

```

                                lcd

/*
 * LCD インターフェイスプログラム (HI-TECH社の原本参考)
 * delay.c を使用します。
 * 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)を初期化しておきます。その後このプログラムを
 * 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
 * 2012.8.30 PIC16F1827
 * 2012.09.23 PIC16F1827 Key pad support
 * tuning timing for LCD&KEY PAD
 */
#define XTAL_FREQ 1600000
#define PIC_CLK 1600000
#ifndef XTAL_FREQ
// XTAL_FREQの指定ない場合はシステムクロックは4MHzだと思ふ
#define XTAL_FREQ 4000000
#endif

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

#define LCD_RS LATB7
#define LCD_EN LATB6

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

void LCD_STROBE(void) {
    pre_port=PORTA;
    LATA=(lcd_data&0xF) | (PORTA&0xF0);
    LCD_RS = lcd_rs;
    __delay_us(10);
    LCD_EN=1;
    __delay_us(20); // タイミング

    LCD_EN=0;
    __delay_us(15); // タイミング
    LATA=0xF | (pre_port&0xF0);
    __delay_us(15); // 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 >> 4 ) & 0x0F ); // 4ビットシフトして上位4ビット
    LCD_STROBE(); // EN線をOn, OFF
    __delay_us(20); // タイミング
    lcd_data = ( c & 0x0F ); // 下位4ビットを出力
    LCD_STROBE();
    __delay_us(20); // 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;
        lcd_write(0x1);
        __delay_ms(10);
    }

/*
 * LCDにバイト列を書きこむ
 */
void
lcd_puts(const char * s)
{
    lcd_rs = 0x01;
    while(*s)
        lcd_write(*s++);
}

/*
 * 1文字分のコード（ニブルを2回）書き込みます
 */
void
lcd_putchar(char c)
{
    lcd_rs = 0x1;
    lcd_write(c);
}

/*
 * 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;
    lcd_write(0x80+pos);
    __delay_ms(1);
}

/*
 * 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;
    lcd_write(c);
}

/*
 * 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_RS = 0;
    LCD_EN = 0;
    __delay_ms(20);
    lcd_data = init_value;
    LCD_STROBE();
    __delay_ms(20);
    LCD_STROBE();
    __delay_us(200);
    LCD_STROBE();
    __delay_us(200);
    lcd_data = 2;
    LCD_STROBE();
    __delay_us(100);
    lcd_write(0x28);
    __delay_us(100);
}

```

```

// lcd
// RSをコマンドモードRS:0に
// クリアコマンド1を書く
// クリア処理には2mSぐらい掛かる

```

```

// データ転送モード RS:1
// バイト列の最後にゼロ
// 1バイト書いては次のアドレス指定

```

```

// 表示データ書き込み設定
// 1バイト書きます（2ニブル）

```

```

// コマンド書き込み設定です
// カーソル位置指定は#7をにします
// タイミング

```

```

// 引数のrs値を引き渡す
// 1バイト書きます（2ニブル）

```

```

// LCDコントローラの初期化コマンド

```

```

// RS信号（PB0）はコマンドモード：0
// エネーブルビットを0

```

```

// 電源投入後最低でも15mS待つてから
// 0x03を設定

```

```

// 4ビットモード指定：0x02を設定

```

```

// タイミング
// インターフェイス長さコマンド
// タイミング

```

```
    lcd_write(0xF);  
    __delay_us(100);  
    lcd_clear();  
    __delay_ms(20);  
    lcd_write(0x6);  
    __delay_us(100);  
}  
  
    lcd  
    // 表示オン、カーソル表示オンでブリンクする  
    // タイミング  
    // 画面の消去  
  
    // 入力モードに設定  
    // タイミング
```