

```

                                lcd
/*
 * LCD インターフェイスプログラム (HI-TECH社参考)
 *
 * LCD制御用ICとして世の中標準である日立製HD44780コンパチ
 * 制御LSIインターフェイスに対応します。4ビットモード接続
 * を使います。LCDの接続コネクタは標準14ピンが使われます。
 *
 * RA0-3をLCDデータビット4-7(高位の4ビット)に接続
 * RB7はLCDのRS入力(レジスタ選択)に接続
 * RB6はLCDのEN入力(イネーブル)に接続
 * いつもWで使う
 *
 * このプログラムを使用するためにまずはポート(TRISA, TRISB)を
 * 初期化するlcd_init()を呼びます。その後他のプログラムを呼ぶ
 * ことができるようになります。
 *
 * 2011.09.29 LCD_STRBとLCD_writeタイミングを短めにした
 * PIC12F1822のMSSPマスターで動作確認
 *
 * 2011.10.21 PIC16F1823対応に変更しました
 * 2011.11.06 PIC16F1823対応8MHzタイミングにした
 * 2012.1.10 PIC16F88で4MHzタイミング DATA:RB4-7,RS:RB1,EN:RBO
 * 2012.1.15 PIC16F1827 8MHz
 * 2012.8.25 PIC16F1827 8MHz with i2c/MSSP1
 */
#define _XTAL_FREQ 8000000

#ifndef _XTAL_FREQ
// XTAL_FREQの指定ない場合はシステムクロックは4MHzだと思う
#define _XTAL_FREQ 4000000
#endif

#include <htc.h>
#include "lcd.h"

#define LCD_RS RB7
#define LCD_EN RB6

unsigned char lcd_rs, lcd_data; // RS command:0,RS data:0x80

void LCD_STROBE(void){ // PIC16F1827 対応RA0-RA3:data
    LATA=lcd_data |(PORTA&0xF0); // Lower 4bits:data Upper bits remain
    LCD_RS=lcd_rs; // RS set:RB7
    LCD_EN=1; // EN
    __delay_us(40); // タイミング
    LCD_EN=0; // EN off:RB6
    __delay_us(50); // タイミング
}

/*
 * 1バイトを2回の4ビットモードでLCDに書く関数
 * 事前にlcd_rsを1:データ、0:コマンドに設定必要
 */
void lcd_write(unsigned char c)
{
    lcd_data=(c)&0xF0)>>4; // 上位4ビット
    LCD_STROBE(); // EN線を0n,OFF
    __delay_us(50); // タイミング
    lcd_data = (c & 0x0F); // 下位4ビットを出力
    LCD_STROBE();
    __delay_us(40); // タイミング
}

/*
 * LCDクリアして、カーソルはホームへ
 */
void lcd_clear(void)
{
    lcd_rs = 0; // RSをコマンドモードRS:0に
    lcd_write(0x1); // クリアコマンド1を書く
    __delay_ms(10); // クリア処理には2mSぐらい掛かる
}

```

lcd

```
/*
 * LCDにバイト列を書きこむ
 */
void
lcd_puts(const char * s) {
    lcd_rs = 0x01;          // データ転送モード RS:1
    while(*s)               // バイト列最後にゼロがあるはず
        lcd_write(*s++);   // 1バイト書いては次のアドレス指定
}

void
lcd_putch(char c)
{
    lcd_rs = 0x1;          // 表示データ書き込み設定
    lcd_write(c);         // 1バイト書きます (2ニブル)
}

/*
 * カーソル位置を設定します HD44780ルールに従ったメモリ位置です
 */
void
lcd_goto(unsigned char pos)
{
    lcd_rs = 0;           // コマンド書き込み設定です
    lcd_write(0x80 | pos); // カーソル位置指定はbit7を1にします
    __delay_ms(20);
}

/*
 * RSを変数でもらってWRITE (2ニブル) します
 */
void
lcd_write_rs(unsigned char c, unsigned char rs)
{
    lcd_rs = rs;          // 引数のrs値を引き渡す
    lcd_write(c);        // 1バイト書きます (2ニブル)
}

/*
 * PORTAを初期化、LCDコントローラを初期化するコマンド書き込み
 */
void
lcd_init()
{
    char init_value;

    init_value = 0x3;     // LCDコントローラの初期化コマンド
    __delay_ms(15);      // 電源投入後最低でも15mS待つてから
    lcd_rs=0;            // command mode
    lcd_data = init_value; // 0x03を設定 3回送る
    LCD_STROBE();
    __delay_ms(15);
    LCD_STROBE();
    __delay_ms(200);     // 電源投入後最低でも15mS待つてから
    LCD_STROBE();
    __delay_ms(200);

    lcd_data = 2;        // 4ビットモード指定 : 0x02を設定
    LCD_STROBE();
    __delay_ms(6);

    lcd_write(0x28);     // インターフェイス長さコマンド
    lcd_write(0xF);      // 表示オン、カーソル表示オンでブリンクする
    lcd_clear();         // 画面の消去
    lcd_write(0x6);      // 入力モードに設定
}

```