EE4930 Advanced Embedded Systems

Section 011, Winter 2019/20

Draven Schilling

2-10-20

Lab 6 – RTOS System Phase 1

# Objectives

The objective for this lab was to experiment with the MSP432 RTOS (sys/bios). We were to build an RTOS system which uses tasks, events, a hardware interrupt, and a software interrupt to make a system which uses an ADC to take readings from a potentiometer which then converts the reading to a temperature value between 50-90F, and displays the results on the LCD. In the process we will learn how to setup and use common RTOS processes and features.

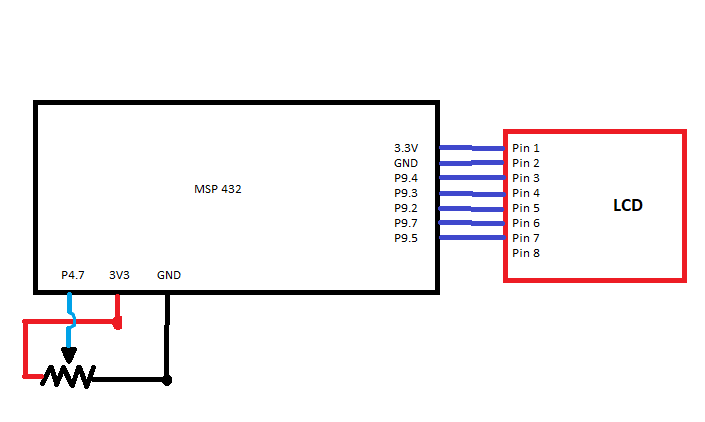
# Description

For this lab, I started by importing the “event” example from CCS to use as a base. From there, I verified operation of the example and then proceeded to strip out the features I didn’t need (Mailbox and Semaphore). Then I read up on the SWI/HWI from the RTOS user manual and CCS documentation and implemented the structure for both of them. Now equipped with the base components to complete the lab, I imported my previous ADC initialization code, wrote each task using the example structure, and added code for the SWI/HWI/clock methods. During debugging, I quickly found that I also had to add additional link and compile libraries that were present in previous labs but not in this example. A little bit of debugging later and the RTOS was performing according to the lab specifications.

# Conclusions

In conclusion, this lab was very interesting. I learned a lot about how the structures of an RTOS works. The main difficulties I encountered were with solving linker errors and a few with the HWI setup. Initially I couldn’t leave the HWI and I ended up finding out that we had to read the ADC value (even if you’re not doing anything with it) in order to clear the interrupt; additionally, I had some problems setting up the HWI as the user guide suggests using hooks, but the CCS documentation uses an implementation similar to all the other structures. Eventually after a bit of debugging I was able to get the lab operational without any major setbacks. I think this lab serves as a good intro to the basic RTOS features because it doesn’t use too many outside peripherals but forces us to explicitly experiment with a lot of the core operating system features.

# Attachments



**Source Code:**