This application note demonstrates a way of interfacing the FT800 to a Raspberry Pi. The FT800 is connected to the USB port of the Raspberry Pi via an MPSSE interface. An application running on the Raspberry Pi communicates with the FT800 through the FTDI Linux D2xx driver. This application note provides a simple example of developing C code to control the FT800, and the principles demonstrated can then be used to produce more complex applications.
Table of Contents

1 Introduction ........................................................................................................... 2
  1.1 Scope ................................................................................................................ 2
  1.2 Software Required ........................................................................................... 2
  1.3 Hardware Required .......................................................................................... 2

2 Hardware Block Diagram and Connection ....................................................... 3

3 Software ............................................................................................................... 4
  3.1 Software Layer Diagram .................................................................................. 4
  3.2 Sample code (main program is in SampleApp.c) .............................................. 4
  3.3 FT_CoPro_Cmds.h ............................................................................................ 5

4 Running the Example ........................................................................................ 6
  4.1 D2XX Linux Driver installation for Raspberry Pi .............................................. 6
  4.2 Run the FT800 Raspberry Sample Code .......................................................... 6

5 Test Results ........................................................................................................ 7

6 Conclusion .......................................................................................................... 8

7 Contact Information .......................................................................................... 9

Appendix A – References .................................................................................... 10
  Document References .......................................................................................... 10
  Acronyms and Abbreviations .............................................................................. 10

Appendix B – List of Figures ................................................................................ 11
  List of Figures ..................................................................................................... 11

Appendix C – Revision History ........................................................................... 12
1 Introduction

This application note documents how to connect a small display to a single board computer (SBC). It uses a Raspberry Pi (RPI) as the SBC and a VM800BU as the FT800 driven small display module. The VM800BU was chosen specifically as it has a USB interface, thus freeing up the RPI SPI ports for other functions.

Project source code may be downloaded from this link.

The sample project can work for any SBC with Linux OS and VM800BU and will 4 colored circles as shown in section 5 to demonstrate the connectivity.

1.1 Scope

This application can be used for most kinds of SBC with a USB host port and is not limited to the Raspberry Pi. However, the Raspberry Pi was chosen for this demonstration due to the widely available hardware platform and OS images.

This document describes how to connect and run an FT800 demo program on a Raspberry Pi system. Creating and configuring a Raspberry Pi board image is beyond the scope of this application note.

1.2 Software Required

This sample application requires the following software resources:

- Raspberry PIDORA (users may use other Linux OS for their systems)
- FTDI Linux D2XX driver for ARM processors (version 1.1.12 or later). Available from the FTDI drivers page

1.3 Hardware Required

- Raspberry Pi B+ board (or other SBC systems) with a TF card (4G bytes or more)
- VM800BU module (see note)
- USB A to Micro B cable (suggest FTDI accessory VA-FC-1M-BKW or VA-FC-1M-BLW)

Note: The VM800BU already includes an on-board USB-SPI interface and is recommended. However a VM800B or VM800C may be connected via an additional USB to SPI bridge such as the C232HM DDHSL-0 cable or VA800A-SPI module or directly to the RPI SPI port.
2 Hardware Block Diagram and Connection

This section summarizes the hardware connections used. A VM800BU is shown connected via the USB A to Micro B cable.

Figure 2.1 Hardware Block Diagram
3 Software

3.1 Software Layer Diagram

The software consists of several different layers, as shown below. The SampleApp.c file is where the actual FT800 application would be created. Users should modify the SampleApp.c file to create different displays. All the Co-processor Engine commands are provided in the project. Refer to Section 3.3 FT_CoPro_Cmds.h for more detail information.

![Software Layer Diagram](image)

**Figure 3.1 Software Layer Diagram**

3.2 Sample code (main program is in SampleApp.c)

The syntax of the sample code is very similar to the FT800 Programmers Guide. Please refer to the following figure for more detail. This is based on the code shown in section 2.5.3 of the FT800 Programmers Guide. Figure 3.2 shows how to draw points with varying radius from 5 pixels to 13 pixels with different colors.

```c
hal_gpi_dli(CLEAR(1,1)); // Clear the screen
hal_gpi_dli(COLOR_RGB(128, 0, 0)); // Set the draw color to Red
hal_gpi_dli(POINT_SIZE(5 * 16)); // Set size to 5 * 16 / 16 = 5 pixels
hal_gpi_dli(BEGIN(POINTS)); // Start the point draw
hal_gpi_dli(VERTEX2F(30 * 16, 17 * 16)); // Draw circle 30 pixels from left and 17 down
hal_gpi_dli(COLOR_RGB(0, 0, 128)); // Set the draw color to Blue
hal_gpi_dli(POINT_SIZE(10 * 16)); // Set size to 10 * 16 / 16 = 10 pixels
hal_gpi_dli(VERTEX2F(30 * 16, 51 * 16)); // Draw circle 30 pixels from left and 51 down
hal_gpi_dli(COLOR_RGB(128, 128, 0)); // Set the draw color to Yellow
hal_gpi_dli(POINT_SIZE(15 * 16)); // Set size to 15 * 16 / 16 = 15 pixels
hal_gpi_dli(VERTEX2F(98 * 16, 51 * 16)); // Draw circle 98 pixels from left and 51 down
hal_gpi_dli(DISPLAY()); // End the display list
hal_spi_wi8(REG_DLSHAP, DLSHAP_FRAME); // Make this display list active on the next frame
```

**Figure 3.2 The real drawing commands of the Sample Code**
3.3 FT_CoPro_Cmds.h

All available Co-Processor Engine commands are defined in FT_CoPro_Cmds.h

```c
ft_void hal_spi_cmd_text(ft_int16_t x, ft_int16_t y, ft_int16_t font, ft_uint16_t options, const ft_char_t* s);
ft_void hal_spi_cmd_number(ft_int16_t x, ft_int16_t y, ft_int16_t font, ft_uint16_t options, ft_int32_t n);
ft_void hal_spi_cmd_loadIdentity();
ft_void hal_spi_cmd_toggle(ft_int16_t x, ft_int16_t y, ft_int16_t w, ft_int16_t font, ft_uint16_t options, ft_uint16_t state, const ft_char_t* s);
ft_void hal_spi_cmd_gauge(ft_int16_t x, ft_int16_t y, ft_int16_t r, ft_uint16_t options, ft_uint16_t major, ft_uint16_t minor, ft_uint16_t val, ft_uint16_t range);
ft_void hal_spi_cmd_repread(ft_uint32_t ptr, ft_uint32_t result);
ft_void hal_spi_cmd_getprops(ft_uint32_t ptr, ft_uint32_t w, ft_uint32_t h);
ft_void hal_spi_cmd_memcpy(ft_uint32_t dest, ft_uint32_t src, ft_uint32_t num);
ft_void hal_spi_cmd_spacer(ft_int16_t x, ft_int16_t y, ft_uint16_t style, ft_uint16_t scale);
ft_void hal_spi_cmd_bgcolor(ft_uint32_t r);
ft_void hal_spi_cmd_swap();
ft_void hal_spi_cmd_inflate(ft_uint32_t ptr);
ft_void hal_spi_cmd_translate(ft_int32_t tx, ft_int32_t ty);
ft_void hal_spi_cmd_stop();
ft_void hal_spi_cmd_slider(ft_int16_t x, ft_int16_t y, ft_int16_t w, ft_int16_t h, ft_uint16_t options, ft_uint16_t val, ft_uint16_t range);
ft_void hal_spi_cmd_interrupt(ft_uint32_t ms);
ft_void hal_spi_cmd_gcolor(ft_uint32_t c);
ft_void hal_spi_cmd#region(ft_uint32_t ptr, ft_uint32_t num);
ft_void hal_spi_cmd_button(ft_int16_t x, ft_int16_t w, ft_int16_t h, ft_int16_t font, ft_uint16_t options, const ft_char_t* s);
ft_void hal_spi_cmd_memset(ft_uint32_t ptr, ft_uint32_t value, ft_uint32_t num);
ft_void hal_spi_cmd_setfont(ft_uint32_t font, ft_uint32_t ptr);
```

Figure 3.3 Parts of Co-Processor Engine commands
4 Running the Example

In order to run the example, the FTDI D2xx driver must first be installed. Then, the sample application can be built and run.

4.1 D2XX Linux Driver installation for Raspberry Pi

Download D2XX Linux ARM driver (Suitable for Raspberry Pi version 1.1.12 in this application note)

tar xvf libftd2xx1.1.12.tar.gz (decompress the D2XX driver)
su (switch to root user)
cd /release/build/arm926

cp libftd2xx.so.1.1.12 /usr/local/lib

ln -sf /usr/local/lib/libftd2xx.so.1.1.12 /usr/local/lib/libftd2xx.so (Creates a symbolic link to the shared object)

4.2 Run the FT800 Raspberry Sample Code

Download the FT800 Raspberry Pi sample code (FT800RPi.tar.gz)

tar xvf FT800RPi.tar.gz

cd /FT800 Raspberry Pi/Build/Linux/

make

rmmod ftdi_sio (remove VCP driver)

LD_LIBRARY_PATH=/usr/local/lib ./FT800RPi (run the application)
5 Test Results

The image below shows the display produced by the sample code. The resulting display is the same as that shown in section 2.5.3 of the FT800 Programmers Guide.

![Test image displayed on screen](image)

**Figure 5.1 Test image displayed on screen**
6 Conclusion

This application note has demonstrated the way in which a Single Board Computer can be interfaced to the FT800. There are many cases where a Single Board Computer may be used within a product and may require a small display with the possibility of touch and sound functionality, and the FT800 provides a good solution for this.

The example uses the Raspberry Pi due to its wide availability and range of resources. The OS images are available from the Raspberry Pi website and debugging can be carried out without additional JTAG interfaces etc. using the monitor attached to the Raspberry Pi. However, the FT800 can be interfaced to many other Single Board Computers.
7 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 8797 1330
Fax: +868 (0) 2 8751 9737

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.

System and equipment manufacturers and designers are responsible to ensure that their systems, and any Future Technology Devices International Ltd (FTDI) devices incorporated in their systems, meet all applicable safety, regulatory and system-level performance requirements. All application-related information in this document (including application descriptions, suggested FTDI devices and other materials) is provided for reference only. While FTDI has taken care to assure it is accurate, this information is subject to customer confirmation, and FTDI disclaims all liability for system designs and for any applications assistance provided by FTDI. Use of FTDI devices in life support and/or safety applications is entirely at the user's risk, and the user agrees to defend, indemnify and hold harmless FTDI from any and all damages, claims, suits or expense resulting from such use. This document is subject to change without notice. No freedom to use patents or other intellectual property rights is implied by the publication of this document. Neither the whole nor any part of the information contained in, or the product described in this document, may be adapted or reproduced in any material or electronic form without the prior written consent of the copyright holder. Future Technology Devices International Ltd, Unit 1, 2 Seaward Place, Centurion Business Park, Glasgow G41 1HH, United Kingdom. Scotland Registered Company Number: SC136640
Appendix A – References

Document References

EVE Product Page
FT800 Datasheet
FT800 Programmers Guide
VM800BU Datasheet
AN_240 EVE From the Ground Up
EVE SampleApp
Raspberry Pi Download
Project source code

Acronyms and Abbreviations

<table>
<thead>
<tr>
<th>Terms</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIDORA</td>
<td>Pidora is a Fedora Remix optimized for the Raspberry Pi computer.</td>
</tr>
<tr>
<td>SBC</td>
<td>A single-board computer (SBC) is a complete computer built on a single circuit board.</td>
</tr>
<tr>
<td>SPI</td>
<td>Serial Peripheral Interface</td>
</tr>
<tr>
<td>TF Card</td>
<td>A TF card stands for a Trans Flash card.</td>
</tr>
<tr>
<td>USB</td>
<td>Universal Serial Bus</td>
</tr>
</tbody>
</table>
Appendix B – List of Figures

List of Figures

Figure 2.1 Hardware Block Diagram.................................................................3
Figure 3.1 Software Layer Diagram.................................................................4
Figure 3.2 The real drawing commands of the Sample Code.................................4
Figure 3.3 Parts of Co-Processor Engine commands........................................5
Figure 5.1 Test image displayed on screen.......................................................7
Appendix C – Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Changes</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>Initial release</td>
<td>2015-02-09</td>
</tr>
<tr>
<td>1.1</td>
<td>Updated the broken link in Section 1 Introduction</td>
<td>2015-11-17</td>
</tr>
</tbody>
</table>