ECE 411 Practicum: Project Phases

Andrew Greenberg

October 2, 2013

Project Phases

- Week 1-2: Concept $\rightarrow$ specification
- Week 3-4: Schematics and prototyping
- Week 5-6: Layout, parts orders, PCB orders
- Week 7-9: Assembly and testing
- Week 10: Demonstration and documentation

Week 1-2: Conception

- Conceptualize
  - Get your team together and brain storm.
  - Choose something cool/fun/interesting, but *most of all* "doable".
- Research
  - Industrial espionage: a time honored tradition (what are other people doing?)
  - What other thing like your concept exists?
  - What's close? What can you rip apart?
- Deliverables:
  - A few good concepts: "Innovate here"
  - Some light requirements (this is too technology driven to have hard core requirements)
  - Some rough L0 and L1 block diagrams
  - Rough schematics (napkins OK)
  - Initial component research - "Hey, this looks cool!"

Week 1-2: Requirements

- What does it do?
- How does it get used?
- Who uses it?
- What does it look like?
- What is novel/interesting about this concept?
- What aspects of it are important?
- Note that *none* of these are "How does it work?". That's a specification!

Week 1-2: Specifications

- Specifications: very specific engineering questions
- Feature list: what *exactly* is your device going to do?
- "Datasheet" numbers: Power? Size? Weight? Environment?
- Use statements: **must, should, may**.
- Use number ranges (minimum,typical,maximum) and tolerances
  - Must weight (,0.25,0.5) kg.
  - Vsupply must be 3.3 +/- 10 % V.
  - Battery capacity must have (250,500,) mAhr of capacity.
  - Device should run (,500,) times before spectacularly exploding.
- Begin filling in L1 block diagrams and component connections/data flow.
- Ramp up on your tools: Redmine, EAGLE, AVR Studio IDE, ?

Week 3-4: Parts Identification and Schematic

- *Most important part of a designer's job.*
- What parts are out there? How do you find one?
- What are their tradeoffs compared to other components?
- You will add and drop features based on what components you find.
- Google, Distributer's websites, and open hardware projects will help you here.
- Begin your schematic and be building component libraries in EAGLE.

Week 3-4: Prototyping

- If you're not *absolutely* sure how something works, you're going to want to prototype it.
- Purchase development boards and/or bare components. Hand wire ("hack") parts together.
- Your job: *reduce design risk!*
- In other words, try things out, experiment, hack, make mistakes, etc. Now is the time to get it wrong.
- Deliverables:
  - A much higher likely hood of success for your project.
  - Complete, error free schematics.
  - Bill of materials (BOM).
  - First compiling bits of firmware.

Prototyping: "The visible circuit"

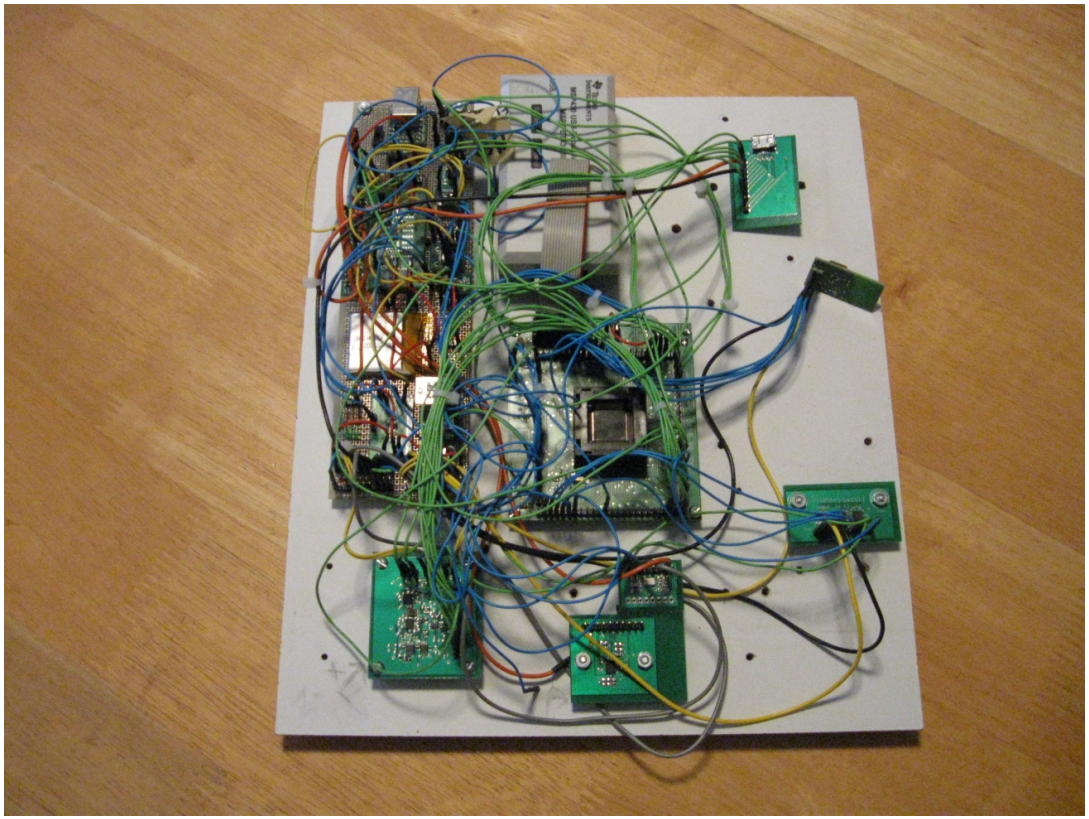

Figure: Easy to probe circuit, as well as interrupt signals and measure current.

Final board: Not-so-visible circuit.

Figure: Final PCB: Very, very hard to probe.

Week 5-6: Layout

- Manufacturing
  - Physical placement of components on the board.
  - Routing traces
  - DFx
- Deliverables:
  - Parts ordered.
  - Last minute changes to schematic, BOM and layout based on availability.
  - PCB layout complete with no DRC errors.
  - CAM files (Gerber files) created and emailed to PCB mfg.
  - Most of the firmware should now be prototyped.
  - GO BACK AND UPDATE YOUR DOCUMENTATION.

Week 7-9: Assembly and testing

- Assembly
  - Hand solder boards.
  - Test/programming jigs as necessary
- Testing
  - This is where the cold, hard universe passes judgment on your design.
  - *Subsection* bring up and testing.
  - Debugging and fixing. Lots of fixing.
- Firmware
  - Unless your prototype was really good, this is the time for firmware people to panic.
  - First boards mean *lots* of code debugging and last minute features.
- Deliverables:
  - Assembled, tested, and programmed boards.

Week 10: Demonstration and Documentation

- Explain and demonstration your design to your peers.
- Deliverables:
  - Final presentation/demonstration
  - Final documentation

Your job in this project

- Pick any two, maybe three: size, power, cost, performance, manufacturability, design time
- Your job is to make these tradeoffs!
- In your documentation, ALWAYS explain your choices. If the decision was arbitrary, *say so!*

Theoretical Design Process

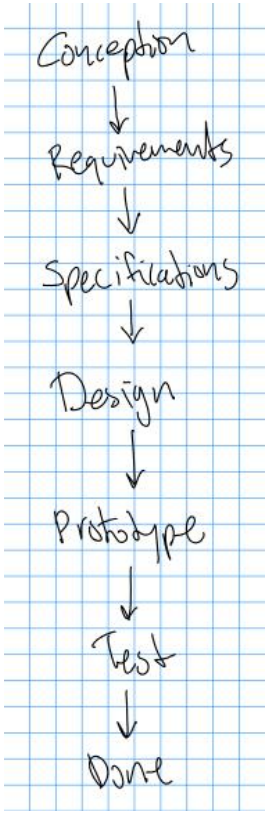

Figure: The Waterfall Model

Real World Design Process

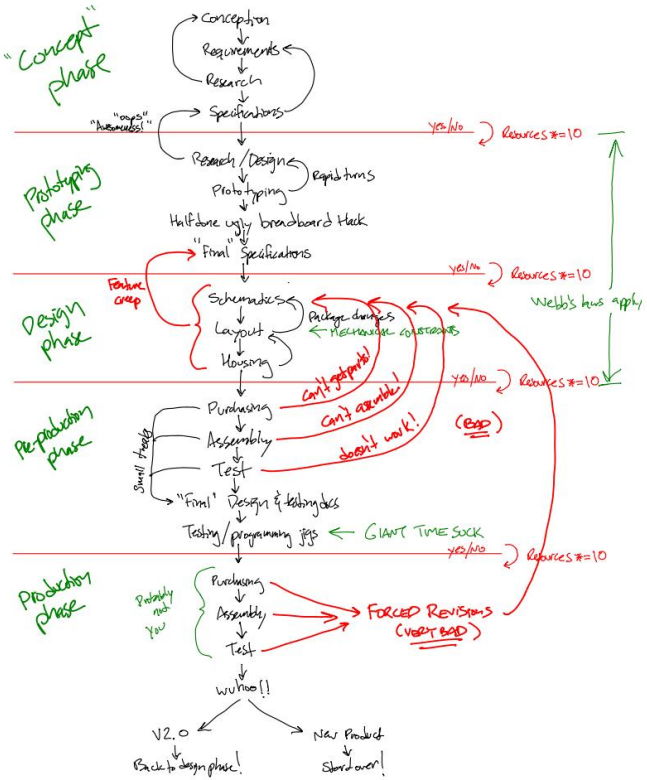

Figure: Not so much the Waterfall Model

A Better System Design Process: the spiral model

- Quick turns of:
  - **design** $\rightarrow$ **implement** $\rightarrow$ **evaluate**
- In ever widening scopes:
  - concept $\rightarrow$ design $\rightarrow$ implementation
- Documentation continuous evolves with project ("Living" documentation)
- Vaguely similar to agile methods in software engineering

Now back to Earth: Projects!

- Who's got some project ideas?
- Let's draw some block diagrams, discuss technology tradeoffs, etc.