Lab Design Exercise: Arduino ADC/DAC

We will spend the next couple of weeks working through some basic functionality of the Arduino UNO and reinforce the fundamental concepts of ADC/DAC. Instead of blindly following a "recipe" of steps, you will have the freedom to come up with your own device design, and you will have the opportunity to use some of the new design tools at your disposal in this process.

Device Objectives

Design an Arduino-based device that:

- Accepts an analog voltage input signal that could range from -5–5 V.
- Protects your Arduino from input voltages that exceed the 0–5 V.
- Displays the input voltage on the LCD shield as a binary number with maximum possible precision. You can choose your method for indicating a negative number.
- Outputs the input analog voltage input signal using the analog output pin.
- Updates the displayed binary voltage and output voltage every second.
- Extra Credit: Adapt your device to accommodate an input sine wave over the same voltage range. What is the maximum frequency that your device can accurately accommodate?

Your device will require analog circuitry pre- and post-Arduino to achieve these functional objectives.

Code Repository

Please create a new GitHub repository for the code associated with this device.

- This can be a public repository using whatever management workflow you want to implement with your group members.
- Please add Dr. Palmeri (mlp6), Nick (nbottenus) and Matt Brown (sphaerobolus) to have at least read access to your repositories.
- Utilize the wiki associated with your repository as a pseudo-ELN for this project.
- Utilize the Issues feature as appropriate during your development / troubleshooting.

Grading

Your device will be graded during lab on 2014-09-12 by Dr. Palmeri, Nick and Matt Brown. The following grading guidelines will be used:

Function (0/5)

- Binary number representations of analog input values are correct.
- LCD display and DAC output updated every second.
- Input voltage range protection.
- Circuit is well-built (Matt Brown standards). Building a ”clean” circuit helps insure correct function.
- Extra Credit: Sine wave is accurately output within your specified frequency range.

GitHub (0/5)

- Well-commented code
- Logical functional layout
- Good use of commits and issue tracker
- Multiple team-member contribution

Testing (0/5)

- Design a procedure to test that your device achieves its functional objectives.
- Perform your testing and present them formally on your GitHub wiki during your device demonstration. Remember, statistics are your friend (or, if not your friend, expected)!

Documentation (0/5)

The wiki associated with your GitHub repository will serve as your “lab report” for this project. Be sure to include the following items:

- Functional Block Diagram
- Circuit Schematic (DipTrace or Eagle)
- Testing Procedure
- Testing Data
- Discussion of your device’s performance, what you could/would change in hindsight, how it could be optimized, etc.

Peer Evaluation (0/5)

You will evaluate your own effort / contributions to the project and those of your group members. Please see the `TeamPeerAssessment.pdf` in the Resources section.