

```

//*****
// Barometer and Temperature for LP810-DEMO board
//
// (C)Copyright 2013 All rights reserved by Y.Onodera
// http://einstlab.web.fc2.com
//*****
#include <stdio.h>
#include <string.h>
#include "LPC8xx.h"
#include "mrt.h"
#include <stdlib.h>
#include "type.h"
#include "lpc8xx_clkconfig.h"
#include "lpc8xx_i2c.h"
#include "I2C.h"
#include "LCD.h"
#include "ADT7410.h"
#include "LPS331.h"
#include "LED.h"

#if defined(__CODE_RED)
// #include <cr_section_macros.h>
// #include <NXP/crp.h>
// __CRP const unsigned int CRP_WORD = CRP_NO_CRP ;
#endif

void SwitchMatrix_Init();
//void IOCON_Init();

int main(void)
{
    char msg[10];
    int temp;
    int l;
    long press;

    // IRC 12MHz
    /* Configure the core clock/PLL via CMSIS */
    SystemCoreClockUpdate();

    /* Initialize the GPIO block */
    // gpioInit();

    /* Initialize the UART0 block for printf output */
    // uart0Init(115200);

    /* Configure the switch matrix (setup pins for UART0 and GPIO) */
    SwitchMatrix_Init();
    // IOCON_Init();

    InitLED();
    InitI2C();
    mrtInit(SystemCoreClock/1000);

    InitLCD();

```

```

// set 16bit resolution
PutI2C(ADT7410_CONFIG, 0xC0, ADT7410_ADDR);
// Power ON Cycle=1Hz
PutI2C(LPS331_CTRL_REG1, 0x90, LPS331_ADDR);

#define PCF8591_ADDR 0x92
#define M24LC64_ADDR 0xA0
// #define EEPROM
// #define ADC
#define TEMP
#define BAR

    while(1)
    {

#ifdef EEPROM
        // EEPROM 0x0002=0x5A
        PutEEPROM(0x00, 0x02, 0x5A, M24LC64_ADDR);
        WaitN(5);

        for(l=0;l<0x1FFF;l++){
            // EEPROM for 24LC64
            GetEEPROM(l>>8, l&0xFF, M24LC64_ADDR);
            temp=I2CMasterRXBuffer[0];
            LocateLCD(0,0);
            PutsLCD("ADR=");
            itoa(l,msg,16);
            PutsLCD(msg);
            LocateLCD(0,1);
            PutsLCD("DAT=");
            itoa(temp,msg,16);
            PutsLCD(msg);
            PutsLCD(" ");
            onLED();
            WaitN(500);
            offLED();
            WaitN(500);
        }
#endif

#ifdef ADC
        // A/D for PCF8591
        GetI2C(0x40, PCF8591_ADDR); // ch0 with D/A enable
        // GetI2C(0x01, PCF8591_ADDR); // ch1
        // GetI2C(0x02, PCF8591_ADDR); // ch2
        // GetI2C(0x03, PCF8591_ADDR); // ch3
        temp=I2CMasterRXBuffer[0];
        itoa(temp,msg,10);
        LocateLCD(0,0);
        PutsLCD("A/D=");
        PutsLCD(msg);
        PutsLCD(" ");
        onLED();
        // D/A for PCF8591
        PutI2C(0x40, temp, PCF8591_ADDR);
        WaitN(500);
        offLED();
#endif
    }
}

```

```

        WaitN(500);
#endif

#ifdef TEMP
    // Temperature
    // RDY?
    do{
        GetI2C(ADT7410_STATUS, ADT7410_ADDR);
        temp=I2CMasterRXBuffer[0];
    }while(temp & 0x80);

    // get temp_high
    GetI2C(ADT7410_TEMP_H, ADT7410_ADDR);
    temp=I2CMasterRXBuffer[0]*0x100;
    // get temp_low
    GetI2C(ADT7410_TEMP_L, ADT7410_ADDR);
    temp+=I2CMasterRXBuffer[0];
//    temp/=128;    // for C
    temp/=13;
    itoa(temp,msg,10);
    l=strlen(msg);
    msg[l]=msg[l-1];
    msg[l-1]='.';
    msg[l+1]=0;
    LocateLCD(0,0);
    PutsLCD(msg);
    PutsLCD("C ");
    onLED();
    WaitN(500);
#endif

#ifdef BAR
    // Barometer
    // get press_high
    GetI2C(LPS331_PRESS_OUT_H, LPS331_ADDR);
    press=I2CMasterRXBuffer[0]*0x10000;
    // get tpress_low
    GetI2C(LPS331_PRESS_OUT_L, LPS331_ADDR);
    press+=I2CMasterRXBuffer[0]*0x100;
    // get press_xl
    GetI2C(LPS331_PRESS_POUT_XL_REH, LPS331_ADDR);
    press+=I2CMasterRXBuffer[0];
//    press/=4096;    // for hPa
    press/=41;
    itoa(press,msg,10);
    LocateLCD(0,1);
    PutsLCD(msg);
    PutsLCD("Pa ");
    offLED();
    WaitN(500);
#endif

    }

}

```