Application Note
AN_418
ME81XAU Sample Application PC Introduction

Version 1.0

Issue Date: 2017-12-27

This application note is provided to introduce the setup of a Sample Application running on a Windows PC with an ME81XAU 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.

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

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 host MCU via a SPI interface. To help customers easily utilize the functionality of the FT81X in a project, a Sample Application is provided here for tutorial purposes.

The ME81XAU modules have one of FT810, FT811, FT812, or FT813 with a FT4222 as USB to SPI bridge chip, mounted on the module PCB. This is intended to interface with a PC host over a USB cable. The FT4222 bridge chip on the module converts USB to an SPI interface and connects to the FT81X, thus enabling the PC to provide stimulus to the FT81X for fast and easy development.

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

Note that although the basic project compiles for the Windows PC with the LibFT4222 library, 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 the Sample Application adapted for the specific device.

Further information regarding the FT81X programming language or pseudo-code can be found in the FT81X Series Programmer Guide. This document introduces how to set up and use the Sample Application with the Bridgetek ME812AU/ME813AU development kits on Windows PC.

For ME812AU or ME813AU development board details please refer to the ME812AU datasheet or ME813AU datasheet.

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 have 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 Microsoft Visual Studio C++ software. It comprises the source code as well as project files.

1.3 Overview

1.3.1 Hardware

The diagram below illustrates the overall hardware setup.

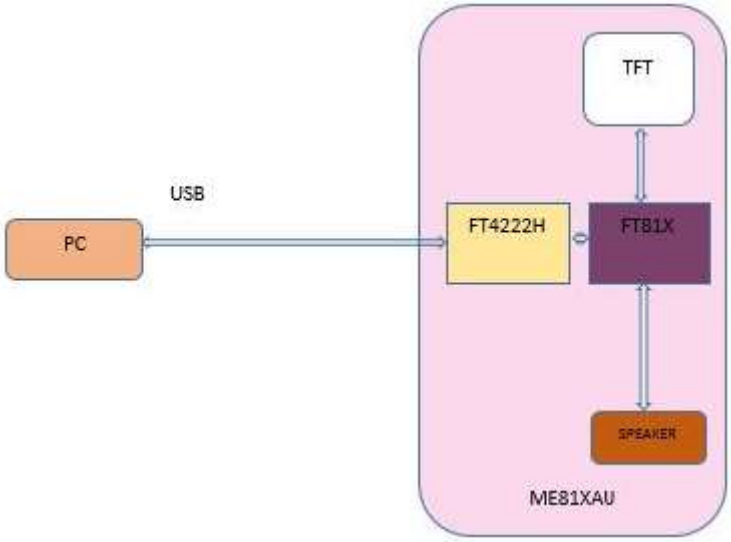

Figure 1-1: Example Block Diagram of Setup for ME81XAU board with Windows PC

The ME812AU and ME813AU units require minimal hardware configuration. The FT81X IC, FT4222H, the TFT display, and the speaker are all included in the module assembly. The micro USB cable provides connection from the PC to the FT4222H which is mounted on the module.

The FT812 provides an interface for resistive touch panels and the FT813 to capacitive touch panels. More about this is mentioned in the FT81X datasheet.

1.3.2 Application flow

The diagram below gives the basic flow to configure the FT81X in an application.

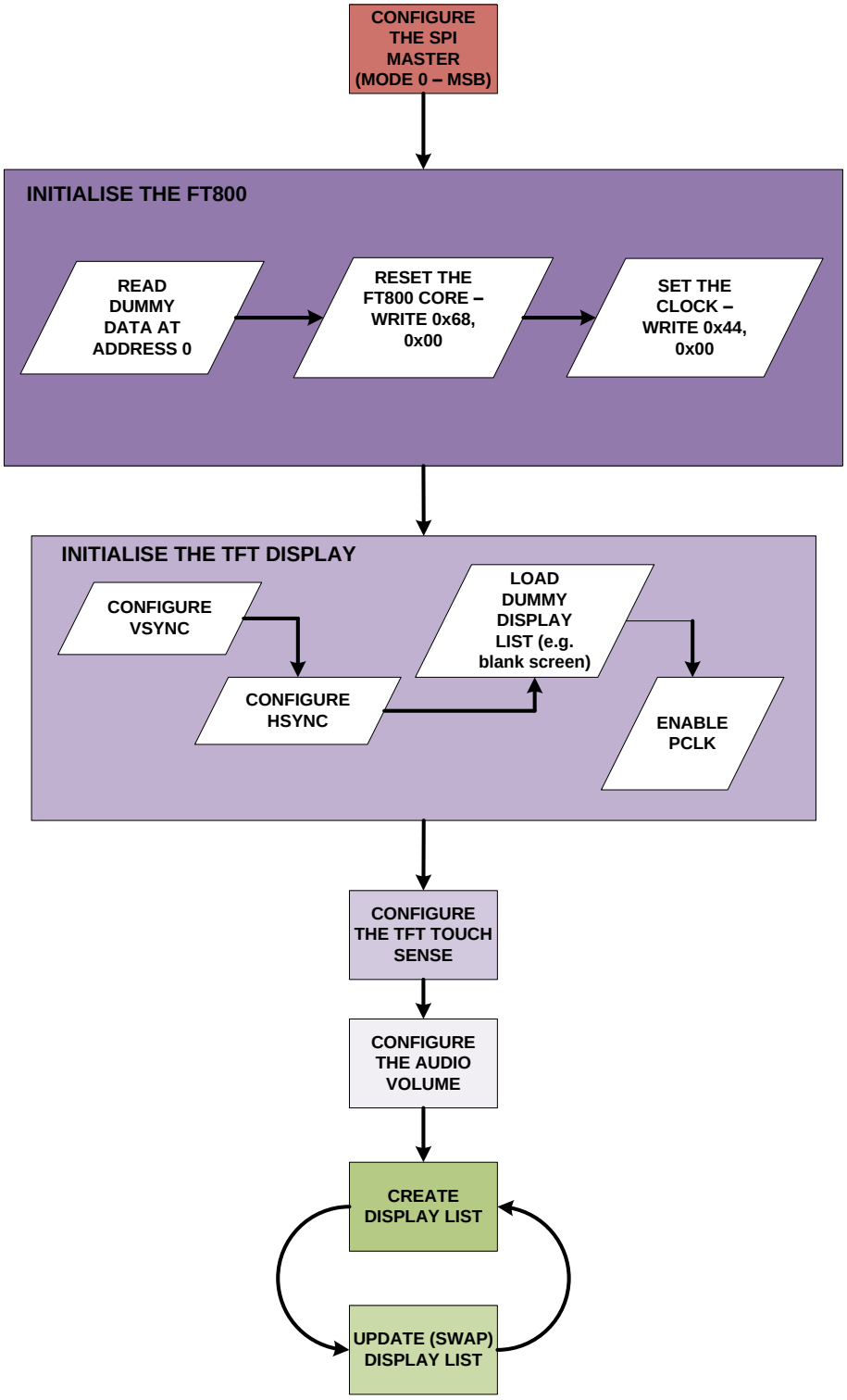

Figure 1-2: Typical Application Flow

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 HAL (hardware abstraction layer) to make the Sample Application code generic for different platforms.

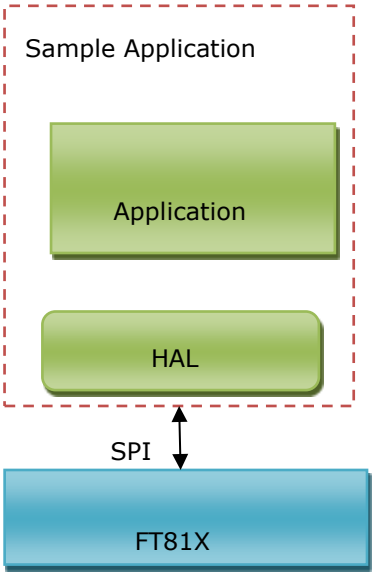

Figure 1.3: Architecture diagram

1.4 Hardware Requirements

- ME812AU or ME813AU development kit.
- Microsoft Windows based PC working as a host. Windows XP onwards is preferred.
- One USB type A to Micro-B cable to provide power and connection to USB-to-SPI bridge chip mounted on ME812AU / ME813AU module.

1.5 Software Requirements

- D2XX driver for the MPSSE cable. Please download and install on the PC.
- Microsoft Visual C++. The IDE (Integrated Development Environment) used to create the Sample Application. It is required by users to build and run the Sample Application. Note: The express version of the tool is a free download from Microsoft.
- FT81X Sample Application release package.

1.5.1 Software package introduction

1.5.1.1 Folder introduction

- 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 MSVC project and solution file.
- Folder "Documentation" includes this document.

1.5.1.2 Dependency

- FT81X Sample Application depends on the D2XX driver to communicate with FT81X through the FT4222 bridge chip.
- FT81X Sample Application is one console application with a GUI interface at the Windows PC side.
- FT81X Sample Application is linked with the libFT4222 library, Please see http://www.ftdichip.com/Products/ICs/FT4222H.html for details.

2 Setup

2.1 Hardware Connection

The picture below shows the connection on the ME81XAU board. This printed circuit board has FT81x and FT4222 mounted on it.

One USB port will be used on the windows PC.

USB cable connected via the MicroB connector provides 5V/100mA power supply from the PC to the module and USB-to-SPI bridge interface.

Figure 2-1: Module Connections

2.2 Source Code Build

Open the file "Project\Msvc_win32\SampleApp\ SampleApp.sln" with Microsoft Visual C++ and the following screen should appear:

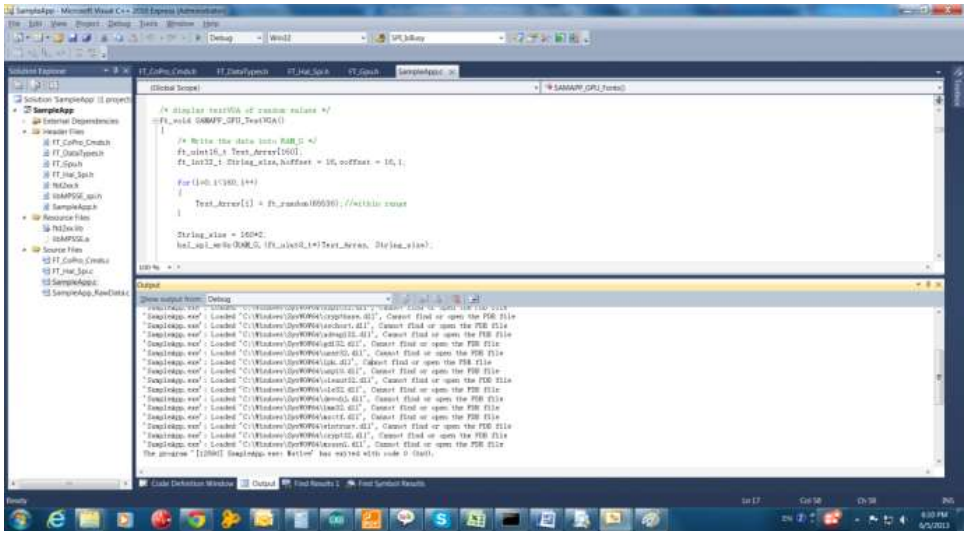

Figure 2-2: Visual Studio Project Page

2.2.1 Determine the Configuration for Hardware under use

In "<...>\Hdr\Msvc_win32\Platform.h ", enable the platform configuration macro (one of the below shown) for the module under use. Screen resolution configuration is already included under the module configuration macro.

```
//#define VM800B43_50                    (1)
//#define VM800B35                       (1)
//#define VM801B43_50                    (1)
//#define VM810C50                       (1)
#define ME812AU_WH50R                    (1)
//#define ME813AU_WH50C                  (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 following categories, which are compiled and built under the following compiler switches in the file "<...>\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 undefined 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 the Windows PC.

2.2.3 Experimenting with the Sample Application Project

Place the breakpoint at the 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.

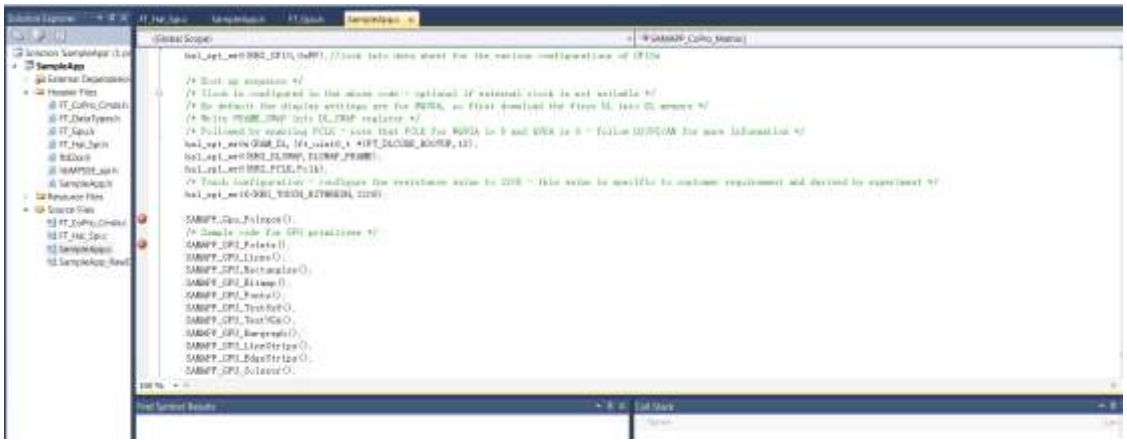

Figure 2-3: Visual Studio Project Page

2.2.4 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 to port 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.

2.2.5 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.

2.2.5.1 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 FT81X 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()
/*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();
/*Font and Points Primitives combination*/
SAMAPP_GPU_String();
/*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();
```

2.2.5.2 Widgets Group

The functions in this group are designed to demonstrate the FT8XX 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 14 in-built widgets and their sample functions are in the form of "SAMAPP_CoPro_Widget_xxx" as below:

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

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_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();
```

2.2.5.3 Audio & Touch Group

The APIs in this group demonstrate how to utilize the audio and touch functionality of the FT8XX.

```
/* 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()
```

2.2.5.4 Host Command Group

The APIs in this group demonstrate the power management feature of the FT8XX.

```
 /*Toggle the PD_N pin of FT800 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 ME812AU and ME813AU development systems before reading this document and starting to use the Sample Application.
- Before running the FT81X Sample Application, please make sure the Micro USB cable is connected with ME812AU or ME813AU board and the D2XX driver is loaded.
- If encountering issues while running the FT81X Sample Application, please check the USB port status in the device manager of the Windows PC and make sure there are no other USB ports in use with the D2XX driver.
- 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@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

- AN_391 EVE Platform Guide
- ME812AU Datasheet
- ME813 AU Datasheet
- FTDI MPSSE for SPI Application Note
- FT8XX Series Programmer Guide
- FT8XX Embedded Video Engine Datasheet
- Sample Application

Acronyms and Abbreviations

Terms	Description
EVE	Embedded Video Engine
IDE	Integrated Development Environment
MPSSE	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 Tables & Figures

List of Figures

List of Tables

NA

Appendix C– Revision History

Document Title: AN_418 ME81XAU Sample Application PC Introduction
Document Reference No.: BRT_000184
Clearance No.: BRT#105
Product Page: http://brtchip.com/product/
Document Feedback: Send Feedback

Revision	Changes	Date
1.0	Initial release	2017-12-27