Due 12/3. Individual effort.

The description below will be updated with additional information so check this page regularly.

**Project specifications:**

* Objective: create a variable frequency triangular wave using the DAC chip connected serially to the AVR microcontroller.
* The user provides the frequency of the signal using a keypad.
* The user entry is echoed to the LCD.
* The LCD 4-bit mode is preferred, but 8-bit mode is OK.
* The DAC can be connected in parallel or serially to the AVR chip via SPI or USART.
* Serially interfacing the DAC requires using 2 shift registers. This can be seen in the DAC & DTA notes. Implementing a serial SPI/USART connection will result in a 25% extra credit.
* Determine the highest sampling rate and confirm it experimentally.
* You should build your project in Simulation IDE and use Microchip Studio for programming.
* Use the function/clock generator and the oscilloscope in Simulation IDE to test your design.

**Algorithm:**

**Parallel DAC Circuit**

1. Create an empty byte array.
2. Connect the keypad, LCD, DAC to various ports of the AVR
3. Prompt the user to enter the frequency of the triangular wave.
4. Read the input and echo it to the LCD. Convert each ASCII digit to an integer and update the frequency. Ignore illegal characters (don't display them).
5. Based on the sampling rate, determine the number of samples needed in a period. The number of samples should be odd.
6. From the number of samples compute the voltage step size. Samples must start at 0 and go to the max value 0x3FF then back to 1 step above 0.
7. Populate the array with the sample values.
8. Use an infinite loop to read each sample and send it to the DAC to generate the signal.

**Submission:**

Submit **1 zip** file that contains

* a report document,
* a simulation IDE file (.sim)
* a C program (.C)
* a hex file for the simulation (.hex)

**Report:**

In your report include the following (don't do a full report format):

* Schematic showing all components and their pin connections (ie screenshot you simlation IDE). The circuit may be too big to fit on a page, so split it into separate images. Label wires if needed (e.g if they're cut-off).
* Derive:
  + The highest sampling rate and confirm it experimentally.
  + MAX frequency that can be produced accurately.
  + MIN frequency that can be produced accurately.
* Code:
  + Code will not be graded if you don't follow instruction below:
    - INDENT code
    - DON'T double space the code or use a font larger that 11pts
    - DON'T use running comment lines (that continue on the next line)
* Test results (image screenshots):
  + MAX freq detected properly
  + MIN freq detected properly
  + Screenshots should clearly show the settings on the input and output of the scope