

การทดลองที่ 7

การออกแบบดิจิตอลเทอร์โมมิเตอร์โดยใช้ DS18B20

(Design of a Digital thermometer using the DS18B20.)

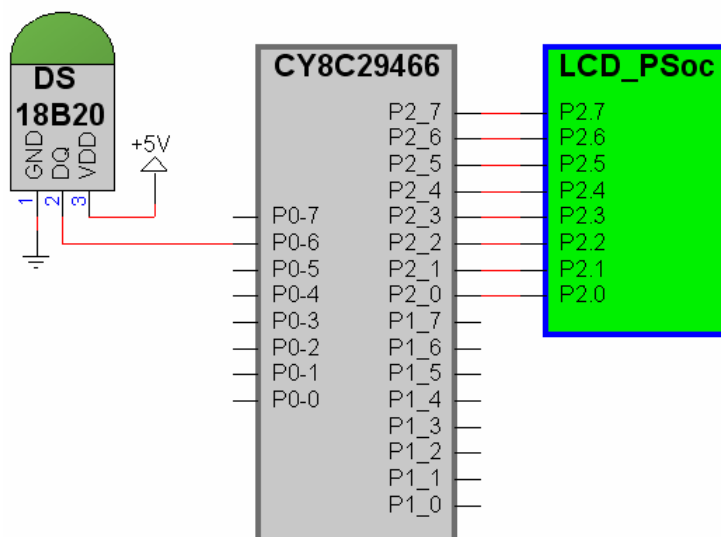
วัตถุประสงค์

1. เพื่อให้เข้าใจถึงวิธีการอินเทอร์เฟซแบบ 1-Wire
2. เพื่อให้เข้าใจการทำงานของ DS18B20 Programmable Resolution 1-Wire Digital Thermometer
3. สามารถใช้งานไมโครคอนโทรลเลอร์ PSoC ทำงานร่วมกับ DS18B20 Programmable Resolution Digital Thermometer โดยผ่านอินเทอร์เฟซแบบ 1-Wire ได้

บทนำ

การทดลองนี้ ได้แสดงวิธีการใช้งานไอซี ดิจิตอลเทอร์โมมิเตอร์ โดยใช้ DS18B20 ซึ่งเป็นไอซี ดิจิตอลเทอร์โมมิเตอร์แบบโปรแกรมค่าความละเอียด และใช้อินเทอร์เฟซแบบ 1-Wire โดยผ่านการควบคุมของไมโครคอนโทรลเลอร์ PSoC ซึ่งได้จัดเตรียม API (Application Programming Interface) สำหรับ 1-Wire ไว้ให้แล้ว ก่อนอื่น นศ.ต้องศึกษารายละเอียดการจัดเตรียม 1-Wire User module และรายละเอียดเกี่ยวกับ 1-Wire interface ได้จาก <http://narong.ee.engr.tu.ac.th/microlab/doc/index.html>

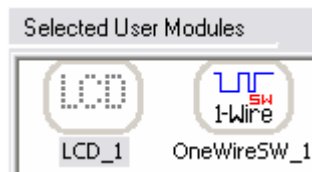
รูปที่ 7.1 เป็น โลจิกไดอะแกรมของวงจรที่ใช้ทดลอง โดยมี DS18B20 ทำหน้าที่เป็นตัวแปลงค่าอุณหภูมิให้เป็นค่าดิจิตอล โดยมี LCD ทำหน้าที่เป็นภาคแสดงผล




รูปที่ 7.1 การเชื่อมต่อ PSoC กับ Real-time clock DS1302 และ LCD

การทดลองที่ 7.1 แสดงการใช้งาน DS18B20 ผ่านไลบรารี OneWireSW

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) และ โมดูล OneWireSW ที่อยู่ในกลุ่ม Digital Comm



3. หลังจากเลือกโมดูลต่างๆแล้ว ให้คลิก  (Interconnect View) เพื่อทำการกำหนดค่าพารามิเตอร์ของพอร์ตต่างๆ โดยให้กำหนดพอร์ต LCD เป็นพอร์ต 2 ส่วนพอร์ต OneWireSW ให้ใช้พอร์ต 0 ดังนี้



User Module Parameters	Value
DQIOPort	Port_0
DQPin	Port_0_6
CRC16	Disable
Search	Disable

เมื่อกำหนดแล้วจะสังเกตได้ว่า ที่พอร์ต P0[6] จะถูกกำหนดให้เป็น Pull Up โดยอัตโนมัติ

OneWireSW_1DQPin	P0[6]	StdCPU	Pull Up	DisableInt
------------------	-------	--------	---------	------------

ส่วน Global Resource ให้กำหนดความถี่ของ CPU_Clock เป็น 12 MHz คือ SysClk/2

Global Resources	Value
Power Setting [Vcc / SysClk freq	5.0V / 24MHz
CPU_Clock	SysClk/2
32K_Select	Internal

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

โปรแกรมที่ 7.1

```

/*****
* FILENAME:    main.c for DS18B20
* DESCRIPTION:    Read Temperator From DS18B20
*                Show on LCD Display    in text and graphs format
*****/

```

```

#include <m8c.h>
#include "PSoCAPI.h"
#include <stdlib.h>
char TextBuff[5];
unsigned char TLSB, TMSB, TempI, j;
int TempP;

void main()
{
    LCD_1_Start();
    LCD_1_Position(0,0);
    LCD_1_PrCString("1-Wire & DS18B20");
    OneWireSW_1_Start(); // Initialize 1-Wire pin

    while(1)
    {
        OneWireSW_1_Reset(); // Reset the 1-Wire device
        OneWireSW_1_WriteByte(0xCC); // Skip ROM command
        OneWireSW_1_WriteByte(0x44); // Convert Temperature command
        OneWireSW_1_Delay10mTimes(90); // Wait 900 ms for the temperature conversion

        OneWireSW_1_Reset(); // Reset the 1-Wire device
        OneWireSW_1_WriteByte(0xCC); // Skip ROM command
        OneWireSW_1_WriteByte(0xBE); // Read Scratch Pad command

        TLSB = OneWireSW_1_ReadByte(); // Read LSB data
        TMSB = OneWireSW_1_ReadByte(); // Read MSB data

        TempP = (TMSB << 4) & 0xF0; // Integer number
        TempI = (TLSB >> 4) & 0x0F;
        TempI = TempI | TempP; // Temperature in interger format
        TempP = TLSB & 0x0F; // Binary point of temperature

        itoa(TextBuff,TempI,10); // Convert data to ASCII
        LCD_1_Position(1,0); // Set LCD position
        LCD_1_PrCString("TEMP = "); // Print data to LCD
        LCD_1_PrString(TextBuff);

        TempP = (TempP & 0x000f)* 625; // Convert data to decimal point for display

        itoa(TextBuff,TempP,10); // Display decimal point
        if (TempP < 1000) LCD_1_PrCString(".0");
        else LCD_1_PrCString(".");
        LCD_1_PrString(TextBuff);
        LCD_1_PrCString(" C ");
    }
}

```

6. ให้ทำการบันทึกและแปลโปรแกรม
7. ประกอบวงจรตามรูปที่ 7.1 เสร็จแล้วดาวน์โหลดโปรแกรมลงชิพ
8. ตรวจสอบการทำงาน

คำถาม

1. ให้อธิบายการทำงานของโปรแกรม ที่ 7.1

2. ให้เขียนโปรแกรมเพื่อแสดงผลอนุกรมในรูปแบบอื่นๆ เช่นบอกทั้งเป็นตัวเลขและกราฟแท่ง