Lab 7

Due Dates:
Submit this lab and the extended learning (optional) to TEACH by Monday at 11:59pm. You must get checked off by a TA by the beginning of lab 8.

Prelab:
1. In the statement below, does A get the value of X or does X get the value of A?
   \[ X = A; \]
2. If there are switches on port B and LEDs on port D, give the commands to:
   a.) Read from the switches on port B and write them to port D.
   b.) Copy the value of the 4 most significant bits of the switches on port B to the 4 most significant bits of port D without changing the value of port D’s lower 4 bits.
3. What value in binary should PORTB have assigned to it to display amber LEDs?
4. In your own words describe the meaning of the term ‘persistence of vision’.

Embedded Programming Overview:
Displaying images is needed in many applications. The LED array on the Wunderboard is an 8x8 pixel matrix capable of displaying red, green, or amber in each pixel. In this lab, we will practice using the display to do some simple visualizations. Before attempting this lab, please read the Wunderboard usage guide on the tekbots.com webpage for a better understanding of the LED array.

Procedure:
The LED array on the Wunderboard has three different control inputs; row, column, and color. The rows are connected to Port C, the columns are connected to the lower 3 bits of port E (PE2 PE1 PE0) and the color selections are connected to two of the port B pins, PB7 (Green) and PB6 (Red). In this lab you will work towards making a program that displays an image of your choice on the display.

Task 1:
The first step is to prove that you can change colors and columns on the display. Starter code is available on the tekbots.com webpage. Start by ‘setting’ the bit for the red LEDs. This could be done by assigning PORTB the value of 0b01000000. You also need to send some binary pattern to PORTC to select rows. Once this is done, you should have an led array with column 0 having some red lights on. See Figure 1. Please note based on what pattern you chose, your display might not look exactly like the figure. Show your TA.

Figure 1: Task 1 Example
Task 2
Now practice controlling the column that is illuminated. Columns
are controlled with the 3 least significant bits of port E. Writing a 0
to the PORTE would enable the LEDs for the right most column.
Figure 2 shows column 3 illuminated. Writing a 7 to PORTE would
enable the left most column. Choose a column other than 0. Once
completed, show your TA.

Task 3
What if you wanted to display two colors at once? To do this, you
need to use something called persistence of vision. Persistence of
vision is the concept that the eye will ‘remember’ what it has seen
and combine several very fast images into a single image. This
concept was used in television for years before digital broadcast.
What it means for the Wunderboard is that you need to switch very quickly between two different
patterns in 2 different colors. Your eye will combine them for you. For example displaying red and
green at the same time might look like Figure 4. Please note though that you would need to actually
turn off one color before turning on the other color to avoid accidentally blending colors as in
Figure 3. Once completed, show your TA.

Task 4
This task is to display an ‘image’ of your own design. Remember you have only 8 by 8 pixels so plan
accordingly. To display an image, again we turn to the idea of the persistence of vision. This time
however rather than changing colors very quickly, we can ‘index’ through the columns very quickly
to display the information. HINT: The first time you try this, it might be very dim. To get the display
to be brighter you would need to leave the LEDs on longer and make the switching between each
column shorter. You may want to look into the delay_ms() function. Once completed, show your
TA. An example image is shown in Figure 5.
Demonstrate and Submit Code
When your code is submitted, it will be processed both to ensure it compiles and runs correctly and to evaluate its comments. Comment every line that you add. The goal is that you understand what and why every line works even if you read your code a year from now.
When ready, submit your source code to TEACH. Your c file should be named main.c
TEACH Assignment Name: Lab7

Study Questions:

1. You want to display red and green LEDs at the same time without accidentally blending them. Write a simple while() loop that will write green LEDs and red LEDs without blending them.

2. You probably noticed when displaying your LED array that one column was brighter than the others. Why is this and how would you prevent it?

Lab 7 Summary:

<table>
<thead>
<tr>
<th>TASK</th>
<th>Completed?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prelab</td>
<td>4pts.</td>
</tr>
<tr>
<td>Task 1</td>
<td>2pts.</td>
</tr>
<tr>
<td>Task 2</td>
<td>2pts.</td>
</tr>
<tr>
<td>Task 3</td>
<td>4pts.</td>
</tr>
<tr>
<td>Task 4</td>
<td>6pts.</td>
</tr>
</tbody>
</table>

Extended Learning:
Modify your code to display an image with red, green, and amber pixels. Once completed, show your TA. The image can be of your choosing, but at least a few rows must have more than one color displayed.

Your file should be named main.c
Demonstrate your program to a TA at the beginning of Lab 8.
Submit your code to TEACH Lab7Extra by next Monday at 11:59pm.