CE223726 – PSoC 6 TFT Display Interface with EmWin Graphics Library

Objective

This code example shows how to control a TFT display using EmWin Graphics Library in PSoC® 6 MCU.

Overview

This code example demonstrates how to display graphics on a TFT display using EmWin Graphics Display Library. EmWin graphics library implements 2D graphics and provides easy-to-use API functions to display text, 2D graphics (lines, rectangles, circles, etc.), and bitmap images. In PSoC Creator™, EmWin graphics library is implemented as a PDL middleware library.

This code example assumes that you are familiar with the PSoC 6 MCU and the PSoC Creator Integrated Design Environment (IDE). If you are new to PSoC 6 MCU, see the application note AN210781 – Getting Started with PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity.

For details of EmWin Graphics Library API, see the EmWin documentation *UM03001_emWin5.pdf* in the *Program Files (x86)\Cypress\PDL\3.x.x\doc\* folder.

Requirements

Tool: PSoC Creator 4.2; Peripheral Driver Library (PDL) 3.0.4

Programming Language: C (Arm® GCC 5.4.1)

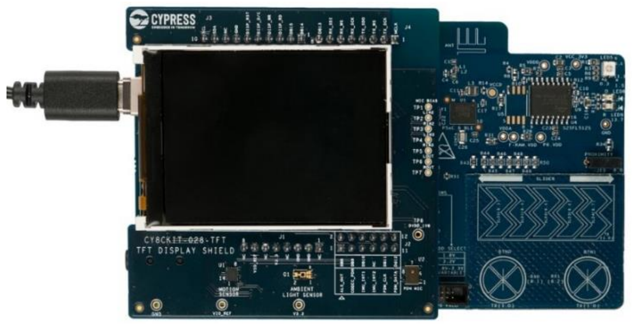
Associated Parts: All PSoC 6 MCUs

Related Hardware: CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit, CY8CKIT-028-TFT TFT Shield

Hardware Setup

1. Plug in the TFT display shield on to the Pioneer Board as Figure 1 shows.

Figure 1. Hardware Setup



2. Set the switches and jumpers on the Pioneer Board as shown in Table 1.

Table 1. Switch and Jumper Selection

Switch/Jumper	Position	Location
SW5	3.3 V	Front
SW6	PSoC 6 BLE	Back
SW7	V_{DDD} / KitProg2	Back
J8	Installed	Back

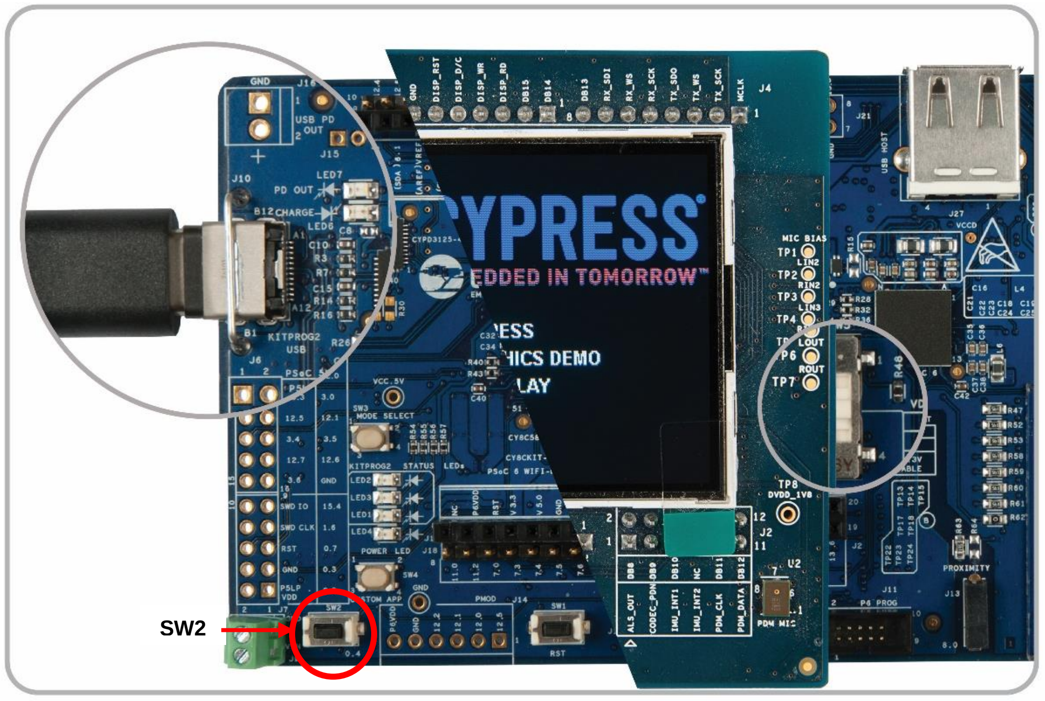
Software Setup

Install the CY8CKIT-62-BLE PSoC 6 BLE Pioneer Kit software, which contains all the required software to evaluate this code example. No additional software setup is required.

Operation

1. Connect the Pioneer Board to your PC using the provided USB cable through the USB connector (J10).

Figure 2. Connecting the USB Cable to the Pioneer Board



2. Program the Pioneer Board with the CE23726_EmWin_TFT_Display_ST7789 project. See the CY8CKIT-062-BLE kit guide for details on how to program firmware into the device.

   The TFT display shows the startup screen for three seconds, followed by a screen that displays instructions to press SW2 to scroll through various demo pages. Press SW2 to advance through the following pages that demonstrate various graphics features in EmWin.

   - Text alignments, styles, and modes
   - Text colors
   - Normal fonts of various sizes
   - Bold fonts of various sizes
   - Color bars
   - 2D graphics with vertical lines, horizontal lines, arcs, and rectangles
   - 2D graphics with concentric circles and ellipses, and rectangles with gradient fills
   - Concentric circles
   - Bitmap image

   Figure 3 shows the startup and instructions screens. Figure 4 shows all the screens that are shown in sequence.

Figure 3. Startup Screen and Instructions Screen

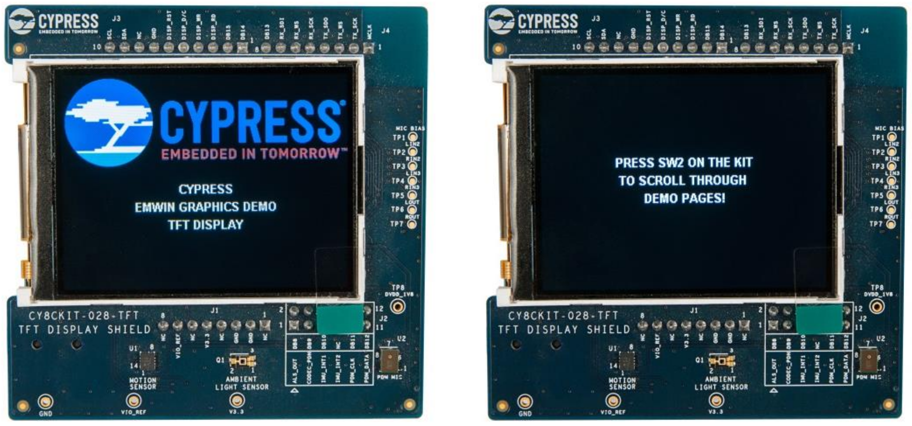
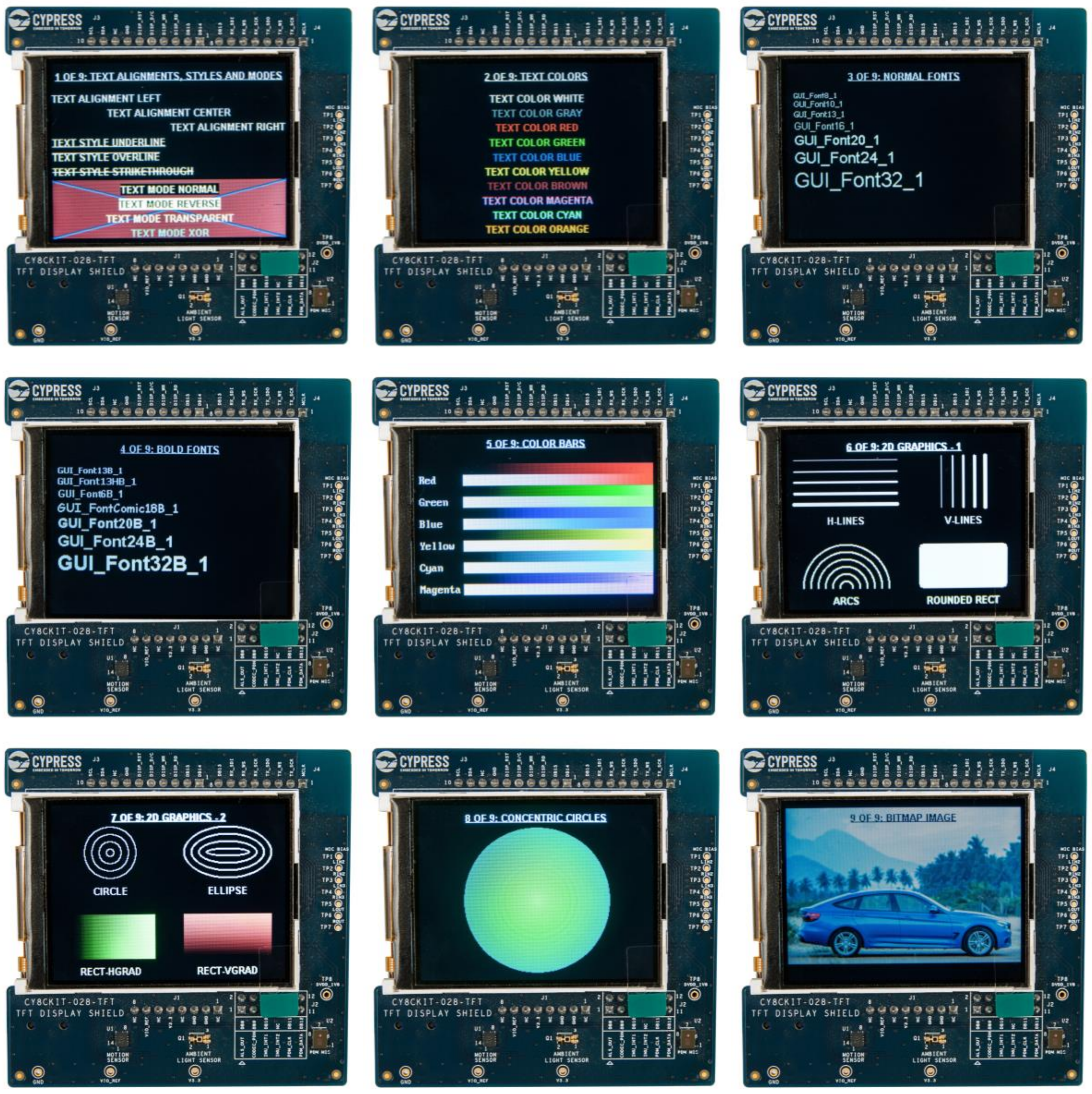


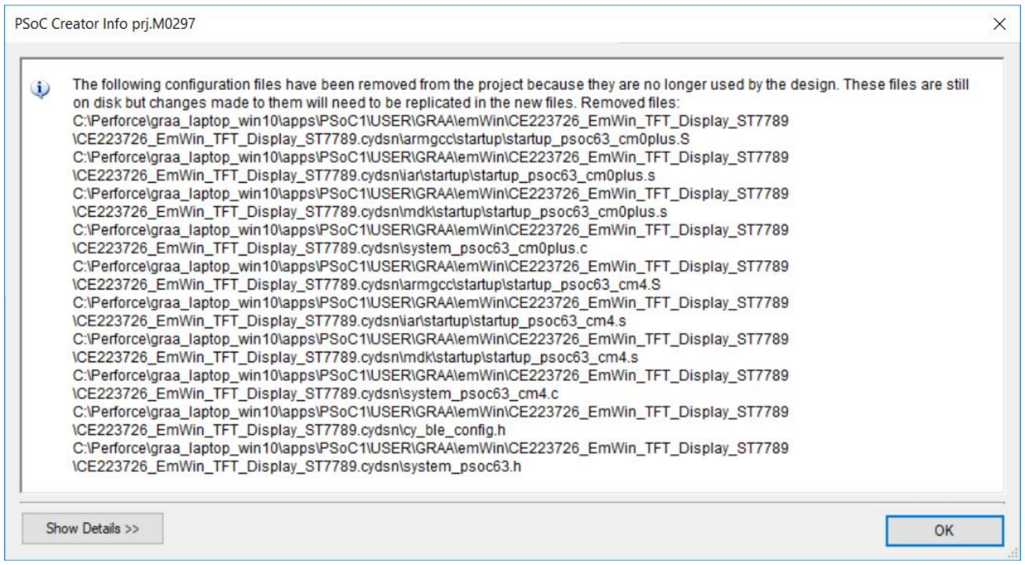
Figure 4. Pages Shown in Sequence



Important Note:

When you build the project, you will see the following notification. Click **OK**.

Figure 5. Build Messages



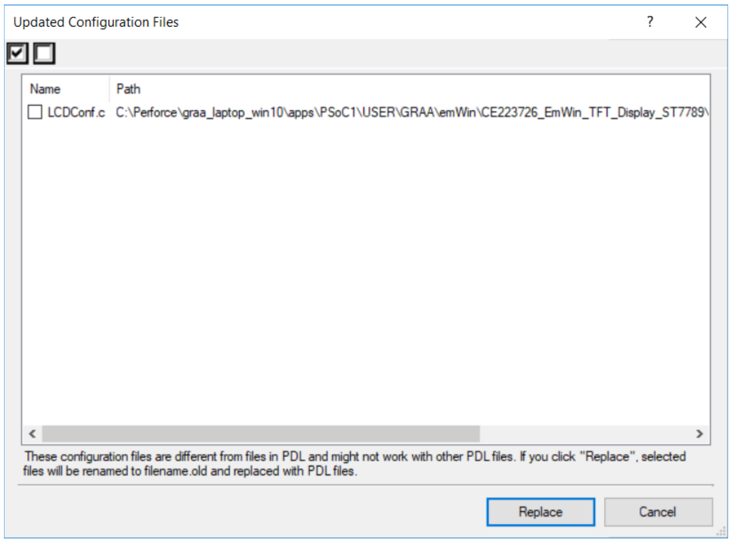
PSoC Creator Info prj.M0297

The following configuration files have been removed from the project because they are no longer used by the design. These files are still on disk but changes made to them will need to be replicated in the new files. Removed files:

```
C:\Perforce\graa_laptop_win10\apps\PSoC1\USER\GRAA\emWin\CE223726_EmWin_TFT_Display_ST7789
\CE223726_EmWin_TFT_Display_ST7789.cydsn\armgcc\startup\startup_psoc63_cm0plus.S
C:\Perforce\graa_laptop_win10\apps\PSoC1\USER\GRAA\emWin\CE223726_EmWin_TFT_Display_ST7789
\CE223726_EmWin_TFT_Display_ST7789.cydsn\iar\startup\startup_psoc63_cm0plus.s
C:\Perforce\graa_laptop_win10\apps\PSoC1\USER\GRAA\emWin\CE223726_EmWin_TFT_Display_ST7789
\CE223726_EmWin_TFT_Display_ST7789.cydsn\mdk\startup\startup_psoc63_cm0plus.s
C:\Perforce\graa_laptop_win10\apps\PSoC1\USER\GRAA\emWin\CE223726_EmWin_TFT_Display_ST7789
\CE223726_EmWin_TFT_Display_ST7789.cydsn\system_psoc63_cm0plus.c
C:\Perforce\graa_laptop_win10\apps\PSoC1\USER\GRAA\emWin\CE223726_EmWin_TFT_Display_ST7789
\CE223726_EmWin_TFT_Display_ST7789.cydsn\armgcc\startup\startup_psoc63_cm4.S
C:\Perforce\graa_laptop_win10\apps\PSoC1\USER\GRAA\emWin\CE223726_EmWin_TFT_Display_ST7789
\CE223726_EmWin_TFT_Display_ST7789.cydsn\iar\startup\startup_psoc63_cm4.s
C:\Perforce\graa_laptop_win10\apps\PSoC1\USER\GRAA\emWin\CE223726_EmWin_TFT_Display_ST7789
\CE223726_EmWin_TFT_Display_ST7789.cydsn\mdk\startup\startup_psoc63_cm4.s
C:\Perforce\graa_laptop_win10\apps\PSoC1\USER\GRAA\emWin\CE223726_EmWin_TFT_Display_ST7789
\CE223726_EmWin_TFT_Display_ST7789.cydsn\system_psoc63_cm4.c
C:\Perforce\graa_laptop_win10\apps\PSoC1\USER\GRAA\emWin\CE223726_EmWin_TFT_Display_ST7789
\CE223726_EmWin_TFT_Display_ST7789.cydsn\cy_ble_config.h
C:\Perforce\graa_laptop_win10\apps\PSoC1\USER\GRAA\emWin\CE223726_EmWin_TFT_Display_ST7789
\CE223726_EmWin_TFT_Display_ST7789.cydsn\system_psoc63.h
```

After this, a notification is displayed.

Figure 6. Updated Configuration Files



Updated Configuration Files

Name	Path
☐ LCDConf.c	C:\Perforce\graa_laptop_win10\apps\PSoC1\USER\GRAA\emWin\CE223726_EmWin_TFT_Display_ST7789\

These configuration files are different from files in PDL and might not work with other PDL files. If you click "Replace", selected files will be renamed to filename.old and replaced with PDL files.

Deselect the file (it is deselected by default), and click **Replace**. If you select the file and click **Replace**, the configuration file in the code example project will be replaced by the default configuration file.

Design and Implementation

This project uses a CY8CKIT-028-TFT TFT Display Shield together with the CY8CKIT-062-BLE Pioneer Board. The TFT shield has a 320x240 resolution and has a Sitronix ST7789 display controller.

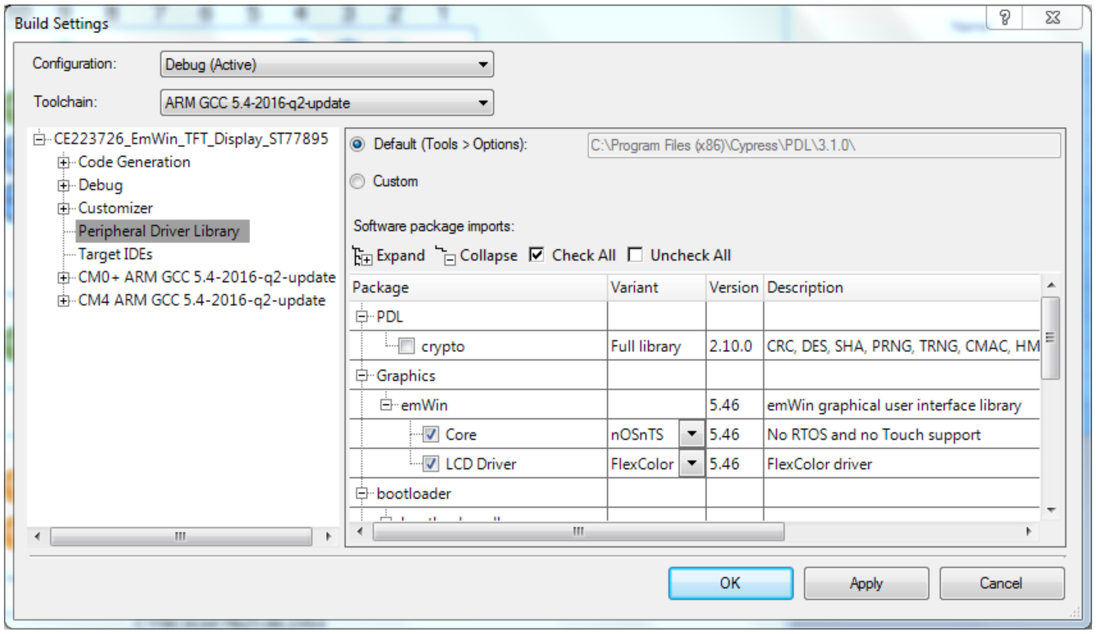
There are two important parts in this code example:

1. **EmWin Graphics Library:** The EmWin Graphics Library is implemented as a middleware in PDL and implements all graphics functions. The library manages a display buffer and updates this display buffer with pixel data according to graphics operations performed. The library also communicates with the display driver using the 8-bit parallel interface implemented using the GraphicLCDIntf Component, and the LCD access routines defined in the *LCDConf.c* file.
2. **Application Code:** The application code calls the EmWin graphics APIs to perform various graphics functions.

Include and Configure EmWin Graphics Library

1. In PSoC Creator, go to **Project** > **Build Settings,** and select **Peripheral Driver Library**. Under the **Graphics** > **emWin** section, select the **Core** and **LCD Driver** options.

Figure 7. Build Settings



Build Settings

Configuration: Debug (Active)

Toolchain: ARM GCC 5.4-2016-q2-update

- CE223726_EmWin_TFT_Display_ST77895
  - Code Generation
  - Debug
  - Customizer
  - Peripheral Driver Library
  - Target IDEs
  - CM0+ ARM GCC 5.4-2016-q2-update
  - CM4 ARM GCC 5.4-2016-q2-update

Default (Tools > Options): C:\Program Files (x86)\Cypress\PDL\3.1.0\

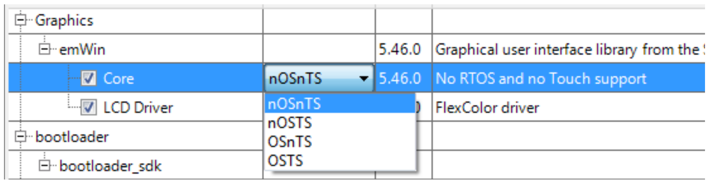
Custom

Software package imports:

Package	Variant	Version	Description
PDL			
crypto	Full library	2.10.0	CRC, DES, SHA, PRNG, TRNG, CMAC, HM
Graphics			
emWin		5.46	emWin graphical user interface library
☑ Core	nOSnTS	5.46	No RTOS and no Touch support
☑ LCD Driver	FlexColor	5.46	FlexColor driver
bootloader			

2. Select the **nOSnTS** option for **Core** because this project does not use RTOS or Touch support.

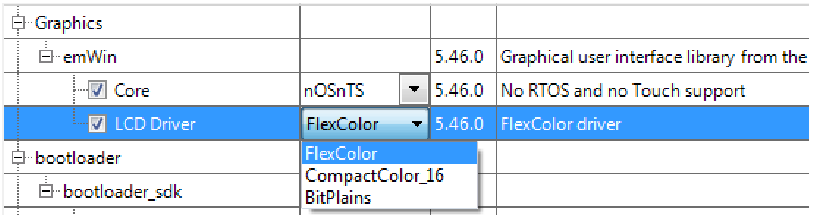
Figure 8. Core Option



Graphics			
emWin		5.46.0	Graphical user interface library from the
☑ Core	nOSnTS	5.46.0	No RTOS and no Touch support
☑ LCD Driver			FlexColor driver
bootloader			
bootloader_sdk			

3. Select "FlexColor" option for the "LCD Driver" parameter. This driver supports the ST7789 display controller.

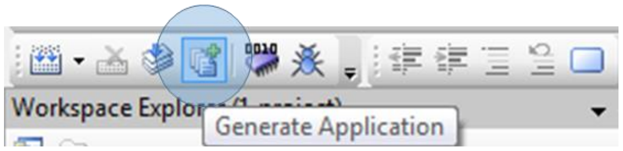
Figure 9. Select LCD Driver Options



Graphics			
emWin		5.46.0	Graphical user interface library from the
☑ Core	nOSnTS	5.46.0	No RTOS and no Touch support
☑ LCD Driver	FlexColor	5.46.0	FlexColor driver
bootloader			
bootloader_sdk			

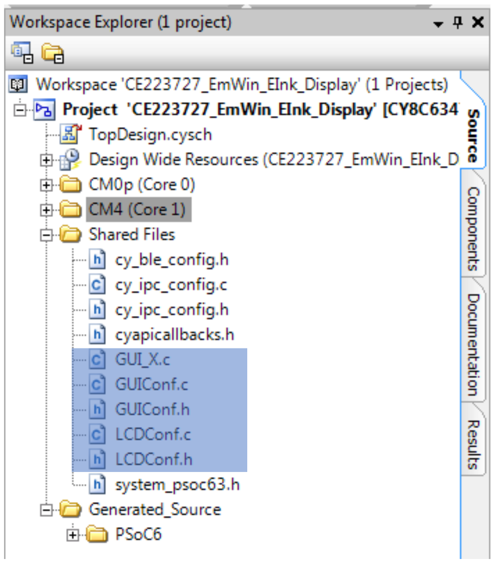
4. Click **Generate Application**.

Figure 10. Generate Application



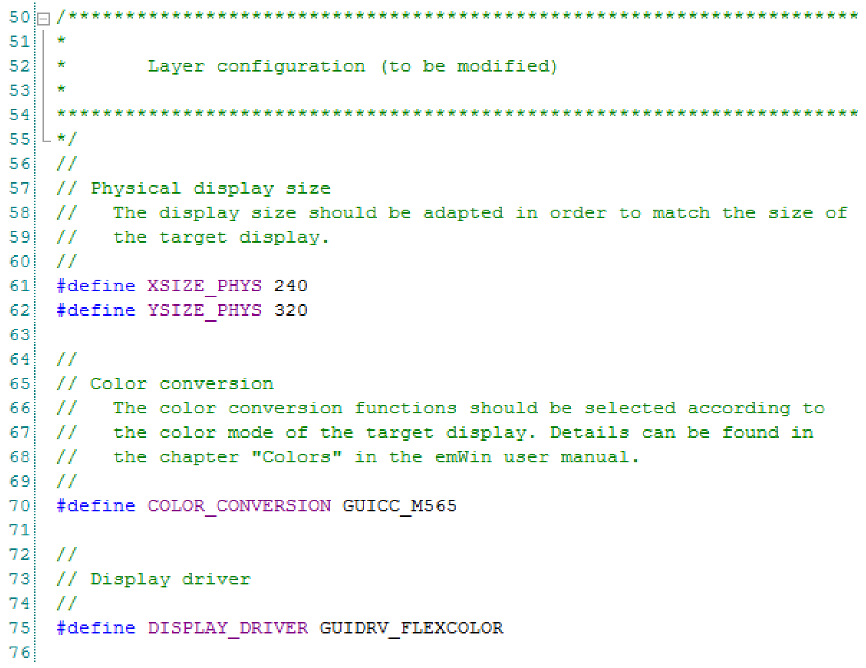
PSoC Creator generates the configuration files for EmWin under the *Shared Files* folder as shown in Figure 11.

Figure 11. Configuration Files Generated



5. Open the *LCDConf.c* file and configure the X and Y sizes of the display, the color palette, and display driver.

Figure 12. Setting the X and Y Values and Color Conversion



```
/*********************************************************************
*
*       Layer configuration (to be modified)
*
**********************************************************************
*/
//
// Physical display size
//   The display size should be adapted in order to match the size of
//   the target display.
//
#define XSIZE_PHYS 240
#define YSIZE_PHYS 320

//
// Color conversion
//   The color conversion functions should be selected according to
//   the color mode of the target display. Details can be found in
//   the chapter "Colors" in the emWin user manual.
//
#define COLOR_CONVERSION GUICC_M565

//
// Display driver
//
#define DISPLAY_DRIVER GUIDRV_FLEXCOLOR

```

The ST7789S controller has a resolution of 240x320 pixels.

This controller supports various color palettes. This project uses the 16 bits per pixel RGB 5-6-5 color palette (5 bits R, 6 bits G, 6 bits G). Per the EmWin user guide Section 15.6 “Fixed Palette Modes”, this color palette is named GUICC_M565.

The ST7789S controller is supported by the FlexColor driver; therefore, set the driver to **GUI_FLEXCOLOR**.

6. In the *LCDConf.c* file, write code in the `_InitController` function to initialize the hardware interface and display controller. The EmWin library calls this function during the initialization stage. Some code snippets from this function are shown below.

```
        /* Start the parallel interface */
        GraphicLCDIntf_1_Start();

        /* Reset - High, Low (reset), High */
        Cy_GPIO_Set(LCD_RESET_N_0_PORT, LCD_RESET_N_0_NUM);
        GUI_Delay(20);
        Cy_GPIO_Clr(LCD_RESET_N_0_PORT, LCD_RESET_N_0_NUM);
        GUI_Delay(100);
        Cy_GPIO_Set(LCD_RESET_N_0_PORT, LCD_RESET_N_0_NUM);
        GUI_Delay(100);
    ….

    GraphicLCDIntf_1_Write8_A0(0x28);
    GraphicLCDIntf_1_Write8_A0(0x11); //Exit Sleep mode
    GUI_Delay(100);
    GraphicLCDIntf_1_Write8_A0(0x36);
    GraphicLCDIntf_1_Write8_A1(0xA0);//MADCTL: memory data access control
    GraphicLCDIntf_1_Write8_A0(0x3A);
    GraphicLCDIntf_1_Write8_A1(0x65);//COLMOD: Interface Pixel format
    ….
```

If a different display controller is used, this function must be modified to send the appropriate command and data bytes as defined in the controller data sheet.

7. In the *LCDConf.c* file, the `LCD_X_Config` function configures various parameters required by the driver. Some important parts of this function are shown below.

   The `GUI_DEVICE_CreatAndLink` function sets the display driver and color palette.

```
GUI_DEVICE * pDevice;
pDevice = GUI_DEVICE_CreateAndLink(DISPLAY_DRIVER, COLOR_CONVERSION, 0, 0);
```

   The configuration structure is updated with the display orientation; the `GUIDRV_FlexColor_Config` function sets up the display driver. The ST7789S controller has a resolution of 240x320. However, the display used in the TFT shield has a resolution of 320x240. Because of this swapping of X and Y resolutions, the orientation is set to swap X and Y axes and mirror the Y axis.

```
CONFIG_FLEXCOLOR Config = {0};
Config.Orientation    = GUI_MIRROR_Y | GUI_SWAP_XY;
GUIDRV_FlexColor_Config(pDevice, &Config);
```

   The PortAPI structure sets the pointers to the APIs that perform read and write through the GraphicLCDIntf parallel interface. The `GUIDRV_FlexColor_SetFunc` function is called to configure the Port APIs, specific display controller, and interface.

```
GUI_PORT_API PortAPI = {0};

PortAPI.pfWrite8_A0  = GraphicLCDIntf_1_Write8_A0;
PortAPI.pfWrite8_A1  = GraphicLCDIntf_1_Write8_A1;
PortAPI.pfWriteM8_A1 = GraphicLCDIntf_1_WriteM8_A1;
PortAPI.pfRead8_A1  = GraphicLCDIntf_1_Read8_A1;
PortAPI.pfReadM8_A1  = GraphicLCDIntf_1_ReadM8_A1;
GUIDRV_FlexColor_SetFunc(pDevice, &PortAPI, GUIDRV_FLEXCOLOR_F66709, GUIDRV_FLEXCOLOR_M16C0B8);
```

   See Section 33.7.4 “GUIDRV_FlexColor” in the EmWin user guide for details.

8. Open the *GUIConf.c* file. This file manages the RAM allocation for the EmWin library. The value of the macro `GUI_NUMBYTES` macro must be set according to the approximate memory requirement based on EmWin library features used by the application. This will depend on the various options enabled in the *GUIConf.h* file. See Section 37.2, “Memory Requirements” in the EmWin user guide for details on the memory usage for various features. For this code example, the memory size has been set to an arbitrary 0x1000 bytes. You can set this to a smaller value based on the features used in your application.

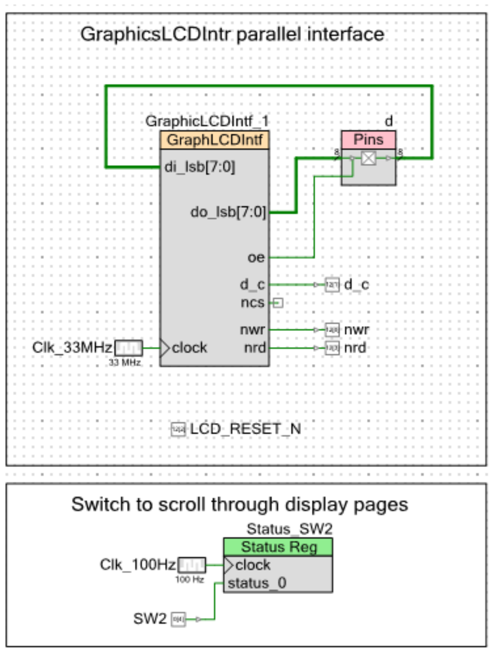
9. The *GUI_X.c* file has timing functions used by the EmWin library. The content of this file varies based on the OS support selected. No modifications are required in this file for this code example.

Main Application:

Hardware:

Figure 13 shows the schematics of the main application.

Figure 13. Schematic

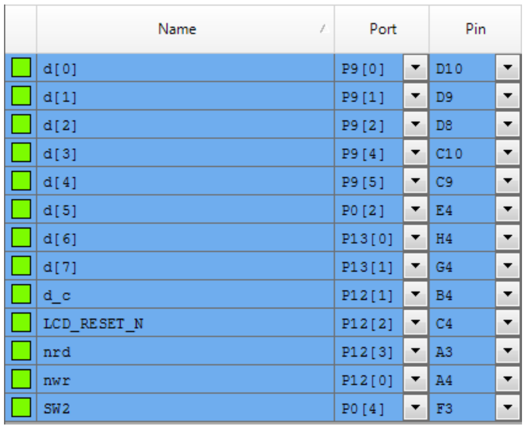


The GraphicLCDIntf Component is used to implement the 8-bit parallel interface to communicate with the display controller. The PortAPI structure of the EmWin driver is updated with the APIs from this Component.

SW2 is a digital input that is connected to SW2 on the CY8CKIT-062-BLE kit. This pin is configured as Resistive Pull Up. SW2 is connected to a Status Register to read the switch status. A 100-Hz clock is connected to the clock input of the Status Register which also provides a 10-ms debounce.

Figure 14 shows the port pin configuration for GraphicLCDIntf and SW2.

Figure 14. Port Pin Configuration for GraphicLCDIntf and SW2



	Name	Port	Pin
🟩	d[0]	P9[0]	D10
🟩	d[1]	P9[1]	D9
🟩	d[2]	P9[2]	D8
🟩	d[3]	P9[4]	C10
🟩	d[4]	P9[5]	C9
🟩	d[5]	P0[2]	E4
🟩	d[6]	P13[0]	H4
🟩	d[7]	P13[1]	G4
🟩	d_c	P12[1]	B4
🟩	LCD_RESET_N	P12[2]	C4
🟩	nrd	P12[3]	A3
🟩	nwr	P12[0]	A4
🟩	SW2	P0[4]	F3

Firmware:

The main application is implemented in the *main_cm4.c* file. The following functions are performed in main:

1. Initializing the EmWin graphics engine
2. Displaying the startup screen

3. Displaying the instructions screen that prompts the user to press SW2 to scroll through various display pages
4. Displaying the following pages in an infinite loop. After displaying each page, waiting for a press and release event of SW2.
   a. Text alignments, styles, and modes
   b. Text colors
   c. Normal fonts
   d. Bold fonts
   e. Color bars
   f. 2D graphics with horizontal lines, vertical lines, arcs, and filled rectangle
   g. 2D graphics with concentric circles, concentric ellipses, rectangles with horizontal, and vertical gradient fills
   h. Concentric circles with color gradient
   i. Bitmap image with overlaid text

Components

Component	Instance Name	Function
GraphicLCDIntf	GraphicLCDIntf_1	8-bit parallel interface for the LCD driver
Digital Output Pin	LCD_RESET_N	This GPIO is used to reset the display controller.
Digital Input Pin	SW2	This pin is connected to an external switch SW2 on the CY8CKIT-062-BLE board.
Status Register	Status_SW2	This status register is used to read SW2 status.

See the PSoC Creator project for more details on PSoC Component configurations and design-wide resource settings.

Related Documents

Application Notes	
AN210781 – Getting Started with PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity	Describes PSoC 6 MCU with BLE Connectivity devices and how to build your first PSoC Creator project
AN215656 – PSoC 6 MCU: Dual-CPU system Design	Describes the dual-CPU architecture in PSoC 6 MCU, and shows how to build a simple dual-CPU design
AN219434 – Importing PSoC Creator Code into an IDE for a PSoC 6 MCU Project	Describes how to import the code generated by PSoC Creator into your preferred IDE
PSoC Creator Component Datasheets	
Pins	Supports connection of hardware resources to physical pins
Clock	Supports local clock generation
Status Register	Can be used to read the state of digital signals
Graphic LCD Interface	Implements 8 or 16 bit i8080 parallel interface
Device Documentation	
PSoC 6 MCU: PSoC 63 with BLE Datasheet	PSoC 6 MCU: PSoC 63 with BLE Architecture Technical Reference Manual
Development Kit Documentation	
CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit	
Training Videos	
PSoC 6 101: Lesson 1-4 FreeRTOS	

Document History

Document Title: CE223726 – PSoC 6 TFT Display Interface with EmWin Graphics Library

Document Number: 002-23726

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	6299269	GRAA	09/18/2018	New code example

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

Arm® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6 MCU

Cypress Developer Community

Community | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709