```
/*      By nobcha   all right reserved  08/14/2010
 *
 *      LCD interface via 4040B         Uses routines from delay.c
 *      This code will interface to a standard LCD controller
 *      like the Hitachi HD44780. It uses it in 4 bit mode, with
 *      the hardware connected as follows (the standard 14 pin
 *      LCD connector is used):
 *
 *      RB0 is connected to the TC4040B CLK and RST(via RC)
 *      RB1 is connected to the LCD EN bit (enable)
 *      TC4040B Q1-Q4 is connected to the LCD data bits 0-3
 *      TC4040B Q5 is connected to the LCD RS input (register select)
 *
 *      To use these routines, set up the port I/O (TRISA, TRISB) then
 *      call lcd_init(), then other routines as required.
 *
 *
 *      Development circumstance MPLAB V4.53 HITECH C V9.71a
 *
 */
#define _XTAL_FREQ 12800000
#ifndef _XTAL_FREQ
 // Unless specified elsewhere, 4MHz system frequency is assumed
 #define _XTAL_FREQ 4000000
#endif


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

#define TC4040B_Q1 1
#define TC4040B_Q2 2
#define TC4040B_Q3 4
#define TC4040B_Q4 8
#define TC4040B_Q5 16
#define TC4040B_Q6 32
#define TC4040B_Q7 64
#define TC4040B_Q8 128
#define TC4040B_Q9 256

#define LCD_EN RB1
#define TC4040B_CLK RB0

#define LCD_STROBE()    ((LCD_EN = 1),(LCD_EN=0) )
#define TC4040B_RESET   ( TC4040B_CLK=1, __delay_us(150), TC4040B_CLK=0, __delay_us(100) )
#define LCD_PULSE       ( TC4040B_CLK=1, TC4040B_CLK =0, __delay_us(2) )



/* nibble write */
void lcd_nib(unsigned char count, char rs_value){
        TC4040B_RESET;                          // TC4040B reset
        if(rs_value){ count = count + TC4040B_Q5; }             // If rs=1 set Q5 count

        while(count>0){
                LCD_PULSE;                      // Put data to TC4040B
                count--;
        }
        LCD_STROBE();                           // Put TC4040B output to SL1602
        __delay_ms(2);
}

/* write a byte to the LCD in 4 bit mode */

void lcd_write(unsigned char char_data, char rs_value){
        char data  = ( ( char_data >> 4)  & 0b00001111 ) ;      // High side nibble
        lcd_nib(data, rs_value);
        char_data = ( char_data  & 0b00001111 ) ;               // low side nibble
        lcd_nib(char_data, rs_value);
}


/*      Clear and home the LCD      */

void  lcd_clear(void) {
```

```
        char rs_value = 0 ;
        lcd_write(0x1 , rs_value );
        __delay_ms(10);
}

/*     write a string of characters to the LCD   */

void lcd_puts(const char *s) {
        char rs_value = 1 ;                             // write characters
        while(*s  )
                lcd_write(*s++ , rs_value );
}

/*      write one character to the LCD     */

void lcd_putch(char c) {
        char rs_value = 1 ;                             // write characters
        lcd_write( c , rs_value );
}


/*      Go to the specified position       */

void lcd_goto(unsigned char pos){       //cursor position set
        char rs_value =0 ;
        lcd_write(0x80+pos , rs_value );
}

/*      initialise the LCD - put into 4 bit mode */
void lcd_init() {
        char init_value;
        TC4040B_RESET ;                                 // TC4040B reset
        init_value = 0x3;                               // Load initial value
        char rs_value  = 0 ;
        LCD_EN = 0;

        __delay_ms(45);                                 // wait 15mSec after power applied,
        lcd_nib(init_value,rs_value);
        __delay_ms(15);
        lcd_nib(init_value,rs_value);
        __delay_us(600);
        lcd_nib(init_value,rs_value);
        __delay_us(600);
        lcd_nib( 2, rs_value );                 // Four bit mode

        lcd_write(0x28 , rs_value);     // Set interface length
        lcd_write(0xF , rs_value );     // Display On, Cursor On, Cursor Blink
        lcd_clear();                                    // Clear screen
        lcd_write(0x6 , rs_value );     // Set entry Mode
}
```