**Queues**

Just like tasks and semaphores, if you want to use a queue, you need to first create it. When you create a queue, you have to state the size of each object and the maximum number of objects the queue must contain.

A queue is referred to in subsequent operations by the handle that is returned when it is created.

The subsequent fundamental operations with a queue are to put objects into it and take objects out of it.

Below are the three primary queue API function declarations from the *AWS FreeRTOS Reference Manual* on pages 162, 199, and 186, respectively.

**#include “FreeRTOS.h”  
#include “queue.h”**

**QueueHandle\_t xQueueCreate(UBaseType\_t uxQueueLength, UBaseType\_t uxItemSize);**

**#include “FreeRTOS.h”  
#include “queue.h”**

**BaseType\_t xQueueSendToBack(QueueHandle\_t xQueue, const void \* pvItemToQueue,   
TickType\_t xTicksToWait);**

**#include “FreeRTOS.h”**

**#include “queue.h”**

**BaseType\_t xQueueReceive(QueueHandle\_t xQueue, void \*pvBuffer,  
TickType\_t TicksToWait);**

**Assignment 5 Update**

You have been given an amended Assignment 4 in which only the original Part 1 is required to be implemented this week. Assignment 5 will involve creating an output task that is fed sensor data via a queue.

**Class Notes**