Application Note

AN_270

FT_App_Sketch

Version 1.3

Issue Date: 2018-01-03

This application note introduces the Sketch Demo Application. The objective of the Demo Application is to enable users to become familiar with the usage of the FT8XX, the design flow, and display list used to design the desired user interface or visual effect.

Use of Bridgetek devices in life support and/or safety applications is entirely at the user's risk, and the user agrees to defend, indemnify and hold Bridgetek harmless from any and all damages, claims, suits or expense resulting from such use.

Bridgetek Pte Ltd (BRTChip)
178 Paya Lebar Road, #07-03, Singapore 409030
Tel: +65 6547 4827 Fax: +65 6841 6071
Web Site: http://www.brtchip.com

Table of Contents

1 Introduction

This design example demonstrates an interactive user interface that provides a drawing, or sketch, area. The touch screen is used to capture user writing inside of the sketch area. The touch events are then drawn within the area interactively. While the sketch area is active, a slider bar is available to change the background colour of the drawing area. A touch button that clears the sketch area is the final element on the screen.

1.1 Overview

The document will provide information on drawing graphics elements through primitives, tagging of touch capabilities and the structure of display lists. In addition, this application note outlines the general steps of the system design flow, display list creation and integrating the display list with the system host microcontroller.

This application note should be read in conjunction with the source code found at:
http://brtchip.com/SoftwareExamples-eve/

1.2 Scope

This document can be used as a guide by designers to develop GUI applications by using FT80x with any MCU via SPI. Note that detailed documentation is available on http://brtchip.com/eve/.

2 Display Requirements

This section describes some of the key components of the design.

2.1 Sketch Area

The Sketch area uses the display and touch screen. As the touch screen is written upon, the corresponding pixels in the sketch area are rendered out to simulate drawing on a page. The sketch area only covers most of the screen. Touch events outside of this area are ignored, except for the button and slider features.

2.2 Slider

A second touch area uses a FT8XX slider widget. This allows a user to change the background colour of what is drawn in the sketch area. It also changes the colour of the slider itself to match.

2.3 Button

A third and final touch area of the screen is assigned to a button widget with the text “CLR”. Tapping this button will generate a touch event called a tag. Upon sensing the tag, the sketch area of the screen is cleared, ready for more drawing.

3 Design Flow

Every EVE design follows the same basic principles as highlighted in Figure 3.1.

Select and configure your host port for controlling the FT8XX then wake the device before configuring the display. The creative part then revolves around the generation of the display list, "** APPLICATION DATA **" in the figure below. There will be two lists. The active list and the updated/next list are continually swapped to render the display. Note, header files map the pseudo code of the design file of the display list to the FT8XX instruction set, which is sent as the data of the SPI packet (typically <1KB). As a result, with EVE's object oriented approach, the FT8XX is operating as an SPI peripheral while providing full display, audio, and touch capabilities.

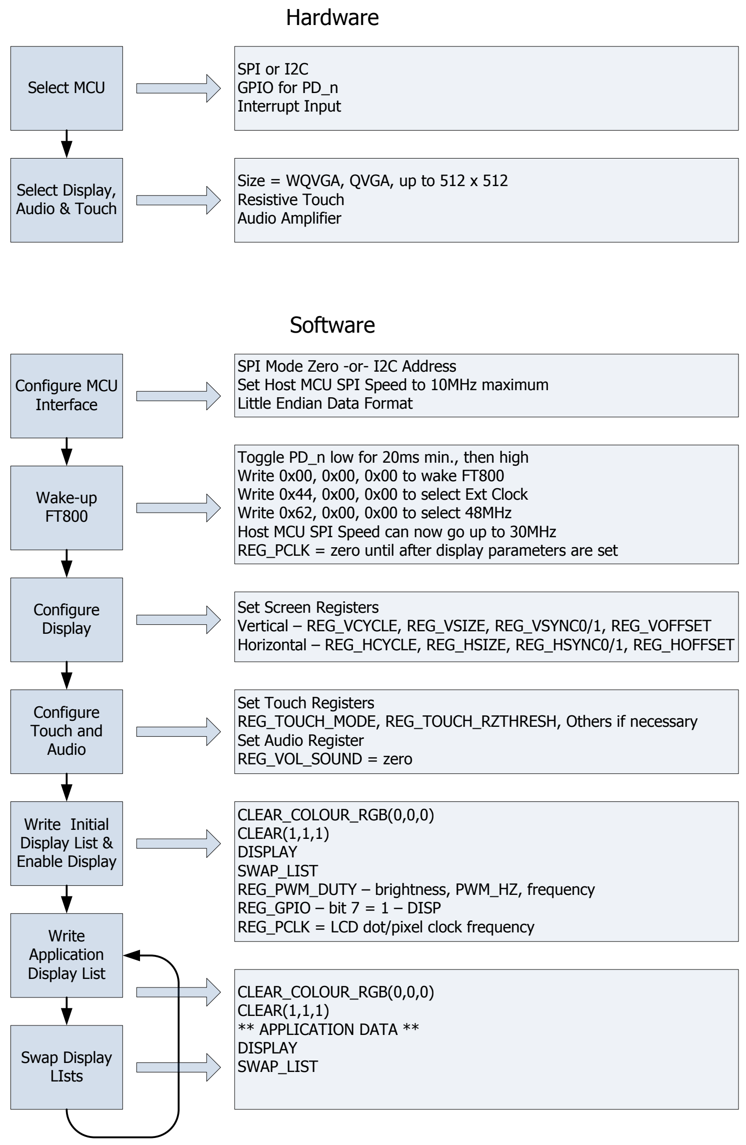

Hardware
Select MCU
SPI or I2C
GPIO for PD_n
Interrupt Input
Select Display, Audio & Touch
Size = WQVGA, QVGA, up to 512 x 512
Resistive Touch
Audio Amplifier
Software
Configure MCU Interface
SPI Mode Zero -or- I2C Address
Set Host MCU SPI Speed to 10MHz maximum
Little Endian Data Format
Wake-up FT800
Toggle PD_n low for 20ms min., then high
Write 0x00, 0x00, 0x00 to wake FT800
Write 0x44, 0x00, 0x00 to select Ext Clock
Write 0x62, 0x00, 0x00 to select 48MHz
Host MCU SPI Speed can now go up to 30MHz
REG_PCLK = zero until after display parameters are set
Configure Display
Set Screen Registers
Vertical – REG_VCYCLE, REG_VSIZE, REG_VSYNC0/1, REG_VOFFSET
Horizontal – REG_HCYCLE, REG_HSIZE, REG_HSYNC0/1, REG_HOFFSET
Configure Touch and Audio
Set Touch Registers
REG_TOUCH_MODE, REG_TOUCH_RZTHRESH, Others if necessary
Set Audio Register
REG_VOL_SOUND = zero
Write Initial Display List & Enable Display
CLEAR_COLOUR_RGB(0,0,0)
CLEAR(1,1,1)
DISPLAY
SWAP_LIST
REG_PWM_DUTY – brightness, PWM_HZ, frequency
REG_GPIO – bit 7 = 1 – DISP
REG_PCLK = LCD dot/pixel clock frequency
Write Application Display List
Swap Display LIsts
CLEAR_COLOUR_RGB(0,0,0)
CLEAR(1,1,1)
** APPLICATION DATA **
DISPLAY
SWAP_LIST

Figure 3.1 Generic EVE Design Flow

3.1 Signature Flowchart

The flowchart below is specific to the Sketch application.

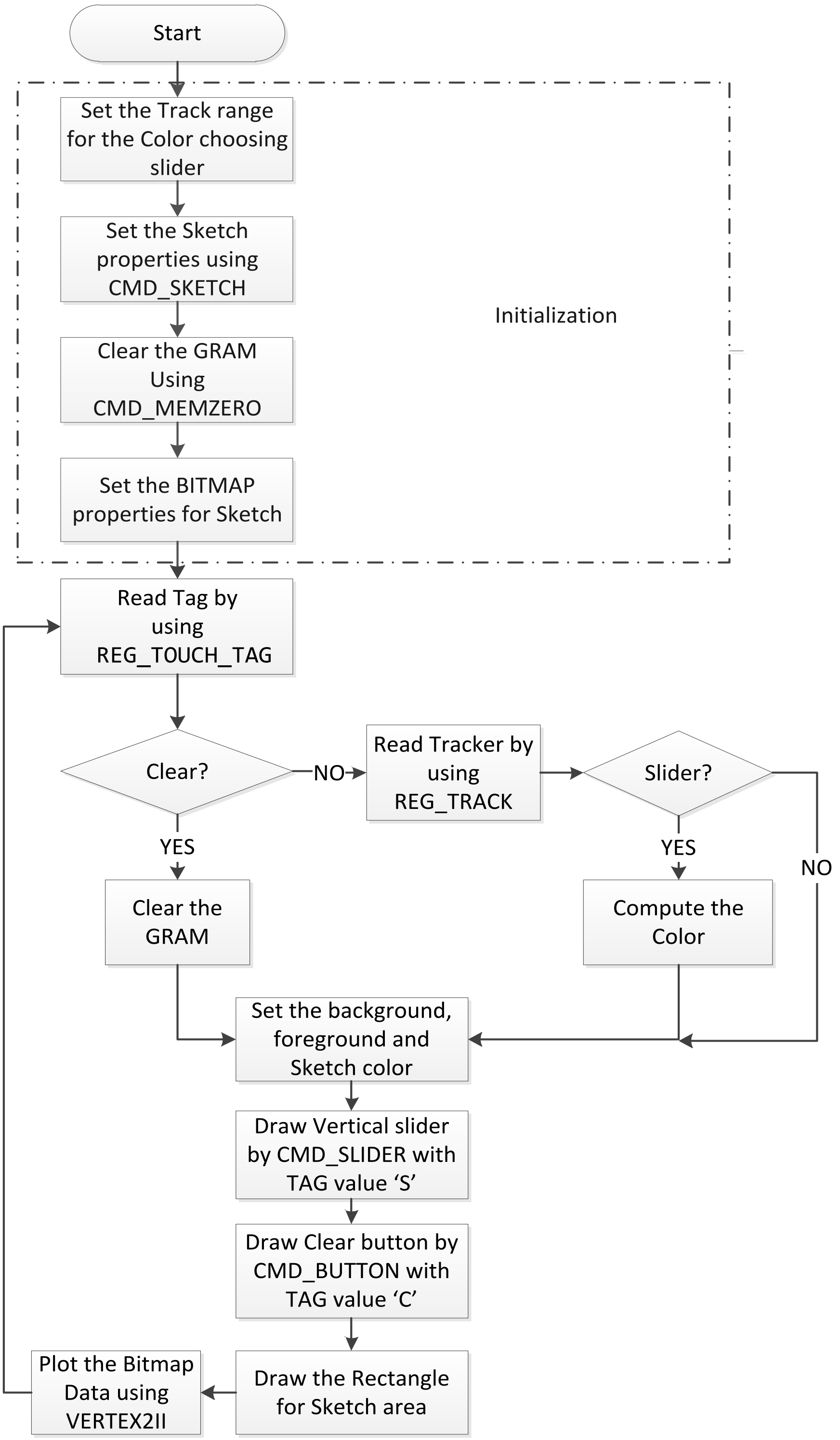

Start
Set the Track range for the Color choosing slider
Set the Sketch properties using CMD_SKETCH
Clear the GRAM Using CMD_MEMZERO
Set the BITMAP properties for Sketch
Initialization
Read Tag by using REG_TOUCH_TAG
Clear?
NO
Read Tracker by using REG_TRACK
Slider?
YES
Clear the GRAM
YES
Compute the Color
NO
Set the background, foreground and Sketch color
Draw Vertical slider by CMD_SLIDER with TAG value 'S'
Draw Clear button by CMD_BUTTON with TAG value 'C'
Draw the Rectangle for Sketch area
Plot the Bitmap Data using VERTEX2II

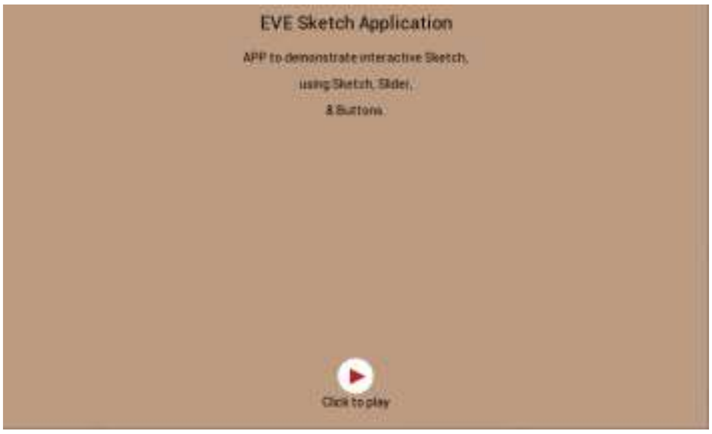
Figure 3.2 Flowchart

4 Function Blocks Description

4.1 Application Start Screen

Refer to AN_391 EVE Platform Guide for information pertaining to platform setup and the necessary development environment.

Upon setting up the platform, the application start screen is displayed.



EVE Sketch Application

APP to demonstrate interactive Sketch,
using Sketch, Slider,
& Buttons

Figure 4.1 Start screen

4.2 Sketch()

An initial display list is created to configure a few items:

```
    uint32_t  tracker,color=0;
    uint16_t  val=32768;
    uint8_t   tag =0;
```

Set the foreground color to white:

```
       Gpu_CoCmd_FgColor(phost,0xffffff);        // Set the bg color
```

Define the area used by the slider bar. This touch area is just to the right of the sketch area, but not all the way to the right edge. This area is tracked so the slider can be moved:

```
       Gpu_CoCmd_Track(phost,(DispWidth-30),40,8,DispHeight-100,1);
```

Start the CMD_SKETCH. This is a large rectangle from 0,0 to 40 pixels in from the right and 30 pixels up from the bottom:

```
#if defined FT801_ENABLE
    Gpu_CoCmd_CSketch(phost,0,10,DispWidth-40,DispHeight-30,0,L8,1500L);
#elif defined FT81X_ENABLE
    Gpu_CoCmd_Sketch(phost,0,10,DispWidth-40,DispHeight-30,0,L8);
#else
    Gpu_CoCmd_Sketch(phost,0,10,DispWidth-40,DispHeight-30,0,L8);
#endif
```

Zero out the GRAM area associated with the sketch area. This clears out any artifacts:

```
Gpu_CoCmd_MemZero(phost,0L,(DispWidth-40)*(DispHeight-20L));
App_WrCoCmd_Buffer(phost,BITMAP_HANDLE(1));
App_WrCoCmd_Buffer(phost,BITMAP_SOURCE(0));
App_WrCoCmd_Buffer(phost,BITMAP_LAYOUT(L8,DispWidth-40,DispHeight-20));
#ifdef FT81X_ENABLE
App_WrCoCmd_Buffer(phost,BITMAP_LAYOUT_H((DispWidth-40)>>10,(DispHeight-20)>>9));
#endif
App_WrCoCmd_Buffer(phost,BITMAP_SIZE(NEAREST,BORDER,BORDER,(DispWidth-40),(DispHeight-
20)));
#ifdef FT81X_ENABLE
App_WrCoCmd_Buffer(phost,BITMAP_SIZE_H((DispWidth-40)>>9,(DispHeight-20)>>9));
#endif
```

Show this display list and clear out the buffers for the next one:

```
    Gpu_CoCmd_Swap(phost);
    App_Flush_Co_Buffer(phost);
    Gpu_Hal_WaitCmdfifo_empty(phost);
```

At this point, the CMD_SKETCH command continually translates the X-Y touch coordinates within the sketch area and changes those locations in the GRAM from the foreground colour to background colour. By doing this, any drawing done on the touchscreen in the sketch area is shown on the screen. Once complete, the GRAM corresponding to the sketch area can be read by the MCU and stored as an image. This example does not store any sketch data.

With the visible display initialization complete, the screen elements can now be drawn. A while() loop will create a new display list each time though. The display list checks for any colour changes as a result of moving the slide and whether to clear the sketch area if the button is tapped.

The clear button and slider bar are assigned touch tags. Tags remove the need to manually correlate the X-Y touch coordinates with a drawn element. Instead, the entire element is assigned a tag number. The FT8XX determines whether the touch event was inside the element boundaries then assigns the appropriate tag. For the slider, the tag is also assigned with the tracker above. The FT8XX automatically sets a value corresponding to the location of the slider. The value is used to assign the background colour. These tags free up considerable host MCU time.

The first item in the while() loop is to see if there are any touch events and whether a tag is assigned. If the CLEAR button is tapped, the signature area is cleared by resetting the GRAM to the original zero values:

```
       // Check the tracker
       tracker = Gpu_Hal_Rd32(phost,REG_TRACKER);
       // Check the Tag
       tag = Gpu_Hal_Rd8(phost,REG_TOUCH_TAG);
       //  clear the GRAM when user enter the Clear button
       if(tag==2)
       {
           Gpu_CoCmd_Dlstart(phost);
              //Clear the gram frm 1024
           Gpu_CoCmd_MemZero(phost,0,(DispWidth-40)*(DispHeight-20L));
           App_Flush_Co_Buffer(phost);
           Gpu_Hal_WaitCmdfifo_empty(phost);
       }
```

If the tag for the slider bar is read, the tracker value is read and a new background colour is assigned:

```
       if((tracker&0xff)==1)      // check the tag val
       {
           val = (tracker>>16);
       }
```

The display list is then started, buffers flushed, foreground colour set to white and background colour set to correspond to the slider (or an initial value for the first time through):

```
        // Start the new display list
        Gpu_CoCmd_Dlstart(phost);                  // Start the display list
        App_WrCoCmd_Buffer(phost,CLEAR(1,1,1));    // clear the display
        App_WrCoCmd_Buffer(phost,COLOR_RGB(255,255,255));  // color
        Gpu_CoCmd_BgColor(phost,color);
```

Next, a tag is assigned to the slider and the slider is drawn:

```
        App_WrCoCmd_Buffer(phost,TAG_MASK(1));
        App_WrCoCmd_Buffer(phost,TAG(1));          // assign the tag value
        Gpu_CoCmd_FgColor(phost,color);
        // draw the sliders
        Gpu_CoCmd_Slider(phost,(DispWidth-30),40,8,(DispHeight-100),0,val,65535);
```

Now another tag is assigned to the "CLR" button and it is drawn:

```
        Gpu_CoCmd_FgColor(phost,(tag==2)?0x0000ff:color);
        App_WrCoCmd_Buffer(phost,TAG(2));          // assign the tag value
        Gpu_CoCmd_Button(phost,(DispWidth-35),(DispHeight-45),35,25,26,0,"CLR");
        App_WrCoCmd_Buffer(phost,TAG_MASK(0));
```

The last item to draw is the word "Color" above the slider bar:

```
    Gpu_CoCmd_Text(phost,DispWidth-35,10,26,0,"Color");
```

With everything drawn, the display list now identifies the sketch area and changes the background colour according to the value from the slider:

```
    App_WrCoCmd_Buffer(phost,LINE_WIDTH(1*16));
    App_WrCoCmd_Buffer(phost,BEGIN(RECTS));
    App_WrCoCmd_Buffer(phost,VERTEX2F(0,10*16));
    App_WrCoCmd_Buffer(phost,VERTEX2F( (DispWidth-40)*16, (DispHeight-20)*16));

App_WrCoCmd_Buffer(phost,COLOR_RGB((color>>16)&0xff,(color>>8)&0xff,(color)&0xff));
    App_WrCoCmd_Buffer(phost,BEGIN(BITMAPS));
    App_WrCoCmd_Buffer(phost,VERTEX2II(0,10,1,0));
    App_WrCoCmd_Buffer(phost,END());
```

The final section of the while() loop is to swap the display list to make it active, flush the buffers and return to the top of the loop:

```
        App_WrCoCmd_Buffer(phost,DISPLAY());
        Gpu_CoCmd_Swap(phost);
        App_Flush_Co_Buffer(phost);
        Gpu_Hal_WaitCmdfifo_empty(phost);
```

4.2.1 FT801 CMD_CSKETCH

The FT801 capacitive display uses a different (capacitive) display touch controller compared to the FT8XX resistive controller which has a slower sample rate than the resistive display. To ensure smooth interaction between the user touch and the application, CMD_SKETCH is replaced with CMD_CSKETCH. To enable this alternative command in the sample program open the Platform.h file and look for:

```
#define FT801_ENABLE
```

By default this is undefined (FT8XX mode). To switch in the CMD_CSKETCH ensure this line is defined. After making the change, rebuild and run the application.

4.3 Functionality

The function has three user interactive elements:

- Draw in the sketch area
- Clear the sketch using clear Button
- Change the sketch colour by using colour slider.

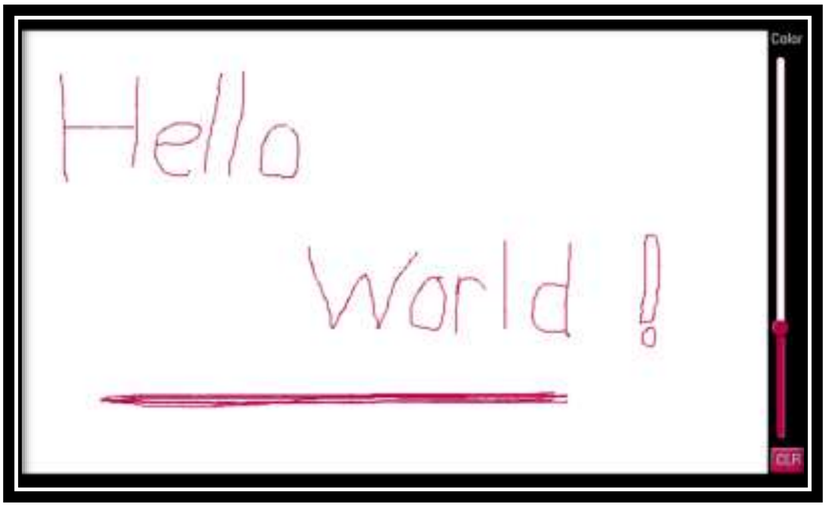


Figure 4.2 SketchDisplay

5 Contact Information

Head Quarters – Singapore

Bridgetek Pte Ltd
178 Paya Lebar Road, #07-03
Singapore 409030
Tel: +65 6547 4827
Fax: +65 6841 6071

E-mail (Sales)	sales.apac@brtchip.com
E-mail (Support)	support.apac@brtchip.com

Branch Office – Taipei, Taiwan

Bridgetek Pte Ltd, Taiwan Branch
2 Floor, No. 516, Sec. 1, Nei Hu Road, Nei Hu District
Taipei 114
Taiwan , R.O.C.
Tel: +886 (2) 8797 5691
Fax: +886 (2) 8751 9737

E-mail (Sales)	sales.apac@brtchip.com
E-mail (Support)	support.apac@brtchip.com

Branch Office - Glasgow, United Kingdom

Bridgetek Pte. Ltd.
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)	sales.emea@brtichip.com
E-mail (Support)	support.emea@brtchip.com

Branch Office – Vietnam

Bridgetek VietNam Company Limited
Lutaco Tower Building, 5th Floor, 173A Nguyen Van Troi,
Ward 11, Phu Nhuan District,
Ho Chi Minh City, Vietnam
Tel : 08 38453222
Fax : 08 38455222

E-mail (Sales)	sales.apac@brtchip.com
E-mail (Support)	support.apac@brtchip.com

Web Site

http://brtchip.com/

Distributor and Sales Representatives

Please visit the Sales Network page of the Bridgetek 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 Bridgetek Pte Ltd (BRTChip) 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 Bridgetek devices and other materials) is provided for reference only. While Bridgetek has taken care to assure it is accurate, this information is subject to customer confirmation, and Bridgetek disclaims all liability for system designs and for any applications assistance provided by Bridgetek. Use of Bridgetek 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 Bridgetek 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. Bridgetek Pte Ltd, 178 Paya Lebar Road, #07-03, Singapore 409030. Singapore Registered Company Number: 201542387H.

Appendix A– References

Document References

- FT800 datasheet
- FT801 datasheet
- Programming Guide covering EVE Command Language
- AN_391 EVE Platform Guide
- AN_240 FT800 from the Ground Up
- AN_245 VM800CB_SampleApp_PC_Introduction - covering detailed design flow with a PC and USB to SPI bridge cable
- AN_246 VM800CB_SampleApp_Arduino_Introduction – covering detailed design flow in an Arduino platform
- VM800C datasheet
- VM800B datasheet

Acronyms and Abbreviations

Terms	Description
Arduino Pro	The open source platform variety based on ATMEL's ATMEGA chipset
EVE	Embedded Video Engine
SPI	Serial Peripheral Interface
UI	User Interface
USB	Universal Serial Bus

Appendix B – List of Tables & Figures

List of Figures

List of Tables

NA

Appendix C– Revision History

Document Title: AN_270 FT_App_Sketch
Document Reference No.: BRT_000190
Clearance No.: BRT#109
Product Page: http://brtchip.com/product/
Document Feedback: Send Feedback

Revision	Changes	Date
1.0	Initial Release	2013-08-21
1.1	Updated Release	2013-11-01
1.2	Added section 4.3.1	2014-06-30
1.3	Document migrated from FTDI to BRT (Updated company logo; copyright info; contact information; hyperlinks)	2018-01-03