



Application Note

AN_280

4DLCD-FT843 SampleApp Arduino Introduction

Version 1.1

Issue Date: 2017-12-28

This application note introduces how to setup the FT8XX Sample Application running on an Arduino - ATmega328 system using the 4DLCD-FT843 system. The objective of the Sample Application is to enable users to become familiar with the usage of the FT8XX, the design flow, and display list language used to design the desired user interface or visual effect.

Table of Contents

1 Introduction	4
1.1 Audience	5
1.2 Scope	5
2 Overview	6
2.1 Hardware Block Diagram.....	6
2.2 Application flow	7
2.3 Architecture	8
2.4 Hardware Requirements.....	8
2.5 Software Requirements.....	8
2.5.1 Software Package Introduction	8
3 Setup.....	10
3.1 Hardware Connection	10
3.1.1 ADAM - Arduino Display Adaptor Module Connections	10
3.2 Source code build and download	11
3.2.1 Determine the Screen Size.....	12
3.2.2 Determine the Group of Functions to Build and Run	12
3.2.3 Source File Brief	12
3.2.4 Project File Brief	13
3.2.5 Major Function Groups in Sample Application	13
4 Helpful Hints.....	16
5 4D- Systems	17
6 Contact Information	18
Appendix A – References	19
Document References	19
Acronyms and Abbreviations.....	19
Appendix B – List of Tables & Figures	20
List of Figures	20



List of Tables.....20
Appendix C – Revision History 21

1 Introduction

The FT8XX combines display, audio and touch functionality into one single chip, powered by Bridgetek's advanced EVE technology (Embedded Video Engine). The FT8XX device interfaces with a system MCU via an SPI interface. To enable customers to more easily utilize the functionality of the FT8XX in a project, a Sample Application is provided here for tutorial purposes.

The sample application has been written for an ATmega328 based platform e.g. Arduino Uno or Sparkfun RedBoard (DEV-10908).

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 FT8XX.

Note that although the basic project is created for ATmega328, the code relating to the creation of the screen shots could be reused in different MCU design environments. In addition, the set-up steps for the ATmega328 would be necessary for other MCU's.

This document introduces how to set up and use the Sample Application with a BRIDGETEK 4DLCD-FT843 development system- **Figure 1** in relation to conjunction with the Sparkfun RedBoard platform. Further information regarding the FT8XX programming language or pseudo-code can be found in the FT8XX Programmer Guide.

For 4DLCD-FT843 development board details, please refer to www.4dsystems.com.au.

To learn more about Arduino Uno and its IDE, please check www.arduino.cc.

To learn more about Sparkfun RedBoard (DEV-10908), please refer to www.sparkfun.com.

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



Figure 1 - 4DLCD-FT843 Display incorporating the FT8XX

Figure 2 shows the 4DLCD-FT843 Display connected to the Sparkfun RedBoard (DEV-10908). The 4DLCD-FT843 enables system designers to rapidly create high quality; human machine interfaces (HMIs). It includes a 4.3" TFT display (with 480 x 272 pixel resolution), PWM audio output (with amplifier enable), 58 synthesized sounds, a 4-wire resistive touch screen, all integrated on a flexible ribbon connector.

The 4DLCD-FT843 Display is connected to the Sparkfun RedBoard (DEV-10908) via the Arduino Display Adaptor Module (ADAM 4Display-Shield-FT843).

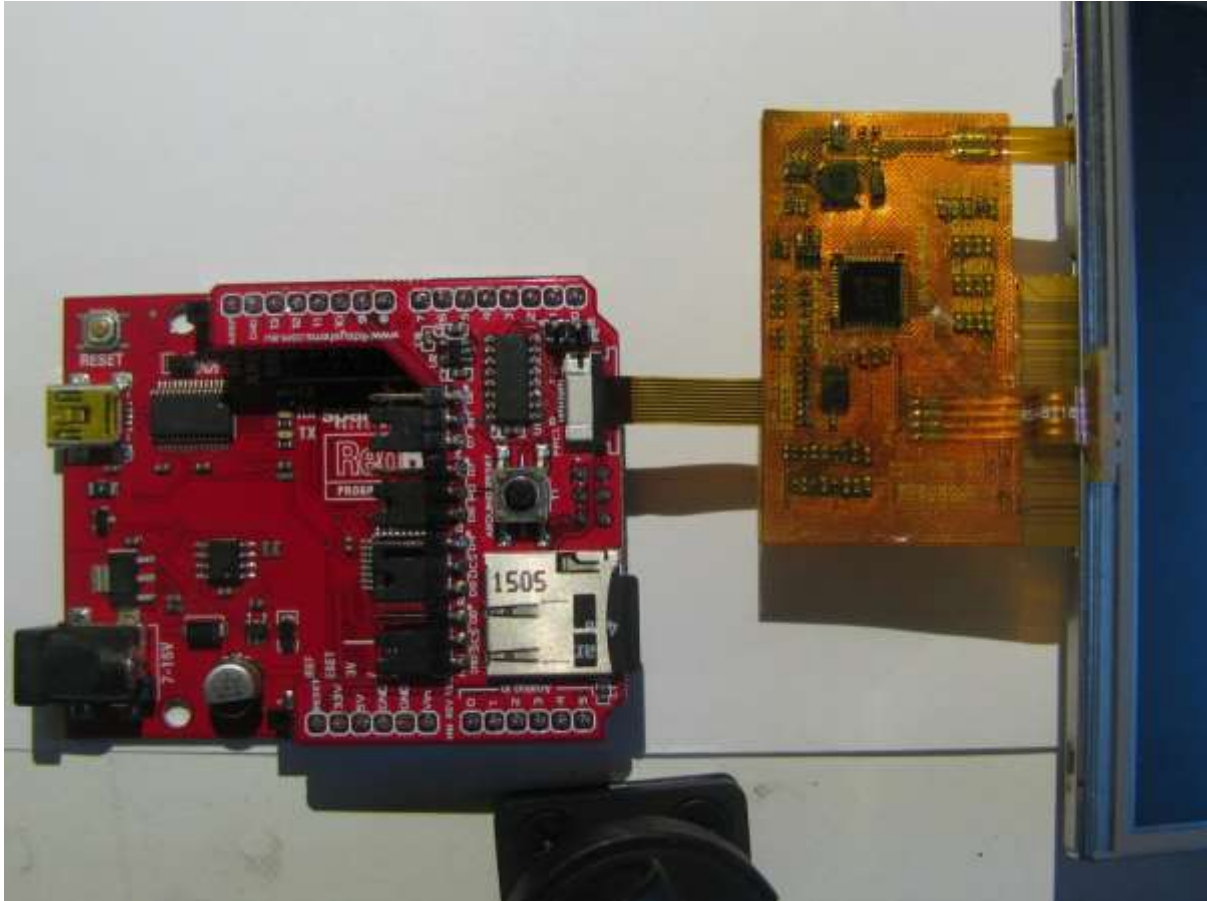


Figure 2 - 4DLCD-FT843 Display connected to the Sparkfun RedBoard

1.1 Audience

This document assumes the audience has read the datasheet and Programmer Guide of the FT8XX. In addition, familiarity of the C/C++ programming language is necessary to understand the Sample Application source code. To understand the SPI of the Arduino Platform, knowledge of ATmega328 microcontroller and IDE (Integrated Design Environment) is required.

1.2 Scope

The Sample Application mentioned in this document is created in the Arduino IDE and runs on Sparkfun RedBoard connected to the 4DLCD-FT843 Display module. It is comprised of the source code as well as project files.

2 Overview

2.1 Hardware Block Diagram

Figure 3 below gives the basic hardware setup with additional audio components.

The Arduino Display Adaptor Module (shown in red dashes) is used to connect the Arduino board to the 4DLCD-FT843 Display via SPI. The SampleApp code includes audio; if this is required a simple filter / amplifier circuit and speaker can be added as shown in figure 3. An alternative is to connect a standalone speaker with built-in amplifier.

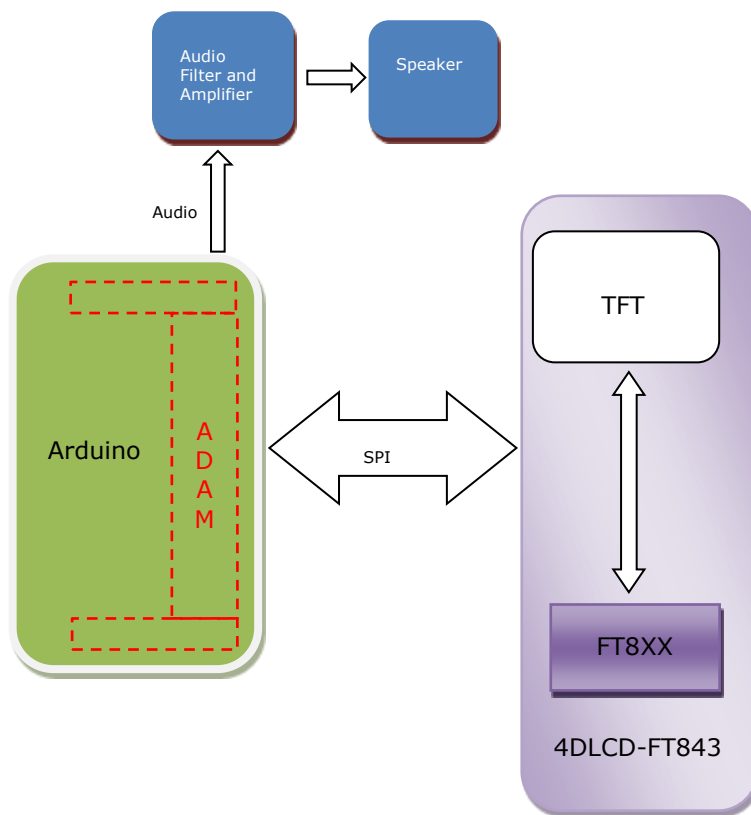


Figure 3 - Hardware Setup Block Diagram

2.2 Application flow

The diagram below gives the basic flow and structure for configuring the FT8XX in an application.

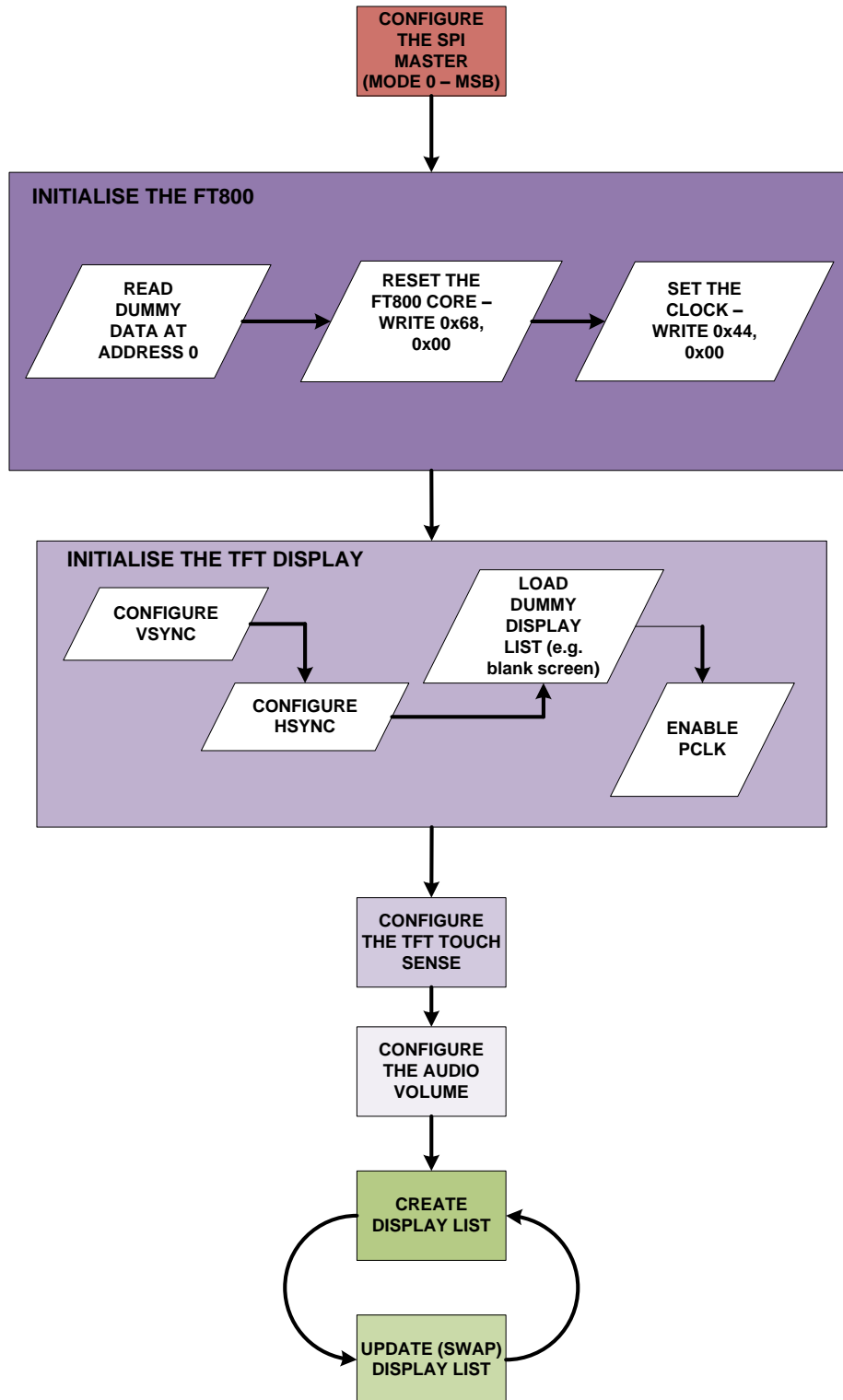


Figure 4 - Application Flow

2.3 Architecture

The Sample Application is designed to easily port to various platforms with SPI host functionality. Therefore, this sample application introduces one generic HAL (hardware abstraction layer) which can be used as a guideline for other platforms. Note additional processor specific HALs are being developed, check www.brtchip.com for the latest support.

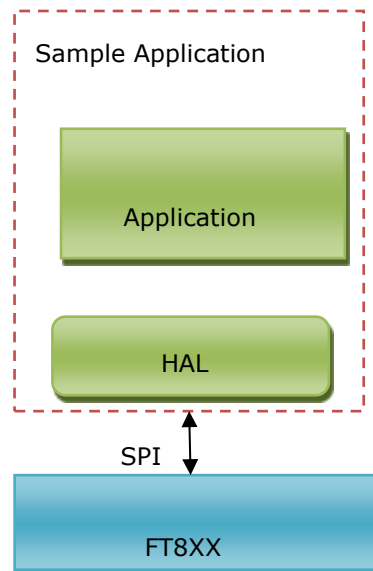


Figure 5 - Architecture diagram

2.4 Hardware Requirements

- 4DLCD-FT843 DISPLAY development kit and additional simple filter and amplifier circuit for Audio.
- Arduino Display Adaptor Module – ADAM (4Display-Shield-FT843)
- ATMEL MEGA 328P microcontroller - Arduino Uno or Sparkfun RedBoard (DEV-10908).
- USB cable with a Mini-B connector to program /power the Arduino.

2.5 Software Requirements

- Arduino IDE 1.0.5
- FT8XX Sample Application release package.

2.5.1 Software Package Introduction

2.5.1.1 Folder Introduction

- Folder "Project\Arduino" contains the entire source code and project file "SampleApp.ino".
- The other folders are not relevant to the Arduino platform.

2.5.1.2 Dependency

The Sample Application uses the SPI library provided by Arduino with Arduino IDE. Please check the [Arduino website](#) for details.

3 Setup

3.1 Hardware Connection

The ADAM module is mated onto the top of the Sparkfun RedBoard (DEV-10908) as shown in **Figure 6**. The 4DLCD-FT843 DISPLAY flat panel cable (FPC) connects into the FPC socket on the top of the ADAM module. USB micro cable is used for power/programming.

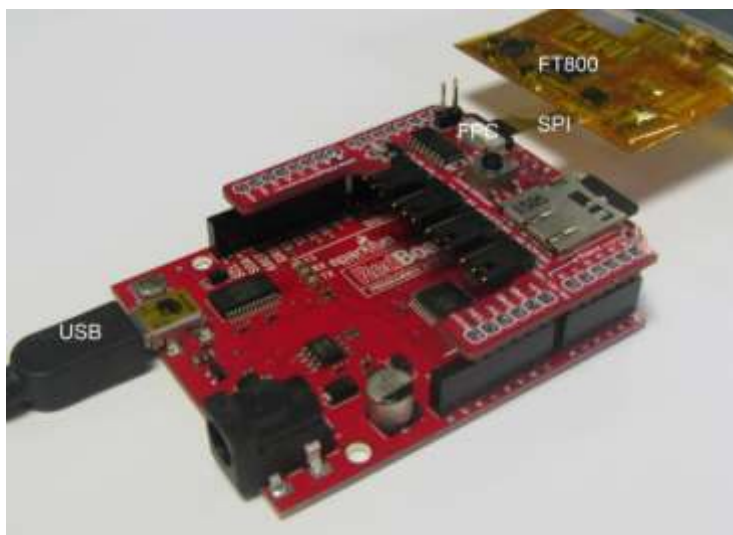


Figure 6 - Hardware Connections

3.1.1 ADAM - Arduino Display Adaptor Module Connections

Figure 7 shows the Arduino Display Adaptor Module connections, **Table 1** gives a brief description of each connection. Jumpers J1 to J4 are shown in their default positions. For full details of the ADAM module (4Display-Shield-FT843) refer to the data sheet available – [ADAM datasheet](#)

Connector	Description
FPC1	10 way Bottom Contact FPC for connecting directly to the 4DLCD-FT843 displays flex
P2	Audio output to a simple filter / amplifier circuit and speaker or a standalone speaker with built-in amplifier.
RESET	Resets the ATmega328
MICRO SD	The micro-SD socket on the ADAM Shield enables the Arduino to access files for use with the FT8XX Graphics Controller, along with being a storage media for general storage used by the Arduino.
J1	SD card Chip Select – default pin 1 and 2
J2	Display Chip Select – default pin 1 and 2
J3	PD – Power down display enable – default pin 1 and 2
J4	INT - Interrupt from the Display – default pin 1 and 2

Table 1 – ADAM – Arduino Display Adaptor Module Connections

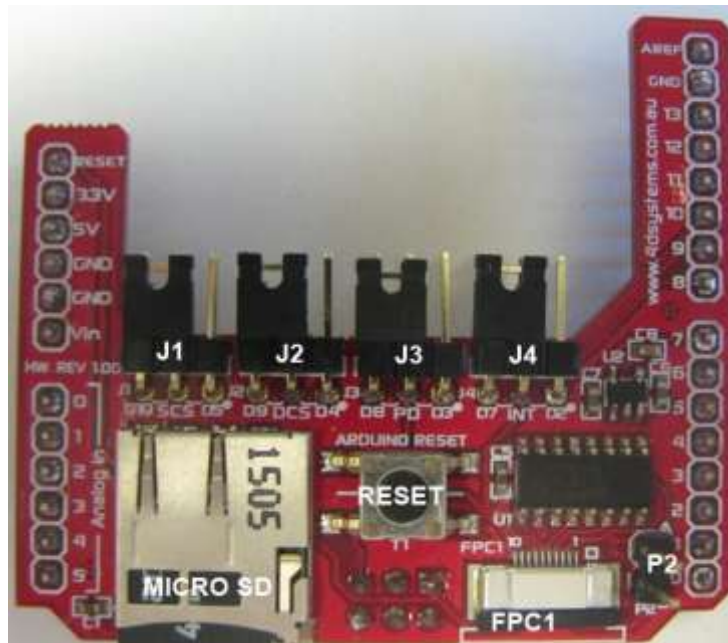


Figure 7 - ADAM Module

3.2 Source code build and download

Please note that all the related source code resides at folder "Project\Arduino\SampleApp".

To build the project, open the file "Project\Arduino\SampleApp\SampleApp.ino" with the Arduino IDE and the following screen will be shown:



```

// Arduino\SampleApp\SampleApp.ino
// SampleApp.ino

#include <Arduino.h>

#define FT_HAL_SPI_H
#define FT_HAL_SPI_H

/* DEFINITIONS USED FOR ARDUINO PLATFORM */
#define ARDUINO_PLATFORM
#define FT_ARDUINO_FREQ_SPI_CS (10)
#define FT400_INT (2)
#define FT400_SPI_4 (4)
#define FT400_SPI_4 (4)
#define FT400_SPI_4 (4)

/* DEFINITIONS USED FOR FT843 OR PINOCCHIO CURRENT BOARD */
#define FT_843_EL_SIZE (8*1024)
#define FT_843_FIFO_SIZE (4*1024)
#define FT_843_BLOCK_SIZE (2*1024)

/* AFTER THE SPI HAL INITIALIZATION */
ft_void hal_spi_init();
ft_void hal_spi_exit();

/* AFTER THE LOWLEVEL START */
ft_void hal_spi_start(ft_uint32_t addr);
ft_void hal_spi_wstart(ft_uint32_t addr);
ft_void hal_spi_wend(ft_uint32_t count);
/* SPI END LOWLEVEL END */
ft_void hal_spi_end();

/* AFTER THE VOID FUNCTIONALITY */
ft_uint32_t hal_spi_rst(ft_uint32_t addr);
ft_uint32_t hal_spi_wst(ft_uint32_t addr);

// Sketch size: 26,252 bytes (of a 30,720 byte maximum)

```

Figure 8 - Arduino IDE with the opened project

Press "Ctrl+R" to rebuild and "Ctrl+U" to download the binary generated into Arduino.

For further details of downloading or programming Arduino, check the website of Arduino.

3.2.1 Determine the Screen Size

For 3.5 inch displays, ensure the compilation macro switch `"#define SAMAPP_DISPLAY_QVGA"` in file "Project\Arduino\Platform.h" is defined.

For other development boards (4.3" and 5.0" displays), make sure the macro above is undefined within the sample application project. This will ensure the correct display resolution is selected to match the correct display size.

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

3.2.2 Determine the Group of Functions to Build and Run

Due to the limitation of the ATmega328 flash size (32KBytes), it is impossible to put all the functions together and program them into the ATmega328. All the functions are grouped into 5 categories, which are compiled and built under the following compiler switches in the file "Project\Arduino\Platform.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
```

Users must enable one API set exclusively, by defining one of the above macros at a time; otherwise the binary build will not fit into the ATmega328 flash.

Note the categories mentioned here are not the same as the groups mentioned in Section 3.2.5. The groups here are defined for the purpose of running on the ATmega328.

3.2.3 Source File Brief

"*SampleApp.cpp*" is the main source file for the Sample Application. The main entry function is inside. It defines all the sample functions.

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

"*Gpu_Hal.cpp*" defines the transportation layer functions, which provides one SPI abstraction layer to access the FT8XX. Editing this file, allows for porting the application to alternative MCU's and compilers with minimal effort. It is more specific to the SPI master interface.

"*CoPro_Cmds.cpp*" defines the APIs of the FT8XX 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 FT8XX specific instruction parameters, register names and memory maps. The contents of this file relate directly to the FT8XX Programmers Guide and is structured to be generic such that it could be ported to other projects for other target MCU's.

"*SampleApp_RawData.cpp*" defines the bitmap data used in the sample application.

3.2.4 Project File Brief

"SampleApp.ino" is the project file used by the Arduino IDE and it includes all the necessary files in this project.

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

3.2.5 Major Function Groups in Sample Application

The major functions in the sample application can be classified into the following group according to its functionality and design purpose.

3.2.5.1 Primitives Group

The functions in this group are designed to demonstrate the usage of FT8XX primitives.

An FT8XX primitive is the basic drawing command e.g. Points are used to draw circles, while Lines is used for straight lines. More information on the primitives may be found in the FT8XX Programmers Guide.

All the function 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();`
- `/*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();`

3.2.5.2 Widgets Group

The functions in this group are designed to demonstrate the FT8XX graphic engine widgets, which are visual components to reduce the effort of GUI programmers.

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

There are currently 14 in-built widgets and the sample functions are in the form of "SAMAPP_CoPro_Widget_xxx".

- 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 FT8XX 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();
- */*Track usage example for touch*/*
SAMAPP_CoPro_Track();
- */*Screenshot example*/*
SAMAPP_CoPro_Snapshot();
- */*Sketch example*/*
SAMAPP_CoPro_Sketch();

3.2.5.3 Audio & Touch Group

- */* 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 example*/  
○ SAMAPP_CoPro_Widget_Keys_Interactive();
```

3.2.5.4 Host Command Group

```
/*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()
```

4 Helpful Hints

- Audio playback functions "SAMAPP_Aud_Music_Player" and "SAMAPP_Aud_Music_Player_Streaming()", are not available on the Arduino Platform due to the audio playback file being too large to fit into the ATmega328 flash.
- Note that a calibration procedure (e.g. SAMAPP_CoPro_Calibrate();) is required if experimenting with the touch screen feature.

5 4D- Systems

4D Systems designs and manufactures compact and cost effective Intelligent Display Modules using the latest state of the art OLED and LCD technology with an embedded custom graphics processor. To find out more about 4D Systems and more importantly the FT8XX support they provide, visit their website at www.4dsystems.com.au.

6 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](http://brtchip.com/) 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

- [AN_391 EVE Platform Guide](#)
- [FT8XX Embedded Video Engine Datasheet](#)
- [FT8XX Series Programmer Guide](#)
- [AN_240 FT8XX From the Ground Up](#)
- 4D Systems (www.4dsystems.com.au)
- Arduino (<http://arduino.cc/>)

Acronyms and Abbreviations

Terms	Description
EVE	Embedded Video Engine
GPIO	General Purpose Input / Output
IC	Integrated Circuit
MCU	Microcontroller
QVGA	Quarter VGA (320 x 240 pixel display size)
SPI	Serial Peripheral Interface
TFT	Thin-Film Transistor
VGA	Video Graphics Array
WQVGA	Wide Quarter VGA (480 x 272 pixel display size)
FPC	Flat Panel Cable

Appendix B – List of Tables & Figures

List of Figures

Figure 1 - 4DLCD-FT843 Display incorporating the FT8XX	4
Figure 2 - 4DLCD-FT843 Display connected to the Sparkfun RedBoard.....	5
Figure 3 - Hardware Setup Block Diagram.....	6
Figure 4 - Application Flow	7
Figure 5 - Architecture diagram	8
Figure 6 - Hardware Connections	10
Figure 7 - ADAM Module	11
Figure 8 - Arduino IDE with the opened project.....	11

List of Tables

Table 1 – ADAM – Arduino Display Adaptor Module Connections.....	10
--	----

Appendix C – Revision History

Document Title: AN_280 4DLCD-FT843 SampleApp Arduino Introduction
Document Reference No.: BRT_000187
Clearance No.: BRT#107
Product Page: <http://brtchip.com/product/>
Document Feedback: [Send Feedback](#)

Revision	Changes	Date
1.0	First Release	2013-10-30
1.1	Document migrated from FTDI to BRT (Updated company logo; copyright info; contact information; hyperlinks)	2017-12-28