CE218133 - PSoC 6 MCU E-INK Display with CapSense

Objective

This code example shows how to create a user-interface solution using an E-INK display and CapSense®.

Overview

This code example demonstrates how to create a user-interface solution using an E-INK display with a CapSense slider and buttons. E-INK displays consume no power for image retention. However, during a display update, the CPU must be active for as long as a second, which consumes CPU cycles and increases average power consumption. PSoC® 6 MCU has an Arm® Cortex®-M0+ low-power core that can alleviate these concerns because it consumes very low power and takes processing overhead away from the main ARM Cortex-M4 CPU. Together with PSoC 6 MCU's CapSense touch sensing, an E-INK display can be used to create user interfaces that have "always-on" functionality. In fact, in this example, the CM0+ is used for all user interface operations—the CM4 core is not used at all.

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, you can find an introduction in the application note AN210781 – Getting Started with PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity.

Requirements

Tool: PSoC Creator 4.2

Programming Language: C (Arm GCC 5.4.1)

Associated Parts: All PSoC 6 MCUs

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

Design

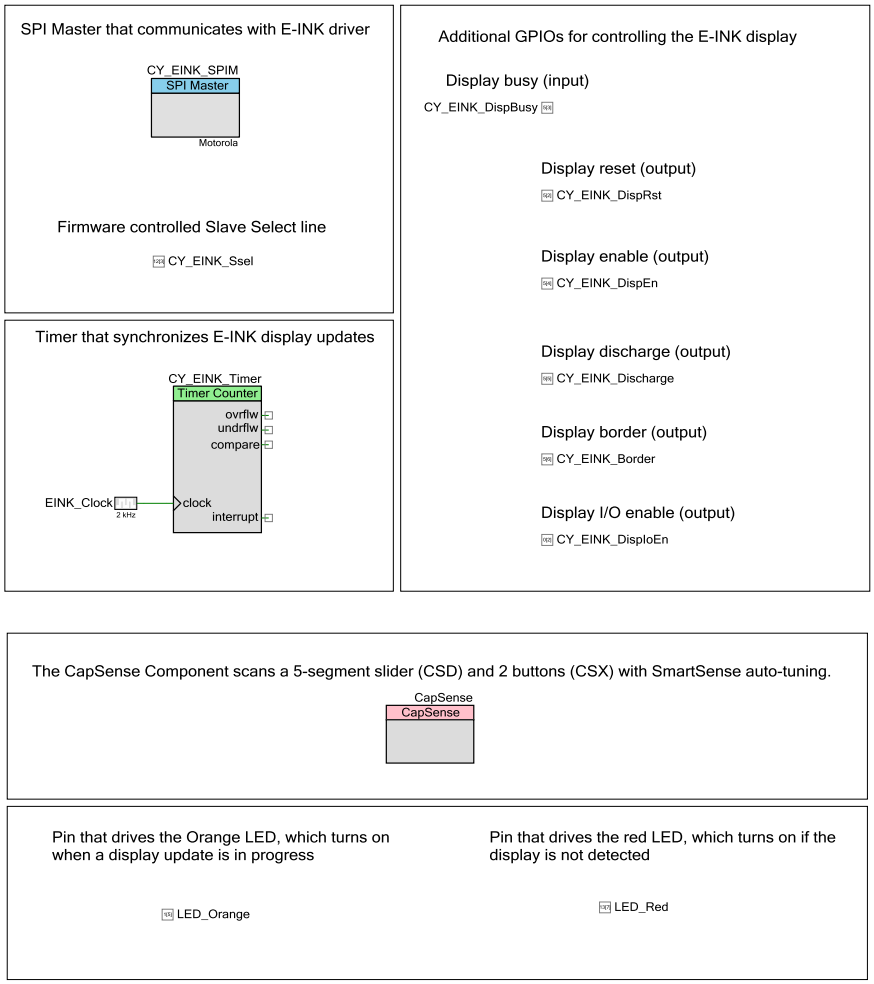
E-INK (electronic ink) is a paper-like display technology, characterized by high contrast, wide viewing angles, and minimal standby power. Unlike conventional backlit, flat panel displays that emit light, E-INK displays reflect light like paper. This makes E-INK displays more comfortable to read, and provides a wider viewing angle than most light-emitting displays. Therefore, E-INK displays are comfortable to read even in sunlight.

This project uses a CY8CKIT-028-EPD E-INK Display Shield together with a 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.

Figure 1 shows the PSoC Creator schematic of this code example.

Figure 1. Schematic Design



The E-INK display in CY8CKIT-028-EPD contains a basic driver IC that interfaces with the PSoC 6 MCU using a custom SPI interface. The driver converts a serial data stream into individual pixel data and generates the voltages required for the E-INK display. PSoC 6 MCU has low-level control of the E-INK display.

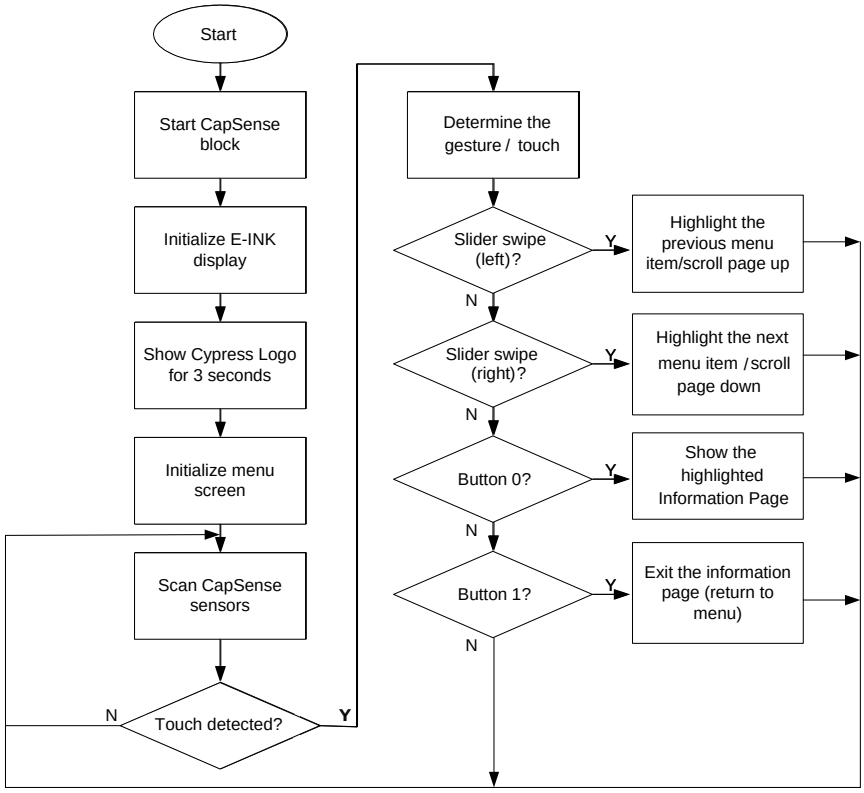
To minimize power consumption, this code example is implemented entirely on the CM0+ core.

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.

The 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.

Figure 2 shows the firmware flow of this project.

Figure 2. Firmware Flow



In this project, PSoC 6 MCU scans a CapSense slider and two buttons for user input. Based on the user input, the E-INK display is updated to scroll through menu items to change information pages, and to move back and forth between the top-level menu and information pages as Figure 2 shows.

The project consists of the following files:

- The *main_cm0p.c* file contains the main function, which is the entry point for execution of the firmware application. The main function contains the routines to initialize the system and a loop that reads touch information and updates the screen accordingly.
- *touch.c/h* files contain functions to initialize CapSense touch sensing and read touch data from buttons and sliders.
- *screen.c/h* files contain functions that update the screen according to a touch input.
- *display.c/h* files provide an adaptation layer between the *screen.c/h* module and the low-level E-INK library.
- *screen_contents.c/.h* files constitute storage files for the images and text displayed on the screen. See Appendix A for a description of the image format.

E-INK Library and Driver Files:

- *cy_eink_library.c/.h* files contain the E-INK library functions and macros.
- *cy_eink_fonts.c/h* files contain the font information used for text to pixel data conversion.
- The *pervasive_eink_configuration.h* file contains display-vendor-provided definitions of register indexes and hardware parameters of the E-INK display.
- *pervasive_eink_hardware_driver.c/.h* files contain display-vendor-provided low-level display hardware driver functions.
- *cy_eink_psoc_interface.c/.h* files contains the PSoC 6 MCU Component-level interface to the display hardware.

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

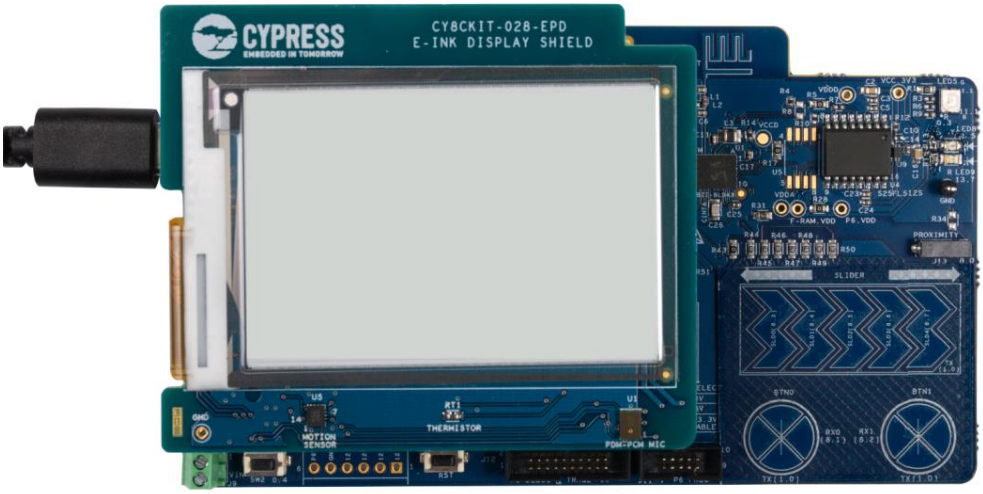
Hardware Setup

Set the switches and jumpers on the Pioneer Board as shown in Table 1 and plug in the E-INK Display Shield to Pioneer Board.

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

Figure 3. Hardware Setup



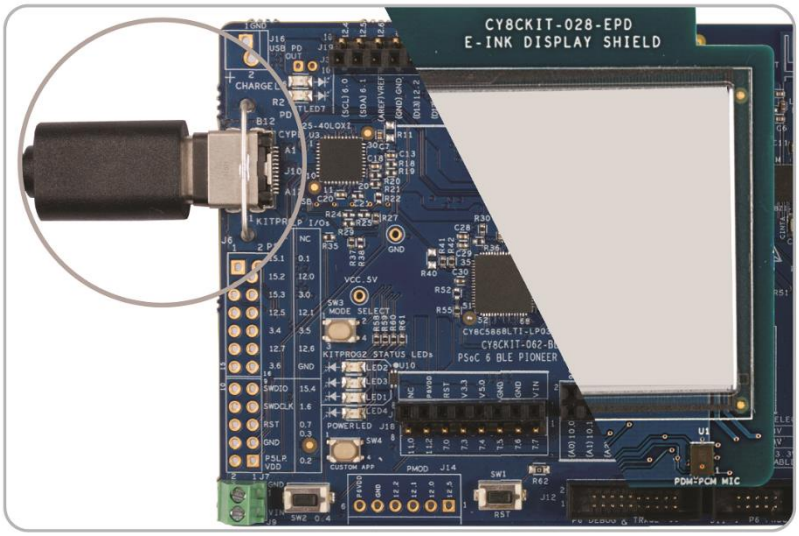
Software Setup

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 4. Connecting the USB Cable to the Pioneer Board



2. Program the Pioneer Board with the CE218133_EINK_CapSense 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 3 seconds, followed by a menu that lists important information about the kit and associated software, as Figure 5 shows. LED9 (Red) turns ON if the E-INK display is not detected. In this case, check the connection between the E-INK Display Shield and the Pioneer Board, and then reset the Pioneer Board.

3. Use the CapSense slider and buttons to navigate the menu, as Figure 6 shows.

   Note that the display takes about a second to refresh the display following a touch input. LED8 (Orange) turns ON when the display is busy. Touch inputs are not processed when the display is busy. Because the main menu uses partial update for faster refreshes (for details, see the `cy_eink_update_t` parameter of the `Cy_EINK_ShowFrame` function in Appendix A, the selection arrow may have slight ghosting.

Figure 5. Main Menu

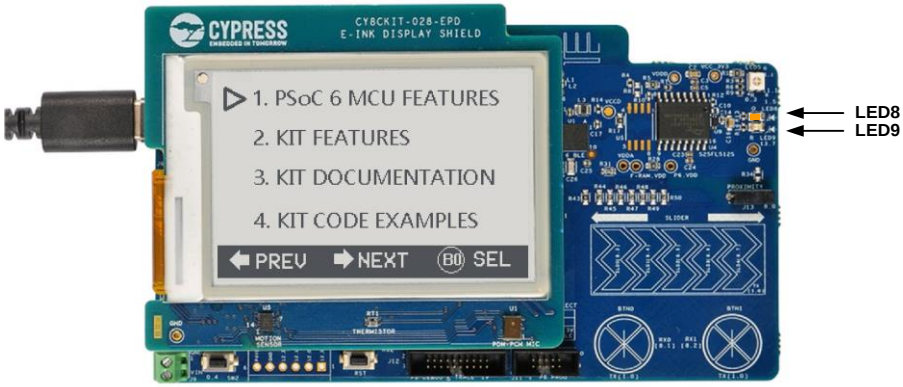
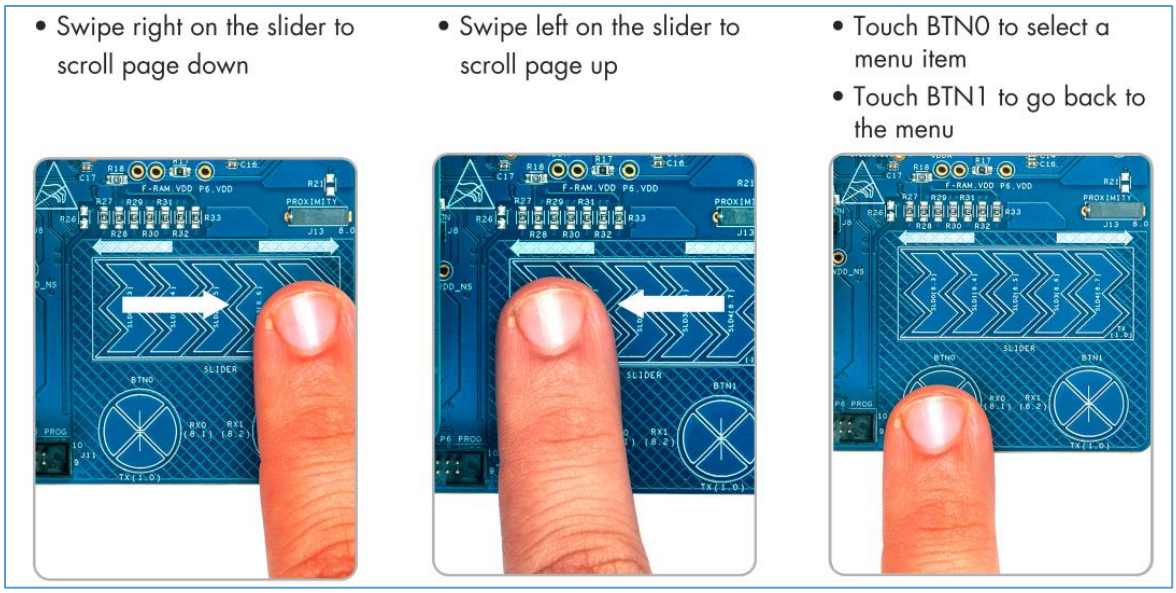


Figure 6. Menu Navigation Options



Components

Table 2. List of PSoC Creator Components

Component	Instance Name	Function
CapSense	CapSense	The CapSense Component is configured to scan a 5-segment slider (CSD) and 2 buttons (CSX) with SmartSense auto-tuning
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.
	LED_Red LED_Orange	These GPIOs are configured as firmware-controlled output pin that control red (**LED9**) and orange (**LED8**) status LEDs.
Digital Input Pin	CY_EINK _DispBusy	This GPIO is a digital input without any hardware connection. It is used to read the status of E-INK display.

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 63 with Bluetooth Low Energy (BLE) Connectivity and how to build your first PSoC Creator project
PSoC Creator Component Datasheets	
Bluetooth Low Energy	Facilitates designing applications requiring BLE connectivity
Device Documentation	
PSoC 6 MCU: PSoC 63 with BLE Datasheet	PSoC 6 MCU: PSoC 63 with BLE Architecture Technical Reference Manual
Development Kit (DVK) Documentation	
CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit	

A E-INK Display Driver Library

This section describes the functions provided in the E-INK display driver library. These functions are in the *cy_eink_Library.c* file.

```
void Cy_EINK_Start (      int8_t     temperature
                    )
```

Initializes the E-INK display hardware and PSoC Components

Parameters

temperature	Ambient temperature in degree Celsius

Returns

None

Note

After initialization of the E-INK hardware, this function turns OFF the power to the display.

```
bool Cy_EINK_Power (      bool       powerCtrl
                   )
```

Turns ON/OFF the power to the E-INK display and initializes the E-INK driver

Parameters

powerCtrl	false – power OFF, true – power ON

Returns

false – driver initialization failed, true – driver initialization was successful

Note

Display contents will be retained even after the display power has been turned OFF.

```
void  Cy_EINK_Clear (     bool       background,
                          bool       powerCycle
                     )
```

Clears the E-INK display to either white or black background

Parameters

background	false – black, true – white
powerCycle	false – does not control the power ON/OFF automatically
	true – automatically controls the power cycle. This function will turn ON the power, clear the display, and then turn the power OFF.

Returns

None

Note

If the `powerCycle` value is false, then E-INK display should be powered on (using the `Cy_EINK_Power` function) before calling this function. Otherwise, the display will not be cleared.

```
void    Cy_EINK_ShowFrame (     cy_eink_frame_t*        prevFrame
                                cy_eink_frame_t*        newFrame
                                cy_eink_update_t        updateType
                                bool                    powerCycle
                           )
```

Updates the display with a frame (image or pixel data stored in flash/RAM). See Appendix B for more information on the image and text formats stored as a frame.

Parameters

prevFrame	Pointer to the frame that is currently displayed on the E-INK display. A frame consists of 5808 bytes (264x176/8) of data in which each bit stores the pixel information of a monochromatic image.
newFrame	Pointer to the frame to be displayed
updateType	CY_EINK_PARTIAL – updates the display from the previous frame to the new frame without any intermediate stages. This is the fastest type of update (~0.4 seconds), however, may produce ghosting if the new frame differs considerably from the previous frame. CY_EINK_FULL_2STAGE – updates the display from the previous frame to the new frame with an intermediate stage that updates the display with the inverted version of the previous frame. This additional stage reduces ghosting, but increases the update time (~0.8 seconds). CY_EINK_FULL_4STAGE – updates the display from the previous frame to the new frame with three intermediate stages that consists inverted version of the previous frame, white frame, and inverted version of the new frame. This type of refresh produces minimal ghosting at the cost of having the longest update time (~1.6 seconds).
powerCycle	false – does not control the power ON/OFF automatically. true – automatically controls the power cycle. This function will turn ON the power, clear the display, and then turn the power OFF.

Returns

None

Note

If the powerCycle value is false, then EINK display should be powered ON (using the EINKCy_EINK_Power_Power function) before calling this function. Otherwise, the display will not be updated.

```
void Cy_EINK_TextToFrameBuffer (     cy_eink_frame_t*        frameBuffer
                                     char*                   string
                                     cy_eink_font_t*         fontInfo
                                     uint8_t*                fontCor
                                )
```

Converts a string of text into pixel data and writes to a frame buffer, which can be then displayed using the `Cy_EINK_ShowFrame` function.

Parameters

`frameBuffer`	Pointer to a frame buffer stored in RAM. A frame consists of 5808 bytes (264x176/8) of pixel data in which each bit stores the pixel information of a monochromatic image.
`string`	Pointer to a string
`fontInfo`	Pointer to a font information structure. The E-INK display driver library supports two constant-sized fonts: `CY_EINK_FONT_8X12BLACK` and `CY_EINK_FONT_16X16BLACK`. See Appendix B for details of these fonts
`fontCor`	Pointer to a two-byte array that stores coordinates starting at which the text needs to be written. Note that this array should point to text coordinates instead of pixel coordinates. See Appendix B for details.

Returns

None

Note

This function does not update the E-INK display. After frame buffer update, use the `Cy_EINK_ShowFrame` function to update the display if required.

```
void E-INK_ImageToFrameBuffer (     cy_eink_frame_t*      frameBuffer
                                    cy_eink_frame_t*      image
                                    uint8_t*              imgCoordinates
                               )
```

Crops an image at the specified coordinates and copies it to the same location in the frame buffer. See Appendix B for more information on the image format.

Parameters

`frameBuffer`	Pointer to a frame buffer stored in RAM. A frame consists of 5808 bytes (264x176/8) of pixel data in which each bit stores the pixel information of a monochromatic image.
`image`	Pointer to a monochromatic image stored in flash/RAM as an array of 5808 bytes. Image should have the same size and format as the frame buffer.
`imgCoordinates`	Pointer to a four-byte array that stores byte coordinates at which the image is cropped (including the final X and Y coordinates) before copying to the frame buffer. See Appendix B for details.

Returns

None

Note

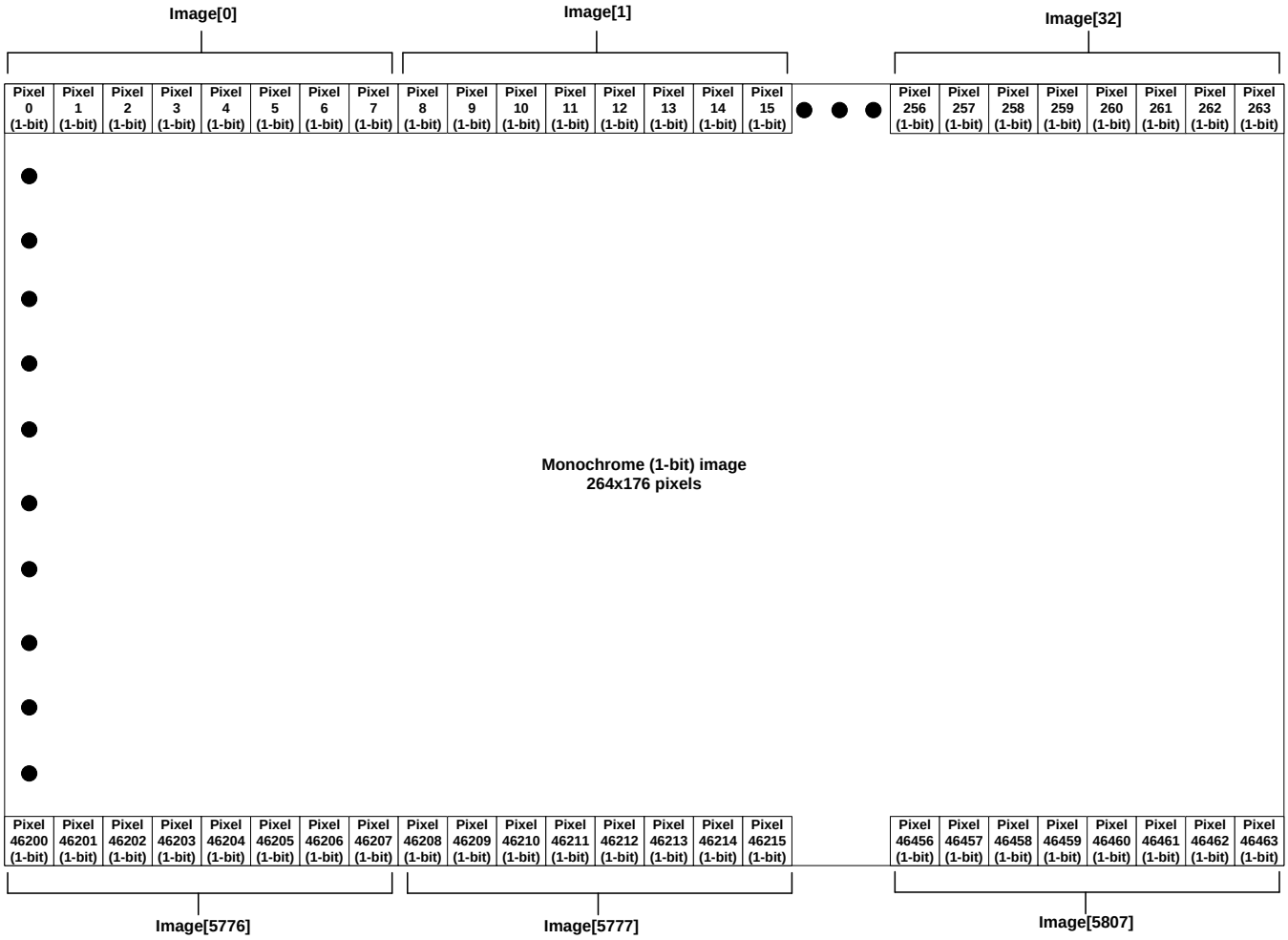
This function does not update the E-INK display. After the frame buffer update, use the `Cy_EINK_ShowFrame` function to update the display, if required.

B Image and Text Formats

Image and Frame Buffer Format

The E-INK display has a resolution of 264×176 pixels. The E-INK display library supports images and frame buffers stored as a uint8 array of size 5808 (264×176/8). Figure 7 shows how the pixel data is stored in an array image[5808].

Figure 7. Image and Frame Buffer Format



You can use the PDi Apps from the display manufacturer to create a variable array from a bitmap image.

B.1 Supported Fonts

The E-INK display driver library supports two constant-sized fonts: CY_EINK_FONT_8X12BLACK and CY_EINK_FONT_16X16BLACK. Figure 8 and Figure 9 show the format of these fonts.

Figure 8. CY_EINK_FONT_8X12BLACK Format

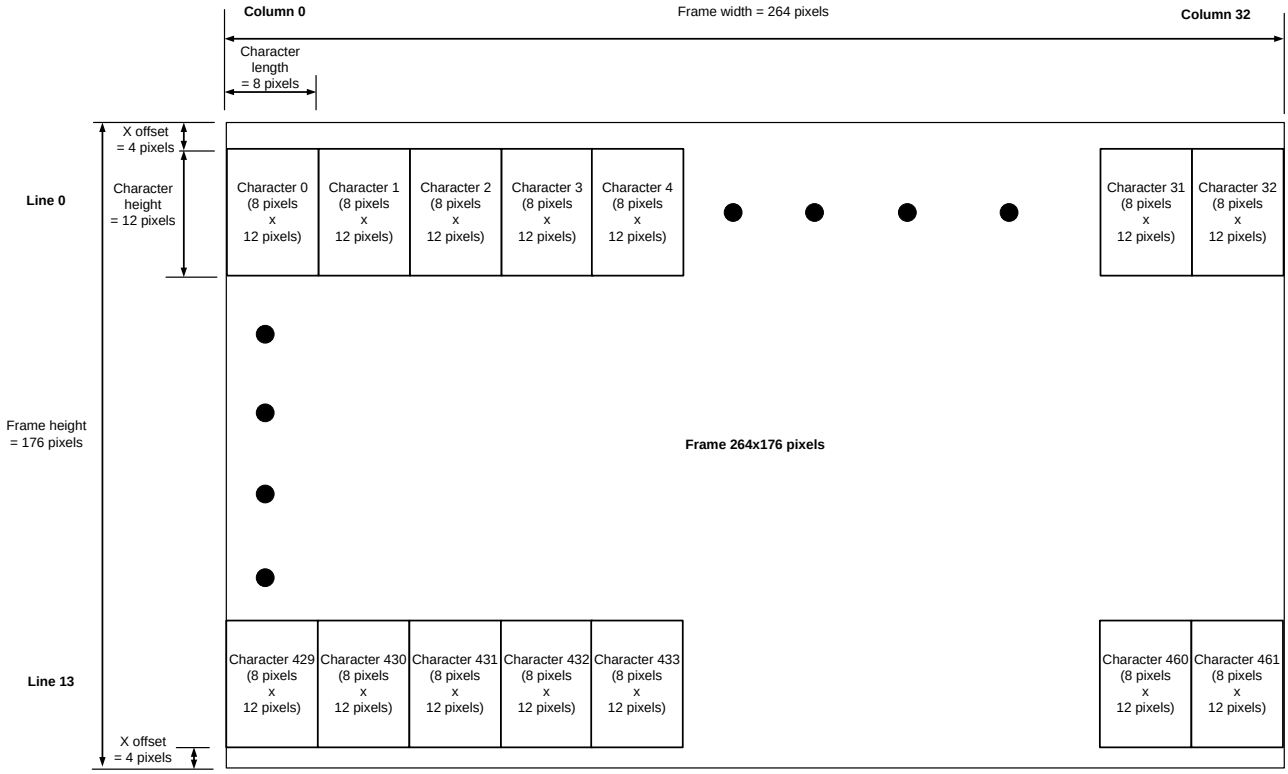
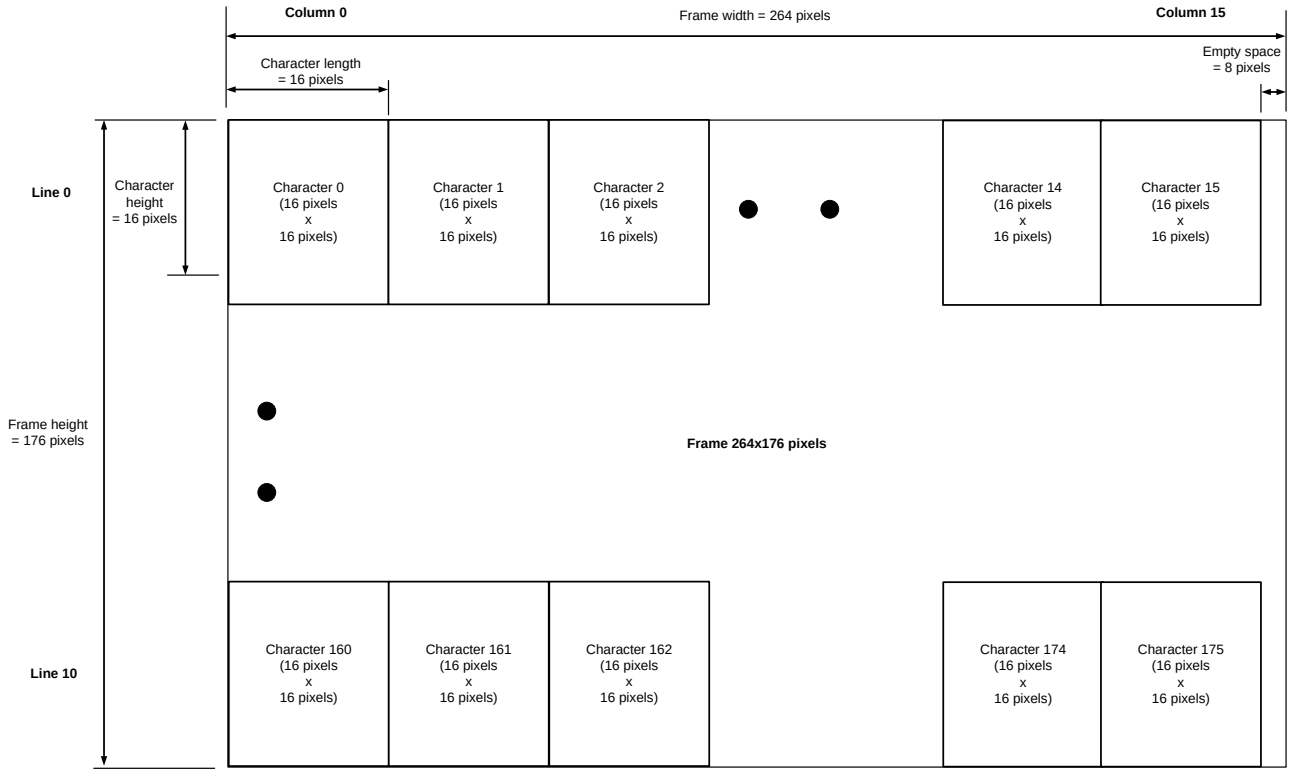


Figure 9. CY_EINK_FONT_16X16BLACK Format



Document History

Document Title: CE218133 – PSoC 6 MCU E-INK Display with CapSense

Document Number: 002-18133

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	5570337	NIDH	12/30/2016	New spec.
*A	5654432	NIDH	03/22/2017	Updated the Top Design, firmware flow, configuration, operation, and library functions. Updated template.
*B	5861714	NIDH	08/23/2017	Initial public release version
*C	6005867	NIDH	12/26/2017	Updated template and minor text changes. Updated project to PSoC Creator 4.2 Beta.

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

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

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