

```

/*****
* LCD表示関数：I2Cインターフェースを呼出し使用
* By nobcha (c)2011
*
* revised for MSSP MASTER mode 09/25/2011
*
* STRAWBERRY LINUX社LCD アドレス 0x7c
* AN001とかST7032iのデータシート参照
*
* パラレル4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
* lcd_str(string) ----- 文字列表示
* lcd_clear(void) ----- LCD画面全消去
*
*****/
#define _LEGACY_HEADERS
#include <htc.h>
#include "mssp_i2c.h"
#define _XTAL_FREQ 8000000

#define LCD_AD 0x7C //

/*****
* コントラスト調整用定数 電圧依存
*****/
// #define CONTRAST 0x20 // for 5.0V
// #define CONTRAST 0x2F // for 3.3V
#define CONTRAST 0x3F // for 2.6V
/*****
* 記号アイコン表示定義
*****/
unsigned char ICON[14][2]={
    {0x00, 0x10}, // アンテナ
    {0x02, 0x10}, // 電話
    {0x04, 0x10}, // 無線
    {0x06, 0x10}, // ジャック
    {0x07, 0x10}, // △
    {0x07, 0x08}, // ▽
    {0x07, 0x18}, // △▽
    {0x09, 0x10}, // 鍵
    {0x0B, 0x10}, // ピン
    {0x0D, 0x02}, // 電池無し
    {0x0D, 0x12}, // 容量少
    {0x0D, 0x1A}, // 容量中
    {0x0D, 0x1E}, // 容量多
    {0x0F, 0x10}, // 丸
};

/*****
* LCDへ表示データ1文字出力
*****/
void lcd_data(unsigned char data){
    _delay_us(20);
    i2c_writeto(LCD_AD); // LCDアドレスをOPEN
    _delay_us(20);
    i2c_putbyte(0x40); // CO:0 (STOP), RS:1 データなので
    _delay_us(20);
    i2c_putbyte(data); // 表示バイトを送る
    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){
    _delay_us(20);
    i2c_writeto(LCD_AD); // LCDアドレスをOPEN
    _delay_us(20);
    i2c_putbyte(0x00); // CO:0 (STOP), RS:0 コマンドです
    _delay_us(20);
    i2c_putbyte(cmd); // コマンドバイトを送る
}

```

```

lcd_i2c_mssp

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 // それ以外コマンドなら50μsec待ち
    __delay_us(40);

/*****
 * 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
    lcd_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
 *****/
void lcd_init(void) { // ST7032iマニュアル掲載お約束
    __delay_ms(100); // 100mS delay 電源安定まで待つ
    lcd_cmd(0x38); // 8bit 2line Normal mode
    lcd_cmd(0x39); // 8bit 2line IS mode
    lcd_cmd(0x14); // OSC 183Hz BIAS 1/5
    lcd_cmd(0x70 + (CONTRAST & 0x0F)); // コントラスト設定
    lcd_cmd(0x5C + (CONTRAST >> 4));
    // lcd_cmd(0x6A); // Follower for 5.0Vの場合
    // lcd_cmd(0x6B); // Follower for 3.3Vの場合
    __delay_ms(100); // 400mS待ち
    __delay_ms(100);
    __delay_ms(100);
    __delay_ms(100);
    lcd_cmd(0x38); // Set back to Normal mode
    lcd_cmd(0x0C); // Display On
    lcd_cmd(0x01); // Clear Display

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

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

/*****
 * オプションアイコン表示
 *****/
void lcd_icon(unsigned char num, unsigned char onoff){
    char cmd;
    lcd_cmd(0x39); // Extend modeにします
    cmd=0x40 | (ICON[num][0]); // アイコン表示のお約束
    lcd_cmd(cmd); // アイコンアドレス指定
    if(onoff){
        cmd= (ICON[num][1]); // アイコンオンデータ
        lcd_data(cmd);
    }
    else // 消す時は0x00かく
        lcd_data(0x00); // アイコンオフデータ
    lcd_cmd(0x38); // Normal Mode
}

```