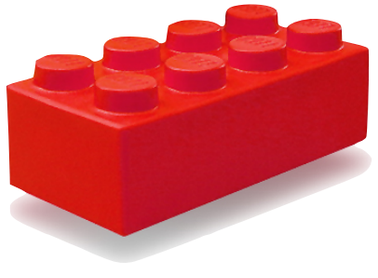
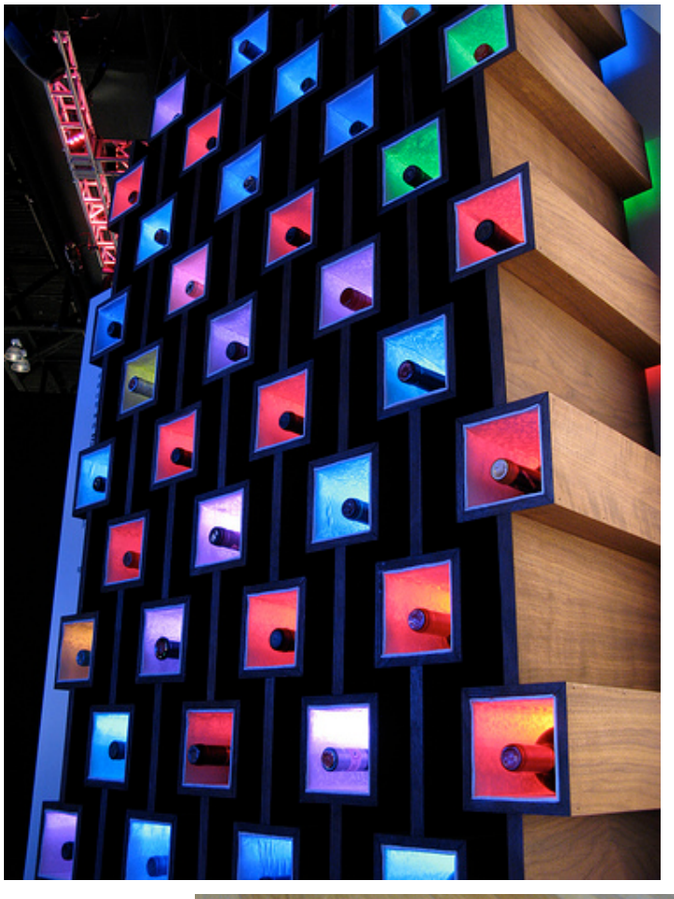
Good Hardware APIs, USB not on Rails, & From 2D→3D



Tod E. Kurt / ThingM

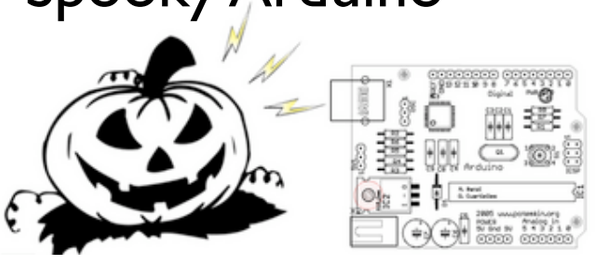
sketching08
27 July 2008

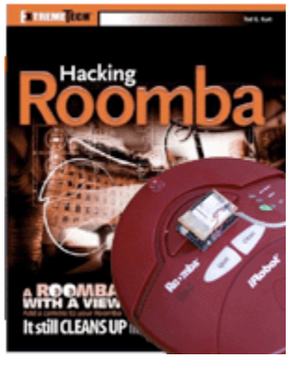


Bionic Arduino

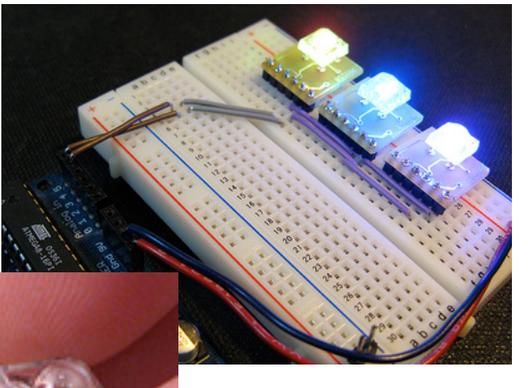
Introduction to Microcontrollers with Arduino

Spooky Arduino

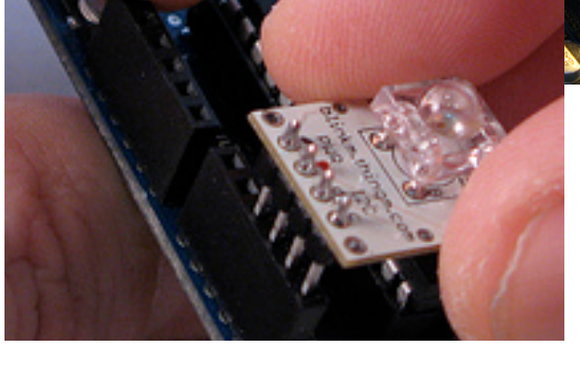




THING M



WineM



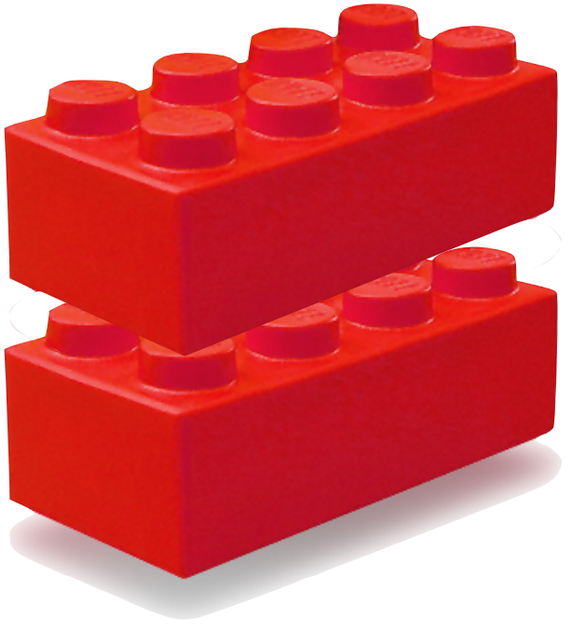
BlinkM

Some of the things I've done.

Good Hardware APIs

- **Hardware API – the physical interface of a device**
  Includes: size, shape, signals, signal ordering, connectors, part placement, labeling, et al

- **Good Hardware APIs:**
  - **use standard & familiar tropes**
  - **are self-describing or obvious**
  - **are flexible to allow novelty**
  - **inspire & be aesthetically pleasing**
  - **or really just, "make sense"**

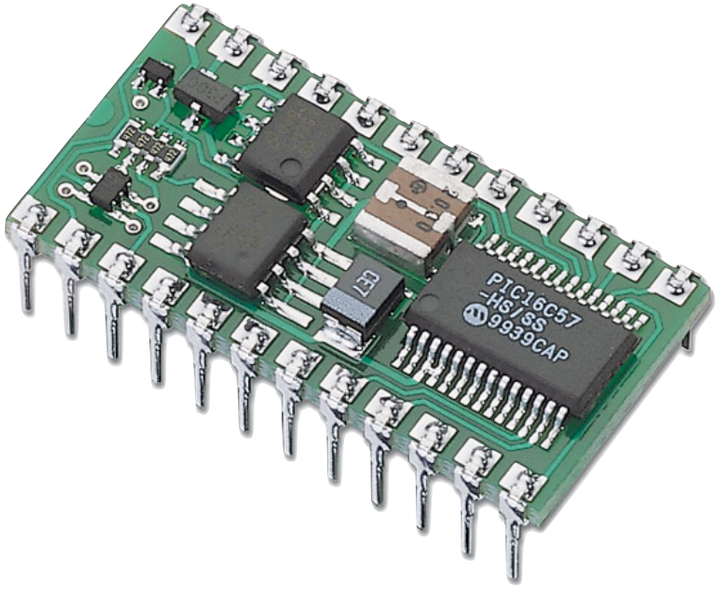


If you're designing hardware with "interfaces" to non-humans, there's still a human component. Does the physical layout, pinout, voltages used, etc. make sense?

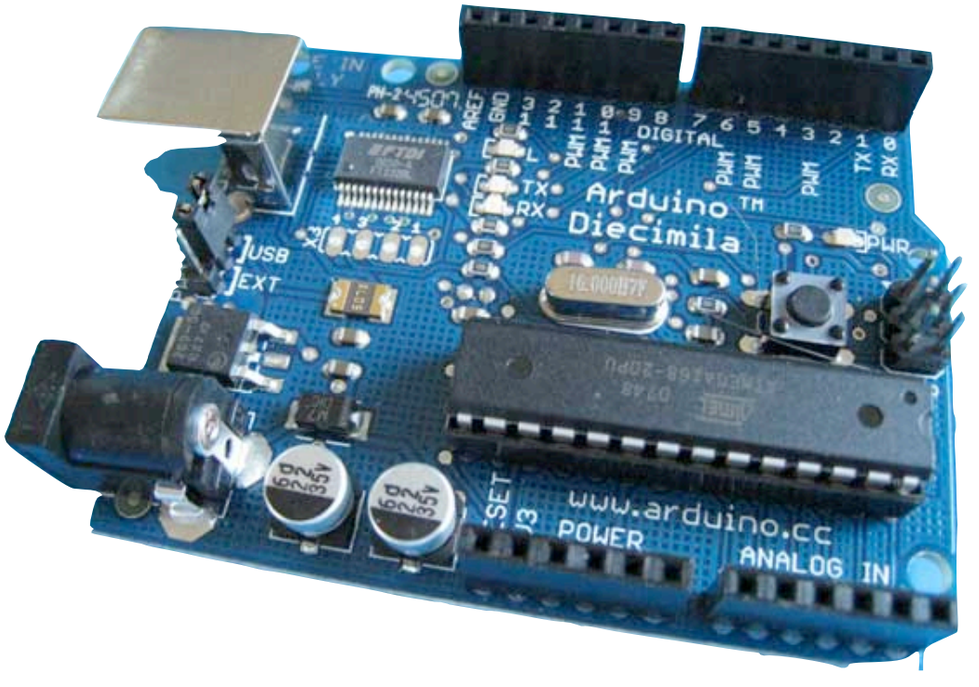
This is pretty obvious to this crowd I think, but I wanted to set the context for my experiences with BlinkM.

Good Hardware APIs

treating the connectivity of two popular microcontrollers as hardware APIs



“jeez, what else do I need to make this work?”



“hey, I know how to plug this in; what else can I hook up to it?”

One of Arduino’s biggest innovations was to treat the concept of “inexpensive microcontroller for beginners” as a system with obvious connectors: USB, DC, wire holes. Instead of a scary looking bug thing.

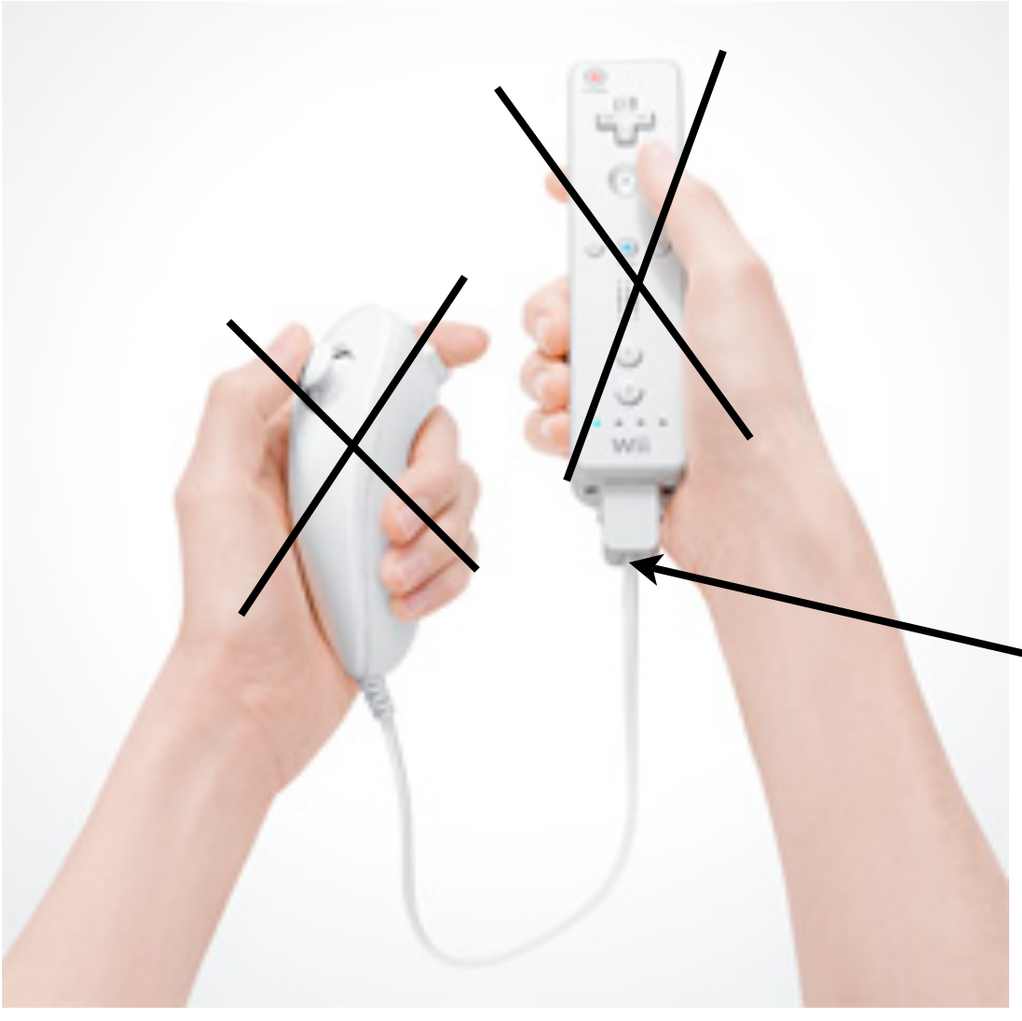
It could be argued that Arduino’s success lies primarily in the fact it has header sockets instead of header pins for its I/O connectors. Arduino’s creators recognized Arduino for the substrate it really is.

Good Hardware APIs

no, not the Wiimote. Or even the Nunchuck.

Let’s take this for example

Good Hardware APIs



Wiimote is I2C master, things that plug into it (nunchuck, etc.) are I2C slaves

I2C Sensors

good!

Nunchuck
(3-axis accelerometer, joystick, buttons)

Motion Plus
(dual 2-axis rate gyroscope)

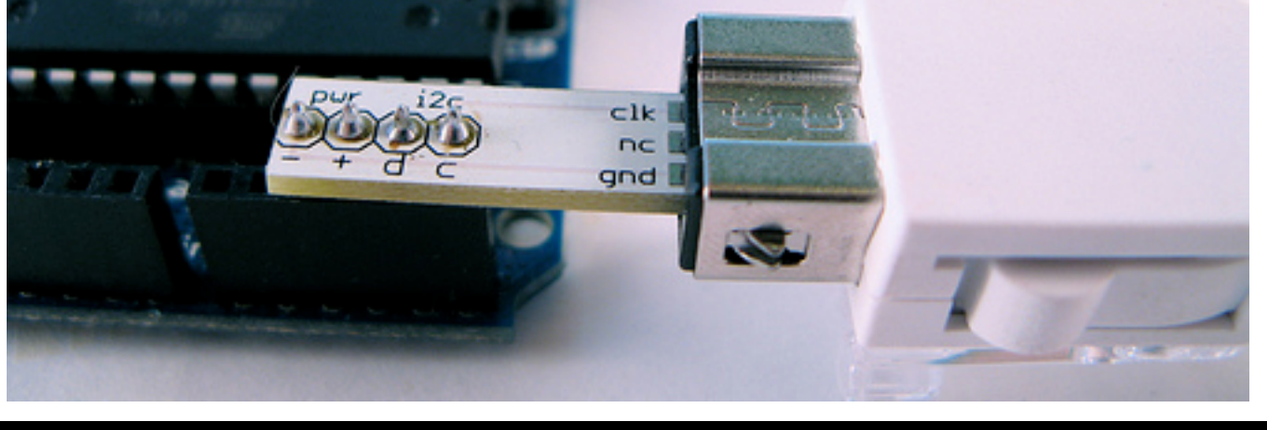
Classic controller
(dual 2-axis joysticks, 12 buttons)

Nintendo rules for doing this. So many awesome sensors

So many I2C devices out there, and now there are 3 good UI devices

“Wiichuck” I2C adapter

makes it easy to use the rad Nintendo I2C sensors



Wiimote connector is weird, but plug thickness is same as compatible with standard PCB thickness.

I’ve sold over 700 of these things.

Wii MotionPlus

cheap gyro,lol

2-axis MEMs rate gyro with I2C interface for $15

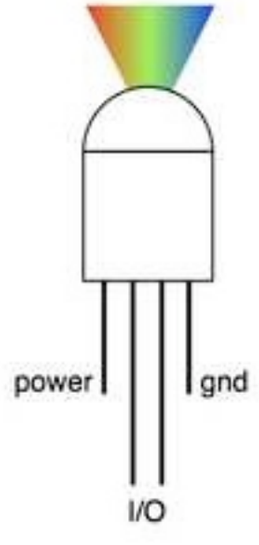
http://www.invensense.com/news/071508.html

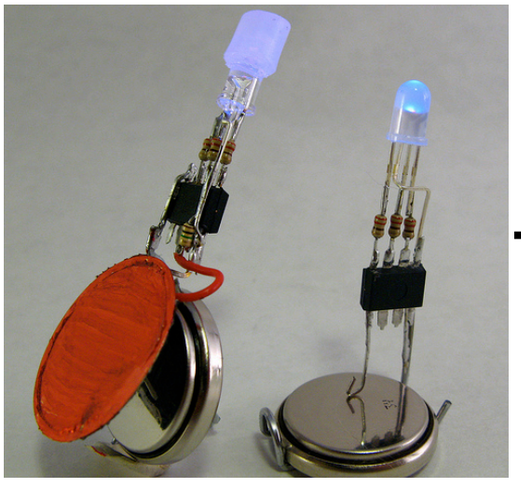
I heard $15 was the price point, but it seems like it should cost more.

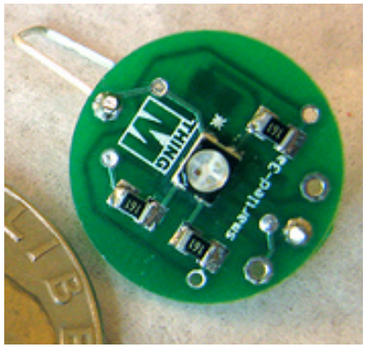
BlinkM

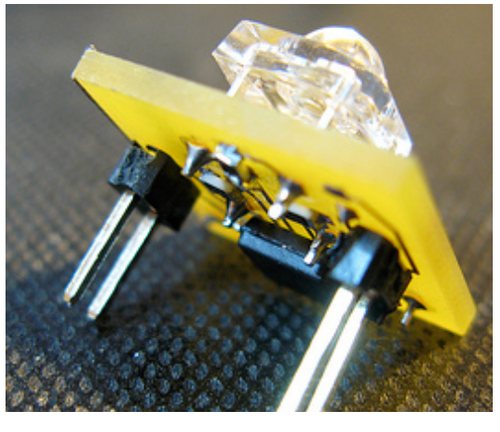
My personal exercise in making a good Hardware API.

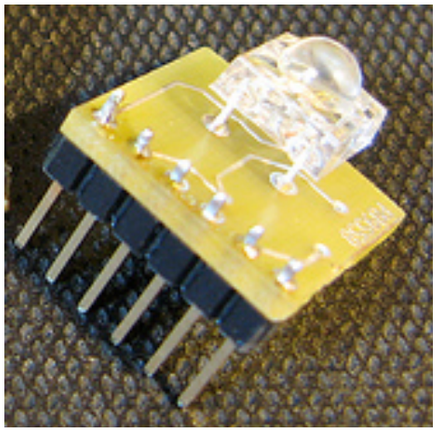
BlinkM Evolution

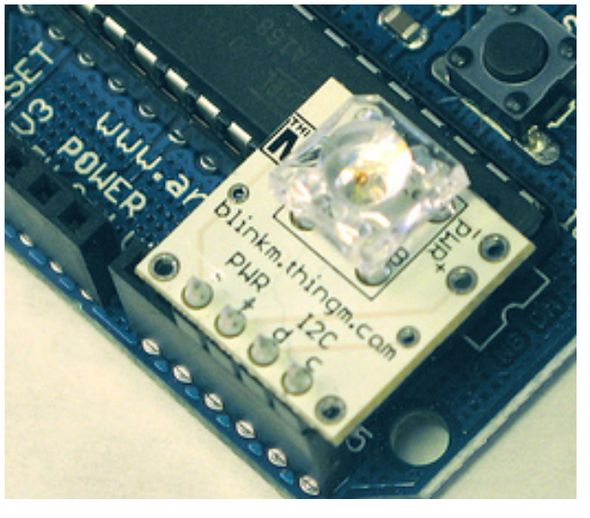










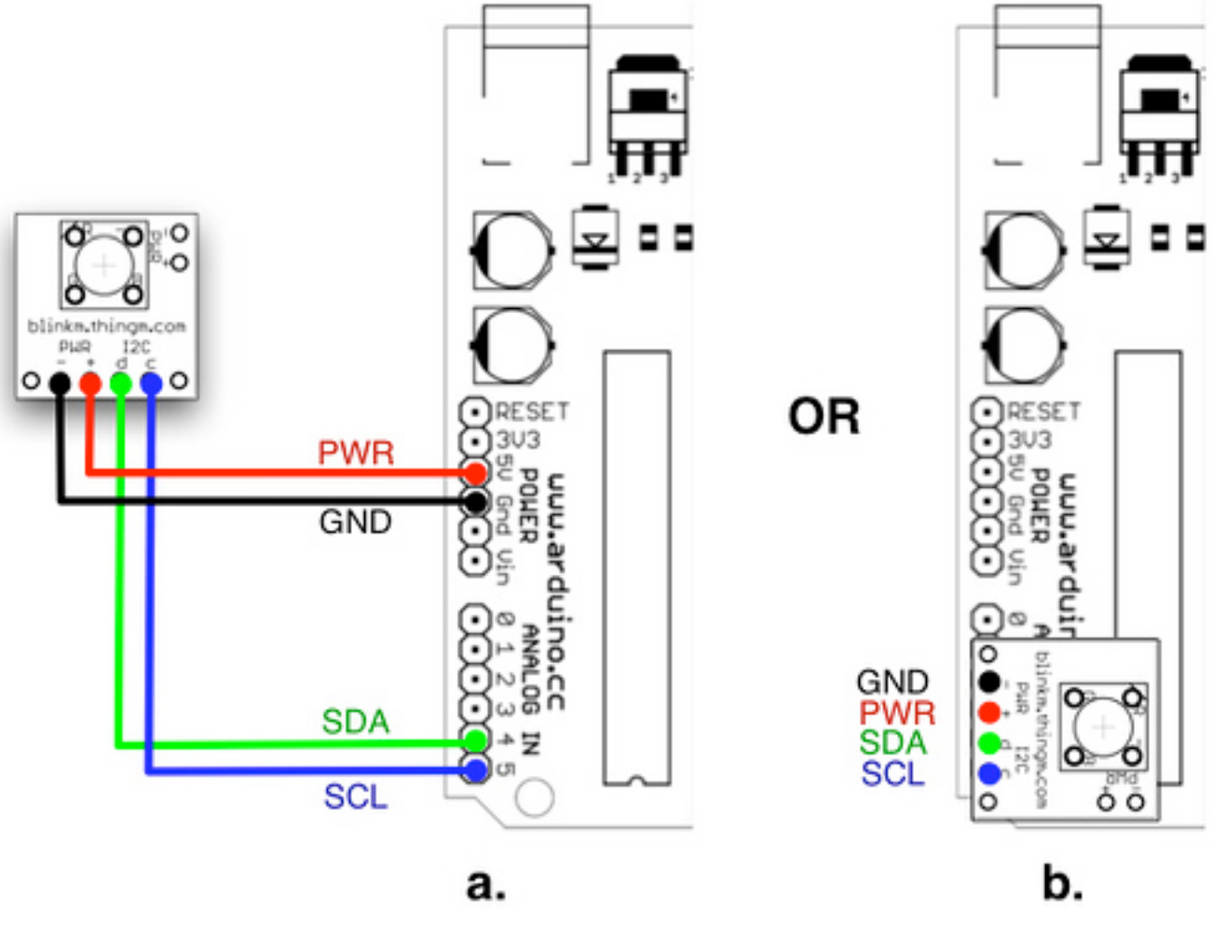


BlinkM started out as a desire to embed a microcontroller in a standard LED package. That's not that easy. So then it became exercises in making a small-as-possible board that's also prototyper-friendly.

BlinkM Hardware API Design Criteria

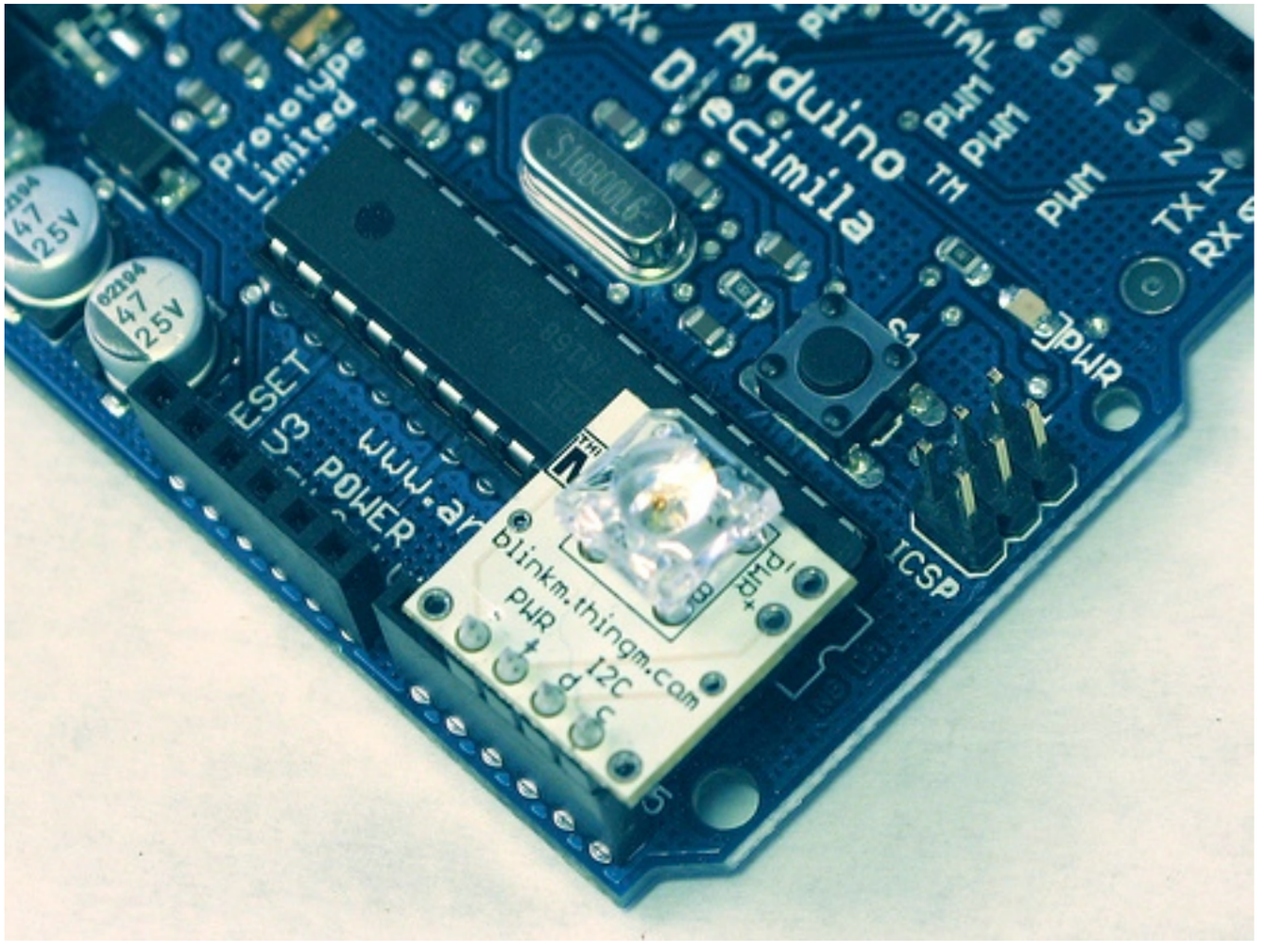
Design constraint	Result
As small as possible	~0.6” square
As bright & wide-angle LED as possible	8000 mcd RGB LED
Multiple BlinkMs on minimal µC pins	I2C, ATtiny45
High degree of non-trivial functionality	ATtiny45 (not tiny13)
Friendly with solderless breadboards	0.1” pin spacing
Easily usable with Arduino, zero-wiring(!) if possible	0.1” spacing & signal ordering
Moddable by intermediate users	SOIC package (not QFN)

BlinkM Wiring API for Arduino

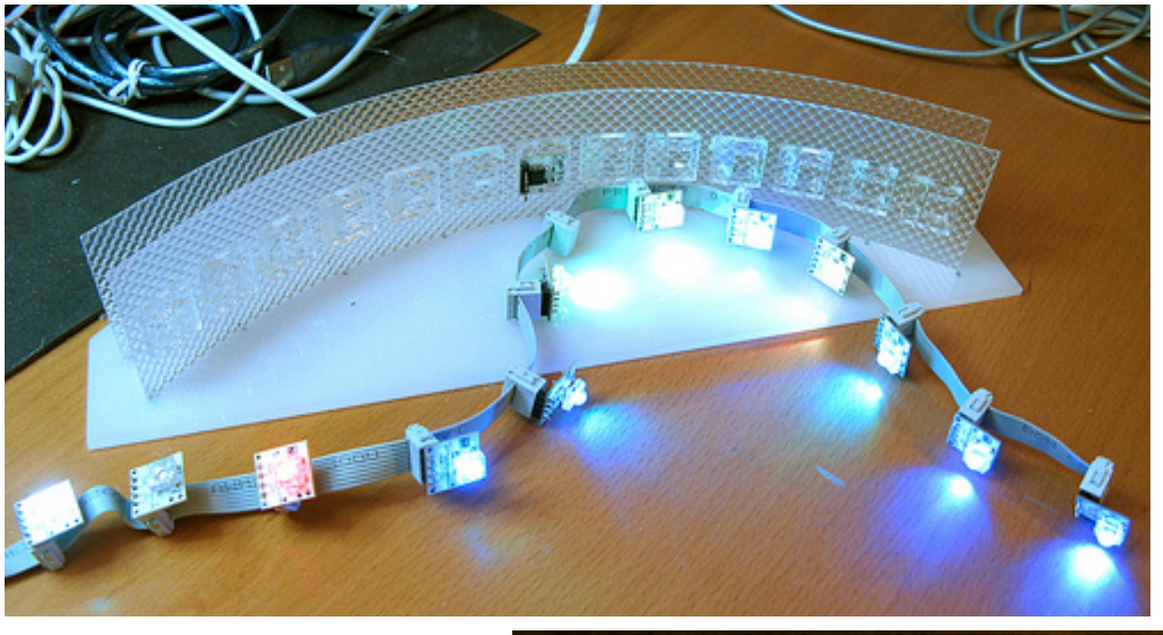


a. b.

I think this looks pretty cool



BlinkM can be thought of as sort of a mini shield for Arduino.
It seems to be pretty popular: we've sold about 4000 of them.

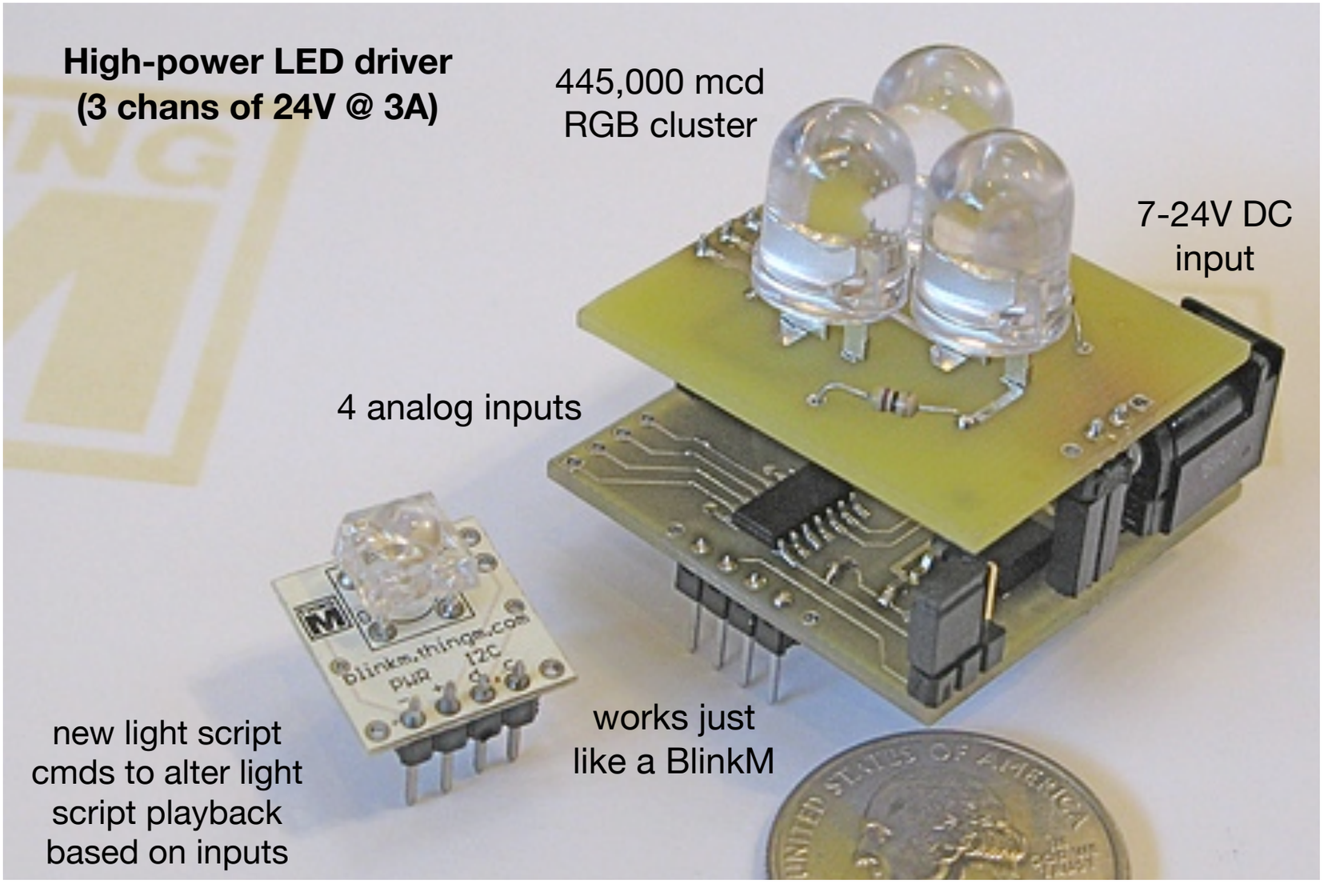


The BlinkM Cylon I did for Maker Faire.
http://todbot.com/blog/2008/06/17/get-on-the-blinkm-bus-with-a-blinkm-cylon/

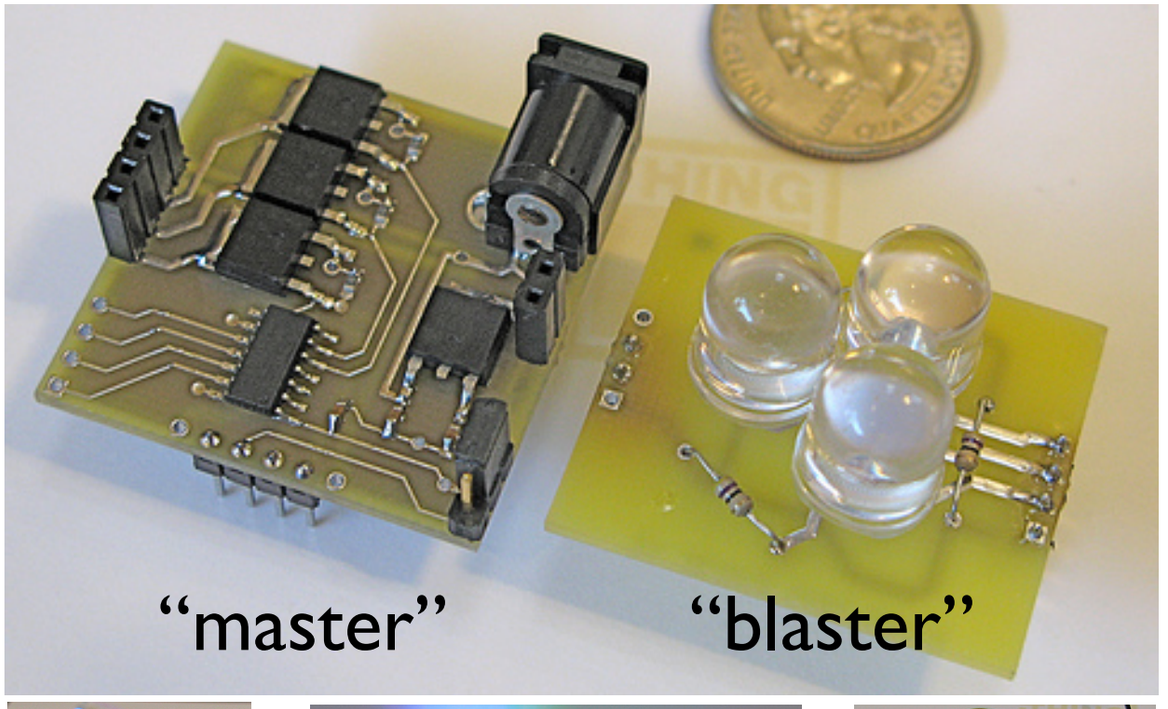
Video of the BlinkM Cylon in action.
http://todbot.com/blog/2008/06/17/get-on-the-blinkm-bus-with-a-blinkm-cylon/

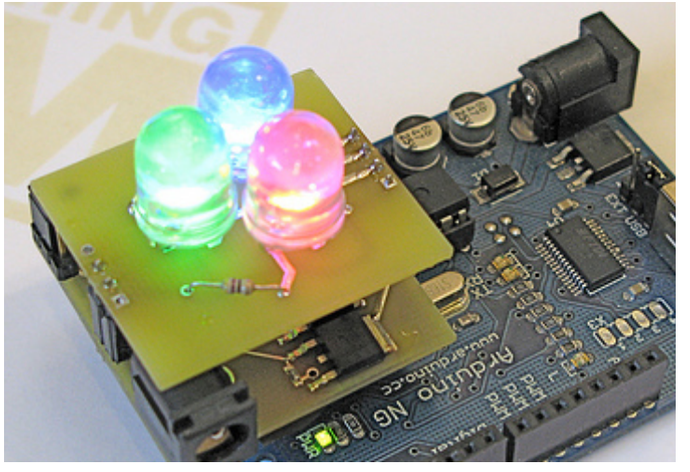
Dave Vondle’s BlinkM Wall

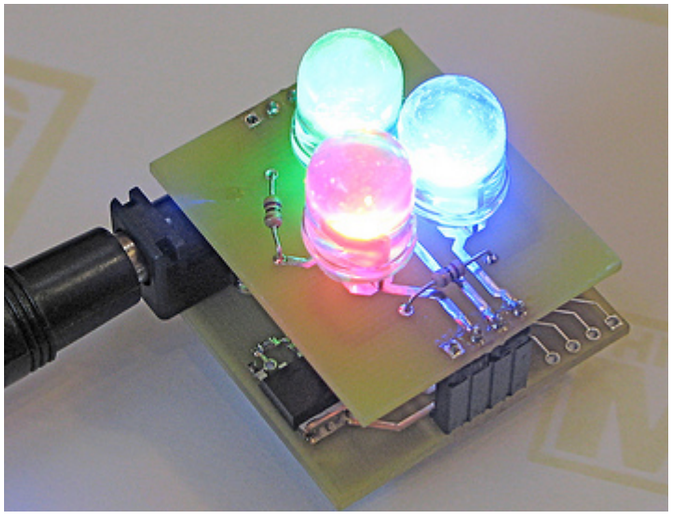
BlinkM MaxM

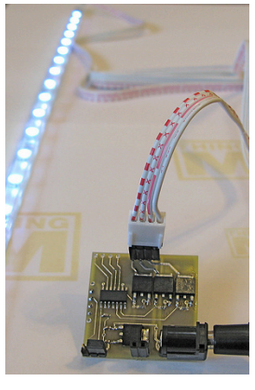

High-power LED driver
(3 chans of 24V @ 3A)
445,000 mcd
RGB cluster
7-24V DC
input
4 analog inputs
new light script
cmds to alter light
script playback
based on inputs
works just
like a BlinkM

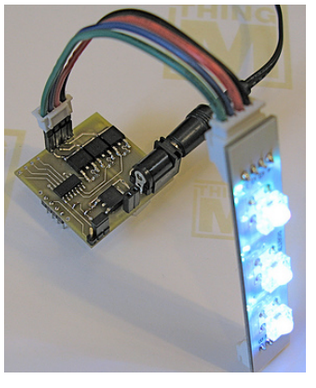
BlinkM MaxM











Examples of MaxM driving an Ikea DIODER and a Sparkfun RGB light bar.
Also a brightness comparison of MaxM to regular BlinkM.

USB not on Rails

- Okay, my original “USB on Rails” idea perhaps not so good
- HID spec is as nutty as USB spec
- Writing HID descriptors as ‘entertaining’ as writing USB device descriptors
- Needs structured data parsers on both computer and peripheral
- No access via Java and other interpreted languages
- Besides, HID is low bandwidth </rationalizations>

USB on !Rails

- USB Communication Device Class (CDC) an alternative
- CDC modems look like modem to OS (i.e. serial port)
- CDC Ethernet also possible (TCP/IP/Ethernet over USB)
- Both CDC modems & CDC Ethernet accessible from Java et al
- No drivers needed for CDC (just .INF file for Windows)
- Data exchange just TX/RX pipe like serial; simple but simple

CDC Ethernet might not be possible on Windows, might have to use RNDIS (which is also driverless, just INF file needed)

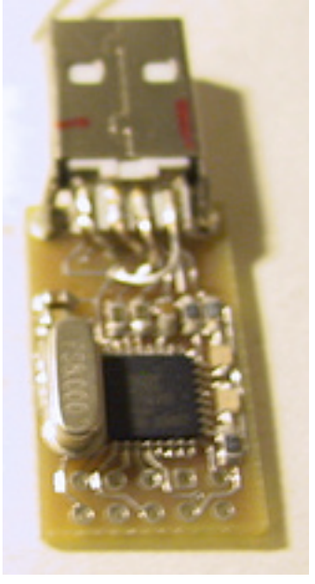
USB with AVR

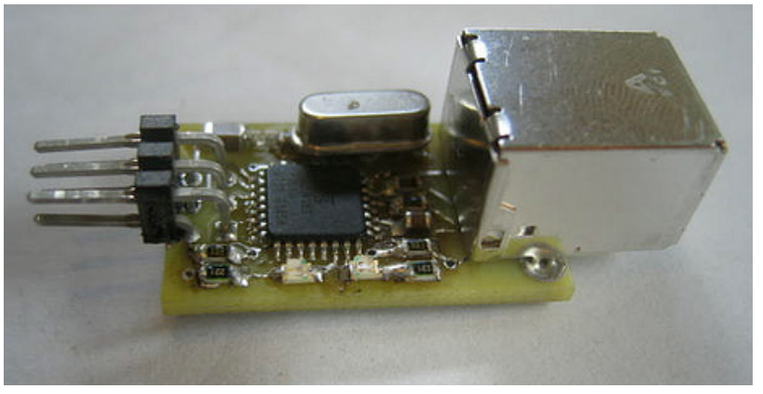
- Software USB stack (AVR-USB)
  - But USB CDC not valid on low-speed USB done by AVR-USB
- Hardware USB
  - AT90USB162 – 16kB, ~ATmega16, no I2C, $3.76/1
  - AT90USB1287 – 128kB, mature, QFN/MLF, $15/1
  - ATmega32U4 – basically ATmega168 w/USB, ~$6/1
  - All can be compound devices (CDC + Flash storage), some can be USB OTG hosts

AVR-USB: http://obdev.at/products/avrusb/index.html
ATmega16U4 not quite released yet.

USB with AVR

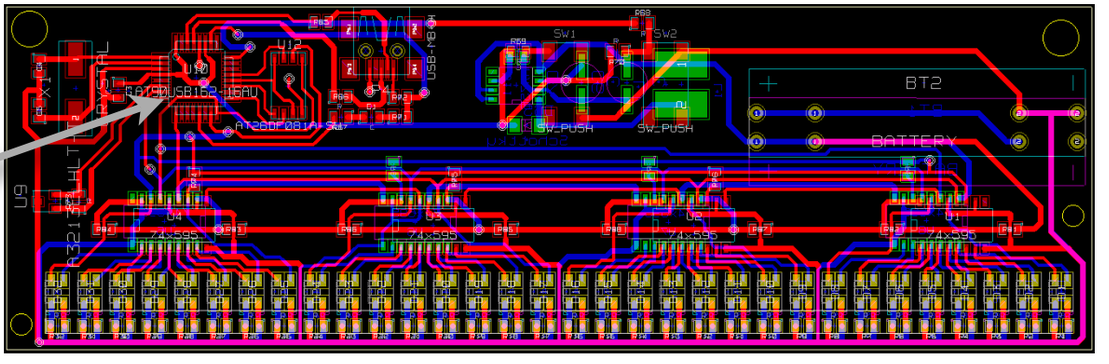
AT90USB162-based





Feurig's benito7 /
dorkboard pdx programmer

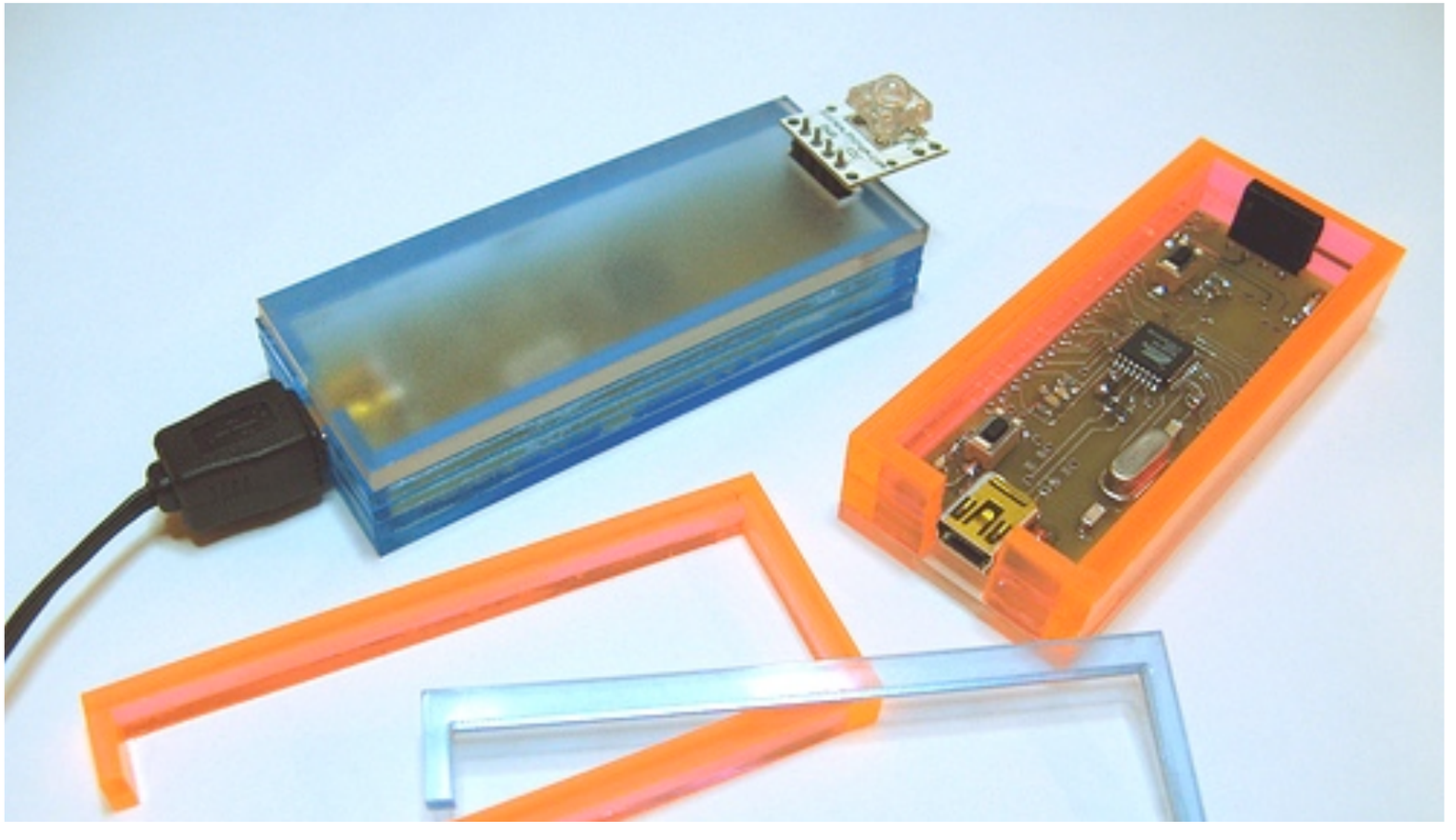
Bicycle LED POV



Feurig's benito7 on Dorkbot PDX : http://dorkbotpdx.org/blog/feurig
Bicycle POV : http://code.google.com/p/bicycleledpov/
MyUSB : http://www.fourwalledcubicle.com/MyUSB.php

BlinkM LinkM

An easier way to play with BlinkMs

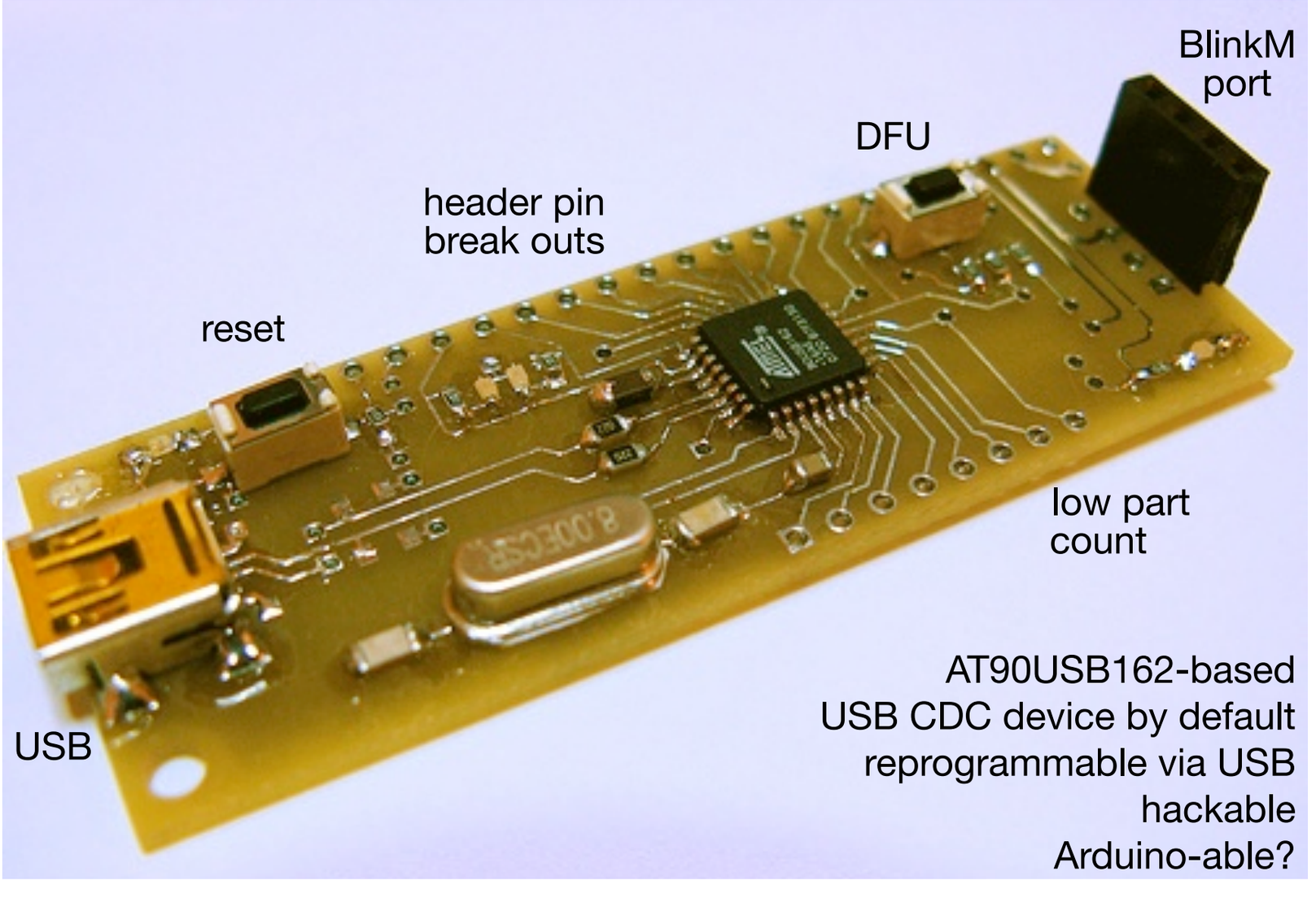


Plug-n-play, no driver install

Drives up to 8 BlinkMs directly from USB, more with external power

Stores complex multi-BlinkM sequences

BlinkM LinkM



AVR USB stack, but better:
http://www.fourwalledcubicle.com/MyUSB.php

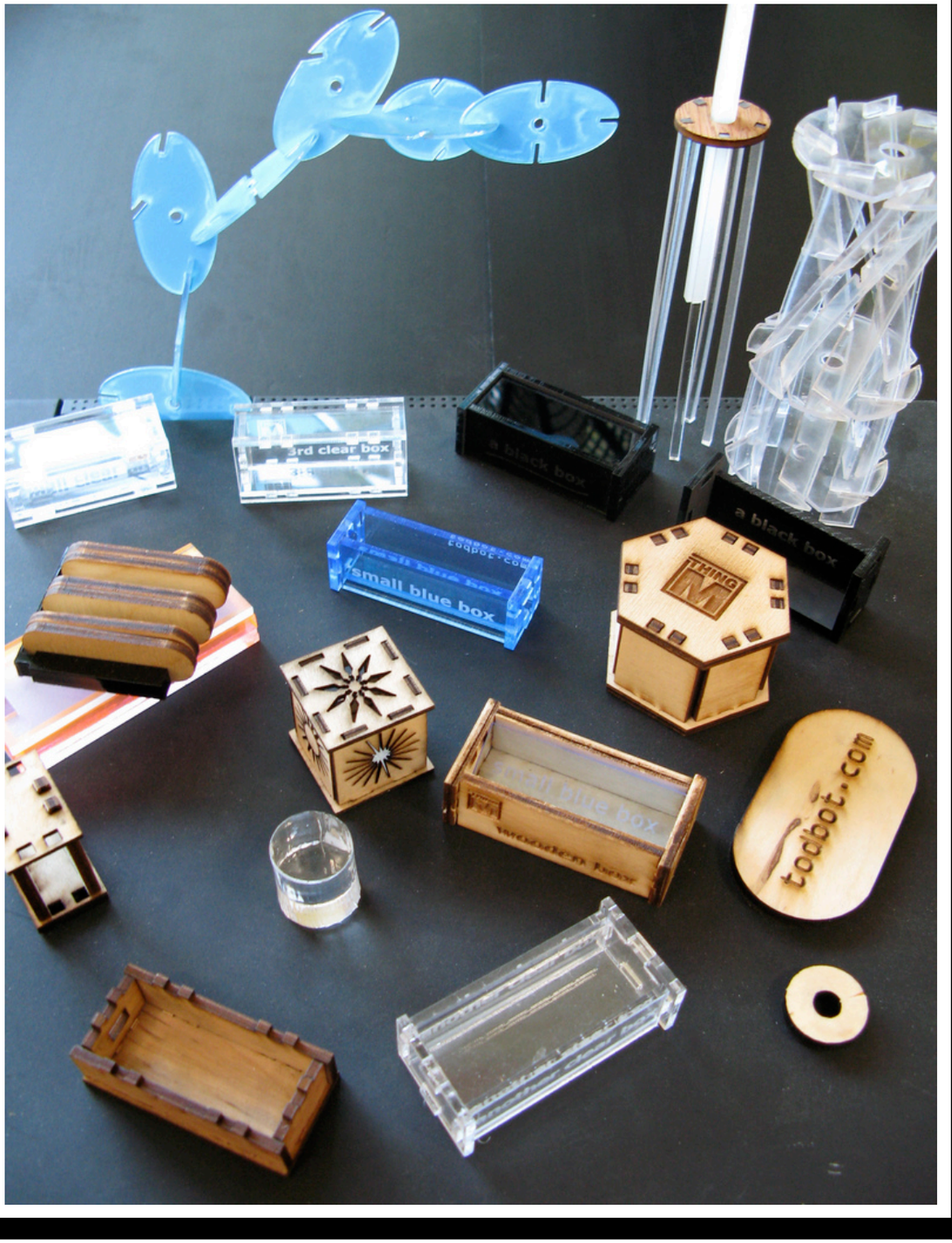
No analog inputs, however.

From 2D→3D

- Laser cutters cutting planar materials is easy
- But how to use it as a rapid prototyping / light production tool for arbitrary 3D shapes?
- How to turn 2D cuttings into 3D objects?
- Must every joint be 90º?

Trying to make a good enclosure for BlinkM LinkM led to many experiments with the laser cutter.

2D→3D: some experiments

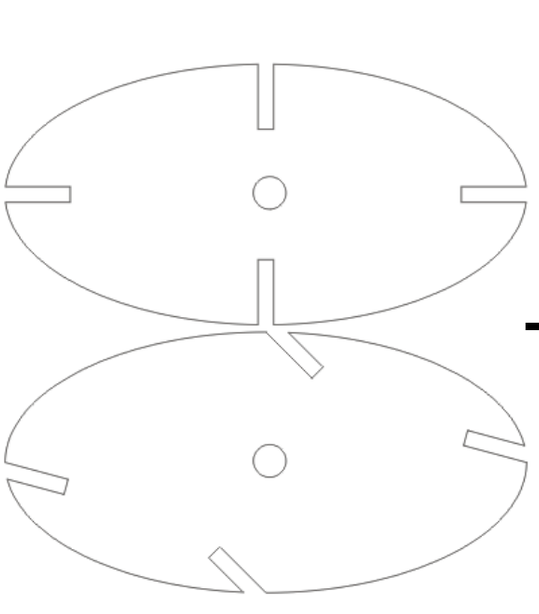


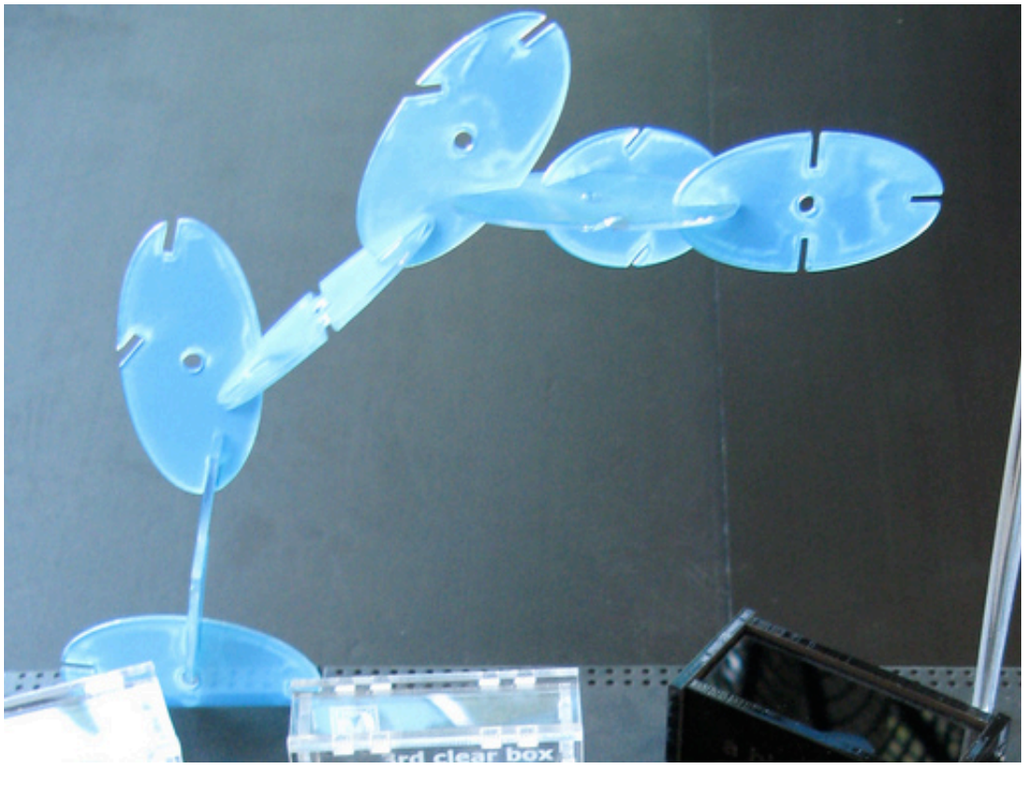
2D→3D: Textures

Not just for decoration. Adds tactility, scores for folding, insets for joinery

3D raster texture test by Ben Franco

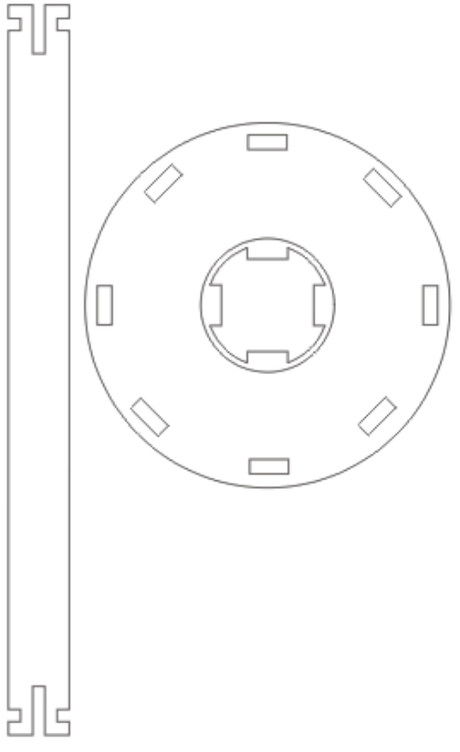
2D→3D: Generative

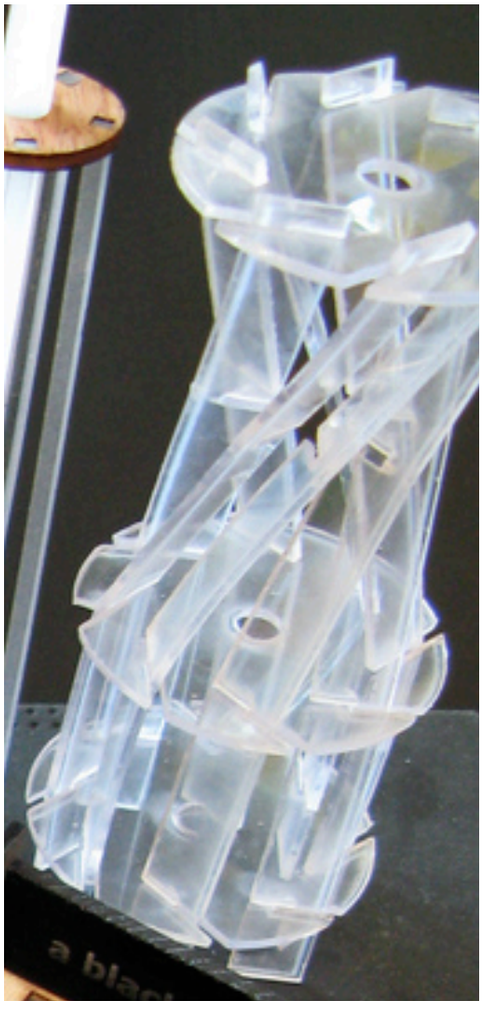




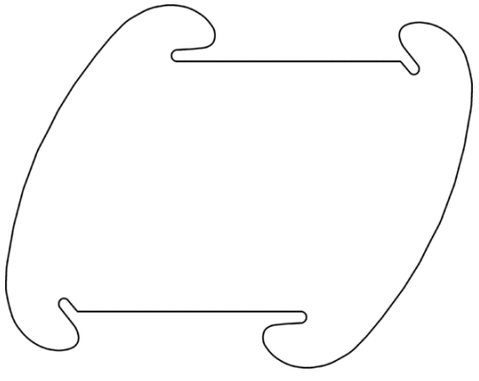
simple components iterated a few times create some complexity

2D→3D: Generative & Flex





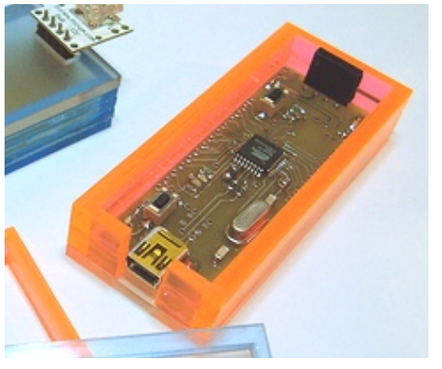
2D→3D: Generative & Flex

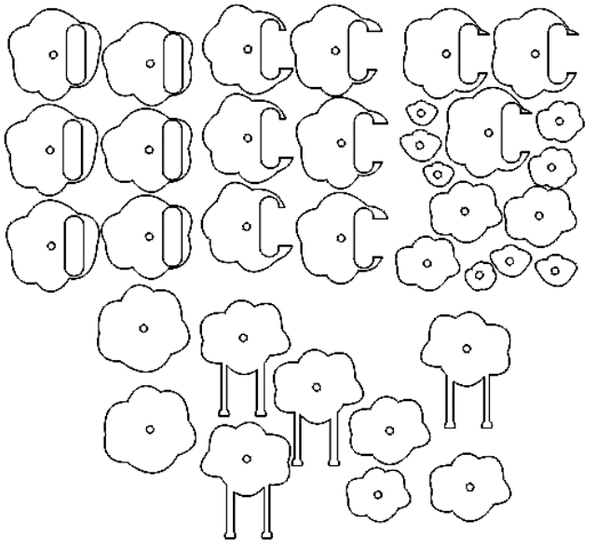


http://www.instructables.com/id/Universal-lamp-shade-polygon-building-kit/

This is not mine, but is really rad.
http://www.instructables.com/id/Universal-lamp-shade-polygon-building-kit/
Check out the variations in the comments.

2D→3D: Slice composition





Jed Burk's Hoggets Roam, http://www.transitionalspecies.com/hoggetsroam.html

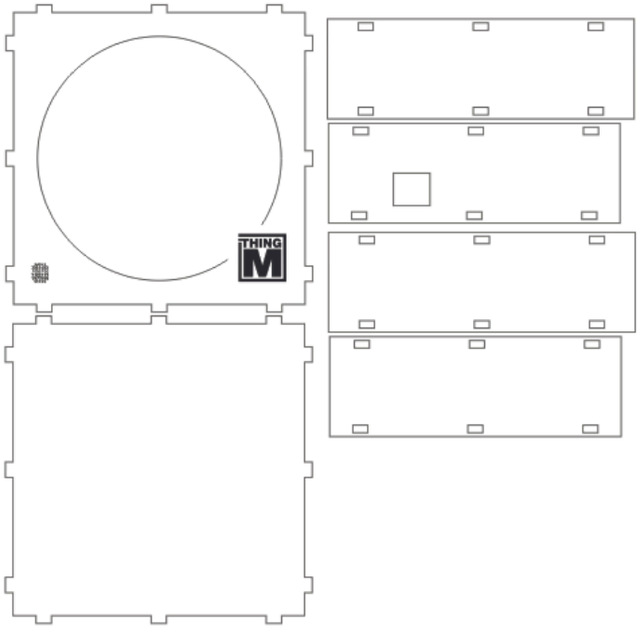
Also not mine, but I've been hanging out with Jed a bit and he's had some really good insights on techniques.

The problem with slice composition for making cases like the LinkM attempt above, is the laser doesn't cut perfectly vertical lines, but rather slightly slanted (the beam is a cone). Stacking slices reveals that, giving a sawtoooth texture, not very professional looking.

2D→3D: Bending

Not that relevant or useful, but it was fun to do.
That's a BlinkM lighting it up.

2D→3D: Boxes

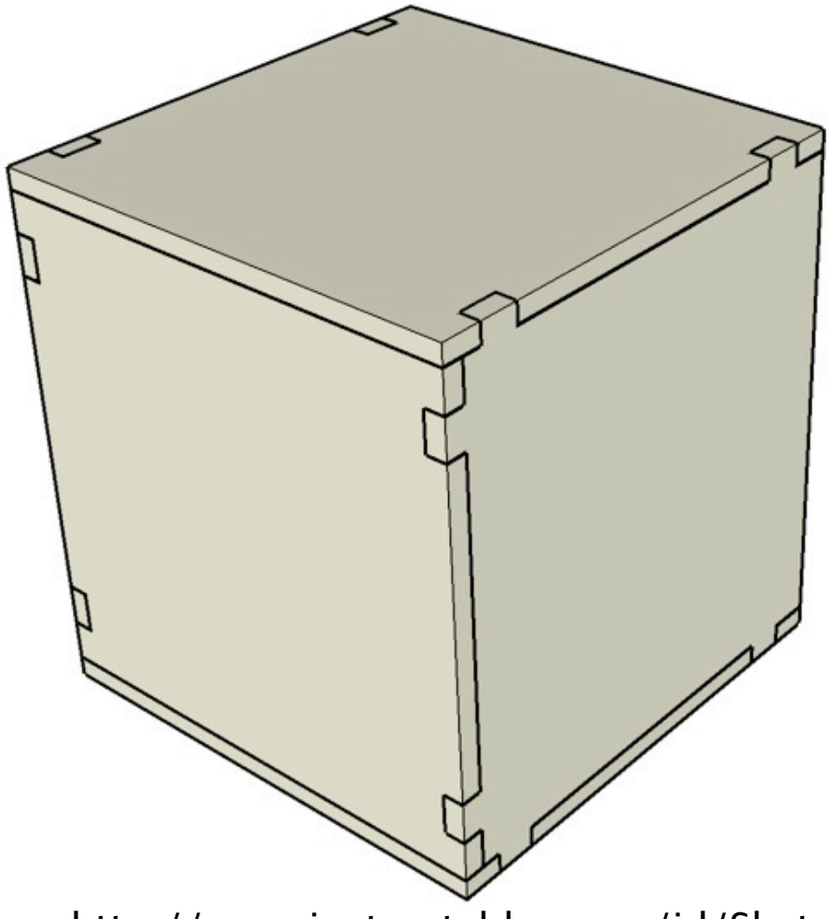


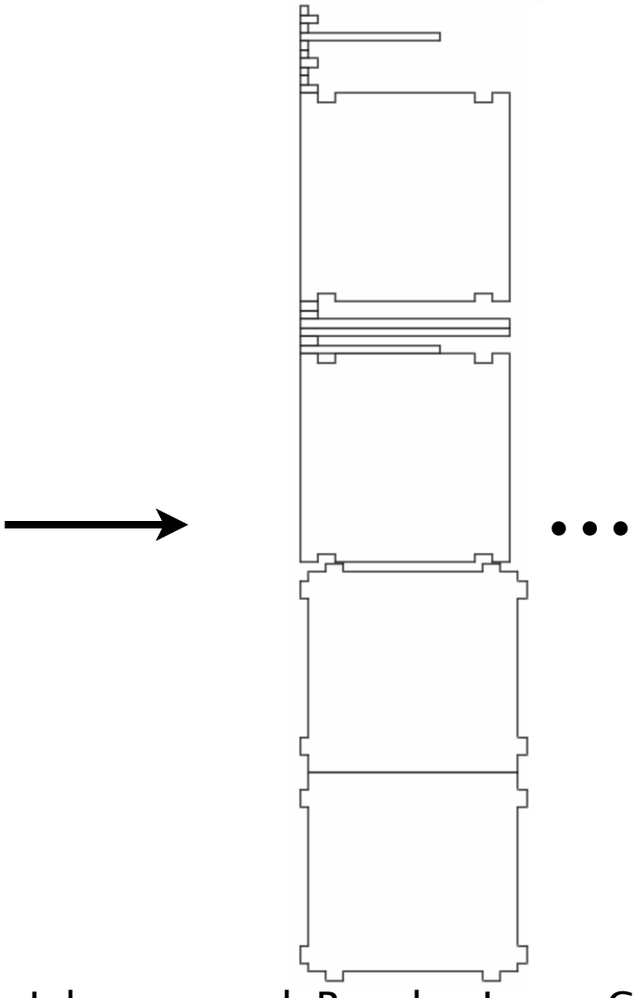
resorting to carpentry techniques

This is an RFID reader “coaster” idea for WineM.

No glue required for the box since friction-fit due to peg & hole dimensions.
Friction-fit not always possible depending on material (acrylic particularly is no good for friction-fit)

3D→2D→3D: Sketchup + SVG Plugin



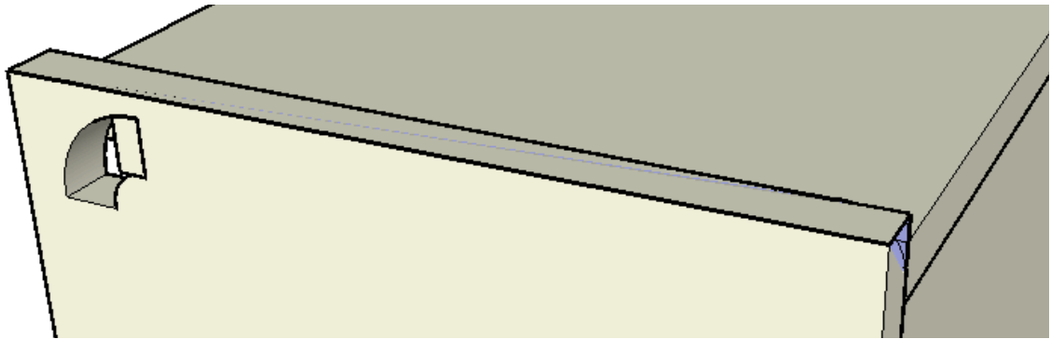


http://www.instructables.com/id/SketchUp-Inkscape-and-Ponoko-Laser-Cutting/

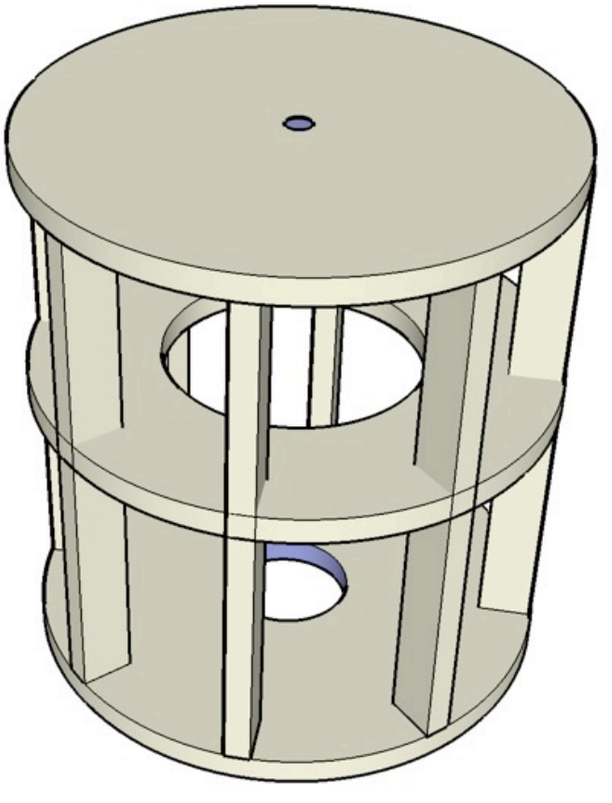
This is how I'll be doing most future laser-cut enclosures.
Construct them in SketchUp, then decompose them into cuttable planar pieces with the SVG plugin.
See:
http://www.instructables.com/id/SketchUp-Inkscape-and-Ponoko-Laser-Cutting/
http://flightsofideas.com/

More complex structures

hinge



non-rectangular



Tod E. Kurt
http://thingm.com/
http://todbot.com/blog/

sketching08
27 July 2008