

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

This week we will start to learn the software that we will use through the course, MATLAB's Data Acquisition Toolbox. This toolbox allows us to interface MATLAB with hardware that allows us to control the USB based data acquisition device. This week we will use a relatively simple sensor, a 3-axis accelerometer, like those found in cell phones and Wii motes.

First you will need to complete the software installation as detailed under Week 0. You should have downloaded and installed the driver software before coming to class. There are some additional software steps that you cannot complete until you have the DAQ plugged in. Be sure to follow these steps now when you are in the lab. If you are in the first lab section (Friday morning), you will need to connect all the wires to the DAQ. If you are in a later lab section, you will need to confirm that the wires are attached properly.

First you should test out your USB data acquisition system and some aspects of the software. Download the template file. Start by just running it and making sure you get no errors. Next, use the function generator to test things a little. Output a sine wave from the function generator and connect it directly to analog input channel 0. By default the analog inputs take differential measurements, so you need something plugged into ai0+ and ai0-. Try modifying some aspects of the software, such as the number of channels acquired, the sample rate, the number of samples, etc. The only point of this testing step is to get you familiar with the hardware and software with a simple signal.

Once you are convinced everything is working, get one of the accelerometer evaluation boards, shown here.

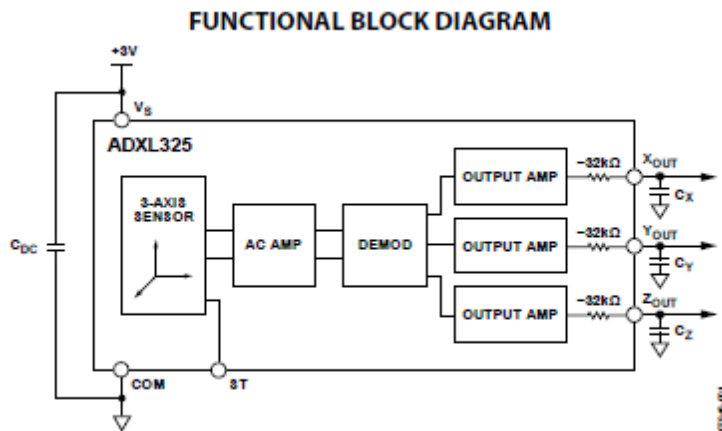


These are reasonably expensive and should be returned after you are done for the day. Do not leave them lying around, do not take them from the lab. The accelerometer chip we use is the ADXL325, the data sheet is found on the website. The small surface mount chip (the square near the top of the board) is attached to the evaluation board so you can easily try things out by plugging the evaluation board directly into your breadboard. On the green accelerometer board, you will see the labels "com" (for common, or ground), "x", "y", "z", "Vss", and "ST" (self test). If you want to use a sharpie to neatly write

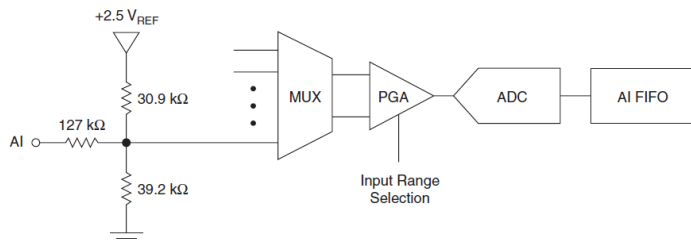
these labels to the correct pins on the top of the chip, you may do so to help you remember which pin is which. Place the board into your breadboard (do not put all the pins in the board on the same row!). On the breadboard, run a wire from the 2.5V supply on the USB DAQ to V_{ss}. Connect ground on the DAQ to com. The X, Y, and Z accelerations can be connected to the DAQ on analog input channels 0, 1, and 2 – on the positive input. The negative analog inputs should be connected to ground.

The chip is designed such that at zero acceleration, the output is ½ the supply voltage. At 5 g, the voltage is the supply voltage 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). In the MATLAB program, convert your measured voltages such that your output on the plot is acceleration in G's. If you look on the data sheet you will find the sensor's sensitivity in mV/g. This value on the data sheet is provided for a supply voltage of 3 V. If you read through the data sheet you will find the sensitivity scales with the supply voltage. Therefore the sensitivity for your setup is not the value on the data sheet, but is proportional to it. Look at the section of the data sheet "Use with operating voltages other than 3 V".

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 you have done everything correct at this point, you will find that basic behavior seems correct as you rotate the sensor, but that the value of the acceleration will not be 1 G. The reason lies 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;

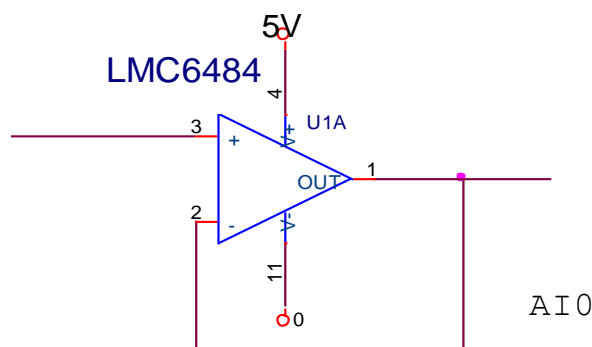


For the USB 6009 you will find the following resistor network exists on each analog input.

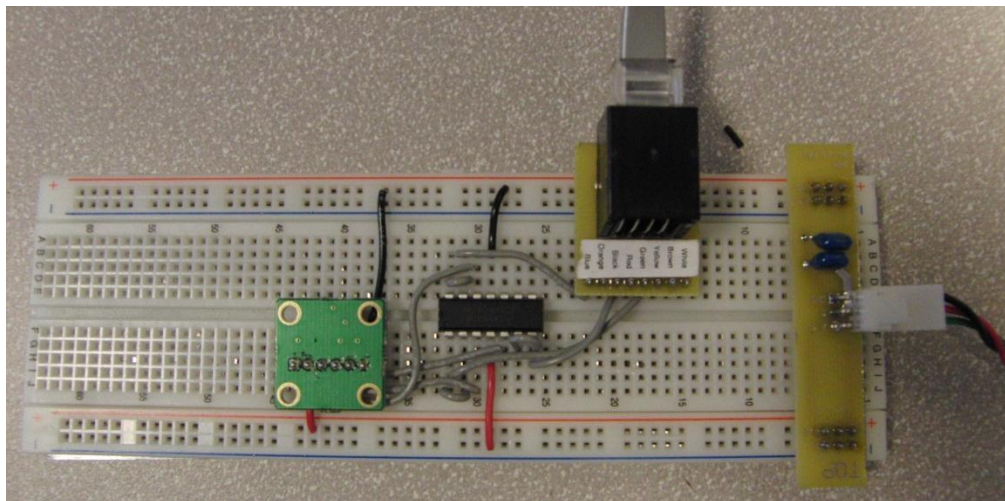


The 32 K resistor on the output of the chip and the resistor network on the input to the DAQ are interfering with each other. On the accelerometer diagram, the voltage at the “output amp” is the accelerometer voltage that we want. An analysis of the circuits connected to each other would show that there is a voltage drop across the 32 K resistor and thus an error in the acceleration measurement.

A simple way to correct for this error is to add an op-amp buffer between the sensor and the DAQ. A simple follower circuit like we built last year will correct the problem, shown below. If you recall, the op-amp inputs draw no current, thus at DC there will be no voltage drop across the resistor on the output of the accelerometer.



This year we will often use a quad op-amp which has 4 op-amps on a single 14 pin chip. See the data sheet linked on the course website for the pin diagram. We will be using the LMC6484. This op-amp can run on the 0 to 5V power supply from the USB. Here is an example of a built circuit, please don't try to trace the wires out, build yours from the schematics.



Once you insert the op-amp buffer between the sensor and the data acquisition system, you should get an accurate reading for the acceleration using the sensitivity from the sensor's data sheet. You need one buffer for each of the 3 acceleration components.

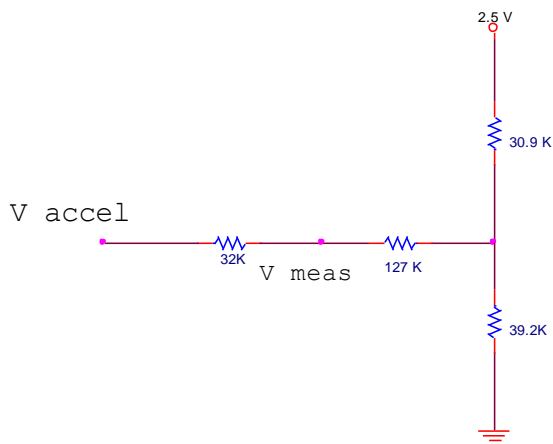
Once everything is working show a plot of you rotating the sensor showing that the x, y, and z accelerations range from about 1 to -1 G. Include this plot in your lab report.

Now convert your 3-axis accelerometer to an orientation sensor. Your program should continuously plot a 3D vector (or some other way to represent the sensor orientation) “continuously” as you rotate the breadboard. The easiest way to do this is put the whole template program in a loop. Then shorten the amount of time that you collect data for each time through the loop. Say, collect data for 0.1 seconds, average all the data to get the average orientation over this time, plot the vector, then repeat.

A little analysis

In addition to the lab, you are also required to do a little bit of analysis to turn in.

Just focusing on the resistor network (at low frequency the capacitor does not do anything) of the accelerometer and the DAQ hooked together WITHOUT THE OP-AMP BUFFER, we would have a network as shown below. The voltage into the network from the accelerometer output amp, V_{accel} , would be considered the true output of the accelerometer chip. The voltage V_{meas} , is what you are measuring with the DAQ. On the DAQ resistor diagram, you can assume that the MUX (multiplexer) draws no current.



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_{meas} and V_{accel} . In your lab report, show a very brief overview of the analysis (it can be hand written), and write your function which relates the actual accelerometer voltage to the measured voltage.

Deliverables:

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 measured results. The only analysis you need to demonstrate is the final question regarding the relationship between the measured voltage and the accelerometer voltage.

Your lab should have

- 1) A plot showing the working 3-axis accelerometer in units of g as you rotate the breadboard through all its possible orientations. Also include your MATLAB code that you used. Be sure to include the sensitivity you used to convert from voltage to acceleration in units of g.
- 2) Your MATLAB code and a short explanation of how you convert the accelerometer signal to orientation.
- 3) An analysis of the resistor network.

Grading

The graders will follow the approximate formula below. 0 points for any aspect of the deliverables that are not provided. Full credit for things that are correct.

- 1) 5 points maximum for deliverable 1.
 - a. 3 points off if things seem blatantly wrong, but you seem to have tried
 - b. 2 points off if it seems correct, but graph is not labeled at all.
 - c. 1 point off if it is fine, but the graph or code are sloppy, poorly labeled or otherwise difficult to read.
- 2) 2 points maximum for deliverable 2.
 - a. 1 point off if it is done, but difficult to read your code, or no explanation of how to convert the sensors signal to orientation.
- 3) 3 points maximum for deliverable 3.
 - a. 2 points off if analysis was attempted but completely faulty from the start.
 - b. 1 point off if the analysis setup is correct, but a mistake in the final answer