EE 444

Embedded Systems Design

Lab 3: MSP430: ADC12\_A

*2/23/2017*

**Chic J O’Dell**

Does your solution work the way it’s supposed to work? □YES □NO1

1If your answer is NO, please explain in your report.

Instructor/TA comments and grading

# Objective and Background

use ADC12\_A to sample various internal and external signals, and  
use the board’s LCD to display the results.

# Equipment

* CrossStudio for MSP430
* TI MSP-EXP430F5438 Experimenter Board
* Oscilloscope

# Procedure

Using SMCLK set to 16 MHz (in the same manner as in the past three labs) as an input to the ADC12\_A. The ADC is then used to measure the internal temperature sensor located on the MSP silicon dye. To accomplish this the control registers for the ADC\_12 were set as follows.

* REFCTL0: Enable the reference modual and make that reference available externally to the ADC
* ADCCTL0: Turn on the ADC, enable multiple sample conversion, set sample hold time to be sufficient for the temp sensors sample time (greater than 30 us).
* ADCCTL1: Set sample hold time to pulse mode, select SMCLK as an input, set the conversion sequence mode to “sequence of channels”.
* ADC12MCTLx: selecting the input for the ADC measurement as the internal temp sensor, set the reference voltage to Vref+ and Vref-, and for ADC12MCTL7 setting the end of sequence bit.
* ADC12IE: in order to use interrupt driven code.

All of the above mentioned registers are setup during main and then waits in LPM0 mode for a button push IR. Note all of the button setup mirrors what had been written in previously in lab2. When button #1 is pushed the interrupt service vector is called and triggers the code seen in “P2IV\_P2IFG6” case. This case statement triggers the start of an ADC measurement using “ADC12CTL0 ^= ADC12ENC + ADC12SC;” as deigned through the control register. As the ADC12SC (ADC sample conversion start bit) is called the ADC takes data and triggers an interrupt upon the ADCMEM7 being written.

In the ADC interrupt service routine all of the MEM registers are summed and then averaged into a single variable and then converted using the provided MSP430 calibration data given in ADC12\_15V\_30 and ADC12\_15V\_85 implemented by following the formulas laid out in the MSP430 users guide documentation. The final calibrated averaged sample is saved into a global variable to be displayed with the LCD screen.

The setup for the LCD is carried out in main directly after the ACD12 control registers are set up. After the LCD has been setup in main upon a button #2 press interrupt driving the LCD screen is run to show the stored temperature value. All of the necessary code used to accomplish this is higher level function calls given in the labs attachment libraries. This being said getting the LCD to display correct entailed a fair bit of trial and error.

# Results

This lab ended up working as hoped, though I learned a lot more about CrossWorks then in the previous labs. Crossworks versions in the lab do not by default support floating point variables in printf/scanf . So trying to get the sampled ADC temp value on the display was a head ache until that setting was found. Another Crossworks Easter egg I found when my LCD brightness would only run in debug mode on the lab computers. That is to say all functionality of the code worked in release vs debug mode except the LCD backlight command, which was only responsive in debug mode. After giving up trying to find the bug in my code I pushed the project to git and left, pulling the code onto my personal laptop later to find the same code worked perfectly well. This is due to differing release build settings in CrossWorks.

# 5. Conclusion

This lab showcased the ease of using the on board LCD screen of the MSP430 dev board (provided you set your CrossWorks release build options correctly)and served as a good introduction to the operation and operating modes of the ADC12.

# 7. Attachments

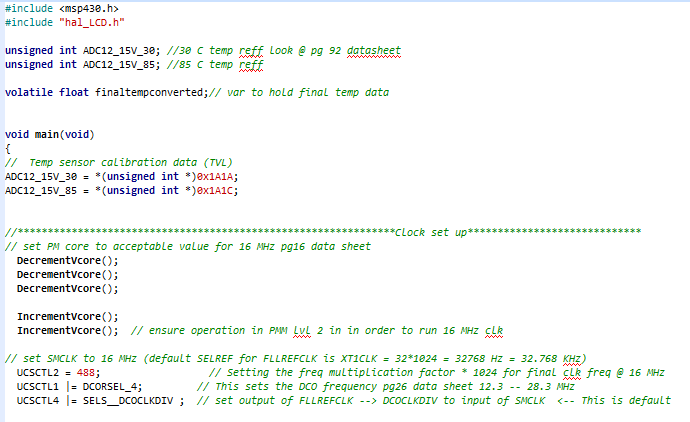


Figure : SMCLK set to 16 MHz.

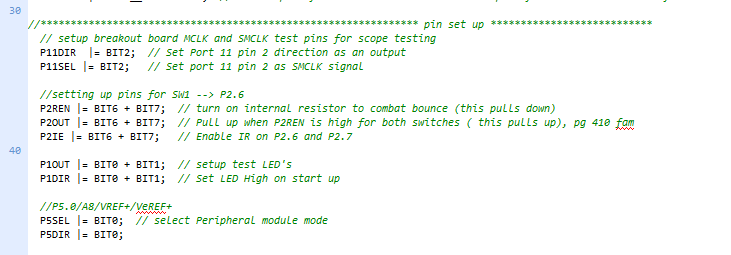


Figure 2: Pin setup for SMCLK breakout, Port 2 IR, SW LED toggle, Vreff Breakout

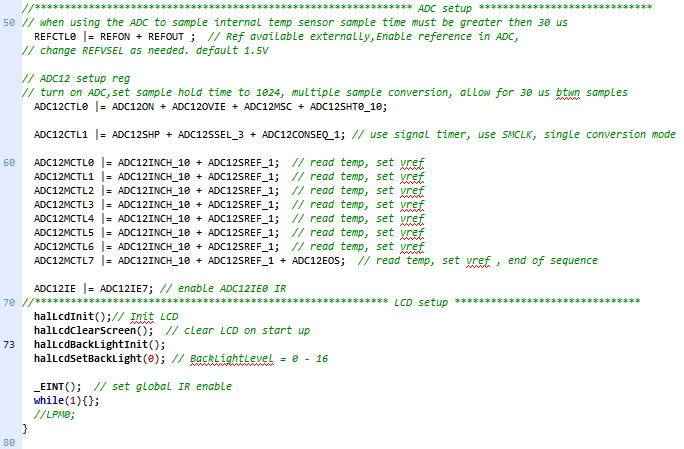


Figure 3: Setup ADC\_12 and enter main LPM0

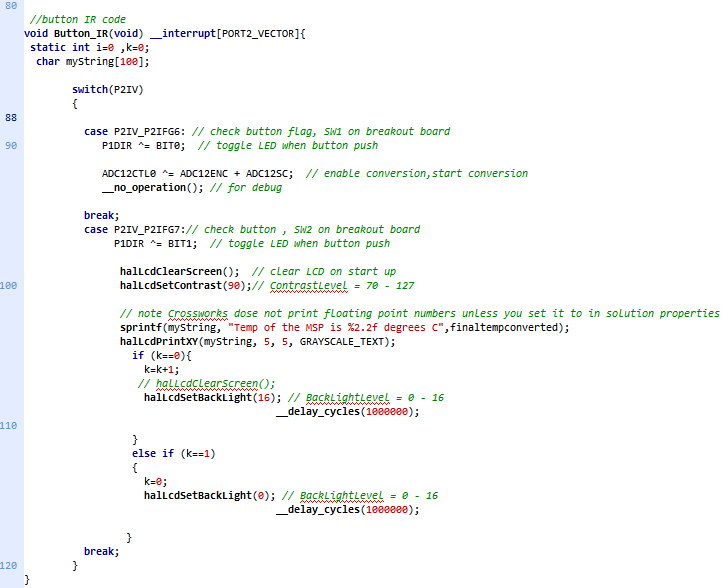


Figure 4: Port 2 interrupt service routine.

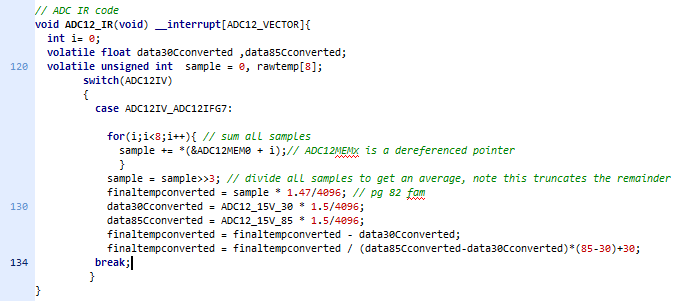


Figure 5: ADC\_12 interrupt.