

```

lcd_i2c_mssp
/*****
* LCD表示関数：I2Cインターフェースを呼出し使用
* By nobcha (c)2011,2012
*
* revised for MSSP MASTER mode 06/04/2012
*
* SUNLIKE社LCD アドレス 0x74
* NXP PCF2119xデータシート参照
*
* 平行8ビット接続のときの関数に合わせる
* lcd_init(void) ----- 初期化
* lcd_cmd(command) ----- コマンド出力
* lcd_data(byte) ----- 1文字表示
* 文字コード空間の関係で0x80をオアし出力
* lcd_str(string) ----- 文字列表示
* lcd_clear(void) ----- LCD画面全消去
* lcd_goto(cursor_pos) --- カーソル位置指定
* lcd_cgram(addr, font) --- CGRAMへフォント格納
*****/
#define _LEGACY_HEADERS
#include <htc.h>
#include "mssp_i2c.h"
#define LCD_AD 0x74
#define _XTAL_FREQ 8000000

/*****
* V LCD 設定
*****/
// #define VLCD 8 // Vlcd for 2.46V
// #define VLCD 15 // Vlcd for 3.02V
// #define VLCD 27 // Vlcd for 3.98V
#define VLCD 39 // Vlcd for 4.94V

/*****
* LCDへ表示データ1文字出力
*****/
void lcd_data(unsigned char data){
    i2c_writeto(LCD_AD); // LCDアドレスをOPEN

    i2c_putbyte(0x40); // CO:0 (STOP) ,RS:1 データなので

    i2c_putbyte(data | 0x80); // 表示バイトを送る (文字コードは0x80以上)

    i2c_stop(); // stop コンディション
}

/*****
* 液晶へ1コマンド出力
*****/
void lcd_cmd(unsigned char cmd){
    i2c_writeto(LCD_AD); // LCDアドレスをOPEN

    i2c_putbyte(0x00); // CO:0 (STOP) ,RS:0 コマンドです

    i2c_putbyte(cmd); // コマンドバイトを送る

    i2c_stop(); // stop

    if((cmd == 0x01) || (cmd == 0x02)) // ClearかHomeかで待ち時間調整
        __delay_ms(2); // Clearなら応答時間は2msec待ち
    else
        __delay_us(40); // それ以外コマンドなら50μsec待ち
}

/*****
* 液晶画面出力カーソル位置指定
*****/
void lcd_goto(unsigned char cursor){ // カーソルアドレスは00~0F, 40~4F
    i2c_cmd(0x80 | cursor); // DRAMアドレス設定#7: 1にして7ビット
}

/*****
* LCD初期化処理
*****/

```

## lcd\_i2c\_mssp

```

/*****
* LCD初期化処理 連続バイト
*****/
void lcd_init2(void) {
    __delay_ms(100);          // PCF2119マニュアル掲載お約束
    lcd_cmd(0x30);           // 100mS delay 電源安定まで待つ
                             // 8bit long IF
    __delay_ms(5);          // 5mS delay
    lcd_cmd(0x30);           // 8bit
    __delay_ms(5);          // 5mS delay
    lcd_cmd(0x30);           // 8bit
    __delay_us(40);          // 40uS delay

    i2c_writeto(LCD_AD);     // LCDアドレスをOPEN
    i2c_putbyte(0x00);       // C0:0 (STOP), RS:0 コマンドです

    i2c_putbyte(0x35);       // FUNCTION_set 8bit M:1:2line H:0
    i2c_putbyte(0x41);       // voltage multi
    i2c_putbyte(0x80|VLCD);   // Vlcd Va Set
    i2c_putbyte(0xC0|VLCD);   // Vlcd Vb Set
    i2c_putbyte(0x34);       // FUNCTION_set 8bit M:1:2line H:0

    i2c_stop();             // stop

    lcd_cmd(0x01);           // Clear display
    __delay_ms(100);         // 100mS待ち
    __delay_ms(100);         // 100mS待ち
    __delay_ms(100);         // 100mS待ち
    __delay_ms(100);         // 100mS待ち

    lcd_cmd(0x0F);           // DISP on, Cursor on
    lcd_cmd(0x06);           // Entry mode I_D:1 S:0
}

/*****
* LCD画面全消去
*****/
void lcd_clear(void) {
    lcd_cmd(0x01);          //初期化コマンド0x01出力
}

/*****
* 文字列出力
*****/
void lcd_str(char* ptr) {
    while(*ptr != 0){
        lcd_data(*ptr++);   // 0x00があるまで文字取り出し
    }                       // 文字表示して、次の文字へ
}

/*****
* CGRAMへ外部フォントを格納：未デバッグです
*****/
lcd_ogram(unsigned char addr, char* font){
    unsigned char i;
    i2c_start();            // i2c start
    i2c_write(0x74);        // i2c addr
    i2c_write(0);           // Function set C0:0, RS:0,
    i2c_write(0x24);        // 2 lines 16 chars 1/18 duty ext com
    i2c_write(addr);        // set ogram addr
    i2c_start();            // Restart condition
    i2c_write(0x74);        // i2c addr
    i2c_write(0x40);        // next byte is data
    for (i=0; i<8; i++){
        i2c_write(*font++); // set ogram addr
    }
    i2c_stop();
}

```