

main\_slave

```
/* ////////////////////////////////////////
 * PIC16F1823にSC1602互換LCD表示パネルを接続してi2cのスレー
 * ブとして表示を行うルーチンです。sc1602とは4ビット
 * モード接続されます。
 *
 * PIC16F88 V0.1 2011.09.30 V1.0 2011.10.03 をベースに改造
 * PIC16F1823 V2.0 2011.10.21
 * V2.1 2011.11.06 Timing for 2.6V operation
 * MPLAB v8.73&HI-TECH C V9.82
 *
 *
 * by nobcha (c)2011
 *
 * i2c経由データは必ず2バイト単位でくるとする。MAX6(2,4,6)
 * 1バイト目はRSビット、2バイト目がLCDに書き込むデータです
 * ストロベリナーナックスの拡張コマンドはサポートしてません
 *
 * RC2-5:SC1602は4ビットモードとし、RC2-5で接続します。
 * RA0:LCD EN bit (enable)はRA0につながります。
 * RA1:LCD RS bit (RS)はRA1につながります。
 * RC1:SDA MSSP
 * RC0:SCL MSSP
 * RA5:動作確認用LED
 * RA2:1:normal 0:Debug (Display hexadecimal received code)
 * RA4/CLK0:Charge pump for minus LCD bias
 *
 * SC1602 pin connection via 4bit mode
 * #1 Vdd=5V
 * #2 Vss=GND
 * #3 LCD contrast center of 2k VOL
 * #4 RS RA1
 * #5 R/W GND
 * #6 E RA0
 *
 * #11-14 DATA RC2-5
 *
 * //////////////////////////////////////// */
#include <htc.h>
#include "lcd.h"
#include "delay.h"

#define _XTAL_FREQ 8000000
#define PIC_CLOCK 8000000
#define I2C_ADR 0x7C
#define DEBUG RA2
#define MON_LED LATA5
```

## main\_slave

```
__CONFIG(  
    FOSC_INTOSC & WDTE_OFF & PWRTE_ON & MCLRE_ON & CP_OFF  
    & CPD_OFF & BOREN_OFF & CLKOUTEN_ON & IESO_OFF & FCMEN_OFF  
);  
  
__CONFIG(  
    WRT_OFF & PLLEN_OFF & STVREN_ON & LVP_OFF  
);  
/* ////////////////////////////////////// */  
  
unsigned char    buffer[20];           /* 受信バッファ */  
unsigned char    stat[20];  
char            i , j, rcv_count=0 , cnt=0;  
  
void            init(void) {  
    PORTA = 0b00001001;                //  
    PORTC = 0b00000000;                //  
    TRISA = 0b00001100;                /* RA2, 3は入力  
    */  
    TRISC = 0b00000011;                /* RC0とRC1はi2c用で  
    */  
    OSCCON = 0b01110000;                /* 内部クロック8MHz  
    */  
    ANSELA = 0x00;                       /* 全ポートデジタル  
    */  
    ANSELB = 0;                          /* オプション設定な  
    */  
    OPTION_REG = 0x00;                    /* オプション設定な  
    */  
    // port direction:  
    1: input  
    __delay_ms(100);  
}  
  
void            ssp_init(void) {  
    SSP1CON1 = 0x36;                    //  
    SSP1EN, CKP, SSP_slave_7bit  
    SSP1ADD = I2C_ADR;                  // SSP ADDRESS SET  
    PEIE = 1;                            // PEIE enable  
    SSP1IE = 1;  
}  
  
unsigned char    lcdrs;  
interrupt i2c_slave() {                 // i2c slave  
    receive int func  
    GIE=0;
```

```

                                main_slave
    if(SSP1IF==1) {
        SSP1IE=0;
        SSP1IF=0;
        buffer[rcv_count]=SSP1BUF;          /* dataをバ
        ッファに入れる */
        stat[rcv_count]=SSP1STAT;          /* statusを
        ステータスバッファに入れる */

        if ((SSP1STAT&0x20)==0x20) {          // DATA?
            if ((rcv_count==1) | (rcv_count==3)
| (rcv_count==5)) {
                if ((buffer[rcv_count]&0x40)==0x40) {lcdrs=1;}
                else {lcdrs=0;}
            }
            else {
                lcd_write_rs(buffer[rcv_count],
lcdrs);
            }
        }
        rcv_count++;                          /* 受信バイ
        ト数 */
        if(rcv_count>7) { lcd_putch(0x3F); }
        SSP1IE=1;
    }
    GIE=1;
}

```

```

char    MSG1[] = "PIC i2c slave v2.0 ";

```

```

void main(void) {
    init();                                  /*
    ポート初期化 */
    MON_LED=1;
    __delay_ms(100);
    __delay_ms(100);
    lcd_init();                              /*
    LCDの初期化 */
    __delay_ms(100);
    cnt=0;
    MON_LED=0;

    if ((DEBUG)==1) {                          /*
    debug mode */
        lcd_goto(40);                          /* 2
    行目にLCD動作表示 */
}

```



```
表示時間を待たせる          main_slave
                              */
                              _delay_ms(100);
                              }
                              }
rcv_count=0;
MON_LED = MON_LED ^= 1;
}
```