EE 445S Real Time Digital Signal Processing Laboratory

Prof. Brian L. Evans

Lab 2 Handout

In this lab you will be generating a sine wave using two outputting methods

 a) **Polling:** You will be using inbuilt library functions for generation of sine wave with Polling. (This program has already been developed for you. Please follow the instructions in the handout to run your project)

 b) **Interrupt**: You will be using difference equations for generation of sine wave with interrupts, as outlined in the presentation slides. You need to write out this program yourself in the appropriate C file.

**Generation of sine wave using polling:**

Please refer to the handout of Lab 1 for instructions on how to create a project. Please go to the same workspace where you created your previous project in Lab 1 after opening Code Composer Studio.

1. Similar to the previous lab, unzip the contents of Lab\_2\_Polling\_Interrupt and create a new project in Code Composer called “sine\_wave\_polling\_interrupt”. Follow steps mentioned in previous handout to create a project till step 7. In step 8, browse to C:\Lab\_445S\Lab\_2\Lab\_2\_Polling\_Interruptand add all files apart from the “bsl”, “gel” and “tests” folder.
2. Follow the subsequent steps of the previous handout to run your project.
3. Go to the function isrAudio() in “isr.c”. In this case, the interrupt has not been used and notice the infinite while loop for generating the sine wave. Also, notice, how the flag XRDY is polled continuously in the function “WriteSample” in order to see if it is ready to transmit a sample.

**Generation of sine wave using interrupts:**

After successfully viewing the sine wave using polling, please make the following changes to the program in order to implement interrupts:

1. Remove the infinite while loop in the function isrAudio() in the file “isr.c”.
2. Go to the function WriteSample. Replace the piece of code

**while**(!CHKBIT(MCASP->SRCTL11, XRDY)){}

MCASP->XBUF11 = dataOut32;

With

**if**(CHKBIT(MCASP->SRCTL11, XRDY)){

 MCASP->XBUF11 = dataOut32;}

1. Open the .tcf file. Click on the “+” sign in order to expand “Scheduling”. Next Click on the “+” sign in order to expand “HWI”. Click on HWI\_INT5. The screen should look like:

Right click on HWI\_INT5. Go to Properties. In the box beside “Interrupt Selection Number”, type 61. (61 correspond to McASP Transmit/Receive Event). For “function”, type “\_isrAudio” (Be careful to use the underscore. This is a way by which an assembly language code can understand that isrAudio is a C program.). Next, click on the “Dispatcher” tab and check the box beside “Use Dispatcher”. Save the .tcf file.

1. In the function isrAudio(), delete the code that implements the sine wave using library function calls and write out your code for implementing sine wave generation using difference equations. Please look at the questions of the lab report and make sure you do everything asked in the questions.

**Generation of sine wave using interrupts and look-up table:**

In the same project which was used for sine wave generation using difference equation and interrupts, write you own program, for generating a sine wave using look-up table (Please refer to the presentation slides about look-up tables.). The sine wave generated has a frequency of 1 kHz, and sampling rate is 8 kHz. In a table, save the values of the sine wave over 64 periods, resulting in an array of 64\*8 = 512 entries. (In this case, one discrete-time period contains one continuous-time period.) Output the values stored in the array using interrupts. Please don’t forget to scale the output by an appropriate scaling factor. Repeat for a sine wave of frequency 2 kHz, using an appropriate array size.

**Lab Report: (Due at the beginning of the lab section next week. One report is to be submitted per group)**

The general format for lab reports is:

* Objective (one sentence)
* Procedure (brief, just enough so the discussion makes sense)
* Discussion of Results (1-2 pages usually)
* Problems Encountered / Other Observations

The point of the **discussion** section is to relate the observations/procedure of the lab to theoretical concepts discussed in the lectures.

For Lab 2, also answer the following questions:

1. Explain (mathematically) what happened when 6 kHz & 7 kHz sine waves were generated for a sample rate of 8 kHz. (HINT: assume the samples are calculated with the COS() function instead of SIN() )
2. Compare and contrast the methods for sending the samples of the sinewave to the codec (polling, interrupt)
3. Scaling factor (the variable which multiplies the generated sine wave value):
	* Why is the scaling factor necessary?
	* What would happen if the scaling factor were set to 50,000 instead of 30,000?
	* What would happen if it was left out?

**HINT:** If you actually try this out in the lab, only take note when either sine wave appears distorted or both sine waves appear distorted. A sine wave at 2 kHz may not appear distorted. Try other frequencies such as 1.7 kHz, 2.4 kHz and 3.3 kHz.

1. Real Time:
	* How many cycles does a single call to **sin()** take?
	* (**HINT**: You can use C profiling tools to get the answer. Use breakpoints. In the debug perspective, go to Target -> Clock -> Enable in order to enable the clock)
	* Calculate the number of cycles that occur between samples. Assume the DSP runs at 375MHz. (And feel free to comment on these two results)