MW 2411 Lab #4
Generation, Processing, and Analysis of Sine Waves

1 Overview

In this lab, you will use the Digital-to-Analog Converter (DAC) and the timer on the Flex-UI/dsPIC33F board to generate a sine wave which will be analyzed in Matlab subsequently. You must write a program that does all of the following. Plot the signal.

1. Use preprocessor defines (`#define`) to define values for the sine waves frequency, sample rate, minimal voltage of the signal and maximal voltage of the signal. Additionally use a macro to convert the interrupt frequency to the number of timer ticks.
2. Use timer 3 to generate an interrupt.
3. Within the timer interrupt use the defined values to calculate the sine waves current value and perform the DAC conversion.
4. Toggle LED 1 at every timer interrupt.
5. Analyze a recorded signal in Matlab by removing noise and determining the frequency of the recorded signal.

2 Procedure

For the generation of sine waves, following steps must be implemented on the Flex-UI/dsPIC33F board:

1. You can find an MPLAB X IDE project with template code on the Moodle course page.
2. Use your knowledge from the previous labs to configure the DAC and timer 3.
3. Update `lab04.c` and `dac.c` such that it fulfills the requirements specified above.
4. Make sure to use the defined macro to set the appropriate timer compare value (`PR3`) based on the interrupt frequency.
5. The output voltage can be calculated with a sine function:
   $V_{\text{out}} = V_{\text{amplitude}} \cdot sin(\omega_{\text{signal}} \cdot t) + V_{\text{offset}}$, with $\omega_{\text{signal}} = 2\pi \cdot f_{\text{signal}}$.

The signal processing and analysis of sine waves consists of the following steps to be implemented in Matlab (use the template `Lab04_Matlab_Signal_Processing.m`):

1. You do not need to record the signal on your own, use the provided signal called `signal.csv` and load it into Matlab using the `readmatrix()` command. Plot the signal.
2. Design a butterworth filter. You can use the built-in functions of Matlab. You must identify the sampling frequency of the signal based on the information given in `signal.csv`. Following characteristics must be fulfilled by the filter you design:

(a) The passband ends at 30 Hz with an attenuation of maximal 3 dB.

(b) Some noise is expected in the range of above 200 Hz. These high frequency components must be reduced by at least 40 dB.

3. Apply the filter to the input signal and plot both the original and filtered signals in the same figure. Add a title, the axis descriptions, and a title to the figure.

4. Fourier transform the *original* signal using MATLAB's built in functions. See the `doc` call and the course slides about how to apply an FFT, e.g., type `doc fft` in the command window. Note that we are only interested in the single-sided frequency spectrum. Plot the resulting spectrum. Take care to remove the DC signal, i.e., the offset V_{offset} by calculating the mean value of the signal. Plot the obtained spectrum into a new figure; only visualize the spectrum up to 500 Hz. Subsequently, Fourier transform the *filtered* signal and plot it into the same figure. Did the filter remove the high frequency components?

5. Write a function that determines the frequency of harmonic signals (you can assume there is only one harmonic wave in the signal). What is the frequency of the recorded signal provided for analysis in this assignment? Print your result. Note: MATLAB requires to define functions at the end of a script.

Due date of code submission can be found on the Moodle submission page for this lab. Only one member of the group must upload the code (all .c, .hhit, and .m files that your project uses compressed in one zip file). At the start of Lab 5, each lab group will be asked to demonstrate and explain their Lab 4 code to the lab instructor. You will also be asked to change the frequency, sample rate and voltages in your code and show the generated sine wave using the oscilloscope.

3 Questions to Ponder

The following questions are provided for your lab group to think about. No written response is required.

1. Is there any other way to generate this and other waveforms from a microcontroller?

2. Is there limit for the sample rate and the signal frequency? And how are they linked?

3. How could you speed up the sine wave generation for a static frequency?

4. What would haven if you do not remove the mean (DC signal part) of the signal before the FTT, i.e., how would the plot of the frequency spectrum look?