Dramaläb


#3
Hacking
Session
ENJOY THE NEW PROJECT BASED SESSIONS
#3 HACK A PROJECTOR,
PLAY IT WITH SENSORS!
06.12.14 - SER F1.FREIRAUM
14:00 - 19:00

TODAY

- working with Processing
- combining Arduino and Processing

processing.org

Processing | p5.js | Processing.py

Processing 2

Cover

Download

Exhibition

Reference
Libraries
Tools
Environment

Tutorials
Examples
Books

Overview
People
Foundation

Shop

» Forum
» GitHub
» Issues
» Wiki
» FAQ
» Twitter
» Facebook

» Download Processing
» Play With Examples
» Browse Tutorials

Processing is a programming language, development environment, and online community. Since 2001, Processing has promoted software literacy within the visual arts and visual literacy within technology. Initially created to serve as a software sketchbook and to teach computer programming fundamentals within a visual context, Processing evolved into a development tool for professionals. Today, there are tens of thousands of students, artists, designers, researchers, and hobbyists who use Processing for learning, prototyping, and production.

» Free to download and open source

» Interactive programs with 2D, 3D or PDF output

» OpenGL integration for accelerated 3D

» For GNU/Linux, Mac OS X, and Windows

» Over 100 libraries extend the core software

» Well documented, with many books available

» Hello Processing Videos

This first look at Processing for total beginners is an introduction to programming in the context of the visual arts. Short video lessons

» Exhibition

Filament Sculptures
by Lia

Fall in Love - Phantogram
by Timothy Saccenti and Joshua Davis

Keyflies
by Miles Peyton

processing.org

Processing | p5.js | Processing.py

Processing 2

Cover

Download

Exhibition

Reference

Libraries

Tools

Environment

Tutorials

Examples

Books

Overview

People

Foundation

Shop

» Forum
» GitHub
» Issues
» Wiki
» FAQ
» Twitter
» Facebook

» Download Processing
» Play With Examples
» Browse Tutorials

Processing is a programming language, development environment, and online community. Since 2001, Processing has promoted software literacy within the visual arts and visual literacy within technology. Initially created to serve as a software sketchbook and to teach computer programming fundamentals within a visual context, Processing evolved into a development tool for professionals. Today, there are tens of thousands of students, artists, designers, researchers, and hobbyists who use Processing for learning, prototyping, and production.

» Free to download and open source

» Interactive programs with 2D, 3D or PDF output

» OpenGL integration for accelerated 3D

» For GNU/Linux, Mac OS X, and Windows

» Over 100 libraries extend the core software

» Well documented, with many books available

» Hello Processing Videos

This first look at Processing for total beginners is an introduction to programming in the context of the visual arts. Short video lessons

» Exhibition

Filament Sculptures
by Lia

Fall in Love - Phantogram
by Timothy Saccenti and Joshua Davis

Keyflies
by Miles Peyton

processing.org

Processing | p5.js | Processing.py

Processing 2

Reference. The Processing Language was designed to facilitate the creation of sophisticated visual structures.

Structure

() (parentheses)
, (comma)
. (dot)
/* */ (multiline comment)
/** */ (doc comment)
// (comment)
; (semicolon)
= (assign)
[] (array access)
{} (curly braces)
catch
class
draw()
exit()
extends
false
final
implements
import
loop()
new
noLoop()
null
popStyle()
private
public
pushStyle()

Shape

createShape()
loadShape()
PShape

2D Primitives
arc()
ellipse()
line()
point()
quad()
rect()
triangle()

Curves
bezier()
bezierDetail()
bezierPoint()
bezierTangent()
curve()
curveDetail()
curvePoint()
curveTangent()
curveTightness()

3D Primitives
box()

Color

Setting
background()
clear()
colorMode()
fill()
noFill()
noStroke()
stroke()

Creating & Reading
alpha()
blue()
brightness()
color()
green()
hue()
lerpColor()
red()
saturation()

Image

createImage()
PImage

Loading & Displaying

processing.org

Processing | p5.js | Processing.py

Processing 2

Libraries. Extend Processing beyond graphics and images into audio, video, and communication with other devices.

The following libraries are created by the Processing Foundation. The PDF Export, Network, Serial, and DXF Export libraries are distributed with Procesing. The Video and Sound libraries need to be downloaded through the Library Manager. Select "Add Library..." from the "Import Library..." submenu within the Sketch menu.

PDF Export
Create PDF files. These vector graphics files can be scaled to any size and printed at high resolutions.

Network
Send and receive data over the Internet through simple clients and servers.

Serial
Send data between Processing and external hardware through serial communication (RS-232).

DXF Export
Create DXF files to save geometry for loading into other programs. It works with triangle-based graphics including polygons, boxes, and spheres.

Video
Read images from a camera, play movie files, and create movies.

Sound
Playback audio files, audio input, synthesize sound, and effects.

Contributions

Contributed Libraries must be downloaded individually. Select "Add Library..." from the "Import Library..." submenu within the Sketch menu. Not all available libraries have been converted to show up in "Add Library...". If a library isn't there, it will need to be installed manually. Follow the How to Install a Contributed Library instructions on the Processing Wiki for more information.

Contributed libraries are developed, documented, and maintained by members of the Processing community. For feedback and support, please post to the Forum. For development discussions post to the Create & Announce Libraries topic. Instructions for creating your own library are on the Processing GitHub site.

3D
Hardware
Sound

processing.org

Processing | p5.js | Processing.py

Processing 2

Tutorials. A collection of step-by-step lessons covering beginner, intermediate, and advanced topics.

Hello Processing
by Daniel Shiffman et al.

Short video lessons introduce coding exercises that lead to designing an interactive drawing program.

Level: Beginner

Getting Started
by Casey Reas and Ben Fry

Welcome to Processing! This introduction covers the basics of writing Processing code.

Level: Beginner

Processing Overview
by Ben Fry and Casey Reas

A little more detailed introduction to the different features of Processing than the Getting Started tutorial.

Level: Beginner

Coordinate System and Shapes
by Daniel Shiffman

Drawing simple shapes and using the coordinate system.

Level: Beginner

Color
by Daniel Shiffman

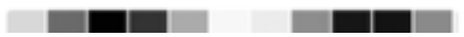
An introduction to digital color.

Level: Beginner

Objects
by Daniel Shiffman

The basics of object-oriented programming.

Level: Beginner



processing.org

Processing | p5.js | Processing.py

Processing 2

Examples. Short, prototypical programs exploring the basics of programming with Processing.

These examples are running online through Processing.js using HTML5 Canvas and WebGL for rendering. There are many more examples included with the Processing application; please look there if you don't find what you're looking for here.

Basic Examples. *Programs about form, data, images, color, typography, and more...*

processing.org

Processing | p5.js | Processing.py

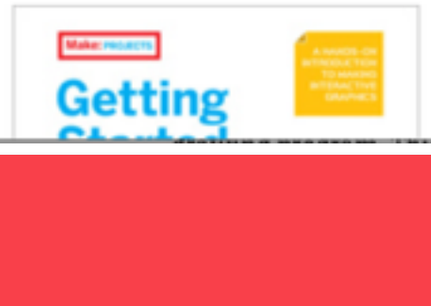
Processing 2

Books. Processing books cover topics from programming basics to visualization. Browse this page to find the right books for you.

Processing: A Programming Handbook for Visual Designers (Second Edition)
Casey Reas and Ben Fry.
Published December 2014, The MIT Press. 720 pages. Hardcover.
» Order from MIT Press
» Order from Amazon

The second edition of the Handbook has been thoroughly updated, influenced by the seven years of Processing being taught in classrooms, computer labs, and studios since the first edition. Every chapter has been revised, and added chapters introduce new ways to work with data and geometry. New "synthesis" chapters offer discussion and worked examples of such topics as sketching with code, modularity, and algorithms. Interviews have been added that cover a wider range of projects and history. "Extension" chapters are now offered online so they can be updated to keep pace with technological developments in such fields as computer vision and electronics.

If you are an educator, you can request a desk/exam copy from the MIT Press website. It's also possible to request a PDF preview.



Getting Started with Processing
Casey Reas and Ben Fry.
Published June 2010, O'Reilly Media. 208 pages. Paperback.
» Order Print/EBook from O'Reilly


sketch_141206a | Processing 2.2.1
Java
sketch_141206a
1

ImageDisplay | Processing 2.2.1

Java

ImageDisplay

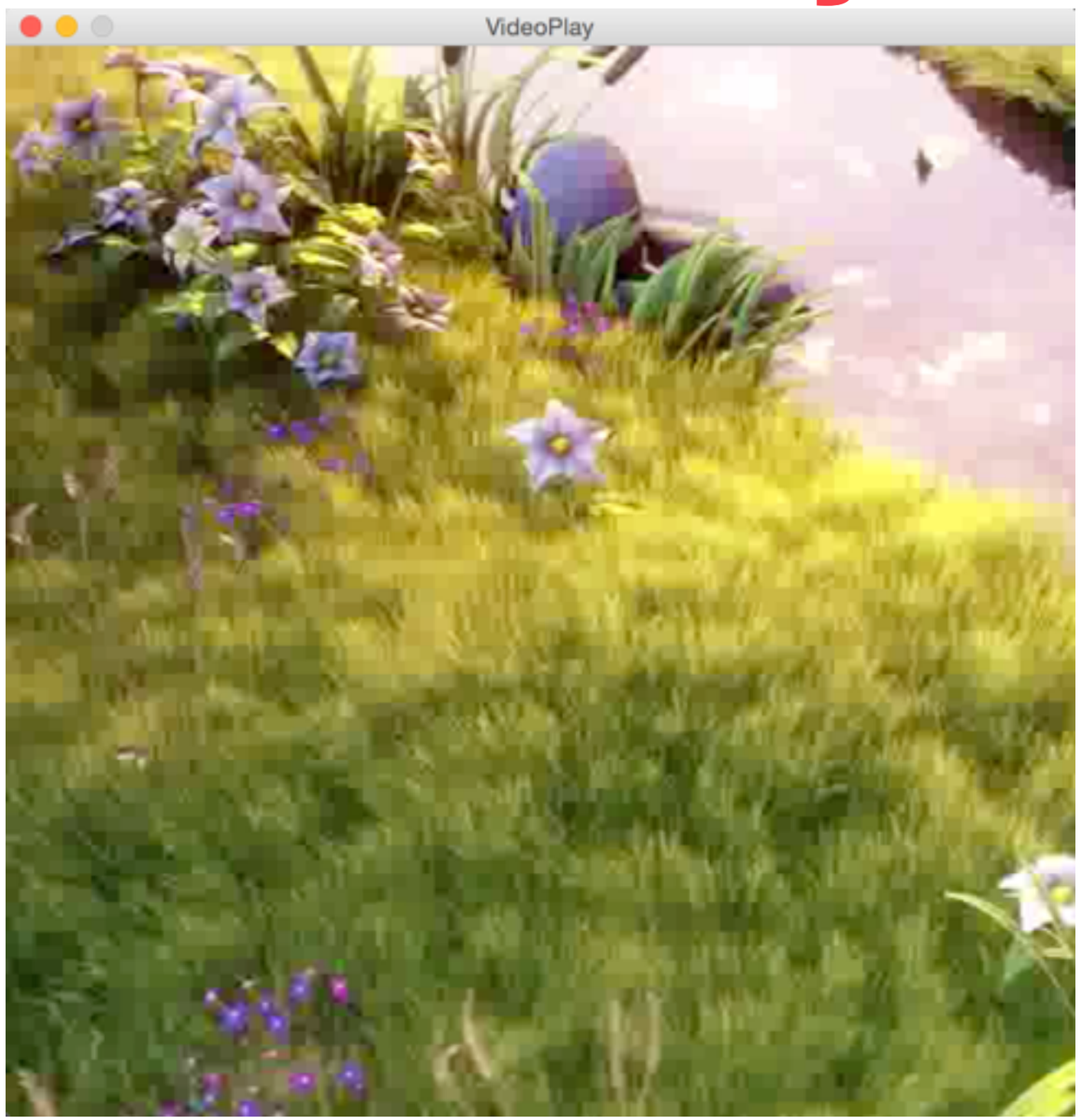
```
PImage img;

void setup() {
  size(780, 300);
  img = loadImage("dramalab.png");
}

void draw() {
  image(img, 40, 40);
}
```

Done exporting.

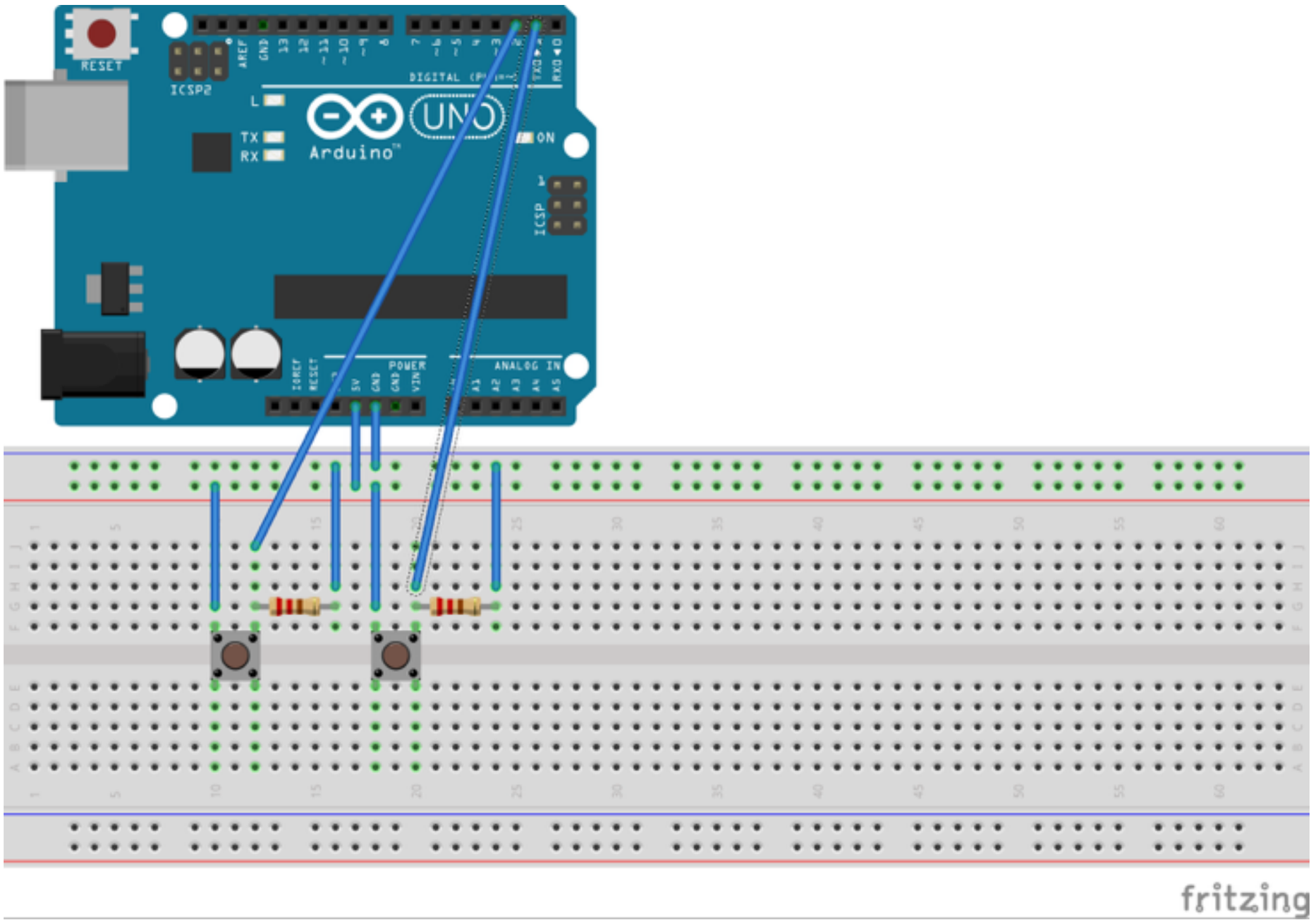
```
No such file or directory
```


Export Application
Export to Application creates double-clickable, standalone applications for the selected plaforms.
Platforms
Windows
Mac OS X
Linux
Full Screen
Full Screen (Present mode)
Show a Stop button
Embed Java
Embed Java for Mac OS X
Embedding Java will make the Mac OS X application larger, but it will be far more likely to work. Users on other platforms will need to install Java 7.
Code Signing
In recent versions of OS X, Apple has introduced the "Gatekeeper" system, which makes it more difficult to run applications like those exported from Processing. This application will be "self-signed" which means that Finder may report that the application is from an "unidentified developer". If the application will not run, try right-clicking the app and selecting Open from the pop-up menu. Or you can visit System Preferences → Security & Privacy and select Allow apps downloaded from: anywhere. To avoid the messages entirely, manually code sign your app. For more information: https://developer.apple.com/developer-id/
Cancel
Export

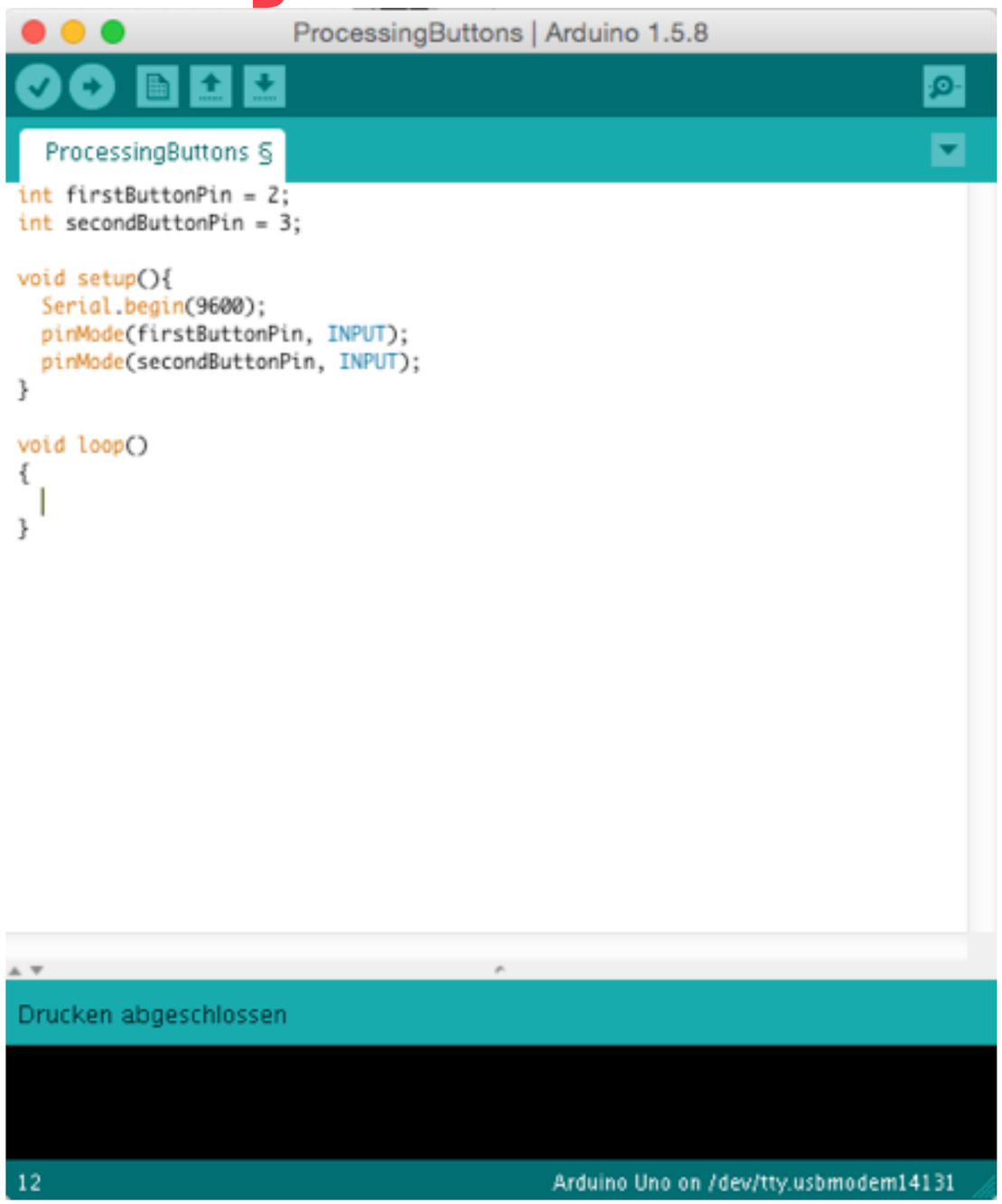
Video Player



Video Player With Arduino

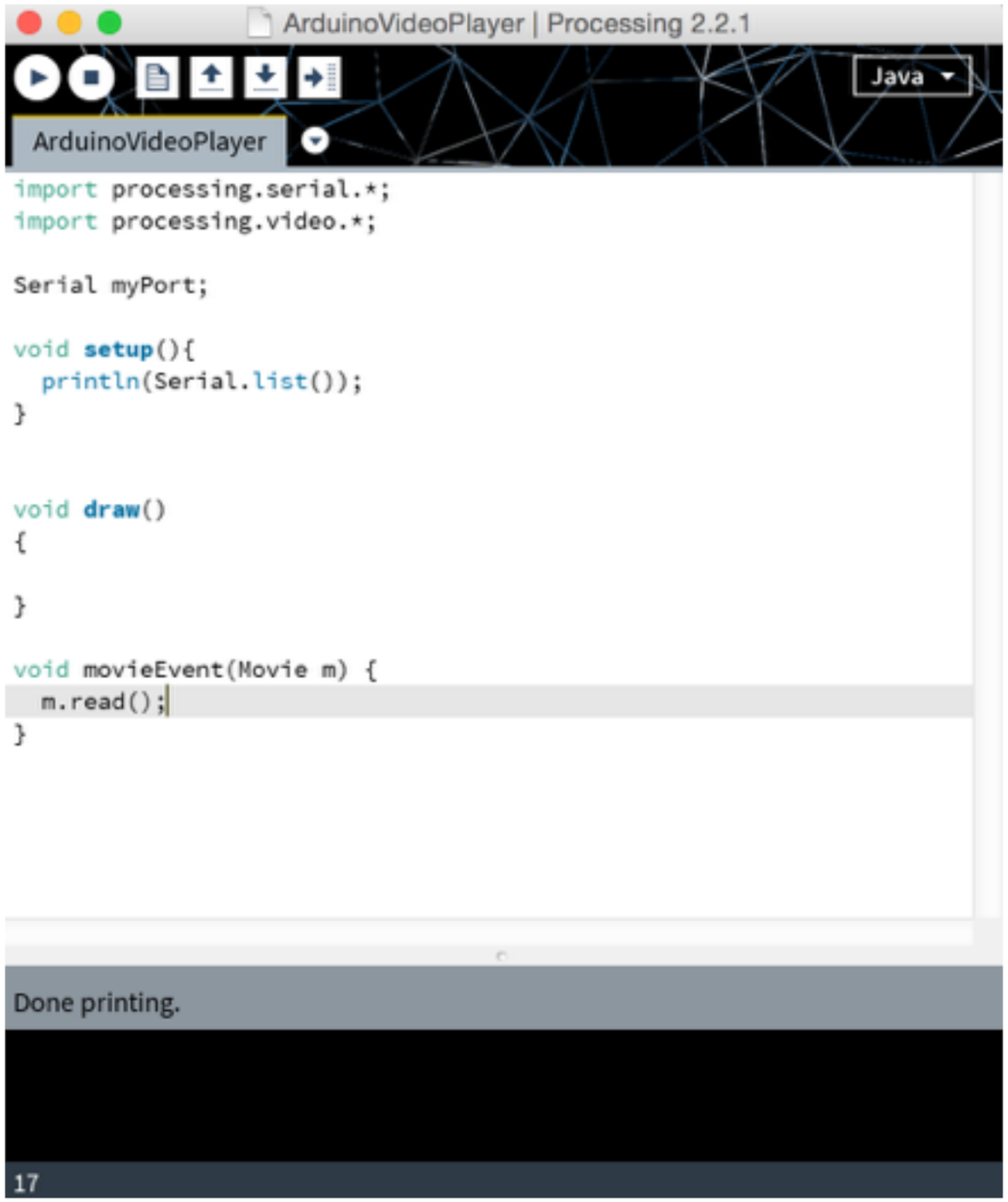


Video Player With Arduino



Max Stricker

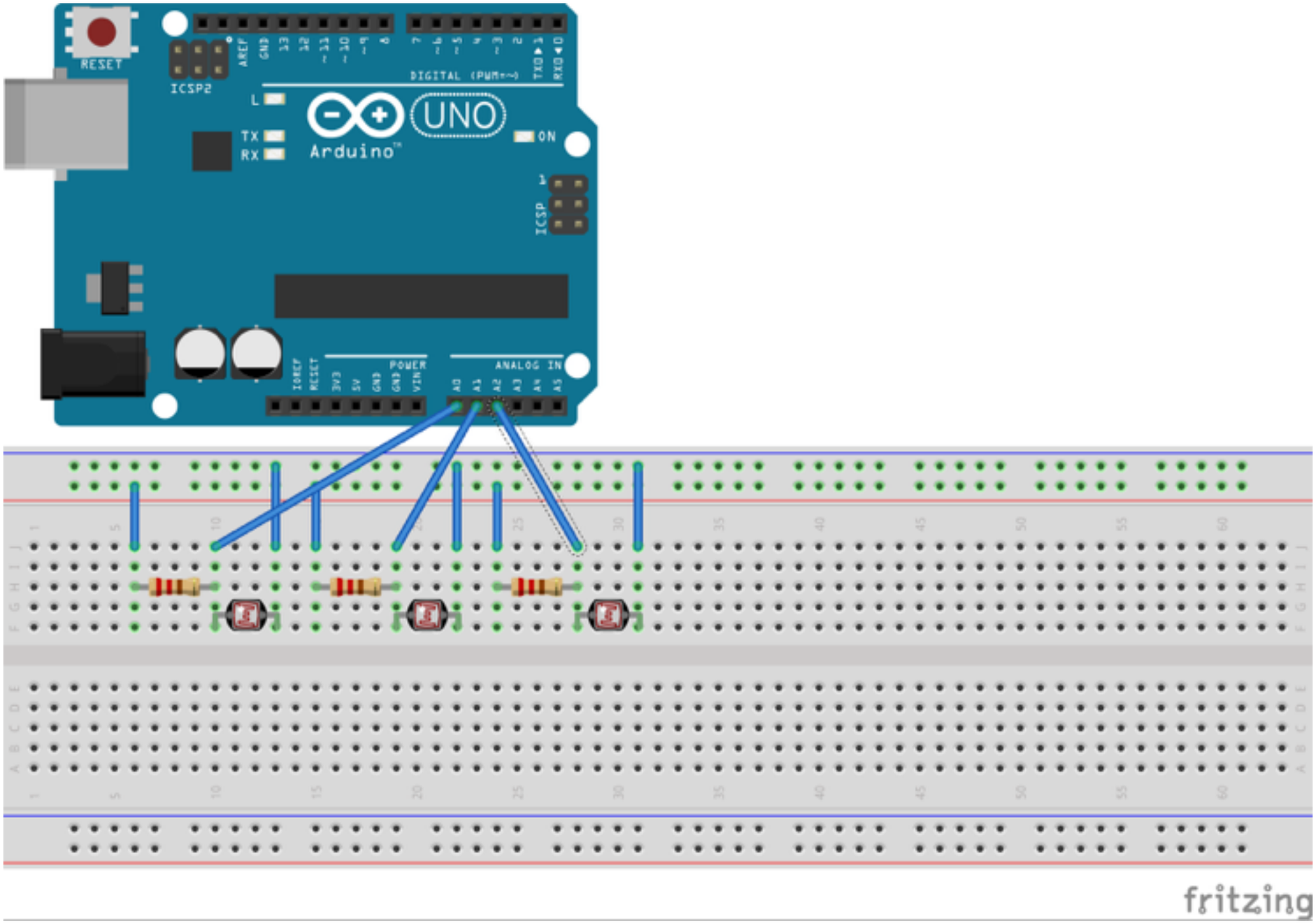
Video Player With Arduino



Max Stricker

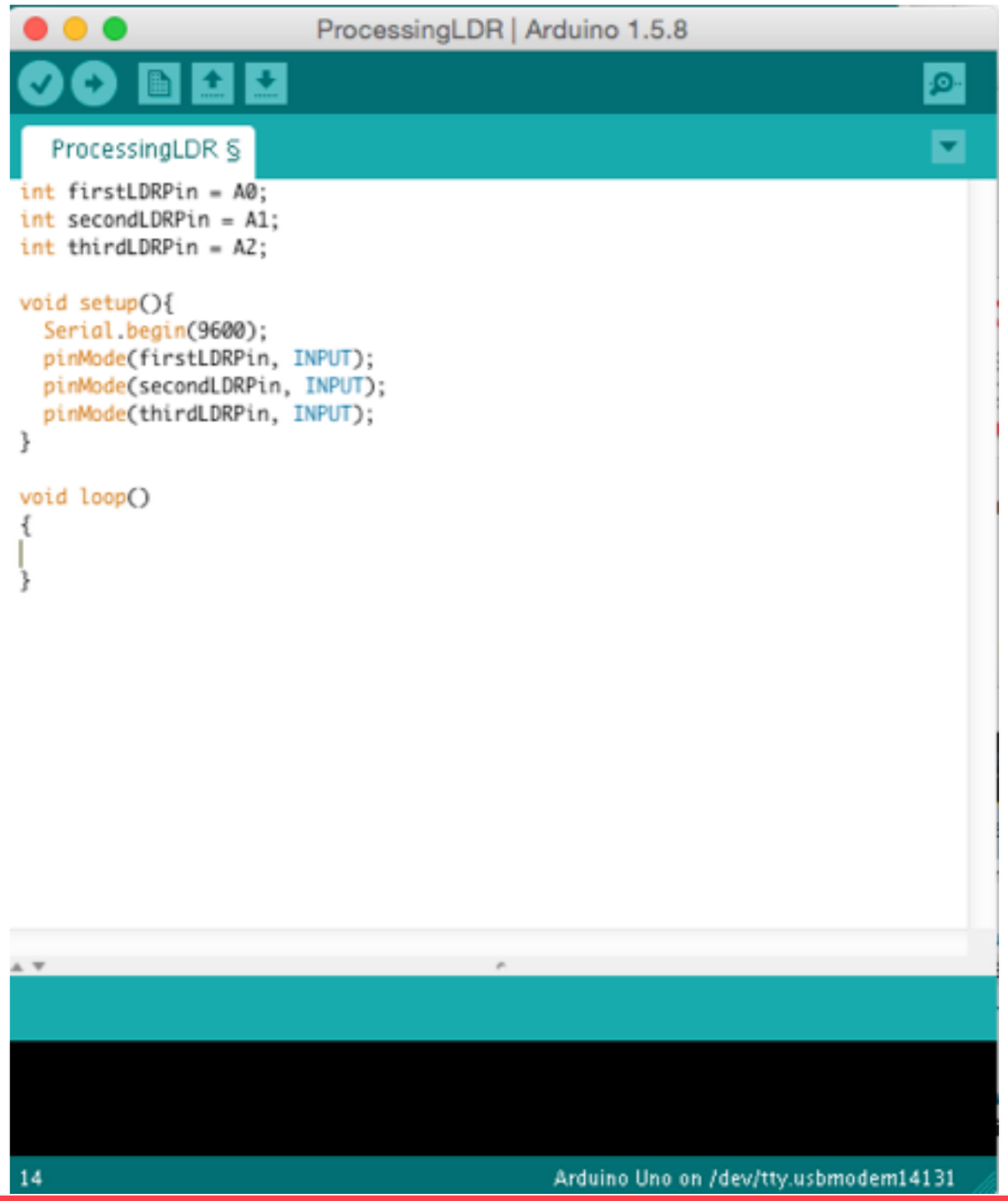
Colored Wall

Colored Wall

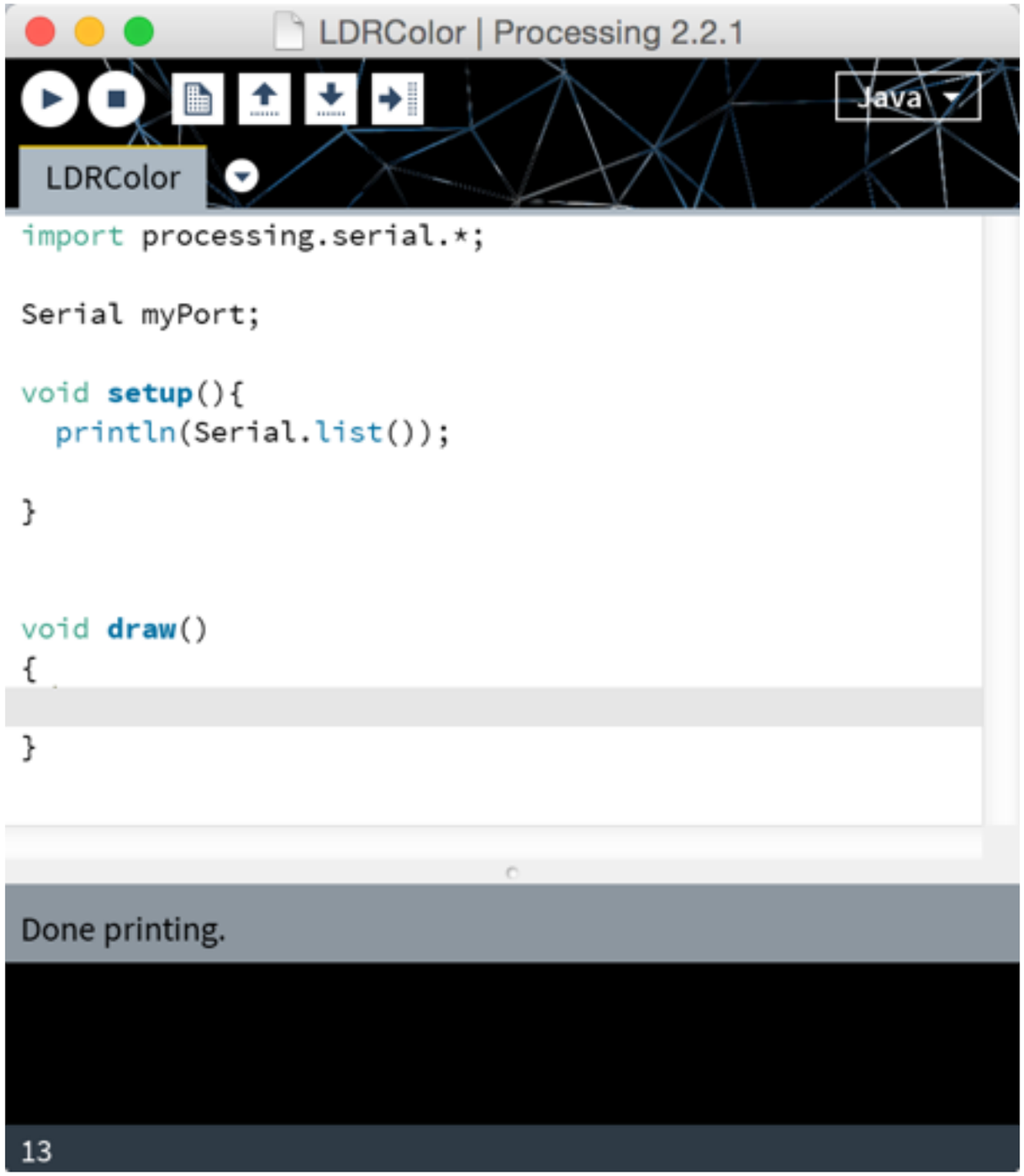


Max Stricker

Colored Wall

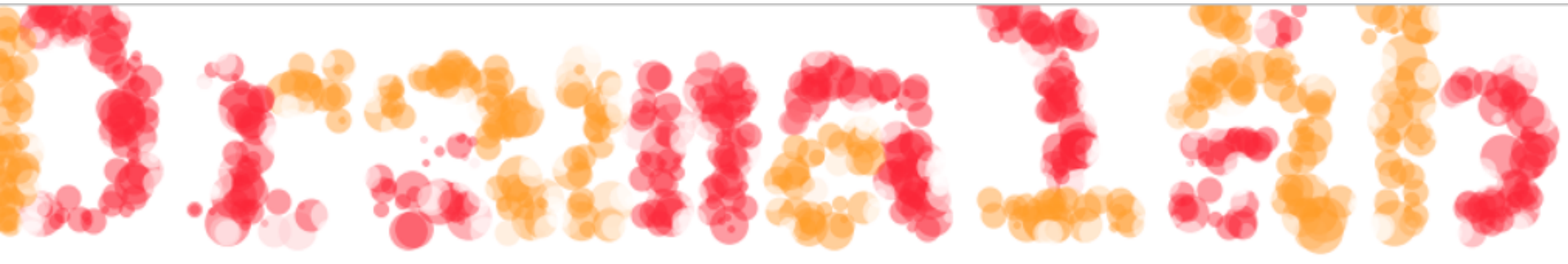


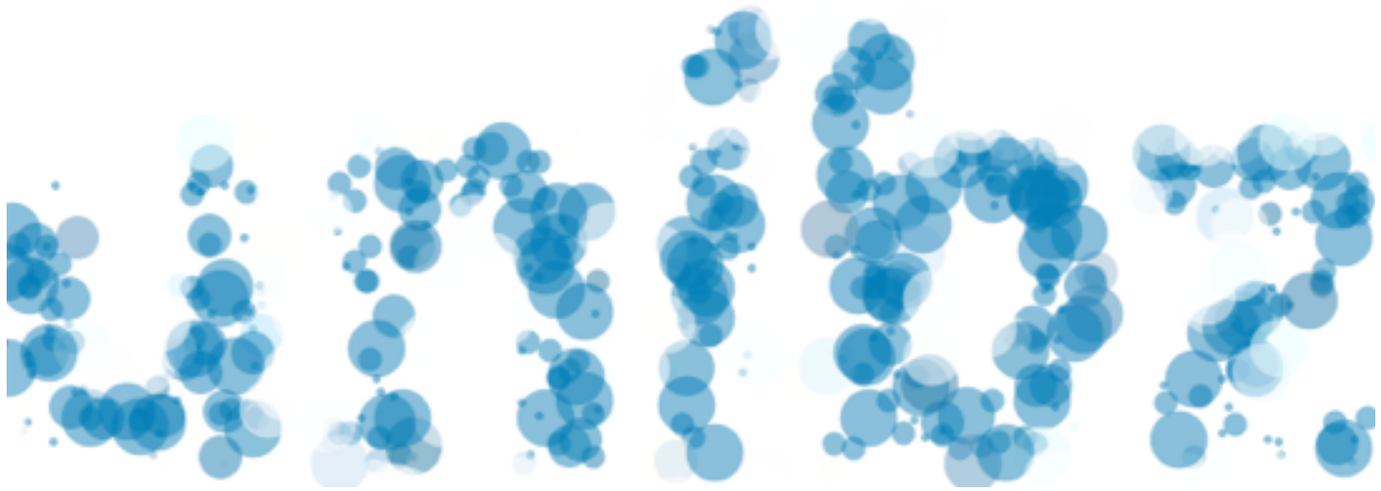
Colored Wall



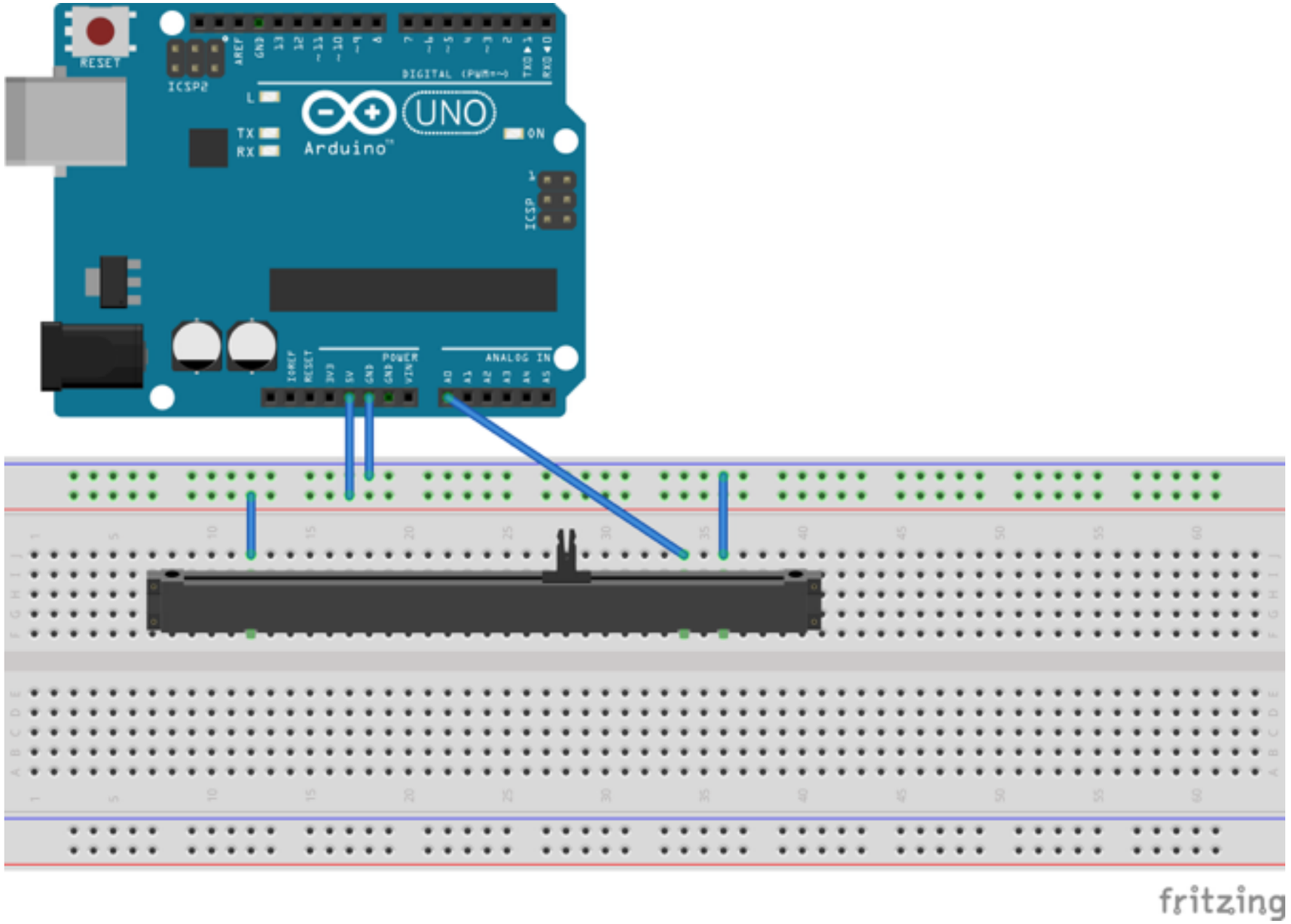
Max Stricker

Dots





Dots


RESET
ICSP2
AREF
GND
DIGITAL (PWM~)
L
TX
RX
UNO
Arduino
ON
ICSP
POWER
ANALOG IN
IOREF
RESET
3V3
5V
GND
GND
VIN
A0
A1
A2
A3
A4
A5
fritzing