**Re-Entrant Functions**

The execution of a function called within a task can be pre-empted before the function returns to allow another task of equal or higher priority to run. The next task, and possibly several other tasks, may call the same function. If the function contains any static variables, this will lead to problems. Functions must be re-entrant (that is, stateless) to be able to be called by several tasks with the possibility of the function being pre-empted by one task and called in another task. If the function must maintain state information, these static variables must be declared within each task calling the function and passed as arguments to the function.  
  
If you use functions from the standard C library, you must consider whether or not they are re-entrant. A good example of a function that is not re-entrant is the strtok() function – it retains the current state of the tokenizing process between calls.

**Priority Inversion & Priority Inheritance**

We will go over pages 250-252 in *Mastering the FreeRTOS Kernel 161204*.

**Deadlock aka “Deadly Embrace”**

We will go over pages 252-253 in *Mastering the FreeRTOS Kernel 161204*.

**Final Project Definition**

ST Micro B-L475E-IOT01A development board peripherals:

* STM32L475 MCU – 80MHz Cortex-M4 core, FPU, MPU, 1MB flash, 128KB RAM
* Serial flash – 64Mb (8MB)
* WiFi – 802.11 b/g/n
* Bluetooth Low Energy (BLE) – V4.1
* Sub-GHz RF – 915MHz ISM band
* Near Field Communications (NFC) Tag – ISO/IEC 14443 Type A and NFC Forum Type 4
* Digital microphones (2)
* Temperature and relative humidity sensor
* Barometric pressure sensor
* 3D magnetometer
* 3D accelerometer + 3D gyroscope
* ToF (time-of-flight) sensor
* Green LED
* Pushbutton
* Potential feature via an Arduino-compatible shield – Graphical LCD

**Class Notes**