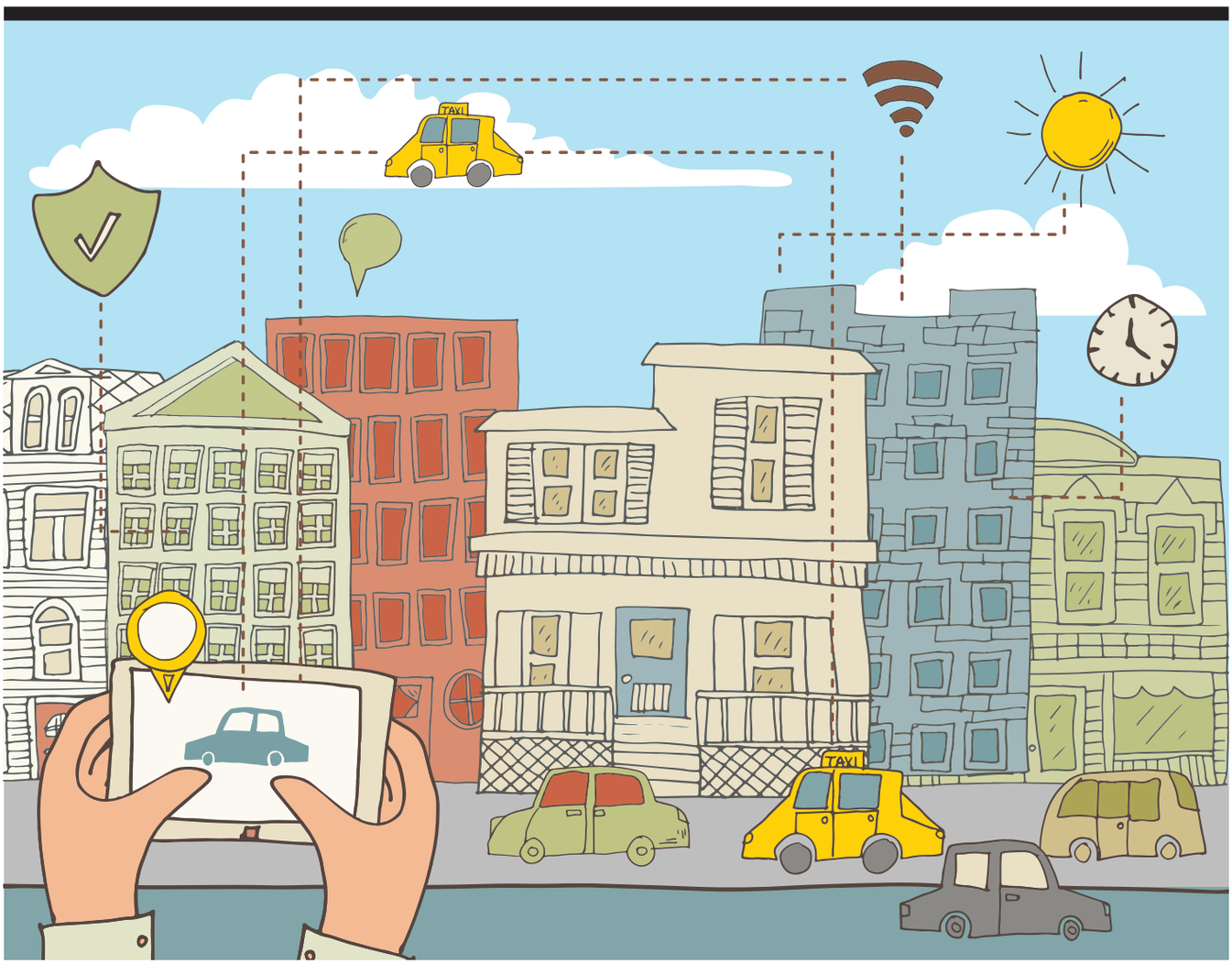
Joshua Adkins, Branden Ghena, Prabal Dutta *University of California, Berkeley* **Editors: Prabal Dutta and Iqbal Mohomed**



Illustration, istockphoto.com

SIGNPOST: Enabling City-Scale Sensing for Citizens and Scientists

The world's population is flocking to city centers at an increasing rate, testing the ability of urban planners and local governments to address new problems in transportation, waste disposal, urban health, and public safety. While one could imagine solutions enabled by an emerging class of smart sensors, current attempts at such systems are difficult to deploy and single purpose in their design. If we expect researchers and citizen-scientists to participate in the data-driven rejuvenation of our urban spaces (and we should, since they are the ones experiencing the day-to-day problems), we must lower the bar to deploying sensors and accessing smart-city data. Towards this goal, we present *Signpost*, an infrastructure-free sensing platform that aims to enable easy, multi-use sensor deployments for citizens, and researchers who have little expertise in building smart and connected sensors.[1]

EASING DEPLOYMENT

The Signpost platform was designed around the idea that the largest impediment to city-scale sensing is the difficulty of deploying sensors. Existing commercial platforms designed GE and Cisco, and research projects, such as the Array of Things [4], all require access to mains power, often necessitating the use of electricians and bucket trucks for installation.

While the importance of these mains-powered sensors in the broader smart-city ecosystem should not be understated – high performance, mains-powered nodes are necessary for some applications – the high deployment burden they impose limits the situations in which they excel. The per-sensor costs of placement and wiring hinders the use of these platforms for dense deployments; the up-front logistical hurdles make them difficult to rely on for experiments with multiple iterations.

The Signpost platform (Figure 1) offers an alternate design point, and only has two deployment dependencies: an existing communications network (such as cellular) and the pervasive sign posts to which it mounts. Rather than relying on mains power, Signpost harvests energy from its vertically mounted solar panel, and can be easily deployed in less than five minutes by a single person with a wrench.

FIGURE 1. The Signpost platform easily mounts to existing street sign posts, harvests energy from an integrated solar panel, and provides pluggable sensor modules with power, communications, processing, storage, time, and location. Signpost is open source, with all hardware and software available on Github.

PROVIDING A SENSING INFRASTRUCTURE

Unlike other smart city sensors, Signpost serves as the infrastructure on which city scale sensing is deployed. Up to five pluggable sensor modules connect to the Signpost using a standard electromechanical interface. Through this interface, they are provided with the key resources necessary for city-scale sensing. These resources – power, networking, storage, time, location, synchronization, and higher-performance compute – were identified by analyzing the building blocks that repeatedly appeared in past city-centric sensor deployments. The resources are provided by core Signpost components, which are themselves modular.

By embracing modularity, the Signpost platform can adapt to emerging applications and support rapid iteration of sensor hardware. The modularity of core platform components enables Signpost to take advantage of improving technology without building and deploying a new sensing platform. The colocation of multiple sensing modalities on a single Signpost allows multiple sensor modules to serve a single application and encourages the sharing of deployment costs between multiple applications. Finally, by providing and sharing key resources among the sensor modules, Signpost eliminates their reimplementation for every sensor deployment, a task that is time-consuming for experienced engineers and difficult or impossible for citizens and domain scientists without embedded design experience.

SUPPORTING DEVELOPERS

Simply removing the need to reimplement the hardware for common services, such as networking and storage, is not enough to make these services accessible to those without embedded design experience. Writing firmware drivers for all of these components would still pose a significant barrier to entry. To address this issue, Signpost makes its system services available through remote procedure calls that occur over a sensor module's common electrical interface. Great efforts have been taken to ensure that this interface remains simple and only requires peripherals that are ubiquitous on embedded hardware; specifically I2C and GPIO are needed for basic operation, but other features such as a GPS-based pulse-per-second (PPS) signal and a high-speed USB link to a high-performance compute module are also included (Figure 2).

In practice, this means that if developers implement a hardware interface layer consisting of I2C and GPIO functions, they get access to all of the Signpost resources. These resources are exposed to applications as a set of library functions (Figure 3). The hardware interface layer is already implemented for common embedded platforms used by the maker community, including Arduino [2] and MBed OS [3] in addition to Tock, a research operating system [5].

Similar to the API exposed by Particle IoT [6], the Signpost backend handles the routing of data between a sensor module and the cloud; that data is tagged with location and timestamps by the Signpost platform. To access a sensor module's data stream, developers simply subscribe to the appropriate topic over MQTT. We hope that by simplifying both the hardware and software interfaces, Signpost can be accessible to both scientists developing new sensors and makers who have experience with off-the-shelf hardware components.

DRAWBACKS OF ENERGY HARVESTING

The deployability that Signpost gains with its energy harvesting design comes at a cost – the limited amount of energy that can be harvested from a solar panel reduces the sensing modalities and applications that Signpost can support. To quantify the amount of energy that could be consumed by each of the five sensor modules, we

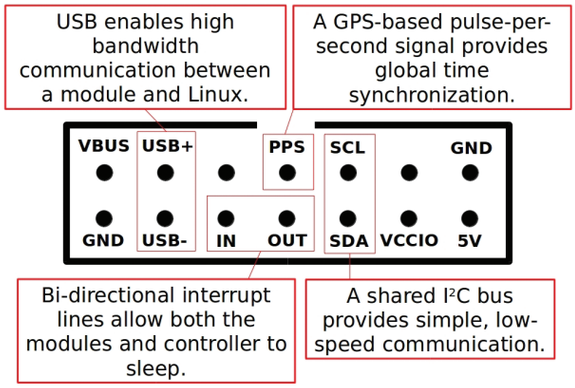


FIGURE 2. The standard header interface for a Signpost sensor module. The only requirements for access to the Signpost resources are a I2C bus and several GPIO pins.

Service	System Call	Description
Init	`i2c_address = module_init(api_handles)`	Initialize module
Network	`response = network_post(url, request)`	HTTP POST data to URL
	`network_advertise(buf, len)`	Advertise data over BLE
	`network_send_bytes(destination, buf, len)`	Send via best available medium
Storage	`record = storage_write(buf, len)`	Store data
Energy	`energy_info = energy_query( )`	Request module energy use
	`energy_set_warning(threshold, callback)`	Receive energy usage warning
	`energy_set_duty_cycle(duty_cycle)`	Request duty cycling of module
Processing	`processing_call_rpc(path, buf, len, callback)`	Run code on Linux compute
Messaging	`messaging_subscribe(callback)`	Receive message from a module
	`messaging_send(module_id, buf, len)`	Send message to another module
Time	`time_info = get_time( )`	Request current time and date
	`time_info = get_time_of_next_pps( )`	Request time at next PPS edge
Location	`location_info = get_location( )`	Request location

FIGURE 3. The Signpost library calls. Sensor modules can access Signpost resources through this API.

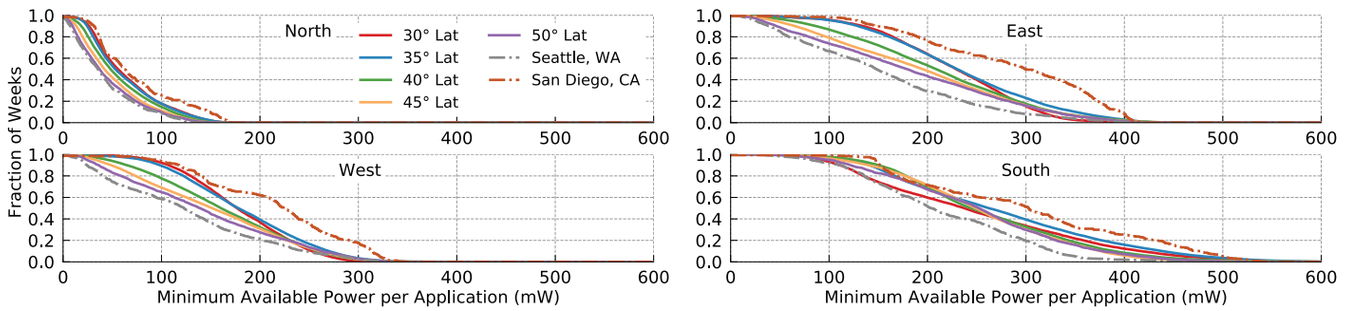


FIGURE 4. Fraction of weeks when a sensor module can expect a minimum power income at different latitudes and cardinal directions. Power available to sensor modules varies by nearly two orders of magnitude.

generate a data-driven model of solar energy harvesting potential for Signposts facing different cardinal directions at different locations across the United States (Figure 4). When compared to the energy harvested on deployed Signposts, this model under-predicts harvesting potential by 3% on sunny days and 22% on cloudy days. From the model, we can see that the average power a sensor module may draw over the course of a week to achieve 95% reliability varies by nearly two orders of magnitude, from only 4 mW for a North facing Signpost in Seattle, WA, to 147 mW for a South facing Signpost in San Diego, CA. A sensor module expecting only 50% reliability could draw 120-210 mW depending on location and direction.

While this power budget may be problematic for some sensing modalities and deployment locations, we provide developers with several Signpost library calls to help manage these constraints. Specifically, rather than requiring sensor modules to enter low-power sleep states, Signpost allows modules to request that they be powered off for a specified period of time.

Combining power constraints with multi-tenancy introduces an additional problem of resource allocation. To ensure the fair sharing of harvested energy, Signpost monitors the energy used by each sensor module and the energy consumed by a module's use of platform resources. For instance, the energy used by a networking call is measured and attributed back to a specific sensor module; the energy required to support time and location services is split among the sensor modules requesting those services. Signpost then provides software methods by which each module can query their available share of energy and their average power consumption.

EXISTING HARDWARE & APPLICATIONS

We create a set of initial sensor modules and deploy them on Signposts on and around campus at the University of California, Berkeley. These modules can sense motion using microwave radar; audio volume at several frequency bands; environmental indicators, such as humidity, temperature, and pressure; and RF spectrum power from 15-2700 MHz. These modules serve not only as example hardware for developers creating their own sensors, but they also are running a suite of example applications, as well, which can help to give a sense of the capabilities of the platform given its power constraints.

To aid the reader, we divide each application, showing the average power draw and breakdown between local computation, local sensing, the platform communication service, and the platform time service (Figure 5).

Weather Monitoring: This application samples temperature, humidity, and pressure and publishes them to the Signpost backend before using the Signpost library to power itself off for four minutes. These measurements are posted to Weather Underground [7] by an application running in the cloud.

Vehicle Counting: The audio volume sensor transmits average volumes of seven different frequency bands every second. In the cloud, a service then analyzes these volumes and counts short-duration peaks (indicative of a car driving by) to estimate the amount of traffic near a Signpost.

RF Spectrum Sensing: The spectrum sensor samples the TV white space channels (every 6 MHz from 470-830 MHz) for 30 seconds of every three minutes. It then calculates the minimum, maximum and average energy on each of these channels and transmits the measurements to the Signpost backend. Like weather monitoring, it saves energy by having Signpost power off the module when it is inactive.

Motion Detection: This application constantly attempts to sense motion using the microwave radar sensor. Because of the sensor's high power consumption, it is often turned off for using greater than its fair share of available energy, and subsequently turned on when Signpost has harvested more energy.

In addition to these sensor modules and applications, we have also created a desktop development kit (Figure 6). This allows developers to test hardware and software without a complete Signpost platform. The development kit includes debugging features, such as serial connections to every module and uses the developer's computer to emulate network connectivity.

CLOSING THOUGHTS

We envision a future in which Signposts are deployed pervasively throughout several test communities, if not more broadly, and we hope these testbeds are used for prototyping smart city applications, running long-term experiments on public health and safety, and ultimately improving the efficiency of the communities in which they are deployed. We also believe that our efforts toward modularity and accessibility could be leveraged in other smart city sensing infrastructure, including higher-performance, mains-powered sensing platforms.

We are only going to achieve this future with buy-in from cities and their citizens. Towards this goal, we are actively seeking involvement from the academic and maker communities, specifically those who want to be involved in designing and building sensor modules. We are also working to make data from existing sensor modules available to those who wish to write city sensing applications. All hardware and software is available at github.com/lab11/signpost. ■

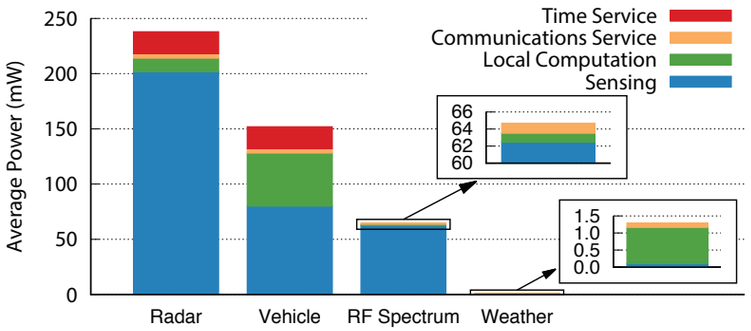


FIGURE 5. The power used by different example applications, and how that power is split between local computation, local sensing, platform communication and platform time services.

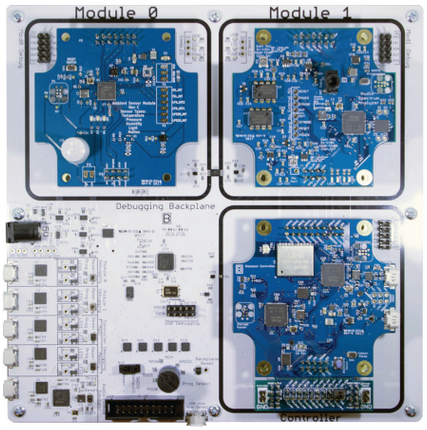


FIGURE 6. The desktop development kit with two sensor modules and the Signpost control module.

Joshua Adkins is a third-year PhD student in Electrical Engineering and Computer Science at the University of California, Berkeley. His current research is focused on easing the programming and deployment of dense sensor networks. He received a BS in Computer Engineering from the University of Michigan.

Branden Ghena is a PhD student at the University of California, Berkeley, studying embedded systems. His research focuses on the design of wireless network protocols that allow for low-power, high-reliability ubiquitous communications between resource-constrained devices, users, and the Internet.

Prabal Dutta is an associate professor of Electrical Engineering and Computer Sciences at University of California, Berkeley. His research interests straddle the hardware/software interface and include wireless, embedded, networked, and cyber-physical systems. He received a PhD in Computer Science from the University of California, Berkeley.

REFERENCES

[1] Joshua Adkins, Branden Ghena, Neal Jackson, Pat Pannuto, Samuel Rohrer, Bradford Campbell, and Prabal Dutta. 2018. "The signpost platform for city-scale sensing." In IPSN'18.
[2] Arduino. 2017. Arduino website. arduino.cc.
[3] Arm. 2017. Mbed OS developer website. os.mbed.com.
[4] Charles E. Catlett, Peter H. Beckman, Rajesh Sankaran, and Kate Kusiak Galvin. 2017. "Array of things: a scientific research instrument in the public way: Platform design and early lessons learned." In SCOPE'17.
[5] Amit Levy, Daniel B. Giffin, Bradford Campbell, Branden Ghena, Pat Pannuto, Prabal Dutta, and Philip Levis. 2017. Multiprogramming a 64 kB computer safely and efficiently." In SOSP'17.
[6] Particle. 2018. Particle website. particle.io.
[7] Weather Underground. 2018. Weather Underground Website. wunderground.com.