Introduction:

Learning how to program an LCD Display.

Equipment:

* Simon Board
* Atmel Studio 7
* 2x16 Character LCD Module
* Connection Wires

Procedure:

* Open Atmel Studio 7 and create a new assembly project.
* Properly Wire the LCD Module to Simon board.
* Define the ports and control pins used by the LCD module.
* Start the LCD initialization process by setting PortD and PortE to output mode and clearing the data pins.
* Set the function set to initialize the LCD in 8-bit, 2 line, and 5x7 pixel mode.
* Set the display turn on and cursor on bit to 1 and clear the display bits.
* Write the individual characters of your first name and then move the cursor to the second line.
* Write the individual characters of your last name and clear the display.
* Return the cursor home to begin the loop again.

Results:

The LCD Display spelled out my name on both lines and clear it after it was displayed.

Questions:

1. The LCD has three important control pins which are PORTE2, PORTE3, and PORTE4. The pins control read/write, enable, and command/data register. The LCD is initialized to 8-bit mode, 2 line, and 5x7 settings.
2. The LCD is level-triggered because the LCD displays the characters by levels.
3. I had my name displayed only on the first line. I fixed the issue by setting my last name on the second loop.

Conclusion:

I learned how to program an LCD Display to print out my name and to clear the LCD after the line is displayed.

Appendices:

Appendix 1:

LCD Display

A circuit board

Description automatically generated

Appendix 2:

Code:

/\*

\* lab4.c

\*

\* Created: 3/4/2020 10:45:14 AM

\* Author : argk4

\*/

#include <avr/io.h>

#define *F\_CPU* 16000000UL

#include "util/delay.h"

#define LCD\_DATA PORTD

#define LCD\_DATA\_DDR DDRD

#define LCD\_CNTRL PORTE

#define LCD\_CNTRL\_DDR DDRE

#define RS PORTE2

#define RW PORTE3

#define EN PORTE4

void lcd\_clear();

void lcd\_write\_data(unsigned char data);

void lcd\_write\_cmd(unsigned char cmd);

void lcd\_ininit();

void lcd\_init();

void lcd\_return\_home();

void move\_cursor(unsigned char addr);

int main(void)

{

lcd\_ininit();

lcd\_init();

while (1)

{

lcd\_write\_data('A');

lcd\_write\_data('n');

lcd\_write\_data('t');

lcd\_write\_data('h');

lcd\_write\_data('o');

lcd\_write\_data('n');

lcd\_write\_data('y');

move\_cursor(0x40);

lcd\_write\_data('R');

lcd\_write\_data('o');

lcd\_write\_data('b');

lcd\_write\_data('l');

lcd\_write\_data('e');

lcd\_write\_data('s');

lcd\_clear();

lcd\_return\_home();

}

}

void lcd\_clear()

{

lcd\_write\_cmd(0x01);

}

void lcd\_return\_home()

{

lcd\_write\_cmd(0x02);

}

void move\_cursor(unsigned char addr)

{

addr |= (1<<7);

lcd\_write\_cmd(addr);

}

void lcd\_write\_data(unsigned char data)

{

LCD\_CNTRL |= (1<<RS); // RS = 1, data register

LCD\_CNTRL &= ~(1<<RW); // RW = 0, write mode

LCD\_DATA = data;

*\_delay\_ms*(100);

LCD\_CNTRL |= (1<<EN); // EN = 1, enable read or write

*\_delay\_ms*(100);

LCD\_CNTRL &= ~(1<<EN); // EN = 0, disable read or write

*\_delay\_ms*(100);

}

void lcd\_write\_cmd(unsigned char cmd)

{

LCD\_CNTRL &= ~(1<<RS); // RS = 0, command register

LCD\_CNTRL &= ~(1<<RW); // RW = 0, write mode

LCD\_DATA = cmd;

*\_delay\_ms*(100);

LCD\_CNTRL |= (1<<EN); // EN = 1, enable read or write

*\_delay\_ms*(100);

LCD\_CNTRL &= ~(1<<EN); // EN = 0, disable read or write

*\_delay\_ms*(100);

}

void lcd\_ininit()

{

LCD\_DATA\_DDR = 0xFF; // port in output mode

LCD\_DATA = 0x00; // clear port pins

LCD\_CNTRL\_DDR |= ((1<<RS)|(1<<RW)|(1<<EN)); // port pin set to output mode

LCD\_CNTRL &= ~((1<<RS)|(1<<RW)|(1<<EN)); // clear port pins

}

void lcd\_init()

{

*\_delay\_ms*(25);

lcd\_write\_cmd(0x30);

*\_delay\_ms*(10);

lcd\_write\_cmd(0x30);

*\_delay\_ms*(5);

lcd\_write\_cmd(0x30);

// function set

// 0,0,1,DL=1(8bit), N=1(2line), F= 0(5x8 set), 0,0

lcd\_write\_cmd(0b00111000);

// display on

// 0,0,0,0,1,D=1(disp on),C=1(Cursor on), B=1(blink)

lcd\_write\_cmd(0x0F);

// clear display

lcd\_write\_cmd(0x01);

// Entry mode set

// 0,0,0,0,0,1,I/D=1, SH=0

lcd\_write\_cmd(0b00000110);

}