

## การทดลองที่ 2

### การเชื่อมต่อไมโครคอนโทรลเลอร์ PSoC กับโมดูลแอลซีดีแบบดอทแมทริก (Interfacing the PSoC microcontroller with Dot Matrix LCD Module.)

วัตถุประสงค์ เพื่อให้สามารถใช้งานโมดูลต่างๆภายใน PSoC ได้

#### บทนำ

การทดลองนี้ ได้แสดงวิธีการใช้งานโมดูลแอลซีดีแบบดอทแมทริก เพื่อให้เห็นข้อความแบบต่างๆ

#### การทดลองที่ 2.1 แสดงข้อความแบบตัวอักษรคงที่

การทดลอง เนื่องจากขั้นตอนต่างๆได้กล่าวไว้โดยละเอียดแล้วในการทดลองที่ 1 ดังนั้นในที่นี้จะกล่าวถึงเพียงย่อๆ หากนักศึกษายังไม่เข้าใจให้ย้อนกลับไปดูการทดลองที่ 1 สำหรับในการทดลองนี้จะได้อธิบายอย่างละเอียดเกี่ยวกับขั้นตอนการเรียกใช้โมดูลและการกำหนดค่าพารามิเตอร์ของโมดูล

1. ให้รัน โปรแกรม PSoC Designer แล้วทำการสร้างโปรเจกใหม่ โดยใช้ไมโครคอนโทรลเลอร์เบอร์ CY8C29466-24PXI ส่วน Main file ให้เลือกเป็นภาษา C
2. เมื่อเข้าสู่ PSoC designer ในหน้าแรกจะเป็น User Module Selection View ดังนี้

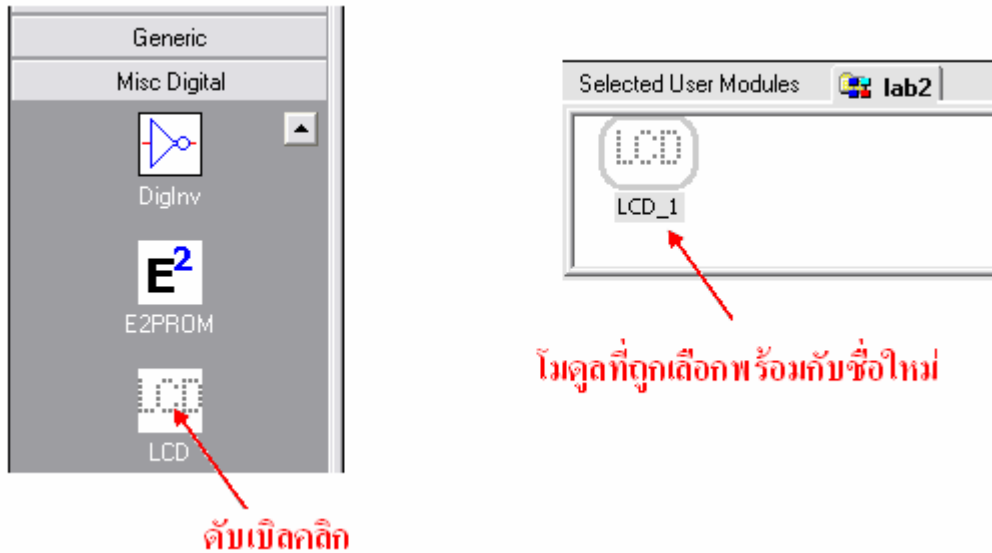
The screenshot shows the PSoC Designer interface for a project named 'lab2'. The main window displays the 'Incremental ADC Data Sheet' for 'ADCINC v1.1'. The data sheet includes a table of resources and their configurations for different PSoC models.


| Resources                       | PSoC™ Blocks |           | API Memory (Bytes) |         | Pins (per External ID) |   |   |
|---------------------------------|--------------|-----------|--------------------|---------|------------------------|---|---|
|                                 | Digital      | Analog CT | Flash              | RAM     |                        |   |   |
| Order Modulators                | 1st & 2nd    | 1st 2nd   | 1st 2nd            | 1st 2nd |                        |   |   |
| CY8C29xxx, CY8C24090, CY7C64215 | 1            | 0         | 1                  | 2       | 273 322                | 8 | 1 |
| CY8C27/22xxx, CY8C24000         | 1            | 0         | 1                  | 2       | 226 275                | 8 | 1 |

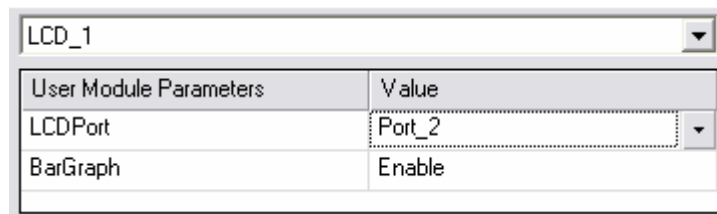
Annotations in Thai:

- Red arrow pointing to the 'ADCINC v1.1' title: "ส่วนแสดงรายละเอียดของโมดูลที่ถูกเลือก (ในขณะนี้อยู่ที่ ADCINC)"
- Red arrow pointing to the 'Resource Meter' table: "ส่วนแสดงจำนวนการใช้ทรัพยากร"
- Red arrow pointing to the circuit diagram: "ส่วนแสดงโมดูลที่ถูกเลือกใช้งาน"



ในหน้าต่างนี้ให้เลือกใช้โมดูล LCD ซึ่งอยู่ในกลุ่ม Misc Digital โดยคลิกที่แถบชื่อ Misc Digital เมื่อมีรายการโมดูลขึ้นมาให้เลื่อนหาโมดูล LCD เมื่อพบให้ดับเบิลคลิกที่สัญลักษณ์ของชื่อโมดูล จะปรากฏโมดูลนั้นในส่วนแสดงโมดูลที่ถูกเลือก และในส่วนแสดงรายละเอียดของโมดูล ก็จะเป็นรายละเอียดของ LCD



3. ให้คลิก  (Interconnect View) เพื่อกำหนดค่าพารามิเตอร์ของ LCD โดยให้ LCD ต่ออยู่ที่ Port\_2 (ตามวงจรในบอร์ดทดลอง) ดังนั้นให้กำหนดค่าพารามิเตอร์ตามรูป








หมายเหตุ หน้าต่าง ต่างๆเช่น หน้าต่าง User Module Selection View หน้าต่าง Interconnect View และหน้าต่าง Application Editor สามารถเปิดขึ้นมาดูเมื่อไรก็ได้ โดยคลิกที่ปุ่มคำสั่งเปิดหน้าต่างนั้นๆ

4. เมื่อเสร็จแล้วให้กด  (Generate Application) เพื่อสร้างซอร์สไฟล์ต่างๆ
5. ให้คลิกที่ปุ่ม  (Application Editor) เพื่อเปิดหน้าต่าง Application เพื่อเขียน Source file ให้แก้ไขโปรแกรม main.c เป็น ดังนี้

## โปรแกรมที่ 2.1

```
//-----  
// C main line  
//-----  
  
#include <m8c.h> // part specific constants and macros  
#include "PSoCAPI.h" // PSoC API definitions for all User Modules  
  
void main()  
{  
    char theStr1[] = "The 5th December"; // Define RAM string 1  
    char theStr2[] = "Father's Day"; // Define RAM string 1  
    LCD_1_Start(); // Initialize LCD  
    LCD_1_Position(0,0); // Place LCD cursor at row 0, col 0.  
    LCD_1_PrString(theStr1); // Print "PSoC LCD" on the LCD  
    LCD_1_Position(1,2); // Place LCD cursor at row 1, col 2.  
    LCD_1_PrString(theStr2); // Print "PSoC LCD" on the LCD  
}
```

แล้วทำการบันทึกไฟล์

6. ให้ทำการแปลโปรแกรมโดยคลิกที่ปุ่ม  (Build) เพื่อแปลโปรแกรม หากไม่มีข้อผิดพลาดจะได้ไฟล์ Hex ที่มีชื่อเหมือนกับชื่อของโปรเจกต์ อยู่ในโฟลเดอร์ Output สำหรับใช้ดาวน์โหลดชิพไมโครคอนโทรลเลอร์
7. ทำการติดตั้ง LCD เข้ากับบอร์ด
8. ให้ต่อสาย USB เข้ากับ PSoC MiniProg แล้วเสียบลงในบอร์ดทดลอง ขั้นตอนนี้อาจเกิดความเสี่ยงต่อคอมพิวเตอร์และบอร์ดทดลองได้  
ถ้า PC เครื่องใดเพิ่งเริ่มทำเป็นครั้งแรก ก็จะถามหาการติดตั้งโปรแกรมสำหรับการเชื่อมต่อแบบ USB ในขั้นตอนนี้อย่าเลือกแบบมัติ เมื่อติดตั้งแล้วโปรแกรมก็พร้อมดาวน์โหลดเพื่อทดสอบ
9. การโปรแกรมลงชิพ ให้ใช้คำสั่ง Program Part ในเมนูโปรแกรม Program > Program Part หรือคลิกที่ปุ่ม 
10. เมื่อปรากฏหน้าต่าง PSoC Programmer ให้เลือก Port กำหนดไฟล์ที่จะดาวน์โหลดโดยคลิกที่ปุ่ม  กำหนด Device Family กำหนด Device แล้วคลิก  เครื่องโปรแกรมก็จะทำการดาวน์โหลดโปรแกรมลงชิพ ในขั้นตอนนี้จะใช้เวลาพอสมควร
11. เมื่อดาวน์โหลดเสร็จ ให้ทำการทดสอบการทำงาน โดยคลิกที่ปุ่ม  เพื่อจ่ายไฟเลี้ยงเข้าบอร์ดทดลอง สังเกตด้านล่างของหน้าต่าง จะปรากฏแถบสีแดงและคำว่า Powered แต่ถ้าไม่มีไฟเลี้ยงเข้าบอร์ดทดลอง ที่ตำแหน่งนี้จะเป็นสีเขียวและปรากฏคำว่า Not Powered
12. ให้สังเกตการทำงาน โดยดูจากข้อความที่ปรากฏบน LCD ถ้าไม่ปรากฏให้ลองปรับ R6 เพื่อปรับความสว่างของ LCD

คำสั่งต่างๆที่เกี่ยวกับโมดูล LCD สามารถดูรายละเอียดได้จากหน้าต่าง User Module Selection View โดยเลือกที่แท็บ API เช่น คำสั่ง LCD\_1\_Start() ก็คือฟังก์ชัน LCD\_Start โดยเปลี่ยนชื่อเป็น LCD\_1\_Start ตามชื่อของโมดูล LCD\_1 คำสั่งอื่นๆ ก็เช่นเดียวกัน

ในการทดลองต่อไปจะแสดงการเรียกใช้ฟังก์ชันแบบต่างๆ ให้นักศึกษาแก้ไขโปรแกรม ให้เป็นไปตามโปรแกรมต่อไปนี้ แล้วตรวจสอบว่าได้ผลการทำงานเป็นเช่นไร

## โปรแกรมที่ 2.2

```
//-----  
// C main line  
//-----  
  
#include <m8c.h> // part specific constants and macros  
#include "PSoCAPI.h" // PSoC API definitions for all User Modules  
#include <stdlib.h>  
  
void main()  
{  
    BYTE Number;  
    WORD i;  
    char TextBuff[6];  
    LCD_1_Start();  
  
    while(1)  
    {  
        for (Number=200;Number>0;Number--)  
        {  
            LCD_1_Position(0,0);  
  
            LCD_1_PrCString("Dec ");  
            itoa(TextBuff,Number,10); //แปลง integer ให้เป็น string ของเลขฐาน 10  
            LCD_1_PrString(TextBuff);  
  
            LCD_1_PrCString(" Hex ");  
            LCD_1_PrHexByte(Number); //Prints a byte as a two-character hex string  
  
            LCD_1_Position(1,0);  
            LCD_1_PrCString("Binary ");  
            itoa(TextBuff,Number,2); //แปลง integer ให้เป็น string ของเลขฐาน 2  
            LCD_1_PrString(TextBuff);  
  
            for (i=1;i<20000;i++)  
            {  
                asm("nop"); // 4 Cycle  
                asm("nop"); // 4 Cycle  
                asm("nop"); // 4 Cycle  
                asm("nop"); // 4 Cycle  
            }  
        }  
    }  
}
```

```
}  
}  
}
```

หมายเหตุ ฟังก์ชัน itoa อยู่ในไฟล์ Header stdlib.h เป็น Library Functions สามารถเปิดดูได้จาก








Help > Document > C Language Compiler User Guide.pdf

### โปรแกรมที่ 2.3

```
//-----  
// C main line  
//-----  
  
#include <m8c.h> // part specific constants and macros  
#include "PSoCAPI.h" // PSoC API definitions for all User Modules  
#include <stdlib.h>  
  
void delay(void)  
{  
    WORD i;  
    for (i=1;i<20000;i++)  
    {  
        asm("nop"); // 4 Cycle  
        asm("nop"); // 4 Cycle  
        asm("nop"); // 4 Cycle  
        asm("nop"); // 4 Cycle  
    }  
}  
  
void main()  
{  
    WORD j, Number;  
    char TextBuff[5];  
    LCD_1_Start();  
  
    while(1)  
    {  
        for (j=0;j<=80;j++)  
        {  
            LCD_1_Position(0,0);  
            LCD_1_PrCString("Progress Bar ");  
            Number = j*1.25;  
            itoa(TextBuff,Number,10);  
            LCD_1_PrString(TextBuff);  
            LCD_1_PrCString("%");  
  
            LCD_1_InitBG(LCD_1_SOLID_BG); //กำหนดให้ LCD ทำงานแบบแท่งกราฟ  
            LCD_1_DrawBG(1,0,16,j);  
            delay();  
        }  
    }  
}
```

}

## สรุปขั้นตอนการใช้งาน PSoC Designer

| ขั้นตอนที่ | รายการ   | ปุ่มคำสั่ง   |
|------------|--|--|
| 1          | สร้างโปรเจกต์  | Start new project     |
| 2          | กำหนดเบอร์ IC และภาษาที่ต้องการเขียน                   |  |
| 3          | เลือกอุปกรณ์ภายในชิพ ทำการวางอุปกรณ์ ดูรายละเอียดโมดูล | Device Editor         |
| 4          | กำหนดพารามิเตอร์                                       | Interconnect View     |
| 5          | สร้างระบบ  | Generate Application  |
| 6          | เขียนโปรแกรม Application                               | Application Editor    |
| 7          | แปลโปรแกรม   | Build                 |
| 8          | Download .Hex → chip                                   | Program Part          |
| 9          | Test program   |  |

### คำถาม

1. ให้สรุปผลที่ได้จากการทดลองของแต่ละโปรแกรม
2. เขียนโปรแกรมเมนู มีตัวเลือกในเมนูไม่น้อยกว่า 5 ตัวเลือก ใช้พอร์ต P1.0 เป็นตัวเลื่อนเมนู และ P1.1 เป็น Enter

คำแนะนำ ใช้คำสั่ง switch...case

```

switch(bTest1)
{
    case 4:
    {
        bTest2 = 1;
        break;
    }
    case 3:
    {
        bTest2 = 2;
        break;
    }
    case 2:
    {
        bTest2 = 3;
        break;
    }
    default:
    {
        bTest2 = 4;
    }
}

```