

### Лабораторна робота № 7

#### Вивчення E-INK дисплею стенду PSoC 6 BLE Pioneer Kit фірми Cypress

**Мета роботи:** Вивчити функціонування E-INK дисплею для відображення на ньому текстової та графічної інформації.

#### Теоретичні відомості

В цьому проекті розглянемо відображення графіки на дисплеї E-INK за допомогою бібліотеки графічних дисплеїв EmWin. Графічна бібліотека EmWin реалізує 2D графіку та забезпечує прості у використанні API функції для відображення тексту, 2D графіки (лінії, прямокутники, кола, т.п.) та растрові зображення. У PSoC Creator графічна бібліотека EmWin реалізована як PDL бібліотека середнього програмного забезпечення. Дисплеї E-INK не споживають енергії для збереження зображення. Разом з PSoC 6 MCU та графікою EmWin бібліотеки, дисплей E-INK може бути використаний для створення інтерфейсів користувача, які завжди відображають інформацію.

Підключіть екран дисплея E-INK до плати Pioneer Kit, як показано на рис. 8.1.

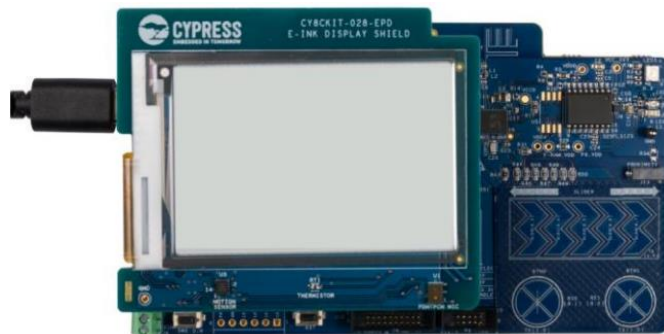


Рис. 8.1. Налаштування обладнання

Встановіть вимикачі та перемички на платі Pioneer, як показано в табл. 8.1.

Табл. 8.1. Вибір перемикачів та перемичок стенду PSoC 6 Pioneer Kit

Switch/Jumper	Position	Location
SW5	3.3 V	Front
SW6	PSoC 6 BLE	Back
SW7	V <sub>DDD</sub> /KitProg2	Back
J8	Installed	Back

1. Підключіть плату стенду до ПК за допомогою USB- кабелю через роз'єму USB (J10). На рис. 8.2 зображено підключення стенду до ПК.

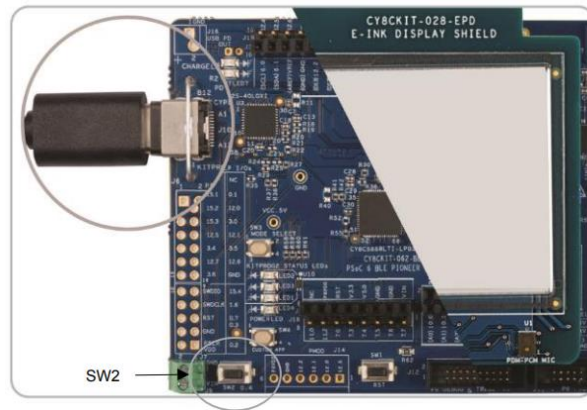


Рис. 8.2. Підключення USB-кабелю до плати стенду

2. Запрограмувати прошивку програми проекту "CE23727. EmWin E-Ink Display". Дисплей E-INK оновлюється та показує екран запуску протягом трьох секунд, а потім екран, на якому відображаються інструкції. Потрібно натиснути кнопку SW2, щоб прокрутити різні демонстраційні сторінки. Ще раз натиснути SW2, щоб перейти до наступних сторінок, які демонструють різні графічні функції в EmWin.

- Нормальні шрифти різного розміру.
- Жирні шрифти різного розміру.
- Вирівнювання тексту, режими та стилі.
- Обтікання тексту та обертання тексту.
- 2D графіка з вертикальними лініями, горизонтальними лініями, дугами та прямокутником.
- 2D графіка з колами та еліпсами.

Для оновлення дисплею новим вмістом потрібно близько секунди.

На рис. 8.3. зображено вигляд екрану при запуску проекту.

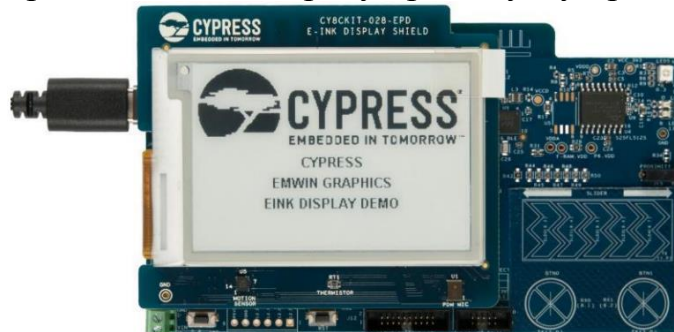


Рис. 8.3. Зображення екрану при запуску проекту



Рис. 8.4. Вигляд сторінок, які відображаються в послідовності проекту

Під час побудови проекту побачимо повідомлення, зображені на рис. 8.5. Натисніть кнопку ОК.

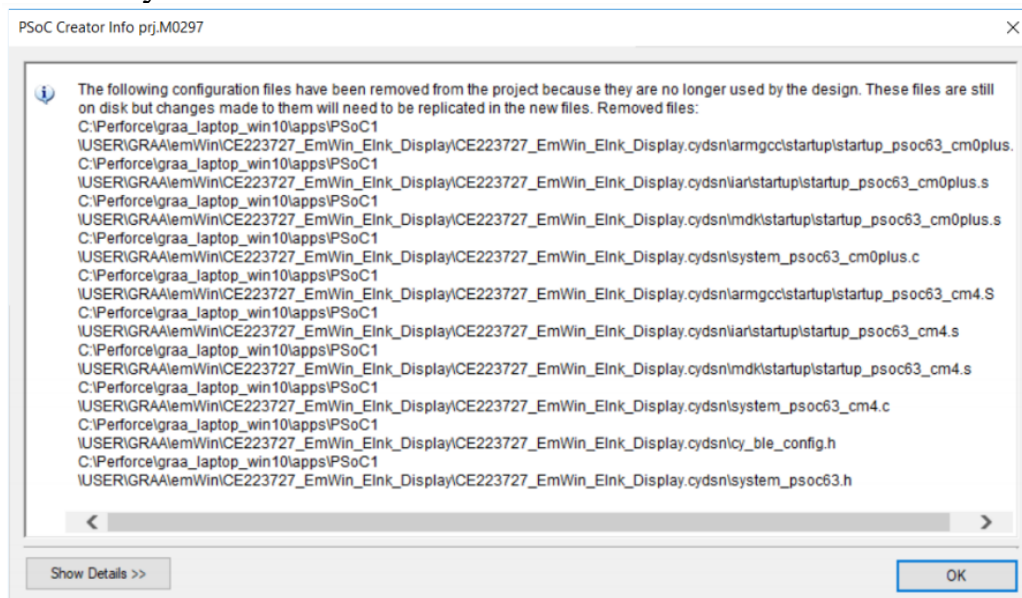


Рис. 8.5. Вигляд вікна повідомлень під час побудови проекту  
Після цього відобразиться наступне повідомлення (рис. 8.6).

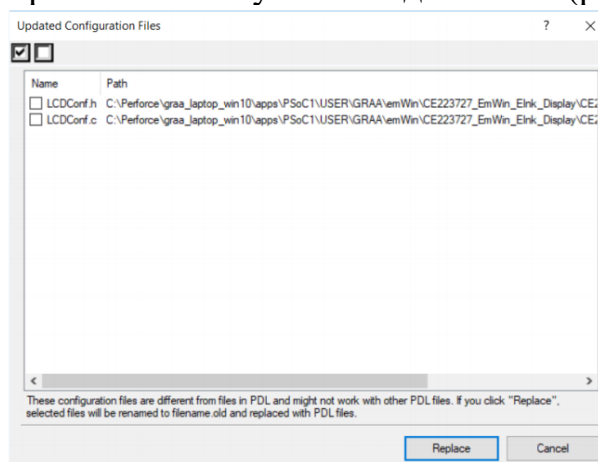


Рис. 8.6. Вигляд вікна повідомлень з оновленими файлами конфігурації

Потрібно зняти виділення обох файлів (їх буде знято за замовчуванням) та натисніть Replace. Якщо вибрати ці файли і натисніть Replace, файли конфігурації в проекті будуть замінені файлами конфігурації за замовчуванням.

### ***Дизайн та реалізація проекту.***

У цій програмній частині проекту є три важливі частини:

1. Бібліотека графіки EmWin: Бібліотека графіки EmWin реалізована як програмне забезпечення в PDL та реалізує всі графічні функції. Бібліотека управляє буфером відображення та оновлює цей буфер дисплея з піксельними даними відповідно до виконаних графічних операцій.

2. Драйвер E-INK: Драйвер E-INK забезпечує передачу піксельних даних з буфера дисплея в IC драйвера E-INK через інтерфейс SPI Master та декілька GPIO.

3. Код програми: Код додатку викликає графічні API- інтерфейси EmWin для виконання графічних функцій, керує буфером дисплею додатку із кешем (необхідний бібліотеці E-INK) та викликає бібліотеку дисплеїв E-INK для оновлення дисплея.

Включення та налаштування графічної бібліотеки EmWin.

1. У PSoC Creator перейдіть до Project > Build Settings та виберіть Peripheral Driver Library. У розділі Graphics > emWin виберіть Core та параметри LCD Driver.

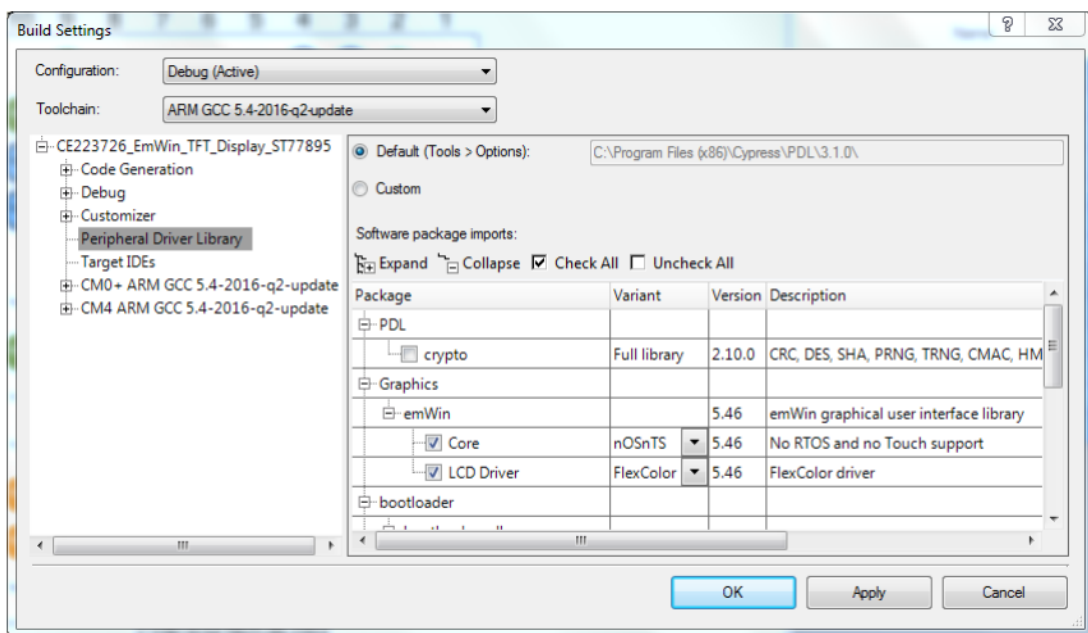


Рис. 8.7. Вигляд вікна для задання параметрів Core та LCD Driver Options

2. Виберіть параметр nOSnTS для Core, оскільки цей проект не використовує підтримку RTOS або Touch.

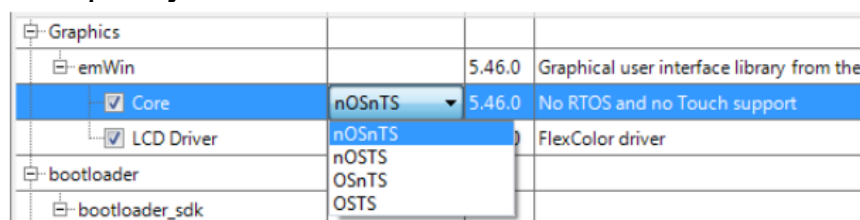


Рис. 8.8. Вікно вибору параметру Core

3. Виберіть параметр BitPlains для параметра LCD Driver. За допомогою драйвера BitPlains бібліотека EmWin управляє лише буфером графічного дисплея в пам'яті. Цей драйвер може підтримувати кольорові профілі від 1 біта на піксель (1bpp) до 8 біт на піксель (8bpp). Оскільки E-INK підтримує чорний/білий кольори, цей проект використовує кольоровий профіль 1bpp.

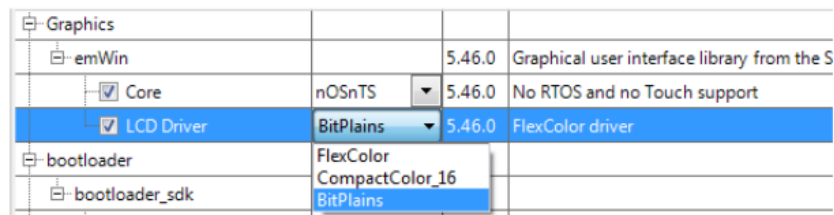


Рис. 8.9. Вікно вибору профілю кольору Core

4. Натисніть Click Generate Application (згенерувати додаток).



Рис. 8.10. Вигляд піктограми генерації додатку

PSoC Creator створює файли конфігурації для EmWin в папке Shared Files (Загальні файли).

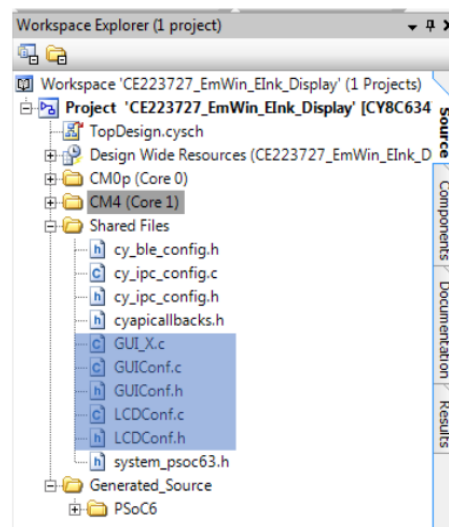


Рис. 8.11. Створені файли конфігурації

5. Відкрийте файл LCDConf.c і налаштуйте розміри X та Y дисплея та перетворення кольорів. Дисплей E-INK, що використовується в екрані E-INK, має роздільну здатність 264x176 пікселів. GUICC\_1 встановлює профіль кольорів у 1bpp.

EmWin виділяє буфер відображення на основі визначених розмірів X та Y.

6. Відкрийте файл GUIConf.c. Цей файл управляє розподілом оперативної пам'яті для EmWin. Значення макросу GUI\_NUMBYTES має бути встановлено відповідно до приблизної потреби в пам'яті на основі функцій EmWin, які використовує додаток. У цьому проекті розмір пам'яті встановлено доволіно (значення 0x1000 байт). Файл GUI\_X.c має функції синхронізації, які використовує EmWin.



Вмістиме цього файлу залежить від вибраної підтримки ОС. Для цього проекту в цьому файлі не потрібно ніяких змін.

```

55 /*****
56 *
57 *   Layer configuration
58 *
59 *****/
60 */
61 //
62 // Physical display size
63 //
64 #define XSIZE_PHYS 264
65 #define YSIZE_PHYS 176
66 //
67 //
68 // Initial color conversion API
69 //
70 #define COLOR_CONVERSION GUICC_1
71

```

Рис. 8.12. Встановлення значень X та Y та перетворення кольорів

CY E-INK Driver. Цей проект містить необхідні бібліотечні функції для управління дисплеєм E-INK. Але фактичні функції апаратного драйвера тут не розглядаються. На рис. 8.13 показана схема PSoC Creator, в якій реалізовано обладнання, необхідне для бібліотеки відображення E-INK.

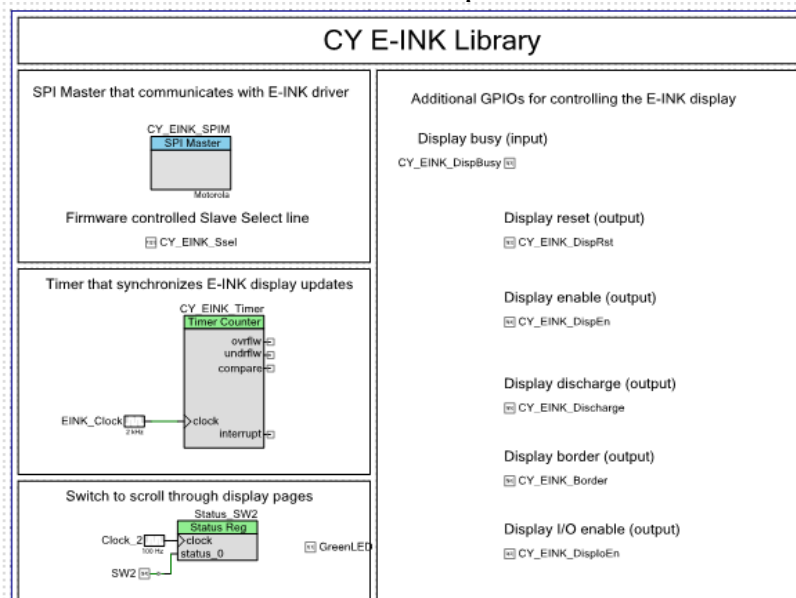


Рис. 8.13. Схема TopDesign: бібліотека E-INK

Мікроконтролер PSoC 6 управляє скиданням, включенням, розрядкою и обмеженням виводів дисплею E-INK. Мікроконтролер PSoC 6 також зчитує стан дисплею, щоб визначити, чи дисплей закінчив попередню операцію. Перемикач навантаження на CY8CKIT-028-EPD, який управляється пристроєм MCU PSoC 6, может використовуватися для включення / виключення дисплею. Транслятор рівня напруги підключений між дисплеєм E-INK та GPIO мікроконтролера PSoC 6, так що мікроконтролер PSoC 6 може працювати зі змінною напругою  $V_{DD}$ . Вхід дозволу перетворювача рівня напруги також підключений до графічного процесору PSoC 6 MCU, так що мікроконтролер PSoC 6 может відключати перетворювач рівня для зниження енергоспоживання, коли дисплей E-INK не використовується.

- CY\_EINK\_SPIM реалізує інтерфейс SPI Master, з допомогою якого PSoC 6 зв'язується з контролером E-INK.

- CY\_EINK\_Ssel - це GPIO, який реалізує сигнал вибору веденого для контролера E-INK.

- CY\_EINK\_Timer реалізує таймер 1 мс, який використовується в функціях синхронізації для функції відображення E-INK.

Бібліотека та файли драйверів E-INK: Наступні файли реалізують драйвер E-INK:

- Файли `cy_eink_library.c/.h` містять функції та макроси бібліотеки E-INK.
- Файл `pervasive_eink_configuration.h` містить визначення індексів регістрів та апаратних параметрів дисплея E-INK, наданих виробником дисплею.
- Файли `pervasive_eink_hardware_driver.c/.h` містять низькорівневі функції апаратного драйвера дисплею, які надаються виробником дисплею.
- Файли `cy_eink_psoc_interface.c/.h` містять інтерфейс рівня компонентів PSoC 6 MCU для апаратного забезпечення дисплею.

Розглянемо функції, що надаються в бібліотеці драйверів дисплея E-INK. Ці функції знаходяться у файлі `cy_eink_Library.c`.

1. `void Cy_EINK_Start(int8_t temperature)`

Ця функція ініціалізує апаратне забезпечення дисплея E-INK та компоненти PSoC.

Параметри функції:

`temperature` – температура навколишнього середовища в градусах Цельсія.

Після ініціалізації дисплею E-INK ця функція відключає живлення на дисплеї.

2. `bool Cy_EINK_Power(bool powerCtrl)`

Ця функція вмикає / вимикає живлення на дисплеї E-INK та ініціалізує драйвер E-INK.

Параметри функції:

`powerCtrl` – `false` – живлення виключено, `true` – живлення включено.

Повертаються:

`false` – ініціалізація драйвера не вдалася, `true` – ініціалізація драйвера є успішною.

Вміст дисплею буде збережено навіть після вимкнення його живлення.

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

Ця функція очищує дисплей E-INK в білий або чорний фон.

Параметри функції:

`background` – `false` – чорний фон, `true` – білий фон;

`powerCycle` – `false` – не керує включенням/виключенням живлення автоматично; `true` – автоматично керує включенням/виключенням живлення. Ця функція включає живлення, очищає дисплей, а потім виключає живлення.

Якщо значення параметру `powerCycle` невірне, то перед тим, як викликати цю функцію, потрібно включити дисплей E-INK, використовуючи функцію `Cy_EINK_Power`. В іншому випадку дисплей не очиститься.

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





uint8\_t\* imgCoordinates)

Ця функція обрізає зображення за вказаними координатами і копіює його в те саме місце в буфері кадру.

Параметри функції:

`frameBuffer` – вказівник на буфер кадру, що зберігається в оперативній пам'яті. Кадр складається з 5808 байтів ( $264 \times 176 / 8$ ) піксельних даних, в яких кожен біт зберігає піксельну інформацію монохроматичного зображення.

`image` – Вказівник на монохроматичне зображення, що зберігається у flash / RAM, як масив 5808 байт. Зображення має мати той самий розмір і формат, що і буфер кадру.

`imgCoordinates` – вказівник на чотирибайтовий масив, який зберігає байтові координати, в яких обрізається зображення (включаючи кінцеві координати X і Y) перед копіюванням в буфер кадру.

### **Формати зображення та тексту.**

Формат буферу зображення та кадру.

Дисплей E-INK має роздільну здатність  $264 \times 176$  пікселів. Бібліотека E-INK дисплеїв підтримує буфери зображень та кадрів, що зберігаються у вигляді масиву `uint8` розміром 5808 ( $264 \times 176 / 8$ ). На рис. 8.14 показано, як дані пікселів зберігаються у зображенні масиву [5808].

Ви можете використовувати програми PDi Apps від виробника дисплеїв, щоб створити змінний масив із растрового зображення.

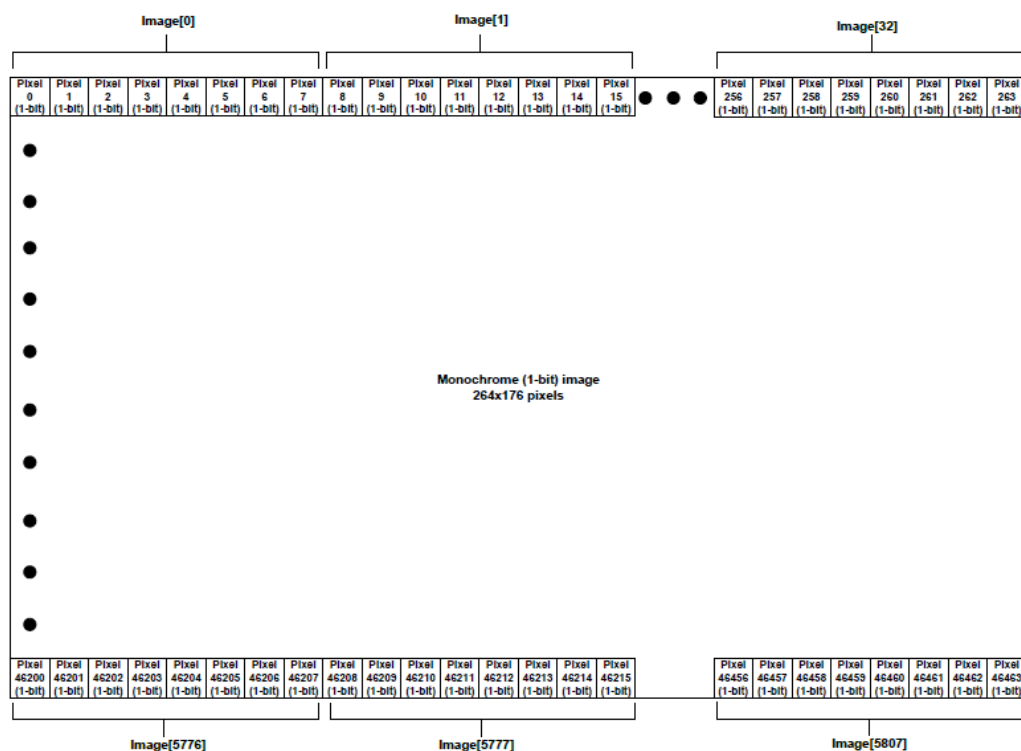


Рис. 8.14. Формат буфера зображення та кадру

*Підтримувані шрифти.*

Бібліотека драйверів дисплея E-INK підтримує два шрифти постійного розміру: `CY_EINK_FONT_8X12BLACK` та `CY_EINK_FONT_16X16BLACK`. На рис. 8.15 та рис. 8.16 показано формат цих шрифтів.

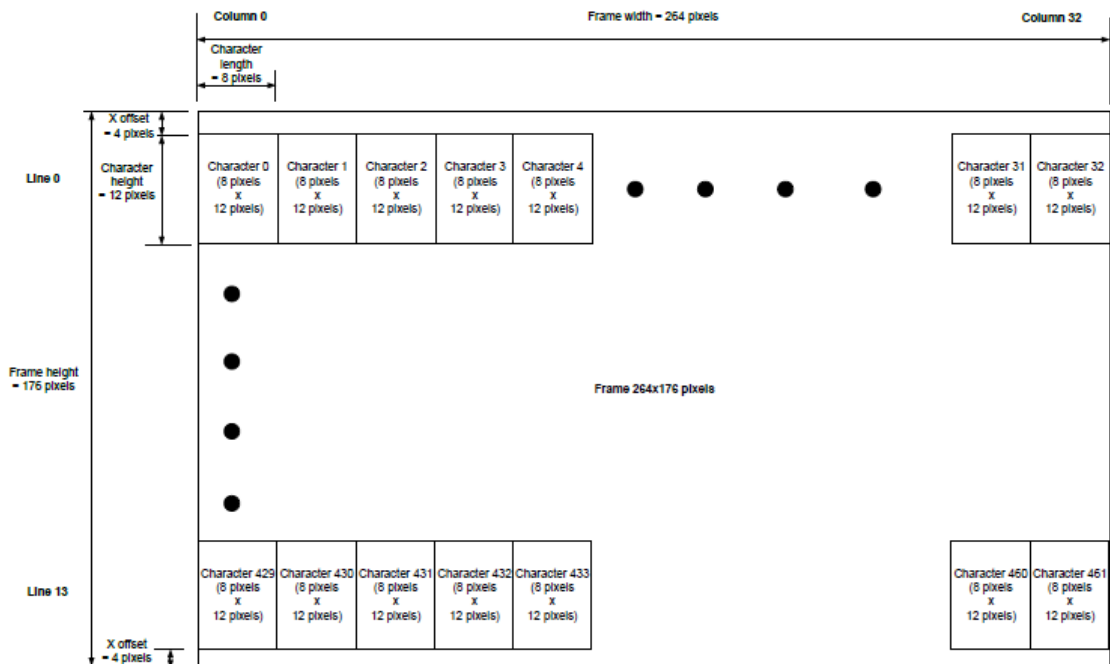


Рис. 8.15. Формат шрифту CY\_EINK\_FONT\_8X12BLACK

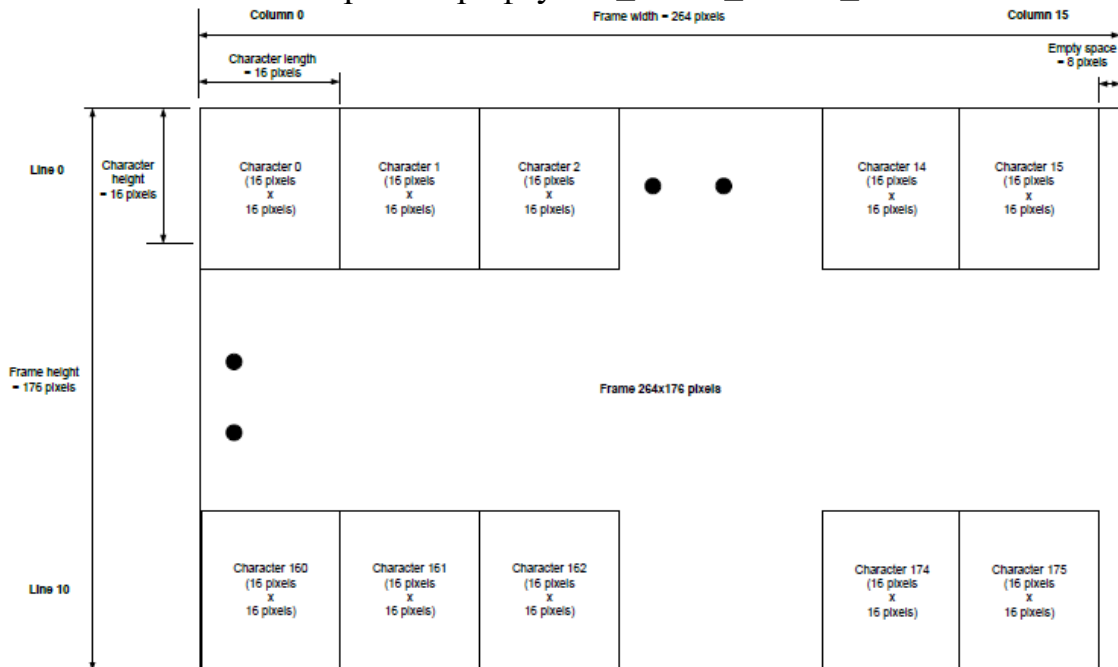


Рис. 8.16. Формат шрифту CY\_EINK\_FONT\_16X16BLACK

Основне використання: Основне використання реалізовано в файлі main\_cm4.c. Наступні функції виконуються тут.

1. Ініціалізація графічного рушія EmWin.
2. Ініціалізація драйвера E-INK.
3. Відображення екрану при запуску програми.
4. Відображення на екрані інструкції, в якій пропонується користувачу натиснути кнопку SW2 для прокрутки різних сторінок дисплею.
5. В нескінченному циклі, відображаються наступні сторінки; після відображення кожної сторінки, очікується подія натискання та відпускання кнопки SW2:

- a) відображення нормальних фонтів;
- b) відображення жирних фонтів;
- c) відображення різних режимів тексту та вирівнювання;
- d) відображення переносу слів і повороту;
- e) відображення 2D графічного екрана #1, на якому зображені вертикальні лінії, горизонтальні лінії, дуги та замальований прямокутник;
- f) відображення 2D графічного екрана #2, на якому зображені концентричні кола та концентричні еліпси.

На рис. 8.17 показана компонента для читання стану перемикача SW2. Перемикач SW2 підключений до компоненти регістру стану, яка працює на частоті 100 Гц. Регістр статусу усуває деренчання та надає стан перемикача SW2.

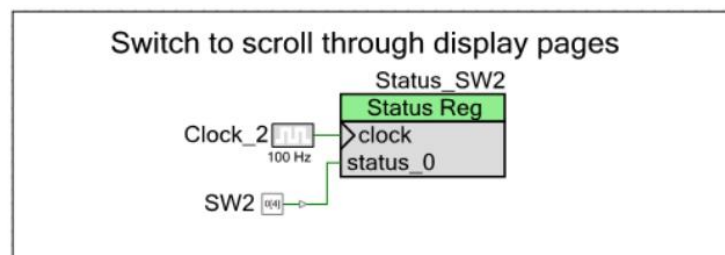


Рис. 8.17. Компонента усунення деренчання контактів перемикача та його стану  
На рис. 8.18, ..., рис. 8.26 зображено налаштування компонент проекту.

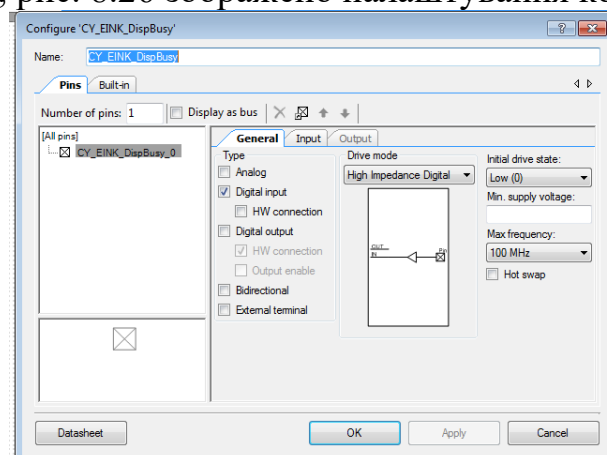


Рис. 8.18. Налаштування компоненти GPIO Cy\_EINK\_DisgBusy

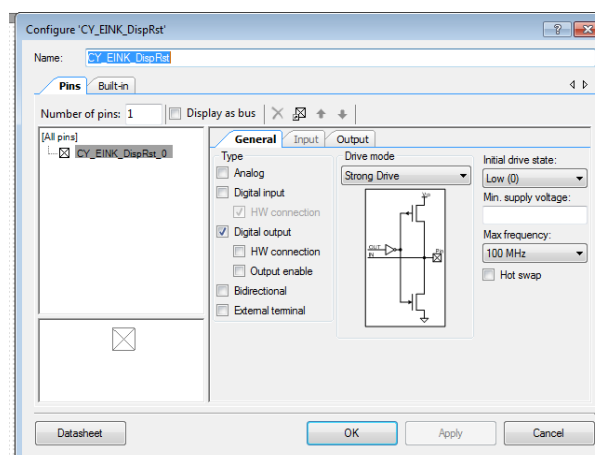


Рис. 8.19. Налаштування компоненти GPIO Cy\_EINK\_DisgRst

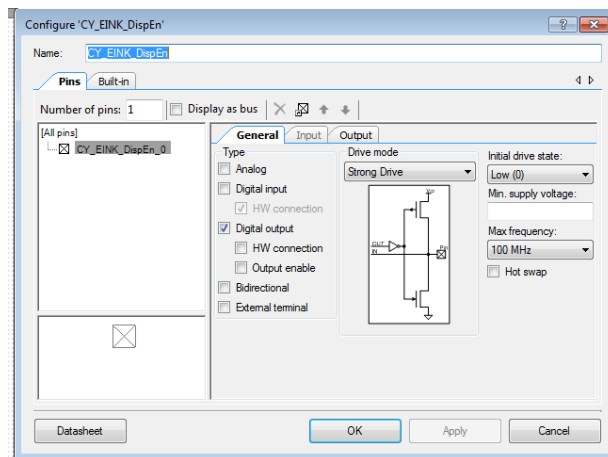


Рис. 8.20. Налаштування компоненти GPIO Cy\_EINK\_Dispen

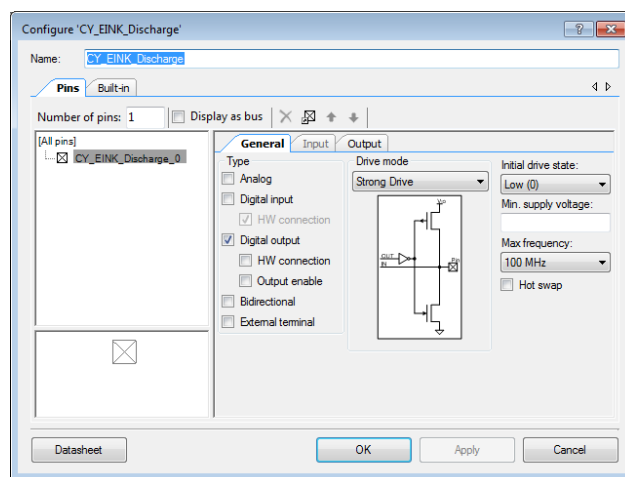


Рис. 8.21. Налаштування компоненти GPIO Cy\_EINK\_Discharge

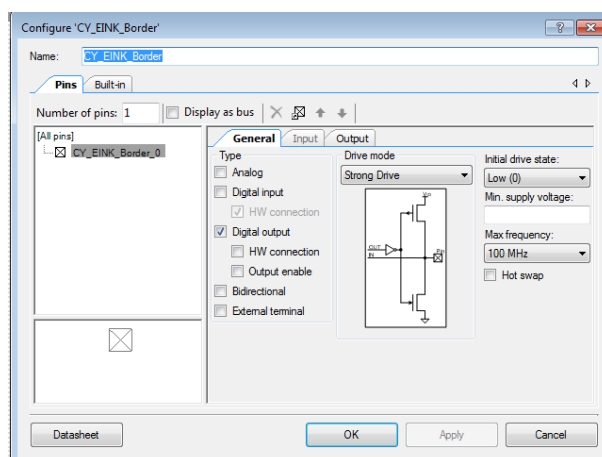


Рис. 8.22. Налаштування компоненти GPIO Cy\_EINK\_Border

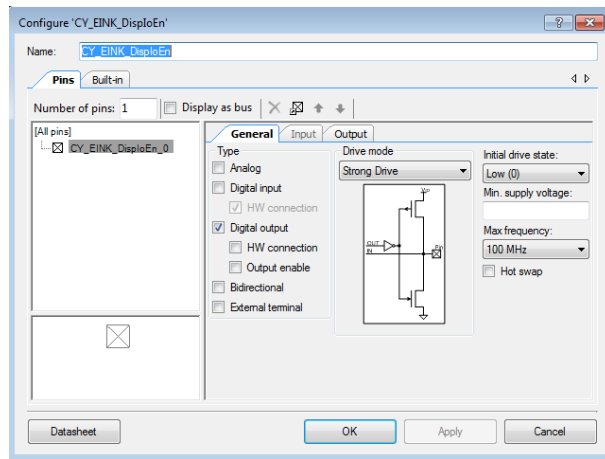


Рис. 8.23. Налаштування компоненти GPIO Cy\_EINK\_DispioEn

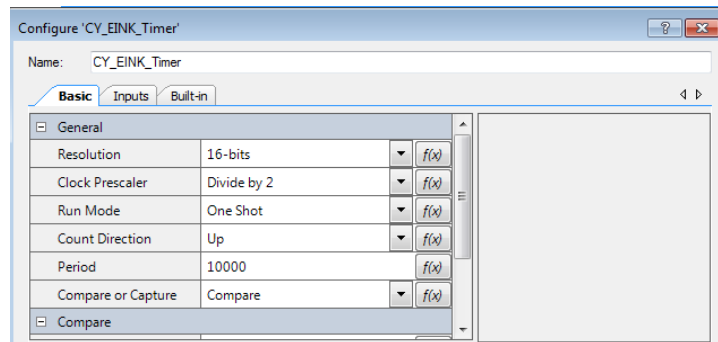


Рис. 8.24. Налаштування компоненти Cy\_EINK\_Timer

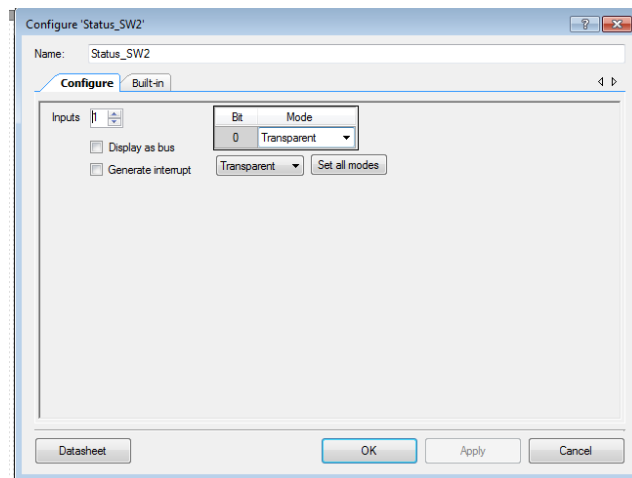


Рис. 8.25. Налаштування компоненти Status\_SW2

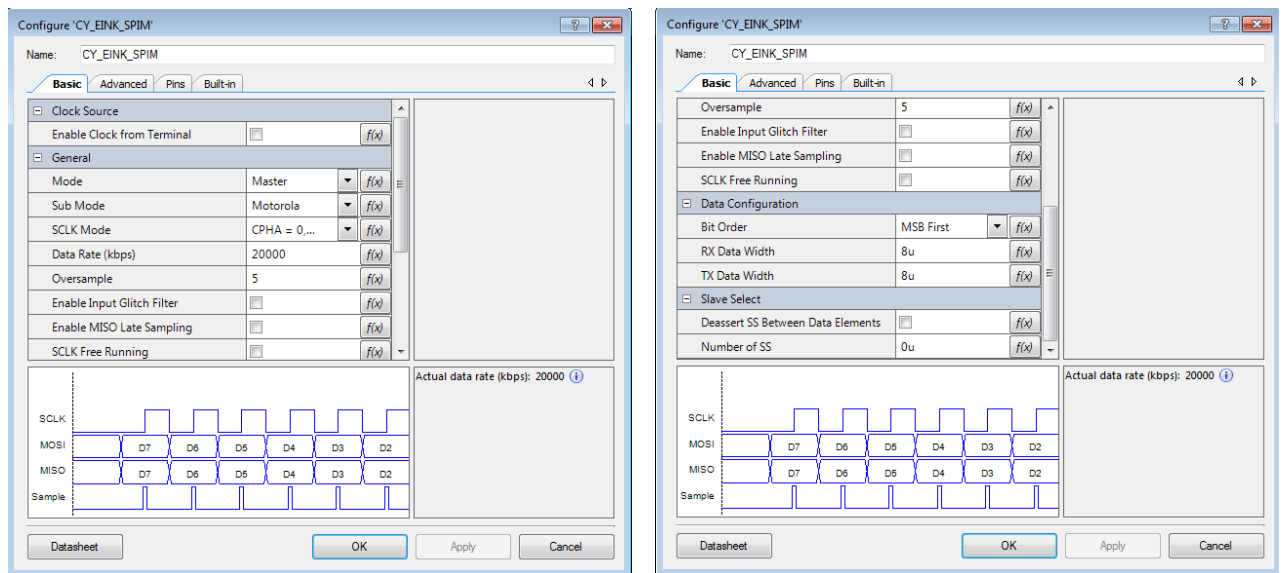


Рис. 8.26. Налаштування компоненти CY\_EINK\_SPIM

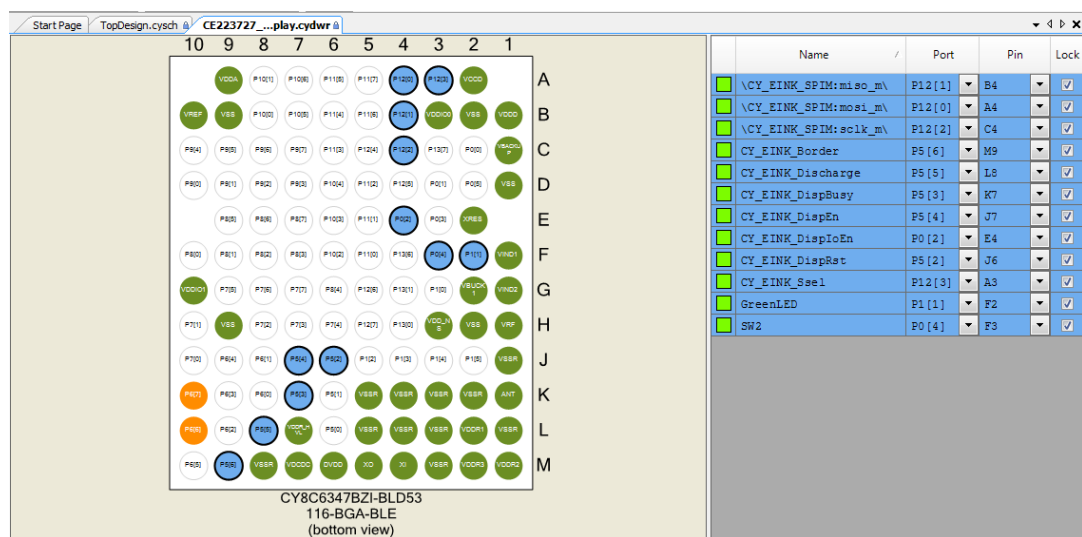


Рис. 8.27. Призначення виводів в схемі проекту

### Компоненти, які використовуються в цьому проекті:

- *SPI (SCB)*. Назва екземпляру CY\_EINK\_SPIM. Компонент SPI налаштований як майстер SPI, який спілкується з драйвером дисплею E-INK.
- *Timer Counter (TCPWM)*. Назва екземпляру CY\_EINK\_Timer. Лічильник таймера налаштований на 1LSB = 1 мсек. Величина рахунку використовується для часу відображення E-INK.
- *Digital Output Pin*. Назва екземплярів CY\_EINK\_Ssel, CY\_EINK\_DispRst, CY\_EINK\_DispEn, CY\_EINK\_Discharge, CY\_EINK\_Border, CY\_EINK\_DispIoEn. Ці GPIO налаштовані як вихідні виводи, які використовуються для забезпечення управління сигналами дисплею E-INK.
- *Digital Input Pin*. Назва екземпляру CY\_EINK\_DispBusy. Цей GPIO є цифровим входом без будь-якого апаратного підключення. Він використовується для читання статусу дисплею E-INK. Назва екземпляру SW2. Цей цифровий входний вивід підключений до входу тригера, щоб прочитати стан перемикача SW2.



- *Status Register*. Назва екземпляру Status\_SW2. Цей регістр стану використовується для зчитування статусу коммутатора. Тактова частота 100 Гц використовується для регістру стану, який також усуває деренчання контактів перемикача.

### Програмна реалізація проекту.

```
/*
 * Програма демонструє управління дисплеєм EInk за допомогою графічної бібліотеки
 * EmWin. Проект відображає екран запуску з логотипом Cypress і текстом "CYPRESS
 * EMWIN GRAPHICS DEMO EINK DISPLAY". Потім в проекті циклічно відображаються
 * наступні екрани:
 *
 * 1. Екран, на якому відображаються різні вирівнювання тексту, стилі та режими.
 * 2. Екран із звичайними шрифтами.
 * 3. Екран із жирними шрифтами.
 * 4. Екран із зображенням двовимірної графіки з горизонтальними лініями,
 *    вертикальними лініями, дугою і замальованим округлим прямокутником.
 * 5. Екран із зображенням 2D графіки із концентричними колами та еліпсами.
 * 6. Екран із текстовим полем із загорнутим текстом
 */

#include "project.h"
#include "GUI.h"
#include "pervasive_eink_hardware_driver.h"
#include "cy_eink_library.h"
#include "LCDConf.h"

/* Кеш буфера зображення */
uint8 imageBufferCache[CY_EINK_FRAME_SIZE] = {0};

/* Посилання на растрове зображення для екрану при запуску програми */
extern GUI_CONST_STORAGE GUI_BITMAP bmCypressLogoFullColor_PNG_1bpp;

/* Function prototypes */
void ShowFontSizesNormal(void);
void ShowFontSizesBold(void);
void ShowTextModes(void);
void ShowTextWrapAndOrientation(void);
void Show2DGraphics1(void);
void Show2DGraphics2(void);

/* Масив функцій демонстраційних сторінок */
void (*demoPageArray[])(void) = {
    ShowFontSizesNormal,
    ShowFontSizesBold,
    ShowTextModes,
    ShowTextWrapAndOrientation,
    Show2DGraphics1,
    Show2DGraphics2
};

/* Число демонстраційних сторінок */
#define NUMBER_OF_DEMO_PAGES (sizeof(demoPageArray)/sizeof(demoPageArray[0]))

/* Ця функція оновлює дисплей з даними буферу дисплею.
 * Функція спочатку передає вміст буферу дисплею EmWin
 * до основного буферу дисплея EInk. Потім викликає
 * функцію Cy_EINK_ShowFrame для оновлення дисплея, а потім
 * вона копіює буфер дисплея EmWin в буфер кешу дисплея EInk */
/*Побічні ефекти:
```

```

* Для оновлення дисплею потрібно близько секунди. Це функція блокує
* і повертається лише після оновлення дисплею.

* Side Effects:
* It takes about a second to refresh the display. This is a blocking function
* and only returns after the display refresh
*
*****/
void UpdateDisplay(cy_eink_update_t updateMethod, bool powerCycle)
{
    cy_eink_frame_t* pEmwinBuffer;

    /* Отримання вказівника на буфер дисплею Emwin */
    pEmwinBuffer = (cy_eink_frame_t*)LCD_GetDisplayBuffer();

    /* Оновлення EInk дисплею */
    Cy_EINK_ShowFrame(imageBufferCache, pEmwinBuffer, updateMethod, powerCycle);

    /* Копіювання EmWin буферу дисплею в буфер кешу дисплея */
    memcpy(imageBufferCache, pEmwinBuffer, CY_EINK_FRAME_SIZE);
}

/*Ця функція відображає екран при запуску програми з Cypress Logo
* та текст опису демоверсії */
void ShowStartupScreen(void)
{
    /* Задання кольору переднього плану та розміру шрифту */
    GUI_SetFont(GUI_FONT_16B_1);
    GUI_SetColor(GUI_BLACK);
    GUI_SetBkColor(GUI_WHITE);
    GUI_Clear();

    GUI_DrawBitmap(&bmCypressLogoFullColor_PNG_1bpp, 2, 2);
    GUI_SetTextAlign(GUI_TA_HCENTER);
    GUI_DispatchStringAt("CYPRESS", 132, 85);
    GUI_SetTextAlign(GUI_TA_HCENTER);
    GUI_DispatchStringAt("EMWIN GRAPHICS", 132, 105);
    GUI_SetTextAlign(GUI_TA_HCENTER);
    GUI_DispatchStringAt("EINK DISPLAY DEMO", 132, 125);

    /* Пересилка на екран даних буферу дисплея */
    UpdateDisplay(CY_EINK_FULL_4STAGE, true);
}

/*Ця функція показує екран із інструкцією для натискання кнопки SW2
* та для прокручування різних сторінок дисплею */
void ShowInstructionsScreen(void)
{
    /* Set font size, background color and text mode */
    GUI_SetFont(GUI_FONT_16B_1);
    GUI_SetBkColor(GUI_WHITE);
    GUI_SetColor(GUI_BLACK);
    GUI_SetTextMode(GUI_TM_NORMAL);

    /* Очистка дисплея */
    GUI_Clear();

    /* Відображення тексту інструкцій */
    GUI_SetTextAlign(GUI_TA_HCENTER);
    GUI_DispatchStringAt("PRESS SW2 ON THE KIT", 132, 58);
    GUI_SetTextAlign(GUI_TA_HCENTER);
    GUI_DispatchStringAt("TO SCROLL THROUGH ", 132, 78);
    GUI_SetTextAlign(GUI_TA_HCENTER);

```

```

    GUI_DispStringAt("DEMO PAGES!", 132, 98);

    /* Передача на екран даних буферу дисплея */
    UpdateDisplay(CY_EINK_FULL_4STAGE, true);
}

/* Ця функція відображує на дисплеї різні розміри шрифту */

void ShowFontSizesNormal(void)
{
    /* Задання розміру шрифту, кольору тла (фону) та режиму тексту */
    GUI_SetFont(GUI_FONT_13B_1);
    GUI_SetBkColor(GUI_WHITE);
    GUI_SetColor(GUI_BLACK);
    GUI_SetTextMode(GUI_TM_NORMAL);

    /* Очистка дисплея */
    GUI_Clear();

    /* Відображення назви сторінки */
    GUI_SetTextAlign(GUI_TA_HCENTER);
    GUI_DispStringAt("1 OF 6: NORMAL FONTS", 132, 5);

    /* Font8_1*/
    GUI_SetFont(GUI_FONT_8_1);
    GUI_DispStringAt("GUI_Font8_1", 10, 30);

    /* Font10_1*/
    GUI_SetFont(GUI_FONT_10_1);
    GUI_DispStringAt("GUI_Font10_1", 10, 40);

    /* Font13_1*/
    GUI_SetFont(GUI_FONT_13_1);
    GUI_DispStringAt("GUI_Font13_1", 10, 52);

    /* Font16_1*/
    GUI_SetFont(GUI_FONT_16_1);
    GUI_DispStringAt("GUI_Font16_1", 10, 67);

    /* Font20_1*/
    GUI_SetFont(GUI_FONT_20_1);
    GUI_DispStringAt("GUI_Font20_1", 10, 85);

    /* Font24_1*/
    GUI_SetFont(GUI_FONT_24_1);
    GUI_DispStringAt("GUI_Font24_1", 10, 107);

    /* Font32_1*/
    GUI_SetFont(GUI_FONT_32_1);
    GUI_DispStringAt("GUI_Font32_1", 10, 133);

    /* Send the display buffer data to display*/
    UpdateDisplay(CY_EINK_FULL_4STAGE, true);
}

/* Ця функція відображує різні розміри шрифту */

void ShowFontSizesBold(void)
{
    /* Задання розміру шрифту, кольору тла (фону) та режиму тексту */
    GUI_SetFont(GUI_FONT_13B_1);
    GUI_SetBkColor(GUI_WHITE);
    GUI_SetColor(GUI_BLACK);
    GUI_SetTextMode(GUI_TM_NORMAL);

```

```

/* Очищення дисплею */
GUI_Clear();

/* Відображення назви сторінки */
GUI_SetTextAlign(GUI_TA_HCENTER);
GUI_DispStringAt("2 OF 6: BOLD FONTS", 132, 5);

/* Font13B_1*/
GUI_SetFont(GUI_FONT_13B_1);
GUI_DispStringAt("GUI_Font13B_1", 5, 25);

/* Font13HB_1*/
GUI_SetFont(GUI_FONT_13HB_1);
GUI_DispStringAt("GUI_Font13HB_1", 5, 40);

/* Font16B_1*/
GUI_SetFont(GUI_FONT_16B_1);
GUI_DispStringAt("GUI_Font16B_1", 5, 55);

/* FontComic18B_1*/
GUI_SetFont(GUI_FONT_COMIC18B_1);
GUI_DispStringAt("GUI_FontComic18B_1", 5, 73);

/* Font20B_1*/
GUI_SetFont(GUI_FONT_20B_1);
GUI_DispStringAt("GUI_Font20B_1", 5, 93);

/* Font24B_1*/
GUI_SetFont(GUI_FONT_24B_1);
GUI_DispStringAt("GUI_Font24B_1", 5, 115);

/* Font32B_1*/
GUI_SetFont(GUI_FONT_32B_1);
GUI_DispStringAt("GUI_Font32B_1", 5, 141);

/* Пересилка даних буферу дисплею на дисплей */
UpdateDisplay(CY_EINK_FULL_4STAGE, true);
}

/* Ця функція відображає наступне:
* 1. Текст вирівнюється зліва, по центру та справа
* 2. Підкреслити, накреслити та прокреслити текст стилю
* 3. Нормальний, зворотний, прозорий та текстовий режими XOR */
void ShowTextModes(void)
{
    /* Задання розміру шрифту, кольору тла (фону) та режиму тексту */
    GUI_SetFont(GUI_FONT_13B_1);
    GUI_SetColor(GUI_BLACK);
    GUI_SetBkColor(GUI_WHITE);
    GUI_SetTextMode(GUI_TM_NORMAL);
    GUI_SetTextStyle(GUI_TS_NORMAL);

    /* Очищення дисплею */
    GUI_Clear();

    /* Відображення назви сторінки */
    GUI_SetTextAlign(GUI_TA_HCENTER);
    GUI_DispStringAt("3 OF 6: TEXT ALIGNMENTS, STYLES AND MODES", 132, 5);

    /* Відображення тексту, вирівняного зліва */
    GUI_SetTextAlign(GUI_TA_LEFT);
    GUI_DispStringAt("TEXT ALIGNMENT LEFT", 0, 30);

```

```

/* Відображення тексту, вирівняного по центру */
GUI_SetTextAlign(GUI_TA_HCENTER);
GUI_DispStringAt("TEXT ALIGNMENT CENTER", 132, 45);

/* Відображення тексту, вирівняного зправа */
GUI_SetTextAlign(GUI_TA_RIGHT);
GUI_DispStringAt("TEXT ALIGNMENT RIGHT", 263, 60);

/* Відображення підкресленого тексту */
GUI_SetTextStyle(GUI_TS_UNDERLINE);
GUI_SetTextAlign(GUI_TA_LEFT);
GUI_DispStringAt("TEXT STYLE UNDERLINE", 0, 75);

/* Відображення накресленого тексту */
GUI_SetTextStyle(GUI_TS_OVERLINE);
GUI_SetTextAlign(GUI_TA_LEFT);
GUI_DispStringAt("TEXT STYLE OVERLINE", 0, 90);

/* Відображення закресленого тексту */
GUI_SetTextStyle(GUI_TS_STRIKETHRU);
GUI_SetTextAlign(GUI_TA_LEFT);
GUI_DispStringAt("TEXT STYLE STRIKETHROUGH", 0, 105);

/* Створення прямокутника, замальованого чорним кольором */
GUI_SetColor(GUI_BLACK);
GUI_FillRect(0, 120, 263, 175);

/* Відображення двох діагональних ліній */
GUI_SetColor(GUI_WHITE);
GUI_SetPenSize(3);
GUI_DrawLine(0, 120, 263, 175);
GUI_DrawLine(0, 175, 263, 120);

/* Встановлення кольору тексту білим на чорному фоні */
GUI_SetColor(GUI_BLACK);
GUI_SetBkColor(GUI_WHITE);

/* Встановлення стилю тексту нормальним */
GUI_SetTextStyle(GUI_TS_NORMAL);

/* Display text in normal mode. This will print black text in
   a white box */
GUI_SetTextAlign(GUI_TA_HCENTER);
GUI_SetTextMode(GUI_TM_NORMAL);
GUI_DispStringAt("TEXT MODE NORMAL", 132, 135);

/* Відображення тексту у інверсному режимі. Тоді відобразиться чорний текст
   на білому фоні */
GUI_SetTextAlign(GUI_TA_HCENTER);
GUI_SetTextMode(GUI_TM_REV);
GUI_DispStringAt("TEXT MODE REVERSE", 132, 150);

/* Пересилка на екран даних з буферу дисплею */
UpdateDisplay(CY_EINK_FULL_4STAGE, true);
}

/* void ShowTextWrapAndOrientation(void) Ця функція викликає функції орієнтації
   тексту та обгортання тексту */
void ShowTextWrapAndOrientation(void)
{
    GUI_RECT leftRect = {4, 19, 24, 166};
    GUI_RECT rightRect = {238, 19, 258, 166};
    GUI_RECT middleRect = {29, 19, 233, 166};

```

```

GUI_RECT middleRectMargins = {31, 20, 232, 165};

const char leftText[] = "ROTATED TEXT CCW";
const char rightText[] = "ROTATED TEXT CW";

const char middleText[] = "This project demonstrates displaying 2D graphics
in an EInk display using Segger EmWin Graphics Library. \n\nThis page shows the
text wrap and text rotation features. In the left rectangle, the text is rotated
counter clockwise and in the right rectangle, the text is rotated clockwise.";

/* Встановлення розміру шрифту, переднього плану та кольору тла */
GUI_SetFont(GUI_FONT_13B_1);
GUI_SetColor(GUI_BLACK);
GUI_SetBkColor(GUI_WHITE);
GUI_SetTextMode(GUI_TM_NORMAL);
GUI_SetTextStyle(GUI_TS_NORMAL);

/* Очищення дисплею */
GUI_Clear();

/* Display page title */
GUI_SetTextAlign(GUI_TA_HCENTER);
GUI_DispStringAt("4 OF 6: TEXT WRAP AND ROTATION", 132, 5);

/* Draw rectangles to hold text */
GUI_DrawRectEx(&leftRect);
GUI_DrawRectEx(&rightRect);
GUI_DrawRectEx(&middleRect);

/* Display string in left rectangle rotated counter clockwise */
GUI_DispStringInRectEx(leftText, &leftRect, GUI_TA_HCENTER | GUI_TA_VCENTER,
strlen(leftText), GUI_ROTATE_CCW);

/* Display string in right rectangle rotated clockwise */
GUI_DispStringInRectEx(rightText, &rightRect, GUI_TA_HCENTER |
GUI_TA_VCENTER, strlen(rightText), GUI_ROTATE_CW);

/* Display string in middle rectangle with word wrap */
GUI_DispStringInRectWrap(middleText, &middleRectMargins, GUI_TA_LEFT,
GUI_WRAPMODE_WORD);

/* Send the display buffer data to display*/
UpdateDisplay(CY_EINK_FULL_4STAGE, true);
}

/* void Show2DGraphics1(void)
* Ця функція відображає наступну двовимірну графіку:
* 1. Горизонтальні лінії різної товщини.
* 2. Вертикальні лінії різної товщини.
* 3. Дуги.
* 4.
* Summary: This function displays the following 2D graphics
* 1. Horizontal lines with various pen widths
* 2. Vertical lines with various pen widths
* 3. Arcs
* 4. Замальований прямокутник. */
void Show2DGraphics1(void)
{
    /* Set font size, foreground and background colors */
    GUI_SetColor(GUI_BLACK);
    GUI_SetBkColor(GUI_WHITE);
    GUI_SetTextMode(GUI_TM_NORMAL);

```



```

GUI_SetTextStyle(GUI_TS_NORMAL);

/* Clear the screen */
GUI_Clear();

/* Display page title */
GUI_SetFont(GUI_FONT_13B_1);
GUI_SetTextAlign(GUI_TA_HCENTER);
GUI_DispStringAt("5 OF 6: 2D GRAPHICS - 1", 136, 5);

/* Display labels */
GUI_SetFont(GUI_FONT_13B_1);
GUI_SetTextAlign(GUI_TA_HCENTER);
GUI_DispStringAt("H-LINES", 68, 85);
GUI_SetTextAlign(GUI_TA_HCENTER);
GUI_DispStringAt("V-LINES", 204, 85);
GUI_SetTextAlign(GUI_TA_HCENTER);
GUI_DispStringAt("ARCS", 68, 163);
GUI_SetTextAlign(GUI_TA_HCENTER);
GUI_DispStringAt("ROUNDED RECT", 204, 163);

/* Horizontal lines */
GUI_SetPenSize(1);
GUI_DrawLine(10, 25, 125, 25);
GUI_SetPenSize(2);
GUI_DrawLine(10, 35, 125, 35);
GUI_SetPenSize(3);
GUI_DrawLine(10, 45, 125, 45);
GUI_SetPenSize(4);
GUI_DrawLine(10, 55, 125, 55);
GUI_SetPenSize(5);
GUI_DrawLine(10, 65, 125, 65);
GUI_SetPenSize(6);
GUI_DrawLine(10, 75, 125, 75);

/* Vertical lines */
GUI_SetPenSize(1);
GUI_DrawLine(168, 20, 168, 80);
GUI_SetPenSize(2);
GUI_DrawLine(180, 20, 180, 80);
GUI_SetPenSize(3);
GUI_DrawLine(192, 20, 192, 80);
GUI_SetPenSize(4);
GUI_DrawLine(204, 20, 204, 80);
GUI_SetPenSize(5);
GUI_DrawLine(216, 20, 216, 80);
GUI_SetPenSize(6);
GUI_DrawLine(228, 20, 228, 80);
GUI_SetPenSize(7);
GUI_DrawLine(240, 20, 240, 80);

/* Дуги */
GUI_SetPenSize(2);
GUI_DrawArc(68, 160, 10, 10, 0, 180);
GUI_DrawArc(68, 160, 20, 20, 0, 180);
GUI_DrawArc(68, 160, 30, 30, 0, 180);
GUI_DrawArc(68, 160, 40, 40, 0, 180);
GUI_DrawArc(68, 160, 50, 50, 0, 180);

/* Rounded rectangle */
GUI_FillRoundedRect(146, 108, 262, 160, 5);

/* Send the display buffer data to display*/
UpdateDisplay(CY_EINK_FULL_4STAGE, true);

```

```

}

/* void Show2DGraphics2(void)
*   Ця функція відображає наступну двовимірну графіку:
*   1. Концентричні кола.
*   2. Концентричні еліпси. */
void Show2DGraphics2(void)
{
    /* Set font size, foreground and background colors */
    GUI_SetColor(GUI_BLACK);
    GUI_SetBkColor(GUI_WHITE);
    GUI_SetTextMode(GUI_TM_NORMAL);
    GUI_SetTextStyle(GUI_TS_NORMAL);

    /* Clear the screen */
    GUI_Clear();

    /* Display page title */
    GUI_SetFont(GUI_FONT_13B_1);
    GUI_SetTextAlign(GUI_TA_HCENTER);
    GUI_DispStringAt("6 OF 6: 2D GRAPHICS - 2", 132, 5);

    /* Display labels */
    GUI_SetFont(GUI_FONT_13B_1);
    GUI_SetTextAlign(GUI_TA_HCENTER);
    GUI_DispStringAt("CIRCLE", 68, 85);
    GUI_SetTextAlign(GUI_TA_HCENTER);
    GUI_DispStringAt("ELLIPSE", 204, 85);

    /* Concentric Circles */
    GUI_DrawCircle(68, 51, 30);
    GUI_DrawCircle(68, 51, 25);
    GUI_DrawCircle(68, 51, 20);
    GUI_DrawCircle(68, 51, 15);
    GUI_DrawCircle(68, 51, 10);

    /* Concentric Ellipses */
    GUI_DrawEllipse(204, 51, 50, 30);
    GUI_DrawEllipse(204, 51, 40, 25);
    GUI_DrawEllipse(204, 51, 30, 20);
    GUI_DrawEllipse(204, 51, 20, 15);

    /* Send the display buffer data to display*/
    UpdateDisplay(CY_EINK_FULL_4STAGE, true);
}

/* void ClearScreen(void)
*   Ця функція очищає екран дисплею. */
void ClearScreen(void)
{
    GUI_SetColor(GUI_BLACK);
    GUI_SetBkColor(GUI_WHITE);
    GUI_Clear();
    UpdateDisplay(CY_EINK_FULL_4STAGE, true);
}

/* void WaitForSwitchPressAndRelease(void)
*   Ця функція реалізує вивід інформації "Wait for button press and release"
*   Спочатку чекає натискання кнопки, а потім чекає відпускання кнопки.
*   Побічні ефекти:
*   Ця функція блокування та працює тільки при натисканні, відпусканні кнопки */
void WaitForSwitchPressAndRelease(void)
{

```

```

    /* Wait for SW2 to be pressed */
    while(Status_SW2_Read() != 0);

    /* Wait for SW2 to be released */
    while(Status_SW2_Read() == 0);
}

/* int main(void)
 * Це головна функція. В ній виконуються наступні функції:
 * 1. Ініціалізація EmWin бібліотеки.
 * 2. Відображення екрану при запуску програми протягом 3 сек.
 * 3. Відображення на екрані інструкцій та очікування на натискання
 *    та відпускання кнопки.
 * 4. Прокручування усередині циклу 6 демонстраційних сторінок під час
 *    кожного натискання та відпускання кнопки. */
int main(void)
{
    uint8 pageNumber = 0;

    __enable_irq(); /* Дозвіл глобальних переривань. */

    /* Ініціалізація emWin графіки */
    GUI_Init();

    /* Запуск інтерфейсу e-Ink та включення його живлення */
    Cy_EINK_Start(20);
    Cy_EINK_Power(1);

    /* Відображення екрану при запуску програми */
    ShowStartupScreen();
    CyDelay(2000);

    /* Відображення інструкції на екрані дисплею */
    ShowInstructionsScreen();
    WaitForSwitchPressAndRelease();

    for(;;)
    {
        /* Включення зеленого світлодіода перед оновленням сторінки */
        Cy_GPIO_Write(GreenLED_PORT, GreenLED_NUM, 0);

        /* Використовуючи pageNumber як індекс, оновіть дисплей за допомогою
         * демонстраційного екрана. Далі приведено функції, які викликаються
         * послідовно одна за одною.
         * ShowFontSizesNormal()
         * ShowFontSizesBold()
         * ShowTextModes()
         * ShowTextWrapAndOrientation()
         * Show2DGraphics1()
         * Show2DGraphics2()
         */
        (*demoPageArray[pageNumber])();

        /* Вимкнення зеленого світлодіода LED після оновлення сторінки */
        Cy_GPIO_Write(GreenLED_PORT, GreenLED_NUM, 1);

        /* Очікування події: натискання кнопки */
        WaitForSwitchPressAndRelease();

        /* Перебір номерів сторінок */
        pageNumber = (pageNumber+1) % NUMBER_OF_DEMO_PAGES;
    }
}

```

/\* [] END OF FILE \*/

***Опис літератури:***

1. CE23727 – PSoC 6 E-INK Display Interface with EmWin Graphics Library. Електронний ресурс. Режим доступу: <https://www.cypress.com/file/448621/download>.

2. CE218133 - PSoC 6 MCU E-INK Display with CapSense. Електронний ресурс. Режим доступу: <https://www.cypress.com/file/385731/download>.

***Завдання.***