



Application Note

AN_417

ME81XA Sample Application MM9XX Introduction

Version 1.1

Issue Date: 2018-01-08

This application note is provided to introduce the setup of Sample Applications running on a Windows PC with a ME81XA development system. The objective of the Sample Application is to enable users to become familiar with the usage of the FT81X, the design flow, and display list language 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>

Copyright © Bridgetek Pte Ltd

Table of Contents

1	Introduction	3
1.1	Audience	3
1.2	Scope	3
1.3	Overview	4
1.3.1	Hardware	4
1.3.2	Application Flow	5
1.3.3	Architecture	6
1.4	Hardware Requirements	6
1.5	Software requirements	6
1.5.1	Software Package Introduction	7
2	Setup	8
2.1	Hardware Connection	8
2.2	Source Code Build	9
2.2.1	Determine the Configuration for the Hardware Being Used	9
2.2.2	Determine the Group of Functions to Build and Run	10
2.2.3	Building and Programming the Sample Application Project	10
2.2.4	Experimenting with the Sample Application Project	10
2.3	Source File Brief	11
2.3.1	Major Function Groups in the Sample Application	12
3	Helpful Hints	15
4	Contact Information	16
	Appendix A– References	17
	Document References	17
	Acronyms and Abbreviations	17
	Appendix B – List of Figures & Tables	18
	List of Figures	18
	List of Tables	18
	Appendix C– Revision History	19

1 Introduction

FT81X combines display, audio and touch functionality into one single chip, powered by Bridgetek's advanced EVE technology, Embedded Video Engine. The FT81X device interfaces with a system MCU via an SPI/QSPI interface. To help customers easily utilize the functionality of the FT81X in a project, a Sample Application is provided here for tutorial purposes.

The Sample Application has been written for usage on an ME81XA module in conjunction with MM900EVxA module where the FT900 IC on the MM900EVXA module acts as host processor. A UMFTPD2A module is used for programming the FT900. The current sample applications run on FT900 and provide stimulus to ME81XA modules.

Users can view the source code of the Sample Application first, and then run the code to observe the operation. Editing the code is also encouraged to help learn the features of the FT81X.

Note that although the basic project compiles for the FT900 module, most of the code could be re-used in different microcontroller design environments. All that the FT81X requires is a host microcontroller device with an SPI master interface and the transportation layer of Sample Application adapted for the specific device.

Further information regarding the FT81X programming language or pseudo-code can be found in the [FT8XX Series Programmer Guide](#). This document introduces how to set up and use the Sample Application with FTDI's MM900EV1A and ME81xA development kits.

For ME81XA development boards, please refer the details in below table.

Part Number	Controller	Display Size	Touch
ME810A-HV35R	FT810Q	3.5 inch	Resistive
ME812A-WH50R	FT812Q	5.0 inch	Resistive
ME813A-WH50C	FT813Q	5.0 inch	Capacitive

Table 1-1 ME81XA development modules

Note: Any source code is provided on an "as is" basis, and is neither guaranteed nor supported.

1.1 Audience

This document assumes the audience has read the Datasheet and Programmer Guide of the FT81X. In addition, familiarity of the C/C++ programming language is necessary to understand the Sample Application source code.

1.2 Scope

The Sample Application mentioned in this document runs on a Windows PC, through Eclipse IDE provided with the [FT9XX Toolchain](#). It comprises of the source code as well as project files.

1.3 Overview

1.3.1 Hardware

Figure 1-1 illustrates the overall hardware setup.

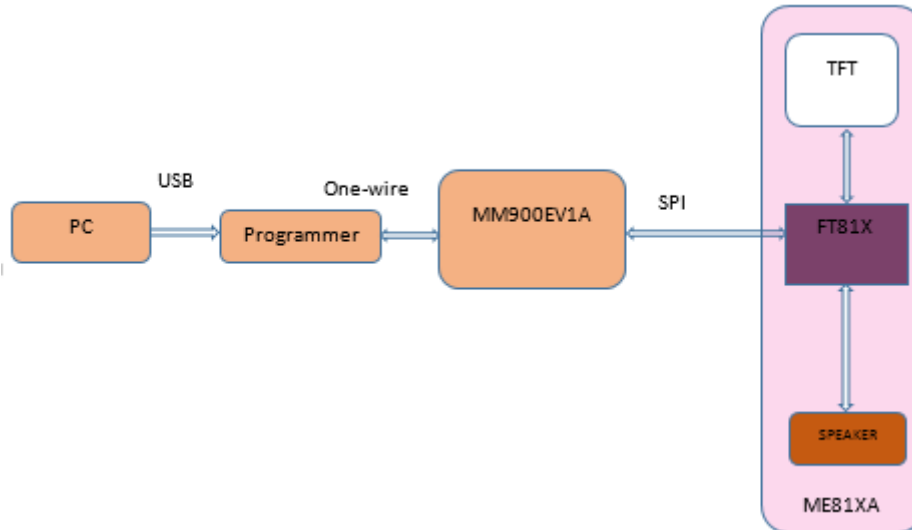


Figure 1-1 Example Block Diagram of Setup for ME81XA board with Windows PC

The ME81XA units require minimal hardware configuration. The FT81X IC, the TFT display, and the speaker are all included in the module assembly. The FT810 / FT812 provide an interface for resistive touch panels with FT811/FT813 providing access to capacitive touch panels. More about this is mentioned in the [FT8XX datasheet](#).

1.3.2 Application Flow

Figure 1-2 gives the basic flow to configure the FT81X in an application.

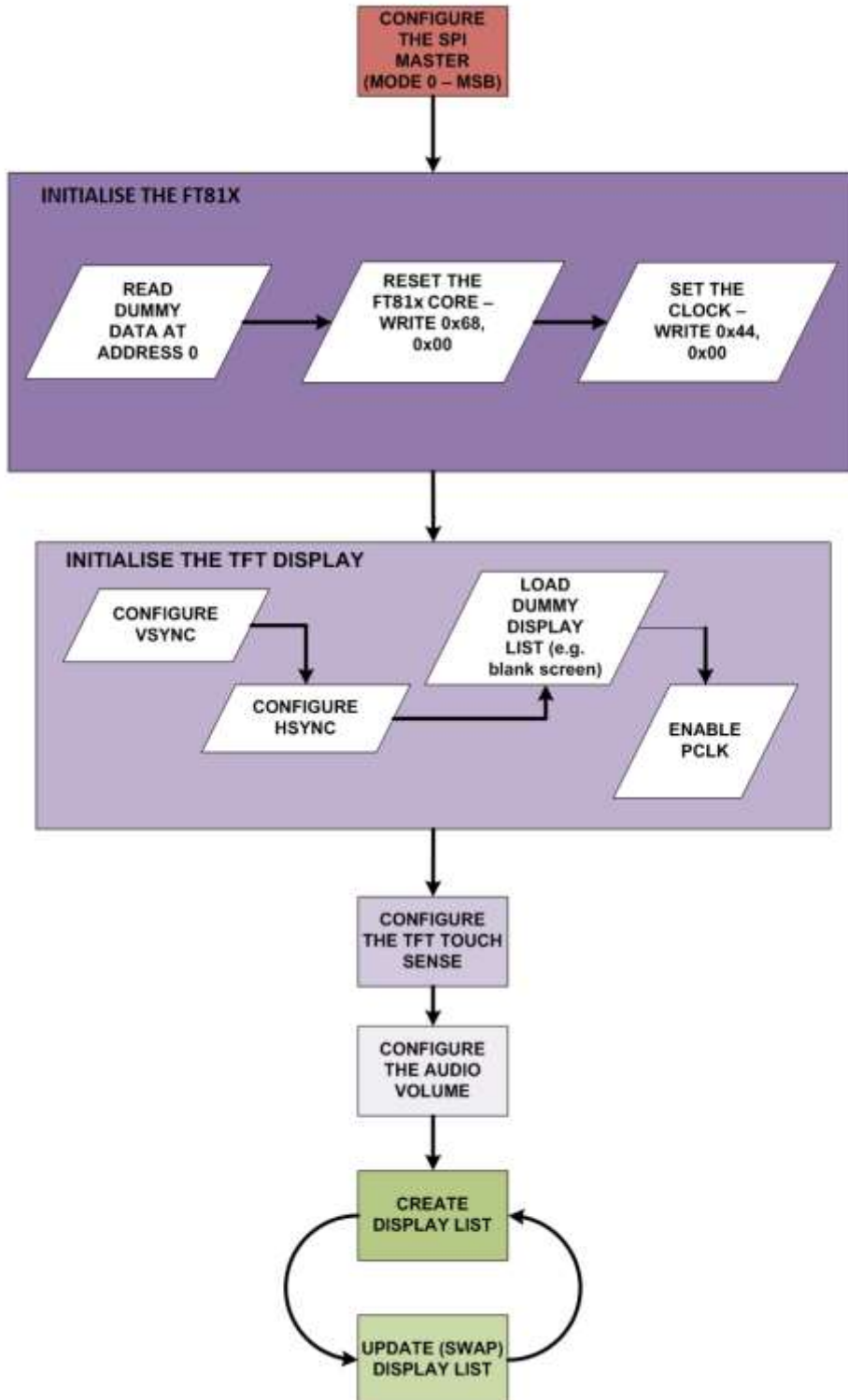


Figure 1-2 Typical Application Flow

1.3.3 Architecture

The Sample Application is designed to easily port to various platforms with SPI host functionality. To achieve this, the Sample Application introduces a HAL (hardware abstraction layer) to make the Sample Application code generic for different platforms as shown in Figure 1-3.

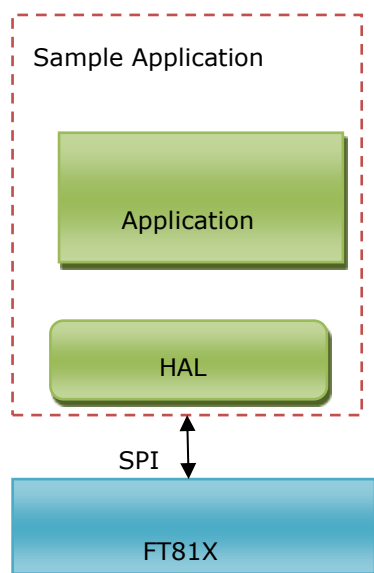


Figure 1-3 Architecture Diagram

1.4 Hardware Requirements

- ME81XA development kit (refer to Table 1-1).
- Microsoft Windows based PC to run the [FT9XX Toolchain](#).
- [UMFTPD2A](#) module for programming FT900.
- Two USB type A to Micro-B cables. One for connecting and providing power to MM900EV1A. Another for programmer (UMFTPD2A) module.

1.5 Software requirements

- [D2XX driver](#) which is a software component needed on the PC to communicate to the programmer (UMFTPD2A).
- Eclipse IDE with [FT9XX Toolchain](#). The IDE (Integrated Development Environment) used to create the Sample Application. It is required by users to build and run the Sample Application.
- [FT81X Sample Application](#) software release package.

1.5.1 Software Package Introduction

1.5.1.1 Project Folder Structure

- Folder "Src" includes the .C source files of the Sample Application.
- Folder "Bin" includes the binary and library files used by the Sample Application
- Folder "Hdr" includes the .h files which define the macros, data structures, and function prototypes for the source .C files.
- Folder "Test" includes the resources files, such as bitmaps and soundtracks used in the application.
- Folder "Project" includes the FT900 project files (as well as other platforms).
- Folder "Documentation" includes this document.\

1.5.1.2 Dependency

- FT81X Sample Application running on FT900 will communicate to FT81X over SPI interface. FTDI D2XX driver plays a role in programming the binary to FT900 through UMFTPD2A programmer.

2 Setup

2.1 Hardware Connection

The picture below shows an example of the hardware setup with the ME813A board.

Two USB ports will be used on the windows PC:

- One USB cable connecting via the Micro-B connector is used to supply power (5V/100mA) from the PC to the MM900EV1A module.
- One USB cable to connect the PC to the UMFTPD2A programmer module.

MM900EV1A and ME813A are connected over an SPI interface.

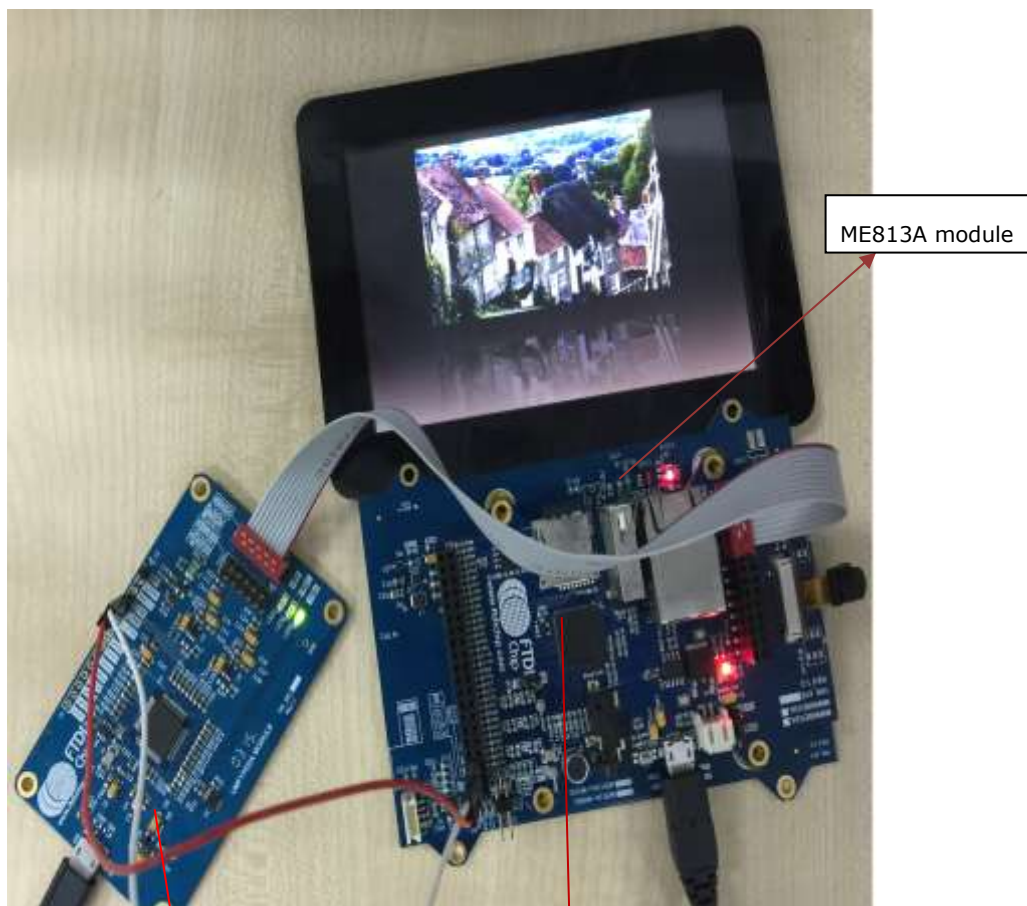


Figure 2-1 Module Connections

UMFTPD2A

MM900EV1A module

Figure 2-2: Module Connections

2.2 Source Code Build

Import the Sample Application project to Eclipse by performing *File --> Import --> Existing Projects into Workspace*. Do not click select 'Copy projects into workspace' as some required directories are not copied with this option.

Figure 2-3 shows the project imported into Eclipse.

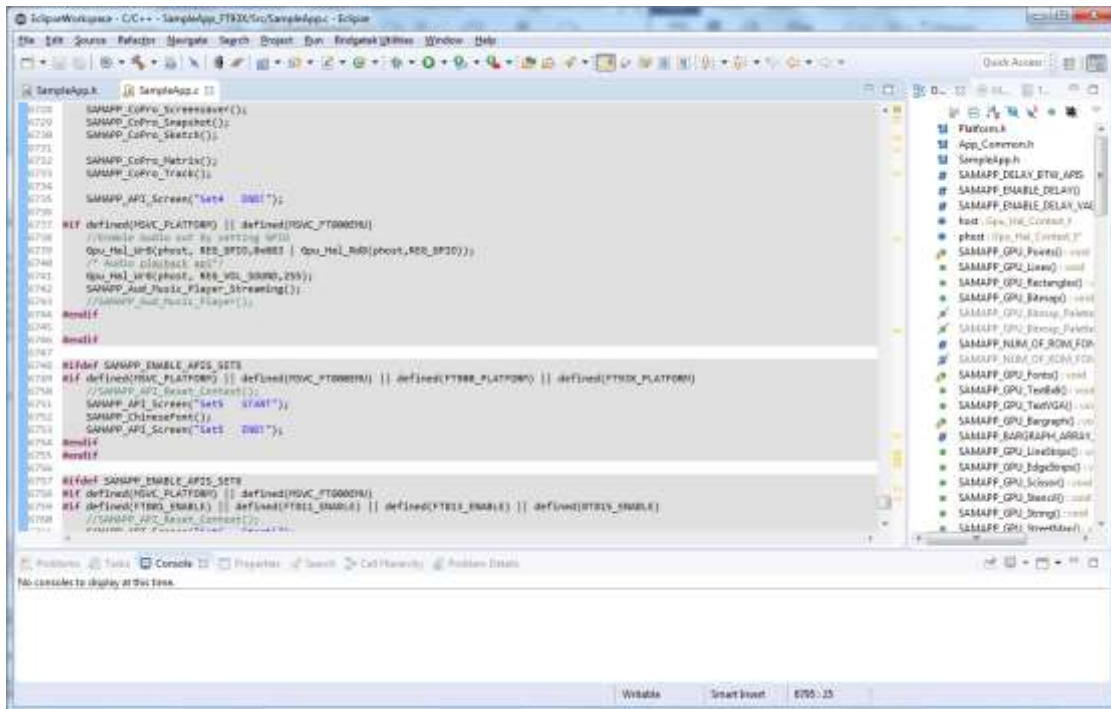


Figure 2-3 Eclipse Project

2.2.1 Determine the Configuration for the Hardware Being Used

In "`<...>\Hdr\FT90x\FT_Platform.h`", enable the platform configuration macro for the module being used. The screen resolution configuration is already included under the module configuration macro.

```

#define ME800A_HV35R (1)
#define ME810A_HV35R (1)
#define ME813A_WV7C (1)
#define ME812A_WH50R (1)
#define ME813A_WH50C (1)
#define ME810A_WH70R (1)
#define ME811A_WH70C (1)

```

This application supports QVGA (320x240), WQVGA (480x272), WVGA (800x480) and HVGA_PORTRAIT (320x480) resolution panels.

After correctly setting the definition, re-build the project.

Display panels other than those stated above are not supported at this time by the Sample Application.

2.2.2 Determine the Group of Functions to Build and Run

All the functions are grouped into the following categories, which are compiled and built under the following compiler switches in the file "Project\Hdr\SampleApp.h"

```
#define SAMAPP_ENABLE_APIS_SET0  
#define SAMAPP_ENABLE_APIS_SET1  
#define SAMAPP_ENABLE_APIS_SET2  
#define SAMAPP_ENABLE_APIS_SET3  
#define SAMAPP_ENABLE_APIS_SET4  
#define SAMAPP_ENABLE_APIS_SET5  
#define SAMAPP_ENABLE_APIS_SET6  
#define SAMAPP_ENABLE_APIS_SET7  
#define SAMAPP_ENABLE_APIS_SET8  
#define SAMAPP_ENABLE_APIS_SET9
```

Users can define all macros to be enabled at the same time or undefine (via #undef) sections to focus on a certain feature.

Note the categories mentioned here are not the same as the groups mentioned in Major Function Groups in the Sample Application. The groups here are defined for the purpose of running on FT900.

2.2.3 Building and Programming the Sample Application Project

After correctly setting the definitions as described in the previous sections, re-build the project. This can be done by right-clicking on the project and selecting Clean, followed by Build. More information can be found in [AN_325 FT9xx Toolchain Installation Guide](#).

To program the executable .bin file into the FT900 IC, use the FT9XX Programming Utility found on the desktop after the FT9XX Toolchain Installation. Select 'Program via One-Wire interface' which uses the UMFTPD2A. More information on this tool usage can be found in the tools About --> Help.

2.2.4 Experimenting with the Sample Application Project

Users can use Eclipse to debug the code which is a good way to learn about the code since you can step through functions.

To do this first place a breakpoint (right-click and select 'Toggle Breakpoint' at the left hand side of the line) at the code line you would like to study before you run the Sample Application. This allows functions to be tested and executed one by one, thus simplifying debugging and observing the functionality of each different function call. A key item to note is that the actual display is not updated until the display list swap is executed.

Information on setting up debugger in Eclipse for FT900 can be found in [AN_325 FT9XX Toolchain Installation Guide](#).

The debug messages from running the application on the FT900 can also be captured on any terminal emulator program. To achieve this, UART0 on the FT900 needs to be connected to the programmer UART interface thus enabling the debug messages to be seen on a terminal program. Specifically make the following connections as shown in Table 2-1 and connect the terminal application to UMFTPD2A Port C.

MM900EVxA	UMFTPD2A
CN3 Pin 4	J2 Pin 5
CN3 Pin 6	J2 Pin 4

Table 2-1 UART0 to UMFTPD2A Connections

Figure 2-4 shows the Sample Application being debugged within the Eclipse IDE.

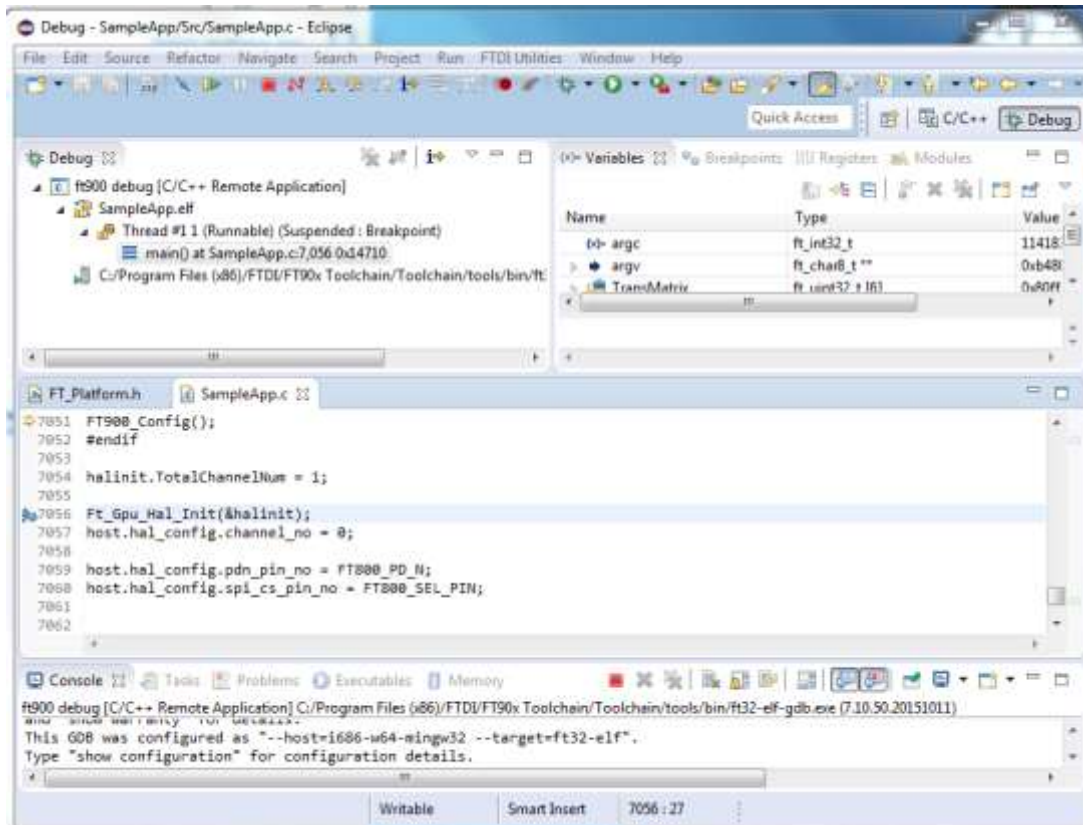


Figure 2-4 Eclipse Debugging

2.3 Source File Brief

"SampleApp.c" under "src\" is the main source file for the Sample Application. The "Main" entry function is inside. It defines all the sample functions. Breakpoints can be set in this file for each of the functions for further study.

The functions in "SampleApp.c" are mostly in the form of "SAMAPP_GPU_xxx" and "SAMAPP_CoPro_xxx".

"Gpu_Hal.c" under folder "src" defines the transportation layer functions, which provide one SPI abstraction layer to access the FT81X. Editing this file allows for porting the application to alternative MCU's and compilers with minimum effort. It is more specific to the SPI master interface than the FT81X.

"CoPro_Cmds.c" defines the APIs of the FT81X coprocessor engine commands. This file is structured to be generic and could be ported to other projects for other target MCU's.

"GPU.h" defines the FT81X specific interface, including the instructions and parameters, register names and memory maps. This file relates directly to the FT81X Programmers Guide and is independent from any MCU.

"SampleApp_RawData.c" defines the bitmap and jpeg data used in the Sample Application.

2.3.1 Major Function Groups in the Sample Application

The major functions in the Sample Application can be classified into the following groups according to the functionality and design purpose.

2.3.1.1 Primitives Group

The functions in this group are designed to demonstrate the usage of FT81X primitives.

An FT81x primitive is the basic drawing command for geometric shapes, for example, points, lines etc. More information on the primitives can be found in the [FT8XX Series Programmers Guide](#).

All the functions are in the form of "SAMAPP_GPU_xxx". Here is the list:

```
/*draw circles*/
SAMAPP_GPU_Points();
/*draw a triangle*/
SAMAPP_Gpu_Polygon();
/*draw lines*/
SAMAPP_GPU_Lines();
/*draw rectangles*/
SAMAPP_GPU_Rectangles();
/*draw bitmaps*/
SAMAPP_GPU_Bitmap();
/*draw palette format bitmap*/
SAMAPP_GPU_Bitmap_Palette()
/*draw palette format bitmap as background*/
SAMAPP_GPU_Bitmap_Palette_Background()
/*draws chars with different fonts*/
SAMAPP_GPU_Fonts();
SAMAPP_GPU_Text8x8();
SAMAPP_GPU_TextVGA();
/*draws a bargraph*/
SAMAPP_GPU_Bargraph();
SAMAPP_GPU_LineStrips();
SAMAPP_GPU_EdgeStrips();
/*example of cutting away an active area on the display*/
SAMAPP_GPU_Scissor();
SAMAPP_Gpu_Cube();
SAMAPP_GPU_Ball_Stencil();
/*Call and Return Primitives combination*/
SAMAPP_GPU_StreetMap();
/*Additive blending of fonts*/
SAMAPP_GPU_AdditiveBlendText();
/*Usage of Macro*/
SAMAPP_GPU_MacroUsage();
/*Additive blending of points*/
```

```
SAMAPP_GPU_AdditiveBlendPoints();
```

2.3.1.2 Widgets Group

The functions in this group are designed to demonstrate the FT81X graphic engine widgets, which are visual components that have been created to reduce the effort of GUI programmers.

A widget will create a complex object with one command as opposed to many. For example, the clock widget provides a large circle for the face, twelve circles for each number, and 3 lines for each clock hand. If this image was created without the widget the programmers would need to draw 13 circles and 3 hands in separate primitive commands.

There are 15 in-built widgets and their sample functions are in the form of "SAMAPP_CoPro_Widget_xxx" as below:

```
SAMAPP_CoPro_Logo();  
SAMAPP_CoPro_Widget_Clock();  
SAMAPP_CoPro_Widget_Guage();  
SAMAPP_CoPro_Widget_Gradient();  
SAMAPP_CoPro_Widget_Keys();  
SAMAPP_CoPro_Widget_Keys_Interactive();  
SAMAPP_CoPro_Widget_Progressbar();  
SAMAPP_CoPro_Widget_Scroll();  
SAMAPP_CoPro_Widget_Slider();  
SAMAPP_CoPro_Widget_Dial();  
SAMAPP_CoPro_Widget_Toggle();  
SAMAPP_CoPro_Widget_Spinner();  
SAMAPP_CoPro_Widget_Text();  
SAMAPP_CoPro_Widget_Number();  
SAMAPP_CoPro_Widget_Button();
```

The following functions are designed to demonstrate additional FT81X commands, which are frequently used by programmers to simplify a project. They are in the form of "SAMAPP_CoPro_xxx".

```
/*Screen calibrate example*/  
SAMAPP_CoPro_Widget_Calibrate();  
SAMAPP_CoPro_Screensaver();  
/*Matrix example for Bitmap manipulation*/  
SAMAPP_CoPro_Matrix();  
/*Appending block of memory to the current display list*/  
SAMAPP_CoPro_AppendCmds();  
/*Decompress functionality example*/  
SAMAPP_CoPro_Inflate();  
/*JPEG decoding functionality example*/  
SAMAPP_CoPro_Loadimage();  
/*Customer Font example*/  
SAMAPP_CoPro_Setfont();  
/*Customer Font used in widget example: Chinese Font*/  
SAMAPP_ChineseFont();  
/*Track usage example for touch*/  
SAMAPP_CoPro_Track();  
/*Screenshot example*/  
SAMAPP_CoPro_Snapshot();
```

```
/*Sketch example*/  
SAMAPP_CoPro_Sketch();
```

2.3.1.3 Audio & Touch Group

The APIs in this group demonstrate how to utilize the audio and touch functionality of the FT81X.

```
/* Audio playback API */  
SAMAPP_Aud_Music_Player();  
/* Audio Playback sample function in streaming way*/  
SAMAPP_Aud_Music_Player_Streaming();  
/*FT800 Built-In Sound sample function*/  
SAMAPP_Sound()  
  
/*FT800 Touch and Tag usage sample function*/  
SAMAPP_Touch()  
/* FT800 Track coprocessor engine command usage sample */  
SAMAPP_CoPro_Track()  
/*FT800 keys widget and touch tag usage example*/  
SAMAPP_CoPro_Widget_Keys_Interactive()
```

2.3.1.4 Host Command Group

The APIs in this group demonstrate the power management feature of the FT81X.

```
/*Toggle the PD_N pin of FT8xx for power cycle*/  
Gpu_Hal_Powercycle ()  
/*  
FT800 Host Command Function: users can send the respective host commands to achieve  
clock source selection, power mode switch, frequency selection as well as core  
reset.  
*/  
Gpu_HostCommand()  
/*  
This API defines 6 scenarios of power mode switch, implemented by calling functions  
above.  
*/  
SAMAPP_PowerMode()
```

3 Helpful Hints

Users are strongly encouraged to read the datasheets of the ME81XA development systems before reading this document and starting to use the Sample Application.

The use of “debug” mode will allow individual sections of the Sample Application to be executed through the placement of breakpoints.

Note that a calibration procedure (e.g. `SAMAPP_CoPro_Calibrate()`) is required if experimenting with the touch screen feature.

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 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
Troï,
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 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

- FT81X Product Page: <http://brtchip.com/i-ft8/>
 FT90X Product Page: <http://brtchip.com/m-ft9/>
[ME810A-HV35R Datasheet](#)
[ME812A-WH50R Datasheet](#)
[ME813A-WH50C Datasheet](#)
[AN_391 EVE Platform Guide](#)
 AN_187 [User Guide For LibMPSSE-SPI](#)
[FT8XX Series Programmer Guide](#)
[FT8XX Embedded Video Engine Datasheet](#)
[Sample Application](#)
[MM900EV Development Modules](#)
[UMFTPD2A](#) MM900EV MCU Programmer and Debugger Module
[FT9XX Toolchain](#)
[AN_325 FT9XX Toolchain Installation Guide](#)

Acronyms and Abbreviations

Terms	Description
EVE	Embedded Video Engine
IDE	Integrated Development Environment
MPSSE	FTDI Multi-Protocol Synchronous Serial Engine
MSVC	Microsoft Visual Studio C++ 2010
SPI	Serial Peripheral Interface
UI	User Interface
USB	Universal Serial Bus

Appendix B – List of Figures & Tables

List of Figures

Figure 1-1 Example Block Diagram of Setup for ME81XA board with Windows PC.....	4
Figure 1-2 Typical Application Flow	5
Figure 1-3 Architecture Diagram.....	6
Figure 2-1 Module Connections.....	8
Figure 2-2 Eclipse Project	9
Figure 2-3 Eclipse Debugging	11

List of Tables

Table 1-1 ME81XA development modules.....	3
Table 2-1 UART0 to UMFTPD2A Connections.....	11

Appendix C– Revision History

Document Title : AN_417 ME81XA Sample Application MM9XX Introduction
Document Reference No.: BRT_000056
Clearance No.: BRT#047
Product Page: <http://brtchip.com/product>
Document Feedback: [Send Feedback](#)

Revision	Changes	Date
1.0	Initial Release	2016-11-08
1.1	Document migrated from Dual branding to BRT (Updated company logo; copyright info; contact information; hyperlinks)	2018-01-08