Requirements Specification for EELE465 Lab Project 5: Analog-to-Digital temperature readings

Academic Dishonesty Clause: It is expected that you, along with your partner, write this code uniquely for this project during this semester. It is acceptable to use code YOU have previously written as examples and guides. It is not acceptable to copy/paste code written by other individuals nor by you during previous semesters or for other classes.

Lab project goal: Read an analog temperature sensor and display the value on the LCD.

The master will sample an analog temperature sensor's output and produce a moving average with a window of n-samples. The user will be allowed to select the resolution, n, using the keypad. The LCD will be used to display the user's choice, the temperature in Kelvin, and the temperature in Celsius. The LED lightbar will continue to function.

Outcomes:

After this lab you should be able to:

- Sample an analog signal using the MSP's ADC module.
- Calculate and display moving averages.

Background:

Remember, the primary point of this class to for you to develop problem-solving and debugging skills, the second objective is to improve communication skills, and the third objective is to develop advanced programming/embedded system skills.

So far, you have been given a significant amount of guidance on setting up and/or developing your code. In this project you will need to apply those skills with less guidance. Although there is not an instruction set within this document on "getting started," it is strongly suggested that you follow good programming/design/engineering skills and *plan* your process before getting started, develop the visuals that will aid you in your development process (and in 'I need help' discussions with the instructors), and write sample test pieces before trying to put it all together.

That said, the following requirement section will be broken into requirements (qualitative, listed by numbers) and specifications (quantitative, listed by letters) to aid in prioritizing your work. You should focus on ALL the requirements before worrying about achieving any/all the specifications.

Acknowledgements: Randy Larimer

Requirements for lab project completion:
At the end of this project, your embedded system must do the following:

1. Upon a restart of the slave, or the first restart of the master, your LCD should display:

Row \ Col	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	E	n	t	e	r		n	:								
4	T		=					°	K						°	C

   a. The # button should refresh the LCD and return to this default-display at any point during operation; this should occur within 0.5s of the button press.

2. Sample the LM19 temperature sensor.
   a. Update and display the average every 3samples/second (or once per 333ms) ±5% (Obviously you cannot do this for the <u>first</u> 2 samples required to fill the averaging window. After the first 2 samples are collected, you should update the value every 333ms)

3. Calculate the moving average of n ADC values, where the user may choose the value of n using the keypad.
   a. Do not, at any point – including start up – display a temperature reading until n-valid samples were collected (ie: if n = 5, you might have the following samples after the first 5 readings:

      T(n) = 23.0°, T(n-1) = 23.1°, T(n-2) = 24.1°, T(n-3) = 22.9°, and T(n-4) = 23.8°

      The average would be 23.38° → 23.4°

      The average should not be displayed until you have collected 5 valid readings. After the first average has been displayed, the average should update every 333ms after that.

4. Display the ADC value on the LCD
   a. Display the value in Deg-Kelvin ±1°
   b. Display the ADC value in °Celsius ±0.1°

   Ideally, your final display will look like this at room temperature:

Row \ Col	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	E	n	t	e	r		n	:								
4	T		=		2	9	6	°	K		2	3	.	5	°	C

<u>Project Stretch Goal</u> (worth +5%, 0.5pts):
1. While maintaining the sampling resolution of 3samples/second, only update the LCD once per 2seconds.
2. Provide *evidence* that the temperature display is correct (calibrate code if necessary).
3. Buttons A-D should still work, but convert D to have two different static patterns, one if the temperature reading is greater than a set point, a different if the temperature is lower.

Acknowledgements: Randy Larimer

Assessment for project completion:

Start the demo with the code running, the microcontroller in the "lock" state, and locked-indicator from Project03 visible.

a. Introduce your project (both partners should be involved), including:

    1. High-level description of what the project does,
    2. Concise discussion of your circuit diagram,
    3. Concise discussion of your high-level flowchart.

Present the aspects of the project you were primarily responsible for. If you shared the workload with a partner, state that while presenting each step.

b. Discussion and Demo of Requirement 1 and 4, including:
    1. Show the LCD upon start up
    2. Show that N, temp-in-Celsius, and temp-n-Kelvin are displayed.
    3. Provide *evidence* that the code refreshes the LCD within 0.5s after pressing #. (Note: "evidence that the code works," cannot be the code itself.)

c. Discussion and Demo of Requirement 2, including:
    1. Demonstrate that the code samples the ADC 3x/sec

d. Discussion and Demo of Requirement 3, including:
    1. Demonstrate that the user can change the size of the sample window
    2. Provide evidence that the first sample displayed is a valid average of that sample size.

e. If you attempted the stretch goals:
    1. Demonstrate that you can still update the average every 1/3 s, but the LCD only updates every 2 s.
    2. Provide evidence that the LCD-displayed temperature is calibrated correctly.
    3. If you are able to use the LEDs to display temperature related to a set point, demonstrate that….note: you could use 2 different buttons with 2 different set points to show the functionality works without having to reprogram to change the set point nor having to adjust the ambient temperature in the room. Alternatively, figure out how to adjust the temperature of the sensor quick enough to be convenient for demonstration purposes – reprogramming mid-demo is not allowed though.
    4. Discuss anything else you may have added to the project.

Acknowledgements: Randy Larimer