Lab 4 Digital Scope and Spectrum Analyzer (changes in red)

Goals
- Interface a microphone and record sounds,
- Design and implement an analog HPF, LPF and digital FIR filters,
- Build a spectrum analyzer,
- Write a real-time application that displays in both the time domain and the frequency domain.

Review
- Operation of the microphone from lecture notes 8, and electret datasheets
- Data sheets on the Texas Instruments LM4041CILPR, TLC2272, TLC2274, or OPA2350.
- Data sheet http://users.ece.utexas.edu/~valvano/Datasheets/gp2y0a21yk.pdf.

Starter files
- Lab 3, your real time operating system
- FIR design templates **FIRdesign51.xls**~~, **FIRdesign64.xls**~~
- LCD graphics driver, new `ST7735.c` posted on the web site (includes graphics)
- **sqrt.c** integer square root function

Background

1) Real-time data acquisition (option 1)

This experiment will use a microphone, an analog circuit, and an ADC converter to sample audio. The electret microphone is a low-cost low-fidelity transducer appropriate for voice recording. An analog band-pass filter will be used to improve signal to noise ratio and to prevent aliasing. You will use the12-bit ADC converter on the microcontroller to convert the analog signal into digital form. You can trigger it either with the hardware timer or the software. The following FIR section designs a digital filter based on a sampling rate of 12.8 kHz, but you are free to **select the range of sampling rates (f_s)** over which your digital scope/spectrum analyzer will operate, as long as the sampling is performed in real time. A background periodic thread implements the real time sampling, and data is passed to the foreground using a FIFO queue. According to the Nyquist theorem, if the signals of interest range up to f_{max}, we need to sample faster than $2*f_{max}$ to reliably capture the information in digital form. Conversely, we need to add an analog low pass filter to remove frequencies above ½ f_s to prevent aliasing. You will be using TI's **FilterPro** design tool to design your LPF.

1) Real-time data acquisition (option 2)

As an alternative (you do not need to do both options) an IR distance sensor, an analog circuit, and an ADC converter to measure distance versus time. The Sharp GP2Y0A21YK is an obsolete and noisy transducer that has an output voltage depending on distance for a range of 10 to 80cm. An analog low pass filter will be used to improve signal to noise ratio and to prevent aliasing. You will use the12-bit ADC converter on the microcontroller to convert the analog signal into digital form. You can trigger it either with the hardware timer or the software. The following FIR section designs a digital filter based on a sampling rate of 12.8 kHz, but you are free to **select the range of sampling rates (f_s)** over which your digital scope/spectrum analyzer will operate, as long as the sampling is performed in real time. A background periodic thread implements the real time sampling, and data is passed to the foreground using a FIFO queue. According to the Nyquist theorem, if the signals of interest range up to f_{max}, we need to sample faster than $2*f_{max}$ to reliably capture the information in digital form. Conversely, we need to add an analog low pass filter to remove frequencies above ½ f_s to prevent aliasing. You may use TI's **FilterPro** design tool or http://users.ece.utexas.edu/~valvano/EE345M/lpf.xls to design your LPF. You may select any sampling rate you wish.

2) FIR digital filter

You will process the real-time data in the foreground by implementing a FIR digital filter. After you have chosen the sampling rate (e.g., 12.8 kHz) you next will **choose a FIR filter length** (e.g., N=51). I recommend you use http://users.ece.utexas.edu/~valvano/EE345M/FIRdesign51.xls and do not use the 64-point version of the Excel sheet. The ratio f_s/N (e.g., 12.8 kHz/51 = 251 Hz) will determine the frequency resolution of the FIR filter design. Let H(z) be the desired filter gain transfer function. Table 1 gives an example desired frequency response. The magnitude of H(k) is selected to implement the **desired gain versus frequency response**. In order to preserve the shape of the audio signals, we will implement linear phase. For frequencies above ½ fs, we make H(k) be the complex conjugate of the N-k term. This will guarantee that the inverse DFT of H(z) will yield real results. The desired filter response, plotted as red dots in Figure 1. The actual FIR filter gain is plotted in blue.

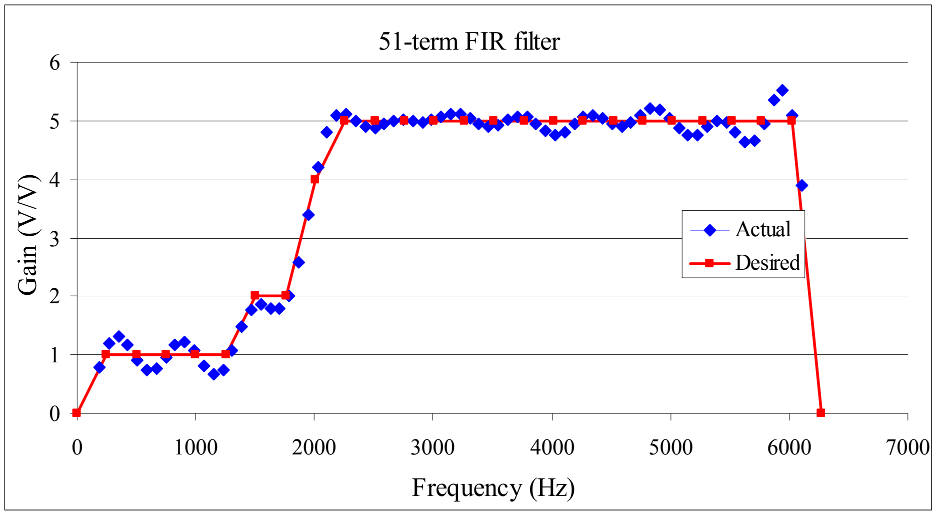

Figure 1. Desired filter response. This is H.

k	f (Hz)	Mag(H(k))	Angle(H(k))	k	f (Hz)	Mag(H(k))	Angle(H(k))
0	0.00	0.00	0.00	26	-6274.51	0.00	77.00
1	250.98	1.00	-3.08	27	-6023.53	5.00	73.92
2	501.96	1.00	-6.16	28	-5772.55	5.00	70.84
3	752.94	1.00	-9.24	29	-5521.57	5.00	67.76
4	1003.92	1.00	-12.32	30	-5270.59	5.00	64.68
5	1254.90	1.00	-15.40	31	-5019.61	5.00	61.60
6	1505.88	2.00	-18.48	32	-4768.63	5.00	58.52
7	1756.86	2.00	-21.56	33	-4517.65	5.00	55.44
8	2007.84	4.00	-24.64	34	-4266.67	5.00	52.36
9	2258.82	5.00	-27.72	35	-4015.69	5.00	49.28
10	2509.80	5.00	-30.80	36	-3764.71	5.00	46.20
11	2760.78	5.00	-33.88	37	-3513.73	5.00	43.12
12	3011.76	5.00	-36.96	38	-3262.75	5.00	40.04
13	3262.75	5.00	-40.04	39	-3011.76	5.00	36.96
14	3513.73	5.00	-43.12	40	-2760.78	5.00	33.88
15	3764.71	5.00	-46.20	41	-2509.80	5.00	30.80
16	4015.69	5.00	-49.28	42	-2258.82	5.00	27.72
17	4266.67	5.00	-52.36	43	-2007.84	4.00	24.64
18	4517.65	5.00	-55.44	44	-1756.86	2.00	21.56
19	4768.63	5.00	-58.52	45	-1505.88	2.00	18.48
20	5019.61	5.00	-61.60	46	-1254.90	1.00	15.40
21	5270.59	5.00	-64.68	47	-1003.92	1.00	12.32
22	5521.57	5.00	-67.76	48	-752.94	1.00	9.24
23	5772.55	5.00	-70.84	49	-501.96	1.00	6.16
24	6023.53	5.00	-73.92	50	-250.98	1.00	3.08
25	6274.51	0.00	-77.00				

Table 1. Desired filter response for one patient that compensates for hearing loss. This is H.

Let x(n) be the input (read from the ADC) and X(z) be the input in the frequency domain. Let y(n) be the FIR filter output, and let Y(z) be the FIR filter output in the frequency domain.

Y(z) = H(z) X(z)

y(n) = IFFT { H(z) FFT{x(t)} }

Take Inverse FFT of the desired gain to get N=51 FIR filter coefficients. Because the negative frequencies in Table 1 are complex conjugates of the positive frequencies, h(n) will be real.

h(n) = IFFT{H(z)} =

0.0170, -0.0020, -0.0316, -0.0566, -0.0614, -0.0399, -0.0035, 0.0233, 0.0196, -0.0135, -0.0503, -0.0590, -0.0303, 0.0104, 0.0208, -0.0187, -0.0794, -0.0982, -0.0306, 0.0968, 0.1810, 0.1010, -0.1922, -0.6206, -1.0035, 3.8431,

-1.0035, -0.6206, -0.1922, 0.1010, 0.1810, 0.0968, -0.0306, -0.0982, -0.0794, -0.0187, 0.0208, 0.0104, -0.0303, -0.0590, -0.0503, -0.0135, 0.0196, 0.0233, -0.0035, -0.0399, -0.0614, -0.0566, -0.0316, -0.0020, 0.0170

Scale to make fixed point coefficients h0 to h50, e.g., 3.8431 ≈ 984/256

```
const long h[51]={4,-1,-8,-14,-16,-10,-1,6,5,-3,-13,
    -15,-8,3,5,-5,-20,-25,-8,25,46,26,-49,-159,-257, 984,
    -257,-159,-49,26,46,25,-8,-25,-20,-5,5,3,-8,
    -15,-13,-3,5,6,-1,-10,-16,-14,-8,-1,4};
```

Multiplication in the frequency domain is equivalent to convolution in the time domain. The FIR filter is the convolution of the data with the inverse transform of the desired filter.

y(n) = h(n) * x(n) = x(n) * h(n) (* means convolution here)

y(n) = sum [h(i)·x(n-i)] as i goes from $-\infty$ to $+\infty$. (· means multiplication here)

Because there are a finite number of h(n) terms, the convolution is a finite sum

y[i]= (h[0]*x[i]+h[1]*x[i-1]+h[2]*x[i-2]+…+h[50]*x[i-50])/256; // * means multiplication here

3) Conversion to the frequency domain using the FFT

After the digital filter, you will calculate the FFT on the y(n) data. You are free to choose an FFT length of N = 64, 256, or 1024 samples. The measurements are displayed on the LCD either in the time domain or the frequency domain. Due to the limitations of the LCD, the graphics will be limited to 128 by 128 pixel resolution.

$$A_k = \sum_{n=0}^{N-1} a_n W_N^{kn} \quad \text{where} \quad W_N = e^{-j2\pi/N} \quad k=0,1,2,\ldots,N-1$$

The range of frequencies is 0 to ½ f_s, and the frequency resolution is f_s/N. Graduate students need to implement FFT windowing to reduce spectral leakage. Graduate students can choose any window function they wish. E.g.,

Hamming	$w(k) = 0.54-0.46*\cos(2\pi k/(N-1))$
Hann	$w(k) = (\sin(\pi k/(N-1)))^2$
Cosine	$w(k) = \sin(\pi k/(N-1))$
Triangle	$w(k) = (2/N)(N/2 - \lvert k - (N-1)/2\rvert)$

Preparation (do this before your lab period)

1 (option 1). You will design an analog interface between the electret microphone and the ADC. Be careful to label the types and tolerances of the capacitors. The DC component will be removed by a high-pass filter, and you will use a shunt reference (LM4041CILPR) to create an analog constant with less noise than the power supply. This way, the sound information, v(t) can be presented to the ADC as a voltage equal to constant+v(t). You will design a low pass filter to reject frequencies above ½ f_s. You will want an amplifier gain so that the ADC swings from about 0.3 to 2.7 volts for normal speaking voices. For now, you can pick any gain around 100. Using TI's FilterPro design tool create an anti-aliasing LPF. Read the help system for this application will help you select the number of poles and filter type (I made a 2-pole Salen-key Butterworth). You are free to adjust the gain and frequency cutoffs later during the testing portion of the lab. Estimate the -3 dB cutoff frequencies for the HPF and LPF.

1 (option 2). You will design an analog interface between the Sharp GP2Y0A21YK and the ADC. Be careful to label the types and tolerances of the capacitors. The range of the analog voltage must be between 0 and 3.3V, and the useful range of the ADC will be 0.3 to 2.7 volts. This way, the distance information, v(t) can be presented to the ADC as a voltage between 0.3 and 2.7 volts. You will design a low pass filter to reject frequencies above ½ f_s. Using TI's FilterPro design tool create an anti-aliasing LPF (alternatively you may use the Excel sheet lpf.xls). Read the help system for TI's FilterPro when selecting the number of poles and filter type (I made a 2-pole Salen-key Butterworth). You are free to adjust the gain and frequency cutoffs later during the testing portion of the lab. Estimate the -3 dB cutoff frequencies for the LPF.

2. Write a background thread that samples the data and puts it into a FIFO, using the RTOS developed in Labs 2 and 3. You may use timer-trigger ADC sampling (better) or software-trigger ADC sampling. Add interpreter commands that allow you
 to set the trigger mode,
 to enable/disable the digital filter
 to print the ADC input and FFT calculations in a form easy to convert to graphs on the PC.

3. Go through the FIR filter design process to create a filter similar to the example shown above. It is ok for you to use the FIR design template FIRdesign51.xls ~~or FIRdesign64.xls~~ as long as you understand the underlying mathematical theory. Write a foreground thread that receives data from the FIFO and implements the FIR filter continuously. A software trigger will determine when to capture a finite length sound recording. The length of this recording will match the FFT length of 64, 256 or 1024 you wish to perform. Your triggering will have at least two modes such as
 1) perform one graphics frame on the SW1 button
 2) start sampling when input goes across a certain voltage threshold
 3) continuous sampling and plotting
You should implement one FIR filter designed for a fixed sampling rate and simply activate or skip the digital filter at run time. I.e., you do not have to dynamically change the filter coefficients based on operator choice of sampling rate.

4. Write a foreground thread that graphs either the voltage versus time or voltage versus frequency on the LCD. Whether you display the voltage versus time or voltage versus frequency can be selected using the interpreter.

Procedure (do this during your lab period)

1. *Gain adjustment*: Perform these tests before connecting the circuit to the microcontroller. Connect the transducer with its amplifier and observe the analog output on an oscilloscope. Verify the analog signal does not saturate the analog amplifiers. Adjust the gain so that the voltage is about 0.3 to 2.7 volts for normal speaking or distances 10 to 80 cm.

2 (option 1). *Dynamic system test*: Perform these tests before connecting the circuit to the microcontroller. ~~Connect a speaker to a signal generator and adjust the amplitude of the sound so that it approximates the loudness of normal speech. Connect the microphone, and connect oscilloscope probes at the input and output of your analog circuit.~~ Remove the transducer and connect the signal generator to the input of the analog circuit. Make sure the voltage level of the signal generator is within range, so that the inputs and outputs of your analog circuit are not saturated. Record the sine-wave amplitudes of the input and output voltages. Collect measurements for about five ~~ten~~ different frequencies. Make sure you choose frequencies above the LPF cutoff and below the HPF cutoff. Calculate the amplifier gain at each frequency. Plot the gain versus frequency response of your audio circuit.

2 (option 2). *Dynamic system test*: Perform these tests before connecting the circuit to the microcontroller. ~~Position a moving object that oscillates an object of variable distance to the sensor (like a spinning bar attached to a motor) and adjust the amplitude of the object motion so that it moves at a distance observable by the sensor. It is important to have the distance traveled by the object known and constant, but allow the rate to vary. Power up the system, and connect oscilloscope probes at the input and output of your analog circuit.~~ Remove the transducer and connect the signal generator to the input of the analog circuit. Make sure the voltage level of the signal generator is within range, so that the inputs and outputs of your analog circuit are not saturated. Record the input and output voltages. Collect measurements for about five ~~ten~~ different frequencies. Make sure you choose slow frequencies that the sensor can observe and fast signals the sensor cannot observe. Calculate the system gain at each frequency. Plot the gain versus frequency response of your system.

3. *Digital scope and spectrum analyzer tests*. Evaluate your oscilloscope operation without the FIR filter. Again connect the ADC input to a sinewave generator at three frequencies. Verify the generator is 0 to 3V before attaching. Using a sinewave shape set the frequency at about 0.1 f_s, 0.25 f_s, and f_s. Place a real oscilloscope on the analog input and adjust the scale on your scope so that one or two cycles are observable on your LCD scope. Take a photograph or hand-draw your LCD display. Using your interpreter, print out the input voltage versus time measured by your system. Plot three graphs of voltage versus time data for the three frequencies
 Analog signal at the ADC input measured with a scope
 Collected voltage versus time data from your system, plotted with a program like MatLab or Excel
 Photographs or drawings of your system in scope mode

Repeat the comparison in the frequency domain. Using your interpreter, print out the voltage versus frequency results of your FFT. Place a real spectrum analyzer on the analog input. <u>Plot three sets of voltage versus frequency data for the three frequencies</u>

Amplitude versus frequency signal at the ADC input measured with a spectrum analyzer
FFT output data from your system (get the units correct), plotted with a program like MatLab or Excel
Photographs or drawings of your system in spectrum analyzer mode

4. *FIR filter test*: Design an experiment to test the digital filter. Feel free to create any test you think is appropriate. Make sure the ADC input remains in the 0 to 3.0V range at all times. Option 1: Capture a spoken word and compare the resulting frequency spectrum of the raw input to the FIR-filtered output. Option 2: Capture a motion typical of a robot turning next to a wall and compare the resulting frequency spectrum of the raw input to the FIR-filtered output.

5. *Determine the bandwidth.* In oscilloscope mode with the FIR filter active, stream data continuously from the ADC to the LCD. Increase the sampling rate until data becomes lost. Use debugging instruments to detect lost data and to determine which module limits the bandwidth.

Deliverables (exact components of the lab report)

A) Objectives (1/2 page maximum)
B) Hardware Design
  Circuit diagram of the microphone interface (preparation 1)
C) Software Design (printout of these software components)
  1) Software used to acquire data, filter, perform FFT and plot graphics (preparation 2,3,4)
  2) Software used to test the FIR filter (procedure 4)
D) Measurement Data
  1) Dynamic circuit performance (procedure 2)
  2) Digital scope data (three*three = 9 graphs) (procedure 3)
  3) Spectrum analyzer data (three*three = 9 graphs) (procedure 3)
  4) FIR filter test data (procedure 4)
E) Analysis and Discussion (2 page maximum). In particular, answer these questions

1) Give the calculations (equations and results) you used to estimate the cutoff frequencies for your HPF and LPF (Preparation 1)? How did the measured frequency response compare to the estimations (procedure 2)?

2) Explain how you measured maximum bandwidth (procedure 5). What was the limiting factor affecting bandwidth?

3) What is the expected FFT output if the input is a squarewave?

4) Look at the noise in your digital samples when it is very quiet? What type of noise is it?

5) We made a big fuss over jitter in labs 2 and 3. Can you estimate the jitter in your ADC samples?

6) Prove your FIR implementation can not overflow.

7) Look at the symmetry in the **`h[51]`** coefficients in the example FIR design. How could you rewrite the following filter equation to reduce the number of multiplies from 51 to 26?
  y[i]= (h[0]*x[i]+h[1]*x[i-1]+h[2]*x[i-2]+…+h[50]*x[i-50])/256;

8) If your system executed the FIR filter using the multiply and accumulate instruction (MLA), you can skip this question. Explain how the MLA instruction could have made your filter execute faster? If you were to have used the MLA instruction, would it have been more accurate?

Checkout (show this to the TA)

You should be able to demonstrate the proper operation similar to procedure 3. Be prepared to discuss how the FIR filter was designed and what the FFT data means. Be prepared to explain spectral leakage. The TA may ask you to discuss the various factors that limit bandwidth on this system.

Hints

1) It does not matter whether your spectrum analyzer outputs units of voltage or dB, as long as you know the difference.

2) It is OK to download software from the web (not from other EE445M students of course), as long as you reference the source of the software and follow all legal procedures. If you find yourself erasing someone else's name off the source code, you are probably doing something wrong.

3) Check blackboard for clarifications and the class web site for updates.