

# Introduction to MATLABs Data Acquisition Toolbox, the USB DAQ, and accelerometers

In class this past week, you should have starting learning about the data acquisition toolbox. You should have at least tried a few things using the sound card in class. Now we will continue to learn about the toolbox using the USB data acquisition system.

- 1) First you should test out your USB system and some aspects of the software using the function generator as the input for you to measure with the DAQ. Start with something like a 100 HZ, 1 Volt amplitude sine wave from the function generator. Write a script using the data acquisition toolbox commands that acquires 1 second of data from the DAQ from one analog input channel. Measure the signal in differential mode and experiment with setting some different parameters, such as the sample rate, the input range, the length of time you acquire data for. Try collecting data in differential and single ended model; is there a difference? Try increasing the frequency on the signal generator, can you get aliasing? There is no need to turn anything in for this part of the lab, simply experiment with the hardware and make sure that you understand how to control it and acquire data. You can confirm that the signal the DAQ is measuring is correct by checking with the scope (though it is possible the scope is out of calibration).
- 2) Once you are convinced this is working, get one of the accelerometer boards. **These are reasonably expensive and should be returned after you are done for the day. Do not leave them laying around, do not take them from the lab.** The accelerometer is the ADXL325, the data sheet is found on the website. The small surface mount chip is attached to the evaluation board so you can easily try things out by plugging the board into your protoboard. On the green board, you will see the labels “com” (for common, or ground), “z”, “y”, “x”, “Vss”, and “ST” (self test). If you want to use a sharpie to neatly write these labels to the correct pins on the top of the chip, you may do so to help you remember which pin is which. Place the board into your protoboard (do not put all the pins in the board on the same row!). Run a wire from the 2.5V supply on the USB DAQ to Vss. Connect ground on the DAQ to com. The X, Y, and Z accelerations can be connected to the DAQ in single ended mode to channels 1, 2, and 3 respectively.
- 3) The chip is designed such that at rest, the output is approximately  $\frac{1}{2}$  the supply voltage. At 5 g, the voltage is 2.5 volts and at -5 g it is zero. Therefore you will only see a slight difference between the components of the acceleration that are zero and the one that is not (gravity). Calibrate your voltages such that your output on the plot is acceleration in G's. You will need to find the voltage when the chip is at rest (you can use the matlab command “mean” to find this value) and you can use the fact that you know g at rest on the surface of the earth to find the sensitivity; i.e. how many mV/g. Convert your voltage output to acceleration in units of g.
- 4) Test the sensor by rotating the board so that you pass through all orientations (z component = 1 g and -1 g, x component = 1 g and -1 g, etc.) If everything is calibrated this should work just fine. In your lab report, show an example of you rotating your sensor.

- 5) At rest, take the data from your sensor and integrate it numerically (use MATLAB's trapz command) to get position versus time. You will find that the drift is very extreme. Not only will the direction that has gravity acting be falling, the x and y position will drift very badly. This is an inherent problem in double integrating accelerometers for position. The drift is so bad that you cannot use them for navigation, perhaps only for very short periods. Try a test where you shake your board quickly back and forth (holding level to the table and moving in the plane) and see if you can even see this in the position measurement. If you carefully calibrate you can almost make this work for a fraction of a second! In your lab report show an example of the drift in your system. See the following for the company's evaluation of the possibility of using accelerometers for navigation:

<http://www.analog.com/library/analogdialogue/archives/43-05/accelerometer.html>

- 6) Try creating a few options for continuously displaying the data in near real time – so that the MATLAB plot is like kind of like an oscilloscope. You might try some different options and see what works the best. At the simplest, you can try putting a loop around the main part of the program that acquires the data. Try collecting the data for a short period, plot it, then repeat. This will appear very choppy because you are starting and stopping the analog input. Try collecting the data continuously on the device (collect infinite samples) – don't start and stop the analog input. Include the program you did come up with as an appendix to your lab report. Post your final MATLAB program for this exercise in your directory so that your ninja can try it out.
- 7) Create a WiiMote that detects the orientation of the protoboard. Create a simple matlab display that shows the orientation as a vector (or something else) in a simple 3D plot. Post your final MATLAB program for this exercise in your directory so that your ninja can try it out.
- 8) Finally, compare the sensitivity (mV/g) that you found above and compare that to the data sheet. Note that in the data sheet, the numbers are for a 3V supply and we are giving a 2.5 V supply. However, if you look at page 11 of the data sheet, it will explain what to do with supply voltages other than 3V. You should find that the agreement between your calibration and the data sheet is not that great. This has to do with the resistors on the output of the sensor and the input of the DAQ. On the data sheet for the ADXL325 you will find the following diagram;

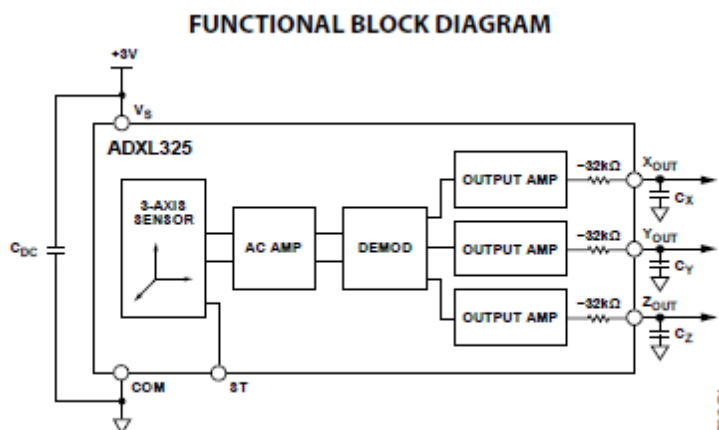


Figure 1.

For the USB 6009 you will find,

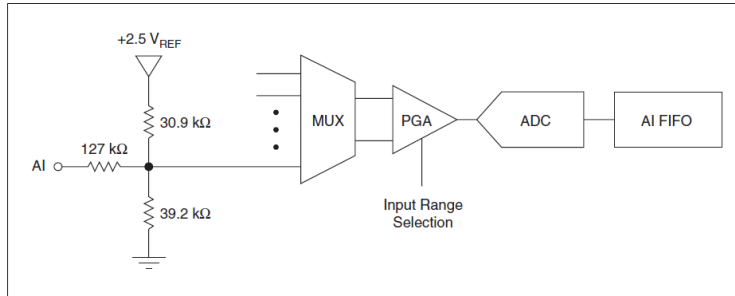
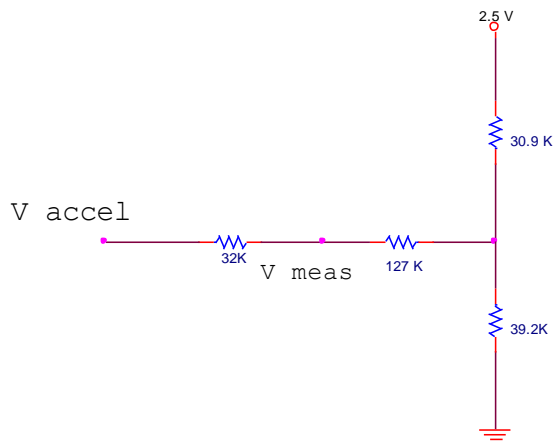


Figure 7. Analog Input Circuitry

The 32 K resistor on the output of the chip and the resistor network on the input to the DAQ are of the same order of magnitude and are interfering with each other. On the accelerometer, the output of the output amp is the accelerometer voltage that we want.

Just focusing on the resistor network of these two components hooked together, we would have a network as shown below. The voltage into the network,  $V_{\text{accel}}$ , would be the voltage from the chip which is proportional to the acceleration, the true output of the accelerometer chip, ADXL325. The voltage  $V_{\text{meas}}$ , is what you are measuring with the DAQ. On the DAQ, you can assume that the MUX (multiplexer) draws very little current. In analyzing this circuit, you will find that  $V_{\text{accel}}$  and  $V_{\text{meas}}$  are not quite the same.



Analyze this circuit, using nothing more than Ohm's law and current conservation at the node (i.e. what current flows in, must flow out), and find an algebraic relationship between  $V_{\text{meas}}$  and  $V_{\text{accel}}$ .

Now in your DAQ program, convert the measured voltages at X, Y, and Z to the actual accelerometer voltages using the relationship you just found. Now repeat your calibration. This time you should find that values you find are much more consistent with the specifications on the data sheet.

In your lab report, show a very brief overview of the analysis (it can be hand written), and write your function which converts the measured voltage. State your values of the sensitivity and the voltage at 0 g that you find, as well as the nominal values expected from the data sheet.

**Deliverables:**

Turn in the requested matlab codes and plots. This lab is mainly to get you up to speed on working with the hardware and software. Nonetheless, make sure your plot all are readable, have axis, are labeled, have units, and that there is a short description of what each plot shows. Since this is mainly an exercise to get started there is little analysis of the results. The only analysis you need to demonstrate is the final question regarding the relationship between the measured voltage and the accelerometer voltage. Turn in the two requested matlab codes so that your grader can try them out easily and see how they work – again, this is mainly an exercise in getting things working.