```
//LCD functions for mega128 style LCD interface
//Roger Traylor 4.25.07

#include <avr/io.h>
#define F_CPU 16000000UL //16Mhz clock
#include <util/delay.h>
#include <string.h>

#include "lcd.h"          //function prototypes defined here

#define NUM_LCD_CHARS 16 //number of chars that the LCD has in a line

void strobe_lcd(void){
        //twiddles bit 2, PORTB creating the enable signal for the LCD
        PORTF |= 0x08;
        PORTF &= 0xF7;
}

void clear_display(void){
        while (!(SPSR & 0x80)) {}       // Wait for any pending SPI transfers to complete
        SPDR = 0x00;    //command, not data
        while (!(SPSR & 0x80)) {}       // Wait for SPI transfer to complete
        SPDR = 0x01;    //clear display command
        while (!(SPSR & 0x80)) {}       // Wait for SPI transfer to complete
        strobe_lcd();   //strobe the LCD enable pin
        _delay_ms(2);   //obligatory waiting for slow LCD (1.64mS)
}

void cursor_home(void){
        while (!(SPSR & 0x80)) {}       // Wait for any pending SPI transfers to complete
        SPDR = 0x00;    //command, not data
        while (!(SPSR & 0x80)) {}       // Wait for SPI transfer to complete
        SPDR = 0x02;   // cursor go home position
        while (!(SPSR & 0x80)) {}       // Wait for SPI transfer to complete
        strobe_lcd();
        _delay_ms(2);   //obligatory waiting for slow LCD (1.64mS)
}

void home_line2(void){
        while (!(SPSR & 0x80)) {}       // Wait for any pending SPI transfers to complete
        SPDR = 0x00;    //command, not data
        while (!(SPSR & 0x80)) {}       // Wait for SPI transfer to complete
        SPDR = 0xC0;   // cursor go home on line 2
        while (!(SPSR & 0x80)) {}       // Wait for SPI transfer to complete
        strobe_lcd();
        _delay_ms(2);   //obligatory waiting for slow LCD (1.64mS)
}

void fill_spaces(void){
        uint8_t count;
        while (!(SPSR & 0x80)) {}       // Wait for any pending SPI transfers to complete
        for (count=0; count<=NUM_LCD_CHARS-1; count++){
          SPDR = 0x01; //set SR for data
          while (!(SPSR & 0x80)) {}     // Wait for SPI transfer to complete
          SPDR = 0x20;
          while (!(SPSR & 0x80)) {}     // Wait for SPI transfer to complete
          strobe_lcd();
          _delay_us(100);   //obligatory waiting for slow LCD (40uS)
        }
}

void char2lcd(char a_char){
        //sends a char to the LCD
        //usage: char2lcd('H');  // send an H to the LCD
        while (!(SPSR & 0x80)) {}       // Wait for any pending SPI transfers to complete
        SPDR = 0x01;   //set SR for data xfer with LSB=1
        while (!(SPSR & 0x80)) {}       // Wait for SPI transfer to complete
        SPDR = a_char; //send the char to the SPI port
        while (!(SPSR & 0x80)) {}       // Wait for SPI transfer to complete
        strobe_lcd();  //toggle the enable bit
        _delay_us(100);   //obligatory waiting for slow LCD (40uS)
}


void string2lcd(char *lcd_str){

        //sends a string to LCD
```

```
        uint8_t count;
        while (!(SPSR & 0x80)) {}       // Wait for any pending SPI transfers to complete
        for (count=0; count<=(strlen(lcd_str)-1); count++){
          SPDR = 0x01; //set SR for data
          while (!(SPSR & 0x80)) {}     // Wait for SPI transfer to complete
          SPDR = lcd_str[count];
          while (!(SPSR & 0x80)) {}     // Wait for SPI transfer to complete
          strobe_lcd();
          _delay_us(100);   //obligatory waiting for slow LCD (40uS)
        }
}

void lcd_init(void){
        uint8_t i;
        //initalize the LCD to receive data
        _delay_ms(15);
        for(i=0; i<=2; i++){ //do funky initalize sequence 3 times
          SPDR = 0x00;
          while (!(SPSR & 0x80)) {}     // Wait for SPI transfer to complete
          SPDR = 0x30;
          while (!(SPSR & 0x80)) {}     // Wait for SPI transfer to complete
          strobe_lcd();
          _delay_ms(7);
        }

        SPDR = 0x00;
        while (!(SPSR & 0x80)) {}       // Wait for SPI transfer to complete
        SPDR = 0x38;
        while (!(SPSR & 0x80)) {}       // Wait for SPI transfer to complete
        strobe_lcd();
        _delay_ms(5);

        SPDR = 0x00;
        while (!(SPSR & 0x80)) {}       // Wait for SPI transfer to complete
        SPDR = 0x08;
        while (!(SPSR & 0x80)) {}       // Wait for SPI transfer to complete
        strobe_lcd();
        _delay_ms(5);

        SPDR = 0x00;
        while (!(SPSR & 0x80)) {}       // Wait for SPI transfer to complete
        SPDR = 0x01;
        while (!(SPSR & 0x80)) {}       // Wait for SPI transfer to complete
        strobe_lcd();
        _delay_ms(5);

        SPDR = 0x00;
        while (!(SPSR & 0x80)) {}       // Wait for SPI transfer to complete
        SPDR = 0x06;
        while (!(SPSR & 0x80)) {}       // Wait for SPI transfer to complete
        strobe_lcd();
        _delay_ms(5);

        SPDR = 0x00;
        while (!(SPSR & 0x80)) {}       // Wait for SPI transfer to complete
        SPDR = 0x0C;                    // cursor off
        while (!(SPSR & 0x80)) {}       // Wait for SPI transfer to complete
        strobe_lcd();
        _delay_ms(5);
}
```