Application Note

AN_336

FT8xx - Selecting an LCD Display

Version 2.0

Issue Date: 2015-09-21

The EVE devices support a wide range of LCD panels due to their programmable display setting registers. This document covers some of the factors which should be considered when selecting an LCD panel to be used with the FTDI FT80x and FT81x devices and when configuring the device for the selected panel.

Future Technology Devices International Limited (FTDI)
Unit 1, 2 Seaward Place, Glasgow G41 1HH, United Kingdom
Tel.: +44 (0) 141 429 2777 Fax: + 44 (0) 141 429 2758
Web Site: http://ftdichip.com

Table of Contents

1 Introduction

The EVE devices support a wide range of LCD panels due to their programmable display setting registers. This document covers some of the factors which should be considered when selecting an LCD panel to be used with the FTDI FT80x and FT81x devices and when configuring the device for the selected panel.

2 Choosing a Display

When selecting an LCD panel and determining it's compatibility with the FT8xx, some important factors are highlighted below:

- Ensure that the resolution of the screen is within the capabilities of the particular EVE series device.
    - The FT80x supports resolutions up to 512 x 512, including WQVGA (480 x 272) and QVGA (320 x 240) displays. These are typically up to 5" screens.
    - The FT81x supports larger resolutions up to 800 x 600, including SVGA (800 x x600). These are typically up to 7" screens, although there are a few 10" displays on the market at this resolution also.

- Check that the ribbon cable of the display is compatible with the EVE board which is intended to be used. Some LCD modules also have different electrical requirements, for example the power and ground positions, or the LED connections for the backlight driver.

- If the panel uses a touchscreen, ensure that it matches the touch support of the particular member of the EVE family:
    - FT800 and FT810/FT812 support resistive touch panels
    - FT801 and FT811/FT813 support capacitive touch. The Capacitive Touch Controller must be one of the supported controllers. See section 6 for details.

- Check that the display is compatible with parallel RGB input signals and is configured for RGB mode. Some panels may require programming over a separate data interface in order to configure the display controller chip, and so the board may need to provide connections between these lines on the LCD ribbon cable and the MCU. This can be confirmed by referring to the datasheet for the LCD panel and also the datasheet for the controller chip itself used in the panel.

  The FT800, 801, 810, 811 support 18-bit RGB, with 6 lines per colour. The FT812 and 813 extend this to 8 bits per colour giving 24-bit colour capability.

- Some FT8xx boards use different GPIO lines to enable the display etc. The sample code provided on the FTDI website is configured to use the GPIO assignments to match the FTDI evaluation modules.

- Consult the datasheets of the LCD panel and the controller chip in the LCD panel, to determine the required values for the FT8xx's display setting registers. Further details on these are given in section 3.

3 Display Parameters

The following figures show the relationship between the timing registers of the FT8xx and the parameters which are typically used in an LCD panel datasheet.

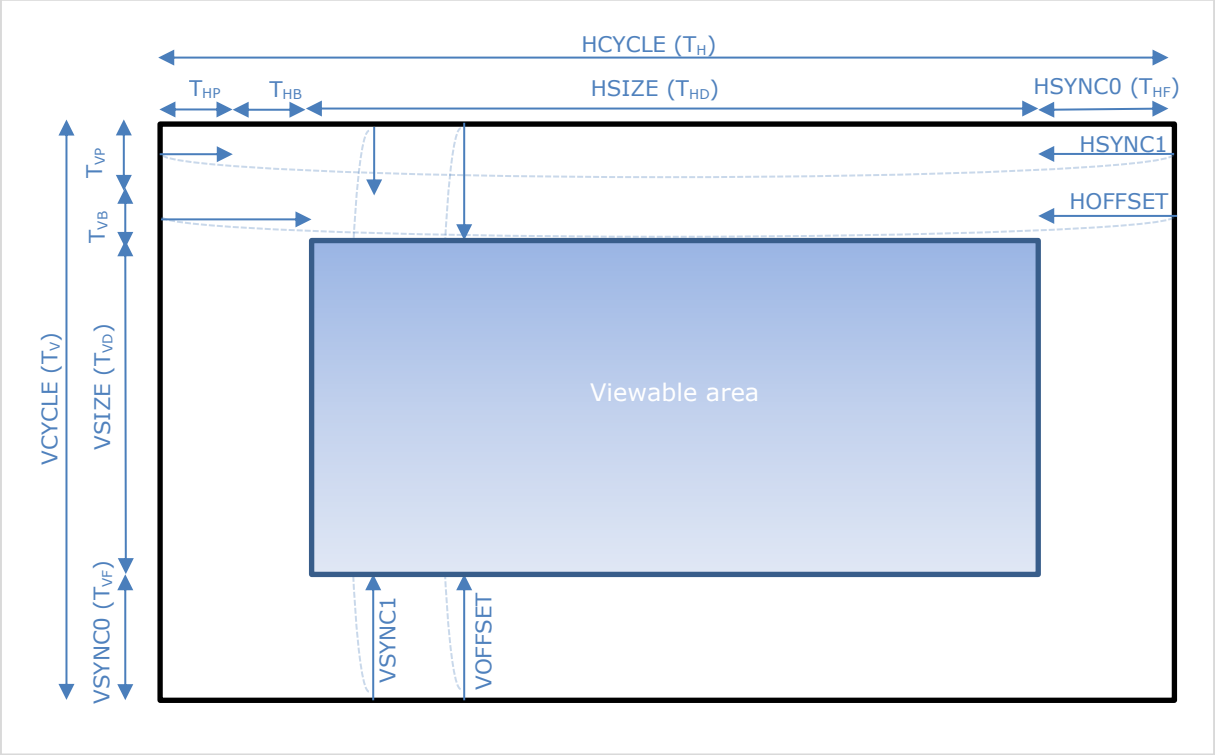

Figure 3.1 LCD Display Parameters

	FT800	Display	Description
Horizontal	REG_HCYCLE	T_H	**Total** length of line (visible and non-visible) (in PCLKs)
	REG_HSIZE	T_{HD}	Length of **visible** part of line (in PCLKs)
	REG_HOFFSET	$T_{HF} + T_{HP} + T_{HB}$	Length of **non-visible** part of line (in PCLK cycles)
	REG_HSYNC0	T_{HF}	Horizontal Front Porch
	REG_HSYNC1	$T_{HF} + T_{HP}$	Horizontal Front Porch plus Hsync Pulse width
Vertical	REG_VCYCLE	T_V	**Total** number of lines (visible and non-visible) (in lines)
	REG_VSIZE	T_{VD}	Number of **visible** lines (in lines)
	REG_VOFFSET	$T_{VF} + T_{VP} + T_{VB}$	Number of **non-visible** lines (in lines)
	REG_VSYNC0	T_{VF}	Vertical Front Porch
	REG_VSYNC1	$T_{VF} + T_{VP}$	Vertical Front Porch plus Vsync Pulse width
	REG_PCLK	T_{PCLK}	System Clock / REG_PCLK = PCLK frequency
	REG_PCLK POL		PCLK polarity (0 = rising edge, 1 = falling edge)
	REG_SWIZZLE		Defines arrangement of the RGB pins of the EVE device

Table 3.1 Translating between FT8xx and Display parameters

The following diagrams show how the timing parameters from Table 3.1 and Figure 3.1 above relate to the actual waveforms produced by the FT8xx.

Note that some display datasheets show the front porch (T_{HF}) at the left-hand side of the timing diagram (as shown below) and others show it at the right-hand side.

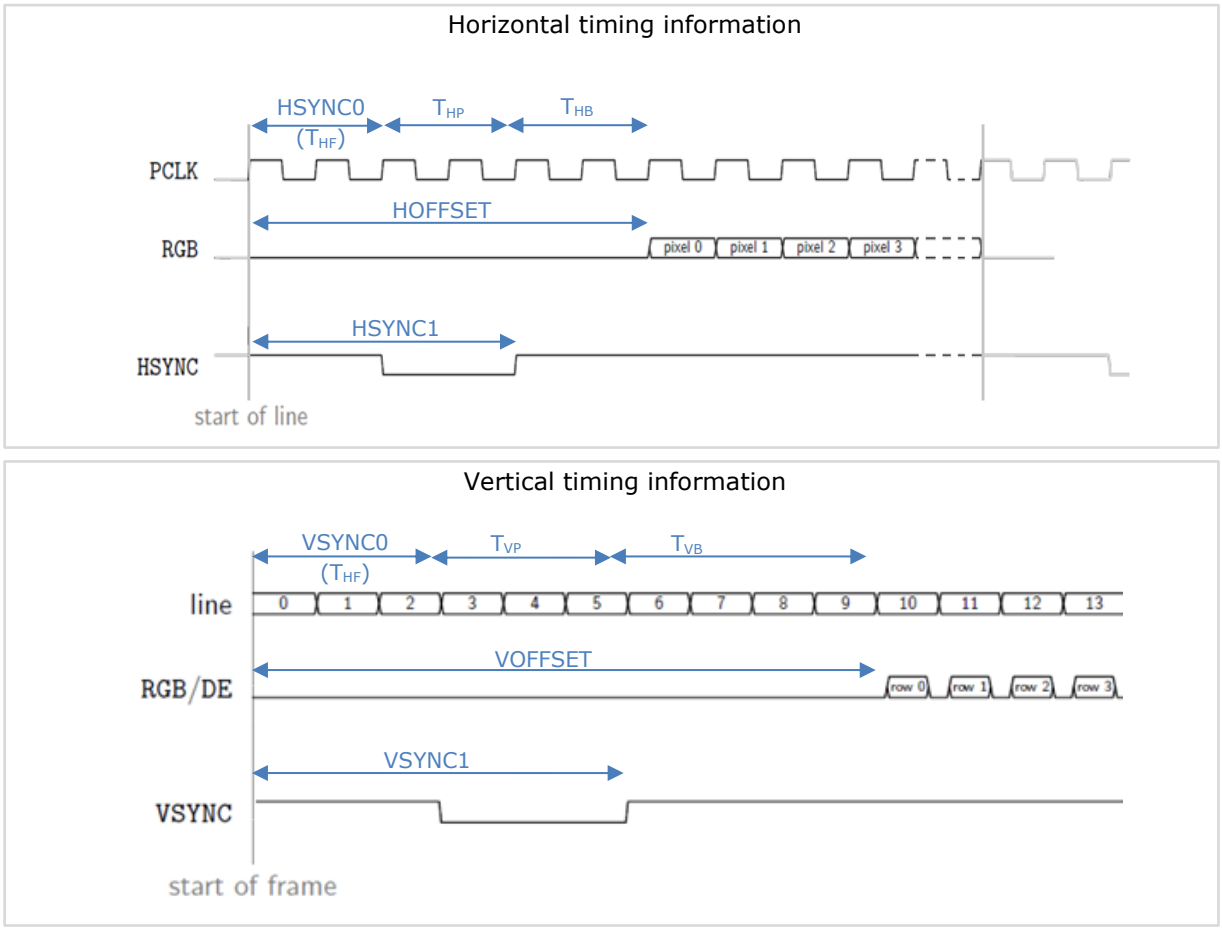

Figure 3.2 Timing Waveforms

In addition to the Horizontal and Vertical settings, Table 3.1 also mentions the PCLK and Swizzle settings, as described below.

PCLK

The FT8xx devices generate the PCLK output based upon dividing down the System Clock. The divider is specified by the REG_PCLK register.

The LCD datasheet will specify an allowable range of PCLK frequencies. When selecting a display, ensure that a value within the panel's supported range can be selected by the divider. The datasheets for the FT80x and FT81x provide further details of the system clock and dividers.

For the FT80x, the system clock can be 48MHz or 36MHz. This can be selected by the CLK48MHz or CLK36MHz system commands. The 48MHz setting is default. The REG_PCLK can be configured to divide this down to a value supported by the LCD panel.

For example, with 48MHz system clock and REG_PCLK = 5, PCLK output = 48MHz / 5 = 9.6MHz.

The FT81x offers an increased range of system clock settings which are 60MHz, 48MHz, 36MHz and 24MHz. 60MHz is the default setting. The REG_PCLK selects an integer division of the selected system clock. With the default 60MHz system clock, the divider allows values of 60MHz (divide-by-1), 30MHz (divide-by-2), and 20MHz (divide-by-3) and so on.

For example an LCD display may support between 26MHz and 48MHz in which case the divider would be set to give a 30MHz output.

PCLK Polarity

The display datasheet will also indicate which edge of PCLK the data must be valid on. The FT8xx supports either polarity through setting the REG_PCLK_POL.

Swizzle

The Swizzle setting allows the R/G/B video signals to be arranged to one of several different orders on the FT8xx's RGB pins. This simplifies the process of laying out the PCB traces between the FT8xx and the display. Ensure that the swizzle is set correctly for the actual connections on the PCB between the FT8xx and the LCD.

4 90 Degree Rotation (Portrait Orientation)

The FT81x (FT810, FT811, FT812, FT813) support 90 degree rotation of the display, allowing a landscape display panel to be used in a portrait orientation. The FT81x handles the rotation of the image, which allows this to be achieved without additional workload on the host MCU.

Whilst the register REG_ROTATE can be written directly, this would not rotate the touch transforms. Therefore, it is recommended to use the co-processor command CMD_SETROTATE instead to rotate both the image and the touch transformations.

Note that the application running on the host MCU must take account of the selected screen orientation and position items accordingly to make best use of the landscape or portrait display area.

At the time of writing, the majority of panels are designed for landscape orientation (where the horizontal parameters are for the longest side, e.g. 480 x 272). For applications requiring a portrait orientation, the FT81x family are therefore recommended.

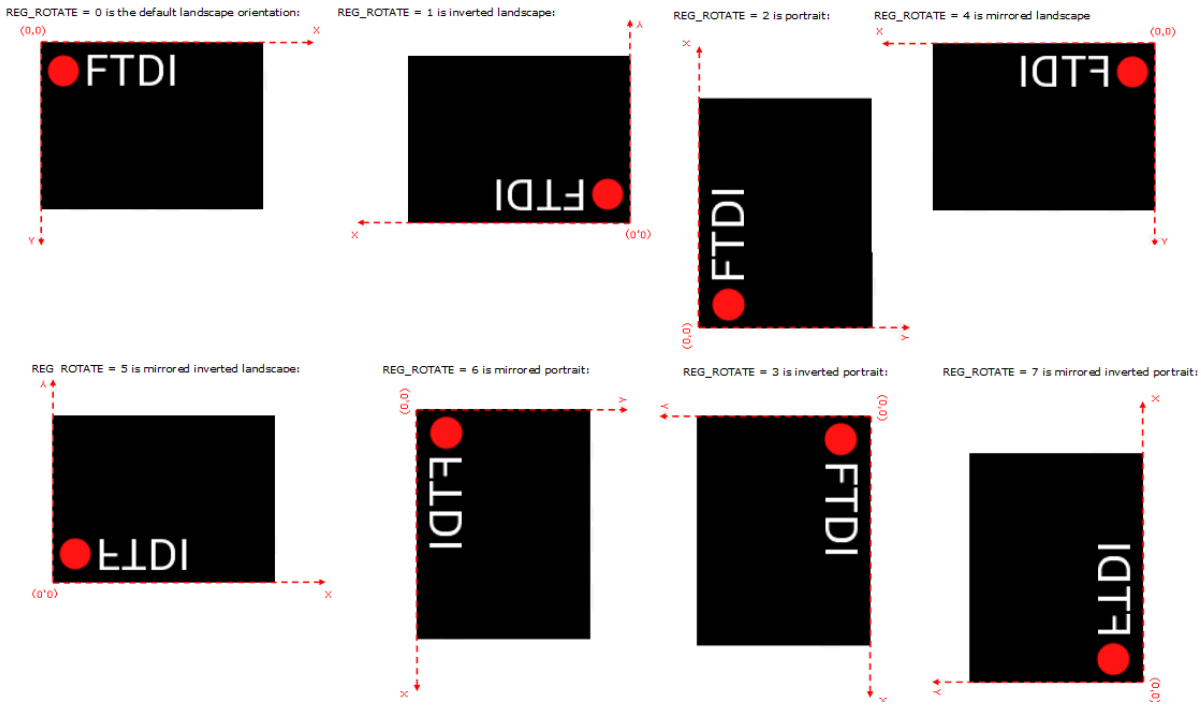

Figure 4.1 Screen rotation options on the FT81x

5 Pixel Ratio

Some of the available 7" LCD panels have been found to use non-square pixels, which can result in some items not displaying as intended. For example, a circle would appear oval and a square would appear as rectangular. Whilst the application could in some cases adjust it's drawing of primitives to reduce the effect, this could still affect the appearance of other items such as the clock widget.

It is recommended to check the datasheet for any display being considered to ensure that the pixels are square. Display datasheets normally list the Active Area size, but some do not list the pixel or dot pitch. If not specified directly in the display datasheet, the pixel width/height ratio can be calculated from the Active Area size.

For example, if the display Active Area size is 154.08 x 85.92 (W x H) and the resolution is 800 x 480, then the average pixel size (assuming no gap between pixels) is

Width 154.08 / 800 = 0.1926mm
Height 85.92 / 480 = 0.179mm

The W/H ratio of each pixel (average) is therefore 1.076

Note: Some datasheets specify the dot pitch. Each pixel contains three 3 dots (R, G and B dot). If dot pitch is 0.063mm (W) x 0.179mm (H), then the pixel pitch is (0.063x3) =0.189 mm (W) x 0.179mm (H). The W/H ratio of each pixel is therefore 1.056.

2.1 Display Attributes >>Return to CONTENTS

ITEM	STANDARD VALUE	UNIT
Display Format	800（RGB）x 480 Dots	--
Display Connector	FPC	--
FPC Connector	40 Pin,0.5mm Pitch, SMD Horizontal Type Top contact	--
Operating Temperature	-20 ~ +70	℃
Storage Temperature	-30 ~ +80	℃
Touch Panel Optional	Yes	--
Sunlight Readable	No	--

2.2 Mechanical Attributes >>Return to CONTENTS

ITEM	STANDARD VALUE	UNIT
Diagonal Size	7.0"	inch
Outline Dimension with FPC Folded	164.9(W) x100(H)	mm
Active Area	154.08(W) x 85.92(H)	mm
Pixel Size	0.063(W) x 0.179(H)	mm
Net Weight	126.0	g

2.3 Electrical Attributes >>Return to CONTENTS

ITEM	STANDARD VALUE	UNIT
IC Package	COG	--
Driver	OTA7001A	--
Interface	RGB Interface	--
Response Time (Typ)	20	ms

Figure 5.1 Display Parameters

Note: FTDI recommend checking the datasheet for any display which is being considered, but the non-square pixels appear to be primarily on 7" panels rather than smaller sizes.

Two example displays which have square pixels are shown below.

Innolux AT070TN83 V1 7"
Module size: 165X104mm
Active Area: 152.4x91.44mm
Interface: RGB666
Datasheet: http://elinux.org/images/0/07/AT070TN83.pdf
pixel pitch = 0.1905 x 0.1905 mm

Microtech MTF070TN83-V1 7"
Module size: 165.00X104.0X5.5mm
Active Area: 152.4X91.44mm
Interface: RGB888
Datasheet: TBD
pixel pitch 0.1926(H) x 0.1790(V)

6 Capacitive Touch Controller Compatibility

This section provides information on capacitive touch controller compatibility for the FT801, FT811 and FT813. The FT800, FT810 and FT812 support resistive screens via their integrated resistive touch screen interface, and so this section does not apply.

Note that this list is correct at the time of writing and is provided for guidance only. FTDI cannot support compatibility of any other touch controllers beyond those specified in this section. FTDI recommend that customers test the intended touch controller as part of their prototyping stage, as specifications of third-party ICs are subject to change by the manufacturer beyond the control of FTDI.

Compatible Touch Controllers

The following controllers are compatible with the FT801, FT811 and FT813. Further testing has enabled the FT5x26, 5x36 and 6x06 to be added to the list of supported controllers.

Azoteq

- IQS5xx

FocalTech

- FT5x06 series: FT5206/FT5306/FT5406
- FT5x16 series: FT5316
- FT5x26 series: FT5426 (see FT801 notes)
- FT5x36 series: FT5336 (see FT801 notes)
- FT6x06 series: FT6306 (see FT801 notes)

The particular models shown in blue text were tested for each family.

Voltage Levels

Some of the supported touch controllers are intended for 1.8V operation on their VCCIO. In order to achieve compatibility with the touch controllers, the I/O voltage of the touch controller must match the FT8xx's VCCIO. In these cases, the following may help:

- The I/O voltage of the touch controller could be set to 3.3V. This may require customisation by the touch control manufacturer.
- If this is not possible, level shifters would be required to translate the levels between 1.8V for the touch controller and 3.3V (in particular, the FT801 device uses a VCCIO of 3.3V).
- For the FT811 and FT813 only, the VCCIO of the FT811/813 could be lowered to 1.8V. In this case, the RGB signals would also be 1.8V and so the LCD selected must be compatible with 1.8V signals. The FT801 does not support 1.8V VCCIO and so this option is not available.

Modes

Some Focaltech controllers support two interrupt modes: query mode and trigger mode. The FT8xx series support only the trigger mode. In addition, some Focaltech controllers also offer two different types of I2C protocol: IIC Standard Protocol and FTS 26 Bytes (IIC/SPI) Protocol. The FT8xx series support only the IIC Standard protocol.

I2C Address

For the FT801, the capacitive touch controller must have its I2C address set to 0x38 as the FT801 does not support configuration of I2C address via the register map.

Note: 0x38 = 7'b0111000 excluding the read/write bit. This translates to an actual address byte value 0x70 for write operation, and 0x71 for read operation, after the R/W bit is added.

For the FT811 and FT813, the FT81x series have the additional feature that the I2C address can be specified via the REG_TOUCH_CONFIG register. Some of the touch controllers listed above have a configurable/programmable I2C address. The register allows the FT81x to be configured to communicate with controllers programmed with a non-default address.

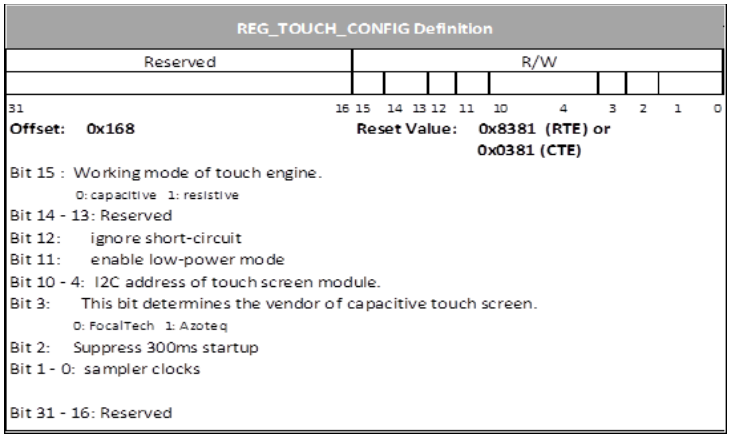

Figure 6.1 Touch Configuration register

For example, if an FT5206 controller has I2C address b'0111000, the REG_TOUCH_CONFIG register would be written as 0x0381.

Hardware

The following diagram shows an example of connecting the FT8xx to a capacitive touch controller.

The controller IC is often integrated into either the display panel itself or within the ribbon cable attached to the panel. The FT8xx board would have a suitable connector to match the connector on the panel's ribbon, such as the FPC connector below.

The communication between the FT8xx and the capacitive touch controller takes place via the I2C lines SDA and SCL in addition to the reset and interrupt lines. The initialisation and communication of the capacitive touch controller via these lines is all handled by the FT8xx transparently to the user application.

The touch controller may have a single power input line or may have separate VCC and VCCIO supply lines. As mentioned in the previous section, the I/O voltage of the CTP must match that of the FT8xx. If the CTP has only one supply pin, it must support operation from the same voltage as the FT8xx's VCCIO. Please refer the datasheet of the CTP to verify its recommended power supply configuration.

Note that the number of connections in the ribbon cable and the order of the signals may vary and so please consult the datasheet of the LCD panel and the CTC device used in the panel to confirm the pinout. For example, some panels may not use the Wake signal.

The circuit uses pull-ups on the interrupt, SDA and SCL lines and also the Wake (where present). The 33 Ohm series resistors help to improve EMI performance.

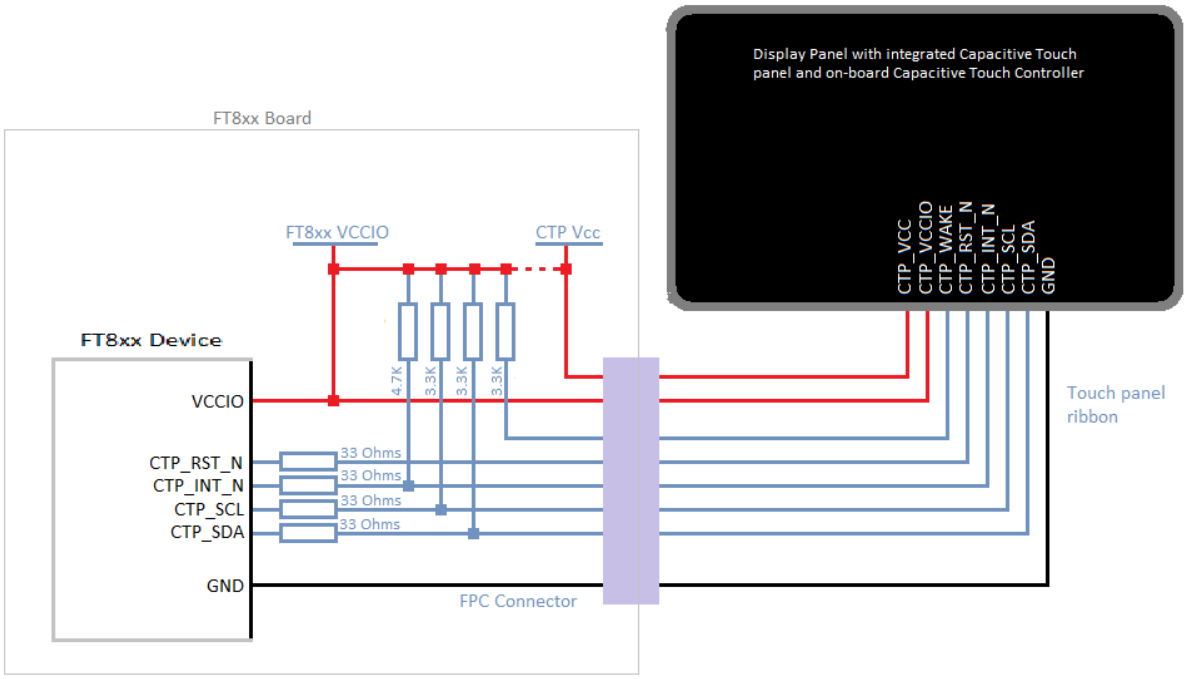

Figure 6.2 Capacitive Touch Connections

7 Touch Screen Calibration

The calibration feature of the FT8xx allows it to determine the alignment of the touch panel relative to the screen. When the application runs the calibration command on the FT8xx, the user is requested to tap three dots on the screen. The FT8xx calculates six transform values during the calibration routine which then allow it to make adjustments so that the users touch is aligned to the graphics underneath. It is required that the screen is calibrated for any application which will use touch to ensure touch accuracy. The calibration is required for both resistive and capacitive touch.

Note that for capacitive panels (FT801, FT811, FT813) the calibration is carried out in Compatibility mode and the transforms still apply when switching to extended mode.

The FT8xx series include a calibration command which makes the above process simple for the host MCU and removes any need to manually calculate transform values or to apply these each time a touch is detected. On completion of the command, the registers are already loaded with the required transforms and the touch system is fully calibrated.

However, on power-down, these transform values are lost. In most of the demo applications, this requires the calibration routine to be run after each power-up. To provide a better end-user experience, it is possible to run the calibration once, store the values on the MCU's non-volatile memory and then restore the values after each power on. Note that the values must be stored within the MCU or a storage device attached to the MCU as the FT8xx does not include non-volatile memory for storage of these values.

Storing Calibration Values

In order to store the values, the calibration could for example be run during a factory test of the finished product. The host MCU would do the following.

During factory test:

- Run a co-processor list containing the Calibrate command
- Wait for REG_CMD_READ == REG_CMD_WRITE (this indicates the completion of the co-processor command list and will only occur once the user has tapped the three dots)
- The command will have populated the registers REG_TOUCH_TRANSFORM_A to REG_TOUCH_TRANSFORM_F with the transform values
- The MCU can now read these six registers with standard 32-bit register read commands and can store the values in the MCU's EEPROM for example

On Power Up:

- The MCU can now write the six values back to their respective REG_TOUCH_TRANSFORM registers with standard 32-bit register writes, instead of running the calibration command.

Re-calibrate Option:

- It may also be desirable to provide a menu option in the application or another way in which the user can re-run the calibration if required (e.g. during maintenance of a machine or if the screen had been damaged and replaced with a new LCD panel).

Example

The pseudo-code example below demonstrates one possible application. In this scenario, on power-up, the MCU would initialise the FT8xx and screen registers and then enters the calibration routine only if the calibration has never been run before or if the user is pressing (touching) and holding the screen during power-up. Otherwise, it is considered that the calibration had been carried out already (and so suitable transform values are available in the MCU EEPROM) and the operator does not wish to re-calibrate. The values are loaded in this case from EEPROM and so the operator does not need to carry out the tapping of the calibration dots.

```
The code below could be called each time the MCU and FT8xx are powered up and before the main
application starts. If the calibration data is not already stored in the MCU's EEPROM or if
the user touches and holds the screen during power-up, the calibration routine is run and the
resulting values are read by the MCU and stored in its EEPROM. Otherwise, the values from the
EEPROM from the previous calibration are written to the FT8xx's REG_TOUCH_TRANSFORM_A to
REG_TOUCH_TRANSFORM_F registers.

This code uses byte 0 of the MCU's EEPROM to indicate whether the calibration data has already
been stored in the EEPROM (0x7C was used to indicate 'values stored') and uses EEPROM bytes 1
- 24 to store the actual calibration data copied from the FT8xx's REG_TOUCH_TRANSFORM_A, _B,
_C, _D, _E and _F registers.

// Check if calibration data exists already or if the user is touching the screen.

If ((IsTouch()) || (EEPROM.Read(0) != 0x7C))
{
        Blank();                        // Blank the screen
        while (IsTouch());              // Wait for user to release touch

        write(REG_PWM_DUTY, 128);       // Ensure the display PWM is at 100%

        BeginCoProList();               // Start a new co-processor command list

        CMD DLSTART                     // DL Start command
        CMD CLEAR(1,1,1)                // Clear the buffers
        CMD COLOR_RGB(255,255,255)      // Set colour for the subsequent text
        CMD TEXT(screen.w/2, screen.h/2, 28, OPT_CENTERX|OPT_CENTERY, "please tap on
               the dot");
        CMD CALIBRATE                   // Run the actual calibration
        CMD DISPLAY                     // Display command
        CMD SWAP                        // Swap command

        FlushCoProBuffer();             // Send above co-processor commands to the FT8xx
        WaitCmdFifoIdle();              // Wait until co-processor finishes execution
                                        // i.e. until REG_CMD_READ == REG_CMD_WRITE

        // The FT8xx's registers REG_TOUCH_TRANSFORM_A to REG_TOUCH_TRANSFORM_F now have
        // their calibrated values. The MCU can read the six 32-bit values. Here, we read
        // them a byte at a time since the EEPROM is programmed on a byte-by-byte basis.

        for (int i = 0; i < 24; i++)
        {
               EEPROM.write(1 + i, rd32(REG_TOUCH_TRANSFORM_A + i));
       }
       EEPROM.write(0, 0x7c);           // Write loc 0 to 0x7C to show the data is stored

        // Now EEPROM(0) has value 0x7C and EEPROM(1) to (24) have the values of the
        // six 32-bit REG_TOUCH_TRANSFORM registers
}
else
{
```

```
        // If the calibration values were already in EEPROM and the user had not touched
        // the screen on power-up then we read the existing EEPROM values and write them
        // to the FT8xx's REG_TOUCH_TRANSFORM registers
        for (int i = 0; i < 24; i++)
                wr32(REG_TOUCH_TRANSFORM_A + i, EEPROM.read(1 + i));
}

// Now, the main application can begin
```

8 Conclusion

The FT8xx supports a wide range of LCD panels thanks to the configurability of its video output timing. This application note provides information to help when selecting an LCD panel and also to illustrate the relationship between LCD datasheet parameters and the FT8xx's display settings registers. It also highlights some of the considerations specific to the new enhanced FT81x series.

9 Contact Information

Head Office – Glasgow, UK

Future Technology Devices International Limited
Unit 1, 2 Seaward Place, Centurion Business Park
Glasgow G41 1HH
United Kingdom
Tel: +44 (0) 141 429 2777
Fax: +44 (0) 141 429 2758

E-mail (Sales) sales1@ftdichip.com
E-mail (Support) support1@ftdichip.com
E-mail (General Enquiries) admin1@ftdichip.com

Branch Office – Tigard, Oregon, USA

Future Technology Devices International Limited (USA)
7130 SW Fir Loop
Tigard, OR 97223-8160
USA
Tel: +1 (503) 547 0988
Fax: +1 (503) 547 0987

E-Mail (Sales) us.sales@ftdichip.com
E-Mail (Support) us.support@ftdichip.com
E-Mail (General Enquiries) us.admin@ftdichip.com

Branch Office – Taipei, Taiwan

Future Technology Devices International Limited (Taiwan)
2F, No. 516, Sec. 1, NeiHu Road
Taipei 114
Taiwan , R.O.C.
Tel: +886 (0) 2 8791 3570
Fax: +886 (0) 2 8791 3576

E-mail (Sales) tw.sales1@ftdichip.com
E-mail (Support) tw.support1@ftdichip.com
E-mail (General Enquiries) tw.admin1@ftdichip.com

Branch Office – Shanghai, China

Future Technology Devices International Limited (China)
Room 1103, No. 666 West Huaihai Road,
Shanghai, 200052
China
Tel: +86 21 62351596
Fax: +86 21 62351595

E-mail (Sales) cn.sales@ftdichip.com
E-mail (Support) cn.support@ftdichip.com
E-mail (General Enquiries) cn.admin@ftdichip.com

Web Site

http://ftdichip.com

Distributor and Sales Representatives

Please visit the Sales Network page of the FTDI Web site for the contact details of our distributor(s) and sales representative(s) in your country.

Appendix A– References

Document References

FT800 Datasheet

FT81x Datasheet

FT800 Programmers Guide

FT81x Programmers Guide

Acronyms and Abbreviations

Terms	Description
CTC	Capacitive Touch Controller
EVE	Embedded Video Engine
LCD	Liquid Crystal Display
SPI	Serial Peripheral Interface

Appendix B – List of Tables & Figures

List of Figures

List of Tables

Appendix C– Revision History

Document Title: AN_336 FT8xx - Selecting an LCD Display

Document Reference No.: FT_001078

Clearance No.: FTDI#400

Product Page: http://www.ftdichip.com/EVE.htm

Document Feedback: Send Feedback

Revision	Changes	Date
1.0	Initial release	2014-07-15
2.0	Updated to include the FT81x series	2015-09-21