**EGR326 PreLab 1 F22**

During this week’s lab exercise you will be exploring the microcontroller (MCU) digital inputs and outputs (digital I/O) by writing a simple program to light a multicolored RGB (red, green, blue) LED display and sequencing through the colors based on pushbutton switch input. You will also incorporate both timing (SysTick) and GPIO interrupts to complete your tasks.

Read the on-line reference articles “A Guide to Debouncing” by Jack Ganssle (part I and part II, <http://www.ganssle.com/debouncing.htm>). Code from the article is copied below.

The Ganssle articles describe the typical switch bounce problem and provides some hardware and software solutions. This week in lab you need to implement a debounce solution when you design and build an interface between a momentary pushbutton switch and the MCU I/O port.

When connecting an external RGB LED (such as the one in your lab kit) to the MSP432, each lead must have a properly chosen current limiting resistor to prevent damage to the MCU I/O ports and the LEDs (remember this from EGR 226?).

To prepare for this week’s lab:

1. Download the lab 1 exercise from the Blackboard Assignments folder and review it so that you will come to lab prepared to make the most efficient use of lab time.
2. Start a new page in your lab notebook and title it lab 1. Write the lab objectives in your own words.
3. In your lab notebook, design an interface between the I/O port pins on the Launchpad that are connected to the on-board RGB LED (see figure 1 of the lab 2 exercise) and the RGB LED from your kit. You will remove the jumpers on the board to gain access to these port pins or use FETs available in the lab and use alternate port pins.

These Port 2 I/O pins have a maximum current rating of 20 mA, but, limit the current in each LED to no **more than 10 mA**. Start with this current for each LED and then you can adjust a current downward if you wish to balance the light intensity.

1. The red, green, and blue LED elements have different forward bias voltages and current vs. brightness characteristics (remember to include the forward voltage drop of each color LED in your calculations, see the data sheet for this information). Pick standard 10% tolerance resistor values and label them on your schematic drawing along with all pin connections and I/O port pin numbers. A table of standard resistance values is posted to Blackboard in the Readings and References folder, use a serial or parallel combination that results in close to the desired value (don’t use more than 2 resistors for each).
2. Summarize what you learned about switch bounce from reading the Ganssle articles. Draw a schematic diagram of one of the hardware solutions.
3. Describe what method you plan to implement to control switch bounce. You can try both hardware and software solutions, but, you need to have a plan for at least one implementation.

If you plan to use a software solution, include the internal pull-up resistor for the switch interface (as discussed in lecture) in your schematic diagram.

After you have completed this assignment and documented it in your notebook, scan the notebook pages (each page should have your name, date, and title of the exercise in the top margin) and submit them **as a single .pdf file** to the Blackboard assignment.

Adjust the image resolution so that the entire document is no larger than necessary (100 Kbytes per page is usually adequate).

GANSSLE switch bounce routine:

**// Service routine called at regular timer intervals**

**uint8\_t DebounceSwitch1()**

**{**

**static uint16\_t State = 0; // Current debounce status**

**// read switch, upper 5 bits of State are don't cares**

**State=(State<<1) | (P1IN & 0x2)>>1 | 0xf800;**

**if(State==0xfc00)return 1; // indicates 0 level is**

**// stable for 10 consecutive calls**

**return 0;**

**}**

**pass switch line value of State return**

**1 pulled up 1111 1000 0000 0001 0**

**10 pulled up 1111 1011 1111 1111 0**

**n switch pressed 1111 1111 1111 1110 0**

**n+1 switch bounce 1111 1111 1111 1101 0**

**n+2 switch bounce 1111 1111 1111 1010 0**

**n+3 switch bounce 1111 1111 1111 0101 0**

**n+4 switch press stable 1111 1111 1110 1010 0**

**n+5 switch press stable 1111 1111 1101 0100 0**

**n+10 switch press stable 1111 1100 0000 0000 1**

**n+11 switch press stable 1111 1000 0000 0000 0**