An Introduction to Sequencing in Pd

BMus/BSc - Level 4

Dr Mat Dalgleish
mathewdalgleish@gmail.com

Assumptions

- Students are diverse - mixed interests, largely inexperienced in programming.
- There’s been a previous session introducing the module (LOs, assignment briefs, etc.) and introducing basics of Pd.
- Access to a Mac lab (Pd, Logic).

Session Overview

- Pure Data (recap)
- Where can Pd used?
- Sequencers - examples
- **Building a sequencer… in a modular way**
- Pd => IAC => Logic
- Task (for later)

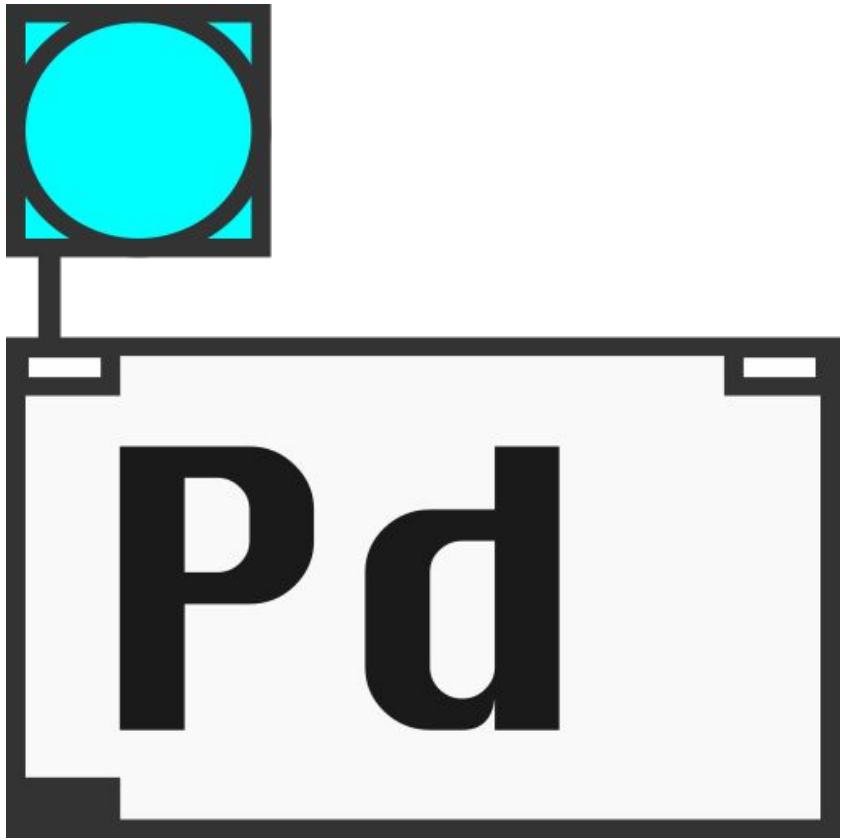
Materials

Today's materials (these slides and Pd patches) are available at:

https://github.com/matdwlv/bcu

or

https://tinyurl.com/2s4jtz6e


Pd

Pure Data

- An open source visual programming environment originally developed by Miller Puckette.
- Part of the Max family.
- Main flavours:
  - Pd vanilla - by Puckette.
  - Purr Data - ported to a HTML5 GUI.
  - Pd-L2Ork - same HTML5 GUI port used in Purr Data but different additional externals.

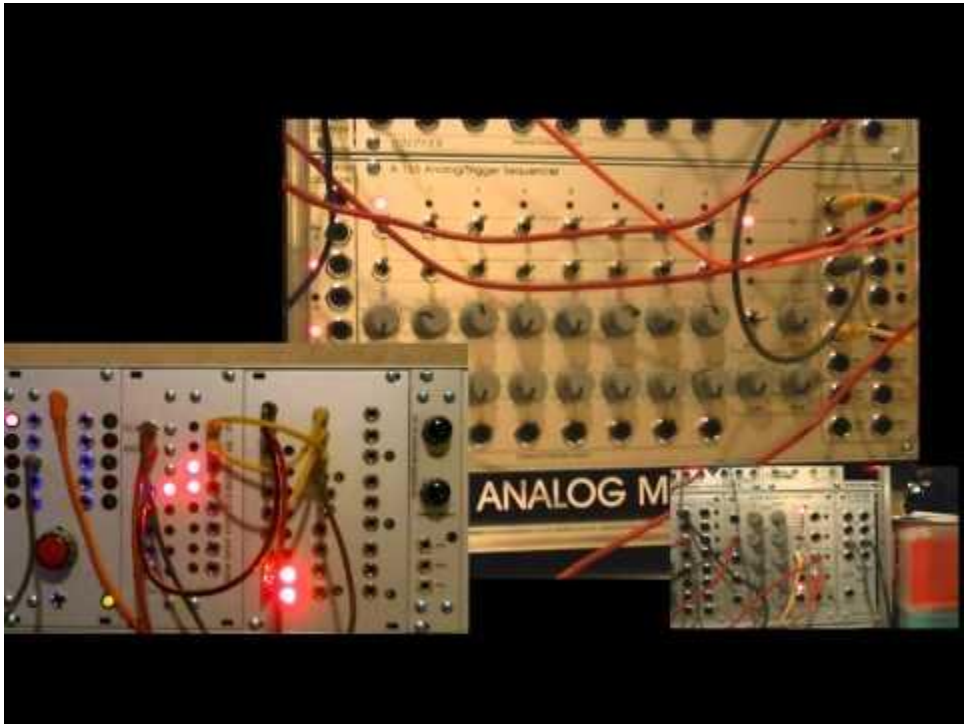
Where can Pd be used?

- Desktop/laptop (Win/OSX/Linux)
- Embedded devices - Bela, Raspberry Pi, Qubit Nebulae (Eurorack), etc.
- Smartphones via LibPD, DroidParty (Android), and PdParty (iOS).
- VST and AU hosts via Camomile.

Sequencing

“[....] programming a set of stored values that can be recalled and sent to any musical destination [....] typically providing note values, rhythms, or articulations” (Farnell, 2010)

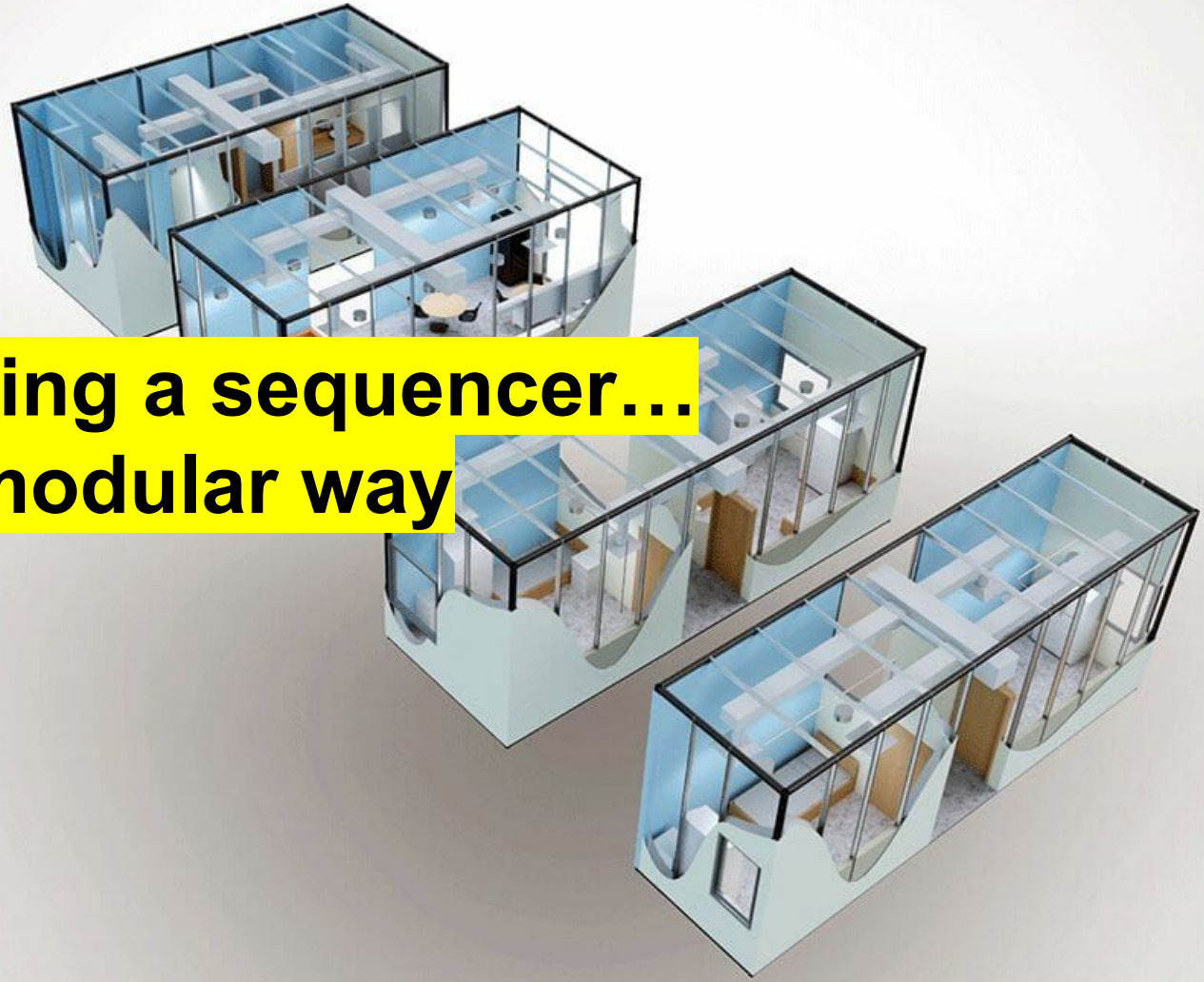
Example: Berlin-style Sequencing



Example: Suzanne Ciani

Example: Spore (EAPd)

Building a sequencer…
in a modular way

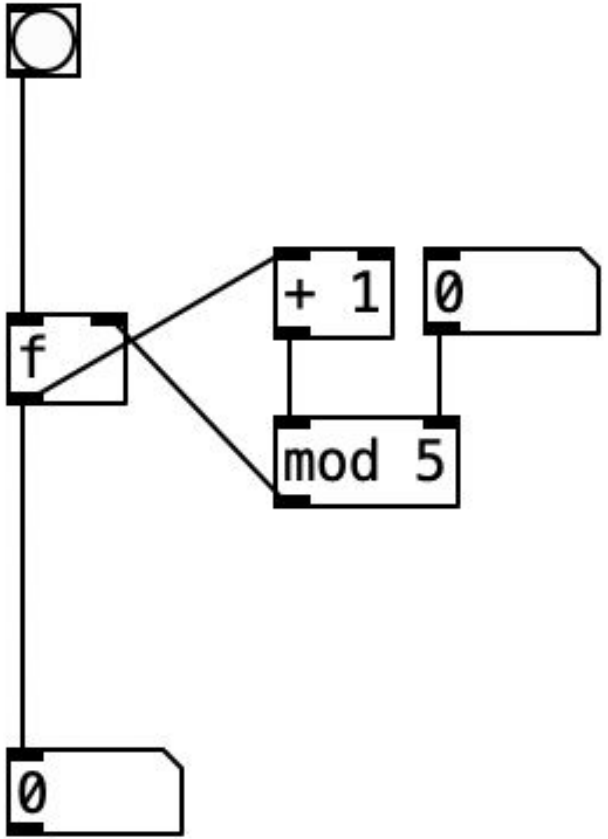


Why Modular?

- Break down more complex systems into simpler functions that are more easily understood.
- Ability to reuse modules - save time/effort in future.
- Development can be largely in parallel.*

* Particularly relevant if working in a team.

Building a Counter



Building a Counter - Key Concepts

- **Hot inlets** - the leftmost inlet of any object is always a hot inlet. Whatever an object receives to the hot inlet will trigger the object and create an output.
- **Cold inlets** - all other inlets are cold inlets. Whatever the object receives to them, it stores as a value, but does not output anything.
- **Modulo operator** - the remainder of dividing two numbers.

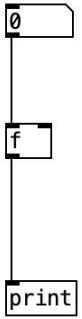
Building a Counter - Development (see Pd patch 1)

1. Number box to HOT inlet of [float]. Value appears at output immediately.
2. Number box to COLD inlet of [f]. Value is stored (no output).
3. Number box to COLD inlet of [f]. Bang to HOT inlet output stored value.
4. Bang outputs the value stored in [f]. That output is passed through [+ 1] and then stored again (infinite counter).

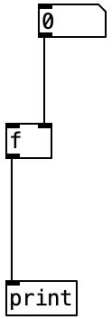
Building a Counter - Development (see Pd patch 1)

5. A modulo function after [+ 1] can be used to set the sequence length. [mod] outputs the remainder of: the counter total divided by the desired sequence length (5 steps in the example).
6. Adding a number box to the cold inlet of [mod] lets us vary the sequence length on-the-fly.

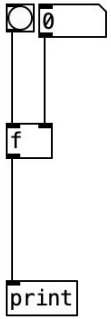
1



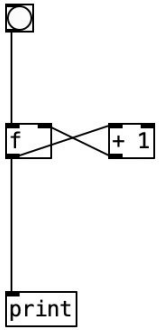
2



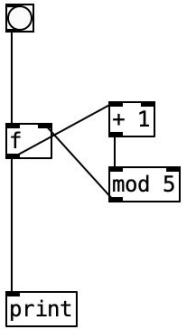
3



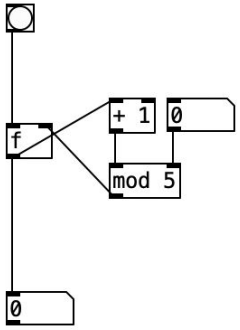
4



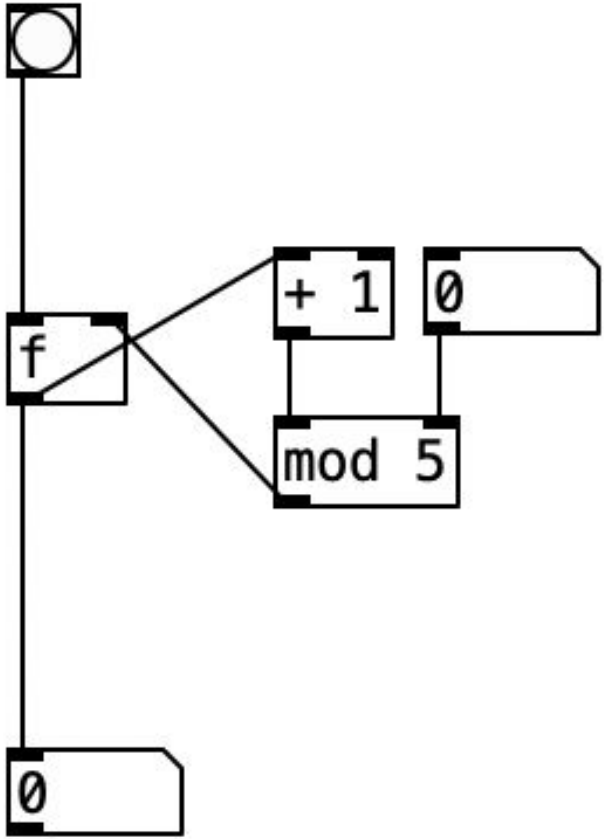
5



6



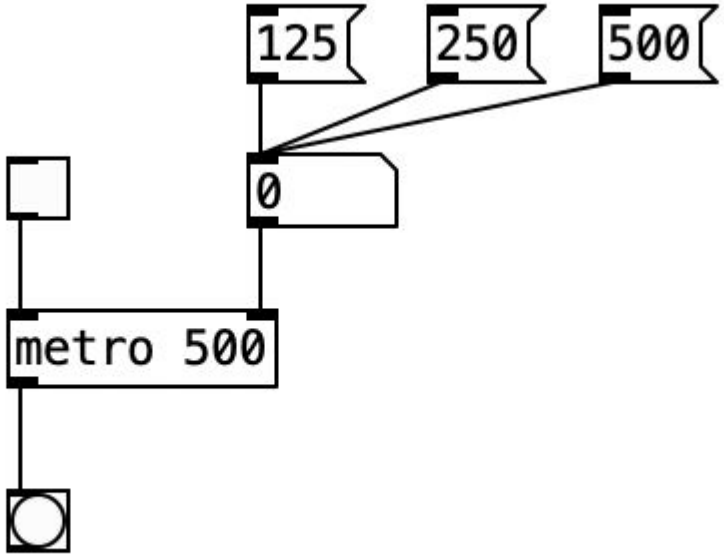
Now, build the counter….



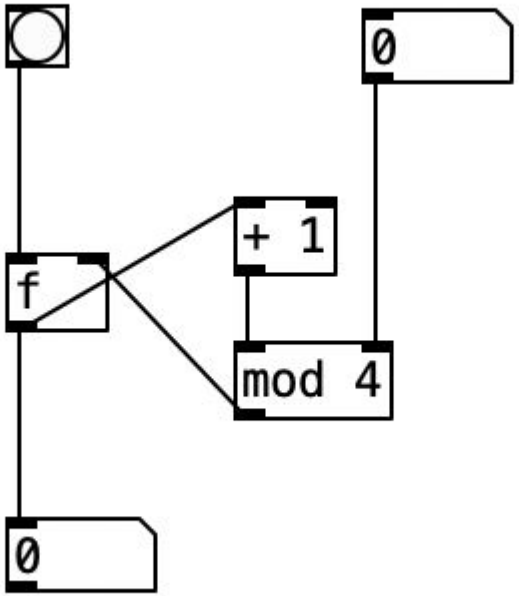
Elements of a Simple Sequencer (see Pd patch 2)

- Metronome (variable rate)
- **Counter**
- Step selection
- MIDI Note generation (or sound generation)

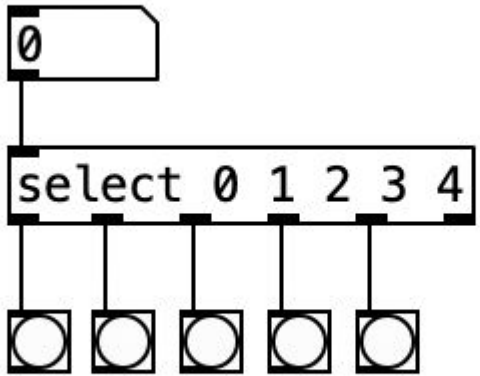
Metronome



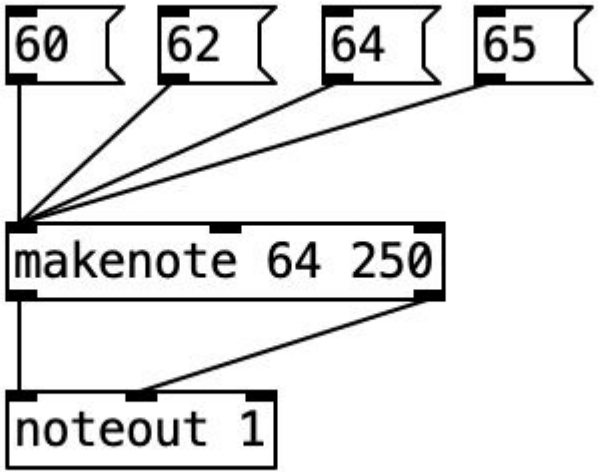
Counter



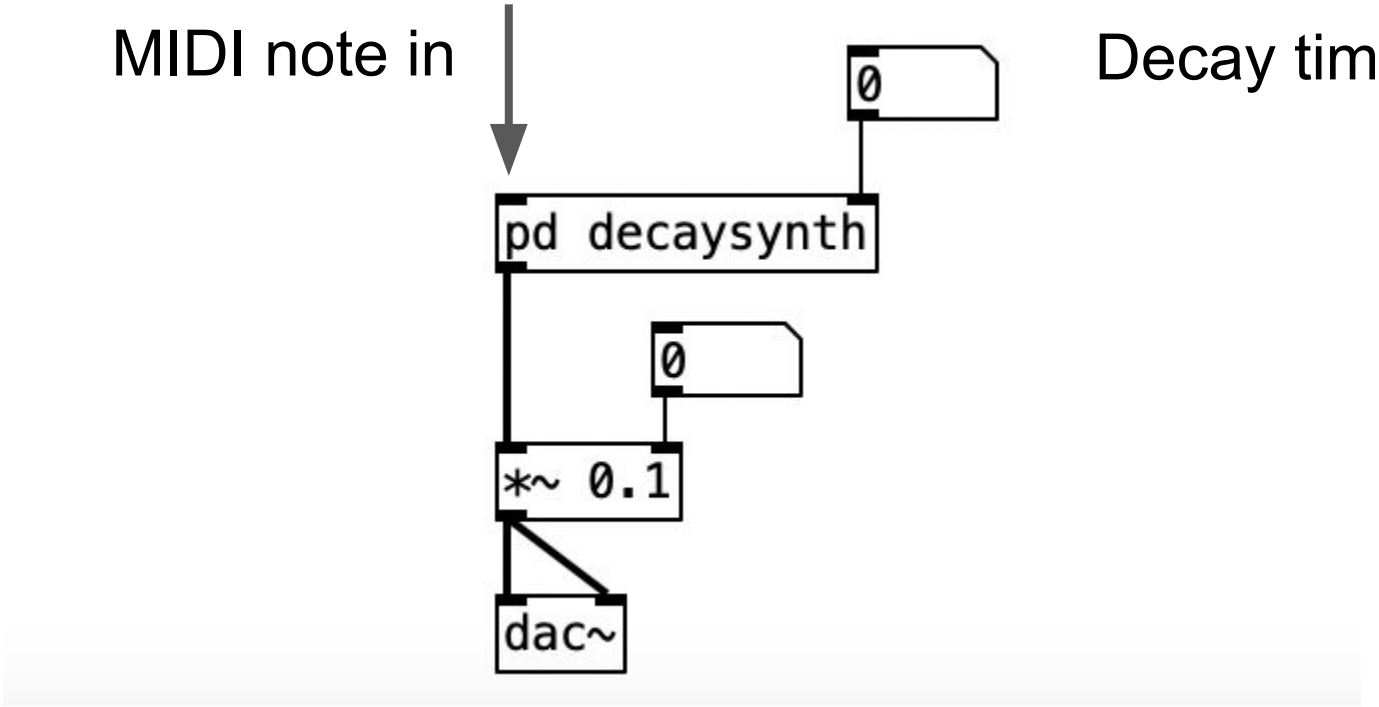
Step Selection



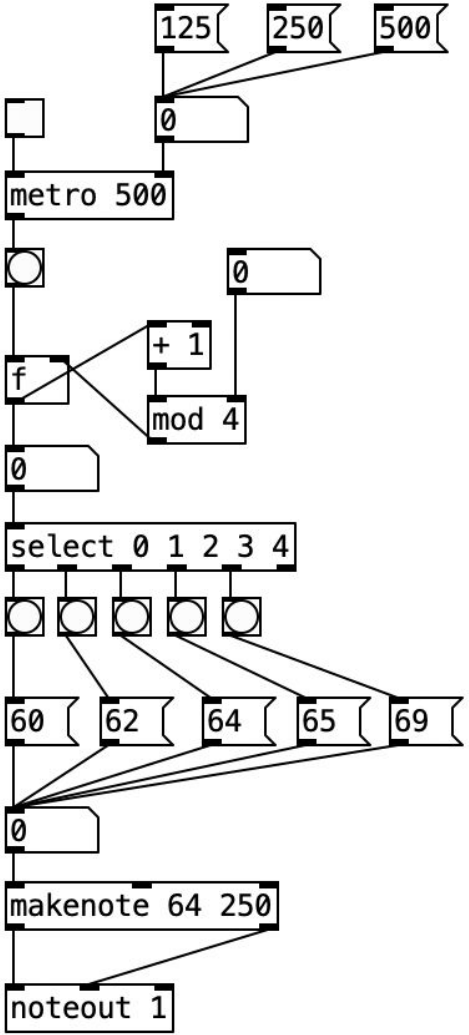
Note Generation

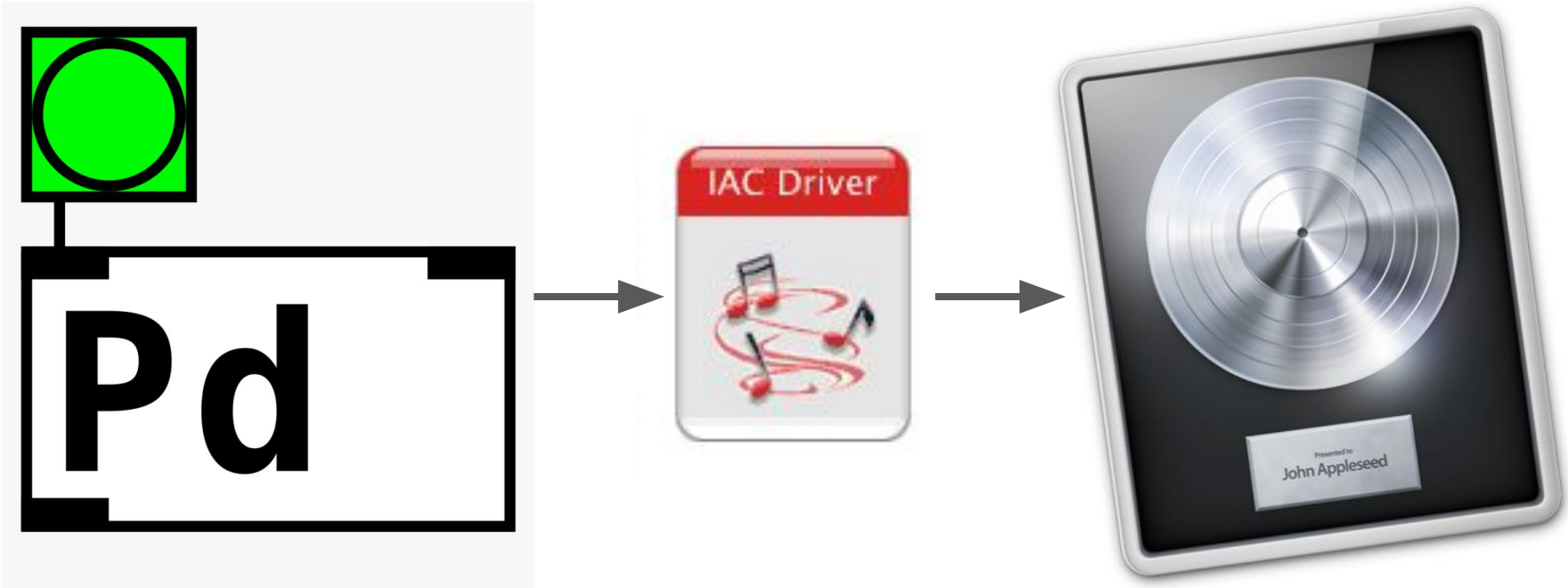


(we could also use simple sound output instead)



Now, put the elements together (see Pd patch 3):




Pd
IAC Driver
Presented to
John Appleseed

IAC Driver

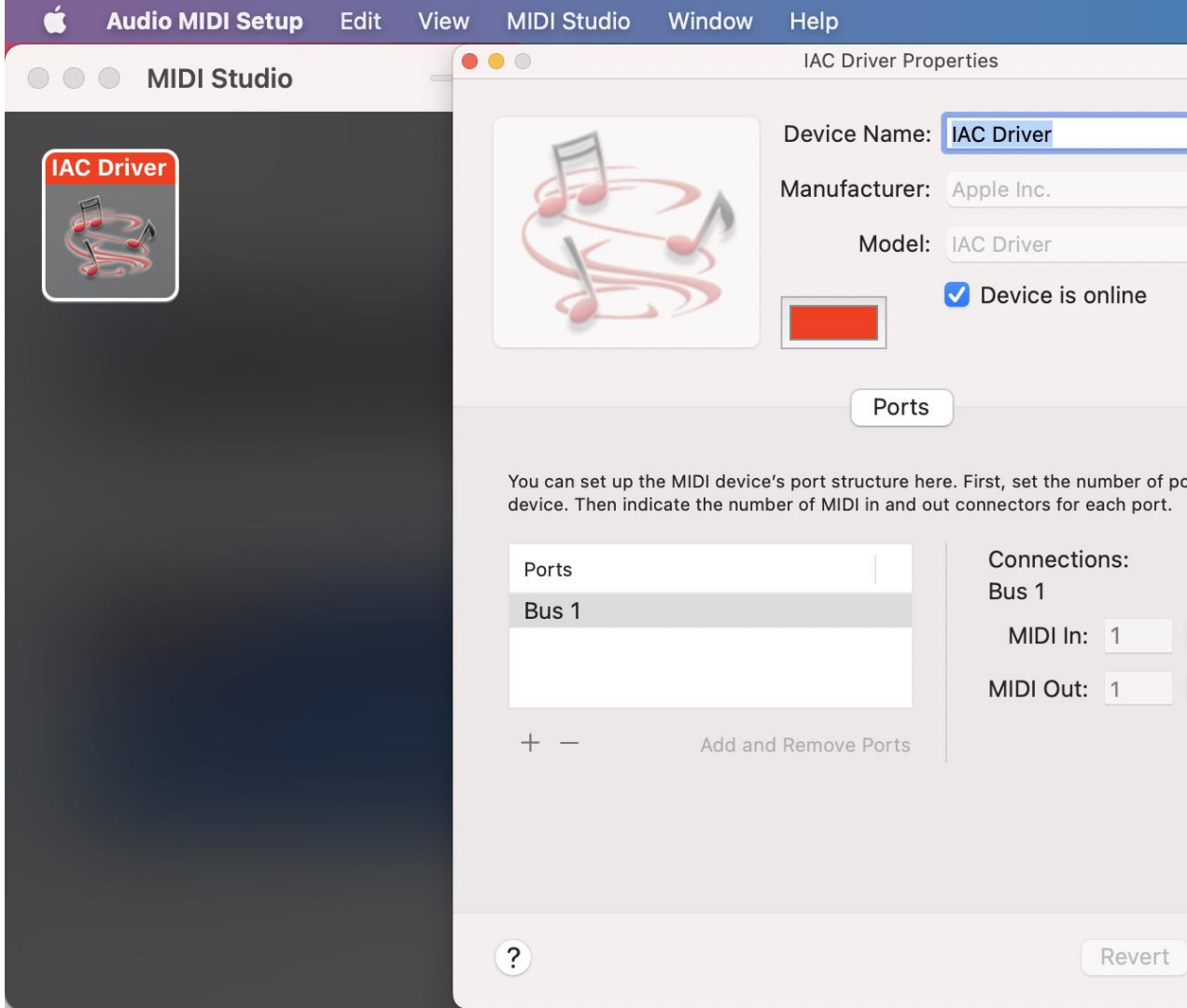
- IAC = Inter-Application Communication
- An inbuilt MIDI device that enables MIDI messages to be routed between applications that support IAC.
- e.g. to route MIDI messages from Pd to Logic.
- Usually offline by default (see next slide)

Go to:

Applications => Utilities

Audio MIDI Setup utility

Ensure Device is online is ticked.

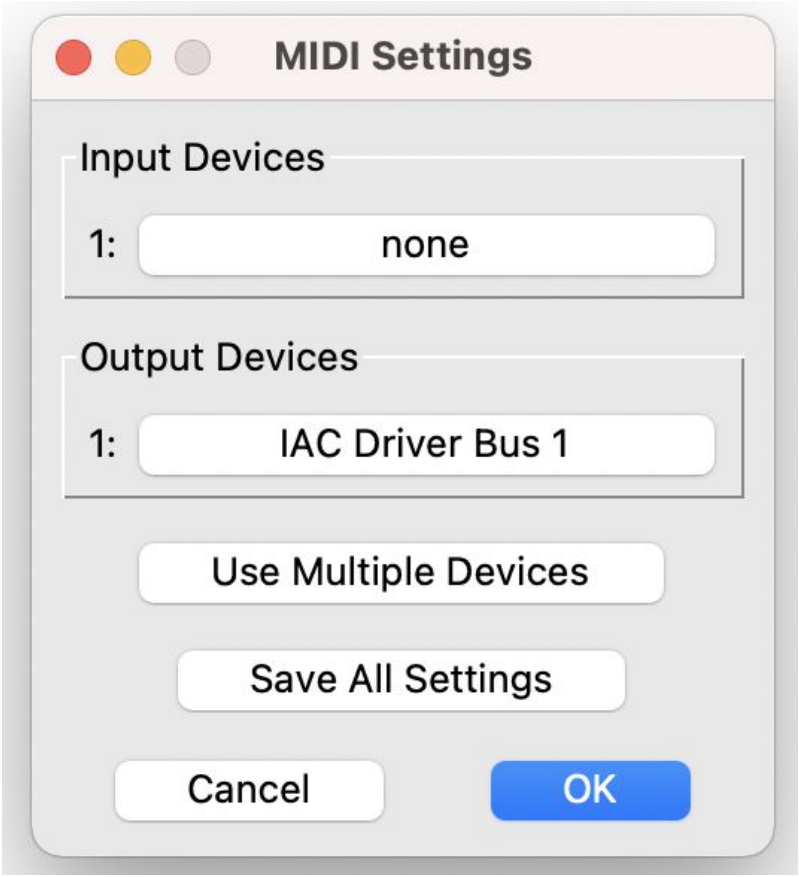


In **Pd**, go to:

Media => MIDI Settings

Set Output Devices (1) to IAC Driver 1

Then click OK (video here)

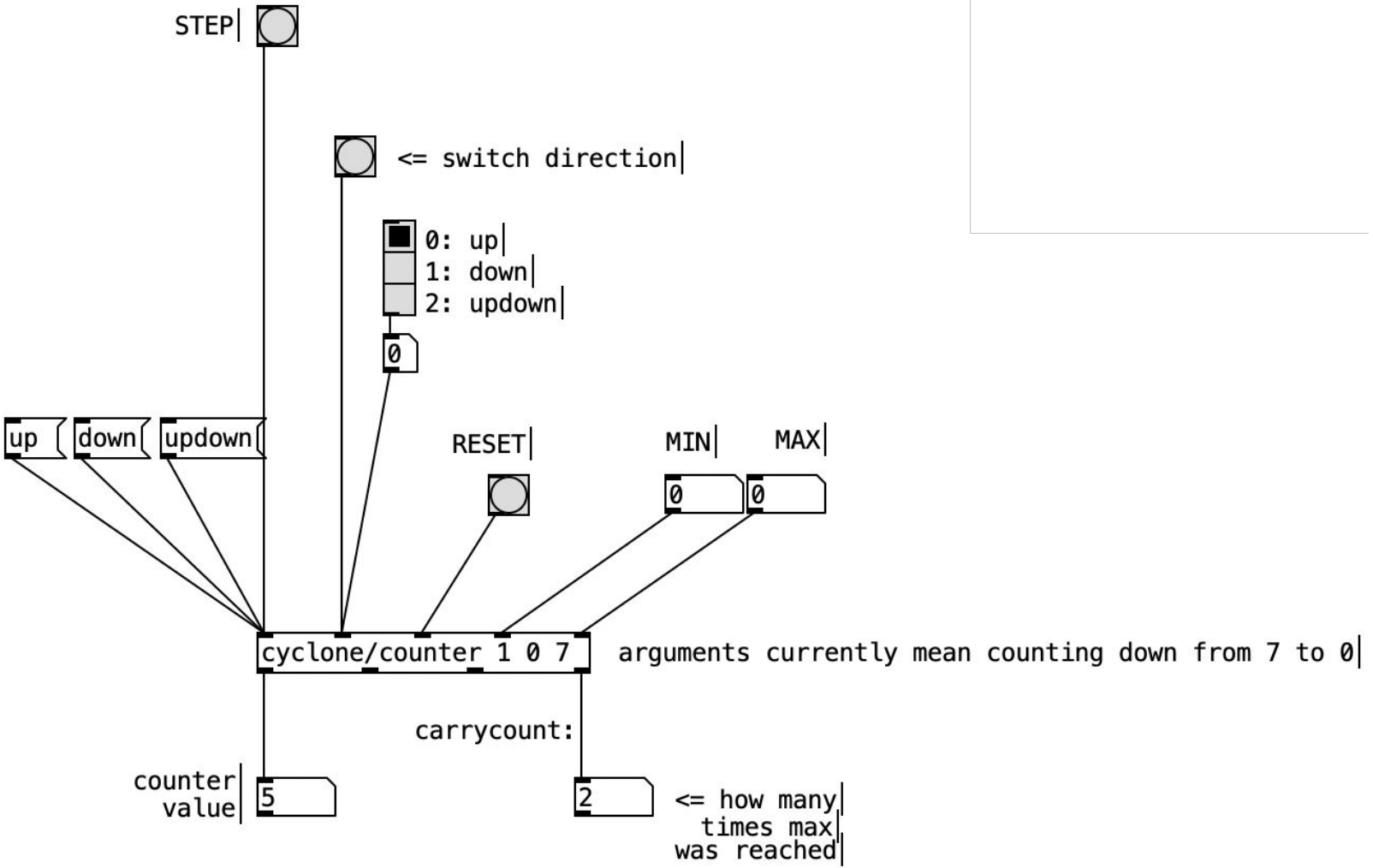


IAC into Logic Pro

- Open the **Logic Pro** application
- File => New Project
- Create a Software Instrument track (video here).
- Experiment with:
  - Manipulating and/or editing the Pd patch
  - The synthesizers available in Logic.

Explore [counter]

- [counter] from the Cyclone library can replace your own counter.
- It adds useful additional functionality:
  - Reset (see Flutter by Autechre)
  - Set/switch direction (up/down/alternating)
  - Set min and max values
  - Count how many times the maximum is reached.


STEP
<= switch direction
0: up
1: down
2: updown
0
up
down
updown
RESET
MIN
MAX
0
0
cyclone/counter 1 0 7
arguments currently mean counting down from 7 to 0
carrycount:
counter value
5
2
<= how many times max was reached

Task (by next week)

- Add at least two extra features (see next slide) to the final Pd patch from today.
- Record a 2 minute video screen capture that talks us through your ideas.
- Upload it to XXXX by XX:XX on XXXX.

Ideas for Further Development

- Subpatch and develop the GUI - foreground main controls, minimise clutter.
- Add multiple rows of sequencing - try sequencing multiple synthesis parameters.
- Use multiple sequencers at the same time - potential for polyrhythms, sequencing the sequencers.
- Your ideas?

Bibliography

Elsea, P. (2018) *Notes on Modular Synthesizers*. Lulu.

Farnell, A. (2010) *Designing Sound.* MIT Press.

Jolly, K. (2011) *Using Pd in Spore and Darkspore* [online]. Available at: https://www.uni-weimar.de/kunst-und-gestaltung/wiki/PDCON:Conference/Using_Pure_Data_in_Spore_and_Darkspore [Accessed 22 June 2022].

Kreidler, J. (2013) *Programming Electronic Music in Pd*. Wolke Publishing House.

Puckette, M. (2007) *Theory and Technique of Electronic Music*. World Scientific Press.