

Running the Processing environment on ARM SBCs

**Lessons learned & what's missing for
having an Arduino equivalent on top
of Linux**

**Gottfried Haider
@mrgohai**

Processing

Processing

“a flexible software sketchbook and a language for learning how to code within the context of the visual arts”

Processing

“a flexible software sketchbook and a language for learning how to code within the context of the visual arts”

started 2001 by Casey Reas & Ben Fry at MIT Media Lab

Processing

“a flexible software sketchbook and a language for learning how to code within the context of the visual arts”

started 2001 by Casey Reas & Ben Fry at MIT Media Lab

since developed by a group of programmers, educators, designers (LGPL/GPL)

Processing

“a flexible software sketchbook and a language for learning how to code within the context of the visual arts”

started 2001 by Casey Reas & Ben Fry at MIT Media Lab

since developed by a group of programmers, educators, designers (LGPL/GPL)

based on Java (but also p5.js, Processing.py)

Demo

Processing

“a flexible software sketchbook and a language for learning how to code within the context of the visual arts”

started 2001 by Casey Reas & Ben Fry at MIT Media Lab

since developed by a group of programmers, educators, designers (LGPL/GPL)

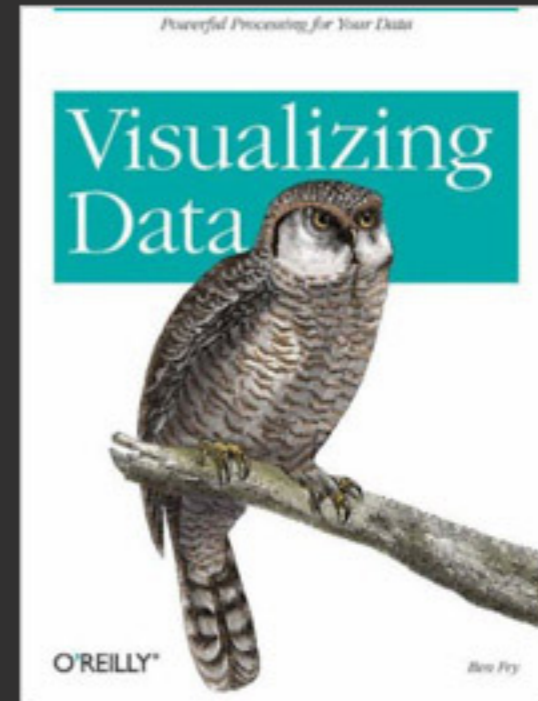
based on Java (but also p5.js, Processing.py)

many resources, especially for education



Processing
A Programming
Handbook for
Visual Designers
and Artists

Revised by John Weeks
Casey Reas
Ben Fry

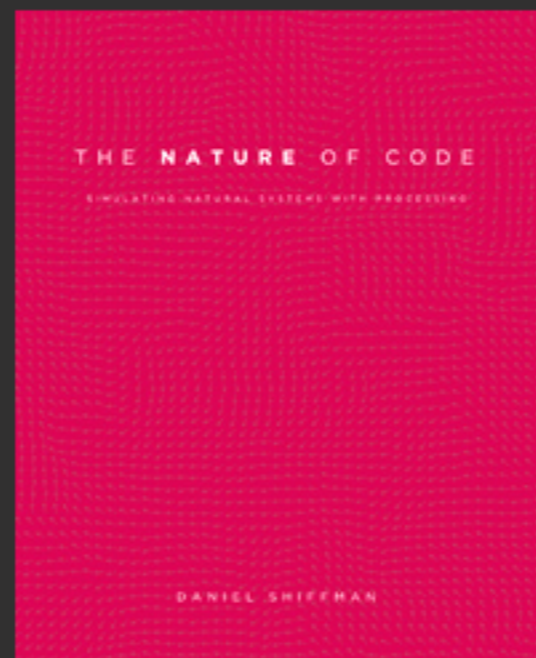


Powerful Processing for Your Data

Visualizing
Data

O'REILLY™

Ben Fry



THE NATURE OF CODE

SIMULATING NATURAL SYSTEMS WITH PROCESSING

DANIEL SHIFFMAN



Second Edition

Make:
Getting
Started with
Processing

A Hands-On Introduction to Making
Interactive Graphics

Casey Reas & Ben Fry

Processing

“a flexible software sketchbook and a language for learning how to code within the context of the visual arts”

started 2001 by Casey Reas & Ben Fry at MIT Media Lab

since developed by a group of programmers, educators, designers (LGPL/GPL)

based on Java (but also p5.js, Processing.py)

many resources, especially for education

<https://github.com/processing/>

Processing on the Raspberry Pi

Processing on the Raspberry Pi

Google Summer of Code

Processing on the Raspberry Pi

Google Summer of Code

Motivation:

- **accessibility through price point (roughly equivalent to Arduino)**

Processing on the Raspberry Pi

Google Summer of Code

Motivation:

- accessibility through price point (roughly equivalent to Arduino)
- more powerful than AVR (FFT, computer vision, networking)

Processing on the Raspberry Pi

Google Summer of Code

Motivation:

- accessibility through price point (roughly equivalent to Arduino)
- more powerful than AVR (FFT, computer vision, networking)
- prospect of a fully open 3D graphics stack (GLES2)

Processing on the Raspberry Pi

Google Summer of Code

Motivation:

- accessibility through price point (roughly equivalent to Arduino)
- more powerful than AVR (FFT, computer vision, networking)
- prospect of a fully open 3D graphics stack (GL ES2)
- access to large repositories of FLOSS software - great to introduce users to the benefits of working with existing communities (but: ARMv6)

Processing on the Raspberry Pi

Google Summer of Code

Results:

- shipped in Processing 3.0.1 - try it out!
- 3D works with the current, closed-source GLES2 driver - thanks to JOGL & Xerxes Rånby
- also works with the in-progress DRM & Mesa Gallium driver by Eric Anholt (image @ <http://sukzessiv.net/~gohai/vc4-buildbot/build/>)
- can build on x86, deploy on ARMv6
- Hardware I/O library!
- should run on any ARMv6+ hard-float SBC (minus GLES2 upbringing)

Processing on the Raspberry Pi

Hardware I/O

	processing.io.*	Arduino
UART	X	X
GPIO	X	X
I2C	X	X
SPI	X	X
PWM	~	X
LED	X	

Processing on the Raspberry Pi

Hardware I/O

	processing.io.*	Arduino
UART	X	X
GPIO	X	X
I2C	X	X
SPI	X	X
PWM	~	X
LED	X	

Caveats (non hard-realtime OS, etc) - *best effort*

Processing on the Raspberry Pi

Hardware I/O - what's missing? (and why is this in the mobile & embedded devroom?!)

Processing on the Raspberry Pi

Hardware I/O - what's missing? (and why is this in the mobile & embedded devroom?!)

<https://github.com/gohai/arduino-like-linux>

Comments?

Processing on the Raspberry Pi

Hardware I/O - Wishlist

- **Software PWM**
- **Runtime pullup configuration**
- **Make PWM sysfs export show up in udev**
- **Race-free export of GPIO, PWM in sysfs?**
- **A way to get from PWM channel to GPIO number**

Processing on the Raspberry Pi

Hardware I/O - Wishlist

- Software PWM

kernel-land implementation using high-resolution timers

Bill Gatliff had a patch in 2010 - there are others e.g. i2c-gpio

wish: `/sys/class/gpio/gpioN/software_pwm`

- Runtime pullup configuration

- Make PWM sysfs export show up in udev

- Race-free export of GPIO, PWM in sysfs?

- A way to get from PWM channel to GPIO number

Processing on the Raspberry Pi

Hardware I/O - Wishlist

- Software PWM

- Runtime pullup configuration

currently only possible through device tree overlays w/ pinconf
vs. `digitalWrite()` on INPUTs in Arduino

wish: `/sys/class/gpio/gpioN/bias` (behind config option?)

- Make PWM sysfs export show up in udev

- Race-free export of GPIO, PWM in sysfs?

- A way to get from PWM channel to GPIO number

Processing on the Raspberry Pi

Hardware I/O - Wishlist

- Software PWM

- Runtime pullup configuration

- **Make PWM sysfs export show up in udev**

writing to `/sys/class/pwm/.../export` doesn't trigger events for udev
(works w/ `/sys/class/gpio/export`)

hence currently root required

- Race-free export of GPIO, PWM in sysfs?

- A way to get from PWM channel to GPIO number

Processing on the Raspberry Pi

Hardware I/O - Wishlist

- Software PWM
- Runtime pullup configuration
- Make PWM sysfs export show up in udev
- Race-free export of GPIO, PWM in sysfs?

exporting a GPIO pin needs to wait for udev to do its thing

currently: `Thread.sleep(500)`

perhaps: default owner & mode taken from *export* node? ideas?

- A way to get from PWM channel to GPIO number

Processing on the Raspberry Pi

Hardware I/O - Wishlist

- Software PWM
- Runtime pullup configuration
- Make PWM sysfs export show up in udev
- Race-free export of GPIO, PWM in sysfs?
- **A way to get from PWM channel to GPIO numbers**

sysfs doesn't tell you this atm

Processing on the Raspberry Pi

Hardware I/O - Any help greatly appreciated :)

<https://github.com/gohai/arduino-like-linux>

Comments?

Thank you FOSDEM!

Gottfried Haider
@mrgohai