This application note is to introduce the Metaballs Demo Application running on both MSVC, Arduino and FT9XX. 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.
Table of Contents

1 Introduction ............................................................................. 2
  1.1 Overview ............................................................................. 2
  1.2 Scope .................................................................................. 2
2 Design Flow ............................................................................. 3
  2.1 Initialization ........................................................................ 3
  2.2 Application Flow ................................................................. 4
3 Description .............................................................................. 5
  3.1 Application Start Screen ..................................................... 5
  3.2 Functionality ........................................................................ 5
4 Running the Demonstration Code ............................................. 7
5 Contact Information ................................................................. 11
Appendix A– References ........................................................... 12
  Document References ............................................................. 12
  Acronyms and Abbreviations ................................................ 12
Appendix B – List of Figures & Tables ........................................ 13
  List of Figures ....................................................................... 13
  List of Tables ......................................................................... 13
Appendix C – Revision History .................................................. 14
1 Introduction

This application demonstrates the usage of graphics primitives such as points and bitmaps.

In this application, 80 random balls and three background fireball are drawn. The fireball is generated by additive colour blending 80 Random balls of different sizes are generated with the primitives "POINTS".

The MCU continuously monitors the touch. If the user touches the screen, the base of the ball origin is changed.

![Figure 1 - Metaball Snapshot](image)

1.1 Overview

The application will be useful to understand the FT8XX Primitives Points, Blend function, and Bitmap. Also the application will be useful to understand the concept of additive blending.

This example demonstrates that in addition to providing an attractive graphical user interface for an application, the FT8xx's tracking features can be used to allow this to be manipulated interactively by the user.

1.2 Scope

This document can be used by designers to develop GUI applications by using the FT8xx with an SPI host. In this case, a PC running Visual Studio (C++) with a C232HM cable is used as the SPI master. Arduino, FT8XX emulator and FT9XX platforms are also supported.

It covers the following topics:
- Brief overview of the Metaball demonstration
- Application code flow including FT8xx initialization and Metaball code
- Description of the Metaball function within the application
- Running the demonstration code

Additional documentation can be found at [http://brtchip.com/eve/](http://brtchip.com/eve/).
## 2 Design Flow

### 2.1 Initialization

Every EVE design follows the same basic principles as highlighted in Figure 2.1. After configuring the SPI Host itself (such as the PC through the C232HM cable, or an MCU), the application will wake up the FT8XX and write to the registers in the FT8XX to configure.

**Figure 2 - Generic EVE Design Flow**
2.2 Application Flow

**Figure 3 – Application Flow Chart**
3 Description

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

3.1 Application Start Screen

Upon setting up the platform, the start-up screen will be displayed.

![Application Start Screen](image)

**Figure 4 – Application Start Screen**

3.2 Functionality

This application starts up with an animation of moving metaballs. When user drags finger on the screen, the velocity of the balls will be increased. And the center of the moving area is the touch point of the screen.

While no touching is detected, the center is

\[
\text{centerx} = \frac{\text{DispWidth} \times 16}{2}; \\
\text{centery} = \frac{\text{DispHeight} \times 16}{2};
\]

If touching is detected, then

\[
\text{sx} = \text{Gpu_Hal_Rd16}(\text{phost},\text{REG_TOUCH_SCREEN_XY} + 2); \\
\text{sy} = \text{Gpu_Hal_Rd16}(\text{phost},\text{REG_TOUCH_SCREEN_XY}); \\
\text{centerx} = 16 \times \text{sx}; \\
\text{centery} = 16 \times \text{sy};
\]

The velocity is 4 times faster when touch is detected.

\[
\text{VEL} = \text{touching} ? 8 : 2;
\]
Here is the code to control the ball velocity and moving range.

```c
for (tval=0; tval<numBlobs; ++tval)
{
    if (blobs[tval].x < centerx)   blobs[tval].dx += VEL;
    else   blobs[tval].dx -= VEL;
    if (blobs[tval].y < centery)   blobs[tval].dy += VEL;
    else   blobs[tval].dy -= VEL;
    blobs[tval].x += blobs[tval].dx << 3;
    blobs[tval].y += blobs[tval].dy << 3;
}

blobs[random(numBlobs)].dx = v();
blobs[random(numBlobs)].dy = v();
for (tval1 = 0; tval1 < h; tval1++)
{
    for (tval2 = 0; tval2 < w; tval2++)
    {
        m = fadein;
        for (tval = 0; tval < 3; tval++)
        {
            bx = blobs[tval].x >> 8;
            by = blobs[tval].y >> 8;
            dx = bx - tval2;
            dy = by - tval1;
            d = SQ(dx) + SQ(dy);
            m += recip[min(d >> 2, recipsz - 1)];
        }
    }
}
```
4 Running the Demonstration Code

This example is shown when running on a PC with Visual Studio (C++) installed. The FT8XX
development module (VM800B/VM800C) is connected to the PC using the C232HM cable which
acts as a USB to SPI converter.

<table>
<thead>
<tr>
<th>Pin</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCK</td>
<td>ORANGE</td>
</tr>
<tr>
<td>MOSI</td>
<td>YELLOW</td>
</tr>
<tr>
<td>MISO</td>
<td>GREEN</td>
</tr>
<tr>
<td>CS#</td>
<td>BROWN</td>
</tr>
<tr>
<td>PD#</td>
<td>BLUE</td>
</tr>
<tr>
<td>GND</td>
<td>BLACK</td>
</tr>
</tbody>
</table>

Table 1 - CM232H Connections to the VM800 Pins

The code can now be compiled and run. The debug button can be used to start the application.

Alternatively the FT9XX platform is also supported for this application. Upon installing the
Toolchain “Eclipse for FT90X”, users can import the project into the workspace. Please refer to
figure 8.
The code can be compiled and generate a `.bin` file, which can be downloaded to the FT900 board with a UMFTPD2A module. Please refer to figure 9 given below -

Figure 6 - Eclipse for FT90X Screenshot

Figure 7 - Connections between MM900EV2A & UMFTPD2A Modules
FT900Prog GUI Launcher is a UI program used for downloading the program to the FT90X board; it is installed together with the FT90X Tool chain. Please see figure 10 given below -

![FT900Prog GUI Launcher Screenshot](image)

**Figure 8 - FT900Prog GUI Launcher Screenshot**

FT900 board (ex. MM900 board) acts as a shield and it can be easily connected to an FT8xx module. Please see figure 11 given below -

![Connections between FT900 & FT8xx Modules](image)

**Figure 9 - Connections between FT900 & FT8xx Modules**

When running the application, the calibration screen will be displayed first. This uses the FT8XX’s built-in calibration routine. It ensures that the FT8XX can align inputs from the touch panel to the image on the screen below accurately. The routine will display a dot and ask the user to tap on this dot. It will then repeat this twice more (with the dot at a different location on the screen in each case).
The Bridgetek logo animation will then appear on the screen (not shown here).

The Metaball introduction screen is then displayed and the application waits for the ‘Click to play’ button to be pressed, before loading the Metaball screen.

The main Metaball screen will now be displayed. Initially, the ball will move at normal speed, and faster when touch is detected. The moving range of the balls also changed.

This application is also supported on an Arduino platform. Users can select different modules in "Platform.h". After that open "App_Metaball.ino" with the Arduino IDE located at "App_Metaball\Project\Arduino\App_Metaball", try to compile the code and upload the program to the Arduino board. Make sure the right COM port and Arduino board is selected before uploading.
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@brtchip.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
- 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
- AN_281 FT800_Emulator_Library_User_Guide - Covering API interface for FT800 Emulator.
- AN_252 FT800 Audio Primer

Acronyms and Abbreviations

<table>
<thead>
<tr>
<th>Terms</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arduino Pro</td>
<td>The open source platform variety based on ATMEAL’s ATMEGA chipset</td>
</tr>
<tr>
<td>EVE</td>
<td>Embedded Video Engine</td>
</tr>
<tr>
<td>SPI</td>
<td>Serial Peripheral Interface</td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
</tr>
<tr>
<td>USB</td>
<td>Universal Serial Bus</td>
</tr>
</tbody>
</table>
Appendix B – List of Figures & Tables

List of Figures
Figure 1 - Metaball Snapshot ................................................................. 2
Figure 2 - Generic EVE Design Flow ...................................................... 3
Figure 3 - Application Flow Chart .......................................................... 4
Figure 4 - Application Start Screen ....................................................... 5
Figure 5 - Visual Studio Screenshot ....................................................... 7
Figure 6 - Eclipse for FT90X Screenshot ................................................. 8
Figure 7 - Connections between MM900EV2A & UMFTP2A Modules .......... 8
Figure 8 - FT900Prog GUI Launcher Screenshot ..................................... 9
Figure 9 - Connections between FT900 & FT8xx Modules ........................ 9
Figure 10 - Calibration Screen .............................................................. 10
Figure 11 - Metaball Snapshot with & without touch ............................... 10

List of Tables
Table 1 - CM232H Connections to the VM800 Pins ............................... 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&lt;br&gt;Document migrated from FTDI to BRT (Updated company logo; copyright info; contact information; hyperlinks)</td>
<td>2018-01-04</td>
</tr>
</tbody>
</table>