

Um TFT-Displays an einen 8 bit Prozessor anzuschließen gibt es einige Möglichkeiten. Auf Grund der Rechenleistung ist es schwierig eine akzeptable Geschwindigkeit zu erreichen. Die Firma FTDI hat dafür eine Lösung geschaffen, welche auf der einen Seite ein sehr gutes Preisverhältnis zu erreichen und auf der anderen Seite eine sehr gute Verarbeitungsgeschwindigkeit zu bieten. Die Chips der FT8xx-Serie sind in Einzelstückzahlen für 5 € erwerbbar, als Module in der Größenordnung von 60..80 € inklusive Display.

Der Chip vereint einen kompletten Displaycontroller, eine Touchcontroller (je nach Chip resistiv oder kapazitiv) sowie einen einfachen Soundcontroller, er wird entweder per SPI oder I2C angesteuert wird. Die Ansteuerung per I2C haben wir bewußt nicht unterstützt, weil diese aufgrund der Geschwindigkeit nicht so optimal ist.

Auf Grund der Architektur ist es nicht möglich einen einzelnen Pixel zu ändern sondern es wird bei jeder Änderung immer das komplette Display neu beschrieben, das hört sich sehr aufwändig und zeitintensiv an, ist es aber nicht, da hier eine völlig andere Herangehensweise gewählt wurde. Man übergibt dem Chip um z.B. einen Button darzustellen nur die Koordinaten, den Text und ein paar Optionen, den Rest erledigt der Controller. Dazu besitzt der Chip 2 interne 4kb große Buffer, wobei einer immer dazu dient die gerade gewünschte Anzeige zu realisieren und der andere um die nächste Anzeige in den Chip zu schreiben, danach werden beide Buffer gewechselt und das Spiel kann von vorn beginnen. Als Einschränkung gilt hier nur, ein Screen kann nur 4096 Byte an Anweisung beinhalten. Trotzdem kann man sehr ansprechende Ergebnisse erzielen. Die Grafiken und die eigenen Fonts, die dargestellt werden sollen, sowie eventuelle eigene Soundschnipsel müssen sich dazu im internen Speicher. Dieser ist bei den FT80x-Typen 256 KB und bei den FT81x-Typen 1 MB groß. Nachdem diese im internen Speicher vorliegen sind nur wenig Bytes an Kommandos notwendig um diese darzustellen. Damit hier auch größere Mengen an Grafiken verwendet werden können unterstützt der Treiber bis zu zwei parallel angeschlossenen SPI-Roms.

Die Initialisierung des Treibers wird wie folgt vorgenommen:

```
Import SysTick, ....
    EVE_Graphic;

Define
EVE_Type      = FT800;                // FT801, FT810, FT811, FT812 or FT813
EVE_SPI= SPI_C, PortA.2, PortA.3, PortA.4; //SPI-Port,CS,PD,Int
EVE_Size      = 4.3;                  // 3.5, 4.3, 5.0, Userdef
EVE_View      = BottomUp_View;       // TopDown_View, PortraitUp_View,PortraitDown_View
EVE_Clock     = ClockIntern; //ClockExtern

{$REUTILIZE SPI_C}
AT25DFxx      = SPI_C, PortA.0;
AT25DF_ROM2 = PortA.1;                // optional 2nd chip select

uses
    uFT800,
    uFT800Ex,
```

uFT800Rom;

Um einen Screen zu erzeugen sind die beiden wichtigsten Befehle der Init- und der Swapbefehl, alles was zwischen diesen beiden Befehlen steht kommt zur Anzeige.

Unit uFT800

Function FT800_BootupConfig:Boolean;

Function FT800_BootupConfig(LCDConfig:TFT800Config):Boolean;

Der Chip muß als erstes auf das entsprechende Display abgestimmt werden, sofern Sie nicht ein Userdefines-Display verwenden, verwenden Sie bitte den ersten Aufruf. Für den anderen Fall bitte den zweiten Aufruf verwenden und vorher die Variable LCDConfig ausfüllen.

TFT800Config=Record

REG_HCYCLE : Word;

REG_HOFFSET : Word;

REG_HSYNC0 : Word;

REG_HSYNC1 : Word;

REG_VCYCLE : Word;

REG_VOFFSET : Word;

REG_VSYNC0 : Word;

REG_VSYNC1 : Word;

REG_SWIZZLE : Byte;

REG_PCLK_POL: Byte;

REG_HSIZE : Word;

REG_VSIZE : Word;

REG_PCLK : Byte;

end;

Als Beispiel hier das 3.5 Zoll Modul von FTDI

```
REG_HSIZE      := FT_DispWidth_35           : Word = 320;
REG_VSIZE      := FT_DispHeight_35          : Word = 240;
REG_HCYCLE     := FF_DispHCycle_35          : Word = 408;
REG_HOFFSET    := FT_DispHOffset_35         : Word = 70;
REG_HSYNC0     := FT_DispHSync0_35          : Word = 0;
REG_HSYNC1     := FT_DispHSync1_35          : Word = 10;
REG_VCYCLE     := FT_DispVCycle_35          : Word = 263;
REG_VOFFSET    := FT_DispVOffset_35         : Word = 13;
REG_VSYNC0     := FT_DispVSync0_35          : Word = 0;
REG_VSYNC1     := FT_DispVSync1_35          : Word = 2;
REG_PCLK       := FT_DispPCLK_35            : Byte   = 8;
REG_SWIZZLE    := FT_DispSwizzle_35         : Byte   = 0;
REG_PCLK_POL   := FT_DispPCLKPol_35         : Byte   = 0; // 1
```

Function FT800_InitScreen(ScreenColor:LongWord): Boolean;

Initialisiert den Screen mit der Hintergrundfarbe ScreenColor nach dem Muster 00RRGGBB und startet die Bufferübertragung an den entsprechenden Buffer.

Procedure FT800_SwapScreen;

Beendet die Bufferübertragung und wechselt den Buffer, nach der Übertragung dieses Befehles wird der gerade übertragene Inhalt sofort zur Anzeige gebracht und der bisdahin verwendete Buffer steht jetzt für die Übertragung eines neuen Inhaltes bereit.

Function FT800_SetColorLong(Value:LongWord) : Boolean;

Setzt die Grundfarbe

Function FT800_SetVColorLong(Value:LongWord) : Boolean;

Setzt die Vordergrundfarbe

Function FT800_SetBGColorLong(Value:LongWord) : Boolean;

Setzt die Hintergrundfarbe

Function FT800_DrawButton (x, y, w, h, Font, options : Word; Caption : String[32]; Track : Byte) : Boolean;

Zeichnet einen Button an die Position x,y mit der Breite w und der Höhe h, der verwendetet Font ist Font (0..31), Caption der darzustellende Text, den Parameter Options und für die Touchrückmeldung eine ID (1..127).

Function FT800_DrawToggle (x, y, w, Font, option, State : Word; onCaption, offCaption : String[16]; Track : Byte) : Boolean;

Zeichnet einen Schalter an die Position x,y mit der Breite w, der verwendetet Font ist Font (0..31), On/Off-Caption der darzustellende Text, den Parameter Options und für die Touchrückmeldung eine ID (1..127).

Function FT800_DrawKey (x, y, w, h, Font, options : Word; Caption : String[32]; Track : Byte) : Boolean;

Zeichnet einen Anzahl Tasten an die Position x,y mit der Breite w und der Höhe h, die Anzahl der Tasten wird durch die Anzahl der Buchstaben in Caption bestimmt, der verwendetet Font ist Font (0..31), den Parameter Options und für die Touchrückmeldung eine ID (1..127).

Function FT800_DrawSpinner (x, y, Style, Scale : Word) : Boolean;

Zeichnet einen Waitcursor an die Position x,y.

Function FT800_DrawClock (x, y, r, options : Word; h, m, S, ms : Byte; Track : Byte) : Boolean;

Zeichnet eine Uhr an die Position x,y mit dem Radius r und der Uhrzeit h,m,s,ms, den Parameter Options und für die Touchrückmeldung eine ID (1..127).

Function FT800_DrawDial (x, y, r, options, Value : Word; Track : Byte) : Boolean;

Zeichnet einen Dialknob an die Position x,y mit dem Radius r und dem Wert Value (0..65535), den Parameter Options und für die Touchrückmeldung eine ID (1..127).

Function FT800_DrawGauge (x, y, r, options, major, minor, Value, Range : Word; Track : Byte) :

Boolean;

Zeichnet eine Gauge an die Position x,y mit dem Radius r und dem Wert Value (0..65535), den Parameter Options und für die Touchrückmeldung eine ID (1..127).

Function FT800_DrawLine (x1, y1, x2, y2 : Word; Width : Word) : Boolean;

Zeichnet eine Linie von der Position x1,y1 nach x2,y2 mit der Dicke Width.

Function FT800_DrawPoint (x, y, d : Word; r, g, b : Byte) : Boolean;

Zeichnet einen Kreis an die Position x,y, Radius d und der Farbe RGB.

Function FT800_DrawRect (x1, y1, x2, y2 : Word; Width : Word) : Boolean;

Zeichnet ein Rechteck von der Position x1,y1 nach x2,y2 mit der Dicke Width.

Function FT800_DrawSlider (x, y, w, h, opt, Value, Range : Word; Track : Byte) : Boolean;

Zeichnet einen Slider an die Position x,y mit der Breite w und der Höhe h, dem Wert Value, den Parameter Options und für die Touchrückmeldung eine ID (1..127).

Function FT800_DrawProgress (x, y, w, h, options, Value, Range : Word; Track : Byte) : Boolean;

Zeichnet eine Progressbar an die Position x,y mit der Breite w und der Höhe h, dem Wert Value, den Parameter Options und für die Touchrückmeldung eine ID (1..127).

Function FT800_DrawScrollbar(x, y, w, h, opt, Value, Size, Range : Word; Track : Byte) : Boolean;

Zeichnet einen Scrollbar an die Position x,y mit der Breite w und der Höhe h, dem Wert Value, den Parameter Options und für die Touchrückmeldung eine ID (1..127).

Function FT800_DrawText (x, y, Font, option : Word; Text : String[64]) : Boolean;

Zeichnet einen Text an die Position x,y mit dem entsprechenden Font.

Function FT800_DrawNumber (x, y, Font, option : Word; Value : LongInt) : Boolean;

Zeichnet eine Zahl an die Position x,y mit dem entsprechenden Font.

Procedure FT800_Boundary (x, y, w, h :Word);

Begrenzt die Ausgabe der darauffolgenden Befehle auf ein entsprechendes Rechteck.

Function FT800_CMDTrack (x, y, w, h : Word; Track : Byte) : Boolean;

Erstellt eine Touchrückmeldung mit der ID (1..127), für ein Rechteck von x,y und der Breite w und der Höhe H.

Function FT800_DrawGradient (x1,y1,x2,y2:Word;SColor,EColor:LongWord;Alpha:Byte) : Boolean;

Zeichnet ein Gradient-Rechteck von der Position x1,y1 nach x2,y2 mit der Startfarbe SColor und der Endfarbe EColor.

Procedure FT800_SetRotateDisplay(Value:Byte);

Setzt die Rotation des Displays im laufenden Programm, beachte Sie dabei, das Sie danach sofort den Touch kalibrieren müssen.

Function FT800_GetRotateDisplay:Byte;

Liefert die aktuelle Rotation des Displays zurück.

Procedure FT800_TouchCalibrate(Value : String[32]);

Ruft die Kalibrierung des Touches auf, in der Mitte des Displays wird der Text Value dargestellt.

Function FT800_IsTouchCalibrated : Boolean;

Testet ob der Touch kalibriert wurde.

Procedure FT800_GetTouchCalibrated(var Temp : TCalibArray);

Liest die Werte der Touchkalibrierung zurück.

Procedure FT800_SetTouchCalibrated(Temp : TCalibArray);

Schreibt die Werte der Touchkalibrierung in den Chip.

Function FT800_GetTouchTrack : Byte;

Liefert die ID des aktuellen Controls sonst 0.

Function FT800_LoadJPGToMem(Source : Pointer; Size : Word; PicType : Byte; Dest : LongWord) : Boolean;

Schreibt eine JPG-Datei in den Speicher des Chips, Source muß auf den Startpunkt der Datei zeigen.

Procedure FT800_LoadBitmaptoDisplay(Handle:Byte;Source : LongWord; Format : Byte; x, y, w, h : Word);

Schreibt eine Pixel-Bitmap in den Speicher des Chips, Source muß auf den Startpunkt der Bitmap zeigen.

Function FT800_CmdFIFOIsReady:Boolean;

Testet den Befehlsbuffer und wartet bis dieser wieder bereit, bei False muß FT800_Reseterror aufgerufen werden..

Procedure FT800_ResetError;

Bei Fehlern in der Bufferübertragung muß dieser Befehl ausgeführt werden, sonst ist keine weitere Übertragung mehr möglich.

Hintergrundbeleuchtung

Procedure FT800_FadeDisplay(Value, TimeSleep : Byte);

Fadet die Helligkeit bis zur Helligkeit-Value in Schritten von TimeSleep in ms.

Procedure FT800_BackLight(Value : Byte);

Setzt die Displayhelligkeit auf den Wert Value.

Sound

Procedure FT800_PlaySound(Note, Effect, Volume : Byte);

Spielt einen Ton mit den entsprechenden Werten ab.

Procedure FT800_StopSound;

Stopt die Soundausgabe.

GPIO

Function FT800_GetGPIONDir(BitNr:Byte):Boolean;

Procedure FT800_SetGPIONDir(BitNr:Byte;Value:Boolean);

Function FT800_GetGPIOValue(BitNr:Byte):Boolean;

Procedure FT800_SetGPIOValue(BitNr:Byte;Value:Boolean);

CustomFont

Function FT800_SetCustomFont(BitmapHandle:Byte;FontData:TFT_Gpu_Fonts) : Boolean;

Procedure StartScreenShot;

Procedure SaveScreenShot(StartX,LenX,Y:Word;Buffer: pointer);

Procedure StopScreenShot;

Interne Befehle

Procedure FT800_WriteCMDEnd;

Procedure FT800_CMDUpdate;

Procedure FT800_CheckWritePointer;

Procedure FT800_IncWritePointer(Size : LongWord);

Function FT800_CheckCMDBufferSizeAndWait(Size : Word) : Integer;

Function FT800_CheckCMDBufferSize(Size : Word) : Boolean;

Function CheckFreeSpace : Word;

Procedure FT800_WriteCMD32(Data : LongWord);

Procedure FT800_WriteCMD16(Data : Word);

Procedure FT800_WriteCMD8(Data : Byte);

Procedure FT800_Write32(Adr, Data : LongWord);

Procedure FT800_Write16(Adr : LongWord; Data : Word);

Procedure FT800_Write8(Adr : LongWord; Data : Byte);

Function FT800_SwapDisplaylist : Boolean;

uFT800Ex

Die in dieser Unit verwendeten Befehl, zeichnen alle Controls mit den Farben der in der Unit uFT800Palette.pas vereinbarten Farben, es werden automatisch die Farben für ein Disabled Control, oder ein gedrücktes Control verwendet. Ansonsten entsprechen diese Befehle den vorher beschriebenen.

Function FT800_DrawButton_Pal (x, y, w, h, Font: Word; Caption : String[32]; Pressed:Boolean; Enable:Boolean; Track : Byte) : Boolean;

Function FT800_DrawHLButton_Pal (x, y, w, h, Font: Word; Caption : String[32]; Pressed:Boolean; Enable:Boolean; HighLight:Boolean; Track : Byte) : Boolean;

HL steht hier für ein Button, dem man den Parameter HighLight mitgeben kann, um z.B. in einem Kalender Termine markieren kann.

Function FT800_DrawToggle_Pal (x, y, w, Font : Word; State:Boolean; onCaption, offCaption : String[16]; Pressed:Boolean; Enable:Boolean; Track : Byte) : Boolean;

Function FT800_DrawKey_Pal (x, y, w, h, Font, options : Word; Caption : String[32]; Track : Byte) : Boolean;

Function FT800_DrawSpinner_Pal (x, y, Style, Scale : Word) : Boolean;

Function FT800_DrawClock_Pal (x, y, r, options : Word; h, m, S, ms : Byte; Pressed:Boolean; Enable:Boolean; Track : Byte) : Boolean;

Function FT800_DrawDial_Pal (x, y, r, options, Value : Word; Pressed:Boolean; Enable:Boolean; Track : Byte) : Boolean;

Function FT800_DrawGauge_Pal (x, y, r, options, major, minor, Value, Range : Word; Pressed:Boolean; Enable:Boolean; Track : Byte) : Boolean;

Function FT800_DrawLine_Pal (x1,y1,x2, y2 : Word; Width : Word; Color:Byte;Alpha:Byte) : Boolean;

Function FT800_DrawPoint_Pal (x, y, d : Word; Color : Byte;Alpha:Byte) : Boolean;

Function FT800_DrawRect_Pal (x1,y1,x2, y2 : Word; Width : Word; Color:Byte; Alpha:Byte) : Boolean;

Function FT800_DrawSlider_Pal (x, y, w, h, Value, Range : Word; Pressed:Boolean; Enable:Boolean; Track : Byte) : Boolean;

Function FT800_DrawProgress_Pal (x, y, w, h, Value, Range : Word; Pressed:Boolean; Enable:Boolean; Track : Byte) : Boolean;

Function FT800_DrawScrollbar_Pal (x, y, w, h, opt, Value, Size, Range : Word; Pressed:Boolean; Enable:Boolean; Track : Byte) : Boolean;

Function FT800_DrawNumber_Pal (x, y, Font,Option:Word; Signed:Boolean; Value : LongInt) : Boolean;

Function FT800_DrawText_Pal (x, y, Font, option : Word; Text : String[64]) : Boolean;

uFT800ROM

Function FT800_LoadStartlogo: LongWord;

Zeichnet ein Startlogo z.B. für einen Bootscreen.

Procedure FT800_LoadFontFromRomToMem (DirIndex:Word;Dest : LongWord);

Läd einen Font aus dem ROM in den Chip.

Function FT800_LoadJPGFromRomToMem (DirIndex:Word;PicType : Byte; Dest : LongWord): Integer;

Läd eine JPG-Datei vom ROM in den Chip.

Function FT800_LoadStreamFromRomToMem(DirIndex:Word;Dest : LongWord) : LongWord;

Lädt einen Stream vom ROM in den Chip, z.B. einen Soundschnipsel

Function FT800_GetRomVersion:Word;

Liefert die Version des ROMs zurück, wichtig nur bei Änderungen der ROM-Behandlung.

Function EveROM_GetRomSize :LongWord;

Liefert die Größe des ROMs

Function EveROM_GetFileInfo (DirIndex:Word):TFT800File;

Liefert die Dateiinformation des Items Dirindex

Function EveROM_Init:Boolean;

Initialisiert den ROM.

function EveROM_Ready : boolean;

function WriteToROM (Source : pointer; dest :LongWord; count : word) : boolean;

function ReadFromROM (dest : pointer; source:LongWord; count : word) : boolean;

function EveROMErasePage (Size:Byte;page_address : Longword) : boolean;