1) From mycourses download this pdf, dxp_Lab9_a1.asm.
2) The **objective** of this lab is to get acquainted with the use of ADC10.

3) For this lab un-install the capacitive touch sensor, but don't install any of the two jumpers back.

4) **Part 1: Build, run and understand the dxp_Lab9_a1.asm code.**

5) Create a new assembly only project in your workspace named fmlxxxx_Lab9_a1. Create a copy of the assembly code dxp_Lab9_a1.asm in the project folder, and rename it fmlxxxx_Lab9_a1.asm. Build the project, enter debug mode, and run the program.

6) Select to see the memory content starting at 0x200.

7) Make sure that line 30 is commented out and on line 51 INCH_10 is being used. These selections will allow you to acquire an analog input value corresponding to the output of the internal temperature sensor. How to convert this value to a temperature is covered in the ADC10 chapter of the UG. Look in 22.2.8.

8) Set breakpoints on the Mainloop line (34), Start a conversion line (55), and on the NOP line (66).

9) Now, run the program to the first breakpoint. Up to this point the code is making settings that don't change during the course of Mainloop execution.

10) Run to the next breakpoint. The Mainloop is calling a routine named Acquire.

11) At the beginning of this, ADC10 is being configured to acquire a single channel / single value from the internal temperature sensor, i.e. channel 10.

12) Read the code and comments. Corroborate the selections with their explanations in the ADC10 chapter of the UG.

13) When you next click RUN, the ADC10 start a conversion. At the end of it, it will trigger an interrupt. Once the interrupt is triggered, the program arrives at the NOP in the ADC10 ISR.

14) Here, the interrupt flag is reset, the acquired value is copied from ADC10MEM into a register or memory location of your choice, and the ADC10 SW FLAG is set.

15) The ADC10 SW FLAG is a means to tell the Acquire routine that a conversion has been completed, once the program returns from the ADC10 ISR. There are certainly other ways one can implement this, but this is one that shows you the use of a SW FLAG. Notice the Acquire routine is basically sitting in a loop that waits for the SW

flag to be set. It will not continue until this happens.

16) Run five times through the code and record manually the converted values. Calculate their average and convert it to a temperature value. Include all these in your report.

17) **<u>Part 2:</u> Change fmlxxxx_Lab9_a1.asm to acquire from channel 0.**

18) Stay in the same project as in part 1.

19) Uncomment line 30 and change to INCH_0 in line 51.

20) Using a test bench DC voltage supply, apply a voltage between 0.5 and 1.0 V at input A0. This is P1.0. Before you connect this signal to the board pin, measure it with the provided voltmeter. Any voltage above 3 V and below -1 V will permanently damage the uC pin circuitry!!! Use small alligator clips to connect to both P1.0 and GND. Don't forget the latter. Both circuits have to reference their voltage against the same reference potential, which is GrouND! When you make the connections, both the Launch Pad and DC voltage supply should be turned off.

21) Power on the Launch Pad first, and then the DC power supply.

22) Run through the program as before 5 times. Manually record the converted values, calculate their average, and compare it against your settings measured with the given voltmeter. Include these results in your report.

23) **<u>Part 3:</u> Change fmlxxxx_Lab9_a1.asm TO fmlxxxx_Lab9_a2.asm OR fmlxxxx_Lab9_c1.c to acquire a sinusoidal signal from channel 0.**

24) This part you can implement in assembly OR in C, i.e. it is your choice.

25) Modify the code provided so that you acquire 32 samples in a data memory array, one sample every 10 ms.

26) The acquisition period should be set by using TimerA. Use any mode you want to trigger an interrupt after 10 ms. Make sure general interrupts are enabled before your timer starts!

27) After this is triggered, you execute the code provided for this lab. Instead of always saving to register 5, you will save the result in an array in the data memory, which you index every loop.

28) You can use a second SW FLAG to signal your Mainloop when you've been in TimerA ISR. The ISR vectors are listed at the end of the msp430g2553.h file.

29) Using a test bench signal generator, apply a sinusoidal signal with the following characteristics: frequency = 10 Hz, DC offset = 0.5 V, and Peak to Peak Amplitude = 1 V.

30) You can use a SW defined variable to keep track of how many samples are left to acquire.

31) After the acquisition is done, go to the memory window, switch the memory window to display decimal, and copy the values into an excel spreadsheet. When the data is pasted into excel, it will paste as text in a single column. To separate the data into columns, use the "*DataText to Columns*" menu option and use the fixed width mode. Note that the array is stored in the RAM, which starts at address 0x0200. Plot the "restored" signal, and make sure the "x-axis" has the sample TIME, not sample number (increment in 10ms increments, starting at 0).

32) Compare, analyze and explain how the sampling frequency affects the conversion. Include the table with values and the plot in your report.

33) **<u>Part 4:</u> Witness ALIASING – Due to undersampling!**

34) Change the acquisition period in part 3 to ~105 ms.

35) Re-acquire as in part 3 the same input signal. Make sure the "x-axis" has the sample TIME, not sample number (increment in 105ms increments, starting at 0).

36) How does the plotted look like? What is its frequency? Why do you think it looks as it looks? What lesson can you draw from this with respect to the use of digital oscilloscopes, or for the use of a digital acquisition system for that matter?

37) Include your acquisition table, plot, and detailed answers to all above questions in your report.

38) Make sure you write the report and upload it along with your project archives on mycourses. As time permits, demo the intermediate steps to the TA.

39) **<u>Grading</u>**:

a. Part 1 = 10 points

b. Part 2 = 10 points

c. Part 3 = 10 points

d. Part 4 = 10 points.