

A2PS - P ke3112 lcd\_test.c  
Default Settings

```

// lcd_test.c
// Roger Traylor
#include <avr/io.h>
#include <avr/delay.h>
#include <string.h>
#include <stdlib.h>

#define delay_count 4000 //up clock in MHz divided by 4000

void cursor_home(void);
void home_line2(void);
void fill_spaces(void);
void string2lcd(char *lcd_str);

int i, j, low, high;
char lcd_str[16]; //holds string to send to lcd
char lcd_test[17] = {"****-LCD_Test-****"};

//delays by 1ms increments
void delay_ms1(uint8_t ms)
{
    uint16_t cnt;
    asm volatile (
        "\n"
        "L_d11%=: " "\n\t"
        "mov %A0,%A2" "\n\t"
        "mov %B0,%B2" "\n\t"
        "L_d12%=: " "\n\t"
        "sbw %A0,1" "\n\t"
        "bme L_d12%=: " "\n\t"
        "dec %i1" "\n\t"
        "bme L_d11%=: " "\n\t"
        : "=&w" (cnt)
        : "r" (ms), "r" (delay_count)
    );
}

void strobe_lcd(void) {
//widdles bit 3, PORTF creating the enable signal for the LCD
PORTF = 0x08;
PORTF = 0x00;
}

void clear_display(void) {
SPDR = 0x00; //command, not data
_delay_loop_1(9); //wait 2us for data to leave SPI port
SPDR = 0x01; //clear display command
_delay_loop_1(9); //wait 2us for data to leave SPI port
strobe_lcd(); //strobe the LCD enable pin
delay_ms1(2); //obligatory waiting for slow LCD
}

void cursor_home(void) {
SPDR = 0x00; //command, not data
_delay_loop_1(9); //wait 2us for data to leave SPI port
SPDR = 0x02; //cursor go home position
_delay_loop_1(9); //wait 2us for data to leave SPI port
strobe_lcd();
delay_ms1(2);
}

void home_line2(void) {
SPDR = 0x00; //command, not data
_delay_loop_1(9); //wait 2us for data to leave SPI port
SPDR = 0x00; //cursor go home on line 2
_delay_loop_1(9); //wait 2us for data to leave SPI port
strobe_lcd();
delay_ms1(2);
}

```

```

}
void fill_spaces(void) {
int count;
for (count=0; count<=15; count++) {
    SPDR = 0x01; //set SR for data
    _delay_loop_1(9); //wait 2us for data to leave SPI port
    SPDR = 0x20;
    _delay_loop_1(9); //wait 2us for data to leave SPI port
    strobe_lcd();
    delay_ms1(1);
}
}

void char2lcd(char a_char) {
//sends a char to the LCD
//usage: char2lcd('H'); // send an H to the LCD
SPDR = 0x01; //set SR for data xfer with LSB=1
_delay_loop_1(9); //wait 2us for data to leave SPI port
SPDR = a_char; //send the char to the SPI port
_delay_loop_1(9); //wait 2us for data to leave SPI port
strobe_lcd(); //toggle the enable bit
delay_ms1(1); //wait the prescribed time for the LCD to process
}

void string2lcd(char *lcd_str) {
//sends a string to LCD
int count;
for (count=0; count<=(strlen(lcd_str)-1); count++) {
    SPDR = 0x01; //set SR for data
    _delay_loop_1(9); //wait 2us for data to leave SPI port
    SPDR = lcd_str[count];
    _delay_loop_1(9); //wait 2us for data to leave SPI port
    strobe_lcd();
    delay_ms1(1);
}
}

/* Run this code before attempting to write to the LCD.*/
void spi_init(void) {
DDRF=0x08; //port F bit 3 is enable for LCD
PORTB=0x00; //port B initialization for SPI
DDRB=0x07; //Turn on SS, MSTR, SCK
//Master mode, Clock=clk/2, Cycle half phase, Low polarity, MSB first
SPCR=0x50;
SPSR=0x01;
}

void lcd_init(void) {
int i;
//initialize the LCD to receive data
delay_ms1(15);
for (i=0; i<=2; i++) { //do funky initialize sequence 3 times
    SPDR = 0x00;
    _delay_loop_1(9); //wait 2us for data to leave SPI port
    SPDR = 0x30;
    _delay_loop_1(9); //wait 2us for data to leave SPI port
    strobe_lcd();
    delay_ms1(7);
}
}

SPDR = 0x00;
_delay_loop_1(9); //wait 2us for data to leave SPI port
SPDR = 0x38;
_delay_loop_1(9); //wait 2us for data to leave SPI port
strobe_lcd();
delay_ms1(5);
}

```

```

Oct 06, 06 10:04          lcd_test.c          Page 1/4
// lcd_test.c
//Roger Traylor

#include <avr/io.h>
#include <avr/delay.h>
#include <string.h>
#include <stdlib.h>

#define delay_count 4000 //uP clock in MHz divided by 4000

void cursor_home(void);
void home_line2(void);
void fill_spaces(void);
void string2lcd(char *lcd_str);

int i,j, low, high;
char lcd_str[16]; //holds string to send to lcd
char lcd_test[17] = (****-LCD_Test-****);

//delays by lms increments
void delay_ms1(uint8_t ms)
{
    uint16_t cnt;
    asm volatile (
        "\n"
        "L_d11%=: *%nl"
        "mov %A0, %A2" *%nl"
        "mov %B0, %B2" *%nl"
        "L_d12%=: *%nl"
        "sbw %A0, 1" *%nl"
        "brne _L_d12%=" *%nl"
        "dec %I" *%nl"
        "brne _L_d11%=" *%nl"
        : "=&w" (cnt)
        : "r" (ms), "r" (delay_count)
    );
}

void strobe_lcd(void){
    //toggles bit 3, PORTF creating the enable signal for the LCD
    PORTF = 0x08;
    PORTF = 0x00;
}

void clear_display(void){
    SPDR = 0x00; //command, not data
    _delay_loop_1(9); //wait 2us for data to leave SPI port
    SPDR = 0x01; //clear display command
    _delay_loop_1(9); //wait 2us for data to leave SPI port
    strobe_lcd(); //strobe the LCD enable pin
    delay_ms1(2); //obligatory waiting for slow LCD
}

void cursor_home(void){
    SPDR = 0x00; //command, not data
    _delay_loop_1(9); //wait 2us for data to leave SPI port
    SPDR = 0x02; // cursor go home position
    _delay_loop_1(9); //wait 2us for data to leave SPI port
    strobe_lcd();
}

```

4 UP  
PRINTING

a2ps -4 -P kec3112  
lcd\_test.c

```

Oct 06, 06 10:04          lcd_test.c          Page 2/4
delay_ms1(2);
}

void home_line2(void){
    SPDR = 0x00; //command, not data
    _delay_loop_1(9); //wait 2us for data to leave SPI port
    SPDR = 0xC0; // cursor go home on line 2
    _delay_loop_1(9); //wait 2us for data to leave SPI port
    strobe_lcd();
    delay_ms1(2);
}

void fill_spaces(void){
    int count;
    for (count=0; count<=15; count++){
        SPDR = 0x01; //set SR for data
        _delay_loop_1(9); //wait 2us for data to leave SPI port
        SPDR = 0x20;
        _delay_loop_1(9); //wait 2us for data to leave SPI port
        strobe_lcd();
        delay_ms1(1);
    }
}

void char2lcd(char a_char){
    //sends a char to the LCD
    //usage: char2lcd('H'); // send an H to the LCD
    SPDR = 0x01; //set SR for data xfer with LSB=1
    _delay_loop_1(9); //wait 2us for data to leave SPI port
    SPDR = a_char; //send the char to the SPI port
    _delay_loop_1(9); //wait 2us for data to leave SPI port
    strobe_lcd(); //toggle the enable bit
    delay_ms1(1); //wait the prescribed time for the LCD to process
}

void string2lcd(char *lcd_str){
    //sends a string to LCD
    int count;
    for (count=0; count<=(strlen(lcd_str)-1); count++){
        SPDR = 0x01; //set SR for data
        _delay_loop_1(9); //wait 2us for data to leave SPI port
        SPDR = lcd_str[count];
        _delay_loop_1(9); //wait 2us for data to leave SPI port
        strobe_lcd();
        delay_ms1(1);
    }
}

/* Run this code before attempting to write to the LCD.*/
void spi_init(void){
    DDRF=0x08; //port F bit 3 is enable for LCD
    PORTB=0x00; //port B initialization for SPI
    DDRB=0x07; //Turn on SS, MOSI, SCLK
    //Master mode, Clock=clk/2, Cycle half phase, Low polarity, MSB first
    SPDR=0x50;
    SPSR=0x01;
}

```

```

Oct 06, 06 10:04          lcd_test.c          Page 3/4
void lcd_init(void){
    int i;
    //initialize the LCD to receive data
    delay_ms1(15);
    for(i=0; i<=2; i++){ //do funky initialize sequence 3 times
        SPDR = 0x00;
        _delay_loop_1(9); //wait 2us for data to leave SPI port
        SPDR = 0x30;
        _delay_loop_1(9); //wait 2us for data to leave SPI port
        strobe_lcd();
        delay_ms1(7);
    }

    SPDR = 0x00;
    _delay_loop_1(9); //wait 2us for data to leave SPI port
    SPDR = 0x38;
    _delay_loop_1(9); //wait 2us for data to leave SPI port
    strobe_lcd();
    delay_ms1(5);

    SPDR = 0x00;
    _delay_loop_1(9); //wait 2us for data to leave SPI port
    SPDR = 0x08;
    _delay_loop_1(9); //wait 2us for data to leave SPI port
    strobe_lcd();
    delay_ms1(5);

    SPDR = 0x00;
    _delay_loop_1(9); //wait 2us for data to leave SPI port
    SPDR = 0x01;
    _delay_loop_1(9); //wait 2us for data to leave SPI port
    strobe_lcd();
    delay_ms1(5);

    SPDR = 0x00;
    _delay_loop_1(9); //wait 2us for data to leave SPI port
    SPDR = 0x06;
    _delay_loop_1(9); //wait 2us for data to leave SPI port
    strobe_lcd();
    delay_ms1(5);

    SPDR = 0x00;
    _delay_loop_1(9); //wait 2us for data to leave SPI port
    SPDR = 0x0E;
    _delay_loop_1(9); //wait 2us for data to leave SPI port
    strobe_lcd();
    delay_ms1(5);
}

//***** main() follows *****/
int main (void) {
    //initialize the SPI port and then the LCD
    spi_init();
    lcd_init();
    clear_display();

    //***** LCD test *****/
    /*
    The string ****-LCD_Test-**** is sent to both LCD lines and held for 1sec.

```

```

Oct 06, 06 10:04          lcd_test.c          Page 4/4
Then LCD is then cleared for 0.6 sec and the pattern is repeated.
*/
while (1){
    //do conversions forever
    string2lcd(lcd_test); //send string to LCD
    home_line2(); //move cursor to line2
    string2lcd(lcd_test); //send string again
    delay_ms1(200); //erase after 0.6 sec
    delay_ms1(200); //erase after 0.6 sec
    delay_ms1(200); //erase after 0.6 sec
    clear_display();
    delay_ms1(200); //write after 0.6 sec
    delay_ms1(200); //write after 0.6 sec
    delay_ms1(200); //write after 0.6 sec
    cursor_home(); //put cursor back to home position
} //while
} //main

```

Oct 06, 06 10:04

lcd\_test.c

Page 1/4

```

// lcd test.c
//Roger Traylor

#include <avr/io.h>
#include <avr/delay.h>
#include <string.h>
#include <stdlib.h>

#define delay_count 4000 //uP clock in MHz divided by 4000

void cursor_home(void);
void home_line2(void);
void fill_spaces(void);
void string2lcd(char *lcd_str);

int i,j, low, high;
char lcd_str[16]; //holds string to send to lcd
char lcd_test[17] = {"***-LCD_Test-***"};

//delays by 1ms increments
void delay_ms1(uint8_t ms)
{
    uint16_t cnt;
    asm volatile (
        "\n"
        "L_dl1%=: "          "\n\t"
        "mov %A0, %A2"      "\n\t"
        "mov %B0, %B2"      "\n"
        "L_dl2%=: "          "\n\t"
        "sbiw %A0, 1"        "\n\t"
        "brne L_dl2%=: "     "\n\t"
        "dec %1"            "\n\t"
        "brne L_dl1%=: "     "\n\t"
        : "=&w" (cnt)
        : "r" (ms), "r" (delay_count)
    );
}

void strobe_lcd(void){
//twiddles bit 3, PORTF creating the enable signal for the LCD
    PORTF = 0x08;
    PORTF = 0x00;
}

void clear_display(void){
    SPDR = 0x00; //command, not data
    _delay_loop_1(9); //wait 2us for data to leave SPI port
    SPDR = 0x01; //clear display command
    _delay_loop_1(9); //wait 2us for data to leave SPI port
    strobe_lcd(); //strobe the LCD enable pin
    delay_ms1(2); //obligatory waiting for slow LCD
}

void cursor_home(void){
    SPDR = 0x00; //command, not data
    _delay_loop_1(9); //wait 2us for data to leave SPI port
    SPDR = 0x02; // cursor go home position
    _delay_loop_1(9); //wait 2us for data to leave SPI port
    strobe_lcd();
}

```

a2ps -1 -P kec3112 lcd-test.c  
↑ "one up"

```
// lcd_test.c
//Roger Traylor
#include <avr/io.h>
#include <avr/delay.h>
#include <string.h>
#include <stdlib.h>

#define delay_count 4000 //uP clock in MHz divided by 4000
```

a2ps -l --font-size=9 -r -P kaes112 lcd\_test.c

landscape mode  
(R = portrait mode (default))

```
void cursor_home(void);
void home_line2(void);
void fill_spaces(void);
void string2lcd(char *lcd_str);

int i,j, low, high;
char lcd_str[16]; //holds string to send to lcd
char lcd_test[17] = {"***-LCD_Test-***"};
```

```
//delays by 1ms increments
void delay_ms1(uint8_t ms)
{
  uint16_t cnt;
  asm volatile (
```

```
    "\n"
    "L_d11%=":
    "mov %A0, %A2"      "\n\t"
    "mov %B0, %B2"      "\n\t"
    "L_d12%=":
    "sbw %A0, 1"        "\n\t"
    "brne L_d12%="      "\n\t"
    "dec %1"            "\n\t"
    "brne L_d11%="      "\n\t"
    : "=&w" (cnt)
    : "r" (ms), "r" (delay_count)
    );
}
```

```
void strobe_lcd(void) {
//twiddles bit 3, PORTF creating the enable signal for the LCD
PORTF = 0x08;
PORTF = 0x00;
}
```

```
void clear_display(void) {
SPDR = 0x00; //command, not data
delay_loop_1(9); //wait 2us for data to leave SPI port
SPDR = 0x01; //clear display command
delay_loop_1(9); //wait 2us for data to leave SPI port
strobe_lcd(); //strobe the LCD enable pin
delay_ms1(2); //obligatory waiting for slow LCD
}
```

```
void cursor_home(void) {
SPDR = 0x00; //command, not data
delay_loop_1(9); //wait 2us for data to leave SPI port
SPDR = 0x02; // cursor go home position
}
```

Oct 06, 06 10:04

lcd\_test.c

Page 1/3

```

// lcd test.c
//Roger Traylor

#include <avr/io.h>
#include <avr/delay.h>
#include <string.h>
#include <stdlib.h>

#define delay_count 4000 //uP clock in MHz divided by 4000

void cursor_home(void);
void home_line2(void);
void fill_spaces(void);
void string2lcd(char *lcd_str);

int i,j, low, high;
char lcd_str[16]; //holds string to send to lcd
char lcd_test[17] = {"***-LCD_Test-***"};

//delays by lms increments
void delay_ms1(uint8_t ms)
{
    uint16_t cnt;
    asm volatile (
        "\n"
        "L_d11%=: "      "\n\t"
        "mov %A0, %A2"   "\n\t"
        "mov %B0, %B2"   "\n\t"
        "L_d12%=: "      "\n\t"
        "sbiw %A0, 1"    "\n\t"
        "brne L_d12%=: " "\n\t"
        "dec %1"         "\n\t"
        "brne L_d11%=: " "\n\t"
        : "=&w" (cnt)
        : "r" (ms), "r" (delay_count)
    );
}

void strobe_lcd(void){
//twiddles bit 3, PORTF creating the enable signal for the LCD
    PORTF = 0x08;
    PORTF = 0x00;
}

void clear_display(void){
    SPDR = 0x00; //command, not data
    _delay_loop_1(9); //wait 2us for data to leave SPI port
    SPDR = 0x01; //clear display command
    _delay_loop_1(9); //wait 2us for data to leave SPI port
    strobe_lcd(); //strobe the LCD enable pin
    delay_ms1(2); //obligatory waiting for slow LCD
}

void cursor_home(void){
    SPDR = 0x00; //command, not data
    _delay_loop_1(9); //wait 2us for data to leave SPI port
    SPDR = 0x02; // cursor go home position
    _delay_loop_1(9); //wait 2us for data to leave SPI port
    strobe_lcd();
    delay_ms1(2);
}

void home_line2(void){
    SPDR = 0x00; //command, not data
    _delay_loop_1(9); //wait 2us for data to leave SPI port
    SPDR = 0xC0; // cursor go home on line 2
    _delay_loop_1(9); //wait 2us for data to leave SPI port
    strobe_lcd();
    delay_ms1(2);
}

void fill_spaces(void){
    int count;
    for (count=0; count<=15; count++){
        SPDR = 0x01; //set SR for data
        _delay_loop_1(9); //wait 2us for data to leave SPI port
        SPDR = 0x20;
    }
}

```

a2ps -1 --font-size=9 -P kec3112 lcd-test.c

↑ font size option