Designing for the Internet of Things: Prototyping Material Interactions

Tom Jenkins
Georgia Institute of Technology
Digital Media Program
Atlanta, GA, 30332, USA
tom.jenkins@gatech.edu

Ian Bogost
Georgia Institute of Technology
Digital Media Program
Atlanta, GA, 30332, USA
ian.bogost@lmc.gatech.edu

CHI 2014, April 26–May 1, 2014, Toronto, Ontario, Canada.

ACM 978-1-4503-2474-8/14/04..$15.00.
http://dx.doi.org/10.1145/2559206.2578879

Abstract
The Internet of Things (IoT) offers fertile ground to consider the nature of electronic prototyping, especially in building systems from the lowest level. While constructing artifacts to interact directly with everyday materials and contexts, we've found it important to approach the IoT from the very lowest levels of hardware to avoid both abstracting away from real knowledge of the platform itself as well as to reduce implementation cost for massive deployment.

Building new, inexpensive platforms that augment everyday objects in minimal ways is our proposal for an alternative to top-down control of IoT devices. We intend to move towards interactions among and between things as a bottom-up design study into ubiquitous small-scale computing and its potential aesthetic applications.

Author Keywords
Prototyping; making; materiality; Internet of Things; IoT; design

ACM Classification Keywords
H.5.2 [Information Interfaces and Presentation]: User Interfaces—prototyping; C.5.3 [Computer System Implementation]: Microcomputers—microprocessors; J.5 [Art and Humanities]—design

Introduction

"The Internet of Things" describes a trend advocating that all sorts of physical artifacts become connected to and controllable from the Internet. According to the Internet of Things (IoT) vision, a coffeepot might be controlled alongside a thermostat to have a home warm and the coffee on when a person wakes up; or sensors in the basement might email you if your basement is flooding.

In some ways, an Internet of Things has been with us for a long time. In addition to end-user devices and products, there are things like algorithmic traders—programs that make thousands of stock transactions per second to obtain a market edge by any means possible. These are things on the Internet, but they are used solely for pragmatic purposes and for financial gain. Likewise, massive and often overlooked examples of the IoT take place in shipment tracking, with pallets of merchandise being tracked by RFID tag from one side of the Earth to the other. On the other end of the pragmatic/aesthetic spectrum, we have the beginnings of an Internet of Things where the viewpoint of the computational object itself is celebrated. Blogs like *The New Aesthetic* [13] emphasize a computational perspective on everyday life. In enacting visions of a connected future, research into early IoT systems may be better placed on the aesthetic end of the algorithmic spectrum, rather than the pragmatic.

From this aesthetic perspective, IoT offers an opportunity to investigate the interrelation of Internet access, materials, and everyday experience, emphasizing particular values through design. Building a new hardware system that devise different types of connections to materiality critically examines the role of objects in the everyday, possibly creating new categories of interaction with and between them.

Alongside the rise of the IoT, access to electronic fabrication tools has been expanding significantly. The Arduino prototyping board [8,9] is used in educational contexts, especially in media arts, as well as in the home workshops of enthusiasts. . These prototyping systems have enabled the creation of custom-built IoT systems by relative amateurs and novices, and have served to help democratize the production of electronic artifacts [11].

These enthusiast technologies offer methods of creating an Internet of things that interacts with everyday materials in novel ways. For example, Botanicalls [14] instruments house plants with a microcontroller, a moisture sensor, and an Internet connection. Depending on the version of the hardware, when the plant needs to be watered, it will either place a phone call or send a message through Twitter to its owner asking to come and give it a drink. The Tweet-a-watt [6] changes the dynamic of another, similar everyday system. It is a wattage-measuring tool that connects to the Internet that reports the current power usage of an electrical outlet over Twitter. Finally, Supermechanical's Twine—born of a successful Kickstarter project that raised over $100,000—promises its backers access to the Internet of Things that "doesn't require a nerd degree" [15]. This enthusiast-oriented device offers an application programming interface (API) to easily create rules for various kinds of events. One example listed on their project page describes using the twine to alert contacts that if the temperature is over a particular temperature, then the owner is at the lake.

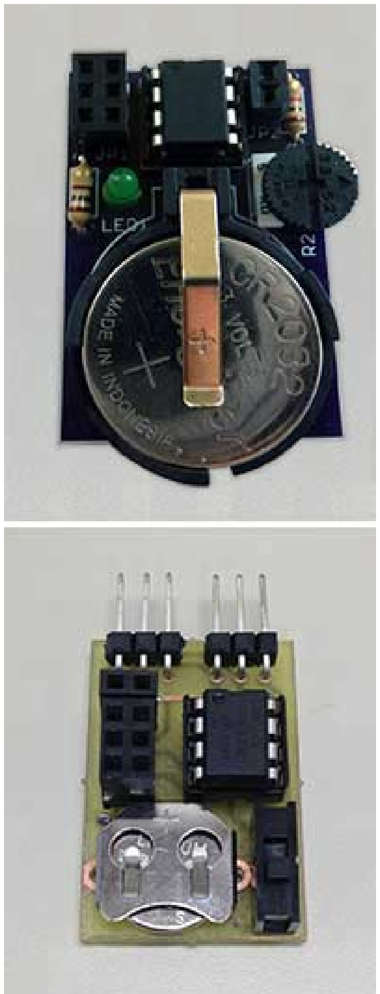

Figure 1: The original version of the tiny tinkering platform, top; and the most recent revision, bottom.

These systems emphasize the creative, aesthetic, and personal potential of the Internet of Things as imagined by a larger, interested community of tinkerers that invest into new uses of technology and shares the applications created among them. Building an open ecosystem for a prototyping platform makes the system useful and ubiquitous. Sensed data can be sent to the web by activating arbitrary triggers, forwarding information to new objects or places. But what happens when the Internet of Things loses its connection?

The Internet of Things Without the Internet

Drawing inspiration from the in-home uses of the examples above, our prototypes begin to sketch out an alternative vision of the Internet of Things. This IoT is bottom-up rather than centrally controlled; it puts an emphasis on material/computational/human collaboration rather than parameterization of the everyday world for a powerful central authority, and most notably, does not require the Internet. Our proposal is to construct small-scale, human-evaluated interactions; or more complicated, communication-driven emergent interactions that still needs humans for assessment. In these concepts, the system's output stays *in situ*, and people around them draw any conclusions themselves.

A Tiny Tinkering Platform

While we are inspired by them, we feel that popular hardware prototyping platforms like the Arduino [9] and Raspberry Pi [16] are often used in aspirational ways, rather than practical ones. We claim that their practitioners orient themselves primarily toward membership into a community of "hackers" or "makers" and only secondarily toward creating working, useful systems. These tools create and reinforce their identity as a "maker," a type of person who is empowered to create with electronic objects, but ends up primarily making novelties or toys.

In contrast, we are creating "tiny tinkering" platforms as an experimental system that explore material interactions in the home. From our perspective, gratification comes not from making something new and remarkable to show off to the world, but in making something unremarkable that still feels important to an individual or small group. Possible applications include when to water a particular plant, when hot coffee has reached the right temperature, when the mail has been delivered, whether it's cold enough outside to warrant wearing a jacket, or a trigger to automatically lower the blinds when the sun is hitting a room in a certain place.

Our aim is to create a speculative ecosystem that reduces the complexity in improvising solutions for small-scale, ubiquitous problems like these. It allows novices to create applications quickly, simply, and inexpensively. To that point, today's electronics platforms are priced for enthusiasts rather than ordinary people. As a result, our platform needs to be as inexpensive as possible in order to encourage applications that are never meant to be precious. Systems should be constructed and left in place doing their job instead of needing to be disassembled and repurposed in a new project down the line. Users should be focused on developing electronics that have a focus on getting something done rather than on doing it.

Initial versions of our tiny low-cost tinkering platform support different types of input and output and are described below. We've focused on the cheapest, most minimal hardware components to force ourselves into a back-to-basics, “hardware store” mentality, where the microcontroller, electronics, and software become integral part of an elemental tool. Our research system is built from very basic hardware, making it very inexpensive, especially at scale. We’re not aspiring to create a platform that could ever be commercially viable, but rather to make something that will let us feel what it is like to create simple solutions to problems like the ones above, and to help others get that feeling as well.

Through our development, we’ve emphasized cost at every step to try to produce a system that could be considered truly ubiquitous. Our ultimate goal is a $1 computer that can be deployed permanently in any capacity to add basic sensor input/output and processing onto any object, system, or circumstance.

Finally, these boards are very close to the bare metal. Abstraction away from the lowest level of computing has been a huge part of the educational mindset behind open-source environments like Processing [17] or Arduino, each wrapping their languages (Java and C, respectively) into simplified frames for consumption. While abstraction has massive value in making very complex systems understandable, in our platform we eschew it, seeking to understand these devices at the lowest level. We do this for two reasons. One, the microcontrollers we’ve experimented with are very simple. In exploring the aesthetic value of minimalist IoT platforms, the very basic parts on our radar do not lend themselves well to complex abstractions away from manual control of individual bits in hardware registries. Due to space limitations and curiosity about what is possible, we find ourselves using assembly and C to program the boards.

Prototyping as Research

The goals for our research system outlined above are lofty, and certainly are not yet fully implemented. Instead, we’ve approached this space considering prototyping new tinkering platforms as a means of testing out different kinds of material constraints and concordant means of processing information.

Material decisions made in the prototyping process have larger implications in the final product. In building out this research platform, the iterative process is essential in revealing the nature of the system in itself: without building a speculative system and encountering drawbacks with it, it is impossible to predict how a system might work in the wild. Running into roadblocks on the fly throws into relief particular constraints and issues. In many cases, the original vision adapts to suit technical validity, or technical implementation shifts to fit a more coherent use-case.

Iterative refinement helps to both construct basic, minimalist examples of projects that work conceptually, while allowing space for revisions that more fully reflect original concepts, adding features with every iteration until a project becomes complete—or something entirely different. Our process of iterative prototyping can help to reveal some of the assumptions we've made about the role of the technological in setting and solving design problems during development. As a lens to this process, our early-stage designs and conceptual use cases are outlined below.

Noise Illuminator

For residents of urban spaces, noise is one of the paramount concerns impacting quality of life. Even something as ubiquitous as traffic can be as loud as an alarm clock. The Noise Illuminator provides visual feedback on noise pollution in a community. The system monitors noise in its surrounding. It signals community members when volume surpasses a set level by lighting LEDs inside a cube that fades over time.

In the future, devices like this could be used to signal noise to IR-sensitive cameras: increasingly-pervasive urban surveillance systems don't take into account aspects of lived experience like noise, focusing instead on the kind of information that can most easily be gleaned from visual imagery. IR LEDs are invisible to people, but can be detected by cameras, making visible to surveillance environmental factors that are often overlooked.

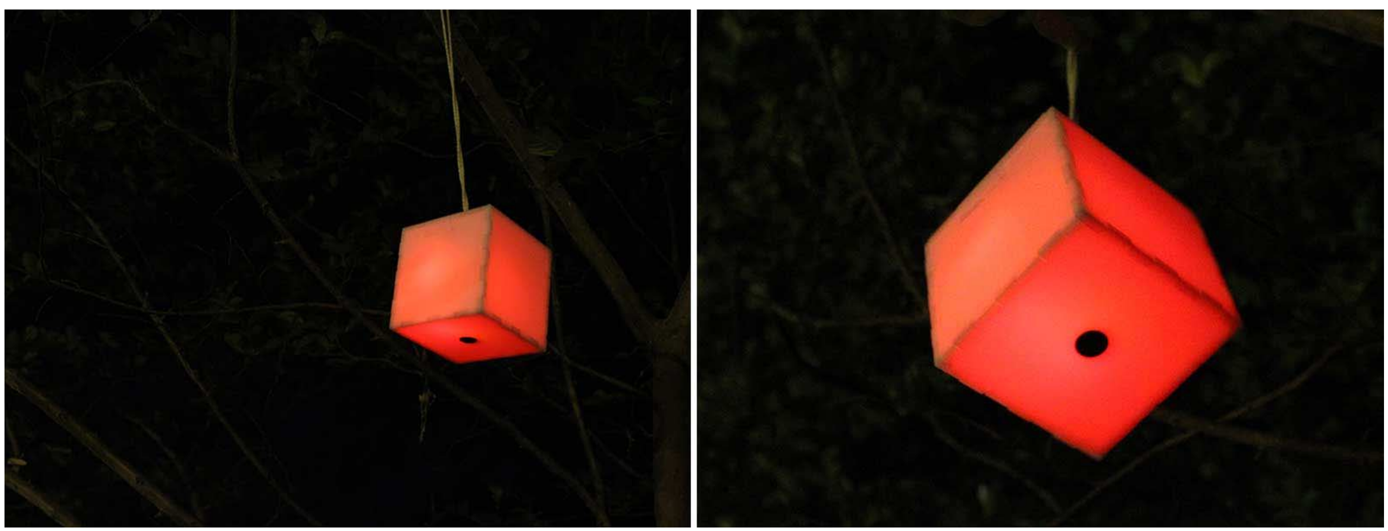

Figure 2. Noise illuminator prototype.

Figure 3 Ripeness detector. The detail at right shows the LED that lights up when the fruit is ripe.

Ripeness Detector
Concrete Jungle [18] is a foraging group that collects fallen fruit from untended trees around Atlanta and donates them to homeless shelters and food banks. As a small nonprofit organization, they have limited manpower to devote to the task. They rely on volunteers to pick and distribute fruit, and interested observers to track of the ripeness of a city's worth of fruit trees.

Casual foraging will never be automated at scale, so no existing electronic solution is feasible. We're applying our tiny tinkering computer to detect ripening fruit and visually signal to passers by that the fruit on a tree is ready to be collected. Our prototype takes a combination humidity and temperature sensor and places it in a Tyvek bag to monitor the progress of fruit as it grows.

Alongside developing and refining these concepts for prototypes, we've been revising the prototyping board as we go—adding different microcontrollers, and trying new sensors— and seeing what kinds of new sensing opportunities arise.

One-bit problems
The first iteration of our prototyping platform used the PIC10F200 as its microcontroller. The 10F series of chips are the baseline parts from Microchip's line. They have only 6 wired pins, even in the 8-pin dual-inline package version, and have very little capability compared to other, later families. When building a device that explores some of the aesthetic qualities of the Internet of Things, that simplicity becomes an advantage: it allows the entire system to become fathomable, something whose material, memory, and state can "fit" into the mind of the developer.

	10F200	10F220	12LF1552
ROM	384	384	3854
I/O	4	4	6
Pins	6	6	8
ADC	0	2	4
SPI	No	No	Yes
Cost	$0.38	$0.46	$0.64

Table 1. PIC microcontrollers that boards have been designed around as part of the prototyping process. "ROM" is the program memory in bytes, "I/O" is input and output pins available on the chip, "Pins" is the total active number of pins on the chip, "ADC" is an analog-to-digital converter, and "SPI" is a serial programming interface, allowing a microcontroller to use a serial communication with sensors or other objects. Cost is based on price per 1000 units.

The 10F200's input has no analog to digital converter (ADC), meaning that the input pins need to be "tuned" to trip analog sensors. This means that rather than having access to the full range of values that the world can present, the microcontroller acts as if the world is binary, like a switch: light sensors are either light enough or dark enough; sound is either present or absent. As far as the system is concerned, a color sensor sees things in literal black and white.

This creates a situation where the user is obligated to find simple problems that can be converted into one bit of information. In practice, this leads to prototypes where configuration and placement become very important. The prototypes above are excellent examples of the kinds of design that one-bit problems lend themselves to. The Noise Illuminator needs to be set to react to a certain, contextual meaning of "loudness" for it to light up. This conception of enough loudness is very different depending on whether it is hung on street signs near a busy intersection or on a tree branch in a public park. Similarly, the Ripeness Indicator needs to be adjusted specifically for particular kinds of fruit. In practice, it would be easier to use it on one or two fruits on a tree to get a sense of when the entire tree is hitting ripeness. In this case,a single augmented fruit becomes a bellwether that allows humans some insight into the state of the fruit on the tree as a whole.

Seeing shades of gray

The second revision of the prototyping boards is based around the PIC10F220. These microcontrollers, like the 10F200, are very basic baseline parts. Unlike that microcotnroller (as can be seen in Table 1), the 10F220 microcontroller includes two ADCs, meaning that a prototyping board can have two analog inputs.

Figure 4. A plant-minding prototype that uses the PIC10F220.

In practical applications, this has led to the addition of shades of gray to the problem space. For example, one prototype (shown in Figure 4, above , and inspired by the Botanicalls project described in the introduction) would intermittently sample soil moisture. Above a certain amount of moisture, the system doesn't do anything—the plant's needs are met. Below a certain level, the device would slowly flash yellow, indicating that the plant is beginning to encounter problems. Even below that level of moisture—and all the way down to none at all—the system would slowly flash a red LED, letting the plant's human caretaker know of its dire predicament. In providing a visual indicator that the plant needs to be watered, our main goal was not to annoy the user. Slow, intermittent sensing allows us to not only use less battery power in registering the current moisture levels, but also provides us with a slow, unobtrusive blink that is useful for conveying information that is less than mission-critical. The analog inputs from the 10F220 allows a user to create different states for different measurements, allowing various degrees of urgency to be represented in the prototype.

Spreading the word
Finally, the PIC12LF1552 is a newer chip that, while still quite basic and inexpensive, offers communications protocols that can talk to different microcontrollers and sensors. The Serial Peripheral Interface (SPI) on the 12LF1552 lets us broaden our imagination to include material augmentations that have messages for us, but also for each other. Devices that collude, flock, or ostracize become possible and interesting. More prosaically, SPI connections to parts such as inexpensive radios that create mesh networks can further refine some of our initial concepts: street noise could be illuminated across distances, appearing in waves or ripples away from their origin; trees wanting to broadcast that their fruit is becoming ripe could sent their messages throughout the orchard, alerting foragers to the exact spot from long distances away.

Discussion

In many ways, the intentions behind this type of technological ecosystem are reminiscent of reflective design strategies [2,10]. In creating a set of tools for material experimentation with instrumenting everyday objects, we hope to create experiences that foster new, exciting interactions with distributed technologies while subverting expectations that the Internet of things exists primarily to be informative, entertaining, or to take some cognitive load off of human users. Similarly, there is an affinity with the strategies and products of critical design [1,3,4,5,7] in that we imagine that these devices aren't solely here to be effective and useful, but that these systems might have their own informational desires and material needs. Our Tiny Tinkering Platform is being built to take a rhetorical stand that asserts a different vision about what everyday interactions with technology can be. Our alternative comes from the bare minimum to see what is possible at the lowest possible level of technical complexity.

The lower cost that comes with using very simple hardware is an added benefit. When using small, autonomous sensors that don't need to connect to the Internet themselves, it becomes much more feasible to create many different kinds of smart things, with little regard to whether adding a microcontroller to something makes sense in the short term. Once our platform becomes more fully developed, it will be possible to build infrastructure-less sensing devices that are cheap, ubiquitous, computational material objects—from coffee lids that tell you when your drink is cool enough to consume, to postboxes that make it clear the letter carrier has been by. At that point, we will find ourselves in a situation where the experiences that everyday objects have are robust, but are only partially translated into the language of the observer. While the functionality is much more minimal than the tabs and pads he described, in some ways the simplicity of the Tiny Tinkering Platform preserves the original promise of Mark Weiser's vision for ubiquitous computing [12]. Rather than emphasizing screen-based interactions with everyday objects, giving directions or setting rules from a cellular phone or personal computer, our small computational artifacts instrument everyday materials, letting them become partners in ubiquitous meaning-making.

Acknowledgements
Thanks to Carl DiSalvo and Sarah Fox, and Thomas Lodato for their comments during prototyping and writing, as well as to the reviewers for their suggestions during the alt.chi jurying. This work was supported by the Intel Science and Technology Center for Social Computing (ISTC-Social).

References
[1] Bardzell, J. and Bardzell, S. What is "critical" about critical design? *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2013), 3297–3306.
[2] Dourish, P., Finlay, J., Sengers, P., and Wright, P. Reflective HCI: towards a critical technical practice. *CHI '04 extended abstracts on Human factors in computing systems*, ACM (2004), 1727–1728.
[3] Dunne, A. and Raby, F. *Design Noir: The Secret Life of Electronic Objects*. Birkhäuser, Berlin, 2001.
[4] Dunne, A. and Raby, F. The Placebo project. *Proceedings of the 4th conference on Designing interactive systems: processes, practices, methods, and techniques*, ACM (2002), 9–12.
[5] Dunne, A. *Hertzian Tales: Electronic Products, Aesthetic Experience, and Critical Design*. The MIT Press, 2006.
[6] Fried, L. Tweet-a-watt. http://www.ladyada.net/make/tweetawatt/.
[7] Gaver, W.W., Beaver, J., and Benford, S. Ambiguity as a resource for design. *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM (2003), 233–240
[8] Gibb, A. New Media Art, Design, and the Arduino Microcontroller: A Malleable Tool. 2010. http://c298394.r94.cf1.rackcdn.com/New%20Media%20Art,%20Design,%20and%20the%20Arduino%20Microcontroller.pdf.
[9] Mellis, D., Banzi, M., Cuartielles, D., and Igoe, T. Arduino: An Open Electronics Prototyping Platform. *Proceedings of the Conference on Human Factors in Computing (alt.chi)*, ACM (2007).
[10] Sengers, P., Boehner, K., David, S., and Kaye, J. "Jofish." Reflective design. *Proceedings of the 4th decennial conference on Critical computing: between sense and sensibility*, ACM (2005), 49–58.
[11] Tanenbaum, J.G., Williams, A.M., Desjardins, A., and Tanenbaum, K. Democratizing Technology: Pleasure, Utility and Expressiveness in DIY and Maker Practice. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2013), 2603–2612.
[12] Mark Weiser, "The Computer for the Twenty-First Century," *Scientific American,* pp. 94-10, September 1991
[13] The New Aesthetic. http://new-aesthetic.tumblr.com/.
[14] Botanicalls. http://www.botanicalls.com/.
[15] Supermechanical : Twine - Listen to your world, talk to the Internet. http://supermechanical.com/.
[16] About us | Raspberry Pi. http://www.raspberrypi.org/about.
[17] Processing.org. http://processing.org/.
[18] Concrete Jungle. http://www.concrete-jungle.org.