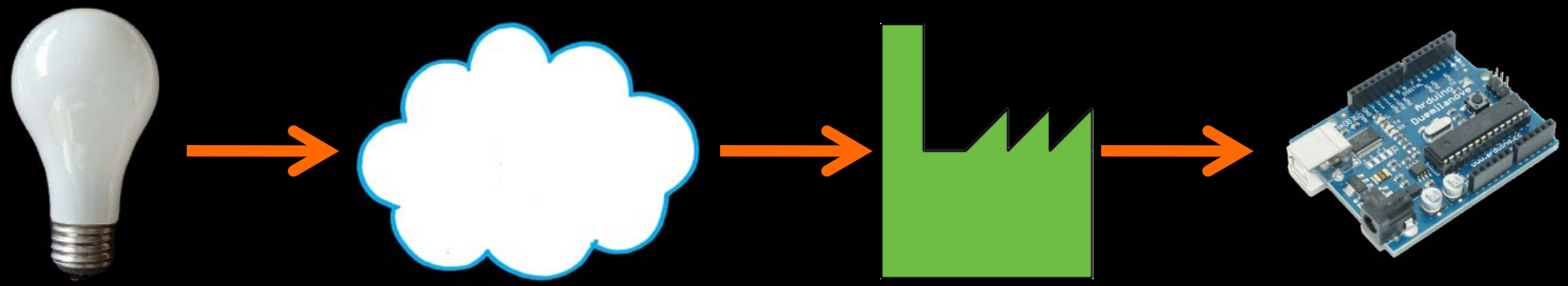
Open-Sourcing The Engineering (Design) Process

Open Hardware Summit 2011

amanda wozniak,
Staff Electrical Engineer, Wyss Institute
Amanda.Wozniak@Wyss.Harvard.Edu

First off, what is the engineering process?



great idea for a project

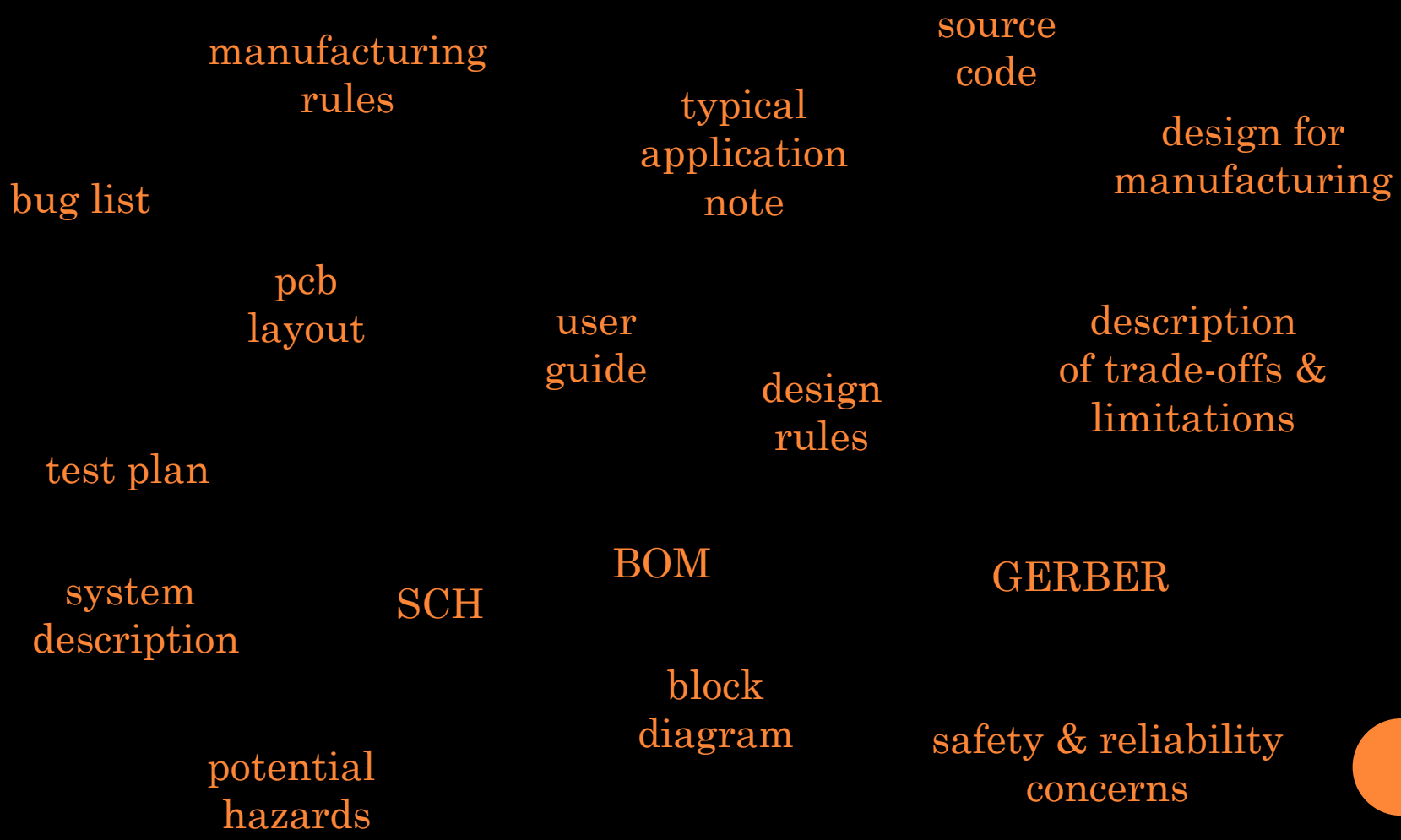
a miracle occurs!

fabrication and assembly

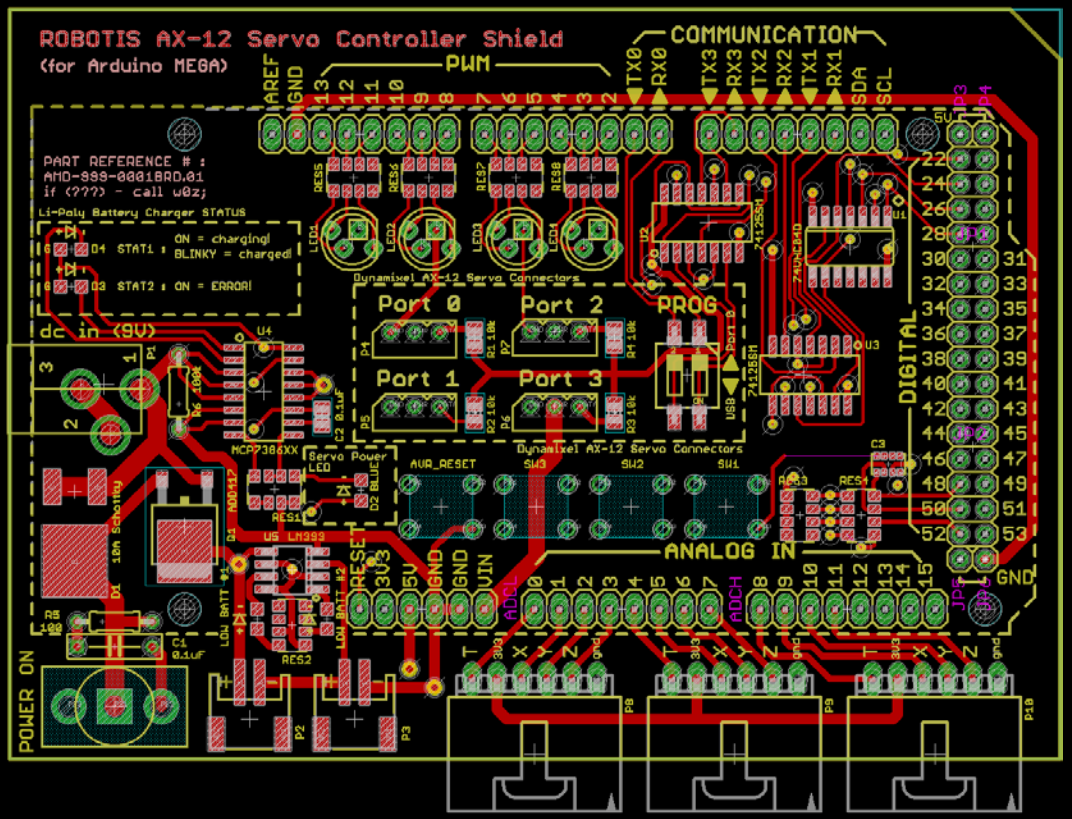
hardware

It's all of this.

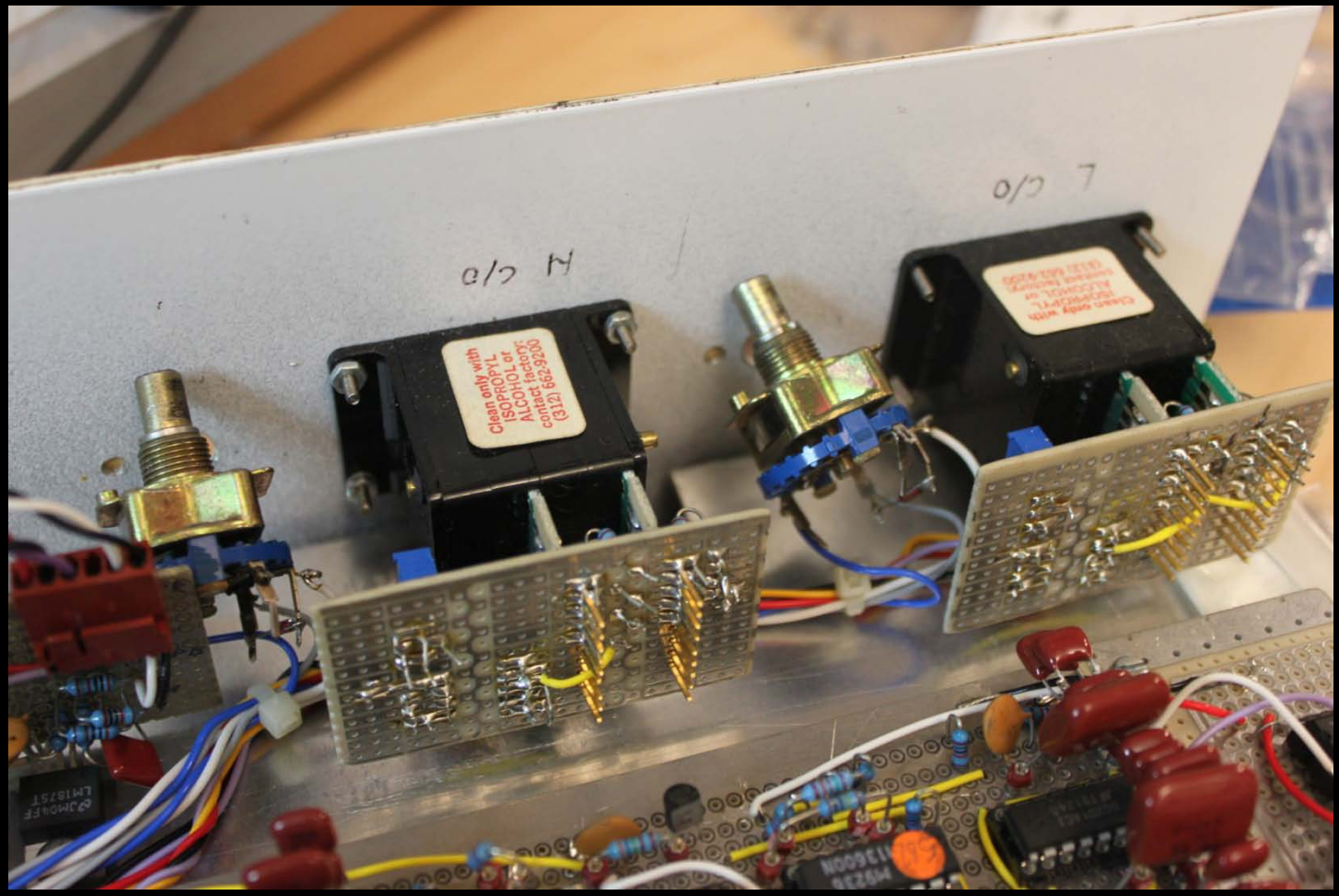
The IDEA of a documenting a process is DEAD SIMPLE. Actually documenting a SPECIFIC process is incredibly HARD.



SCHEMATICs, GERBERs and BOMs are a sufficient for manufacturing. why bother with a process? Shouldn't what we MAKE be enough?



What's so OPEN about your hardware if I have to REVERSE-ENGINEER it to contribute?



Industry is so successful, because underneath all the NDAs, everything is exposed.

In addition to being open, FORMAL design processes often SAVE YOUR BACON

Sure, society could collapse from insufficient documentation.

Don't let that intimidate you.

THIS IS THE STORIED ENGINEERING PROCESS

have an idea

ask important questions

design (record your rationale)

hold a review

make one (and test it)

does it work?

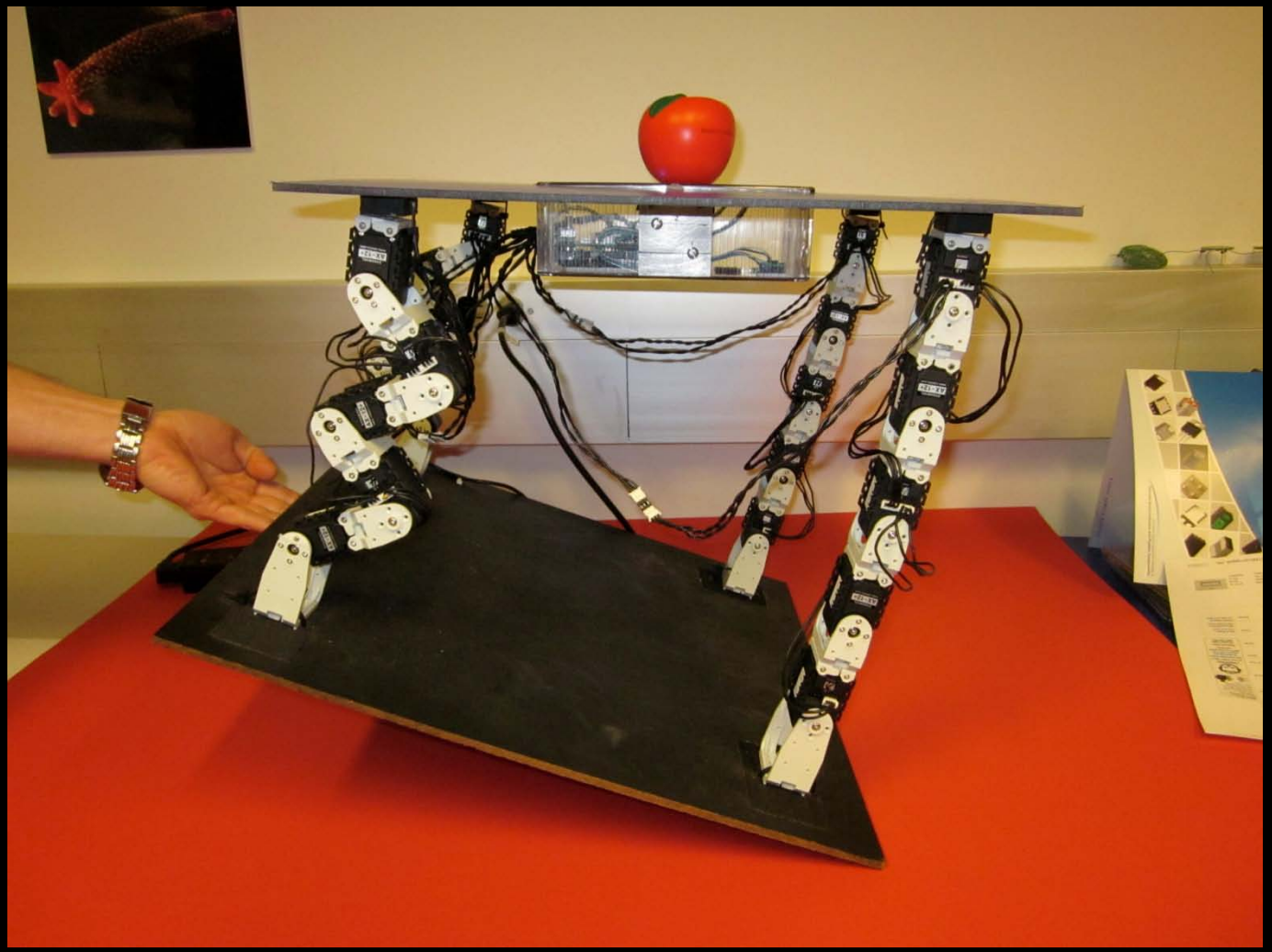
did you address the questions?

LATHER. RINSE. REPEAT.

make many (and profit)

THAT'S IT.

PROJECT GOAL: AUTONOMOUS ROBOT

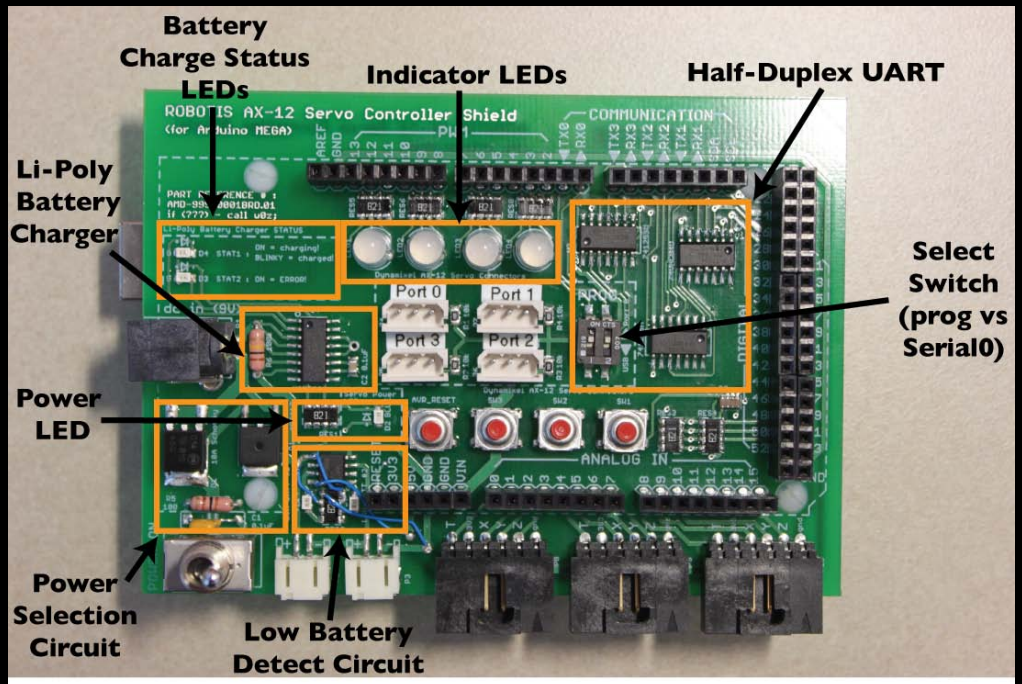


First, I asked some questions…

- Why are we making this?
- Who is this for?
- How will this be used?
- What features does it need to have (now)?
- What features does it need to have (later)?
- What are the legacy requirements?
- Who's going to build this?
- How many do we want to make?
- What is the budget?
- What is the timeline?

All the answers pointed to making a shield for the ARDUINO MEGA.

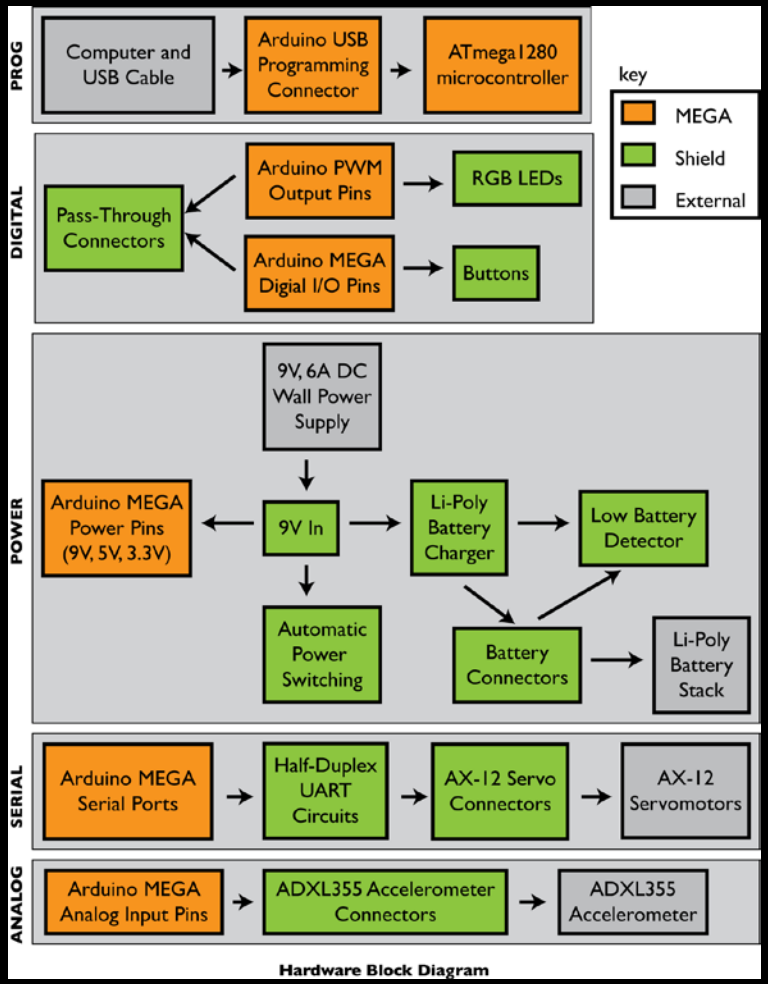
- All the benefits of open-source hardware
- Enough resources to get the job done



ROBOTIS AX-12 Servo Controller Shield: Circuits

design (record your rationale)

I grouped the required features into a block diagram and looked for existing solutions – REDUCE, REUSE, RECYCLE

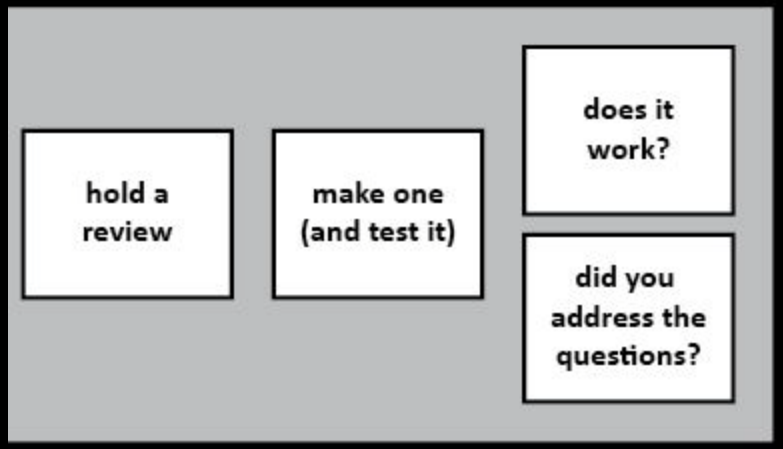


- application notes
- vendor datasheets
- old projects
- cookbooks
- open-source community

design (record your rationale)

Hardware Design work-flow

- Parts Selection and Schematic Capture
- Schematic Review – REVISIT QUESTIONS
- Layout Floor-planning (mechanical)
- PCB Layout
- Schematic + Layout Review – REVISIT QUESTIONS
- Pre-Tapeout Verification
- Manufacturing Tape-out
- Test and Characterization
- Iterate (if necessary)
- Document
- Release



BEST PRACTICES: SCH/BOM

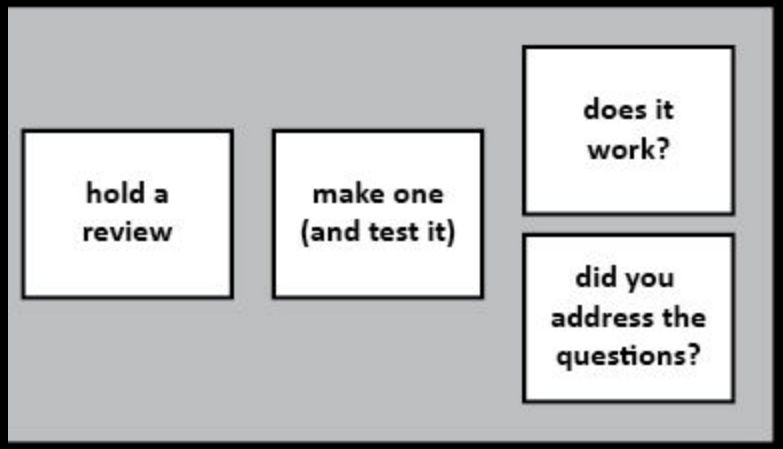
- as you select parts for your schematic, curate your CAD library and update your Bill of Materials as you go. This helps you to avoid footprint and pinout mistakes

- for every part on your BOM, take the extra time to find multiple vendors and list both the FUNCTION of the part and its CRITICAL SPECIFICATION (tolerance, size, cheapness, etc).

- for each part make a note of any physical Design Rules for the PCB layout.

BEST PRACTICES: PCB/PROTOTYPE

- Take your Design Rules and follow them to the best of your ability.
- Verify against Schematic.
- Review with others. REVISIT QUESTIONS.
- Built. Test.
- Iterate until satisfied.



Pre-Tapeout checklist

- Have you fixed all DRC/ERC errors?
- All part footprints on PCB match BOM?
- All part pin-outs on schematic match data sheet?
- Does your schematic match your working proto?
- Did you verify the critical spec for each part?
- Did you find the right vendor part number for each part?
- Is your part in stock? (BUY IT NOW)
- All Pin-1 designators correct?
- All RefDes labels correct?

Fabrication Package Checklist

- GERBERS
- NC Drill File
- Assembly Drawing
- Pick & Place Coordinates
- BOM
  - Part ID
  - Reference Designator(s)
  - Part Type
  - Package Footprint
  - Value/Description/Critical Spec
  - Manufacturer's Part Number
  - Vendor's Part Number

So, now your design documentation is just compiling what you already know

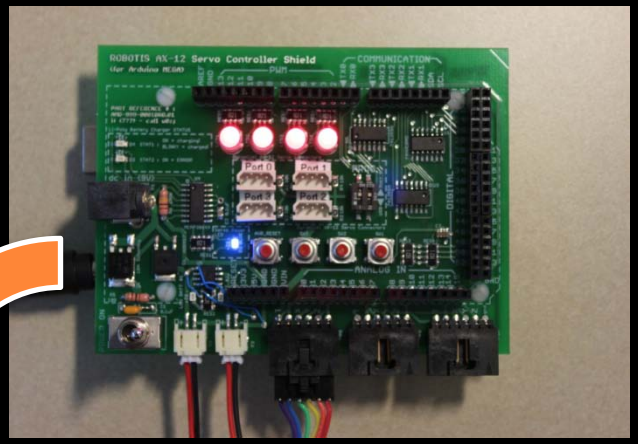
- Project Introduction (Goals, Overview)
- System Block Diagram
- Discussion of Essential Features/Trade-offs
- Block-by-Block Breakdown
  - Function
  - Schematic block
  - Layout block
  - Parts selection (and critical specs)
  - Performance metrics (if applicable)
- Software/Firmware Summary
- Typical Application
- User's Quick-Start Guide
- Errata

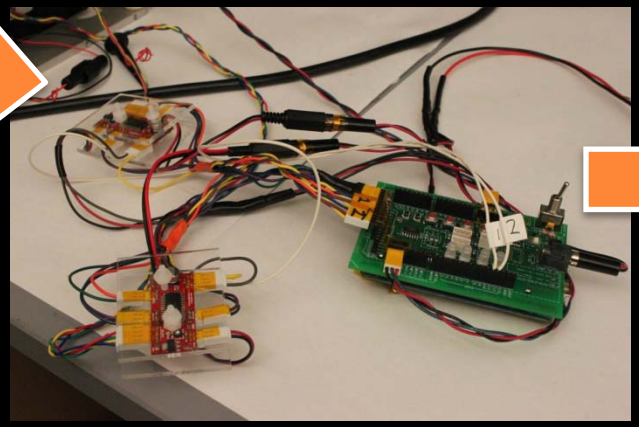
make many
(and profit)

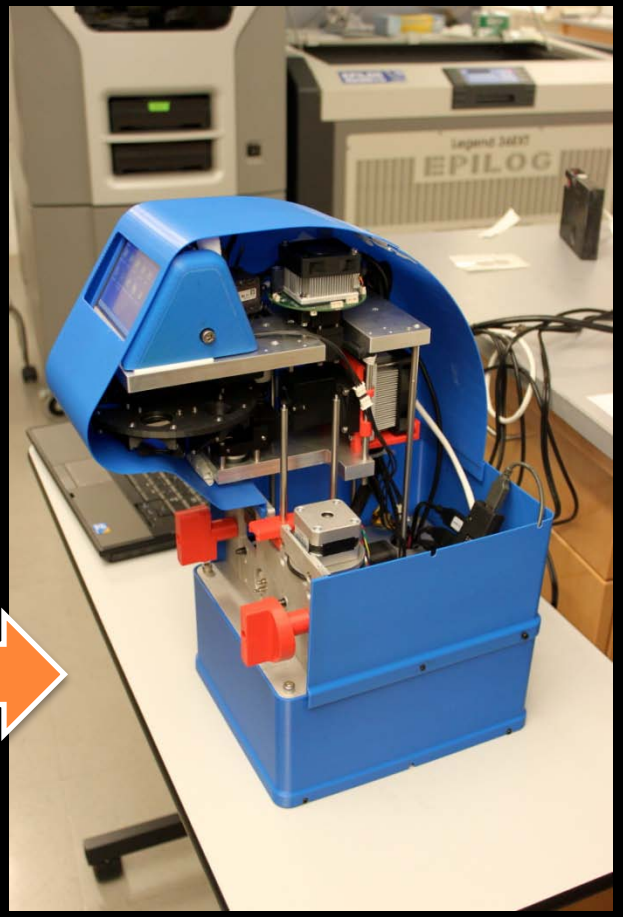
Why bother with all of that process and detail for a demo?

make many
(and profit)

When you follow a process, you gain efficiency. When we needed to prototype a diagnostic microscope, I had all the parts.

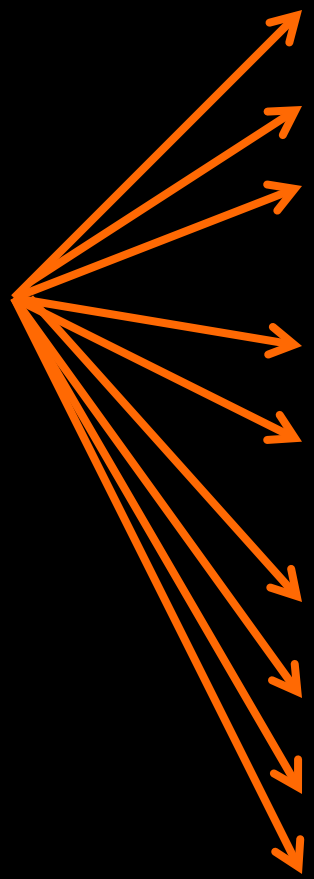






Hobby project or industry product, the gist of the process remains the same – only the details change

- README.txt
- Schematic
- Gerber
- BOM



- Design History File
- Hazard Analysis
- Failure Mode Effects Analysis
- Risk Mitigation
- Statistical Performance
- Margins/Tolerance
- Lifetime Reliability
- SOPs
- etc… etc… etc…

PROCESS TRANSPARENCY ALLOWS YOU TO CONQUER:

- ANALYST BIAS – “The previous guy knew what he was doing better than I do.”
- HUBRIS – “I know best, that other guy is an idiot.”
- CARGO CULT THINKING – “Engineering must be magic. If we copy it and it looks the same, it’s got to work, right?”

The more we share, the more others can question our design… the faster we can learn from our collective mistakes and the sooner we can celebrate our collective successes.

Questions? Comments? Drop a line.

- professional contact: amanda.wozniak@wyss.harvard.edu
- personal contact: woz@mit.edu