Lab 4 - Continuous Sensors

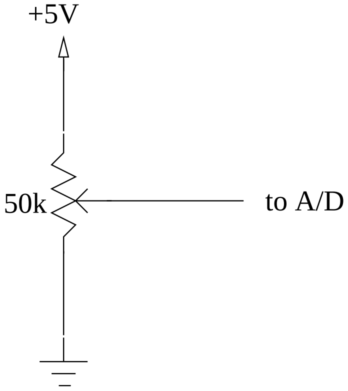
Music 250a - CCRMA 2002

DUE WEDNESDAY OCT 30, 2002

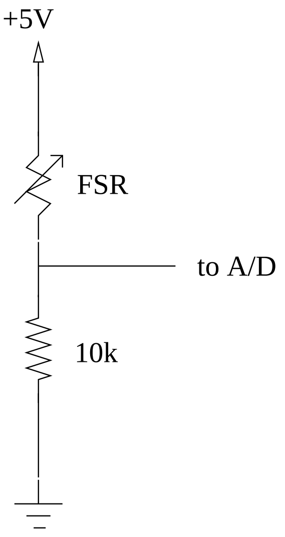
1. Voltage from sensors:

Hook up each of the sensor circuits below. Before connecting them to the A/D pins of the Atmel, look at the voltage output of the sensor on an oscilloscope or multimeter. What is the voltage range? The Atmel uses a 0-5 Volt range. For maximum resolution, you should have as close to 0V-5V as possible. Try using different resistance values in the circuits to see if you can optimize the voltage range.

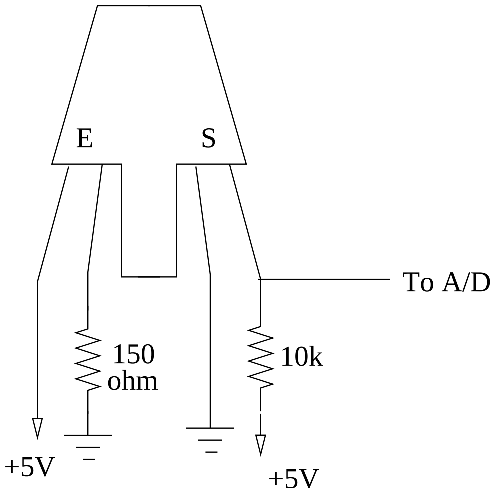
A. Potentiometer circuit:



B. FSR circuit:

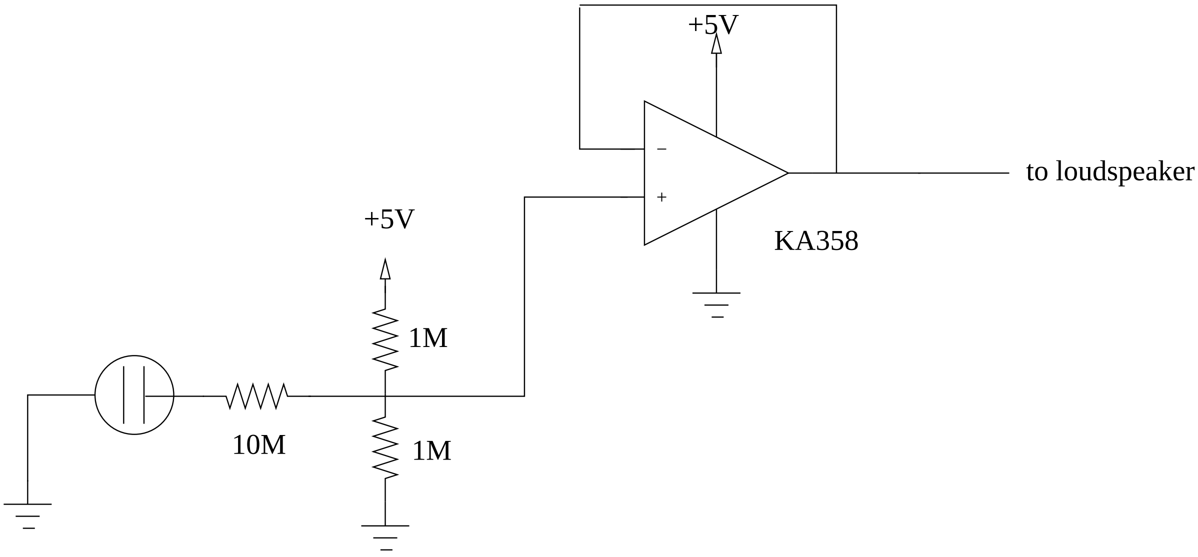


C. Optical proximity sensor circuit:



D. Piezo sensor circuit:

- remember that the piezo won't give a very nice steady voltage reading, but look at it anyway...



2. A/D conversion and MIDI/OSC out:

Now, connect your sensors to the A/D pins of the Atmel (A0-A3). Using the sample code from the website as a start, read the value of the sensors and send the value via MIDI and OSC to Pd to continuously control some aspect of sound. You can use or modify your patch from last week if you want. You can do one sensor at a time, or all 4 at once. Remember that MIDI can only handle values from 0-127 (7 bits), and OSC can handle 32 bit values. The A/D converter can return 8 or 10 bit samples. So if you want to send an 8-bit sample as MIDI, you will have to throw away one bit. If you are using OSC, you could use the full 10 bits.

You can use the provided Pd patches from the website to start making sounds and to visualize your data.

3. Optional / Extra credit ideas:

- Using MIDI, pack together two 7-bit bytes in order to send all 10 bits of a sample.
- Avoid redundancy and only send out values when the input changes.
- Use the LCD to debug your A/D readings.
- Use multiple sensors to control different aspects of your patch.

- Implement a simple digital filter to smooth out the data.