klesne na nulu, je instance ovladače uvolněna z paměti.

4.4.2. Procedura TCDRelease

Funkce **TCDRelease** provádí uvolnění odkazu na ovladač komunikační linky řadiče dotykového panelu.

```
procedure TCDRelease;
```

Parametry:

Procedura nemá žádné parametry.

Poznámky:

Poté, co je procedura **TCDRelease** zavolána tolikrát, kolikrát byla zavolána funkce **TCDInitialize** je ovladač komunikační linky uvolněn z paměti a proměnná **TCDriver** je nastavena na hodnotu **nil**.

4.4.3. Procedure TCDInterruptProc

Procedura **TCDInterruptProc** provádí krok přijímacího a vysílacího automatu ovladače komunikační linky dotykového panelu. Tato procedura musí být volána z aplikace alespoň jednou za 10ms.

```
procedure TCDInterruptProc;
```

Parametry:

Procedura nemá žádné parametry.

Poznámky:

Tuto proceduru lze (a je vhodné) volat z přerušení od časovače.

4.5. Globální proměnné

4.5.1. Proměnná TCDriver

Proměnná **TCDriver** obsahuje odkaz na instanci ovladače komunikační linky řadiče dotykového panelu, tj. třídy **TTouchCommDriver**. Tato proměnná je nastavována funkcemi **TCDInitialize** (viz. kapitola 4.4.1) a **TCDRelease** (viz. kapitola 4.4.2). Proměnná **TCDriver** je určena pouze pro čtení.

```
TCDriver : PTouchCommDriver;
```