HI

GOALS?

GOALS?

See a music generator built using a VPL.
Learn when, how and why to use a VPL.
Mash the DOM, JS, and the Monome together.
Party.

MONOME & LILY

by Justin Marney

	●	
		●
●	●	●

CONWAY'S GAME OF LIFE

An evolving game played on a grid of cells where each cell is either alive or dead.

CONWAY'S GAME OF LIFE

Choose a start state, apply a set of rules each round.

CONWAY'S GAME OF LIFE

Live cell with fewer than two live neighbors dies.
Live cell with more than three live neighbors dies.
Live cell with two or three live neighbors lives on.
Dead cell with three live neighbors becomes a live cell.

CONWAY'S GAME OF LIFE

These rules lead to intriguing, complex patterns.

AN IDEA!

Use the Game of Life to generate MIDI patterns.
Use the MIDI patterns to trigger a sound generator.
Control the game in real-time with a Monome.
Play the Game of Life like an instrument.

AN IDEA!

Edit the start state on the Monome.

AN IDEA!

Edit the start state on the Monome.
Evolve every n milliseconds.

AN IDEA!

Edit the start state on the Monome.
Evolve every n milliseconds.
Interpret each state as a series of MIDI notes.

AN IDEA!

Edit the start state on the Monome.
Evolve every n milliseconds.
Interpret each state as a series of MIDI notes.
Display each state on the Monome.

AN IDEA!

Edit the start state on the Monome.
Evolve every n milliseconds.
Interpret each state as a series of MIDI notes.
Display each state on the Monome.
Reset the game to the start state every x evolutions.

AN IDEA!

Edit the start state on the Monome.
Evolve every n milliseconds.
Interpret each state as a series of MIDI notes.
Display each state on the Monome.
Reset the game to the start state every x evolutions.
Duplicate the system across multiple MIDI channels.

THE RESULT

A multilayer Game of Life MIDI sequence generator controlled by the Monome triggering an external sound generator.

Do it without writing code?

MIND ALTERING DEMO

HOW IS THIS POSSIBLE?

HOW IS THIS POSSIBLE?

Monome

Visual Programming
(via PureData)

Open Sound Control

FRINGE TECHNOLOGIES?

These tools allow you to **quickly** create a working application with a **usable interface**.

FRINGE TECHNOLOGIES?

They allow you to **mash together** sources from **different domains** with very little effort.

LOTS OF FREEDOM.

VERY LITTLE EFFORT.

THE WEIRD MASHUP

We'll take a look at some tools used to create mashups using unconventional sources.

THE MASHUP TOOLBOX

The physical computing interface called the Monome.
(which is a box of buttons and lights)

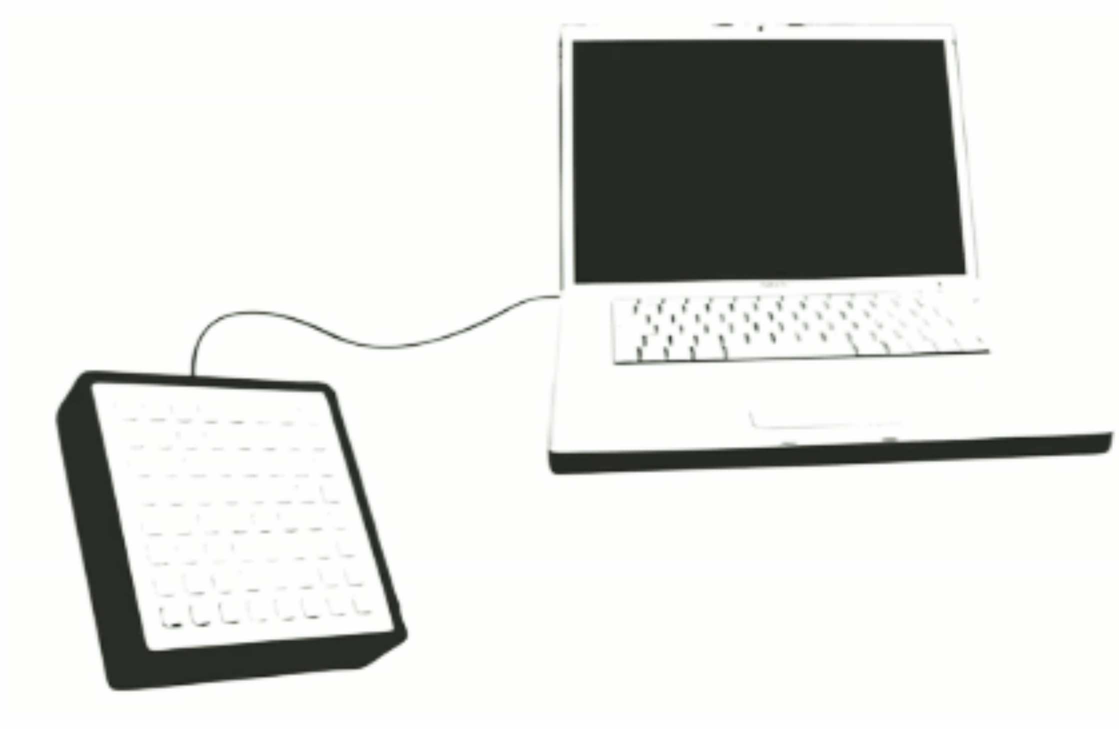
The visual programming language called Lily.
(which is just like PureData but functions in your browser)

A community of smart people and open-source software.

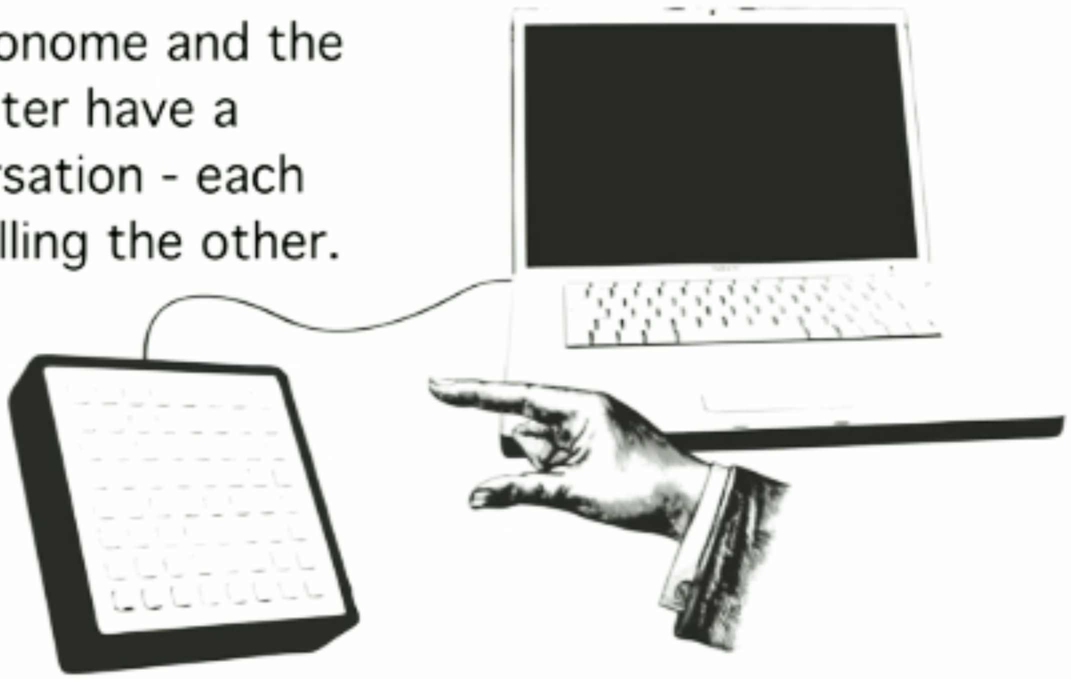
MONOME

A **physical computing interface** with an **open-source hardware** platform, **hand-crafted** by Brian Crabtree and Kelli Cain.

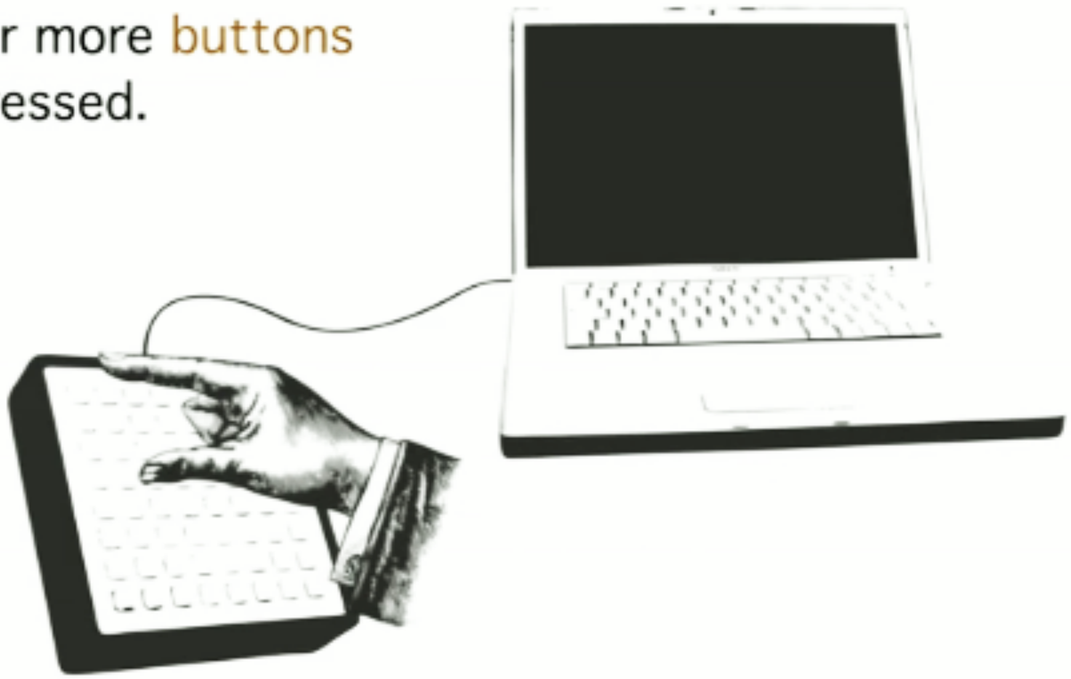
A monome is a reconfigurable grid of backlit keypads which connects to a computer.

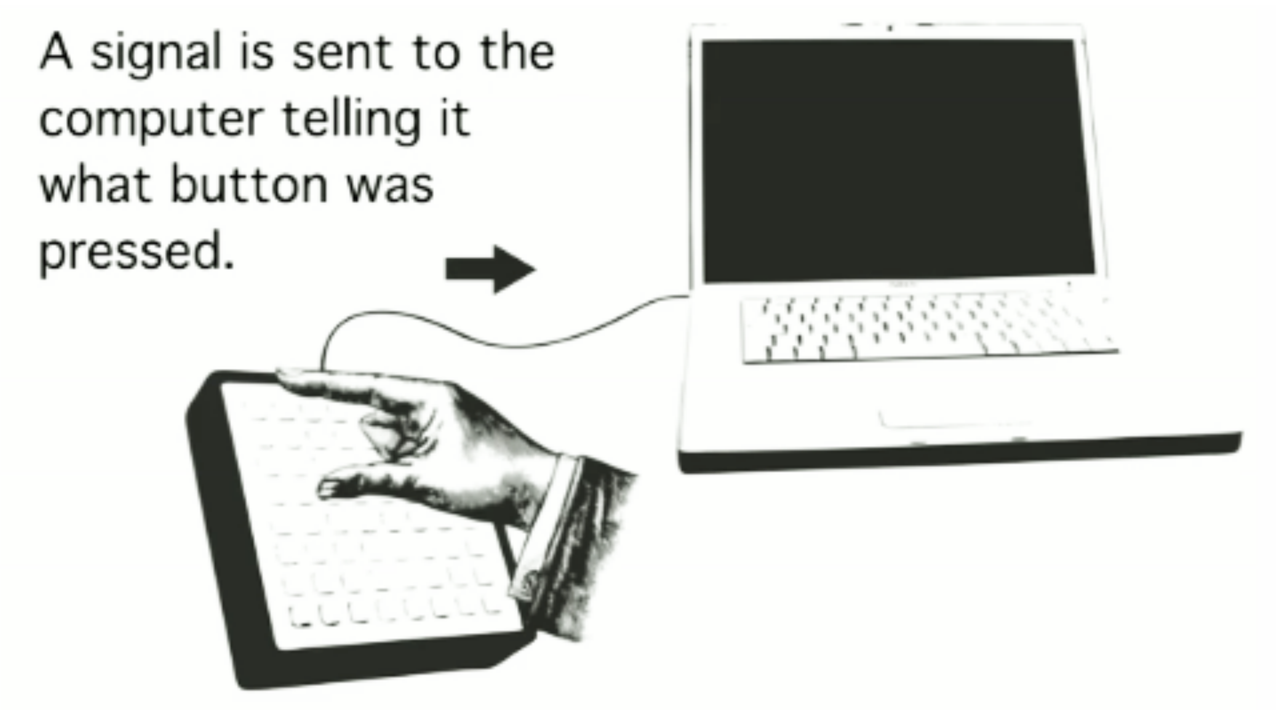


The monome and the computer have a conversation - each controlling the other.



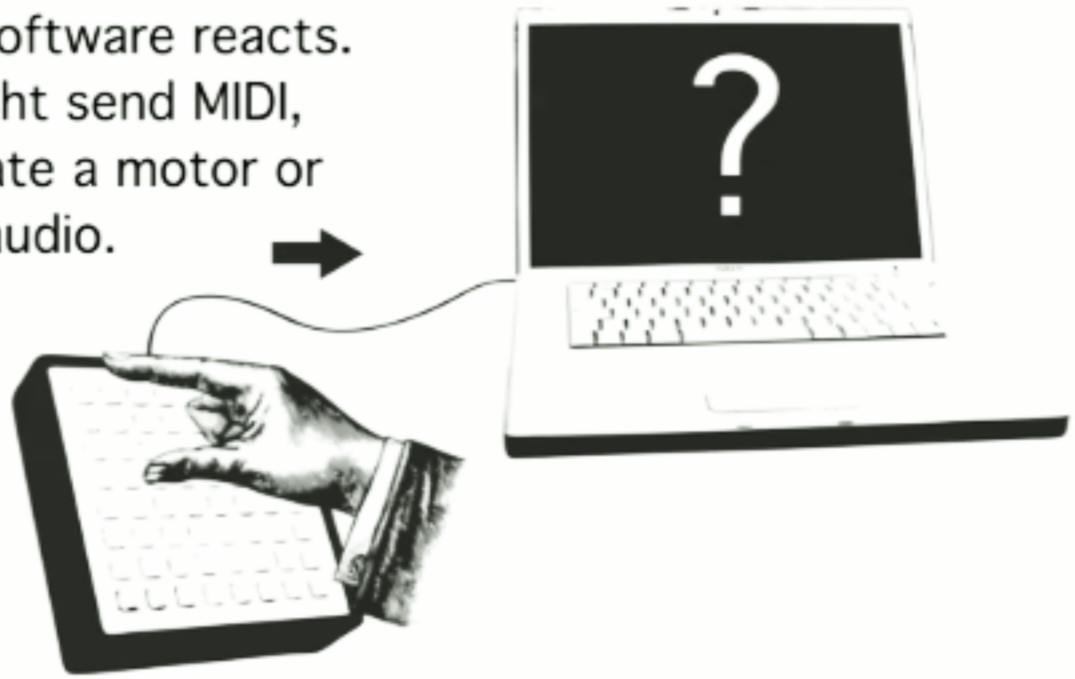
One or more buttons are pressed.

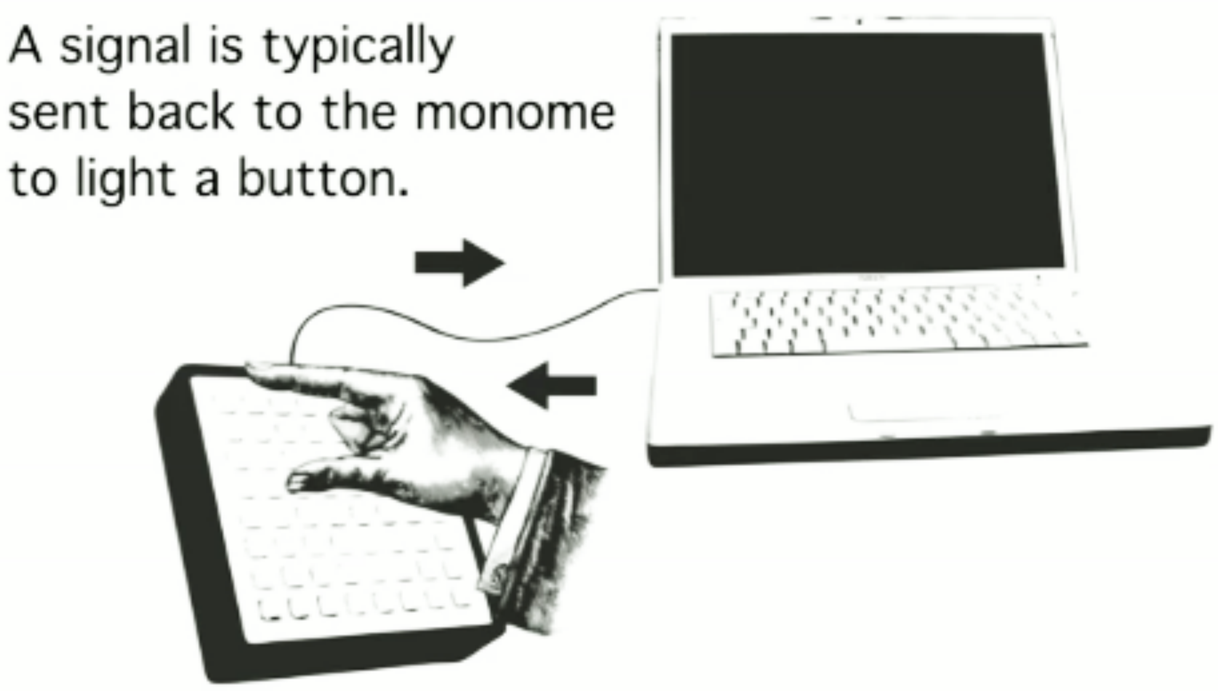




A signal is sent to the computer telling it what button was pressed.

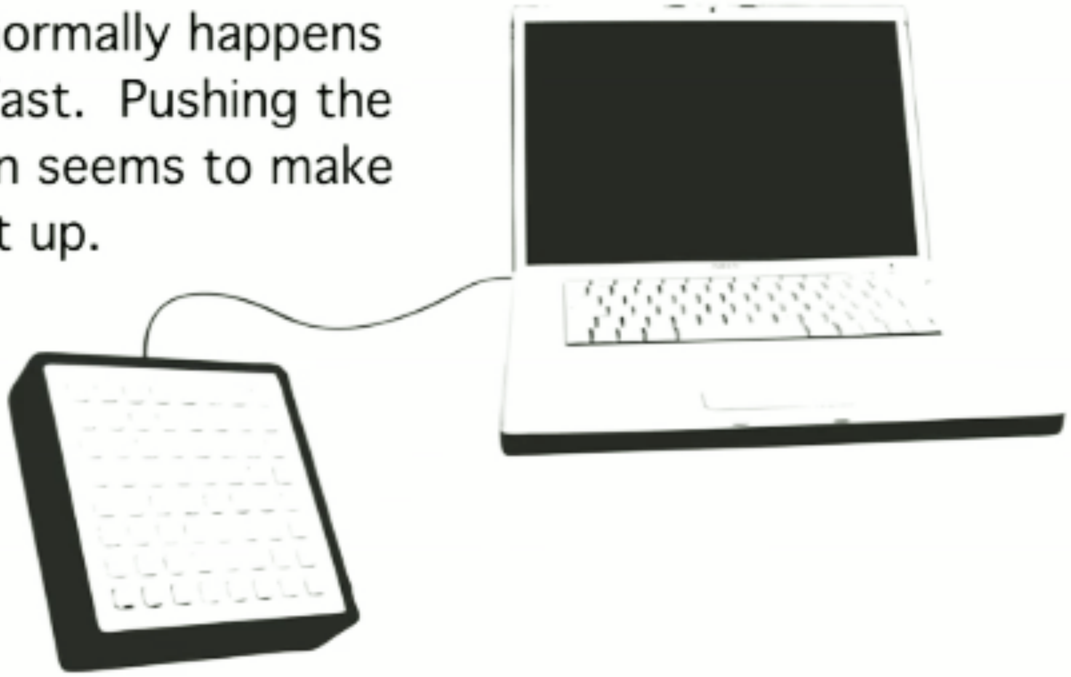
The software reacts.
It might send MIDI,
activate a motor or
play audio.



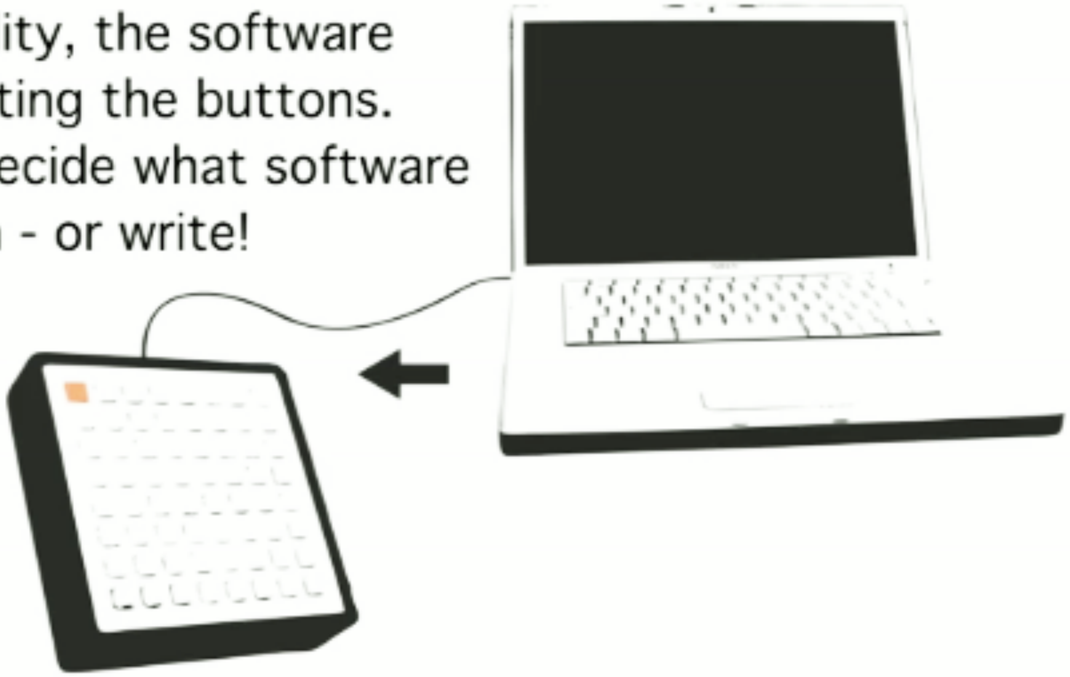


A signal is typically sent back to the monome to light a button.

This normally happens very fast. Pushing the button seems to make it light up.



In reality, the software
is lighting the buttons.
You decide what software
to run - or write!



MONOME

Large community of people creating **open-source** Monome applications using both VPLs and text-based languages.

MONOME

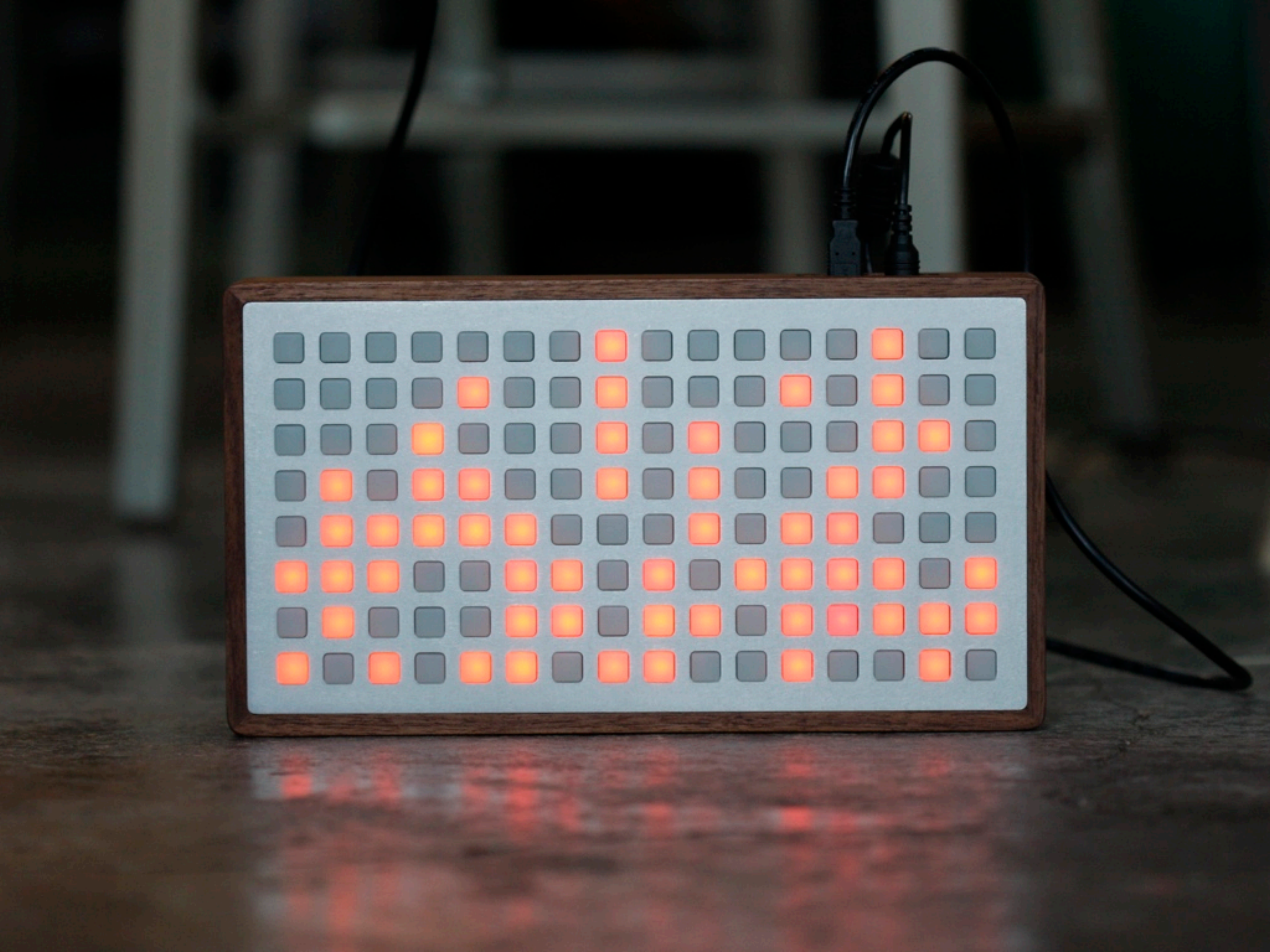
Ruby, Python, Chuck,
Max/MSP, PureData,
SuperCollider, Java,
Processing, Isadora

A lot of languages.

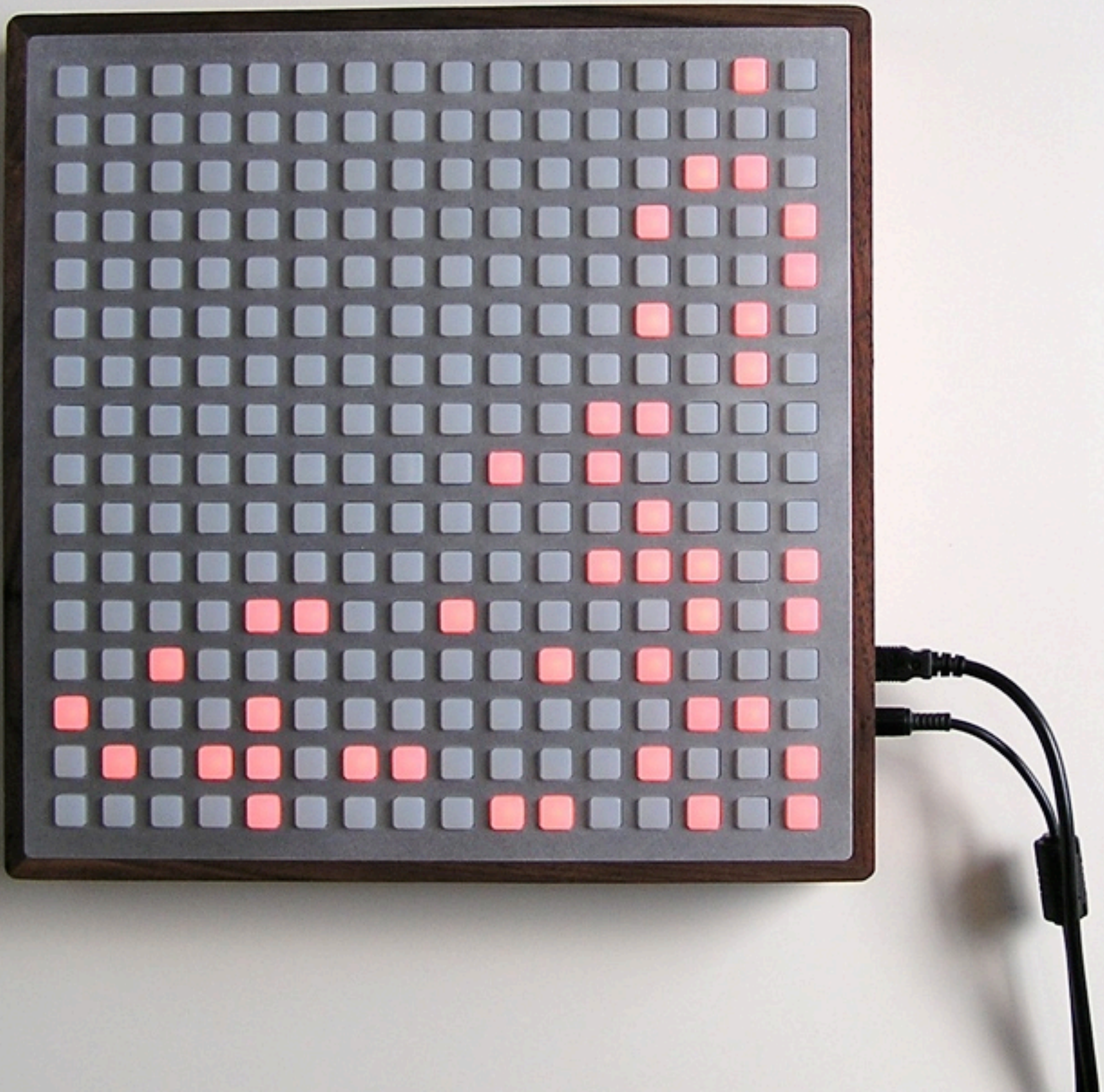
MONOME

Sound generators,
MIDI sequencers,
pixel animators,
games, application
control surfaces,
video controllers

A lot of applications.



$800



$1400

Constraints fuel creativity.

VISUAL PROGRAMMING

Create applications by manipulating graphical elements.
(instead of text)

VPL programs are like big state machines.

THE BASIC IDEA

The fundamental elements of a VPL are **boxes** which implement functionality, and **arrows** which connect boxes and allow them to pass data.

WHY?

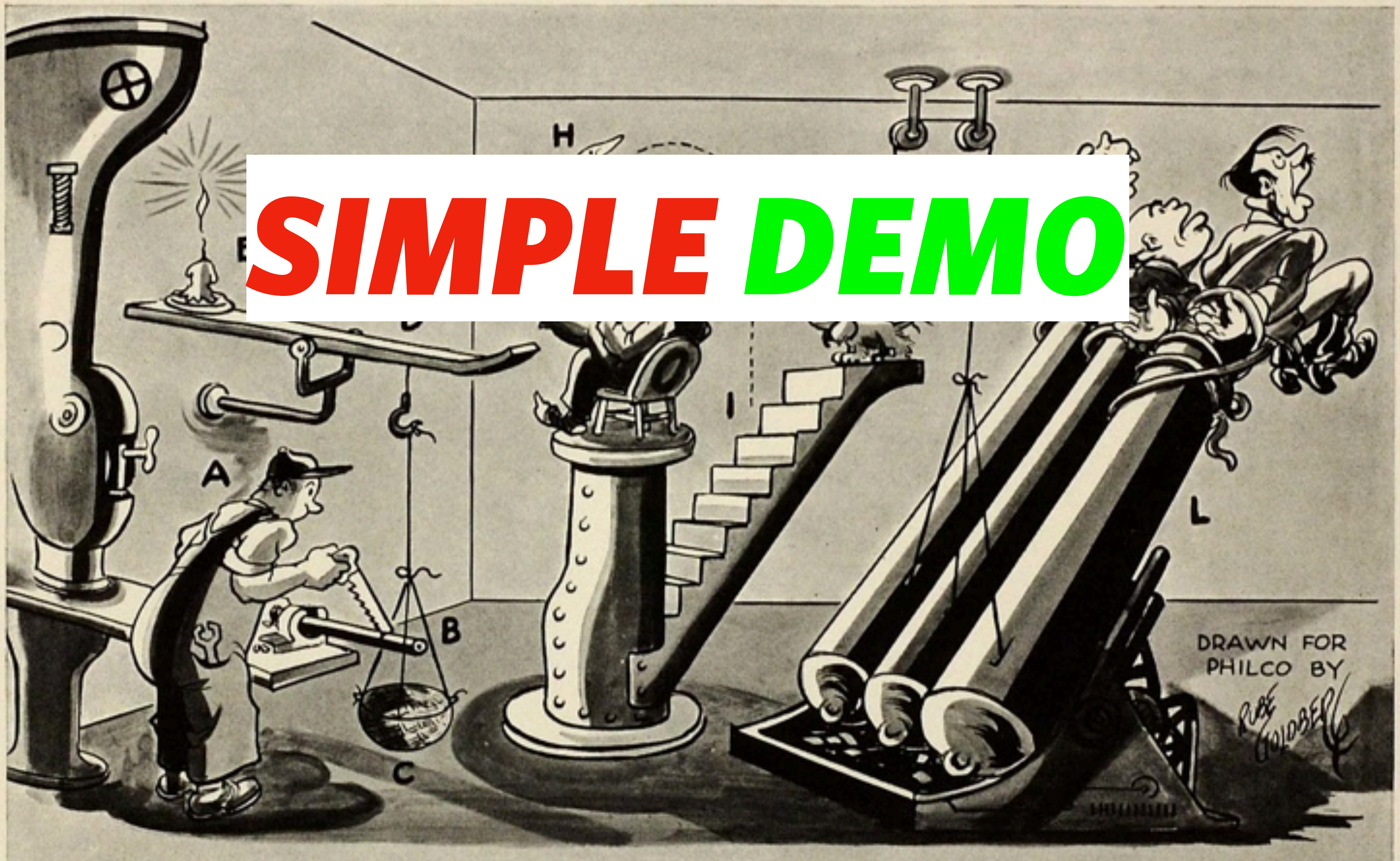
No syntax to learn before creating something useful.
Connected boxes are intuitive and easily understood.
Programs can be manipulated while they are running.
(i.e. during a performance, and the simple box/arrow interface makes this easy to do.)

WHO?

People use visual programming software to create or assist in live performance applications such as audio generators, musical instrument control, art installations, etc.

Rube Goldberg's Latest War Machine!

SIMPLE DEMO



WAR WORKER (A) SAWS OFF END OF IRON PIPE (B), WHICH DROPS INTO BASKET (C)—WEIGHT OF PIPE TIPS SEE-SAW (D), CAUSING LIGHTED CANDLE (E) TO SLIDE DOWN UNDER FOOT (F) OF WATCHMAN (G)—WATCHMAN RAISES FOOT SUDDENLY, KICKING STUFFED DUCK (H), WHICH LANDS ON STAIRS (I)—BIRD-DOG (J) WALKS DOWN STAIRS AFTER DUCK, PULLING STRING (K), WHICH SETS OFF BIG GUNS (L) AND BLOWS ADOLF, MUSSO AND HIRO SO FAR THEY'LL NEVER TROUBLE THE CIVILIZED WORLD AGAIN!

301

LILY

Developed by Bill Orcutt.
Installs and runs as a Firefox Add-On.
Built using Chrome and Javascript.
Open-source.

YOU CAN

Create a patch that opens a URL.
Analyze the DOM elements of that page.
Manipulate the DOM elements.
(mini demo!)

Lily is a Javascript application.

Patches run on the JS engine.

Externals are Javascript classes.

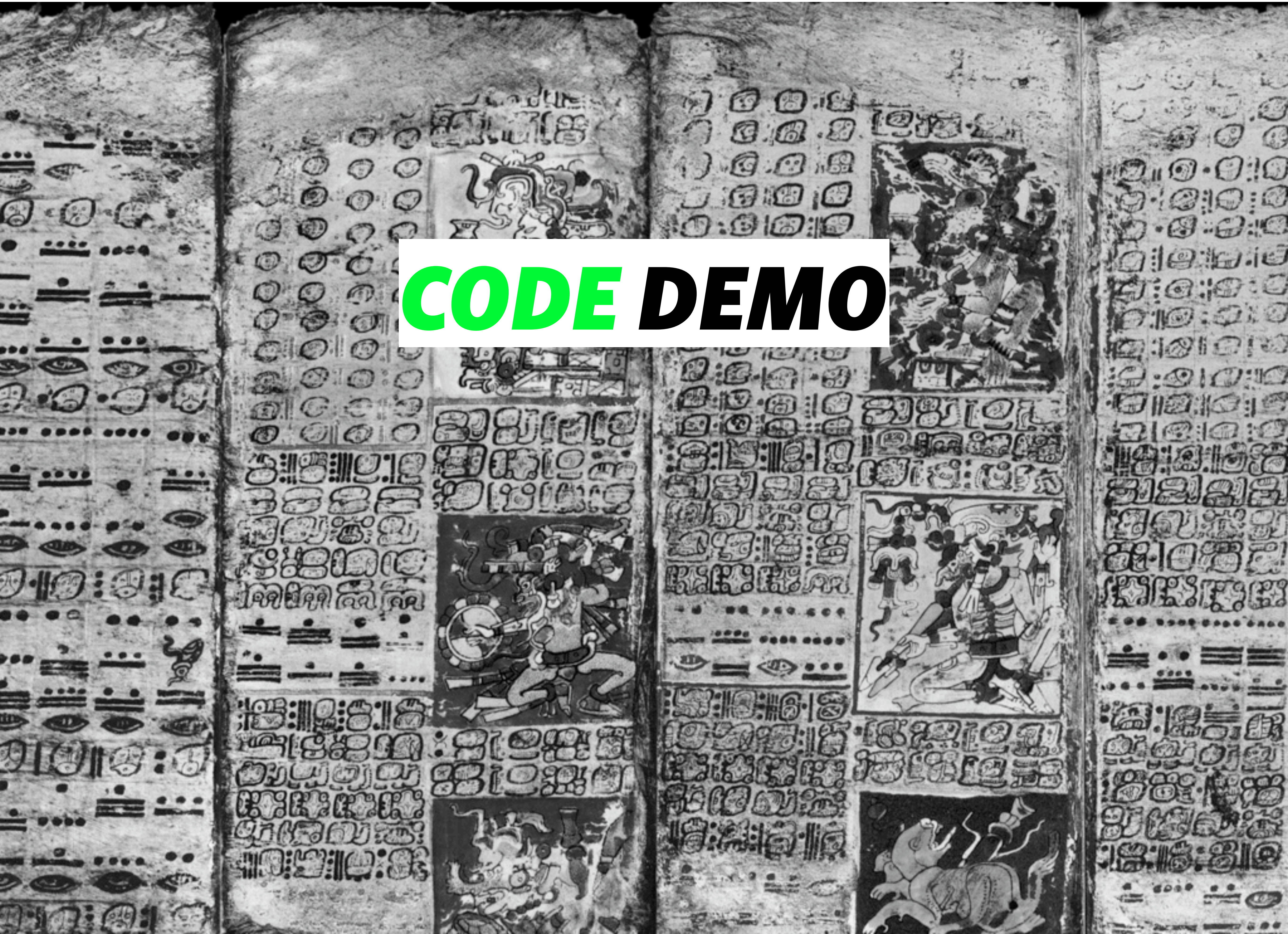
YOU CAN

Create new externals in Lily using Javascript.
Use jQuery, AJAX, or any other JS libs from within Lily.
Save your Lily patches as Firefox Add-Ons.

I thought you said no code?

I thought you said no code?

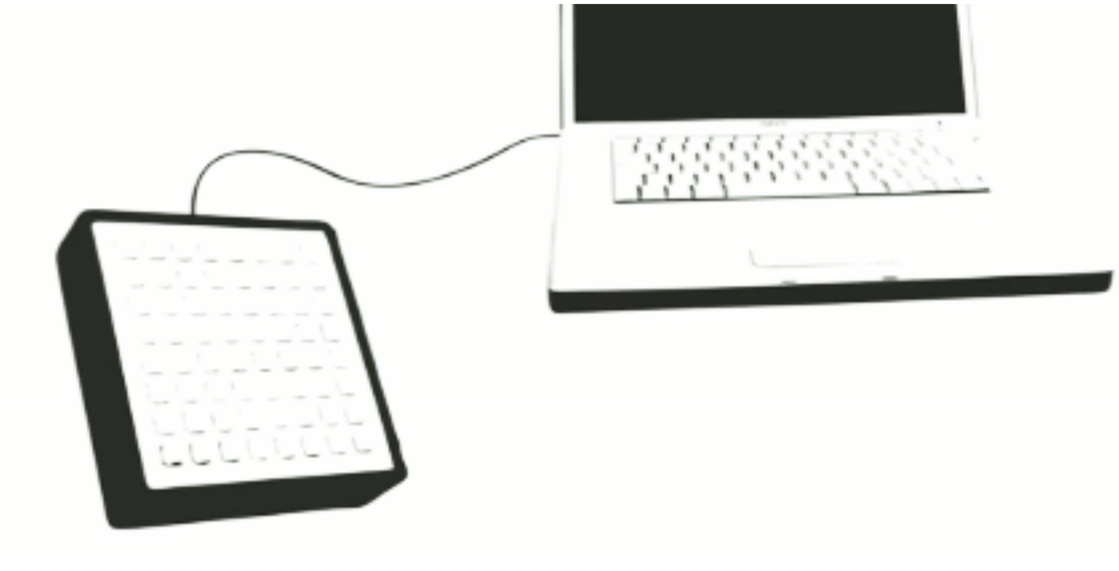
(If you are here at **developer day** and are really excited by the idea of not writing code, see me after class.)

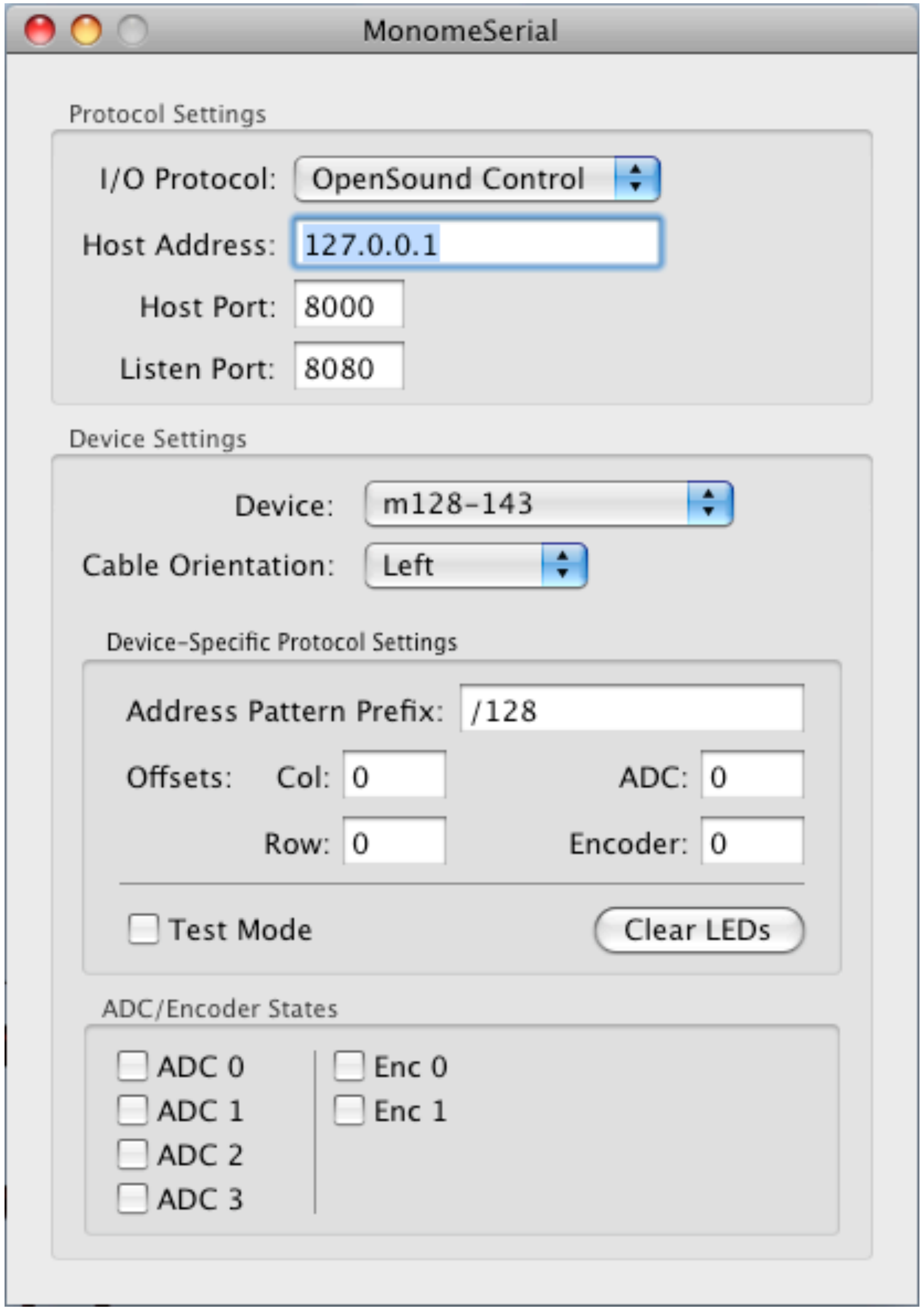


CODE DEMO

MonomeMonomeserialOSCLily

The Monome transmits data to the computer via a serial protocol.





MonomeSerial

Protocol Settings

I/O Protocol: OpenSound Control

Host Address: 127.0.0.1

Host Port: 8000

Listen Port: 8080

Device Settings

Device: m128-143

Cable Orientation: Left

Device-Specific Protocol Settings

Address Pattern Prefix: /128

Offsets: Col: 0 ADC: 0

Row: 0 Encoder: 0

□ Test Mode

ADC/Encoder States

□ ADC 0
□ ADC 1
□ ADC 2
□ ADC 3

□ Enc 0
□ Enc 1

OPEN SOUND CONTROL

A network communication protocol that uses URL-style naming.

MONOME OSC PROTOCOL

When a button is **pressed** Monomeserial sends **/name/press x y 1** to the host IP and port.

(The name, IP & port are configured in Monomeserial. x, y is the coordinate of the button pressed.)

MONOME OSC PROTOCOL

When a button is **pressed** Monomeserial sends **/name/press x y 1** to the host IP and port.

(The name, IP & port are configured in Monomeserial. x, y is the coordinate of the button pressed.)

When a button is **released** Monomeserial sends **/name/press x y 0** to the host IP and port.

MONOME OSC PROTOCOL

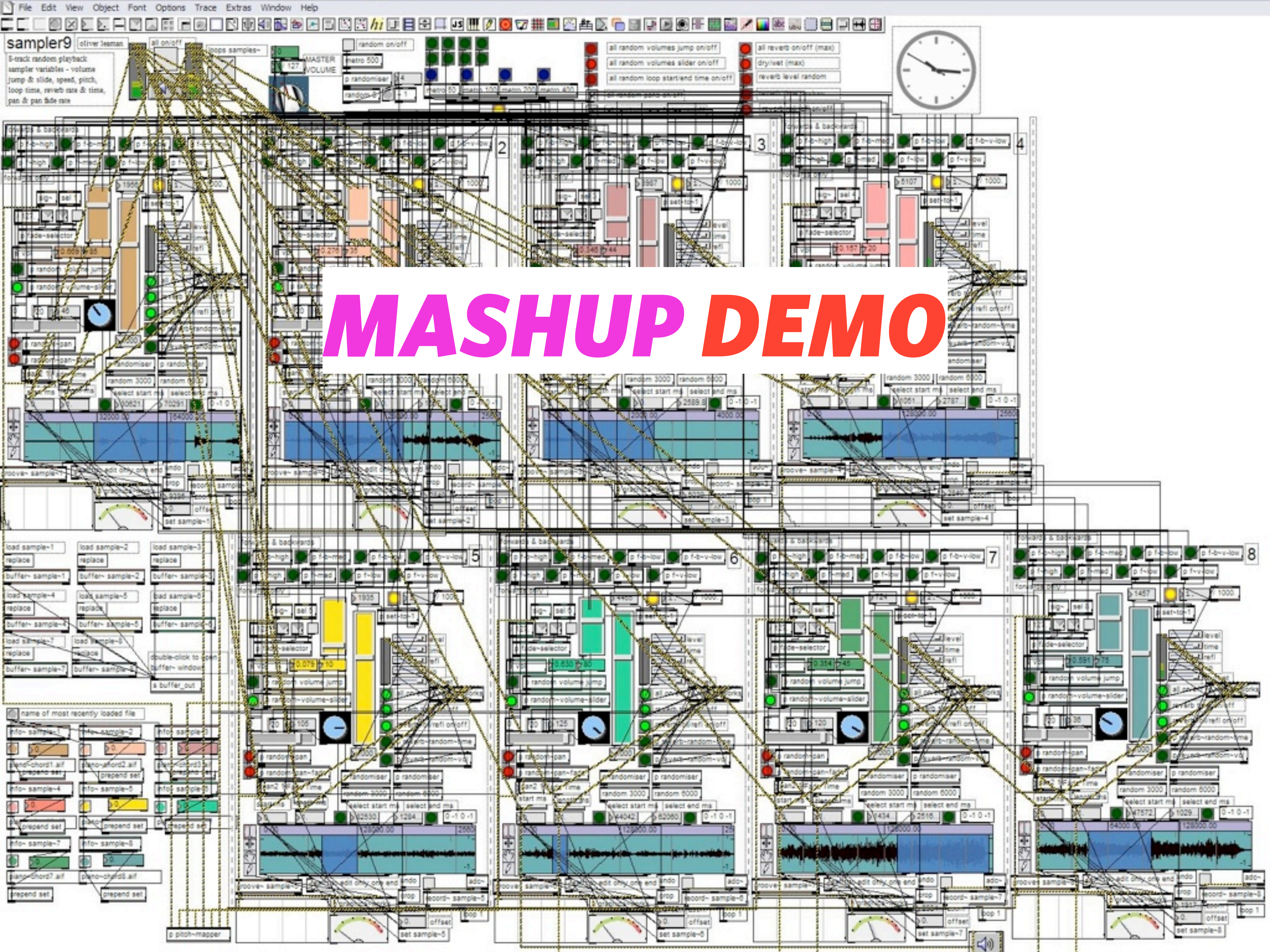
Sending **/name/led x y 1** to the listen port will **light** the led at coordinate x, y.

MONOME OSC PROTOCOL

Sending **/name/led x y 1** to the listen port will **light** the led at coordinate x, y.
Sending **/name/led x y 0** to the listen port will **turn off** the led at coordinate x, y.
(The listen port is configured in Monomeserial.)

OSC IN LILY

oscsend and **oscreceive** externals.


File Edit View Object Font Options Trace Extras Window Help
sampler9
oliver leaman
8-track random playback sampler variables - volume jump & slide, speed, pitch, loop time, reverb rate & time, pan & pan fade rate
all on/off
loops samples
MASTER VOLUME
random on/off
metro 500
p randomiser
metro 50
metro 100
metro 200
metro 400
all random volumes jump on/off
all random volumes slider on/off
all random loop start/end time on/off
all reverb on/off (max)
dry/wet (max)
reverb level random
forwards & backwards
p f-b-high
p f-b-med
p f-b-low
p f-b-v-low
p f-high
p f-med
p f-low
p f-v-low
2
3
4
5
6
7
8
sig~
sel 1
set-to-1
fade-selector
random volume jump
random-volume-slider
random-pan
random-pan-fader
p randomiser
random 3000
random 6000
select start ms
select end ms
groove~ sample-1
record~ sample-1
set sample-1
offset
load sample-1
replace
buffer~ sample-1
double-click to open buffer~ window
s buffer_out
name of most recently loaded file
info~ sample-1
piano-chord1.aif
prepend set
p pitch-mapper
level
time
refl
reverb
dac~

MASHUP DEMO

RESOURCES

http://gotascii.com/2009/6/2/monome-lily
http://github.com/gotascii/sonifier
http://github.com/gotascii/thunderdome
http://lilyapp.org
http://monome.org
http://cycling74.com
http://puredata.info