Application Note

BRT_AN_018

FT90x Camera to EVE

Version 1.0

Issue Date: 2017-09-11

This Application Note describes an application where a camera on an FT90x connects to an FT81x display to show the output of the camera and a graphical overlay.

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 3
1.1 Overview 3
1.2 Scope 3
1.2.1 Features 4
1.2.2 Possible Enhancements 4
2 Project Overview 5
2.1 Sources Folder 5
2.2 Eve Folder 5
2.1 Images Folder 5
3 Using the Camera to EVE 7
3.1 Required Hardware 7
3.2 Use of Application Note Software 7
3.3 Camera Display 8
4 Contact Information 10
Appendix A– References 11
Document References 11
Acronyms and Abbreviations 12
Appendix B – List of Tables & Figures 13
List of Tables 13
List of Figures 13
Appendix C– Revision History 14

1 Introduction

This Application Note describes a display showing video obtained from a camera module with an FT90x device and an FT81x. The display also shows a graphical overlay for the camera output to resemble a car reversing camera. There is no need to connect the device to a host computer, apart from programming the FT90x, as the application runs standalone.

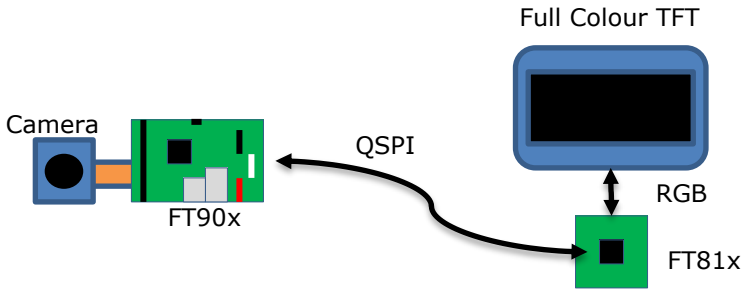

Figure 1 - Block Diagram

The camera is attached to the FT90x device. The FT8xx device controls the TFT display.

The document should be read in association with the example code provided in the references section.

1.1 Overview

This document describes the design and implementation of the FT90x Camera to EVE code. The FT90x Camera to EVE allows a user to:

- Display JPEG images from a camera on a display.
- Draw a graphical overlay on the video.
- Displays statistical text on the display.

This document is intended to demonstrate the capabilities of the FT90x family of microcontrollers and interfacing to FT8xx display ICs. It specifically shows drawing overlays on top of JPEG images on the FT81x devices.

Third-party open source code is used to implement this application note:

- Printf – tinyprintf.

Links to resources for these libraries are in Appendix A – References.

1.2 Scope

The program shows how to interface with an Omnivision OV5640 camera and obtain a stream of JPEG images (MJPEG). Only camera modules that support MJPEG output are allowed. The OV5640 module is used in the *CleO* camera module, but requires modification to the MM900EVxA boards to function.

The FT90x program detects a JPEG image in the data stream and controls an EVE display on the FT90x's QSPI interface.

Decoding the JPEG image is performed by the FT8xx IC on the EVE module. A series of lines is then drawn on top of the JPEG image. These can be used to show significant fixed points on the image obtained.

1.2.1 Features

This application note shows how to interface with an FT8xx family display IC and send JPEG images and other graphical elements to the display.

1.2.2 Possible Enhancements

This application note can be seen as a start for customisation or extension. Some example enhancements could be:

- More features for image display e.g. overlaid controls for instrumentation.
- Touchscreen controls for adjusting the camera settings.
- Higher resolutions or lower frame rates.
- Other camera modules.

2 Project Overview

The project files for the application are divided into the following folders.

Folder	Description
Source	Application source code and abstraction files.
Includes	Application specific header files.
Images	JPEG images for this application.
eve	EVE API libraries.
eve_arch_ft9xx	EVE FT9xx architecture specific code.
lib	Library files.
lib\tinyprintf	tinyprintf library.

Table 1 - Project Files Overview

2.1 Sources Folder

The main part of the application is found in the "Sources" folder. This is split into 3 main sections and has 3 source code files.

- The "main.c" file is generally responsible for the FT90x setup, receiving data from the camera, detecting JPEG images in the data stream;
- The "camera.c" file abstracts the camera device allowing different camera modules to be used;
- A camera abstraction file "ov5640_camera.c" to interface with the supported camera module;
- The "eve.c" file performs all FT81x operations and is responsible for displaying the JPEG images and drawing graphical overlays.

The other files in this folder are:

- "images.c" contains JPEG images encoded into C array declarations. These images are displayed by the FT8xx display.
- "crt0.S" a modified startup file (in FT90x assembly language) to allow the application to write to a protected section of FlashROM on the device.

Files in these folders use the "Includes" folder for application specific header files.

2.2 Eve Folder

This folder holds the FT81x API code which abstracts the FT81x register and processing list writes into C functions. A separate folder "eve_arch_ft9xx" contains FT9xx specific code for the library.

2.1 Images Folder

The FT81x can display JPEG images. This folder contains the raw JPEG images which are encoded into the "images.c" file in the "Sources" folder.

The method for converting these to C code is as follows. Use the HexEdit utility to open the JPEG file, Select All, select Edit -> Copy As… -> C Source. The C code can be pasted into a source file to be used in the application.

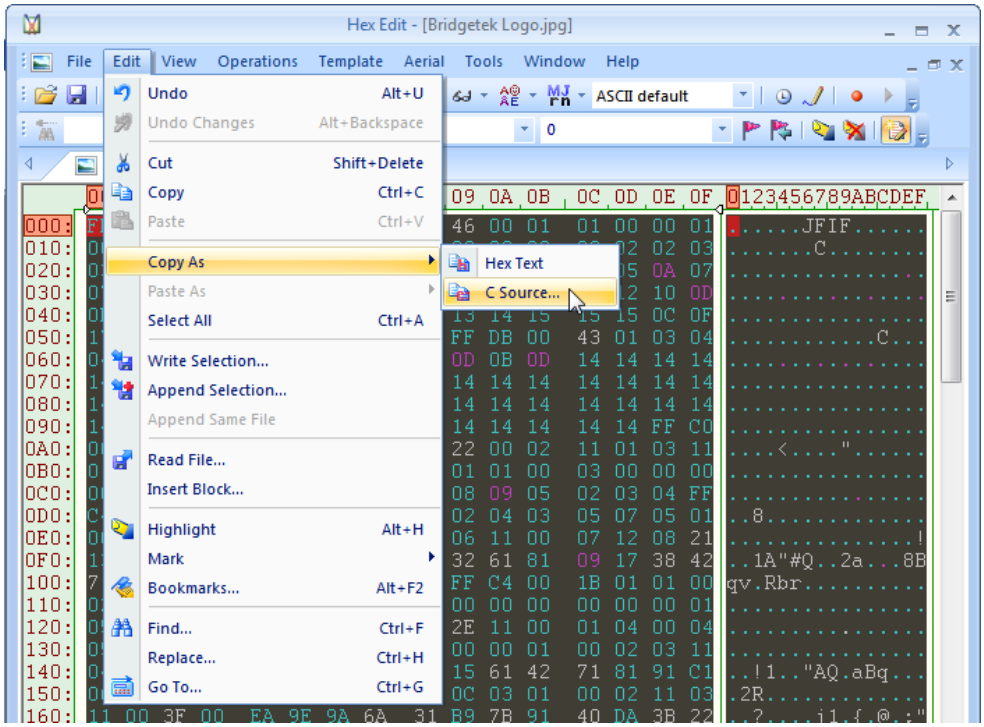

Figure 2 - HexEdit Copy As C Source

If the code is declared as "**const uint8_t __flash__**" then it will reside in Program Memory and will not be copied to RAM.

3 Using the Camera to EVE

3.1 Required Hardware

This application note is intended to be used on an ME812A-WH50R or ME813A-WH50C EVE development module with an MM900EVxA module. This application note shows how to work on an 800 by 480 display. Changes to the EVE module or FT90x device can be made in the "EVE_config.h" file in the "eve" folder of the source code.

The MM900EVxA module connects directly to the EVE development module with a set of pin headers. The QSPI interface on the FT90x device is taken through the pin headers to the FT81x on the EVE module. The host PC connects via USB to the MM900EVxA module.

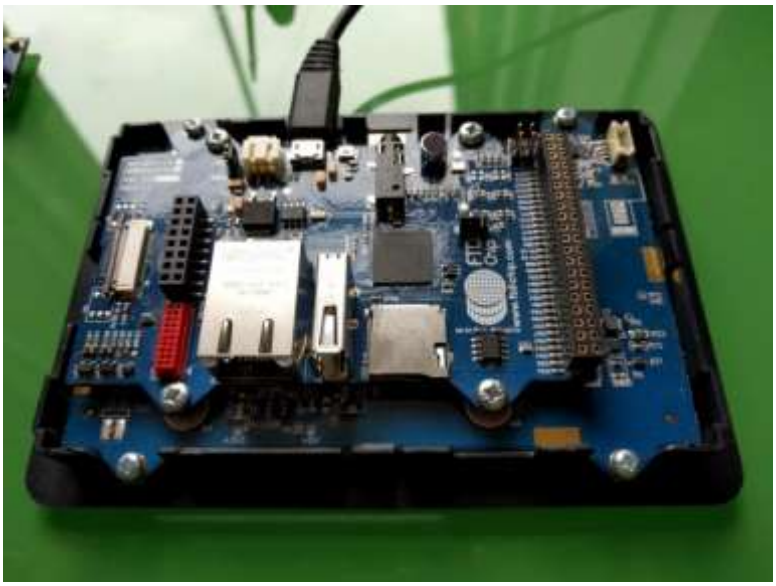

Figure 3 - MM900EV2A and ME812A-WH50R module

3.2 Use of Application Note Software

The Camera to EVE application will wait until it receives data from the camera. The FT81x display will indicate this with the Bridgetek logo and the caption "Waiting for images".

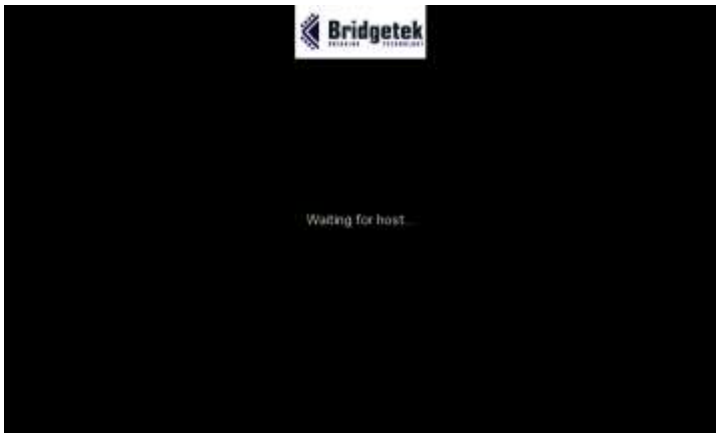

Figure 4 - Waiting for Host Screen

3.3 Camera Display

A sample output of the EVE display is shown in Figure 5.

Figure 5 – EVE Display with Camera Video

The overlay shown is the red lines projecting a box.

The camera information is received as a stream of data from the camera. This is expanded into an in-memory bitmap by the LOADIMAGE command.

```
EVE_LIB_BeginCoProList();
EVE_CMD_DLSTART();
EVE_BITMAP_HANDLE(CUSTOM_BITMAP2);
EVE_CMD_LOADIMAGE(image_start_address, OPT_NODL);
```

As the data is streamed in, a display list is created for the expansion of the JPEG. This display list sends the LOADIMAGE command and directs the expanded bitmap into a convenient memory address of the FT81x. Each time data is received from the camera, it is sent to the FT81x using the SPI bus.

```
EVE_LIB_WriteDataToCMD(buffer, length);
```

When the end of the JPEG image is detected, then the display list is finished and the code waits for the GPU to complete.

```
EVE_LIB_EndCoProList();
```

It is then able to draw the JPEG image and the overlay.

The FT81x commands to display a 640x480 JPEG image in the centre of the screen are as follows:

```
EVE_LIB_BeginCoProList();
EVE_CMD_DLSTART();
EVE_CLEAR(1, 1, 1);
EVE_BITMAP_HANDLE(CUSTOM_BITMAP2);
EVE_BEGIN(BITMAPS);
EVE_BITMAP_SOURCE(image_start_address);
// Tell GPU the size of the received image (can be done with GETPROPS)
EVE_BITMAP_LAYOUT(RGB565, IMAGE_WIDTH * 2, IMAGE_HEIGHT);
EVE_BITMAP_LAYOUT_H((IMAGE_WIDTH * 2) >> 10, IMAGE_HEIGHT >> 9);
EVE_BITMAP_SIZE(NEAREST, BORDER, BORDER, IMAGE_WIDTH, IMAGE_HEIGHT);
EVE_BITMAP_SIZE_H(IMAGE_WIDTH >> 9, IMAGE_HEIGHT >> 9);
// Display image centred on screen
EVE_VERTEX2II(LEFT_X, TOP_Y, CUSTOM_BITMAP2, 0);
EVE_END();
```

The values of IMAGE_WIDTH and IMAGE_HEIGHT are 640 and 480 respectively. The relevant code is in the `eve_loop()` function in the file "eve.c". The code is slightly different in the source code as it has code to support different sizes of images.

Once the JPEG image has been added to the display list, then the overlays may be added. This is achieved with the following code:

```
EVE_BEGIN(LINES);
EVE_COLOR_RGB(128, 0, 0);
EVE_LINE_WIDTH(2 * 16);
// Draw overlay lines
EVE_VERTEX2F(LEFT_X * 16, BOTTOM_Y * 16);
EVE_VERTEX2F((LEFT_X + (IMAGE_WIDTH / 3)) * 16, (BOTTOM_Y / 2) * 16);
EVE_VERTEX2F((LEFT_X + (IMAGE_WIDTH / 3)) * 16, (BOTTOM_Y / 2) * 16) ;
EVE_VERTEX2F((LEFT_X + (IMAGE_WIDTH * 2 / 3)) * 16, (BOTTOM_Y / 2) * 16);
EVE_VERTEX2F((LEFT_X + (IMAGE_WIDTH * 2 / 3)) * 16, (BOTTOM_Y / 2) * 16);
EVE_VERTEX2F((LEFT_X + (IMAGE_WIDTH)) * 16, BOTTOM_Y * 16);
```

The LEFT_X and BOTTOM_Y macros indicate a reference point on the bottom left of the image displayed on the screen. The value of IMAGE_WIDTH is the width of the image. Once the overlay has been added, then the DISPLAY and SWAP commands can be sent to render the image and overlay on the screen.

```
EVE_DISPLAY();
EVE_CMD_SWAP();
EVE_LIB_EndCoProList();
```

4 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 1330
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.

Appendix A– References

Document References

FT90x Product Page

FT93x Product Page

FT900/901/902/903 Datasheet

FT905/906/907/908 Datasheet

FT81x Product Page

FT81x Datasheet

FT9xx Development Modules

ME81x Modules

MM900EVxA datasheet

AN_324 FT9xx User Manual

AN_365 FT9xx API Programmers Manual

CleO camera module

tinyprintf

HexEdit

BRT_AN_018 FT90x Camera to EVE Source Code

Acronyms and Abbreviations

Terms	Description
API	Application Programming Interface
EVE	Embedded Video Engine
HID	Human Interface Device (Keyboard, Mouse etc)
IC	Integrated Circuit
JPEG	Joint Photographic Experts Group
MCU	Microcontroller Unit
QSPI	Quad Serial Peripheral Interface
RAM	Random Access Memory
TTF	Thin Film Transistor
USBD	Universal Serial Bus Device

Appendix B – List of Tables & Figures

List of Tables

Table 1 - Project Files Overview 5

List of Figures

Figure 1 - Block Diagram 3
Figure 2 - HexEdit Copy As C Source 6
Figure 3 - MM900EV2A and ME812A-WH50R module 7
Figure 4 - Waiting for Host Screen 7
Figure 5 – EVE Display with Camera Video 8

Appendix C– Revision History

Document Title: FT90x Camera to EVE
Document Reference No.: BRT_000169
Clearance No.: BRT#091
Product Page: http://brtchip.com/i-ft9/
Document Feedback: Send Feedback

Revision	Changes	Date
1.0	Initial Release	2017-09-11