Anthony Bartman

EE 4930

Laboratory 6: RTOS System – Phase 1

2/11/2021

Objectives:

The objectives of Lab 6 pertain to learning how to use a potentiometer, ADC and the LCD screen while using an RTOS system to display a temperature value. The student will learn how to use hardware and software interrupts, a clock, 2 events and 2 tasks within a RTOS C file. The student will have to choose which tasks will have different priority levels and how to set that hierarchy. The LCD will be updated based on a potentiometer change which will then flow through the RTOS system to update the LCD with the current temperature value. Another objective is to determine how to set up the flow of the RTOS system with task priority levels, HWI and SWI setup, and Clock, Task and Event setups. The RTOS system flow, task priority assignment, and the chosen clock period will be discussed within the description of the lab report.

Description:

Lab 6 has students create an RTOS system that reads a potentiometer that simulates a temperature value that will then be displayed on the LCD. I initialized the HWI, SWI, a Clock, 2 tasks and 2 Events by reading TI RTOS Kernel User Guide and documentation for the HWI and SWI libraries. After initializing the RTOS system flow from the Lab 6 writeup, I began to determine how to create the flow of this lab's RTOS system. The RTOS system flow is as follows: The clock will trigger every 0.25 seconds to check if there is a new potentiometer reading. If there is a potentiometer reading change, then the hardware interrupt will trigger because it is linked to the ADC interrupt. Once the HWI is triggered, it will use a custom event to trigger the software interrupt that will convert the ADC value into temperature range from 50 to 90 degrees Fahrenheit. In terms of the task priority assignment my thought process was similar to the RTOS flow I just described. The priority hierarchy of tasks follows from HWI to SWI to Input Task to Update LCD. Updating LCD is the least priority task because that happens after the HWI and SWI have completed their data reading and conversion. HWI interrupt is prioritized first because without the HWI, this lab would not function properly since we potentially wouldn't be reading that data from the potentiometer. I choose a 0.25s clock period because it was a fast enough period to update the display as seamlessly as possible but still not consuming as much power.

Conclusion:

The results of the lab can be shown in the lab demo on box. I had some trouble understanding how the RTOS system works and where to learn more about how the HWI and SWI functions. It was very difficult to read through the documentation of the HWI and SWI libraries as well as using the basic event.c program from the resource explorer. I learned how important it is to read and understand documentation to the fullest and how to write a simple RTOS system. Overall, RTOS systems are a new way to think about how to efficiently code an embedded system. It is also crucial to continue learning how to read documentation to produce effective, yet efficient code for an embedded system.

Attachments:

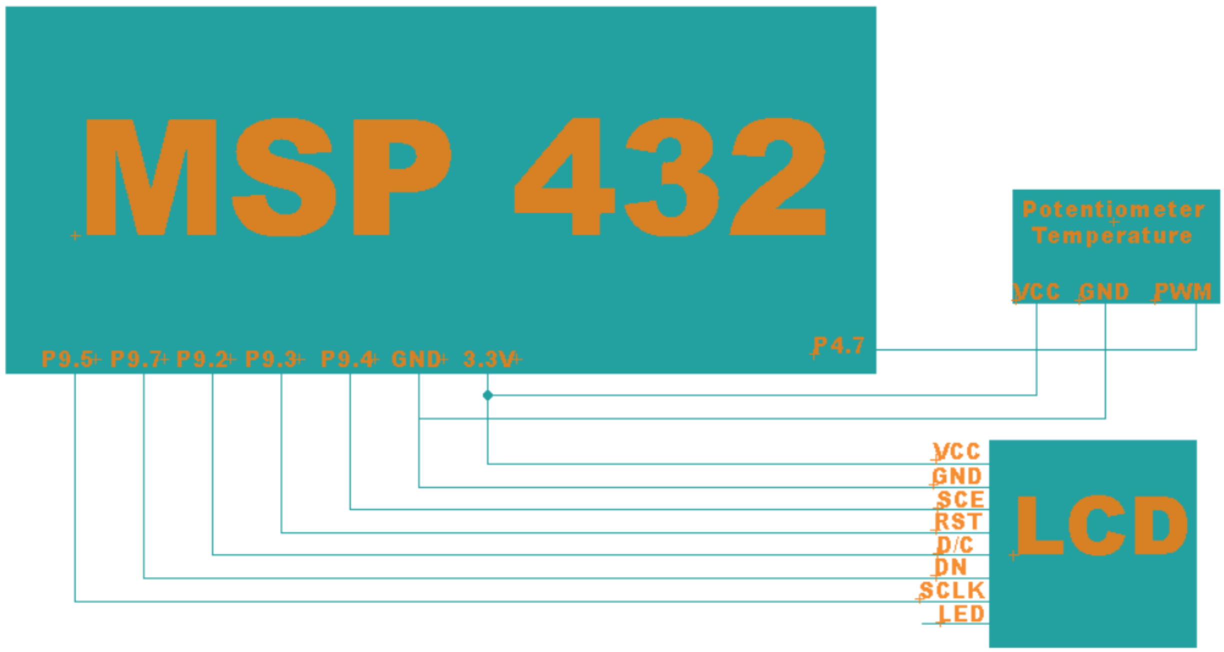

Figure 1: This figure shows the schematic of the lab wired up physically.