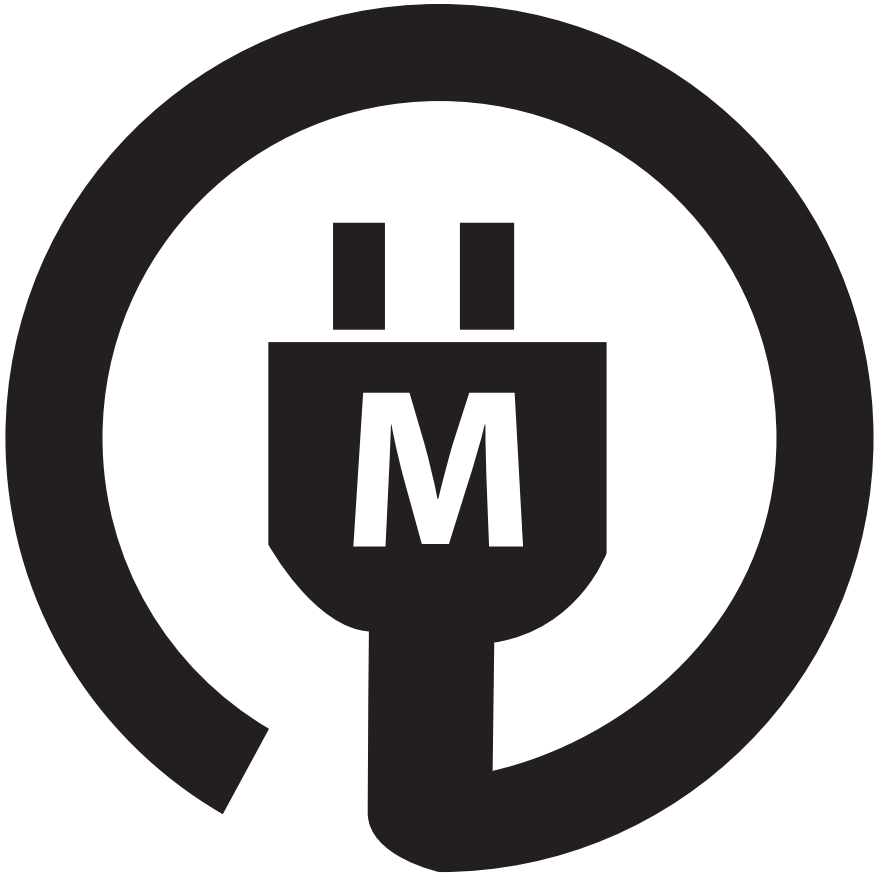


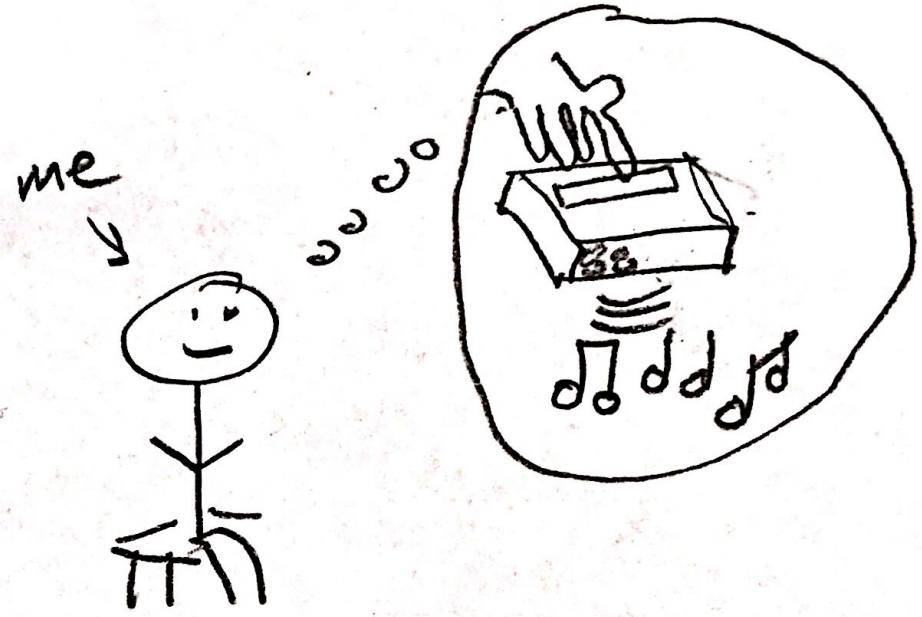
MakerNet

The fast, intuitive way to make awesome portable projects where you don't have to reinvent the wheel

Jeremy Gilbert
jgilbert20 at gmail
@makeitblink42

Of course, it starts with an idea…

Isabella – 3 Years Old



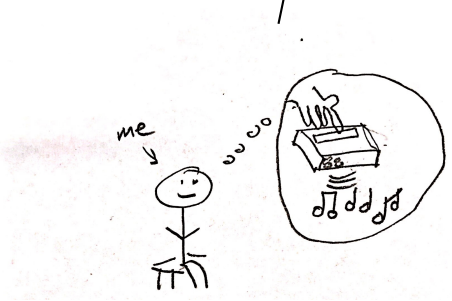
what if my daughter had a Keyboard she could carry?

Makers gonna make

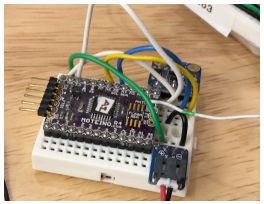
(But what do they make..?)

Things!

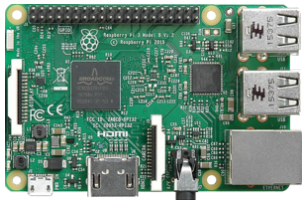
Thinginess scale:



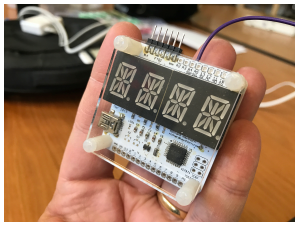
Ideas:
Not a Thing



Breadboards:
Not yet a Thing



RaspberryPi:
Awesome, but
not really a
Thing on its own



Hardened prototypes
Very Thing-like

Art:
Definitely a Thing

iPhone:
Definitely a Thing

So what exactly is a Thing?

Experiential

Finished

Shareable

Inspiring

Functional*

Self-contained

Permanent*

Things are important to Makers

They inspire us to make other Things

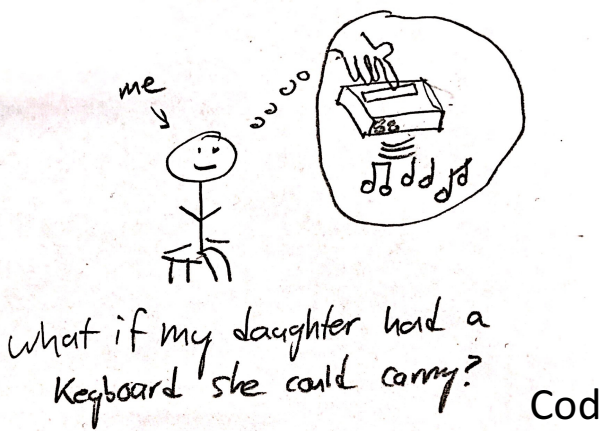
And they teach us what is possible

They also let us share the things we love

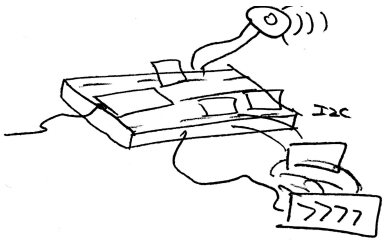
* Usually; there are always fun exceptions of course

Making Things is really hard

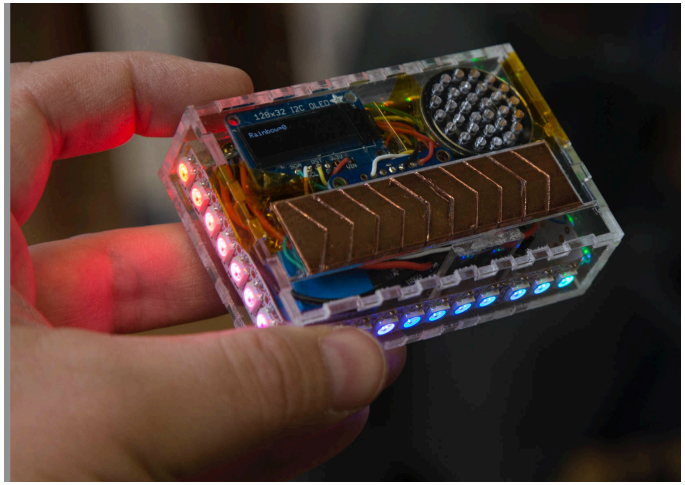
Idea



Proof of concept



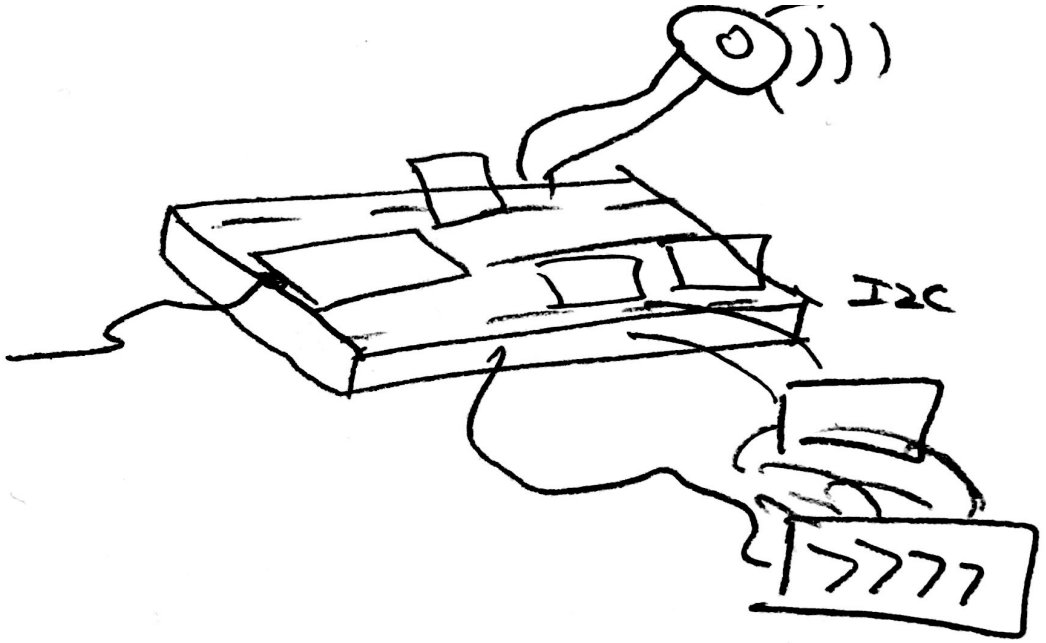
Finished "Thing"

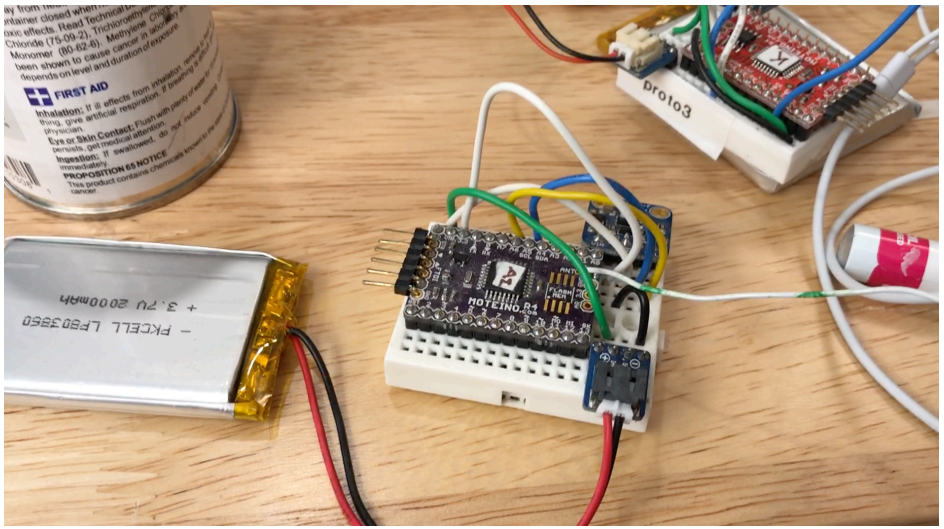


Coding
Wiring
Basic analog
Basic digital
Microprocessor code
Wire-striping
Multimeter

Soldering
Case-making
Resilience
Battery management
Stability
Mounting
3D design
Laser cutting, FDM, etc

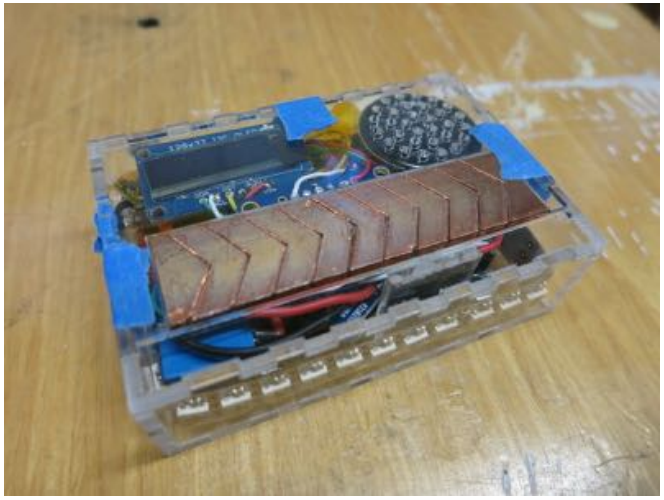
For me, breadboards are the stopping point of many great project ideas



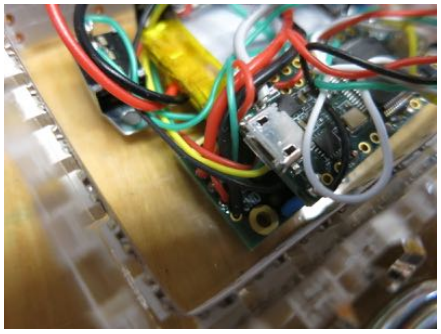


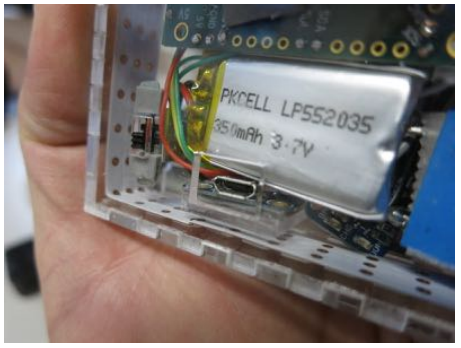
- Great for proving out an idea relatively quickly
- Completely un-portable
  - Cannot show to a 3-year-old
  - Can't bring into work easily to show co-workers
- Most of your effort spent with glue logic (level shifting, I2C address debugging, event loop programming), not on the really fun stuff
- Not permanent
- Not shareable

The transition to “finished project” often requires all sorts of special skills

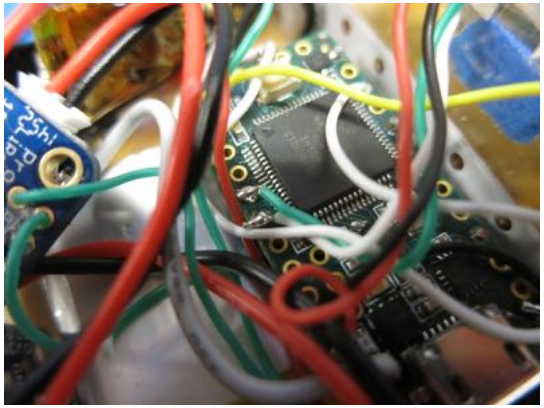


Casemaking

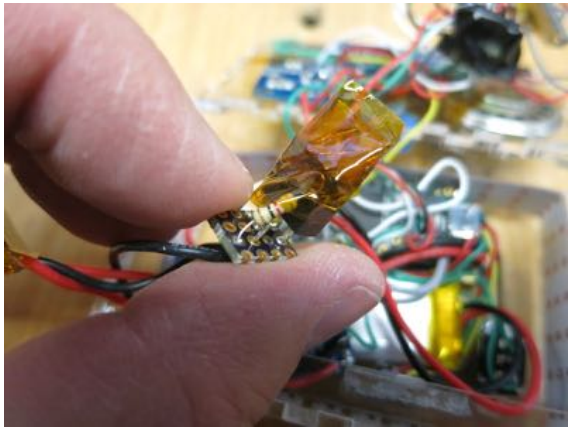




Routing USB to edge of case

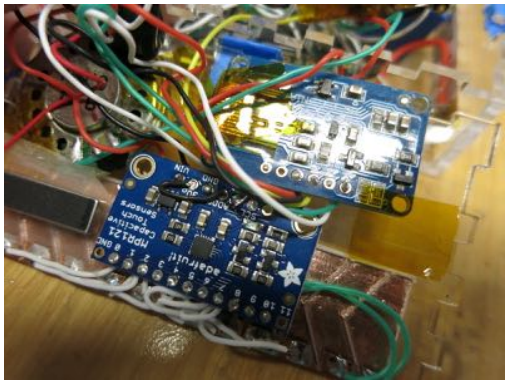


Fine pitched soldering



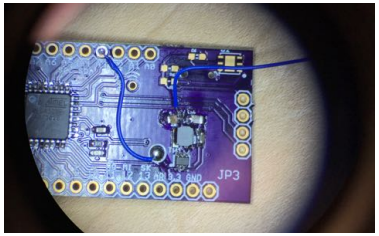
Hacking small pieces together

Bus wiring

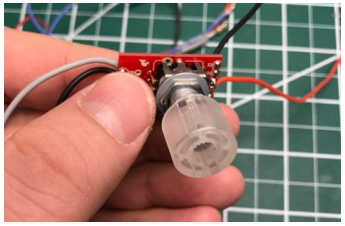


Not to mention some incredibly esoteric, hard-won skills such as low-power hacking and SMT soldering

Getting below 0.00mA



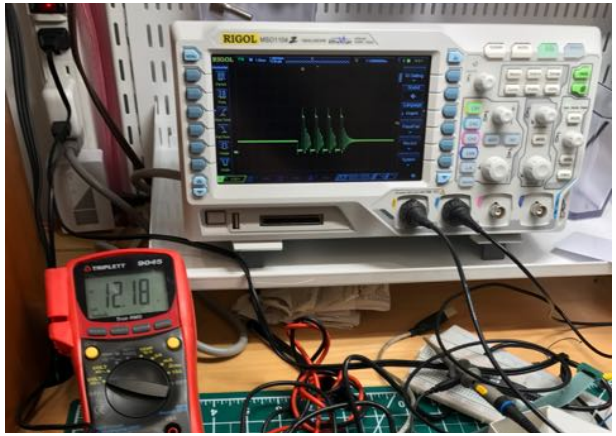
SMT Soldering



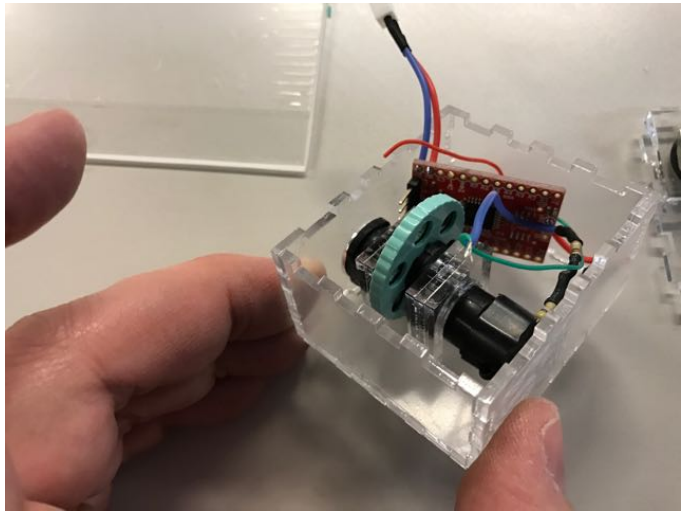
Interrupt handling

SMT Reflow

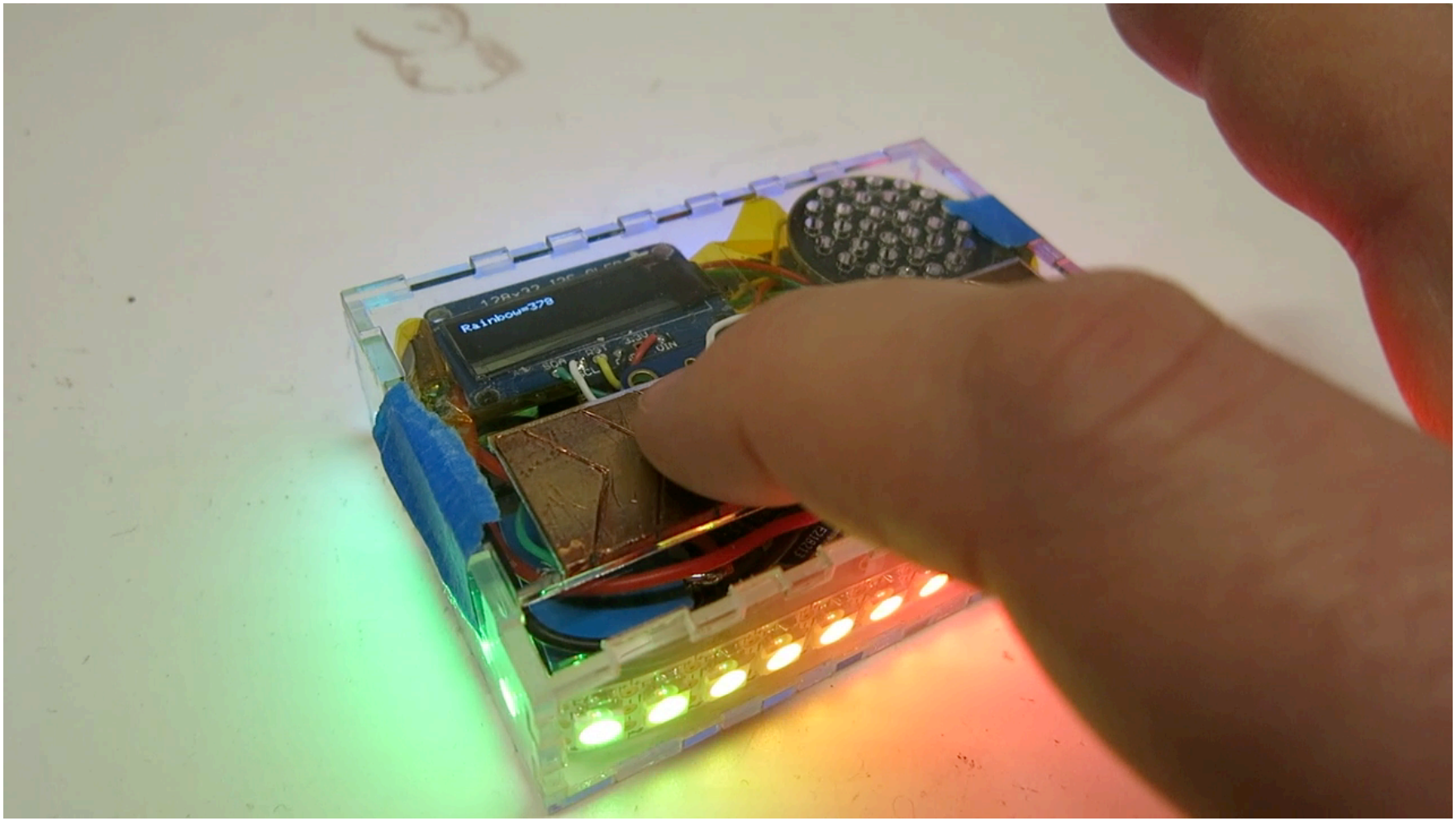
Mixed signal debugging



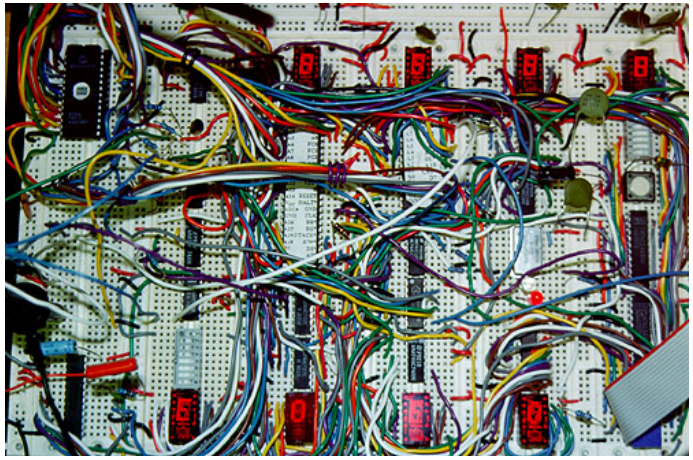
Acrylic welding

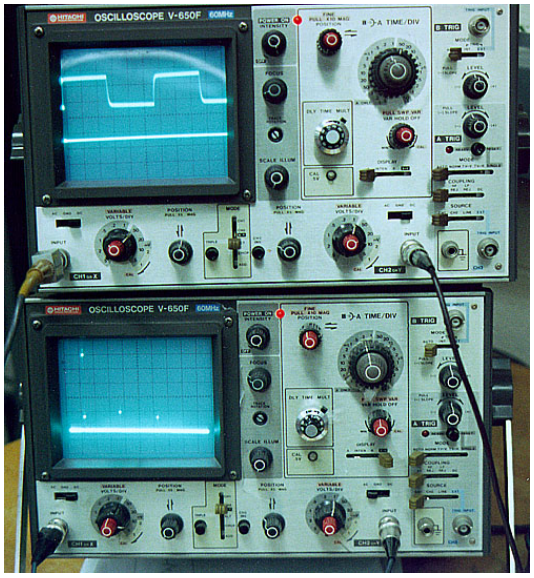


A recent result: Box of Clicky Awesomeness



Making anything used to be a whole lot harder…





FFT implemented on a 68000 DIP

- In 1996, when I trained in computers at Brandeis, it took a whole semester to assemble a working CPU
- PIC chips were new and few of my professors had ever heard of them
- Building computers seemed like such a complex activity…
- I naturally turned to coding and left EE behind

Limor Fried, SFE, Arduino all have made it possible for electronics projects to open up to wider audiences

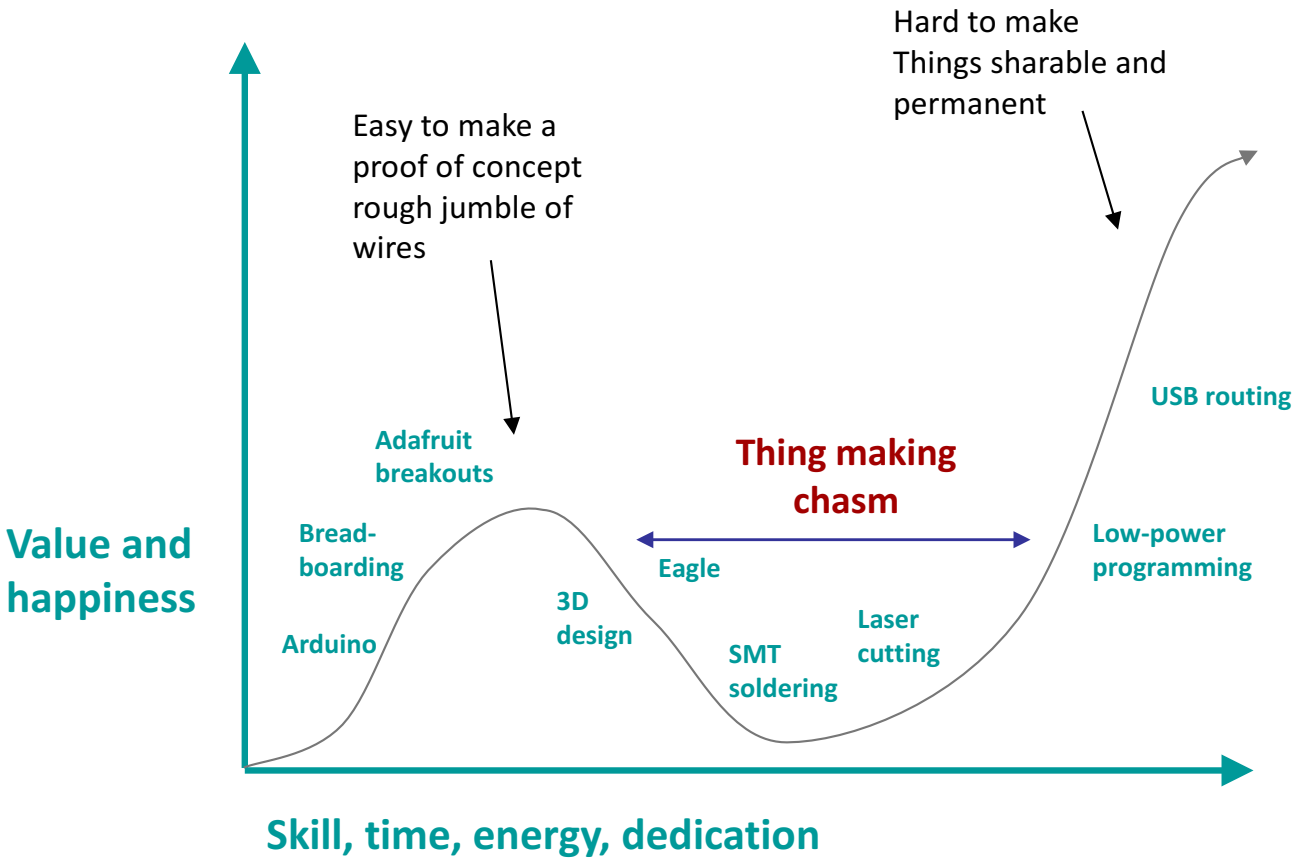
sparkfun ELECTRONICS

adafruit

- We've seen some awesome, open source tools that inspire and teach
- These tools make it easy and low-cost to get your feet wet (no data sheets, no assembly, etc)
- Having **early wins** as you explore is vital to accelerating and growing our community

- If Limor Fried, Paul Stoffregen, and the Arduino team hadn't come along, **I would never be a maker**

But there is still a pretty steep learning curve between prototype and finished “Thing”

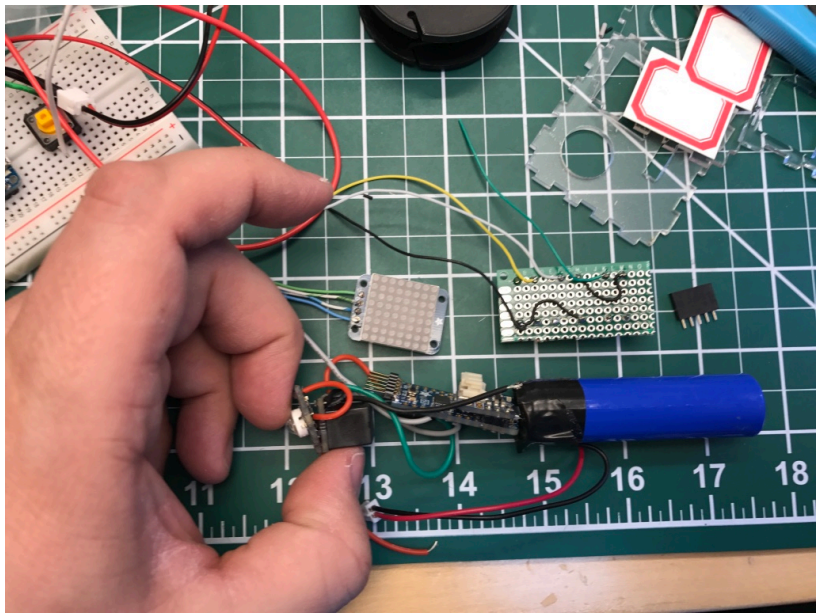


- To get more Makers in the world, we need to reduce the burden that it takes to go from hacked-up prototype to “sharable” project
- Not a replacement for learning, but a way to get more early wins
- Today we push Makers unnecessarily into the “weeds”

Making is a way of thinking… so how do we think faster?

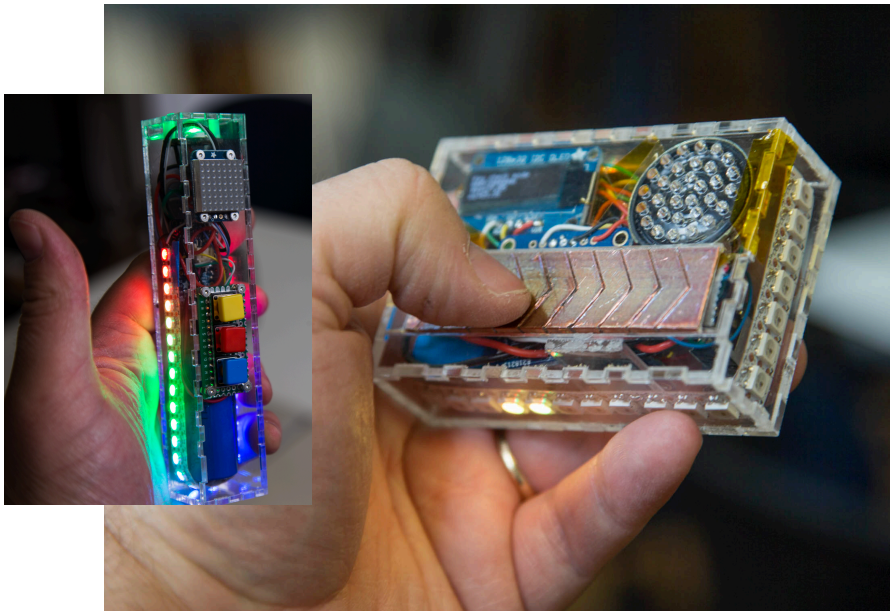
- ”the act of making and refining prototypes will change your product's design and teach you what works and what doesn't work”

Bread-boarding and loose wire



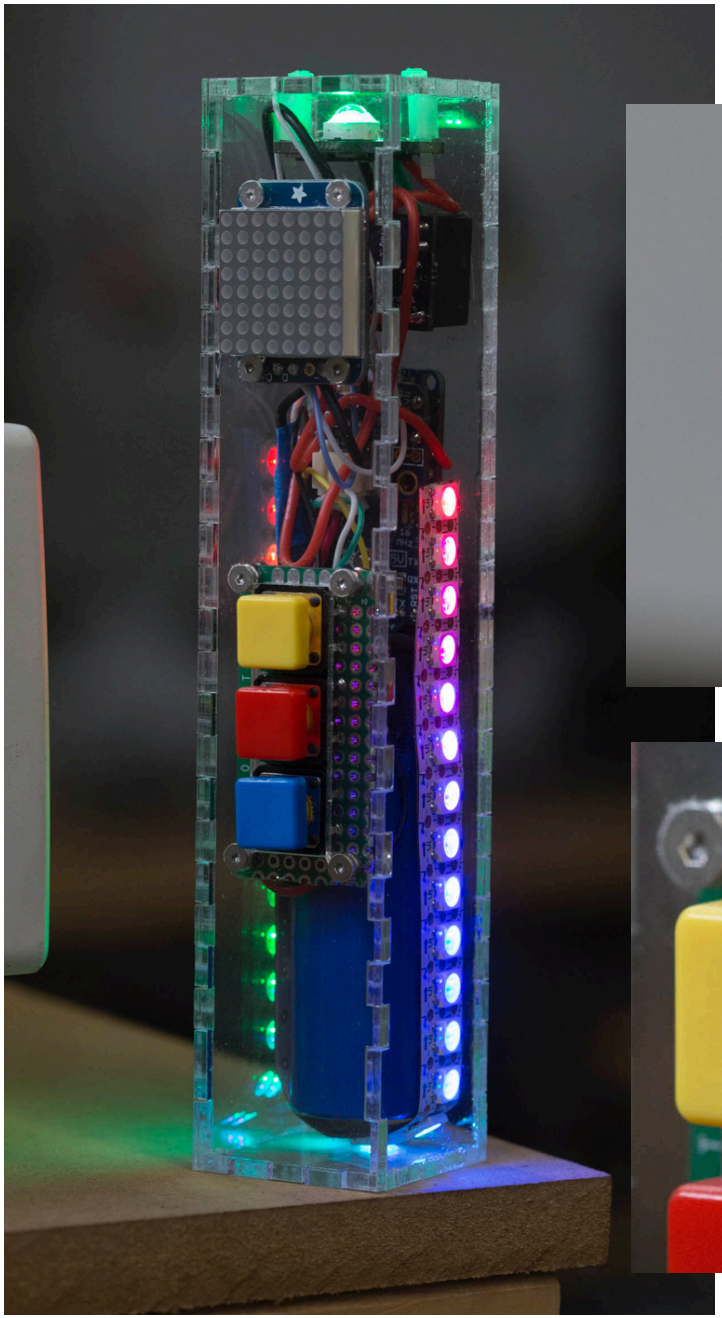
- Teaches you if the components will wire up together, if the software works without crashing, etc

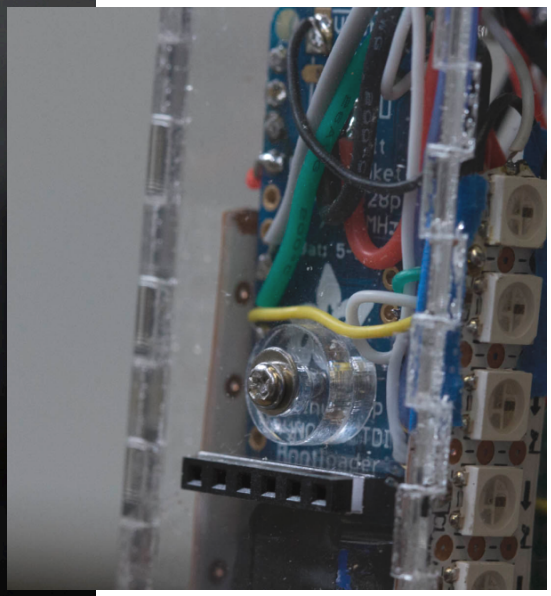
Self-contained builds

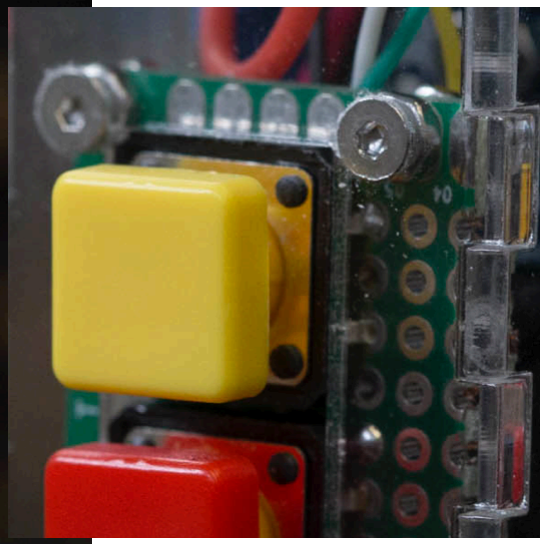


- Teaches you if your concept delights and engages yourself and others
- Let’s you understand utility, functionality and design

Challenge 1: Making projects self contained







- Casemaking requires skill, 2D/3D design expertise, precision tools, laser cutting, 3D printing, etc
- Battery power and field-programming are desirable, but requires charging circuits and easy ways to route USB/FTDI to case edge
- Low power modes for portable Things are essential but a particular challenge

Challenge 2: Integrating lots of peripherals

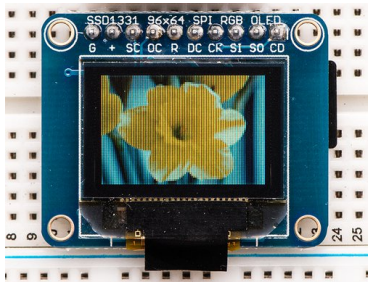
Parallel

SPI Serial

OneWire I2C

Multiple standards

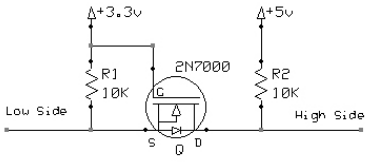
- Arduino libraries are a good start but can introduce conflicts and complexity



CPU-bound graphics

- Discovery, addressing, arbitration usually not handled by libraries, requiring hard coding

Level shifting



- Color graphic displays eat substantial central MCU time and add lag

Interrupt management

```
attachInterrupt()
```

- Level shifting introduces more components

Wiring

- Wiring messes impede case-fitting and assemby

The Maker community needs higher level abstractions and modularity for Things

Peripheral

Adapter

Adapter

"Makernet"

Handles power, level shifting, interfacing, arbitration, naming, and low-power support

Multi-drop, multi-master network

Adapter

RotaryKnob

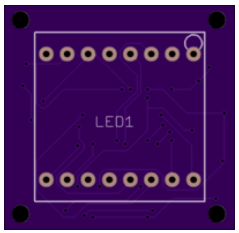
8x8 Matrix

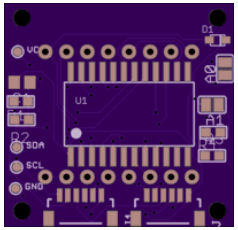
MCU

Clean OO software interfaces

Each device type would have identified SKUs for mounting, 2D models for cases, and software abstraction

Common PCB





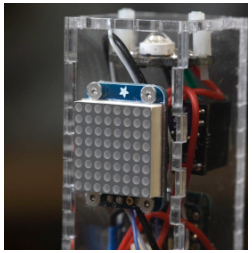
Software library

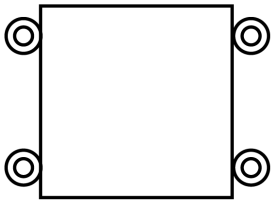
Std8x8Matrix()

Mounting kit

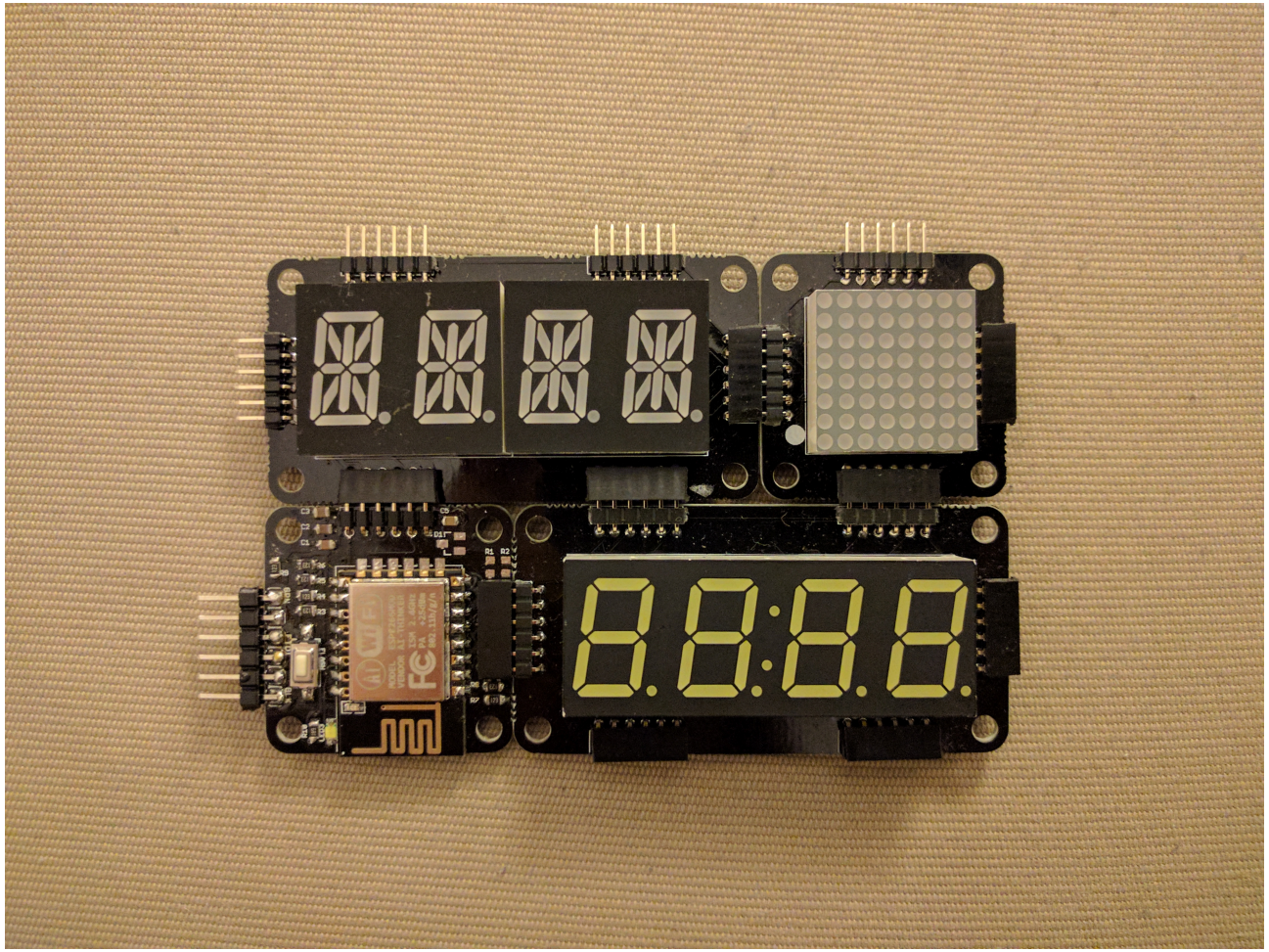
You get the hardware and the thing needed to put it into a fun case; common drill-points

Case plans





Richard Hawthorne, found of CHS, has been experimenting with modules based on I2C



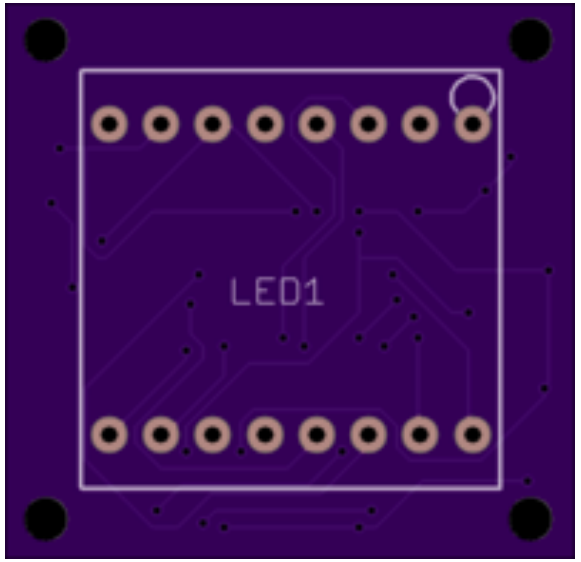
- Common sizes for the modules
- Interconnects are 1” pitch

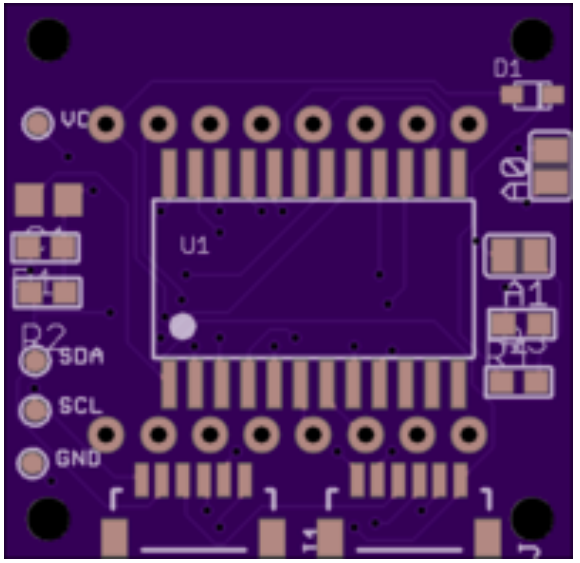
See http://deluxecapacitor.com/projects

MakerNet would tie each component together

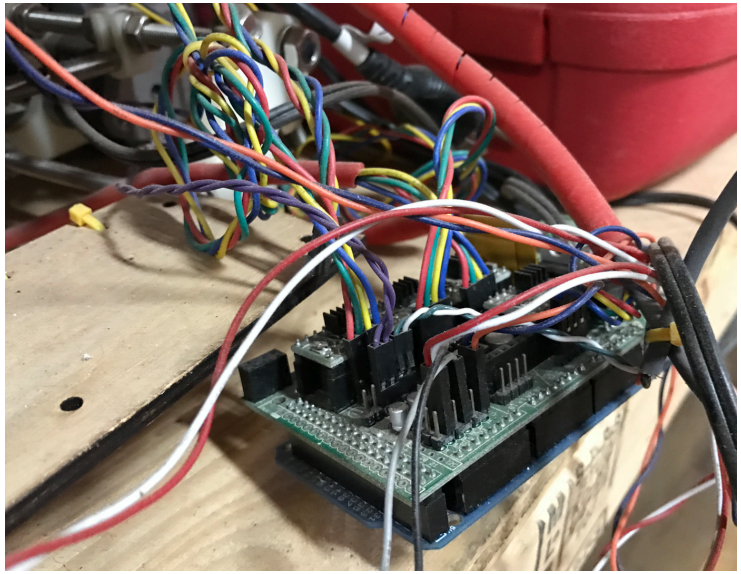
- All are connected by the MakerNet bus – handles address conflicts, power requirements, low-power, latency and events in a modular framework
  - I2C w/ discovery + arbitration
  - Ethernet-like bus connects all devices
  - Think CANBus but cheap, open and fast

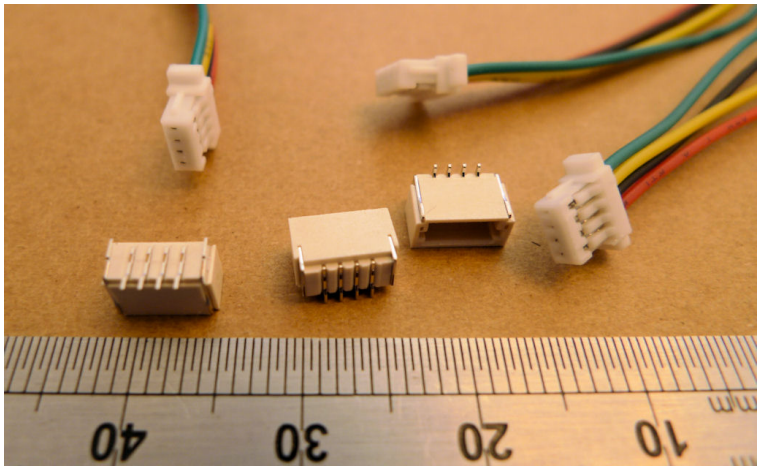
- No soldering required for simple things – everything connects with a 6-pin JST-SH





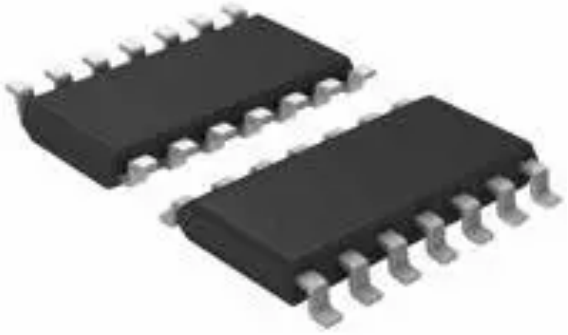
We’re thinking of JST-SH as a cabling standard





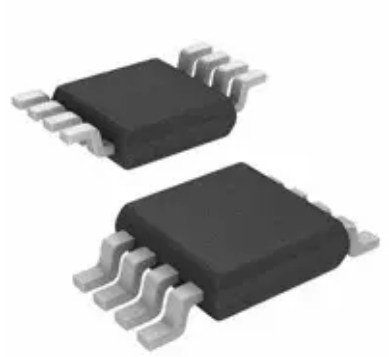
- 1mm pitch
- Easy to source 6 pin cables (Due to use in RC/GPS)

The core makernet framework could be built on on RS485 and adds <$2.00USD to each component BOM



AT SAM D09: 48MHRZ ARM Cortex-M0, two external components, $0.88 ea

Price Break	Unit Price	Extended Price
1	1.90000	1.90
10	1.71300	17.13
25	1.52960	38.24
100	1.37660	137.66
250	1.22364	305.91
500	1.07068	535.34
1,000	0.88714	887.14



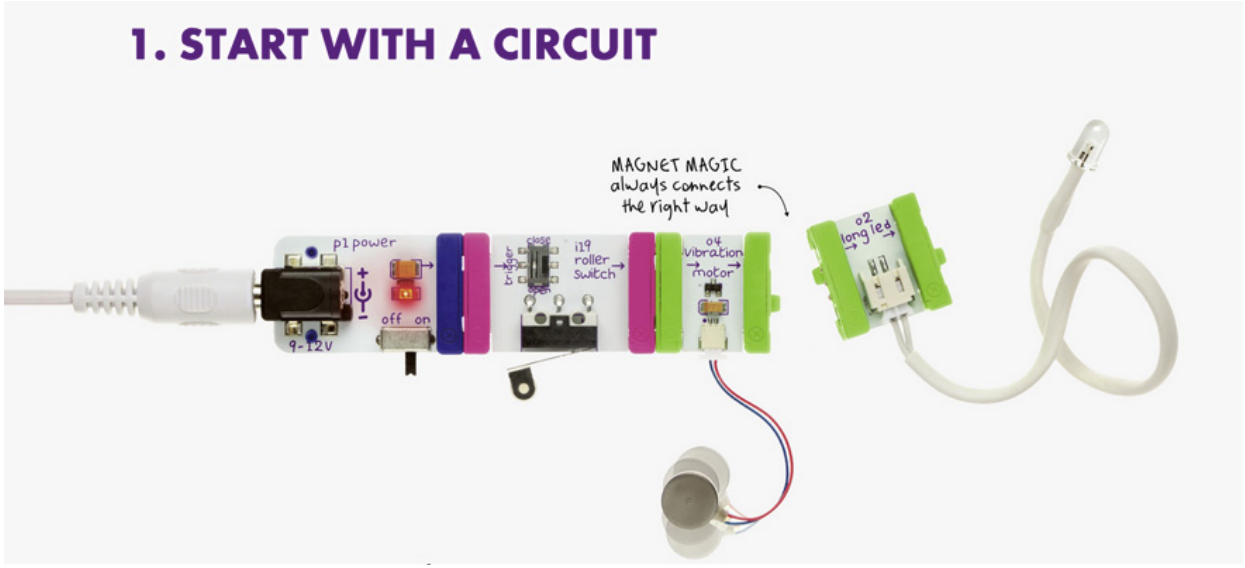
ISL83078EIUZA – 3.3V RS485 transciever, half-duplex, 10MBPS

Price Break	Unit Price	Extended Price
1	1.74000	1.74
10	1.56400	15.64
100	1.25670	125.67
500	1.03250	516.25
1,000	0.85550	855.50

Submit a request for quotation on quantities greater than those displayed.

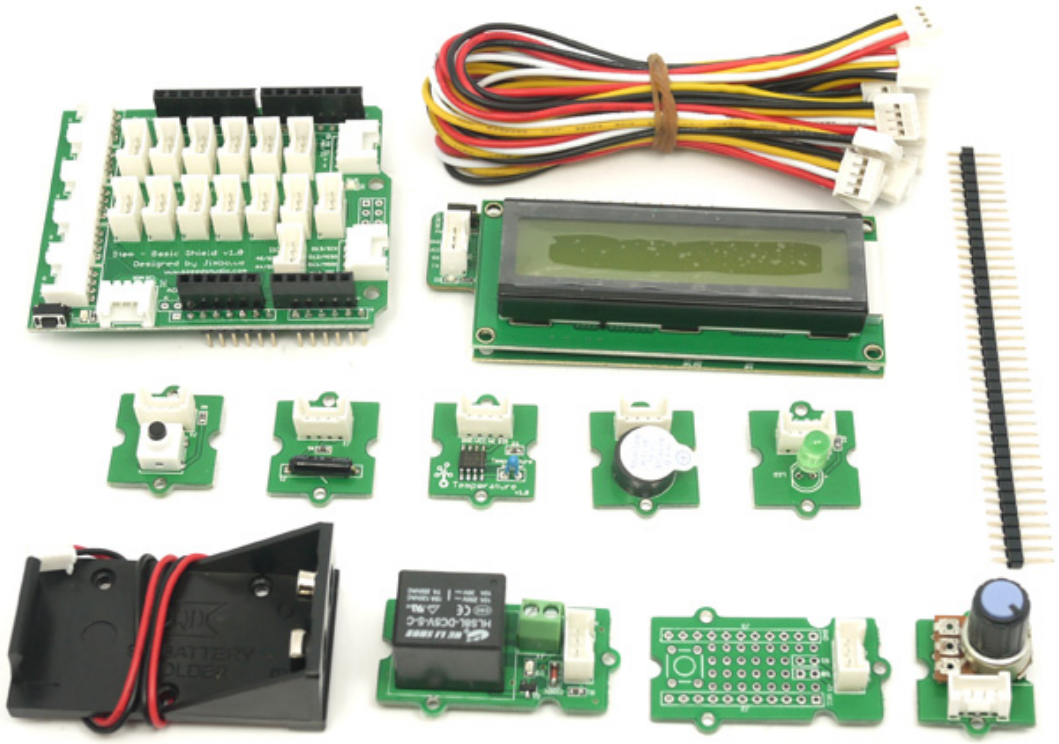
- MakerNet packets are translated to I2C, SPI or GPIO as needed
- MakerNet handles bus arbitration, address assignment
- Data encoded on a differential pair highly resistant to noise
- Low-power mode and device shutdown built in

Ideas/inspiration/thoughts: LittleBits



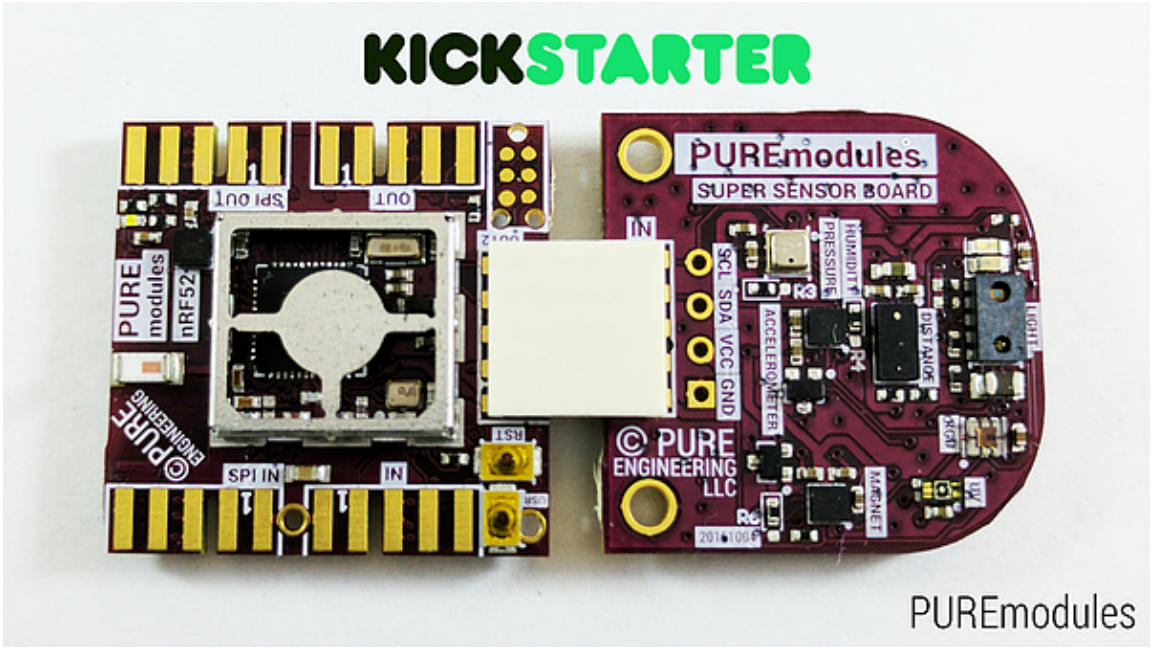
- Great for sharing joy of electronics
- No soldering required!
- Magnet links don't hold very well -- very hard to make anything "permanent" or Thing-like
- Target: Great for kids and early electronics learners

Ideas/inspiration/thoughts: Grove



- Good:
  - Simplifies cabling
  - Documented
  - Mounting holes
- Challenges
  - No universal bus or software toolset
  - Difficult to panel mount cleanly
- Target: Learning electronics and prototyping ideas quickly

Ideas/inspiration/thoughts: PureModules



- Good:
  - Common bus architecture
  - Nice and small, highly functional parts
- Questions:
  - Does the bus support arbitration, discovery, etc?
  - Does it help address the case and fitting problem?

Ideas/inspiration/thoughts: Modulo

- Super high-quality documentation + code
- SAM D21 based architecture
- I2C with some great tricks (looks like it could support arbitration, discovery, etc?)
- How could this be adapted to cases and footprints?
- Does the I2C architecture serve what we need in terms of discovery, renumbering, plug and play?