21M.370 Digital Instrument Design
Lab assignment 4 - Due March 17 at 2:00pm

Deliverables:

1. A link to a 1-2m video of you performing with the knuckles instrument

Assignment description:

Before working on this lab you should have completed assembling your basic kit, including soldering headers onto the ESP32 and the breakout board, and attaching headers to two potentiometers and four buttons.

Once that is done, you will:

1. mount your pots and buttons in an enclosure
    1. Here is a link to a video showing how I did this, with some practical advice for prototyping
2. connect them to the breakout board using jumper cables
    1. You will want to monitor either usi
3. program your esp32 with the 'knuckles' firmware:
   NIME/Class/Labs/lab4/Knuckles/knuckles/knuckles.ino
4. open the PD patches:
   NIME/Class/Labs/lab4/Knuckles/main.pd
   NIME/Class/Labs/lab4/Knuckles/knuckles
5. run the knuckles.py python script
   NIME/Class/Labs/lab4/Knuckles/python/knuckles.py
6. use your mouse or type 'z d t' on your laptop keyboard to enable some toggles.

This should get you to the point where you can generate sound using the instrument. Here is a video overview of the knuckles instrument, which also shows all of the code running.

For your deliverable, you are going to record a 1-2m long performance using knuckles. Think about it as a composition - how does it begin, develop, and end? What range of sounds can you get to? Explore the parameters of the instrument to make it sound very different than the default sounds.

There are several places to debug if it doesn't immediately work:

1. In arduino, set <SERIAL_DEBUG> to 1, and open the serial monitor. You will get a stream of all sensor data.
    1. You should make sure your pots range smoothly from 0 to 4095, and that all of your buttons trigger events. You might need to turn off autoscroll to find button events.
2. In python set the <PACKET_INCOMING_SERIAL_MONITOR> variable to 1 at the top of the knuckles.py file to see incoming data from the esp32:

    ```
    packet [1, 4, 223]
    packet [0, 1, 254]
    packet [1, 4, 223]
    packet [0, 1, 254]
    packet [1, 4, 215]
    ```

    1. If this doesn't work then python isn't finding your esp32. At the very top of the console look for the list of available serial ports. You should see one which matches the port used by Arduino to program the ESP32.
    2. Make sure you don't have the Arduino Serial Monitor open, as that will conflict with the python script.

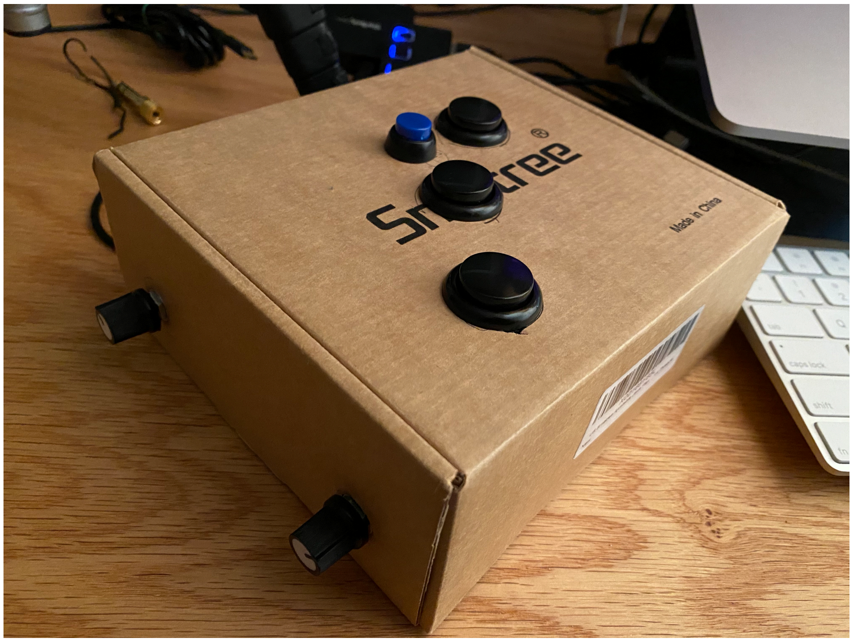
3. In PD there is a visual monitor for the controls. You should see the sliders / toggles change state as you manipulate the controls. If you don’t then there may be a problem with OSC connecting. Try close and reopening the knuckles_ctrl patch.

Once you have sound being controlled by the interface, you will want to explore the instrument for a while and plan on how to make a short 1-2m video

---

Overview of the knuckles instrument

Knuckles uses four buttons and two potentiometers. When you build you own you can lay these out however you wish - here is what I chose to do:



BUTTONS

Three of the buttons are assigned to drum voices:
button 0: bass drum
button 1: snare drum
button 2: hihat

The third button (the blue one in my case) is a <mode shift> button, which will allow you to use the potentiometers to change parameters for the drum voice.

POTENTIOMETERS

The two potentiometers are used to control tone and sequence parameters. The tone potentiometer will:

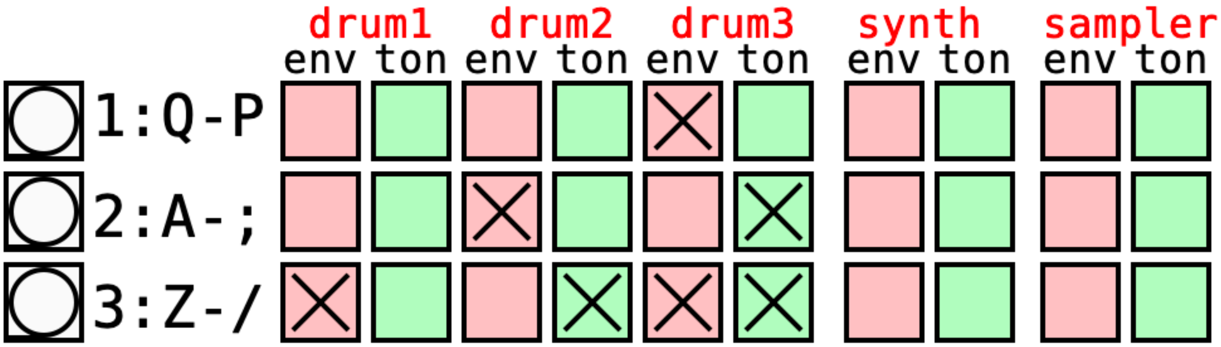
1. control the tone of a drum voice when both the <drum voice select> button and the <mode shift> button are pushed. At lower values the tone know will control volume and decay time, and as the knob is turn up it will more strongly affect the tone.
2. Control the tone of the synthesizer - both the filter frequency and decay time.

The sequencer potentiometer will:

1. control the number of pulses in a euclidian rhythm when both the <drum voice select> button and the <mode shift> button are pushed.
2. control which steps of the ‘16steps’ sequencer will be played to control the pitch of the synthesizer.

In addition, changing the values of the potentiometers controls the amplitude of the synth voice. The more movement, the louder the synth voice.

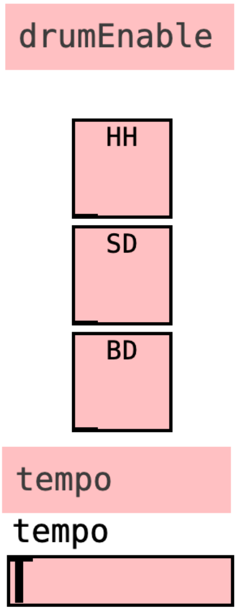
There is also a 5th voice, a sampler. You will need to manually load a sound file into the sampler to use it.



	drum1 env	drum1 ton	drum2 env	drum2 ton	drum3 env	drum3 ton	synth env	synth ton	sampler env	sampler ton
○ 1:Q-P	□	□	□	□	☒	□	□	□	□	□
○ 2:A-;	□	□	☒	□	□	☒	□	□	□	□
○ 3:Z-/	☒	□	□	☒	☒	☒	□	□	□	□

The three euclidian sequencers can be assigned to control the five different voices using this toggle array. Each voice has an amplitude envelope (env) and also a tone envelope (ton). The env for a voice must generally be triggered to hear that voice (except for the synth voice).

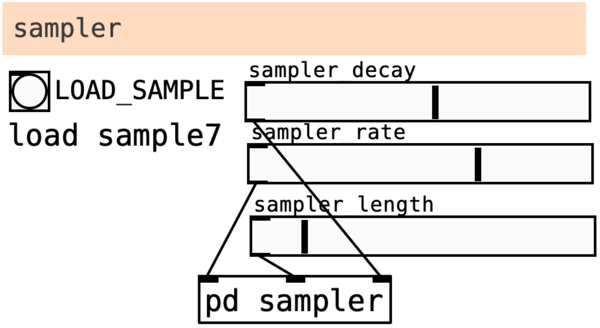
Each row of toggles is mapped to a row of your computer keyboard, as labelled above.

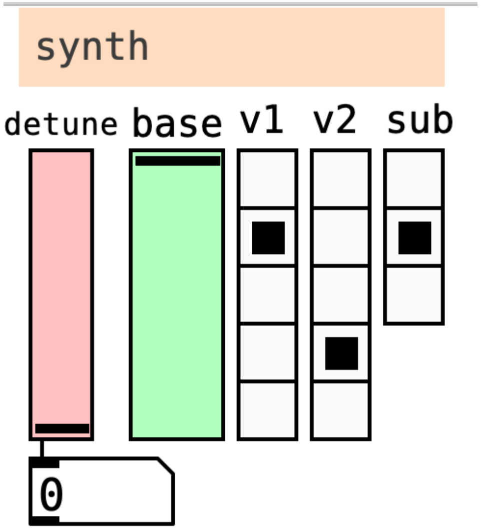


The drum enables in the control panel are just for reference - you will control these using the buttons of your instrument.

Below the enables is a tempo slider for controlling the tempo of the sequencers.

Click the “load sample” button to open a file browser. I suggest using the ‘sample7’ which comes with automatonism, but feel free to explore other samples. You may need to tweak the rate of the sampler to tune the sampler to your synthesizer - and the other parameters just to tweak the sound.

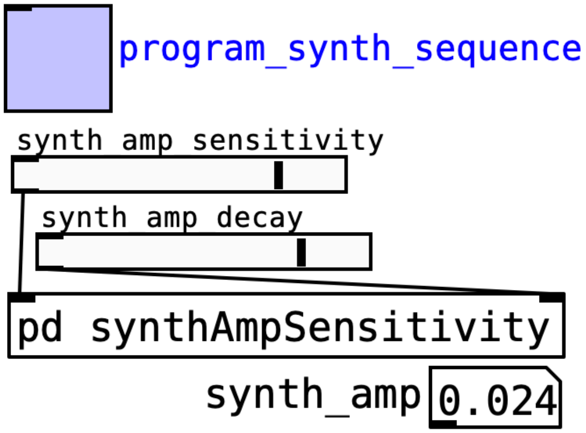




There are four voices in the synth. The ‘base’ slider changes the main frequency of your synth - v1 and v2 transpose the frequency for two additional voices, and sub shifts the frequency down one or two octaves for a sub-octave. The detune slider will detune v1 and v2 relative to the base frequency.

Also note that for the synth, the base voice and sub are triggered by the ‘env’, while v1 and v2 are triggered by ‘ton’.

There are a few additional synth controls. ‘program_synth_sequence’ lets you use the potentiometers to



change the pitch sequence in the 16steps objects. You should leave this disabled, but feel free to play around with it and see how it affects the sequence. As a hint, each button controls a waveform with pots controlling amplitude and frequency.

You can also change the sensitivity for the amount of potentiometer movement needed to control the amplitude of the synth.