ELEC 5260/6260 Problems Set #7 - REVISED
Due Monday, 3/25/2013 – PART 1
Due Friday, 3/29/2013 – PART 2

The goals of this exercise are to study:
(1) digital-to-analog conversion,
(2) serial device interfaces (I^2C in this case),
(3) real-time operations, coordinated by programmable timers.

You are to design a musical “tone generator” for the DISCOVERY board, with button-selectable frequencies. These tones should be heard via the audio mini jack on the board (most standard “ear buds”, headphones, or similar listening device can be plugged into this jack.)

1. Tones are to be produced by generating sinusoidal signals at the output of the microcontroller’s built-in digital-to-analog converter (DAC). The output of the DAC is connected on the board (via GPIO pin PA4) to analog inputs AIN1A/AIN1B of the Cirrus CS43L22 Audio DAC. The CS43L22 drives the headphone (HP) jack on the board.
2. The tone frequencies are to be those of the eight notes of an increasing C-major scale in music (C-D-E-F-G-A-B-C).
3. The sampling frequency is to be 20KHz. This information plus the tone frequencies will be used to generate the data samples to be converted by the DAC.
4. Initially, no tone should be generated when the board is reset.
5. The first press of the user button should result in generation of the low C tone.
6. Subsequent presses of the button should change the tone to the next higher note in the scale. (Only one change per button press.) After the high C tone, continue again from the low C tone.
7. If the button is pressed twice within a one-second interval, generation of tones should stop. Generation of tones should begin again on the next button press, with the last tone that was generated.
8. LEDs 4-3-5 (in that order) are to display the current tone number (binary numbers 000-111 correspond to tones from low C to high C.)
9. LED 6 is to blink at a rate that is proportional to the tone being generated. Select blinking rates that can be distinguished visually.

	LED3		LED3 (orange) = I/O port pin PD13
			LED4 (green) = I/O port pin PD12
LED4		LED5	LED5 (red) = I/O port pin PD14
			LED6 (blue) = I/O port pin PD15
	LED6		User button (blue) = I/O port pin PA0

(Continue on next page for assignment Parts 1 & 2)

PART 1 – Generation of data samples.

Design and program an algorithm (ex. a series approximation) to generate 16-bit data samples at the required sampling rate for each selected tone frequency. Ideally, this should be one algorithm, with the tone frequency and sample rate as parameters. For testing purposes, you may use a sample frequency of 1KHz, which corresponds to the default "System Tick Timer" frequency, or you may change the initialization of the System Tick Timer to generate interrupts at a frequency of 20KHz.

Test the algorithm by writing each sample to a global variable and displaying that variable in the debugger Logic Analyzer window. (This should show the sinusoidal signal at the tone frequency.) The document "Cortex-M4 training for STM32F4-Discovery board using ARM Keil MDK toolkit" (especially sections 15 and 19) describes the use of Keil Logic Analyzer to graphically display variable values and signals. This document is on the course web site.

Submit your source program and printouts of the Logic Analyzer window, showing two different tones corresponding to the musical notes E and A.

PART 2 – Generation of tones at the headphones.

1. Use one of the microcontroller's general-purpose programmable timers, **other than the system tick timer**, to produce the required 20KHz data sampling frequency (generate new data samples every 1/20KHz = 0.05msec.)
2. Convert data samples to analog form with the microcontroller's on-chip digital to analog converter (DAC). *(Not the DAC on the CS43L22.)*
3. Initialize the CS43L22 to "pass through" to the headphones the analog signal from the output of the microcontroller DAC.
4. Implement the rest of the button press and LED display requirements.

Testing and submitting the final program:

1. As before, the program can be tested in RAM on the board or in flash memory. The final version should be programmed into the flash memory of the board, so that the program can be demonstrated without being connected to the Keil debugger. Bring your programmed board to my office for a demonstration.

2. Write a short report, briefly describing how you addressed/implemented each of the project requirements. Attach a printout of your source program to this report. Submit the report both in hard copy and electronic form (***email)***.