ECE 3710 Lab 6 – Fall 2017

Due Date: Week of November 13 at the beginning of your lab section (20 points)

Objectives

The student should be able to operate an analog to digital converter (ADC) and a digital to analog converter (DAC) on the microcontroller to produce a varying frequency sine wave.

Overview

In this lab we will write a C program that generates a sinusoid using a DAC connected to the microcontroller at a frequency specified by the user—i.e. we'll turn the microcontroller into a sinusoidal waveform generator. The user will be able to select the frequency of the sine wave by varying the level of a voltage source connected to the ADC.

Preparation

1. Become familiar with the operation of your microcontroller's onboard ADCs (see the corresponding sections in the TM4C123GH6PM Data Sheet). You will find the ADC corresponding pins in section 13-2.

2. Familiarize yourself with the following:
    a. MCP4725 External DAC data sheet
    b. Analog test board (speaker) schematic
    c. LM386M Amplifier datasheet

3. Prepare the following parts
    a. Tiva C series development board
    b. Analog test board module
    c. Jumpers and wires
    d. 10nF capacitor
    e. MCP4725 12-bit DAC
    f. Power Supply

4. Answer the following questions:
    a. What is the input voltage (Vss) range for the DAC?
    b. What is the function of the A0 pin of the DAC? Select a value for A0. Using your value of A0, what is the I2C address of the DAC? (A2 = 0, A1 = 1).
    c. Draw a wiring diagram connecting your microcontroller to the needed components.
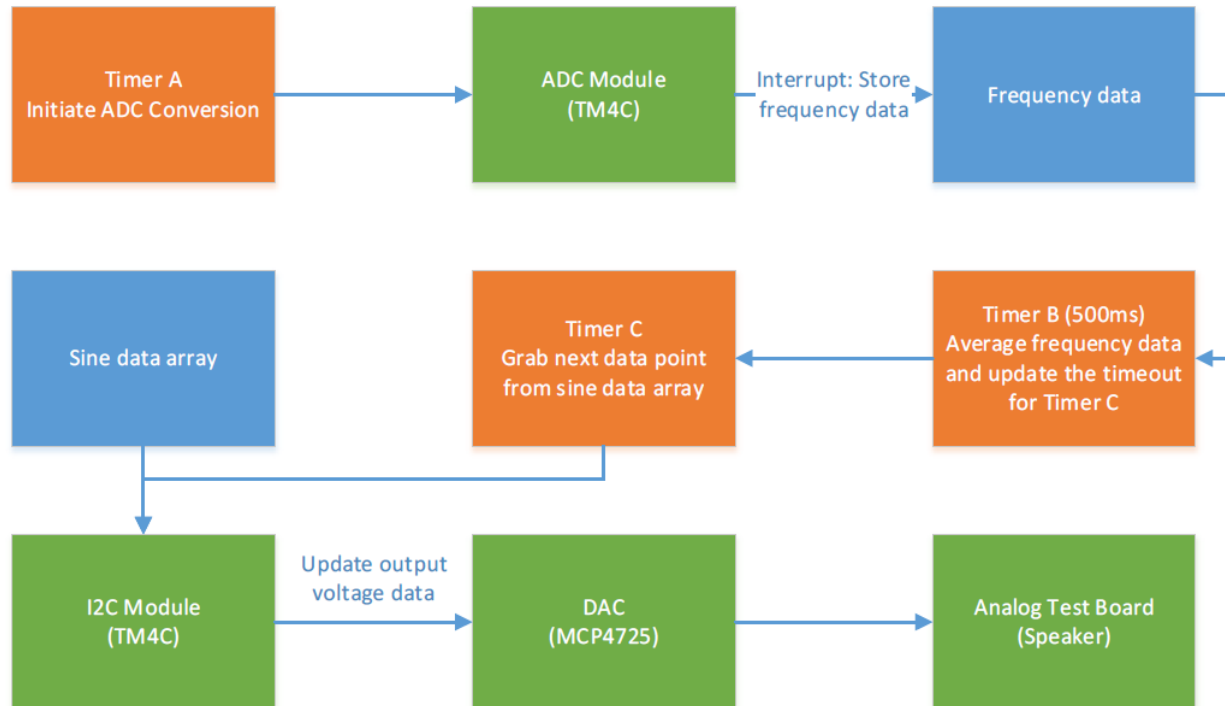
Requirements

1. Use the external 16 MHz crystal with the PLL to set the system clock (configure the PLL to produce whichever frequency is most convenient). This is necessary to produce consistent output on the DAC.

2. Connect one of the analog board potentiometers to ADC0. Connect the output of the DAC to the analog test board (AIN or AUDIO_IN), you will be able to hear as the frequency changes. It is not a pleasant noise.

3. Use a timer to configure the ADC to automatically sample the input line (i.e. initiate a conversion) every 2 ms.

4. Make use of an interrupt (no polling) to copy the converted voltage value from the ADC's registers.

5. Produce a sine wave using a look-up table with 40 entries.

6. Configure I2C to communicate with the DAC in fast mode (400 kHz).

7. Update the DAC as a result of a timer (auto-reload) using an interrupt. The frequency range is 100 Hz to 1000 Hz.

8. Set the DAC output as a result of an interrupt.

9. Use the capacitor to filter the square edges of the DAC waveform.

10. Update the output frequency every 500 ms (averaging conversions).

Table 1: List of timers and interrupts

Timer / Interrupt needed	Function	Frequency
Timer A	Initiate ADC conversion	2ms
ADC Interrupt	Copy voltage from ADC to appropriate place	n/a
Timer B	Update the Timer C frequency using data from the ADC Interrupt	500ms
Timer C + Interrupt	Send to I2C to update the output voltage of the DAC	Varies

Verify Requirements

1. Demonstrate to your TA that you have met the above requirements
2. Include in your lab report a discussion of how the voltage is used to determine the frequency.
3. Use the oscilloscope to view the filtered waveform from the DAC and verify the frequency. Include a picture of the waveform from the oscilloscope in your lab report.
4. Show that the correct frequencies are produced for the following codes: 0, 1023, 2047, 3071, and 4095. Measure the voltages these correspond to. The frequencies must be accurate to within 2 Hz.

Notes and Tips

- Since you are sampling the line 250 times as often as you're checking to see if the user wants to output a different frequency, you should use the extra samples to mitigate the effects of noise and power supply jitter on the measurements; i.e. you should average the ADC code values for the 250 samples to determine the actual ADC code to use for the frequency calculation.
- The DAC has an output buffer that changes the output impedance. This may be something to consider when filtering the rigid edges of the waveform.