

LCD Display B

Objective:

Students will design and write a C language program to interface the LCD display. They will use outline and logic from last week's lab to incorporate more advanced functionality. Students will define constants and add code features to control the cursor, display characters on the second line when strings longer than 16 characters are passed, perform a busy check (in place of the loop delay) when writing characters and commands to the screen after the initialization has been completed, and send commands to change the display background color.

Program Development Design worksheet:

Read lab instruction completely, and then complete steps 1-3 for your program planning and outline. You will turn your completed design worksheet in as a well-documented word .docx.

Lab 07 description and requirements:

- **Cursor Commands**

Return the display to the home position (command 0x02) and clear the display (command 0x01) before displaying a string. This will not make a difference for the first string (but it won't hurt to clear the display on the first string), but it will matter for subsequent strings. After 16 characters are sent to the display, move the cursor to the second line by sending the command 0xc0. All commands are binary values (we will use Hex notation). So, don't mistake them for ASCII values as used for letters displayed on the display.

Defines should be used for constants. These defines are placed in a defs.h file

```
#define HOME_DISPLAY 0x02
#define CLEAR_DISPLAY 0x01
#define SECOND_LINE 0xc0
```

- **Busy Check**

Add a subroutine that is called to check the display status. The LCD status is checked by reading Port K bit 7. When the bit goes low the display is no longer busy. While the bit is high the display is busy. To read this bit: Port k bit 7 must be programmed as Input (Throughout the program PORK has been programmed as Output) and RS, RW and enable lines must be assigned as follows:

In your function

1. Prepare the lines to read the busy bit:
DDRK bit 7 must be placed in the input mode (ddr K – bit 7 low). Next, place the RS line in the command mode (Port D – bit 5 low), the R/W* line in the read mode (Port D – bit 6 high), and the enable line in the enable mode (Port D – bit 7 high).
2. Reading the display status at Port K bit 7:
While Port K bit 7 is high, the display is busy. If the display is busy the E line must be taken high and then low again to read the updated status.
3. Once the busy port K bit 7 is low: before leaving the function, re-set ddr k bit 7 back to an output.

IMPORTANT: The busy check cannot be used during the display initialization process, but can be used for all display writes (both data and commands) after the initialization.

- **Background Color**

The display background color can be changed by writing to Port E bits 3-2. The following table shows the relationship between the values written and background color. Bits 2 and 3 of Port E are the backlight color select lines. The backlight may be Red, Green, or Blue. See table below. Bit 2 is the LSB and Bit 3 is the MSB; these pins are connected to a decoder and inverter in order to select and drive the color of the backlight.

Port E Bits (3:2) Combination	LCD Background Color
00 ₂	Off
01 ₂	Red
10 ₂	Green
11 ₂	Blue

Add a function to your display code that can be called with the following constants that are defined in a header file named defs.h. Assign appropriate macro values for each #define constant listed below.

```
DISPLAY_OFF
DISPLAY_RED
DISPLAY_GREEN
DISPLAY_BLUE
```

Code similar to the following can be used to test your display routines. The test software can be placed in function main. **Note: even though you are using the check busy you will still need the delay function for the initialization and to stall the program enough for the hardware to keep up when passing new strings.**

```
LCDout("My Short Message");
Delay(??); //pick enough time to see changes
LCDout("My Longer Message to the LCD");
Delay(??); //pick enough time to see changes
```

Test your subroutines by calling the initialization routine and then calling the display subroutine with several different strings that are different lengths. Make sure one of the messages has more than 16 characters. As shown above, delay several seconds between each message, using your Delay() function, so that you can verify correct operation. Only after this works should you add the check busy function for the data lines and the display color changes.

Before you submit: ensure each of your files includes a commented header, a brief description of the program or file purpose and well organized and commented code.

Submission: Submit your completed well organized design worksheet, main.c, LCD.c defs.h, and protos.h files as well as a zipped version of your entire project to Canvas assignments for Lab 07