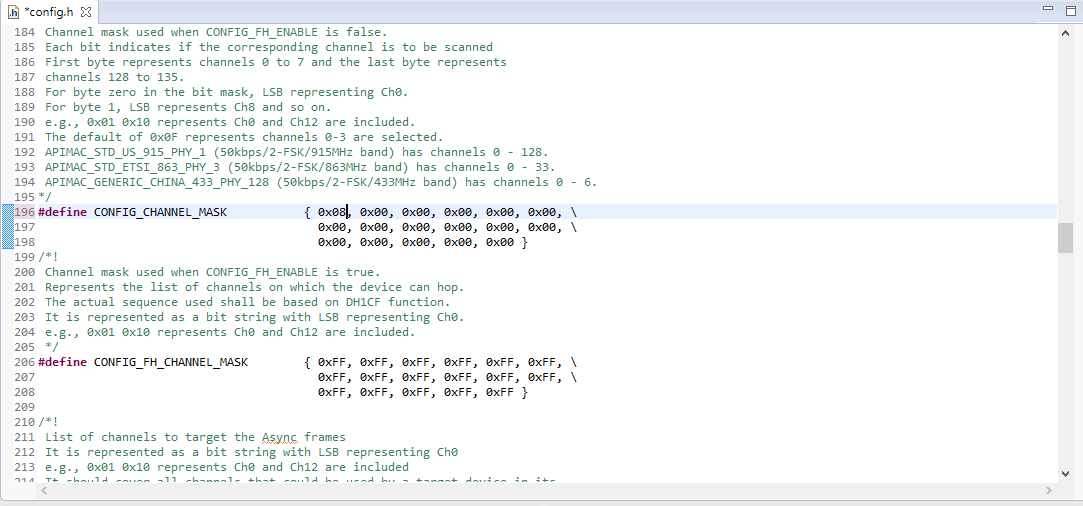
**Date Submitted: 12/4/2018**

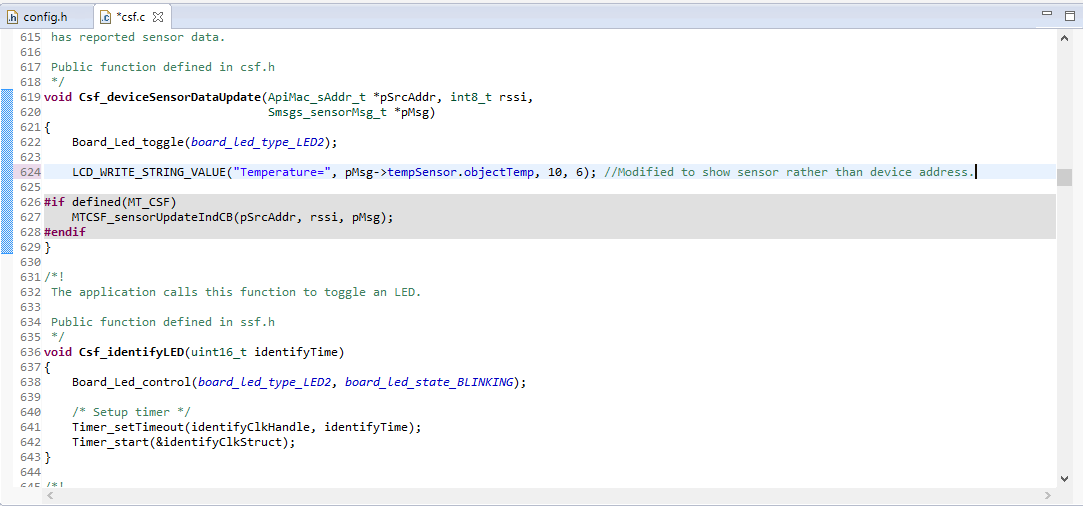
**Assignment Youtube Playlist:** **https://www.youtube.com/playlist?list=PL4oTyvRrubXdJRKEUOwTKdQsdkmt-vUxo**

# Task 01: Building and loading the collector example

**Youtube Link:** **https://youtu.be/1HbnL3HWp60**

For Task 1 we import the collector example to CCS. We then open the config.h file and adjust our channel mask . We also modify our code to write our temperature to LCD, although an LCD was not connected to our device for this lab.





**-----------------------------------------------------------------------------------**

# Task 02: Building and loading the sensor example

**Youtube Link:** **https://youtu.be/xxStDpsIfz0**

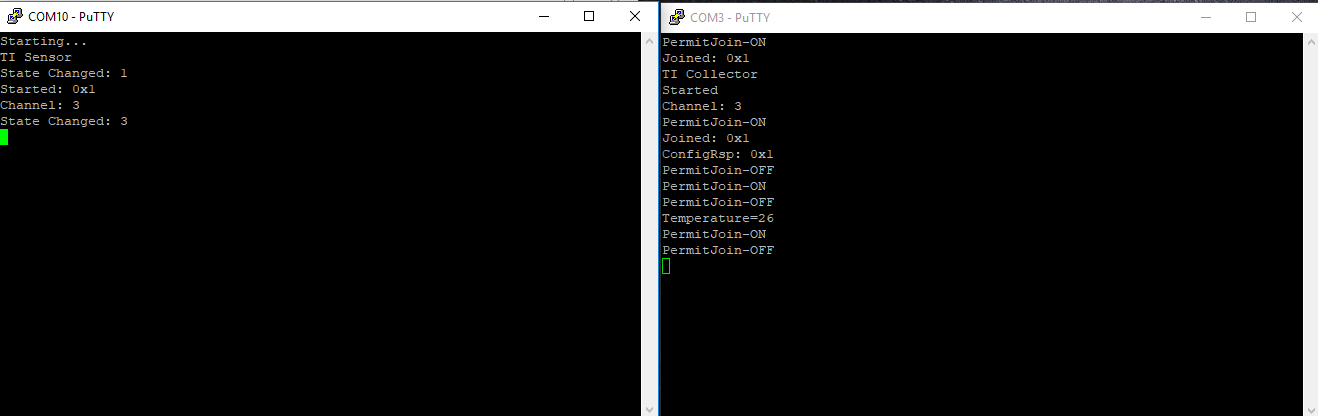
For Task 2 we import the sensor example and modify it in the same fashion as our collector. The code adjustments look nearly identical to above.

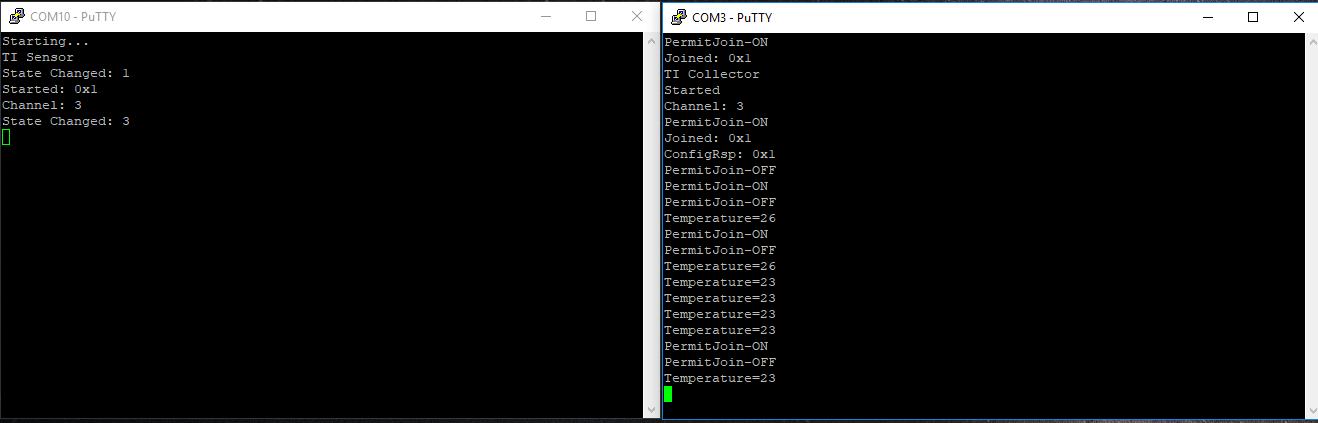
**-----------------------------------------------------------------------------------**

# Task 03: Using the Collector and Sensor

**Youtube Link:** **https://youtu.be/1TZ18QsCOg8**

For Task 3 disconnect both CC1350 Launchpads from CCS and connect them to one computer, displaying a UART connection for both. We display the data from both devices in UART as they connect then begin to transfer data.



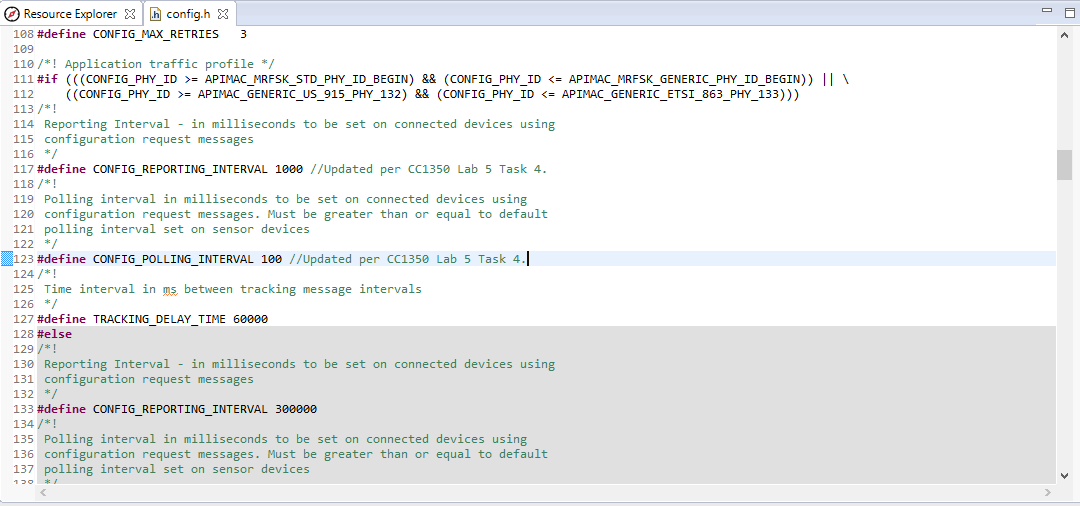


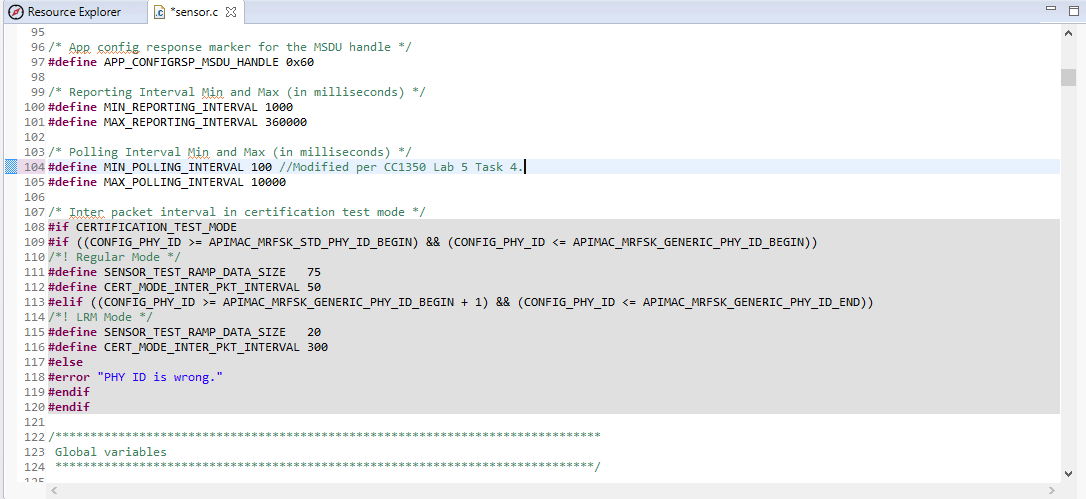
**-----------------------------------------------------------------------------------**

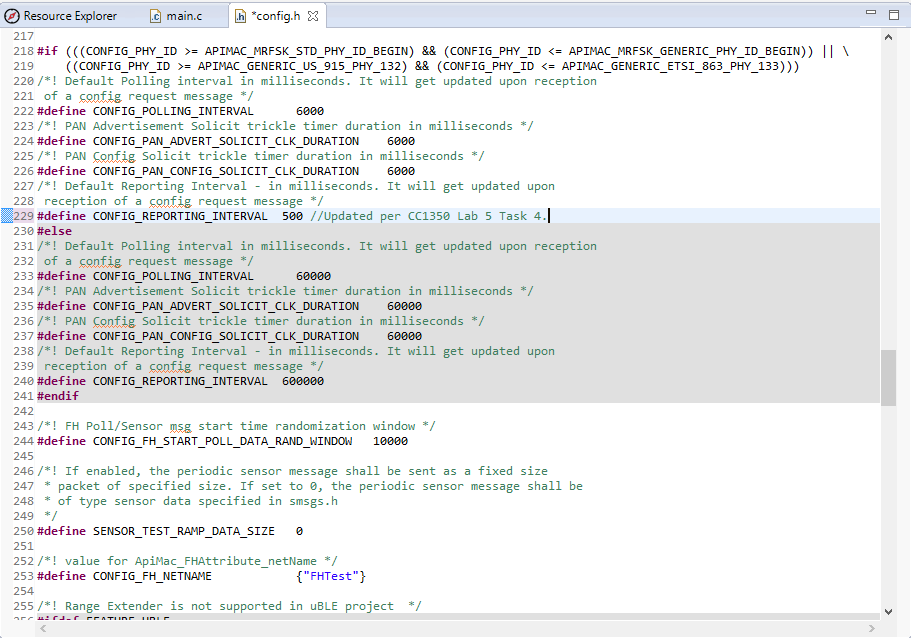
# Task 04

**Youtube Link:** **https://youtu.be/9QU15SIQQi4**

For Task 4 we update the sensor’s reporting rate to generate results more quickly. However, we found that a rate of 100ms was not recognized by our devices, so we used a polling rate of 500ms. Adjusted code is shown below.







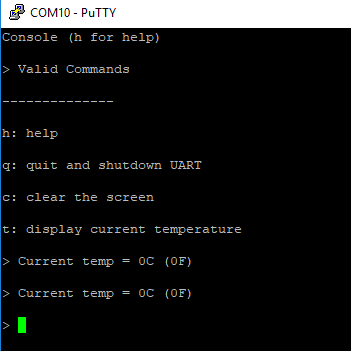
**-----------------------------------------------------------------------------------**

Lab 5 Part 2

# Task 01: Building and loading the portable app

**Youtube Link:** **https://youtu.be/QTEJfjpIXKw**

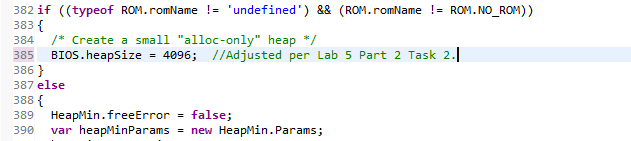
In this lab we are going to import the “portable” code to our sensor LaunchPad. This will allow us to work directly with our sensor device rather than have it act as a passive gatherer. For task 1, we simply import the project and run. The portable app uses data from an external device that we do not have, so the values it displays are invalid but are displayed in the manner we are expecting.

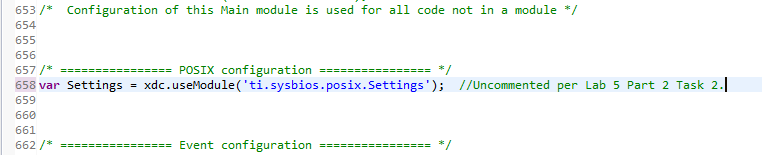


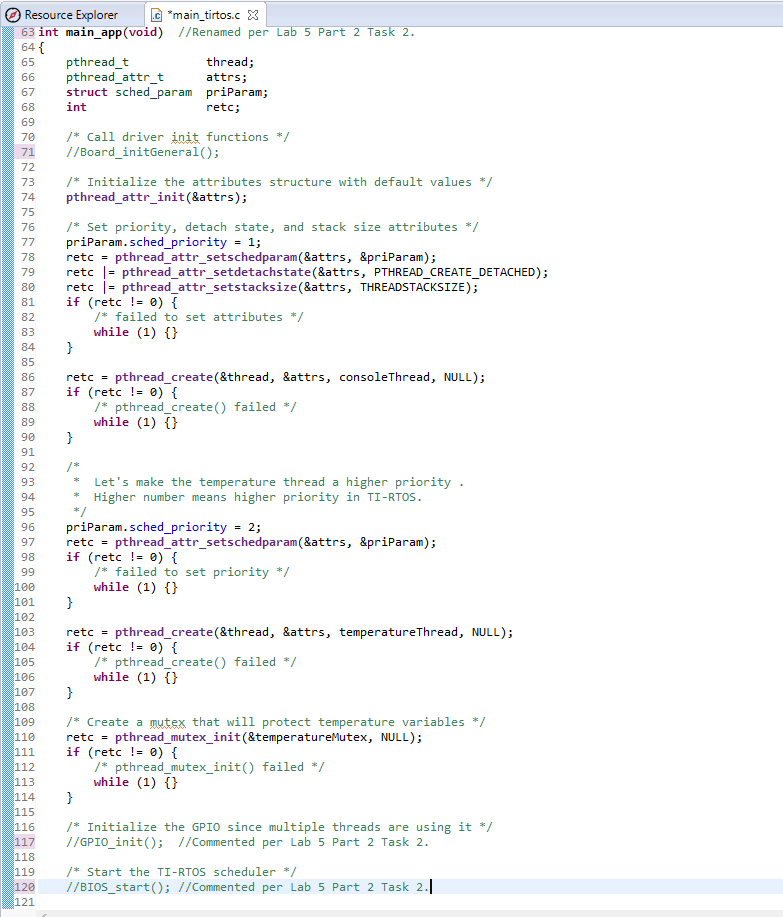
# Task 02: Combine the portable app with the TI 15.4-Stack app

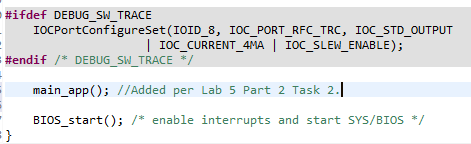
**Youtube Link:** **https://youtu.be/aRz5YDH498A**

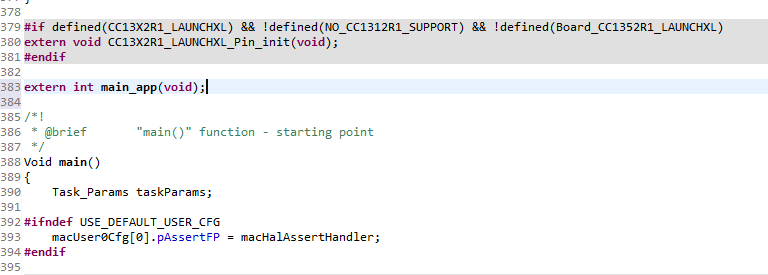
For Task 2 we combine the functionality of the portable app with the previous functionality of Lab 5 Part 1. We update the heap size to 4096 as well as add POSIX support, then copy over the console.c, main\_tirtos.c, and temperature.c files from the portable project. We also comment out various now-duplicated function calls in main\_tirtos.c. Our end result is that we are essentially running both the portable app and our remote projects in parallel.

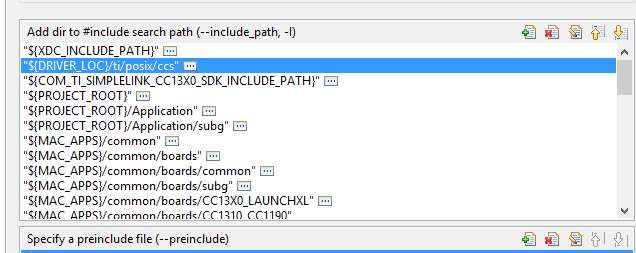


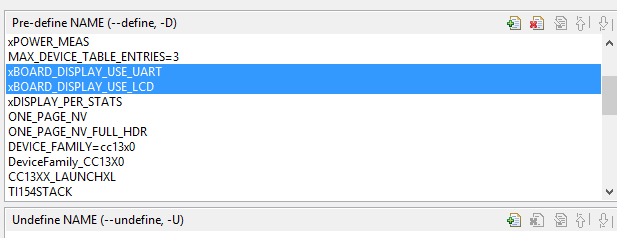


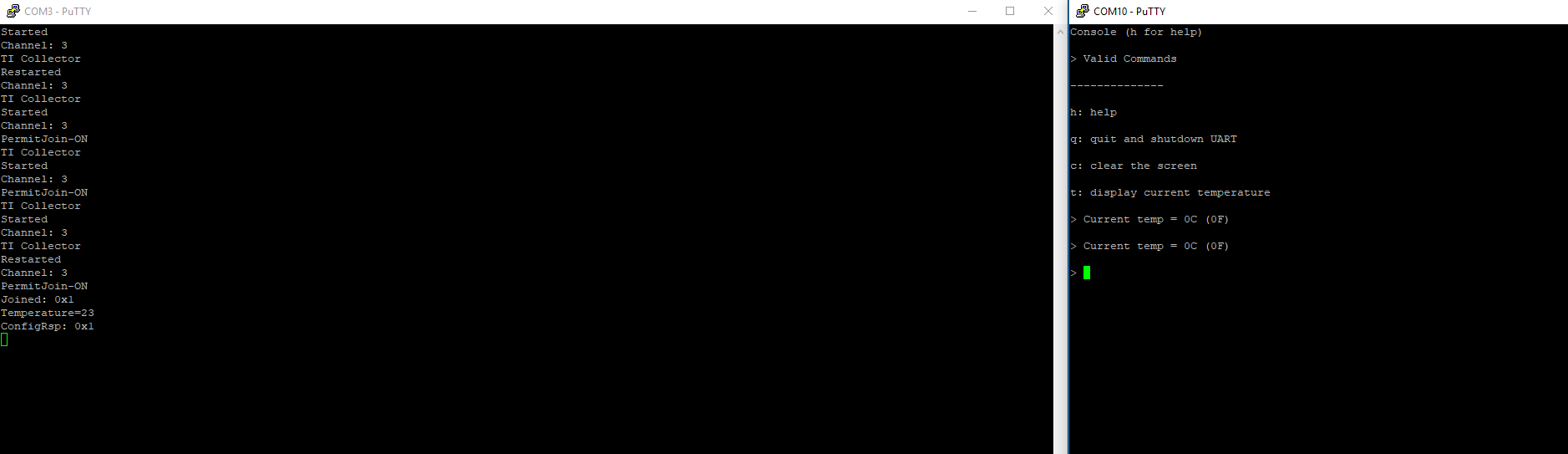












Tasks 3-5 have been skipped as it refers to the temperature data source that we do not have.