

## Data Acquisition

Name \_\_\_\_\_

Partner (s) \_\_\_\_\_

Grade \_\_\_\_\_/10

### Introduction

The goal of this lab is to learn to apply proper sampling theory when gathering data using an Analog-to-Digital Converter (ADC) system.

### Objectives

- Learn about ADC systems;
- Learn proper technique for sampling an analog signal;
- Learn how to interface an ADC board to Matlab;
- Learn the basics of signal processing.

### Equipment Provided

- Oscilloscope;
- Signal generator;
- Computer with ADC and connection boards (new).

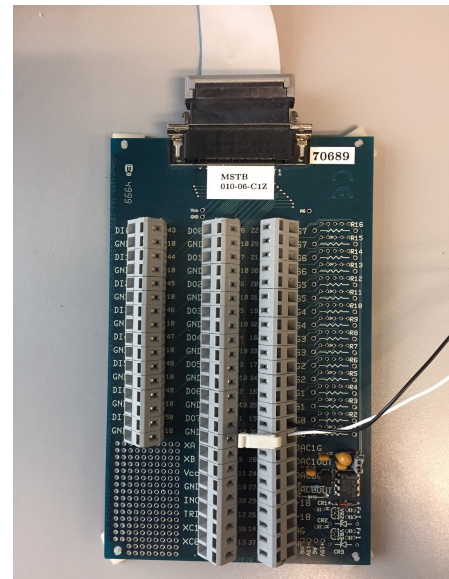
### References

- Instrumentation text book;
- Course web page.

### Procedure

Your lab this week includes a new piece of equipment – the Microstar DAP840 analog-to-digital (ADC) interface board. The board is installed internal to the computer at your work station. You connect signals to the ADC interface board via a ribbon cable and connector strip shown at the right. The first thing to do for this lab is to setup the Agilent signal generator to output a signal which you will capture via the DAP840 board and read into Matlab using a script that has been written for you.

- 1) Connect the Agilent 33250A signal generator output to the DAP840 board by connecting the signal generator ground to the **G0** terminal of the ADC interface board and the signal generator output to the **S0** terminal of the A/D interface board. Use the little white “pry bar” to open the connection port. You can see the pry bar on the long grey strip at the right. Take care that you do not lose this pry bar.



- 2) Set the signal generator to output a sine wave with an amplitude of 2 volts peak-to-peak, 1-volt DC offset, and a frequency of 500Hz.
- 3) Use the Tektronix scope to verify your waveform looks like it should at the connection terminals to the A/D board. This is a good time to learn more about the features of the oscilloscope, so put the scope in **DC Coupling** mode and then use the **Measure** function to display the peak-to-peak voltage, the frequency, and the RMS voltage and record the values below. Calculate what the RMS voltage should be and record below.

$V_{pp}$  \_\_\_\_\_ V

Freq \_\_\_\_\_ Hz

$V_{RMS,meas}$  \_\_\_\_\_ V

$V_{RMS,calc}$  \_\_\_\_\_ V

- 4) Place the scope in **AC Coupling** mode, which removes the DC component of the signal it is displaying. Record the RMS voltage from the scope and the calculated RMS value.

$V_{RMS,meas}$  \_\_\_\_\_ V

$V_{RMS,calc}$  \_\_\_\_\_ V

- 5) What conclusions can you draw about the method used by the scope to calculate RMS values?

- 6) Place your scope back in the **DC Coupling** mode. Start Matlab on the PC.

- 7) Download the lab32020.m file from the class web site and save it to your STEM drive or to the desktop. This file uses a Matlab function named **msgets0** to gather data from channel 0 on the Dap840 board installed inside the lab computers. The format of the **msgets0** function is:

```
[data,time] = msgets0(rate, numpoints, gain);    % Reads A/D channel 0 "single-ended"

data = vector of voltage values returned from the converter, in decimal;
time = time vector corresponding to the data points;
rate = number of samples per second;
numpoints = number of data points;
gain = optional selectable gain (1, 10, or 100 with a default of 1);
```

Here is a sample program to acquire and plot data:

```
[d,t] = msgets0(2000,100);    % rate = 2000 samples/second, numpoints=100, gain=1
d = d*5/32768;                % Convert A/D counts to volts – the DAP840 returns the
                              % signed 14-bit results in a 16-bit word with zeros padded in
                              % the two LSB positions.
plot(t,d);                    % Plot the waveform
```

- 8) A copy of the lab32020.m Matlab script is shown below:

```
close all;
clear all;

Sample_frequency = 50000;      % sampling frequency
number_points_to_gather = 10000;

[d,t] = msgets0(Sample_frequency, number_points_to_gather);
d = d * 5 / 32768;            % Convert ADC values to voltages

knownF = 500;                 % Known frequency
numberCycle = 2;              % desired number of waveforms displayed

ptswave = Sample_frequency / knownF;    % number of data points per wave

                              % length = numberCycle * ptswave * ( 1 / knownF);
length = numberCycle * ptswave * ( 1 / Sample_frequency);

plot(t,d);
axis([0 length -1.1 1.1]);
hold on
plot(t,d,'ok');
axis([0 length -1.1 1.1]);

msstop;    % This needs to close the execution otherwise frequency values don't seem to work.
```

- 9) Modify the file above, if necessary, to display *approximately two periods* of the waveform on a Matlab plot. Run the program and print your plot. Also, print a copy of your code.

- 10) Modify the **lab32020.m** script as indicated below and comment on your results. Capture 500 data points. Set the DC offset to 0V.

Sample Rate (Hz)	Frequency(Hz) (Agilent)	Amplitude (V <sub>p</sub> ) (Agilent)	Comments
20,000	500	1	
2,000	500	1	
100	500	1	

- 11) Using the **Arb** function on the signal generator, choose *Select Waveform*, then *Built-in waveforms*, then the **Cardiac** option to output an approximation for an EKG signal. Record the lowest frequency that you can sample at and still see the entire waveform. Print a plot.

Sampling frequency \_\_\_\_\_ Hz

- 12) Now that you understand the basics of signal acquisition using an ADC board and Matlab, the next step is to do some signal processing from within Matlab. Set your function generator to output a sine wave of 1000Hz, 0.0 V offset, and 2.0 V<sub>pp</sub>.
- 13) Capture the waveform in Matlab. Then, add Matlab code to the lab32020.m function to calculate and display the following parameters on a plot of the waveform. Note that there may be built-in Matlab functions to do some of these. Print the plot.
- Frequency.
  - Period.
  - Peak-to-peak voltage.
  - Stretch goal - RMS voltage.
- 14) Repeat step 13 for a square wave.
- 15) Run the Matlab function *msstop* to shut down the connection between A/D board and Matlab. Then exit Matlab before signing off.

## To Turn In

- This handout stapled to your plots.