ECE2049: Embedded Systems in Engineering Design
Lab Exercise #3 – A Term 2022

Making a Time and Temperature Display

In this laboratory you will use MSP430 and several of its peripherals to implement a system measures temperature using ADC12's internal temperature sensor once per second (measured with Timer A2) and displays time along with the temperature to the LCD. The main purpose of the assignment is to gain some experience using MSP430's analog-to-digital converter. However, you will have to make use of some elements from previous labs like timers, decimal-to-ASCII conversion and text display of results.

Assignment:

Digital time and temperature displays are common sight. In this lab you will make a mini-time and temperature display using the MSP430F5529. First you will implement a UTC style clock with one second resolution. The temperature sensor which you will use is the MSP430's internal temperature sensor which is connected to ADC12_A analog input channel 10 (ADC12INCH_10). You will also need to implement the Scroll Wheel and use it to modify the date and time settings. You may refer to the schematic for pins settings of Scroll Wheel.

Pre-lab: This pre-lab is to be completed before your lab session. You should have your original pre-lab paper signed and dated by a member of the course staff at the start of lab. They should also *sign & date* the pre-lab box on your sign-off sheet.

1) *READ THE ENTIRE LAB ASSIGNMENT!!*

2) Write a displayTime(long unsigned int inTime) function which takes a copy of your global time count as its input argument. Your function should convert the time in seconds that was passed into Month, Day, Hour, Minutes and Seconds. It should then create ascii array(s) and display date and time to the LCD in the format specified Step 6 below. *Explain why it is important to pass a copy of the time into the function rather than just using the global variable*.

3) Write a displayTemp(float inAvgTempC) function which take a copy of the floating point temperature in C and displays temperature in both C and F to the LCD as specified in Step 6 below. *This function is just a display function. It can convert C to F but it should not do the averaging of past temperature readings described below*.

Requirements:

In implementing the time and temperature display you are required to complete each of the following tasks. You do not have to complete the tasks in the order listed. Be sure to answer all questions fully in your report.

1) No welcome screen this time just right to business. As soon as the program is started the systems should start displaying date (month and day), time and temperature.
2) The system will take temperature measurements using the MSP430's internal temperature sensor (see User's Guide Ch 28, class notes)once per second. Timer A2 should be configured to measure 1 second intervals. Implement a UTC-style clock count capable of holding 1 year of time (sec). What data type did you use to store your time count? Initialize you UTC count to your birth date (minus the year, of course).
3) Configure the ADC12 to make single channel, single conversion readings from the internal temperature sensors. Select your reference voltages to give the best resolution for the problem. ***Justify your choices in your report.*** What will the resolution of your readings be in volts and in ^{o}C ?
4) Form a moving average of the last 36 seconds of temperature readings. That is, each second form the average of the current reading and the previous 35 readings. Convert your averaged reading to degrees C and degrees F. You can achieve this by storing the last 30 seconds worth of averaged data in an array: tempC[]. You should use "circular indexing" (i.e., something like index = time count modulo 30) to index your array rather than shifting the arrays around. It's much more efficient. Explain why.
5) Your LCD should continually display the following items in a loop. Displaying each for 3 seconds. *The temperature displayed should be the average temperature* from Part (4).

Date => MMM DD (i.e., OCT 10)

Time => HH:MM:SS (i.e. 10:02:49)

Temp (C) => ddd.f C (i.e. 24.7C)

Temp (F) => ddd.f F (i.e. 72.3 C)

6) Implement the scroll wheel. See lecture notes. Same design approach with Digital Current meter.
7) When the left Launchpad button is pressed once you should enter Edit Mode. The months should be selected and change by turning the scroll wheel. If the left button is pressed again, move over to the day. Again, turning the scroll wheel will change the value. Successive presses of button will move the user over to the next date/time field (i.e. hours, minutes, seconds). If after editing the seconds field, the left button is pressed again cycle back and select the month again. *How will you map the output of the scroll wheel to months, to days, etc.? Explain in your report?*

8) Pressing the right Launchpad button at any time accepts the changes made and exits Edit Mode to Display Mode. You should convert the new date entered to be your current global time in seconds. If you were now to press the left button the you would start back at editing the month.

9) Write a high-quality lab report answering all the questions in the requirements section!

10) ***BONUS (5 pts):*** Use ADC12 interrupts instead of busy bit polling to signal completion of ADC conversions.

11) ***BONUS (5 pts)***: When editing, either underline, blink or invert the colors of the display of to indicate the field currently selected.

SAVE OFTEN!
All lab partners who participated equally should save their own copies of ALL code to their R drive

To submit your code for grading, you will need to create a zip file of your CCS project so that the TAs can build it. You can also use this method to create a complete backup copy of your project (perhaps to archive or send to your partner) for later. To do this:

1. Right click on your project and select "Rename..."

2. If you are submitting your project, enter a name in the following format: *ece2049c20_lab3_username1_username2*, where username1 and username2 are the usernames of you and your partner. (NOTE: Failure to follow this step will result in points deducted from your lab grade!)

3. Click OK and wait for CCS to rename your project.

4. Right click on your project again and select "Export..." then select "Archive file" from the list and click Next.

5. In the next window, you should see the project you want to export selected in the left pane and all of the files in your project selected in the right pane. You should not need to change which files are selected.

6. Click the "Browse" button, find a location to save the archive (like your M drive) and type in a file name using the EXACT SAME NAME used in Step (2).

7. Click "Finish". CCS should now create a zip file in the directory you specified.

8. Go to the Assignments page on the class ***Canvas*** website. Click Lab **3 Code Submit**. Attach the archive file of your project that you just created and hit the Submit button. Only one submission per team.

ECE2049 A-2021 Lab 3
Sign-off Sheet

Bonus Sign-off: **Friday 09/30/2022** ***Report due***: **Tuesday 10/05/2022**

Student 1: _____

Student 2: _____

Board #: _____

YOU ARE RESPONSIBLE FOR *ALL* THE REQUIREMENTS LISTED IN THE REQUIREMENTS SECTION OF THIS ASSIGNMENT!

PRE-LAB (students graded individually)	5	Student 1: Student 2:
Timer A2 measuring seconds	5	
ADC12 making single channel, single measurements for ADC12_A temp sensor once per second	10	
Implement Scroll Wheel	10	
Proper conversion and display of date & time (month and day – FEB 2 and hr min sec = HH.MM.SS)	10	
Proper conversion and display temperature in degrees C and F	5	
Edit mode using the scroll wheel	20	
Use ADC12_A interrupts rather than busy bit polling to get results	***5***	
BONUS: Indicating field being edited by underline, blink or color inversion or such	***5***	
Answer to TA Questions at Sign-off	5	Student 1 Student 2
Report (answering ***all*** questions from the requirements section)	30	
Total points	100	

***** Both Students MUST be present at Sign-off for all parts!!***

5% Early Bonus: _____