Application Notes

AN_390

FT80x to FT81x Migration Guide

Version 1.0

Issue Date: 2015-09-30

This document highlights the enhanced features in the FT81x Series of graphics controller ICs and the changes required to migrate from FT80x to FT81x.

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

Building on the success of the first generation of FT80x Embedded Video Engine (EVE) devices with integrated touch and audio functions, the series has now been extended to include a further four devices which offer a plethora of improvements and extended capabilities, the FT81x series. The FT810, FT811, FT812 and FT813 provide increased pixel resolution for sharper imagery, portrait orientation capabilities, increased speed for faster data transfer and image/video loading, and larger memory capacity.

2 FT81x vs FT80x - Features Comparison

This section lists some of the enhanced features available in the FT81x compared to the FT80x series.

- Maximum screen resolution 800x600 pixels (increased from 512 x 512 on FT80x)
- Colour signalling options of 18-bit (262K Colours) or 24-bit RGB (1.6 Million)
- Object memory 1Mbyte (increased from 256kByte on FT80x)
- MCU transfer with SPI or Dual SPI or QSPI interface option
- Available in QFN-48 / QFN-56 package (RoHS compliant)
- Extra-large ROM fonts added
- Orientation switching support added with REG ROTATE and CMD SETROTATE (allows Portrait operation)
- Master clock frequency improvement - REG PCLK can now be 1
- Smooth playback - motion JPG encoded AVI video playback
- Multiple 16/32 bit colour palettes supported with transparency
- L2 format supported for efficient DXT1-style bitmaps
- CMD MEDIAFIFO specifies an area of main memory to use as a FIFO for JPG, AVI and PNG loading
- Multiple numeric formats supported including binary, octal, decimal and hex (FT80x was decimal only)
- Simplified font loading with CMD SETFONT2
- 500-1000 times faster JPG loading
- Pixel clock drawing increased from 4 to 16 pixels per clock
- Quicker Firmware memory operations (e.g. copy, fill, CRC) due to tuned inner loops
- CMD SNAPSHOT hundreds of times faster than FT80x
- Auxiliary registers for cmd fifo for ease of command buffer management.

3 Application Migration from FT80x to FT81x

3.1 Memory map addresses for RAM/ROM/Registers

The register map has changed between the FT80x and FT81x to accommodate the extra object RAM and new registers. The Memory Map addresses for FT80x vs FT81x are given in Table 1 - FT81x Memory map

and Table 2 - FT80x Memory Map

. For details of the full register map, refer to the programmers guide for the FT80x and FT81x families.

Header files containing the register address definitions can also be found within the project zip files for the sample projects. These can be included in other software projects to provide definitions for the register map. Please refer to the EVE2 compatible samples on the project samples web page, denoted by (+EVE2).

http://www.ftdichip.com/Support/SoftwareExamples/FT800_Projects.htm

Note that the EVE2 compatible headers have both FT80x and FT81x definitions, with a #define used to select the required set.

Start Address	End Address	Size	NAME	Description
00 0000h	0F FFFFh	1024 kB	RAM_G	General purpose graphics RAM
0C0000h	0C0003h	4 B	ROM_CHIPID	FTDI chip identification and revision information: 0C0000h: 08h 0C0001h: 10h (FT810), 11h(FT811), 12h(FT812), 13h(FT813) 0C0002h: 01h 0C0003h: 00h
1E 0000h	2F FFFBh	1152 kB	ROM_FONT	Font table and bitmap
2F FFFCh	2F FFFFh	4 B	ROM_FONT_ADDR	Font table pointer address
30 0000h	30 1FFFh	8 kB	RAM_DL	Display List RAM
30 2000h	30 2FFFh	4 kB	RAM_REG	Registers
30 8000 h	30 8FFFh	4 kB	RAM_CMD	Co-processor command circular buffer

Notes: The addresses beyond this table are reserved and shall not be read or written unless otherwise specified.

Palette now uses an area within RAM_G instead of the dedicated area in FT80x.

ROM_CHIPID is within the RAM_G memory. These are the reset values. The values can be overwritten and so it is recommended to read them before the application writes to these locations.

Table 1 - FT81x Memory map

Start Address	End Address	Size	NAME	Description
00 0000h	03 FFFFh	256 kB	RAM_G	Main graphics RAM
0C 0000h	0C 0003h	4 B	ROM_CHIPID	FT800/801 chip identification and revision information: 0C0000h: 08h 0C0001h: 00h (FT800) 01h (FT801) 0C0002h: 01h 0C0003h: 00h
0B B23Ch	0F FFFBh	275 kB	ROM_FONT	Font table and bitmap
0F FFFCh	0F FFFFh	4 B	ROM_FONT_ADDR	Font table pointer address
10 0000h	10 1FFFh	8 kB	RAM_DL	Display List RAM
10 2000h	10 23FFh	1 kB	RAM_PAL	Palette RAM
10 2400h	10 257Fh	380 B	REG_*	Registers
10 8000 h	10 8FFFh	4 kB	RAM_CMD	Graphics Engine Command Buffer
1C 2000 h	1C 27FFh	2 kB	RAM_SCREENSHOT	Screenshot readout buffer

Note: The addresses beyond this table are reserved and shall not be read or written unless otherwise specified.

Table 2 - FT80x Memory Map

3.2 SPI packet structure

Since the FT81x supports SPI packet structures for dual and quad SPI modes, an extra dummy byte needs to be added. Single channels for the FT81x are similar to the FT80x. For SPI memory read transactions, the host sends two zero bits, followed by the 22-bit address. This is followed by a dummy byte. After the dummy byte, the FT81x responds to each host byte with read data bytes.

3.3 Display settings parameters

Display scan-out parameters may need to be changed to meet the LCD panel requirements. Please refer to the guide below that explains the factors to consider when selecting a display panel and configuring the FT8xx registers accordingly.

Selecting an LCD Display

3.4 Default system clock

The default system clock for the FT80x is 48MHz, and for the FT81x it is 60MHz

By default the FT81x system clock is 60MHz when the input clock is 12MHz. The host is allowed to switch the system clock to other frequencies (48MHz, 36MHz, and 24MHz) via the host command "CLKSEL". The clock switching command shall be sent in SLEEP mode only.

3.5 Extension instructions for GPU primitives

The maximum resolution for FT80x is 512 x 512 pixels, whereas for FT81x the maximum resolution is enhanced to 800 x 600 pixels. Additional registers with a _H notation accommodate the upper bits of the values (e.g. the higher bits of the stride / width / height).

Like the FT80x, the graphics engine in the FT81x takes the instructions from display list memory RAM_DL in the form of commands. Each command is 4 bytes long and one display list can be filled with up to 2048 commands as the size of RAM_DL is 8K bytes. The graphics engine performs the respective operation according to the definition of commands.

The following set of commands allows setting of the Graphics State –

BITMAP_SIZE_H

Specify the 2 most significant bits of bitmaps dimension for the current handle.

Encoding

31 24	23 4	3 2	1 0
0x29	reserved	width	height

Parameters

Width: 2 most significant bits of bitmap width. The initial value is zero.

Height: 2 most significant bits of bitmap height. The initial value is zero.

Description

This command is the extension command of BITMAP_SIZE for bitmap larger than 511 x 511 pixels.

Graphics context

None

BITMAP_SIZE_H

Specify the 2 most significant bits of the source bitmap memory format and layout for the current handle.

Encoding

31 24	23 4	3 2	1 0
0x28	reserved	linestride	height

Parameters

Linestride: The 2 most significant bits of the 12-bit linestride parameter value specified to BITMAP_LAYOUT.

Height: The 2 most significant bits of the 11-bit height parameter value specified to BITMAP_LAYOUT.

Description

This command is an extension of BITMAP_LAYOUT for large bitmaps. This command is not needed if the specified linestride parameter value to BITMAP_LAYOUT is less than 1024 and the height parameter value is less than 512.

3.6 PALETTED8 Bitmap

The FT80x supports Paletted bitmaps. The new FT81x uses the Paletted8 format instead, which requires some additional display list items in order to display the Paletted8 bitmap in colour. The format can be set as follows:

BITMAP_LAYOUT

Encoding

31 24	23	22	21	20	19	18 9	8 0
0x07	format					linestride	height

The table below shows the format options for the FT81x including PALETTED8.

Name	Value	Bits/pixel	Alpha bits	Red bits	Green bits	Blue bits
ARGB1555	0	16	1	5	5	5
L1	1	1	1	0	0	0
L4	2	4	4	0	0	0
L8	3	8	8	0	0	0
RGB332	4	8	0	3	3	2
ARGB2	5	8	2	2	2	2
ARGB4	6	16	4	4	4	4
RGB565	7	16	0	5	6	5
TEXT8X8	9	-	-	-	-	-
TEXTVGA	10	-	-	-	-	-
BARGRAPH	11	-	-	-	-	-
PALETTED565	14	8	0	5	6	5
PALETTED4444	15	8	4	4	4	4
PALETTED8	16	8	8	8	8	8
L2	17	2	2	0	0	0

Table 3 - BITMAP_LAYOUT Format List

An example of the PALETTED8 format is shown below

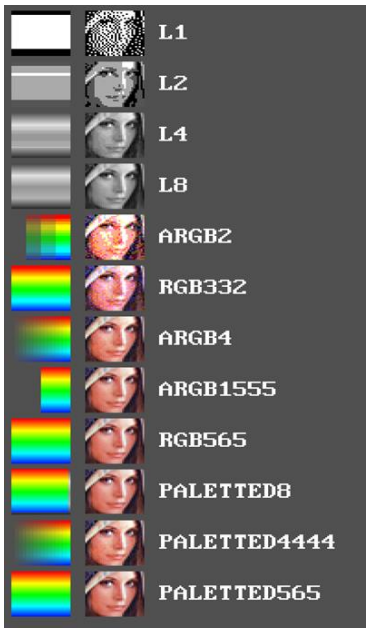

Figure 1 - PALLETED8 Format

4 FTDI HAL based Migration - FT80x to FT81x

The migration recommendations mentioned in Section 3 are intended for users who are developing their own source files.

The FTDI website has a range of examples which have been updated to support the FT81x in addition to the original support for FT80x. Please refer to the samples with the (+EVE2) note.

http://www.ftdichip.com/Support/SoftwareExamples/FT800_Projects.htm

These examples use the #defines FT_80X_ENABLE and FT_81X_ENABLE in order to support both families in the same source files. These defines are used throughout the main application code and supporting c and header files.

To simplify the selection, these defines can be controlled from a single file called platform.h which is present in each of the demo code projects. This file contains a high-level set of defines to select the type of demo board, screen size and host platform. These in turn enable the associated defines throughout the rest of the code.

Please refer to the sample projects from the link above and also the accompanying platform guide for further details.

EVE Platform Guide

5 Recommendations for better performance

The following sections show some of the ways in which the FT81x's features can be used to improve performance and/or reduce workload on the host MCU.

5.1 Auxiliary registers for command FIFO

To offload work from the MCU for checking the free space in the circular buffer, the FT81x offers two auxiliary registers "**REG_CMDB_SPACE**" and "**REG_CMDB_WRITE**" for bulk transfers. It enables the MCU to write commands and data to the co-processor in a bulk transfer, without computing the free space in the circular buffer and increasing the address. As long as the amount of data to be transferred is less than the value in the register "**REG_CMDB_SPACE**", the MCU is able to safely write all the data to "**REG_CMDB_WRITE**" in one write transfer.

5.2 Media FIFO for image decompression

The CMD_MEDIAFIFO command is used to set up a streaming media FIFO in RAM_G.

C prototype

```
void cmd_mediafifo ( uint32_t ptr,
                     uint32_t size );
```

Parameters

Ptr: starting address of memory block

size: number of bytes in the source memory block

Command layout

+0	CMD_MEDIAFIFO (0xffffff39)
+4	Ptr
+8	Size

Examples

To set up a 64-Kbyte FIFO at the top of RAM_G for JPEG streaming, and report the initial values of the read and write pointers:

```
cmd_mediafifo(0x100000 - 65536, 65536);//0x100000 is the top of RAM_G
printf("R=%08xW=%08x\n",rd32(REG_MEDIAFIFO_READ),rd32(REG_MEDIAFIFO_WRITE));
```

prints:

000f000 00f000

5.3 VERTEX_FORMAT for pixel / subpixel vertex instructions

The FT81x has support for configuring the vertex format. The purpose of using vertex_format is to set the precision of VERTEX2F coordinates. Precisions of 1, 1/2, 1/4, 1/8 and 1/16 can be selected.

Encoding

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16 15 13 12 11 10 9 8 7 6 5 4 3	2 1 0
0x27	RESERVED	Frac

Parameters

Frac: Number of fractional bits in X, Y coordinates. Valid range is from 0 to 4. The initial value is 4.

Description

VERTEX2F uses 15 bit signed numbers for its (X, Y) coordinates. This command controls the interpretation of these numbers by specifying the number of fractional bits.

By varying the format, an application can trade range against precision.

frac	Units in pixel precision	VERTEX2F range in pixels
0	1	-16384 to 16383
1	1/2	-8192 to 8191
2	1/4	-4096 to 4095
3	1/8	-2048 to 2047
4	1/16	-1024 to 1023

Table 4 - VERTEX_FORMAT and pixel precision

Graphics context

The value of **frac** is part of the graphics context

5.4 VERTEX_TRANSLATE_X and VERTEX_TRANSLATE_Y

The VERTEX_TRANSLATE commands are added to ease the movement of a set of objects. These commands allow translation of whole sets of primitives/widgets.

The VERTEX_TRANSLATE_X is used to specify the vertex transformations X translation. The command is structured as follows:

Encoding

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17	16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
0x2B	RESERVED	x

Parameters

X: signed x-coordinate in 1/16 pixel. The initial value is 0

Description

Specifies the offset added to vertex X coordinates. This command allows drawing to be shifted on the screen.

Graphics context

The value of x is part of the graphics context.

The VERTEX_TRANSLATE_Y is used to specify the vertex transformation's Y translation. The command is structured as follows:

Encoding

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17	16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
0x2C	RESERVED	y

Parameters

y: signed y-coordinate in 1/16 pixel. The initial value is 0

Description

Specifies the offset added to vertex Y coordinates. This command allows drawing to be shifted on the screen.

Graphics context

The value of *y* is part of the graphics context

5.5 CMD_SETFONT2 to facilitate easy calculation of address

In FT80x, the CMD_SETFONT was used to register one custom defined bitmap font into the co-processor engine. In FT81x, CMD_SETFONT2, a new co-processor command, is added to setup a custom font. To use a custom font with the co-processor objects, create the font definition in RAM_G and issue CMD_SETFONT2, as described in section 5.5 of the FT81x Programmers Guide.

C prototype

```
void cmd_setfont2( uint32_t font,
                   uint32_t ptr,
                   uint32_t firstchar );
```

Command layout

+0	CMD_SETFONT2(0xffffff3b)
+4	Font
+8	Ptr
+12	firstchar

Parameters

Font: The bitmap handle from 0 to 31

Ptr: 32 bit aligned memory address in RAM_G of font metrics block

Firstchar: The ASCII value of the first character in the font.

Examples

With a suitable font metrics block loaded in RAM_G at address 100000, first character's ASCII value 32, to use it for font 20:


```
cmd_setfont2(20, 100000, 32);
cmd_button(15, 30, 130, 20, 18, 0, "This is font 18");
cmd_button(15, 60, 130, 20, 20, 0, "This is font 20");
```

Table 5 - CMD_SETFONT2 Example

5.6 CMD_SETBITMAP co-processor command

This command facilitates easy construction of a bitmap using a set of GPU instructions.

It will generate the corresponding display list commands (BITMAP_SOURCE \ BITMAP_LAYOUT \ BITMAP_SIZE) for a given bitmap information, sparing the effort of writing the display list manually.

The parameters filter / wrapx / wrapy in BITAMP_SIZE is always set to NEAREST / BORDER / BORDER value in the generated display list commands.

C prototype

```
void cmd_setbitmap( uint32_t   addr,
                    uint16_t   fmt,
                    uint16_t   width,
                    uint16_t   height  );
```

Parameters

Addr: Address of bitmap data in RAM_G.

Fmt: Bitmap format, see the definition in BITMAP LAYOUT.

Width: bitmap width, in pixels.

Height:bitmap height, in pixels

Command layout

+0	CMD_SETBITMAP(0xffff ff43)
+4	Addr
+8	Fmt
+10	Width
+12	Height

Examples

NA

Note: Two bytes need to be appended after last parameter to provide 4 bytes alignment as required by the co-processor

5.7 DXT1 Usage

The FT81x series provides support for the L2 bitmap format. If an FT80x application currently uses DXT1, it could use the L2 bitmap format feature instead of the two L1 bitmaps supported in the FT80x series.

6 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

FT81x Series Programmer's Guide

FT800 Series Programmer's Guide

FT81x EVE Datasheet

FT800 EVE Datasheet

Acronyms and Abbreviations

Terms	Description
API	Application Programming Interface
AVI	Audio Video Interleave
FIFO	First In First Out
GPU	Graphics Processor Unit
HSYNC	Horizontal Synchronization
IO	Input / Output
JPG	Joint Photographic Group
LCD	Liquid Crystal Display
MCU	Micro Controller Unit
MPU	Memory Protection Unit
PNG	Portable Network Graphics
QFN	Quad Flat No leads
QSPI	Quad Serial Peripheral Interface
RAM	Random Access Memory
RoHS	Restriction of Hazardous Substances Directive
ROM	Read only Memory
VSYNC	Vertical Synchronization

Appendix B – List of Tables & Figures

List of Tables

List of Figures

Appendix C – Revision History

Document Title: AN_390 FT80x to FT81x Migration Guide

Document Reference No.: FT_001246

Clearance No.: FTDI#471

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

Document Feedback: Send Feedback

Revision	Changes	Date
1.0	Initial Release	2015-09-30