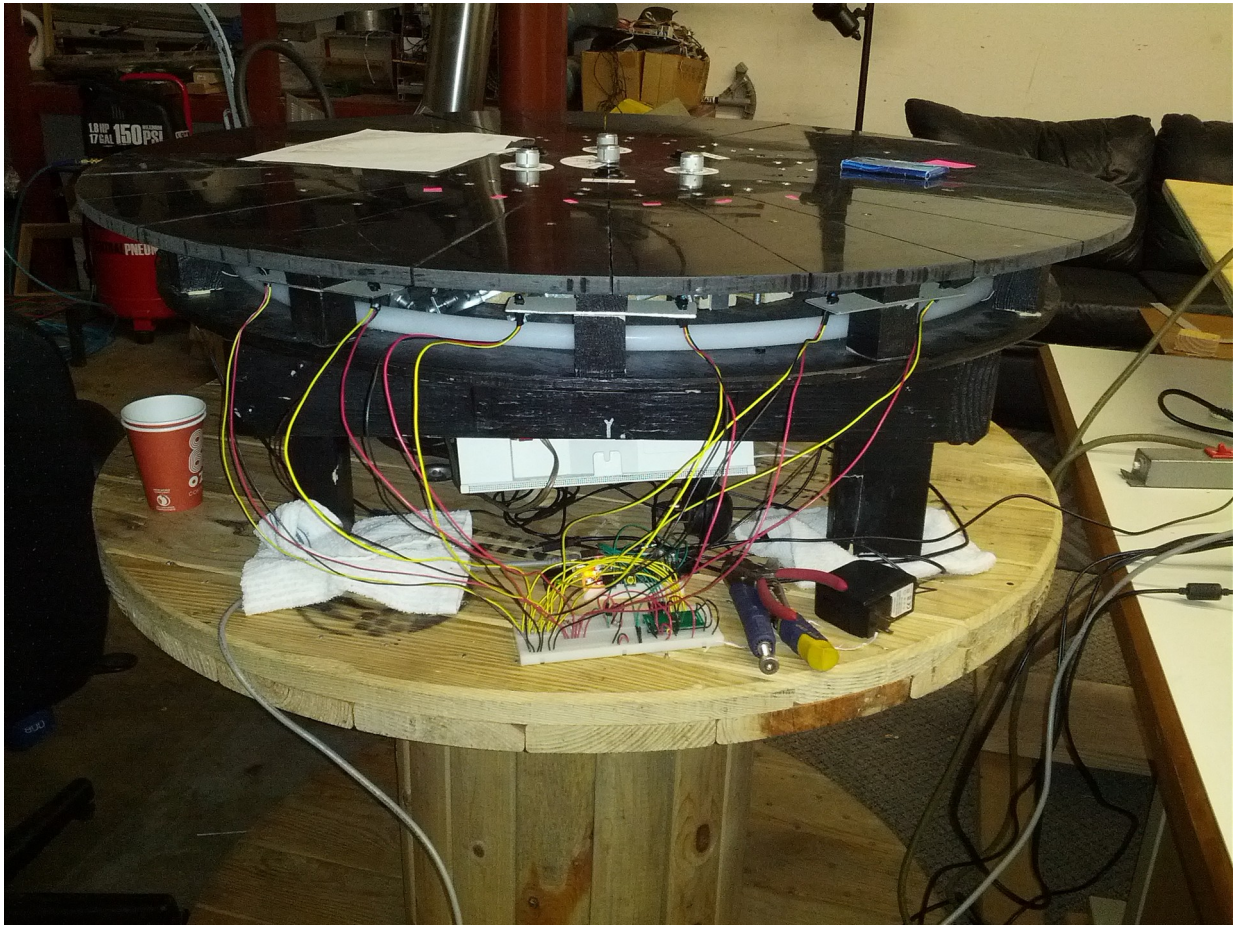
Cyclophone

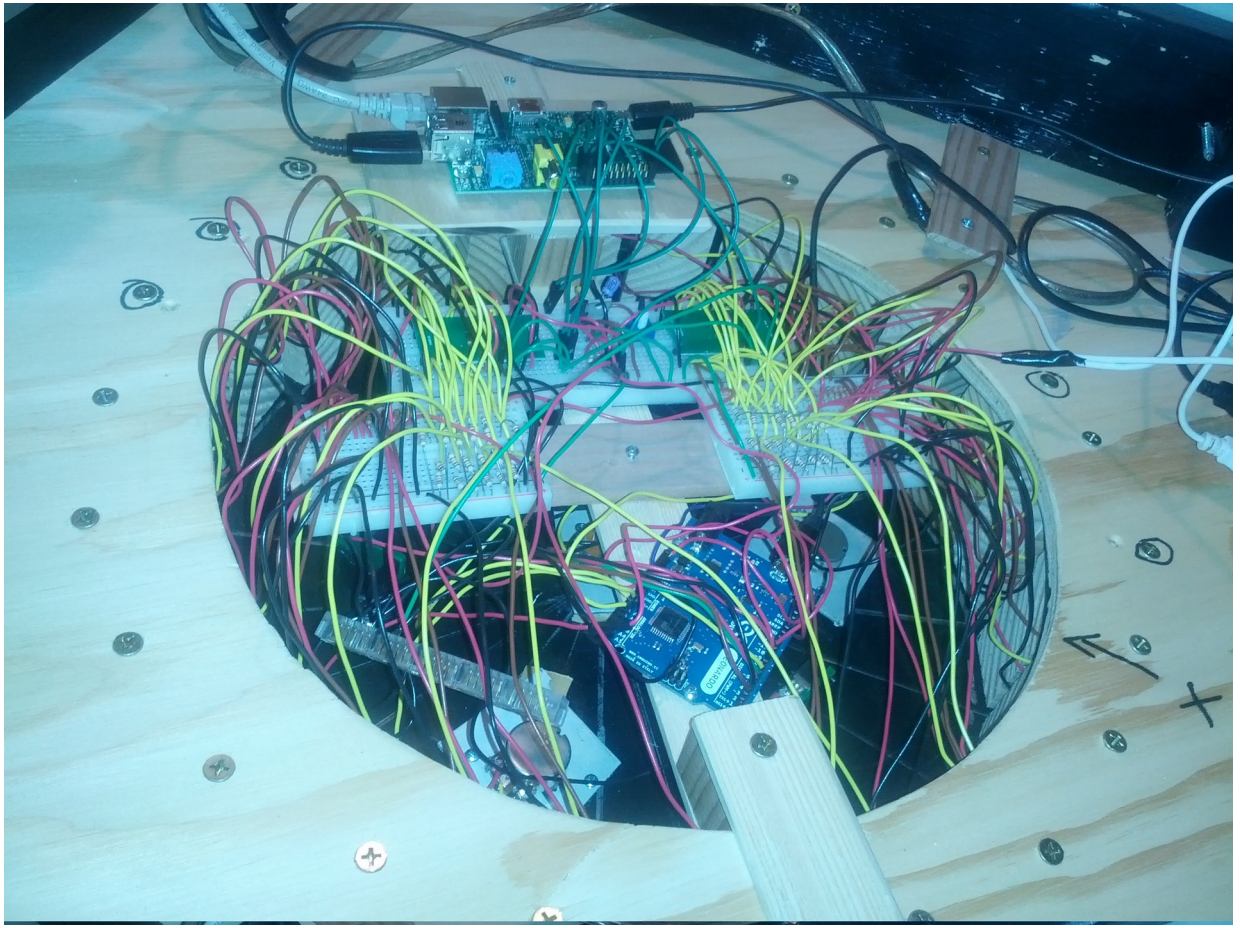
An ARM powered instrument
using Haskell.

Motivation

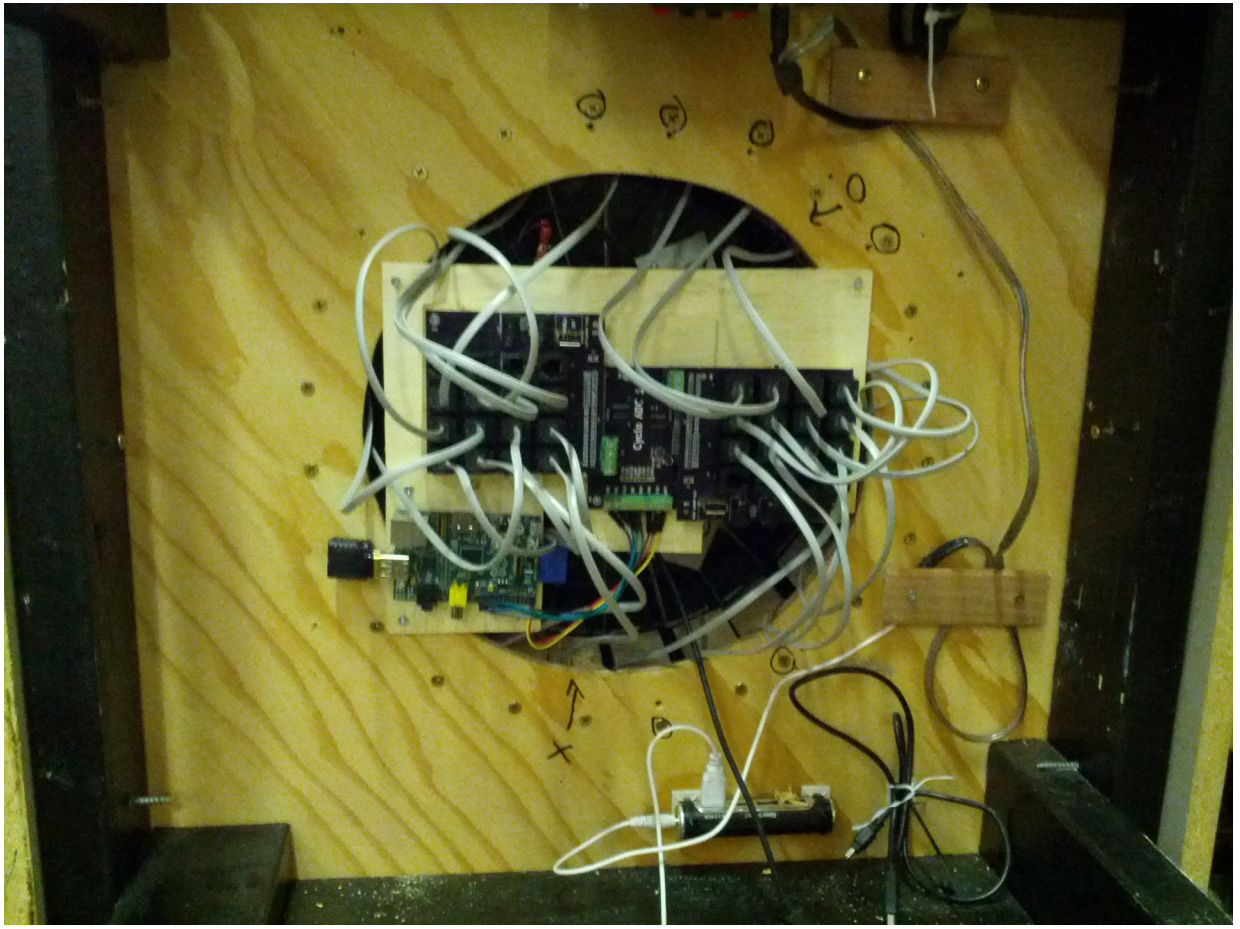
- Got involved in the Boulder Hackerspace (SSD) in 2013.
- Somehow got caught up in making an accessory for the SSD flagship project, Soundpuddle.
- Why not functional?


SOUNDPUDDLE


150PSI




Cyclo ADC 1.0



Design Goals

- Solar/battery powered.
- Raspberry PI or similar for low power.
- Realtime sound synthesis

Hardware

- Arduino #1 for arcade buttons and knobs.
- Arduino #2 for LED control
- Infrared phototransistors for key position
- Banana pi for key position scanning, sound synthesis.

Audio Toolkits

- Initially Overtone/Clojure. Too slow on ARM.
- Euterpea has a nice book. Not realtime.
- Csound-expression. Potentially realtime, but more oriented towards composition.
- Hsc3/supercollider.

Software

- Arduino “C+” code
- C++ for key position scanning (1500 hz vs 500 hz for haskell)
- Haskell recieves OSC messages from key scanner
- Haskell controls supercollider for sound synthesis (hsc3 library).
- Haskell LED server, bridges OSC and arduino

Difficulties

- Typical linux audio problems. Jack/dbus/etc.
- Latency – partly hardware, partly Jack.
- Raspbian haskell out of date for audio toolkits.
- Compiling GHC problematic on raspberry pi, and when compiled had bugs.

Wins

- Eventually found solutions to the major problems.
- Good haskell ARM experience on Arch, even ghci works there.
- Compile on laptop, often works the first try on ARM.
- Crash free and reliable.
- Reasonably speedy.

Future Plans

- Reduce latency by eliminating Jack. Maybe another try with csound.
- Web server for extra controls, chord sequencing, sound sampling. yesod/purescript.
- Robotic noisemakers in addition to audio synthesis.