Lab 2: Liquid Crystal Display (LCD) Driver in C
Instructor: Prof. Yifeng Zhu
Spring 2016

Goals

1. Understand alternative function of GPIO pins
2. Understand basic concepts of an LCD driver, particularly ***Bias*** and ***Duty Ratio***
3. Understand concepts of double buffer memory to ensure the coherency of the displayed information
4. Understand clock configurations of GPIO pins and LCD drivers

Pre-lab Assignment:

1. Read Chapter 17 of Textbook. (Note: Table 17-1 and Table 17-2 are changed for STM32L4. This lab description gives the updated tables)
2. Complete the pin configuration tables included in this handout

In-Lab Assignment:

1. Complete *LCD_PIN_Init*( ) and *LCD_Configure*( ).
2. Complete *LCD_Display_Name*( ) to display the first six letters of your last name. You cannot call *LCD_DisplayString*( ) in *LCD_Display_Name*( ).
3. Complete *LCD_DisplayString*( ) to display a short string. The string has only letters and numbers only, with a length less than 7.
4. Something cool. This following gives a few examples.
    a. LCD cool animations
    b. LCD scrolling to display a long string
    c. Set LCD contrast min-->max-->min by pressing user button

Introduction

PIN configuration: A total of 28 GPIO pins from Port A, B, and C drive the LCD display, as shown below. The duty ratio of this LCD is 4 and therefore there are four common terminals (COM0-COM3), which are connected to four GPIO pins. The other 24 GPIO pins are mapped to pixel bits stored in the internal LCD RAM. The mapping between GPIO pins and LCD RAM are given in the textbook. Each pin should be configured as Alternative Function 11 (LCD Driver).

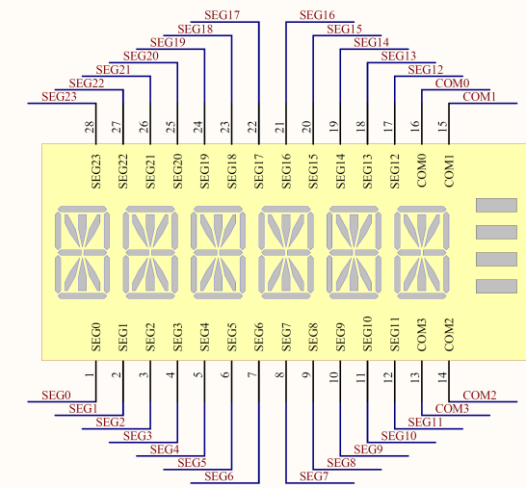


Figure 1. PIN connection to six 14-segment digits and 4 bars.

Modification to the program flowchart (Figure 17-7 in Textbook): For the STM32L4 discovery kit, the MUS_SEG bit in the LCD_CR must be ***cleared***. SEG[31:28] are not multiplexed with SEG[43:40].

LCD (24 segments, 4 commons, multiplexed 1/4 duty, 1/3 bias) on DIP28 connector

VLCD = PC3			
COM0 = PA8 (LCD_COM0)	COM1 = PA9 (LCD_COM1)	COM2 = PA10 (LCD_COM2)	COM3 = PB9 (LCD_COM3)
SEG0 =PA7 (LCD_SEG4)	SEG6 = PD11 (LCD_SEG31)	SEG12 = PB5 (CD_SEG9)	SEG18 = PD8 (LCD_SEG28)
SEG1 = PC5 (LCD_SEG23)	SEG7 = PD13 (LCD_SEG33)	SEG13 = PC8 (LCD_SEG26)	SEG19 = PB14 (LCD_SEG14)
SEG2 = PB1 (LCD_SEG6)	SEG8 = PD15 (LCD_SEG35)	SEG14 = PC6 (LCD_SEG24)	SEG20 = PB12 (LCD_SEG12)
SEG3 = PB13 (LCD_SEG13)	SEG9 = PC7 (LCD_SEG25)	SEG15 = PD14 (LCD_SEG34)	SEG21 = PB0 (LCD_SEG5)
SEG4 = PB15 (LCD_SEG15)	SEG10 = PA15 (LCD_SEG17)	SEG16 = PD12 (LCD_SEG32)	SEG22 = PC4 (LCD_SEG22)
SEG5 = PD9 (LCD_SEG29)	SEG11 = PB4 (LCD_SEG8)	SEG17 = PD10 (LCD_SEG30)	SEG23 = PA6 (LCD_SEG3)

	LCD					
STM32L Pin	**LCD Pin**	**COM3**	**COM2**	**COM1**	**COM0**	**LCD Pin**
PA7 (LCD_SEG4)	1	1N	1P	1D	1E	SEG 0
PC5 (LCD_SEG23)	2	1DP	1COLON	1C	1M	SEG 1
PB1 (LCD_SEG6)	3	2N	2P	2D	2E	SEG 2
PB13 (LCD_SEG13)	4	2DP	2COLON	2C	2M	SEG 3
PB15 (LCD_SEG15)	5	3N	3P	3D	3E	SEG 4
PD9 (LCD_SEG29)	6	3DP	3COLON	3C	3M	SEG 5
PD11 (LCD_SEG31)	7	4N	4P	4D	4E	SEG 6
PD13 (LCD_SEG33)	8	4DP	4COLON	4C	4M	SEG 7
PD15 (LCD_SEG35)	9	5N	5P	5D	5E	SEG 8
PC7 (LCD_SEG25)	10	BAR2	BAR3	5C	5M	SEG 9
PA15 (LCD_SEG17)	11	6N	6P	6D	6E	SEG 10
PB4 (LCD_SEG8)	12	BAR0	BAR1	6C	6M	SEG 11
PB9 (LCD_COM3)	13	**COM3**				
PA10 (LCD_COM2)	14		**COM2**			
PA9 (LCD_COM1)	15			**COM1**		
PA8 (LCD_COM0)	16				**COM0**	
PB5 (LCD_SEG9)	17	6J	6K	6A	6B	SEG 12
PC8 (LCD_SEG26)	18	6H	6Q	6F	6G	SEG 13
PC6 (LCD_SEG24)	19	5J	5K	5A	5B	SEG 14
PD14 (LCD_SEG34)	20	5H	5Q	5F	5G	SEG 15
PD12 (LCD_SEG32)	21	4J	4K	4A	4B	SEG 16
PD10 (LCD_SEG30)	22	4H	4Q	4F	4G	SEG 17
PD8 (LCD_SEG28)	23	3J	3K	3A	3B	SEG 18
PB14 (LCD_SEG14)	24	3H	3Q	3F	3G	SEG 19
PB12 (LCD_SEG12)	25	2J	2K	2A	2B	SEG 20
PB0 (LCD_SEG5)	26	2H	2Q	2F	2G	SEG 21
PC4 (LCD_SEG22)	27	1J	1K	1A	1B	SEG 22
PA6 (LCD_SEG3)	28	1H	1Q	1F	1G	SEG 23

Lab 2:Pre-Lab Assignment

Student Name: _____
TA: _____
Time & Date: _____

1. Configure Port A: Pin 6, 7, 8, 9, 10, 15 as Alternative Function Mode

GPIO Mode: Digital Input (00, reset), Digital Output (01), Alternative Function (10), Analog (11)

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[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]		MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
MASK																																
VALUE																																

GPIOA Mode Register MASK Value = 0x_____ (in HEX)
GPIOA Mode Register Value = 0x_____ (in HEX)

Configure Port A: Pin 6, 7, 8, 9, 10, and 15 as Alternative Function 11 (0x0B)

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																																

GPIOA Alternative Function Register [0] MASK = 0x_____ (in HEX)
GPIOA Alternative Function Register [0] = 0x_____ (in HEX)
GPIOA Alternative Function Register [1] MASK = 0x_____ (in HEX)
GPIOA Alternative Function Register [1] = 0x_____ (in HEX)

2. Configure Port B: Pin 0, 1, 4, 5, 9, 12, 13, 14, and 15 as Alternative Function Mode

GPIO Mode: Digital Input (00, reset), Digital Output (01), Alternative Function (10), Analog (11)

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[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]		MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
MASK																																
VALUE																																

GPIOB Mode Register MASK Value = 0x_____ (in HEX)
GPIOB Mode Register Value = 0x_____ (in HEX)

Configure Port B: Pin 0, 1, 4, 5, 9, 12, 13, 14, and 15 as Alternative Function 11 (0x0B)

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																																

GPIOB Alternative Function Register [0] MASK = 0x_____ (in HEX)
GPIOB Alternative Function Register [0] = 0x_____ (in HEX)
GPIOB Alternative Function Register [1] MASK = 0x_____ (in HEX)
GPIOB Alternative Function Register [1] = 0x_____ (in HEX)

3. Configure Port C: Pin 3, 4, 5, 6, 7, and 8 as Alternative Function Mode

GPIO Mode: Digital Input (00, reset), Digital Output (01), Alternative Function (10), Analog (11)

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[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]		MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
MASK																																
VALUE																																

GPIOC Mode Register MASK Value = 0x_____ (in HEX)
GPIOC Mode Register Value = 0x_____ (in HEX)

Configure Port C: Pin 3, 4, 5, 6, 7, and 8 as Alternative Function 11 (0x0B)

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																																

GPIOC Alternative Function Register [0] MASK = 0x_____ (in HEX)
GPIOC Alternative Function Register [0] = 0x_____ (in HEX)
GPIOC Alternative Function Register [1] MASK = 0x_____ (in HEX)
GPIOC Alternative Function Register [1] = 0x_____ (in HEX)

4. Configure Port D: Pin 8, 9, 10, 11, 12, 13, 14, and 15 as Alternative Function Mode

GPIO Mode: Digital Input (00, reset), Digital Output (01), Alternative Function (10), Analog (11)

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[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]		MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
MASK																																
VALUE																																

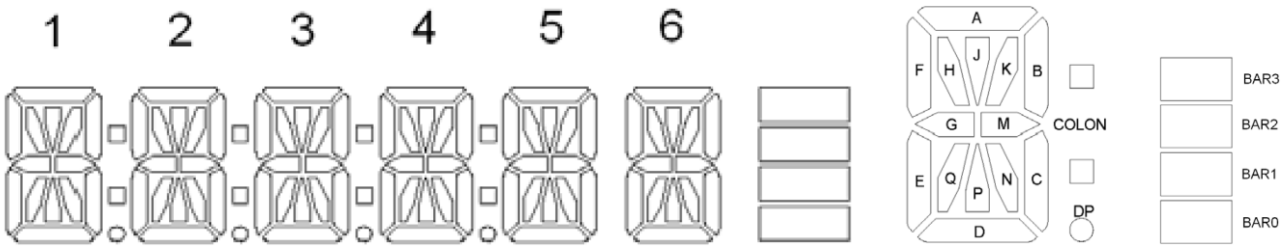
GPIOD Mode Register MASK Value = 0x_____ (in HEX)
GPIOD Mode Register Value = 0x_____ (in HEX)

Configure Port D: Pin 8, 9, 10, 11, 12, 13, 14, and 15 as Alternative Function 11 (0x0B)

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																																

GPIOD Alternative Function Register [0] MASK = 0x_____ (in HEX)
GPIOD Alternative Function Register [0] = 0x_____ (in HEX)
GPIOD Alternative Function Register [1] MASK = 0x_____ (in HEX)
GPIOD Alternative Function Register [1] = 0x_____ (in HEX)

Write down your last name, and complete the following table.



Your Last Name: _____ (First Six Characters)

	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_RAM[0]	4E	4G	3M	3B		6G	5M	5B	1M	1B					6E		3E	3G	2M	2B			6B	6M		2E	2G	1E	1G			
LCD_RAM[1]																													5E	5G	4M	4B
LCD_RAM[2]	4D	4F	3C	3A		6F	5C	5A	1C	1A					6D		3D	3F	2C	2A			6A	6C		2D	2F	1D	1F			
LCD_RAM[3]																													5D	5F	4C	4A
LCD_RAM[4]	4P	4Q	3 Col	3K		6Q	3 Bar	5K	1 Col	1K					6P		3P	3Q	2 Col	2K			6K	1 Bar		2P	2Q	1P	1Q			
LCD_RAM[5]																													5P	5Q	4 Col	4K
LCD_RAM[6]	4N	4H	3 DP	3J		6H	2 Bar	5J	1 DP	1J					6N		3N	3H	2 DP	2J			6J	0 Bar		2N	2H	1N	1H			
LCD_RAM[7]																													5N	5H	4 DP	4J

LCD_RAM is an array of 32-bit unsigned integers.

LCD_RAM[0] = 0x_____ (in Hex)
LCD_RAM[1] = 0x_____ (in Hex)
LCD_RAM[2] = 0x_____ (in Hex)
LCD_RAM[3] = 0x_____ (in Hex)
LCD_RAM[4] = 0x_____ (in Hex)
LCD_RAM[5] = 0x_____ (in Hex)
LCD_RAM[6] = 0x_____ (in Hex)
LCD_RAM[7] = 0x_____ (in Hex)

Complete the following configuration table for LCD registers.

1. Refer to Figure 17-7 of Textbook and STM32L4 Reference Manual to complete the following table.
2. For STM32L4 discovery kit, the mux segment of the LCD_CR must be ***disabled,*** i.e. the MUX_SEG bit in the LCD_CR register must be cleared. SEG[31:28] are not multiplexed with SEG[43:40]. The flow chart (Figure 17-7 in textbook) is for STM32L1 discovery kit and it sets MUX_sets bit in the LCD_CR register to make output pins SEG[43:40] have function SEG[31:28].

Offset	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
0x00	LCD_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUFEN	MUX_SEG	BIAS[1:0]		DUTY [2:0]			VSEL	LCDEN
	value																																
0x04	LCD_FCR	Res.	Res.	Res.	Res.	Res.	Res.	PS[3:0]				DIV[3:0]				BLINK[1:0]		BLINKF[2:0]			CC [2:0]			DEAD [2:0]			PON [2:0]			UDDIE	Res.	SOFIE	HD
	value																																
0x08	LCD_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FCRSF	RDY	UDD	UDR	SOF	ENS
	value																																
0x0C	LCD_CLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UDDC	Res.	SOFC	Res.
	value																																

Lab 2: In-Lab Assignment

The basic requirement of this lab is to display your last name on the LCD. Refer to Textbook for the flow charts.

Notes:

1. The code uses the ***stm32l4xx.h*** head file provided in the STM library. It includes many useful macro definitions and data structures. Using them makes your code easier to understand and debug.

2. The program code to initialize the LCD clock is provided to you. The function is ***LCD_Clock_Init***();
   a. The LCD clock is the same clock as the Real-time clock (RTC). RTC clock domain is protected by default. To configure the LCD clock source, the RTC domain needs to be unlocked first by writing "0xCA" and "0x53" to the RTC->WPR register.

3. You are required to implement four functions:
   a. ***LCD_PIN_Init***() that enables GPIO clocks and configures GPIO pins as the alternative function 11 (LCD)
   b. ***LCD_Configure***() that performs the LCD configuration in the flow chart
   c. ***LCD_Display_Name***() that display the first six letters of your last name
   d. ***LCD_Display_String***() that sets up the LCD_RAM and displays the input string on LCD.
   e. ***LCD_Clear***() that clear the LCD screen.

4. Examples of something cool
   a. LCD cool animations
   b. LCD scrolling to display a long string
   c. Set LCD contrast min → max → min by pressing user button
   d. Something really cool

Lab Demo Questions:

1. In the LCD_WriteChar( ) function, why do we use the following while statement
   ```
   while ((LCD->SR & LCD_SR_UDD) == 0);
   ```
2. What clock is used to drive the LCD? How can you find out? (Hint: check RCC register value in debug environment)
3. Explain to TA why double-buffering can ensure the coherency of the displayed information

Lab 2: Post-Lab Assignment

Answer the following questions in the file Readme.md and submit it with your lab code to the gitlab server.

1. Suppose the duty ratio of a LCD display is ¼ and it has a total of 120 display segments (pixels). How many pins are required to drive this LCD?

2. Can a GPIO pin perform all alternative functions simultaneously?

3. Is the LCD driver (programmed in this lab) built in within the processor chip? What is the function of the COM driver and SEG driver?

4. How many pixels can the STM32L4 processor LCD driver drive? How large is the LCD_RAM in terms of bits? (Read STM32L4 Reference Manual)

5. How many pixels does the LCD installed on the STM32L4 discovery kit have? Explain why many LCD_RAM bits are not used for this LCD?