Build IT: The Internet of Things

Today's small networked devices increasingly are based on 32-bit microcomputers, which can represent a daunting level of hardware complexity. The use of an intuitive tool to customize such designs at a high level of abstraction can possibly speed the adaption of ever more intelligent devices into the Internet of Things.

by W. Gordon Kruberg, M.D. and Andrew Simpson, PhD, Gumstix

Surrounding us are hundreds of devices that, once connected, will know a lot about our everyday lives. Think of a light switch in the foyer of a home. An innocuous piece of equipment with a very simple purpose; a light switch reveals much about the homeowner: what time they leave for work in the morning, what time they get home in the evening, and what time they turn in for the night, as examples. If all the light switches in that house could be networked together and connected to a smart enough computer, all that data would give a reasonably clear picture of the homeowner's life as they move from room to room.

That same data not only illustrates the occupants' lives, but could also be used to make their house smarter. In fact, this is exactly what the Nest Learning Thermostat does for climate control. While programmable thermostats have certainly existed for a long time, they are usually complicated to program and difficult to adjust.

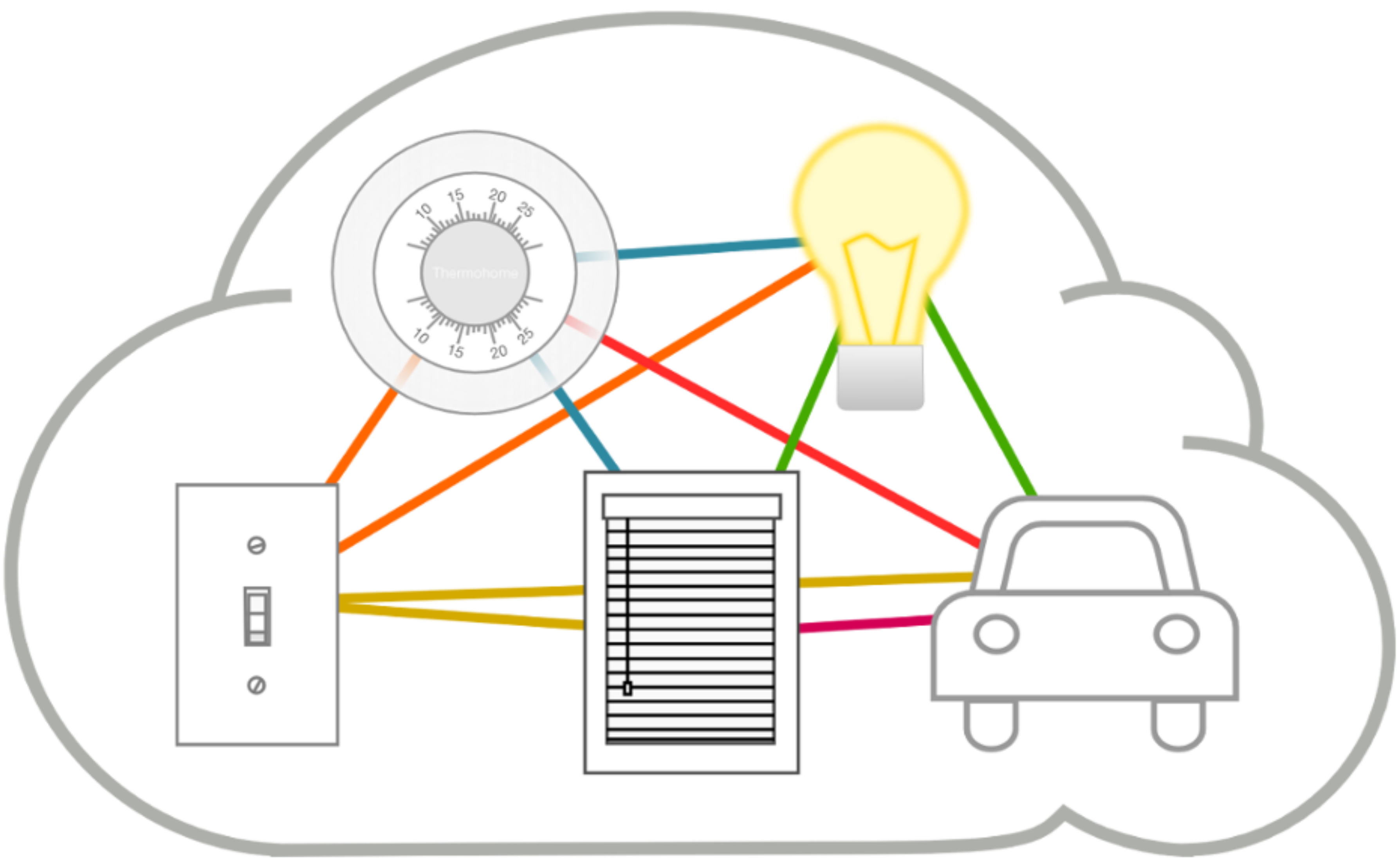

FIGURE 1

A conceptualization of the Internet of Things, where many everyday devices talk to each other.

The Nest is a thermostat that can "program itself" by learning from the user, who simply turns a dial to adjust the temperature. The Nest remembers what temperatures were set at what time. It can be controlled from the Internet and can also detect when no one is home for automatic shutoff, leading to cost savings and reduced energy use.

The Internet of Things

The Nest is a device that is one node in the "Internet of Things." The Internet of

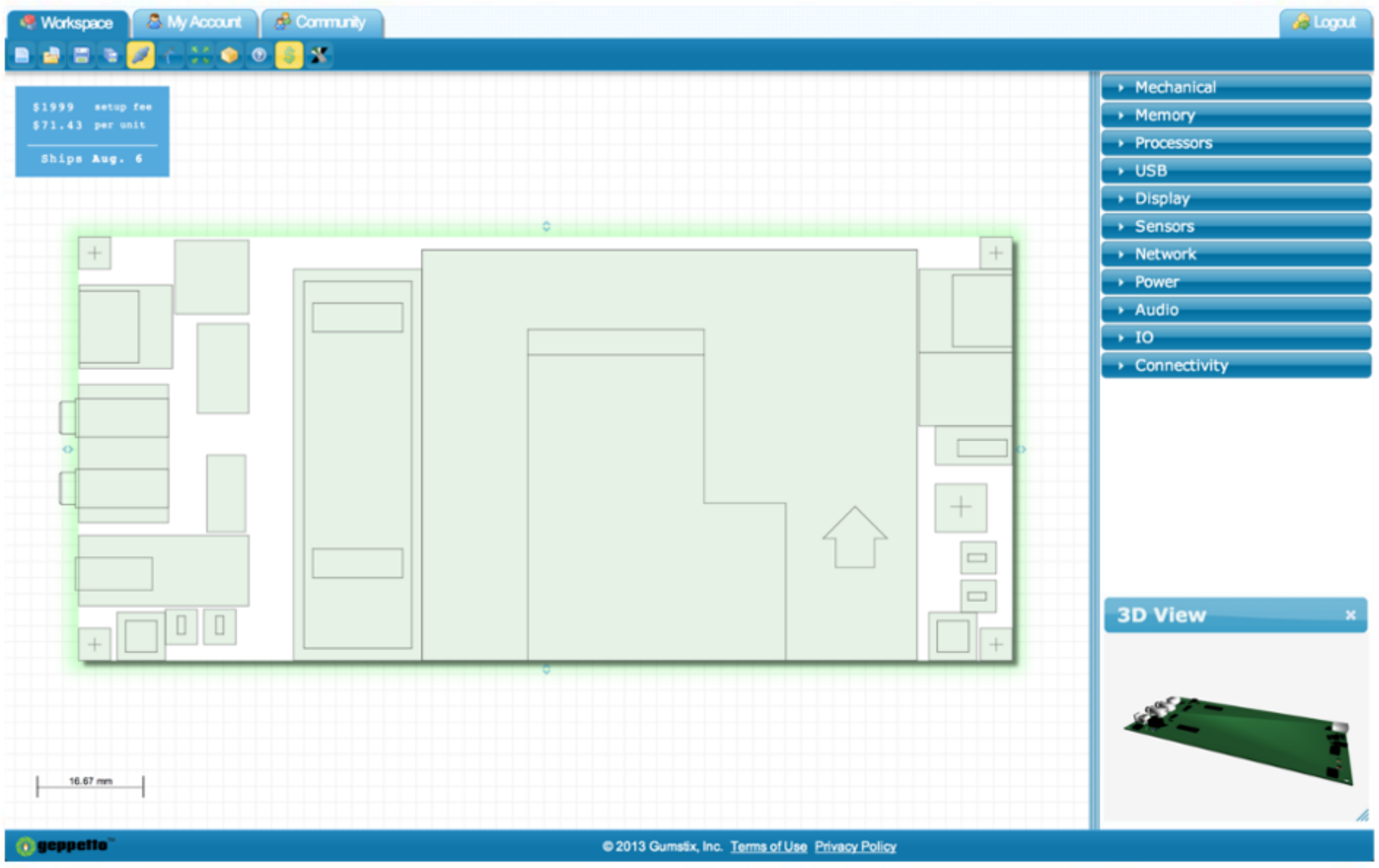

FIGURE 2

The Geppetto design platform user interface showing a design. The green blocks represent modules that contain specific hardware functionality.

Things means different things to different people, but it originally referred to ubiquitous RFID tags used to track items within a supply chain. Today the things in the Internet of Things are not limited to passive technology being incorporated into real-world objects—like RFID tags on supply chain components—but are now being used in smaller and smarter devices with processors and networking directly onboard (Figure 1). A teardown of the Nest thermostat, for example, revealed support for Wi-Fi, ZigBee and an ARM Cortex-A8 processor.

The addition of high-level computing capabilities to "dumb" objects in our physical environment makes the potential of the Internet of Things even greater than before. Bill Wasik's article in *Wired* magazine earlier this year welcomes us to the "Programmable World" by giving us a tour of SmartThings CEO Alex Hawkinson's smart house, enabled by devices that can be controlled over the web. Devices like those in Hawkinson's smart house comprise the next generation of the Internet of things, where tiny, full-fledged computers collect data, analyze it and send it to other such devices for the process to repeat, when necessary. For example, your phone could let your house know when you're on you're way home from work; your house would then know the exact moment to turn on the lights, turn the thermostat up, even unlock the door, to maximize not only convenience, but things like energy savings and enhanced security.

Embedding Intelligence: Software Is King

Small, ubiquitous computers in our everyday physical world are nothing new. For years, embedded computers have been used in many things from home appliances like washing machines and microwave ovens, to our cars and on buses, to building security systems and innumerable other applications. These systems, however, have always relied on hardware tailor-made to the specific application and oftentimes do not include even a microprocessor, opting for cheaper, easier to work with, but ultimately less-capable, microcontrollers. With such a limited scope for their hardware, these "things" have historically never been able to "talk" to each other, even when that might have been useful.

In recent years, however, embedded computers have become much more powerful. Platforms based on ARM's architecture have come to dominate this space, offering incredible performance with very low power requirements. Embedded computers also frequently include wireless networking capabilities, whether Bluetooth or Wi-Fi, allowing them to communicate with each other using existing networking infrastructure. More capable embedded computers also allow developers to deploy increasingly sophisticated software solutions on full-fledged operating systems like Linux. The ability to use high-level systems like Linux offers greater flexibility for developers to create a more intelligent, more capable and more connected solution than otherwise possible in hardware alone.

While the focus has shifted away from expensive hardware development to more flexible software development, the need to customize hardware for specific functionality remains. With hardware design out of the grasp of many software developers—and vice versa—innovation on devices that could become part of the Internet of Things is currently constrained by the considerable resources needed for both hardware and software development.

Intuitive Embedded Design

Embedded design has always been a complex field. Embedded engineers are required to make numerous decisions with long-ranging consequences when specifying a device—for example, ARM or Intel? Low-power or performance? Answering questions like these culminates in a series of trade-offs, not only in terms of specifications, but also in terms of cost. Developing a solution using a microcontroller, for example, may be cheaper and easier than using a microprocessor when possible, but microcontrollers also offer no room for further expansion should the need arise. However, while microprocessors are the more capable choice, they are significantly more complex and require more expertise.

For these reasons, laying out the specifications for an embedded device ultimately locks the engineer into a long-term manufacturing path that is difficult and expensive to change later on. Development kits exist to mitigate some of these factors, allowing engineers to begin developing with existing, generic hardware that meets their needs, but suffer from the problem of being equally complicated to customize once the engineer is ready to do so. With the introduction of intuitive embedded design tools, however, pre-built development kits designed with a tool can also be easily customized using high-level specifications, and the manufacturer can provide such a board at a fraction of the cost otherwise.

Intuitive embedded design tools are a new way of designing electronic devices. With an interface that abstracts hardware design to a higher level than electrical components, intuitive embedded design tools let users without extensive electrical design knowledge create hardware solutions on which comprehensive software solutions can be deployed. By removing the hardware barrier and providing an easy way for anyone to design electronic devices, the new Internet of Things—one that relies on sophisticated computing power coupled with comprehensive software solutions—is being made possible.

Gumstix' Geppetto is one intuitive embedded design platform that launched earlier this year. In Geppetto, users simply drag, drop and connect modules on a board and then order it at the touch of a button. Geppetto-built boards are either small single board computers (SBC) with the processor built in, or a small expansion board that can be used with Gumstix' Overo and DuoVero computers-on-module. The resulting device made up of an SBC or COM with expansion board runs a full Linux operating system on ARM Cortex-A8 processors, while Wi-Fi, Bluetooth and DSP are just some of the optional features that are included. Geppetto-created designs can also be shared with the Geppetto community, allowing users to build their own features into solutions that already exist.

Online Design Tools

Geppetto is not the only online embedded design tool. Circuits.io, Upverter and HackEDA are just three examples of tools that users can use to design, share and order electronic devices with relative ease. All three take a more traditional electronic design application (EDA) approach, where users connect circuit blocks in a schematic to get the functionality they need. Geppetto, however, approaches embedded design conceptually at a high level. Instead of connecting circuit blocks in a schematic, specific functionality is represented on high-level building blocks (e.g., an Ethernet connector, DVI port or a USB hub). Users simply drag the building blocks that offer the functionality they want onto a board and connect them to each other in a way that suits their design. This is different from a schematic in that all routing takes place behind the scenes, leaving users only to worry about high-level connection specifications. By grouping components into modular building blocks while offering specificity over connections, engineers also retain a large degree of control over their design.

FIGURE 3

The Alto35 development kit designed using Geppetto. Users can expand upon this design at the touch of a button to create their own, custom device.

Completed designs can be shared with other users, cloned by other users for modification, and ordered at the touch of a button for delivery within twenty business days. In this way, Geppetto aims to give users scalability in their designs, while also minimizing time-to-market by cutting development and manufacturing time significantly. Users are also able to order pre-built development kits, the designs for which are customizable on Geppetto, giving embedded developers an even faster way to get started on their design.

Making the Next Generation Internet of Things a Reality

Intuitive embedded design makes the process of creating an electronic device as simple as connecting building blocks available in a standard library. This allows for rapid development of a base configuration and shifts focus from the core functions of a computer to the more complex peripherals. With shared designs, users are able to expand ideas by incorporating their own needs for an even faster design process. An example of this is the Alto35, a Geppetto-designed development kit that offers a touchscreen, networking and even an RC servo (Figure 3). A Geppetto user could, for example, adapt the Alto35 design to create his or her own smart home control center complete with touchscreen controls and Wi-Fi communication with other custom built devices controlling lighting, heating, security and even window coverings. The devices are simply Linux-powered computers, and so the only limitations lie in software design, which means they are generally easier to deal with than those that exist in hardware.

By simplifying the design process and making smarter things easier to design, intuitive design platforms like Geppetto offer enormous potential for experimentation and development in a more connected world. While the first generation Internet of Things relied on "dumb" things like RFID tags, the next generation will increasingly rely on smarter things, such as the Nest thermostat, with better software and networking. Intuitive embedded design allows developers who are experienced in software solutions to design the hardware they need quickly and easily; conversely, it allows hardware developers to bring their designs to life with minimal effort. ◢

Gumstix
Redwood City, CA.
(650) 206-2464.
[www.gumstix.com].