

# Designing Interactive Systems 2

## Lecture 13: Software Prototyping

Prof. Dr. Jan Borchers  
Media Computing Group  
RWTH Aachen University

[hci.rwth-aachen.de/dis2](http://hci.rwth-aachen.de/dis2)



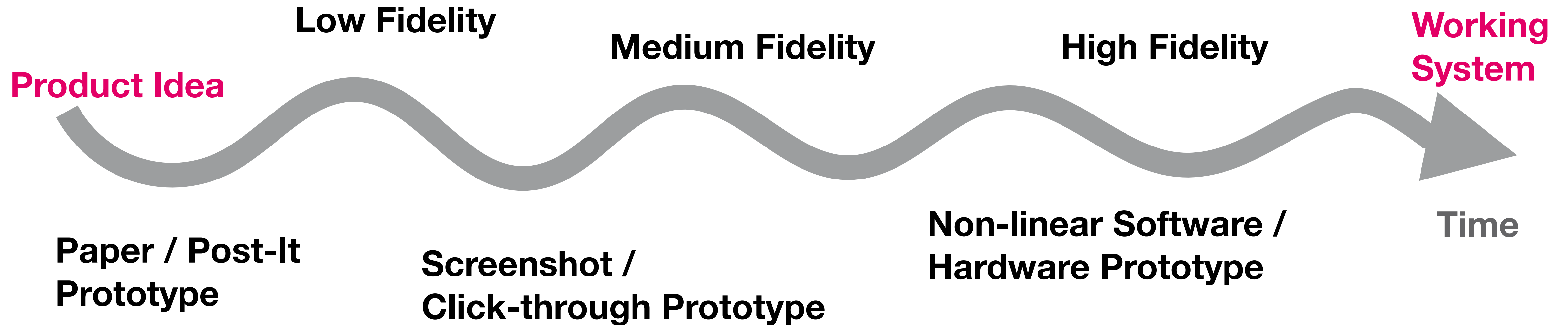
**RWTH**AACHEN  
UNIVERSITY

## CHAPTER 38

# Software Prototyping



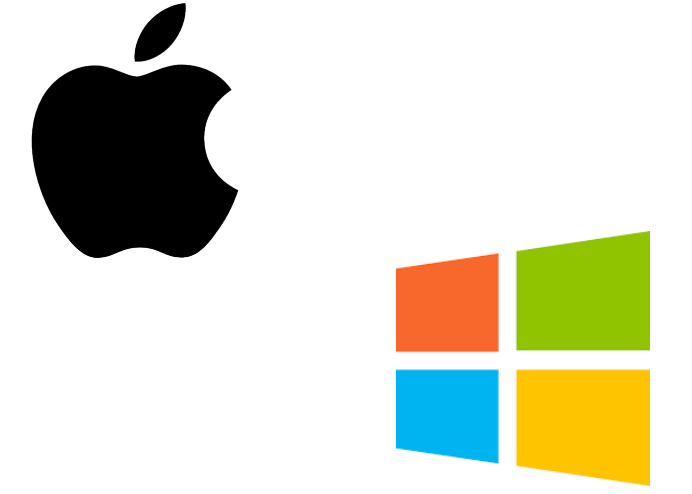
# Prototyping Stages (DIS1)



## CHAPTER 39

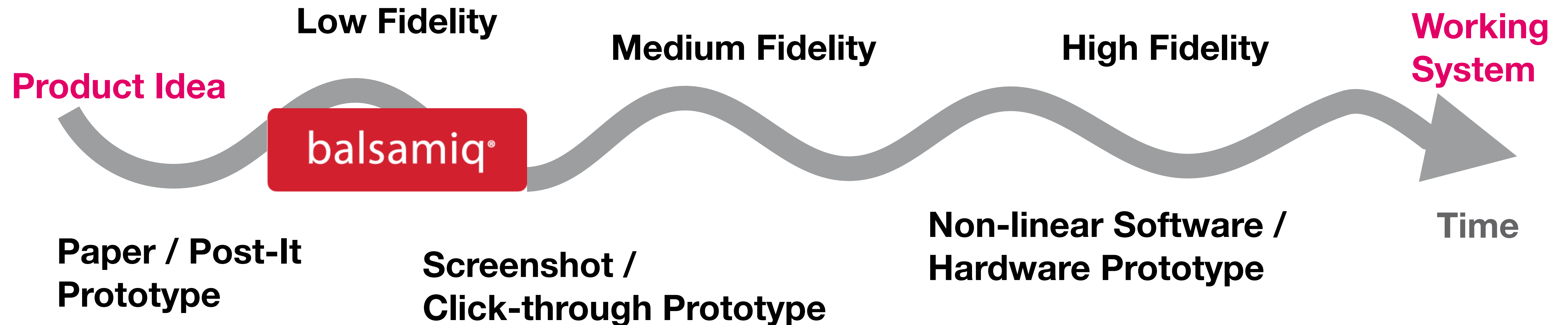
# Prototyping Standard Apps

# balsamiq®



- Wireframing tool
- Design mockups
- First released in 2008
- **WYSIWYG** editor

# When to Use



Navigator Panel

Toolbar

UI Library

Canvas

Properties Panel

The screenshot shows the Balsamiq Mockups application interface. At the top is a dark toolbar with icons for undo, redo, copy, paste, and other functions. Below the toolbar is a horizontal row of tabs for the UI Library, including All, Assets, Big, Buttons, Common, Containers, Forms, Icons, iOS, Layout, Markup, Media, Symbols, and Text. The main canvas area is a grid with the text "Mockups is Easy". To the left is the Navigator Panel, which shows a tree view of the project structure, including a "Welcome" screen, a "Mockups is Easy" screen, and a "Site Map". To the right is the Properties Panel, which shows the properties of the selected element, including Position (X: 217, Y: 365), Size (width: 137, height: 39), Layering, Auto-Size, Links, and Text. Annotations in yellow boxes with red arrows point to various parts of the interface: "Start by adding controls from the UI Library" points to the UI Library tabs; "Double-click to edit the controls you add" points to a button in the UI Library; "Open the inspector panel to change properties like the selected tab" points to the Properties Panel; "See what else you can build with Balsamiq" points to the Navigator Panel. A red arrow also points from the "Mockups is Easy" text on the canvas to the "Mockups is Easy" screen in the Navigator Panel.

Mockups is Easy

Start by adding controls from the UI Library

Double-click to edit the controls you add

Open the inspector panel to change properties like the selected tab

See what else you can build with Balsamiq

Mockups is Easy

Welcome to Balsamiq Mockups!

Welcome

Adding controls

Site Map

Radio Button Group

Position

217 X

365 Y

Size

137 width

39 height

Layering

Auto-Size

Links

Hide

(o) option 1 (... No Link

( ) option 2 No Link

Text

B i u 13

**Demo:** balsamiq®

# Balsamiq: Features

- Focused on structure rather than colors and icons
- For rough wireframes
- Library of UI elements
- Can design widgets



# CHAPTER 40

# Sketching



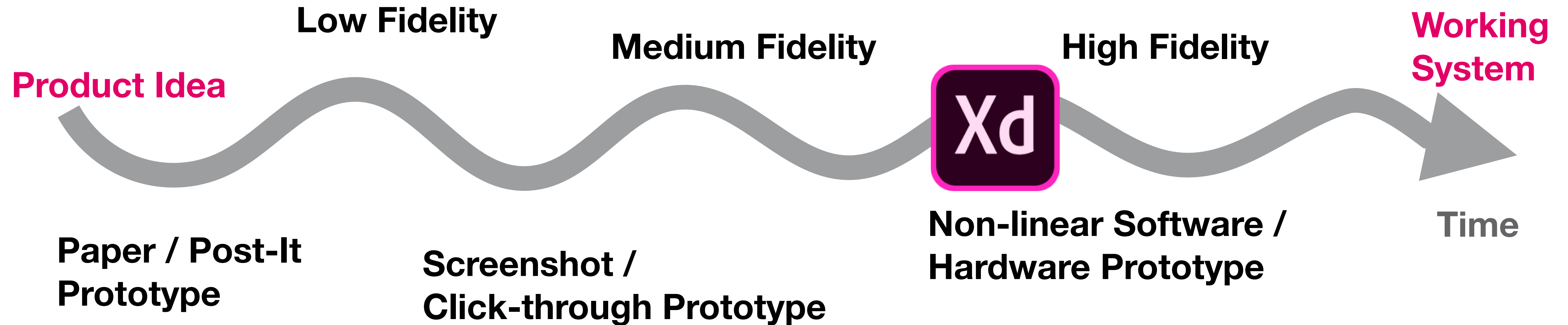


# Adobe Xd

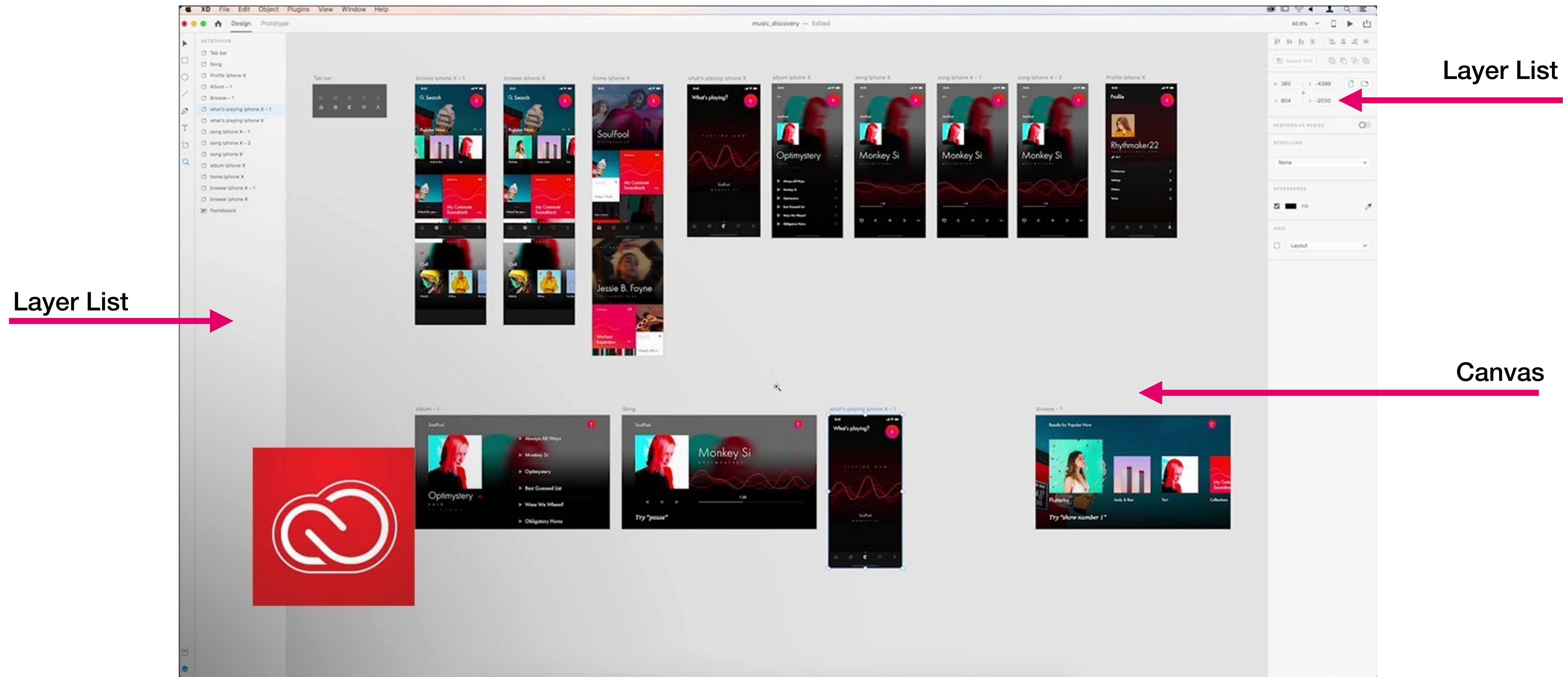
- UI sketching and prototyping tool
- Create user interfaces for mobile and web applications
- First released in 2017
- Free



# When to Use



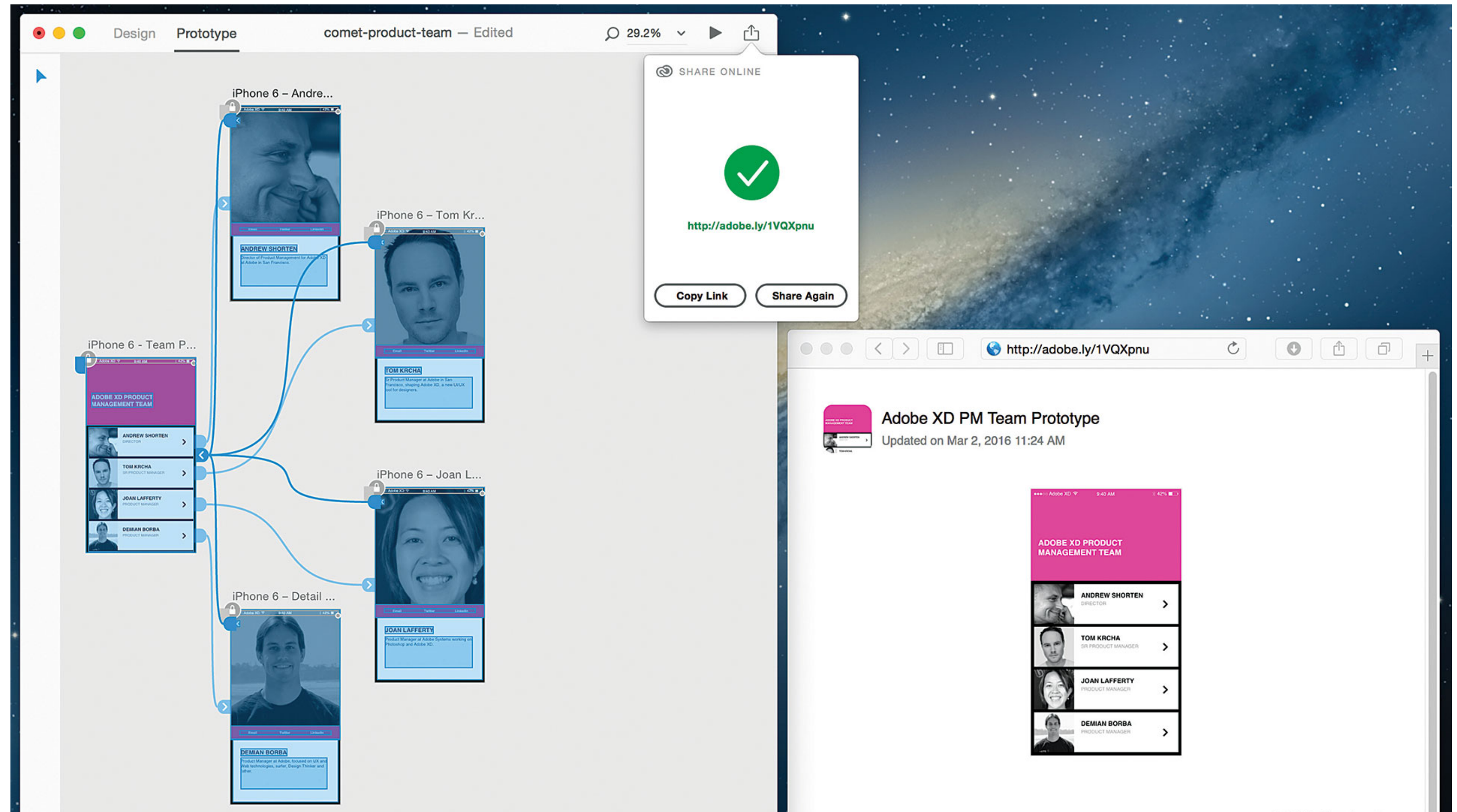
# Interface



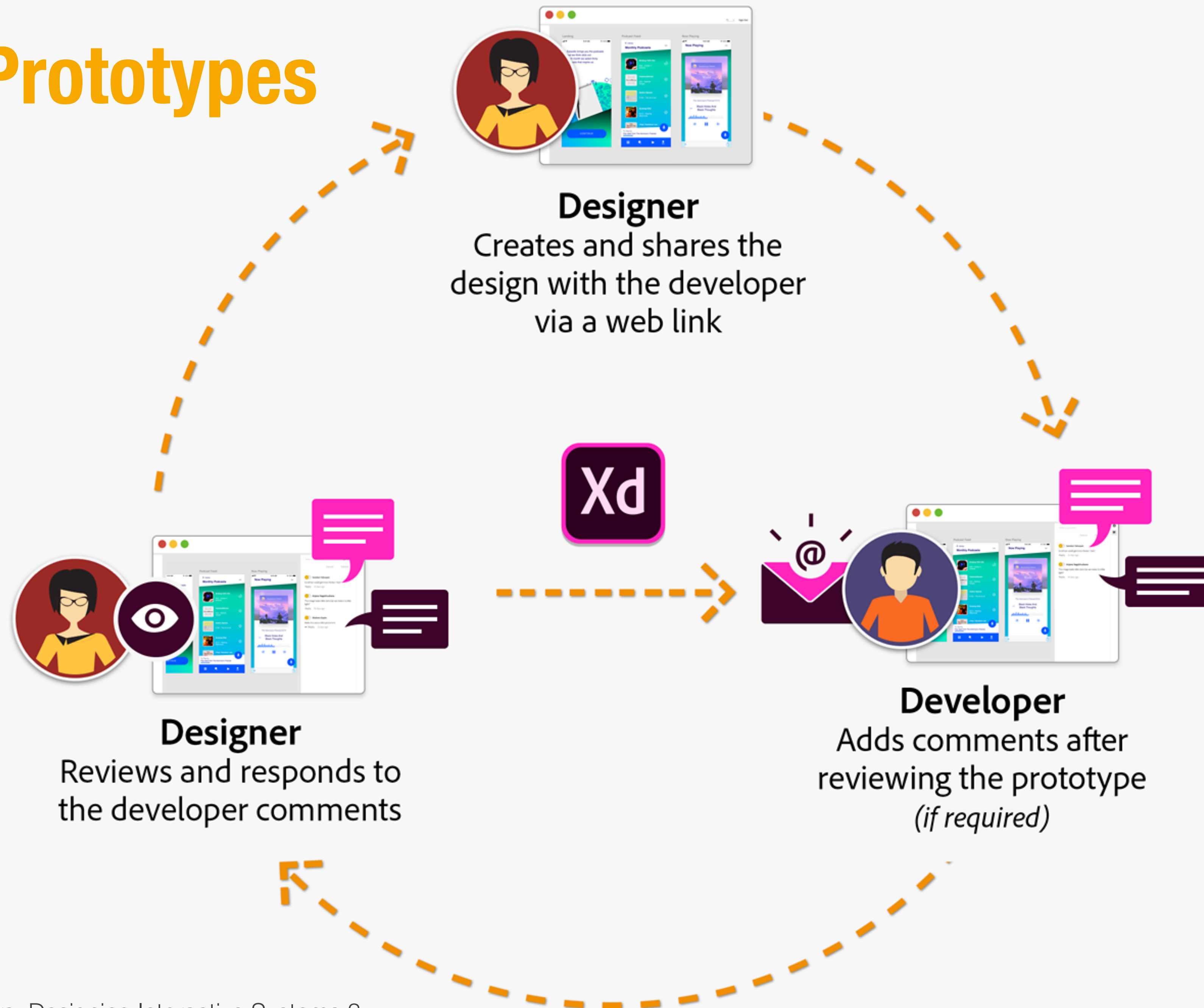


# Nonlinear Software Prototypes

- In-App preview
- Mobile preview app for iOS and Android



# Sharing Prototypes



# Demo: Adobe Xd

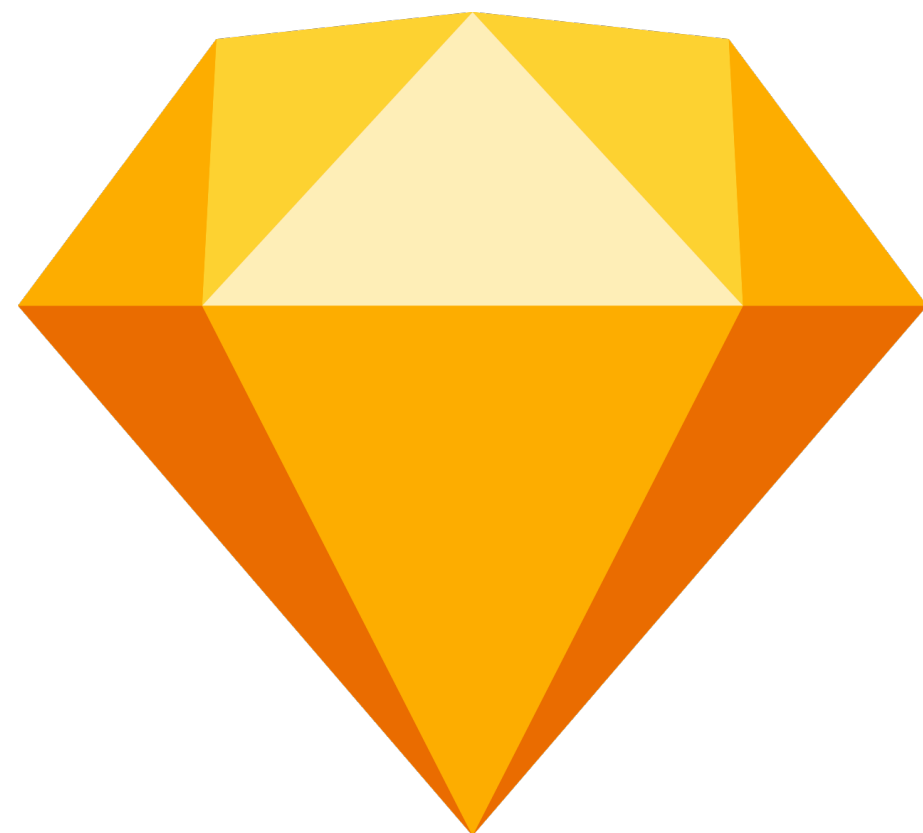


# Adobe Xd: Features

- **Scrolling** by resizing canvas
- Repeat Grid
  - Tool that lets you replicate a group of objects
- Voice Design
- Supports applications in the Adobe Creative Suite (Illustrator, Photoshop, Photoshop Sketch)
- **Developer file**

# Sketch Software

- First released in 2010
- Designing **UI** and **UX** of **mobile apps** and **websites**



Google

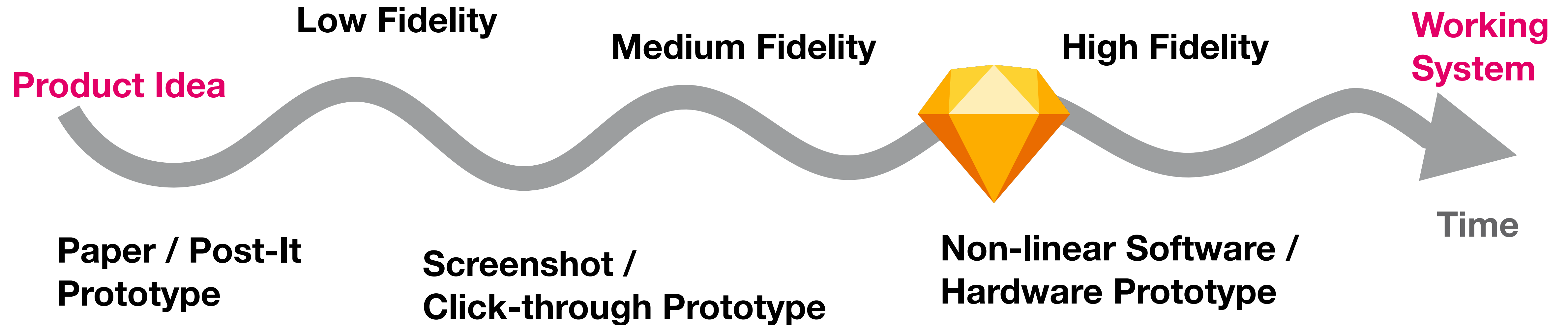


PORSCHE

stripe



# When to Use



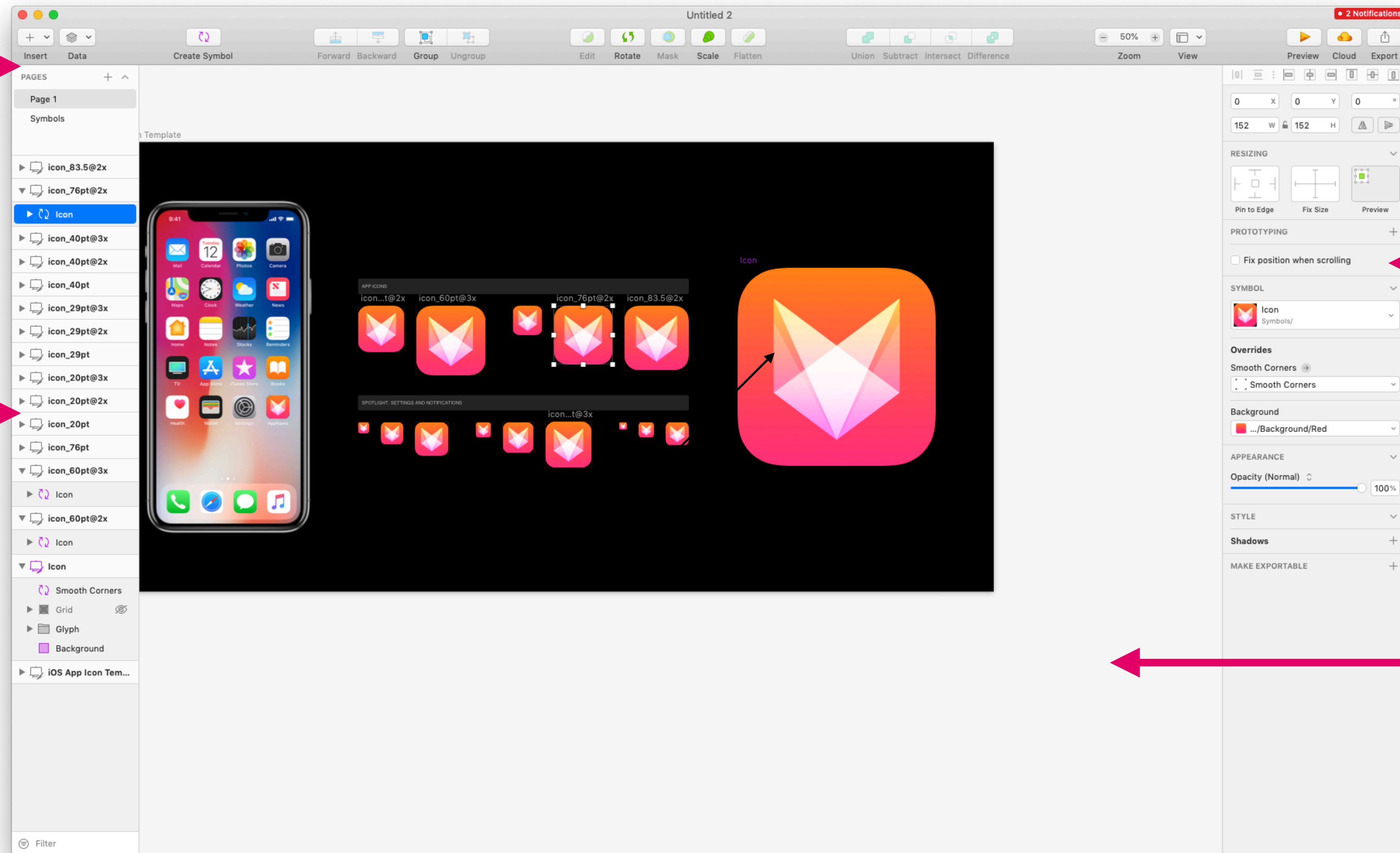
# Interface

Toolbar

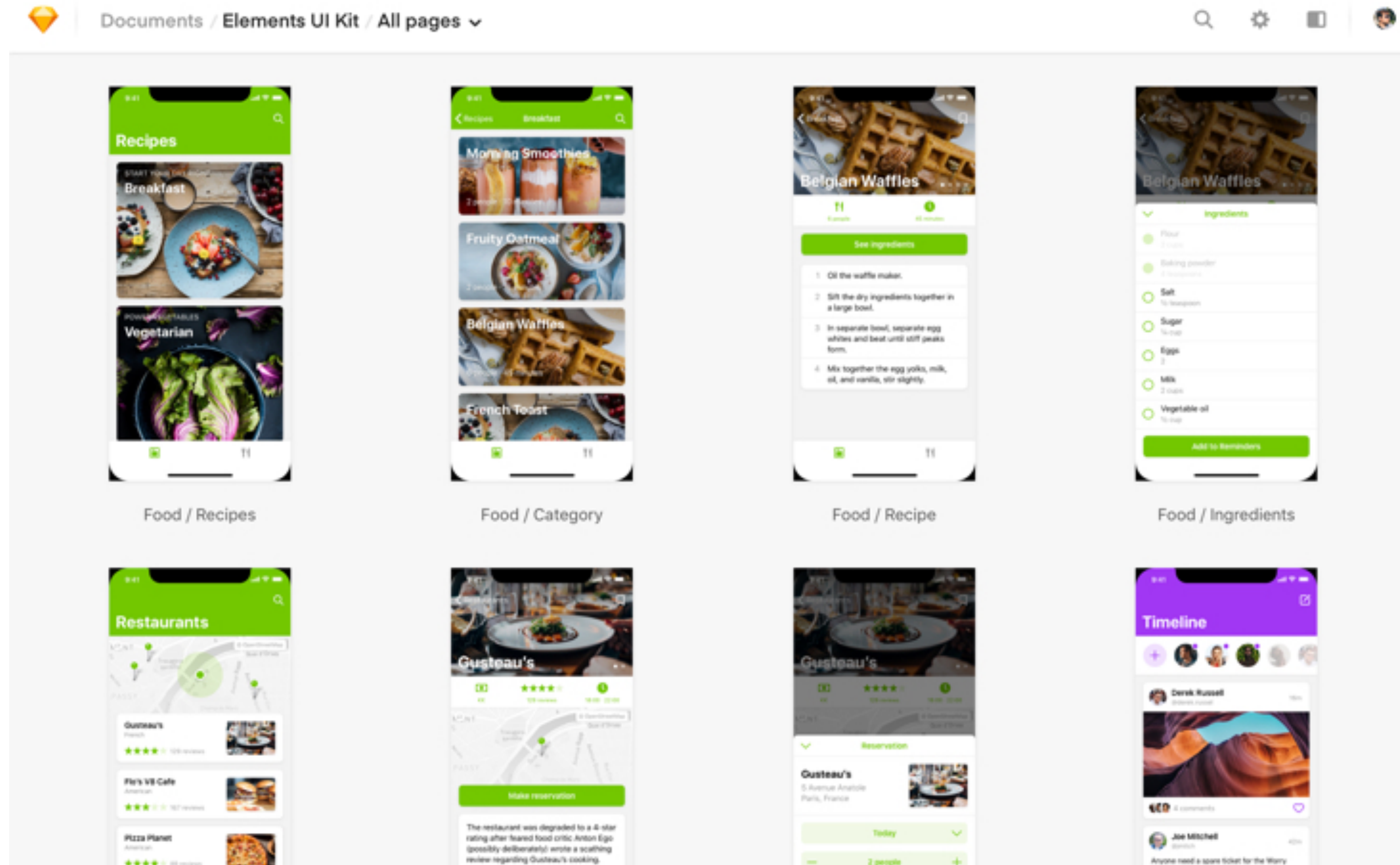
Layer List

Inspector

Canvas



# Sketch Cloud



# Demo: Sketch



# Sketch: Features

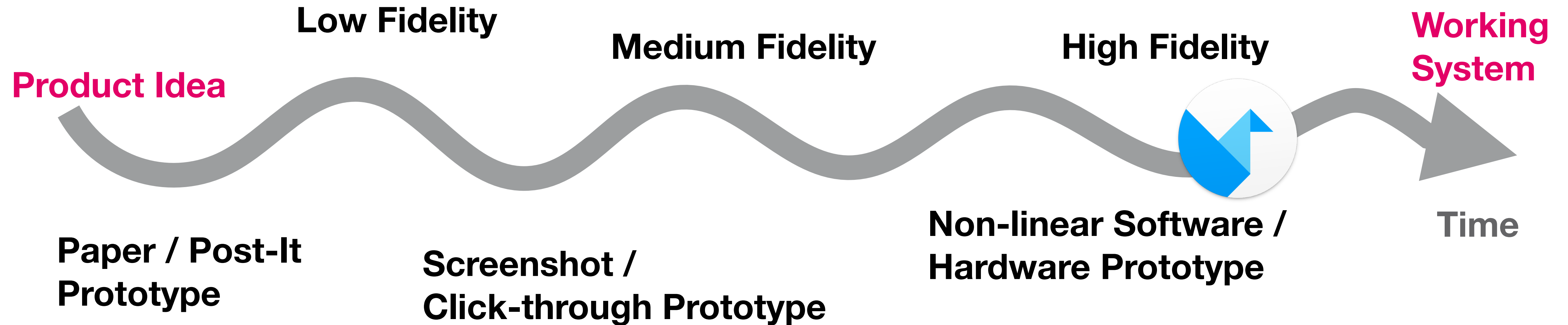
- Sketch vs. Illustrator and Photoshop
  - Combines Design and Prototyping
  - Focused on the needs of the UI and icon designer
  - Good for newbies
- **Scrolling** by resizing canvas
- **Symbols** allow reusing elements across different Artboards and Pages so you can save time and stay consistent
- **Libraries** with designed widgets, e.g., for iOS, provided on Apple website

# Origami Studio

- Free design tool by Facebook
- Plug-in for Quartz Composer
- Visual programming environment
- Programming in Quartz Composer by connecting patches to each other
- Good companion to Sketch



# When to Use



# Interface

The screenshot displays the Axure RP software interface for designing interactive systems. The main workspace is divided into three primary sections:

- Viewer:** Located on the left, it shows a simulated mobile device (iPhone 7) displaying a video player interface with a play button.
- Logic Flow Diagram:** The central area contains a visual programming logic. It starts with a 'Switch' component connected to a 'HYPER-REALITY-360p.mp4' video asset. A 'Not' condition leads to a 'UI-Play' component. A 'Classic Animation' component is linked to three 'Transition' components, which in turn control the 'Opacity', 'Scale', and 'Position Y' of a 'UI-Title' component.
- Properties Panel:** On the right, the 'UI-Title' component's properties are listed, including 'Enable', 'Position' (X0, Y-150), 'Anchor', 'Size' (W375, H150), 'Opacity', 'Scale', 'Rotation', and 'Pivot'. Below this, a 'Layer List' shows the hierarchy of elements: Status Bar, UI-Title, UI-Play, HYPER-REALITY-360p.mp4, and Color Fill.

Annotations with pink arrows point to specific interface elements:

- Add Components:** Points to the '+' button in the top toolbar.
- Properties:** Points to the 'UI-Title' properties panel.
- Layer List:** Points to the 'HYPER-REALITY-360p.mp4' entry in the layer hierarchy.
- Viewer:** Points to the simulated mobile device on the left.



# Demo: Origami



# Origami: Features

- Building small high fidelity prototypes
  - Rather complex in its functions
- Tweaking the interaction you have in mind
- Not suitable for quick wireframing
- **Scrolling** possible
- Pasting of Sketch UI files possible
- Good for logic of realistic interaction sequences — state changes, timeouts,...

# Features

<b>Feature \ Tool</b>	Balsamiq	Adobe Xd	Sketch	Origami
Plattform specific options				
Sharing Designs	Only PNG (not interactive)	Online Link	Online Link	Via Mail
Libraries				
Scrolling				
Developer file	PNGs without specifications	Sharing includes all assets, dimensions	objects are exported with specifications	Export code snippets
Real Testing		App for iOs and Android		Origami Live app on Mac
Advanced Interactions		Voice commands		Microphone & Sound

# Which Tool to Use When?

- balsamiq:
  - Standard (productivity) apps
  - Fast wireframing by using widgets for medium fidelity
- Adobe Xd & Sketch:
  - Standard UIs via libraries, and custom UIs (immersive apps, games)
  - Medium to high fidelity
- Origami:
  - Small, high fidelity, model interaction sequence
  - Reuse Sketch files



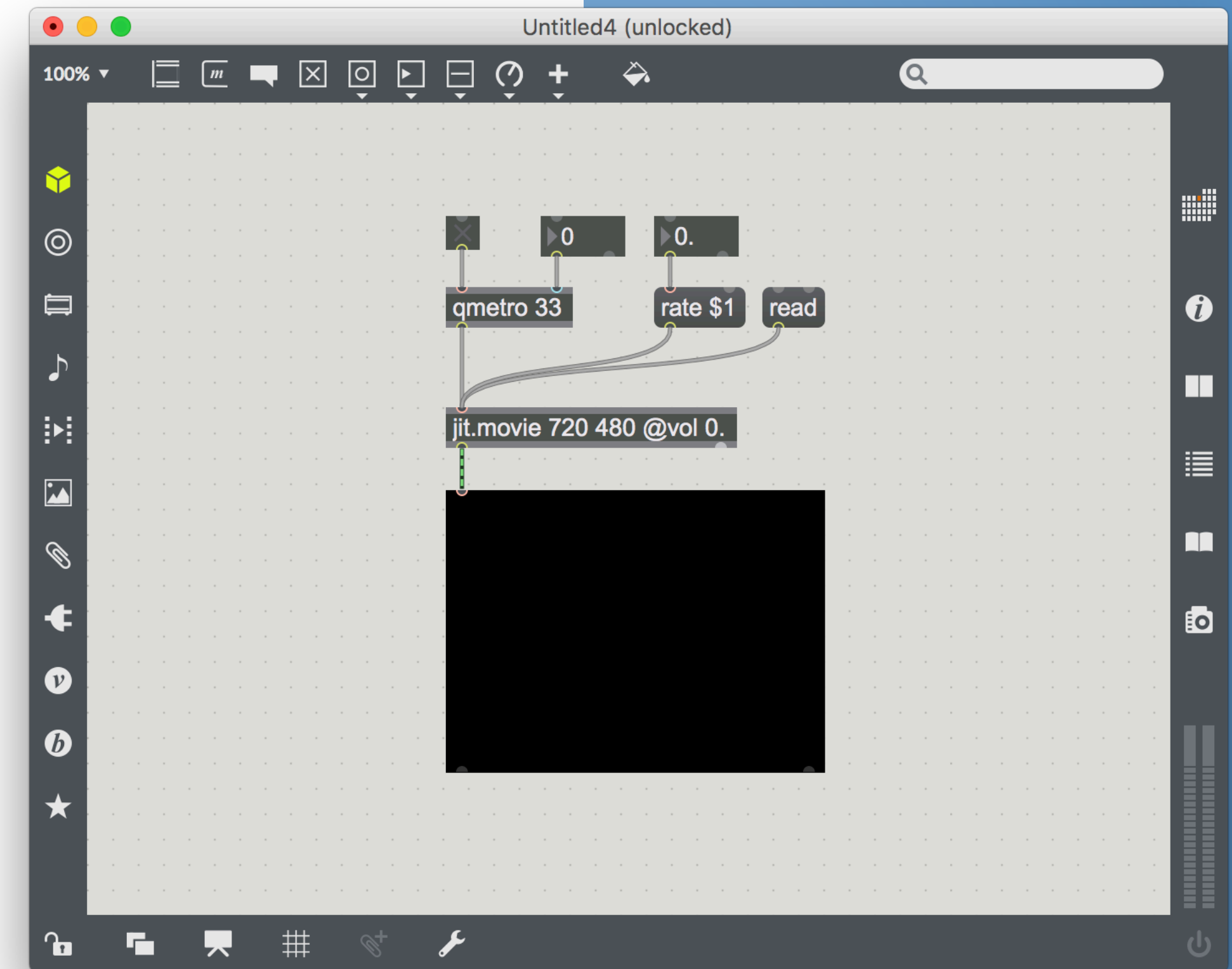
## CHAPTER 41

# Multimedia Prototyping: Max/MSP



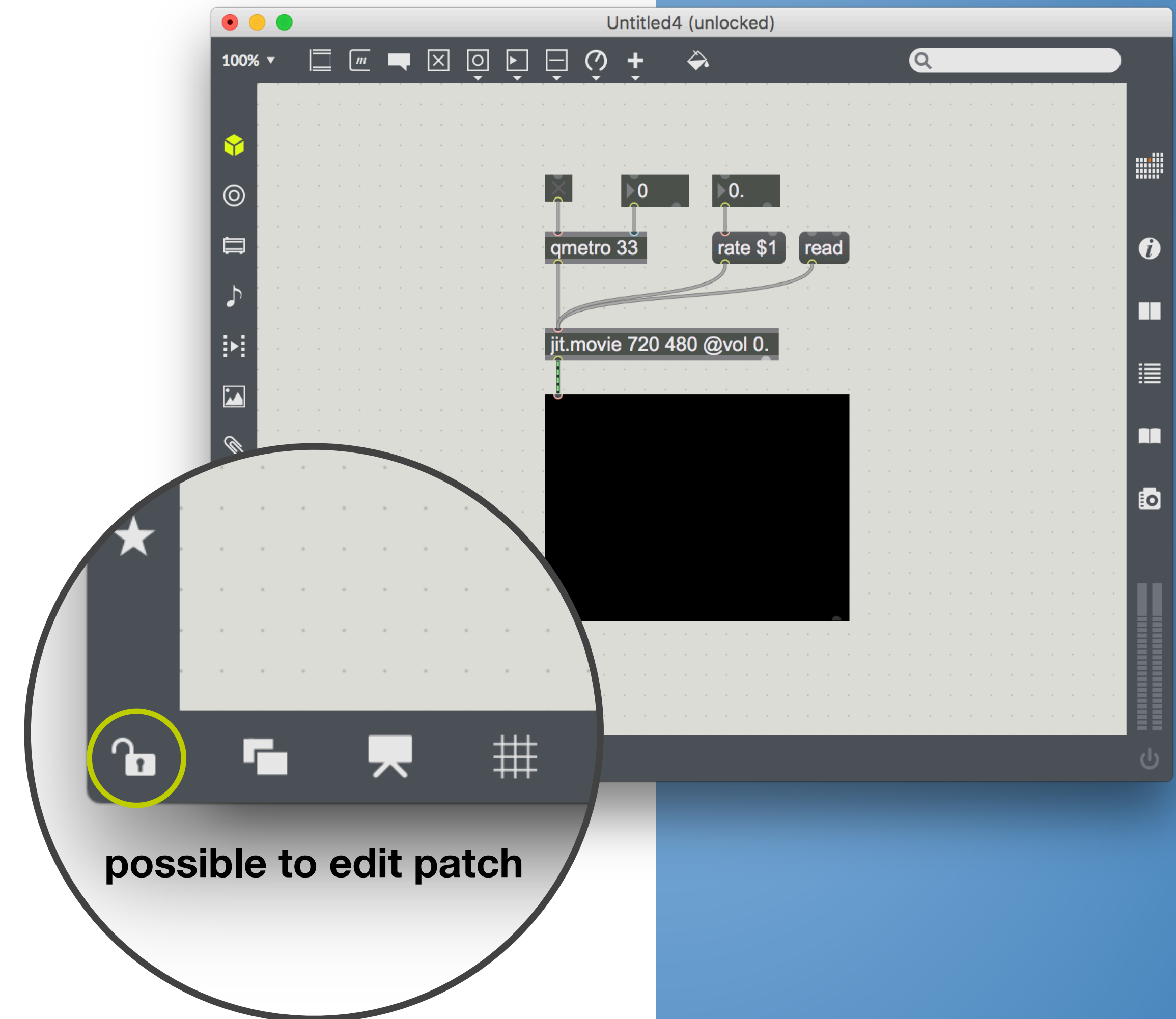
# Intro to Max

- Visual **dataflow-based programming** inspired by patch panels in recording studios
- Tuned for working with time-based input streams
- Programs are called **patches**
- **Objects** contain functionalities and are visually represented by boxes
- **Patch cords** connect objects



# HCI View on Max

- **Prototyping** UIs
  - For working with time-based media
  - Using time-based media as input/output devices
- No need to be a DSP guru
- **Liveness** allows fast development
  - Locked: use patch
  - Unlocked: edit patch



“If most MIDI apps are 🍕 **pizza**,  
then Max is a **kitchen**.”

- Miller Puckette

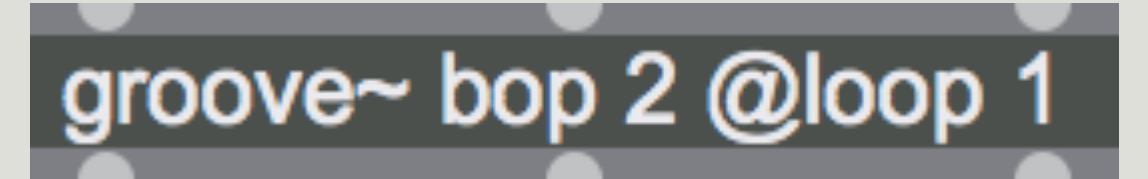


# History

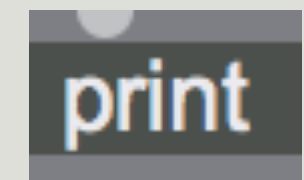
- 1986: **Patcher** for Mac  
MIDI and control processing
- 1989: **MSP** to process digital audio
- 1989: Ircam Musical Workstation  
featuring Max containing
  - GUI on NeXTStep
  - „Faster Than Sound“ card
- 1995: **pd** (pure data)  
ability to share patches between  
multiple projects
- 2000s: **Jitter** real time processing of  
matrix data, e.g. video, 3D;  
Windows support
- 2010s: **Vizzie** visual processing  
module, **Ableton Live** integration,  
**Gen** integrates imperative  
programming blocks, **NodeJS**  
support

# Objects

- **Normal objects**
  - Squares with object name in it
  - Object name can be followed by arguments
  - Inputs are at the top, outputs at the bottom
- **UI objects**
  - Sliders, dials, number fields, buttons, etc.
- **Comments**



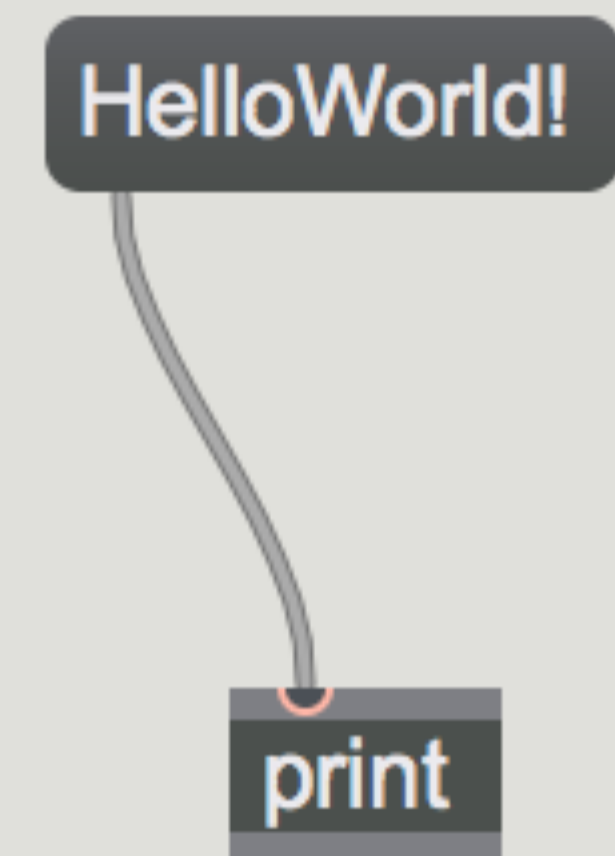
groove~ bop 2 @loop 1



print

# Messages

- Sent through patch cords
- Various **types**
  - number: int, float
  - list of numbers, separated by spaces
  - word ("symbol")
  - bang
  - combinations
- **Message order**  
right to left, bottom to top



# Math in Max

- Basic operations as objects like **\***, **+**, **-**, **/**
- Common functions like **sin**, **abs**, **sqrt**
- **expr** object for more complex terms
- **if** for conditions

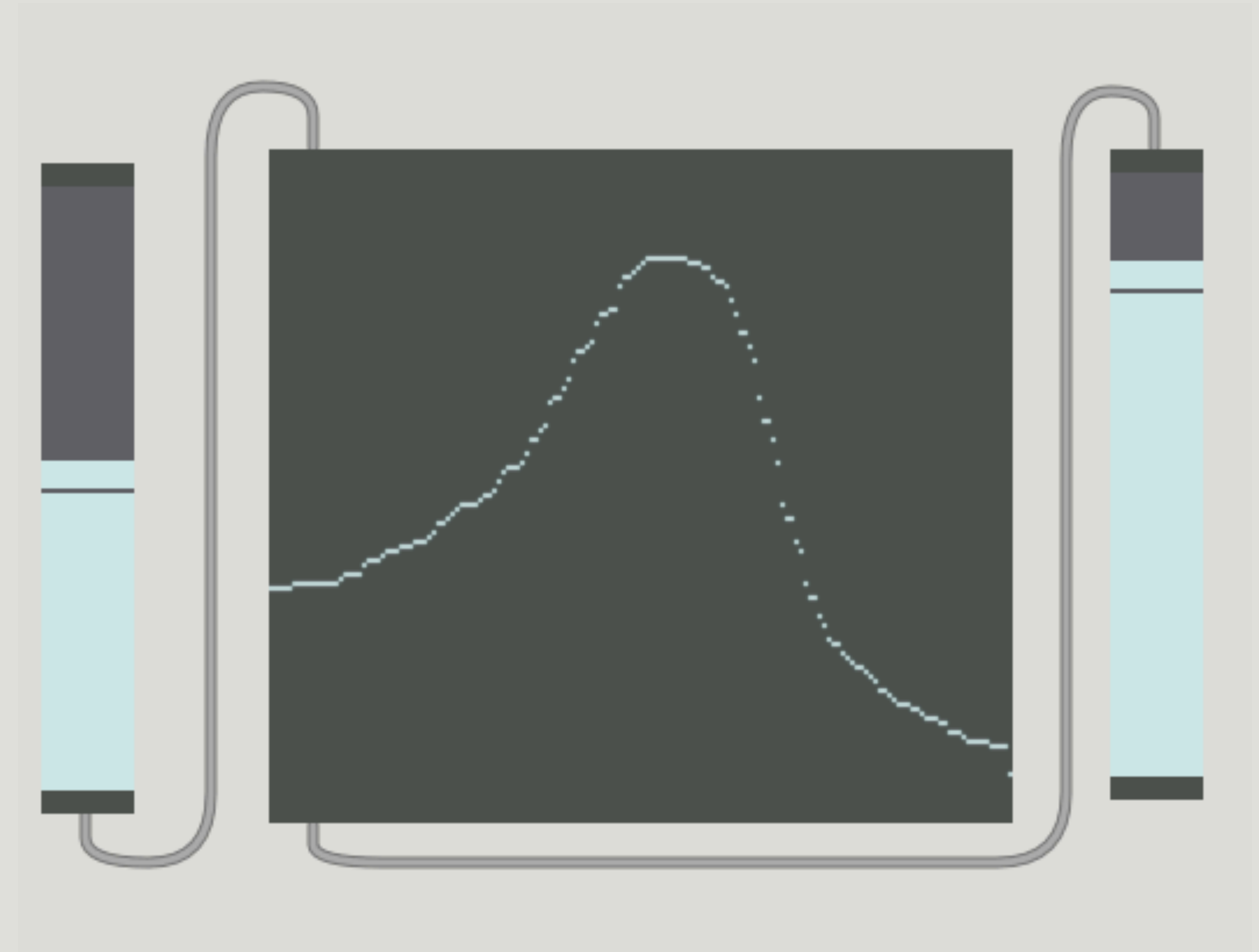
# Creating Patches

## DEMO 1

Using a table to map slider input to slider output

## DEMO 2

Adding integers



# MIDI Handling

- Objects for input and output of notes, control messages and raw data, e.g. **ctlin**, **midiout**, **notein**
- Objects for translation between MIDI notes and frequencies
- MIDI messages are ordinary messages:  
MAX handles MIDI as sequence of numbers

# MSP

- MSP object names end with ~
- MSP signal patch cords are curly
- curly signals cannot be connected to regular objects
- DSP operations happen at sample timing precisions

# MSP Features

- **adc~** and **dac~** for audio in-/output
- **fft~** and **ifft~** convert from time to frequency domain
  - phase vocoder (timestretching)
  - complex filters
- **buffer~** for storing sample data
- UI elements for visual analysis

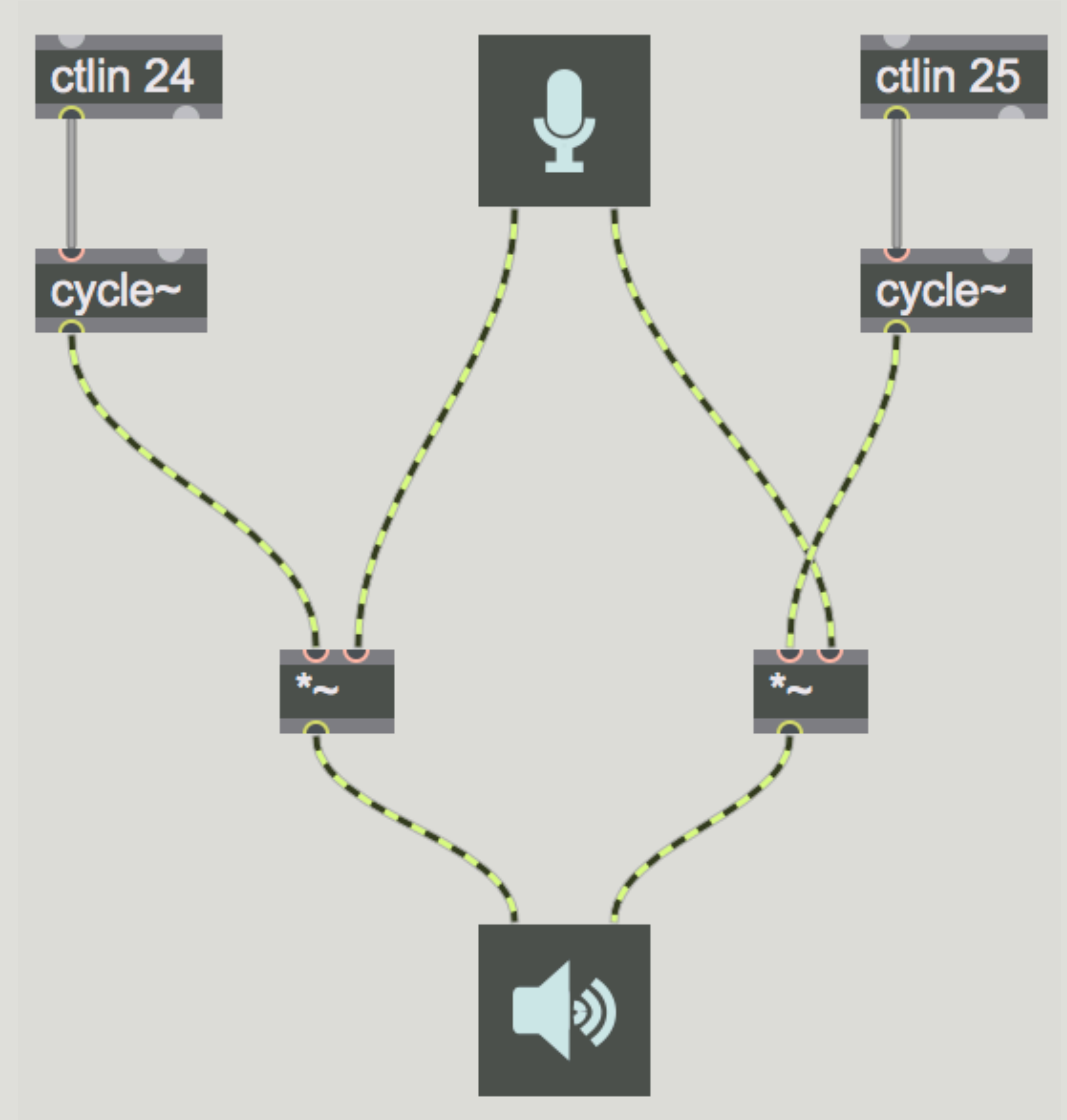


# Simple MSP Patch

## DEMO 3

Performing ring modulation

- **cycle object**  
generates a periodic waveform
- **ctlin object**  
outputs MIDI control signals



# Jitter

- 130 additional objects for Max/MSP
- processing of matrix data (rows and columns)
  - video, still images, 3d geometry, text, spreadsheets, particle systems, voxels or audio

# Jitter Objects

- most names start with **jit.**
- attributes instead of dozens of arguments
- the green matrix patch cords cannot be connected with other Max/MSP objects

brightness 0.5 attribute passed as a message

jit.brcosa

jit.brcosa @brightness 0.5

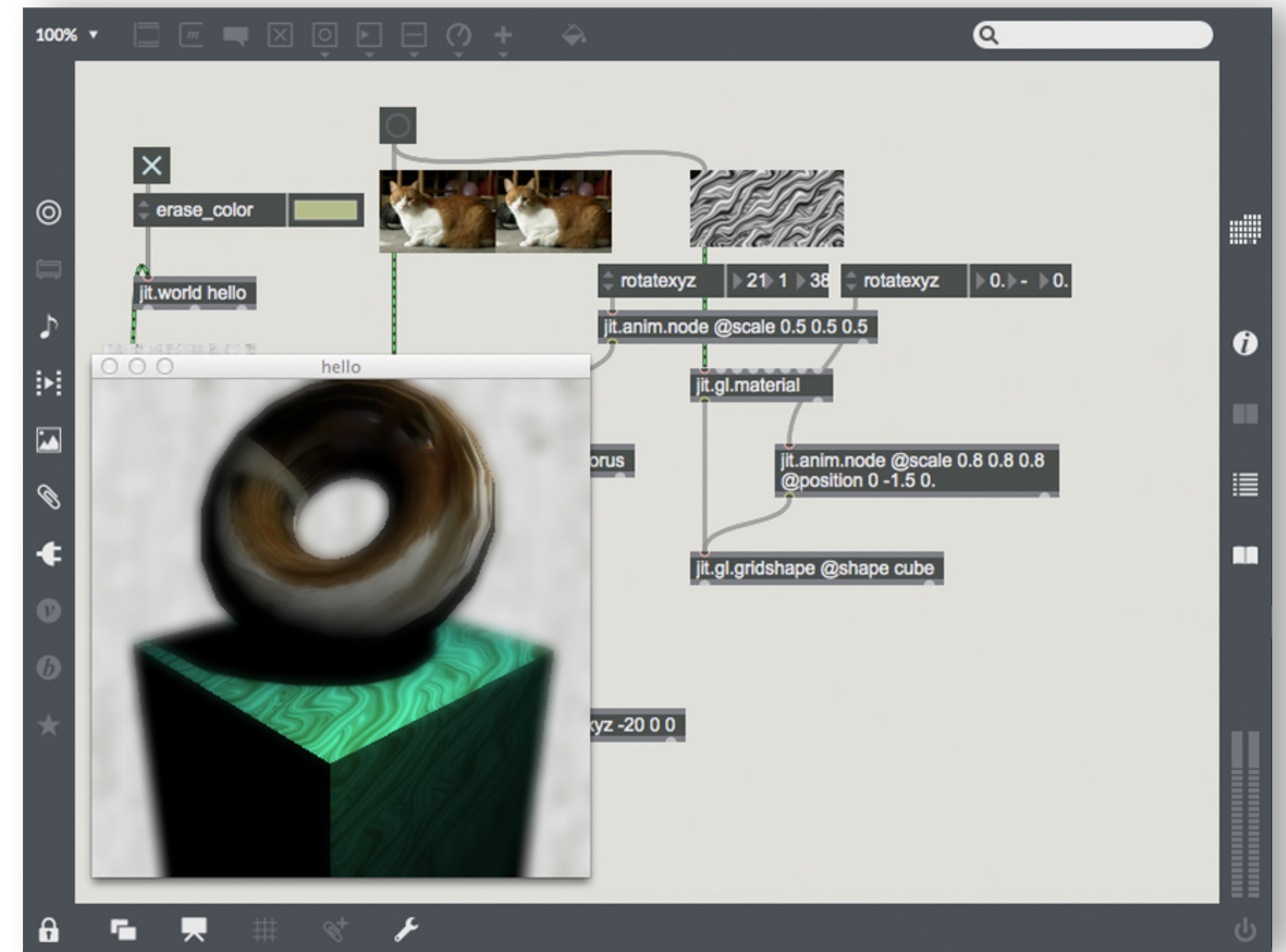
attribute provided as argument

jit.grab 320 240 1

jit.brcosa

# Jitter Objects

- **jitter.qt.\*** for video
- **jitter.gl.\*** for 3d/OpenGL
- **jitter.p.\*** for particle effects
- **jitter.la.\*** for linear algebra
- **jitter.peek~** and **jitter.poke~** for connection to MSP



# Extending Max/MSP

- **Software**

- write plugins in C that appear in Max as new objects
- free SDK available

- **Hardware**

- MIDI controllers and synthesizers
- DMX Lighting
- Arduino

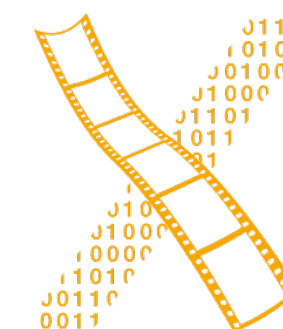


# Designing Interactive Systems 2

## Lecture 14: Hardware Prototyping

Prof. Dr. Jan Borchers  
Media Computing Group  
RWTH Aachen University

[hci.rwth-aachen.de/dis2](http://hci.rwth-aachen.de/dis2)



**RWTH**AACHEN  
UNIVERSITY



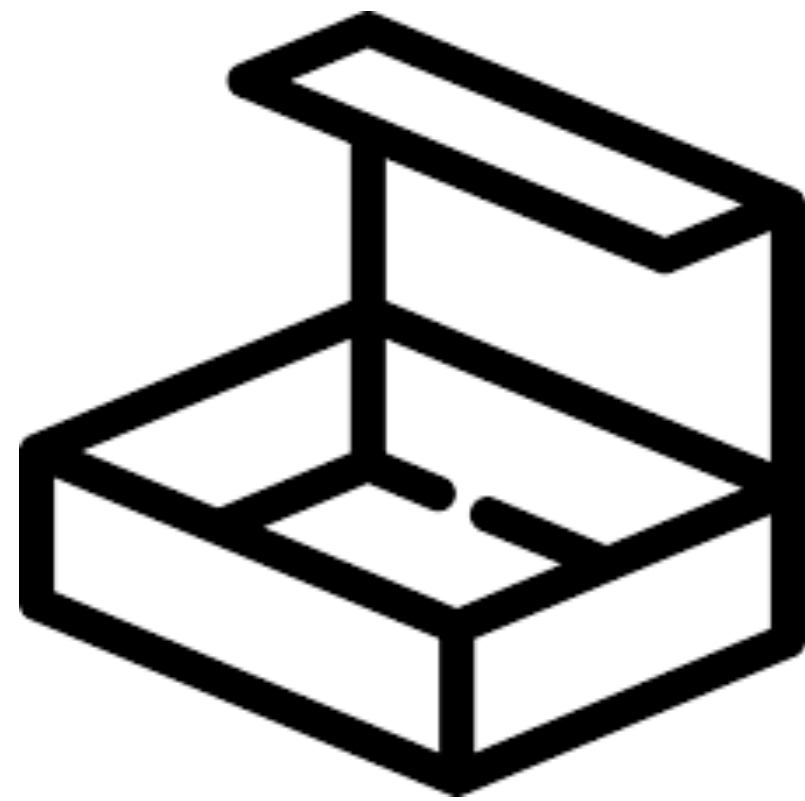






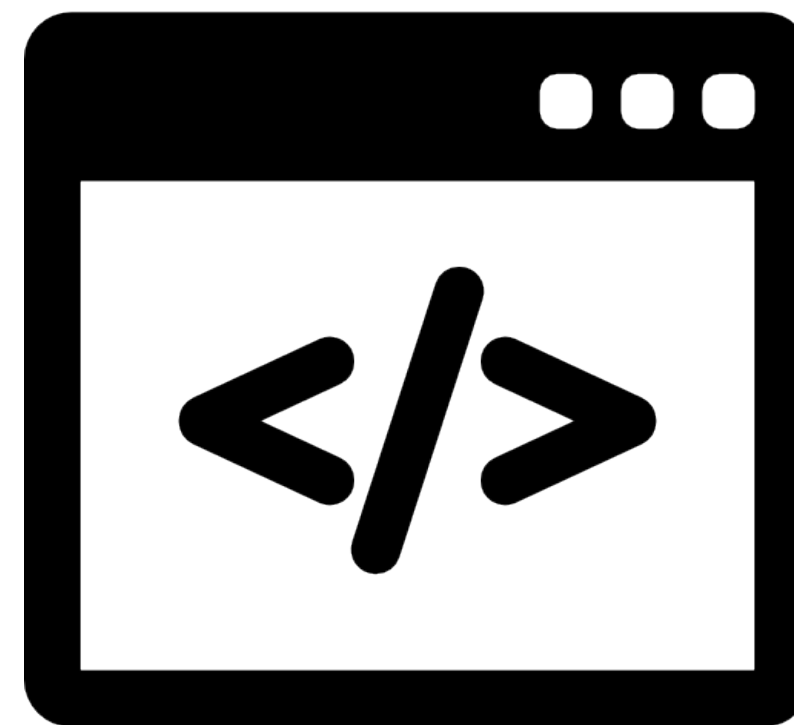
# Topics today

UI =



**Form**

+



**Function**

+



**Sensors &  
Actuators**



**CHAPTER 42**

# **Prototyping Form: Mechanics**



# Why Hardware Prototyping? – Ubiquitous Computing



WED • NOV 5

8:00 meet ed

9:00

10:00 Staff meeting

11:00 Send price list

12:00 Lunch

1:00 Touchdown spec

2:00 review meeting

3:00 Forecast meeting

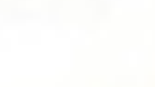
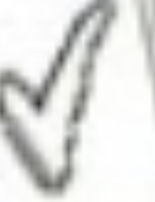
4:00

5:00

6:00

Today S M T W T F Goto

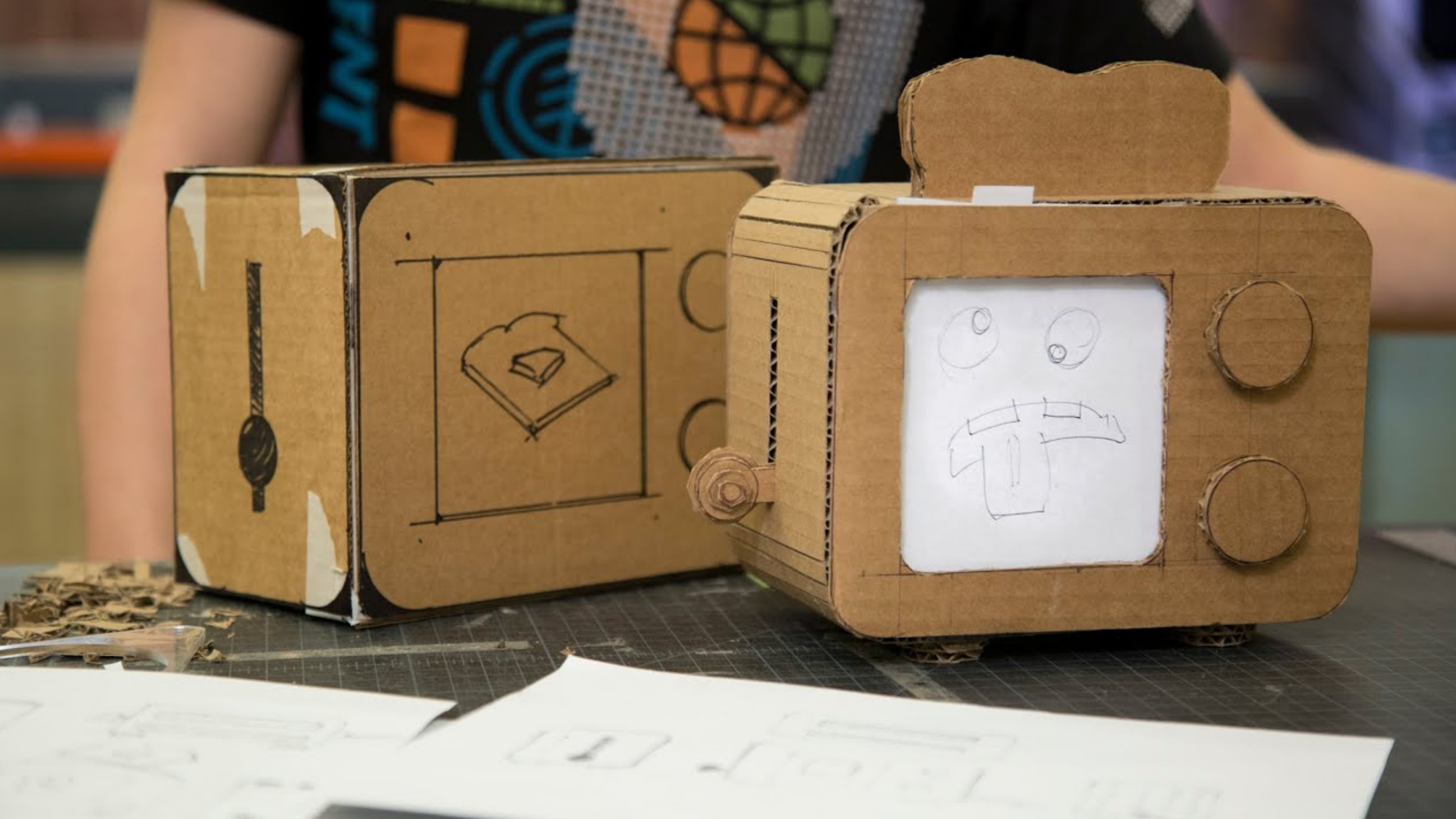
Menu



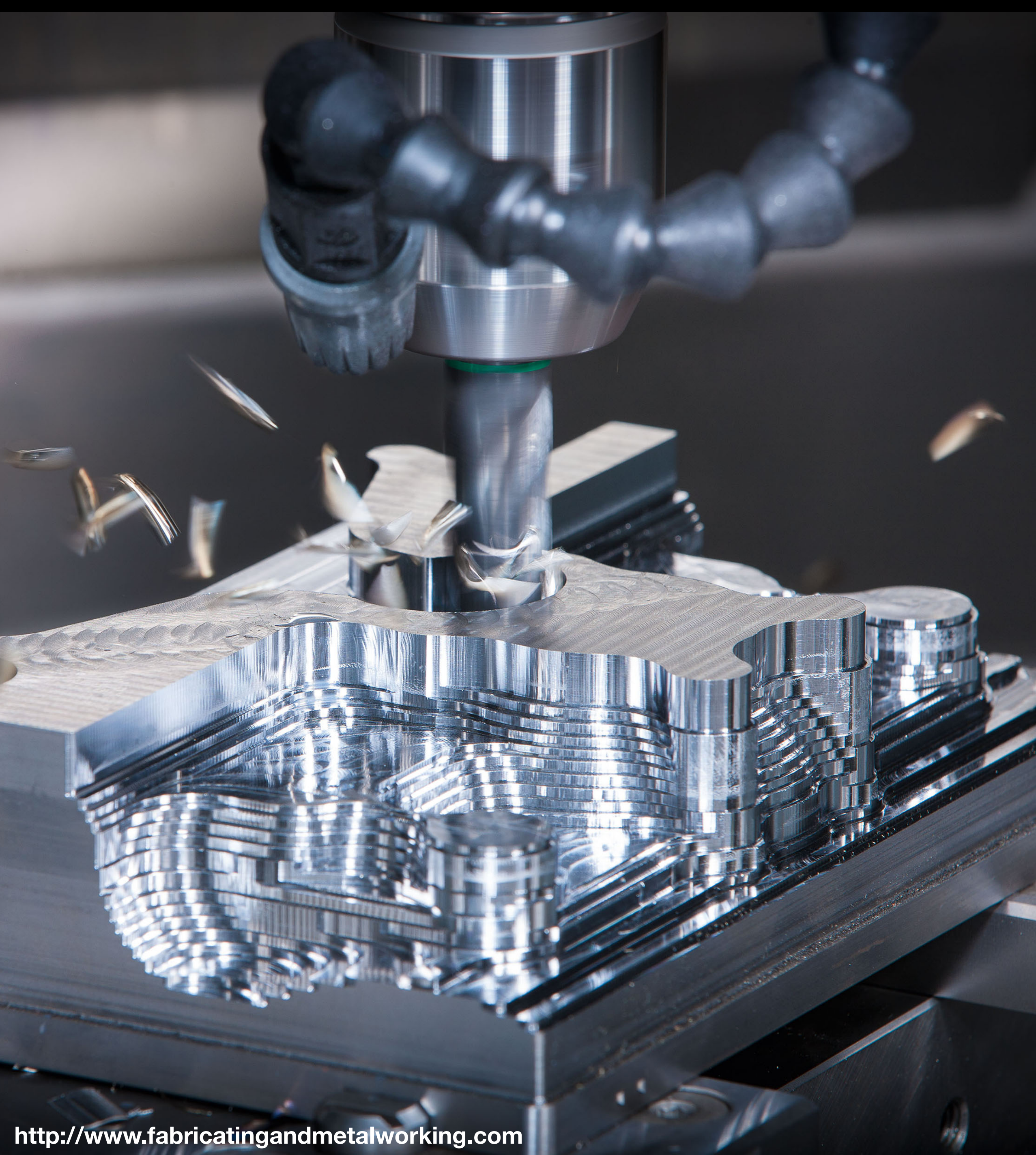












<http://www.fabricatingandmetalworking.com>



<http://en.hglaser.com>



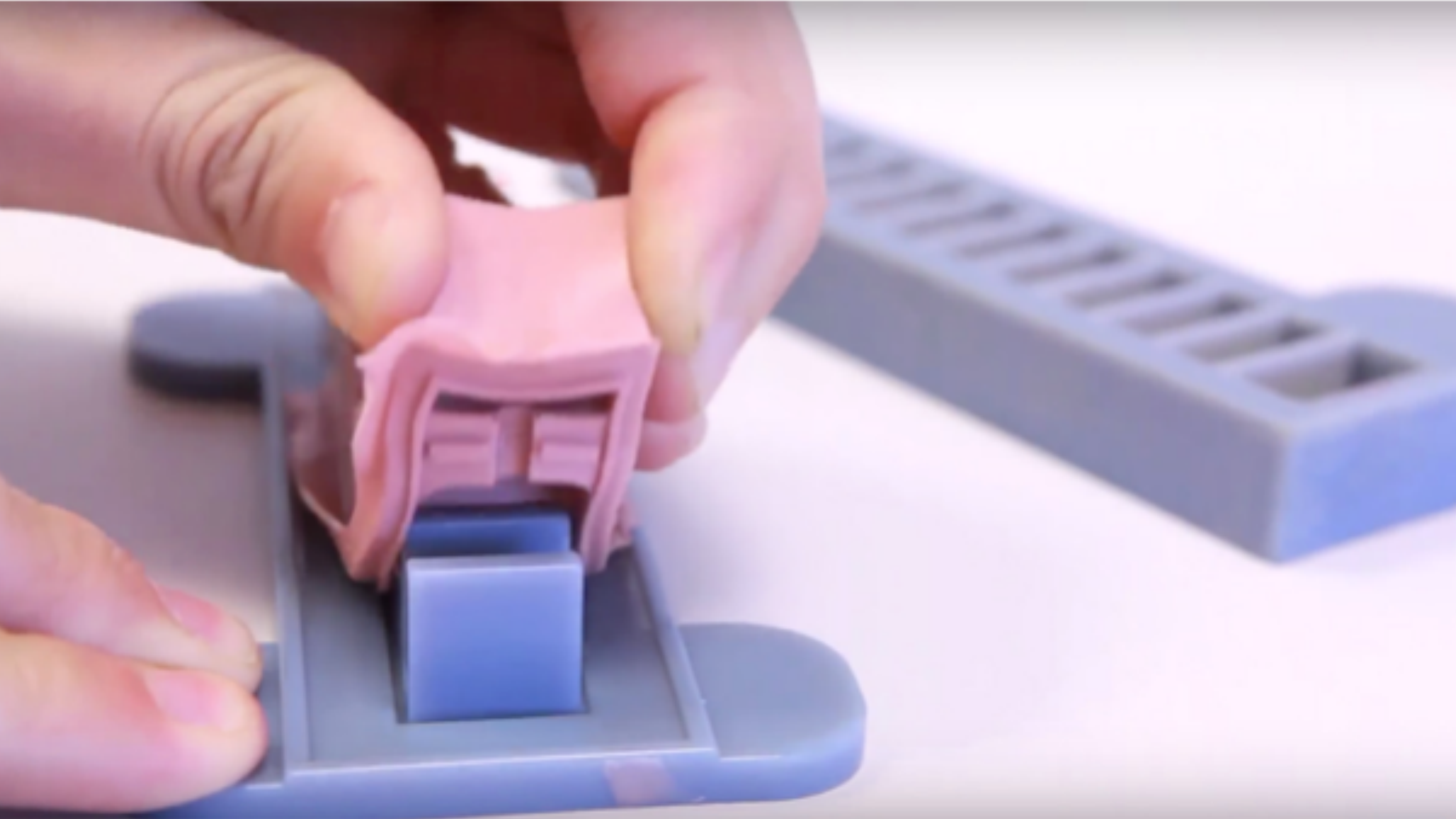
**RWTHAACHEN**  
**UNIVERSITY**





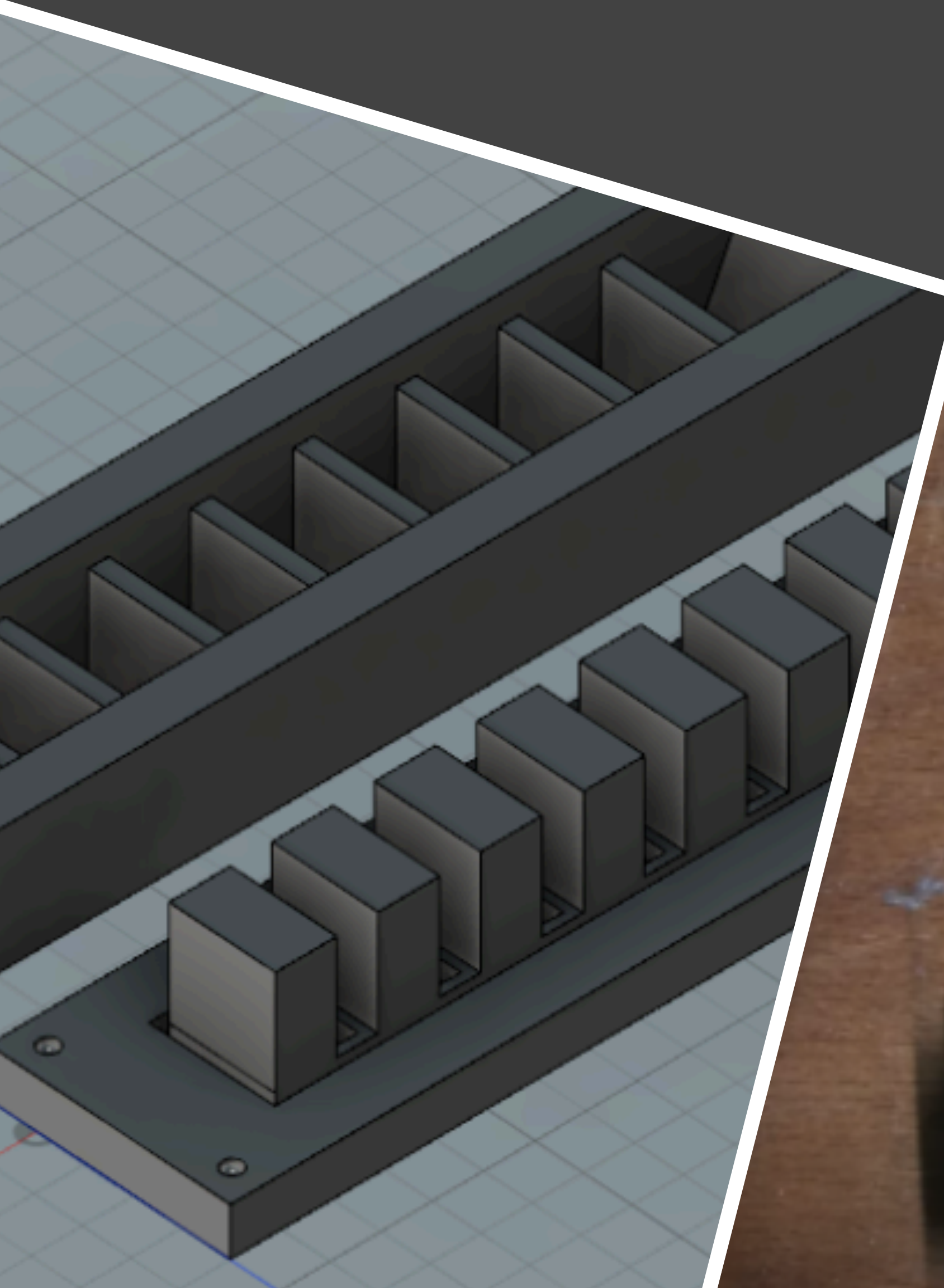
Source: Ultimaker Ultimaker Press Room Images







# 1. CAD Design



# 2. 3D Printing



# 3. Casting







## CHAPTER 43

# Prototyping Function: Embedded Programming

# Adding function to hardware prototypes

—

## Embedded programming & electronics



14 digital I/O (40mA)  
incl. 6 PWM 'analog' out

ICSP

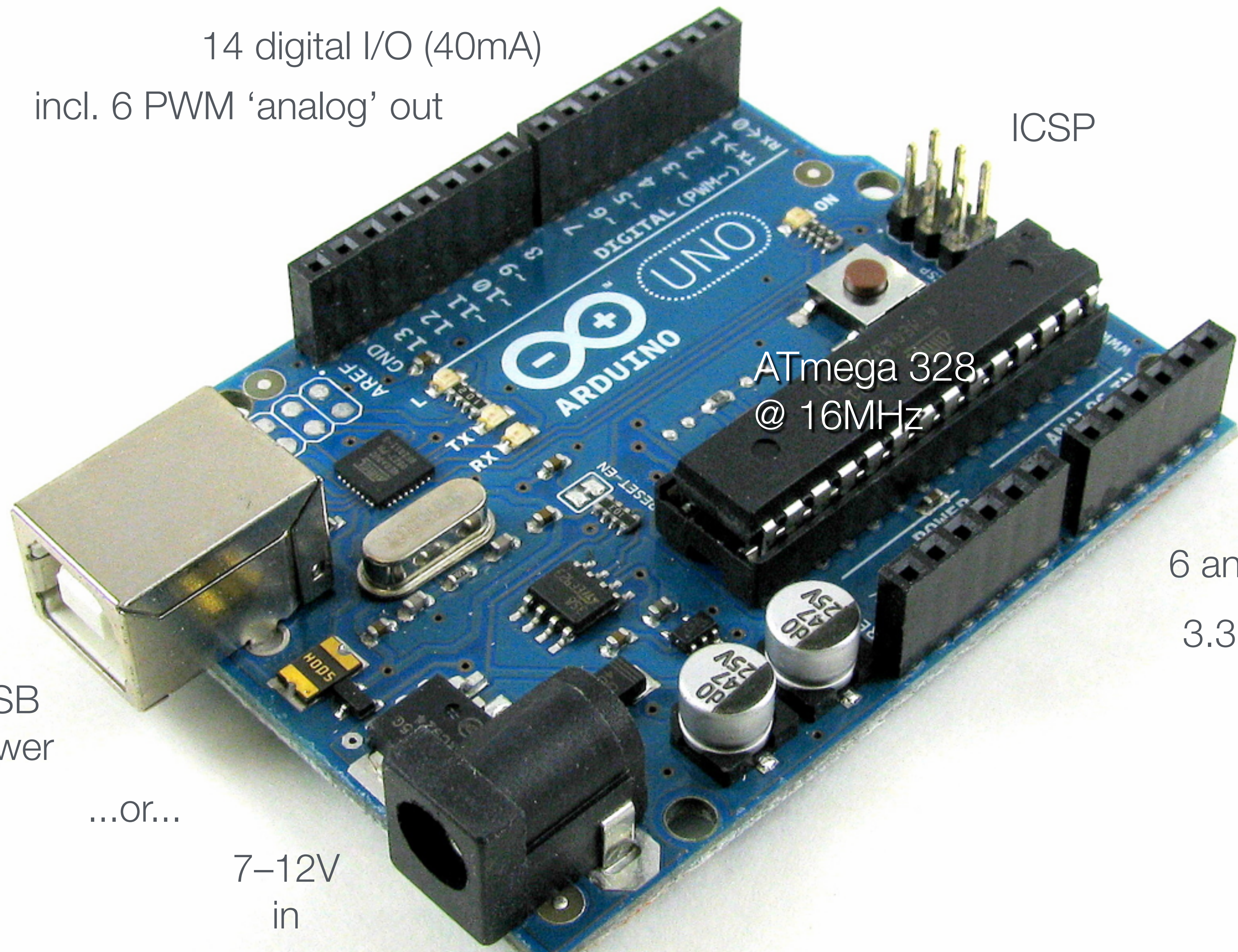
ATmega 328  
@ 16MHz

6 analog in  
3.3 & 5 V

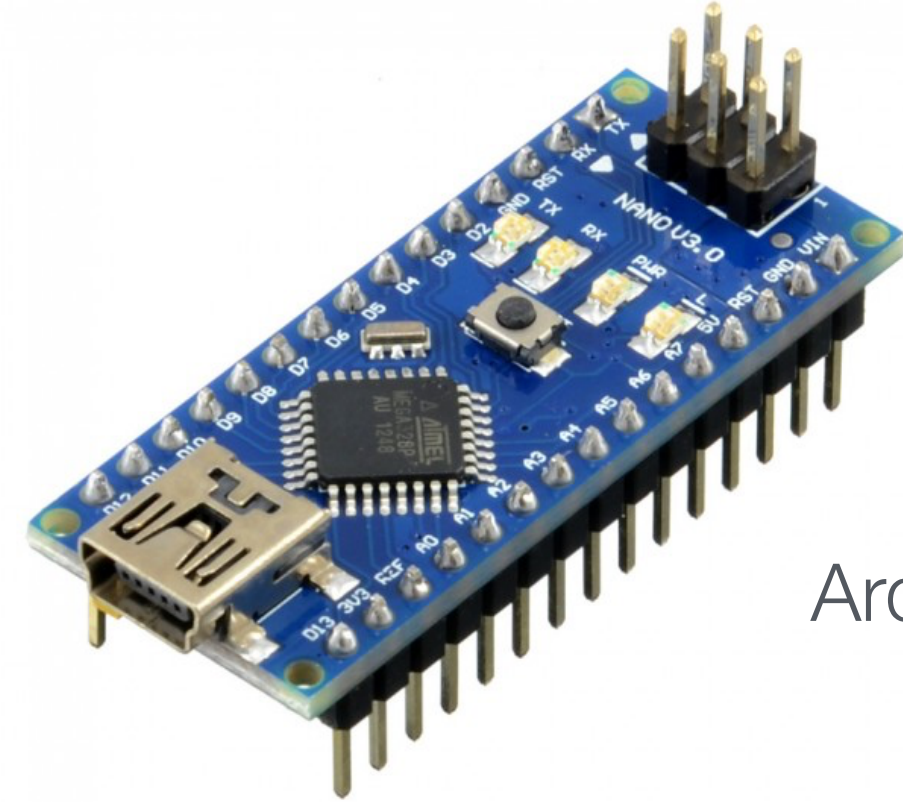
USB  
power

...or...

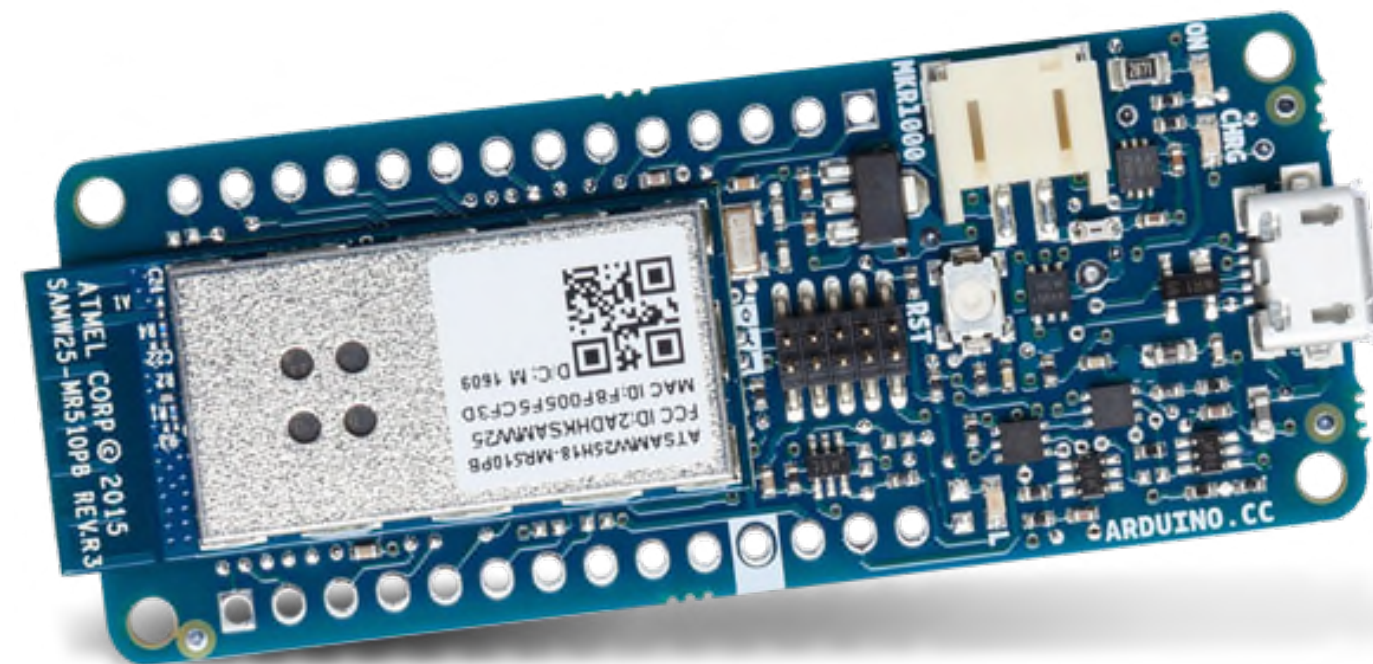
7-12V  
in



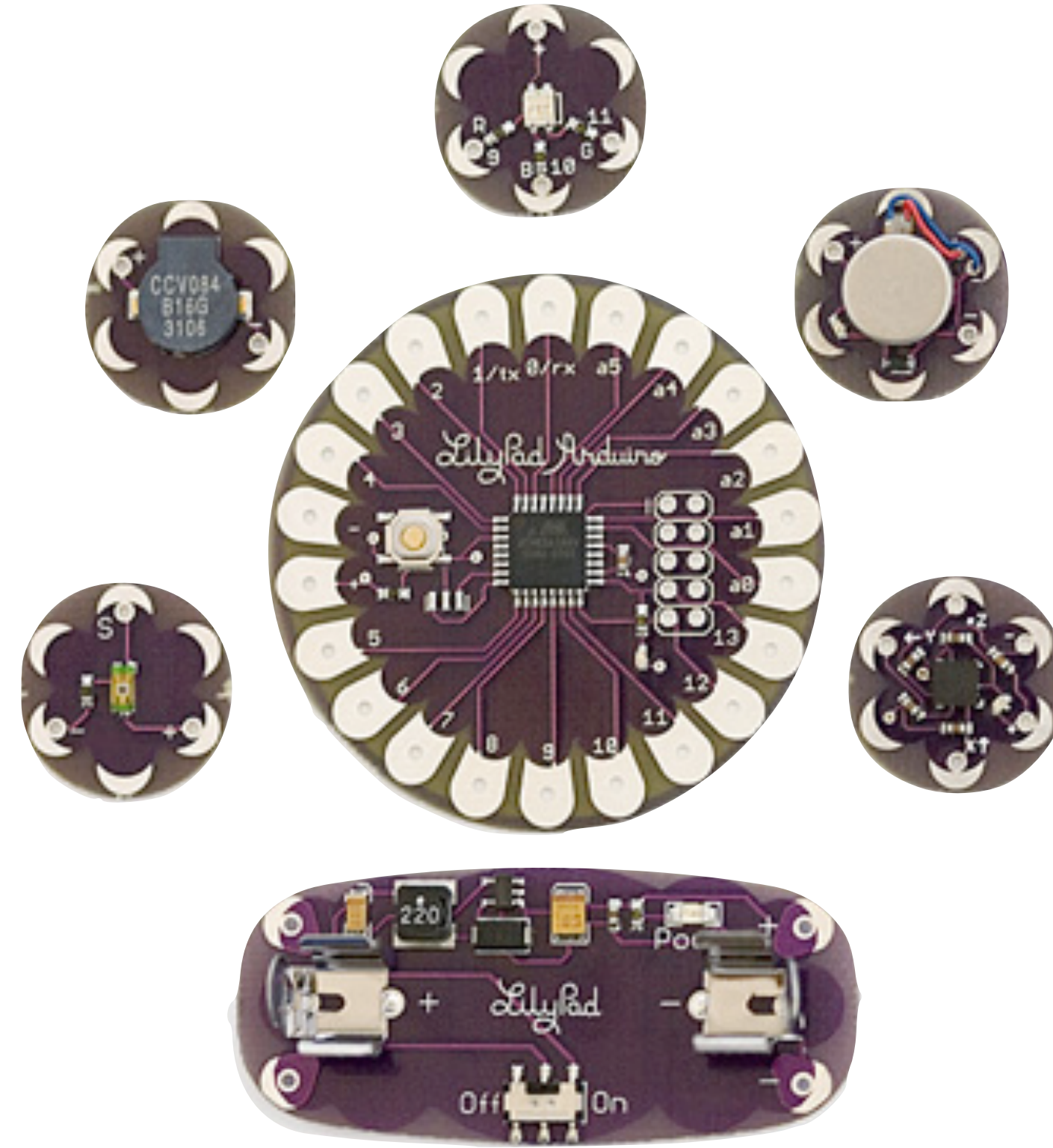




Arduino Nano



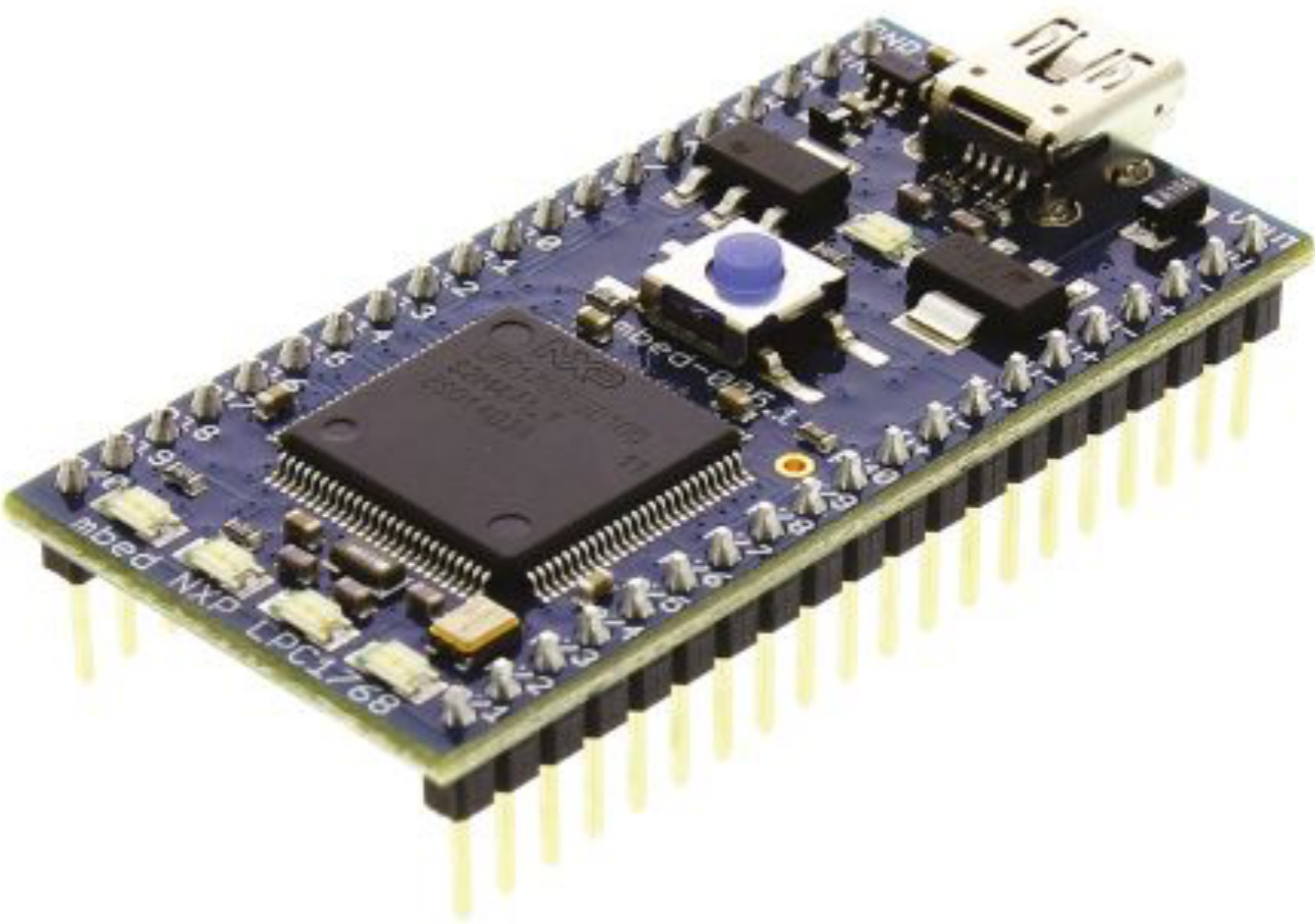
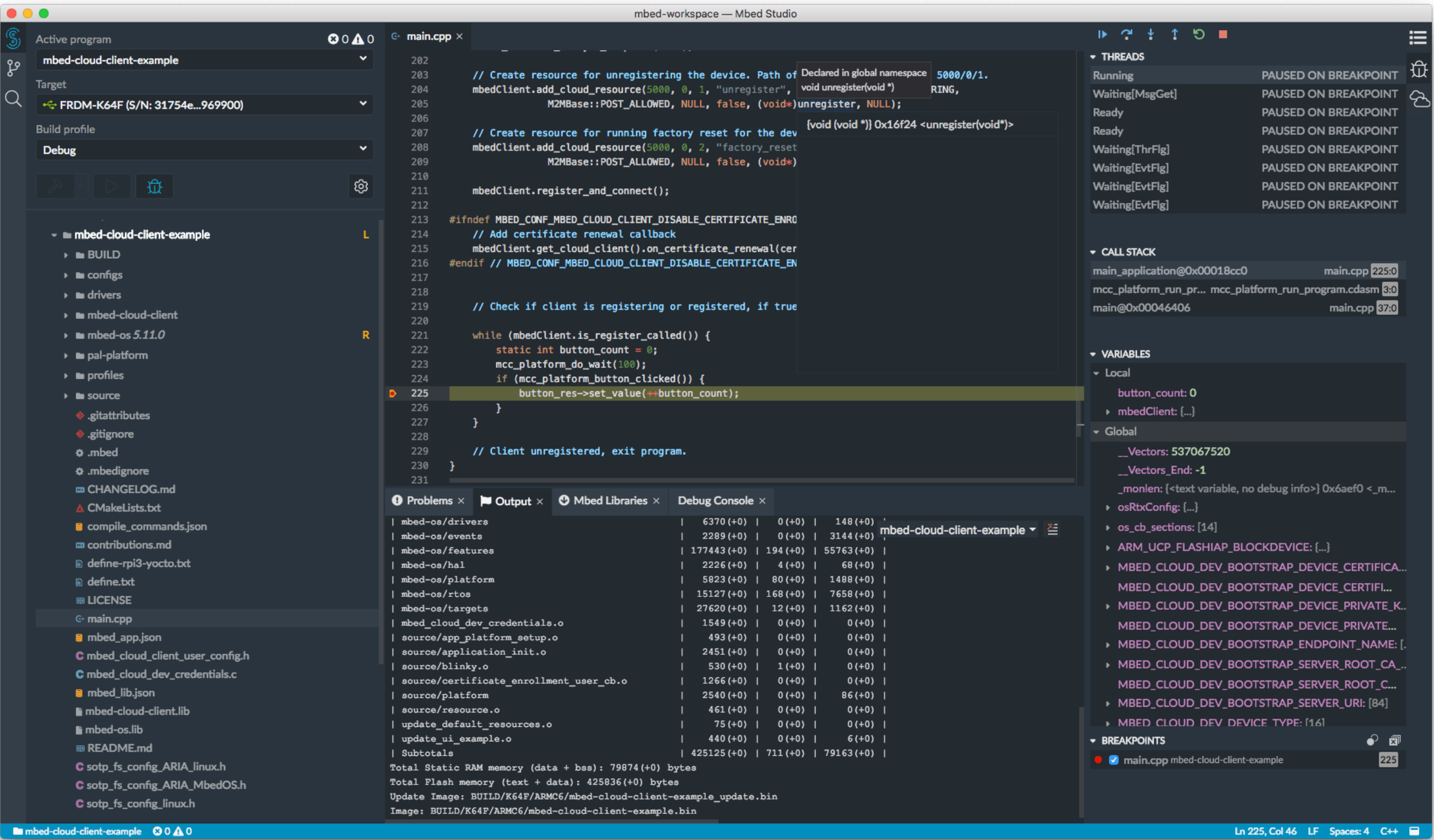
Arduino MKR



Arduino Lilypad



# mbed



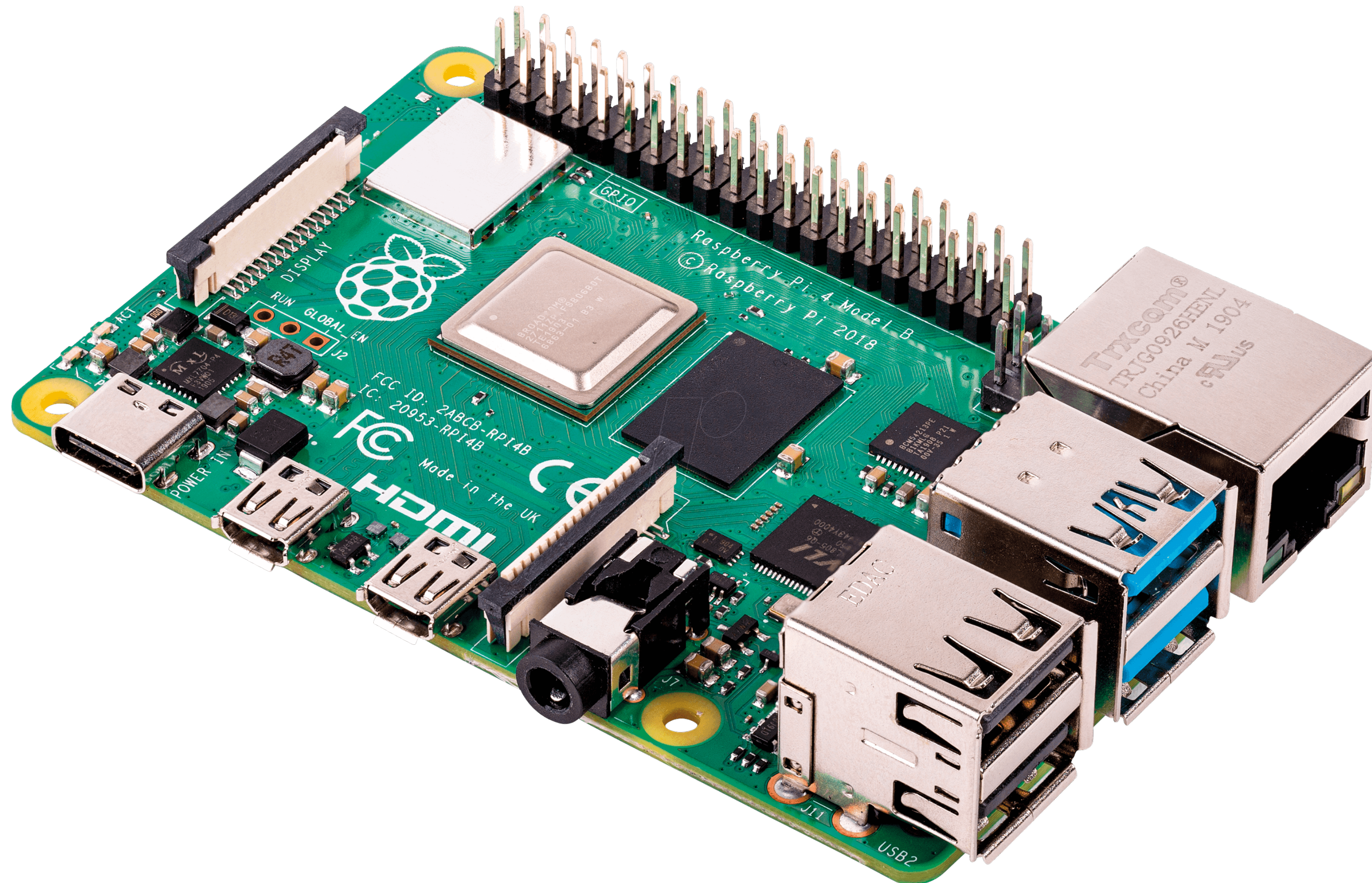


# BBC micro:bit



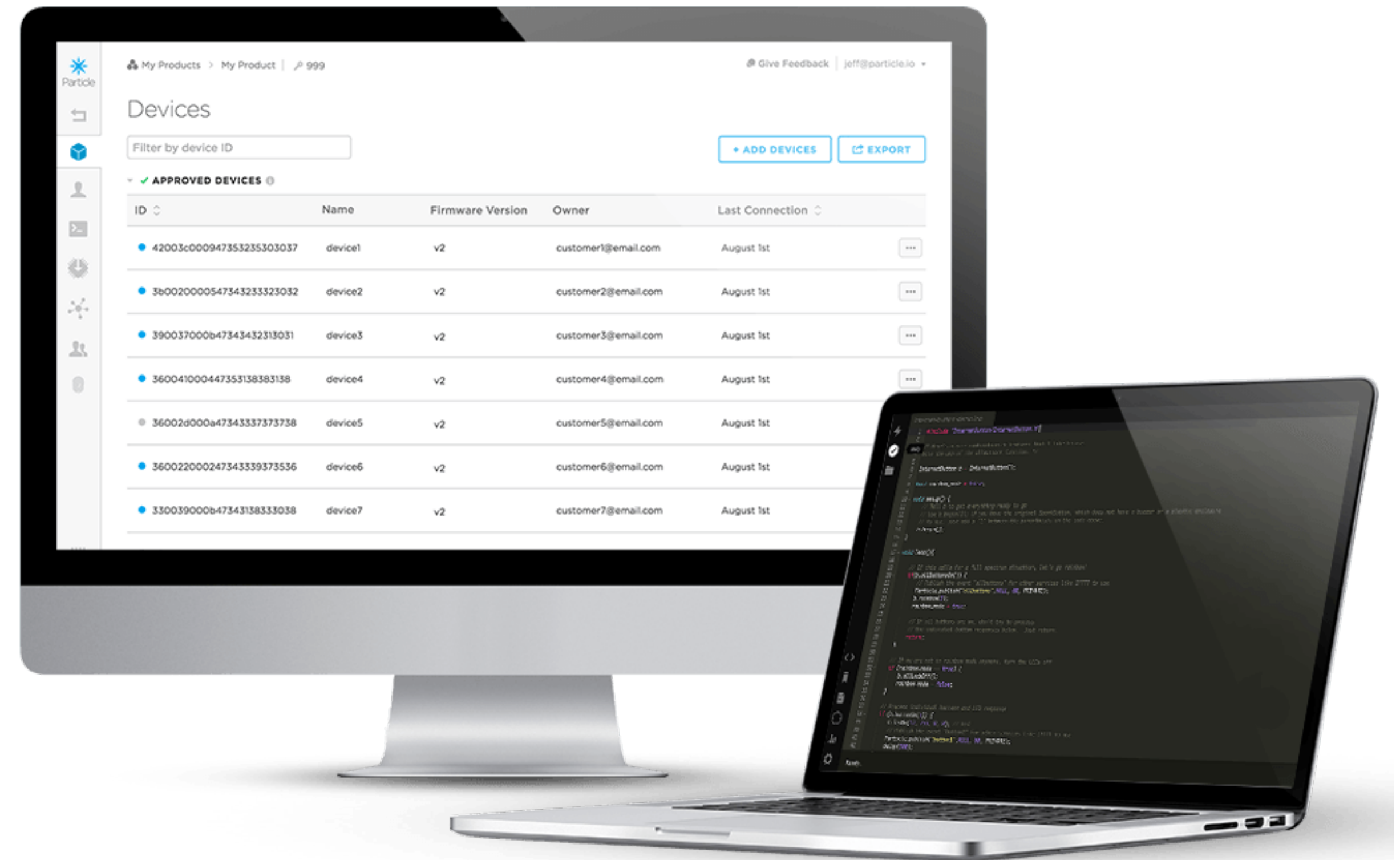
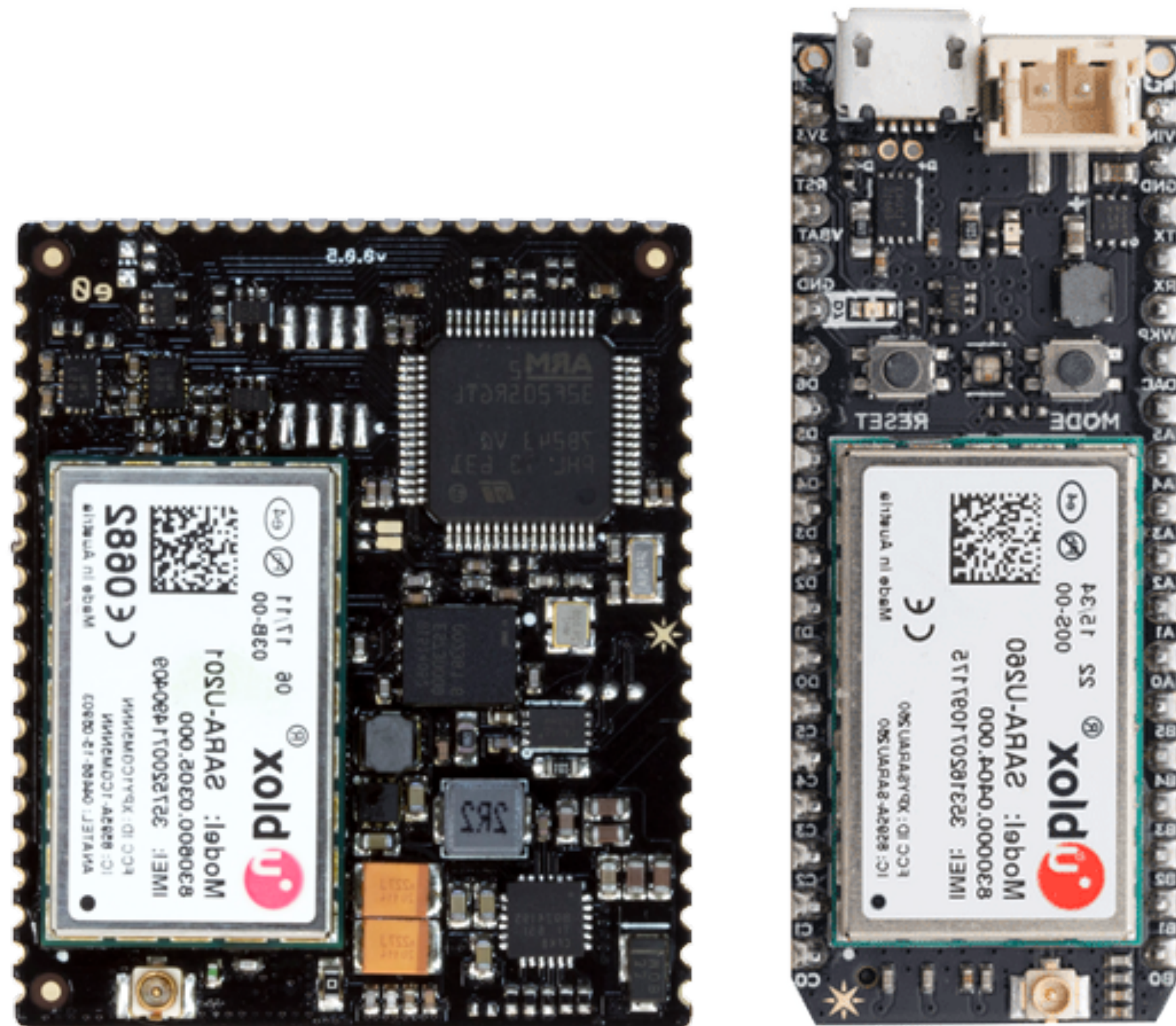


# Raspberry Pi





# Particle





## CHAPTER 44

# Prototyping Peripherals: Electronics



In

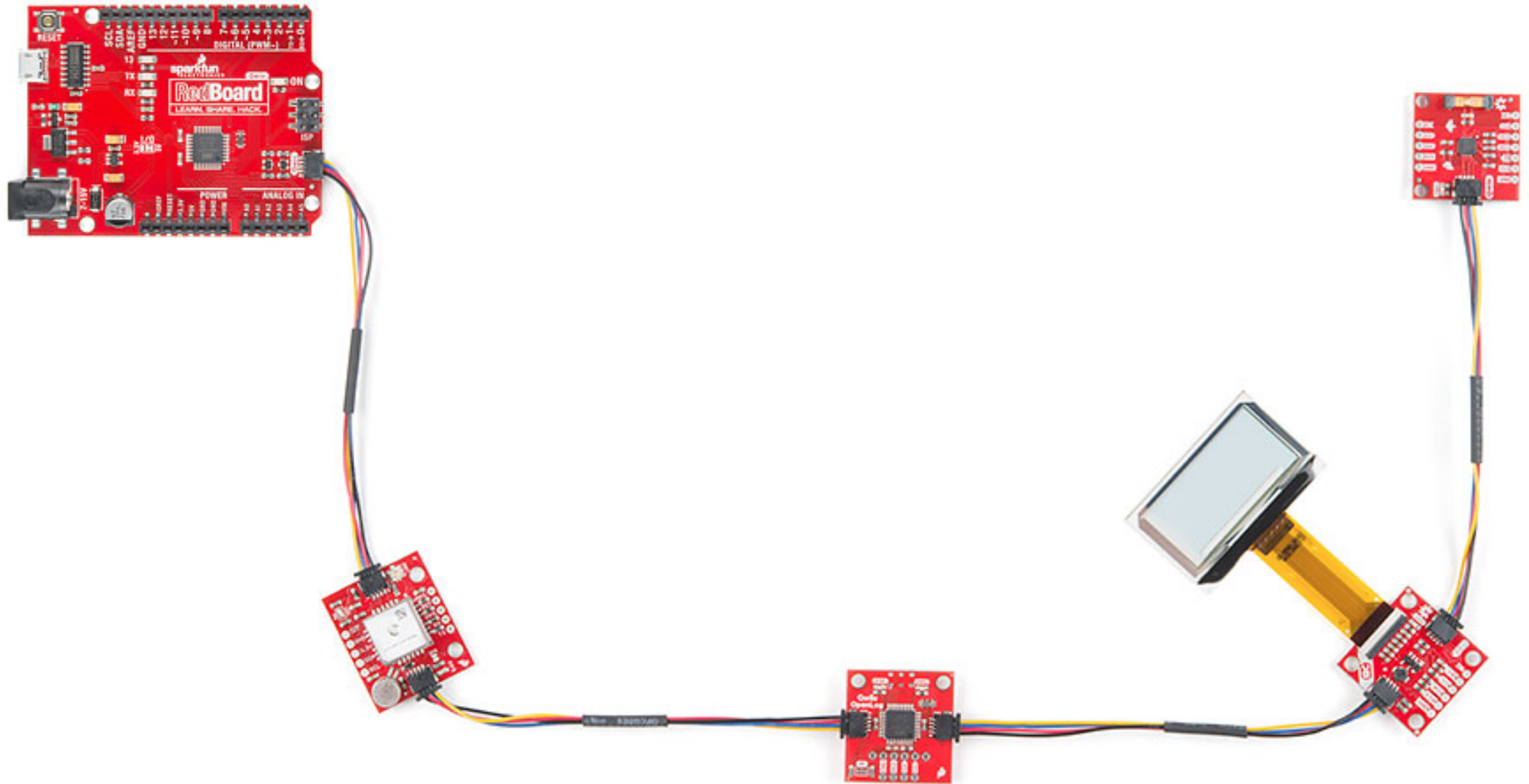
Out

Digital



Analog







**Peripherals:**  
LED, switch,  
colour sensor

**Form**  
3D design,  
3D print,  
casting

**Function:**  
Programmed  
Arduino Board

**Form**  
2D design, laser cutter





## CHAPTER 45

# Arduino Crash Course



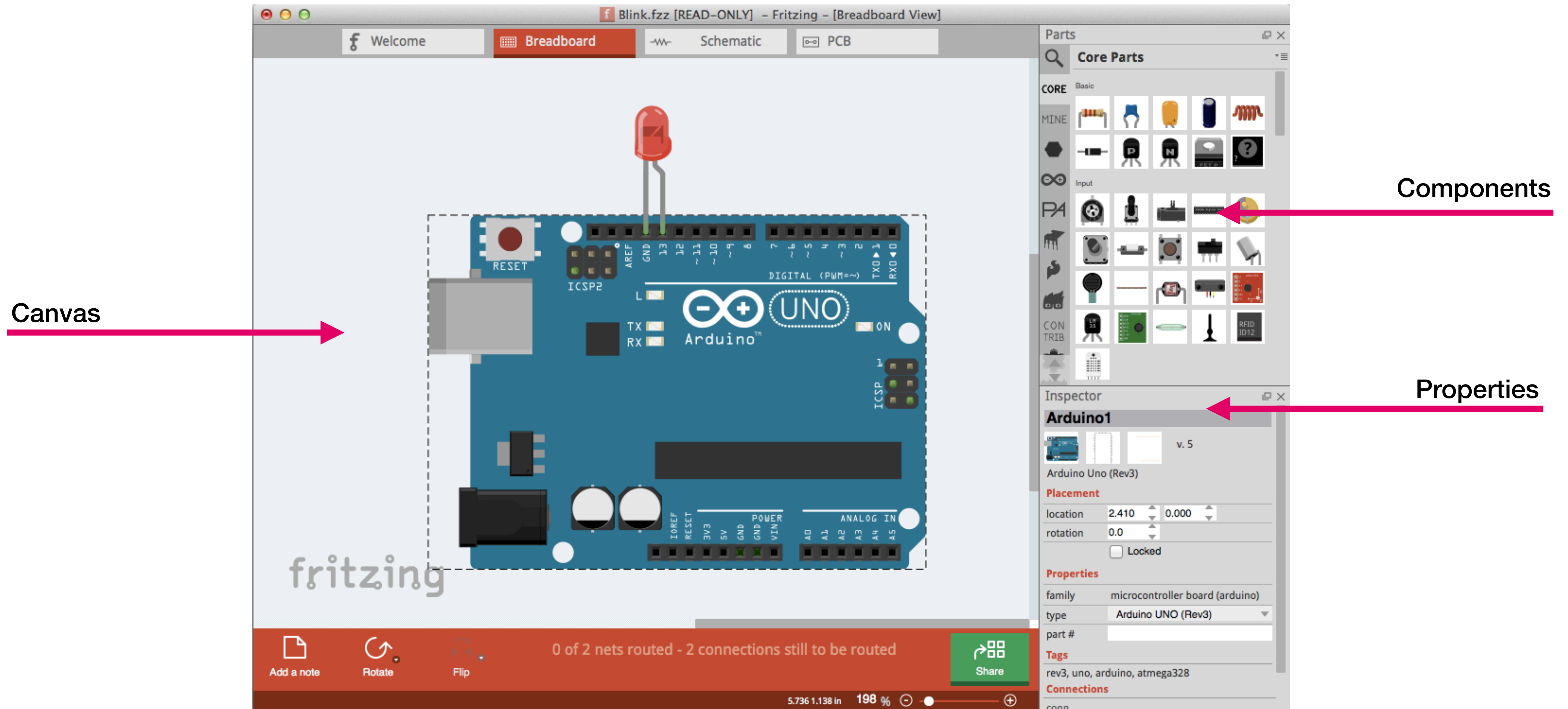
## CHAPTER 46

# Electronics Prototyping: Fritzing



- Open-source initiative
- Release: 2008
- Electronic design automation (EDA) tool for non-engineers
- Key advance:
  - Drag & Drop realistic components
  - Software generated PCB design for manufacturing an Arduino shield

# Interface





# Demo Fritzing

