**ECE271: Microcomputer Architecture and Applications — University of Maine**

# Prelab for Lab #2: Liquid Crystal Display (LCD) in C

Week of 4 February 2019

## Pre-lab

### Part A – gitlab

If you haven't already, set up your account on gitlab and the git software as described by the gitlab handout on the website.

### Part B – Textbook Readings / Videos

The following might be helpful in preparing for the prelab.

1. Textbook `Chapter 17.1-17.3` to review the LCD setup
2. The classnotes posted to the course website.
3. There's a textbook Youtube Tutorial here:
   `http://web.eece.maine.edu/~zhu/book/tutorials.php`
   Lecture 14 on the LCD display.
4. If you're really interested in how LCD displays work you can search online for the *Microchip AN658: LCD Fundamentals* document.

### Part C – Prelab assignment

Just like the previous lab, we will have you plan in advance the pin configurations that you are going to write. The description of LCD-related info for the board can be found in the STM32L4 manual, Chapter 25.

The display takes 28 GPIO pins in Port A, B, and C. The display is also split up into four parts, so there are four common terminals COM0-COM3. The pins should be configured for the LCD Driver Alternative Function #11 (0xB).

## 1. Configure Port A

We want to put Port A pins 6, 7, 8, 9, 10, and 15 in Alternative Function Mode. This is done in the GPIOA MODER register, which can be found described in section 8.4.1 of the manual. To do this you will first need to clear the two bits to zero, then OR in the value (which is binary 10).

So calculate the mask below, then the value to OR in, first in binary then in hex.

| Register | 31 30 | 29 28 | 27 26 | 25 24 | 23 22 | 21 20 | 19 18 | 17 16 | 15 14 | 13 12 | 11 10 | 9 8 | 7 6 | 5 4 | 3 2 | 1 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MODER | MODER15 | MODER14 | MODER13 | MODER12 | MODER11 | MODER10 | MODER9 | MODER8 | MODER7 | MODER6 | MODER5 | MODER4 | MODER3 | MODER2 | MODER1 | MODER0 |
| Mask | | | | | | | | | | | | | | | | |
| Value | | | | | | | | | | | | | | | | |

Mask in hex:_____ Value in hex:_____

Now set the Alternative Function Register (AFR) for Port A: Pin 6, 7, 8, 9, 10, and 15. Set this to function 11 (0xB) which indicates LCD. The manual calls these AFRL and AFRH (see 8.4.9 and 8.4.10). You need 4-bits for each pin because there are 16 alternate functions. This means the values for 16 pins are too big to fit in one 32-bit register so it is split across two.

| Register | 31 30 29 28 | 27 26 25 24 | 23 22 21 20 | 19 18 17 16 | 15 14 13 12 | 11 10 9 8 | 7 6 5 4 | 3 2 1 0 |
|---|---|---|---|---|---|---|---|---|
| AFR[0] | AFRL7[3:0] | AFRL6[3:0] | AFRL5[3:0] | AFRL4[3:0] | AFRL3[3:0] | AFRL2[3:0] | AFRL1[3:0] | AFRL0[3:0] |
| Mask | | | | | | | | |
| Value | | | | | | | | |
| AFR[1] | AFRH15[3:0] | AFRH14[3:0] | AFRH13[3:0] | AFRH12[3:0] | AFRH11[3:0] | AFRH10[3:0] | AFRH9[3:0] | AFRH8[3:0] |
| Mask | | | | | | | | |
| Value | | | | | | | | |

Mask AFR[0] in hex:_____ Value AFR[0]in hex:_____
Mask AFR[1] in hex:_____ Value AFR[1]in hex:_____

## 2. Configure Port B

We want to put Port B pins 0, 1, 4, 5, 9, 12, 13, 14 and 15 as Alternative Function Mode. This is done in the GPIOB MODER register. Again, set this to binary 10.

| Register | 31 30 | 29 28 | 27 26 | 25 24 | 23 22 | 21 20 | 19 18 | 17 16 | 15 14 | 13 12 | 11 10 | 9 8 | 7 6 | 5 4 | 3 2 | 1 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MODER | MODER15 | MODER14 | MODER13 | MODER12 | MODER11 | MODER10 | MODER9 | MODER8 | MODER7 | MODER6 | MODER5 | MODER4 | MODER3 | MODER2 | MODER1 | MODER0 |
| Mask | | | | | | | | | | | | | | | | |
| Value | | | | | | | | | | | | | | | | |

Mask in hex:_____ Value in hex:_____

Now set the Alternative Function Register (AFR) for Port B: Pin 0, 1, 4, 5, 9, 12, 13, 14 and 15. Set this to function 11 (0xB) which indicates LCD.

| Register | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AFR[0] | AFRL7[3:0] | | | | AFRL6[3:0] | | | | AFRL5[3:0] | | | | AFRL4[3:0] | | | | AFRL3[3:0] | | | | AFRL2[3:0] | | | | AFRL1[3:0] | | | | AFRL0[3:0] | | | |
| Mask | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Value | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| AFR[1] | AFRH15[3:0] | | | | AFRH14[3:0] | | | | AFRH13[3:0] | | | | AFRH12[3:0] | | | | AFRH11[3:0] | | | | AFRH10[3:0] | | | | AFRH9[3:0] | | | | AFRH8[3:0] | | | |
| Mask | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Value | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Mask AFR[0] in hex:_____ Value AFR[0]in hex:_____
Mask AFR[1] in hex:_____ Value AFR[1]in hex:_____

### 3. Configure Port C

We want to put Port C pins 3, 4, 5, 6, 7, and 8 as Alternative Function Mode. This is done in the GPIOC MODER register. Again, set this to binary 10.

| Register | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MODER | MODER15 | | MODER14 | | MODER13 | | MODER12 | | MODER11 | | MODER10 | | MODER9 | | MODER8 | | MODER7 | | MODER6 | | MODER5 | | MODER4 | | MODER3 | | MODER2 | | MODER1 | | MODER0 | |
| Mask | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Value | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Mask in hex:_____ Value in hex:_____

Now set the Alternative Function Register (AFR) for Port C: Pin 3, 4, 5, 6, 7, and 8. Set this to function 11 (0xB) which indicates LCD.

| Register | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AFR[0] | AFRL7[3:0] | | | | AFRL6[3:0] | | | | AFRL5[3:0] | | | | AFRL4[3:0] | | | | AFRL3[3:0] | | | | AFRL2[3:0] | | | | AFRL1[3:0] | | | | AFRL0[3:0] | | | |
| Mask | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Value | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| AFR[1] | AFRH15[3:0] | | | | AFRH14[3:0] | | | | AFRH13[3:0] | | | | AFRH12[3:0] | | | | AFRH11[3:0] | | | | AFRH10[3:0] | | | | AFRH9[3:0] | | | | AFRH8[3:0] | | | |
| Mask | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Value | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Mask AFR[0] in hex:_____ Value AFR[0]in hex:_____
Mask AFR[1] in hex:_____ Value AFR[1]in hex:_____

**4. Configure Port D**

We want to put Port D pins 8, 9, 10, 11, 12, 13, 14 and 15 as Alternative Function Mode. This is done in the GPIOD MODER register. Again, set this to binary 10.

| Register | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MODER | MODER15 | | MODER14 | | MODER13 | | MODER12 | | MODER11 | | MODER10 | | MODER9 | | MODER8 | | MODER7 | | MODER6 | | MODER5 | | MODER4 | | MODER3 | | MODER2 | | MODER1 | | MODER0 | |
| Mask | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Value | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Mask in hex:_____ Value in hex:_____

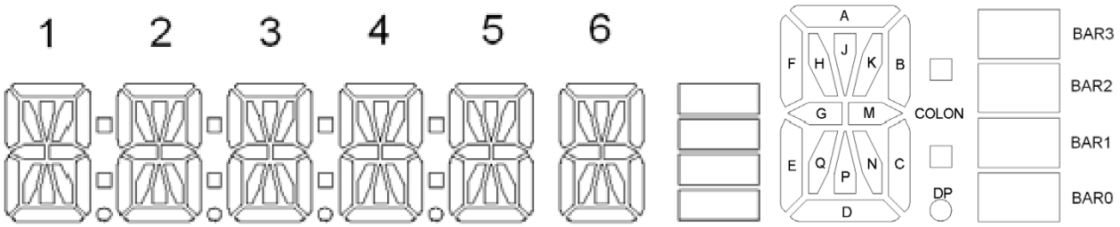
Now set the Alternative Function Register (AFR) for Port D: Pin 8, 9, 10, 11, 12, 13, 14 and 15. Set this to function 11 (0xB) which indicates LCD.

| Register | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AFR[0] | AFRL7[3:0] | | | | AFRL6[3:0] | | | | AFRL5[3:0] | | | | AFRL4[3:0] | | | | AFRL3[3:0] | | | | AFRL2[3:0] | | | | AFRL1[3:0] | | | | AFRL0[3:0] | | | |
| Mask | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Value | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| AFR[1] | AFRH15[3:0] | | | | AFRH14[3:0] | | | | AFRH13[3:0] | | | | AFRH12[3:0] | | | | AFRH11[3:0] | | | | AFRH10[3:0] | | | | AFRH9[3:0] | | | | AFRH8[3:0] | | | |
| Mask | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Value | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Mask AFR[0] in hex:_____ Value AFR[0]in hex:_____
Mask AFR[1] in hex:_____ Value AFR[1]in hex:_____

### 5. Display your last name

In the lab we will use the display to print the first 6 letters of your last name. Use the below diagram to sketch out in advance which segments need to be turned on.




Now set the bits in the table for the bits you've colored in above. Yes, I know this is a pain.

| Register | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RAM[0] | 4E | 4G | 3M | 3B | | 6G | 5M | 5B | 1M | 1B | | | | | 6E | | 3E | 3G | 2M | 2B | | | 6B | 6M | | 2E | 2G | 1E | 1G | | | |
| RAM[1] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 5E | 5G | 4M | 4B |
| RAM[2] | 4D | 4F | 3C | 3A | | 6F | 5C | 5A | 1C | 1A | | | | | 6D | | 3D | 3F | 2C | 2A | | | 6A | 6C | | 2D | 2F | 1D | 1F | | | |
| RAM[3] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 5D | 5F | 4C | 4A |
| RAM[4] | 4P | 4Q | 3: | 3K | | 6Q | -3 | 5K | 1: | 1K | | | | | 6P | | 3P | 3Q | 2: | 2K | | | 6K | -1 | | 2P | 2Q | 1P | 1Q | | | |
| RAM[5] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 5P | 5Q | 4: | 4K |
| RAM[6] | 4N | 4H | 3. | 3J | | 6H | -2 | 5J | 1. | 1J | | | | | 6N | | 3N | 3H | 2. | 2J | | | 6J | -0 | | 2N | 2H | 1N | 1H | | | |
| RAM[7] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 5N | 5H | 4. | 4J |

LCD_RAM is an array of 32-bit unsigned integers
LCD_RAM[0] in hex:_____
LCD_RAM[1] in hex:_____
LCD_RAM[2] in hex:_____
LCD_RAM[3] in hex:_____
LCD_RAM[4] in hex:_____
LCD_RAM[5] in hex:_____
LCD_RAM[6] in hex:_____
LCD_RAM[7] in hex:_____

## 6. Complete the following config table for LCD registers

Refer to the third rectangle ("LCD Configuration") in Figure 17-10 in the textbook and calculate the settings that need to be set in the LCD registers to get the display configured. The values can be found in the STM32L4 manual in Section 25.6 (p790).

If you don't have the textbook handy, these settings are also discussed in the Lab document.

| Register | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LCD_CR | | | | | | | | | | | | | | | | | | | | | | | | BUFEN | MUX_SEG | BIAS[1:0] | | | DUTY[2:0] | | | VSEL | LCDEN |
| Value | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| LCD_FCR | | | | | | | PS[3:0] | | | | DIV[3:0] | | | | BLINK[1:0] | | BLINKF[2:0] | | | | CC[2:0] | | | DEAD[2:0] | | PON[2:0] | | | UDDIE | | SOFIE | HD |
| Value | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| LCD_SR | | | | | | | | | | | | | | | | | | | | | | | | | | | FCRSF | RDY | UDD | UDR | SOF | ENS |
| Value | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| LCD_CLR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | UDDC | | SOFC | |
| Value | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |