Do-It-Yourself Devices

Personal Fabrication of Custom Electronic Products

David Adley Mellis

SB Mathematics
Massachusetts Institute of Technology, June 2003

MA Interaction Design
Interaction Design Institute Ivrea, June 2006

SM Media Arts and Sciences
Massachusetts Institute of Technology, September 2011

Submitted to the Program in Media Arts and Sciences
School of Architecture and Planning
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Media Arts and Sciences at the
Massachusetts Institute of Technology
September 2015

Signature redacted

Author: David A. Mellis
Program in Media Arts and Sciences
August 7, 2015

Signature redacted

Certified by: Mitchel Resnick
LEGO Papert Professor of Learning Research
Program in Media Arts and Sciences

Signature redacted

Accepted by: Pattie Maes
Academic Head
Program in Media Arts and Sciences

Do-It-Yourself Devices
Personal Fabrication of Custom Electronic Products

David Adley Mellis

Submitted to the Program in Media Arts and Sciences
School of Architecture and Planning on August 7, 2015
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Media Arts and Sciences at the
Massachusetts Institute of Technology

Abstract

Many domains of DIY (do-it-yourself) activity, like knitting and woodworking, offer two kinds of value: the making process itself and using the resulting products in one's life. With electronics, the sophistication of modern devices makes it difficult to combine these values. Instead, when people make electronics today, they generally use toolkits and other prototyping processes that aren't well suited to extended use.

This dissertation investigates digital fabrication (of both electronic circuit boards and enclosures) as an alternative approach to DIY electronics, one that can support individuals in both making devices and using them in their daily lives. The dissertation explores three questions: (1) What are the scope and limits of the personal fabrication of electronic products? (2) How can we engage people in the personal fabrication of electronic products? (3) Why make electronic products using personal fabrication?

These questions are explored through two investigations. The first is a DIY cellphone, including an autobiographical approach exploring my making and use of the device. Also documented are workshops and other dissemination in which others have made their own phones. The second investigation is a six-week workshop in which participants designed and made internet-connected devices.

The investigations reveal personal fabrication as a robust, open-ended, and nuanced means of making devices for use in daily life, but with limitations and constraints imposed by the commercial ecosystem surrounding this DIY practice and by the nature of electronic products. Analysis of the workshops reveals multiple trajectories that people take in these activities; the computational concepts, skills, and practices they develop; and strategies for engaging them. Finally, the investigations reveal multiple values for the personal fabrication of electronic products, including its ability to transform people's relationships with the technology in their lives.

Thesis Supervisor: Mitchel Resnick
Title: LEGO Papert Professor of Learning Research

Do-It-Yourself Devices

Personal Fabrication of Custom Electronic Products

David Adley Mellis

Signature redacted

Leah Buechley
Associate Professor of Media Arts and Sciences (2009–2014)
Massachusetts Institute of Technology

Do-It-Yourself Devices

Personal Fabrication of Custom Electronic Products

David Adley Mellis

Signature redacted

Björn Hartmann
Associate Professor, Electrical Engineering and Computer Science
University of California, Berkeley

Acknowledgements

Thank you to my advisors, past, present, and future. To Leah Buechley, for creating High-Low Tech, for inviting me to be a part of it, and for your passionate dedication to changing the culture of technology. To Mitchel Resnick, for welcoming me into Lifelong Kindergarten and for sharing your commitment to creativity and learning. To Björn Hartmann, for inviting me to continue my work with you at UC-Berkeley, and for your insight and clarity.

Thank you to all the members of High-Low Tech and Lifelong Kindergarten, for your ideas, collaboration, and encouragement over these six years: Hannah Perner-Wilson, Emily Lovell, Jie Qi, Ed Baafi, Sam Jacoby, Jennifer Jacobs, Kanjun Qiu, Amon Millner, Jay Silver, Eric Rosenbaum, Ricarose Roque, Sayamindu Dasgupta, Tiffany Tseng, Abdulrahman Idlbi, Champika Fernando, Alisha Panjwani, Shrishti Sethi, Juliana Nazaré, Natalie Rusk, Kreg Hanning. And to everyone that has made the Media Lab and MIT an amazing place to be: Amit Zoran, Sean Follmer, Lining Yao, David Moinina Sengeh, Nadya Peek, Che-Wei Wang, Taylor Levy, Pau Ten, Jonathan Ward, Peter Schmitt, Nan-Wei Gong, Benjamin Mako Hill, Jean-Baptiste Labrune, Dana Gordon, Mark Feldmeier, Daniel Leithinger, Roy Shilkrot, Edwina Portocarrero, Marcelo Coelho, Dimitris Papanikolaou, David Cranor, and so many more.

Thanks to Neil Gershenfeld, Hiroshi Ishii, Neri Oxman, and the rest of the Media Lab faculty for your inspiring research visions, and to Linda Peterson, Keira Horowitz, John DiFrancesco, Tom Lutz, Abisola Okuk, Karina Lundahl, Stephanie Gayle, Paula Aguilera, and all the others who make this place work.

Thanks as well to Massimo Banzi, David Cuartielles, Tom Igoe, and the rest of the Arduino team for your collaboration and friendship over the past ten years.

Finally, thank you to my family, for everything, and to Dena Molnar, for your love, support, and understanding during these past four years.

Contents

1. Introduction

Some things give us value through use. They serve a purpose, allow us to do things we couldn't do otherwise, or to do them faster, better, easier. Other things provide us with an opportunity to exercise our creativity, to enjoy an experience, to develop our abilities. Making things for use in our own lives is a chance to combine these values. When we knit a hat, build a chair, or even just cook dinner, we exercise our creativity and invest meaning in an activity — but we also have something to keep us warm, a place to sit, and food to eat.

In some areas of our lives, it's difficult to combine these values. Today's electronic products provide us with incredible value. With these devices, we can stay in touch, access information, and entertain ourselves wherever we are. And yet, while these products increasingly pervade our lives, most of us have little idea of how they're made and almost no involvement in their production. This wasn't always the case. In the 1950s and 60s, it wasn't uncommon for technically-minded individuals to assemble their own products from do-it-yourself kits. These kits were often comparable in quality to commercial products and significantly cheaper. Assembling them for oneself involved similar processes and tools as the ones used in industry, just on a smaller scale. These kits didn't necessarily provide much flexibility or creative freedom to the person assembling them — but at least they were an opportunity to participate in the process of making an electronic product. The complexity and sophistication of today's electronic products — and the tools and processes used to assemble them — means that it's no longer possible for individuals, no matter how skilled, to replicate these devices on their own.

Instead, over the past three decades efforts to engage amateurs in working with electronics have frequently taken the form of toolkits, collections of higher-level physical modules which can be combined in a variety of configurations. These toolkits have a number of advantages. They encapsulate the underlying complexity of the electronic circuits involved. They facilitate tinkering by making it quick and easy to try out new configurations of modules. They provide a curated set of components which work well together and suggest possible uses. These properties make them well-suited to creative exploration and meaningful experiences.

Toolkits, however, are less well-suited to the construction of products for use in daily life. The overhead imposed by wrapping basic components in higher-level building blocks restricts the size and aesthetics of the objects built with them. The ease with which modules can be connected and disconnected means that they may not be robust enough for long-term use. The curated set of components included in the toolkits limits the nature and variety of the products that can be built with them.

Other approaches to prototyping electronics are also well-suited to experimentation but not ideal for the creation of finished products. The Arduino electronics prototyping platform, for example, includes a core set of hardware and software modules to facilitate the creation of interactive prototypes. Rather than providing a fixed set of modules like many toolkits, the Arduino platform works with a variety of off-the-shelf electronic components using solder-less breadboards and other techniques. This enables experimentation with a wide range of technologies and projects. The use of breadboards and jumper wires, however, can lead to bulky and fragile prototypes and makes it difficult to work with surface-mount components. As a result, these techniques are better suited to the creation of early-stage prototypes rather than products intended for extended use.

The increasing accessibility of digital fabrication technology suggests an alternative approach, one that combines opportunities for individual creativity with the ability to make products for daily use. Digital fabrication allows individuals to create one-off objects — including printed circuit boards (PCBs) — directly from digital design files. Individuals can gain access to the technology via low-cost machines targeted at hobbyists, community spaces hosting more expensive machines, and on-demand services offering a variety of technologies. Free, open-source, and low-cost computer-aided design (CAD) tools facilitate creation of the digital design files. These designs can be shared online for others to make or modify (a practice known as open-source hardware). This personal fabrication approach allows for a precision, optimization, and reproducibility not found in more manual construction methods. At the same time, it offers more flexibility than mass production — unique artifacts can be created from the direct specifications of a single individual and the design can be revised for each subsequent iteration.

The combination of personal fabrication and DIY electronics means that individuals can design and produce sophisticated electronic products. Complex and optimized printed circuit boards can be designed using open-source or inexpensive software, precisely fabricated in small quantities using relatively low-cost digital fabrication machines or on-demand online vendors. When combined with the broad range of available electronic components, these PCBs can reliably yield a unique electronic circuit. Together with enclosures produced using 3D-printing, laser-cutting, or other fabrication processes, these circuits can form complete, attractive, and robust electronic products, produced in small quantities or as one-offs.

This approach has been championed by MIT professor Neil Gershenfeld, founder of the FabLab initiative and author of *Fab: The Coming Revolution on Your Desktop — From Personal Computers to Personal Fabrication* [2007]. In Gershenfeld's "How To Make (Almost) Anything" course at MIT (and the distributed online version called Fab Academy), students learn to combine digital fabrication with circuits to produce custom electronic products. There are also some commercial electronic kits and products made using digital fabrication. For example, Adafruit sells a variety of clocks that combine open-source PCBs with laser-cut acrylic or other enclosures.

Still, the design and personal fabrication of electronic products by individuals has not been well-studied or understood. Even in the case of relative experts, like MIT students, it's not clear how to best go about making an electronic product using digital fabrication, or to what extent this approach can match the capabilities of industrial production. Furthermore, it's not clear who else might be interested in these activities, what might motivate them to participate, and what support they'd need to do so.

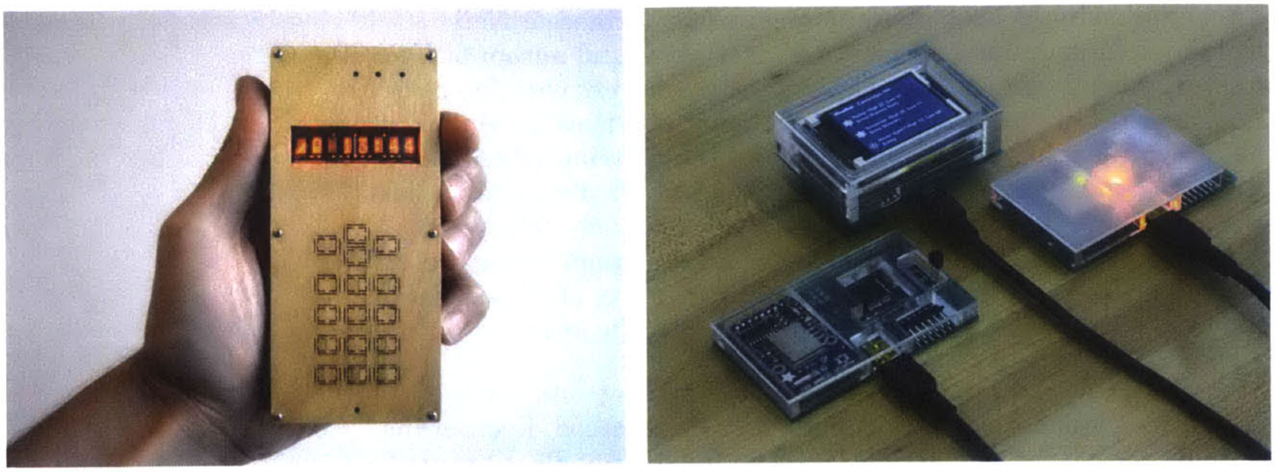
This dissertation builds on the work of my master's thesis [Mellis 2011], which explored the design space of personally-fabricated electronic products, including technological and aesthetic best practices. These lessons are summarized here in the chapter entitled "The Personal Fabrication of Electronic Products". This dissertation builds on this practical understanding to explore three more fundamental research questions:

- What are the scope and limits of the personal fabrication of electronic products?

- How can we engage people in the personal fabrication of electronic products?

- Why make electronic products using personal fabrication?

For the first of these questions, I consider the perspective of the expert practitioner, exploring the possibilities and constraints that come from the technology and processes themselves, rather than the knowledge or skills of particular individuals. The second research question concerns the ways in which those with relevant skills and expertise can engage novices in the process of fabricating their own devices. Partly, this is about understanding the different trajectories that people will take through the process. Partly, it's about understanding the computational concepts, practices, and skills they'll need to be successful in these activities. It's important to note that in exploring this question, I'm interested in the possibilities that new activities and contexts, combined with personal support and facilitation, can bring to currently-available tools and technologies — rather than in building new software or tools. The third question



The DIY cellphone (left) and my examples for the connected devices workshop (right).

explores the values that novices and experts can derive from engaging in these activities, in terms of both the products themselves and the process of making them.

These research questions are explored through two primary investigations. The first is my DIY cellphone, a device I've designed, built, and used in my daily life over the course of two and a half years. It's a basic GSM phone that combines a custom PCB with a variety of digitally-fabricated enclosures. The cellphone illuminates the possibilities and limitations of the personal fabrication of electronic products as carried out by an expert practitioner. Workshops in which others made and customized the DIY cellphone provide lessons on engaging people in relatively constrained (but still potentially meaningful) personal fabrication activities. Sharing the design files and instructions for making the phone online reveals possibilities for the distributed creation of electronic products.

The second investigation is a six-week workshop in which I guided eight adult participants through the process of designing and fabricating their own internet-connected electronic products. These are relatively simple circuits containing a microcontroller, wifi module, and other basic electronic components, sensors, and actuators. Participants in this workshop defined and implemented their own products based on my examples, a process that required more extensive involvement and understanding than the DIY cellphone workshops. This workshop offers lessons about the roles of tools, materials, information, and in-person support in engaging new groups in the personal fabrication of an electronic device.

It's important to note that although programming is an essential part of the process of making many electronic devices, it's not been a focus of the research described here. There is much existing work on building new tools to support programming and on engaging new

	DIY Cellphone	Connected Devices
Approach	• Autobiographical investigation • Short workshops • Online dissemination	• Extended workshop
Technologies	• GSM module • Arduino-compatible microcontroller • Multiple fabrication processes	• Wifi module • Arduino-compatible microcontroller • Emphasis on PCB fabrication
Results	• Design reflections • Personal reflections • Workshop trajectories • Dissemination lessons	• Participants trajectories and profiles • Computational concepts, skills, and practices • Discussion of values

Summary of the investigations.

audiences in this activity. I've chosen to focus on the more novel aspects of building devices using electronics and digital fabrication.

The investigations described here showcase the rich design space made possible by a personal fabrication approach to electronic products. It is open-ended, nuanced, and continuously-variable in a way that permutations of toolkit modules are not. It yields robust devices for use in daily life that would be difficult to make with other prototyping processes. On the other hand, the investigations also make clear that personal fabrication isn't capable of matching the sophistication of mass production. The commercial ecosystem surrounding individual DIY practice constrains what's possible. Electronic products are heavily dependent on electronic components and individuals don't have access to many of the latest components, and couldn't work with them even if they did. Furthermore, these attempts to create relevant and timely devices highlight the uniques challenges that today's electronic products pose for DIY practice. Their sophistication and that of the processes used to produce them far exceeds what individuals have access to. Frequent changes in technology mean that devices need to be redesigned frequently, to accommodate both the latest components and rapidly-changing user expectations. This creates additional burdens for an individual maker. In addition, many regulatory structures were created with mass production in mind and impose disproportionate overhead on the personal fabrication of a small number of devices.

The dissertation workshops reveal many lessons about engaging people in the personal fabrication of electronic products. They highlight multiple trajectories that people may take in the process, only some which are well served by a personal fabrication approach.

They reveal computational concepts, skills, and practices that people develop in the process, expanding on those discussed in prior related work. They highlight strategies for engaging people in the process and reveal the importance of support from people as well as support from technology in doing so.

Despite its limitations, personal fabrication offers an opportunity to combine creativity and individual involvement with the ability to create robust electronic products for use in daily life. Personal fabrication makes it possible to build products that wouldn't exist otherwise and to test out new ideas with extended, real-world trials. It leverages the power and flexibility of software to enable the creation of specific and optimized devices that would be difficult to match with fixed toolkits or breadboard prototypes. Perhaps more importantly, it offers the possibility of changing people's relationships with the electronic products in their lives. A personal fabrication approach fosters an understanding of the way that electronic products are made and a sense that it's possible to make a device if you want to.

Looking to the future, there are many opportunities for new tools and technologies to expand the possibilities for novices to engage in the personal fabrication of electronic products. Software tools could encode more domain knowledge, providing additional scaffolding to individuals. Better workflows could facilitate distributed collaboration on complex devices. Personal fabrication could be applied to make products for people with specialized needs or wants. Still, there are many questions about the extent to which personal fabrication will gain or lose in relevance as technology continues to evolve. The possibilities that personal fabrication offers for individual empowerment are tempered by the fact that engagement in these activities can illuminate the gap between the capabilities of the individual and the power of industry. It shows that people may perhaps be able to do much more than they imagined possible but still much less than they might want.

Contributions

This dissertation contributes the following:

- A framework for understanding the personal fabrication of electronic products, its elements, and the ways in which it compares with other means of making electronics.

- Design guidelines and best practices for personally-fabricated electronic products.

- Examples of personally-fabricated electronic products and the process of designing, making, using, and iterating on them. This includes:

    - Potential aesthetic and functional qualities.
    - The constraints and limitations imposed by the available tools and technologies.

- Lessons on engaging new audiences in the personal fabrication of electronic products, including:
    - Examples of and principles for tools and contexts that can foster these experiences.
    - Different trajectories that people take through the process.
    - A breakdown of the computational concepts, practices, and skills that are required for, and can be developed through, these activities.
    - Strategies for motivating people to participate in these activities and helping them succeed in them.

- Analysis of the values that can be derived from the personal fabrication of electronic products, including new possibilities for the products themselves and the personal significance of participating in the process of making them.

Dissertation Roadmap

Chapter 2 provides relevant background on the relationship between technology and making. It contrasts the qualities found in craft, mass production, and digital fabrication, as discussed in contemporary works. It explores the nature of DIY and hobbies, with a specific focus on the DIY electronic kits popular after World War II. Finally, it provides an overview of today's "maker movement" of people engaging with DIY and technology.

Chapter 3 discusses related research from the human-computer interaction (HCI) community. It starts with a brief overview of computation and making. The chapter includes a longer discussion of strategies for engaging people in electronics, including toolkits for children and designers as well as the relationship between electronics and materials, craft, activities, and critical thought. This is followed by a survey of new interfaces for digital fabrication. The chapter then discusses the combination of digital fabrication and electronics, as explored in the HCI community. It closes with a brief survey of other research on the relationship between technology and making.

Chapter 4 provides a framework for the personal fabrication of electronic products. It starts by contrasting this approach with other means of making electronic devices. This is followed by an in-depth discussion of the elements of personally-fabricated electronic products, including fabrication processes themselves, electronic components, and embedded software. The discussion includes criteria for selecting these elements. The chapter then discusses three case studies of personally-fabricated electronic products (a radio, speakers, and a computer mouse). This is followed by a set of general principles for making and sharing personally-fabricated electronic products.

Chapter 5 discusses the DIY cellphone. It starts with a brief summary of related projects and a short description of the DIY cellphone itself. This is followed by a discussion of the phone's iterations and reflections on this design process. I then discuss my personal experiences using the phone in my daily life. This is followed by a discussion of the two workshops, including an overview, a summary of the participant trajectories, and an overview of the participants' reflections. The chapter concludes with a summary of the online and other dissemination of the phone's design.

Chapter 6 covers the connected devices workshop. It starts by discussing my process in preparing the example devices for the workshop. After a short description of the workshop itself, the chapter delves into four trajectories and profiles of the workshop participants. This is followed by an analysis of the computational concepts, skills, and practices that participants developed in the course of the workshop. The chapter ends with a summary of the values that workshop participants identified in DIY electronics generally and a personal fabrication approach more specifically.

Chapter 7 synthesizes the findings of the two investigations to address the three research questions. It discusses the scope of the personal fabrication of electronic products and the limits on this practice. It highlights multiple strategies for engaging people in the process. It discusses the values that I and others have found in going through the personal fabrication of electronic products, for both the products themselves and the process of making them.

Chapter 8 concludes the dissertation. It starts with a recapitulation of the document as a whole. This is followed by a discussion of opportunities for future work. Then, I discuss three questions that are key to the future possibilities for the personal fabrication of electronic products. I end with some ambivalent reflections on the possibilities for empowering people in their relationships with the technology in their lives.

2. Background

This chapter places the dissertation in the context of the relevant social and technical backgrounds. The first section looks at how society makes things by contrasting digital fabrication with craft and mass production. The next section looks at how individuals make things through a discussion of DIY and hobbies. This is followed by an overview of the electronic kit era that followed World War II. The chapter concludes with a discussion of today's DIY technology practices, popularly known as the "maker movement".

Qualities of Craft, Mass Production, and Digital Fabrication

To better understand what it means to make things with digital fabrication, it's instructive to compare it with the two other dominant paradigms for making objects: craft and mass production. Each of these three areas is a massive subject and a short discussion like this one can't provide even a brief summary. Instead, this discussion focuses specifically on the relationship between these different paradigms of making, as discussed by contemporary authors.

From today's perspective, mass-production appears as the dominant paradigm, only beginning to be challenged by digital technologies. But, as Mario Carpo [2011] points out,

> "In the long duration of historical time the age of mass-produced, standardized, mechanical, and identical copies should be seen as an interlude, and a relatively brief one — sandwiched between the age of hand-making, which preceded it, and the digital age that is now replacing it." (p. 10)

Furthermore, digital fabrication restores some of the qualities of craft that were lost in mass production: the possibility for individual variation in the artifacts produced, for instance, and the integration of design and making.

In distinguishing these paradigms of production, the key criteria is not necessarily the specific means by which an artifact is produced — i.e. by hand, by machine, or from a digital file — but rather the qualities of the process and the artifacts it creates. For, as Carpo points out, the ability to precisely create mechanical copies dates back at least as far as the advent of printing. This ability led to a mechanization of the making of many physical artifacts:

> "Standardized images preceded industrial assembly lines, and a culture of standardized architecture was already well

> established at a time when all visually standardized architectural parts ... had to be carefully handcrafted in order to *look* identical to one another."

Conversely, McCullough [1998] points out that many digital artifacts, while trivial to make copies of, are the product of manual, continuous, and personal processes of construction:

> "Think of a digital artifact, shaped by software operations, made up of data assemblies. Although lacking in physical substance, it is a thing with an appearance, spatiality, structure, workable properties, and a history. Although it does not bear the mark of someone's hands, as a clay pot does, neither is it the product of a standardized industrial process, like an aluminum skillet. It is individual, and reveals authorship at the level of its internal organization. It is unique, for although flawless copies can be made, nobody is going to make another just like unless by copying." (p. 155)

So if making something by hand can reproduce the qualities of mechanical reproduction, and making something on the computer can allow for the exercise of craft, what are the qualities that distinguish these paradigms from each other? We can think of these qualities along two dimensions (paralleling my first two research questions): designing artifacts and engaging people. The following discussion refers back to the context of this dissertation: the personal fabrication of electronic products, a process which involves digital design but also the assembly of existing physical parts (electronic components).

Designing Artifacts

Three qualities express some of the core contrasts between craft, mass production, and digital fabrication: similarity and variability; iteration and experimentation; materials and processes.

Similarity and Variability

One core concern is the notion of similarity and its inverse, variability. This has implications for how we identify objects. Carpo discusses three examples in the domain of money: a signature on a check, a banknote, and a credit card:

> "When objects are handmade, as a signature is, variability in the process of production generates differences and similarities between copies, and identification is based on visual resemblance; when objects are machine-made, as a banknote is, mass-produced, exactly repeatable mechanical imprints generate standardized products, and identification is based on visual identicality; when objects are digitally made, as are the

> latest machine-readable or chip-based credit cards, identification is based on the recognition of hidden patterns, on computational algorithms, or on other nonvisual features." (p. 4)

Craft and digital processes enable different kinds of variability.

> "Handmade objects can be made on demand, and made to measure. This makes them more expensive than comparable mass-produced, standardized items, but in compensation for their extra cost, custom-made objects are as a rule a better fit for their individual user." [Carpo 2006] (pp. 6–7)

In contrast, with digital processes "differentiation can now be scripted, programmed, and to some extent designed." (p. 7) Here variability is not the inevitable result of an inherently imprecise process (manual craft) but a deliberate and precise (although potentially probabilistic) result of computational processes. This kind of algorithm variability is not, however, the focus of this dissertation. Instead, it focuses on the opportunities allowed by digital process for designs to be modified and combined by multiple people — the individual creation of personal variations. The outcomes this enables are illustrated through the investigations described in this dissertation.

Iteration and Exploration

Another crucial characteristic of these different modes of production is the potential for continuity between one version of a design and the next, or changes in the nature of iteration. In mass production, the overhead of establishing a production line imposes a strong separation between one version of a product and any revisions to its design. The ability of digital fabrication to produce an object from a new or modified design in a few hours facilitates much faster iteration. McCullough [1998] points out:

> "Tightening the loop between conception and execution has the potential to reconcile some of the separation of design and fabrication that industrialization had previously imposed on craft. Thus, after two centuries of separation, the conception and the execution of everyday objects are once again in the same hands. (So to speak — hands actually touch less than ever before.)" (p. 178)

Experimentation is further enabled by the ability to explore alternatives in software before producing any physical artifacts. CAD software makes it possible to rapidly generate and evaluate alternatives, particularly with respect to geometry. In electronics, this digital experimentation is more difficult, since evaluating the behavior of a circuit design requires either sophisticated simulation or physical fabrication. The nature of iteration and experimentation allowed by the personal fabrication of electronic product is one of the core qualities explored by this dissertation.

Mediums and Materials

Both Carpo and McCullough celebrate the richness and diversity of forms that can be achieved with digital design processes, and their overcoming of many of the limits of older manufacturing processes. Carpo notes that digital design tools can provide the precision and unambiguousness needed to construct complex forms that would be difficult or impossible to specify with 2D drawings and construct with non-CNC machines. That is, digital processes can overcome the "notational bottleneck" created by the need for construction drawings to mediate between the architect's intent and the physical materials of a building, both of which are capable of much complexity and irregularity.

McCullough suggests that digital design software provides something akin to a physical medium:

> "Increased notational density supports quasi-continuous operations formerly only available from physical materials.... It is fair to assert that despite the lack of physicality there exists a growing possibility of constructing the experience of a medium in the world of the computer." (pp. 214–215)

Furthermore, he points out that the qualities of these digital media are in large part shaped by the "vocabularies and operators" provided by particular software applications.

Neither Carpo nor McCullough devote much consideration to the question of components or parts, perhaps because they pose a challenge to the transformations enacted by the shift to digital design processes. In designing the electronic products in this dissertation, however, the properties and availability of electronic components has played a key role in shaping the products that can be made. This re-imposes some of the constraints of mass production that might otherwise be transcended through the use of digital design processes.

Engaging People

Digital fabrication foreshadows a return to the integration of designing and making found in craft production. This has implications for notions of authorship as they relate to physical products and for what it means to engage people in producing those products.

Authorship

Craft, mass production, and digital fabrication have different implications for the notion of authorship. When the same person both designs and makes an artifact, the notion of authorship is relatively unambiguous. McCullough gives the example of painting as an area in which these two activities are typically combined. When the activities of design and production are separated, notions of authorship become more complicated. In some areas, it is common to associate authorship with the creator of the design. Writing itself is the canonical example — from which the term "authorship" derives. Classical music is another, in which the creator of the score is seen as the author of the piece of music, although it may be performed by others (an example given by McCullough).

Architecture is somewhere in the middle. "It does use intermediate representations — drawings — yet we do not think of these as the work." [McCullough 1998] (p. 93) Some, however, argue for the idea of architect as author. Carpo situates the original force for this idea (if not its earliest expression) with Leon Battista Alberti, author of a famous Renaissance treatise on architecture. He emphasized the importance of precise architectural drawings and, with them, the identification of the architect as the author of the building. This requires seeing the design as "notationally identical" with the final product, despite their obvious differences in form, scale, etc.

Similarly, to the extent that a digital fabrication process reliably creates objects according to an individual's specification, the individual can be clearly identified as the author of that object. The more human skill, judgement, and interpretation is required in the assembly process, the more variation is introduced in the resulting artifact and the more the notion of authorship is blurred. We see this distinction in the work of this dissertation. Some processes operate reliably and autonomously, closely identifying the author of the object with the designer of its digital file. Others require significant manual skill and judgement (e.g. in the form of troubleshooting), giving more agency to the maker and, perhaps, lessening the importance of the original designer. The DIY cellphone workshops, in which participants primarily assembled a device that I designed, illustrate some of these ambiguities and tensions.

Engagement

The separation of designing and making doesn't simply change our notion of the author of a work; it profoundly transforms the nature of the labor involved in producing an artifact. This division of labor underlies a long history of criticisms of mass production, summarized in relationship to today's digital craft practices by David Gauntlett in *Making is Connecting* [2011]. Gauntlett quotes John

Ruskin, one of the most prominent critics of the industrial revolution, on the division of labor:

> "It is not, truly speaking, the labor that is divided; but the men: — Divided into mere segments of men — broken into small fragments and crumbs of life; so that all the little piece of intelligence that is left in a man is not enough to make a pin, or a nail, but exhausts itself in making the point of a pin, or the head of a nail." (John Ruskin, "The Nature of the Gothic" in *The Stones of Venice vol. II,* 1853)

Instead, we should combine the roles of designer and maker, as was the case in the craft tradition. As Ruskin puts it:

> "We are always in these days endeavoring to separate the two; we want one man to be always thinking, and the other to be always working, and we call one a gentleman, and the other an operative; whereas the workman ought often to be thinking, and the thinker often to be working, and both should gentlemen, in the best sense." ("The Nature of the Gothic")

For McCullough, digital fabrication offers opportunities to re-integrate these roles, albeit with fabrication machines taking over many of the labors of the workman. He discusses the nature of the engagement enabled by the computer as a medium, and the opportunities it offers for play and mastery. These are qualities that I'm seeking in my personal fabrication approach to the making of electronic products.

DIY and Hobbies

In order to understand the meaning and relevance of the personal fabrication of electronic products, it's useful to consider it in the context of other types of DIY and hobby activities. Looking at other domains in which people make things for themselves helps clarify which aspects of the current activity are unique to the specific technologies and activities involved and which derive from the general nature of DIY. This isn't a particularly well-studied field, but there are enough detailed analyses to begin to draw some common themes and useful lessons.

In his book *Hobbies* [1999], Steven Gelber traces the history of this category of activities. He situates the origin of the modern notion of a hobby in the 1880's, prior to which it referred to any preoccupation, including political causes or personal fixations. The notion of a hobby as a productive use of leisure time depended on the separation between work and leisure that occurred during the industrial revolution. Gelber broadly distinguishes two classes of hobbies, collecting and crafts, of which only the latter is relevant here. He discusses several periods of craft hobbies, including women's handcrafts in the 19th and early 20th centuries, DIY activity during the Great Depression, the kit craze of the 1950's, and the flourishing of DIY home improvement after World War II.

Gelber describes hobbies as productive leisure, activities that share the pleasurable and voluntary nature of leisure but that also embody many qualities of work. Like work, hobbies involve the application of personal skill and effort towards the construction of some (at least notionally) useful outcome. Unlike work, the form and outcome of the activity is chosen and controlled by the individual. Hobbies therefore reflect, for Gelber, both a spillover of work values into leisure time and a compensation for qualities lacking at work. This combination is some ways contradictory but can also prove deeply satisfying:

> "Both men and women employees perceive craft hobbies as a perfect job, freely started and freely stopped, controlled from beginning to end by the worker, and resulting in a product that reflects the skill and imagination of the hobbyist. For workers satisfied with their jobs, this perfect job is part of the integrated pattern of their lives and validates both the skills and the values that have made them successful at their vocations. For the dissatisfied worker the hobby is an escape into the job they wish they had." (pp. 156–157)

This notion of hobbies as the perfect job forms some of the motivation for the current research. It attempts to provide an opportunity for individuals to exercise and develop their technical and creative abilities in voluntary, pleasure context. As McCullough [1998] points out, "recreational craft is more satisfying than mere amusement, precisely because it is merged with work." (p. 222)

In his introduction to a special issue of the Journal of Design History devoted to the topic, Paul Atkinson [2006] provides a brief survey of DIY and its history. He sees DIY as an explicit counter to mass-production — "the antithesis of the prescribed design of the mass marketplace". Atkinson points out the difficulty of providing a precise definition or categorization of DIY:

> "Where are the boundaries to be drawn between different levels of activity ranging from handicrafts to home maintenance, interior decorating, interior design, garden design, vehicle maintenance and customization, home improvement and self-build homes?"

(To which I would add electronics and personal fabrication.) He notes that while many types of DIY could be grouped under the areas "the making of objects" and "maintenance of the home", these categories include activities with a wide range of creative input and different motivations for carrying them out.

Atkinson divides the motivations for participating in DIY into four broad categories: pro-active DIY, reactive DIY, essential DIY, and lifestyle DIY. Pro-active DIY consists of those activities that involve a high degree of personal creativity and self-direction, whereas reactive DIY involves simpler, guided activities like kit assembly. Essential DIY covers those activities undertaken for reasons of financial or other practical need, whereas lifestyle DIY consists of "activities undertaken as emulation or conspicuous consumption". An even simpler classification might reduce this to a single spectrum, with economic or other practical motivations at one end and motives of self-expression or personal identity at the other, with perhaps another dimension reflecting the degree of creativity and effort involved. Some forms of DIY, like home improvement, are often associated with practical motivations, while others have more to do with self-expression or identity. DIY electronics, as we'll see in more detail shortly, has changed with respect to this dimension of motivation. Early DIY electronics often offered practical cost-saving benefits whereas more recent activity tends to emphasize creativity and experimentation.

Electronic Kits

In the years after World War II, kits of all kinds exploded in popularity, with sales going from $44 million in 1945 to $300 million in 1953. [Gelber 1999] They came in many diverse forms, but popular materials included plastic, leather, and wood. The Tandy Corporation, for example, begin producing leather kits in 1950, which included pre-cut leather parts and the other materials needed to assemble wallets, purses, and other objects. The Revell company produced plastic kit models of a broad range of cars, trains, planes, and other forms of transportation. These plastic models dominate the documentation of the kits of the period. Other kits yielded larger, more complex objects. The Mirror Dinghy, for example, was a popular UK kit for constructing a wooden boat. Over 70,000 were sold between its launch in 1963 and 2004. [Jackson 2006]

Part of this movement was a range of kits for building electronic products. Electronics kits did exist in some form prior to World War II, for example in the domain of amateur radio (also known as "ham" radio). Ham radio kits existed as early as the 1920's but early kits tended to be simple collections of parts, without much documentation. Some of these early kits didn't even include all the pieces needed to construct a project, merely the most specialized or expensive ones. [Haring 2007]

Electronic kits took off after World War II. Through the 1950's, 60's, and 70's, countless versions allowed individuals to build their own radios, hi-fi audio equipment, test equipment, and more. The most popular producer of these kits was the Heath company, maker of

Heathkits. They created their first electronics kit (an oscilloscope) in 1947 and produced many more over the next 45 years. These kits allowed hobbyists to assemble electronic devices for less (sometimes significantly less) than buying similar pre-assembled products. As Chas Gilmore, former Manager of Design Engineering at Heath, described their customers, "they were hungry for electronic products, and they were very willing to exchange pleasurable sweat-equity (i.e., their time) for a substantial reduction in the cost to acquire them."[1] An retrospective article in the New York Times states that "Heathkits used to cost about 30 percent less than assembled components."[2] This ability to assemble devices comparable to commercial products, and at reduced prices, has faded with the increasing sophistication of today's electronic products, making it difficult to justify today's electronics DIY with the same practical motivations.

Some sense of the scale of the post-World War II DIY electronics activity can be gained from data on the revenues and product offerings of Heath and related companies. A 1955 article announcing the sale of the Heath company listed its annual sales as $7 million,[3] or about $60 million in today's dollars. The next year, Heath offered more than 60 products.[4] At this point, revenues were a small fraction of the total kit market (given by [Gelber 1999] as $300 million in 1953). By 1961, however, total sales of electronic kits had reached $50 million (nearly $400 million in today's dollars). By then, Heath offered more than 250 products.[5] A year later, the company had $20 million in sales (more than $150 million in 2015 dollars) and, in 1972, that had grown to $66 million (almost $375 million in today's dollars).[6]

Electronic kits were not the only means of supporting hobbyist engagement in the construction of DIY electronics. A range of publications offered project ideas and instructions. *Popular Electronics* started in 1954 and included a range of electronics projects. Circulation was around 400,000 from the mid-1960s until the original publisher, Ziff-Davis, ceased publication of the magazine in 1985. The January 1975 issue famously featured the early personal computer (kit) the Altair 8800. Articles in the magazine

[1] Lou Frenzel, "Heathkit: A Right-Time, right-Place Business", Electronic Design, Sept. 24, 2012.

[2] Lawrence M. Fisher, "Plug Is Pulled on Heathkits, Ending a Do-It-Yourself Era", *New York Times*, March 30, 1992.

[3] "Michigan Concern Sold to Daystrom", *New York Times*, February 1, 1955.

[4] "Switch to Electronics Sparks Rise In Profits This Year by Daystrom, Inc.", *Barron's National Business and Financial Weekly*, Oct 1, 1956.

[5] John Johnsrud, "Producers of Electronic Kits Assemble Profits", *New York Times*, Nov 26, 1961.

[6] Evelyn Keene, "Kit provides spark for do-it-your-selfer", *Boston Globe*, Aug 19, 1973.

often came with images of printed circuit board designs for the reader to reproduce. Kits of the parts required for a project were offered for sale as well. Today, the possibilities for individual DIY are similarly shaped both by the availability of physical parts as well as information on the uses to which those technologies can be put.

According to one article, "most of the people who built Heathkits were not electronics engineers, or even engineer- wannabes"[7] — this was a hobby with appeal beyond the professional engineer. No previous electronics knowledge was required to assemble a Heathkit and the company prided itself on the high probability of successfully completing their kits. Their slogan was "we won't let you fail." Many customers built multiple kits and "it was not uncommon for Heathkit loyalists to exceed the 100 mark."[8] It's not clear how often Heathkit customers moved on to more open-ended forms of DIY electronics. As Gelber [1999] notes in reference to other forms of kits:

> "A kit could become the first step in a lifetime journey of crafting, but the proliferation of different kinds of kits suggests that rather than moving vertically to more skilled or creative expressions of the same craft, hobbyists moved laterally to an equally elementary kit in another medium or simply to another object in the same medium." (p. 262)

The step-by-step and packaged nature of kits may restrict the amount of flexibility and creative expression involved in their construction. While other forms of DIY (like instructional articles) may have involved similarly constrained processes or simple projects, kits also limit the need for effort in other portions of the process. As Gelber puts it (again in reference to non-electronic kits):

> "The package meant that the hobbyist did not have to engage the hobby at a higher level of abstraction. Nonkit crafters needed to think about what sort of craft they wanted to do, what projects they should pursue, what materials they needed to do it, and what tools were called for. Kit assemblers needed only to buy the box. There were no preliminary steps, no planning or organizing, no thinking about the process. In other words, the hobbyists did not have to engage the craft intellectually." (p. 262)

Haring [2007] describes similar limitations in electronics. For example, she writes of a warning from Heath to their customers: "In the majority of cases, failure to observe basic instruction fundamentals is responsible for failure to achieve desired level of performance." Their instruction manuals included a location to check

[7] Fisher, "Plug Is Pulled on Heathkits"

[8] Ibid.

off the completion of each step. Still, she describes amateur radio enthusiasts as finding values in kit assembly not present in buying pre-assembled products. Kits provided an opportunity to practice patience and good workmanship, and to engage, in some way, in the process of making a product. Digital fabrication provides new possibilities for individuals to go beyond assembling a kit designed by someone else and instead design and make their own device (or to create a personalized version of an existing design).

Despite their popularity, Heathkits didn't appeal to everyone. More than 95% of Heathkit builders were men and the average customer had at least one college degree.[9] Amateur radio reflected a similar gender disparity. Haring [2007] describes a culture that "deliberately advanced a masculine identity for radio hobbyists and radio technology." Ham radio practitioners were largely male. The Northern California DX Club, for example, didn't admit its first female member until 1963, almost twenty years after its founding. Descriptions of some ham radio gatherings suggested a "freewheeling men's retreat" (Haring's words) that involved copious drinking and some recreational sexual activity. Wives of ham radio hobbyists were portrayed as obstacles to the practice. The few women who did engage in amateur radio tended to de-emphasize their technical proficiency as a means of accommodating to the male-dominated culture. Gender and economic diversity continues to be a challenge in today's maker movement.

The rise of more sophisticated electronic products helped to bring about the end of the kit era. Heath stopped selling kits in 1992 but the problems started earlier. "Heath's kit sales have steadily declined since 1981, victims of reduced leisure time, declining prices that make it cheaper to buy fancy radios and electronic equipment than to build them, and the seduction of technically oriented consumers by personal computers."[10] The falling cost advantages of DIY assembly resulted from increased integration of components (in the form of integrated circuits) and automated assembly processes. These advanced technologies were also key to the viability of the personal computer, the programming of which became a focus of much technical hobbyist activity in the 1980's (described, for example, in Steven Levy's *Hackers*). In 2003, well-known electronics author Forrest Mims was quoted as saying that "hobby electronics is in a free fall."[11] Since then, however, DIY technology has undergone something of a renaissance in the forms of today's "maker" movement.

[9] Ibid.

[10] Ibid.

[11] Mark David, "A Gift Of Hobby Electronics Inspires Future Engineers", Electronic Design, Dec. 18, 2003.

Today's Maker Movement

The last decade or so has seen the emergence of a new generation of hobbyist technology activity, often referred to as the "maker movement". The term was popularized by Make Magazine, which launched in 2005, and the Maker Faires, in-person hobbyist gatherings that began in 2006.[12] Make Magazine and the Maker Faires (both initiatives of Maker Media) cover a wide range of hobbyist activities with a general focus on technology. Make Magazine has a paid circulation of 125,000[13] and Maker Faire attendance totaled over three-quarters of a million people in 2014.[14] Maker Faires have been held around the world, including at the White House. The maker movement is often described as a broad wave of people creating things with new technologies — the White House Maker Faire fact sheet, for example, talks about access to fabrication software and tools "enabling more Americans to design and build almost anything".[15] The maker movement, however, can also be seen as a brand [Bean & Rosner 2014]. In fact, Maker Media explicitly takes credit for naming the movement, including in their historical timeline a 2005 event: "[Maker Media founder, Dale] Dougherty coins the terms: 'Makers' and 'Maker Movement'".[16] Despite this confusion about the exact definition of today's maker movement, it's possible to point to a number of specific new technology platforms and resources that have emerged over the past decade or so.

One key domain for today's technology hobbyists is physical computing — microcontrollers capable of interfacing with sensors, actuators, and a variety of other electronic components. While people have been experimenting with electronics for decades, some significant new platforms and resources have started in the last ten or so years. (The history of electronics toolkits is discussed in more detail in the next chapter.) One popular platform is Arduino, which includes hardware and software for programming microcontrollers. I helped co-found the Arduino platform as a master's student at the Interaction Design Institute Ivrea and it has since spread around the world. As one article put it, "Arduino has spawned an international

[12] http://makermedia.com

[13] http://makermedia.com/wp-content/uploads/2013/01/Make-and-Makezine-rates.pdf

[14] http://makermedia.com/brands/maker-faire/

[15] https://www.whitehouse.gov/nation-of-makers

[16] http://makermedia.com/press/fact-sheet/

do-it-yourself revolution in electronics."[17] Two popular hobbyist electronics distributors, SparkFun Electronics and Adafruit Industries, began around the same time — SparkFun in 2003 and Adafruit in 2005. These sites offer custom electronic modules alongside third-party components and tools, along with tutorials, forums, and other online resources. SparkFun, for instance, offers around 500 custom products together with about 2500 third-party parts. Both SparkFun and Adafruit have reached annual revenues of over $30 million.[18] In 2011, the Raspberry Pi, an inexpensive single-board computer went into mass production; over five million have been sold since.[19] A host of other platforms provide access to a whole range of physical computing technologies.

In contrast with many previous electronics platforms, both the Arduino hardware and many of the products from SparkFun and Adafruit are open-source, with the design files shared for others to study or modify. This corresponds with a general increase in the availability of the means for individuals to create their own printed circuit boards (PCBs) from digital design files. In a 1999 article on PCB design in *Popular Electronics,* for instance, the author noted:

> "In the past ten years, the software packages used to lay out printed circuit boards (PCBs) have dramatically come down in price, while their usefulness has skyrocketed. Because of that, very few hobbyists design boards by hand any more. They've made the transition from designing boards using rub-on transfers on sheets of Mylar to designing with mouse clicks on a computer screen."[20]

The instructions given in the article for making PCBs involved a photographic transfer process and chemical etching. Soon, though, other options were available to the individual hobbyists. PCB manufacturer Advanced Circuits offered on-demand manufacturing of PCBs for $33 each starting in 2001.[21] (The special currently requires a minimum order of four boards.) By 2006, SparkFun sister company BatchPCB was pooling orders to offer cheaper prices to

[17] David Kushner, "The Making of Arduino: How five friends engineered a small circuit board that's taking the DIY world by storm", *IEEE Spectrum* online, 26 Oct 2011. http://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino

[18] SparkFun: http://www.fastcompany.com/3044271/tech-forecast/radioshack-for-the-maker-crowd
Adafruit: https://www.facebook.com/ServInt/posts/10153172594036177

[19] https://www.raspberrypi.org/five-million-sold/

[20] James Edwards, "Designing and Building Printed-Circuit Boards" in *Popular Electronics,* October 1999.

[21] http://www.4pcb.com/advanced-circuits-history/

individual customers[22] and by 2009, China-based Gold Phoenix would make 155 square inches worth of PCB for $110.[23]

Hobbyist access to other forms of digital fabrication has expanded over a similar period. (See [Mota 2011] for an overview.) The RepRap project (short for "*rep*licating *rap*id-prototyper"), consists of 3D-printers capable of making many of the parts needed to produce more 3D-printers. It first appeared online in 2004 and gained widespread notice in 2005.[24] The RepRap community has since created and shared many 3D-printer designs. RepRap helped give rise to MakerBot, a commercial manufacturer of hobbyists 3D-printers that released its first machine in 2009.[25] In 2013, Stratasys, an established maker of commercial 3D-printers, acquired MakerBot for approximately $400 million.[26] The past few years have seen the launch of numerous other hobbyist 3D-printers — a review guide published by Make Magazine in 2015 included 26 different models. Ponoko and Shapeways, two services that offer on-demand digital fabrication, launched in 2007. They provide access to a range of 3D-printing and laser-cutting services in a variety of materials.

The increasing availability of digital fabrication processes has gone along with a rise in CAD software targeted at individual makers and hobbyists. In 2006, Google released a free version of SketchUp, a 3D modeling program, shortly after acquiring its creator, @Last software.[27] Tinkercad, a free browser-based 3D modeling tool, was launched in 2011 and acquired by CAD giant Autodesk in 2013.[28] Autodesk launched their own free 3D CAD tool, 123D Design, in 2012.[29] OpenSCAD is an open-source editor for creating 3D models with textual scripts; development started in 2009.[30] These tools mean that the individual hobbyist has access not only to the technology needed to fabricate an existing design but also the tools needed to create or modify a design for themselves.

[22] "BatchPCB downtime", https://forum.sparkfun.com/viewtopic.php?f=16&t=5486&p=23200

[23] http://hackaday.com/2009/01/15/how-to-prepare-your-eagle-designs-for-manufacture/

[24] http://reprap.org/wiki/About

[25] CupCake CNC by MakerBot, http://www.thingiverse.com/thing:457

[26] http://www.businesswire.com/news/home/20130619006431/en/Stratasys-Acquire-MakerBot-Merging-Global-3D-Printing

[27] http://googleblog.blogspot.com/2006/04/great-day-for-3d.html

[28] https://www.tinkercad.com/about/

[29] http://www.core77.com/posts/23824/Autodesk-Releases-123D-Design-Modeling-Software-Gratis

[30] https://github.com/openscad/openscad/commit/9da231a4ece4b125c77ba0f6968c68c89a461d1a

The internet further enables the aggregation of information and designs from a wide range of individuals. Instructables, a website that hosts user-written tutorials on a wide range of topics, was started in 2005 and now has more than 100,000 tutorials.[31] Thingiverse, a site created by MakerBot, hosts user-created 3D models that can be printed on a MakerBot 3D-printer or made with other fabrication machines. It launched in 2008[32] and now hosts over 100,000 things.[33] The existence of online media makes it difficult to compare the audience of the maker movement with that of more traditional publications. For instance, while *Make Magazine* has only an eighth of the paid circulation of *Family Handyman* (a DIY home improvement magazine) and a third of that of the former *Popular Electronics*, its website has 2.1 million unique monthly visitors, more than 15 times the subscriber base of its print magazine.

It's instructive to compare the maker movement with other hobbies and DIY practices. *Forbes* lists the revenues of arts and crafts chain Michaels as $4.4 billion (as of December 2013)[34] and of Jo-Ann Fabric and Craft Stores as $2.3 billion (as of October 2014)[35] compared with a little over $30 million for Adafruit and SparkFun. These low-tech DIY and craft practices may not generate the same media attention as the "maker movement" but they serve a large and established audience. Today's hobby electronics companies don't match the revenues of Heath at its peak either.

Issues of gender, class, and race continue to trouble DIY technology communities today. Four-fifths of Make Magazine's readership is male and a similar percentage have post-graduate educations.[36] Their median household income is $106,000, about double that of the U.S. as a whole. The statistics for Maker Faire attendees are similar; Maker Media describes them as "well-educated" and "affluent".[37] Make has been criticized for this lack of diversity, e.g. in a talk given by Leah Buechley at the Eyeo festival in 2014.[38]

Today's maker tools and technologies enable a wide range of experimentation and creation. There are, however, few examples of

[31] http://www.instructables.com/about/

[32] http://geekdad.com/2008/11/thingiversecom/

[33] http://www.thingiverse.com/about

[34] http://www.forbes.com/companies/michaels-stores/

[35] http://www.forbes.com/companies/jo-ann-stores/

[36] http://makermedia.com/press/fact-sheet/

[37] http://makermedia.com/wp-content/uploads/2013/01/MFBA-2014-research-deck_FINAL.pdf

[38] https://vimeo.com/110616469

kits for the self-assembly of commercial-like products of the kind prevalent in the 1950s and 60s. Both Adafruit and Sparkfun have a "kit" category in their product listings but it's only one among many — and, furthermore, many of the kits listed are collections of parts for experimenting with technology ("invention kits") rather than means of assembling specific products. Overall, today's maker movement seems to place more emphasis on creativity and self-expression and less on cost-savings and the practical uses of the objects made. This dissertation explores the possibility of harnessing today's maker technologies for the making of finished products. It asks whether its possible to combine the creativity and self-expression enabled by open-ended technologies with the practicality and usefulness of robust, functional devices.

3. Related Work

There are a number of strands of HCI research that explore the relationship between technology and making. This chapter starts by briefly describing some foundational work in engaging new audiences (particularly children) in computational making in the domain of software. It then discusses efforts to engage people in two areas closely related to the topic of this dissertation — electronics and digital fabrication — as well as the combination of the two. Finally, it summarizes work that takes a broader perspective on the role of technology in making.

Computational Making

Much of the foundational research that has inspired and grounded this dissertation explores software and programming as a medium for creativity and learning. While this chapter primarily focuses on research in the directly-related domains of electronics and digital fabrication, here I discuss a few references in the general area of computational making and learning.

One of the most influential people in the area of computers and their relationship with creativity and learning is Seymour Papert. In his book *Mindstorms: Children, Computers, and Powerful Ideas* [1980], Papert discusses ways in which computers can allow children to engage and experiment with powerful ideas in mathematics and other domains. He developed the theory of constructionism, the idea that people both construct knowledge for themselves (the constructivist philosophy) and that constructing public entities, including physical objects, is a powerful means of doing so [Papert & Harel 1991]. This dissertation is inspired by Papert's emphasis on finding forms of technology that provide both a personal connection to the learner and access to deeper ideas of science and engineering.

Traditional approaches to technology (including programming and electronics) often emphasize certain styles of learning or working that may not appeal to everyone. Sherry Turkle and Seymour Papert [1991] argue for epistemological pluralism, or a respect for multiple ways of knowing. In particular, they emphasize the importance of recognizing and respecting "soft" forms of knowledge that may not align with the abstract and hierarchical approaches commonly associated with engineering. Turkle and Papert draw on Levi-Strauss's [1966] notion of bricolage, or the "science of the concrete", discussing it as a contrast with top-down planning:

> "The bricoleur scientist does not move abstractly and hierarchically from axiom to theorem to corollary. Bricoleurs construct theories by arranging and rearranging, by

> negotiating and renegotiating with a set of well-known materials." [Turkle & Papert 1991]

This approach resonates with the findings of this dissertation, which emphasize hands-on knowledge of electronic components and fabrication processes over abstract theories of electronics.

More recently, several research papers (e.g. [Brennan & Resnick 2012], [Qiu et al. 2013], and [Kafai et al. 2014]) have explored the specific computational concepts, practices, and perspectives that people can develop in the course of learning about programming and technology. Inspired by this work, this dissertation investigates the additions and changes to these frameworks that emerge from engaging with personal fabrication and DIY electronics.

Engaging People in Electronics

Engaging people in making and learning with electronics involves paying attention both to the form of the technology itself and thc kinds of activities and contexts in which it occurs. This section starts by discussing some of the technological tools that researchers and companies have built to engage people with electronics, specifically toolkits for children and those for designers and hobbyists. It then discusses work that combines electronics with craft, a process that involves changes to both the technologies themselves and the activities to which they're applied. This is followed by a discussion of work that specifically looks at the way that electronics and activities can work together to engage new audiences. Finally, it discusses research that explores critical approaches to electronics making.

Toolkits for Children

Electronics toolkits have a long history, both within and outside of the HCI community. These toolkits encapsulate electronics functionality into higher-level physical building blocks or modules that can be combined and re-combined to build a variety of interactive prototypes. In this sense, electronics toolkits can be seen as a special case of the notion of a toolkit as discussed in [von Hippel & Katz, 2002]: a method for transferring expertise from the maker of a technology to its users. Many early electronics toolkits were targeted specifically at children and explored the combination of making and learning.

Inspired by Seymour Papert and his vision of the potential of computation as a medium for learning [Papert 1980], Mitchel Resnick and collaborators have developed a series of projects to introduce programming and engineering to children. Resnick [1993] discusses three such projects: LEGO/Logo, electronics bricks, and programmable bricks. These combine the ability to construct physical structures using LEGO bricks with various techniques for specifying interactive behaviors. Resnick et al. [1998] discuss four different domains of "physical manipulatives": blocks, beads, balls, and badges. In each domain, the development of new technology has been accompanied by a consideration for how it might connect to children's interests and activities. Building on this work, the commercial LEGO Mindstorms platform was first released in 1998. It integrates sensors, motors, and a programmable module with the LEGO construction system. LEGO has since developed a variety of electronics platforms, including the WeDo, which targets younger children. More recent is the littleBits platform, which consists of a set of modules that connect to each other with magnets, enabling simple and quick exploration for even very young children.

Some of these toolkits for children use simple textual programming languages. LEGO/Logo for example, allows children to write textual Logo programs which execute on the computer and control tethered electronic components. The subsequent "programmable brick" interprets similar programs independent of a computer [Martin & Resnick 1993]. Other toolkits use visual programming environments, with graphical blocks that can be connected together to generate a program's logic. With most littleBits modules, no programming is required; instead, a project's behavior derives from the physical arrangement of the modules.

Other publications distill lessons from the development of these toolkits and their use with children. Resnick et al. [1996] advocate for "pianos not stereos" — i.e. technology that enables children to be designers and to learn through the process of making rather than simply being users or consumers of technology. Resnick and Silverman [2005] discuss ten guiding principles for designing construction kits for kids. These principles have been an important reference as I conduct my own work.

Toolkits for Designers and Hobbyists

Other toolkits have focused more specifically on designers. An early project, MetaCricket [Martin et al. 2000] was initially intended for children but was also adopted by professional designers. Other toolkits for designers were inspired by toolkits for the creation of graphical user interfaces (GUIs), as discussed, for example, in [Meyers et al, 2000]. These include Phidgets [Greenberg & Fitchett, 2001], iStuff [Ballagas et al. 2003], and Calder [Lee et al. 2004]. (The Phidgets research project spun out into a commercial platform in the early 2000's.) Lee et al. discuss two qualities of particular importance for these toolkits: (1) allowing interactivity and form to be prototyped together and (2) preserving fluidity and flexibility. Some platforms, like d.tools [Hartmann et al. 2006], combine electronics modules with tools to help in the creation of on-screen interfaces. Other toolkits are targeted at more specific audiences. For example, Voodoo IO [Villar et al. 2006] facilitates the creation of custom video game controllers.

Phidgets and Calder provide a means for the toolkit to communicate with a computer along with libraries for programming them using traditional (computer-based) programming languages and development environments. "This allows us to reuse rather than reinvent the extensive programming and debugging environments that have been developed for GUI implementation." [Lee et al. 2004] On the other hand, this does little to simplify the process for people without prior programming experience and requires the toolkit be tethered to a computer.

A more recent toolkit, .NET Gadgeteer [Villar et al. 2012], contains more sophisticated modules, including cameras and graphical displays. It has also been translated into a commercial product. The .NET Gadgeteer toolkit works with existing programming languages (e.g. C#) and development environments (Visual Studio) but it generates byte code that is interpreted on the device itself. This eliminates the need for tethering but, again, provides little simplification of the programming language and environment.

These toolkits provide custom high-level modules for many components, including sensors and actuators. Blikstein [2013] calls these the "Cricket" model (based on the toolkit that inspired the LEGO Mindstorms). Other toolkits provide a high-level module only for the main computational element (microcontroller) and connect to a variety of off-the-shelf sensors and actuators. Blikstein calls this the "breakout" model. Sadler et al. [2015] elaborate, first discussing systems using the breakout model:

> "By staying at the circuit level, prototypers are not limited by a ceiling of fixed-functionality black boxes. This makes the modifiability of the system high, and it is possible to add and remove components at a raw circuit level.... However, encouraging manual circuit creation comes at a cost of a higher technical barrier to entry. On the other side of the spectrum are more modular toolkits that give novices a set of pre-fabricated, and functionally rigid, building blocks.... While a building-block approach may reduce technical barriers to entry, it comes at a cost of making it harder to modify the system."

A popular early platform on the breakout model was the BASIC Stamp from Parallax, which became popular in the early 1990's. It combines microcontroller breakout boards (based on PIC or other processors) with a BASIC interpreter. Like .NET Gadgeteer, the BASIC Stamp uses a virtual machine running on the microcontroller, programmed in a variant of the BASIC programming language using a free editor provided by Parallax. The choice of BASIC is an attempt to simplify the programming by using a language targeted at educational contexts.

Another popular platform along these lines is Arduino, discussed in the previous chapter (in the section on "Today's Maker Movement"). Arduino builds on the Wiring platform by Hernando Barragan [Barragan 2004] and Processing by Ben Fry and Casey Reas. Arduino includes microcontroller breakout boards (for the AVR family of 8-bit processors and other microcontrollers) along with a software development environment. Arduino and Wiring use an existing programming language (C with some C++ features) and compiler (gcc) to generate native microcontroller code. On top of this, they provide custom libraries and some minor pre-processing of user code to simplify the programming required for common tasks. They also offer custom development environments that encapsulate the process of building code and uploading it to the hardware. This approach is inspired by Processing, which similarly provides custom libraries, pre-processing, and a development environment which runs on top of a standard programming platform (Java). Arduino, Wiring, and Processing are all open-source, allowing third-parties to modify and extend the software libraries and development environment as required. The firmware that runs on the products described in this dissertation use the Arduino software and development environment.

As mentioned in Chapter 2, a whole host of embedded platforms for hobbyists have been released recently, including the popular Raspberry Pi single board Linux computer.

The personal fabrication approach pursued in this dissertation can be seen as an attempt to take the modifiability of systems like Arduino and Wiring to its logical extreme, eliminating even the single high-level physical module found in those platforms.

Electronics and Craft

There a number of different strategies for constructing physical prototypes around an electronics toolkit. Hodges et al. [2013] discuss eight of these, including the use of existing construction systems (like LEGO or metal sheets or extrusion); constructions with paper, tape, glue, etc.; laser-cutting or 3D-printing enclosures; and retrofitting existing devices. Some toolkits are designed to be easily and reversibly mounted onto substrates like foam [Avrahami & Hudson 2002] or thick fabric [Villar et al. 2006] to allow for quick construction of custom interface layouts.

One domain that's received in-depth investigation is the combination of electronics and craft materials like fabric or paper. These combinations can be a valuable means of engaging new audiences in electronics making, particularly those that may not be interested in traditional forms of technology. This area has been notably explored by Leah Buechley and her students. Through the development of toolkits, examples, documentation, workshops, and other resources, this work attempts to expand the culture of technological production, involving new people and new activities in electronics making [Buechley & Perner-Wilson, 2012]. This research includes the LilyPad Arduino toolkit [Buechley 2010], techniques for constructing textile sensors [Perner-Wilson et al. 2011], and work on paper circuits ([Qi & Buechley 2014] and [Mellis et al. 2013]). Many of the tools and techniques developed in these efforts have been evaluated through the use of workshops. Reflecting on the experience and comments of workshop participants allowed the researchers to gain general insights about the culture and form of technology. In this dissertation, I'm similarly interested in the ability of workshops to reveal and change people's relationships with technology. I'm also interested in the ways that placing technology in new domains can engage new people in making it, although with a focus on everyday electronic products rather than craft materials.

Activities for Engaging with Electronics

Other research has looked more generally at the combination of custom tools or toolkits with workshops or other activities as a means of introducing new audiences to working with electronics. Moriwaki and Brucker-Cohen [2006] describe the "scrapyard challenge", a series of workshops using found materials with simple circuits to introduce novices to electronics. Silver [2009] discusses the relationship between a tool, Drawdio, and the activities and mindset that surround it. Drawdio is a circuit that uses a 555 timer chip to create musical tones based on the resistance of everyday objects. Silver describes multiple workshops and activities he facilitated using the Drawdio and uses them to extract principles for a "maker methodology". Makey Makey [Rosenbaum 2015] similarly combines a specific technological object with a set of examples and activities that suggest ways it might be used. In this case, the tool allows arbitrary conductive materials to be used as switches to generate key-presses on a computer (as if coming from a keyboard). Or, as the Makey Makey website puts it: "Ever played Mario on Play-Doh or Piano on Bananas? Alligator clip the Internet to Your World." For both Drawdio and Makey Makey, videos of example projects were an important method for suggesting possible uses of the devices.

In my work, I've attempted a combination of product development and activity facilitation inspired by the projects discussed here. By curating both the digital and physical systems participants work with and the activities they conduct, I try to provide meaningful creative and learning experiences.

Critical Engagement with Electronics

A few researchers have looked at electronics as an element of activism or critical inquiry. Kuznetsov et al. [2011] conducted workshops in which participants constructed air quality-sensing balloons as means of generating public awareness and discussion of environmental issues. Ratto [2011] proposes critical making as an approach for tying hands-on activities to reflection on broader issues about technology and society. In my work, I also hope to use making as a means of helping people to engage in broader questions about the role of technology in society.

Interfaces for Digital Fabrication

In recent years, a variety of HCI research projects have explored new interfaces for digital fabrication, a topic summarized in a recent workshop at CHI [Mellis et al, 2013]. One approach is the creation of new 3D modeling software. For example, Oh et al. [2006] describe custom software interfaces for designing furniture and skeleton animals. SketchChair [Saul et al. 2011] provides a custom user interface for designing and simulating chairs which can be fabricated using a number of different machines. Johnson [2008] and Jacobs and Buechley [2013] present software tools to support novices in designing objects for fabrication using code, an approach similar to that found in the open-source OpenSCAD software application.

Other tools combine software with novel tangible interfaces. KidCAD [Follmer & Ishii 2012], for example, allows children to create 3D models by pressing existing objects into a gel, whose shape is reconstructed with structured light. Weichel et al. [2014] describe an augmented reality system, MixFab, that allows users to use existing physical objects and gestures as part of the modeling process. SPATA [Weichel et al. 2015] includes a digital caliper and protractor that facilitate the transfer of measurements between physical objects and digital models.

Another approach is directly augmenting digital fabrication machines to allow for tighter coupling between the design process and the resulting artifacts. Examples of this approach include CopyCAD [Follmer et al, 2010] and Constructable [Mueller et al, 2012]. Some projects go as far as creating custom fabrication machines which allow for more interactive control ([Willis et al. 2012] and [Zoran & Paradiso 2013]).

In my work, instead of building new tools and technologies, I've explored the ways in which existing ones can be applied to the creation of devices for use in everyday life. In the process, I've discovered opportunities and lessons for improving existing tools and creating new ones.

Digital Fabrication of Interactive Objects

Closest to my research (from a technical perspective at least) are the projects that combine digital fabrication with the construction of interactive objects. This work has taken a variety of forms: aesthetic exploration, building enclosures for electronic modules, integrating interactive behaviors into fabricated objects, and digital fabrication of circuit elements themselves. I discuss these in turn.

Plywood Punk [Schmitt & Seitinger 2009] and Wooden Logic [Cottam 2009] focus on the aesthetics of the combination of electronics with digitally-fabricated materials, particularly wood. I'm also interested in exploring these aesthetic dimensions, with the additional constraint of seeking to make products for use in daily life.

Some projects focus specifically on the task of making enclosures for electronic circuits. For example, .NET Gadgeteer [Villar et al. 2012] provides support for modeling enclosures in SolidWorks that fit the toolkit's modules. Enclosed [Weichel et al. 2013] facilitates the creation of laser-cut enclosure for circuits composed of .NET Gadgeteer modules. In my work, I've tended to manually design the enclosures for my products, but these projects suggest promising directions for automating at least some of this process.

Other work feature tighter integration between the digitally-fabricated forms and the interactive functionality. OriginalMachines [Schmitt 2011] describes a custom software tool for fabricating gears and other mechanisms that mate with DC motors to form custom servo mechanisms. Printed optics [Willis et al. 2012] is a set of techniques for guiding light using 3D-printed structures and using them in a variety of interaction modalities. Sauron [Savage et al. 2013] supports the design and fabrication of physical input devices whose use can be sensed by a camera mounted on the object, in place of electronic circuits or components. In my work, I've maintained a more traditional separation between electronics and fabricated forms, but these research projects suggest intriguing opportunities for future investigation.

Finally, some projects have investigated the digital fabrication of electronic circuits themselves — typically conductive traces to which traditional electronic components will be attached. In some cases, this involves the use of traditional printed circuit board fabrication. For example, Martin et al. [2000] discuss custom PCBs as a means of integrating and reproducing prototypes that were initially composed of multiple toolkit modules. Fritzing [Knörig et al. 2009] is a software tool that seeks to make PCB design more accessible to novices by enabling them to start by designing their circuit in an interface that looks a breadboard. In general, though, PCB design and fabrication is often seen solely as the domain of experts, with electronics toolkits providing the novice-friendly alternative. This dissertation challenges that perspective, exploring the extent to which PCB fabrication itself can be made accessible to novices.

More recent work has explored alternative techniques for constructing circuits. Midas [Savage et al. 2012] provides a software tool for designing, fabricating, and interacting with capacitive sensors made of copper cut with a vinyl cutter. Kawahara et al. [2013] explore the use of ink-jet printed silver as a quick and easy means of circuit production. In my work, I've stuck with PCBs because they seemed best suited to the making of consumer products for extended use. That said, I think it would be fascinating to explore the new product categories and constraints created by ink-jet printed or other thin and flexible circuits.

Technology and Craft / DIY

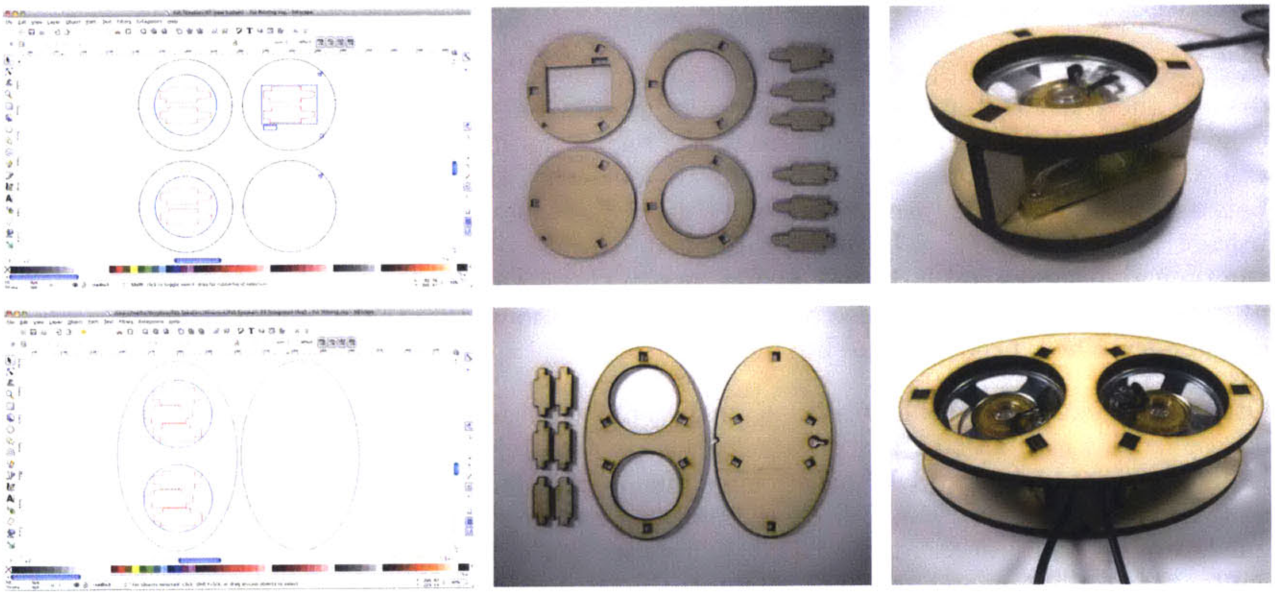
Another strand of HCI research has looked at the effect of technology on DIY and craft practices, both traditional and technological. Buechley et al. [2009] provides an overview of the subject. Kuznetsov and Paulos [2010] surveyed contributors to online DIY communities. They discuss respondents' motivations for engaging in DIY activities and for participating in online DIY communities. Torrey et al. [2009] studied the ways that people search for craft knowledge online, exploring the relationship between digital information and embodied practices.

Tanenbaum et al. [2013] examine maker culture more broadly. They discuss the relationship between practical and personal motivations, echoing some of the themes discussed in the background chapter (in the "DIY and Hobbies" section). They also discuss the relationship between individual DIY practices and industrial products and infrastructures. Sun et al. [2015] highlight an example from outside the typical discourse (elderly radio tinkerers in China). Lindtner et al. [2014] discuss maker culture as a source of HCI innovation that can generate new products and businesses.

Other work (e.g. [Maestri & Wakkary, 2011]) has focused specifically on repair as a form of DIY and creativity. Other researchers have looked at technology and DIY practice within specific domains, such as furniture hacking ([Rosner & Bean 2009]) or knitting and gardening ([Goodman & Rosner 2011]). This research provides a context for reflecting on the implications of the case studies composing the dissertation.

4. Personal Fabrication of Electronic Products

This chapter outlines a framework for designing and making electronic products using personal fabrication — that is, an individual combining low-volume PCB fabrication, off-the-shelf electronic components, digitally-fabricated enclosures, and embedded software to form electronic devices for use in daily life. As mentioned in the introduction, this allows for a precision and optimization that's difficult to achieve with manual methods while simultaneously providing more flexibility than mass production. The use of digital fabrication enables a unique combination of reproducibility and customization — multiple instances of a product can be made from its digital design files, but the design can be modified for each instance. (As discussed in the background chapter, "Qualities of Craft, Mass Production, and Digital Fabrication" section, this type of variability differs from that found in craft or mass production.) An individual can start from an existing design, modify only the portions they want to change, and then fabricate a custom variation. This ability to build on previous designs facilitates iteration throughout a design process, personalization of products to suit individuals' needs and tastes, and the integration of the work of multiple people in open-source hardware or other collaborative development processes.



A personal fabrication approach enables the customization of a digital design file (left) and the subsequent fabrication (middle) and assembly (right) of a custom product.

This chapter compares personal fabrication with other approaches to making electronic products, details the elements of such products, discusses some case studies of personally-fabricated electronic

	Personal Fabrication	**Electronics Prototyping**	**Electronic Kits**	**Digital Design for Mass Production**	**Hardware Startups**
Degree of Individual Control	High	High	Low	Low	Medium
Reproducibility	High	Low	Medium	High	Medium
Robustness	High	Low	High	High	High

Contrasting personal fabrication with other approaches to making electronic products.

products, and then provides some general principles for making devices in this way. The practical framework presented in this chapter formed the basis for the design and fabrication of the other electronic products described in this dissertation and served as a starting point for investigating its broader research questions.

Comparison with Other Approaches to Making Electronics

To clarify what I mean by a personal fabrication approach to making electronics, I compare it with four other popular approaches: electronics prototyping (with breadboards and toolkits), mass-produced electronic kits, digital design for mass production, and hardware startups.

vs. Electronics Prototyping

Perhaps the most significant difference between personal fabrication and other kinds of electronics prototyping (like breadboards and the electronics toolkits discussed in Chapter 3) is the extent to which the former can yield artifacts which are optimized and robust enough for extended use. The combination of PCB plus soldered components in particular is a robust construction which is becoming increasingly accessible to individuals (discussed in more detail in “Elements of Fabricated Products” below). Conversely, laser-cutting, 3D-printing, CNC milling and other techniques typically considered as prototyping processes are finding increasing applicability for the manufacture of final products. Breadboarded circuits and those built from electronics toolkits, in contrast, are often not designed for extended use. The overhead of the breadboard or the individual toolkit modules can make it difficult to optimize the size or performance of a device. The ease with which they can be connected and disconnected can make them less likely to achieve the robustness required for extended use (but does provide a significant advantage in tinker-ability, discussed below).

In comparison with prototyping tools like breadboards and electronics toolkits, a personal fabrication approach also yields a digital design file that can be directly shared and reproduced. While it's possible to document other physical prototypes in ways that make it possible for others to re-create them, doing so is a relatively labor-intensive and error-prone process. For example, in re-creating a circuit on a breadboard, one needs to manually make all the connections between the individual components. When assembling a fabricated circuit board from an existing design, on the other hand, the connections between components are embedded in the PCB itself, requiring only that one connect each component to a given location on the board. This changes the assembly and debugging process from a global one, in which one needs to create and verify the entire circuit, to a local one, in which one only needs to successfully connect each component to the PCB (which then provides the desired connections between components).

Another advantage that personal fabrication has over other kinds of electronics prototyping is the ability to work with a broader range of components. Electronics toolkits typically encapsulate a fixed set of electronic components. Breadboards generally work with through-hole parts with 0.1" pitch, a form that's increasingly being replaced by smaller surface-mount components. Custom PCBs can provide footprints for a wide range of through-hole and surface mount parts, allowing for the use of components beyond those contained in any particular toolkit or those available in breadboard-compatible forms (or on breadboard-compatible breakout boards). This increases the range of components that can be used and reduces cost and size.

One notable disadvantage of personal fabrication versus other methods of electronics prototyping is a decrease in tinker-ability. When working with breadboard circuits or electronics toolkits, it's typically quick and easy to connect and disconnect various components, rapidly exploring a number of different circuit configurations. This can facilitate the process of figuring out how to work with an individual component as well as the process of evaluating multiple components to determine their suitability for a given task (e.g. deciding which sensor can most reliably detect a particular physical phenomenon). To some extent, this decrease in tinker-ability is mitigated by the ability to build on existing digital circuit designs whose correctness and applicability to a given domain have already been proven. Often, however, it is helpful to try out a new component or circuit on a breadboard before incorporating it into the digital design for a larger fabricated circuit.

vs. Electronic Kits

Mass-manufactured electronic kits, like the Heathkits of the post-World War II years (discussed in Chapter 2), can yield products similar to those created by a personal fabrication approach. Both include electronic components hand-soldered to a manufactured PCB. The main difference is the process by which the PCB (and, therefore, the product) is designed and produced. The boards in these traditional kits are mass manufactured, giving the individual little or no control over the design of the resulting product. Furthermore, the kits may include other mass-produced elements, like enclosures, that would also be difficult for an individual to customize. A fabricated product, in contrast, is produced in small quantities from a digital design file, giving individuals the opportunity to create or modify the product for themselves. That allows for a level of personalization and an ease of iteration that would be difficult with a traditional mass-manufactured electronics kit.

vs. Digital Design for Mass Production

Today, many mass-produced products are designed using CAD software and these digital files are then translated into the corresponding physical product. Often, however, this process involves lengthy tooling or setup phases, like the milling of molds or the configuration of production lines. In discussing a personal fabrication approach to the making of electronic products, I'm specifically focusing on processes that allow for the relatively efficient creation of individual products — requiring the use of manufacturing processes without involved or costly setup procedures. For volume production, this is likely to be less efficient than more setup-intensive methods.

vs. Hardware Startups

Individuals seeking to make electronic products have the option of going into business — raising money, manufacturing a product, and selling it to customers. This hardware startup approach seems to be growing in popularity and feasibility. There are new startup incubators, like Hax (formerly HAXLR8R), Bolt, and Highway1, focused specifically on hardware companies. Chinese manufacturers seem increasingly open to working with startups with small production runs. Kickstarter makes it possible to raise money from a large group of potential customers.

This model has some qualities in common with a personal fabrication approach. Both allow small groups of people to create and distribute new electronic products. But the hardware startup route tends towards a more traditional structure, in which a company designs, produces, and markets many identical copies of a product, which is bought and used by customers. Personal fabrication, in contrast, suggests a different means of scaling, one in which products are designed and made by many different people, including the ones who will be using the device. It involves sharing products by publishing their design files (rather than shipping the physical objects), which enables evolution and personalization of the device.

Elements of Fabricated Electronic Products

Making an electronic product involves a combination of fabricated materials and forms, electronic components, and software. Here, I discuss the considerations involved in effectively using these technologies in the personal fabrication of electronic products.

Digital Fabrication Processes and Materials

Digital fabrication machines translate digital designs into physical objects. As discussed in the introduction (Chapter 1), many of these machines are (or are becoming) accessible to individuals, whether through the existence of low-cost models that people can purchase directly, through shared facilities that provide access to more expensive machines, or through online services that fabricate parts on-demand. While these methods of access are very different, all allow an individual to fabricate individual parts from digital design files.

Making effective use of digital fabrication to create a product from its design files requires more than just access to those files. Many aspects of the resulting parts or product may not be specified in the digital design but may be essential to ensuring an appropriate result. Effective application of a personal fabrication approach requires both appropriate design files and an understanding of these additional factors. Here, I discuss various popular fabrication processes with an emphasis on the considerations involved in using them to produce a satisfactory physical artifact from a digital design. I then discuss some general criteria relevant to selecting appropriate fabrication processes.

Fabrication Processes

3D Printing. The purest of digital fabrication processes are the various forms of 3D printing. These turn digital designs into physical objects by gradually adding material in the desired locations, allowing for a wide range of possible geometries. The term "3D printing" encompasses a broad range of machines, from personal plastic printers costing a few hundred dollars to industrial machines that sinter metal and cost hundreds of thousands of dollars. Different machines work with different materials and offer different resolutions and tolerances. The materials may have different strengths, optical properties, appearances, finishing possibilities, and so on. Depending on the object being fabricated, some or all of these characteristics may be crucial to creating a usable result. In designing and sharing objects for 3D printing, therefore, it's important to specify not just their geometries, but also the required tolerances, materials, and other characteristics — most of which are less easily captured in digital form. In addition, many 3D-printing processes need some form of manual post-processing, such as removal of support material, finishing, or curing. These require an operator with appropriate knowledge and skill — and can create variations from one print to the next, even with the same file and machine. Finally, 3D-printing technology is evolving and diversifying rapidly. For all these reasons, it's important not to think of 3D printing as a way to automatically create things from information, but rather as a process with specific material qualities and affordances.

3D-printing is perhaps the most promising future direction for fabricated enclosures (a subject returned to in the "Future Work" section of Chapter 8). It allows for a wide range of geometries and, as a result, often eliminates the need for most manual assembly. This places great importance on the digital design process and the software with which it's carried out. In my experience, designing 3D models for printing requires significantly more CAD skill than 2D or 2.5D design processes. On the other hand, the process seems particularly well-suited to advances in software and one can imagine a wide range of different 3D modeling tools for different kinds of design. The flexibility of 3D printing makes it feasible to fabricate the geometries resulting from this potential diversity of design tools. The many variations and rapid advance of 3D-printing technology promise a wide range of material properties. All of which makes 3D-printing the most software-driven of the fabrication processes considered in this dissertation.

Similarly, 3D-printing is the fabrication process that comes closest to allowing for the creation of parts similar to those made with mass-production (e.g. injection molding). This is, of course, what makes it so valuable as a means of prototyping parts that will be mass produced. On the other hand, today's 3D printers, particularly those accessible to hobbyists, can't match the surface finish or many of the other material properties of injection molded parts. As a result, 3D-printed enclosures can end up looking like inferior versions of mass-produced parts, rather than having a specific aesthetic of their own. Certainly, there are beautiful, unique, and complex forms made with 3D printing but the skill required makes this quality difficult for novices to achieve.

Milling and Cutting. Other fabrication processes work by cutting or removing pieces of a larger stock material. Laser cutters cut 2D shapes out of plywood, cardboard, acrylic, and other flat materials. Vinyl cutters do the same, but with a knife that cuts through thin materials like paper or adhesive-backed vinyl. The water-jet cutter handles stronger and thicker materials like wood, metal, and glass, cutting with a stream of hard particles in a powerful jet of water. CNC (computer-numeric control) machines, like mills or routers, work in three (or more) dimensions, removing material from solid blocks of stock with a variety of cutting bits. They are often capable of very precise operations, albeit only within specific axes of movement. Compared with 3D printers, these cutting and milling tools have the advantage of being able to work with a variety of existing materials, including natural ones with complex structures that are difficult or impossible to replicate with the homogenous stock of most 3D printers. They are more limited in the geometries they can produce, however, and often require more steps in fabricating or assembling the parts.

In addition to specifying the geometry of a design itself, it's important to be explicit about the nature of the stock material and the characteristics of the cutting process. Whether two parts press-fit tightly together, slip past each other, or don't fit at all depends as much on the precise thickness of the stock (which can vary even across nominally equivalent materials) and the thickness of the cut as on the shape in the file. Some constructions may be infeasible to achieve given the tolerances of a particular machine. (For example, laser cutters may yield slightly different cut thicknesses on different sides of their working area; water-jet cutters can give rough, non-vertical edges.) Traditional engineering drawings often capture the required tolerances for various surfaces and the material to be used. A quickly created CAD file used for a prototype and then thrown up on a webpage may not. Parts might be sanded, glued, pounded together, or otherwise tweaked in ways not reflected in the design files. Generating tool paths for a CNC machine is a complex process with a significant impact on the form and finish of the resulting object; this complexity may not be possible to capture in a way that can be easily shared with others, particularly if they are using a different machine. Finishing and assembling parts created with CNC devices requires careful craft,which might be difficult to communicate or learn. All of these factors need to be kept in mind when designing or sharing a digital file for someone else to replicate.

With a CNC mill and, to a greater extent, with a laser-cutter, it's difficult to make the complex 3D geometries often found in injection molded parts. As a result of these limitations, products to be housed in laser-cut or CNC-milled enclosures need to be designed with them in mind. Available thicknesses of stock material, for example, can set important constraints on a product's design. Optimizing for efficient fabrication also yields unique geometries based, for example, on minimizing cut times. This can suggest simpler forms, a consideration reinforced by the appeal of a simple design process.

One successful strategy is the use of 2.5D designs (as illustrated by the DIY cellphone and connected devices described later in this dissertation). That is, enclosures designed as two-dimensional shapes that gain their 3D forms from the thickness of the stock materials or, on the CNC mill, the depth of cuts. These enclosures can consist of two or more parallel, flat forms stacked on top of each other, with a PCB sandwiched in the middle. Using 2.5D shapes allows for simpler software design processes. For novices, working with 2D vector drawing tools like Adobe Illustrator or Inkscape can be significantly easier than 3D-modeling. With the laser-cutter in particular, the combination of easy CAD software with a fast fabrication process allows for rapid iteration, maximizing the advantages of a personal fabrication approach. The laser-cutter also allows custom designs to be engraved onto enclosures, an easy way of personalizing a device. The assembly necessitated by laser-cut enclosures also provides an opportunity to engage people in making their own devices.

Despite their limitations, the affordances — in materials, geometries, and process — that laser-cut and CNC-milled enclosures bring to electronic products make them a good fit for today's personal fabrication possibilities. They allow for unique aesthetics, for rapid iteration, and for the involvement of individuals in the process. In turn, they require more reinvention of electronic products (from the designs used in mass production). This is the sweet spot where re-imagining the design of devices enables new relationships between people and their electronic products.

Printed Circuit Boards. The production of printed circuit boards (PCBs) can also be considered a digital fabrication process — and a relatively mature one. Digital designs are etched from copper or other materials using a photographic process, then covered with an isolating layer and text and other annotations. While the processes for creating circuit boards in this way are generally toxic and the automated systems for doing so are expensive, many services will produce PCBs on demand for individual customers with small or nonexistent minimums and standard specifications and tolerances. (As a board's specifications get more demanding, however, costs can sometimes increase dramatically.) Circuit boards can also be manually etched or milled on a CNC machine, processes that are more directly accessible to individuals but also less robust and precise. While some circuits are sensitive to the precise characteristics of circuit board's substrate or the exact tolerances of the fabrication process, a great many can be shared with relative confidence that they will work when made on a different machine from a different provider.

Other Fabrication Machines. A variety of other digital fabrication processes exist, each with its own affordances and constraints. For example, a host of machines are available for working with soft materials: CNC embroidery machines apply custom designs to fabric, knitting machines generate colors and constructions based on digital files, and Jacquard looms are possibly the oldest digital fabrication machines in existence. Industrial production uses a variety of automated machines, including robot arms and other adaptable parts of an assembly line. Furthermore, as digital fabrication becomes more established, more people are creating their own machines for custom purposes of various kinds.

Selection Criteria

In selecting a fabrication process and designing for it, there are many considerations and criteria, including material properties, repeatability, cost, accessibility, and design considerations.

Material properties. This covers a whole range of criteria, including aesthetics, durability, flexibility, optical qualities, conductivity, ease of fabrication, etc. Choice of an appropriate material depends on many factors, so it's difficult to generalize. That said, it's worth noting that digital fabrication can work well with many materials that aren't often found in electronic products, like wood, paper, fabric, and cardboard. I've had good luck with both plywood (on the laser-cutter) and solid wood (w/ CNC machines), as well as laser-cut acrylic. The set of materials available for use with 3d printers continues to expand, although at present 3d-printed parts can sometimes look like bad approximations of molded parts rather than a unique medium. 3D printing seems uniquely suited to produced geometrically-complex parts, which can be more difficult to design (see below).

Repeatability. This includes a few different characteristics that together influence the likelihood that a given fabricated part will be usable for its intended purpose. A crucial but often overlooked property is the tolerance of a given process, i.e. the margin of error on the geometry of fabricated parts. A common mistake is designing a part in such a way that it relies on a tolerance better than that achievable on a given machine. (Note it's important to consider tolerances on stock materials like plywood or acrylic as well as on the fabrication processes themselves.) Another aspect of repeatability is the likelihood that a given fabrication process will complete successfully (i.e. its yield). While many fabrication machines are reliable when well-maintained and used by a skilled operator, personal fabrication can often happen under less than these ideal circumstances. Bits can break, material can catch on fire, shatter or collapse, the machine may work intermittently, etc. This reliability is the product of a number of factors but, in general, the easier-to-use and better constructed a fabrication machine is, the more likely an individual is to end up with a part they can use. In addition, the more steps or processes required to make a product, the more likely that one of them will go wrong.

Scalability. While not directly applicable to a strict personal fabrication model, another important consideration is the extent to which a fabrication process lends itself to the production of multiple copies of the same design. Often a product will stabilize at various points in its design, at which time it might be useful to create multiple copies — e.g. to sell to others or as part of a workshop. Because of their origins as prototyping processes, many fabrication machines offer few economies of scale. For example, laser-cutting ten copies of something often takes about ten times as long as cutting one (once you've figured out the correct settings). Other processes, however, are more scalable. For example, the cost per board of making PCBs drops dramatically as you order more copies of a given design. In this sense, PCB fabrication seems like a mass-production process that has been made available for one-off production, rather than a prototyping process that is being repeated to make multiples. For some processes, there may be straightforward ways to transition from prototyping processes to larger production runs. For example, laser-cut cardboard could be replaced with a custom die capable of cutting many identical pieces. Sometimes, however, transitioning to a more scalable process may require extensive redesign of a product or a large jump in the quantity and cost of production (as in, for example, going from 3d-printed to injection-molded plastic).

Accessibility. This, again, covers multiple factors that determine the extent to which individuals are able to gain use of a given fabrication process. The cost of the machines themselves is, of course, one important factor — for example, many 3d printers are now targeted at consumers, with prices from hundreds to a few thousand dollars. (The cheapest digital fabrication machine may be traditional 2D printers, which can be used to transfer digital designs to physical objects in various ways.) The existence of online services that make fabrication processes available to individuals is also crucial. Here, considerations include not just the existence of the service itself, but the ease of using it to order individual custom parts — e.g. automated quotes and ordering, visual previews, well-specified design rules, short and reliable turnaround times, etc.

Post-processing. Parts don't necessarily emerge ready-to-use directly from a digital fabrication process. They may require finishing, coating, assembly, or other forms of post-processing. Even hand-soldering of electronic components could be seen as a form of post-processing performed on a digitally-fabricated PCB. To optimize for translation for digital design to complete product, it's important to minimize the amount of post-processing required. On the other hand, post-processing can be an opportunity to involve individuals in making a product, whether to customize aspects of the product's design, to invest the product with meaning through the individual's labor, or to provide an opportunity for learning and skill development.

Difficulty of designing for the fabrication process. Different fabrication processes require very different kinds of input, which can greatly influence the difficulty of designing for those processes. When using a 2D process like the laser-cutter, it's relatively straightforward to create design files for individual parts, although it can be more difficult to understand how flat parts will combine into a 3D form. The 3D design files required for a 3D printer, on the other hand, typically require the use of more complex 3D modeling software which can be a significant barrier to entry for novices. One advantage of these tools, however, is the ability to get a better understanding of the overall assembly and form of a product before fabricating it. CNC machines impose their own constraints. It's possible to use them with 2.5D design files that can be designed, like laser-cut parts, in easy-to-use 2D design software. Creating a design that can be successfully cut out on the machine, however, requires more understanding of the details of the process (like the cutting tools and paths involved) than is usually needed for the laser-cutter or 3D printers.

Electronic Components

In selecting electronic components for use in personally-fabricated products, the main challenge is finding parts that are optimized enough for production use but still accessible for use by an individual. This means looking for off-the-shelf components (rather than hobbyist modules) that are both accessible to individuals and reproducible when others attempt to recreate a product.

Components vs. Modules

I make a distinction between *components* — industrially-produced parts intended for use in commercial products — and hobbyist *modules* (or toolkits) that wrap these commercial parts in easier-to-use forms. While the line between the two can be ambiguous, there are multiple properties that distinguish components from higher-level modules:

- their *form factor* allows for embedding within a product (i.e. is both small enough and possible to mount robustly)
- they are *complex relative to their constituents* — for example, a microcontroller may contain millions of transistors whereas a breakout board might have only one or two chips
- they provide a coherent abstraction / *black box-ness*, i.e. you don't have to know what its pieces are, just what a component does and how to talk to it (whereas information about a hobbyist module is likely to lean heavily on the documentation of the components it contains)
- they are *available in large quantities* (e.g. for use in a mass-produced device)
- they are *standardized*, i.e. you can get an equivalent item again later (although this is not always possible, as discussed in "Reproducibility" below)

The use of components rather than hobbyist modules offers a number of advantages for a personally-fabricated electronic product. They tend to be cheaper, smaller, and more widely distributed, allowing for the creation of more optimized, robust, and reproducible devices. Not all components, however, are equally suitable for a personal fabrication approach.

Criteria for Component Selection

In pursuing a personal fabrication approach to making devices, two criteria are crucial in selecting electronic components: *accessibility* (an individual's ability to get and work with the components) and *reproducibility* (the likelihood of being able to get the same, or compatible, components in the future). Accessibility determines the overall feasibility of using the component in a personally-fabricated devices and reproducibility the likelihood that the device can be iterated on or personalized by others.

Note that these considerations don't just apply to the electronics — stock material or even simple hardware like screws or bolts may not be equally accessible or reproducible.

Accessibility

Accessibility in a physical sense depends on the compatibility of a given component with the circuit assembly processes available to individuals. At present, individuals are likely have access to techniques like hand-soldering or solder-paste, stencils, and hot-plates. There are signs, however, that automated assembly (pick-and-place machines) are becoming increasingly accessible, both through lower-cost machines and lower-volume and more-standardized service providers.

In thinking about accessibility, it's important to consider not just whether or not individuals have access to a particular set of tools or assembly processes, but also the cost of those tools and processes. A reasonably-priced bill of materials for a product might be dwarfed by the need for expensive tools (e.g. specialized soldering equipment). Even if the tools themselves are not particularly expensive, having to purchase them specifically for a particular project can significantly increase the cost of fabricating that design.

Accessibility also depends on factors of software and information. The existence of, say, an Arduino software library for communicating with a component can make it accessible for more individuals, as can access to examples of other circuits that use the component. For some components, access to even basic information, like the datasheet, is restricted, making individuals even more reliant on third-party efforts (like reverse-engineering of communication protocols). Finally, many parts are not even sold in small quantities or to individuals, making them inaccessible in a most basic sense.

Not all accessible electronic components are equally convenient. The fewer distributors carry a part, for instance, the higher the likelihood that getting it will require a separate order, adding to the hassle and shipping costs. Some parts have existing Eagle footprints, making them easy to incorporate into PCB designs; others don't. Another benefit of sourcing components from hobbyist sites like Adafruit and SparkFun is that they tend to provide these Eagle footprints for their parts, unlike Digi-Key. Some components have existing Arduino libraries, greatly reducing the amount of work required to interface with them. Some parts have ambiguous, incomplete, or non-existent datasheets, meaning that some experimentation may be required to figure out how to get them to work.

Although not strictly part of the personal fabrication process, accessibility is further served by the use of components that are compatible with breadboard prototyping as well as custom PCBs. This enables the evaluation of components in order to decide which ones to include in a fabricated designs. Some components come in breadboard-compatible form factors that are still relatively compact and, therefore, suitable for production use. Some integrated circuits are available in both breadboard-compatible packages and smaller surface-mount forms, making it easy to transition between the two. Other surface-mount components are popular enough to be available on existing modules targeted for hobbyist prototyping. All of these options facilitate the transition between evaluating the suitability of a component and incorporating it into a PCB design for fabrication.

Note that here I'm concerned with accessibility as it applies to individuals generally, without regard to their individual skills. Even for an expert, there are many components that aren't accessible on an individual basis for the reasons just discussed.

Reproducibility

Reproducibility is best served by the use of generic components that are widely available from multiple sources. Unfortunately, this isn't always possible. Instead, it's frequently necessary to use either specialized parts or underspecified parts.

Specialized parts are ones that provide advanced functionality crucial to the functioning of a device, like the GSM module in the DIY cellphone or the wifi module in the connected devices. While there may be multiple options for the same basic functionality, they may differ in significant ways (like their physical form or software interface) that make them difficult to substitute for one another. This makes the device's design reliant on a single manufacturer, who may decide to stop producing a part — as happened with some essential parts discussed in the "Case Studies" section below. More subtly, the manufacturer may introduce changes in parts of the behavior of the component's functionality, causing hard-to-debug issues in the device. In cases where it is necessary to use specialized parts, it's important to consider factors that can suggest the likelihood of being able to get the parts again in the future. These include the length of time the component has been available thus far, the tendency of its manufacturer to maintain parts for extended periods, and how widely the part is distributed.

Underspecified parts are less specialized but still vulnerable to the same reliance on a single source. These are components that provide a common (sometimes even generic) function but one for which there isn't a common standard or specification. For example, there are many options for components like speakers, RGB LEDs, potentiometers, and displays but each one tends to differ in small or large ways from the others. This often means that even though there are many choices for a component, the design for a device is dependent on the details of a particular one. When ordering these underspecified parts from a different supplier (e.g. in another country), it can be difficult to tell if they will be compatible with the device design — because some crucial characteristics may not be documented. In this situation, or if the original part becomes unavailable, the devices may need to be redesigned, even though there are many similar components available on the market.

Reproducibility can also be at odds with prototyping, as it suggests avoiding the use of parts that are on hand if there's no good source for buying more of them in the future. Collections of miscellaneous electronics junk, for instance, can be a convenient and cheap source of parts for quick prototypes but can make it difficult to recreate a device later. On the other hand, the speed and breadth of many online electronic component distributors means that it's often possible to order needed parts quickly, while being confident of a future supply.

Another consideration is the nature of the steps required to fabricate a product. Unusual or hard to specify steps will make a product harder to reproduce, even by someone who may have the necessary skills. Anything that requires judgement or a sense of feel or fit can be difficult to convey through a digital design file.

Many components don't satisfy the criteria of accessibility and reproducibility, leaving only a sub-set available for electronic products that are intended for others to people to make and reproduce. Furthermore, achieving accessibility and reproducibility can be at odds with other goals for a personally-fabricated electronic product, such as the desired form and function. Components small enough to fit into the desired form factor may be too small for an individual to solder, for example. Gaining access to the specialized parts that are essential to particular functionality may require negotiations with and large minimum orders from a supplier, making it infeasible for the production of small numbers of devices. Components that have been around long enough to be supported by third-party documentation and software libraries may no longer be sophisticated enough to compete with the latest (more difficult-to-use) alternatives. Much of the art of the personal fabrication of electronic products is identifying and making good use of these components.

Embedded Software

Software is the element of an electronic product least affected by the shift to a personal fabrication approach. While the choice of processor may be constrained by an individual's ability to work with it, given a particular processor, the software possibilities are similar to those found in other approaches. It is useful, however, to avoid expensive, operating-system specific software tools and use open-source, cross-platform tools whenever possible.

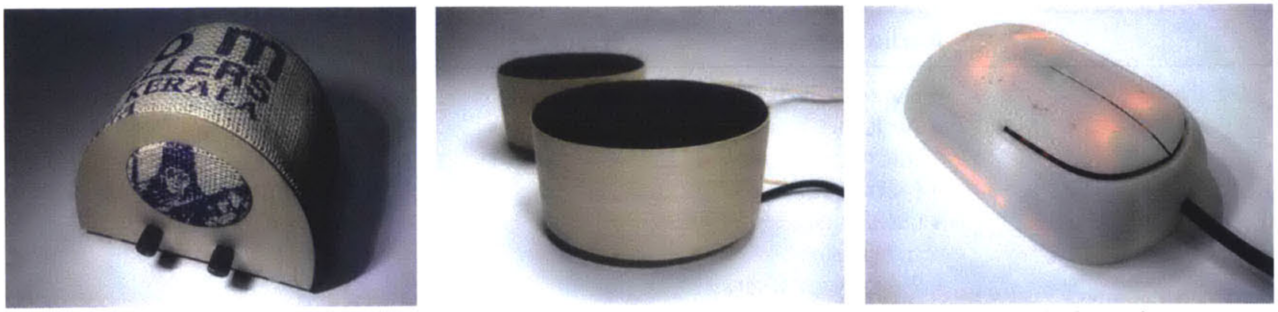
Of course, in replicating or modifying a personally-fabricated device, the ability to reuse and change its software is essential. Sharing the design files for a product's hardware without open-sourcing its software makes it much harder for someone else to make use of the design. Conversely, building on existing open-source software facilitates the personal fabrication of electronic products. In fact, working just with electronic components that are accessible to individuals seems to increase the likelihood of finding relevant open-source software. One reason is that hobbyists and other developers of this software are likely to face similar constraints on their choice of components. Another reason is that the distributors (like SparkFun or Adafruit) that make components available to individuals often provide open-source libraries and examples along with them.

One difficulty exacerbated by a personal fabrication approach is that of building software that can run on many different versions of a device. Small changes to the hardware may require significant changes to the code. Regardless of the complexity of the changes required, targeting multiple hardware versions complicates the software development process, as work needs to be test on multiple devices.

A further complication is that testing changes to a device's software typically requires access to the hardware itself. This can limit the potential contributors to the software, as there may not be that many people with access to the physical device itself.

Case Studies

Here, I present three examples of digitally-fabricated electronic products to illustrate the framework. They are a radio, a pair of speakers, and a computer mouse — all developed as a part of my master's thesis [Mellis 2011] in Leah Buechley's High-Low Tech group at the Media Lab. I explain the ways in which the form and



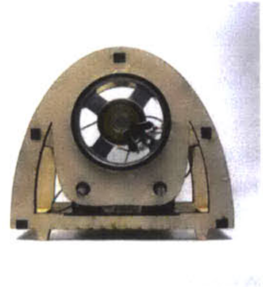
Three personally-fabricated electronics products: the Fab FM radio (left), the Fab Speakers (middle), and a computer mouse with 3d-printed enclosure (right).

design process of these devices were adapted to a personal fabrication approach.

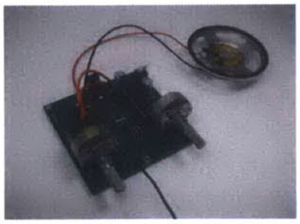
Fab FM (Radio)

The Fab FM is an FM radio receiver that I designed and built with Dana Gordon, starting in Neil Gershenfeld's course "How to Make (Almost) Anything". It includes a custom PCB with a FM radio receiver module and an enclosure made of fabric and laser-cut wood.

The radio highlights some of the opportunities of the laser-cutter as a means of making an electronics enclosure and suggests some strategies for managing its constraints. We used wood (1/4" plywood and veneer) as our primary material, both because it's relatively easy to laser-cut and because it highlights that machine's ability to work with natural materials. We wanted a geometry that would be easy to assemble from the flat pieces produced by the laser-cutter but would avoid the finger-jointed box aesthetic commonly seen with laser-cut parts. Our solution was a frame consisting of two parallel faces connected by press-fit struts. This gives flexibility in the shape of the faces (and, thus, the form of the product as a whole) while being straightforward to cut and assemble. The plywood frame has openings on the sides and top which we covered with fabric — a process that requires additional manual labor but also provides an opportunity for customization. Finally, we used veneer to conceal the holes in the plywood faces, the rough edges of the fabric, and the speaker.



Fab FM plywood frame w/ electronics.



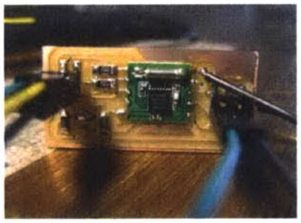
Fab FM circuit board.

The main challenge for the radio's electronics was finding a reliable means of receiving radio stations using components that were straightforward to both purchase and work with. While there are many designs for radio-receiving circuits using only fundamental components (capacitors, inductors, etc), building one that offered good reception seemed like it would require a complex circuit containing many parts that are no longer widely available (like variable capacitors). There were a variety of FM radio receiving chips available, but we couldn't find one that would be reasonable to solder with a manual iron. As a result, we ended up using a module,

	Radio	Speakers	Mouse
Electronics	• FM receiver module • Arduino-compatible microcontroller (ATmega328P) • LM386 audio amplifier • 9V power supply	• TPA701D audio amplifier • 3 x AAA batteries	• Integrated mouse sensor (ADNS-2620) • Arduino-compatible microcontroller (ATmega328P)
Software	Modified SparkFun example (AVR C)	None	Modified Objective Development example (AVR C w/ USB HID emulation)
Fabrication Processes & Materials	Laser-cut plywood & veneer; fabric.	Laser-cut plywood; veneer strip; fabric.	3D-Printed ABS or nylon plastic.
Workshop Audience	Designers & technologists	General public	Architects and industrial designers (w/ CAD experience)

Overview of the case studies.

distributed by SparkFun, that broke out the AR1010 FM radio receiver chip from Airoha into a relatively coarse surface-mount package.



Milled circuit board for testing the FM radio module.

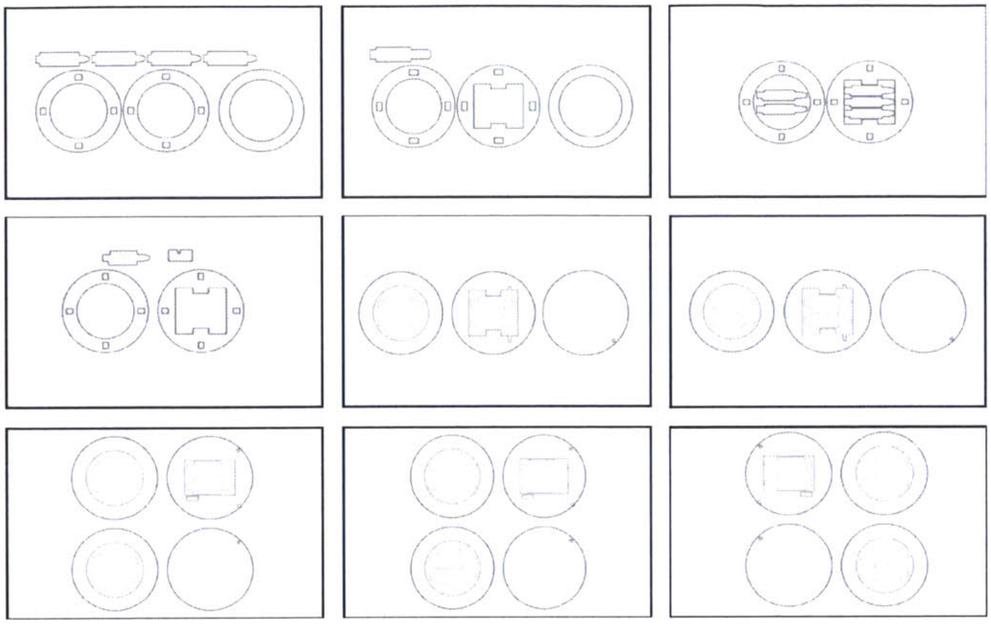
Another challenge was posed by our desire for a classic interface consisting of two knobs: one for tuning and the other for both volume and, the source of the difficulty, an on-off switch. To find a potentiometer with an integrated switch we had to order from a relatively obscure electronics distributor, increasing the number of vendors required to reproduce the radio.

The rest of the components were straightforward to source. We selected an ATmega328P microcontroller, the same one found on the popular Arduino Uno. This made it possible to build on the Arduino example code provided by Sparkfun, the distributor of the radio receiver module. The audio amplifier is the long-lived LM386.

SparkFun's Fab FM radio kit.

Early prototypes used milled circuit boards to test individual components and arrangements, particularly for the radio module (pictured). Having digital design files for each individual PCB made it possible to combine them into a single unified circuit board with relative confidence that the new design would work.

The use of digital design files enables precise alignment between the circuit board and enclosure (e.g. for the knobs which are mounted to the PCB and protrude through the wooden structure). This alignment, however, had to be manually maintained between revisions to the design of the PCB and the enclosure, since the CAD tools used for each didn't interoperate.



Selected iterations of the laser-cutter files for the speaker enclosure.

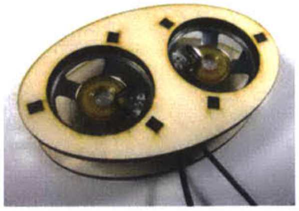
The Fab FM gives some insight into the process of transitioning from a pure personal fabrication approach to a commercial product. The radio is now available as a kit from SparkFun, who had to make some adjustments to our design to suit their production needs. These included swapping out the radio module (which had been discontinued) for a breakout board with a different FM receiver chip, replacing the potentiometers and speaker with ones from other vendors, replacing the veneer pieces with an additional layer of plywood, and switching from 1/4" to 1/8" plywood.

Fab Speakers

The Fab Speakers are a pair of portable, battery-powered speakers similar in construction to the Fab FM. Here, the goal was to design a product that would be as simple as possible for others to make and modify. I wanted to minimize the number of parts and distributors required, the digital fabrication processes used, and the difficulty of the assembly process.

The core of the electronics design challenge was finding appropriate amplifiers together with a compatible power source. I settled on the TPA701D because it was relatively easy to solder (with its 8-pin SOIC package) and possible to power from three AAA batteries (avoiding the additional complications posed by either a rechargeable LiPo battery or a DC boosting circuit). All the electronic components for the speakers are available from a single supplier (Digi-Key). No microcontroller (or code) is required.

To facilitate the production of the speakers by individuals without direct access to a laser-cutter, the enclosure was designed so that the digitally-fabricated pieces required could be ordered from Ponoko. That led me to a design which used a commercially-available veneer strip, since Ponoko didn't include veneer in the materials available for laser-cutting. I created a design for the plywood frame which fit onto Ponoko's smallest size of material, minimizing its cost. (This required going from four legs to three.) I designed the frame to hold the circuit board and speaker in place without additional adhesives, minimizing the steps required to assemble the speakers. The batteries imposed a minimum size on the speakers; they are exposed through a hole in the bottom of the speakers so they can be replaced as required.



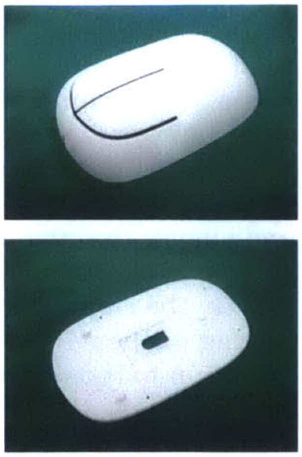
Frame of my wall-mounted speaker variation (above) and an owl-shaped speaker designed by a workshop participant (below).

The speakers also demonstrate the power of hands-on access to a laser-cutter in facilitating iteration on a design. Once appropriate settings are determined for the stock material (in this case, 1/4" plywood) and for the press-fit tolerances, it's possible to cut the parts for a new enclosure in one go. As a result, I was able to quickly iterate through a number of revisions in searching for means of achieving the desired design objectives. In addition to iterating on the main design, I created a variation that combines the left and right speakers into a single housing. This version is designed to mount on a wall and plug into an outlet (rather than run on batteries). In a workshop, one participant created a customized, owl-shaped version of the speakers. By starting from the design files for my wall-mounted variation, he was able to design a suitable enclosure in a single iteration and make a custom pair of speakers in a single day.

3D-Printed Mouse

This computer mouse reveals the many differences that come from using 3D-printing (rather than, say, laser-cutting) in creating an enclosure for an electronic product. Most significantly, it shifts the effort involved almost entirely into the digital realm: more and harder modeling is required but almost no assembly is needed. There are also important shifts in the potential forms and aesthetics: 3D-printing provides greater geometric flexibility but loses the ability of the laser-cutter to work with natural materials like wood.

To maximize the value of the use of 3D-printing, I held a workshop in which architects and industrials designers designing custom enclosures for the mouse. By providing them with a simple 3D model of the circuit board, I made it easy for them to leverage their existing 3D modeling skills in the new domain of electronic product design. The participants created a diversity of forms that would have been difficult for me to achieve on my own.



3D-printed enclosure for the mouse.

The mouse also illustrates the ways in which the availability of individual parts can enable or constraint the kinds of devices that individuals are able to produce. The feasibility of building a working mouse was largely due to the existence of dedicated mouse sensor chips — in this case, the Avago ADNS-2620. This combines a low-resolution camera with computation and image processing algorithms to calculate the motion of the mouse across a surface. The ADNS-2620 comes with a matching lens and LED holder, which help ensure proper operation. At the time I originally designed this mouse, the ADNS sensors were cheap (about $2) and widely available in multiple versions. Avago has since stopped making the sensors, and while it appears that they're now available from another company (PixArt), they are no longer stocked by popular U.S. electronics distributors. This makes it difficult for an individual to create the mouse.

General Principles

These principles are an attempt to break down my experience into a few practical principles for making electronic products using personal fabrication. They're targeted at anyone seeking to design a device for themselves. The principles are divided into two groups: one on the design process and the other on sharing these designs for others to build on.

Designing Devices

Use standard parts and materials (in conjunction with your open source design files). For others to make use of an open source design, they need to be able to get the parts that it relies on, whether those are electronic components, screws, stock material, or something else.The more standard and widely available the parts you use are, the easier it will be for someone else to reproduce your design.That might require foregoing components that are convenient for you if they're not available to others. Note that this guideline is in some ways opposed to some quick prototyping techniques, which may favor the materials at hand regardless of their future availability.

Understand and design for the fabrication process used. Different fabrication processes are good for different things—and they also have different processes and constraints. By designing for a specific fabrication process, you can take advantage of its strengths, avoid its weaknesses, and optimize for its parameters. Be specific: different kinds of 3D printers have very different possibilities, as do different stock materials that you might cut with a laser cutter or CNC machine.Working with a particular machine or process as you iterate on your design allows you to learn the capabilities of the machine and ensure that your designs are compatible with it. Of course, other people trying to reproduce your design might not have access to exactly the same machine or process, so try to find ways to avoid relying too heavily on individual quirks or features. Pay special attention to the tolerances of your chosen fabrication process. Don't create designs that rely on a precision that's not possible to reliably achieve with the machine (e.g., if you have to laser-cut 10 parts to get 2 that actually work, you might want to rethink your design). Hand-soldering is not a particularly exact process; when designing enclosures for a circuit, remember that some components may not end up exactly where the design file specifies they should.

Pursue unique meanings, functions, and aesthetics. The power and efficiency of mass production make it difficult to compete with this approach on its own terms. Instead, try to find unique values for your personally-fabricated devices.Those might come from solving a problem that's of interest to only a small group of people, albeit possibly of great value to them. It might mean using unusual materials or aesthetics to differentiate your devices in ways that might not appeal to a mainstream consumer but might be appealing to someone looking for an alternative. For example, you might look for ways to take advantage of the diverse material possibilities offered by fabrication processes. The use of wood, for example, resonates with the aesthetics of other hand-crafted objects and contrasts with the materials found in most mass-produced devices. It's also easy to cut with a laser-cutter or CNC mill. Acrylic is also great for laser-cutting, comes in a variety of colors and thicknesses, and allows for a range of optical properties. Other unique values might flow simply from finding ways to meaningfully involve individuals in the production of the devices.Take advantage of the fact that personal fabrication allows you to make devices in small quantities to find audiences that aren't well served by existing commercial products.

Design the electronics with the enclosure in mind. The form of an electronic circuit imposes many constraints on the enclosure that will contain it. If you design the circuit first, it's easy to make decisions that will make it difficult to create an attractive, robust enclosure later. Therefore, it's important to keep the enclosure in mind when designing the circuit board. Consider whether the shape of the enclosure (and of the stock materials, if any, used to construct it) imposes any limits on the height or size of the electronic components. Consider whether elements of the enclosure will need to pass between electronic components and, if so, how much space they'll need. Also consider the composition of the overall product — e.g. alignment and symmetry — when deciding where to place electronic components that will be visible. Iteration helps, of course, because once you've made a complete device (circuit and enclosure), it's easier to go back and revise the electronics to suit the overall product's form. Keeping the enclosure in mind from the beginning, though, will help reduce the amount of iteration required to achieve a satisfactory result.

Find ways to make iteration faster, cheaper, and easier. A key benefit of digital fabrication is that every part it produces can be different.To take full advantage of this ability, find ways to iterate on your design rapidly. Getting direct access to a laser cutter, for example, might mean you can try out a few designs in an afternoon instead of waiting a week or two to get a single one in the mail. Similarly, having the electronic components on hand to solder them to a newly fabricated circuit board will allow you test that board more quickly and update its design accordingly. Identify the biggest barrier or barriers to iteration and try to find ways to remove them, whether by getting hands-on access to a machine, using software tools to refine your design before fabricating it, or being able to modify or update a part after it's been made.

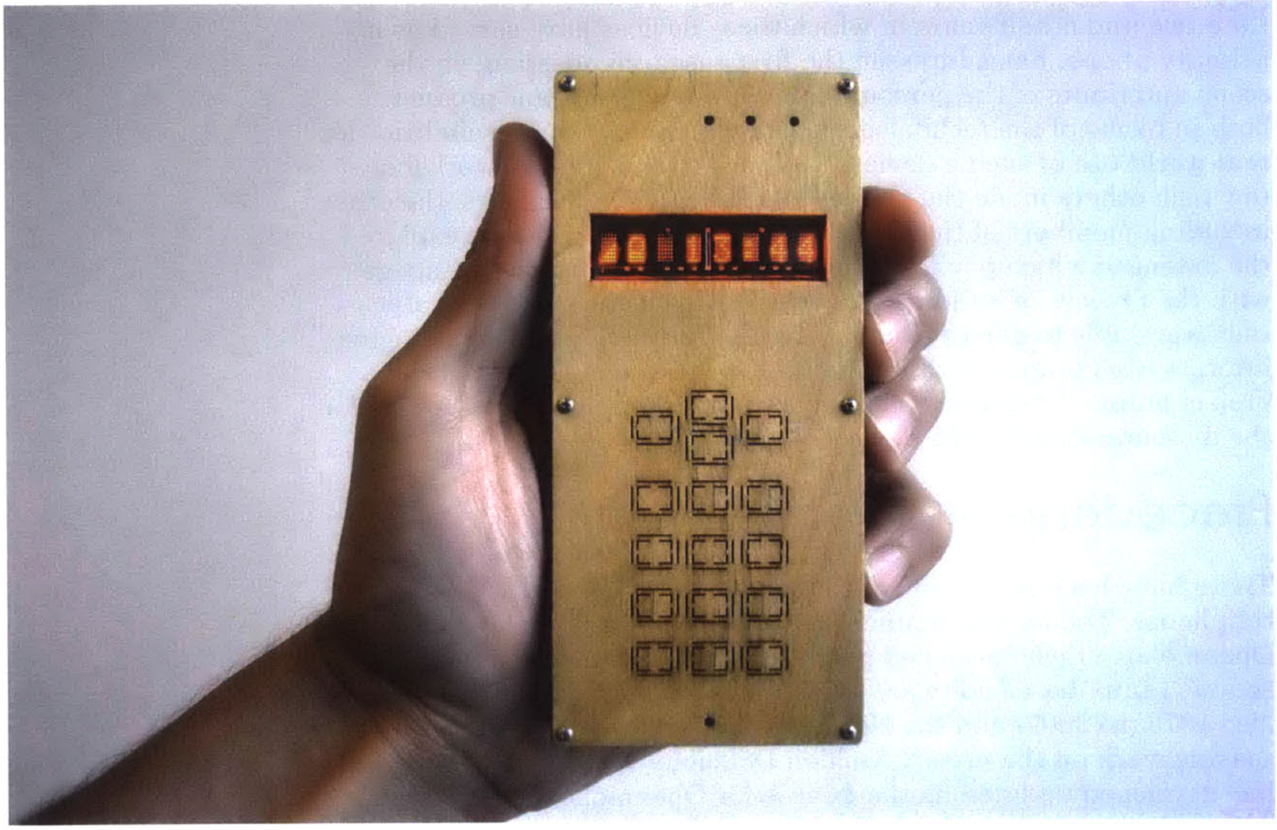
This ability to produce one-off (but reproducible) prototypes facilitates the testing and integration of the various elements of a product. For example, in designing 2D parts to be assembled into a 3D object, the easiest means of testing the assembly may be to simply laser-cut the parts and put them together — rather than trying to build a 3D model in software. More critically, fabricating and assembling a product can help with the integration between the electronics and the enclosure. This is particularly useful when the two are designed in separate and non-interoperable CAD tools or when aspects of the interaction aren't easily modeled in software. The diffusion of an LED's light through a particular thickness of 3D-printed plastic, for instance, may be easiest to evaluate by actually printing the plastic and assembling it on the circuit board.

Sharing Designs

Open source the complements to the hardware itself. Someone who wants to re-create or modify your design will likely need more than just a raw CAD file. Provide whatever additional information seems likely to be useful—for example, parts lists, assembly instructions, firmware, and user documentation. Furthermore, by providing the original sources for these additional resources (not just compiled binaries or hard-to-edit documents like PDFs), you enable others to update them together with your hardware files when creating new variations on a design.

Clearly distinguish between open source design files and the products based on them. Selling a physical product is very different from sharing a hardware design file, even if the former is based on the latter. Someone who buys a product may have higher expectations for its functionality, reliability, and safety than someone who makes a device for himself or herself based on your design. If you make and sell products based on someone else's design, be sure to distinguish between the two, making it clear that the product is from you but giving credit to the original designer.

Make sure the design files reflect the final artifact. In the course of fabricating an electronic product from its digital files, it's not unusual to make tweaks or changes to the object's design. Laser-cut parts may be sanded to press fit together, for example, or a different value of resistor substituted to control the brightness of an LED. While these changes are a normal part of the process, it's important to reflect them by updating the digital design files accordingly. Otherwise, the next time someone makes the product, they'll have to rediscover and reimplement the same tweaks. If it's not possible to update the files, try to at least document the changes that you made.



5. Do-It-Yourself Cellphone

This chapter explores the use of digital fabrication to produce what is perhaps the quintessential device of our day: the cellphone. This is a deliberate attempt to push the limits of DIY practice in order to better understand its scope and limitations. The requirements and constraints of mobile phones – in terms of functionality, size, battery life, usability, and more – make them a particularly enlightening case study. A cellphone is something that most of us carry with us every day, rely on for many purposes, and have a complex relationship with. And yet, cellphones (at least in a Western context) lack the rich DIY heritage of, say, early radios or personal computers. As such, it seems important to ask what will happen when we try to make cellphones for ourselves. To what extent can we even build a phone that will function well enough for us to use it in our daily life? What parts of the design space will we and others be able to explore – and what will prevent us from going further? How much will people be able to understand of a device that's composed of sophisticated and often opaque components and designed using complex tools?

I've taken multiple approaches to this investigation. The first is an autobiographical method, in which I designed, made, and used multiple iterations of a DIY cellphone. This process, including more than two and a half years in which these devices have served as my primary phone, has addressed the first research question, on the scope and limits of the personal fabrication of electronic products, both in terms of the technological constraints and the possibilities for real-world use of such a device. Second, I conducted two workshops in which others made the phones: one focused on designers, the other including members of the general public. These workshops explored the extent to which participants were able to meaningfully engage with the process of building a device for themselves and the value they were able to derive from it. Finally, the cellphone has attracted interest from others, both my friends and strangers who saw the project online. I draw on their experiences to discuss some lessons for the dissemination of DIY devices.

Precedents and Related Work

There have been some previous attempts to build open or DIY cellphones. The most prominent early example is probably Openmoko, which developed phones capable of running a fully open-source, Linux-based software stack. They released two products, the Neo 1973 (in 2007) and the Neo FreeRunner (2008) before largely ceasing work on the project. Golden Delicious, a German company, has developed updated motherboards for Openmoko phones. Although these companies have released design files for the phones, their emphasis has been on creating an open-source software stack combined with commercially-manufactured hardware — as opposed to hardware that individuals could make for themselves. More recently other organizations, including FireFox and Ubuntu, have pursued similar efforts (open-source software running on commercially-manfactured hardware).

Another notable project is the Port-O-Rotary,[39] a cellphone embedded in an old rotary telephone created by SparkFun Electronics in 2005. These were sold for a few years on the SparkFun website. While the Port-O-Rotary phones were designed for hand-assembly, they don't seem intended for extended use.

Another group attempted to build a more practical DIY phone in 2005 and 2006, a project known as TuxPhone.[40] It's unclear whether the project ever progressed far enough to yield a working phone.

SeeedStudio built a mobile phone using a combination of an Arduino Uno, a GSM/GPRS shield, a touch-screen shield, and a custom power circuit, together with a 3D-printed case.[41] This project may be the

[39] https://www.sparkfun.com/tutorials/51

[40] http://web.archive.org/web/20090105165700/http://opencellphone.org/

[41] http://www.instructables.com/id/ArduinoPhone/

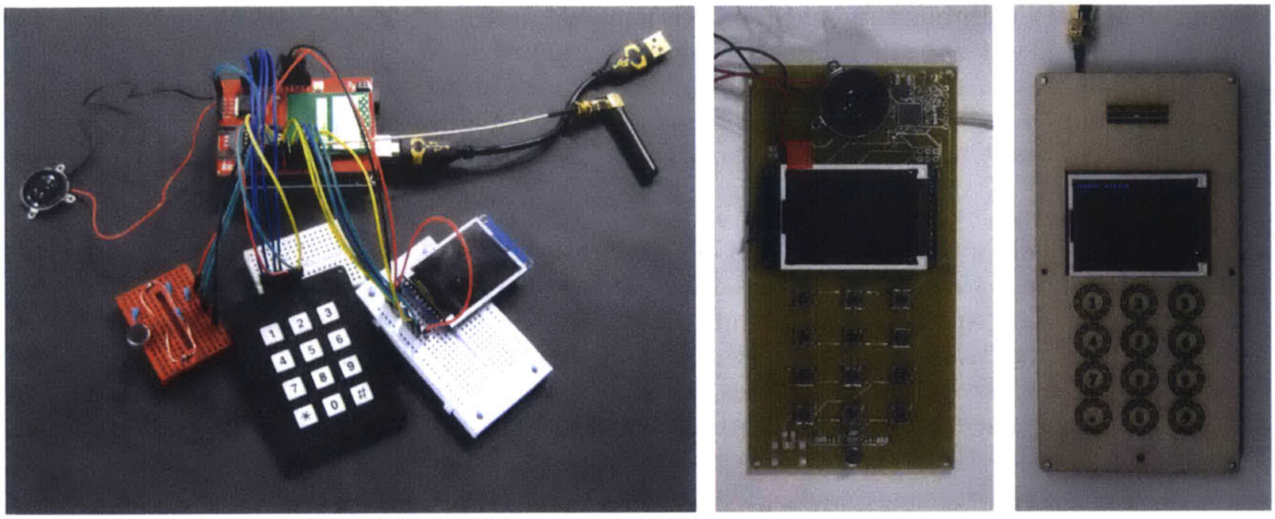
closest to my DIY cellphone in that it's both possible for an individual to make it and it's relatively compact and practical in its form factor. It differs significantly, though, in its use of higher-level modules (Arduino board and shields) as opposed to the single custom PCB and industrial electronic components found in my DIY cellphone.

Given the relatively long availability of GSM modules capable of powering cellphones (and sold for use with Arduino boards), I'm surprised that there aren't more prior efforts to build DIY cellphones. Instead, these modules have been used mainly to provide data or SMS connectivity for electronics devices. Development platforms in this space include the Adafruit Fona and the forthcoming Particle Electron.

The DIY Cellphone

The DIY cellphone is an attempt to satisfy two somewhat contradictory objectives: it should be as easy as possible to individually assemble by hand; and it should be functional enough to serve as one's primary phone. (The tension between objectives these was discussed in more detail in the "Electronic Components" sub-section of Chapter 4.) The phone's electronics and enclosure have been iteratively designed to satisfy these objectives. For the electronics, this requires balancing the overall size and functionality of the device against the difficulty of soldering the components. For the enclosure, the challenge has been to create a robust, attractive form that can be produced using digital fabrication, minimizing the time, cost, and number of processes required.

In its current form, the DIY cellphone makes and receives phone calls and text messages, stores up to 250 phone numbers and names, shows the time, and serves as an alarm clock. It connects to GSM networks (like AT&T and T-Mobile in the United States) and includes a socket for a standard, fullsize SIM card. The phone is powered by a 3.7 volt, 1000 milli-amp hour rechargeable lithium polymer (LiPo) battery and rechargeable via a mini-USB port. In total, there are a little over 60 individual components on the board, including 16 buttons, a microphone and speaker, a magnetic buzzer for generating ring tones and the alarm, and either a black and white LCD display (84x48 pixels) or an 8-character LED matrix (where each character is a 5x7 grid of LEDs). The phone uses the same GSM module (the Quectel M10) and antenna as those found on the Arduino GSM Shield; these components handle the major functionality of connecting to the cellular network and processing audio input and output. The phone's firmware is a~1000 line Arduino program running on an ATmega1284P micrcontroller; it controls the user interface and communicates with the GSM modules, using the Arduino GSM library, custom extensions to it, and other Arduino



Initial proof of concept: breadboard prototype w/ Arduino and shield (*left*), circuit board (*center*) and assembled phone (*right*).

libraries. All the components can be individually ordered from three suppliers (Digi-Key, SparkFun, or the Arduino store). The total cost of the components (in the quantities required to make a single phone) is around $105 for the LCD version, $135 for the LED matrix version. The cost of the circuit board varies from around $10 to $60 each, depending on the desired quantity and turn-around time.

A variety of enclosures have been designed for the phone, both by the author and others. The standard one consists of two halves sandwiching the PCB, which is visible from the side. Each half is made from laser-cut 0.25" plywood covered with laser-cut veneer. The halves are held together with small bolts.

The phone is open-source, with design files for both the electronic circuit and enclosure available online,[42] along with the source code for the Arduino-based firmware and libraries.[43] The circuit was designed in Eagle, the enclosure in Inkscape.

Design Iterations

The DIY cellphone underwent a number of design revisions during its development. These are briefly described here to illustrate the scope and limits of the process of designing an electronic product for personal fabrication.

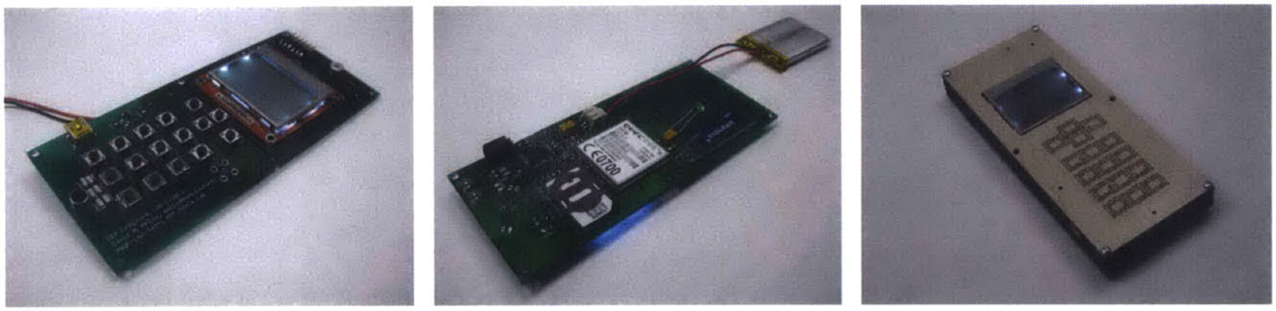
[42] https://github.com/damellis/cellphone2hw

[43] https://github.com/damellis/cellphone2

Initial Proof-of-Concept

The initial prototype of the DIY cellphone was intended as a proof-of-concept, validation of the fact that I could build a device capable of making a phone call. It was built using the SM5100B GSM module, available from SparkFun Electronics along with an Arduino shield designed for prototyping with the module. I did initial tests using the SparkFun shield, then designed a custom PCB for the SM5100B, along with an ATmega328P, a TFT display (on a breakout board from Adafruit Industries), 12 buttons, a speaker, microphone, SIM socket, LiPo battery connector, and external antenna. Soldering the connector for the SM5100B module required the assistance of a microscope, a high-end soldering iron, and an expert, as it had a pitch too fine for regular manual soldering.

This initial board was not intended for daily use and lacked many essential features, including a way to recharge the battery, enough buttons for a reasonable user interface, and a size small enough to fit in a pocket. Still, I was able to write a simple Arduino program that enabled the board to make and receive phone calls. I also designed an enclosure for the phone using laser-cut plywood (two 1/4" thick halves) and veneer, along with some flexure-based buttons.

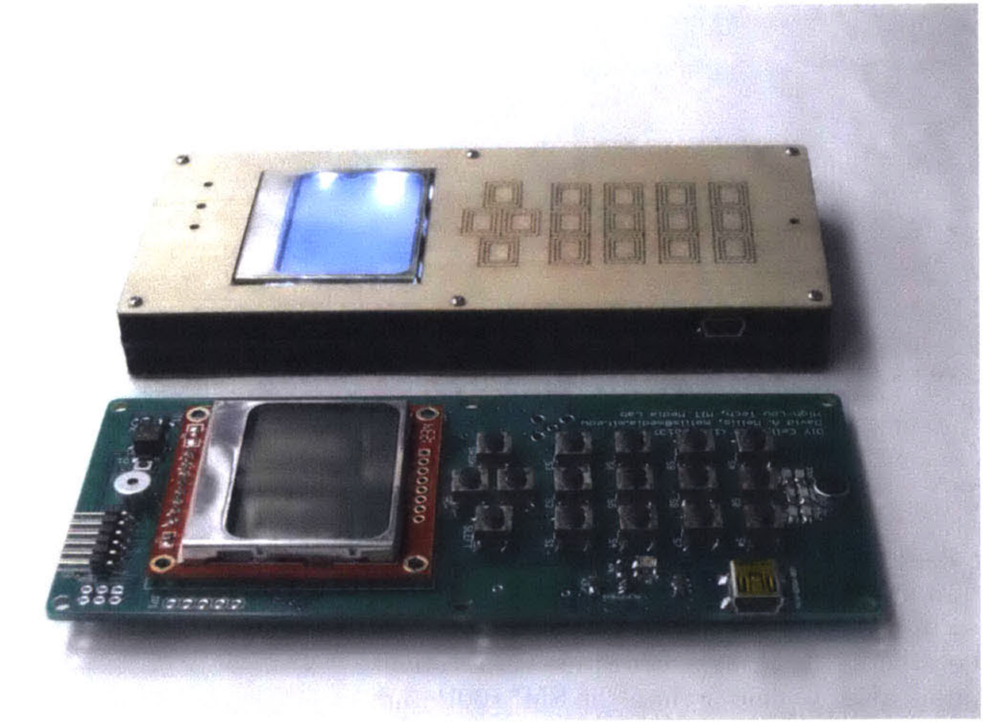


The first DIY cellphone that I used in my daily life: circuit board (left and middle) and laser-cut enclosure (right).

First Viable Phone

This prototype was a substantial redesign, building on lessons from the proof-of-concept in an attempt to create a device that would be viable for daily use. I switched from the SM5100B to the Quectel M10, the module used on the newly released Arduino GSM Shield. The M10 was smaller and easier to solder than the SM5100B and the Arduino GSM library promised to simplify the creation of robust software for the phone. I also borrowed the smaller antenna (and antenna PCB layout) used on the Arduino GSM shield, which fit inside the enclosure of this new prototype. I went from the ATmega328P on the proof-of-concept to the ATmega1284P, a compatible microcontroller with four times the flash memory (needed for storage of the phone's firmware). I switched to a smaller, more robust LCD and added four buttons for navigation to complement the 12 button numeric keypad. I also added a LiPo charging circuit (its design borrowed from a SparkFun product), which allowed the phone's battery to be recharged using a USB cable. A buzzer provided audible ring tones and I found a smaller speaker for call audio. Finally, I shrank the overall length and width of the phone, making it possible (although not particularly comfortable) to carry it in my pocket. Like the proof-of-concept, this prototype was housed in laser-cut plywood and veneer.

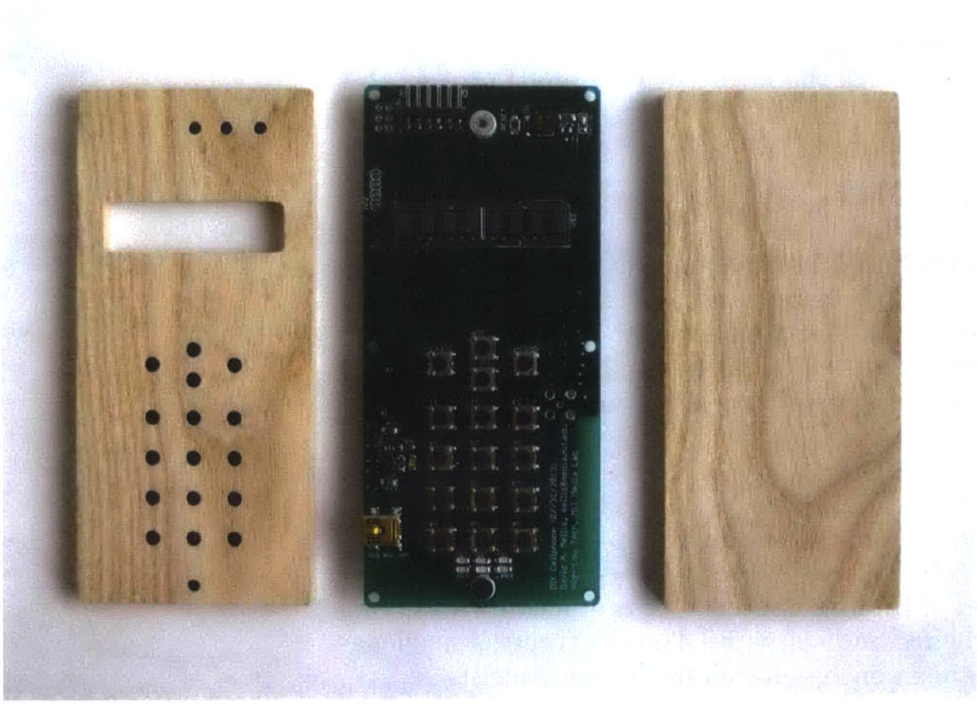
There were many rough aspects to this phone: the corners were pointed and uncomfortable, there was a hole in the back of the enclosure for a programming header, I soldered on an extra LED to show the status of the M10 module, and the bolts holding the enclosure together extended far enough beyond the back of the phone to catch on my pocket. The software was even rougher: it displayed only the phone numbers (not the names) of incoming callers, there was no way to adjust the volume, no notification of a missed call, and many other limitations. Still, I immediately started using this phone in my daily life (around the middle of December, 2012).



Refined prototype with rounded corners and tidier component layout.

Refinements

Over the course of the next few months, I refined both the phone's circuit board and enclosure. It gained rounded corners, more aesthetic placements of the components, improved flexure buttons, a



Phone with LED matrix display and CNC-milled white ash enclosure.

reset button (in case of software crashes), and other minor improvements. I also made various improvements to the software, based largely on problems encountered during my use of the phone. For example, I added volume control, the ability to change the LCD contrast, and the generation of DTMF tones (for navigating interactive voice menus). By the spring of 2013, I had a relatively clean DIY cellphone design.

In parallel with this process, I held my first workshop for others to make the phone (described in more detail later in this chapter).

From LCD to LED Matrix

The final substantial change to the phone's design was a switch from an LCD to an LED matrix, completed in May 2013. This was prompted by the unreliability of the LCD over extended use (it would break after about a month in my pocket). The LED matrix displays just eight characters at a time, which required many changes to the software interface. It also complicates the use of the phone, although this generally hasn't been a problem for me since I know the interface well.

CNC Milled Enclosure

A few months after switching to the LED matrix, I created a CNC-milled, solid-wood enclosure for my personal DIY cellphone. The only change required to the electronics was replacing the buttons with ones with a taller plunger. This enabled the buttons to extend through the wood enclosure, since it wasn't feasible to cut flexures into the enclosure as I did with the laser-cut versions. I've used this phone for the last two years.

Design Reflections

The different versions of the DIY cellphone provide a good opportunity for evaluating my framework (described in Chapter 4) for the personal fabrication of electronic products. Here, I consider the electronic components, embedded software, and digitally-fabricate parts that comprise the phone, drawing lessons from each.

Electronic Components

In designing the phone's circuit board, I've been surprised at how little theoretical knowledge of electronics has been required. Most of the work has consisted of finding appropriate components and tying them together in hardware and software. The electronics knowledge required has typically been encapsulated in the components themselves or documented in their datasheets or other circuits. For example, I'm using an off-the-shelf antenna specifically designed for GSM frequencies, together with a PCB trace borrowed from the Arduino GSM shield. For audio, the microphone and speaker connect directly to the GSM module, requiring no amplification and only minimal filtering (described in the M10's datasheet). Similarly, the power curves required to safely and efficiently charge a LiPo battery are embedded in the charging chip I'm using. The PCB layout required for proper dissipation of the heat built-up during charging is described in the datasheet for the charging chip. And, most significantly, the protocol used to communicate with the GSM network is encapsulated by the GSM module I'm using.

The DIY cellphone provides a useful test case of the accessibility of electronic components. That is, whether an individual (in this case, a relative expert) can acquire all the parts needed to create a functioning device of this type and to what extent it's necessary to rely on hobbyist modules. Here, with the possible exception of the GSM modules, all the components used in the various iterations of the DIY cellphone were industrial components, mass produced for use in products — not modules specifically targeted at educational or hobbyist customers. They are all, apart from the GSM module, available from well-known distributors (Digi-Key and SparkFun). Importantly, the datasheet for the GSM module is available as well. The fact that such a diversity of industrial electronic components are available for purchase and use by individuals is a promising sign for the feasibility of a DIY approach to electronic products. Still, sourcing appropriate parts can be a challenge, as it requires digging through offerings from multiple vendors, trying to find something that performs well, fits the size constraints, and is feasible to buy and solder. Even worse, there are many components that were not feasible to include in the phone's circuit, either because they're not sold to individuals, they're too difficult to solder by hand, or they're too difficult to write code for.

It's important to note that it would likely have been impossible to design a viable smart phone in a way that would be possible for an individual to make for themselves. The processors powerful enough to run the necessary software come almost exclusively in very difficult to solder packages; screens matching the resolution and quality of commercial products are not available for individual purchase; and the overall complexity of the hardware and software would make it extremely difficult for one person to design or assemble.

Given the primacy of component selection in the design process, it's essential to be able to test individual components for their suitability for a given function. I've adopted multiple prototyping strategies for testing individual parts. Some components (like the LED matrix) can be used directly with a standard breadboard, facilitating the prototyping process. Other components (like the GSM modules I used) can be purchased on breakout boards specifically designed for prototyping (e.g. as Arduino shields or with breadboard compatible layouts). Other components (like the buzzer) come in surface-mount or other packages that are difficult to prototype with directly. For these, I sometimes milled small custom breakout boards to test the component. In general, the aim was to verify that a particular component met my needs before incorporating it into a new circuit board design for the whole phone. There's sometimes a tension between the qualities that make a component easy to prototype with and those that make it good for incorporation into the design of a product. For example, the same through-hole pins that make the LED matrix possible to test in a breadboard also made it more difficult to fit into my phone's circuit board, because they required re-arranging the components on the other side of the board.

The other major difficulty in building the circuit boards for the phones has been fitting everything into a form factor small enough to be comfortable for daily use. This places constraints on component selection but also creates hard work in laying out the circuit board and routing the traces between components. This process was much more demanding than the creation of the phone's schematic. That is, even in a making a relatively sophisticated device like a cellphone, much of the complexity comes not from knowing how the components need to connect logically, but physically arranging them in a optimized way. The more one tries to optimize the design, the harder it is. On a bigger board, connecting the components together would be relatively straightforward. The complexity comes from packing them together, not from a need for theory in making individual connections. This is an area amenable to improvement through better software tools; these possibilities are discussed in the "Future Work" section of Chapter 8.

Embedded Software

The phone's software is comprised mostly of a single 1200-line Arduino program. The program uses a collection of Arduino libraries, including the GSM library developed for the Arduino GSM Shield, libraries for talking with the displays in the different phone models, a library for reading from a keypad, and a library I developed of extensions to the GSM library. The software was developed over the course of about a year, from October 2012 until November 2013. It evolved in parallel with my use of the phone, with development priorities determined primarily by my personal needs as a user of the phone.

The ability to draw on existing Arduino libraries certainly facilitated the firmware development process. For example, I was able to find an existing open-source library for each display possibility I tried. The fact that those libraries were open-source was important, as it allowed me to understand and fix them as needed (e.g. patching a bug in the LED matrix display that crashed the phone). The availability of relevant open-source libraries was likely facilitated by my use of an Arduino-compatible microcontroller. More important, perhaps, is the opposite point: that if I needed to use a microcontroller that wasn't compatible with Arduino or another popular hobbyist platform for, say, performance reasons, it would be difficult to find code to draw on in the process. Tapping into an existing ecosystem facilitates the development process but creates another constraint on component selection.

Digital Fabrication

Over the course of the project, I and others have made enclosures for the DIY cellphone using a variety of fabrication processes. These include laser-cutting, CNC milling (of solid wood), 3D-printing, and even hand-cut cardboard. In almost every case, however, a given enclosure has been made using only one of these fabrication processes. Minimizing the number of different processes required simplifies the reproduction and modification of the phone's design, so it's a hopeful sign that the people making enclosures seemed satisfied with the use of a single process for a given enclosure.

By laser-cutting and CNC milling enclosures, I've been able to use natural materials like wood that give the DIY cellphone a very different look and feel than most existing phones. This has been valuable as a means of emphasizing the uniqueness of the DIY artifact. In contrast, the 3D-printed plastic enclosures have sometimes felt like less polished versions of a commercial, injection-molded device. On the other hand, the use of existing materials imposes more constraints on the phone's geometry. For example, the desire to use 1/4"-thick wood stock places limits on the height of the electronic components that could be used. With a 3D-printed enclosure, it may have been easier to accommodate a few components of greater height.

With these flat, stacked enclosures, a key challenge has been finding ways to connect the different layers. Ideally, the process should be easy, attractive, and reversible. Different approaches can be seen in the various cellphone enclosures. For example, the laser-cut cellphone enclosures are held together with screws. This is a relatively straight-forward solution but the screws complicate the appearance of the devices and can catch on one's pocket. The CNC-milled enclosures are held together with magnets, which aren't visible and make it easy to open and close the enclosure. A good strategy for connecting these flat layers is essential to making an attractive, robust enclosure using simple, 2D design tools.

Using the Phone in My Daily Life

In retrospect, the version of the phone that I first started using was very rough. It had sharp corners, large bolts extending out the back, a hole in the back for a programming header, and many missing features in the software. While it had a phone book, it wouldn't show people's names when they called. There was no way to generate the DTMF tones needed to interact with automated menus and, consequently, no way to enter the password to check my voicemail. There was no way to change the volume or check the battery life.

In large part, my willingness to tolerate all these shortcomings was the result of the excitement I felt at being able to start using the phone in my daily life. The initial prototype of the second generation phone went into my pocket immediately after I screwed it together, a process that was repeated with every subsequent iteration. Typically, I'd intend these as prototypes to be refined before I started using them – but, inevitably, as soon as they were finished, they became my primary phone. This was true even in one case when I had finished the circuit board but not the case, so I carried the bare PCB for a couple of days before finishing the enclosure.

This excitement was sometimes shared by others seeing me use the phone. Even people who knew that I was working on DIY cellphones were often surprised to see me actually pull one out of my pocket. They would express astonishment that was this was the phone I had been previously talking to or texting them with. I was actually conscious to keep the phone in my pocket most of the time to avoid conversations with strangers about it, but was pleasantly surprised by the attention it received from the waiters in a variety of restaurants. Friends who had seen the phone would often encourage me to show it to others.

At other times, the phone led to friendly teasing from friends, for example, about the fact that I couldn't use it to look up something on the web. One friend joked "what, do I have to call you now?" when I went a weekend without receiving an email he had sent me, and knowing that the phone doesn't sound an alert when I get a text message. On another occasion, my aunt called to wish me a happy birthday, and the call dropped a couple of times. She gave me a hard time about it, although it wasn't clear if the problem was my phone, the connectivity in the spot I was standing, or on her end. Friends would chide me for reading about new iPhones, as if I were cheating on my phone.

Any problems with the phone were a source of intense stress. The frustration with a non-working device seemed increased because I knew that I had no one else blame: I designed it, built it, and was the one using it. For example, the LCD on early iterations of the phone would became unreliable, I would compulsively check to see if it would turn on, and developed elaborate tricks involving pressing on various spots of the display as I turned it on. Once, my phone was continuously crashing, even when I removed and reinserted the battery; I was pre-occupied until I could get to my computer and figure out the cause (an overly-long text message that crashed the software each time it received it, preventing the text from being deleted).

In short, the current state (working or not) of the latest prototype could become a proxy for my feelings towards the whole endeavor: excitement and satisfaction when everything worked, stress and frustration when anything was wrong. I have to remind myself that certain problems (like not having a friend's number in my phone book or forgetting my charger) have nothing to do with the DIY cellphone per se but could happen with any device. Now that the device is relatively stable, it's become less of an emotional experience — but the process of getting it to that point has been intense and personal.

Workshop 1: Designers

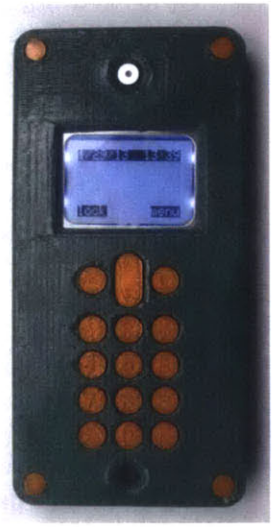
As an initial effort to understand how others might respond to the process of making or modifying the cellphone, I invited colleagues and friends with experience in design and technology to attend a workshop. Nine people participated, three female and six male. Their ages ranges from 21 to 36; six were white, two black, and one Asian. On a pre-workshop survey, many of the participants self-reported an expertise in design but only some experience with electronics and programming. Participants had some experience with soldering, less

	Workshop 1	**Workshop 2**
Audience	Designers and technologists	Including members of general public
Motivation	Exploring new form factors and interactions	Making a phone to use
Values	• Excitement and challenges • Changing perspectives on technological identity	• Understanding devices typically taken for granted • Alternatives to commercially-available devices

Overview of the DIY cellphone workshops.

with microcontroller programming, and even less with circuit board design. All reported regular smartphone use and liking their phones.

The workshop started with an opening discussion. Then participants brainstormed and sketched a concept for a phone that would be ideal for either themselves or someone they knew. Most of the rest of the first day was spent soldering the phones together. In theory, the second day was meant to be used for modifying the circuit or designing and making an enclosure. In practice, we had to finish programming and debugging that day. Some people had other obligations as well, and could only attend for part of the time. Still, everyone was able to successfully assemble the electronics and program the firmware. After completing the assembly and debugging, some participants used their cellphone circuit boards (without enclosures) to call friends or experimented with calling each other from various locations in the building. The workshop ended with a closing discussion, including thoughts from participants on the directions they would take the DIY cellphones if they continued to work on them.



3D-printed cellphone enclosure created by Ben Peters.

Two participants did create custom enclosures for the phone, although both did so in advance of the workshop itself. One enclosure was modeled in SolidWorks and 3d-printed on a MakerBot Replicator. It consisted of seven parts (top, bottom, keypad, and plugs to cover the screws) and required about five hours in total to print, including two iterations of the bottom and keypad. The other enclosure had a bottom half of wood, which was also modeled in SolidWorks and CNC milled on the ShopBot. The fabrication took about five hours (including two iterations), after approximately the same amount of time for modeling. The top half of the enclosure was created by placing silkworms on top of the circuit board and allowing them to spin a silk covering, a process which was only somewhat successful.

Trajectory: Exploring Interactions

The trajectory of most interest to the participants in the workshop was one that wasn't necessarily well-served by its emphasis on low-level assembly. Instead, many participants wanted to be able to explore and experiment with a variety of design possibilities, both form factors and interactions. Their ability to do so in the workshop was limited by the monolithic design of the phone's circuit board, leading to a discussion about a potential modules or a toolkit for cellphone design and prototyping:

> "If you made basically different boards for each of the sub-assemblies of the cell phone, like say there's a screen breakout like there is right here- screen breakout and a keyboard breakout and a logic board breakout with like the antenna and the modem and the microcontroller and the SIM card, and the speaker and headphone - or the speaker and the microphone - breakout, are all connected by flexible cables so you can just mount them in any orientation that you want, so you can make a weird geometry phone that doesn't look like a phone."

> "In the beginning I was mostly excited about like changing the display, changing the form factor, maybe no buttons, maybe a slider, maybe a scroll wheel, and this [the cellphone PCB] kind of puts me back in the form factor that I wanted to get away from."

Participants distinguished between these desired modules or toolkit and both adding accessories to an existing smartphone or prototyping with the existing Arduino GSM shield:

> "Yeah, because then I have control over the shape. And what kind of- Like plugging in accessories into an iPhone: an iPhone is like a huge display and that's what annoys me the most about having this loaf of bread in my pocket."

> "I think this [the Arduino GSM shield] would be fun for doing a GSM-enabled project where I can text to a- home automation or doing stuff like that. Basically controlling / actuating and reading out electronics from a microcontroller that communicates over GSM. But this would not, form factor-wise, it's like even worse than that thing [the DIY cellphone]. The footprint is so big and inflexible."

Going into the workshop, I expected that participants would request more advanced hardware: large touch-screens, a proper operating system, additional storage, etc. Instead, participants seemed interested in the other end of the design space, phones with even simpler interfaces than the DIY cellphone. They described some of their ideas:

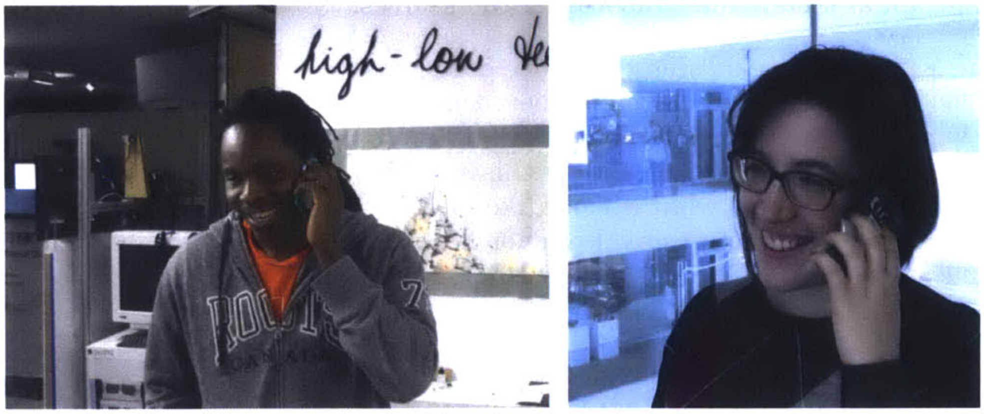
> "It's basically, a screen with two buttons, so you can either accept or reject a call. And you can have a menu to see who it is that called. You can charge it and use your headphones, and that would be it."

> "I just need two functions: one is alarming me whenever there is an event showing up on my schedule, so I won't miss this meeting; and then I just need to receive phone calls. I don't even need any button to call because I don't call people normally. Just receiving calls and alarm me my schedule."

These kinds of design possibilities may have been better served by a toolkit for making cellphones: i.e. higher level modules containing the required functional elements. Participants' interest in technically simple devices suggest that it would be feasible to construct a toolkit providing an interesting range of design possibilities. That would have allowed workshop participants to quickly combine the modules in different configurations to experiment with different form factors and interactions, rather than spending most of their time assembling an existing PCB without much control over its design. Still, even with this mismatch between participants' ideal trajectory and the workshop structure, the participants gained a lot of value from the experience.

Workshop Value

In a closing discussion at the end of the workshop, participants reflected on their experiences. Two themes that emerged were around their excitement and frustration during the process, and about the possibilities and limits of individual empowerment and understanding.



Workshop participants making calls with their phones.

Excitement and Challenge

Participants were visibly excited when placing their first calls on their newly assembled and programmed phones. Two later reflected:

> "There was an excitement coming into the project. I thought it was very cool; I wanted to do it. But once I was able to do it, the coolness I felt was much bigger than what I thought it would be even."

> "I wasn't expecting to be as excited when I first made my first phone call. It was really exciting!"

In part, this excitement may have reflected a reaction to the long and sometimes frustrating process of soldering the components together:

> "Soldering the components correctly was frustrating because they were quite small, fiddly and the soldering was time consuming. Making the circuit board overall was fun and having a working cell phone at the end was extremely rewarding. The excitement from each member of the workshop upon conducting a phone call was evident."

> "It was fun and rewarding to talk with the phone I made. The soldering part was a little bit frustrating but less that I expected."

The excitement may also have reflected the lack of a sense that a phone is something that it's possible to make for oneself, which I discuss next.

Changing Perspectives on Technological Identity

In our society, we typically interact with cellphones as finished, assembled, enclosed devices. The workshop participants reported that assembling the DIY phone helped them to understand that a cellphone is a collection of components like other devices, and that they might be able to design and build such devices for themselves:

> "When I'm choosing a phone, I'm choosing the phone based on what I see, how it looks, I never thought about where the speaker might be, what's the speaker like underneath it, a speaker to me is this set of holes, but then there's this speaker component inside that. I don't know what it looks like in my phone but I know what it looks like in this other phone [the DIY cellphone]. All the different elements, I know what they look like and what they are."

> "I was very happy this morning when you explained me how the capacitors works. I probably know it's not that hard to build a phone but I can never make it by myself. But you know, it's just nice to see an exposed version that you solder by yourself. And then you explained, oh those capacitors were for control the voltage and the others are for protecting or for making the buzzer work. And by the end, you feel like, yes, maybe someday you can customize your own gadget. Cause

> phone, comparably saying, it is not a very simple gadget, so I mean, if you can make a phone, possibly we can some other stuff by ourself."

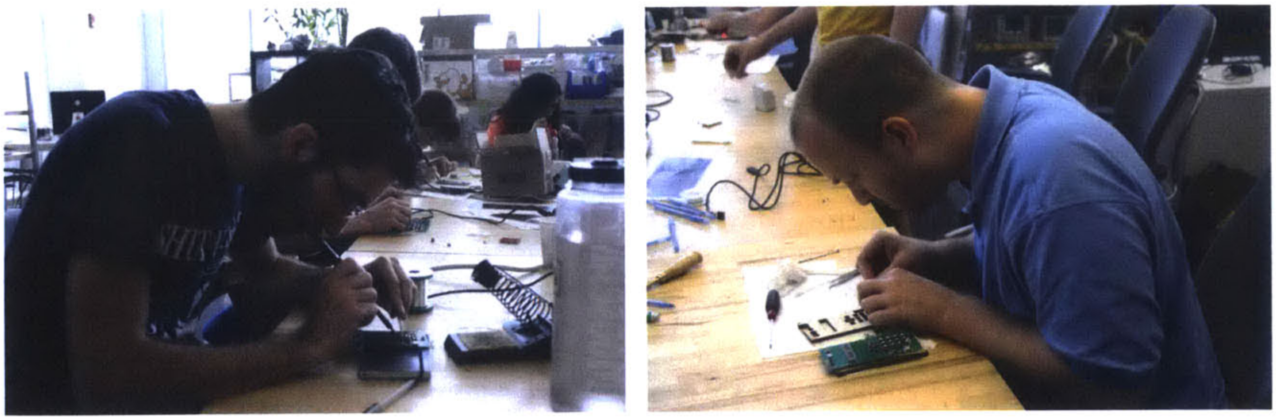
> "It's awesome to see that with these components you can actually build a functional phone, and seeing the code is not rocket science, just one big file. Maybe it's not the easiest thing to overlook but it makes- but it brings down this complex device into something that's feasible and you're like alright, that means if I can do a cellphone, like someone else said, I may be comfortable approaching other devices that maybe are less complex, where I either will be less scared redesigning them."

Understanding how a device works is a starting point for being able to change it. Now that they understand what it's made of and how it works, participants stated that a longer activity would give them the time to further customize the phone:

> "I feel like two days... is enough time to have discussions and stuff, but it's still not enough time to let people work together and doing prototypes and fabrication stuff. It might make more sense if you want to do larger sort of workshops... so you get access to the machines and you can actually do stuff the outcome can be really good, I think. And it's a fun thing to brainstorm on top of phone, and then you can start to customize even this board, to modify the size and maybe modify the keyboards and stuff. That can be very, very interesting. "

> "I think the workshop's definitely long enough to assemble the stuff that you've designed, but I think then, once you've done that, it takes time for that to sink into people's brains and then be able to think about not just how they would want to customize it but the ways in which they would be able to. Cause at the beginning of yesterday morning, you can have ideas about the kind of phone that you want, but now I have a better idea of how that might be manifested: what things might be easier to include or not include."

These quotes reflect a change in these participants' technological identities. In this two-day workshop, they may not have learned all the skills they need to make a device but they've gained the sense that it's possible, and something they could learn to do for themselves. This transformation of an individual's sense of their own capabilities may be more powerful than the acquisition of any particular set of skills, since it can provide the confidence to acquire those skills as desired.



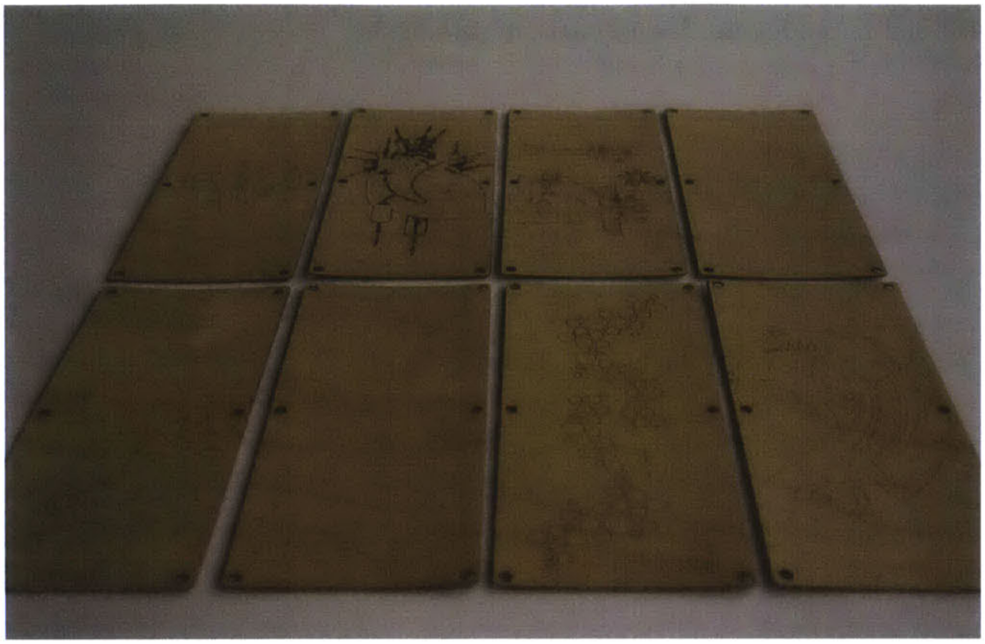
Workshop 2 participants assembling their cellphone circuits (left) and enclosures (right).

Workshop 2: General Public

For this workshop, I wanted to work with a more general audience to see how they would respond to the process of making a cellphone for themselves. I recruited using fliers in local coffee-shops and other locations. Five participants found out about the workshop in this way, six others joined through word-of-mouth from the authors. Participants were charged a $150 materials fee. Three of the participants were female, eight male. Their ages ranged from 24 to 45. Professions varied, although many participants had experience in an area involving engineering or technology but not specifically electronics, like mechanical engineering, computer science, or graphic design. Participants reported less extensive use of cellphones and expressed less positive feelings towards them than the participants in the previous workshop.

Poster for the workshop hanging in Toscanini's, a local ice cream parlor. (Thanks Gus!)

This workshop was two days long (1 to 6 pm on a Saturday and Sunday) and deliberately focused on assembling the DIY cellphone in its existing form. The only customization was selecting a graphic to be laser-etched into the veneer on the back of the phone. After an opening discussion, each participant received a PCB and the necessary electronic components, and most of the first day was spent soldering the boards together. The second day consisted of programming the boards and, mostly, debugging various problems. These issues took a variety of forms, including one unexpected problem: the buzzers of nine of the eleven phones failed to ring when an incoming call was received. This was eventually identified (after the workshop) as a result of a difference in the firmware between two batches of GSM modules. Many participants (six of the eleven) returned after the workshop to continue to troubleshoot or to implement a workaround to get their buzzers to work. Some returned multiple times to continue to debug and would coordinate their visits to work on their phones together. Eventually, all but one participant was able to complete a functional phone.



Participants engraved personalized designs on the back of their cellphone enclosures.

While many participants reported a desire to use their DIY cellphones in their daily lives, a variety of practical obstacles have interfered. At least two participants are still lacking a SIM card from a GSM carrier. Another reported that the phone doesn't get good reception in their apartment. Two mentioned that the lack of a silent mode would be an obstacle to adoption.

Workshop Trajectory: Assembling a Compelling Device

The structure of this workshop fit well with the interests and motivations of many of the participants. It illustrates the assembly of a compelling device as a trajectory for engaging people in the personal fabrication of electronic products. Here, the question of what to do was largely determined in advance: make a cellphone. Despite this narrow focus, making a cellphone was an activity that excited and motivated participants. Those who saw the workshop flier largely came with the intention of making a usable phone. Not all DIY projects are equally relevant and motivating; placing DIY in the context of today's technologies and products can give it extra resonance and meaning for people.

Assembling a project that someone else has designed may not offer much creative freedom but it can still provide a meaningful challenge and an opportunity to develop skills and familiarity with technology. Participants in this workshop spent a focused two days assembling and debugging their boards, in the process developing their skills with surface mount soldering. Many participants in the workshop returned later to complete their phones, a sign of their dedication to the project.

The activity of assembling the phone also exposes the parts that go into an electronic product. On the other hand, it's important to be clear that this type of activity doesn't, in itself, teach people how to understand or design their own electronic circuits. The process of debugging can be a chance to learn more about how a circuit works but, as the experience of participants in the workshop showed, it can also be a frustrating obstacle to the main goal of getting a device to work.

If the goal is to engage people in designing and making their own electronic products, assembling devices designed by others isn't enough to learn the relevant skills and mindsets. It can, however, be a starting point — one that highlights the reasons why you might want to design your own devices and showcases some of the possibilities. Furthermore, for some people, the process of assembling an electronic product can be a meaningful and engaging challenge in its own right, and they may prefer to continue by assembling other products rather than shifting to designing their own.

Workshop Values

Participants in this workshop expressed a value similar to participants in the first workshop on the subject of understanding a device they had previously taken for granted. Among this group, however, there was a stronger interest in the DIY cellphone as a viable alternative to purchasing products from existing companies — and less interest in prototyping of novel form factors or interactions.

Understanding Devices Typically Taken For Granted

In their comments, participants in the workshop expressed gaining an understanding of the elements of a cellphone that echoed the sentiments of the participants in the first workshop:

> "It's kind of cool that, to me, it seems like battery, GSM, antenna, microcontroller, and that's like essentially what you can make a cellphone out of. And that's kind of cool because before, honestly, I understand cellphones to be like fairly simple devices, but I had no scope of, like organs in a human body, like what they're actually- what makes it. So that really just made it seem much more simplified and easy."

> "It's cool that you can treat this all as a black box. You can pull out components and you can make a phone that works with stuff you can buy at home, off-the-shelf. You don't have to know

> how the part works to know the part is working. You don't have to know how the GSM works but how to connect it."

Furthermore, they talked about how this appreciation could extend beyond just the phone itself, to a more general attitude towards the things in their lives:

> "I like that you grabbed our attention and forced us to slow down and say this is how you build a thing you use every single day. It's just, I guess you know the DIY movement is saying don't take everything for granted, see how it actually works."

> "But things like this would definitely make a person more open to re-purposing things they already have rather than buying them. Just things like that, like general- more of a conservative effort, or just like looking at things around them and being like a table is just a piece of wood with some legs on it.... Just taking things apart and not taking them for granted."

They pointed out that this was different than, but not necessarily inferior to, a more general knowledge of electronics:

> "I think there are just deeper questions for me that don't make sense to describe in a two day workshop. I mean you can pick up an electronics book and figure out why you use one versus the other and you know, go to the GSM technical data to figure out how it works but that just doesn't make sense for what we're doing here."

> "I mean, I still don't know anything about circuits. I have no idea of like what capacitors will actually do in a circuit, effectively. Which is not what I really expected to take away from the class, but my suspicion was correct, that I didn't."

It's true that this workshop didn't teach the participants much about the theory of electronics. On the other hand, it also shows that you don't necessarily need to know that much theory to make a working device. By showing participants the extent to which electronics functionality can be encapsulated by components, the workshop has brought them closer to being able to assemble these components together into their own devices.

Alternatives to Commercially-Available Devices

Participants in the second workshop emphasized the ways in which the DIY cellphone differed from those that are commercially available:

> "I wanted a phone that you can't buy because- not particularly this one- but they don't make phones like this any more. I'm a purist in terms of electronics and I don't like how they're

> making computers handheld. I like a radio being a radio, a phone being a phone."
>
> "I just like how it's just like a phone, and that's all I really need. If I need to check things, I can open a computer and check them on that. It's just, I like how it's devoted to one singular purpose."
>
> "You don't see too many wood phones on the market, either."

Some further expressed an interest in DIY devices as an explicit alternative to buying commercial devices, highlighting the political associations of DIY practice:

> "If there's way to have choices so you're not dependent on monopolies for things. That way there's a check and balance to some of this political power, too. It gives more freedom to all people, you know."

This theme was elaborated on in an exchange between two of the participants:

> "What I think is really exciting about these things is they just give more competition to large- like consumers are just kind of like caged animals in a way, where you just have to choose who's giving you the best deal rather than making a better deal for yourself. And that's- I'd gladly pay 150 dollars just to make this than give that to a company with very weak morals or ethics."
>
> "All of the parts are still mass-produced mostly, right?"
>
> "Oh, yeah, that's true but it just feels better than going into like the T-Mobile store and being like, yeah, alright, give me your spiel. Like which one of these things am I going to inevitably going to purchase."

Its ability to prompt this sort of debate is an important value of the workshop. It gives participants a chance to explore their values with respect to the technology in their lives: what aspects of the devices they want control of, whether they care where the parts of those devices come from, etc.

Other Dissemination

In addition to working directly with others in workshops, I've found other ways to disseminate the phone. I've worked with friends to make their own phones and published the design files (both hardware and software) online for others to build on. In addition, the DIY cellphone project has been written up in publications including

DAS IST EIN TELEFON.
AUS ST. PAULI.
WIR
BAUEN DAS
SELBER.
TECHNIK
FÜR ALLE
FAB LAB FABULOUS ST. PAULI ZEIGT
LA FABRICA EN BETA.
DER TESTLAUF FÜR EINE TELEFONPRODUKTION AUF ST. PAULI
16. - 18. DEZEMBER 2013 GEWERBE 5 BEIM GRÜNEN JÄGER 5 20359 HH
OFFEN AB 19:30 INFOS: WWW.FABLAB-HAMBURG.ORG AM PLATTENTELLER C.B.P.

The DIY cellphone around the world: poster from a FabLab in Germany (left) and a maker of the phone at the Maker Faire Shenzhen (right).

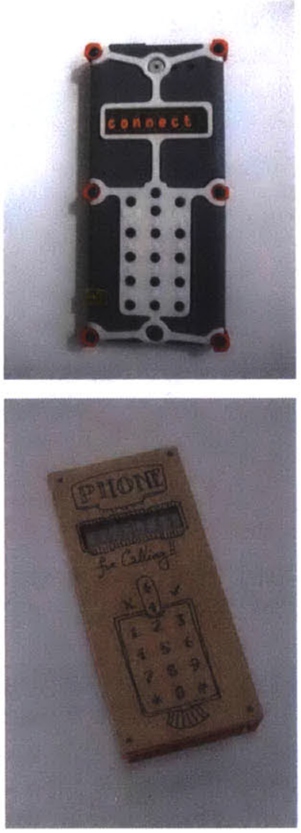
Wired UK, Make Magazine, New Scientist, and various tech blogs, and I was interviewed about the project for a BBC podcast. This coverage probably paid a significant role in generating interest in assembling the phone. It has been reproduced and modified by multiple people in different countries and different ways. Here, I discuss some lessons from this process.

Different motivations have driven different people to make their own DIY cellphones. One master's student in Sweden, for instance, held a DIY cellphone workshop as part of her research work. A German FabLab used the phone as part of an exhibition aimed at rethinking the possibilities for bringing manufacturing back to local urban environment[44]. An individual in Shenzhen, China (who I met at the Maker Faire there) was motivated by the desire to highlight the possibility for individual makers to build devices like a cellphone. My friends may have been partially motivated by their friendship and the opportunity to engage in a social activity. Some of these friends have used the phone in their daily lives, for periods ranging from a few days to many months.

This dissemination has expanded the range of design possibilities shown for the DIY cellphone. Friends and others have built variations on the phone's enclosure, in a variety of materials. These include CNC-milled wood, 3d-printed plastic, and even hand-cut cardboard. Others have created variations on the cellphone's circuit. In preparation for their exhibition, the German FabLab created a

[44] http://www.fablab-hamburg.org/fabrica/

version of the cellphone with additional pins broken out.[45] Another group made a 3G prototyping board[46] based on the Quectel UC15 module, which is similar to the Quectel M10 I'm using in the DIY cellphone. This suggests the possibility of the distributed, decentralized design of electronic products. However, despite these examples of variations on the phone's enclosure and circuit, merging changes back into a canonical version of the design is still difficult, making it hard to integrate the work of many people. (This topic is returned to in the concluding chapter.)



Phones made by friends: 3D-printed enclosure by Amit Zoran (top); hand-cut and illustrated cardboard enclosure by Jeffrey Warren (bottom).

Other people's experiences reproducing the phone offer an opportunity to evaluate the device's reproducibility, an important criteria discussed in Chapter 4 ("Electronic Components" sub-section"). Component availability has been one challenge. The group in Sweden, for example, wasn't able to find the small M0 screws I used to hold the enclosure together. The man in Shenzhen substituted a different and, apparently, much cheaper speaker for the one I used. (It's reasonably easy to find a speaker with the same electrical characteristics as the one I used, but there are very few options in the small form factor I selected.) Availability of the GSM module has been an ongoing challenge, as it's not available from mainstream electronics distributors. For a while, I convinced Arduino to sell the module on the EU-based online store, which provided a reasonable source for Europe. There is a U.S. based distributor of the modules, but they are relatively obscure and, until recently, didn't offer online sales. The module is often available on AliExpress, a collection of mostly-Chinese based vendors, but shipping from China can be slow or expensive. This reinforces the importance of using parts which have widespread, consumer-facing distribution.

A related difficulty is the need for specific tools. The equipment involved (e.g. soldering iron, microcontroller programmer, multimeter) is relatively standard for this kind of electronics project. Some people, however, were interested in the DIY cellphone as introduction to electronics. For them, the need to purchase and learn to use new equipment added to the cost and effort involved. Assembling the phone also requires reasonable skill at using these tools, i.e. for the soldering of surface mount components. Some people have struggled to develop this skill independently. Even some people with existing SMD soldering experience have struggled with some specific steps in the assembly process. For example, the speaker in the DIY cellphone is soldered in an unusual way (its pads are located under the body of the speaker and soldered through holes in the PCB). This process has been difficult to communicate, even to those

[45] https://github.com/axelaxel/cellphone2hw/tree/fabrica

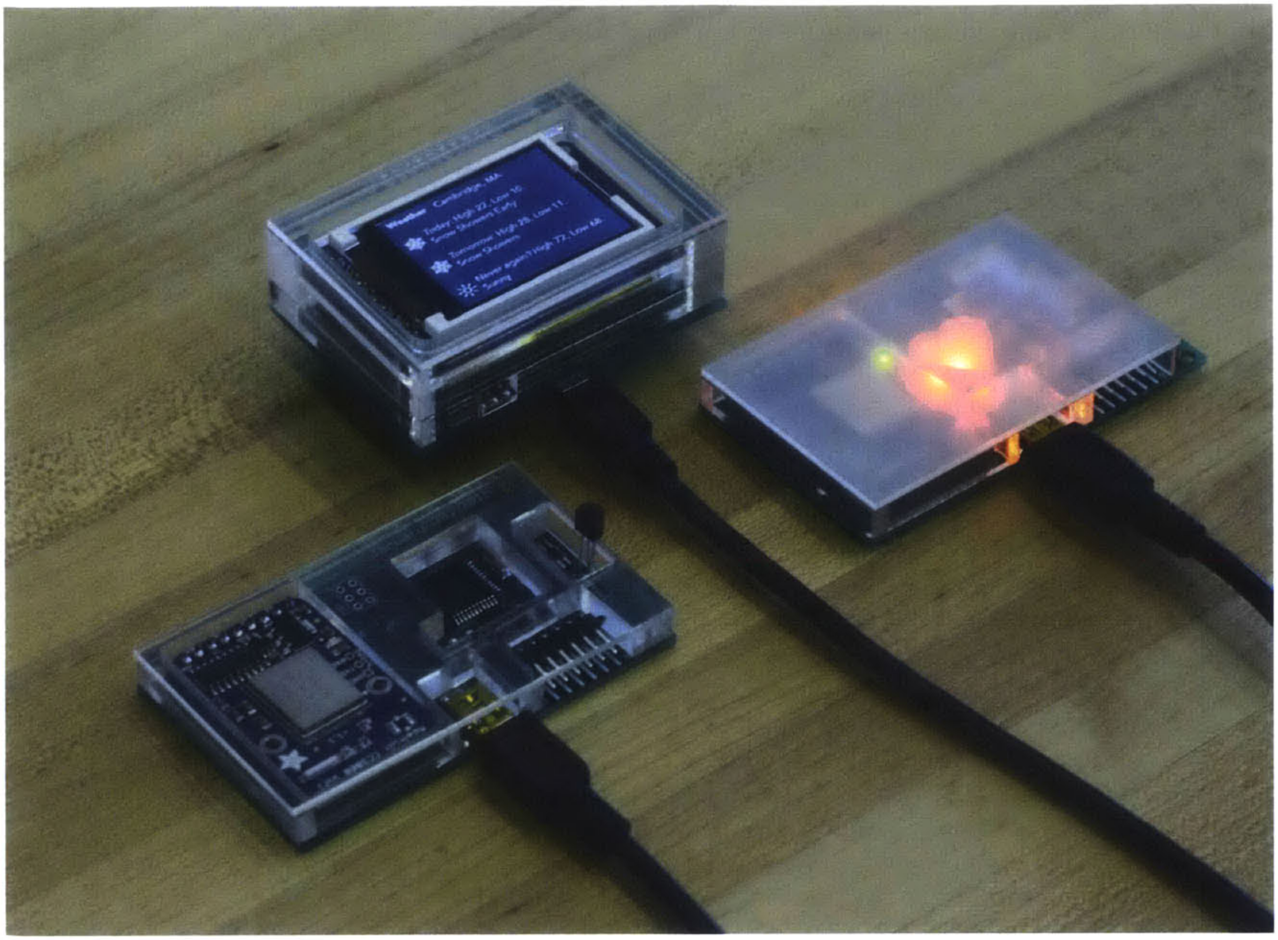
[46] https://github.com/wicker/cellphone3g/

who have had no trouble with the other difficult but more standard forms of surface-mount soldering.

Another challenge has been helping remote individuals troubleshoot their DIY cellphones. Many people have emailed me, or posted on my online forum, asking for help. It's often difficult to get good information about the status of their phones from these written communications. Debugging is an iterative process that requires testing of many different hypotheses as to potential problems, a process that's difficult to do by email. Just checking for a single potential issue (like missing power on the microcontroller) can take multiple exchanges over a few days (e.g. to explain how to test for the problem). If I've run through my list of known potential issues and the device still isn't working, it's almost impossible to know how to proceed remotely. Without the rich information available from in-person observation, it's difficult to formulate and test new hypotheses about potential problems. In many cases, I was able to help people resolve some issues but they still seem to have abandoned the project before getting everything working. On the other hand, for those with experience with the relevant assembly and debugging processes, making the phone wasn't necessarily a problem. These experienced individuals were often able to make the phone without any help from me.

The individuals that did finish making and start using the phone had requests for additional software features. One asked for battery-life and signal-strength indicators, which I hadn't bothered with for my own use. He also requested a silent mode, which I also hadn't felt a personal need for. I subsequently added these features to the phone's software. They have since been useful in my own use of the phone, but I may not have added them (or added them as soon) if not for these requests from someone else. Other features were requested by users in other countries: the ability to enter a + sign for international dialing (which I added), and the ability to properly display text messages containing international non-ASCII characters (which I haven't). One person mentioned needing to supply a PIN for use with their SIM card, something I hadn't encountered with U.S. carriers. These experiences point out that different users may have different requirements for a fabricated electronic product and disseminating its design can be useful as a means of discovering those requirements. It also suggests that there may be enough flexibility in the software that runs on a device to satisfy the needs of many different users, once those needs have been identified.

Finally, publishing the DIY cellphone has generated a number of emails from individuals interested in making cellphones as a commercial product. Unfortunately, the experience I've acquired and the information I've shared in making the DIY cellphone doesn't necessarily translate directly onto commercial production. My assumption is that large portions of the technology behind the phone and the process of making it would need to be redesigned to allow for higher-volume production.



6. Connected Devices

This workshop was an attempt to engage novices in an extended process of designing and making their own electronic product, including component selection, circuit board design, soldering, embedded programming, and debugging. I chose the domain of internet-connected devices both because of current interest in the "Internet of Things" and because it seemed like an area in which a variety of applications could be explored with relatively simple circuits. Through this process, I hoped to learn about what might motivate people to participate in a process like this, what help they need to do so, what difficulties they encounter, and what value they derive from the experience. The workshop context was not intended as a neutral means of examining people's processes but as an explicit attempt to support their efforts. It reveals some of the different trajectories that people can take through the process of fabricating an electronic product and the computational concepts, skills, and practices they can develop in the process. This workshop approach yields some lessons that are specific to workshop contexts and others that generalize to novices engaging in personal fabrication on their own.

In the following section, I discuss my process in developing examples for the participants to build on. Then I describe the workshop itself: recruitment, activities, and outcomes. This is followed by a description of four trajectories taken by the workshop participants, each illustrated by a profile of an individual participant. Then, I detail some of the computational concepts, skills, and practices that participants developed in the course of the workshop. The chapter closes with participants' comments on the value of DIY electronics and personal fabrication.

Workshop Preparation and Examples

As a starting point for the workshop, I created my own examples of internet-connected devices. This served multiple purposes: helping workshop participants imagine possible applications, scaffolding the creation of their own custom devices, and ensuring that I had the technical foundation required to support the participants. There were two main stages in developing the examples: component selection and example creation. I describe them each in turn, partly as a record of my process but also because I think they point to more general issues in scaffolding novice creation of interactive products. First, however, is a short aside on my personal use of an internet-connected device related to those developed for the workshop.

An Inspiration: My Weather Widget

Part of the inspiration for the workshop came from a device I previously built for my own use: an Arduino Yun with an LCD shield that displayed the day's weather forecast. The weather data was retrieved every four hours from Yahoo Weather using Temboo, a system for facilitating embedded connections to web services. The device remained in my apartment for months and I frequently checked it in the morning to find out the forecast for the day.

Revised weather widget with graphical display and 3D-printed enclosure.

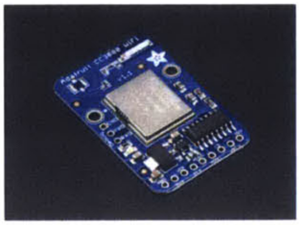
After finishing the workshop, I built a revised version of the weather widget based on the display example discussed below. This revised version has a higher-resolution screen and displays more information, including the current temperature and the forecast for the following day. It also shows a graphical icon depicting the current weather conditions. The use of a custom PCB means that the new version is smaller and cheaper than the initial design. I also designed a 3D-printed enclosure to house this revised weather widget. This updated weather widget is now in my apartment and I check it on many mornings.

Component Selection

As with the development of any electronic product, identifying and selecting core electronic components was an essential step in creating my internet-connected device examples. Specifically, here I was primarily concerned with finding suitable solutions for wifi connectivity and for embedded processing. While the particular components I considered and selected may be of only passing interest, the process highlights some of the ongoing challenges in finding components appropriate for use by novices.

As a baseline, for both processing and wifi connectivity, I wanted something that satisfied the considerations described in the "Personal Fabrication of Electronic Products" chapter above. That is, I had a preference for small, cheap, mass-produced components over larger, more expensive, higher-level modules but I still wanted parts that would still be possible for someone to solder by hand and for which it would be relatively easy to write code. By envisioning the future experience of the workshop participants, I was able to refine my criteria for component selection. Specifically, for the soldering irons I planned to use in the workshop, I wanted components whose pitch (spacing between adjacent legs) was at least 0.8 mm. I also wanted components that would be compatible with the Arduino software, both because of its popularity and flexibility and because of my familiarity with the platform.

The possibilities for wifi connectivity are evolving rapidly, making it difficult to find components that offer both up-to-date functionality and accessible software libraries. I considered a range of options, none of which fully satisfied my criteria. In the end, I selected the CC3000 module from Texas Instruments, on a breakout board from Adafruit. In doing so, I compromised on performance criteria (like size, cost, and functionality) in order to achieve the desired ease-of-use (solderability and Arduino compatibility). The CC3000 is relatively popular and Arduino libraries for it are available from multiple sources. This gave me confidence that it works reasonably well (since I didn't find many complaints about it online) and that I'd be able solve any problems encountered in using it. The bare CC3000 module, however, seemed infeasible to solder by hand, as all the pads are under on the underside of the module. To test this, I designed some boards using the CC3000 module directly but was unable to solder them successfully myself. As a result, I decided to stick with the breakout board for the workshop.



Adafruit breakout board for the CC3000 wifi module from TI.

To further highlight the tradeoffs involved in selecting components that are accessible to novices, here are some details about other wifi options I considered. TI provides newer alternatives to the CC3000: the CC3100 and CC3200 are both single-chip wifi solutions (the latter of which includes a user-programmable microcontroller as well). These are smaller and cheaper than the CC3000 (about 40% of

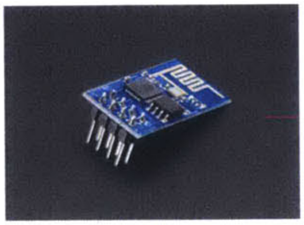
	ATmega1284	ATmega32U4	SAMD21
Solderability	Medium	Medium	Low to Medium
Software Support	High	High	Low (at the time)
Flash memory	128KB	32KB	Up to 256KB
Cost	$8	$6	$3 – $6

	CC3000 Breakout	CC3000	CC3100	ESP8266
Solderability	High	Low	Low	High
Software Support	High	High	Low	Low (at the time)
Cost	$35	$28	$14	$7

Selection criteria for key components: wifi modules (top) and microcontrollers (bottom).

the size and 50% of the cost.). They also supported SSL encrypted network connections. These newer options, however, didn't yet have breakout boards or Arduino libraries, making them infeasible for use in the workshop.

Another appealing wifi option that wasn't quite ready for use by the time of my workshop is the ESP8266. This is a wifi system-on-a-chip made by Shanghai-based Espressif Systems. Breakout boards containing the ESP8266 and a flash memory holding a serial-to-wifi firmware are available for under $10 from Seeedstudio in China (and Adafruit and Sparkfun in the U.S.). While these breakout boards are larger and more expensive than the underlying components, they're still small and cheap compared with other wifi options, and are easy to solder. Unfortunately, as I was planning my workshop, I couldn't find a good Arduino library to use with the ESP8266. I attempted to develop my own (building on some existing code examples from Adafruit) but decided it would take too much time. A vibrant community has grown up around the ESP8266[47] which has, since the workshop, released Arduino-compatible development software for the module.



ESP8266 module.

[47] See, for example, http://www.esp8266.com/

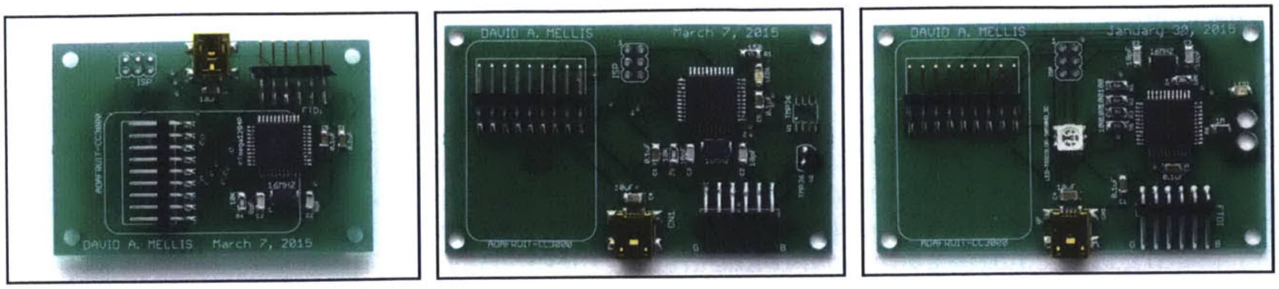
Selecting an appropriate microcontroller involved similar considerations and tradeoffs, although in a less-pronounced fashion. Here, I was able to find a processor (the ATmega1284P AVR from Atmel) that satisfied my main criteria: it's a mass-produced component that's possible to solder by hand and can be programmed with the Arduino software environment. This selection still required some tradeoffs on secondary criteria. For example, programming the ATmega1284P requires additional configuration files to be added to the Arduino software. I would have preferred a processor that was built-in to Arduino but none of those options had both the required pitch and the memory needed to communicate with the CC3000. I also would have preferred the SAMD21 ARM Cortex M0 processor (also from Atmel), which is both cheaper and more powerful than the ATmega1284P and which is the basis of the Arduino Zero. Unfortunately, as I was planning the workshop, the Arduino Zero and the corresponding Arduino support for the SAMD21 were not yet available.

One advantage to the combination of CC3000 module and ATmega1284P was the existence of an Arduino-compatible board with these components — the WildFire from Wicked Device. I ordered some of these for use in the early stages of the workshop.

Example Development

To suggest possible applications of internet-connected devices, I created three examples showcasing different possibilities: pulling information from the internet, logging sensor data to an online repository, and communication between internet-connected devices. The first device includes a display to show information (like a weather forecast) downloaded from the web. The second includes a temperature sensor whose values can be logged. The third example consists of two paired objects, each with a touch sensing electrode and RGB LED. Touching the electrode on one device causes the LED on the other to glow. All the examples include a in-system programming header for burning the bootloader onto the ATmega1284P, an FTDI connector for subsequent programming and communication with the microcontroller, a mini-USB connector for power, a connector for the CC3000 breakout board, and a few passive components.

Developing these examples involved three main challenges, echoing (and therefore, serving as an opportunity to create scaffolding for) those that would be faced by the workshop participants. These challenges are selection of the additional components required (beyond the common CC3000 breakout board and ATmega1284P microcontroller); the design, assembly, and testing of a circuit board; and the programming, including internet connectivity and communication with web sites and services.


ISP
FTDI
ADAFRUIT-CC3000
DAVID A. MELLIS March 7, 2015
DAVID A. MELLIS March 7, 2015
ADAFRUIT-CC3000
DAVID A. MELLIS January 30, 2015
ADAFRUIT-CC3000
FTDI

Example circuit boards: for a display (left), with temperature sensor (middle), RGB LED and touch electrode (right).

Sourcing Secondary Components

Selection of the remaining components for these examples was relatively straightforward but also provides some additional insight into the opportunities for and constraints on personal fabrication. Many of the components needed (like resistors, capacitors, and connectors) are available in standard packages from a variety of manufacturers and distributors. This was also true of the temperature sensor on the second example; multiple mass-produced and easy-to-use options were readily available.

For displays, the situation is more complicated. While both Adafruit and SparkFun sell a variety of different screens, many of them are on custom breakout boards available only from the individual distributor. (This is an example of specialized parts, discussed in the “Electronic Components” sub-section of Chapter 4.) The displays may use a common chipset supported by an Arduino library, but it’s often impossible to find an alternative source of physically-compatible displays. That creates a risky dependency on a single source of future components. Furthermore, many of the screens available from these distributors are relatively low-resolution or otherwise lacking compared with the displays found on modern consumer products. This may reflect a rapid evolution in technology that limits the standardization of displays and their availability to individuals.

Another difficulty was posed by the RGB LED on the paired example devices. Here, the issue is slight incompatibilities between what are considered essentially generic components. RGB LEDs come in many packages with slight differences in size or pin-out, making them an underspecified component, as discussed in Chapter 4. These differences are not necessarily codified in individual package names, making it difficult to find compatible components. Sometimes the LED’s package does have a name, but not one that specifies all aspects of its layout. These issues make it tricky to accurately specify and source the LEDs used in a given circuit design, despite the fact that RGB LEDs are readily available.

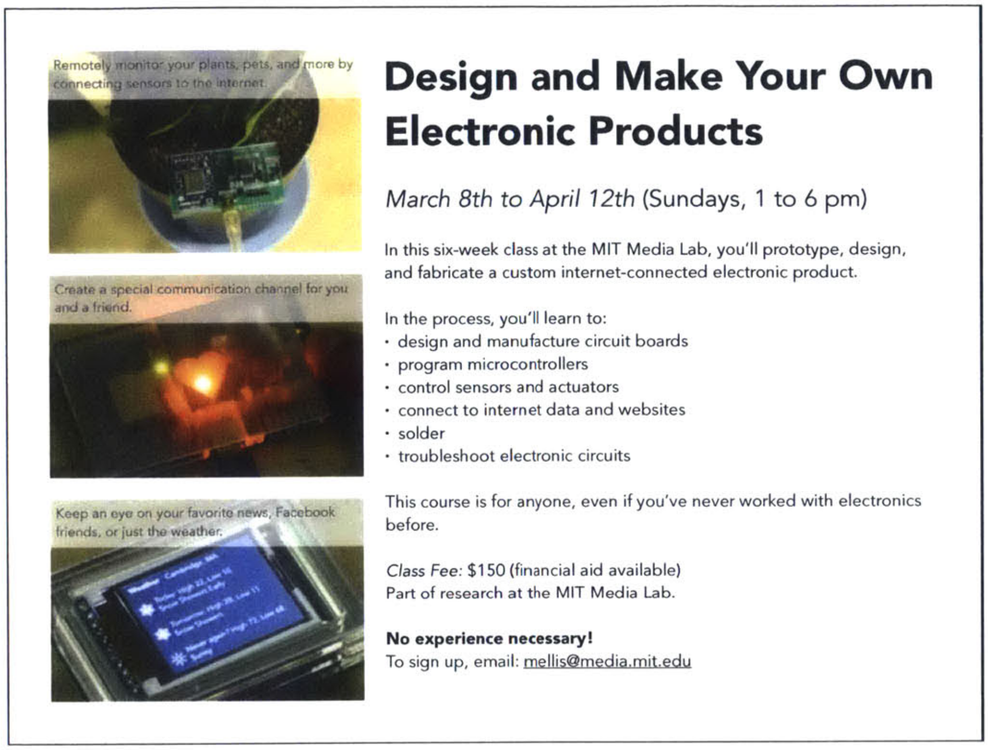
Developing Circuit Boards

For me, it was straightforward to create the schematics and layout for the example circuit boards in Eagle. I created custom Eagle footprints for some of the components, notably the CC3000 and display breakout boards. This involved converting the Eagle files for the breakout boards themselves into a form that allowed them to be incorporated as individual components on other PCBs. One consideration in designing the boards was that it be possible to layout their traces using Eagle's auto-routing tool. (I didn't want the participant to have to manually route the traces as they modified the examples.) Luckily, given the relatively low-density of the examples, this wasn't a problem. I made a few other minor adjustments in refining the examples, like switching from a through-hole to surface-mount connector for the FTDI cable (to allow the board to sit flat on a surface). I also made the connections between the ATmega1284P and the CC3000 module compatible with the configuration of the Wicked Device WildFire board. All of my example boards were ordered from AP Circuits and manually assembled with a soldering iron.

It's worth noting the differences between my examples and the WildFire boards. Both include similar core functionality (microcontroller and CC3000 module) but differ in their intended use: as a digital design to be modified or a physical module to connect to. These differences in intended use impose different requirements on the design of the boards. The WildFire includes additional circuitry, like a real-time clock, SD card slot, and USB to serial convertor, that adds to its flexibility and usefulness as a prototyping platform. These additional components, however, complicate the design files for the WildFire, making them less suitable as a starting point for the design of a custom PCB. My examples, in contrast, are much simpler in design, making them less flexible as a prototyping platform but better suited to their role as examples to be modified.

Programming Internet Connectivity

The software I developed for the examples depended heavily on online services specifically targeted at individual hobbyists and developers. I used two main platforms: Temboo, which simplifies the process of connecting to many existing web APIs, and Phant (hosted at data.sparkfun.com), which provides an online data-store for connected devices. Without Temboo, it would have been difficult or impossible to connect my devices directly to many web services, as they require authentication that's not feasible to implement on the ATmega1284P. Without Phant, or another similar service, I or the workshop participants would have had to develop custom server software to receive and store data, distracting from the focus on making the devices themselves.



Remotely monitor your plants, pets, and more by connecting sensors to the internet.

Create a special communication channel for you and a friend.

Keep an eye on your favorite news, Facebook friends, or just the weather.

Design and Make Your Own Electronic Products

March 8th to April 12th (Sundays, 1 to 6 pm)

In this six-week class at the MIT Media Lab, you'll prototype, design, and fabricate a custom internet-connected electronic product.

In the process, you'll learn to:
- design and manufacture circuit boards
- program microcontrollers
- control sensors and actuators
- connect to internet data and websites
- solder
- troubleshoot electronic circuits

This course is for anyone, even if you've never worked with electronics before.

Class Fee: $150 (financial aid available)
Part of research at the MIT Media Lab.

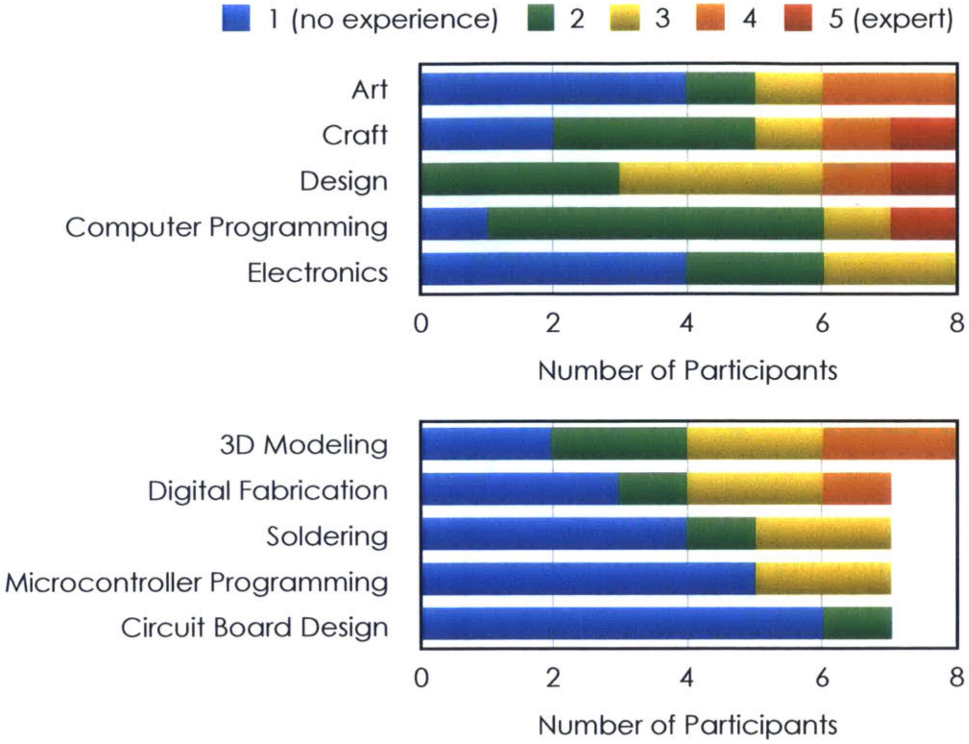
No experience necessary!
To sign up, email: mellis@media.mit.edu

Workshop Activities and Outcomes

I recruited the participants by posting fliers (pictured) in local coffeeshops. When people contacted me, I asked them to fill out an application form. I selected eight participants (out of the first 14 to fill out the application), based on their ability to attend all the classes and in attempt to put together a diverse set of participants. (In total, 45 people emailed me to express interest in the class.)

The eight participants ranged in age from 29 to 44 (median 39.5); three were female, five male; and six were white, one Hispanic, and one Asian. Seven of the participants had a master's degree; the eighth, a PhD. Their professions ranged from architecture and construction to business analyst and tech strategy consultant. They were older and better educated than the participants in other workshops for which I've recruited in the same way, but it's hard to draw any firm conclusions from this anecdotal evidence.

The class ran for six weeks, meeting at the Media Lab from 1 to 6 pm each Sunday afternoon. Participants paid a $150 fee, which covered part of the costs of the class materials. Alexis Hope and Santiago Alfaro (both Media Lab graduate students) assisted participants during some of the sessions. Participants completed a survey at the start and end of the class, along with a brief questionnaire at the end of each session. I also recorded discussions at the start and end of the class and at a few points in-between.

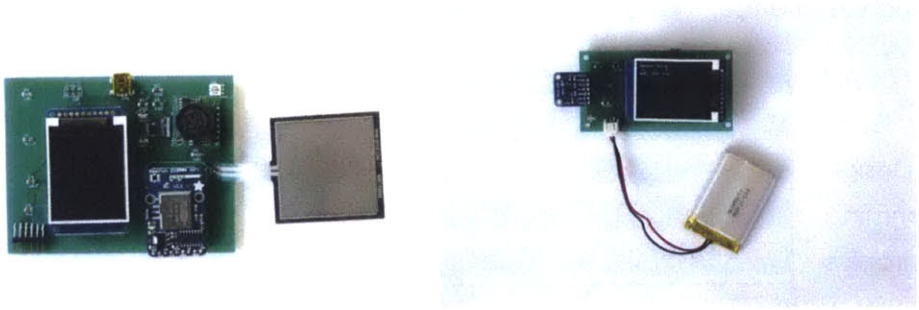


Participants experience on entering the workshop (according to pre-workshop survey). (One participant did not indicate their experience on some of the second set of topics.)

In the first class, I introduced Arduino and participants built basic circuits (and corresponding code) with an Arduino Uno and a breadboard. In the second session, we used the Wicked Device WildFire boards and worked with code to connect to Temboo and Phant. During the third session, participants each assembled one of my example circuit boards and programmed them with existing example code. This included any debugging necessary to get the boards working.

In the fourth, fifth, and sixth sessions, participants designed, assembled, and programmed (respectively) their own circuit boards. They chose a project of interest and distilled their concepts down to a circuit board they could create within the context of the class. Seven of the participants started from one of my examples, using Eagle to add sensors or actuators to the ATmega1284P, CC3000 connector, and other core components. The eighth participant designed their board, an Arduino shield, from scratch. By the end of the workshop, seven of the eight participants had successfully assembled and tested their boards, writing some initial software for them.

While many of the participants' projects added only a small number of components to my examples, they addressed a number of different application areas. One participant created a board with an



Two participant projects: John's snow detector, with ultrasonic range-finder and pressure sensor (left) and Nathan's accelerometer to be mounted on a fencing foil (right).

accelerometer that he planned to attached to a fencing foil as a means of detecting incorrect posture. Another created a board with an ultrasonic range finder and a pressure sensor that he intended to use to detect snow (as part of a networked system of snow removal). A third participant added a current sensor to his board so he could monitor and log the power consumption of his appliances. Another participant added pressure, humidity, and temperature sensors that she wanted to correlate with online weather information to predict the likelihood of getting a migraine headache.

Participant Trajectories & Profiles

Not everyone takes the same trajectory in learning to work with technology. Understanding these different trajectories is a crucial part of engaging people in the personal fabrication of electronic products. Here, I discuss four different trajectories that emerged from the connected devices workshop, each illustrated by a profile of one of the participants. The four trajectories are:

1. Applying an engineering mindset (Tyler)
2. Interest in an application domain (John)
3. Finding meaning in the process itself (Ernesto)
4. Struggling with technological agency (Nicole)

All names used are pseudonyms to protect the identities of the participants. (Also, for legibility, I've elided most occurrences of "like" and "um" in participant quotes.)

Trajectory 1. Applying an Engineering Mindset

For three of the participants — Tyler, Nathan, and Akash — the connected devices workshop was a chance to extend their existing engineering experience and mindset to the personal fabrication of electronic products. None of these three had extensive experience with DIY electronics before the workshop. On the opening survey, all reported their prior experience with electronics, microcontroller

Participant	Age	Gender	Degree	Profession	Project
Nicole	29	Female	Masters	User experience research	Unfinished
Tyler	30	Male	Masters	Product manager	Water weather station
Akash	31	Male	Masters	Software development	Smart home hub
Ernesto	37	Male	Masters	Business analyst	Current sensor
John	42	Male	PhD	Construction builder	Snow detection system
Luisa	42	Female	Masters	Architecture and urbanism	Headache prediction system
Nathan	44	Male	Masters	Strategy consultant	Fencing posture detector
Brooke	44	Female	Masters	Environmental management / jewelry	Email-controlled LED

Backgrounds and projects of the workshop participants.

programming, and soldering as three or less on a five point scale (with 1 being "no experience" and 5 being "expert"). On the other hand, all three of these participants have experience with engineering. Akash is a software developer and Tyler and Nathan work in technology firms, although Tyler is in marketing and Nathan does little hands-on work. These backgrounds meant that these participants came into the workshop with experience in several important computational concepts and practices, particularly debugging, forming abstractions, the connection between code and circuits. (These concepts and practices are discussed in more detail in the next section.) These basic engineering mindsets or practices seem to have been more important to Tyler, Nathan, and Akash's success in the workshop than any specific electronics knowledge or experience. Tyler was perhaps the most successful of the three.

Tyler

Tyler has degrees in aeronautical engineering but never worked as a professional engineer. His education included some programming and electronics, but these subjects weren't ones he enjoyed or excelled at:

> "Did circuits and computer science on the intro classes as an undergrad. Didn't always see the application to them. It was all just like rate theory and op-amps and didn't say what you could do with an op-amp, so it kind of went over my head."

After school, he spent some years in the armed services as part of the ROTC program. Now he works in sales and marketing for a technology company.

Tyler recently started to see more relevance in electronics:

> “When Arduino came out, that was a real cool thing, cause it was like oh cool, I can actually- it gave me some motivation to learn these things, but also lets you see the real world application much more easily. Oh, cool, well if I figure this out, then I can do that.”

Still, he had trouble taking it beyond a certain point:

> “At one point, I bought a LED matrix and was like: I'm going to make a bike turn signal… I saw the words shift register online. And I was like, I'm going to buy that and wire it up on a breadboard and figure it out and I just couldn't figure it out because there's this snake nest of wires coming out to go to a shift register from an Arduino.”

Something similar occurred in other instances as well:

> “Figuring, oh I'll buy the parts and pieces and then I'll figure out how to use them together without a lot of really coherent thought- You know, I'll just figure it out when I get it. And it didn't really work that well.”

Tyler reflected that the obstacles to finishing his projects may not have been exclusively technical:

> “There was no one around me who was doing it, so there was no one I could talk about it with… I know all this stuff is online but there's so much online that it's almost overwhelming… It just seemed like you had to be more of an expert to even get started. That might not have been true. And actually with experience in the class, I know now that that wasn't true. But that was my perception of it.”

For Tyler, the class seemed to foster increased independence and follow-through. He had come in with a project idea in mind already, for a weather buoy that could remotely report the conditions from a location out on a body of water. He even had some notion of what it would take to build the project:

> “I know I need to have a thermistor, I know I need to have a power supply management charging system for the solar panel. I need to have the GSM. And put all that stuff together.”

The class helped provide him with the skill and confidence he needed to better understand and integrate the relevant technologies:

> “It was really helpful to have a nice breakdown for me on that first and second week of here's all this stuff about physical sensors, here's all of the stuff about in-out, you can go with wifi, you can go with Zigbee, you can go with everything else. So yeah, having that kind of systematic approach helped me

> kind of break it down a little bit better and figure it out from there."

One area in particular in which Tyler advanced was programming. Before the class, as he put it: "the coding of it [was] maybe the most alien to me... while loops vs. for loops and maybe with my poor CS performance in the past, I'm like 'I don't know'". Through the in-class exercises, he gained in confidence and ability.

> "I really like how there was- you guys showed us to do the one switch with the one LED and then you said, 'alright now put two LEDs in make them do different things'. And then, 'now think of something else you can do that's more complicated'. It was nice to have that kind of like 'okay, you're starting to learn how to swim, you can do a little bit of this, now go a little bit deeper, go a little bit further. We'll be here but you should try to lead yourself out there.'"

For Tyler, the workshop was a particularly suitable context in which to engage in this kind of learning.

> "It was nice to have that kind of support as we were poking around. If for no other reason other than when I got frustrated or stumbled into a wall, I couldn't just put it away and go watch TV cause you guys were right there and we had three hours left."

Tyler's project took the form of an Arduino shield with connections for two thermistors (for measuring the air and water temperature) and an accelerometer (for detecting waves). It was the only project that didn't start from one of my example boards. Instead of wifi connectivity with the CC3000, Tyler used an Arduino GSM shield to enable connectivity from more remote locations using the cellular network. He planned to use a solar panel and rechargeable battery to power his circuit, enabling it to operate autonomously. Tyler worked very independently throughout, requiring much less assistance than the other participants. As he described it:

> "Reading temperatures and making a bunch of different readings and then averaging them out to smooth out any noise. Getting all that. I've almost finished a way to look at the accelerometer really quickly over a period of and find my max and my min acceleration and try to infer something about wave height from that. But that's all going pretty smoothly and that went pretty smoothly."

When asked why he seemed to have a smoother experience than other participants in the class, Tyler speculated that it may have been related to being around engineers and engineering in his work life.

> "Being around through several rounds of defects being discovered, oh well a supplier gave us a bad batch of this, or there's a firmware bug here that drove and saturated the MOSFETs and the power supply blew up... I was never the one designing it, or designing the fixes, but I was out there in the field sending wireless firmware updates... I've seen people running PCB layouts and going 'oh hey, what's that little thing' and it was like 'this is a this, that's a that'. So maybe my baseline environmental exposure has been a little bit higher."

Tyler expressed a strong belief in the value of DIY electronics. For him personally this was partly because of its potential to serve as a "satisfying personal hobby." Tyler also saw possible professional benefits:

> "It's definitely something that I've put on my resume now... open-source, Arduino... You know, I might work in marketing but I can speak a little bit of this stuff and ... if there's some technical aspect or interaction that I can have that someone else can't do or doesn't have. So I think there's a little bit of that kind of professional skill development ... as well."

Furthermore, he sees opportunities to make things that otherwise wouldn't exist:

> "And some of the stuff just isn't out there, like for swimming there's really not that great electronics... I've been thinking about ... what if I could have a heading indicator on my goggles. That's just a thing that no one makes. And there's nothing I could buy for that."

Tyler also highlighted the values of DIY electronics for society more generally. "It's almost like literacy, of being able to interact with [technology] on a meaningful basis beyond simply consuming it or using it."

And later:

> "I think if you're mystified and baffled by the world around you, whatever it is, it could be electronics, it could be something completely different, I think that gives you a lot less agency in your day to day life... That's how I would pitch it. Know more about the stuff around you that has such a strong effect on you."

Tyler also expressed privacy concerns about getting technologies from companies:

> "I though the Nest was really cool, the thermostat, and then Google bought it and then I got really creeped out. And then they bought DropCam, and I am like I am not putting any of

> those in my house. And I know Google already knows everything about me already but that was just like a thing where I did not feel- that was the privacy line. And so, there's definitely a thing where if I have to make that choice, if I could like build my own Nest, I'm totally all about that versus buying something readymade that has baggage."

For Tyler, the workshop has had a seemingly transformational effect. Before taking it, he had some interest in and exposure to electronics but struggled with projects and his own motivation. Through the class, he developed an ability to complete projects independently and confidently. In doing so, he built on his existing understanding of engineering and developed the concepts, skills, and practices needed to apply that mindset to the domain of electronics.

Trajectory 2. Interest in an Application Domain

Both Luisa and John came into the workshop with an interest in the applications of sensors and network connectivity to the city or built environment. This interest relates to their professional lives: Luisa is an architect and John has a PhD in construction management. Like Tyler, Nathan, and Akash, these participants were able to define and execute a feasible project with support from the workshop facilitators. Unlike those participants, however, Luisa and John seemed less satisfied with this process of picking a single project for the sake of going through the technical steps involved in realizing it. Both later stated that they would have preferred to skip the PCB design aspects of the workshop, instead leaving more time to gain familiarity with electronics and programming and to experiment with their applications in their area of interest. Interestingly, this was despite the fact that both Luisa and John expressed an enjoyment of Eagle itself, fostered by their existing familiarity with architectural CAD software. Luisa also did excellently with soldering, assembling two boards without errors. Again, their enjoyment of or skill with these aspects of a personal fabrication approach didn't compensate for the fact that they distracted from their primary interest in exploring an application domain.

These participants may have been better served by a workshop using other electronics prototyping processes, like electronic toolkits and breadboards. This would have allowed more time to experiment with different possibilities for monitoring the environment using sensors and collecting this information over a network. A prototyping approach would also have given these participants more opportunities to develop the necessary technical concepts and practices (discussed in the next section), many of which they struggled with. These tradeoffs emerge clearly in John's story.

John

John is a 42-year old with a background in civil and environmental engineering, including a PhD in the area of lean construction. During the dot-com era (late 1990's), he explored the role of IT in construction, including research and some work in CAD and programming. More recently, as a researcher at a research lab, he had some tangential exposure to electronics through his interactions with another group working on wireless sensor networks. Still, his decision to enroll in the class seems to have been relatively spontaneous:

> "It was kind of a random event seeing your flier in a coffeeshop in Davis Square. It was kind of like, 'Goddammit, I need to find out more about this.' And what better way to- someone's giving this gift, if you like, and go find out about it." (one-on-one conversation)

Despite (or perhaps because of) his experience with engineering, John felt apprehensive coming into the class: "It was a bit of a frontier for me. I think I felt very vulnerable walking in the first day going 'what am I doing here?'" (closing discussion)

John's project was a snow advisory management (SAM) system, inspired by the preceding winter's record-setting snowfall. He worked on developing a system that could detect the presence of un-shoveled snow, particularly in critical locations like around fire hydrants, and route that information to relevant parties. As a starting point, John created a presentation discussing the context, use cases, and potential implementation of the SAM system. He also developed a physical model of a house to help think through and communicate the system. We had some email exchanges about the best way of sensing snow and, after some Googling, I suggested an ultrasonic distance sensor to measure the height of snow piles. John derived his circuit from my display example, adding an ultrasonic distance sensor, pressure sensor, and RGB LED. Additionally, he re-laid out the circuit board from his modified schematic.

John had to miss one session of the class due to family commitments. As a result, he wasn't able to finish assembling and testing his board within the six class sessions. When he returned for a follow-up interview, I helped him finish the board and test each of its individual components. This included soldering a jumper wire to compensate for a connection that was missing from the schematic and board layout. In the end, the entire board seemed to work correctly.

In his comments on the class, John highlighted the tension between constructing a project to address a particular functional goal and the process of learning how to work with electronics. In a conversation at the start of the third class session, he talked about the project as motivating his work: “At the high level, I'm making the project drive my thinking or having a solid project to understand- using it as a pull mechanism to learn more.”

On the other hand, he acknowledged that in order to focus effectively on potential applications, the class may have required more incoming knowledge on the part of the participants:

> “I've seen what, I guess, pre-requisites could be needed for coming into this class to really build out an entrepreneurship-style product, proof-of-concept, thing.”

He even speculated that the class may have been more valuable if it had assigned participants specific projects to allow them to focus more on the technical content and learning. Alternatively, he suggested assigning out-of-class homework to provide additional opportunities for working with the technology.

This tension between solving problems and personally learning about or working with technology persisted in John’s speculations on future opportunities. He described one pathway focused on personal engagement with technology:

> “There's the hobby thing and playing with SAM and kind of feeling that you're in your little fiefdom of your garage and you're trying to save the world.”
>
> “Then there's just watching what's happening and being a spectator and keeping up-to-speed.”

On the other hand, in thinking about possible uses for technology in his professional construction work, he expressed a preference for commercial solutions:

> “And you'd probably always be leaning to working with a vendor versus- that's the thing, you know, what's the saying about 'nobody ever got fired for hiring IBM?', is that right? ... When you're working with a company that's conservative, they're not going to say we'll pay you to go off and put 20% of your time into researching this.”

John seemed interested in combining a personal involvement in technology with addressing real-world problems, but described this as something that would require technically-experienced partners:

> “And probably once I figure out how the work scene myself, there is a John Weller Inc. that would like to try keep playing in this space of coming up with applications, but I know that, personally, myself, I think I could have a keen eye for an application but I would definitely need to team with people,

> software engineers, others who might be more technically savvy and can do just a lot quicker and faster."

Even if he personally had the technical knowledge to build a prototype, John was conscious of the larger factors involved in developing a scalable solution:

> "I will say this class has helped me conceive a way of thinking about a proof-of-concept. I suppose if I had more time on the internet-of-things and understanding a way of seeing that future that I'd even be more confident of saying actually we can create our own proof-of-concept here. But then you've got that business block- or that entrepreneurship block of like okay, well, okay you can make a set of gadgets here for this building; who's gonna design, develop it, maintain it, is there a business model for it, who's going to pay for this. Can you grow a business?"

The relationship between real-world applicability and personal practice is a theme that was mentioned by many of the participants. John perceived a tension between his desire to work with technology himself and his interest in real-world commercial applications, which he felt would be better served by professionals. This tension may have been aggravated by the personal fabrication approach, which forced participants to decide on and execute a specific project. Had John been able to spend more time tinkering — developing his own skills and exploring potential real-world applications — he may have felt more alignment between these two activities. For him, personal fabrication should probably come later, after his explorations have led him to a concrete project that he wished to take to the next level.

Trajectory 3. Finding Meaning in the Process Itself

For a third group of participants, Ernesto and Brooke, the workshop was an opportunity to derive meaning from the process itself, rather than through a specific project focus. Both made simple modifications to one of the example circuits without a specific conceptual idea. This gave them an excuse to experiment with the technology and gain familiarity with its capabilities. For Brooke, this technical capability was something she was interested in adding to her art practice. For Ernesto, it was an opportunity to understand the possibilities of electronic products and begin to imagine new products for particular situations or problems. Compared with Tyler, Nathan, and Akash, these participants developed fewer new technical concepts and practices — but rather changed their perspective on the relevance of electronic circuit design and fabrication to their interests and lives.

Ernesto

Ernesto, 37, is from Colombia. He studied industrial engineering and then worked for a number of years in real estate, earning a degree in economics as well. He then got a master's in business intelligence and now works as a business analyst.

Ernesto did some programming in C++ as part his industrial engineering studies and has experimented some with HTML and CSS. He came into the class with little experience in electronics and few specific ideas about what he was hoping to make. As he put it in the pre-class survey, he was there "To acquire knowledge on this area. To explore new avenues for entrepreneurial purposes."

Ernesto deliberately chose a constrained project to work on. As he put it in the closing discussion: "I really want to be able to see the whole process from the beginning to the end so I picked a very simple project." He worked from the display example, adding a connector for a current sensor available on SparkFun. Because we couldn't find documentation of the pin out of the sensor, Ernesto included additional connectors for jumper wires. That way, when we figured out the pin out, we could connect the appropriate pins of the sensor to the rest of the circuit.

Working from the temperature sensing example I provided, Ernesto was able to read data from his sensor and log into the SparkFun data service. This required changing the code slightly, to average multiple readings, as the values from the sensor changed rapidly. Once his circuit was working, Ernesto used it to measure the current draw of Tyler's project. They observed spikes in the current when the GSM module on Tyler's circuit communicated with the network.

Ernesto described learning about the pieces that go together into making an electronic product, stating:

> "How all the pieces interact, Eagle - you got two different files, you send them to Canada. They send you something back. You get sensors from Adafruit. you have SparkPlug [SparkFun]. Like how all those pieces interact together just to be able to control a physical device from code." (closing discussion)

He was also surprised to find some aspects of the process easier than he expected:

> "Using code to create an output on the physical world. I thought it was much more complex. Or I probably never thought about how complex it could be so I guess that it was not ... as complex as I would have imagined at one point. Especially connecting to the internet. Probably the advances of technology have made it easier but- or- Arduino itself made it easier, but I was not expecting it to be so straightforward in the

> sense that with just one, no antenna, cause I was expecting to find an antenna on the wifi connector, it was less chaotic than I had expected before." (closing discussion)

Still, he questioned his ability to make a project on his own:

> "What I learned too is that in order for me to be able to jump into one of these projects, I don't think that I have the whole knowledge. I would need some electrical engineering knowledge to know what resistors, what capacitors, how to create.. how to design the board, programming skills, and I would not say that that's frustrating, cause that's expected but yeah, I learned that I would need to have more knowledge in order to jump into one of these projects or to build something meaningful." (closing discussion)

And later:

> "I think that I would still need someone like you or like you or probably like Tyler to be able to go over the roadblocks that may come from difficult steps on the project."

So in contrast to some of the other participants, like Tyler, Ernesto didn't come out of the workshop confident in his ability to do future work in electronics. He did, however, talk enthusiastically about the benefits of his experience in the workshop:

> "I have to say that now that I- every single time that I encounter a situation, I am thinking how these- every single time that I'm seeing a problem, I'm thinking about 'oh, how can this solve this situation?'. So, at least, that's how the knowledge has worked for me." (one-on-one conversation)

This change in perspective was also captured in his responses to some of the survey questions. On the pre-workshop survey, he indicated that he strongly agreed with the statement "electronic products seem like magic" and strongly disagreed with the statement "I would have something to contribute in a conversation about the design of electronic products." On the post-workshop survey, his opinions on these two questions were reversed.

He also saw more practical opportunities for electronics:

> "As the class progressed, I started seeing a world of possibilities. If I would have been in Colombia, I would have just jumped straight into setting a business of some- because I see a huge market there, a necessity for these products. It wouldn't have been difficult at all to start a business there." (one-on-one conversation)

Finally, he reflected on the value of electronics for society more generally:

> "It empowers you, it expands your knowledge, it's- I also think that, honestly, I think that something like this has to be taught at some level of school... I think that very few people will take advantage of creating devices that will improve their quality of life or that will benefit directly from creating them. But I think ... if everyone knew how this works, it would help ... to innovate more quickly as more people can participate ... can bring their own thoughts into different uses for the devices." (one-on-one conversation)

Overall, then, the class helped Ernesto to understand the value of making electronics, both personally and for society as a whole. While he may not feel confident in his ability to complete a project on his own, he has a new appreciation for how electronics are made, what they might be used for, and why these things are important to understand. More importantly, this experience made him feel that he could be an empowered and useful participants in the process of defining and creating an electronic product. This is a significant shift in his perceptions of his own abilities, independent of the specific technical skills he did or did not gain in the workshop.

Trajectory 4. Struggling with Technological Agency

The final participant in the workshop, Nicole, struggled, finding difficulty in developing a sense of agency with respect to the technologies and activities covered.

Nicole

Nicole, 29, works in user experience at a financial institution, working on usability testing for digital product development. She has a background in art and advertising. In the opening class survey, she discussed her background and interest in electronics and programming:

> "I'd really love to be able to make the things I want! I get really frustrated with web design and coding attempt. I have lots of ideas, though! I took a creative visualizations course through Harvard last spring... and got my hands on a lot of things I don't know what to do with-arduinos, bread boards, etc. These things fascinate me and I have no idea what to do with them or how to teach myself how to make something. Im so happy to be here!"

After her class last spring, however, Nicole found it difficult to continue with electronics on her own.:

> "I take a lot of books and I kind of try to read about it, but since that spring, no I haven't done anything but like hold all the little things that I've accumulated and not know what to do with them."

In addition, work and other hobbies left little time for electronics and "I just figured I would eventually take maybe another class or something."

Nicole struggled in the class. She would try out the activities but tended to stop before finishing. In the fourth session, with assistance she completed a design for a circuit board that included an LED and a connector for a resistive sensor. The following week, she struggled with soldering the components to her board, misaligning the microcontroller on her initial attempt. On a trying again with a second board, her soldering improved but she stopped before completing the process. Nicole missed the last session of the class. Later she explained that this was because she didn't want to slow down the rest of the participants: "The rest of the group was moving so fast, I didn't want to take up space and, you know, put another- sometimes it's hard to manage a classroom like that."

Reflecting on the activities in the class, Nicole explained that it helped her to understand the relationship between programming and the physical world:

> "I really liked that, especially that first day, where we used the wires and LED and that makes it a very real- connects- just makes the understanding of the code to this, you know, to the physical."

Designing circuit boards in Eagle also helped her to demystify the process by which technology is produced:

> "I may not know the- how to build my own board but going to Eagle and seeing the connection, that was really cool, I think it was really helpful. I like that a lot. [Which connection?] Like the computer to the physical. Cause it can seem like 'oh, that's magic' or 'that's MIT wizardry' or something but, it just made- it explained things."

She also discussed her appreciation in getting a better sense of the technological possibilities.

> "I really liked seeing what everyone else made- just to see 'how do you do that?' you know and that was- kind of opens a different perspective on, you know, this is how this is made ... I feel now that there's this knowledge that's available to me in a way that before I didn't know how to get to it."

When it came to making a project herself, though, Nicole expressed more doubts.

> “Nothing I make ever works. So that's- I mean I'm just not- I can try really hard to be exact about things but there's always a problem. I'm like not a math person, you know. I feel like y'all can do something and then- I don't know, it just doesn't ever work out for me.”

These doubts carried over to Nicole’s thinking about future possibilities:

> “I feel like if I watched a video or something, I could follow it. The most simple things... I don't know what I could make. I don't think I could make too much. But I could see someone else and be like 'oh, okay, that's that' and that's fine with me.”

These comments suggest that the class may have discouraged Nicole from engaging in electronics. Before starting the class, she had expressed a strong interest in learning to make things for herself, saying on the opening survey “I want to be confidently familiar so I can go forward and teach myself more and really make things.” When she encountered difficulties and obstacles within the class sessions, however, Nicole seemed to give up, even in situations in which she seemed (from my perspective) to be making progress.

One possible explanation for some of Nicole’s difficulties is that she came into the class lacking confidence in her technological abilities. The specific challenges she encountered may have reinforced this self-perception, leading her to give up on some of the activities. The fast-paced nature of the class may have exacerbated this issue by limiting the opportunities that Nicole had to gain familiarity and confidence with specific technologies before we moved on to the next activity.

Another possible interpretation of Nicole’s challenges is that this class may not have had the right focus to provide her with an appropriate experience. The emphasis on designing and assembling a circuit board created a high bar to making of a successful product. Participants had to complete a design on schedule in order to get their PCBs back on time. They had to solder many components onto the board before any of the circuit would function. A more incremental approach might have helped Nicole by allowing her to tinker with individual components, making progress at a more self-defined pace and creating a lower initial bar to making a functional circuit.

Concepts, Skills, Practices

To engage novices in working with technology, it's important to understand the concepts, skills, and practices they may develop in the process. This understanding can help educators in structuring and facilitating activities to develop the necessary concepts, skills, and practices. It can also provide potential participants with a preview of the abilities they may develop in the course of an educational activity. It's important to note that not all activities involving making things with technology contribute equally to participants' development. As pointed out in Chapter 2 (in "Electronic Kits"), it's possible for someone to assemble many pre-packaged kits without ever proceeding to the next level of understanding or activity. The challenges in this workshop, on the other hand, provided significant opportunities for participants to develop.

This section discusses the concepts, skills, and practices demonstrated by participants in the connected devices workshop. It discusses some new concepts and practices not found in prior work ([Brennan & Resnick 2012], [Qiu et al. 2013], and [Kafai et al. 2014]). It also adds the new category of skills, the physical abilities required to create with technology. The connected devices workshop also reinforces and provides some new takes on some of the concepts and practices discussed in those prior works. Some of these additions and modifications come from expanding the notion of computational thinking to include embedded computing and circuits, others from the specifics of the technologies and approaches used in the connected devices workshop. (Note that changes in perspective, another theme of the previously-cited papers, were discussed in the previous section on participants trajectories and therefore aren't included here.)

Specifically, this section looks at three concepts:
- Connection Between Code and Circuits
- Making Sense of Sensor Data
- Non-Determinism and Error Handling

Two skills:
- Soldering
- Schematic and PCB Layout

And eight practices:
- Project Scoping
- Abstracting and Modularizing
- Component Selection
- Integrating Multiple Sources
- Articulating Coherent Questions
- Being Incremental and Iterative

- Debugging
- Reusing and Remixing

Similar to my own experience with the DIY cellphone (discussed in Chapter 5), detailed knowledge of electronics theory hasn't proved to be an important pre-requisite for the making of electronic products. For example, while the connected devices rely on sophisticated computation and communications technology, it wasn't necessary to understand the theory behind them in order to make a working device. Instead, the electronic components encapsulate the theory; the core challenges relate to finding the right components and connecting them successfully. Of course, this process isn't always smooth or easy, but the difficulties encountered don't typically involve the electronics theory encapsulated by the component — e.g. debugging the CC3000's ability to connect to a wifi network doesn't require an understanding of the structure of the radio waves used by the 802.11 protocol.

Concept: Connection Between Code and Circuits

Kafai et al. [2014] list input / output and analog / digital as among the concepts developed by participants. Participants in the connected devices workshop articulated a more fundamental principle, the general understanding of a relationship between code and circuits in the real world. As Ernesto put it in response to a question about what was easier than expected:

> "Using code to create an output on the physical world. I thought it was much more complex. Or I probably never thought about how complex it could be so I guess that it was not … as complex as I would have imagined at one point." (closing discussion)

Luisa, too, discussed the connection between virtual code and data and the physical world as something she began to understand through the class:

> "I think last class was the first class that I started to have a certain sense of how to connect the hardware, the hard things that we are touching, with data that is virtual." (session three)

This connection between code and physical circuitry is something that's easy to take for granted after you've become comfortable with embedded programming — but it's an essential part of the practice. This isn't necessarily about the specific technical knowledge required to do so (like the particular syntax of an Arduino program) and it doesn't mean understanding the internals of a microcontroller — simply being comfortable with the notion that text typed onto a computer screen or data downloaded from the internet can have an effect on the physical world.

In the connected devices workshop, for example, this understanding seemed to emerge from the experience of using Arduino code to control LEDs or other electronic components. Writing this code for oneself, rather than just using existing examples, seemed to have been crucial in developing this understanding. If the whole process seems like magic (as it can when you just use an existing example), it's less likely to elicit the kind of engagement with the technology that's required to get someone into the idea of controlling the world with code.

Concept: Making Sense of Sensor Data

Simply connecting a sensor to a microcontroller and reading data from it often isn't sufficient to make sense of the physical phenomenon of interest. This often requires both additional reasoning about which sensors to use and how to physically configure them — as well as additional processing of the resulting sensor data. John discussed his growing awareness of this concept:

> "Understanding a little bit more about that physical- I guess understanding how a sensor operates and how that's translated into- how that translates back into the electrical circuit and how do you- what are the mathematical algorithms that you're creating to derive intelligence out of that, to actually to provide some decision support eventually or some useful information to the user." (post-workshop interview)

John's interest in a real-world application (snow detection) played a crucial role in developing this sophisticated understanding of sensing. It forced him to consider which sensors would be most useful for this application, how they might be deployed, and how to process the resulting data.

In general, understanding how to extract higher-level meaning from low-level sensor data is crucial to understanding the potential applications for which a sensor can be used. Nathan, for example, went back-and-forth with me to develop his understanding of which sensors were suitable for which kinds of projects. It was clear to him that an accelerometer could be used to measure some kind of motion, but not necessarily what inferences could be drawn from its readings. With his more advanced skills, Tyler was able to seek out information on using an accelerometer to measure the intensity of waves.

The importance of understanding the potential higher-level uses of a particular set of low-level technological capabilities applies to more than just sensors. For example, an ATmega microcontroller and the CC3000 wifi module can communicate sensor data, weather conditions, the text of email, and other simple information but not audio, images, or complex web pages. This type of high-level understanding of the capabilities of a component is often neglected in descriptions or tutorials, which tend to focus more specifically on how to connect and extract data from a component, not how to interpret or use that data.

Concept: Non-Determinism and Error Handling

The use of networking introduces many more opportunities for latency or failure than those found in more basic embedded systems, such as those discussed in [Qiu et al. 2013] and [Kafai et al. 2014]. With many components, once the circuit is correct and the soldering good, simple code should work reliably without need for error handling. With network, on the other hand, even if the physical component (e.g. wifi module) is connected correctly, the network might be down, DHCP registration might fail, the remote server may fail to respond, data may be in a different format that expected, etc. Or, the data arrive may arrive successfully but late, creating problems if the circuit is supposed to be doing other things in the meantime, like reading sensors or responding to user input. All of this requires the programmer to consider the different errors or delays that might happen, to decide on an appropriate response in each situation, and to turn these decisions into code. Even in cases when this error-handling is provided in an example, it complicates the code and makes it more difficult for people to understand what's going on. These difficulties were hinted at in one of Nathan's comments:

> "I think the Temboo piece, that was a little tricky because it sort of seems like it's all being done behind the scenes. It's doing something one step away from magic and it just works or it doesn't work and it's almost like it would be nice to see a little more granularity there." (Nathan, closing discussion)

This comments highlights the difficulties in understanding caused by unreliable systems. I would argue that there's as much or more "magic" (complexity hidden behind opaque abstractions) in, for example, microcontrollers — but they're far more deterministic, so it's not as necessary to understand how they work.

Tyler wrestled with indeterminacy in his use of the cellular GSM network:

> "Honestly, it was like the GSM interactions and getting something that could handle- I don't know, it just seemed like there were a lot more hangups when you're turning a card off and turning it back on and trying to connect to the network and just trying to go through and debug stuff. And I'm making progress on that, the forums are pretty helpful. That was the single frustrating thing. Cause, I dunno, it seemed like it's a little more glitchy than wifi." (post-workshop interview)

Here, we see a more sophisticated understanding. Not only does he understand errors and indeterminacy as an important aspect of working with technology, he's also distinguishing between different technologies on the basis of this factor.

Skill: Soldering

Successfully creating an electronic product using personal fabrication requires an understanding of computational concepts and practices, but it also needs certain manual skills. In the connected devices workshop, soldering of surface-mount components was a key part of the process. An individual's degree of proficiency in this skill has important implications for other aspects of the process, like the range of components that they can use, the amount of debugging they'll need to do, and the speed with which they'll be able to try out and iterate on designs. It's important to note that these skills aren't necessarily correlated with someone's conceptual understanding. For instance, Luisa excelled at the soldering but struggled with other aspects of the process, while Nathan had more trouble with the soldering despite his better understanding of the technology. Also worth noting is the fact that skills like soldering are not defined exclusively by an individual's capabilities but are greatly influenced by the nature of the tools they use and the materials they're manipulating. Even many expert solderers would struggle to assemble a fine-pitch surface-mount component with a cheap, fat-tipped soldering iron.

Skill: Schematic and PCB Layout

Designing a PCB in Eagle or other CAD software requires knowledge of the components and understanding of electronics, of course, but it also involves many aspects that are better considered as a manual skill rather than a conceptual understanding. Routing traces is the clearest example, although in the connected devices workshop all the participants relied on the auto-router instead of doing this manually. Component placement, however, was something that participants did engage with and that some had more success with than others. Even laying out a schematic, where the connections have only logical, not physical meaning, involves aspect of manual skill in creating an organized and understandable arrangement of components. The classification of these activities as a manual skill is analogous to much of the discussion in *Abstracting Craft* on CAD as a medium.

Practice: Project Scoping

The process of matching one's personal interests with the possibilities of technology (and one's skill) is a complex and subjective one. Nathan described his process in detail:

> "My problem was I didn't have a problem I wanted to solve and so I was like I'm just going to try to eyeball something and come up with something out of the air. And, David and I bounced a few ideas back and forth. And I settled on one and I'm happy with it and I almost got it working but then part of the issue was scale. And I thought of a few projects that I would do and I was thinking 'yeah that would be really interesting and would take two years to put together so I'm going to put that one aside for now.' And then others, were like 'oh, yeah, that's a great idea and I can go buy that at Walmart today for five dollars so I'm not going to do that either.' And I think it was just scoping down to something that I thought was interesting and executable in the time we had with my limited knowledge of the tools we had. So that's kind of- and I don't know how you solve that problem, right, cause even coming in you knew you had to work on a project but you didn't know what tools you were going to use and to say you're going to connect a device to the internet, what does that mean? Am I going to connect some time of industrial application to the internet? I just had no idea. So it was, for me, I think that was honestly the most challenging was just scoping down something that I thought I could do." (closing discussion)

In this quote, Nathan mentions multiple complex criteria: his ideas and interests, technical feasibility, lack of commercial equivalents, and appropriateness to the workshop. Balancing these criteria to find a project involved multiple rounds of back and forth between me and him. In each of these exchanges (as with other participants), I was using my knowledge of the technological possibilities to try to suggest or comment on possibilities of interest. With more experience, I would expect the workshop participants to get better at scoping potential projects, but it's worth noting that this practice is one that even professional engineers struggle with.

Practice: Abstracting and Modularizing

Brennan and Resnick [2012] discuss this practice with respect to code, but in the connected devices workshop it found different expressions with respect to circuits. Specifically, because the workshop's personal fabrication approach dealt directly with individual components — in PCB design, soldering, and debugging — the process of abstraction occurred more in participants' thinking and less in concrete structures. That is, abstraction took the form of being able to think about the higher-level behavior or use of a group of individual electronic components — rather than creating a new structure or module containing that group. Nathan expresses this practice in terms of focusing on the big picture:

> "And someone said that they want to get more and more into the details and I'm sort of- I appreciate that desire and I agree

> with it, but also I'm sort of like I want to solve a problem and I care about the minutiae on some level but ultimately I want to solve some problem and it looks like I have enough of the tools that I can, you know, if there's some little thing that I don't understand I can figure it out, or just drive over it really. I was thinking about- I had some circuit last week and the resistor- the LED was in backwards cause I just completely didn't even think about that. That's something that's just a little speed bump along the way of getting the project done. So it's just nice to- like, three weeks ago, if you had said 'here's a problem, how would you solve it?' I would have just said 'I'm going to go drink a beer' now it's okay, let's break down the problem because I'm at least a little bit more comfortable with all the pieces." (Nathan, session three)

As with code, it's sometimes necessary to open up an abstraction and understand and fix what's happening within it, as Nathan discusses here with respect to having his LED backwards. But, having solved that problem, he's able to return to thinking about his project at a higher level, abstracting away the details of the LED.

This need to view circuits at different levels of abstractions occurs in many situations. For example, placing a pull-up resistor on the reset line (10K ohms) of a microcontroller or a decoupling capacitors (0.1 uF) between power and ground requires understanding enough to know what that means, but not necessarily exactly what it does or why it's necessary. Communicating with a component using SPI means that you need to know enough to connect it to the MISO, MOSI, and SCK pins of the microcontroller, but not the details of the structure of the protocol. Again, much of this is about being comfortable with the vocabulary and form of technical documentation, as opposed to having a detailed knowledge of electronics theory.

Some workshop participants struggled to work at a higher level of abstraction. As Luisa put it:

> "What I found challenging was to understand what are the components that have to be paired with the basic components that you are buying. Like how many resistors, how many capacitors, where is the crystal, why one and not two or not three. Why do I have- the reasoning behind the selection of all these things beyond the sensors that we buy directly." (closing discussion)

While Luisa is expressing a concern about the details of individual circuits, I think a key part of her difficulty is in being able to work at different levels of abstraction. Being able to copy pieces of existing circuits or incorporate details provided by an expert, without necessarily understanding all the pieces, is an important practice in its own right, independent of an understanding of the particular pieces. As Nathan put it:

> "I'm still slightly confused by our little diagram here about the push-button and the resistor. I kind of get it; I get that it works. I could look at something and copy it and solve my problem, I think, but looking at a blank page and trying to do it would be problematic." (Nathan, closing discussion)

Here, even though he doesn't understand all the details, he's comfortable enough with his ability to abstract to feel confident that he could successfully incorporate this example into a larger circuit.

Practice: Component Selection

Core to the making of an electronic product is selecting and making use of the components it contains. Selecting from the multitude of available parts is a multi-faceted and ill-defined problem. There are many criteria: cost, quality of documentation, availability of code or circuit examples, capabilities, reliability, ease-of-assembly, etc. It's not always clear how components rank on individual metrics nor how to weigh these factors against each other.

The connected devices workshop provided participants with two ways of developing their working knowledge of components. Part of the process is working with a set of components that has been curated by the workshop facilitator (in this case, me). This exposes participants to a set of purportedly good component choices that they can draw on in the future. Perhaps more importantly, though, workshops give participants an opportunity to go through the process themselves. In the connected devices workshop, most of the participants had to find an appropriate sensor or actuator specific to their project. Tyler discussed his strategy for selecting a temperature sensor:

> "What do I want to do with this? I want to measure temperature. Go on Adafruit, go on SparkFun, all the different ways of measuring temperature. Do I use a thermocouple? Do I use a thermistor? Read a bunch of reviews. See people- read the tutorials. Okay, thermistor it is; it's waterproof, it's simple. I don't have to do any weird calibration or logic shifting." (Tyler, post-workshop interview)

This shows his consideration of various factors: availability (from a familiar source), ease of interfacing, and robustness. It also showcases some of his processes for understanding these factors: reviewing e-commerce sites as a source of possible parts and using online reviews to understand which parts may best serve his purposes.

The practice of component selection is essential to making use of one of the key advantages of a personal fabrication approach: the ability to work with a variety of components, outside the constraints of any particular toolkit or system. It's also a practice whose development is facilitated by a process in which people need to seek out at least some relevant components on their own, rather than working entirely within a pre-selected set of parts.

Practice: Integrating Multiple Sources

When working outside of a curated set of technologies (like an electronics toolkit or a single-purpose kit), it's often necessary to integrate information from multiple sources. For example, in trying to work with a humidity sensor on her PCB, Luisa was attempting to integrate information from the sensor's datasheet, from the documentation for its Arduino library, from her PCB's schematic, and from a table listing the Arduino pin mapping for the microcontroller she was using. Even if someone understands each of this pieces, tying them all together becomes its own challenge. It often requires multiple steps of translation, each of which may have its own vocabulary. For example, the datasheet of Luisa's temperature sensor referred to an SPI interface, the microcontroller pins in the schematic were labelled with the names of the individual SPI pins (MISO, MOSI, SCK), and the Arduino library needed Arduino pin numbers. This points out the importance of helping people navigate technical documentation even if they don't, and don't have to, understand most of what it contains.

Similar difficulties arise in selecting technologies for use in a project. Many different sources of information must be understood in order to determine whether different parts will work together — e.g. whether the Arduino library for a given component will fit within the flash memory of a given microcontroller. Even the basics processes involved in the personal fabrication of an electronic product can involve multiple sources. As Ernesto put it:

> "I gained a lot of knowledge. How all the pieces interact: Eagle- you got two different files, you send them to Canada, they send you something back. You get sensors from Adafruit. You have Sparkplug [SparkFun]. Like how all those pieces interact together just to be able to control a physical device from code." (closing discussion)

As this comment points out, simply knowing what kind of different resources are required — like sensors and component suppliers, PCB vendors and CAD software — is a crucial part of understanding how to make an electronic device. Some of this need to integrate multiple sources of information can be avoided by getting all of the different technologies from a single source or platform, but this greatly restricts the diversity of technologies that can be used.

Practice: Articulating Coherent Questions

Being able to ask for help in a way that can yield useful information is important technological practice, especially for independent work. In contexts when others can't see what you're doing in person, or clarify your meaning in a live conversation, it's important to be able to unambiguously describe your goals and current status. Luisa expressed doubts as to her ability to do this successfully:

> "But then, the question is I feel like I need more help on that, I don't know even if within the community, Arduino community, I'm going to utilize the right terms to express what I'm looking for, you know." (closing discussion)

As she says, you don't always have the right words to describe the things you're trying to do — and, without them, it's difficult to ask for help. For Nathan, the workshop increased his confidence in this practice:

> "Now I think I could send someone an email and say 'hey, I did XYZ and it didn't work' vs. like 'Hi, I have no idea what's going on and I need some help.'" (Nathan, discussion)

Having reached a certain level of technical understanding, he's now better placed to progress on particular projects because he's more able to get help. Before someone gets to this point, in-person support can be crucial, a point returned to in the "Provide Support from People and Support from Technology" sub-section of Chapter 7.

Practice: Being Incremental and Iterative

This practice, from [Brennan & Resnick 2012], was applied by workshop participants in developing their code. The workshop, however, also highlights the difficulty of being incremental and iterative with respect to circuits when taking a personal fabrication approach.

Tyler not only engaged in iterative coding practices but was able to articulate his use of them:

> "There's just so much example code out there and kind of feeling like, woah, here's this thing with 500 lines that someone just posted online somewhere and they call three or four different libraries and there's just so much, where do I even start with this? And having to learn- just find the simplest thing you can. Find the thing you can understand, find the next level of complexity up from that without any extra bells and whistles and try to figure out that. And then once you have that figured out, okay, take your next step. That's kind of slow and frustrating, though, and I wanted to go five steps ahead. So that was difficult, to manage that gradual introduction and

> sometimes it's really tough to find those incremental pieces." (closing discussion)

This use of increasingly complex examples, chosen according to his current level of conceptual understanding, is a sophisticated form of an incremental approach to technology development.

Some of the workshop participants talked about finding this type of incremental or trial-and-error approach easier with code than with their circuits. As Nathan put it:

> "I was sort of less concerned with the coding because that's to some extent a trial and error issue." (closing discussion)

With code, the time between one version and the next can be minutes or even less. With PCB design and fabrication, on the other hand, it may be a week or more before you can go from one revision to the next. While this allows for more iteration than mass production, it's not nearly as amenable to incremental approaches as code, electronics toolkits, or breadboard prototyping. In the workshop, participants had to decide on the composition and layout of their PCB in week four and retain it through week six, eliminating most opportunities for iterating on their circuits.

Practice: Debugging

Several of the workshop participants talked specifically about the difficulty of debugging their circuits:

> "My biggest stumbling block was not even debugging the code but just debugging my board.... Honestly my first reaction when I couldn't get this to upload was 'screw it, I've got another one, I'll start over' because I don't where to begin, I don't know where to step into the process." (Nathan, closing discussion)

> "Debugging the board is also more difficult than debugging code." (Akash, closing discussion)

Some of this may have been simply been due to the previous experience that Nathan and, especially, Akash (a professional programmer) have with programming. But it also reflects some of the specifics of the process of debugging circuits. For example, it's more difficult to try to figure out what's going on by making small changes and seeing what happens — as they may cause further damage to the circuit. It also may not be obvious which parts of a circuit are involved in which operations, and so it's not necessarily clear what connections to check with a multimeter, for example. This difficulty was exacerbated by the fact that participants soldered all the components to their boards before attempting to test any of them, meaning that when something went wrong, they couldn't isolate the problem to recent changes. Finally, some aspects of a circuit are just difficult to inspect — the location of a short circuit between power and ground, for example, or whether or not a particular component is damaged.

Software is complex, but it's often possible to dig into the code and isolate a problem. In electronics, you can easily run into a literal black box that can't be opened. As a result debugging circuits (like debugging other opaque systems) relies on a mental model of what's happening inside the components and what could cause them not to work as expected. This mental model needs to be developed through experience with circuits and, especially, experience debugging them.

Practice: Reusing and Remixing

The structure of the connected devices workshop emphasized the reuse and remixing of existing examples, a practice discussed in [Brennan & Resnick 2012] and one readily adopted by the participants. Nathan talked about his use of example code:

> "I knew that what I was trying to put together, there were three example files that did the three things I wanted to do, so I just had to figure out how to put them together." (closing discussion)

While Akash talked about drawing on example circuit design files:

> "Yeah, I mean, the designing the boards seemed a lot easier than I anticipated, I guess because we started with some template and just like made some modifications." (Akash, closing discussion)

This illustrates one advantage of a personal fabrication approach, that someone can directly build on and modify an example for a circuit by downloading its digital file rather than, say, having to recreate a circuit on a breadboard.

Tyler talked about the relationship between his use of example code and his learning process:

> "Before I felt like I've been finding and copying example code from online without fully understanding it. It was kind of- not cheating but lazy. And- someone said something in the class that was 'no, that's how you learn. is you go and you use it.' And maybe part of my block before was that I felt like if I'm going to copy this I have to make sure I understand every single reference or instructions within it. Well, you don't, if it goes in and if it works, it works, you're probably going to learn all these different details as you're editing it in the future and you drop a colon or a parenthesis or something and all of a sudden it doesn't work and then you have to figure it out." (Tyler, post-workshop interview)

With both the code and circuits, providing workshop participants with examples that they modified was an important scaffold for learning. Had we started from scratch, building a project would have taken much longer and required understanding many more pieces all at once. Starting from examples allowed participants to concentrate on specific parts at different times and understand or change them before moving on to the next one. On the other hand, if they hadn't changed the examples at all, it's doubtful they would have learned nearly as much.

Value of DIY Electronics & Personal Fabrication

Participants in the connected devices saw a clear role for DIY electronics generally and discussed multiple values it could serve. Their attitudes towards the particulars of the personal fabrication approach pursued in the workshop were more nuanced, with many seeing a place for this kind of digital design and fabrication but a more limited one. This sub-section is divided into three sections: (1) on the value of the process of making DIY electronics, (2) on the value of the products of this process, and (3) on the value of the knowledge generated by the process. The first of these (the value of the process) stems primarily from DIY electronics generally, while the latter two relate more specifically to a personal fabrication approach.

The Value of the Process (Opportunities for Creativity)

Many participants stressed the value and meaning of electronics-making activities for their own sake (independent of any particular outcomes):

> "I heard the word 'hobby' many times. You know, if I think about my grandfather or my father, they liked to play with little trains and mount these little trains and the trains were replaced by Lego and now we're not talking about Lego any more, now there's like tons of Lego formats. I think this could be for me it could be the next step of the Lego because it becomes iterative [interactive?]. My son was telling me 'oh, I want to make a robot'. Great! Let's buy some things and let's make a robot. Why not? Cause people like to see things moving. It's always attracting to see things that are interactive. Why not this? So I this as a natural sequence of these play structures." (Luisa, closing discussion)

Brooke talked about the value of custom electronics for art:

> "I think art- these little crazy art projects that have- that need unique solutions." (Brooke, closing discussion)

John contemplated continue to pursue his project as a hobby:

> “So where would I go? There's the hobby thing and playing with SAM and kind of feeling that you're in your little fiefdom of your garage and you're trying to save the world.” (John)

Nathan, although he came into the class with an engineering, problem-solving mindset, found enjoyment in the act of making itself:

> “This was kind of neat to put together like a Lego set but there was also this, 'I see this nebulous problem, how do I use these tools to solve that problem.' - that engineering aspect to it. But if Brooke calls me up one night and says, 'hey I need help soldering something', okay, I'll go help because that was enjoyable in and of itself.” (Nathan)

This perspective of electronics as hobby places an emphasis on the activity and the making process itself. That might suggest that it’s actually more suited to the tinker-ability and experimentation enabled by toolkits and breadboards, as opposed to the planning and up-front design required by a personal fabrication approach.

The Value of the Products (Compared to Commercial Products)

Many of the participants struggled with identifying the value of DIY electronic products, especially compared with the wide range of existing commercial products.

Ernesto was perhaps the exception, seeing many possibilities for the emerging countries, like his homeland of Colombia:

> “As the class progressed, I started seeing a world of possibilities. If I would have been in Colombia, I would have just jumped straight into setting a business of some- because I see a huge market there, a necessity for these products. It wouldn't have been difficult at all to start a business there. That was my first thought.” (one-on-one conversation)

In general, however, he was skeptical of the potential for people to make devices for use in their own lives:

> “I think, on a first level. I think that very few people will take advantage of creating devices that will improve their quality of life or that will benefit directly from creating them.” (one-on-one conversation)

Tyler emphasized the difficulty in the electronics domain of creating DIY products that could compete with commercial alternatives:

> “I don't think I'm going to be building the next iPhone or making my own- like some people learn how to make their own beer, and they never drink any beer cause they just 'I'm going to drink my own beer cause it's the best and I'm the best'- I don't think that's going to happen with this.” (closing discussion)

On the other hand, he saw opportunities to create products that simply don’t exist:

> “Some of the stuff just isn't out there, like for swimming there's really not that great electronics. There's GPS loggers but there's no way to actually get information back to the user who's swimming so I've been thinking about - once I figure out waterproofing in salt-water, which is a big challenge on the buoy - what if I could have like a heading indicator on my goggles. That's just a thing that no one makes.” (post-workshop interview)

John found his interest in developing new products tempered by a consideration of the factors involved:

> “I will say this class has helped me conceive a way of thinking about a proof-of-concept. I suppose if I had more time on the internet-of-things and understanding a way of seeing that future that I'd even be more confident of saying actually we can create our own proof-of-concept here. But then you've got that business block- or that entrepreneurship block of like okay, well, okay you can make a set of gadgets here for this building; who's gonna design, develop it, maintain it, is there a business model for it, who's going to pay for this. Can you grow a business?” (John, one-on-one conversation)

This comment highlights an important issue: the more emphasis one places on the product potential of a DIY project, the more like a business (and the less of a DIY activity) it becomes. John might develop a proof-of-concept as a DIY or hobbyist activity. But growing a business around it would involve very different questions and activities. This points out the space that personal fabrication exists within. The product has to be useful enough to justify its fabrication — but, if it turns into a business, it will take on a very different character. Nathan, for one, envisions products within this space:

> “I'm not going to go and now build a connected home solution because I can buy one off the shelf. But certainly, this opens up the realm of interesting solutions for other problems that only I think hey, there's a problem here. I'm going to go- maybe it takes me six months to put together but it's still an interesting project to approach.” (Nathan, one-on-one conversation)

Despite questioning the value of personal fabrication as a replacement for existing commercial products, participants saw it as a realistic practice to include in their hobbyist activities:

> "The Eagle was really interesting to learn and I'm sort of like, you know for a hundred bucks if I think of something I want to do, maybe I go down that road. Cause it was kind of interesting and if anything for me it's an interesting conversation piece." (Nathan, closing discussion)

Brooke, too, talked about cost as a consideration, as well as the quantity of circuits she'd need:

> "I think though if I had a project that needed a lot of boards like the same boards over and over, I would get it to this stage. But I think if I just had a one-off, I don't think I would- I mean, I don't really know that much about pricing or if it's worth it to get one board done. I probably wouldn't pay $80 for a board that I only needed- like I knew that I was only going to need one of in my whole life. But if I had a reason to, I would definitely go down that route." (Brooke, closing discussion)

Tyler was more explicit about some of the potential advantages of designing a PCB, specifically being able to work with a variety of components and to create an optimized, robust form factor:

> "I like the idea of it because often times you're just limited to whatever you can buy: what they decided to put on a breakout board at Adafruit, or SparkFun, or Pololu. This kind of opens it up a little more widely. I doubt I'll be going out and picking, spec'ing my own microcontrollers but- Or even things as simple as putting my package together of the accelerometer, the terminal blocks for the thermometers or thermistors and the power supply device, the buck-boost regulator. Being able to just make it all compact and fit neatly, as opposed to I have this one little RadioShack perma-breadboard that I soldered stuff on and then I'm going to have little jumper wires coming over. I really liked that, that was neat. So it's probably faster to simply buy a breadboard that's been stuffed and pre-assembled especially when it's not just the device but there's all kinds of other crystals and caps and resistors on there but I think that when it's time to really optimize or make something good, it's nice to be able to have that skill to think about laying out a board for yourself." (Tyler, one-on-one conversation)

The comments from Brooke and Tyler emphasize that PCB design is not necessarily something that makes sense at all stages in a project, or for all projects. Nathan expressed a similar sentiment: "I think I would end up trying to mock something out on an Arduino before I committed everything else." (closing discussion) Tyler, too, described a similar perspective:

> "But, yeah, I mean, if I'm making a one-off thing and none of my friends are ever going to use it, I'll probably- I might just make it but I wouldn't do a PCB, I'd do it on a solderable breadboard. If I was making something and a bunch of people around me showed interest, yeah, maybe I'd go ahead and whip up Eagle and start doing some things." (Tyler, one-on-one conversation)

For these participants, PCB design is something that makes sense once they have a clear idea for a device and a need for more than one of them. It's not the obvious way of exploring possibilities or simply experimenting with electronics. Still, now that they've gone through the process of designing a circuit board, they consider it as part of their repertoire of possible approaches. They see personal fabrication as an approach that can make sense in a personal or hobbyist context, even if it can't be justified from a purely economic perspective. This echoes the values seen in other areas of DIY and hobby activities (as discussed in Chapter 2), where practitioners value practical applications of their activities even in cases where there are cheaper commercial alternatives.

This question, of when personal fabrication provides practical value in making products, is one that I return to in the next chapter. Next, though, I discuss a value that was more clearly endorsed and articulated by the workshop participants: personal fabrication as a means of understanding and demystifying electronic products.

The Value of the Knowledge (Demystifying Existing Technology)

Multiple participants mentioned that the workshop had demystified the functionality of electronic products. The workshop's approach of designing and assembling PCBs is particularly well-suited to this type of demystification because of its use of processes and parts analogous to those found in commercial products. In contrast, when working with pre-assembled circuit boards, even hobbyist modules like an Arduino board, it's easy to see them as magic. As Akash put it:

> "One thing that I still don't understand, that I'd like to learn more about, I mean- this Arduino board is still kind like a black box to me." (session three)

This comment highlights the fact that the use of electronics toolkits often substitutes, for one black box (an electronic product), other equally opaque ones (a toolkit's modules). Because these toolkit modules are not the parts from which commercial electronic products are constructed, learning how to work with them doesn't translate to an understanding of how commercial products are made. In the workshop's personal fabrication approach, on the other hand, participants created electronic devices from digital PCB designs and industrial electronic components, a process much closer to that found in industry. This led many participants to talk about gaining a better understanding of the way that commercial electronic devices are put together.

Tyler discussed the example of his clothes dryer:

> "I think it demystified it a little bit. Like you mentioned appliances. I was working on my dryer, my clothes dryer, and there's this little circuit board in the back. I open it up- I wasn't- I was fixing the roller belt or something, but I was looking at it going 'well, I guess it makes sense that that was in there.' No idea what it's doing, probably something related to the control panel of the timer and the heat setting, but just to be able to look at something, crack it open and be like 'okay well I think that's what that is, I think that's where the power comes in'. It's kind of like, it's almost like literacy for electronic devices. And I like that." (Tyler, closing discussion)

Akash expressed a similar sentiment:

> "Having gone through this process, looking at other electronic devices, I think I will have a better understanding- or at least try to think about how that thing works, right, and what's going on inside it." (Akash, closing discussion)

John discusses the scope of the knowledge:

> "Yeah, I think at some systems level or sub-systems level I think I've a better understanding now of how things work. Maybe down at the minute level or- you know- do I still understand how this sensor works? I probably don't." (closing discussion)

That is, he's gained an understanding of how devices can be composed from components even without a detailed knowledge of how those components work internally.

Brooke referred specifically to the use of Eagle as important to helping her understand the way in which devices are designed:

> "I think that the Eagle part was really useful because I didn't know... I really had no idea how hardware designers pull everything together. I didn't know about libraries where you can just grab parts and it definitely demystified that part." (Brooke, closing discussion)

For some participants, this demystification was accompanied by a continued respect for the difficulty of the work that goes into making an electronic product:

> "I'd say I've had circuit board design and production demystified. But I don't- I don't think I had like a huge change in how I thought of this stuff. I've known some parts are hard and- I think it's basically the same except for the circuit board." (Brooke, closing discussion)

> "I'd say that it increased my level of respect for any electronic device and how much is involved on just putting a display on an alarm clock." (Ernesto, closing discussion)

Nathan and Tyler discussed some benefits of this demystification. Nathan emphasized its usefulness in evaluating the sophistication of new electronic products:

> "I think this did give me a little bit more insight into how some of those things are put together. And actually for work we were looking at a bunch of these Kickstarter projects and now I can sort of speak- I can look at them and read what they're doing on their circuit boards and stuff and go 'oh, that's really interesting' or 'oh, that's kind of trivial if you think about it'. So it's given me a little bit more insight which is actually really useful." (Nathan, closing discussion)

Tyler emphasized the agency to be gained by developing a better sense of the electronics we encounter:

> "I think if you're mystified and baffled by the world around you, whatever it is, it could be electronics, it could be something completely different, I think that gives you a lot less agency in your day to day life.... That's how I would pitch it. Know more about the stuff around you that has such a strong effect on you." (Tyler, one-on-one conversation)

Ernesto expressed a similar sentiment:

> "It empowers you, it expands your knowledge, it's- I also think that, honestly, I think that something like this has to be taught at some- at some level of school. It's- we need to know how to- really on this era of- digital era- just how to talk to devices." (Ernesto, one-on-one conversation)

For these participants, feeling empowered and a sense of agency in today's world involves an understanding of how electronic products work. Making with electronics is one promising way of developing this understanding. Furthermore, as discussed above, this process also gave some of the participants the perspective that they could meaningfully participate in the process of defining and creating an electronic product, even if they don't have all the technical skills needed to do so on their own. This is a significant transformation of these participants' sense of their own technological identities and an important step towards feeling like an empowered member of today's technological world.

7. Discussion

This chapter discusses the broader implications of both the DIY cellphone and the connected devices workshop. The chapter is divided into three sections, corresponding to the research questions posed in the introduction. The first section discusses the scope and limits of the personal fabrication of electronic products. The second looks at ways to engage people in these processes and the third explores the values of these activities.

What Are the Scope and Limits of the Personal Fabrication of Electronic Products?

This section begins by discussing the ability of personal fabrication to allow skilled individuals to design products for use in their own lives. It goes on to discuss the limitations on these possibilities, beginning with the acknowledgement that all DIY activity takes place in an ecosystem and that the structure of this ecosystem limits the things an individual is able to make. This is followed by a discussion of some of the unique challenges of applying DIY in the context of today's electronics.

Scope: Making Electronic Products for Use in Daily Life

The DIY cellphone and, to a lesser extent, my weather widget demonstrate that personal fabrication indeed offers the possibility to create electronic products for use in daily life. While this approach can't match all aspects of mass production, it yields the precise, robust, and optimized circuits and enclosures required for a product. My cellphone, for example, has been in continuous use for more than a year without changes. This robustness stems from the use of electronic components soldered to an industrially-produced (in very small quantities) PCB, a standard and successful construction. The various enclosures for the cellphone, while made with digital fabrication techniques rather than mass production processes, are similarly robust and optimized. Both electronic circuit and enclosure can be — and have been — precisely reproduced, another hallmark of a product. (In addition, of coursc, it's possiblc to make custom variations on the design of fabricated electronic products, a flexibility not typically possible with mass-produced devices.)

Again, this kind of product-ness differs from the electronic devices made using breadboards or electronics toolkits, with their less-optimized forms, less robust connections, and more restricted component choices. As discussed by the participants in the connected devices workshop, however, personal fabrication can be a way of taking these other types of prototyping forward by translating them into a more robust and optimized form for extended use.

In contrast to the DIY electronic products of the post-World War II era (discussed in Chapter 2), the personally-fabricated electronic devices in this dissertation also differ significantly from their commercial counterparts. This design space (of personally-fabricated electronic products) is a unique one, requiring its own design principles and best practices, as discussed in Chapter 4.

Scope: Nuanced, Open-Ended Creative Practice

Personal fabrication offers open-ended, nuanced, and continuously-variable possibilities, where an individual can undertake a self-directed design process — not just experiment with finite permutations of a fixed set of elements. Both Eagle and the CAD tools used here for designing enclosures allow for the construction of an unlimited number of subtly different forms.

Certainly, each fabrication process has its constraints (e.g. the flat forms of most laser-cut parts) but the possibilities are open-ended and continuously-variable in a way that's not true of say, LEGO bricks. In addition, because the constraints of digital fabrication processes differ from those used in mass production, fabricating a device's enclosure requires rethinking its materials, geometries, and other characteristics. This is an interesting, creative challenge, as demonstrated by the many diverse cellphone enclosures that have been produced.

Similarly, while not every electronic component is accessible to individuals working with personal fabrication (as discussed in the next sub-section), there are countless possibilities from a variety of sources. This makes the process of selecting and combining components a complex design challenge in itself (detailed in my development of the connected devices examples in Chapter 6). Even for the connected devices workshop participants making only small modifications to an existing design, the possibilities were nuanced and open-ended. The participants sought out components online and were free to layout their PCB in any number of ways. In designing the DIY cellphone and the connected devices examples, I was able to chose from an even wider variety of components and explore a truly open-ended design space in combining them. Again, this is quite different from the fixed set of possibilities found in an electronics toolkit and the discrete connections and fixed dimensions of solder-less breadboards.

Most of the variability seen in the fabricated devices in this dissertation is of the digital sort discussed in Chapter 2. That is, it is the result of precise changes made to a digital representation of a design rather than the natural product of imprecise manual processes. On the other hand, the creation of electronic devices depends heavily on the use of existing components and parts in a way that's not discussed by Carpo [2011] or McCullough [1998]. While the PCB connecting those parts is largely analogous to the 3D models discussed by those authors, the designer's freedom is limited by the availability of parts and their form and function.

Limits: The Ecosystem of DIY Practice

Notwithstanding the literal meaning of the phrase "do-it-yourself", much of DIY takes place within an ecosystem. Just as companies buy materials, parts, and machines from other companies, individuals depend on the products of industry to make things for themselves. In fact, as discussed in the background section, the very notion of DIY as a distinct activity stems from the industrial revolution and its separation between work and leisure. In light of this reality, I would argue that DIY is better understood as an individual's ability to assemble available materials and processes into a desired product rather than their ability to make everything themselves from scratch. Even the making of simpler objects depends heavily on the products of industry. *The Toaster Project* by Thomas Thwaites offers an amusing and informative account of an attempt to make an electronic product without relying on existing industry; Thwaites's difficulties suggest that this is not a feasible approach. (For a literary account of what is perhaps one of the more complex things that an individual can make entirely on their own, see *The Survival of the Bark Canoe* by John McPhee.) More complex objects, like the devices described in this dissertation, are effectively impossible for an individual to make without the use of commercial technologies.

This perspective, of DIY within an ecosystem, highlights the extent to which DIY practice depends on the decisions of actors, like industrial manufacturers, that may or may not have an interest in supporting it. For example, while I was able to find the components necessary to build a functional phone, I had few choices available. Furthermore, I'm not able to access many of the technologies in modern smartphones, like the latest processors or the highest-resolution display. Even if I could get these parts, I probably wouldn't be able to work with them. Similarly, some digital fabrication processes are available through relatively low-cost machines, others can be accessed through online service, and still others aren't generally available to individuals. There are technical reasons for some of these limitations but others stem from business decisions that may not align with the desires of the DIY community or even with the potential of making money in that market. Electronics manufacturing expert Bunnie Huang has pointed to this problem in the domain of processors: "I've reviewed business plans of over a hundred hardware startups by now, and most of them are using overpriced chipsets built using antiquated process technologies."[48] The work of these startups is limited because they can't buy the latest processors, not because of their own abilities or the technology that exists in the world.

The specifics of the ecosystem surrounding DIY practice change over time, particularly in the domain of technology. (This rapid change brings its own challenges, discussed in the next sub-section.) This can sometimes expand the possibilities for DIY practice. For example, the availability of new hand-solderable components could dramatically expand the possibilities for DIY devices, as could easier access to automated assembly processes. On this latter point, there are some promising signs, with some companies working to develop lower-cost pick-and-place machines and others provided easier, more standardized access to PCB assembly at smaller volumes. On the other hand, changes in the ecosystem can also create new limits. For instance, the discontinuation of essential components or changes to required infrastructure (like the cellular network) could prevent continued making or use of DIY devices.

The relationship between an industrial manufacturer and the individual do-it-yourself practitioner is often mediated by companies or other entities that focus specifically on hobbyist or DIY audiences. For example, SparkFun or Adafruit supply breakout boards for many electronic components; Arduino provides a higher-level platform for programming microcontrollers; Ponoko and Shapeways offer individuals access to expensive digital fabrication machines. On the other hand, relying on these intermediate platforms (as I did with the CC3000 breakout in the connected devices) makes the resulting devices less product-like and more restricted in their scope.

These intermediaries also highlight the importance of factors other than strict availability. Being able to order a part, with clear documentation, from a known supplier, using an easy online ordering form with a quick turn-around and low shipping costs is far easier than calling up an unknown company and going through a lengthy negotiation, even if both technically offer access to the individual. This ease-of-use is an important part of the value offered by companies like Adafruit or Ponoko.

Effecting changes in these supply chains can expand the possibilities for DIY practice as much as the creation of new tools or toolkits or the production of new educational resources. These intermediaries provide a potential point of leverage for influencing the ecosystem around the individual DIY practitioner. Our experiences with the phone suggests some ways that individuals can effect changes in these ecosystems. By working with Arduino to make the GSM module available, for example, we expanded the number of people

[48] "From Gongkai to Open Source", http://www.bunniestudios.com/blog/?p=4297

that could build the phone. By making the phone's hardware and software available online as open-source, we enabled others to build a phone without designing it themselves. Similarly, we were enabled by the decisions of others to open-source libraries for talking to various electronic components. These examples show that while industrial and commercial entities may shape much of the ecosystem surrounding high-tech DIY, it's still possible for individuals to make meaningful changes in the technologies available for DIY practice.

Limits: The Unique Challenges of Today's Electronics

You can't treat personal fabrication simply as an alternative means of making existing mass-produced devices. There are a number of challenges that constrain the possibilities. Processes like laser-cutting, 3D-printing, and CNC-milling don't necessarily scale well to mass-production quantities (Apple's MacBooks notwithstanding). PCB designs that work well for hand-assembly may not be optimized for automated production, and better or cheaper components may be available to people ordering in large volumes. As a result, lessons and design guidelines from the world of mass-production don't necessarily translate to personal fabrication, and vice-versa. This section discusses some of the particular challenges that today's electronic products pose for a personal fabrication or DIY approach.

The Sophistication and Complexity of Electronic Devices and Their Production

Making electronics products is a particularly challenging domain for DIY practice, such as the personal fabrication approach taken here. Commercially-available electronics products are incredibly complex and, in many modern instances, quite well designed. This creates high expectations that are difficult to match with DIY devices. As participants in the connected devices workshop discussed (in the "Value of the Products" sub-section of Chapter 6), in imagining the kinds of electronic products that can be made with personal fabrication, it may not make sense to start by thinking about existing commercial devices. (Instead, it's better to start from a consideration of other values, discussed in "Why Electronic Products Using Personal Fabrication?" below.)

Today's electronic products are manufactured using increasingly sophisticated machines and processes, processes that are difficult to match with the technologies that are available to individuals. This isn't true in other domains. For example, clothes are typically sewn by an individual operating a sewing machine that's not much different from the one an individual hobbyist might buy. The disparity wasn't always true of electronics either. In the post-World War II era, for example, a store-bought electronic product was likely soldered by hand in a process not that different from the one a hobbyist would have used to assemble a DIY kit at home. The commercial activity took place on a larger scale and required more organization, but an individual could achieve relatively comparable results, if perhaps somewhat less efficiently. The processes used to assemble the devices in this dissertation, on the other hand, are completely different from those used in the production of commercial electronic products. As a result, it's not a question of doing the same thing as a big company but for oneself. Instead, the challenge is to achieve similar results with much more limited means, a far more difficult goal.

Much of this difference stems from the fact that commercial products are often highly optimized. Component selection limits the extent to which this optimization can be achieved in personally-fabricated devices. As discussed in the previous sub-section, many of the components in commercial devices aren't even available to individuals, further increasing the challenge of competing with commercial products. Both the GSM module in the cellphone and the wifi module in the connected devices, for example, were significantly bigger than the equivalent parts in commercial products, restricting the sizes of the devices. The nature of the available displays also proved to be a significant limiting factor, as the thin and high-resolution screens found in modern smart phones weren't available. The sheer quantity and density of the components found in modern mass-produced products would be nearly impossible to achieve with hand soldering, further limiting the possibilities.

The sophistication of commercial devices extends to software as well. There's a huge difference between the operating system and robust applications found on a modern smartphone and the single Arduino program powering my DIY cellphone. While open-source development methods have proven capable of developing complex pieces of software, the need for this level of sophistication is another significant barrier to matching the capabilities of commercial devices with personal fabrication.

Rapid Changes in Electronic Components and Products

Another challenge is the rapid pace of change of electronics and electronic products. New components are developed all the time and may not be available to individuals when they're first produced (or ever). For example, in preparing for the connected devices workshop, I had to forego newly available options for both the wifi module (the ESP8266) and the processor (the SAMD21) because adequate software support didn't yet exist. Components also become obsolete frequently, preventing the fabrication of designs that depend on those parts. This happened in the case of the components at the heart of both the Fab FM radio and my 3D-printed mouse.

Even the general nature of the products that people are interested in changes rapidly, making it harder to keep up using a personal fabrication approach. In fact, most of the products I've made (radio, speakers, mouse, and cellphone) are obsolete in some way or another, even though they continue to function. The speakers don't support Bluetooth connectivity, the radio can't play music from the internet, the mouse depends on components that are now difficult to buy, and the cellphone lacks many of the features of today's smart phones. In other areas, like clothing or furniture, tastes change and styles evolve, but there isn't the same relentless progress that's found in the electronics industry.

Regulations and Policies Designed for Mass-Manufactured Products

There are also regulatory challenges specific to the electronics industry. Devices are subject to FCC regulations on electronics interference. Testing a design for compliance is an expensive and time-consuming process. This makes sense if you're planning to manufacture many identical products, but is poorly suited to personal fabrication, where each device may be different (and, therefore, require re-certification). Other regulations come not from governments but technical organizations. For example, many USB devices require a USB vendor ID, which has to be purchased for a few thousand dollars. Again, this may not be a large barrier for someone that intends to manufacture many devices, but it's significantly more expensive than making one of the devices described in this dissertation. In general, many of the rules and regulations around electronics seem to assume a mass-production approach, creating additional barriers for the individual production of devices.

How Can We Engage People in the Personal Fabrication of Electronic Products?

This section returns to the second research question: How can we engage people in the personal fabrication of electronic products? It discusses five strategies:

- Showcase Relevant Technology
- Curate Technologies and Examples
- Emphasize Problem Setting as Well as Problem Solving
- Provide Support from People and Support from Technology
- Leverage New Models for Collaboration

In addition to these strategies, it's important to consider the different trajectories that people may take, as discussed in reference to the DIY cellphone workshops and particularly in the discussion of the connected devices workshop. It's also crucial to consider the computational concepts, skills, and practices that they may develop, as discussed in Chapter 6.

This section closes with a short note about the role of programming in engaging people in the personal fabrication of electronic products.

Showcase Relevant Technology

Not all electronic products are equally relevant and appealing to all audiences. If we want to engage more people in the personal fabrication of electronic products, we need to identify and highlight products and product types with broad and diverse appeal. The differences between products have been clear in my investigations. The idea of building one's own cellphone, for example, is an activity with relevance to many people — both because of the importance and ubiquity of phones and because many people never realized it was possible to make one themselves. A radio, in contrast, is both less exciting as a device (one many people seem no longer to use) and something that people are less surprised (or not surprised) to discover that it's possible to make. Similarly, many of the participants in the connected devices workshop were specifically interested in the possibilities of internet-connected devices.

This need for relevance means extra work for those of us interested in engaging people in fabricating electronic products. We can't stick with the same few examples, as they're likely to become obsolete and won't appeal to everyone. Instead, we need to keep up, to the extent possible, with changes in technology and the kinds of electronic products that people encounter in their lives. We also need to look for products that will appeal to different audiences. At the very least, we should showcase different possible uses and applications of technology, as I did in creating and advertising the connected devices examples. This need for relevance can clash other criteria for engaging people, like speed and simplicity. The devices of most relevance to people are often relatively technically sophisticated, making them harder to design and make. (For example, the DIY cellphone is much more complex than the Fab FM.) Still, unless we look for relevant products, we will miss many opportunities to engage new audiences in fabricating electronic products.

Curate Technologies and Examples

Selection of appropriate elements for an electronic product is complicated by the incredible diversity of components available and the multitude of sources from which they can be acquired. Sorting through this abundance to find parts that not only have the required functionality but also work together requires skill and time. Experienced makers may have their own carefully accumulated set of technologies but assisting novices by curating an appropriate selection of components is an essential aspect of engaging novices in working with electronics. With electronic toolkits, this curation is done through the selection of which parts to include in the toolkit. This is a valuable but inflexible approach, since it's difficult to change these elements. With a personal fabrication approach, it's possible to curate a set of technologies or components for each device, workshop, or tutorial. This curation can happen through documentation (like the online instructions for the DIY cellphone), through technology (like the parts libraries that come with Eagle), or through in-person activities (like the connected devices workshop). (The relationship between these different kinds of support is discussed in "Provide Support from People and Support from Technology" below.)

In preparing for the connected devices workshop, I put significant effort into finding a combination of technologies (primarily microcontroller and wifi module) that could be programmed with Arduino and soldered by hand. The components also had to be capable of connecting to a range of online data, requiring further curation of appropriate web services and libraries. As many of the participants noted, these are technologies they wouldn't have discovered if not for the workshop. To further support these workshop participants, I developed examples showing how the individual technologies could be combined into complete devices. Because the goal of these examples was to serve as a starting point for participants' own designs, I deliberately made them as simple and clear as possible. This is very different from the design goals of a physical module (like the WildFire boards we used in the second week of the connected devices workshop). In that case, the flexibility and power of the module tends to be more important than the simplicity of its design.

Much of the value that hobbyist technology companies provide is in curating of sets of effective and usable components and helping people use them. Some of this involves creating breakout boards or writing software for connecting components, but partly it's just about picking out common and useful components from the larger set of parts available. While not necessarily specific to a particular activity or device, this kind of curation is more flexible than the incorporation of modules into a toolkit. Another key value of these companies is providing information about the components, like what they can do and how to connect them. In addition, the websites of these companies can bring together users of the particular technologies to share experiences and help each other troubleshoot.

This kind of curation is also provided by the community spaces and online vendors that offer access to digital fabrication processes. This goes beyond the literal provision of access to the machines to include things like documentation of the design rules for different processes. The communities that form around these resources can serve as repositories of experience and know-how that novices can tap into.

Whether in electronics or fabrication, effective making requires much practical knowledge about appropriate use of technology; sharing this knowledge is a crucial element in engaging people in working with the technologies. It's also essential, however, to help people learn how to curate technologies for themselves. In part, this can happen by sharing suggestions for particular components and the criteria by which they were chosen. More useful, though, is helping people learn to find and evaluate parts on their own, a process that requires support from both people and technology (discussed below).

Emphasize Problem Setting as Well as Problem Solving

Broadly speaking, the process of making something can be divided in two parts: deciding what to do (problem-setting) and figuring out how to do it (problem-solving). Some people might focus more attention on one or the other of these halves but to some extent both are always involved. Much of the work on engaging people in electronics focuses on problem-solving — i.e. building the tools and teaching the skills that people need to be able to build devices. My experiences engaging people in personal fabrication highlight problem-setting as an equally important aspect of helping people make things with technology.

Curating a set of technologies and examples (as discussed above) is not only a way of helping people make a given device (problem-solving), it's also a means of helping them figure out what device to make (problem-setting). The technological possibilities can suggest potential applications, providing inspiration grounded in realistic possibilities. For example, for the second investigation, the theme of connected devices was chosen from among many different possibilities. It suggests certain potential uses and applications, which are different than those that would be present with other technologies. My examples further showcased the possible uses of these technologies.

Another part of this problem-setting process is showing people possible uses for a set of technologies. This goes beyond documenting the individual building blocks to showcasing complete devices and their potential uses. There's a huge difference in appeal between "learn to interface with a GSM module" and "build your own cellphone", even if the underlying skills and technologies are similar. In advertising the connected devices workshop, I highlighted potential applications of internet-connected devices rather than simply describing the technology that would be covered.

Related to showing people products they can make is explaining the potential uses of particular components, like sensors. This was discussed in "Skills, Concepts, Practices" in chapter 6, for example, Nathan's need to understand the potential uses of an accelerometer. Scaffolding people's understanding of these possibilities is a crucial part of supporting them in the problem-setting process. Without a sense of what's technically feasible, it's difficult to evaluate the feasibility of potential projects. Technical capabilities can even serve as a means of brainstorming project ideas — for example, browsing the web services supported by Temboo or the sensors sold on Adafruit could suggest potential projects, like Luisa's headache warning system or Ernesto's current sensing project. On the other hand, product descriptions don't always do a good job of describing the potential uses of a particular sensor, instead assuming that customers will already know whether it's the right component for their particular needs. Considering problem-setting as its own activity emphasizes the need for illustrating the possible applications of these individual technologies.

It's important to acknowledge that personal fabrication may not be the best approach for problem setting. Because it's less tinker-able than other forms of prototyping, it's not as well suited to quickly experimenting with different possibilities, an useful component of problem setting. It may be better to use other approaches in the problem-setting process and transition to problem "solving" with personal fabrication once someone has set the problem for themselves.

Provide Support from People and Support from Technology

Learning to do something in a workshop setting, with in-person support and guidance, is very different than undertaking a similar activity on your own, with just the assistance of software and information. Because the connected devices workshop involved both support from people and support from technology, its lessons don't directly generalize to situations in which people are attempting similar activities on their own. On the other hand, by observing participants' processes and struggles in the workshop, I've gained a perspective on which challenges are better suited to support from people and which might be addressed with better software tools or information. Specifically, in-person support seems particularly important for ill-specified aspects of the process, like defining projects and selecting and integrating diverse technologies, and for physical activities like debugging circuit boards. On the other hand, participants' experiences revealed the need for better software tools and support for embedded programming and PCB design.

As discussed in the "Concepts, Skills, and Practices" section of Chapter 6, building devices outside the context of an electronics toolkit or a single-purpose kit requires multiple sophisticated practices, like selecting from a wide range of components according to diverse criteria and making sense of information from a variety of sources. Similarly, the problem-setting process, or figuring out what project to work on, is complex and ill-defined. These practices are difficult for novices and this type of complex and subjective judgement feels difficult to encode into a software tool. Instead, these challenges feel well-suited to in-person support from workshop facilitators or other experts or educators. In the connected devices workshop, for example, the participants' project-scoping processes benefited from the back and forth enabled by a workshop context and by the facilitators' understanding of the technological possibilities. This is an area in which an expert can listen to a novice's general interests and reflect them back through the framing of a more tractable design space. That is, the expert doesn't necessarily tell the novice what to do, but helps them frame a design space in which they're capable of finding their own project definition.

The importance of personal support applies to project definition on a small scale as well — for example, in figuring out the next change or addition to try to make to one's program. At its core, this is a pedagogical challenge, not a technological one — i.e. workshop participants were looking for appropriate challenges or tasks that would foster their own learning and progress, rather than simply code that would accomplish a particular function. Again, this question of what to work on next depends on many factors, like someone's overall goals and interests, their current understanding and skill, and the technical possibilities. Making sense of these criteria and suggesting next steps requires sympathy and judgement on the part of workshop facilitators. (A concept explored in more detail in John Dewey's *Experience and Education.*)

A workshop also provides a social context and structure for work in a particular area. As Tyler described it:

> "It was nice to have that kind of like 'okay, you're starting to learn how to swim, you can do a little bit of this, now go a little bit deeper, go a little bit further. We'll be here but you should try to lead yourself out there.' It was nice to have that kind of like support as we were poking around. If for no other reason other than when I got frustrated or stumbled into a wall, I couldn't just put it away and go watch TV cause you guys were right there and we had three hours left." (Tyler, one-on-one conversation)

For him, it provided appropriate challenges and a setting on which to work on them. Similarly, for the other participants, the workshop seemed to provide a way to carve out time to dedicate to the activities. Participants were devoted to the class, with only two participants missing even one session. On the other hand, participants struggled to find time outside of the class to work on these topics. They often borrowed hardware to take home but returned the next week without having found the time to work with it. The structure of a workshop, therefore, seems particularly valuable as a way of helping people dedicate time to the activity. These social contexts can also provide emotional support and encouragement, helping people persevere through challenging activities.

On a more concrete level, the debugging of a circuit by a novice is something that seems difficult to accomplish without in-person support. It requires a consideration of many potential hypotheses, each of which must be checked and confirmed or rejected. The difficulties are compounded because the circuit can appear opaque and mysterious. Even if someone knows how to use a multimeter, it's not clear what to measure or how to connect those measurements with an understanding of the potential problems. In trying to support people debug their DIY cellphone remotely, I've come to appreciate the difficulty of debugging from afar. In many cases, the person in need of support may not realize that a particular issue could even occur and therefore wouldn't think to describe the symptoms necessarily to recognize it. There are typically many possibilities to distinguish, each of which requires some diagnostic process that may or may not be easy to explain. The distance adds overhead to each step of what is often a many-step process. Having to rely on asking specific questions makes it difficult to notice other issues that you might not have thought of, also important in debugging. The small size of some electronic components means that even photographs of a circuit are often not particularly helpful.

With in-person support, debugging circuits can be a fruitful process, both in identifying and correcting the problem and in helping the novice to learn. In the connected devices workshop, facilitators would suggest potential issues or fixes to be tried by the participants. This provided a useful division of labor, allowing facilitators to support multiple participants while allowing participants to draw on the expertise of the facilitators. Moreover, by investigating potential hypotheses for the source of a given problem, the participants added to their stock of future debugging strategies.

In-person support can be particularly valuable for novices, who may not know enough to clearly articulate a question in writing. Once someone has more technical expertise, they may be better able to seek and receive support remotely. That is, they will have more of the knowledge and vocabulary needed to articulate the problem they're encountering and a better sense of the information relevant to figuring out what's happening. Importantly, workshops can help people develop their ability to get help from those with more experience. Nathan, for instance, said that he felt more confident is his ability to articulate a specific question as a result of the workshop. Luisa, with less experience, felt that she would still struggle to communicate her problems in a way that would help her get assistance. As a result, she's more likely to need in-person support in the future.

In programming the microcontrollers in their projects, participants struggled with many of the issues that confront novices writing code of any kind: understanding the syntax, making sense of the flow of control, working with low-level or confusing API's, etc. Here, it seems that there are many opportunities to leverage the lessons learned about helping novices program to these kinds of embedded systems, a topic discussed in the future work section of the next chapter.

In the domain of PCB design, the auto-router provides a hint of the power of software to aid novices. As participants commented:

> "Eagle was nice. I could definitely see how you could have to spend a lot of time needing to become a real pro at it but the auto-router was magical. It was cool watching all the wires go bo-bo-bo-bop." (Tyler, closing discussion)

> "Yeah, laying out traces was way more easy than I thought it would be. I had no idea that there was such a thing as the auto-router." (Brooke, closing discussion)

One can imagine many other algorithms or interfaces that would similarly leverage the power of software to help novices design custom PCBs. This is also discussed in more detail in the next chapter.

Even with better software tools for programming and circuit design, the connected devices participants would likely have struggled with many parts of the process. These include the process of scoping their projects, of selecting and integrating new components, and debugging their circuits. Some of these practices may also be amenable to technological solutions (e.g. automated component assembly could avoid the need to debug hand-soldered circuits). Still, open-ended activities like designing and making an electronic product for oneself involve many complex and subjective steps that seem likely to rely on support from other people for the foreseeable future.

Leverage New Models for Collaboration

By capturing the designs for electronic products as digital files, personal fabrication enables new means of sharing and collaborating on those designs. This open-source hardware practice echoes open-source software but comes with its own set of challenges and opportunities. My experience with the DIY cellphone suggests that it is, in fact, possible to share a relatively complex design online for others to reproduce and modify. People around the world have made their own DIY cellphones and customized its design. This global activity requires some adjustments to the components and processes used in order to expand the number of locations in which its feasible to make a device. More significantly, however, are the barriers to integrating the work of many people on a centralized design. For one thing, the software design tools I've used aren't good at tracking or merging changes. Another problem is the extensive overhead required to test a given change. Typically, you need to fabricate the entire device, which is a significant barrier for a small revision. Packing many components together onto a single PCB can make the design difficult to modularize, since changes to any part may require shifts and adjustments elsewhere. This makes it more difficult to portion out the work.

Open-source hardware does offer, however, opportunities to engage the efforts of many different people. It requires skills in electronics, in programming, and in industrial design, among others, and someone with expertise in one of those areas can make valuable contributions. Many individuals designed unique enclosures for the DIY cellphone, for instance, without having to understand its electronics. I drew on software libraries written by a variety of Arduino users, who may or may not know anything about designing PCBs. As discussed previously, the process of just assembling a device or even just its enclosure, can provide an opportunity for personal involvement and meaning in ways that aren't found in software. Small efforts to make the process accessible to experts in different domains can go a long way. The basic 3D model I made of the mouse PCB, for example, enabled architects and industrial designers to model enclosures for it. Even two-dimensional vector diagrams exported from PCB design software can be the basis for a 3D model, if a slightly less convenient one. Uploading a PCB design to a site like OSH Park means that someone who just wants to assemble a device can order the board without having to download or understand PCB design software. If we look for ways to engage people in the personal fabrication of electronic products, we can find many opportunities to work with a broad range of people.

These new models of collaboration create interesting questions of authorship, as discussed in Chapter 2 (in the "Qualities of Craft, Mass Production, and Digital Fabrication" section). When one person designs a PCB but someone else orders it, assembles it, and creates a custom enclosure for it, who's designed the device? What about when one person makes a few small changes to someone else's PCB design but puts the resulting circuit to a completely different use? Etc.

A Note on Programming

One of the most complex and open-ended parts of the process of making electronic products has been writing the code that runs on them. Connecting each sensor, actuator, or communications module to the microcontroller is a relatively straightforward and finite activity involving a few traces and maybe some secondary components. The software that runs on the microcontrollers in the devices does much of the work of tying together the various components into an integrated device. Not only is programming difficult, but it's a very different skill and activity than designing circuit boards, soldering, and the other parts of physically making an electronic product. As a result, it's difficult to learn all these things at once. For the participants in the connected devices workshop that already knew how to program, the workshop seemed to work well as an opportunity to learn how to apply those skills to electronics. They were able to gain more familiarity with the Arduino software and the process of debugging code and circuits together. For participants without prior programming experience, however, the workshop probably didn't offer sufficient opportunity to develop those skills. Someone interested in learning to program the physical world might be better served by a more standard Arduino workshop that omitted the design and assembly of custom circuit boards.

Helping participants write the programs for their projects was difficult to do in the context of the workshop. Unlike debugging circuits, which is a finite process with a definitive end (a working circuit), programming can be an open-ended and exploratory activity. It requires the knowledge of many individual programming concepts and the ability to combine them into a desired result. With the Arduino code we were using in the connected devices workshop, participants often needed continuous one-on-one attention as they wrote their programs, support that wasn't possible to provide given the participant to facilitator ratio. When troubleshooting a circuit, the facilitator could ask a participant to investigate a potential problem and then report back. After a few rounds of this, it was generally possible to solve the problem. With the programming, on the other hand, the support needed was both more continuous and more extended. That made it a much more difficult activity to support.

As mentioned in the introduction, this dissertation explicitly focused on the other aspects (besides programming) of making an electronic product using personal fabrication. Still, it's important to acknowledge that programming is an essential part of the process. To help people make their own electronic products using personal fabrication, we need to draw on the lessons of those working to support novice programmers and combine them with the insights from this dissertation.

Why Make Electronic Products Using Personal Fabrication?

This section returns to the third research question: why make electronic products using personal fabrication? It should be clear from the first section of this chapter that this personal fabrication approach is not necessarily capable of matching commercial, mass-produced devices. In addition, personal fabrication may not be the best approach for quickly experimenting with new technologies or interactions. That kind of tinkering is often better served by the use of electronics toolkits or breadboards that make it easy to quickly "sketch" a particular circuit, as mentioned by the participants in the first DIY cellphone workshop. Still, as the case studies in this dissertation illustrate, personal fabrication is a rich, diverse, and meaningful means of making electronic products. Here are some of the reasons why someone might want to do it, organized into two categories: changing *the nature of* electronic products, and changing *how people feel about* electronic products.

To Change the Nature of Electronic Products

Personal fabrication allows for changes in the nature of electronic products themselves. Here, I discuss three reasons, related to the nature of the products themselves, that someone might engage in this approach:

- To make things you can't get otherwise
- To support extended, real-world prototyping of a device design
- To leverage the power of software

To make things you can't get otherwise

Because it allows for the production of electronic devices in small quantities, personal fabrication can support the making of electronic products that wouldn't make sense in the volumes required for mass production. For instance, in today's mobile phone industry, there is an overwhelming emphasis on the development of increasingly sophisticated and powerful — and quite similar — smart phones. Presumably that's because this is what most of the market wants. For someone like me, though, that was looking for a simpler mobile device — but not an old, obsolete one — there were few appealing options. With personal fabrication, I was able to explore this design space of simpler — and different — mobile phones. This sentiment was echoed by some of the participants in the second DIY cellphone workshop. As one put it:

> "They don't make phones like this any more. I'm a purist in terms of electronics and I don't like how they're making computers handheld. I like a radio being a radio, a phone being a phone."

In this age of universal, all-powerful devices, personal fabrication offers opportunities to re-examine dedicated devices from a new perspective. There are fewer technical constraints; instead, we can take general purpose technologies (which may be developed for smart phones or other convergent devices) and use them in special-purpose designs. As Nathan put it in the closing discussion of the connected devices workshop, "I explained what I put together in the class and someone said to me, 'so you made one function on the iPhone?'" Even if similar functions can be achieved with a general-purpose smart phone, there are opportunities to provide optimized user interfaces, custom form factors, and simply fewer distractions. My weather widget is a good example — it's easy to check the weather on a smart phone, of course, but having a dedicated device makes it possible to get constant, at-a-glance access to this specific piece of information. This is especially applicable in cases where information (or interface) needs are tied to a particular location — as in the case of wanting to check the weather before deciding what to wear out of the house in the morning. The weather buoy system that Tyler made in the connected devices workshop, or the the snow removal system that John pursued, are other good examples of this kind of location-specific application.

This application of personal fabrication to special-purpose devices works well when you can achieve a simple function with a simple technological implementation. Other kinds of functionality (e.g. anything that needs a full-fledged web browser) requires much of the hardware and software capabilities of a smart phone. This makes it difficult to create given the constraints on personal fabrication.

Personal fabrication also enables the exploration of alternative aesthetics and materials than those found in most mass-produced electronic products. The veneer and fabric of the Fab FM and the solid wood in the milled DIY cellphone enclosure look very different than the glass, metal, and plastic of most consumer electronics. Jeff Warren's cardboard enclosure for the DIY cellphone emphasizes this contrast with industrial products. The electronic components themselves can provide specific, retro associations. The LED matrix on the DIY cellphone reminds some of old electric calculators and the phone with the LCD prompts inevitable queries about Snake, the classic game found on old Nokia phones. The steampunk community, in particular, revels in alternative aesthetics for electronic devices. [Tanenbaum et al. 2012] These unique aesthetics have the added advantage of provoking curiosity and prompting questions in strangers, which provides an opportunity to spread the idea of DIY electronics (discussed in "Communicating the Possibility of DIY Electronics" below).

One possibility that's not explored in this dissertation is making custom electronic products for users with specific or unusual needs. This is discussed in the future work section of the conclusion chapter.

To support extended, real-world prototyping of a device design

In my experience, a personal fabrication approach is particularly useful in creating integrated, robust prototypes combining electronic functionality with specific physical forms. It offers the freedom to experiment within the capabilities of a large set of processes and parts. This is much broader than the possibilities enabled by any given toolkit system or set of modules. A PCB can create almost any set of connections between electronic components, provided you can get those components and assemble them onto the board. A laser-cutter can cut almost any 2D outline out of a range of flat materials (although there are only so many ways to combine them into a 3D structure). A desktop 3D printer can print complex and varied geometries (although possibly only in rough plastic). The set of components for which you can get some breakout board is bigger than the set of components available with a littleBits connector, for example, or as a LEGO brick.

In fabricating electronic circuits, the use of physical building blocks (electronic components) is, of course, essential. The use of PCBs however, fundamentally transforms the process of connecting those components together into more complex circuits. It allows for far more integrated, optimized, and robust circuits than those possible with breadboards or electronics toolkits. There's no need for each building block to have its own PCB, connector, or enclosure; instead they can all be combined on a single PCB. As an example, the initial breadboard prototype of the DIY cellphone would never have fit into a pocket and would have been difficult to transport without it falling apart.

All of this enables the creation of interactive devices that it may not be possible to build with other kinds of prototyping processes. Devices that need to communicate with specific protocols or devices, use specific types of sensors, provide specific interaction modalities or form factors, use a particular kind of display, etc. may not be possible with a toolkit but could be accommodated using a custom PCB. Many geometries may not be feasible to achieve with manual working of off-the-shelf materials but could be created with a 3D printer.

These kinds of prototypes are particularly valuable for real-world testing of a product, where the combination of form and function is crucial to understanding how a device will fit into someone's life. With the DIY cellphone, for example, it was critical to have a prototype that would both fit in my pocket (so I would carry it with me) and make phone calls. These integrated prototypes are also important for testing the relationship between electronic components and enclosures or other mechanical parts. For example, in using the DIY cellphone in my daily life, I discovered that the LCD screen died after a month or so of use, prompting the switch to the LED matrix.

This is something that I wouldn't have discovered without real-world testing of the prototype. There were also many software problems and enhancements that I found through using the phone in a variety of real-world situations. Perhaps as importantly, using the phone myself helped me discover features that I didn't need and, therefore, was able to avoid taking the time implement.

Furthermore, personal fabrication blurs the distinction between prototype and product. It allows an individual to (relatively) rapidly produce an individual prototype from digital design files that also serve as a reproducible definition of a product. With the DIY cellphone, for example, each prototype was robust and optimized enough that I could immediately start using it in my daily life. As I discovered problems or opportunities with the phone, I could iterate on its design, as you'd expect with a prototype. At the same time, each iteration could be reproduced in more-or-less identical form, as you'd expect with a product. This is very different than mass-production, whose processes don't lend themselves to use in making prototypes. It's also very different than manual construction, which is poorly-suited to making identical copies of a product.

Creating a prototype using digital fabrication requires the creation of a digital design and the translation of that design into a physical object. The more overhead involved in this process (in terms of time, resources, etc.), the less useful it is as a form of prototyping. This stresses the importance of access to fabrication at low cost and with rapid turn-around. It also places increased emphasis on the speed and fluidity of the software tools used for design (discussed in "To leverage the power of software" below). With the right access to machines and the necessary fluency in design software, however, personal fabrication can be a valuable means of prototyping.

In making the connected devices prototypes, for example, I was ordering a new PCB prototype of one of the three examples approximately once a week. These took approximately three days to get produced and sent back to me, and another day or two to assemble and test. Because the devices were relatively simple, I could make changes to their designs in a couple of days and order new revisions. In the case of the cellphone, the time required to get a new PCB made was similar, but the overhead was much greater. Soldering the components took longer, testing the device was more complicated, and it was more necessary to make a new enclosure each time. Still, in a relatively short period, I was able to make a new device and begin using it in my daily life.

To leverage the power of software

Much of the work of designing an electronic product for personal fabrication happens in software, which gives it very different possibilities than purely physical forms of making. The flexibility of software can allow for the creation a wider array of forms and functions than would be possible from the combination of standard physical building blocks (whether modules of an electronics toolkits or, say, pre-cut wooden blocks). Software enables precision and repeatability. Designing in software also facilitates leveraging the work of others, whether by building on files they've shared or by creating software that encapsulates their knowledge.

In making the enclosures for the DIY cellphone and connected devices, I've benefited from the precision and repeatability of software. Lacking skills in woodworking or sculpture, I would have had a difficult time making these objects by hand. And, having made one enclosure, I would have had to start over in making the next. Whereas by designing the enclosures in software, I could capture the many details required in a reproducible form. If I made mistakes or had new ideas, I could make adjustments in the next iteration without having to recreate the entire object from scratch, as I might have to if it were built directly in the physical world.

Designing circuits in software also allows abstractions to be shared as digital designs rather than physical modules. This process of posting a design file online is significantly easier, cheaper, and faster than manufacturing and distributing physical modules. That makes it possible to share far more abstractions than it would be feasible to incorporate into an electronics toolkit. It also means that you're not tied into a particular system of making connections, making it easier to incorporate new components into a design.

When access to digital fabrication was limited to small numbers of professional experts, the accessibility of CAD software was less important. It was reasonable to assume that those professionals would take the time to learn complex interfaces and could afford to pay for expensive software packages. As access to fabrication becomes more widespread, however, the accessibility of software design tools becomes an increasingly important factor in the ability of individuals to fabricate custom objects. The software tools haven't always kept up with the advances in the fabrication technology itself. Specific opportunities for improvements in this area are discussed in the future work section of the conclusion chapter.

To Change How People Feel about Electronic Products

Perhaps even more important than changes to the devices themselves are the transformations that personal fabrication can enact in people's relationships with those devices. Four themes are discussed here:

- The Personal Satisfaction of Making Something for Use in Your Own Life
- Understanding how Devices Work

- Transforming People's Technological Identities
- Communicating the Possibility of DIY Electronics

The Personal Satisfaction of Making Something for Use in Your Own Life

The process of designing, making, and using the DIY cellphone, while not always easy, has been personally rewarding. Everyday, I have the satisfaction of using a device that I created; I understand how it works; I choose to make it the way it is. (And I've come to terms with its limitations.) Pre-requisite to this satisfaction is the fact that the phone is functional and attractive enough for me to make regular use of it. The breadboard prototypes of the phone, on the other hand, while rewarding in the fact that I was able to make them, don't offer the same sense of ongoing satisfaction through use. Personal fabrication was what made it possible for me to create a cellphone circuit small and robust enough to fit in my pocket and an enclosure clean enough that I'm proud of it. In those respects, it serves as an important enabler of satisfaction through the use of electronic products.

Much of this satisfaction is despite (or, perhaps, because of) the fact that I could have simply bought a commercial cellphone. As discussed in Chapter 2, hobbies often provide personal satisfaction — and the satisfaction of doing something productive and useful — even in cases where there may have been other means of satisfying the practical need. In that respect, the personal fabrication of electronic products is no different than other hobbies in offering real enjoyment and satisfaction despite the lack of purely economic or functional justification.

Participants in the cellphone workshops experienced something of a similar reward in the use of a device they had made themselves. The excitement and joy that many expressed upon making the first phone calls with their newly assembled phones hinted at similar senses of accomplishment. While the precise form of the phone may have been less important in this case — many participants made their first calls before completing the enclosures — my sense is that the feeling of *having made a phone* was critical in these emotions.

Understanding how Devices Work

Personal fabrication provides a unique opportunity to interrogate the nature of today's electronic products. Despite their increasing functionality and ubiquity, these devices are largely opaque to most people. As discussed in the background chapter, this wasn't always the case — in the decades following World War II, it was relatively common for someone to assemble their own electronic products from a kit. This process didn't necessarily allow for much flexibility or creativity, but it did at least allow people to see what made up the electronic products in their lives. Assembling a fabricated electronic product offers a similar opportunity to understand today's devices. As one of the participants in the first DIY cellphone workshop put it:

> "When I'm choosing a phone, I'm choosing the phone based on what I see, how it looks, I never thought about where the speaker might be, what's the speaker like underneath it, a speaker to me is this set of holes, but then there's this speaker component inside that. I don't know what it looks like in my phone but I know what it looks like in this other phone [the DIY cellphone]. All the different elements, I know what they look like and what they are."

For this participant, the workshop was a chance to see the way in which a complete product could be assembled from individual components, along with a fabricated PCB and enclosure. Furthermore, as in the discussion in the second cellphone workshop, this process of assembling a device from parts can create debate about the origin of those parts and about who has control of the way in which they're put together.

This personal fabrication approach supports reflection on the nature of commercial devices in a way that experimenting with toolkits may not. On the one hand, it shows how an individual can draw on existing, commercially-produced parts and processes of the sort that might go into a commercial product — whereas toolkit modules involve an additional layer of abstraction that must be penetrated to understand how products are composed from their fundamental parts. On the other hand, the final result of personal fabrication processes can be suggestive of — even if less sophisticated than — commercial devices in a way that prototypes made with toolkits often are not. In this way, then, the personal fabrication of an electronic product allows an individual to replicate, in its own way, the process that industry undertakes in producing an electronic device. This makes it a particularly valuable lens for reflecting on those industrial activities.

Transforming People's Technological Identities

Designing and making a device — even a simple one — using personal fabrication or assembling a more complex device designed by someone else can transform an individual's perception of the feasibility of making electronic products for themselves. After the connected devices workshop, Tyler, Nathan, Brooke, and, to some extent, Akash talked about the design of a custom PCB as a practice they could draw if circumstances called for it. Participants in the first cellphone workshop expressed similar sentiments. As one put it:

> "By the end, you feel like, yes, maybe someday you can customize your own gadget. Cause phone, comparably saying,

> it is not a very simple gadget, so I mean, if you can make a phone, possibly we can some other stuff by ourself."

These individuals haven't necessarily learned everything they'd need to know to make an electronic product on their own — but they've learned to think of it as a possibility, as something they could do if they wanted to. In part, this stems from having seen someone else (i.e. me) do this for themselves and, in part, it stems from having gone through some of the steps for themselves. This perspective of oneself as the kind of person that can participate in the process of creating an electronic product may be even more significant than any specific technological skills, because it gives people the confidence to acquire those skills if they want to.

Communicating the Possibility of DIY Electronics

My experience with the DIY cellphone suggests that some of this feeling can be communicated just by providing an example to others. Many strangers, after seeing my phone, have asked what it is, how I made it, how they could create their own. This may be only a starting point for someone feeling like they could make an electronic product for themselves, but it is, at least, that. In today's world, many of these people might not have otherwise encountered examples of DIY electronic products. In this sense, using a DIY device in one's life can be a way of communicating and promoting the possibility of making electronic products for oneself.

The appearance of a device plays a key role in communicating these possibilities. The wooden enclosure on my DIY cellphone, for example, seems to have been an important factor in drawing notice and conversation. Being able to see what's inside a device also provides an opportunity for education — facilitated, for example, by the magnetic closure on my DIY cellphone enclosure or the clear acrylic enclosures on the connected device examples. Other aesthetic qualities can draw notice and comment — the exposed PCB edge on the cellphone, for instance, or its retro LED matrix display. By using DIY electronics in our daily lives, we share the message that electronic products aren't magic and that people can learn make them for themselves if they want to.

8. Conclusion

This chapter starts by summarizing and recapitulating the research described in this dissertation. It then discusses some opportunities for future work, followed by a discussion of some of the ongoing challenges in the personal fabrication of electronic products. Finally, the chapter closes with some reflections on the relationship between this sort of DIY activity and an individual's sense of empowerment or agency with respect to the technology in their life.

Recapitulation

This dissertation explored individuals' creation of electronic products using digital fabrication and electronic components. It focused on three research questions:

- What are the limitations of and constraints on the personal fabrication of electronic products?

- How can we engage people in the personal fabrication of electronic products?

- Why make electronic products using personal fabrication?

These questions were investigated through two projects: (1) a DIY cellphone that I created and used in my daily life and (2) a workshop in which participants designed their own connected devices with my support. These investigations built on the framework for the personal fabrication of electronic products laid out in Chapter 4. That chapter discussed the elements of these devices (fabrication processes and materials, electronics, and software) and best practices for combining them into complete products.

Chapter 5 documented my personal experience using the DIY cellphone as well as my efforts to help others make and modify the device. Those efforts include two short workshops, which highlighted the way in which a personal fabrication approach can help individuals to understand and question the composition of electronic products and their relationship to those devices. I also discussed lessons from the online dissemination of the DIY cellphone, which showed that it is possible to distribute electronic products solely as digital files but that it's difficult to support others remotely and to integrate the work of multiple people.

Chapter 6 described the connected devices workshop, in which participants designed and fabricated their own wifi-connected devices over the course of six sessions. This workshop highlighted four different participant trajectories, each illustrated by a profile of one of the participants. It also yielded a rich set of computational concepts, skills, and practices that emerge in the course of the personal fabrication of electronic products. Finally, participants discussed the value of DIY electronics and personal fabrication for themselves and for society.

Chapter 7 returned to the research questions listed above in discussing the lessons of the DIY cellphone and the connected devices. These devices demonstrate that personal fabrication offers an open-ended and nuanced design space for electronic products but they also highlight the specific limitations and challenges of electronics as a domain for DIY practice. In particular, the sophistication of electronic products, the rapid changes they undergo, and the ecosystem mediating access to the underlying technologies makes it difficult for personal fabrication to compete with mass production on its own terms. Acknowledging these limitations, the chapter discussed ways of engaging people in the personal fabrication of electronic products, offering multiple strategies that acknowledge the importance of both technological and personal or social factors. Chapter 7 concluded with a discussion of the value of the personal fabrication of electronic products, including the ability to create unique devices and prototypes that would be difficult to make in other ways. More importantly, it highlighted the ways in which these activities can transform people's understanding of their own capabilities with respect to the technology in their lives.

Opportunities for Future Work

This dissertation explored the possibilities for the personal fabrication of electronic products, as offered by existing tools and technologies. In the process, it revealed many opportunities for the creation of new tools and technologies, discussed here — along with the possibility for the creation of unique, domain-specific devices.

New CAD Tools for New Fabrication Processes

One lesson of this dissertation is that different fabrication processes have very different affordances, in terms of the geometries they support and the uses to which they can be put. Many of today's software CAD tools, however, were not designed for today's fabrication processes. There is an opportunity for new CAD software targeted specifically at the capabilities of these fabrication processes and which encodes design rules and other best practices for working with them.

For example, designing for the laser-cutter involves a lot of 2D forms — but it also requires thinking about how those 2D parts will combine into a 3D structure. This kind of construction is difficult to visualize with a standard 2D vector drawing tool and may not require the full complexity and sophistication of a full 3D modeling package. Instead, it may be better served by a specialized, hybrid design tool. Enclosed [Weichel 2013] is one example along these lines but the space feels ripe for further experimentation.

Similarly, the geometric freedom of 3D printing creates a vast design space. As a result, the capabilities and interfaces of particular CAD tools play a crucial role in shaping the forms that are created. Here, there are opportunities to build design software tailored to specific types of constructions (e.g. enclosures for electronic devices) within a given fabrication process (3D printing). By targeting a specific subset of the possibilities, this approach could provide a simpler learning curve than more general 3D modeling tools. Meshmixer [Schmidt & Singh 2010] is an example of a tool optimized for free-form mesh modeling, but one can imagine many other potential use cases and corresponding interfaces.

This need for specialized and easy-to-use CAD interfaces is compounded by the increasingly accessibility of digital fabrication processes and their relative speed compared with other methods of making. As more people gain the ability to easily and quickly translate digital designs into physical objects, the speed and ease with which those digital designs can be created becomes the bottleneck of the creation process. As McCullough points out in *Abstracting Craft*, these digital design tools are mediums in themselves; we should take advantage of our ability to create new ones.

Encoding More Knowledge into Circuit Design Tools

In the domain of circuit design in particular, there are many opportunities to provide hobbyists with more powerful tools. Eagle, the PCB design tool used through this research and popular with many electronics hobbyists, encodes very little knowledge of electronic circuits beyond what's strictly required to generate appropriate PCB geometries. Someone designing a circuit, however, needs to know a lot more about the components and connections involved — like which pins of a microcontroller support which types of components, or what voltages are acceptable in different parts of a circuit, or what supporting components are required for a particular sensor or actuator. Incorporating this kind of information into a circuit design tool would allow users to work at a higher level of abstraction, focusing on the overall design and functionality rather than low-level details. This should make it easier for novices to work independently and lessen the need to coordinate many diverse sources of information. Fritzing [Knörig et al. 2009] attempts to address similar concerns by providing users with a breadboard view of their circuit, but it does little to incorporate other kinds of information that a novice needs to create a circuit. Another motivation for incorporating a higher-level understanding of circuits into a design tool would be to help people debug their circuits, e.g.

suggesting which connections to verify with a multimeter depending on the problems observed.

Increased automation of circuit layouts is another area in which the power of software could be used to further support novice activity. Allowing people to specify the location of important components (like user-interface elements) while automatically positioning and connecting the remaining parts would allow for a focus on the product design aspects of the process. Additionally, automated tools and algorithms could help in optimizing the layout of components, minimizing the size of devices or allowing for specific form factors, important considerations for real-world use (as in the case of the DIY cellphone). Professional PCB software has some of this functionality, but a novice interested in simplifying the layout of a few components has very different requirements than a professional interested in optimizing a complex circuit.

As with the digital fabrication processes just discussed, PCB fabrication (and even automated assembly) are becoming increasingly accessible to individuals. This again emphasizes the importance of software design tools as a limiting factor in a novice's ability to create a custom circuit.

Integration of Electronic and Mechanical CAD

The easier it becomes to iterate on the design of both PCBs and their enclosures, the more tedious it becomes to keep the two synchronized. We need for better tools to coordinate these domains, allowing someone to change a device's circuit together with its enclosure and to understand how the two relate (without having to physically assemble the entire device). Part of what's missing are smoother processes for synchronizing geometries between domains (e.g. PCB and mechanical CAD). Another are 3D representations of electronic components — some exist, but many don't and they're not well integrated into Eagle and other hobbyist CAD tools. Again, this is an area in which professional tools provide more capabilities but one in which novices or individual hobbyists may have different needs than professionals. Specifically, for novices it's important to make simple products easy to create, whereas experts may be more interested in making complex devices possible.

Better Tools for Integration and Testing of Revisions

This dissertation includes many examples of someone taking a digital design file, modifying it, and fabricating a custom artifact from that modified digital file. Comparing different versions of a file, incorporating changes back into the original design, and integrating the work of multiple people remains a difficult challenge. This is an area which would benefit from the mature tools and workflows found in the world of open-source software (and other large-scale software development). In software, the ability to review, merge, and test revisions is crucial to the ability to combine the work of many people in the creation of complex artifacts. Similar capabilities would be of great benefit in the hardware realm as well. There are challenges, of course, like the difficulty of testing hardware without making it and the difficulty of dividing up physical modules into distinct modules. Still, it seems there are opportunities in this space — even basic tools to visualize the differences between two versions of a file, or merge individual changes from someone's else work could provide a big boost towards the practice of open-source hardware. This is another way in which people's ability to create new things is more limited by their ability to design rather than their ability to translate those designs into physical form.

A related challenge is coordinating the software for a device with its hardware. Of course, code is relatively easy to share and significant pieces of software (like Linux and Firefox) have been written in a distributed open-source fashion. In customizing the design of a personally-fabricated product, it's possible to tweak the software that goes with it, too. If this software is large and complex, though, even small changes to the hardware might involve complicated software development and testing efforts.

Better Tools for Supporting Novice Embedded Programmers

Much research has looked more specifically at the ways that people learn to program and the skills and mindsets they learn in doing so. Other work has built tools to simplify the process of programming and make it more accessible and engaging. Systems like Scratch, for example, that allow users to snap together visual building blocks instead of typing in code would help avoid syntax errors. Scratch also visually displays the available blocks, helping people get a sense of and explore the possibilities. I think there's a lot of opportunity to apply this work to embedded development. This domain can provide unique motivations for learning to program as well as unique technical challenges.

On the other hand, the personal fabrication approach of the workshop involved working with a wide variety of electronic components and their associated libraries. A visual programming environment that simplified the programming process by limiting the available functionality would not have been well-suited to this kind of open-ended component selection. Instead, there's a need to simplify the programming process while still providing an open-ended tool. In addition, there are efficiency concerns, as many of the microcontrollers that can be soldered by hand have relatively slow speeds and little memory. Balancing these conflicting requirements offers a good challenge for future work.

Custom Products for Specific User Groups

Personal fabrication seems particularly well-suited to people or situations requiring custom solutions. It would be instructive to focus on a particular domain — for example, individuals engaged in high-performance sports, or with particular physical disabilities — and create custom electronic products (or other objects) for specific individuals. This would take advantage of personal fabrication's ability to create prototypes for real-world use and to rapidly iterate on designs based on these experiences.

Ongoing Challenges

It's not clear what the future holds for personal fabrication. The pace at which technologies of digital fabrication and embedded computation are evolving shows few signs of slowing down (notwithstanding the impossibility of the exponential growth of Moore's law continuing forever).The extent to which these improvements will extend the capability of individuals and the possibilities for personal fabrication of electronic products, however, is not so easy to predict. Here are three questions about the future of personal fabrication — questions that I hope will encourage us to think about the future we'd like to see and to work toward making it a reality:

Will the technologies that can be made by individuals keep pace with those produced by large companies? Although technology continues to improve, it doesn't necessarily do so in ways that are accessible to everyone. As a result,it's unclear to what extent personal fabrication and DIY will be able to keep up with the devices that are produced and sold by large companies. While the potential scope of personal fabrication continues to expand as technology improves, the gap between it and proprietary products may limit the extent to which it can serve as a feasible substitute for them. We should remember that the decisions we make influence the potential scope of personal fabrication. If we encourage manufacturers to make their technologies available to individuals, support open tools, make use of open standards, and make our own hardware open source, we can expand that extent to which individuals are able to create, modify, and control the technologies they use in their lives.

Will we find diverse applications for the technological possibilities? While personal fabrication and DIY electronics allow for many activities and outputs, it's less clear to what extent these possibilities will provide meaningful for diverse audiences. For early adopters, it's easy to get caught up in the technologies themselves as opposed to their potential contexts and applications. This interest in the technology itself can be helpful, as its uses may not be immediately clear or accessible. Even so, this emphasis on technology for technology's sake will not appeal to everyone.Thus, as we think about the future of personal fabrication, we should remember to not just play with the technology, but also find ways to make it relevant and useful to new people and situations. In part, this evolution may happen naturally as technologies mature and we come to take them for granted but it also relies on those of us with early access to and expertise in technology to think about how to make it relevant and useful to others.

Will distributed collaboration on hardware get easier? Although there are exceptions, open source hardware currently seems less likely than other domains (e.g., open source software) to involve collaboration between many individuals on a centralized design or repository, in which small contributions are combined together into a complex whole. Although there are many reasons for this pattern, if open source hardware is to thrive, it seems crucial to facilitate better collaboration between large numbers of distributed and diverse individuals.This will require improved tools (discussed in "Future Work" above), more efficient processes, and, perhaps most importantly, a focus on fostering communities that have a shared interest in the development of open source hardware.

Depending on the answers to these questions, the future of personal fabrication will look very different. My hope is that we will find ways to make it increasingly relevant and valuable, by expanding the technologies it can make use of, the applications and contexts to which it can be applied, and the collaborations that can develop around it. If our practices can keep pace with the growth of technology, personal fabrication should offer a powerful alternative to mass production for making technology in our lives.

Fostering Agency and Empowerment?

When we don't understand the electronic products that increasingly make up our world, we can experience a loss of agency with respect to our technological surroundings. As Tyler put it:

> "I think if you're mystified and baffled by the world around you, whatever it is, it could be electronics, it could be something completely different, I think that gives you a lot less agency in your day to day life.... That's how I would pitch it. Know more about the stuff around you that has such a strong effect on you."

While there are many values for the personal fabrication of electronic products (discussed in Chapter 7), fostering a sense of agency with respect to the devices in one's life is, for me, probably the most important. Unfortunately, this goal is not necessarily easy or straightforward. Making a device for oneself shows that this is possible — a source of empowerment — but it also points out the gulf between the capabilities of the individual and those of industry — a source of disempowerment. As individuals, we may be capable of doing much more than we thought possible but far less than we might want.

References

Atkinson, Paul. 2006. Do It Yourself: Democracy and Design. In *Journal of Design History* (Spring 2006) 19 (1): 1-10.

Ballagas, Rafael, Meredith Ringel, Maureen Stone, and Jan Borchers. 2003. iStuff: a physical user interface toolkit for ubiquitous computing environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '03). ACM, New York, NY, USA, 537-544.

Barragán, Hernando. 2004. Wiring: Prototyping physical interaction design. Master's Thesis. Interaction Design Institute Ivrea, Italy.

Bean, Jonathan and Daniela Rosner. 2014. Making: movement or brand? in Interactions 21, 1 (January 2014), 26-27.

Benkler, Yochai. Coase's Penguin, or, Linux and the Nature of the Firm. Yale Law Journal, 112, 3 (2002), 367–445.

Brennan, Karen and Mitchel Resnick. 2012. "Using artifact-based interviews to study the development of computational thinking in interactive media design." Paper presented at annual American Educational Research Association meeting, Vancouver, BC, Canada.

Buechley, Leah and Hannah Perner-Wilson. 2012. Crafting technology: Reimagining the processes, materials, and cultures of electronics. *ACM Transactions on Computer-Human Interaction.* 19, 3, Article 21 (October 2012), 21 pages.

Buechley, Leah and Benjamin Mako Hill. 2010. LilyPad in the wild: how hardware's long tail is supporting new engineering and design communities. In *Proceedings of the 8th ACM Conference on Designing Interactive Systems* (DIS '10). ACM, New York, NY, USA, 199-207.

Buechley, Leah, Daniela Rosner, Eric Paulos, and Amanda Williams. 2009. DIY for CHI: methods, communities, and values of reuse and customization. In CHI EA '09, ACM (2009), 4823–4826.

Carpo, Mario. 2011. *The Alphabet and the Algorithm.* MIT Press, Cambridge, MA, USA.

Cottam, Matt. 2009. Wooden logic: In search of heirloom electronics. Master's Thesis. Umeå Institute of Design, Umeå University.

de Bruijn, Erik. 2010. "On the Viability of the Open Source Development Model for the Design of Physical Objects: Lessons Learned From the RepRap Project". Master's Thesis. Tilburg University.

Dewey, John. 1938. *Experience and Education.* Kappa Delta Pi.

Follmer, Sean, David Carr, Emily Lovell, and Hiroshi Ishii. 2010. CopyCAD: remixing physical objects with copy and paste from the real world. In *Adjunct proceedings of the 23nd annual ACM symposium on User interface software and technology* (UIST '10). ACM, New York, NY, USA, 381-382.

Follmer, Sean and Hiroshi Ishii. 2012. KidCAD: digitally remixing toys through tangible tools. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '12). ACM, New York, NY, USA, 2401-2410.

Gauntlett, David. 2011. *Making is Connecting.* Polity, Cambridge, UK.

Gelber, Steven M. 1999. *Hobbies: Leisure and the Culture of Work in America.* Columbia University Press, New York, NY, USA.

Gershenfeld, Neil. 2007. *Fab: The Coming Revolution on Your Desktop – From Personal Computers to Personal Fabrication.* Basic Books, Inc., New York, NY, USA.

Goodman, Elizabeth and Daniela Rosner. 2011. From garments to gardens: negotiating material relationships online and 'by hand'. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '11). ACM, New York, NY, USA, 2257-2266.

Greenberg, Saul and Chester Fitchett. 2001. Phidgets: easy development of physical interfaces through physical widgets. In *Proceedings of the 14th annual ACM symposium on User interface software and technology* (UIST '01). ACM, New York, NY, USA, 209-218.

Haring, Kristen. 2007. *Ham Radio's Technical Culture.* MIT Press, Cambridge, MA, USA.

Hartmann, Björn, Scott R. Klemmer, Michael Bernstein, Leith Abdulla, Brandon Burr, Avi Robinson-Mosher, and Jennifer Gee. 2006. Reflective physical prototyping through integrated design, test, and analysis. In *Proceedings of the 19th annual ACM symposium on User interface software and technology* (UIST '06). ACM, New York, NY, USA, 299-308.

Hartmann, Björn, Daniel MacDougall, Joel Brandt, and Scott R. Klemmer. 2010. What would other programmers do: suggesting solutions to error messages. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '10). ACM, New York, NY, USA, 1019-1028.

Hodges, Steve, Stuart Taylor, Nicolas Villar, James Scott, and John Helmes. 2013. Exploring physical prototyping techniques for functional devices using .NET gadgeteer. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction* (TEI '13). ACM, New York, NY, USA, 271-274.

Hook, Jonathan, Thomas Nappey, Steve Hodges, Peter Wright, and Patrick Olivier. 2014. Making 3D printed objects interactive using wireless accelerometers. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '14). ACM, New York, NY, USA, 1435-1440.

Jacobs, Jennifer and Leah Buechley. 2013. Codeable objects: computational design and digital fabrication for novice programmers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '13). ACM, New York, NY, USA, 1589-1598.

Kafai, Yasmin B., Eunkyoung Lee, Kristin Searle, Deborah Fields, Eliot Kaplan, and Debora Lui. 2014. A Crafts-Oriented Approach to Computing in High School: Introducing Computational Concepts, Practices, and Perspectives with Electronic Textiles. *ACM Transactions on Computing Education.* 14, 1, Article 1 (March 2014), 20 pages.

Kawahara, Yoshihiro, Steve Hodges, Benjamin S. Cook, Cheng Zhang, and Gregory D. Abowd. 2013. Instant inkjet circuits: lab-based inkjet printing to support rapid prototyping of UbiComp devices. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing* (UbiComp '13). ACM, New York, NY, USA, 363-372.

Kelleher, Caitlin and Randy Pausch. 2005. Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. ACM Comput. Surv. 37, 2 (June 2005), 83-137.

Knörig, André, Reto Wettach, and Jonathan Cohen. 2009. Fritzing: a tool for advancing electronic prototyping for designers. In *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction* (TEI '09). ACM, New York, NY, USA, 351-358.

Kuznetsov, Stacey and Eric Paulos. 2010. Rise of the expert amateur: DIY projects, communities, and cultures. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries* (NordiCHI '10). ACM, New York, NY, USA, 295-304.

Kuznetsov, Stacey, George Noel Davis, Eric Paulos, Mark D. Gross, and Jian Chiu Cheung. 2011. Red balloon, green balloon, sensors in the sky. In *Proceedings of the 13th international conference on*

Ubiquitous computing (UbiComp '11). ACM, New York, NY, USA, 237-246.

Lee, Johnny C., Daniel Avrahami, Scott E. Hudson, Jodi Forlizzi, Paul H. Dietz, and Darren Leigh. 2004. The calder toolkit: wired and wireless components for rapidly prototyping interactive devices. In *Proceedings of the 5th conference on Designing interactive systems: processes, practices, methods, and techniques* (DIS '04). ACM, New York, NY, USA, 167-175.

Johnson, Gabe. 2008. FlatCAD and FlatLang: Kits by code. In *IEEE Symposium on Visual Languages and Human-Centric Computing,* 15-19 Sept. 2008 (VL/HCC '08), pp.117–120.

Lévi-Strauss, Claude. 1966. *The Savage Mind.* University of Chicago Press, Chicago, IL. (Originally published as *La Pensée Sauvage,* Librairie Plon, Paris, 1962.)

Levy, Steven. 1984. *Hackers: Heroes of the Computer Revolution.* Doubleday.

Lindtner, Silvia, Garnet D. Hertz, and Paul Dourish. 2014. Emerging sites of HCI innovation: hackerspaces, hardware startups & incubators. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '14). ACM, New York, NY, USA, 439-448.

Maestri, Leah and Ron Wakkary. 2011. Understanding repair as a creative process of everyday design. In *Proceedings of the 8th ACM conference on Creativity and cognition* (C&C '11). ACM, New York, NY, USA, 81-90.

Martin, F., B. Mikhak, and B. Silverman. 2000. MetaCricket: a designer's kit for making computational devices. IBM Systems Journal, 39, 3-4 (July 2000), 795-815.

Martin, Fred and Mitchel Resnick. 1993. LEGO/Logo and electronic bricks: creating a scienceland for children. In: Ferguson D (ed) *Advanced educational technologies for mathematics and science.* Springer, Berlin Heidelberg New York.

McCullough, Malcolm. 1998. *Abstracting Craft: The Practiced Digital Hand.* MIT Press, Cambridge, MA, USA. (First published in 1996.)

Mellis, David A., Dana Gordon, and Leah Buechley. 2010. Fab FM: the design, making, and modification of an open-source electronic product. In Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction (TEI '11). ACM, New York, NY, USA, 81-84.

Mellis, David Adley. 2011. Case Studies in the Digital Fabrication of Open-Source Consumer Electronic Products. Master's Thesis. Massachusetts Institute of Technology.

Mellis, David A. and Leah Buechley. 2012. Case studies in the personal fabrication of electronic products. In *Proceedings of the Designing Interactive Systems Conference* (DIS '12). ACM, New York, NY, USA, 268-277.

Mellis, David, Sean Follmer, Bjoern Hartmann, Leah Buechley, and Mark D. Gross. 2013. FAB at CHI: digital fabrication tools, design, and community. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '13). ACM, New York, NY, USA, 3307-3310.

Mellis, David A. and Leah Buechley. 2014. Do-it-yourself cellphones: an investigation into the possibilities and limits of high-tech diy. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '14). ACM, New York, NY, USA, 1723-1732.

Moriwaki, Katherine and Jonah Brucker-Cohen. "Lessons from the scrapyard: creative uses of found materials within a workshop setting." AI & SOCIETY 20.4 (2006): 506-525.

Mota, Catarina. 2011. The rise of personal fabrication. In *Proceedings of the 8th ACM conference on Creativity and cognition* (C&C '11). ACM, New York, NY, USA, 279-288.

Mueller, Stefanie, Pedro Lopes, and Patrick Baudisch. 2012. Interactive construction: interactive fabrication of functional mechanical devices. In *Proceedings of the 25th annual ACM symposium on User interface software and technology* (UIST '12). ACM, New York, NY, USA, 599-606.

Myers, Brad, Scott E. Hudson, and Randy Pausch. 2000. Past, present, and future of user interface software tools. *ACM Transactions on Computer-Human Interaction,* 7, 1 (March 2000), 3-28.

Nardi, Bonnie A.. 1993. *A Small Matter of Programming: Perspectives on End User Computing.* MIT Press, Cambridge, MA, USA.

Oh, Yeonjoo, Gabriel Johnson, Mark D. Gross, and Ellen Yi-Luen Do. 2006. "The Designosaur and the Furniture Factory," Design Computing and Cognition, 2006, Part 2, 123- 140.

Papert, Seymour. 1980. *Mindstorms: Children, computers, and powerful ideas.* Basic Books, Inc.

Papert, Seymour and Idit Harel. 1991. "Situating Constructionism" in *Constructionism* (eds. Idit Harel and Seymour Papert). Ablex Publishing, Norwood, NJ. Accessed online at: http://www.papert.org/articles/SituatingConstructionism.html

Perner-Wilson, Hannah , Leah Buechley, and Mika Satomi. 2010. Handcrafting textile interfaces from a kit-of-no-parts. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction* (TEI '11). ACM, New York, NY, USA, 61-68.

Qi, Jie and Leah Buechley. 2014. Sketching in circuits: designing and building electronics on paper. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '14). ACM, New York, NY, USA, 1713-1722.

Qiu, Kanjun, Leah Buechley, Edward Baafi, and Wendy Dubow. 2013. A curriculum for teaching computer science through computational textiles. In *Proceedings of the 12th International Conference on Interaction Design and Children* (IDC '13). ACM, New York, NY, USA, 20-27.

Raasch, Christina, Cornelius Herstatt and Kerstin Balka. On the open design of tangible goods. R&D Management 39, 4 (2009), 382–393.

Ratto, Matt. 2011. Critical Making: Conceptual and Material Studies in Technology and Social Life. In The Information Society: An International Journal, 27:4, 252-260.

Resnick, Mitchel. Behavior construction kits. In *Communications of the ACM* 36, 7 (July 1993), 64–71.

Resnick, Mitchel, Amy Bruckman, and Fred Martin. Pianos not stereos: creating computational construction kits. In *Interactions* 3, 5 (September 1996), 40-50.

Resnick, Mitchel, Fred Martin, Robert Berg, Rick Borovoy, Vanessa Colella, Kwin Kramer, and Brian Silverman. 1998. Digital manipulatives: new toys to think with. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '98), Clare-Marie Karat, Arnold Lund, Joëlle Coutaz, and John Karat (Eds.). ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 281-287.

Resnick, Mitchel and Brian Silverman. 2005. Some reflections on designing construction kits for kids. In *Proceedings of the 2005 conference on Interaction design and children* (IDC '05). ACM, New York, NY, USA, 117-122.

Rosenbaum, Eric. 2015. Explorations in Musical Tinkering. PhD Dissertation. Massachusetts Institute of Technology.

Rosner, Daniela and Jonathan Bean. 2009. Learning from IKEA hacking: i'm not one to decoupage a tabletop and call it a day. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '09). ACM, New York, NY, USA, 419-422.

Ross, Douglas T. 1958. Papers on Automatic Programming for Numerically Controlled Machine Tools. Servomechanisms Laboratory, MIT Dept. of Electrical Engineering.

Sadler, Joel, Kevin Durfee, Lauren Shluzas, and Paulo Blikstein. 2015. Bloctopus: A Novice Modular Sensor System for Playful Prototyping. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction* (TEI '15). ACM, New York, NY, USA, 347-354.

Saul, Greg, Manfred Lau, Jun Mitani, and Takeo Igarashi. 2010. SketchChair: an all-in-one chair design system for end users. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction* (TEI '11). ACM, New York, NY, USA, 73-80.

Savage, Valkyrie, Xiaohan Zhang, and Björn Hartmann. 2012. Midas: fabricating custom capacitive touch sensors to prototype interactive objects. In Proceedings of the 25th annual ACM symposium on User interface software and technology (UIST '12). ACM, New York, NY, USA, 579-588.

Savage, Valkyrie, Colin Chang, and Björn Hartmann. 2013. Sauron: embedded single-camera sensing of printed physical user interfaces. In *Proceedings of the 26th annual ACM symposium on User interface software and technology* (UIST '13). ACM, New York, NY, USA, 447-456.

Schmidt, Ryan and Karan Singh. 2010. meshmixer: an interface for rapid mesh composition. In *ACM SIGGRAPH 2010 Talks* (SIGGRAPH '10). ACM, New York, NY, USA, 1 page.

Schmitt, Peter. 2011. originalMachines: developing tools and methods for object-oriented mechatronics. PhD Dissertation. Massachusetts Institute of Technology.

Schmitt, Peter and Susanne Seitinger. 2009. Plywood punk: a holistic approach to designing animated artifacts. In *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction* (TEI '09). ACM, New York, NY, USA, 123-126.

Silver, Jay. 2009. Awakening to maker methodology: the metamorphosis of a curious caterpillar. In *Proceedings of the 8th International Conference on Interaction Design and Children* (IDC '09). ACM, New York, NY, USA, 242-245.

Sun, Yuling, Silvia Lindtner, Xianghua Ding, Tun Lu, and Ning Gu. 2015. Reliving the Past & Making a Harmonious Society Today: A Study of Elderly Electronic Hackers in China. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing* (CSCW '15). ACM, New York, NY, USA, 44-55.

Tanenbaum, Joshua, Karen Tanenbaum, and Ron Wakkary. 2012. Steampunk as design fiction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '12). ACM, New York, NY, USA, 1583-1592.

Tanenbaum, Joshua G., Amanda M. Williams, Audrey Desjardins, and Karen Tanenbaum. 2013. Democratizing technology: pleasure, utility and expressiveness in DIY and maker practice. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '13). ACM, New York, NY, USA, 2603-2612.

Torrey, Cristen, Elizabeth F. Churchill, and David W. McDonald. 2009. Learning how: the search for craft knowledge on the internet. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '09). ACM, New York, NY, USA, 1371-1380.

Turkle, Sherry and Seymour Papert. 1991. "Epistemological Pluralism and the Revaluation of the Concrete" in *Constructionism* (eds. Idit Harel and Seymour Papert). Ablex Publishing, Norwood, NJ. Accessed online at: http://www.papert.org/articles/EpistemologicalPluralism.html

Villar, Nicolas, Kiel Mark Gilleade, Devina Ramduny-Ellis, and Hans Gellersen. 2006. The VoodooIO gaming kit: a real-time adaptable gaming controller. In *Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology* (ACE '06). ACM, New York, NY, USA, , Article 82 .

Villar, Nicolas, James Scott, Steve Hodges, Kerry Hammil, and Colin Miller. 2012. .NET gadgeteer: a platform for custom devices. In *Proceedings of the 10th international conference on Pervasive Computing* (Pervasive'12), Judy Kay, Paul Lukowicz, Hideyuki Tokuda, Patrick Olivier, and Antonio Krüger (Eds.). Springer-Verlag, Berlin, Heidelberg, 216-233.

von Hippel, Eric, and Ralph Katz. "Shifting Innovation to Users via Toolkits." Management Science 48, no. 7 (July 2002): 821–833.

Weichel, Christian, Manfred Lau, and Hans Gellersen. 2013. Enclosed: a component-centric interface for designing prototype enclosures. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction* (TEI '13). ACM, New York, NY, USA, 215-218.

Weichel, Christian, Manfred Lau, David Kim, Nicolas Villar, and Hans W. Gellersen. 2014. MixFab: a mixed-reality environment for personal fabrication. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '14). ACM, New York, NY, USA, 3855-3864.

Willis, Karl, Eric Brockmeyer, Scott Hudson, and Ivan Poupyrev. 2012. Printed optics: 3D printing of embedded optical elements for interactive devices. In *Proceedings of the 25th annual ACM symposium on User interface software and technology* (UIST '12). ACM, New York, NY, USA, 589-598.

Zoran, Amit and Joseph A. Paradiso. 2013. FreeD: a freehand digital sculpting tool. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '13). ACM, New York, NY, USA, 2613-2616.