

Міністерство освіти і науки України
Національний університет „Львівська політехніка”



Звіт

до лабораторної роботи № 1

з дисципліни: «Програмування систем на кристалі»

на тему: «ВИВІД ІНФОРМАЦІЇ НА LCD ДИСПЛЕЙ СТЕНДУ PSOC3»

Виконав:

ст.гр. КІ-47

Мартиш М.В.

Прийняв:

Цигилик Л.О.

Мета роботи: ознайомлення з принципами роботи стенду PSoC 3 для практичного засвоєння базових навичок проектування систем на кристалі, засвоєння методів та засобів конфігурування програмованих систем на кристалі та перевірки їх функціонування на прикладі роботи з символьним дисплеєм.

Завдання

Згідно заданого варіанту написати програму котра динамічну інформацію на дисплеї з 2х16 елементів. Для кожного варіанту задана конфігурація із 2 активних зон дисплея. Для кожної зони реалізувати динамічне відображення текстової/графічної інформації. Швидкість зміни інформації в обидвох зонах однакова. Рух тексту/графіки повинен відбуватись із зсувом на 1 елемент вліво чи вправо, вгору чи вниз. Частота зміни тексту відповідає часу фіксації тексту/графіки в певному положення (наприклад 200мс, це означає що кожні 200мс дисплей має оновлюватись і текст/графіка повинен зміститись на один елемент вліво, вправо, вгору чи вниз. В залежності від завдання текст/графіка може рухатися групою зсуваючись разом на 1 елемент вліво, вправо, вгору та вниз або рухатись частинами.

Варіант-3

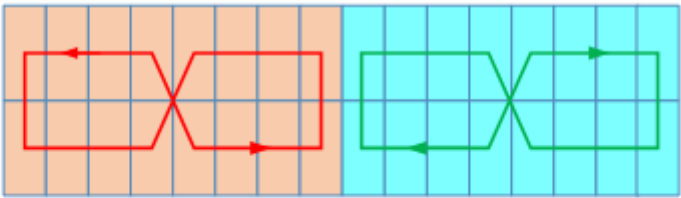


4			
			1200

Рис.1. Завдання варіант 4

Хід роботи

- 1) Запускаємо програму та створюємо проект

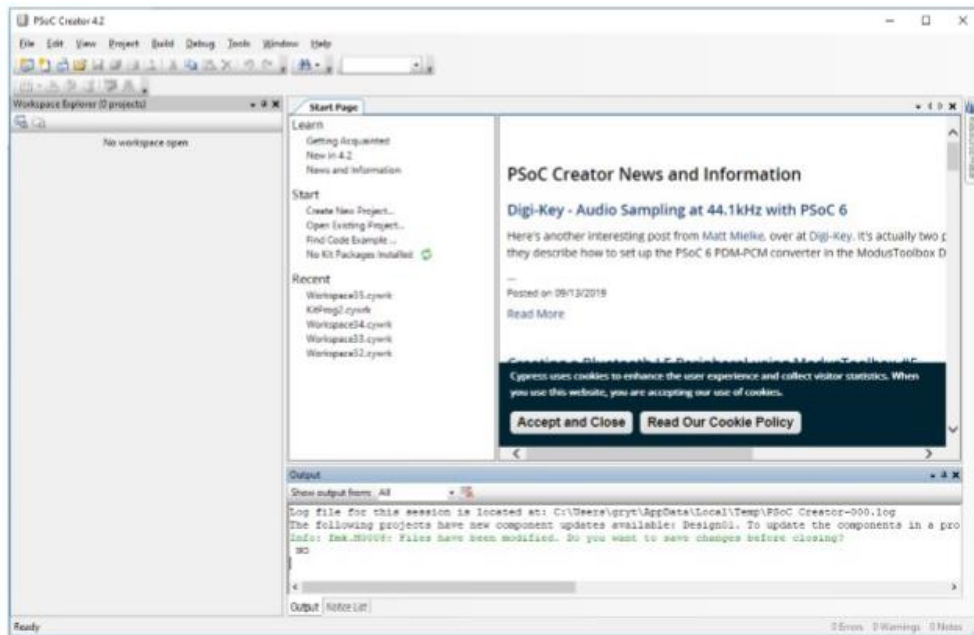


Рис.2. Вікно PSoC Creator

- 2) Обираємо пристрій

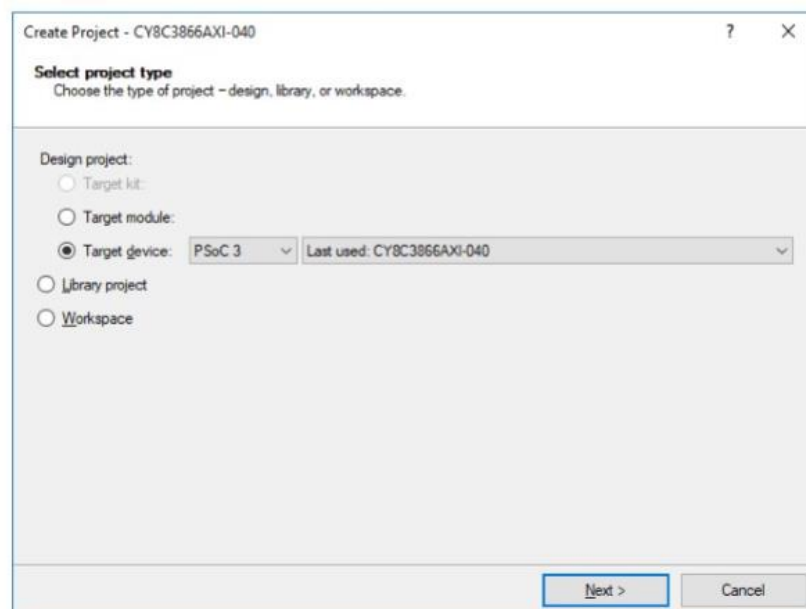


Рис.3. Вибір пристрою

- 3) Обираємо компонент “CharacterLCD” з каталогу компонентів Cypress в пункті та переміщуємо його на «Файл схеми».

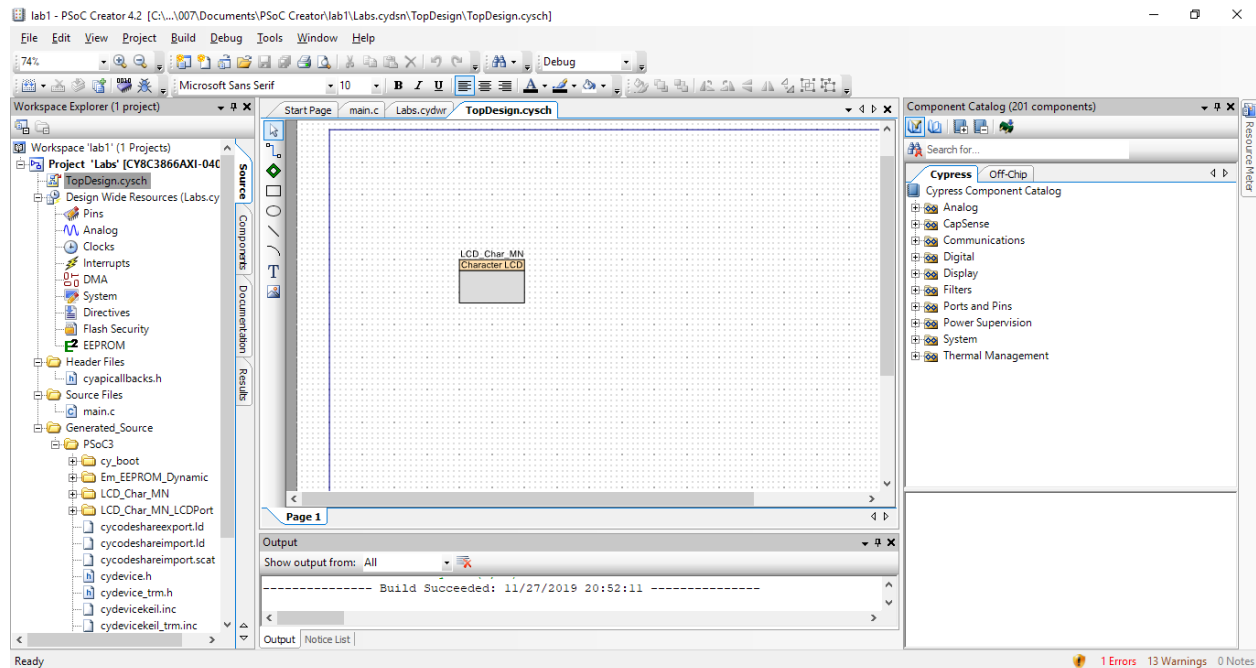


Рис.4. Розміщений компонент CharacterLCD

- 4) Конфігуруємо елемент “LCD_Char_1”: змінюємо ім’я, а в пункті “LCD Custom Character Set” вибираємо “UserDefined” та в пункті “CustomCharacterEditor” вводимо символи відповідно до варіанту.

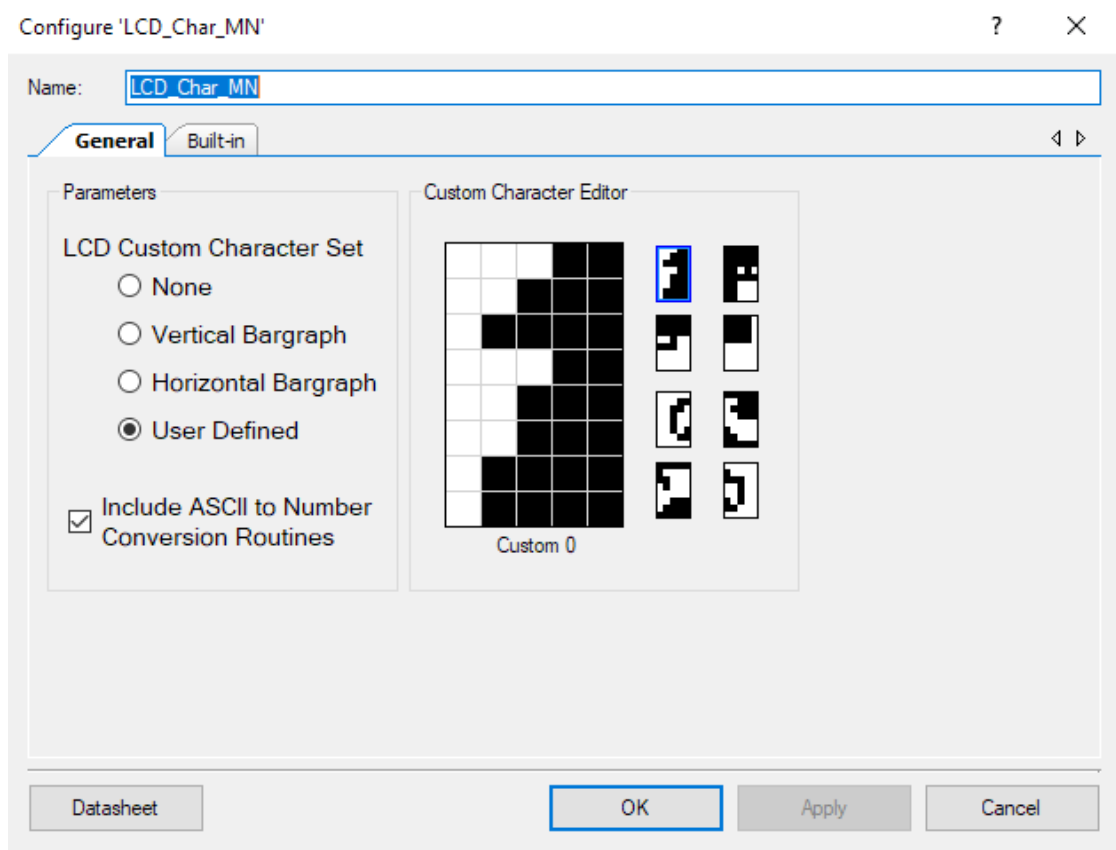


Рис.5. Вибір пристрою

- 5) У файлі main.c пишемо та компілюємо код програми

```
Ax51.exe Generated_Source\PSoC3\CyBootAsmKeil.a51 CASE EP "INCDIR(., Generated_Source\PSoC3)" DB MACRO DF(DEBUG) NOCO PR(.\\DP8051\\I
Ax51.exe Generated_Source\PSoC3\\cymem.a51 CASE EP "INCDIR(., Generated_Source\PSoC3)" DB MACRO DF(DEBUG) NOCO PR(.\\DP8051\\DP8051_Ke
Ax51.exe Generated_Source\PSoC3\\KeilStart.a51 CASE EP "INCDIR(., Generated_Source\PSoC3)" DB MACRO DF(DEBUG) NOCO PR(.\\DP8051\\DP805
LIX51.exe TRANSFER '.\\DP8051\\DP8051_Keil_951\\Debug\\CyDmac.obj', '.\\DP8051\\DP8051_Keil_951\\Debug\\CyFlash.obj', '.\\DP8051\\DP8051_Kei
LX51.exe '.\\DP8051\\DP8051_Keil_951\\Debug\\main.obj', '.\\DP8051\\DP8051_Keil_951\\Debug\\cyfitter_cfg.obj', '.\\DP8051\\DP8051_Keil_951\\De
*****
* RESTRICTED VERSION WITH 0800H BYTE CODE SIZE LIMIT; USED: 041AH BYTE (51%) *
*****
Program Size: data=14.0 xdata=102 const=109 code=3370
OHex51.exe Labs.omf HEXFILE(Labs.ihx)
c8051-elf-omf2elf.exe Labs.omf Labs.elf
CyHexTool -o "C:\\Users\\007\\Documents\\PSoC Creator\\lab1\\Labs.cydsn\\DP8051\\DP8051_Keil_951\\Debug\\Labs.hex" -f "C:\\Users\\007\\Documents
Flash used: 3479 of 65536 bytes (5,3%).
SRAM used: 102 of 8192 bytes (1,2%).
----- Build Succeeded: 11/27/2019 20:52:11 -----
```

Рис.6. Результат компіляції

6) Налаштовуємо виводи проекту

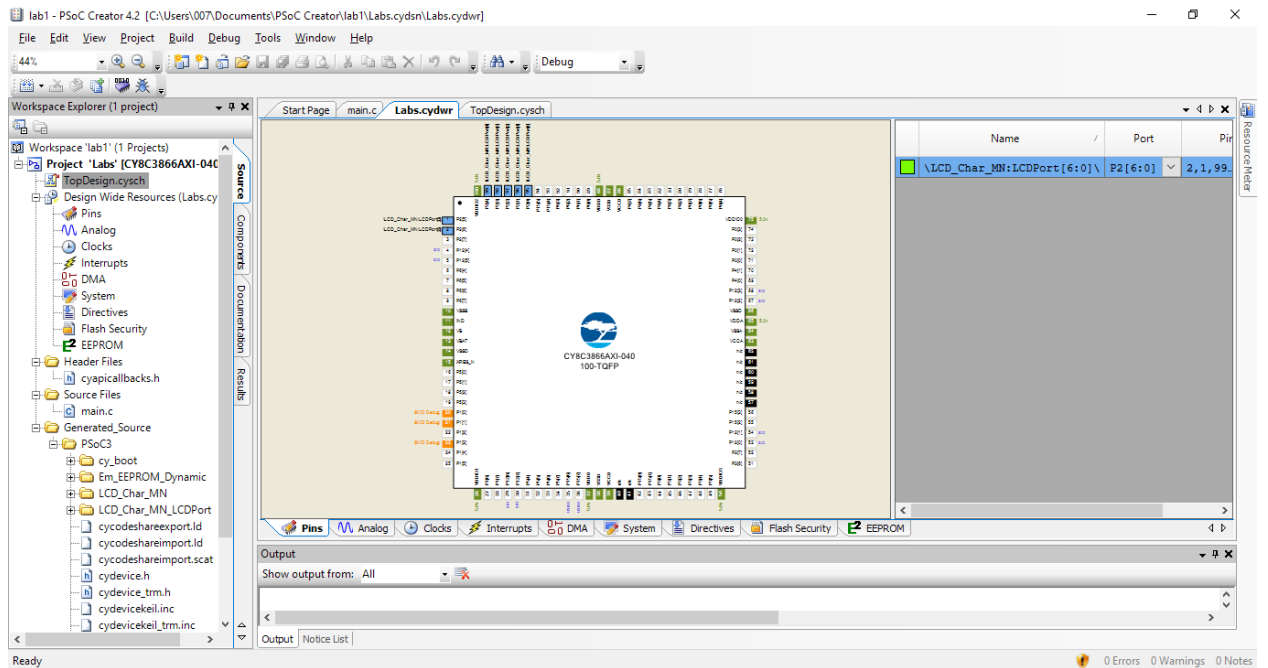


Рис.7. Вибір pinів

7) Для прошивання файлу у PSoC5 запускаємо програму PSoC Programmer.



Рис.8. Перевірка роботи дисплею

Код програми

```
#include "project.h"
```

```

void DrawGun(uint8 y, uint8 x)
{
    LCD_Char_MN_Position(y, x);
    LCD_Char_MN_PutChar(LCD_Char_MN_CUSTOM_0);

    LCD_Char_MN_Position(y, x+1);
    LCD_Char_MN_PutChar(LCD_Char_MN_CUSTOM_1);

    LCD_Char_MN_Position(y, x+2);
    LCD_Char_MN_PutChar(LCD_Char_MN_CUSTOM_2);

    LCD_Char_MN_Position(y, x+3);
    LCD_Char_MN_PutChar(LCD_Char_MN_CUSTOM_3);
}

void DrawBall(uint8 y, uint8 x)
{
    LCD_Char_MN_Position(y, x);
    LCD_Char_MN_PutChar(LCD_Char_MN_CUSTOM_4);

    LCD_Char_MN_Position(y, x+1);
    LCD_Char_MN_PutChar(LCD_Char_MN_CUSTOM_5);

    LCD_Char_MN_Position(y, x+2);
    LCD_Char_MN_PutChar(LCD_Char_MN_CUSTOM_6);

    LCD_Char_MN_Position(y, x+3);
    LCD_Char_MN_PutChar(LCD_Char_MN_CUSTOM_7);
}

int main(void)
{
    uint8 iter = 0u;
    LCD_Char_MN_Start();
    for(;;)
    {
        CyDelay(1200u);
        LCD_Char_MN_ClearDisplay();

        if(iter == 3u)
        {
            DrawGun(1u, 5u);
            DrawBall(1u, 8u);
            iter = 0u;
            continue;
        }
        if(iter == 2u)
        {
            DrawGun(0u, 4u);
            DrawBall(0u, 8u);
            iter += 1u;
            continue;
        }
        if(iter == 1u)
        {
            DrawGun(1u, 0u);
            DrawBall(1u, 12u);
            iter += 1u;
            continue;
        }
        if(iter == 0u)
        {
            DrawGun(0u, 0u);
            DrawBall(0u, 12u);

```

```
        iter += 1u;  
        continue;  
    }  
  
}  
  
}
```

Висновок: під час виконання роботи було вивчено принципи роботи стану PSoC 3 та практично засвоєно базові навички проектування систем на кристалі, засвоєно методи та засоби конфігурування програмованих систем на кристалі та перевірено їх функціонування на прикладі роботи із символьним дисплеєм.