Microcontrollers as Material: Crafting Circuits with Paper, Conductive Ink, Electronic Components, and an "Untoolkit"

David A. Mellis, Sam Jacoby, Leah Buechley, Hannah Perner-Wilson, Jie Qi
MIT Media Lab
75 Amherst St., Cambridge, MA 02142
{mellis, sjacoby, leah, plusea, jieqi}@media.mit.edu

ABSTRACT
Embedded programming is typically made accessible through modular electronics toolkits. In this paper, we explore an alternative approach, combining microcontrollers with craft materials and processes as a means of bringing new groups of people and skills to technology production. We have developed simple and robust techniques for drawing circuits with conductive ink on paper, enabling off-the-shelf electronic components to be embedded directly into interactive artifacts. We have also developed an set of hardware and software tools – an instance of what we call an "untoolkit" – to provide an accessible toolchain for the programming of microcontrollers. We evaluated our techniques in a number of workshops, one of which is detailed in the paper. Four broader themes emerge: accessibility and appeal, the integration of craft and technology, microcontrollers vs. electronic toolkits, and the relationship between programming and physical artifacts. We also expand more generally on the idea of an untoolkit, offering a definition and some design principles, as well as suggest potential areas of future research.

Author Keywords
microcontrollers, craft, paper, conductive ink, toolkits, Arduino

ACM Classification Keywords
J.5 Arts and Humanities

General Terms
Design, Human Factors

INTRODUCTION
The modes of production in craft and electronics differ dramatically. Craft and other creative media (e.g. painting, sculpture, or wood-working) involve the manual use of tools on raw material, allowing for infinite and subtle variation. They emphasize creative expression, diversity, manual skill, and individual autonomy. The same tools are used by experts and amateurs. Through continued practice and experimentation, an individual can develop their abilities, and, eventually, master the medium. This diversity of craft processes and outcomes offers opportunities to engage many different people at many different levels.

In contrast, in the domain of electronics, many attempts at education and accessibility take the form of toolkits: collections of standardized building blocks designed specifically for novices. While these modules allow for fast, easy construction and experimentation, they impose a number of limitations and constraints. In packaging electronic components into higher-level modules, toolkits can obscure the technology they seek to make accessible. In addition, modular toolkits typically introduce their own, proprietary means of making connections, making them difficult to interface with other toolkits or with off-the-shelf components. In most cases, users are limited to the set of modules provided by the maker of the toolkit. The size and shape of the modules also constrains the aesthetics and forms of the artifacts one can make. The toolkit itself—unless concealed—typically remains a substantial, if not central aesthetic component of any project. Even users expert with a particular toolkit may remain locked in by its constraints. There is a significant discontinuity in moving from a toolkit's modules to working directly with the underlying components.

Of course, many of these limitations also apply to off-the-shelf electronic components like standalone microcontrollers. In our discussion of toolkits, we focus on those higher-level modules specifically intended for educational or artistic practice rather than the standard components found in industrially-produced devices. It is those toolkits that are designed with pedagogical or creative affordances in mind and it is those affordances that we're interested in exploring and improving. In contrast, components like microcontrollers are optimized for mass production, emphasizing standardization and reliability.

In this paper, we discuss our efforts to overcome some of the limitations of these electronics toolkits through craft practice. In particular, through a series of workshops, we have developed techniques for drawing circuits using conductive ink on paper and embedding off-the-shelf electronic components. This serves as an accessible method for crafting

TEI 2013, February 10 - 13, 2013, Barcelona, Spain.

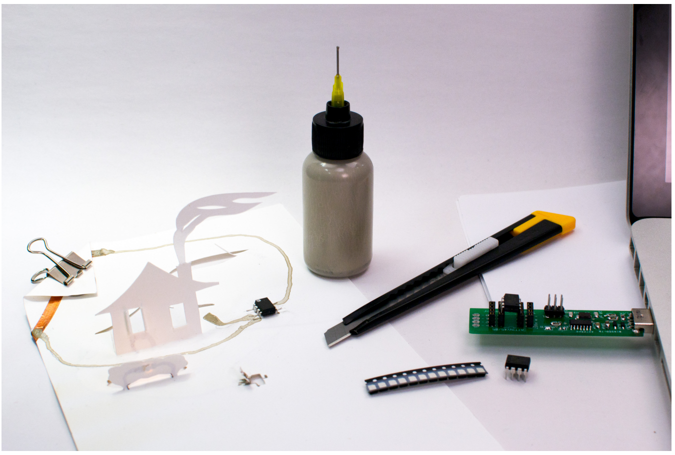

Figure 1: Workshop materials and example

self-contained interactive objects. We've also developed a complementary set of tools to facilitate the programming of the microcontrollers included in these circuits. Because these tools provide scaffolding similar to that of a toolkit, but are not embedded in the final artifacts produced, we refer to them as an "untoolkit," a concept we develop later in the paper. To probe more deeply the combination of these circuit construction techniques with our tools for microcontroller programming, we conducted a final workshop, which we describe in detail below. This workshop yielded insights about using craft – instead of high-level electronics toolkits – to introduce microcontrollers and circuit design.

In the following section, we discuss related work, both on electronics toolkits and craft approaches to technology. We then describe in more detail our techniques for constructing circuits with conductive ink and our microcontroller untoolkit. This is followed by an account of the workshop, which provides the basis for a longer discussion about people's relationships with technology and our techniques. We then offer a generalized definition of an untoolkit and some principles for their design. This is followed by a discussion of opportunities for further research and a short conclusion.

RELATED WORK

Here we discuss related work in the areas of electronic toolkits, craft approaches to technology, and paper circuits.

Electronic Toolkits

There's a long history of electronic toolkits, both in research and as commercial products. The Braun Lectron, designed by Dieter Rams, is an early commercial example. Inspired by Seymour Papert and Valentino Braitenberg, Mitchel Resnick and his colleagues developed a series of toolkits to introduce children to programming and engineering. They include the electronic and programmable bricks [19] and the Crickets [20], which helped inspire the Lego Mindstorms robotics products. Phidgets [8] and Calder [13] are two examples of toolkits intended to help designers in the prototyping process. The d.tools platform [9] complements an electronics toolkit with on-screen interface tools, while .NET Gadgeteer [23] augments their electronics modules with software for developing custom 3D-printed enclosures. The Basic Stamp and Arduino are two popular platforms that combine a toolkit-like microcontroller module with the use of breadboards off-the-shelf electronic components. In [1], the authors distinguish this approach (which they call the "breakout model") from toolkits which offer also custom sensing and actuating modules (the "Cricket model"). In working directly with microcontrollers, we seek to extend the advantages of the breakout model by removing the need for any custom modules. We are, however, inspired and guided by many of the goals and principles of the toolkits mentioned here.

Craft Approaches to Technology

More recently, researchers have begun to experiment with ways to more tightly couple electronics with craft or other hacking and do-it-yourself (DIY) practice. For example, the practice of circuit bending [7] encourages people to hack existing musical devices in order to create new interfaces and sounds. The Scrapyard Challenge [14] series of workshops similarly encourages participants to reuse existing objects to create interfaces for musical control. In [10], the authors discuss methods for rapidly creating interfaces from cardboard, tinfoil, and pushpins. Squishy Circuits [11] is a technique for creating electronic circuits with conductive and insulating play-dough.

An area of research with particular relevance is the domain of e-textiles, which integrates electronic components with conductive threads, fabrics, and other soft materials. In [2], the authors describe their techniques for and experiences with these practices. These practices are supported by the LilyPad Arduino toolkit [3], a set of modules for e-textiles. In [16], the authors describe a set of techniques for creating textile sensors using a range of conductive materials and craft techniques. The EduWear project [12] discusses the effect of e-textile experience on children's attitudes towards technology. We draw inspiration from the ability of e-textiles to merge electronics into artifacts whose aesthetics are shaped by their textile and craft components and to engage different skills and groups than other forms of technology construction. The value of this integration of craft and technology is discussed in more detail in [5].

Paper Circuits

Other work has looked at specifically at the integration of electronics with paper. In [4], the authors describe a toolkit of magnetic components that can be moved between circuits made with conductive ink. In [6], the authors show techniques for embedded electronic components into hand-made paper. Saul et al. [22] describe a set of technological paper artifacts, including robots with circuits made from goldleaf. Russo et al. [21] describe a conductive ink pen they've developed for drawing circuits on paper. Qi and Buechley ([17], [18]) describe a set of artistic artifacts made using paper circuits. Graffiti Research Lab [1] have done multiple projects using conductive paint and other materials to create circuits on walls and other surfaces. Perner-Wilson [15] describes many different approaches to the integration of craft

[1]http://www.graffitiresearchlab.com/

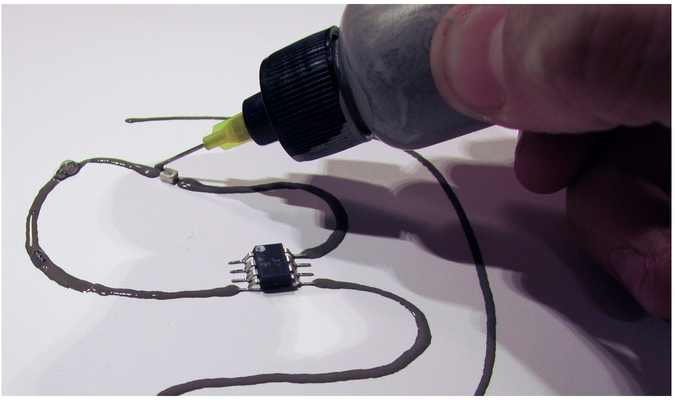

Figure 2: Constructing a circuit with silver ink

and technology, including the use of conductive paint on various substrates. Our work has focused on finding ways to make these materials accessible and reliable in conjunction with bare microcontrollers and other off-the-shelf components, allowing us to try them with others.

PAPER AND CONDUCTIVE INK CIRCUITS

We construct our circuits using conductive ink on paper, connecting various off-the-shelf electronic components. After experimentation with a variety of approaches, we've found that dispensing silver ink from syringe-tipped squeeze bottles offers a good balance of cost, reliability, conductivity and control. We use the WB-101 water-based silver conductive ink from Conductive Compounds, thinned with distilled water to a 10:1 ink-to-water ratio by weight (e.g. 20 grams of ink to 2 grams of water). The diluted mixture is loaded into 1-ounce squeezable bottles and dispensed through disposable 20-gauge syringe tips. So prepared, a line roughly 1mm in diameter can be dispensed on a variety of surfaces. After five to ten minutes drying-time, the line has a resistivity of 0̃.3 Ohm/cm, which is adequate for circuits constructed on a sheet of letter-size or A4 paper.

When dry, the ink adheres well to both the paper as well as the metal leads of electronic components, forming durable conductive bonds. We include components like microcontrollers, LEDs (e.g. 1206 surface-mount packages), and coin-cell batteries (like the three-volt 2032), along with sensors like capacitive electrodes, microphones and light sensors (LDRs). All of these components are available for less than a dollar (USD). In order to integrate through-hole (DIP) microcontrollers, we bend their legs outwards and glue the plastic package directly to the paper. We then trace lines of conductive ink from the legs of the microcontroller onto the paper using the squeeze bottles. Similar techniques apply to components like LEDs or light sensors. We've developed a range of techniques for holding the battery, from simply securing it in place with a binder clip to enclosing it in a laser-cut paper pouch.

We've tested these techniques with a variety of people and settings. In one workshop, participants made interactive LEDs lamps with pre-programmed microcontrollers. In another, children constructed simple circuits containing just an LED and battery—an activity taking only 10 to 15 minutes. These preliminary workshops didn't involve any programming (or, indeed, the use of computers), allowing us to focus specifically on the techniques themselves and the integration of papercraft with electronics. Participants' engagement with these activities and the quality of the artifacts they produced gave us confidence in our techniques and a desire to extend them with the inclusion of microcontroller programming.

OUR MICROCONTROLLER UNTOOLKIT

To make standalone microcontrollers—specifically the ATtiny45 and ATtiny85 from Atmel—as easy to use as an Arduino board, we developed a hardware programmer and two software tools. These tools provide similar scaffolding to a toolkit but work directly with bare microcontrollers (which cost dramatically less). They allow someone to upload and run a program on a microcontroller without a custom circuit and without any programming, or with only a few lines of simple code. This allows the user to focus on the appearance and behavior of the artifact they're making. Because they build on standard tools, however, our hardware and software allow someone to gradually progress to expert embedded development. These tools inspired our general concept of an untoolkit, discussed in detail later.

TinyProgrammer

The TinyProgrammer loads compiled programs from the computer onto an ATtiny45 or ATtiny85 microcontroller, without the need for any additional connections or components. The microcontroller goes directly into a socket on the TinyProgrammer, which plugs into the USB port of a computer. The TinyProgrammer provides power and all necessary connections to the ATtiny45/85. It also provides additional headers breaking out each pin of the microcontroller, facilitating testing of uploaded programs while the microcontroller remains in the TinyProgrammer. The TinyProgrammer uses the open-source software projects V-USB and USBtiny and is derived from the open-source hardware USBtinyISP. The TinyProgrammer's hardware and software are, in turn, open-source and freely available for download

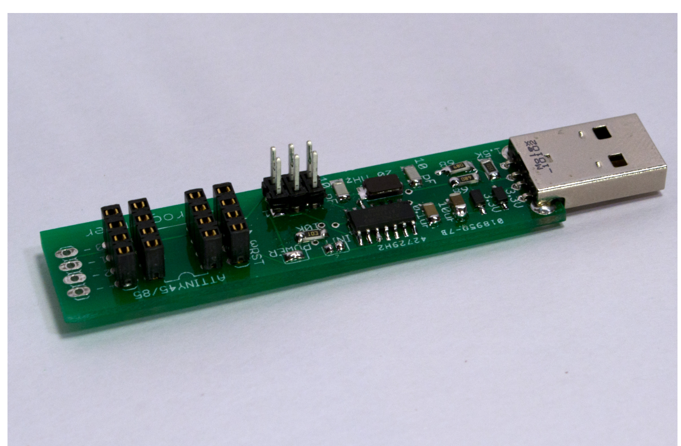

Figure 3: TinyProgrammer

Figure 4: TinyUploader Software

online[2]. We are also working to make the TinyProgrammer commercially available.

TinyUploader software

The TinyUploader software allows users to upload a set of pre-compiled programs to the microcontroller (via the TinyProgrammer or other programmer). This introduces newcomers to idea of loading a program onto a microcontroller without requiring them to first write (or even view) any code. Drop-downs in the software allow for the selection of the program and the choice of microcontroller (ATtiny45 or ATtiny85). Internally, the TinyUploader uses the standard avrdude command-line tool, translatings its sometimes-cryptic error messages into clear, human-readable language. The TinyUploader is open-source[3] and based on the open-source ReflashBlinkM software.

Arduino support for the ATiny45/85

The Arduino software provides a core library of functions that ease the process of programming AVR microcontrollers. We developed an open-source plugin[4] for the Arduino software that allows it to program the ATtiny45 and 85. Once installed, the user can compile and upload programs for the ATtiny microcontrollers in the same way that they would for an Arduino board. (Not all the functionality of the Arduino software is supported; for example, the ATtiny lacks a hardware serial port and I^2C hardware.) By building on Arduino, we can take advantage of existing examples and documentation, introducing people to an active development community and providing pathways for future development.

WORKSHOP

To test our techniques for paper circuits and our microcontroller untoolkit, we conducted a workshop entitled "Papercraft and Programming." Through a series of guided activities, participants programmed and designed circuits on paper, producing a set of small, interactive projects. Surveys, as well as interrogative group-discussions, were conducted before and after the workshop to solicit feedback on participants' understanding and perceptions of the materials, as well as technology-production as a whole.

A dozen participants were recruited from the local community by way of flyers, as well as from the authors' public email lists. Participants ages varied from 23 to 60. Ten of the twelve were women. All had attended at least some college, and seven had at least a bachelor's degree. Technical backgrounds varied considerably, with one participant working professionally as a computer programmer, and others never having seen a line of code. Among the group, were a retiree, a teacher, an administrative assistant, and a chef.

After an initial discussion, participants were guided through the use of TinyProgrammer and TinyUploader, uploading one of a small library of pre-compiled programs. Each participant configured the required software and programmed an ATtiny for their own use. We then introduced squeeze bottles with conductive silver ink, as well as a simple paper template on which to affix a battery, the microcontroller itself, and an LED. Workshop participants familiarized themselves with the basic principles of design with the materials, and completed a small circuit. We then introduced the Arduino IDE, a conventional programming environment, and the basics of embedded-systems C-programming with the Arduino libraries. After writing a small program, either from scratch or from included examples, participants constructed a papercraft project.

The final artifacts that participants constructed expressed a wide variety of interests. One woman built a busy street-scene with blinking lights on paper cut-out cars, another, a flickering campfire scene. One participant, drawing on her own art experience, drew a human figure with a blinking eye. Other projects were more utilitarian: a small, working lamp and an interactive greeting card. While some chose to use the conductive ink as a design element, others chose to hide it beneath layers of paper. All projects were self-contained, incorporating a microcontroller, one or more LEDs, and battery. The microcontrollers were programmed with different behaviors, from simply fading or blinking various lights to responding to touch input.

DISCUSSION

Over the course of the workshop, four central points of interest emerged. We focused on the accessibility and appeal of our techniques, the affordances of craft processes in making technology, participants' relationship with programming, and the relationship between toolkits and off-the-shelf electronic components.

Accessibility and Appeal

Working with paper and conductive ink provides new avenues to engage people's interest. Some participants were intrigued by the papercraft from the beginning: "We were like, 'papercrafting, that sounds awesome . . . let's go.' I don't think we even read what it was about. But if you had started with 'Microcontroller Adjustment of Paper Material,' I would have been like I don't know what that means." But papercraft isn't just appealing to those specifically interested in the medium. The familiarity of paper and ink provides

[2] http://www.thingiverse.com/thing:12461
[3] http://github.com/damellis/TinyUploader
[4] http://github.com/damellis/attiny

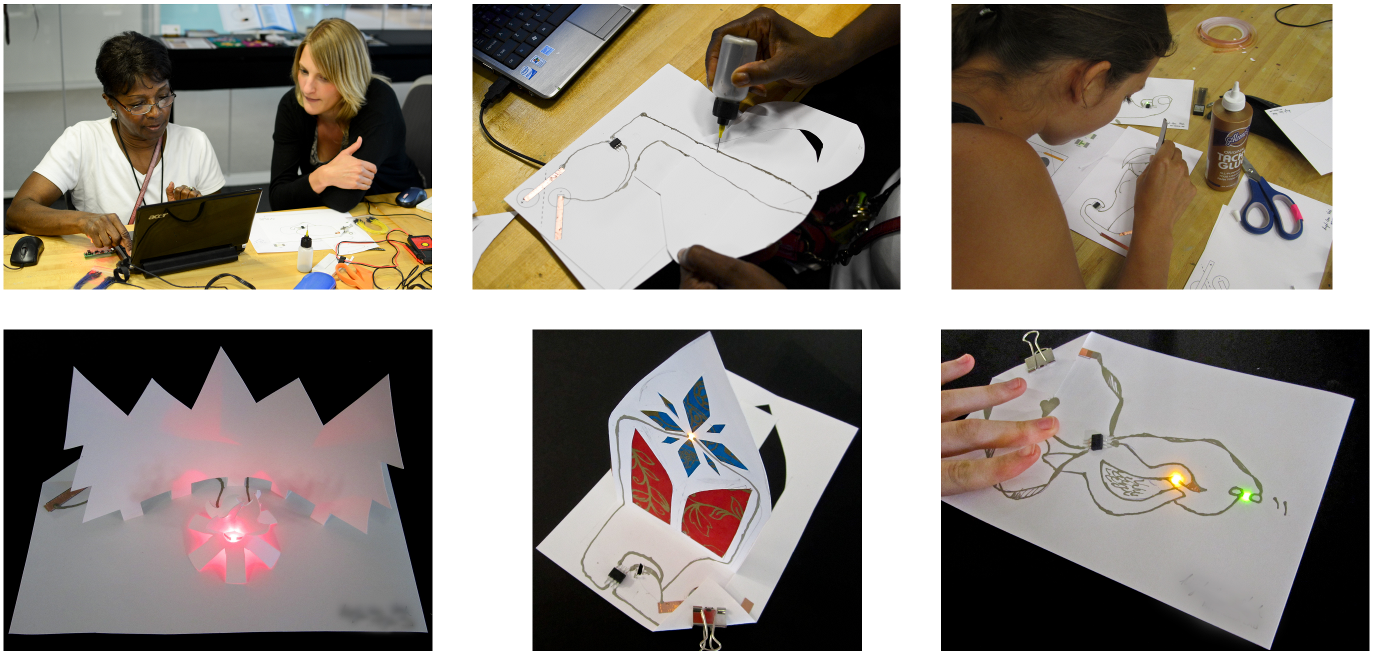

Figure 5: Papercraft and Programming

comfort. One participant spoke about why she was interested: “I think for me, it wasn’t even the papercraft, because I’m not really into papercrafting. But it was more that it seemed accessible. Papercraft is kind of like painting and drawing or writing. It seemed like it was easily transferable. . . It seemed like it wasn’t terribly expensive, it seemed like it wasn’t terribly out of reach for a regular person to do on their own.”

The simplicity and versatility of the components garnered praise: “Great way to introduce [microcontrollers]. Easy to work with. Equipment wasn’t so complicated.” Another said: “It was a lot of fun. Really simple tools you can use to make almost anything, create almost anything. Almost everyone made something completely different from paper, paint, and like two circuit elements.”

Participants discussed the way that the language used to describe technology can influence the groups of people that are interested. One participant speculated that the workshop was specifically geared to appeal to groups typically excluded by the conventional language of electronics and technology design: “I was wondering, if with your title and your description. . . if you were trying to get more women involved or if you expected more women to be drawn to the craft aspect. Cause I generally think of craft as more feminine than more masculine. Whereas if it was microcontrollers, I would think you would get more men than more women.” Though the workshop was not explicitly targeted at any single group, we find it noteworthy that the attendants were overwhelmingly female.

Affordances of a Craft Approach to Technology

Integrating craft processes into the construction and design of technology introduces new skills, goals, and outcomes. One respondent commented: “The whole experience of working through and drawing the lines, you develop a sensitivity to where are they going to do and how are they working. They become part of the artwork or the craft.” Workshop participants who were inexperienced with electronics and programming found comfort and confidence in working with craft materials: “The silver ink was a little challenging to use at first, but I loved the way it looked. . . I loved drawing circuitry.” Another participant excelled in quickly mastering the craft aspects of the project because of prior experience with other forms of art: “I just like drawing, animation, so I thought, make her eye glow.” Another mentioned the challenge of integrating the various aspects: “Circuits and microcontrollers were new and so, actually, was working with paper. It was a great creative challenge at the end to combine them in a way that was functional.”

The nature of the materials also introduced new challenges. The unpredictability of the ink—which can spatter or smear, disrupting an aesthetic design or the conductivity of a trace—demanded participants attention, and rewarded mastery of the materials. Maintaining both a functional circuit and a satisfying aesthetic process was difficult: “Drawing with conductive paint was surprisingly hard. The conductive painting needs to be done with great care.’ For those who were experienced with conventional electronics design, the radically different process introduced new challenges: “I’ve used [microcontrollers] before, it’s always been breadboard to printed circuit board. This is definitely a different medium. I don’t know if it’s more or less frustrating. There are different things, like the ink coming out.” We see these challenges as evidence of a rich medium that provides opportunities for skill development and mastery—ones quite different from those found with traditional circuit design or electronic toolkits.

Differences Between a Toolkit and Microcontrollers as a Material

In reflecting on the artifacts made in the workshop, we see a difference with those constructed from electronics toolkits. As the modules of a toolkit tend to be larger and more expensive than bare electronic components, they are often used to assemble one artifact with one function–and then disassembled to form another artifact with another. In an untoolkit, however, the expense is loaded primarily onto the tools themselves, which provide the means to utilize relatively low-cost materials. Our workshop participants created finished, stand-alone artifacts, ones in which the electronic components and circuit were integrated directly with the paper and other craft materials. The shape and appearance of these artifacts derives predominantly from the these other materials and from the conductive ink itself, unconstrained by the need to incorporate bulky or awkward toolkit modules. The artifacts are rarely taken apart; instead, iteration happens by starting anew, an action more akin to that of an artist, setting aside one sketch and beginning another.

Using an inexpensive microcontroller and inexpensive materials inspired workshop participants to experiment with future projects: "I really liked how we were only using a few components, because if you buy a $25 microcontroller board. . . you feel like it's really precious. You don't want to embed it in a project because you spent almost $30 on it. But just using the ATtiny and a couple of LEDs, that's not intimidating at all. It encourages people to make these kinds of things more frequently. Instead of thinking you have to dedicate your microcontroller board to this really awesome project. You feel like you can experiment, do stuff whenever you feel like it." Cost is a vital factor in the diversity of the artifacts that can be made. "An Arduino is $30, it's precious. I've got to use it well." The small size of the microcontroller and its affordability, thrilled some participants: "The ATTiny can be used in many things! It is very tiny, it can be incorporated into clothing and books." In addition, we think there's value in introducing people to the standard components (like the ATtiny) that are found in the devices they use in their daily lives.

Giving People an Understanding of the Relationship between Programming and Devices

In introducing microcontrollers and programming alongside a set of easily-manipulable craft materials, we hoped to create an understanding of the role that programming plays in the design and functionality of interactive objects. We also hoped to create an experience in which participants felt excited and comfortable about working with technology. On one of those metrics, we were successful: before the workshop, five participants did "not enjoy programming." Asked in a post-survey, all but one said they did.

Linking the crafting of a physical artifact with a conventional programming environment—in this case, the Arduino programming environment—allowed for novices to find traction quickly. One participant said: "For people like me that don't know how to write software, in any shape, in any form. It looks like a bunch of letters on the screen. It helped a lot to be like this goes here, this is an LED, put that there. It helped a lot for someone to point things out, what makes what actually light up and what didn't."

Some workshop participants were excited to be working with code at all: "The programming was what sold it for me. . . when you bring programming to it, it's at a whole other level. . . " Even the limited amount of programming that we shared was enough to revise a participant's overall perception of digital devices: "To me, making a light blink was interesting, and I think about a computer, and it just blows my mind. How many lines of code must that be? And no one, they just take it for granted."

Another participant, who had formal experience programming, noted the different motivations that microcontroller programming introduced: "I've done some programming years ago and it was like, it was such a fun manipulation of it. To make the connection was really fun. How I might make that little bug do what I ask it do."

UNTOOLKIT: DEFINITION AND DESIGN PRINCIPLES

We use the term untoolkit to refer to an attempt to provide scaffolding similar to that of a traditional electronics toolkit while avoiding the creation of custom, high-level building blocks or modules. Instead, an untoolkit approach provides tools and techniques that allow existing components and materials to be leveraged in new ways or by new groups of people. It is these existing parts that are incorporated into the constructed artifacts, not the elements of the untoolkit itself. Because the parts are already being made and sold for other uses, they are likely to be cheaper and more widely available than a custom module enclosing them would be.

In making an untoolkit, we try to identify the resources that would make these existing parts accessible and appealing. In doing so, we apply many of the same principles as do other electronic toolkits: lowering barriers to entry, creating evocative examples, fostering communities of use, establishing a clear identity, etc. Like a toolkit, an untoolkit frames a domain of things you can make and stuff you can do. By using the word "untoolkit", we seek to emphasize the possibility of pursuing these goals without necessarily using the high-level modules found in other electronics toolkits. In this respect, we intend the term as much as a provocation as a new category of tools.

As demonstrated through the paper, our notion of an untoolkit has much in common with traditional notions of a medium, with its materials, tools, and techniques. By drawing on existing parts, untoolkits introduce novices to many of the same components and materials as those used by experts. It provides for subtlety and mastery in crafting artifacts from low-level parts and raw materials. It suggests possibilities for the incorporation of diverse elements, instead of restricting flexibility with proprietary modules or connectors. In these ways, we think it offers an useful complement to the attributes of other electronics toolkits.

Isn't this just another word for a toolkit?

Sort of. In its original meaning, a toolkit refers simply to a set of tools for a particular purpose, a definition close to the one we use for "untoolkit". In electronics, however, the term “toolkit” has become so strongly identified with a set of modular building blocks that we thought it was important to use a different word to suggest an alternative approach.

But aren’t you still creating custom modules (e.g. the TinyProgrammer)?

Yes, but these modules aren’t embedded into the artifacts built with the untoolkit. This makes them more like a traditional tool than like a toolkit’s modular building block.

How is this different from just buying a microcontroller?

Standalone microcontrollers are typically found in an engineering context, requiring technical knowledge and complicated setup. By creating tools to make them easier to program and techniques for embedding them in craft practices and artifacts, we transform the process of working with these existing components.

Isn’t a microcontroller a high-level, modular building block?

It is, but it’s one that’s already ubiquitous – small, cheap, widely available, and self-contained. In that way, it’s similar to paint, paper, or other industrially-produced materials that we treat as fundamental elements in the construction of artifacts.

Our experience with the untoolkit described here yielded principles that may be useful in the design of future untoolkits:

Frame the technology for the target audience. As mentioned above, it is important to suggest the kinds of techniques and artifacts that can be made with the untoolkit. For example, our TinyProgrammer, TinyUploader, and ATtiny plugin for Arduino are all relatively general purpose, applicable to many different kinds of circuits and objects. By pairing them with techniques for constructing circuits from paper and conductive ink, we provided them with a context that attracted and guided our audience. Even the name you give to the untoolkit and where you advertise it can have an important influence on who uses it and for what. All of these issues are crucial to the design and success of an untoolkit.

Leverage existing hardware and software. In designing untoolkits, it’s important to leverage existing hardware and software. Open-source projects are an obvious source for programs or objects that can be customized for inclusion in an untoolkit. Another useful approach is in combining existing materials and components, possibly from a diverse set of suppliers or industries. Mass production yields parts that can be sourced globally at reasonable prices; these are often preferable to custom parts that may be more expensive or less-widely available. It can be worth spending some extra time to find an existing tool or part that does what you need, although custom components may be preferred for maximum ease-of-use and suitability to the target domain or audience.

Provide paths for further exploration and knowledge. Because of their dependance on existing materials and components, untoolkits offer natural pathways for further exploration and knowledge. An explanation of the relationship between the elements of an untoolkit and related tools or techniques can facilitate transition to more standard or professional methods. Even simple things like links to the sources for the elements of the untoolkit or the materials and components it manipulates are invaluable for enabling people to pursue independent exploration and learning. Open-sourcing the designs of an untoolkit’s software and hardware provide another way for people to understand how they work or adapt them to suit their developing interests and activities.

FUTURE RESEARCH

We are interested in exploring the potential of untoolkits and microcontrollers as material in a number of different ways.

Untoolkits for Other Domains

While craft provides a particularly rich domain for untoolkits, we’re also interested in exploring the kinds of untoolkits that would work with more traditional circuit construction techniques. In these contexts, it is crucial to provide the untoolkit with a strong focus and conceptual identity in order to suggest the kinds of things that people can make with it. For example, an untoolkit for audio projects might include a hardware probe that makes it easy to listen to the audio signal at any point in the circuit as well as a software design tool for simulating the behavior of filters and other effects. An untoolkit for working with sensors might include a module for recording and replaying sensor data and a software tool for processing the data.

Digital Fabrication as Untoolkit

Digital fabrication provides numerous possibilities for software-focused untoolkits: digital design tools offering high-level, easy-to-use interfaces that can be translated into physical objects. Because of the flexibility of the fabrication processes, the final object can be custom-made from standard materials and components even while the digital design is assembled from higher-level virtual building blocks. Transferring modular abstractions from hardware to software offers a number of advantages: the final product can avoid redundant enclosures and connectors, designs can be more easily indexed and shared, software can generate complex or intricate forms. While tools for digital fabrication abound, we think an untoolkit approach suggests new possibilities for interfaces that focus on particular classes of objects.

From a microcontroller to understanding a device.

While the workshop provided participants with an understanding of the relationship between microcontroller code and its effect on the physical world, it seemed less successful in providing them with an understanding of the overall functioning of other electronic devices. For example, one participant said of understanding their own devices, “Short of taking apart devices that I need and would probably break, and then no longer have. . . it’s like. . . where do you go to figure that out?” We would like to explore combinations of microcontroller programming with device hacking to see if they could provide people with a more comprehensive understanding of the functioning of electronic devices.

Artistic practice and mastery.

While the focus of the workshop described here was on providing participants with an accessible introduction to microcontrollers, we think the same basic techniques can provide for sophisticated artistic practice and aesthetic exploration. We're interested in exploring the potential for mastery and expression in crafted circuits, both in our own projects and through working with others. We're also interested in exploring the range of artifacts that can be produced using these circuit construction techniques and the diversity of functionality that can be integrated into them.

CONCLUSION

In this paper, we've presented our techniques for constructing interactive artifacts with conductive ink, paper, off-the-shelf electronic components, and our hardware and software tools for microcontroller programming. These techniques demonstrate a tight integration of craft and technology, generating different artifacts and engaging different skills and people than traditional embedded development or electronic toolkits. The accessibility of papercraft offers an appealing entry into electronics and programming. Our tools for programming microcontrollers suggested the idea of an un-toolkit, which offers an alternative to traditional approaches for scaffolding the construction of interactive devices. This combination of tools and techniques allows us to treat microcontrollers as a material, merging the domains of technology and craft in new and powerful ways.

REFERENCES

1. P. Blikstein and A. Sipitakiat. Qwerty and the art of designing microcontrollers for children. In *IDC '11*, pages 234–237. ACM, 2011.
2. L. Buechley and M. Eisenberg. Fabric pcbs, electronic sequins, and socket buttons: techniques for e-textile craft. *Personal Ubiquitous Comput.*, 13(2):133–150, Feb. 2009.
3. L. Buechley, M. Eisenberg, J. Catchen, and A. Crockett. The lilypad arduino: using computational textiles to investigate engagement, aesthetics, and diversity in computer science education. In *CHI '08*, pages 423–432. ACM, 2008.
4. L. Buechley, S. Hendrix, and M. Eisenberg. Paints, paper, and programs: first steps toward the computational sketchbook. In *TEI '09*, pages 9–12. ACM, 2009.
5. L. Buechley and H. Perner-Wilson. Crafting technology: Reimagining the processes, materials, and cultures of electronics. *ACM Trans. Comput.-Hum. Interact.*, 2012.
6. M. Coelho, L. Hall, J. Berzowska, and P. Maes. Pulp-based computing: a framework for building computers out of paper. In *CHI EA '09*, pages 3527–3528. ACM, 2009.
7. Q. R. Ghazala. The folk music of chance electronics: Circuit-bending the modern coconut. *Leonardo Music Journal*, pages 96–104, 2004.
8. S. Greenberg and C. Fitchett. Phidgets: easy development of physical interfaces through physical widgets. In *UIST '01*, pages 209–218. ACM, 2001.
9. B. Hartmann, S. R. Klemmer, M. Bernstein, L. Abdulla, B. Burr, A. Robinson-Mosher, and J. Gee. Reflective physical prototyping through integrated design, test, and analysis. In *UIST '06*, pages 299–308. ACM, 2006.
10. S. E. Hudson and J. Mankoff. Rapid construction of functioning physical interfaces from cardboard, thumbtacks, tin foil and masking tape. In *UIST '06*, pages 289–298. ACM, 2006.
11. S. Johnson and A. P. Thomas. Squishy circuits: a tangible medium for electronics education. In *CHI EA '10*, pages 4099–4104. ACM, 2010.
12. E.-S. Katterfeldt, N. Dittert, and H. Schelhowe. Eduwear: smart textiles as ways of relating computing technology to everyday life. In *IDC '09*, pages 9–17. ACM, 2009.
13. J. C. Lee, D. Avrahami, S. E. Hudson, J. Forlizzi, P. H. Dietz, and D. Leigh. The calder toolkit: wired and wireless components for rapidly prototyping interactive devices. In *DIS '04*, pages 167–175. ACM, 2004.
14. K. Moriwaki and J. Brucker-Cohen. Lessons from the scrapyard: creative uses of found materials within a workshop setting. *AI Soc.*, 20(4):506–525, Aug. 2006.
15. H. Perner-Wilson. A kit-of-no-parts. Master's thesis, MIT, 2011.
16. H. Perner-Wilson, L. Buechley, and M. Satomi. Handcrafting textile interfaces from a kit-of-no-parts. In *TEI '11*, pages 61–68. ACM, 2011.
17. J. Qi and L. Buechley. Electronic popables: exploring paper-based computing through an interactive pop-up book. In *TEI '10*, pages 121–128. ACM, 2010.
18. J. Qi and L. Buechley. Animating paper using shape memory alloys. In *CHI '12*, pages 749–752. ACM, 2012.
19. M. Resnick. Behavior construction kits. *Commun. ACM*, 36(7):64–71, July 1993.
20. M. Resnick, F. Martin, R. Berg, R. Borovoy, V. Colella, K. Kramer, and B. Silverman. Digital manipulatives: new toys to think with. In *CHI '98*, pages 281–287. ACM Press/Addison-Wesley Publishing Co., 1998.
21. A. Russo, B. Y. Ahn, J. J. Adams, E. B. Duoss, J. T. Bernhard, and J. A. Lewis. Pen-on-paper flexible electronics. *Advanced Materials*, 23(30):3426–3430, 2011.
22. G. Saul, C. Xu, and M. D. Gross. Interactive paper devices: end-user design & fabrication. In *TEI '10*, pages 205–212. ACM, 2010.
23. N. Villar, J. Scott, S. Hodges, K. Hammil, and C. Miller. .net gadgeteer: A platform for custom devices. In *Pervasive '12*, 2012.