The Signpost Platform for City-Scale Sensing

‡Joshua Adkins, †Brad Campbell, ‡Branden Ghena, ‡Neal Jackson, ‡Pat Pannuto, and ‡Prabal Dutta
‡University of California - Berkeley, †University of Virginia

A city-wide sensing platform that is deployable, scalable, and driven by applications.

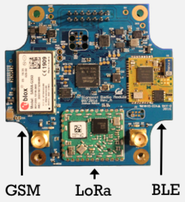
Key Research Themes

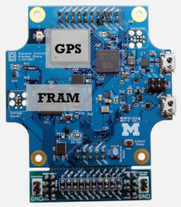
1. **Energy-proportional computing:** Sensing and communication must scale to current harvesting conditions and adjust to energy variability.
2. **Modular system design**: Hardware and software interfaces need to support a wide variety of module implementations.
3. **Distributed applications:** The platform must balance Signpost-local resources, communication bandwidth, and cloud interactions.
4. **Private by design:** Do not collect what must be kept private. Filtering done at the hardware level can help ensure that no identifying data is sent to the cloud.

Signpost provides power, networking, storage, compute, time, location, isolation, and installation to sensor modules through a standardized interface.

Shared Resources. Providing power, networking, storage, location, and higher-performance computation **lowers the bar** to building and deploying a module.

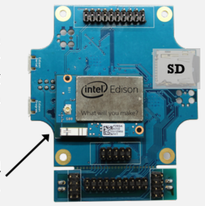
The Radio Module provides cellular (GSM), long range 915mHz (LoRa), and Bluetooth Low Energy (BLE) networking.





The Control Module provides non-volatile storage on an SD card and FRAM. It uses a GPS module to provide location and time data. The Control Module also manages and arbitrates power provided by the solar panel.

Modules will be able to access the general, higher performance computation of the Intel Edison through an in development RPC interface.



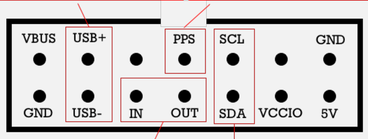
Intel Edison embedded Linux computer.

These components are the most technically difficult parts of designing a sensor system. The signpost platform provides them for you.

Hardware Interface. Sensor modules are added to the platform through a standard electrical and mechanical interface. The interface is designed to provide the necessary features we envision for modules.

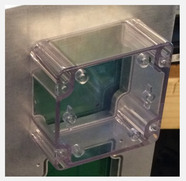
USB enables high bandwidth communication between a module and Linux.

A GPS-based pulse-per-second signal provides global time synchronization.

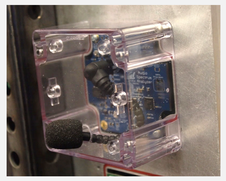


Bi-directional interrupt lines allow both the modules and controller to sleep.

A shared I²C bus provides simple, low-speed communication.



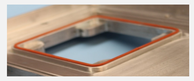
An off-the-shelf case seals modules to the waterproof sensing platform.



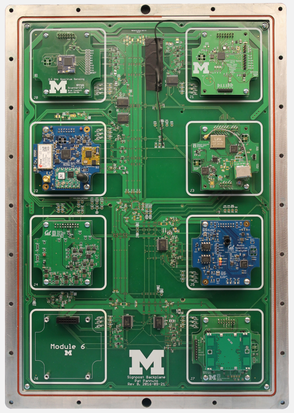
The case can be easily modified to accommodate different sensors.

Isolation. Integrated mechanisms for physical isolation, electrical isolation, and fair distribution of resources ensure reliability and security.

Gaskets limit water damage to a single sensor module if a leak occurs.

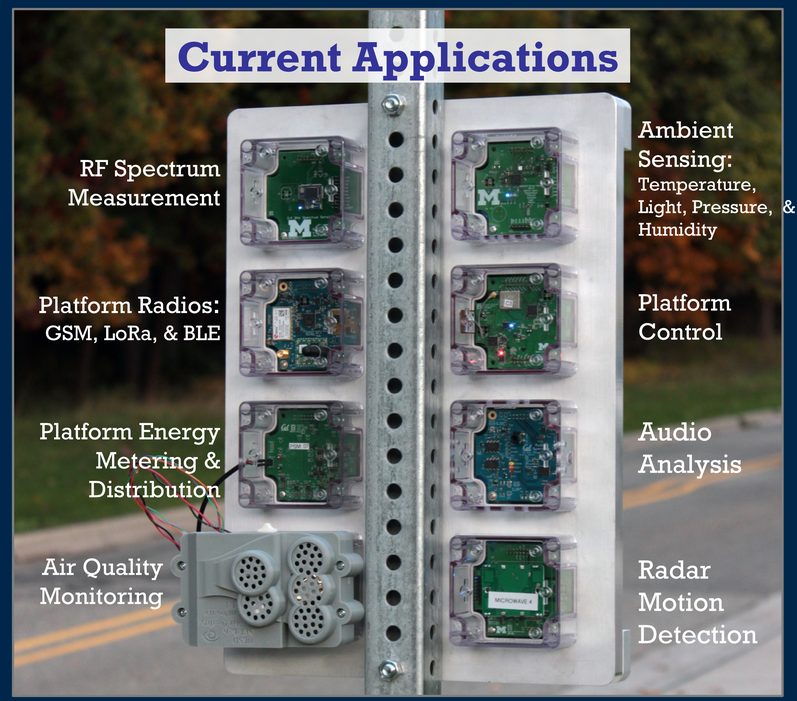


The platform Backplane allows the Control Module to completely electrically isolate a module.



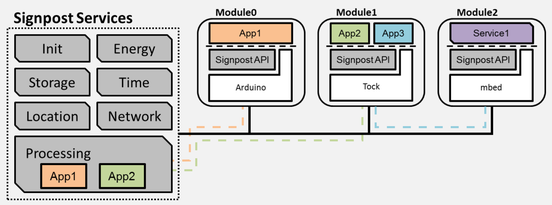
This prevents faulty, malicious, or greedy modules from negatively impacting the entire Signpost platform.

It also allows a module to share private information (such as a key) on the shared bus.



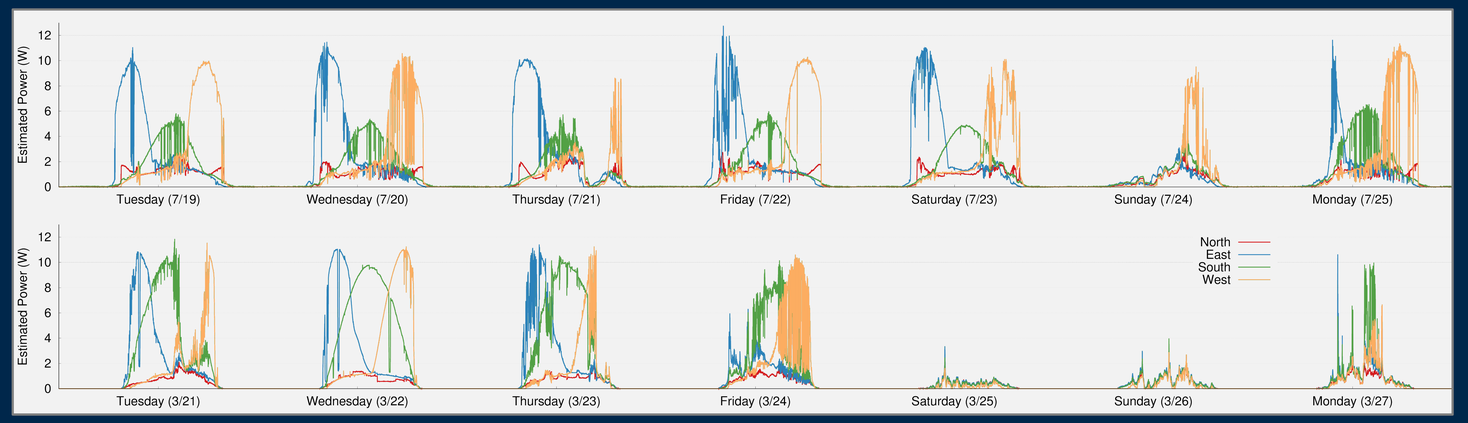
Software Interface. Providing a standard library for accessing system resources supports application developers. Serializing commands over the data bus allows for easy interoperability between many software architectures.

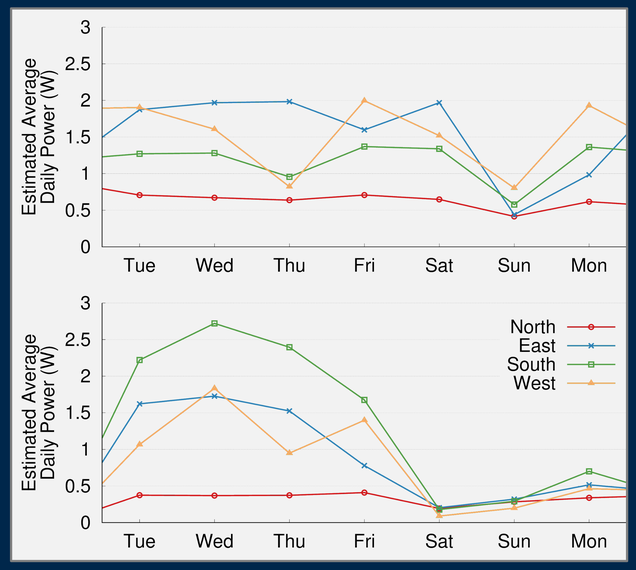
Service	System Call	Description
Init	`i2c_address = module_init(**api_handles)`	Initialize module
Network	`response = network_post(url, request)`	HTTP POST data to URL
	`network_advertise(buf, len)`	Advertise data over BLE
	`network_send_bytes(destination, buf, len)`	Send via best available medium
Storage	`record = storage_write(buf, len)`	Store data
Energy	`energy_info = energy_query( )`	Request module energy use
	`energy_set_warning(threshold, callback)`	Receive energy use 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



Modules can run any software stack they choose. Implementing the Signpost libraries only requires an I²C peripheral, timers, and GPIO.

Signpost currently supports the Tock, Arduino, and Mbed environments.





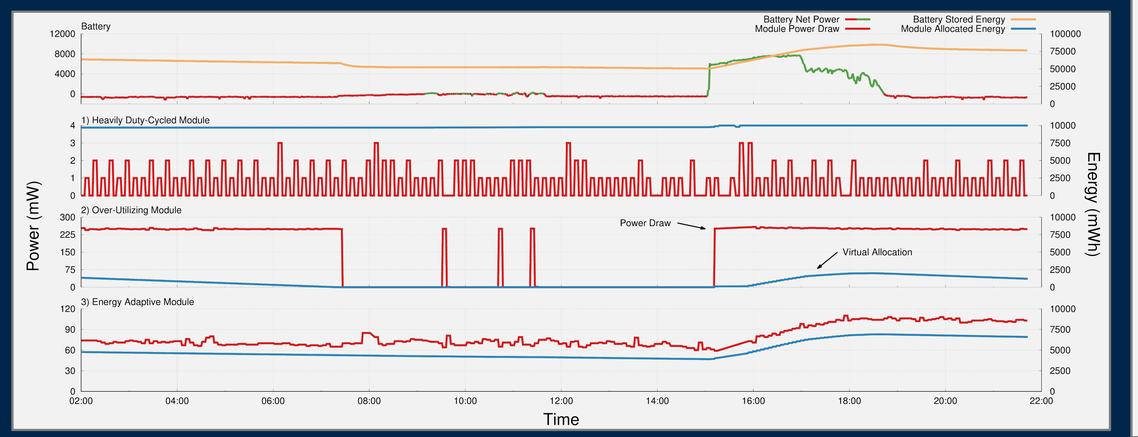
Variability in Harvesting. Emphasizing the ability to deploy Signposts facing any direction leads to increased variability in harvested energy.

Differences in direction, season, and weather lead to instantaneous power varying between 0 Watts and 12.9 Watts, with average daily power ranging from 0.09 Watts to 2.7 Watts. The Signpost platform needs to be able to adapt to varying energy availability, capitalizing in times of plenty and conserving in times of famine.

Energy Adaptivity in Practice. Adjusting to platform energy variability requires software primitives to support energy awareness and adaptivity.

Energy on the system is allocated to modules as "virtual batteries", ensuring that each module gets a fair share. Incoming energy is distributed evenly between modules. In the future, these could become priority based allocations.

Shown at left, modules can request to be duty-cycled at particular rates in order to operate continuously (1), ignore energy constraints altogether and run only when energy is available (2), or dynamically adapt to current conditions (3).



Technology Transfer. Signpost has been demonstrated at the Terraswarm Annual Review (2016), the 14th ACM Conference on Embedded Networked Sensor Systems (SenSys 2016), and at Intel (2017). A demo paper has been published at SenSys'16: "Demo Abstract: The Signpost Network".

Signpost is an entirely open-source project, dual licensed under MIT/Apache-2.0.

All source files are available on Github:
https://github.com/lab11/signpost
https://github.com/lab11/signpost-software

Embedded Systems Research at the University of California, Berkeley
https://lab11.eecs.berkeley.edu https://github.com/lab11