ECE2049: Embedded Systems in Engineering Design
Lab Exercise #3 – A Term 2018

Making a Time and Temperature Display

In this laboratory you will use the MSP430 and several of its peripherals to implement a system that measures temperature (using ADC12's internal temperature sensor) and time (using Timer A2) then 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 and homework like timers, decimal-to-ASCII conversion and text display of numeric results.

Assignment:

Digital time and temperature displays are a common sight (e.g. banks, shopping malls). In this lab you will make a mini time and temperature display using the MSP430. First, you will implement a UTC-style clock (see Homework 3) with one second resolution. The temperature sensor 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 (HW#4).

Pre-lab (*Due in Lab Weds 9/26/18*): 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()` function which takes a copy of your global time count as its input argument. Your function should convert the time in seconds (which was passed in) to 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()` 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. While it may convert C to F, 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 once per second. Timer A2 should be configured to measure 1 second intervals. Implement a UTC-style clock count capable of holding 1 yr of time (sec). What data type did you use to store your time count? Initialize your UTC count to the last 7 digits of your student ID number. Manually convert that number to date and time to confirm that your code is correct. Include your validation (for both partners' ID numbers) in your report.

3) The system will measure temperature using the MSP430's internal temperature sensor (see User's Guide Ch 28 and class notes).

4) Configure the ADC12 to make single channel, single conversion readings from the internal temperature sensors. Select your reference voltages to achieve the *best* resolution while being able to measure temperature in the range -40 to 100°C. ***Justify your choice in your report.*** What will the resolution of your readings be in volts and in C°/bit? Convert your ADC readings to degrees C.

5) Store the last 256 readings into 2 arrays: times[] and tempC[]. You should use "circular indexing" with your array index defined to be of type char rather than shifting the arrays around. It's much more efficient. Explain why circular indexing is efficient in your report and why you can use a char as the array index with out any additional index math (other than incrementing the array index).

6) Your LCD should continually display the following items holding the display for ***4 seconds*** between updates. Month should be the abbreviated name not a number. ***YOU MAY NOT USE sprintf()!!*** See Intro slides for why.

   Date => MMM-DD (i.e. SEP-22)

   Time => HH:MM:SS (i.e. 16:02:49)

   Temp (C & F) => ddd.f C / ddd.f F (i.e. 24.7C / 72.3F)

IMPORTANT: The readings from the temperature sensor are rather noisy. They jump around a lot from reading to reading. Therefore, *the temperature displayed should be the average of the last 4 temperatures measured. Show TA/tutor exactly how you did this!*

7) Implement the scroll wheel (see HW#4). Select your reference voltages and configure your code such that they cover *the entire range of voltages* than can be generated by the scroll wheel. Explain your choices for the scroll wheel reference voltages. ***Leave V_{ref+} for the temperature sensor as it was.*** Explain where and how your set the reference voltages for the two input channels.

8) When the button S1 on the lab board is pressed the buzzer should sound but the pitch of the buzzer should be controlled by the scroll wheel. When the wheel is turned in one direction the pitch of the buzzer should go up. When the wheel is turned in the other direction then the pitch of the buzzer should go down. Pressing button S2 should turn the buzzer off . How will you map the output of the scroll wheel to buzzer pitch? Explain in your report?

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. Explain how ADC12 interrupts are configured and what your ISR does in your report.

11)***BONUS (5 pts):*** Have your clock toll the hour (like a church tower) using the buzzer. Explain completely how you achieve this in your report.

WARNING and a HINT on number conversion: Here's a fun(?) issue that students can encounter when converting the time and date, which is related to how the CCS MSP430 compiler handles constant data types. If you type a line like this:

```
        long x = 24 * 3600;
```

You'll get a warning reading "integer operation is out of range" because the compiler treats the constants as `int`'s (i.e. as 16-bit integers) before multiplying them, so the multiplication will overflow before the result is assigned. This may seem silly, but it kind of makes sense because our MSP430 is a 16-bit machine. It's default word size is 16-bits and operations on longs are emulated in software. To make the MSP430 actually multiply the numbers as longs, you can either use integer suffixes to tell the compiler that the constants are longs, like this:

```
        long x = 24L * 3600L;
```

... or cast either of the operands:

```
        long x = ((long)24) * 3600;
```

Writing a High Quality Lab Report:

Your lab should be written in a professional style. It should be an electronically prepared technical document like what you would submit to a co-worker or your boss. The report should include the following:

1) *Cover sheet* = An appropriate, professional-looking cover sheet with the title of the assignment and the names and ECE box numbers of each student.

2) ***YOUR SIGN-OFF SHEET!!!*** Most of the points for the lab are listed on the sign-off sheet. If you do not submit it then you will not receive the points.

3) *Introduction* = 1-3 paragraphs (1/2 page tops) succinctly stating the objectives of the lab and giving an overview of what you accomplished.

4) *Discussion and Results* = As many pages as it takes (without padding!). In this section you should thoroughly discuss what you did in each part of the lab. You should describe the approach you took to solving any problems. Include pseudo code descriptions and/or flow charts for any code developed. Results should also be thoroughly discussed. Any measurements should be tabulated, questions (especially those in yellow!) should be answered completely, figures should not be hand drawn, snippets of code may be included where useful but a full code listing is not required or desired. Instead, code should be submitted electronically (see below).

5) *Summary and Conclusion* = 1-3 paragraphs. Wrap-up and summarize what you accomplished in the lab. This should be a “bookend” to the introduction.

6) *Appendices* = Include any pre-labs, relevant raw data sheets or reference material, etc.

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, 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: *ece2049A18_lab3_username1_username2*, where username1 and username2 are the user names 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-2018 Lab 3
Sign-off Sheet

Bonus Sign-off: **9/28/2018** ***Report due***: **10/2/2018**

Student 1: ______________________________ **ECE mailbox:** __________

Student 2: ______________________________ **ECE mailbox:** __________

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	10	
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 – APR 2 and hr min sec = HH:MM:SS). ***NO sprintf()!!***	10	
Proper conversion and display temperature in degrees C and F. ***NO sprintf()!!***	5	
Scroll wheel controlling buzzer	15	
Use ADC12_A interrupts rather than busy bit polling	***5***	
Make your clock toll the hour with buzzer	***5***	
Answer to TA Questions at Sign-off (students are graded individually)	5	Student 1 Student 2
Report (*answering **all** questions from the requirements section*)	30	
Total points	100	

5% Early Bonus: ________________ ***Board #***:________

BOTH PARTNERS MUST BE PRESENT FOR ALL SIGN-OFFs!!!

ECE2049 Lab Report Grading Rubric

Format -- 5 pts

Did you follow instructions given above as to the format of your report?

Is you code (submitted on-line) formatted, properly commented, etc.?

What is expected for the following parts was already described above.

Introduction – 3 pts

Discussion – 15 pts

Conclusion – 3 pts

Appendices – 1 pt

Professionalism – 3 pts

Spelling, grammar, neatness, presentation, etc.