



ABU DHABI UNIVERSITY

MICROPROCESSORS AND FIRMWARE PROGRAMMING

Lab Report 6

Interfacing the LCD with AtMega16

Author:
Muhammad Obaidullah
1030313

Supervisor:
Dr. Mohammed Assad Ghazal

Section 1

June 27, 2012

Abstract

In this Lab we learned how to display a string on to the LCD.

1 Introduction

Question 1: To copy the code from the blackboard and change it in order to display a string eg. your name.

Question 2: Connect 8 switches to any port of Atmega except PortC and display the value on the Port as a integer on LCD.

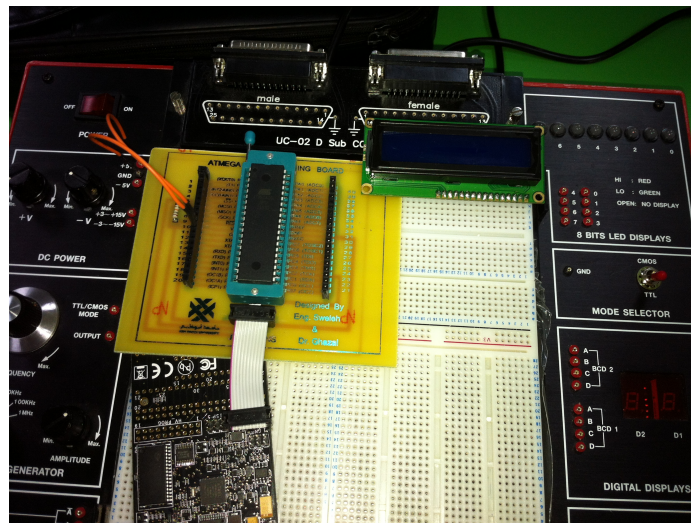


Figure 1: We were provided with a header file for displaying string on LCD and also the LCD, Atmega16, and AVR Dragon

2 Experiment Set-up

The ATmega16 chip was already mounted on a safety bracket. We had to place the bracket with the micro-controller on to the breadboard. Then we connected the micro-processor to the AVR Dragon programmer. We connected LCD with AtMega16 as shown in the *Figure 2*.

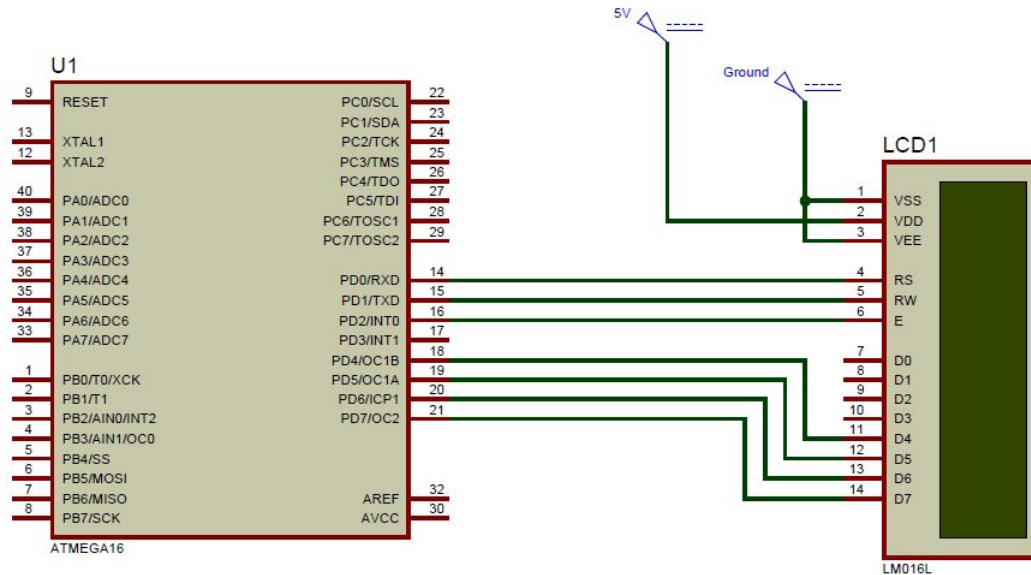


Figure 2: This is how we connect LCD to the to the ATMega16 PORTD

3 List of Equipment used

- ATmega16 micro-controller chip.
- JTAG MKII programmer.
- Wires.
- Breadboard.
- Mounting bracket for micro-controller.
- 16x2 LCD.
- 8 Switches.
- 5V power supply.
- AVR Studio IDE.
- HAPSIM.

4 Procedure

4.1 Question 1 Code

- Start AVR Studio and click on File/New/New Project.
- Write the following code into the AVR .c file.

```
#include <avr/io.h>
#include <avr/pgmspace.h>
#include <util/delay.h>
#include "lcd_lib.h"
#include <string.h>
#include <stdlib.h>
#include <stdio.h>

const char LCDwelcomeln1[] PROGMEM= "*LCD IS WORKING*\0";
const char LCDwelcomeln2[] PROGMEM="FOR ATMEGA 16\0";

void WriteLineOne(char string[])
{
    //LCDclr();
    LCDGotoXY(0,0);
    int Size = strlen(string);
    if (Size > 16) Size = 16;

    for ( int i = 0 ; i < Size ; i++)
    {
        LCDsendChar(string[i]);
    }
    for ( int i = 16-Size; i < 16 ; i++)
    {
        LCDsendChar(' ');
    }
}

void WriteLineTwo(char string[])
{
    LCDGotoXY(0,1);
    int Size = strlen(string);
    if (Size > 16) Size = 16;

    for ( int i = 0 ; i < Size ; i++)
    {
        LCDsendChar(string[i]);
    }
    for ( int i = 16-Size; i < 16 ; i++)
    {
        LCDsendChar(' ');
    }
}
```

```

}
int main(void)
{
    LCDinit();//init LCD bit, dual line, cursor right
    LCDclr();//clears LCD

    while(1)//loop demos
    {
        WriteLineOne(LCDwelcomeln1);
        WriteLineTwo(LCDwelcomeln2);
    }
}

```

4.2 Question 2 Code

- Start AVR Studio and click on File/New/New Project.
- Write the following code into the AVR .c file.

```

#include <avr/io.h>
#include <avr/pgmspace.h>
#include <util/delay.h>
#include "lcd_lib.h"
#include <string.h>
#include <stdlib.h>
#include <stdio.h>

//const char LCDwelcomeln1[] PROGMEM= "*LCD IS WORKING*\0";
//const char LCDwelcomeln2[] PROGMEM="FOR ATMEGA 16\0";

void WriteLineOne(char string[])
{
    //LCDclr();
    LCDGotoXY(0,0);
    int Size = strlen(string);
    if (Size > 16) Size = 16;

    for ( int i = 0 ; i < Size ; i++)
    {
        LCDsendChar(string[i]);
    }
    for ( int i = 16-Size; i < 16 ; i++)
    {
        LCDsendChar(' ');
    }
}

```

```

}
void WriteLineTwo(char string[])
{
    LCDGotoXY(0,1);
    int Size = strlen(string);
    if (Size > 16) Size = 16;

    for ( int i = 0 ; i < Size ; i++)
    {
        LCDsendChar(string[i]);
    }
    for ( int i = 16-Size; i < 16 ; i++)
    {
        LCDsendChar(' ');
    }
}

int main(void)
{
    LCDinit();//init LCD bit, dual line, cursor right
    LCDclr();//clears LCD
    char variableString[3];

    while(1)//loop demos
    {
        unsigned char number = PINA;
        sprintf(variableString,"%d",0b00001111&number);
        WriteLineOne(variableString);
    }
}

```

4.3 Uploading the code to ATMega16.

- Connect AVR Dragon to the computer through a USB cable and connect the AVR Dragon Pins to the micro-controller.
- connect the LEDs and the push-buttons to the ATMega16 as shown in *figure 2*.
- Click build and compile in AVR Studio.
- Run the code.

5 Results and Discussions

At the end of these exercises we got the following results:-

- Successful operation of LCD was achieved.
- The value on Port is casted into a string and then displayed on the LCD.

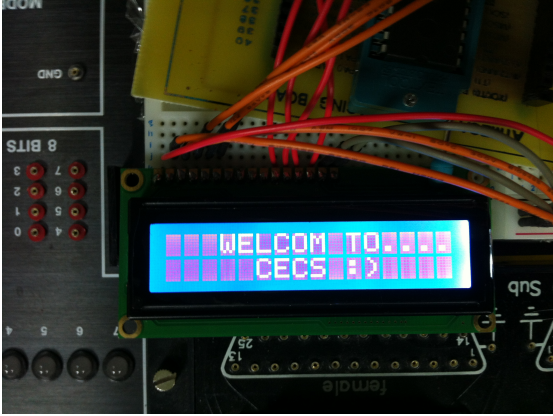


Figure 3: The LCD Prints “WELCOME TO CECS.”

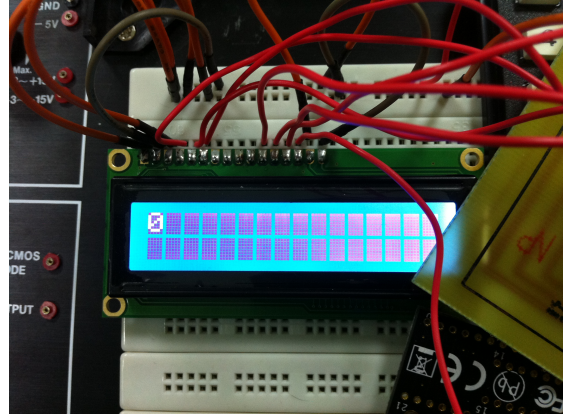


Figure 4: LCD shows the integer value on PORTB and since none of the switches are On, 0 is shown

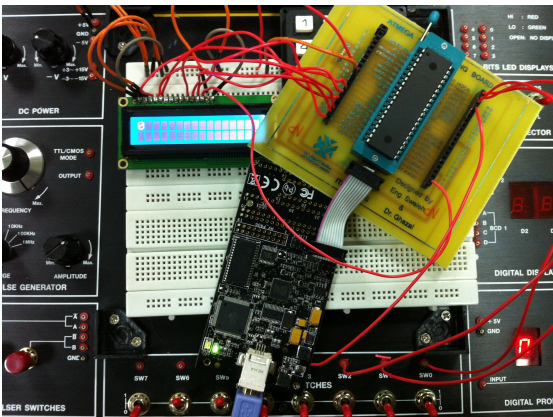


Figure 5: Overview of the experiment.

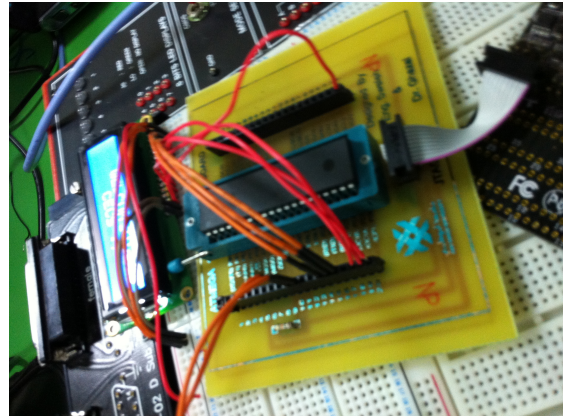


Figure 6: Connected LCD according to the schematics

- LCD should be enabled before we send the data to it.
- commands can also be sent to LCD, such as go to this position etc.

6 Conclusion

- Once you have a header file to communicate with the LCD, it gets much easier to communicate with LCD.
- LCDs have option either to receive data from only 4 pins or 8 pins, we choose to send data to 4 pins only because that way extra port is not required..