Critical Sign Tracking

May 14-18

Team Members: Aaron Cannon, Zheng Luo, David-Paul Adeola, Simeng Liu

Client: Dr. Halil Ceylan Advisor: Dr. Daji Qiao

Mission:

To reduce response time in fixing critical warning signs that have become non-functional by means of severe weather or vandalism. By reducing response time we hope to reduce potentially fatal accidents that can occur if a critical warning sign is non-functional for a great period of time.

Design Requirements:

Functional:

- Tilt Threshold: Once the sign tilt becomes greater than threshold, an alert will be sent.
- Send Alerts: Alerts are only sent out if an event occurs.
- Wireless Communication: End user should be notified by a means of wireless communication.
- Self-Calibrated: Once turned on the system will self-calibrate.
- False Reports: There should be false notifications or lack of notification.
- Power Supply:The system should be supplied a constant 5V.

Non-functional:

- Size: The enclosure shouldn't exceed more than 20% of the enclosed hardware.
- Weather Resistant: The system should be enclosed so that it is resistant to all possible weather conditions in Iowa. The conditions should include temperature (-22 F to 120 F), waterproof, and wind resistant.
- Cost-Effective: The price for the package should be no more than $155.
- Easy to Install: Installation should take no longer than 10 minutes.
- Lifetime: System should be able to sustain itself for 10 years.

Design Details:

Data Management:

To manage the data that our sensors were receiving, we elected to use an Arduino microcontroller board. The Arduino offered us a lot of flexibility in sensor choices and an ease it interfaces each peripheral device.

The Arduino monitors the tilt and vibration of the sign using an accelerometer. The vibration is calculated based on the standard deviation of the acceleration encountered and is used to adapt the sleep cycle. If the tilt reaches a threshold, a SMS alert is sent using a GPRS shield.

Power System:

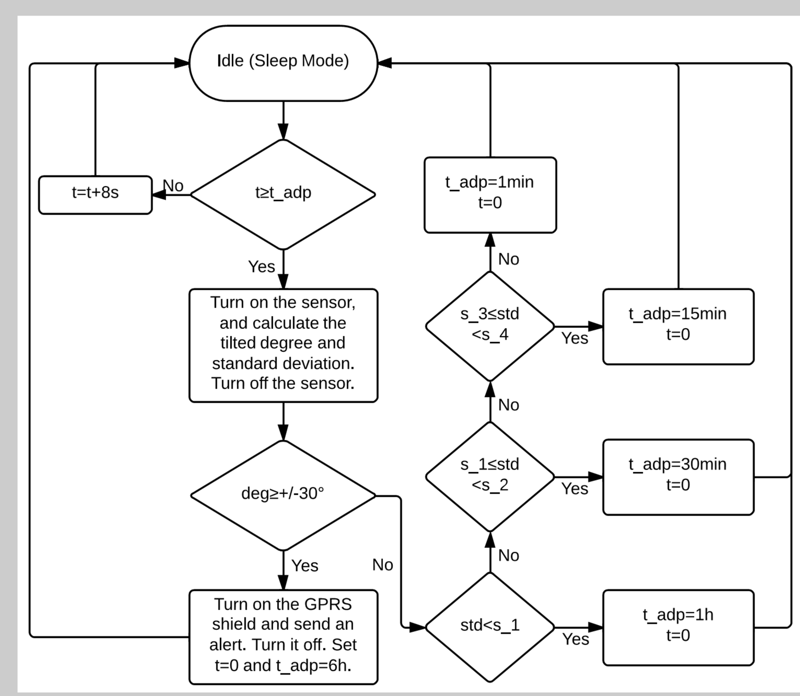
Since our system would be attached to signs in remote locations we had to harvest our own and store it in a manner that would supply reliable power for an extended amount of time. To harvest energy, a 7.2V, 200mA solar cell is used. To store energy we have a 3.7V, 1100mAh Li-Ion battery which should last a minimum of 3 days. To regulate charging and usage by Arduino, a charging circuit is used.

Embedded Program:

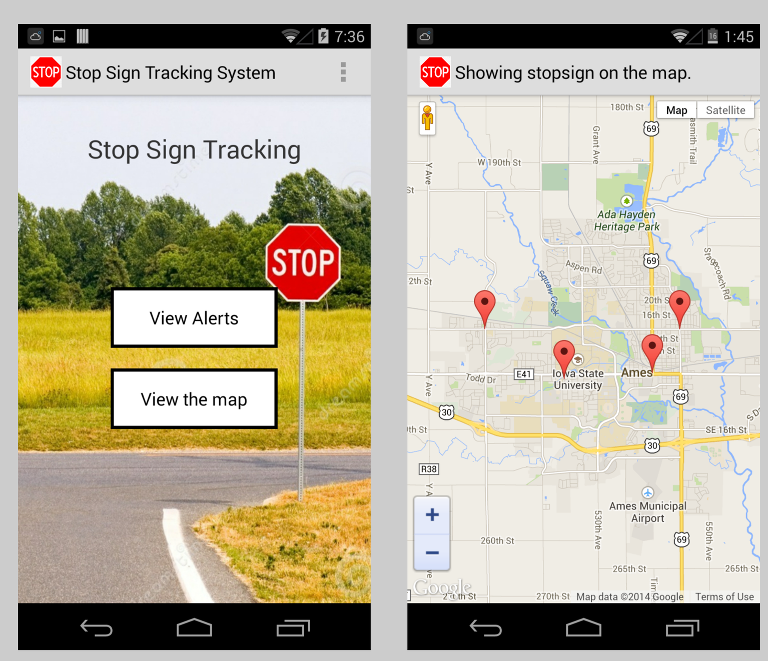
To link all our hardware together, an embedded program was written and ran on the Arduino. The program interfaces the accelerometer and gprs shield. Using the calculated standard deviation from the accelerometer an adaptive sleep cycle is implemented to conserve energy. If total tilt exceeds threshold of 15 degrees a SMS alert is sent out to user. SMS sent using GPRS shield with the correct AT commands.

User Interface:

The android application is an easy way to view all alerts received by the end users cell phone. It can filter all alerts from the messages in the cell phone's inbox. It can also show all issued systems on a google map at the same time, which is convenient to monitor the status of signs.



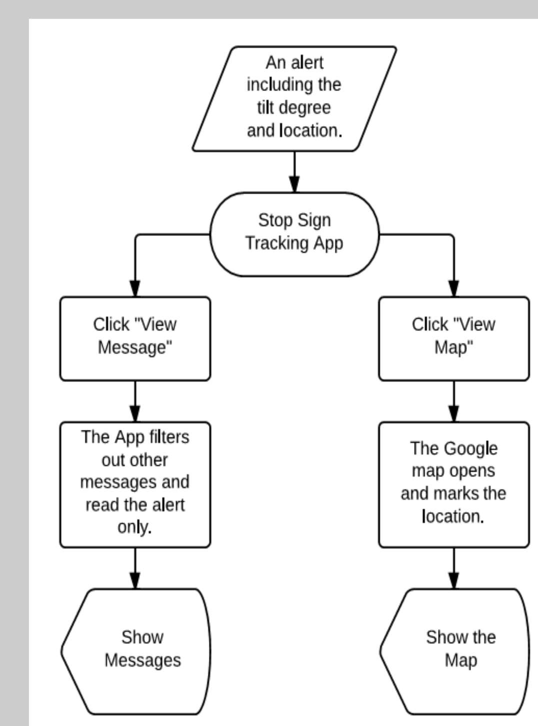
Embedded System Flow Diagram



App Screenshot

Software

- We used a GPRS shield attached with an AT&T sim card to send the alert text message to our clients. There will be a location and a hyperlink in the text. After clicking the link, a google map with maker of the stop sign location will jump out.
- Another easier way to manage the alert text message is to use our own android app. If the clients got a lot of alerts messages, it can show the multiple alerts location and markers on google map at the same time.



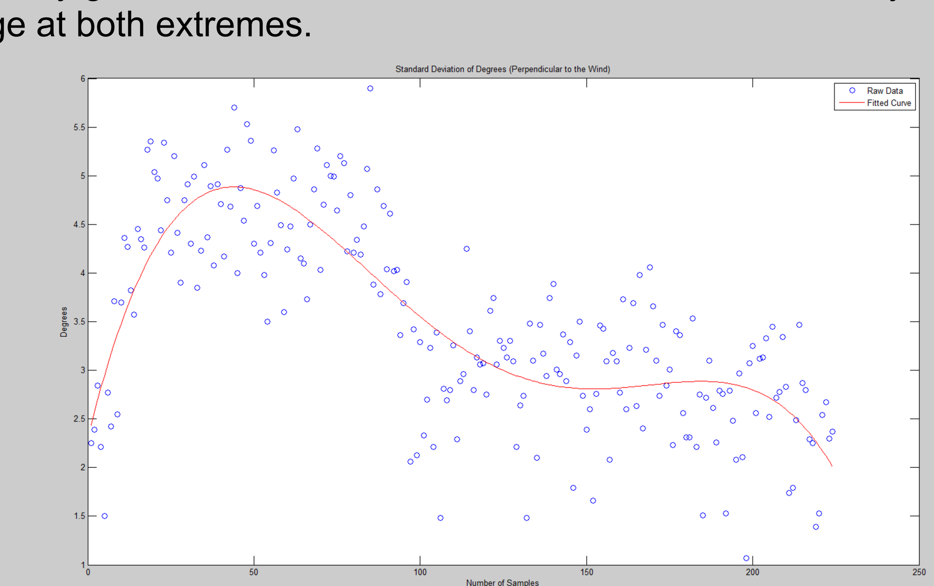
User Side: Cellphone Software Design

Testing

Cell Test: Intermittently issued AT+CSQ command to view the cell signal strength while driving. Results showed that reception in central Iowa is strong and reliable.

Wind Test: We gained access to the wind tunnel so we could simulate a stop sign shaking in the wind. Using this data we can reliably adapt our sleep cycle based on conditions. We were also able to verify that our system is strongly secured within the enclosure.

Temperature Test: To verify our system would operate in the extreme Iowa temperatures, we tested the extremes -30 F and 120 F. The accelerometer successfully gave valid values and the GPRS shield successfully sent a text message at both extremes.

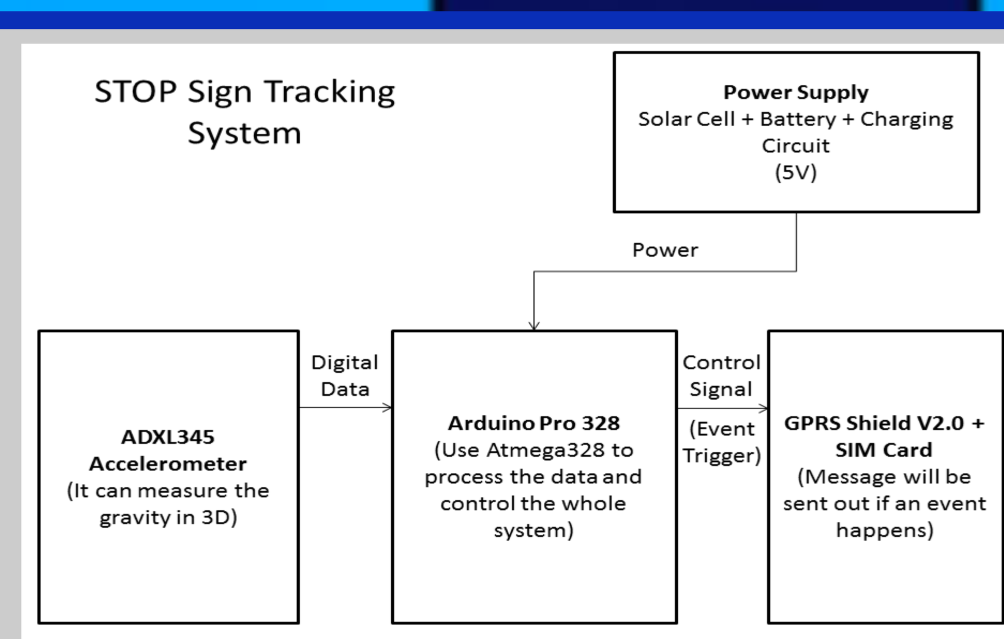


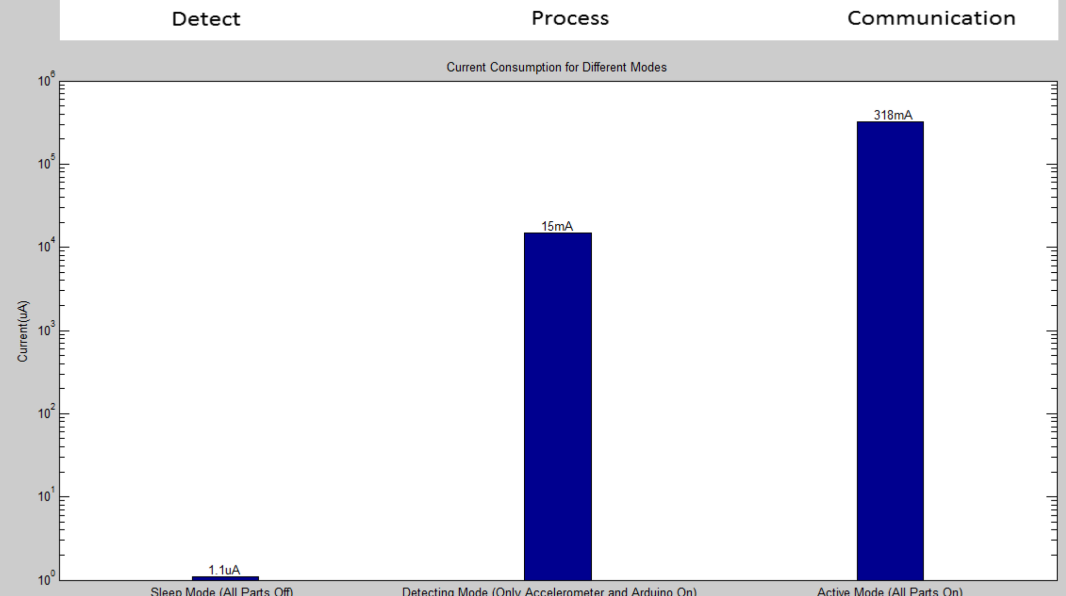
Hardware

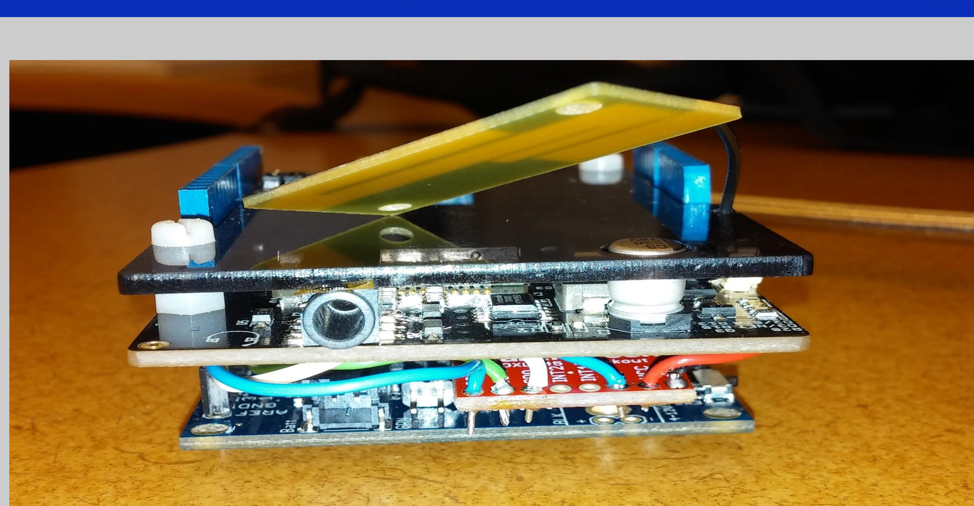
A 3.7V, 1100mAh Li-Ion battery is used to store the energy for the system. Based on different duty cycles, the battery can last for different time.

- If all of parts are turned on:
  - GPRS shield: 303 mA (for 1900 MHz frequency band used by AT&T in Iowa).
  - Accelerometer: 40 μA in measurement mode.
  - Microcontroller: 15 mA in running mode (5V, 16MHz).
- If all of pats keep in sleep mode:
  - GPRS shield: no power consumption (disconnected).
  - Accelerometer: 0.1 μA in standby mode.
  - Microcontroller: 1 μA in sleep mode.

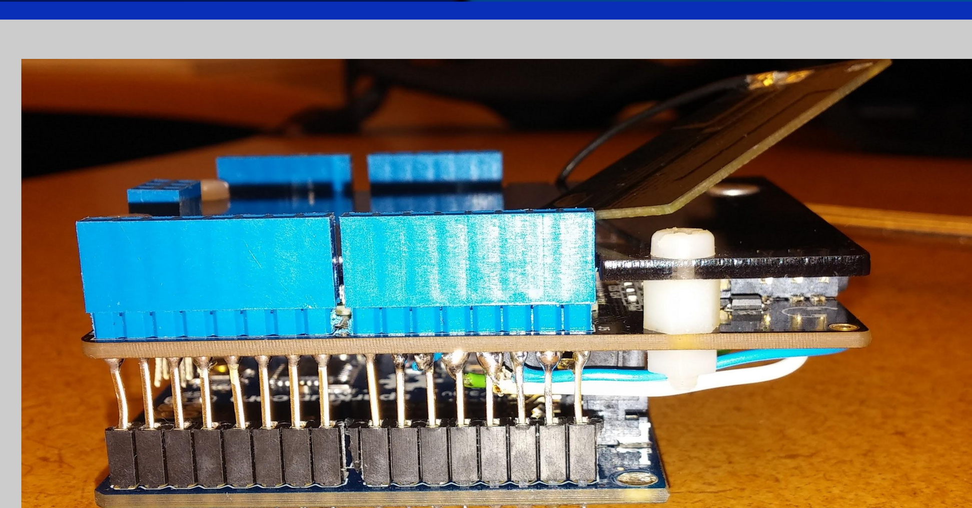
In general, the system can last for 3 days in a windy season without charging.







Front View of the Arduino, Accelerometer, and GPRS Shield



Side View of the Arduino, Accelerometer, and GPRS Shield

Final Design