Lab 3 - Music with MIDI and Open Sound Control

Music 250a - CCRMA 2002

DUE WEDNESDAY OCT 23, 2002

- **Read through the entire lab first, because items in later parts may affect your strategy on earlier parts.**

1. MIDI->Pd:

The purpose of this lab is to control and create music (or sound) in Pd. Using buttons as inputs to your Atmel microcontroller, you will send messages from the Atmel that relay your button presses to your Pd patch via MIDI. You can use as many buttons as you want, and you **must** use at least one button on the solderless breadboard. The result should be music, either a short composition or a framework for improvising. Think about the system as whole: the inputs, the processing of the inputs, and the output sounds you will produce. Your choices in any one of these aspects will affect all the others.

You will need to write a C program for your Atmel and create a Pd patch. You can use the avrlib-demo programs (demo6 and demo7 for MIDI and OSC) to help you get started. See the website for Pd patches as well.

2. OSC->Pd:

Do the same as #1 using Open Sound Control instead of MIDI. You may want to incorporate them in the same program using something like this:

```
//#define USE_MIDI
#define USE_OSC


.
.
```

```
.

#ifdef USE_OSC
oscSendMessageOneArg(PSTR("/button/down"),button);
#endif
#ifdef USE_MIDI
midiNoteOnOut(60+button, 100, 1);
#endif
```

3. Performance:

Find one of your classmates and teach him/her how to use your system. They will need to perform your composition or improvise with your system for one or both of the TA's. **Note that this is mandatory, we are not joking.**

4. (Bonus - OPTIONAL) ADC - continuous control:

Use one of the continuous sensors on your board (potentiometer, FSR) as a continuous controller. Use the built-in ADC on the Atmel (see avrlib-demo #5) and send the values as MIDI or OSC messages to your Pd patch.