UMass Dartmouth ECE 263: Embedded System Design

Lab 1: 8-bit LCD Display

August 11 2022

We certify that this work is original
and not a product of anyone's work but our own

X	X
Sai Abhishek Bhattiprolu	Ayyappan Rajesh

Contents

1 Abstract

The goal for this lab is to print the names of both the team members on an 8-bit LCD Display connected to the ATMega 328pb board. The idea is to write the character array/string into the board to display on the screen.

2 Introduction

In this lab, students will code for AVR in C to achieve the task. This lab will use basic concepts in C and as well as in digital logic and embedded systems. An 8-bit LCD with a four other bits that include read/write and power bits. A potentiometer is connected to the display to control the contrast.

3 Methods/Procedure

System Under Test: A 328pb board and an LCD Display

Hardware Needed:

- Breadboard

- LCD Display

- Potentiometer

- Wires

- 328 Pb board

Software Needed:

- ATmel Studio

-> CODE:

1) Define the macros and bit numbers where you want your RS and E bits to go.

2) Define the state the bits are in the port of your choice (either B or C, PORT B in this case.)

3) Define your desired delay.

4) Call the functions written in this program in order to achieve the target.

5) Start out writing the main function by declaring the output portd as portd on the 328pb board has 8-bits and portb for the readstring and enable bit.

6) Initialize the LCD in the main function, refer to the tech manual.

7) For the "writecommand" function, set enable bit to 0 and RS bit to zero for ELOW and enable bit to 1 for EHIGH. Furthermore, set PORTD value to an unsigned integer value and call the delay function for time and transfer the data with a delay time of 39us.

8) In the "writedata" function, Let everything be the same as the writecommand function but set the RS bit to 1 to write the data.

9) Declare a character string array in the printstring" function along with an unsigned integer value that sets the quantity from -255 to +255. Use a while loop where the character string is not equal to null, the data will be stored in the array.

10) Finally, write a mapping function to address the co-ordinates of the array to print on the lcd display using the writecommand function with 0x80 being the MSB.

-> Circuit

1) Connect the lcd to the atmega board with their respective ports and bits.

2) Connect the lcd to the potentiometer and connect the potentiometer to the same power from the board.

3) Debug the code after setting the project properties to appropriate selections and run it.

4) The LCD should now display the names and the knob on the potentiometer should be able to control the contrast.

4 Experimental Results

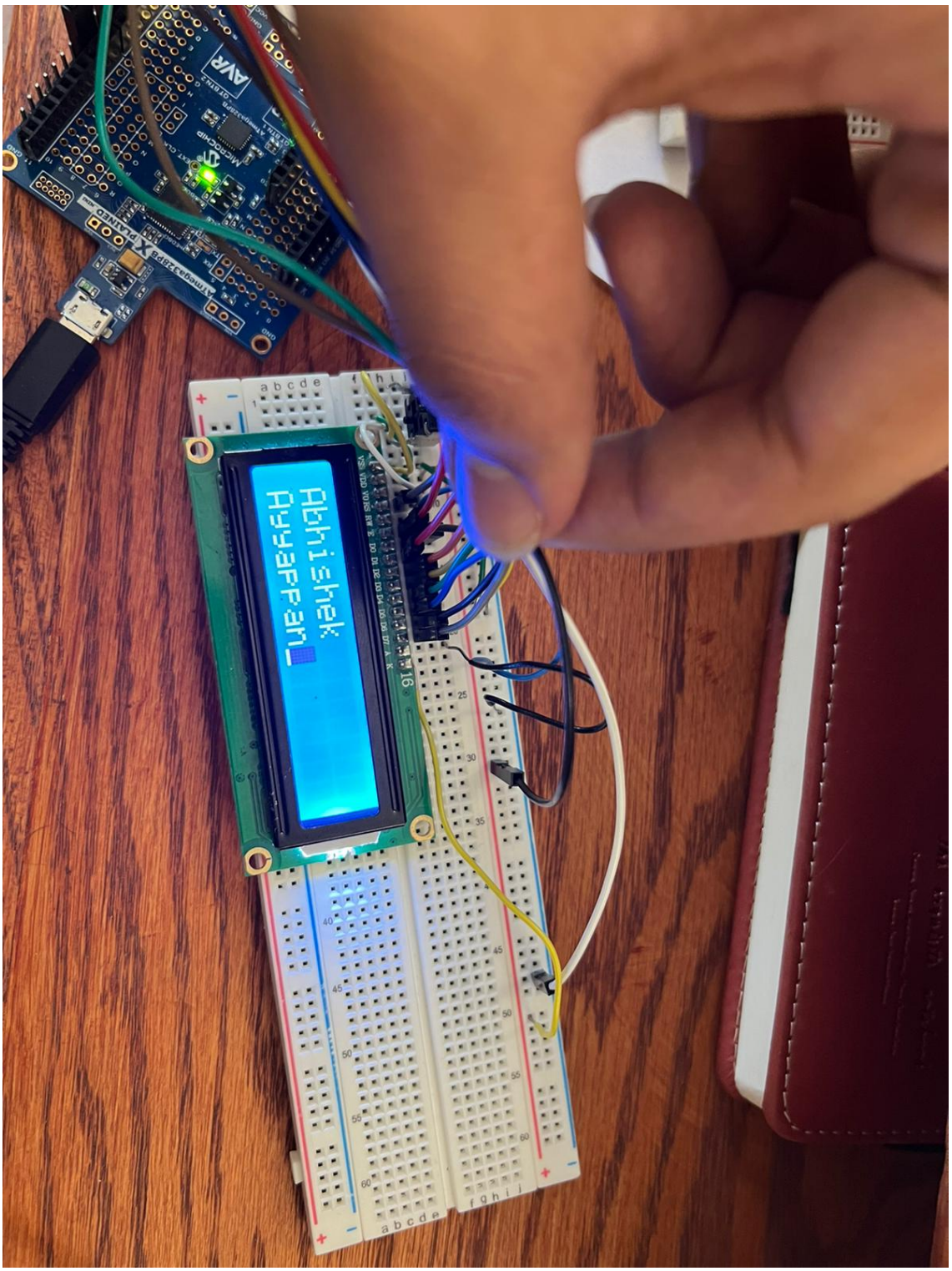

Figure 1: LCD Display

5 Discussion of Important Results

We initially had issues with the solution not being debugged properly and the lcd not initializing as it should. But after consulting the professor for help, we realized we have been working on the wrong project file. Once that issue has been fixed, all the debugging issues and the lcd issues have been solved. The lcd now displays both the names and in different contrast settings. With a little tweaking in the co-ordinates in the code, the position of the names can be shifted as well.

6 Conclusion and Connection with Theory

The program had run successfully and had displayed the required text which was our first names. Various concepts from the theory, such as functions and variables was used to achieve the results for this lab. This lab is a wonderful learning experience for students beginning to learn AVR in C and about micro-controllers along with embedded systems and digital logic/design.

7 Appendices

D:\UMass\ECE263\Labs\LCD Lab\LCD Lab\main.c 1

```c
/*
 * LCD Lab.c
 *
 * Created: 8/8/2022 2:21:46 PM
 * Author : sabhi
 */


#include <avr/io.h>
#define  F_CPU 16000000
#include <util/delay.h>



#define EBIT 1
#define RSBIT 2

inline void E_HIGH() {
    PORTB |= 1 << EBIT;
}

inline void E_LOW(){
    PORTB &= ~(1 << EBIT);
}

inline void RS_HIGH(){
    PORTB |= 1 << RSBIT;
}

inline void RS_LOW(){
    PORTB &= ~(1 << RSBIT);
    }

inline void delay250ns()
{
    asm volatile("nop\t\n" "nop\t\n" "nop\t\n" "nop\t\n");
}

void writecommand(uint8_t p);
void writedata(uint8_t p);
void printstring(char s[]);
void gotoXY(uint8_t x, uint8_t y);

int main(void)
{
    DDRD |= 0xFF;
    DDRB |= 0x3F;
    _delay_ms(100);
    writecommand(0x30);
```

Figure 2: Code Part-1

```
    _delay_ms(5);
    writecommand(0x38);
    _delay_us(100);
    writecommand(0x08);
    writecommand(0x01);
    _delay_ms(2);
    writecommand(0x06);
    writecommand(0x0F);
    gotoXY(0,0);
    printstring("Abhishek");
    gotoXY(0,1);
    printstring("Ayyappan");

    while(1)
    {
    }
}

void writecommand(uint8_t p)
{
    E_LOW();
    RS_LOW();
    E_HIGH();

    PORTD = p;
    delay250ns();

    E_LOW();
    _delay_us(39);
}

void writedata(uint8_t p)
{
    E_LOW();
    RS_HIGH();
    E_HIGH();

    PORTD = p;
    delay250ns();

    E_LOW();
    _delay_us(39);
}

void printstring(char s[])
{
    uint8_t i = 0;
    while(s[i]){
        writedata(s[i++]);
```

Figure 3: Code Part-2

```
D:\UMass\ECE263\Labs\LCD Lab\LCD Lab\main.c                                    3
    }
}

void gotoXY(uint8_t x, uint8_t y)
{
    writecommand(0x80 | (y*0x40+x));
}
```

Figure 4: Code Part-3