

การทดลองที่ 6

การใช้งานไมโครคอนโทรลเลอร์ PSoC กับ Real-time clock (Using the DS1302 real-time clock with PSoC microcontroller.)

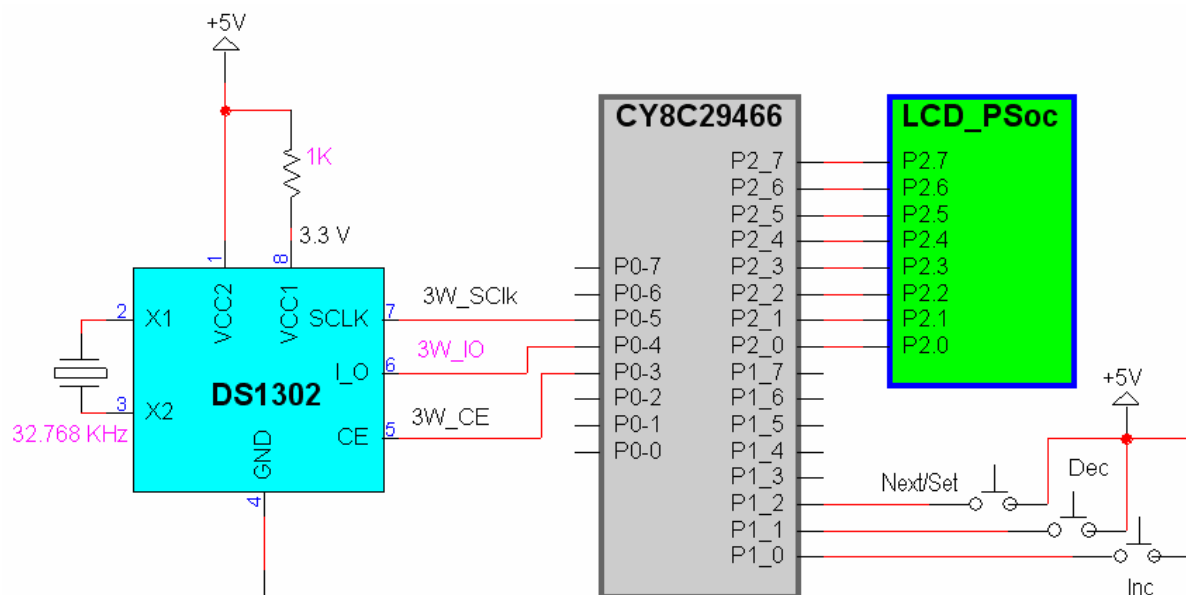
วัตถุประสงค์

1. เพื่อให้เข้าใจถึงวิธีการอินเตอร์เฟสแบบ 3 Wire
2. เพื่อให้เข้าใจการทำงานของ Real-time clock
3. สามารถใช้งานไมโครคอนโทรลเลอร์ PSoC ทำงานร่วมกับ Real-time clock โดยผ่านอินเตอร์เฟสแบบ 3-Wire ได้

บทนำ


การทดลองนี้ ได้แสดงวิธีการใช้งานไอซี Real-time clock เบอร์ DS1302 ซึ่งมีการอินเตอร์เฟสแบบ 3-Wire โดยผ่านการควบคุมของไมโครคอนโทรลเลอร์ PSoC

รูปที่ 6.1 เป็น Logic Diagram ของวงจรที่ใช้ทดลอง โดยมี DS1302 ทำหน้าที่เป็นตัวสร้างค่าเวลา และวันเดือนปี LCD ใช้สำหรับการแสดงผล ส่วนสวิทช์ทั้ง 3 ตัวใช้สำหรับการตั้งเวลาและวันที่ รายละเอียดของ DS1302 และ ไลบรารี สำหรับ DS1302 สามารถศึกษาได้จาก <http://narong.ee.engr.tu.ac.th/microlab/doc/index.html>



รูปที่ 6.1 การเชื่อมต่อ PSoC กับ Real-time clock DS1302 และ LCD



การทดลองที่ 6.1 แสดงการใช้งาน DS1302 ผ่านไลบรารี DS1302.h

1. ให้รันโปรแกรม PSoC Designer แล้วทำการสร้างโปรเจกใหม่ โดยใช้ไมโครคอนโทรลเลอร์เบอร์ CY8C29466-24PXI ส่วน Main file ให้เลือกเป็นภาษา C
2. เมื่อเข้าสู่ PSoC designer ในหน้าแรกจะเป็น User Module Selection View ในหน้าต่างนี้ให้เลือกใช้ โมดูล แอลซีดี (LCD)
3. หลังจากเลือกโมดูลต่างๆแล้ว ให้คลิก  (Interconnect View) เพื่อทำการกำหนดค่าพารามิเตอร์ของพอร์ตต่างๆ ให้กับหนดดังนี้

```
Port_0_3 P0[3] StdCPU Strong DisableInt
Port_0_4 P0[4] StdCPU Pull Up DisableInt
Port_0_5 P0[5] StdCPU Strong DisableInt

Port_1_0 P1[0] StdCPU Pull Down DisableInt
Port_1_1 P1[1] StdCPU Pull Down DisableInt
Port_1_2 P1[2] StdCPU Pull Down DisableInt
```

ส่วน Global Resource ไม่ต้องเปลี่ยนแปลงใดๆ

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

โปรแกรมที่ 6.1

```
//-----
// Experiment 6.1 Using the DS1302 real-time clock with PSoC microcontroller.)
//-----

#include <m8c.h> // part specific constants and macros
#include "PSoCAPI.h" // PSoC API definitions for all User Modules

BYTE Time_date[7]={0x10,0x12,0x01, // Seconds and minutes in BCD 1:12:10am
                  0x02,0x01, // Date-Month
                  0x03, // Day of Week, Wendsday
                  0x08}; // year 08

#define sec Time_date[0]
#define min Time_date[1]
#define hrs Time_date[2]
#define dte Time_date[3]
#define mon Time_date[4]
#define day Time_date[5]
#define yr Time_date[6]

#include "ds1302.h"
```

```

void main()
{
    LCD_1_Start();           // Initialize LCD
    init_ds1302();          // Initialize DS1302
    setup_time_date();      // Setup user time

    while (1)
    {
        LCD_1_Position(0,0);
        LCD_1_PrCString("Time: ");

        read_time_date();
        LCD_1_PrHexByte(hrs);
        LCD_1_PrCString(":");
        LCD_1_PrHexByte(min);
        LCD_1_PrCString(":");
        LCD_1_PrHexByte(sec);

        LCD_1_Position(1,0);
        LCD_1_PrCString("Date: ");
        LCD_1_PrHexByte(dte);
        LCD_1_PrCString("/");
        LCD_1_PrHexByte(mon);
        LCD_1_PrCString("/");
        LCD_1_PrHexByte(yr);
    }
}

```

6. ให้ทำการบันทึกและแปลโปรแกรม
7. ประกอบวงจรตามรูปที่ 6.1 (การทดลองที่ 6.1 นี้ยังไม่ต้องใช้สวิทช์ก็ได้) เสร็จแล้วดาวน์โหลดโปรแกรมลงชิพ
8. ตรวจสอบการทำงาน

คำถาม

1. ให้อธิบายการทำงานของโปรแกรม ที่ 6.1
2. ให้เขียนโปรแกรมเพื่อทำเป็นนาฬิกาที่บอกได้ทั้งเวลาและวันที่ และสามารถตั้งเวลาได้โดยใช้สวิทช์ (ที่ตัวก็ได้ แต่ต้องตั้งได้หมดทุกค่า)

หมายเหตุ สำหรับข้อสองการคิดคะแนนจะคิดจาก 100% หากด้วยจำนวนกลุ่มที่ใช้โมดูลแบบเดียวกัน