

Lab 4: FreeRTOS

Goals

1. Transition to the FreeRTOS embedded real-time operating system
2. Emphasizing the ability to extract information from the datasheet to correctly setup registers

Objective of the lab

Porting over some of the tasks from previous labs to the FreeRTOS operating system.

What you need for the lab

1. The EK-TM4C123 Launchpad (<http://www.ti.com/tool/EK-TM4C123GXL>)
2. TM4C123 data sheet (<https://canvas.uw.edu/courses/1205180/files/folder/Ek-TM4C123GXL?preview=49165887>)
3. IAR workbench or other IDE
4. Getting Started with RTOS (<https://canvas.uw.edu/courses/1205180/files/folder/labs?preview=49823635>)
5. FreeRTOS support archive (https://www.freertos.org/FreeRTOS_Support_Forum_Archive/March_2016/freertos_Getting_Started_with_Tiva_C_series_5236d16bj.html)
6. FreeRTOS jump start (<https://jj09.net/freertos-jump-start/>)
7. LEDs (<https://learn.adafruit.com/all-about-leds/the-led-datasheet>)
8. LCD (EB-LM4F120-L35)(http://www.kentecdisplay.com/uploads/soft/Products_spec/EB-LM4F120-L35_UserGuide_04.pdf)

Scheduler

We will transition to the FreeRTOS real-time operating system for this lab. It comes with its own timing interrupt, please remove the timer initialization and ISR from previous labs to get started. We would suggest you start with some simple tasks with the scheduler, then start porting the previous project over to FreeRTOS. We also suggest you read the files provided in the above section to get started.

Tasks 1: Timing

Using FreeRTOS Sleep(n) call to control timing.

TASK 2: Display

Display any relevant information to the LCD. Please be creative in the point. Furthermore, the touch feature is also required in this task.

TASK 3: LED

- Creating three tasks that blink red, green, and yellow LEDs at different periods.
- Implementing the following state machine. Note here, you did the same FSM in lab 2.
 - i) Please refer to the lab1 regarding the hardware setup.
 - ii) Replace the physical push buttons with two virtual buttons that displayed on the LCD. Let's call the start/stop button as virtual button 1, and the button for the passenger as virtual button 2 for the following sections. You can display the button either vertically or horizontally. Please remember you need to display the relevant information to the LCD at the same time.
 - iii) When a virtual button is pressed, the system will respond only if the user holds down the button (on the LCD) at least 2 seconds.
 - iv) If the user presses the virtual button 1 (start/stop button, hold for 2 seconds), but not the virtual button 2 (passenger button), the system will start with the stop stage (where the red LED is on, and other LEDs are off). After 5 seconds, the system will move to go stage. Then wait for another 5 seconds to change from go stage to stop stage. In other words, the go and stop stage will last for 5 seconds and switch to each other.
 - v) If the user presses the virtual button 2 (passenger button, hold for 2 seconds) to indicate a passenger tries to across the street, the system will stop the current stage and move the warn stage immediately. The warn stage will last 5 seconds, and move to go stage.

Deliverables:

1. A lab report along
2. Demonstrate the output to your TA
3. Upload the report and source files for both sections to the canvas. One submission is expected per team.