EE572
HW#2
Due Monday, January 24

0. Pick a Lab partner and form a group of 4-5 people for the project. It is helpful if at least one of your group members has some experience in assembly language programming.

1. a) Represent the number -3.625 in 8 bit fixed point notation (two's complement) then represent it in 32 bit IEEE floating point.

   b) Later in EE572, we will see that minimizing the necessary sampling time is crucial in many control applications. Given this, state why it is important for digital control engineers to work in either fixed point or integer numbers instead of floating point numbers whenever possible.

   c) In the time domain, the results of **sampling** the signal, $x(t)=5\cos\pi t+5$, at a sampling period of $T_S = 1/4$ second.

   d) Now, find the Fourier Transform of x(t) and sketch the results of sampling the signal using $T_S = 1/4$ second in the frequency domain (i.e., sketch $X_s(f)$).

   e) According to Nyquist, have we sampled x(t) quickly enough to recover it?

2. a) Suppose we have a 2-bit (25 cents – Ha!) A/D D/A converter. Sketch the results of **quantizing** the signal, $x(t)=5\cos\pi t+5$ in the time domain

   b) In reality, we have a 12-bit A/D converter in lab with a dynamic range of 20 volts (i.e., ±10 volts). What is the mean square error (MSE) due to quantizing with our system?

   c) With our 12-bit system, we use a linear encoding scheme such that the Dynamic range is 10 –(-10) = 20 volts. The width of each quantizing band is $D/2^N = 20/2^{12} = 4.883$ milivolts. Thus, in our encoding scheme 0000000000 represents –10 + 0.004883/2 = -9.99756 volts and 111111111111 represents the maximum of +10 - 0.004883/2 = 9.99756 volts volts. With this information, determining the corresponding analog voltage if the encoded value is:
   i) 001001010011 ii) 111001011100 iii) 000001001000 iv) formula for any 12-bit integer, x

   d) With our 12-bit A/D D/A system described above, first quantize then encode the following analog voltages into 12 bit numbers:
   i) -3.1 volts ii) 0.1 volts iii) 5.8 volts iv) formula for any analog voltage, x

   e) Although the most prevelent type of encoding found in digital control systems is the kind discussed in class today, occassionally a more sophisticated encoding scheme is used. One such encoding scheme is called the **Grey Code**. In Grey Code, only 1 bit changes in the encoded value from one quantizing level to the next level. For example, if n=2 bits, a possible Grey Code encoding scheme is: 00 - lowest level; 10 - next lowest; 11 - next lowest; 01 - highest. See how only one bit is toggled in successive encoded values as we move from one quantizing level to the next? This code would facilitate error checking if we knew our signal would not change more than one quantizing level for each sampling period. Can you devise such a Grey Code scheme for n=3 bits?