Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

CE23727 – PSoC 6 E-INK Display Interface with EmWin Graphics Library

Objective

This code example shows how to control an E-INK display using the EmWin Graphics Library in PSoC® 6 MCU.

Overview

This code example demonstrates how to display graphics on an E-INK display using the EmWin Graphics Display Library. The 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™, the EmWin Graphics library is implemented as a PDL middleware library. E-INK displays consume no power for image retention. Together with PSoC 6 MCU and EmWin graphics library, an E-INK display can be used to create user interfaces that have the "always-on" functionality.

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.

Details of EmWin Graphics Library API can be found in 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)

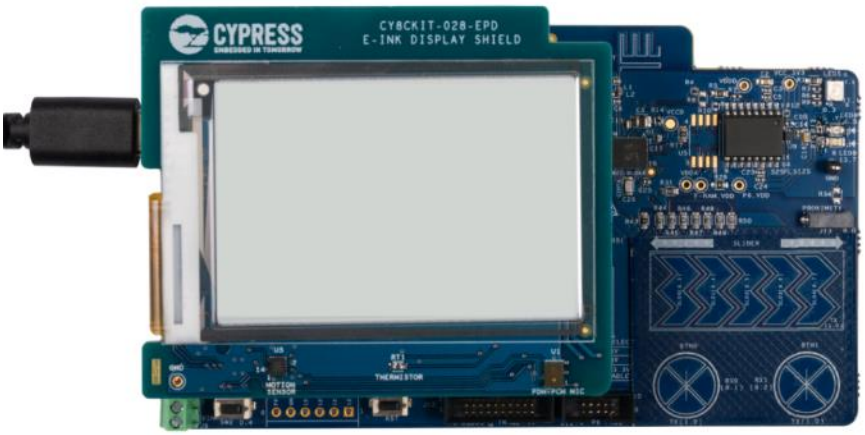
Associated Parts: All PSoC 6 MCUs

Related Hardware: CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit

Hardware Setup

1. Plug in the E-INK 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

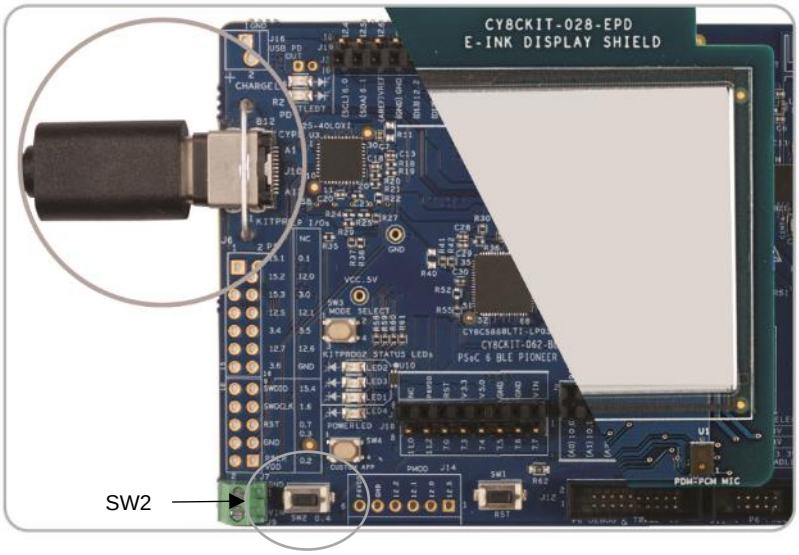
Software Setup

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 'CE23727_EmWin_Eink_Display' project. See the CY8CKIT-062-BLE kit guide for details on how to program firmware into the device.

   The E-INK display refreshes and 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.

   - Normal fonts of various sizes
   - Bold fonts of various sizes
   - Text alignments, modes, and styles
   - Text wrap and text rotation
   - 2D graphics with vertical lines, horizontal lines, arcs, and rectangle
   - 2D graphics with circles and ellipses

   Note that it takes about a second to refresh the display with new content.

Figure 3. Startup Screen

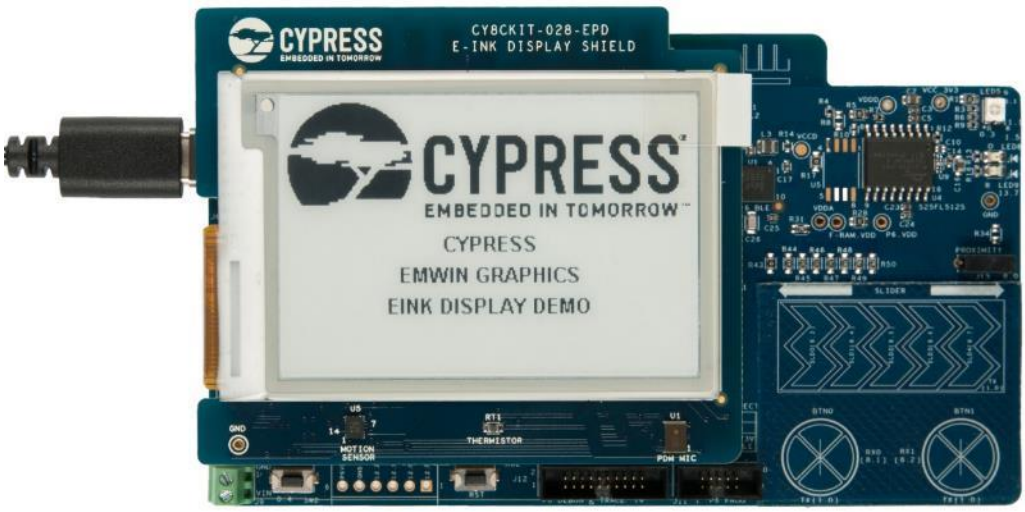
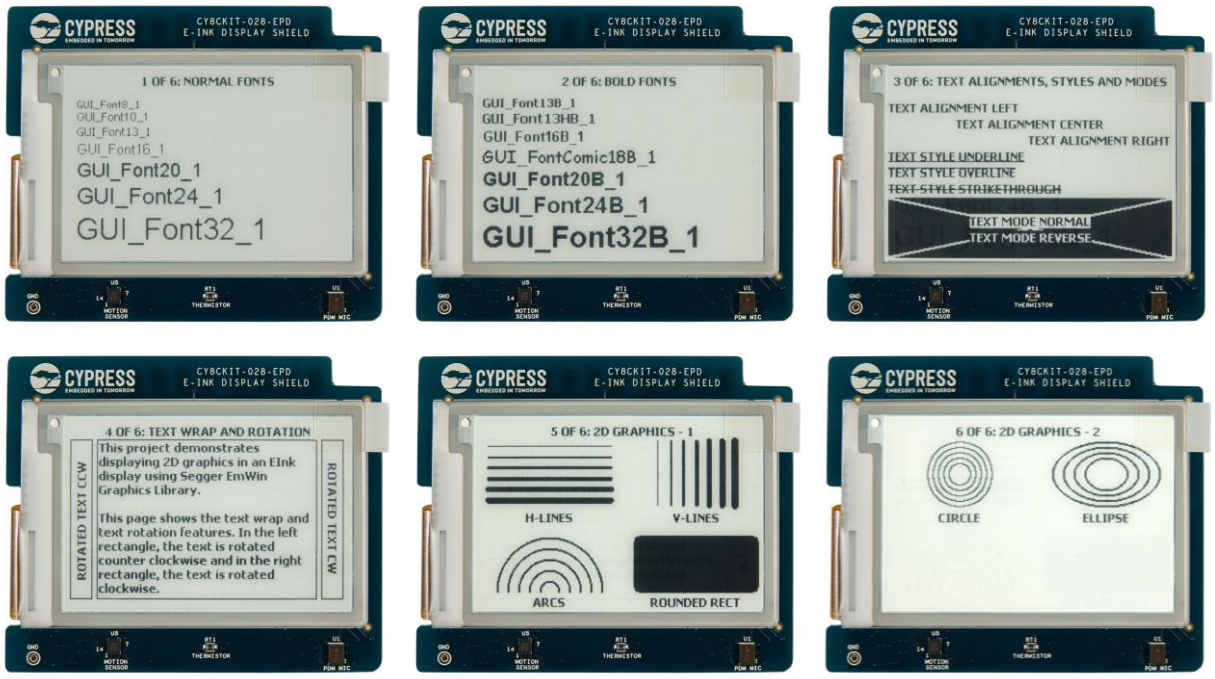


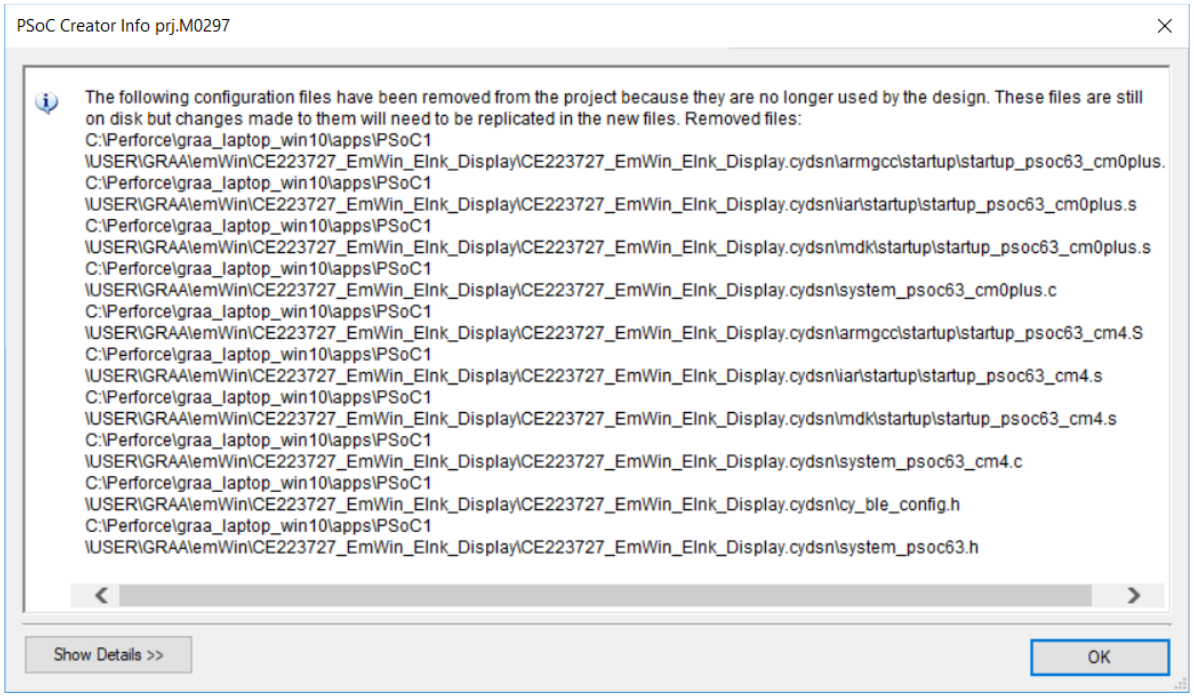
Figure 4. Pages Shown in Sequence



Important Note:

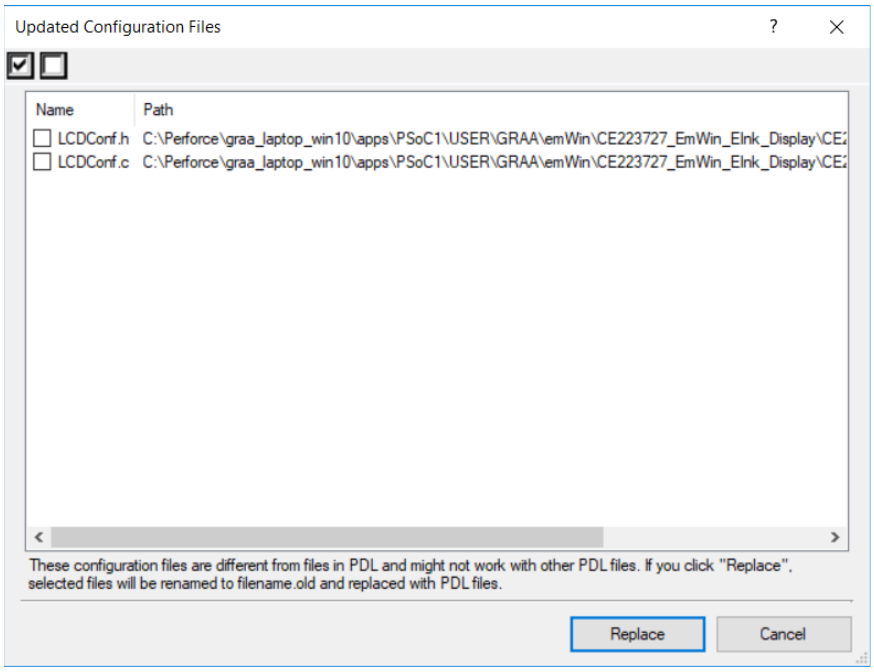
When you build the project, you will see the following notification. Click **OK**.

Figure 5. Build Messages



After this, the following notification is displayed:

Figure 6. Build Messages Showing Updated Configuration Files



Deselect both the files (they will be deselected by default) and click **Replace**. If you select these files and click **Replace**, configuration files in the code example project will be replaced by the default configuration files.

Design and Implementation

This project uses a CY8CKIT-028-EPD E-INK Display Shield together with CY8CKIT-062-BLE Pioneer Board. The E-INK Shield has a 2.7-inch E-INK display with a resolution of 264×176 pixels.

For details on the Pioneer Board and E-INK Display Shield, see the Pioneer Kit Guide.

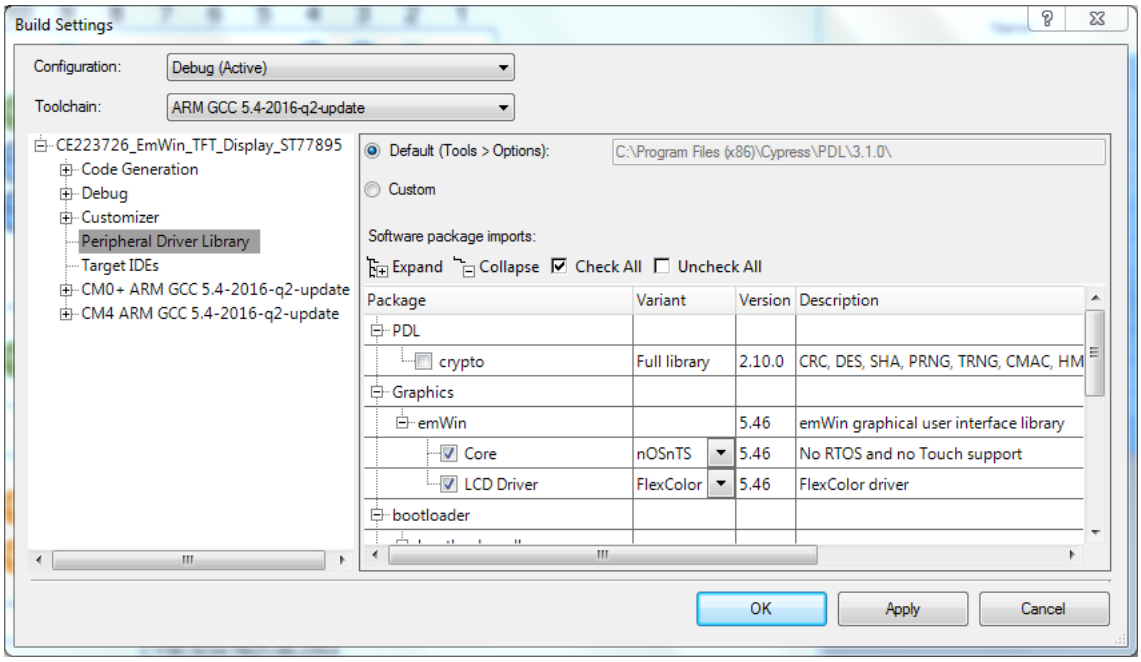
There are three important parts in this code example:

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.
2. **E-INK Driver:** The E-INK driver takes care of transferring the pixel data from the display buffer to the E-INK driver IC through a SPI Master interface and several GPIOs.
3. **Application Code:** The application code calls EmWin graphics APIs to perform graphic functions, manages an application display buffer with cache (needed by the E-INK library), and calls the E-INK display library to update the display.

Include and Configure EmWin Graphics Library

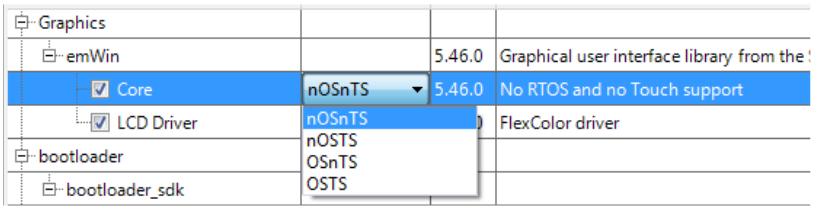
1. In PSoC Creator, go to **Project** > **Build Settings** and select **Peripheral Driver Library**. Under the **Graphics** > **emWin** section, select the **Core** and **LCD Driver** options.

Figure 7. Set Core and LCD Driver Options for the Project



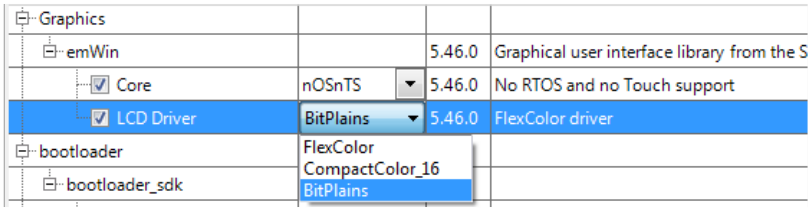
2. Select the **nOSnTS** option for **Core** because this project does not use RTOS or Touch support.

Figure 8. Core Option



3. Select the **BitPlains** option for the **LCD Driver** parameter. With the BitPlains driver, EmWin library manages only the graphics display buffer in memory. This driver can support color profiles from 1 bit per pixel (1bpp) to 8 bits per pixel (8bpp). As the E-INK supports Black/White, this project uses the 1bpp color profile.

Figure 9. Select Color Profile



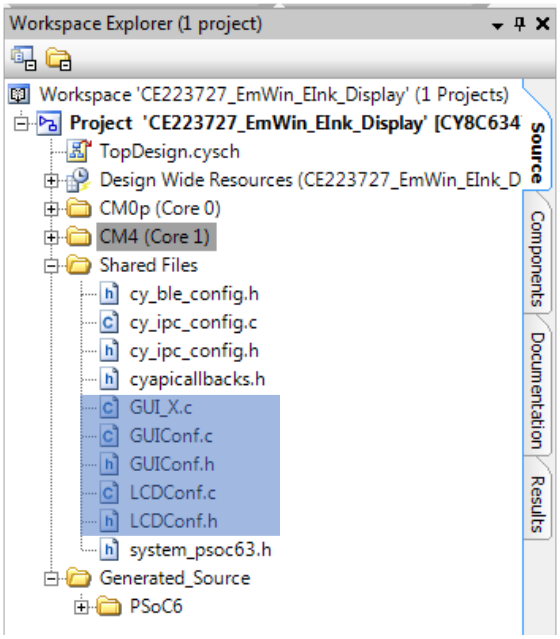
4. Click **Generate Application**.

Figure 10. Generate Application



PSoC Creator generates the configuration files for EmWin under the *Shared Files* folder.

Figure 11. Configuration Files Generated



5. Open the *LCDConf.c* file and configure the X and Y size of the display and color conversion. The E-INK display used in the E-INK shield has a resolution of 264x176 pixels. GUICC_1 sets the color profile to 1bpp. See the EmWin user guide for details of the color profiles.

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

```
/*********************************************
*
*       Layer configuration
*
*********************************************
*/
//
// Physical display size
//
#define XSIZE_PHYS 264
#define YSIZE_PHYS 176

//
// Initial color conversion API
//
#define COLOR_CONVERSION GUICC_1

```

EmWin allocates the display buffer based on the defined X and Y sizes.

6. Open the *GUIConf.c* file. This file manages the RAM allocation for EmWin. The value of the `GUI_NUMBYTES` macro must be set according to the approximate memory requirement based on EmWin features used by the application. 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 value of 0x1000 bytes.

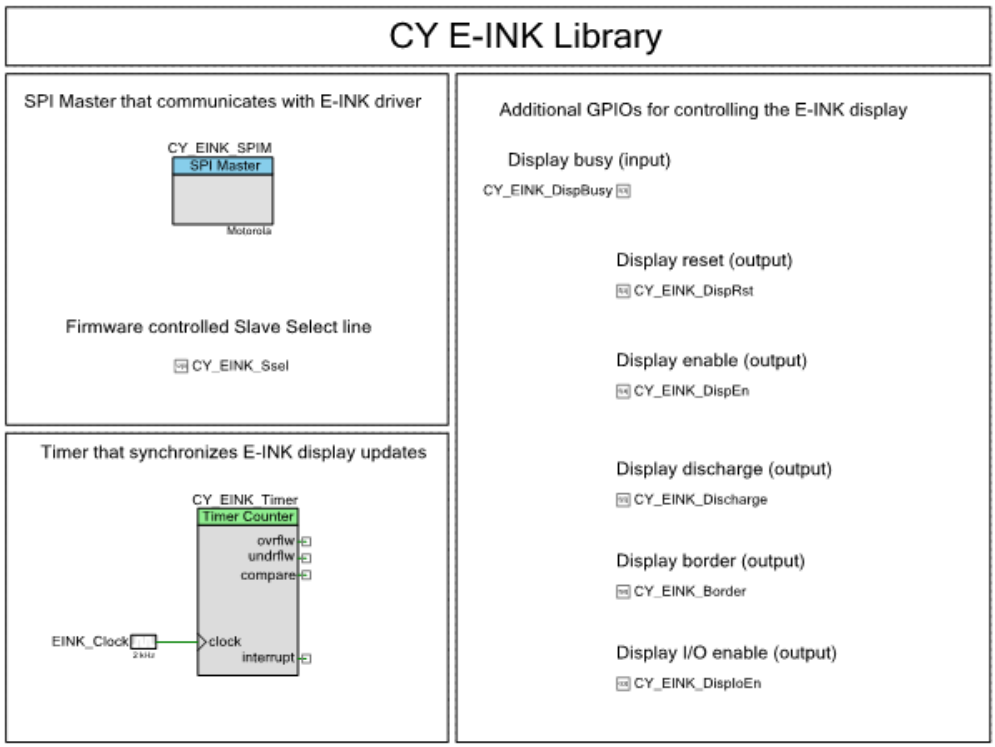
   The *GUI_X.c* file has timing functions used by EmWin. The content of this file varies based on the OS support selected. No modifications are required in this file for this code example.

CY E-INK Driver

This code example contains the required library functions for driving the E-INK display. However, the actual hardware driver functions are not covered in this document. See the E-INK display driver document for more details.

Figure 13 shows the PSoC Creator schematic that implements the hardware required for the E-INK display library.

Figure 13. TopDesign Schematic: E-INK Library



PSoC 6 MCU controls the E-INK display's reset, enable, discharge, and border pins. PSoC 6 MCU also reads the status of the display to determine whether the display is busy with a previous operation. A load switch on CY8CKIT-028-EPD, which is controlled by the PSoC 6 MCU device, can be used to turn the display ON/OFF. A voltage level translator is connected between the E-INK display and PSoC 6 MCU GPIOs so that PSoC 6 MCU can operate with variable V_{DD}. The enable input of the voltage level translator is also connected to a PSoC 6 MCU GPIO so that PSoC 6 MCU can disable the level translator to reduce power consumption when the E-INK display is not used.

CY_EINK_SPIM implements a SPI Master interface using which the PSoC 6 communicates with the E-INK controller. CY_EINK_Ssel is the GPIO that implements the Slave Select signal for the E-INK controller.

CY_EINK_Timer implements a 1-ms timer that is used for timing functions for the E-INK display function.

E-INK Library and Driver Files:

The following files implement the E-INK driver:

- *cy_eink_library.c/.h* files contain the E-INK library functions and macros.
- *pervasive_eink_configuration.h* file contains definitions of register indexes and hardware parameters of the E-INK display provided by the display vendor.
- *pervasive_eink_hardware_driver.c/.h* files contain low-level display hardware driver functions provided by the display vendor.
- *cy_eink_psoc_interface.c/.h* files contain the PSoC 6 MCU Component-level interface to the display hardware.

Note: Do not edit these files because it may cause an undesirable operation of the E-INK display.

See CE218133 – PSoC 6 MCU E-INK Display with CapSense® for details of the APIs in the E-INK display library.

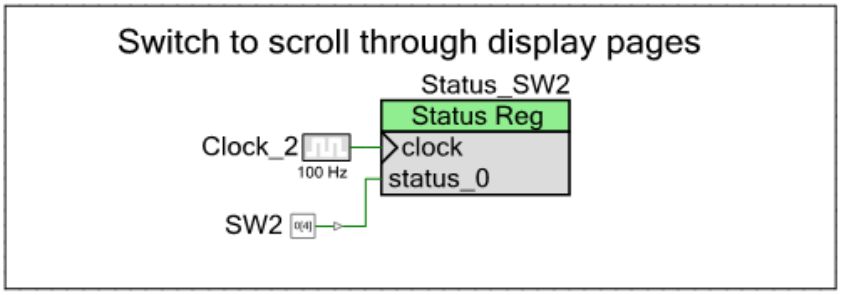
Main Application:

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. Initializing the E-INK driver
3. Displaying the startup screen
4. Displaying the instructions screen that prompts the user to press SW2 to scroll through various display pages.
5. In an infinite loop, displaying the following pages; after displaying each page, waiting for a press and release event of SW2.
    a. Displaying normal fonts
    b. Displaying bold fonts
    c. Displaying various text modes and alignment
    d. Displaying word wrap and rotation
    e. Displaying 2D graphics screen #1 that shows vertical lines, horizontal lines, arcs and filled rectangle
    f. Displaying 2D graphics screen #2 that shows concentric circles and concentric ellipses.

Figure 14 shows the hardware to read SW2. SW2 is connected to a Status Register Component which is clocked at 100 Hz. The Status Register provides the debounced state of the switch.

Figure 14. Switch Debounce and Status



Components

Table 2. List of PSoC Creator Components

Component	Instance Name	Function
SPI (SCB)	CY_EINK_SPIM	The SPI Component is configured as a SPI master that communicates with the E-INK display driver.
Timer Counter (TCPWM)	CY_EINK_Timer	The Timer Counter is configured to have 1LSB = 1 ms. The count value is used for E-INK display timing.
Digital Output Pin	CY_EINK_Ssel CY_EINK_DispRst CY_EINK_DispEn CY_EINK_Discharge CY_EINK _Border CY_EINK _DispIoEn	These GPIOs are configured as firmware-controlled output pins that are used to provide control signals to the E-INK display.
Digital Input Pin	CY_EINK _DispBusy	This GPIO is a digital input without any hardware connection. It is used to read the status of the E-INK display.
	SW2	This Digital Input Pin is connected to the input of the flip-flop to read the SW2 status.
Status Register	Status_SW2	This status register is used to read the status of the switch. A clock of 100 Hz is used for the Status Register, which also acts as a debounce to the switch.

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
CE218133 – MCU E-INK Display with CapSense	Describes how to implement a E-INK display solution with CapSense®
PSoC Creator Component Datasheets	
Pins	Supports connection of hardware resources to physical pins
Timer Counter (TCPWM)	Supports fixed-function Timer/Counter implementation
Clock	Supports local clock generation
Interrupt	Supports generating interrupts from hardware signals
CapSense	Supports touch sensing
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: CE23727 – PSoC 6 E-INK Display Interface with EmWin Graphics Library

Document Number: 002-23727

Revision	ECN	Submission Date	Description of Change
**	6299270	09/18/2018	New code example
*A	6634885	08/05/2019	Updated the CE project 1. Increased the SPI data rate from 8.33Mbps to 20Mbps 2. Instead of application code maintaining two buffers (current and cache), changed application code to use only one buffer for cache. emWin's display buffer is directly used as current buffer in EInk page update function calls 3. Minor code cleanup

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 | Code Examples | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

All other trademarks or registered trademarks referenced herein are the property of their respective owners.

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

© Cypress Semiconductor Corporation, 2018-2019. This document is the property of Cypress Semiconductor Corporation and its subsidiaries ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. No computing device can be absolutely secure. Therefore, despite security measures implemented in Cypress hardware or software products, Cypress shall have no liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. CYPRESS DOES NOT REPRESENT, WARRANT, OR GUARANTEE THAT CYPRESS PRODUCTS, OR SYSTEMS CREATED USING CYPRESS PRODUCTS, WILL BE FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (collectively, "Security Breach"). Cypress disclaims any liability relating to any Security Breach, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from any Security Breach. In addition, the products described in these materials may contain design defects or errors known as errata which may cause the product to deviate from published specifications. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. "High-Risk Device" means any device or system whose failure could cause personal injury, death, or property damage. Examples of High-Risk Devices are weapons, nuclear installations, surgical implants, and other medical devices. "Critical Component" means any component of a High-Risk Device whose failure to perform can be reasonably expected to cause, directly or indirectly, the failure of the High-Risk Device, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from any use of a Cypress product as a Critical Component in a High-Risk Device. You shall indemnify and hold Cypress, its directors, officers, employees, agents, affiliates, distributors, and assigns harmless from and against all claims, costs, damages, and expenses, arising out of any claim, including claims for product liability, personal injury or death, or property damage arising from any use of a Cypress product as a Critical Component in a High-Risk Device. Cypress products are not intended or authorized for use as a Critical Component in any High-Risk Device except to the limited extent that (i) Cypress's published data sheet for the product explicitly states Cypress has qualified the product for use in a specific High-Risk Device, or (ii) Cypress has given you advance written authorization to use the product as a Critical Component in the specific High-Risk Device and you have signed a separate indemnification agreement.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.