The Signpost Platform for City-Scale Sensing

Joshua Adkins, Brad Campbell, Branden Ghena, Neal Jackson, Pat Pannuto, Samuel Rohrer, and Prabal Dutta
Wouldn’t it be cool to...

- Measure air quality throughout downtown
- Monitor traffic and pedestrian flow
- Detect emergency situations

Cities could use data to make policy changes

Real-time applications could use the data to improve process efficiency

Individuals could use data to understand their environment and change their behavior
SONYC - Distributed Audio Sensing

Array of Things - Environmental sensing


[2]
These existing systems are hard to deploy

Limits potential deployment scenarios

- Limited deployment opportunities
  - Requires existing power
  - Must be in range of WiFi
- Political
  - Infrastructure changes are difficult
- Expensive
  - Hiring city works to perform deployments

The Signpost Platform: Infrastructure-free Infrastructure

No dependencies except for a standard signpost
- Solar energy harvesting
- Multiple wireless networking options
- Easy (two bolt) installation

Provides the infrastructure to enable city-scale sensing
- Modular design adapts to different cities
- Sensor modules access key services
- Isolation enables sharing of the platform
Modules plug into a standard interface

Audio spectrum module

Environmental sensing module

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.
<table>
<thead>
<tr>
<th>Deployment</th>
<th>Services Needed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Power</td>
</tr>
<tr>
<td>Caraoke [3]</td>
<td></td>
</tr>
<tr>
<td>Bouillet et al. [4]</td>
<td></td>
</tr>
<tr>
<td>Aircloud [5]</td>
<td></td>
</tr>
<tr>
<td>Girod et al. [6]</td>
<td></td>
</tr>
<tr>
<td>Ledeczi et al. [7]</td>
<td></td>
</tr>
<tr>
<td>SenseFlow [8]</td>
<td></td>
</tr>
<tr>
<td>Argos [9]</td>
<td></td>
</tr>
<tr>
<td>SONYC [1]</td>
<td></td>
</tr>
<tr>
<td>Kyun Queue [10]</td>
<td></td>
</tr>
<tr>
<td>Micronet [11]</td>
<td></td>
</tr>
</tbody>
</table>
Core Modules Provide Services to Sensor Modules

- **Power Module**
- **Control Module**
- **Radio Module**
Power module provides and monitors power

- Regulates power from the battery
- Monitors energy usage by each module
- Charges battery from the solar panel
Control Module Manages the Platform

- Time, Location, Synchronization
  - Provided by GPS
- Bulk storage on SD Card
- Energy usage statistics
- Higher performance compute
  - Runs on Intel Edison
  - Accessible through RPC Interface
Radio Module Provides Networking

- Cellular
  - Fast/higher reliability
- LoRa
  - Long Range 915 Mhz band (100-1000kbps)
- Bluetooth Low Energy
  - Signpost-to-phone communication
Platform Provides Isolation

- Guarantees module access to the other services
- Mechanical isolation in the case design
- Electrical isolation in backplane (interconnect)
- Control module manages resource isolation
  - Energy
  - Networking
  - Storage
  - Compute
## Signpost Software API

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

- The directions are comparable
- A vertical solar panel is not too detrimental
- Enough energy to run many city-scale applications
- Not enough energy to run all applications on a Linux Computer
Can modules adapt to varying energy?

- Implemented a simple energy policy
- Incoming energy is split between module’s “virtual batteries”
- Module energy usage is subtracted from their “virtual battery”
- If a module uses too much energy it is cut off
- If a battery is full the energy is redistributed

Three modules running
- Very low power duty-cycled module
- High power module
- Module adapting to a target lifetime
Energy Adaptivity in Practice
Future Vision

Deployment on Berkeley campus
- Starting with 5
- Grow to 25 by October

Collaboration to build applications
- SeaGlass IMSI Catcher Detection [12]
- ChemiSense air quality monitoring
- Dynamic wireless channel selection through RF spectrum sensing

Signpost Development Kits

Emulates a running Signpost
- Platform for module development, bring-up
- Test applications with different energy profiles
- Exposes more debugging output
Diverse and Growing Set of Modules

15-2700 MHZ RF Spectrum

Environmental Data

Speed and Motion

Audio Spectrum
The Signpost Platform for City-Scale Sensing

Joshua Adkins, Brad Campbell, Branden Ghena, Neal Jackson, Pat Pannuto, and Prabal Dutta
References

[8] Li et al. An Experimental Study on People Tracking. 2015