Critical Sign Tracking System

Design Document

Aaron Cannon, David-Paul Adeola

Zheng Luo, Simeng Liu

Table of Contents

I. Introduction

In this project, which is a stop sign tracking system, we need to design an event-trigger tracking system which can send out alerts when the stop sign is tilted. So we need to have sensors and a communication system. In order to connect those two parts together, we also need a good microcontroller. Besides, a power supply is necessary to support all of those three parts. In this project, we have to think about the price of each part because the cost efficiency for stop signs is important. Additionally, we must consider carefully about the operating temperature because the whole system will work outside. The remaining things are the integration of those components and the embedded system.

1. Functional Requirements:

- Tilt threshold: Once the tilt becomes greater than 30 degrees, an alert should be sent out.
- Send Alerts: Alerts should be event driven and only a message is sent out if an event occurs. Events are something that is being monitored that will render the sign nonfunctional or an incoming request for information. The sensor should specifically figure out what happens to the STOP sign.
- Wireless Communication: Communication to mobile devices should be wireless to keep cost low.
- Self-Calibrate: System should calibrate itself to reduce maintenance and installation time.
- False-Positives: 10% of alerts can be false-positive alerts (alert sent out that sign broke constraints but constraints weren't actually broken).
- False-Negatives: No false-negatives are allowed (No alert sent out even though constraints were broken).
- Power Supply: System (the battery) should be supplied 5V. Some of which will be used to directly supply system and the rest to be used to charge a rechargeable source.
- Tracking: Should be able to locate the sign from the signal.

2. Non-functional Requirements:

- Size: The package shouldn’t be larger than 20% of the hardware being used.
- Package: The system should be packaged with ESD materials that are weather resistant.

- Weather Resistant:

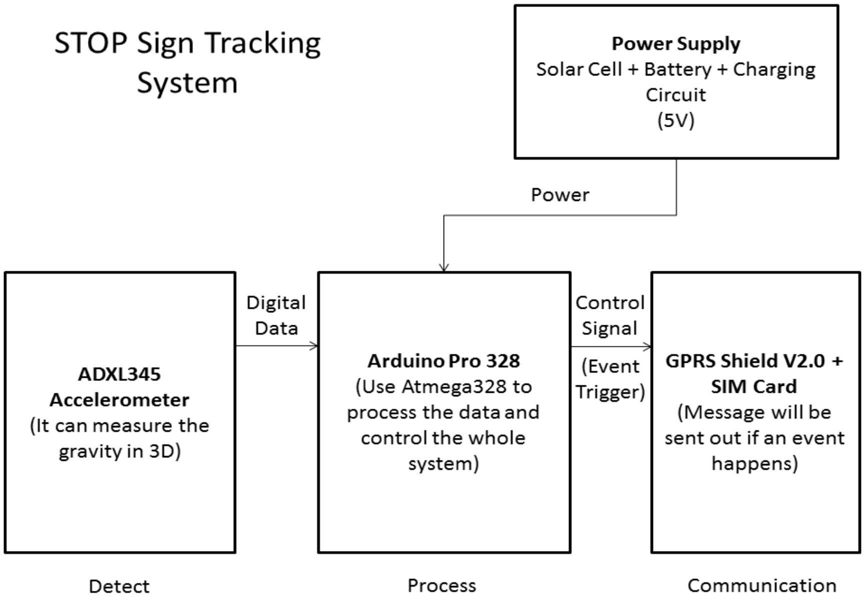
  -The package should operate in temperatures from -30 F to 120 F.

  -Water-proof (rain, snow, fog…).

  -Wind resistant: should stay attached to sign under strong wind (~30MPH).

  -Easy to Install: Installation onto stop sign should be no more than 10 minutes.
- Lifetime: System should be able to sustain itself for 10 years.

II. Device Systems



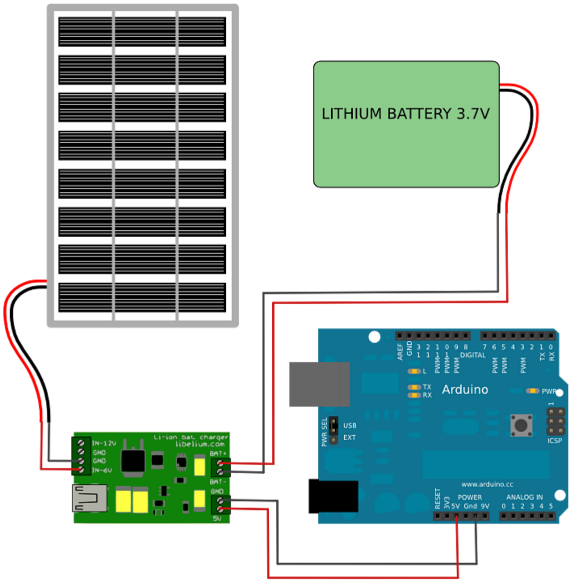
1. Power Supply System

The power supply system consist of a 7.2 V flexible solar panel, 5V Regulator/Charger Module w Li-Ion Battery, connected to the rest of our communication system (Arduino, accelerometer, and GPRS shield). The solar panel will connect to the charging circuits which will output to the Li-Ion battery and the communication system.

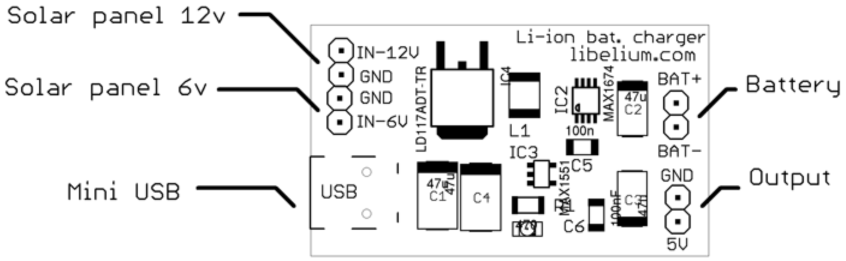
The module provides a regulated 5V DC (300mA) output via screw terminals to power the communication system. The regulator and charger accept DC input power to charge the Li-Ion battery via one of three connections:

- Mini-USB connector allows you to charge the battery via a PC's USB port or any other USB power source.

- 2-pin screw terminal accepts 3.7V DC ... 6V DC voltage sources, such as a solar panel.
- 2-pin screw terminal accepts 3.7V DC ... 12V DC voltage sources.



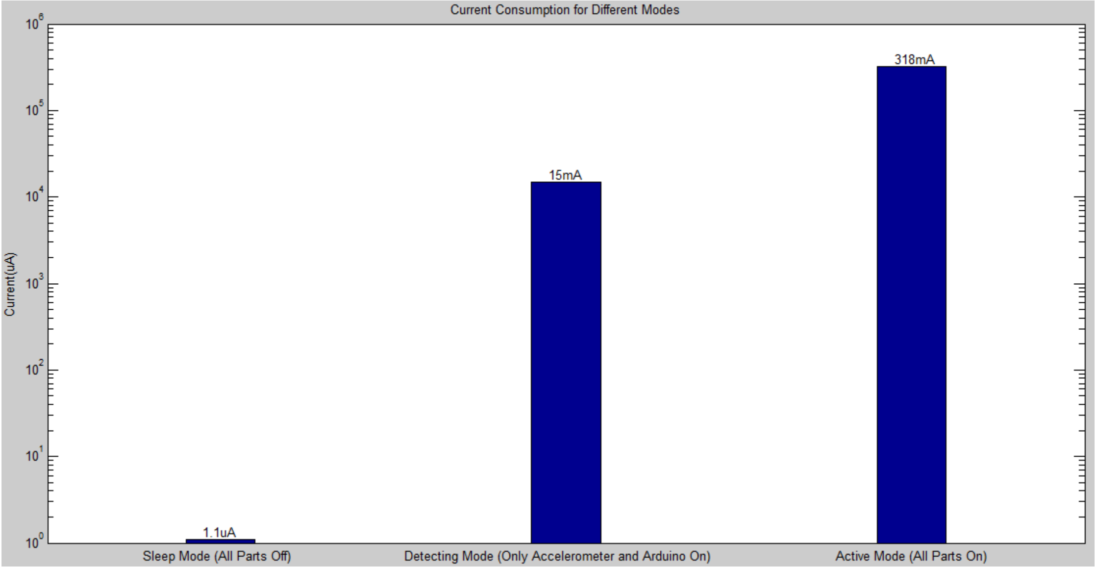
The 3.7V...12V DC input is connected to a step down voltage regulator, while the other two voltage inputs connect directly to the Li-Ion battery charger, thus providing better efficiency. A MAX1674 step-up regulator converts the Li-Ion battery voltage to a regulated 5V DC output voltage that is accessible via a 2-pin screw terminal to power the device.



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: 40µA in measurement mode.
    -Microcontroller: 15mA in running mode (5V, 16MHz).

- If all of pats keep in sleep mode:
  - -GPRS shield: no power consumption (disconnected).
  - -Accelerometer: 0.1μA in standby mode.
  - -Microcontroller: 1μA in sleep mode.



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

- The energy for this system is collected by the solar film from daylight.
- The thickness of the solar film is 1mm with weather proof.
- The solar film has a dimension of 270mm×175mm, and a 7.2V, 200mA output in standard operation.

2. Tilt Sensor

A tilt sensor is necessary to measure if the stop sign is tilted. In fact, it is an accelerometer. When the stop sign is tilted, the sensor will be able to know how many degrees the sign is tilted by. If the range exceeds ±30 degrees, an alert should be sent out. Typically, a tilt sensor is very cheap compared to other components. It can be just connected to the header of the board of the microcontroller. After comparing with other choices, we decided to use ADXL345 accelerometer.

ADXL345 Accelerometer

-Temperature Range: -40 ℃ to +85 ℃

-Price: $27.95

-Supply Voltage: 3 to 5V

3. GSM Module

A GSM module is the communication system that we would like to use. Alerts should be event driven and only a message is sent out if an event occurs. The message should contain the position of the stop sign and also specify if it is tilted by certain degrees. Besides, the communication to mobile devices should be wireless to keep cost low. The GPRS Shield V2.0 can work in the temperature range demanded and has a good interface for the Arduino Pro 328. The price is also acceptable.

GPRS Shield V2.0 (SIM900)
- Temperature Range: -30 ℃ to +80 ℃
- Price: $59.90
- Supply Voltage: 3.4 to 4.5V
- Interface: UART

4. Microcontroller

A microcontroller is necessary to process the data from sensors and control the GSM module to send out alerts when events happen. The ATmega328 has an excellent operating temperature range and is able to control the whole system. Arduino Pro 328 which uses ATmega328 has a very cheap price compared to other boards.

Arduino Pro 328 (ATmega328)
- Temperature Range: -40 ℃ to +85 ℃
- Price: $14.95
- Supply Voltage: 3.3V to 12V, or 5V to 12V (Two Models)
- Clock Frequency: 8MHz (3.3V Model) or 16MHz (5V Model)
- Memory: 32k Bytes of ISP Flash
- Interface: UART
- Size: 53.34×54.08mm (L×W)

III. User Interactions with the System

1. Clients

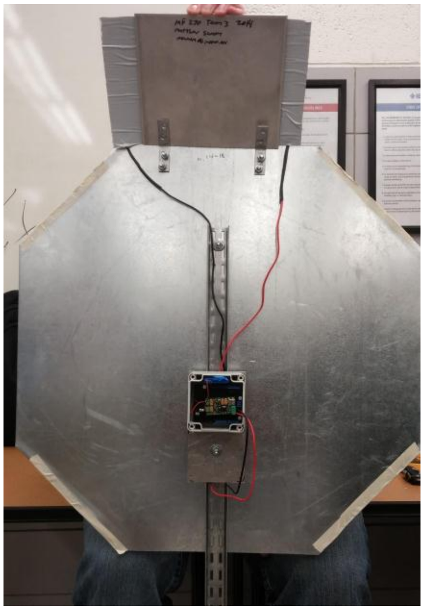
First of all, our clients are people who are working at traffic departments, city or county office. One of their responsibilities is to check status of stop sign. If the stop sign falls down or is stolen, the user will be notified via text message.

2. Product Installation

We are using NEMA enclosure box type to contain our devices inside. NEMA is rated to protect against designated environmental conditions. A typical NEMA enclosure might be rated to provide protection against environmental hazards such as water, dust, oil or coolant or atmospheres containing corrosive agents such as acetylene or gasoline.

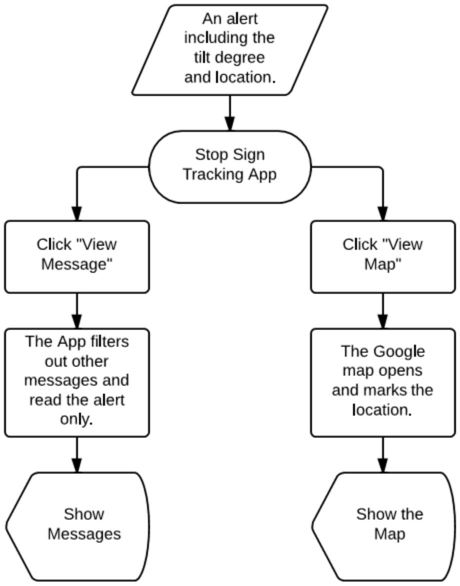
We are going to put devices connected in NEMA box. There are usually four holes for screws to hold the box with high stability. Then we will use screws to connect NEMA box to the pole of the stop sign. The solar cell will be put on the top edge of stop sign for getting enough solar energy. The wire to solar cell will go into NEMA box so that it can supply power to each device.

Figure below: Final setup of device.



3. Communication between Clients and Products

We are going to write the software so that once the modules are installed and powered on for the first time the system should self-calibrate. Once powered on the tilt sensor keeps checking the angle between stop sign and horizontal ground. The sensor will transmit the data to the microcontroller, which determines if the tilt has reached a pre-determined tilt threshold of 30 degrees. Once the data reaches or exceeds this threshold, the microcontroller is going to control GSM module to send a text message to certain clients' cell phones. There are two methods to monitor the status of stop signs.

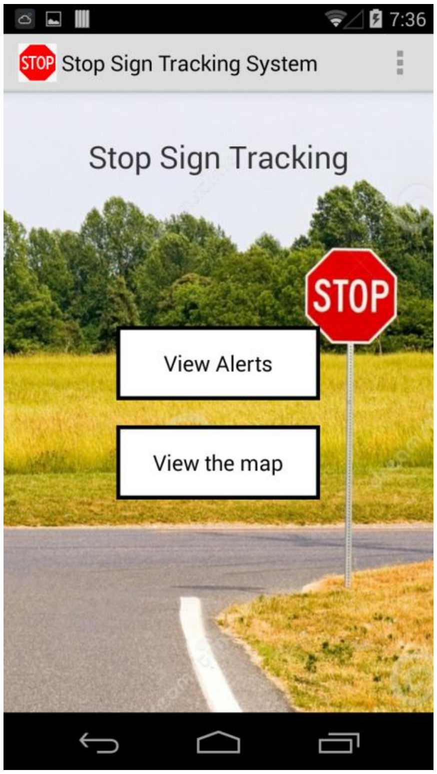


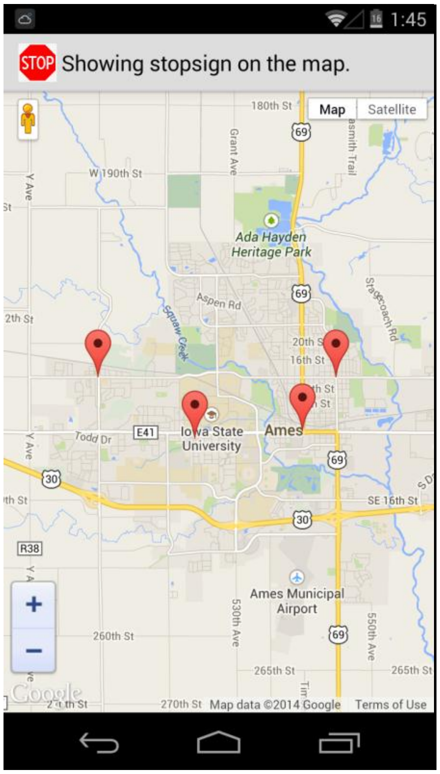
a) Text message notification:

Before the product is installed, each GSM module will be programmed and be ready to send specific information including location of each devices with stop sign. Besides, more information like strength of wind, temperature can be edited. Then, the clients' phones will be notified by a text alarm that contains location of the stop sign once stop signs tilted.

b) Android application interface:

An android application is developed as a user interface. It is an easy way to view all alerts received by our client' cell phone. It can filter all alerts from many messages in cell phone's inbox. It can also show all issued stop signs on Google map at the same time, which is convenient to monitor the status of stop signs and collect data from cell phone.





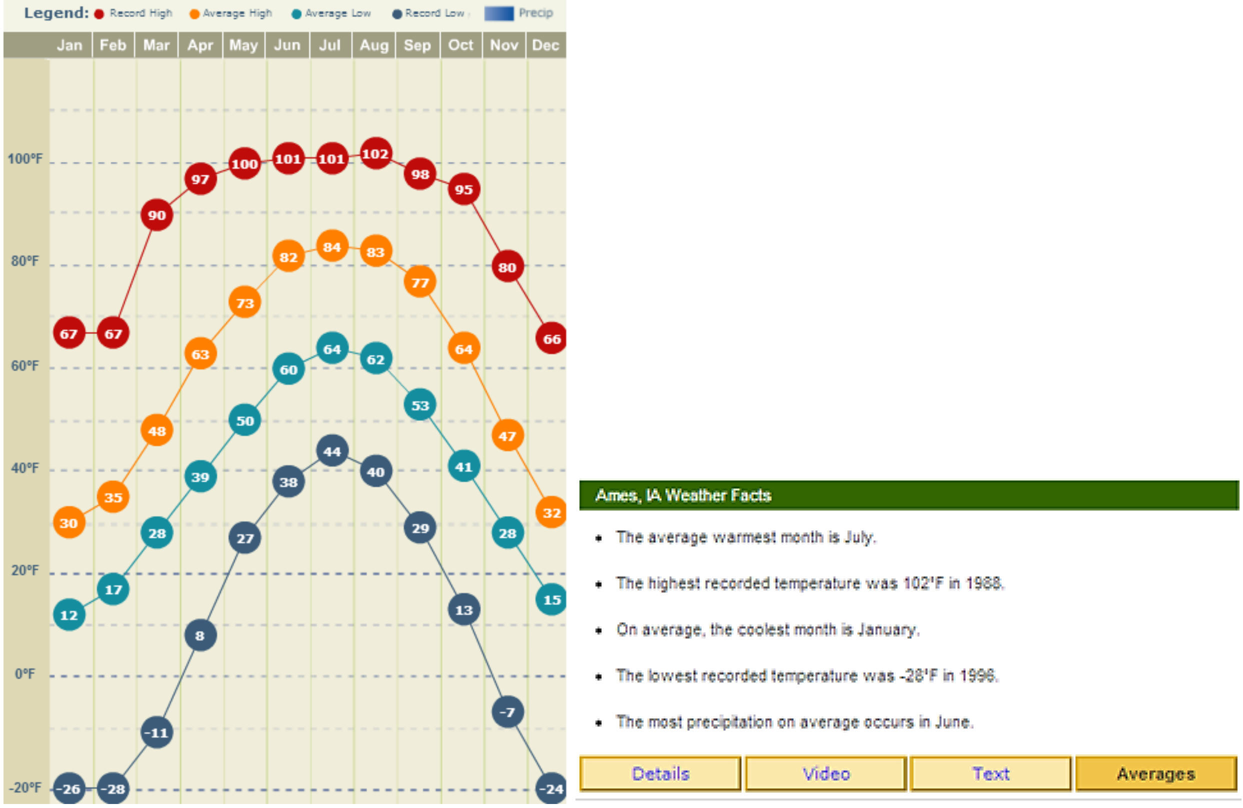
Showing stopsign on the map.

IV. Testing methods/verification

For this project to be successful the device needs to function under Iowa weather. The device needs to be able to work during the hottest parts of the summer and the coldest days in the winter. The device needs to work properly through all-weather including snow and rain. To do this we need to perform some durability tests to make sure the device can function properly.

1. Temperature Test

For first Test we need measure is temperature, we will test the device in extreme cold and hot weather. Each part of the device has its own temperature range specified on their data sheets. The accelerometer has a temp range -40 to 85 ℃, the GSM has a range from -30 to 70 ℃ with a max of -40 to 85 ℃, and the Arduino pro (ATmega328) -40 to 85 ℃. In Coover Hall, there are heating chambers in several of the hardware labs we plan on using the heat stations to test the heat range up to 120 degrees. Research showed that the highest recorded temperature in Iowa was 102 ℉ in 1988. And to test the cold we will use a freezer box up to test temperature at -35 degrees F. The lowest recorded temperature in Iowa was -28 ℉ in 1996.



Ames, IA Weather Facts

- The average warmest month is July.
- The highest recorded temperature was 102°F in 1988.
- On average, the coolest month is January.
- The lowest recorded temperature was -28°F in 1996.
- The most precipitation on average occurs in June.

We did both hot and cold testing in the temperature chambers found in Coover 2046. We tested both extremes hot and cold which provided a range of -22 F to 120 F. To conduct the tests we placed the device in the chamber and allowed the chamber to reach the desired temperature. While the device was in the chamber we printed data from the accelerometer out to the attached computer. We also turned the GPRS shield on and off. Once the chamber reached the desired temperature we kept it at that temperature for 5 minutes while monitoring the output. This test verified that the components that we have chosen will be able to withstand the typical weather found in Iowa.

2. Wind and Vibration Tests

To test the shock/ vibration and wind test we will use the labs in Howe Hall, Howe Hall has labs for mechanical vibrations, acoustics, and stress analysis. The average wind speed in Story county of Iowa is 21.21 mph another option is using a vibrating plate to measure the vibrations of the device and to see the after effects.

We were successfully able to gain access to the low-speed wind tunnel located in Howe Hall. We tested our system attached to a prototype set-up at three different wind-speeds and four different angles to the wind. We tested at wind speeds of 10 mph, 20 mph, and 30 mph and angles of 0, 30, 45, and 60 degrees. We were testing for average values returned by the accelerometer to correctly calibrate our embedded program according to the weather conditions being experienced. The general trend we observed was that at higher wind speeds the sign shook more and as the wind speed decreased the amount of shaking observed also decreased. The data can be seen in the figures below.

Figure Set 1: acceleration of degree

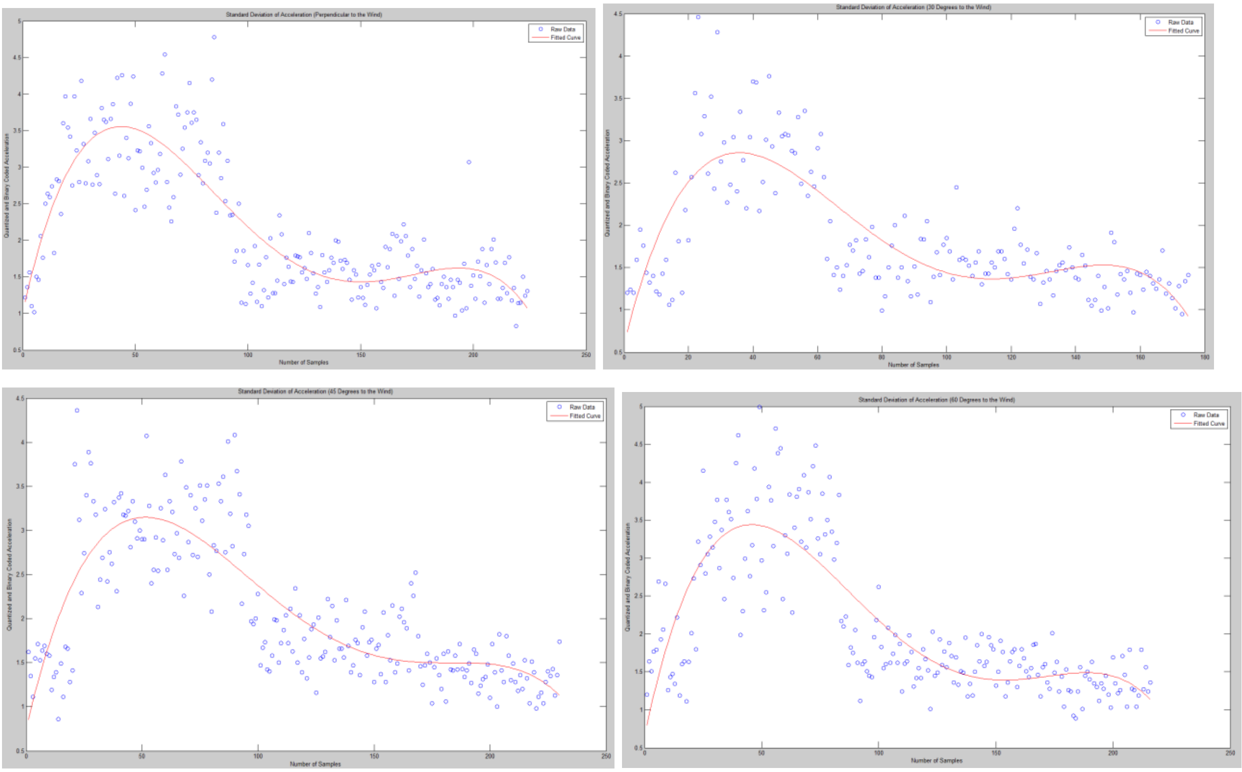
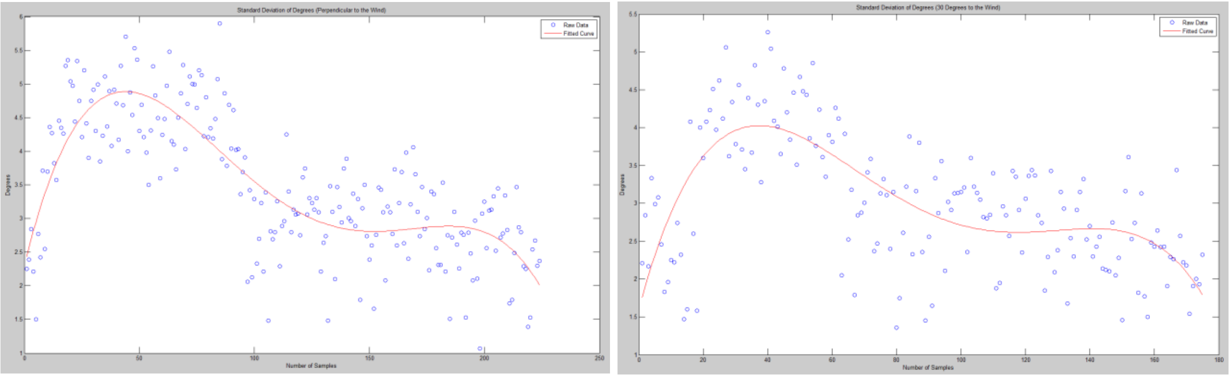
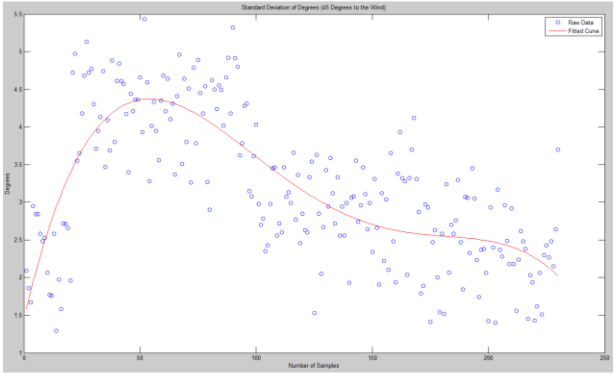
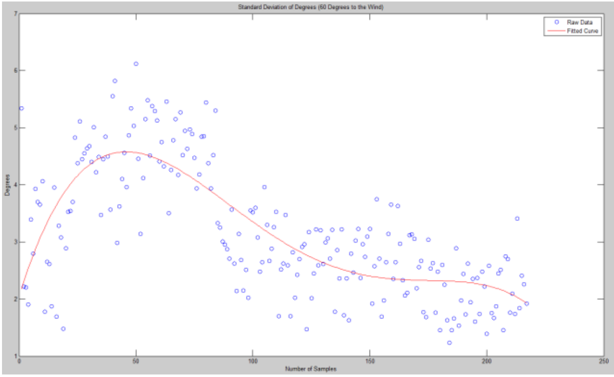


Figure set 2: Standard Deviation of angle







3. Water Test

To test the enclosure box to see if its water resistance we will place the enclosure outside in a rain storm. We will fill the enclosure with a water absorbent material. The enclosure was left out in a rain storm for 3 hours. Upon completion of the three hours the material was taken out. The material showed no signs of being exposed to moisture.

4. Cell Signal Reliability

To test the cell signal reliability we will measure the strength of the signal available at various locations. We want to find how reliable the signal available to the GPRS shield is. To test the signal the command "AT+CSQ?" was issued. The command returns a numerical value representative of the cell strength. The worst signal received was 25 which is still a very reliable signal. Overall the signal was consistently a strong signal available for use.

5. Accuracy of the Tilt Sensor and Calibration.

To accurately test the sensor we will manually perform adjustments to the stop sign tilting to 15 degrees, 30 degrees, 45, degrees, and knocking the sign down. We want to determine the reaction of the sensors and how it will detect changes to the sign.

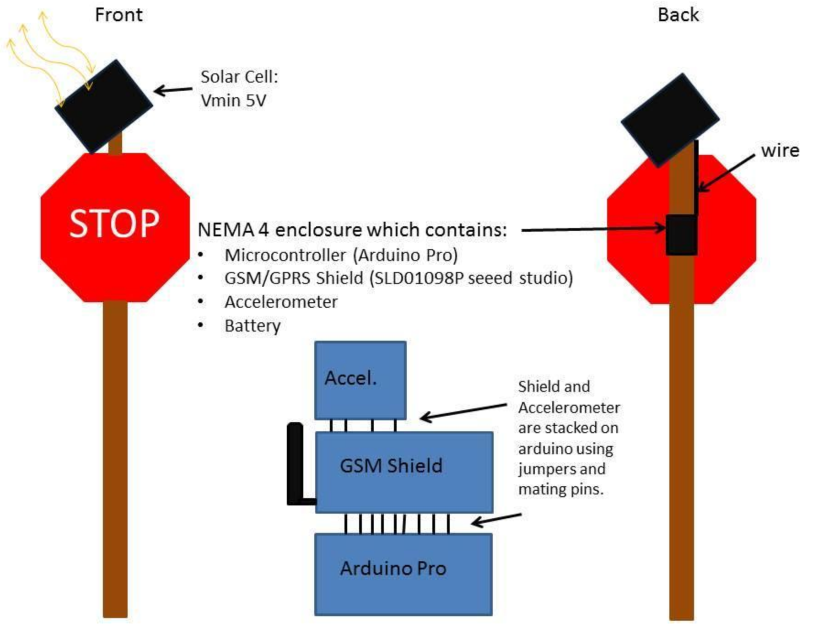
6. Longevity of Power System

To test the longevity of the power system we will run the GSM and Arduino to max output to see how they function. The power supply can be tested to by using fluorescent light instead of sunlight to determine the max energy that is absorbed. The expected battery life is three days. To test the battery life we placed the system into shortest sleep cycle without the solar cell connected. We were able to verify that the battery was able to last the calculated three days.

V. Schematics/Models

Found below is sketch of how we envision this project attaching to the sign as well as interfacing with each other. In addition to the sketch logic flow has been provided based on our final design of this project. For each function where the GSM/GPRS module would be used, the AT command to accomplish the desired task has been provided.

1. Schematic Drawing

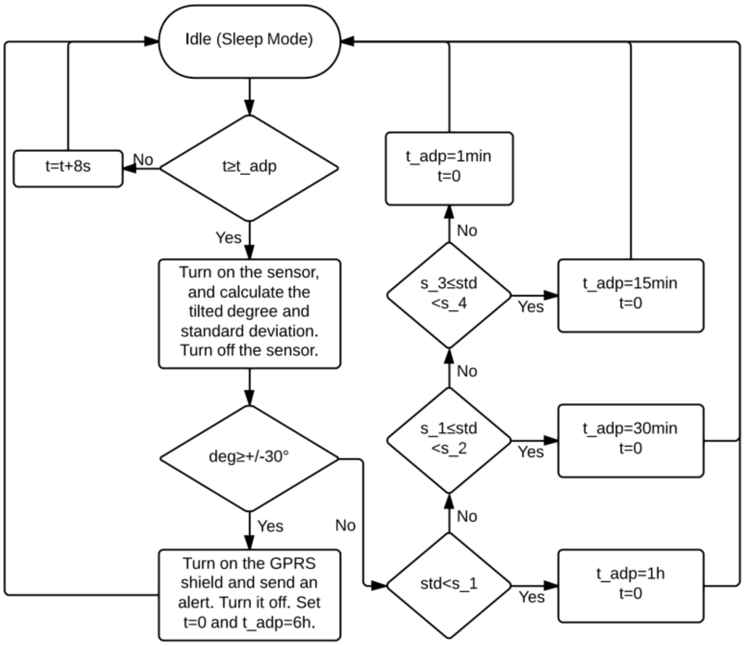


2. Embedded Software

In order to interface with the accelerometer and the GPRS shield, an embedded program had to be written and compiled on the Arduino. The program read the information from the accelerometer and calculated the standard deviation. If the total tilt of the stop sign exceeded +/- 30 degrees a text message was sent.

The standard deviation calculation was used in the adaptive sleep cycle. The adaptive sleep cycle changes based on the estimated weather conditions which are included in the code. We have set the standard deviation benchmarks based on the wind testing conducted. The longest the system will sleep without reading information from the accelerometer is 6 hour. The shortest it will sleep is 1 minute.

The highest wind speed in Iowa is 30 miles/hour with an average lifespan of 30 minutes, which produces a typical path length of about 15 miles. This should be the judgment of the shortest sleep time. For a given mile along the path, the wind would be within the mile for an average of 2 minutes. So a sleep cycle of 1 minute should be good to detect the wind. Other values of sleep time are gradually increased from the shortest time. On the other hand, if a message has been sent out, the sleep time will be set to 6 hours in order to wait for fixing up. Otherwise, a reminder message will be sent out after 6 hours. The flow diagram of the program can be seen below.



3. Useful AT Commands

AT+CMGR --> Read SMS Message

AT+CMGW --> Write SMS Message to Memory

AT+CMSS --> Send SMS Message from Storage

AT+CSAS --> Save SMS Settings //if we change settings it would be important to save them

VI. Part List

- Solar Panel - $25.00
- Lithium Ion Battery Rechargeable $50.00
- GPRS Shield V2.0 (SIM900) - $59.90
- ADXL345 Accelerometer - $27.95
- Arduino Pro 328 (ATmega328) - $14.95
- SIM Card and Plan $10 AT&T Plan 200 Texts - $4.99
- NEMA Enclosure Polycarbonate Enclosure (NEMA 4x), Lift-Off Cover, 3.1x 3.2 x 2.6 Inches, Gray - $30.77

The total cost of the project is $223.44.

Appendix I: Operation Manual

Initial Steps to Start up Sets:

You will need to start a texting message plan with your local phone provider.

Have a stop sign ready for installing the device.

Text Message Manual:

You will receive text message once there is a stop sign tilted. The content of coming message has location and degrees of tilt. There is also a link at the end of content. You can click it to view where the tilted stop sign is on the Google map.

Android Application Manual:

- How to install user interface:
    - Step 1: check the box of unknown sources from you cell phone setting -> security-> unknown resources.
    - Step 2: download the apk setup package from our senior design website.
    - Step 3: click “install” button once download is done. It will install automatically on your cell phone.
- How to view alerts and map:

Our own android application is an easy way to view all alerts received by our client' cell phone. It can filter all alerts from many messages in cell phone's inbox. It can also show all issued stop signs on Google map at the same time, which is convenient to monitor the status of stop signs and collect data from cell phone.

- Step 1: Click “View alerts” button, then it will jump to another page that lists all issued stop signs including location directly.
- Step 2: Click “View on the map” button, then it will jump to a page where a Google map demonstrates markers of all issued stop signs.

Programming System:

For the system to work you need to compile the code onto the Arduino. The following steps are written starting from installing the Arduino environment. If you already have the Arduino environment on your PC you may skip to step 2.

- Step 1: Download the Arduino IDE installer found here: http://arduino.cc/en/main/software. (Note: Be sure to select correct operating system.)
- Step 2: Once the environment is successfully installed launch the application.
- Step 3: Once the application is running you need to select the correct board and com port which the software will be programmed onto. The board is the Arduino pro.
    - You can change the board by selecting Tools>board>Arduino pro (5V, 16MHz).
    - You can find which serial port your board is connected to by navigating to Control Panel>Device Manager>Ports (COM & LPT). The device should be listed there.
    - To change the serial port in the Arduino environment select Tools>Serial Port>COMXX.
- Step 4: Copy and paste provided embedded program into sketch window.
- Step 5: Click upload button. (The upload button is a right arrow)

Hardware Connections:

This section is to provide instructions how put all of the parts together. This section will be broken into four parts. They will be attaching GPRS shield to Arduino, attaching accelerometer to GPRS shield, attaching charging circuit to peripherals.

1. Attaching GPRS Shield to Arduino:

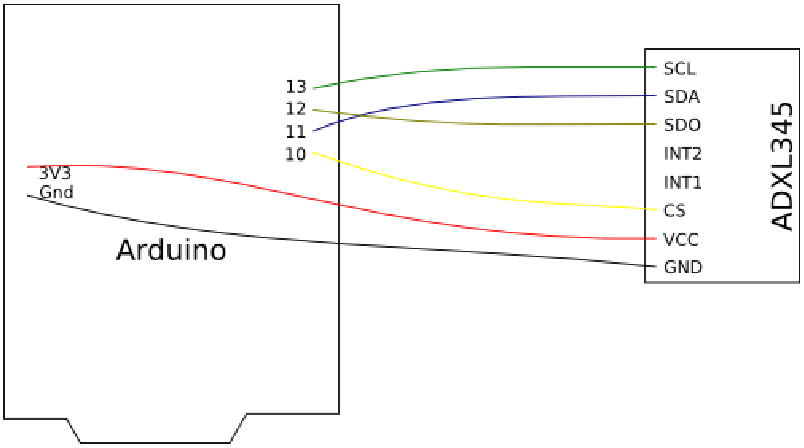
- Step 1: Solder header pins provided with GPRS shield through the holes on the GPRS shield. The female side of the pins should be located on the side of the board that the components are located, i.e. headphone jack, antenna.
- Step 2: Solder Arduino header pins to Arduino through designated holes. The female side of the pins should be located on side of the Arduino board where there is printed text visible.

Hint: header pins for GPRS shield have longer a much longer pin component than Arduino pins.

- Step 3: Place the SIM card into the SIM card holder located on the bottom side of the GPRS shield. *Note: Metal contacts of the SIM card face the board.*
- Step 4: Stack GPRS shield on top of the Arduino. The male pins of the GPRS shield should fit into the female sockets of the Arduino.

2. Attaching Accelerometer to Arduino

Solder wires according to the following diagram.



3. Attaching Charging Circuit to Peripherals

(1) There are three things that need to be attached to the charging circuit with wires. The wires should have approximately ¼ inch of insulation stripped away. The three things attached are the Arduino, battery, and solar panel.

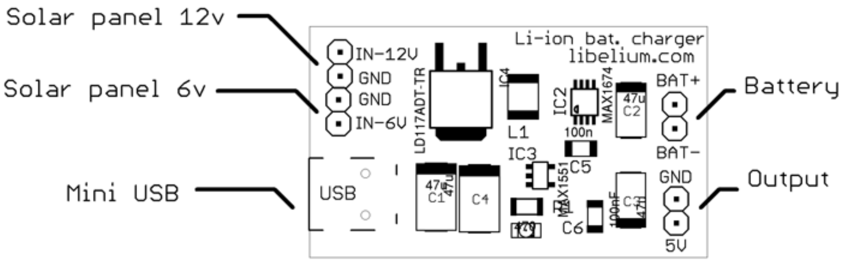
(2) The wires connected to the solar panel should be soldered to the exposed leads and then heat shrinks to protect from weather elements.

(3) The wires connected to the Arduino should be soldered to the holes that are labeled battery found directly behind the micro USB port.

(4) To connect the leads of the devices to the charging circuit do the following:

- Step 1: loosen all set screws on wire ports. (Wire ports are green and have flat head screw on top)
- Step 2: place red wire + port and black wire into - or GND port.

Note: +/- of battery go to ports labeled +/- bat, + of Arduino goes to port labeled 5v and - goes to GND, + of solar panel goes to 12v and - goes to gnd. Screen shot is shown below.



Start-up:

Before startup, make sure you have correctly programmed the Arduino, connected all hardware, and given the battery a charge. To charge the battery simply plug a mini USB cable into the charging circuit with the battery connected for at least an hour. To start the system, do the following:

- Step 1: Place system in a vertical position.
- Step 2: Flip switch on Arduino from EXT position to BATT position. The switch is found on the side with 12 pins.

Maintenance:

Maintenance for the system is minimal. If a part is discovered to be operating abnormally it is advised to simply replace the part on the device. This is the most cost effective solution as many of the parts are inexpensive.

Appendix II: Alternative versions of the Design

At the start of this project we knew this would be the first system to specifically track the function of critical traffic/warning signs. Currently there are traffic signs equipped with solar cells but for the purpose of powering lights for increased visibility. There are also various tracking systems available but are used as geophone detection for higher profile equipment such as mobile stop lights for construction sites. From there we looked at several different options for each part in the system.

Tilt Sensor

An accelerometer is being used as our tilt sensor to measure the stop sign's tilt. When the stop sign is tilted, the sensor will be able to know how many degrees the sign is tilted by. If the range exceeds ±15 degrees, an alert should be sent out. Typically, an accelerometer is not too expensive compared to other components. It can be just connected to the header of the board of the microcontroller. After comparing with other choices, we decided to use ADXL345 accelerometer. We compared prices and functionality to gyroscopes and a basic tilt sensor.

A gyroscope alone is not accurate because of drifting. Gyroscope doesn't always return to the initial position which can sometimes change the data. Most gyroscopes are coupled with accelerometers to increase accuracy. This strategy is significantly more expensive than just simply using an accelerometer.

We also looked at a position sensor. The qualities in comparison to accelerometer measured the same properties under the same conditions. It also comes already pre-packaged (No exposed boards) but we were unsure of how to interface with microcontroller.

GSM module

The GSM module will be alert driven and only a message is sent out if an event occurs. The message should contain the position of the stop sign and also specify if it is tilted by certain degrees. The GPRS Shield V2.0 will work in the temperature range demanded from 115 to -35 degrees F and has a good interface for the Arduino Pro 328. The price is the lowest we that found acceptable. Other options we thought about using were RFID tags and Wi-Fi hotspots. With further research these options didn't fit our project goals.

Microcontroller

A microcontroller is necessary to process the data from sensors and control the GSM module to send out alerts when events happen. The ATmega328 has an excellent operating temperature range and is able to control the whole system. Arduino Pro 328 which uses ATmega328 has a very cheap price compared to other boards. We compared the Arduino Pro to Raspberry Pi and Pic microcontroller.

Arduino: Micro and Leonardo both used the same microcontroller (ATmega32u4).The micro would be advantageous for this project due to its reduced size. Both boards are the same price and could potentially save ~$2 by ordering board without headers.

The PIC microcontroller is $10 cheaper than the Arduino. It is unknown if the GSM module and accelerometer will interface with the PIC. The PIC requires 12V power compared to 5V for an Arduino.

The Gumstix was too expensive. The entry level board we found costs $100.

Power Supply

A power supply is needed for the whole system. Our idea is to use a rechargeable battery plus a solar panel. We also use a charging circuit to connect the solar panel with the battery. Other options included wind charging system and supercapacitor.

A supercapacitor is easy to use and a complicated charging circuit is not needed (lowering down the total size of the device). It also is quicker to recharge (several minutes to full recharge), much longer lifetime (rapid charging and discharging is allowed). And works very well under extreme temperatures but more expensive than a comparable battery.