Experiment 7: FreeRTOS Introduction

Objectives:

- To gain experience with the TM4C123 port of FreeRTOS.
- To gain experience with the FreeRTOS API.
- To gain experience with the TI Educational BoosterPack MKII

Reading:

- Tiva C Series TM4C123x ROM User's Guide (course website or http://www.ti.com/lit/pdf/spmu367)
- TivaWare Peripheral Driver Library User's Guide course website or http://www.ti.com/lit/spmu298)
- FreeRTOS Documentation "Hands On Tutorial Guide" and "Reference Manual" (http://www.freertos.org/Documentation/RTOS_book.html)
- FreeRTOS Documentation: "Running the RTOS on a ARM Cortex-M Core" (http://www.freertos.org/RTOS-Cortex-M3-M4.html)
- BOOSTXL-EDUMKII User's Guide (http://www.ti.com/lit/ug/slau599a/slau599a.pdf)
- Valvano's Board Support Source Code (TM4C123Valvanoware/inc/BSP.c and TM4C123Valvanoware/inc/BSP.h)

Introduction:

In this experiment, you will experiment with the popular FreeRTOS RTOS. The project in the zip file on fht ecourse website assumes that you have both the TI TivaWare distribution and the FreeRTOS distribution unzipped with a root in the same directory as the directory that contains the project file. I.e.,

C:/MyLabDir/FreeRTOS
C:/MyLabDir/TI
C:/MyLabDir/Lab7demo

Where the project file is:

C:/MyLabDir/Lab7demo/freertos_demo.uvproj

Because the include path in the project includes:

..\TI
..\TI\examples/boards\ek-tm4c123gxl
..\FreeRTOS\Source\include
..\FreeRTOS\Source\portable\RVDS\ARM_CM4F

If you use a different directory structure, you will have to modify the project file to find the FreeRTOS and TI files.

Experiment:

1. First, download the project file from the course web site. Download the latest version of FreeRTOS and the latest version of TivaWare and place them in the directory structure described above.
   Build the freertos_demo and confirm that it works.

2. We will integrate Valvano's board support pack for the TI Educational BoosterPack MKII. This file includes several functions to initialize the GPIO, A2D, SPI, I2C subsystems to communicate with the various sensors and actuators on the MKII board. Build Tm4C123Valvanotware/Lab1 and confirm that your MKII board is working.

3. First, we will develop a simple electronic level app. Copy BSP.c and BSP.h into your project directory. The 3-axis accelerometers on the MKII board provide analog outputs for the acceleration on each of three orthogonal axes. When the board is not being accelerated, these sensors can be used to determine an Earth nadir vector. We will use this vector to determine when the board is "level" and to measure the angle between successive orientations of the board. We will develop a rudimentary UI (user interface) for our new system using the push buttons and the Adafruit ST7735 LCD.

   User Interface:
   - On the MKII Switch1 (Mode_Button) will determine the mode of operation and with each press the mode will toggle between BUBBLE and TEXT.
     - The system will support two modes: TEXT and BUBBLE. In TEXT mode, the current raw X, Y, Z acceleration data as well as the angle between the hearer pins on the board and the nadir vector will be reported as asci text. In BUBBLE mode the angle between the Nadir vector and the pin headers will be reported graphically with a small circle centered on the display when the difference = 0.
   - Switch2 (Lock_Button) on the MKII will "lock" the display.
     - If the Lock_Button is pressed the system will enter LOCK mode and the display will hold the most recent sample (in either TEXT or BUBBLE modes)
     - If the system is in the LOCK mode and the Lock_Button is pressed again, the system will remain in LOCK mode and will report (in ASCII) the difference between the Nadir vectors recorded at the time of the two most recent presses of the Lock_Button.

   Architecture:

   - A task will be dedicated to monitoring the buttons,
   - A task will be dedicated to monitor the accelerometers
   - A task will be dedicated to updating the display.

   Demonstrate the operation of everything except the BUBBLE mode display for a signature.

4. Demonstrate your BUBBLE Mode display.

5. Add any other enhancements you think are appropriate. Perhaps the speaker col chirp on button presses to provide feedback to the user. Perhaps the LEDs on the MKII could change color or intensity based on the mode and maybe even he quality of the level when not in the LOCK mode. Demonstrate your final design for a signature.

NAME:_____

1-3: Instructor Signature:_____ Date/Time:_____

4-5: Instructor Signature:_____ Date/Time:_____