Driver Project 4: IO Interrupts, Timers and UART Display Drivers (Group Project)

Due Date: As per D2L.

Assignment:

Using the Microcontroller and the driver topics on IOs, Timers, interrupts and UART display drivers covered so far, you will design a simple IO controller to test out IO and Timer Interrupts with display drivers. Design a state machine to turn on, turn off and blink a LED connected to port RB8 based on the push buttons (PBs) connected to the input ports RA2, RA4 and RB4 as shown in the schematic in the lecture slide 'HW and IO Control.pdf'. PB1, PB2 and PB3 represent push buttons connected to ports RA2, RA4 and RB4 respectively. The state machine should operate as follows:

User input(s)	**Output(s)**
While PB1 is pressed	LED blinks at approx. 0.5 sec intervals (0.5 sec on and 0.5 sec off) "PB1 is pressed" is displayed on PC terminal
While PB2 is pressed	LED blinks at approx. 2 sec intervals (2 sec on and 2 sec off) "PB2 is pressed" is displayed on PC terminal
While PB3 is pressed	LED blinks at approx. 3 sec intervals (3 sec on and 3 sec off) "PB3 is pressed" is displayed on PC terminal
While 2 PBs are pressed together	LED stays on without blinking "PB_ and PB_ are pressed" is displayed on PC terminal. " The blanks __ should display the corresponding PB numbers pressed
While all 3 PBs are pressed together	LED stays on without blinking "All PBs pressed" is displayed on PC terminal
No PBs pressed	LED stays off "Nothing pressed" is displayed on the PC terminal

Additional info:

1. Implement the above controller using the hardware kit and your code, which will be designed using basic ANSI C commands; IO and Timer interrupts; and Display driver functions provided.
2. Use of polling and manufacturer-provided timer functions instead of interrupts will lose points. Function names: Students can use any convention when naming functions or organizing code.
3. Display instructions: All displays on the PC terminal window should be on a single line. Note that display functions carried out at 32 kHz (300 Baud) can affect timer delays. Your code should account for such delays when producing delays specified in the table above.

4. Interrupts: Interrupt ISR names are provided in the lecture slides. As specified in lecture, IO (CN interrupts) are triggered on rising and falling edges and due to any debounce effects of the push buttons. A debounced switch will result in several hi to lo and lo to hi fluctuations at the Microcontroller input before stabilizing to a steady and fixed voltage when the switch is pressed. Your code should filter out any such effects.

Note:

Port RA2 is one of those exceptional ports that is also multiplexed to the input for an external oscillator and an analog input port. To be able to use it as a digital input with a pushbutton, it's multiplexed analog input has to be disabled by including the following line of code in your IOinit() function. We will revisit this multiplexing when we look at ADC converters in a couple of weeks.

```
AD1PCFG = 0xFFFF; // Turn all analog pins as digital
```

Deliverables:

This is a group project. Each group should upload the following onto their respective group D2LDropbox folder created:

1. **Zipped up file of the project. MPLAB projects can be zipped up by right clicking on the project and selecting package (See screenshot below). The zipped project is saved in the same project folder created by user. Make sure your driver code is commented properly especially any mathematical computations used.**
2. **Link to your video demo uploaded on youtube, Vimeo or similar video hosting website along with the zipped up project. Dropbox or Google or OneDrive links are allowed as well but ensure that videos are in .mp4 or .mov format. Videos uploaded in any other format will lose points. Video demo should be as follows:**
    a. **Single recording no more than 2 mins long**
    b. **Show UCID or government issued ID cards of all 3 group members placed in front of the computer with MPLAB and/or hardware running**
    c. **Demo of the code and hardware operation showing the following:**
        i. **Each of the PBs pressed individually - showing LED status and UART display**
        ii. **2 PBs pressed simultaneously – showing LED status and UART display**
        iii. **3 PBs pressed simultaneously – showing LED status and UART display**
        iv. **No PBs pressed - Showing LED status and UART display**
        v. **Repeat one of the PBs pressed individually - showing LED status and UART display (to show if there are any code hangups in previous states)**
    d. **Explanation of the code organization in MPLAB including any special power or time saving features (i.e. interrupts, clock switching, sleep/idle) used, and respective contribution of each group member towards code development and hardware/software testing.**

Grading rubric: (Total = 10 points)

- Correct setup and use of timers, IOs, UART, interrupts and clock modules = 5
- Code optimization for better power savings and time resolution i.e clock switching, interrupts, sleep/idle modes for speedy message displays and power savings = 2
- Proper video and code upload format including commenting of all driver lines of code = 2
- Group participation = 1