Create a new FreeRTOS-based project called *Assign05*. Copy the contents of the *src* and *inc* directories from *Session11.zip* into the project, refresh the project, clean the project, then build the project. Run and test the project using Tera Term or CoolTerm to see the 4-digit distance measurement on a single line updating every 100 msec.

* Use the xQueueCreate() (page 162) function to create a queue called ProxQueue that will hold 10 distance measurements of type uint16\_t.
* Instead of using printf() in the proximity measuring task to output the distance, use the xQueueSendToBack() (page 199) function to add the distance to the queue. Do not block for any time when adding an item to the queue (i.e., if the queue is full, the distance will be thrown away).
* Create a new task called OutputTask that initializes the UART and uses the xQueueReceive() (page 186) function to take a distance measurement from the queue and print it out in the same format at Assignment 4. If the queue is empty, block indefinitely.
* Once you have this working, record the size of the text segment after building the project.
* Use the eprintf() function instead of the printf() function. Once you have this working, record the size of the text segment after building the project.
* Clean up the code to remove all unnecessary functions (e.g., Serial\_Scanf()).

**This assignment must be emailed to tlupfer@sandiego.edu by midnight on Sunday, September 27th.**

**Make sure your name appears in the module comments at the top of any C and header files you create or modify.**

**Create a zipfile containing the *inc* and *src* directories of the project (and no other directories).**

**You should attach a single file named:  
  
 *lastname05.zip*  
  
In other words, my file would be named *lupfer05.zip.***

**In your email, provide the size of the text segment when printf() was used and when eprintf() was used.**