Application Note

AN_334

FT801 Polygon Application

Document Reference No.:FT_001075

Version 1.0

Issue Date: 2014-07-22

This document introduces the setup of the FT801 Polygon Application running on MSVC. The objective of the Polygon Application is to enable users to become familiar with the usage of the multi-touch functionality of FT801, the design flow, and the display list used to design the desired user interface or visual effect.

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

This application demonstrates the multi-touch functionality of the FT801 chip by drawing a polygon on the screen, where the corners are defined by the points on the screen touched by the user.

The polygon created is used to lighten the selected area of a bitmap image.

Note: a maximum of 5 touch points may be used.

1.1 Overview

The document will provide a basic understanding of the FT801 multi-touch functionality.

The Arduino version of the application requires an SD card to read and load the bitmaps. The bitmaps must be copied from the Test folder to the SD card root folder.

1.2 Scope

This document can be used by software programmers to develop GUI applications using the FT801 with any MCU via SPI.

For information on how to use the project file and source code, refer to the following application note: **FT_App_Gradient Application note**

Note that detailed documentation is available at www.ftdichip.com/EVE.htm including:

FT801 Datasheet

FT800 Series Programming Guide

2 Application Flow

2.1 Flowchart

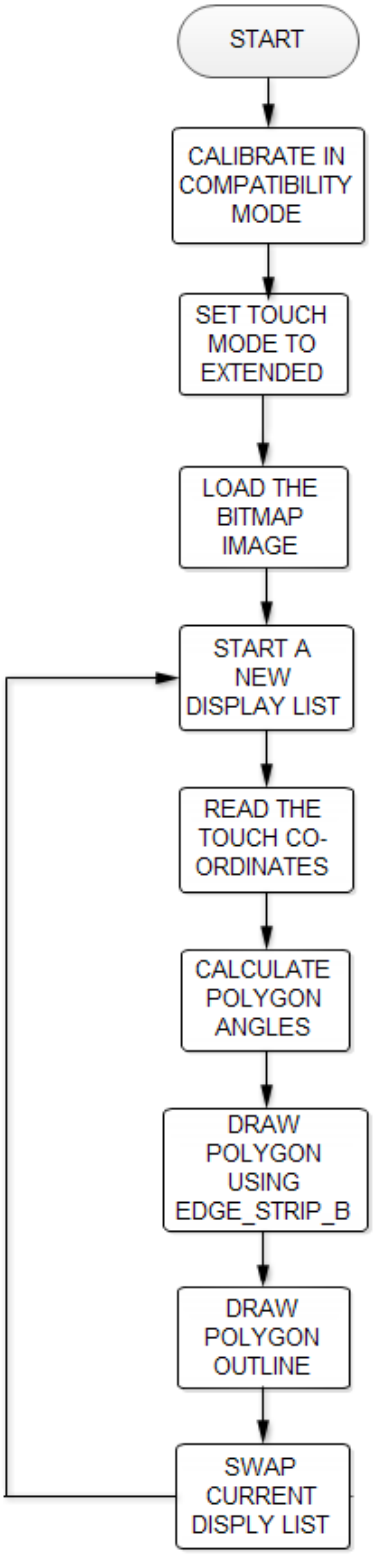

Figure 2-1 Flowchart

3 Description

The parameters which need to be initialized when constructing the display list are detailed below.

3.1 Intialization

3.1.1 Set Extended mode for multitouch

By default, the FT801 touch engine works in compatibility mode and in this mode only one touch point is detected (in order to be backward-compatible with the FT800). In extended mode, the FT801 touch engine can detect up to 5 touch points simultaneously.

Before entering in extended mode, user needs to do calibration in compatibility mode.

A co-processor command list is started. This command will clear the display parameters.

```
Ft_Gpu_CoCmd_Dlstart(phost);
Ft_App_WrCoCmd_Buffer(phost,CLEAR(1,1,1));
```

The following commands set the colour and then print a text message to the user which tells them to tap on the dots during the following calibration routine. The FT800's built-in calibration routine is then called.

```
Ft_App_WrCoCmd_Buffer(phost,COLOR_RGB(255,255,255));
Ft_Gpu_CoCmd_Text(phost,FT_DispWidth/2,FT_DispHeight/2,28,OPT_CENTERX|OPT_CENTERY,"Ple
ase tap on a dot");
Ft_Gpu_CoCmd_Calibrate(phost,0);
```

The display list is then terminated and swapped to allow the changes to take effect.

```
Ft_App_WrCoCmd_Buffer(phost,DISPLAY());
Ft_Gpu_CoCmd_Swap(phost);
Ft_App_Flush_Co_Buffer(phost);
Ft_Gpu_Hal_WaitCmdfifo_empty(phost);
```

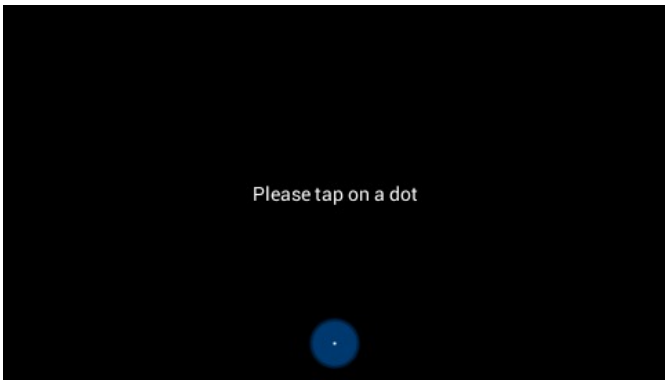

Figure 3-1 Calibration screen

As this application is designed to demonstrate FT801's multitouch functionality, it is necessary to set the mode to extended. For more information, refer to the FT800 Series Programming Guide (see Appendix A– References).

```
Ft_Gpu_Hal_Wr8(phost, REG_CTOUCH_EXTENDED, CTOUCH_MODE_EXTENDED);
```

3.1.2 Download the Bitmap

This application uses one bitmap, which will be in the background. The bitmap is downloaded into the desired location in the FT801's Graphics RAM. The bitmap handles are also assigned here.

After downloading the bitmap into the FT801, it will be present in the background, as shown in the following screenshot:

Figure 3-2 Bitmap loaded in background

The application then uses the COLOR_RGB command to set the red, green and blue values of the FT801 colour buffer, which will be applied to the downloaded image. The following screenshot shows the resulting effect on the bitmap image.

```
Ft_App_WrCoCmd_Buffer(phost, COLOR_RGB(0x40,0x40,0x40));
```

Figure 3-3 Bitmap after COLOR_RGB command

Note: These commands, followed by a swap display list command, must then be sent to the FT801 using the Flush command. See the provided sample project for the full code listing.

3.2 Functionality

The Polygon Demo is an interactive demonstration where the user can touch the screen with a maximum of 5 touch points simultaneously. This section shows the steps taken to detect the touch and display the polygon.

3.2.1 Read ctouch control registers

As the FT801 has a different touch engine and touch control registers compared to the FT800, and is capable of detecting upto 5 simultaneous touches, the application needs to read the 5 ctouch registers. The code below shows this:

```
ft_void_t read_extended(ft_int16_t sx[5], ft_int16_t sy[5])
{
  ft_uint32_t sxy0, sxyA, sxyB, sxyC;
  sxy0 = Ft_Gpu_Hal_Rd32(phost, REG_CTOUCH_TOUCH0_XY);
  sxyA = Ft_Gpu_Hal_Rd32(phost, REG_CTOUCH_TOUCH1_XY);
  sxyB = Ft_Gpu_Hal_Rd32(phost, REG_CTOUCH_TOUCH2_XY);
  sxyC = Ft_Gpu_Hal_Rd32(phost, REG_CTOUCH_TOUCH3_XY);

  sx[0] = sxy0 >> 16;
  sy[0] = sxy0;
  sx[1] = sxyA >> 16;
  sy[1] = sxyA;
  sx[2] = sxyB >> 16;
  sy[2] = sxyB;
  sx[3] = sxyC >> 16;
  sy[3] = sxyC;

  sx[4] = Ft_Gpu_Hal_Rd16(phost,REG_CTOUCH_TOUCH4_X);
  sy[4] = Ft_Gpu_Hal_Rd16(phost,REG_CTOUCH_TOUCH4_Y);
}
```

3.2.2 Draw polygon

To draw the polygon, the application reads the touch co-ordinates in a loop. After reading the coordinates, it calculates the angles based on these co-ordinates, using the atan8 and atan2 functions.

To draw the polygon itself, the application first uses the COLOR_MASK to disable writing of the colour components. It then uses the STENCIL_OP and STENCIL_FUNC commands to fill the drawn polygon with color. This is done by the code below:

```
    Ft_App_WrCoCmd_Buffer(phost,(COLOR_MASK(0,0,0,0)));

    Ft_App_WrCoCmd_Buffer(phost,STENCIL_OP(KEEP, INVERT));

    Ft_App_WrCoCmd_Buffer(phost,STENCIL_FUNC(ALWAYS, 255, 255));
```

It then uses the EDGE_STRIP_B and SCISSOR_XY commands to draw the polygon:

```
  ft_void_t poly_paint() {
    poly_x0 = max(0, poly_x0);
    poly_y0 = max(0, poly_y0);
```

```
      poly_x1 = min(16 * 480, poly_x1);
      poly_y1 = min(16 * 272, poly_y1);
      Ft_App_WrCoCmd_Buffer(phost,SCISSOR_XY(poly_x0, poly_y0));
      Ft_App_WrCoCmd_Buffer(phost,SCISSOR_SIZE(poly_x1 - poly_x0 + 1, poly_y1 -
poly_y0 + 1));
      Ft_App_WrCoCmd_Buffer(phost,BEGIN(EDGE_STRIP_B));
      poly_perim();
}
```

The screenshot below shows when the user has touched 3 points.

Figure 3-4 Polygon drawn with 3 points

To draw the outline for the polygon, the application uses the LINE_WIDTH, LINE_STRIP and VERTEX_2F functions, as shown below:

```
    Ft_App_WrCoCmd_Buffer(phost,RESTORE_CONTEXT());
    Ft_App_WrCoCmd_Buffer(phost, COLOR_RGB(0x80,0x80,0xff));
    Ft_App_WrCoCmd_Buffer(phost, LINE_WIDTH(24));
    poly_outline();
```

```
ft_void_t poly_perim() {
   for (j = 0; j < poly_n; j++)
   Ft_App_WrCoCmd_Buffer(phost,VERTEX2F(poly_x[j], poly_y[j]));
   Ft_App_WrCoCmd_Buffer(phost,VERTEX2F(poly_x[0], poly_y[0]));
 }
 ft_void_t poly_outline() {
   Ft_App_WrCoCmd_Buffer(phost,BEGIN(LINE_STRIP));
   poly_perim();
 }
```

When the user touches the screen with 3 points, the outline is drawn as shown below.

Figure 3-5 Polygon outline drawn

The screenshot below shows the application with 5 simultaneous touches:

Figure 3-6 Polygon drawn with 5 touch points

4 Conclusion

This application has demonstrated the configuration of the FT801 for multi-touch operation and has also presented an example of the way in which the multi-touch capabilities can be used. The techniques shown here can be used to add a multi-touch interface to a wide variety of applications.

5 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

1. FT800 Series programmer guide
2. FT801 Embedded Video Engine Datasheet
3. Polygon App

Acronyms and Abbreviations

Terms	Description
SPI	Serial Peripheral Interface
GUI	Graphical User Interface

Appendix B – List of Tables & Figures

Appendix C– Revision History

Document Title: AN_334 FT801 Polygon Application

Document Reference No.: FT_001075

Clearance No.: FTDI#402

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

Document Feedback: Send Feedback

Revision	Changes	Date
1.0	Initial release	2014-07-22