Commuter Tracking Sensor Network

"A wireless sensor net for outdoor tracking and localization."

Project Principles:

Seth Hendrick
Alessandro Sarra
Jared Mistretta

Project Collaborators:

Dr. Jeffrey Wagner - mjwgse@rit.edu

Submission Date:

May 20, 2014

Table of Contents

Overview

Needs - Quantifying a dollar value for a public good or a public trail, like any public land, can be a difficult task. As the product is being paid for via third party channels, (e.g., state and local taxes), the utility gained by the actual user (who may have not paid any taxes for the good) of the trail cannot be directly correlated with the taxes paid. The actual usage of the trail needs to be tracked to gain a better understanding of the trail value in terms of usage.

Objective - The objective of the project is to capture the usage data for the Lehigh Valley Trail. These data include the mode of transportation and the entry and exit points commuters use. The data will then be accessible via a web interface to a limited set of users.

Description - A wireless sensor network that will utilize computer vision and mesh networking will be used to track the usage of the Lehigh Valley trail. The network will be comprised of a series of modules that use image sensors to recognize commuters and their modes of transportation. Additionally these modules will communicate with one another to determine what are the entry and exit points commuters are using to access the trail. The network will have a gateway node that allows the data to be backhauled to the internet. These data will then be stored via cloud storage and accessible via a web interface.

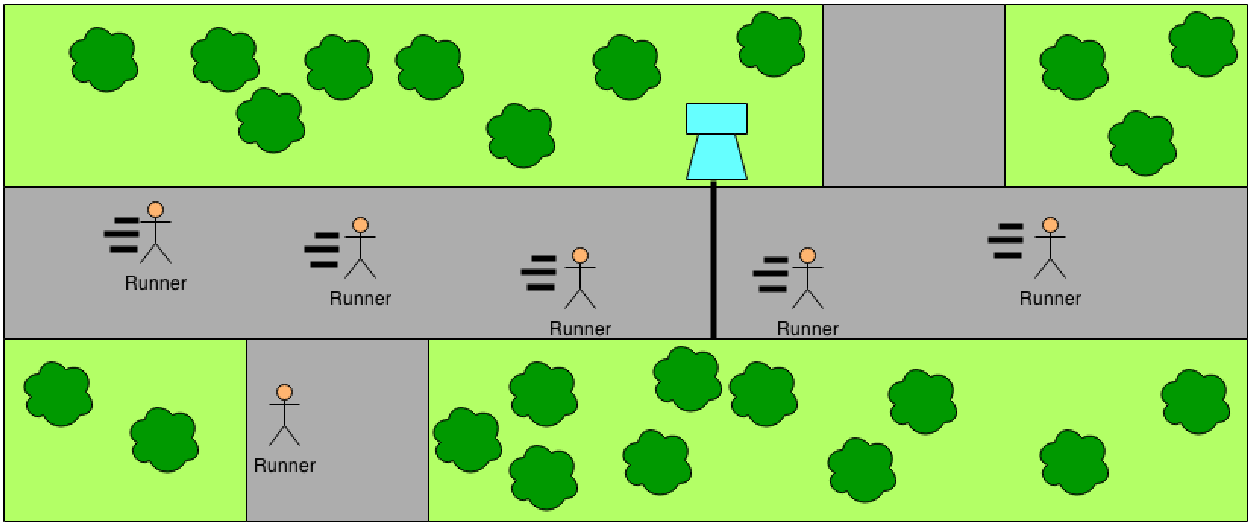

Figure 1 - Marketing Diagram

Figure 1 is a marketing diagram that targets the average user. It shows several subjects passing in front of the blue sensor node located on the Lehigh Valley trail. The sensor node is shown detecting the size of the object passing in front of it, as well as the direction the runner is travelling. The node is also located near two trail entrances, which make it possible to estimate which access points are used to enter the trail.

Requirements Specs -

Customer Needs (Marketing Requirements) -

1. Modules will detect commuters and the path they are taking.
2. Modules will determine the mode of transportation used by the commuter.
3. Modules will be deployed for a week of time without need for maintenance.
4. Trail modules will be able to intercommunicate via a network.
5. Data gathered by the network will be stored via a cloud solution.
6. Data will be accessible via a website interface to a limited set of users, including stakeholders and developers.
7. The website will display the status of each trail node, and provide basic control of the nodes.

Engineering Specs -

Table 1 - Engineering Specifications

Spec #	Marketing Reqs.	Engineering Specification	Justification
1	1,2	The image sensor must be able to capture the image.	The image is needed so that CV can be performed.
2	1,2	The image sensor must be able to perform in a variety of light intensities and directions including overcast weather, dusk and dawn visibility, and midday brightness levels.	The image sensor will be outside and exposed to various weather conditions and light levels throughout the day.
3	3	The image sensor must draw a small amount of power relative to other image sensors on the market, which is 50 µA in standby and 110 mW in active mode on average.	The image sensor will be running on a battery, and must draw a minimal amount of current to maintain charge for prolonged periods.
4	1,2	The lens responsible for image acquisition must have a field of view that creates a consistent image size across the various positionings that the modules will have relative to the image subjects.	The image subjects will vary in size and must produce the same size image for increased consistency and accuracy in CV algorithm output.

5	1,2	The processor must be able to acquire images from the sensor and perform CV algorithms on the acquired images.	The commuter’s mode of transportation must be determined.
6	3	The processor must draw a relatively small amount of power, which is 140mA on average while performing CV algorithms.	The processor will be running on a battery, and must draw a small amount of current to maintain battery charge for prolonged periods.
7	4	The trail nodes must communicate on a band that is allowable by the FCC.	The project must adhere to IEEE ethical standards, and therefore must be legal in all regards.
8	4	The trail nodes must have a range of at least 3 miles.	The greatest distance between any two nodes is approximately 1.75 miles assuming line of sight is unobstructed.
9	5,6	A module must provide gateway services that will link the network to a database service.	The data from the network must be transferred to a database, so it can be accessed through the Internet.
10	5,6	The cloud storage solution must have enough memory to hold all of the data set that is collected.	The data acquired must not be lost due to insufficient space.
11	6	The data from the database must be accessible via a standard web browser for both mobile and desktop devices.	The data must be easily accessible.
12	6	The data from the database must be viewable in a variety of graphical representations.	The data must be easily readable.
13	6	Web site access must be limited to a select user base.	The data should not be viewed by the public due to privacy concerns.
14	3	The trail nodes should enter a low-power state in the evening.	The image sensors will not be able to capture images in the dark, and all power-saving

			options must be considered.
15	3	The trail nodes must be able to operate in the weather conditions of Rochester, NY.	Replacement of nodes would be costly, and loss of data would also result.
16	3	The modules should be hidden from view when deployed in the field.	Strategically placing the modules will deter theft.
17	3	The trail modules' communication method should draw approximately 120mA while active and 30mA while on standby.	The trail modules are battery operated, and must draw a minimal amount of current in order to maintain charge for prolonged periods.
18	3	A sustainable energy source shall be provided to each trail module.	The module's battery source is recharged, and will last longer.
19	4	Modules will use a wireless mesh protocol.	A mesh protocol will provide the needed flexibility.
20	6	The server that hosts the data must be secure to prevent any unauthorized access.	The server's data must be private, and cannot be tampered with from outside forces
21	4	Mesh protocol must have encryption services.	Needed to maintain security of data.
22	6	Server must have an uptime of 95%. This means that protections from DDOS and similar attacks must be built in.	With no server, the data is not accessible to stakeholders.
23	6	There should be a "Status" page hosted somewhere other than where the main server is to notify the user if the server is down, or if maintenance is going on.	Users (and developers) should be kept in the loop if the service is down, and why.
24	4	The network will also be configured so that nodes have a common operating picture.	Information will be made redundant and synchronization of data will be maintained between all nodes and the server.

25	3	The sensor nodes should not provide an environmental threat of any nature.	It is plausible for the battery unit within the sensor nodes to plate, potentially causing a hazardous situation. The enclosure should prevent an internal hazard from escaping to immediate surroundings.
26	7	The website user interface should allow an admin user to power-cycle a node.	Performing a manual power-cycle on a node is time consuming, as the user must travel to the trail, walk to the node and reset it.
27	7	The admins should be alerted through the web interface or email if a node requires maintenance.	If notifications of the node's status are automatic, it prevents maintainers from walking out to the trail every day and ensuring all the nodes still work.

Concept Selection

While preliminary research has shown that there are several systems with wireless sensor nets that use video for tracking and localization, one all-inclusive solution that performs this task outdoors has yet to be found. Additionally none of the systems that were researched offer the breadth of services that are being proposed: on board CV, low power operation, diverse deployment environments and configurations, and a web interface for accessing data. None of the systems utilize the trickle power mechanism that is being considered. It is believed that the use of a windbelt for trickle charging in this sort of application is a novel use of the technology, and would serve as an identifying factor for the project, allowing it to stand out among others.

Table 2 - Concept Selection Chart

Problem	Solution concept
Detect Commuter	1. Computer Vision - Selected 2. Pressure Sensor 3. Laser Sensor
Module Networking	1. Wireless Mesh - Selected 2. Wi-Fi 3. SDR Ham
Power Charging	1. WindBelt - Selected 2. Solar 3. Turbine 4. None
Data storage and access	1. Cloud Solution 2. Custom web server - Selected 3. Local storage and access
Gateway	1. Raspberry Pi and Ethernet - Selected 2. Wi-Fi 3. 3G/4G 4. Proprietary Radio

CV was selected as the method to gather the data as it is considered the most versatile method to accomplish our tasks. Multiple systems would be required to gather the same data if other technology was used.

Wireless mesh technology was also chosen for its versatility and adaptability. Regular IP networking does not apply well to the wireless arena. By using a mesh protocol there is no need to preconfigure the network topology. It is desired that the user be able to apply the modules and technology being developing for a variety of applications, not just our specific use case.

The windbelt was chosen for its compact design and high efficiency when compared to other systems of this size. Solar charging may be added to the system if the windbelt proves inadequate to meet our power needs. Experimentation will provide an adequate amount of data needed to make these decisions.

Signal Acquisition Method

Power must be provided to the MCU and connected peripherals for an extended period of time. For this reason, use of a sustainable energy resource is required to keep nodes deployed on the trail for longer periods. If this is the case, the need for manual recharge of the connected energy storage device will occur fewer times over a given period. This is ideal, since manually charging the storage devices would require traveling to each node, disconnecting the batteries, bringing them to the base station and connecting them to one charger, which would have the ability to charge only one battery at a time. If this were the case, it would require extended periods of time with the nodes not being deployed and not collecting valuable data for that time period. On all accounts, the project would be halted for that duration. Several sustainable energy sources were researched for their various positives and negatives. It was determined that the energy source should be efficient, converting a reasonable amount of AC current from a small amount of natural energy, that the source should be unique in nature, that it should provide an adequate amount of energy over time to the storage solution, that it should be relatively low-profile so as not to attract the attention of passersby, and that it should be of relatively low cost.

Table 3 - Windbelt Alternatives Pugh Chart

Alternatives	Design 1 Windbelt	Design 2 Solar	Design 3 Turbine	Design 4 None
Efficient	5	5	3	1
Unique	5	3	3	1
Adequate	3	5	4	1
Low-Profile	5	3	1	5
Cost	5	3	1	5
Sum	**23**	**19**	**12**	**13**

Table 3 shows the four different sources that were considered, including windbelt energy, solar energy, turbine, and not having a sustainable solution. The windbelt solution was chosen because it was rated highly in all required areas except adequate current. The smaller current supplied by the windbelt was considered a reasonable downside to using the technology. When compared to other sources, such as solar, it is less costly and lower profile. Turbines would drive up the cost, and make low-profile implementation next to impossible, and not having a sustainable solution would be a less costly yet time-consuming alternative.

Node Communication Method

Each node must be able to communicate with other nodes over the entire length of the trail. Figure 2 shows the length of the trail along with the potential node positions.

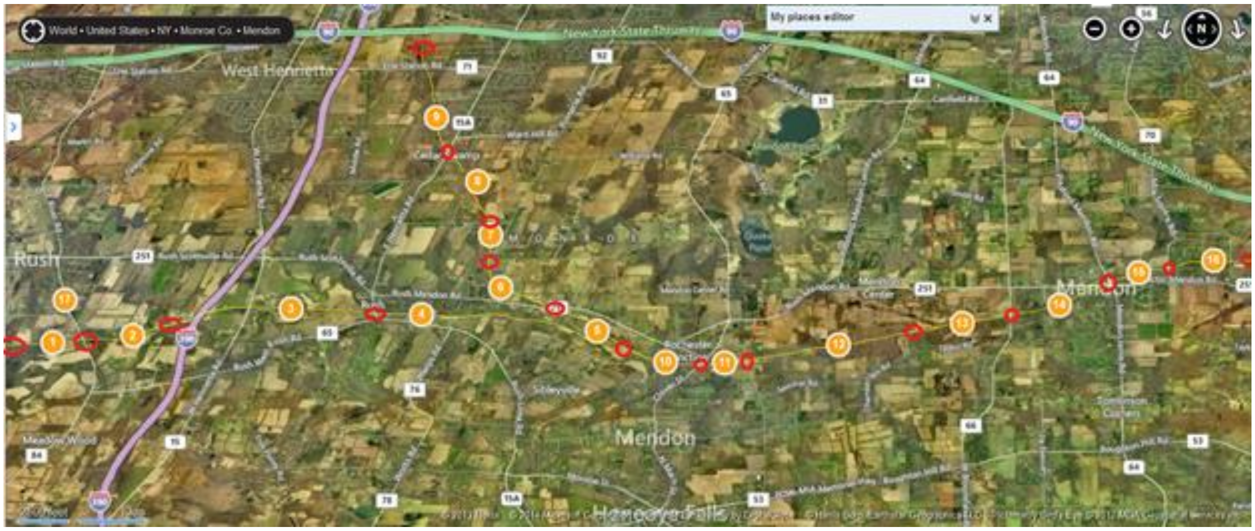

Figure 2: Approximate Trail Node Locations and Trail Entry Points

Each numbered orange circle in Figure 2 is the approximate location of where a node should go. The hollow red circles are trail access points. There is one node in between each access point. The orange circle labeled "17" is Jared's house, and that is where the gateway node will be. For scale, the trail left-to-right stretches about 15 miles, and the vertical distance between the top-most and bottom-most nodes (9 and 11) is roughly 4 miles.

The nodes must be able to send data to each other and to the database. The greatest estimated distance between two nodes is roughly 1.75 miles. Therefore, the selected solution must be able to communicate data at a distance of at least 2 miles, although the greater the distance the better. There were three main methods considered: a giant directional antenna, SMS communication, and ZigBee radio modules.

The first idea considered was to deploy a giant Wi-Fi antenna at Jared's house (position 17 in Figure 2), which is only half a mile from the south-west end of the trail. By pointing 1 or 2 antennas in the direction of the trail, most, if not all, of the trail can be covered. By going with the directional antennas, it means that there is a centralized node at the antenna location that is collecting data from each of the trail nodes, and syncing that data with all the other trail nodes, along with the database. Some of the antennas researched could reach distances of over 28 miles, and were cheaper than buying 17 ZigBee modules. However, upon further investigation, the legality of deploying such devices was unclear. Most directional antennas are designed for point-to-point communications; that is communicating with one and only one other point. For the purposes of the sensor network, the antennas need to do point-to-multipoint communication, or communicate with more than one node at a time. Using antennas designed for point-to-point communication for point-to-multipoint can be illegal depending on the antenna. In addition to the legality issue, the trail is not straight, and therefore a directional antenna might not fan out enough to cover the entire trail. An omnidirectional antenna, or an antenna that goes in all directions, could be used to get a good fan out, but no omnidirectional antennas were found that had a long enough range. The final problem with using a directional antenna is that each of the nodes still need some kind of Wi-Fi dongle. USB Wi-Fi dongles are less than $10, which are cheaper than buying 17 ZigBee modules. However, the MCU may not be compatible with the USB Wi-Fi dongles. Due to the questionable legality, and the fact that the fan-out of the antenna is unknown, this idea was rejected.

Another plan to have the nodes communicate with each other was to connect GSM chips to each node and have the nodes communicate with each other through SMS. GSM chips are designed for low-powered mobile devices, so the power footprint would probably be small. With SMS, the range between each of the nodes is no longer a problem, as they can be deployed anywhere, and still be able to send a text message, assuming a phone signal. Although the GSM chips are cheap, being $10 to $20 apiece, sending SMS messages requires a carrier such as AT&T or T-Mobile, and they are not free. Eventually, sending SMS messages will cost more than using a ZigBee module. Due to the recurring cost, this idea was rejected.

The chosen solution was to use the Digi International XBee 900 HP radio module. The XBee has built in mesh networking, so no mesh networking software needs to be written. It also has a range of 9 to 28 miles, depending on the antenna. The XBee has a UART interface, which allows it to send data from the MCU. Most importantly, the documentation also clearly states how to use the device legally so the project does not get shut down by the FCC.

Gateway to the internet

The trail nodes need some kind of access to the internet to send its data to a database. One of the concepts considered was to use 3G or 4G attached to one or two of the nodes, and send data to the database over 3G through the cellular network. The good thing about 3G is that range is not a problem. As long as there is a good cell phone signal, data from the trail nodes can go anywhere in the world. The problem with 3G is that it requires a data plan from a wireless carrier, making a recurring cost per month which is less than desirable. Also, 3G seems to use a lot of power, which is a problem since the trail nodes are running on a battery, and power is limited.

Another idea was to purchase a gateway device that would convert the data from the trail nodes to either Ethernet or USB, and connect it to a PC at Jared's house (node 17 in Figure 2). Jared's house has an internet connection which will be the access point to the World Wide Web so the data from the trail nodes can be sent to the database. This idea would make the most sense, but there is a cost problem as gateways would incur an additional expense. There is, however a cheaper alternative.

The choice for the gateway to the internet was to use a Raspberry Pi and an XBee. The XBee will communicate with the trail nodes, and send any data that needs to be written to the database through the Raspberry Pi's UART connection. The Pi has an Ethernet port, so it can send data from the trail directly to the internet. The Pi has an added bonus as well. It can act as both the gateway and the web server at the same time.

Web Server / Cloud storage

The data from the trail nodes needs to be saved to a database somewhere. That data must also be able to be accessible through a web browser via web pages. One idea that was considered was using a third party for the cloud storage of the database and for hosting the web site. There are free hosting services, such as x10Hosting, so cost is not a problem. However, the host might give little control over what can or cannot be installed on the server. Also, additional software needs to be written to send data from the Raspberry Pi to the third party host so the host has the most up-to-date data from the trail. One advantage of having a third party web host is that it can handle a large number of users. However, since heavy traffic is not expected on the site due to the limited number of users that will have access, this advantage does not hold much weight. Another advantage is that a third party site might do automated backups, but this can be easily done on a local server as well.

Another concept would be to use the same Raspberry Pi that will function as a gateway node for the web server to manage the “cloud” storage for the database as well. Using the Pi in this way gives total control over what software can or cannot be installed on the system. The Pi also saves some coding, as there is no need for software to be written that will send the trail node data to some third party host from the Pi; the data from the trail nodes will go directly to the database, which will live in the Pi’s memory. The Pi does come with a few drawbacks however. The first one is that it cannot handle as many users as a third party host. This was not considered an issue due to the low volume of users that will be allowed to access the server. The second downside is if Jared’s house loses power, then the website will go down as well. A “status” webpage page might need to be hosted somewhere else, (e.g., another team member’s house), which will ping the Pi every few minutes and display on the status webpage whether or not the trail data website is down or not.

The selected concept, for now, is to use the Raspberry Pi as the server. The main reason is so there is total control over what is able to be installed on it, and the choices are not limited to whatever the third-party hosting site offers. However, if it turns out the Pi does not have enough computing power or hard drive space, or is insecure to use, then the third-party hosting option will be considered

Design

The commuter tracking system overview is shown in Figure 3, and is composed of 16 nodes established along the length of the Lehigh Valley Trail that communicate traffic information to a gateway node through use of external antennas. The gateway continuously broadcasts the statistical data to a cloud server, where it is stored in an associated database. Users will have access to the traffic information and the sensor net through a web based graphical user interface.

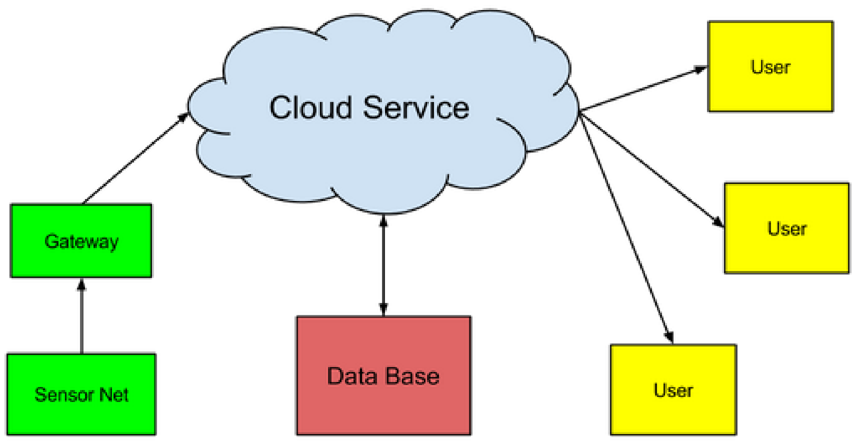

Figure 3 - System Overview

Each sensor node shall meet a low power requirement while providing the ability to capture and process images. The network will also be configured so that nodes have a common operating picture. In this manner, information will be made redundant and synchronization of data will be maintained. The approximate radio frequency range of each node shall be at least 3 miles to accommodate 15 miles of trail with some overlap of neighboring nodes.

Figure 4 shows a hardware block diagram illustrating flow of power and data throughout the system from a high-level perspective. Power is provided via a trickle-charge, sustainable, low output energy resource and fed into a power regulation device, where it is converted from AC to DC. The regulation device will also rectify voltage output prior to supplying power to the image sensor, image processing component, MCU and radio module to ensure they are operating in nominal conditions. A Lithium Polymer battery will be used to store the harvested energy, which is split using a simple voltage regulation device that provides unregulated current to a secondary boost converter, and 3.6V to an XBee radio module. The secondary boost converter will provide a stable current to two Cortex M processors located on the camera module. An image sensor on the same module will provide motion capture information to one of the processors, and the other will oversee the transfer of data to and from a radio module connected to an external antenna. An infrared module will also be connected to the auxiliary processor, allowing the node to operate during low light conditions and waking the image sensor from sleep mode when it senses a commuter.

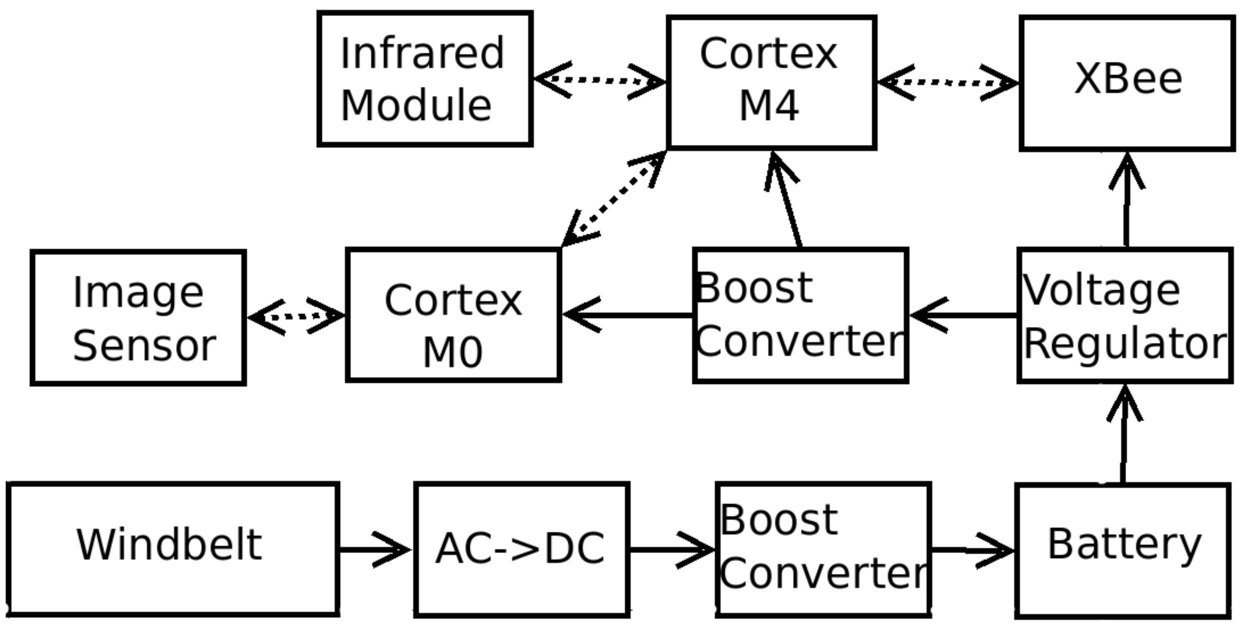

Figure 4 - Hardware Block Diagram

The Power System

A windbelt will be used to trickle-charge a lithium polymer battery due to the inherent need for sensor nodes to be deployed for longer durations with less manual intervention. Table 3 shows the alternative sustainable energy sources that were considered, along with point values for several different categories of need. It was desired that the chosen design provide sustainable energy efficiently and adequately with a low profile and low cost. Utilization of a unique design was also considered to give the project an interesting touch. The windbelt design met requirements more than other choices, which included solar and turbine, and the choice of not having a sustainable energy element. Due to its potential effect on overall sensor network deployment, the windbelt is considered a high-risk design component.

The commuter tracking network is powered using a pre-charged lithium polymer battery capable of providing 3.7 V to the voltage regulation device and then to connected peripherals. The windbelt will produce a small AC current based on the aeroelastic flutter effect to provide a trickle-charge to the battery. This is done for a fraction of the cost that would be incurred if using turbine or solar sources, and can provide exponentially more power for any given wind speed than turbine technology. As shown in Figure 5, this can be done on a specific model windbelt for wind speeds as low as 3.5 m/s, supplying 0.2 mW of AC power across a 390-Ω load at 70 Hz. This power output improves exponentially as wind speed increases, taking full advantage of variability of environmental conditions.

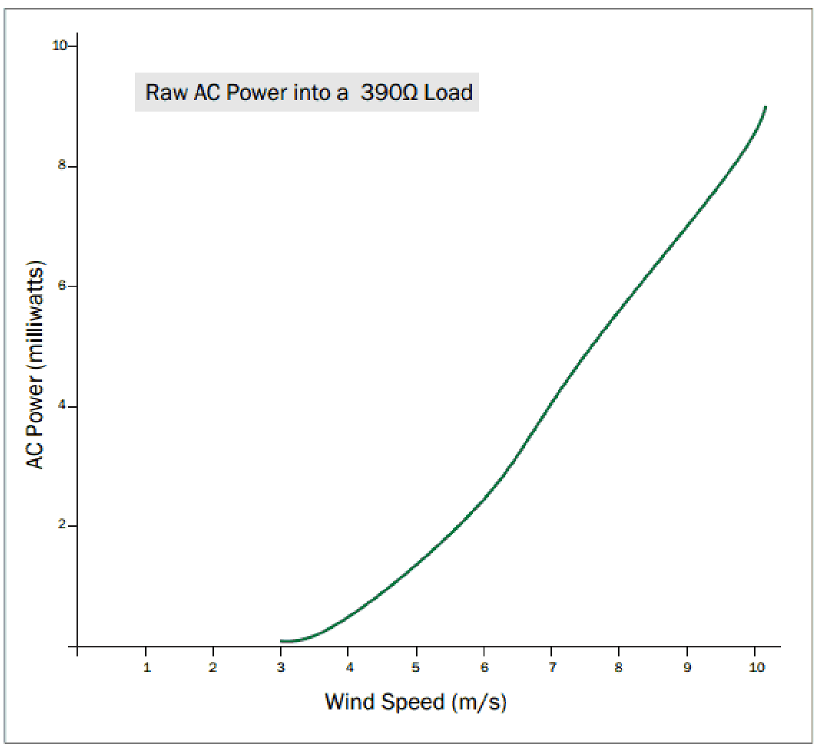

Figure 5 - Wind Speed (m/s) vs. AC Power (mW)

Likewise, power will be provided by the LiPo battery during time periods where wind is not present. The flutter effect produces power from wind by vibrating a band pulled taut across a small opening. The band repeatedly moves a small magnet across one or more coils, inducing current upon connected wires. As an example, the µicroWindbelt shown in Figure 6 is able to power wireless sensors through the attached 3-V DC buffered supply. This model exhibits characteristics similar to those we wish to provide in our own custom implementation of windbelt technology. The design will be developed in an evolutionary manner, generating working prototypes that will be adequately tested and implemented through use of RIT’s 3-D printing facilities.

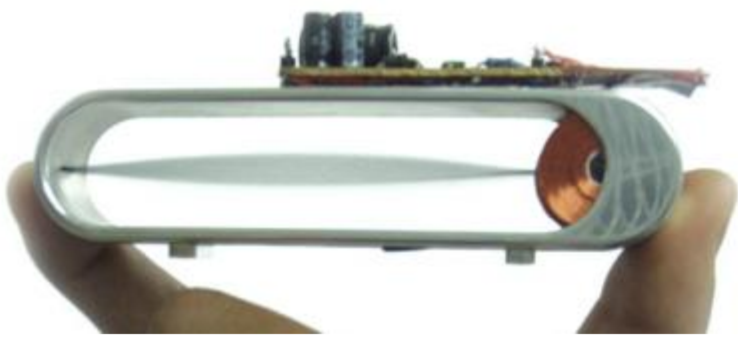

Figure 6 - The Humdinger µicroWindbelt

Dimensions of the design shown in Figure 6 are 13 cm by 3 cm by 2.5 cm. The design was developed by Shawn Fraye of Humdinger LLC, and was specifically derived for use with microcontrollers, providing 0.2 mW at 3.5 m/s, 2.0 mW at 5.5 m/s, and 5.0 mW at 7.5 m/s at 70 Hz. It can provide anywhere from 50-200+ Wh at the listed dimensions. These dimensions may not be the end design that is chosen due to size restrictions on the internal components, but the micro design of that shown in Figure 6 is considered optimal due to its low profile and relative efficiency.

The trickle-charge supplied to the LiPo battery should ideally be of a low enough current to keep a fully charged battery at full capacity while not causing damage to the battery. This would be ensured through use of a rectification circuit positioned after the windbelt that would cut charge of the battery to at most 7 C, or 7 charge hours to full battery capacity. This limit is imposed for safety reasons, since a LiPo battery could potentially overcharge, causing damage to the cells and possibly plating out the metal on the battery. Other safety features of the battery should include reverse polarity protection, a charge temperature limiter that prohibits charge when the battery is lower than and discharge current protection to prohibit reverse flow of current from inhibiting the generation of inductive current.

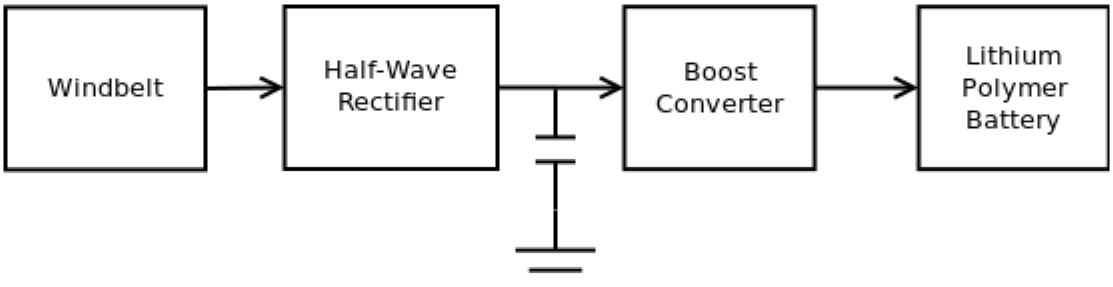

Figure 7 - Signal Conditioning Block Diagram

Figure 7 shows a block diagram of the input signal as it passes from the windbelt to the rectification circuit, which will be composed of simple Schottky diodes. The diodes will function to keep the AC signal from going into negative values, essentially cutting off the lower portion of the waveform and maintaining an adequate DC range for the boost converter to exploit. A capacitor is added after the rectifier to smooth its output, also keeping signal levels from dropping too far between periods. The complete half-wave rectifier along with smoothing capacitor design is shown in Figure 8.

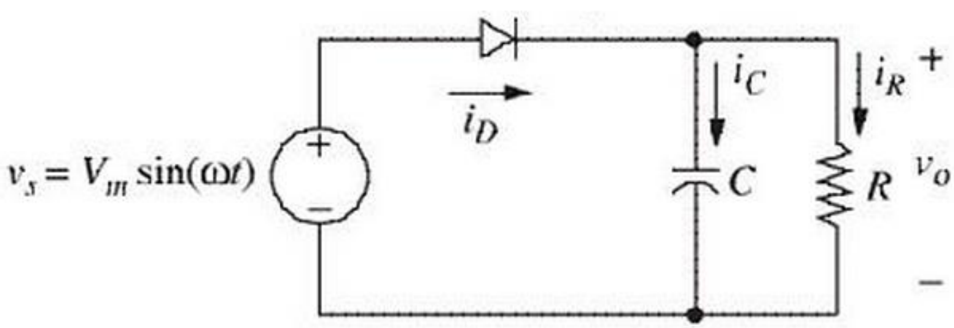

Figure 8 - Half-Wave Rectifier with Smoothing Capacitor

As shown, the current ID will pass from the 70 Hz AC source provided by the windbelt through the diode and be divided into IC and IR. The voltage levels across the resistor are represented by the waveform shown in Figure 9.

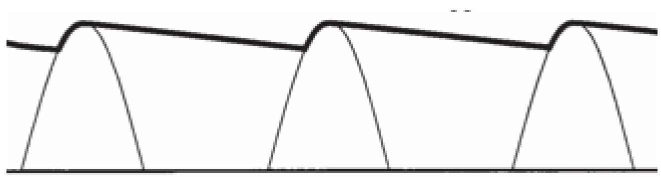

Figure 9 - Waveform Result of Half-Wave Filter

The AC signal is kept at a high enough value for the remainder of the signal conditioning circuit to work effectively. Negative AC values are eliminated due to reverse current protection of the diode, and risk associated with applying reverse current to the battery is also reduced. To further reduce risks associated with providing an unstable source to the battery, a TI bq25504 boost converter will be connected to the output of the rectification circuit via the VIN_DC port. If initial testing shows that the half-wave rectifier is not sufficient in providing a DC source to the converter, a full-wave rectifier alternative can be used. This will result in the inversion of the bottom portion of the AC signal, providing higher signal levels and less average fall-time between periods.

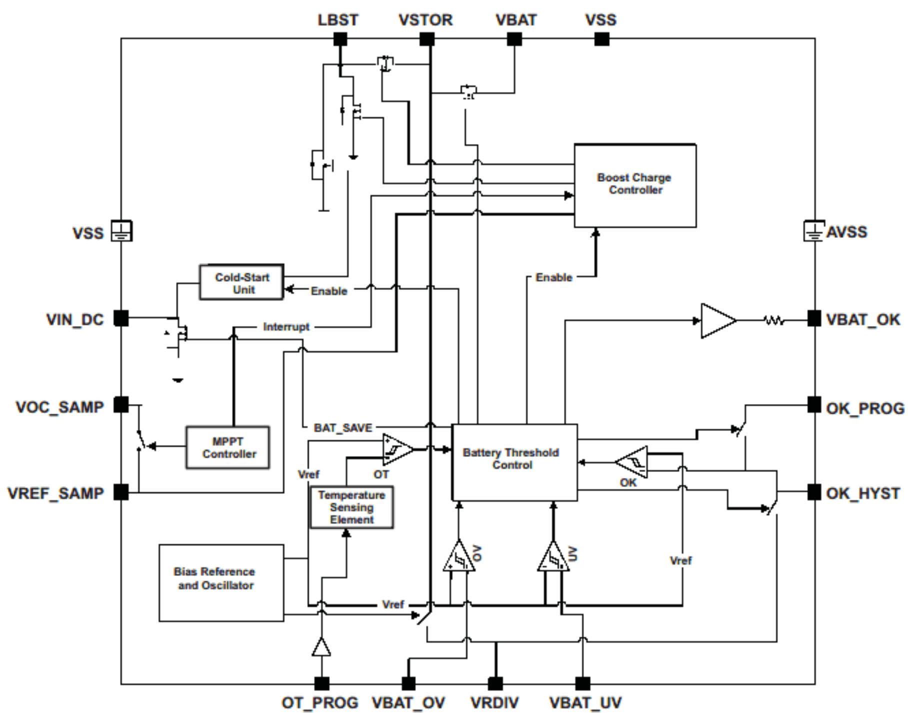

Figure 10 - Texas Instruments bq25504

As shown in Figure 10, the bq25504 boost converter takes the DC signal provided by the rectifier and drives it up to a solid 3.3 V, which can then be used to charge the LiPo battery. The TI device was chosen due to its ultra-intelligent, nano-power design that is ideal for implementation in a wireless sensor node application. It also provides safety features to the LiPo battery such as programmable overvoltage and undervoltage protection, and thermal shutdown protection. The converter has programmable hysteresis features that allow it to retain past voltage values for power level tracking (MPPT), can turn on and remain running with minimal power-on voltage, and can even warn attached microcontrollers of pending loss of power. A low voltage cold-start of 330 mV or higher is needed to start the device, along with a V_{IN} of at least 80 mV to allow it to continue harvesting, allowing the battery to safely charge for longer time periods. When the device is not connected to an analog load and is not amplifying, it consumes only 330 nA.

MPPT, or maximum power point tracking, is a technique used in energy conversion systems that allows for continuous sampling of the output of a power source, and the dynamic application of resistive load that will allow the circuit to obtain maximum power during changes in environmental conditions and variations of input voltage. Internal potentiometers are automatically adjusted to divide the voltage differently when it reaches a certain threshold value. An external reference voltage can also be set by MCU to keep voltage levels at a specified minimum. All threshold values are fully programmable.

The boost converter also has a “battery good” port that can be programmed to monitor voltage of the storage device, and send a signal when it reaches a certain level. If signals exceed a programmable overvoltage or undervoltage threshold, the connected MCU can send a signal to shut off all connected peripherals. As reassurance that the boost converter works with most modern day cell chemistries, Texas Instruments provides documentation confirming compatibility with Lithium Polymer and Lithium Ion batteries. The boost converter is of great importance in an energy acquisition scenario, as changes in the environment, specifically wind speed and temperatures, would otherwise have a much greater effect on the power being provided to the LiPo battery, and the health of the battery as well.

The battery chosen for the design was the Tenergy Li-Polymer 3.7V 10000mAh DGR-A (model number 7872196). This model was chosen for its relatively stable cell-chemistry, large capacity, relatively small dimensions and weight, and larger cycle life and charge rates. The battery is capable of charging at a rate of 10000 mA for 1 C. At 80% charging efficiency at 10A, the battery would reach full capacity in approximately 1 hour and 11 minutes. Reducing the charge rate to 1A at about 50% efficiency increases the time to full charge by about 14 hours. Discharge rates are similar, which means that the method used for charging the battery should be able to keep up with about half a day's worth of battery consumption at about 1A.

The battery can be charged up to 500 times before full capacity begins to degrade. There is a limit to the Dimensions are roughly 20 cm by 7 cm by 1 cm, which can fit a slightly larger windbelt enclosure than the micro enclosure previously specified. Preliminary testing will continue to aid in which decision is made for the final design of both the enclosure and the battery. Currently, the design is leaning toward having a larger battery size and life at the expense of a larger enclosure design. The Tenergy battery will be able to provide a 3.7V nominal signal with a cut-off voltage of 2.75V at 0.2C discharge rate and requires 4.2V or less to accumulate charge. If this signal cannot be obtained by using a windbelt device in preliminary testing, then an alternative sustainable source of energy must be considered.

Image Capture and Processing

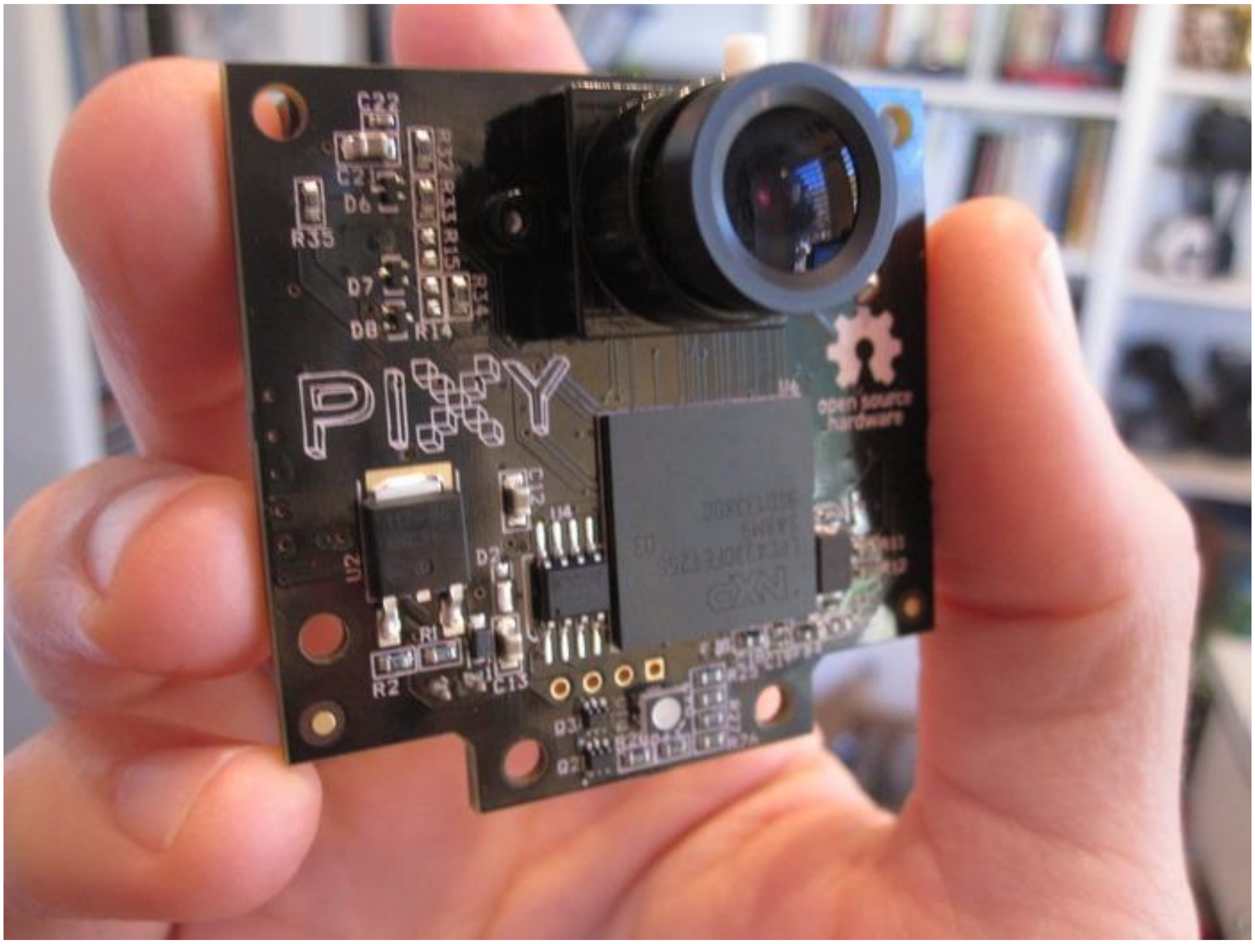

Figure 11 - Pixy Board Image Processing Module

To accomplish the task of capturing and processing images the CMUCam 5, or Pixy, was selected. A picture of the Pixy board is shown in Figure 11. Pixy is a sensor module that includes an image sensor and on board image processing. The Pixy utilizes an Omnivision OV9715 image sensor capable of a 1280x800 pixel resolution. A lens with a 70-degree FOV comes standard with the Pixy. This lens may need to be replaced depending on the specifics of the module deployment. The NXP LPC4330 processor is the onboard MCU that is used to perform the onboard image processing. This powerful dual core processor can handle the image processing without too high of a power consumption. The power consumption for the unit averages 140mA. A diagram of the LPC4330's architecture can be seen in Figure 12.

Figure 12 - LPC43XX Architecture Diagram

The NXP LPC4330 MCU that is onboard the Pixy will be used to control the modules. The processor is a dual core ARM Cortex-M architecture, and includes an M4 and M0 core that can intercommunicate between a common bus system. This processor is fully capable of doing the image processing that is needed and any additional control that may be needed as well. It should simply need to send communications over UART to the XBee modules. The XBee modules operate on their own without any need for control, and the power system also operates without the need for a control system. Since all of the image processing is done onboard, the Pixy simply sends a small amount of statistical data. For example, data that can be send by the Pixy can be the Cartesian coordinates of an object, or how many objects are on the screen. This is accomplished at a rate of 50Hz.

The Pixy also provides a diverse set of communication ports and protocols to interface with a variety of devices, as shown in Figure 13. This will be used to communicate with the XBee modules.

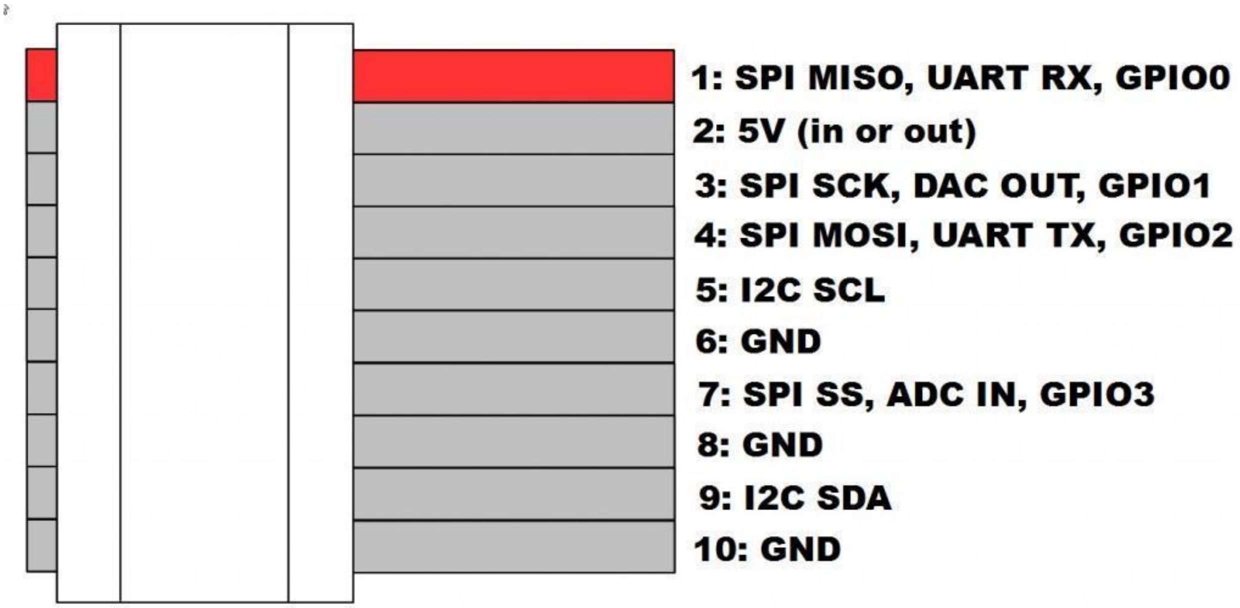

Figure 13 - Pixy Module Pin-out

Sensor data (e.g. XY coordinates, object speed, etc.) will be sent to the XBees via UART. This data will then be routed through the network to the cloud storage solution.

Currently, the Pixy includes software only for color detection. While new facilities are being added to the Pixy constantly we will probably need to write some application specific software. We will need descriptors for a person walking, riding a bike and riding a horse. We should be able to adapt existing open source software to our purposes.

The Pixy is also ideal for our purposes due to its physical specifications. With dimensions of only 2.1” x 2.0” x 1.4” the module can be hidden in an inconspicuous manner. The light weight of 27 grams also aids in the versatility of its deployment.

Network Architecture

ZigBee XBee RF modules will be used to coordinate data among the sensor nodes, utilizing the 900 MHz experimental band along with 2-3 dB gain antennas to intercommunicate. An approximate RF range of 3 miles is necessary to allow overlap of each node with at least two other nodes along the length of the trail.

The XBee name refers to a family of form factor compatible radio modules that utilize the 802.15.4 network protocol. A network of this IEEE standard specifies the physical layer and access control for low-rate LR-WPANs, as shown in Figure 14.

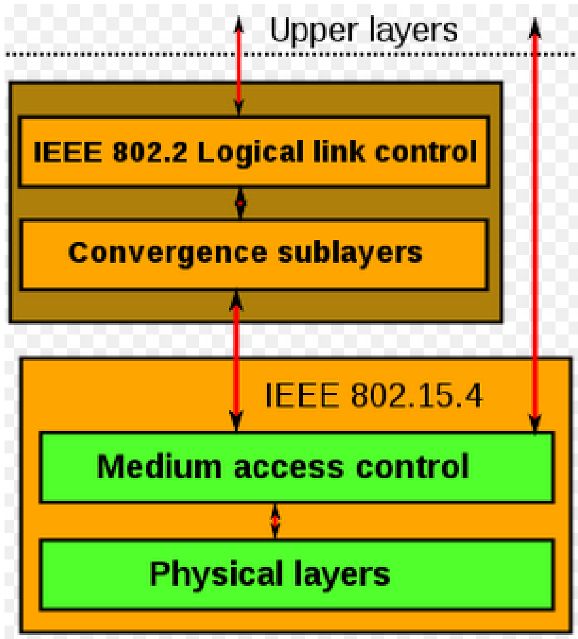

Figure 14 - IEEE 802.15.4 Protocol Stack

The protocol architecture is conceptually simple, based on the OSI model. Only the two layers shown in Figure 14 are defined by the protocol. In North America, the physical layer utilizes the 902 to 928 MHz band, including up to 30 channels. Data rates range from 100 kbps at 868 MHz to 250 kbps at 915 MHz.

The MAC layer allows for network beaconing, controlling frame validation, guaranteeing time slots and handling node associations. Frame size is limited to 127 bytes for most 802.15.4 applications, and fragmentation schemes are available to support larger network layer packets.

The ZigBee protocol makes up the remainder of the stack, adding four main components: the network layer, application layer, ZigBee device objects (ZDOs) and manufacturer-defined application objects. The application objects allow for customization and complete integration of the network. ZigBee allows for low power consumption and secure networking in personal area networks. A small form factor is characteristic of modules that use the protocol, along with the ability to transmit data over long distances while passing data through intermediate devices. This is accomplished while attempting to reach more distant nodes, creating a highly reliable mesh network topology and eliminating the need for a centralized control for certain applications. ZigBee devices utilize 128-bit symmetric encryption for security purposes.

The Digi XBee-Pro 900HP S3B is the specific 900 MHz module that will be used, coupled with a 2.1-dB antenna. The radio can operate with a range of up to 9 miles provided that LOS (line of sight) should not be an issue. This will allow for creation of a true mesh network, since all project nodes are within a 15-mile range limit of each other along the trail. The radio utilizes an ADF7023 transceiver with a Cortex-M3 operating at 2.8 MHz. It can transmit at the 900 MHz experimental band that is desired, and transfers at 10 Kbps worst case range. Speeds of up to 200 Kbps are possible at minimum range. The most efficient transmittable range has been proven to be 4 miles with 2.1-dB dipole antennas. The radio will be connected to the MCU via UART, and operates at 2.1 to 3.6 VDC with a transmit current of 215 mA, a receive current of 29 mA, and a sleep current of 2.5 µA.

Infrared Sensor

The Pixy cam has one weakness. When is powered, it will be consistently be filming the trail, even if no one is walking past. This will waste power as the Pixy cam is active and running image processing on nothing. To help mitigate this, an infrared sensor can be put in place that will “wake up” the Pixy Cam whenever someone walks past the trail.

The infrared sensor that was chosen was the Zilog ZMotion Detection Module (Z8FS040), which is shown below in Figure 15.

Figure 15 - The ZMotion Detection Module

The ZMotion module comes with a variety of pins that can be used. The pinout is below in Figure 16.

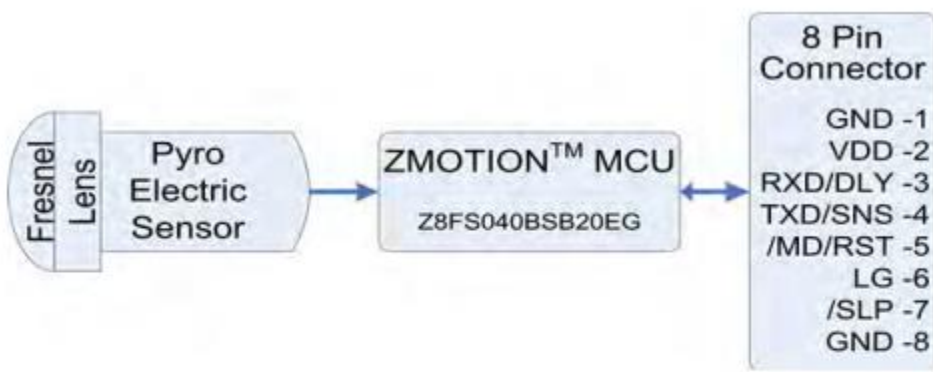

Figure 16 - ZMotion pinout

The ZMotion module can operated in two different modes: hardware interface and serial interface. For the purposes of this project, the ZMotion module will be in hardware interface mode. When in hardware interface mode, pin 5 of the ZMotion module will be set to logic ‘0’ when motion is detected. This pin can be tied to one of the Pixy Cam’s

GPIO ports, and the Pixy Cam will be woken up via an interrupt whenever the logic of the pin is ‘0’. When the Pixy Cam is woken up, it can do image processing before going back to sleep.

Another advantage of the ZMotion is that is has a low-power mode, which is activated whenever pin 7 is low. During this time, the motion sensor will not detect any motion, as it is basically asleep. The pin that controls low power mode can be connected to the XBee’s “module status pin”, which is set to low whenever the XBee is asleep. At night, when the gateway node tells all of the XBees to go to sleep, not only will the XBees go to sleep, but the infrared sensor will also go to sleep. When the infrared sensor goes to sleep, the Pixy Cam will stay asleep. This technique will reduce the total power consumption during the night, when the Pixy cam can’t see anything anyways.

Connecting it all together

Due to the modularity of the design, and a split in voltage levels at the node after the battery in the circuit, it is required that the 3.7V produced by the battery be passed into a secondary boost converter to be ramped up to a level that is acceptable by the Pixy camera module. The second line that is split off of the battery node will be attenuated to provide levels acceptable by the XBee radio. These connections will all be made on a custom PCB like that shown in Figure 17.

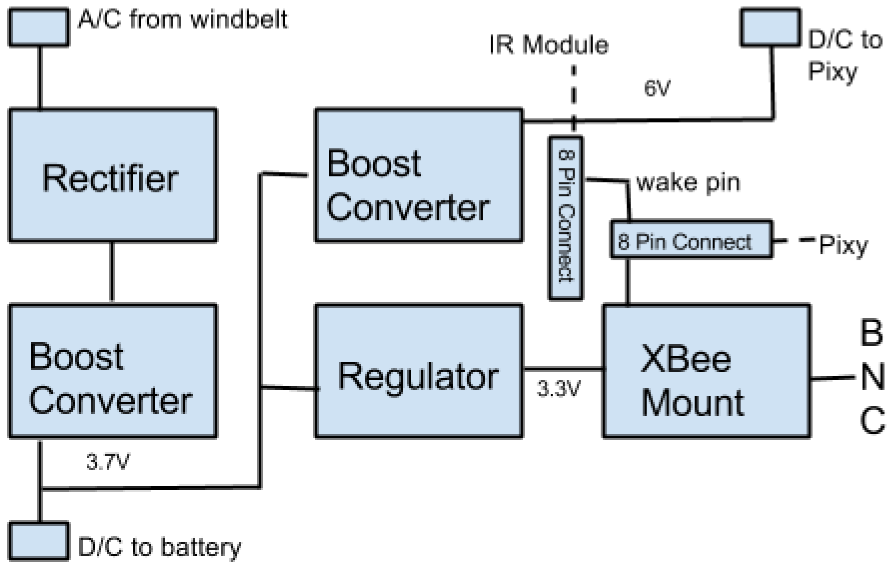

Figure 17 - Custom PCB Design

Other important features on the PCB include an XBee mount that will allow for interfacing 20 pins with the board. The mount will be routed to an 8-pin interface that will be used to attach to a Pixy camera module. Another 8-pin connector will be used to attach the IR module, and a wake on detect pin will be connected to the power pin for both the Pixy camera and the XBee. This will allow for the devices to be in sleep mode during low traffic periods. When a commuter passes by, the infrared module will set the wake pin, turning on the peripherals needed to perform the tracking algorithms. The PCB will also have two pin connectors to attach the windbelt to the rectifier, the boost converter to the battery, and the Pixy camera to the secondary boost converter. A BNC connector will allow for an antenna to be attached to the XBee, and be positioned outside of the node enclosure.

Cloud Server, Database, and Gateway node

The Digi XBee 900 HP will allow communication between each of the trail nodes, but there needs to be some way to save the data to a web-accessible database through some kind of gateway node. This can be achieved by using a Raspberry Pi. The Pi can be used to both read data from the trail nodes, write these data to a database, and then serve web pages using the data from the database to users from the internet. Figure 18 shows a block diagram on how this will work.

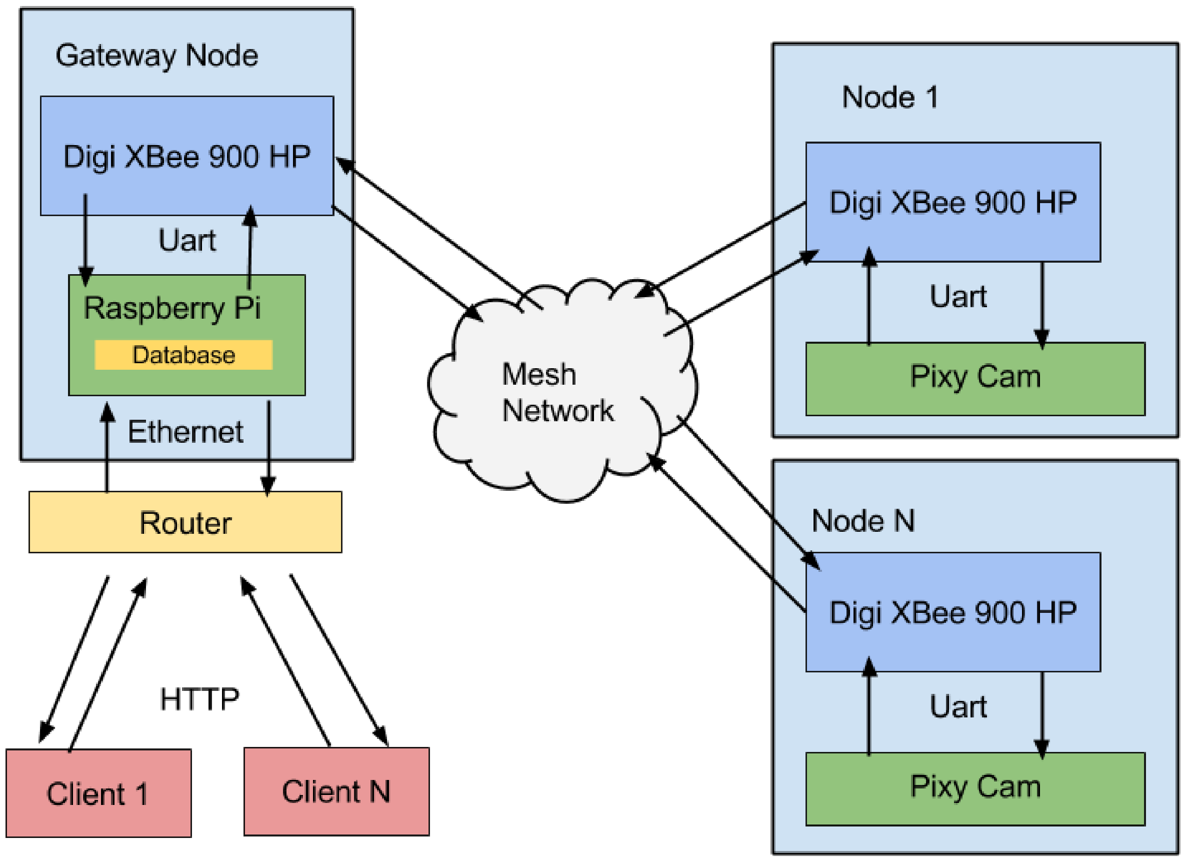

Figure 18 - Network Communication Diagram

Each of the trail nodes will have the MCU connected to the Digi XBee module via UART. The XBees will form a mesh network. Meanwhile, the gateway node will have the Raspberry Pi also connected to a Digi XBee through the Pi's UART interface. If one of the trail nodes needs information to be sent to the database, it will communicate with the Raspberry Pi, which will then write the information to the database that is in the Pi's memory. The Pi will also act as the web server as well. The Pi has a built-in Ethernet jack that can be used to connect with the World-Wide-Web. The Pi will be able to run server software, such as Apache, and a web framework, such as Django, to handle http requests and send web pages with data from the database to clients. The location of the gateway node will be point 17 in Figure 2, which is Jared's house.

There should be a “Status” webpage hosted somewhere else, such as on RIT’s Gibson server. Every 5-10 minutes, a HTTP get request to the main server will occur to get the server status. If the server returns “Ok”, the status page will display that everything is up and running. If the server returns “maintenance”, the status page will display that the server is undergoing maintenance. If a 404 is returned, then the server is down, and the status page will display “down.” The purpose of this is so if a user cannot access the server, they can go to this status page and see if there is maintenance going on, or if the site is down completely.

User Interface and Controls

The Django web framework will be used to implement the sensor network website. Django uses the Python programming language along with an HTML/Python hybrid templating language, and allows for the user to import CSS and external and internal media and JavaScript. A user interface will be developed that will allow users to access the database end of the sensor network through the Raspberry Pi’s web server. Django allows for an Apache server connection, and also interfaces with several database formats, such as SQLite, which will provide all of the necessary features for our project within a relatively small footprint.

The web interface will allow access to two user groups, user and admin. The standard user will have the ability to view sensor node information live while it is being captured, and to view accumulated data as it is received as well. This information can be filtered on a per-node, per date, per time period, or per weekday basis. Accumulated data from the entire network can also be viewed all at once. The data stored in the database will include, but will not be limited to the amount of CV entities that have passed in front of a node, the nature of the entity (horse, human, or bicycle), the direction in which the entity was travelling, the access point at which the entity entered the trail, the access point at which the entity left the trail, and the overall time that the entity spent on the trail.

Admin capabilities extend the standard user’s access to view sensor data such as temperature, battery charge, AC current being generated, output from the “battery ok” port, DC voltage output from the boost converter, and whether or not a node is "online". When a node is not "online" or a sensor is processing an alert, such as battery temperature exceeding a certain threshold, the website will process the alert and post a view alert message popup, at which point it may either be viewed or dismissed. These messages will be accumulated in a message center for the admin to view when desired. Messages will also be sent via email to the administrator's provided account. The admin will also have the option to power down a node remotely, attempt to power on a node, or reboot a node. History of the alerts processed at a node will be viewable by date and time. All information from the database (within reason) can be removed or modified by the admin.

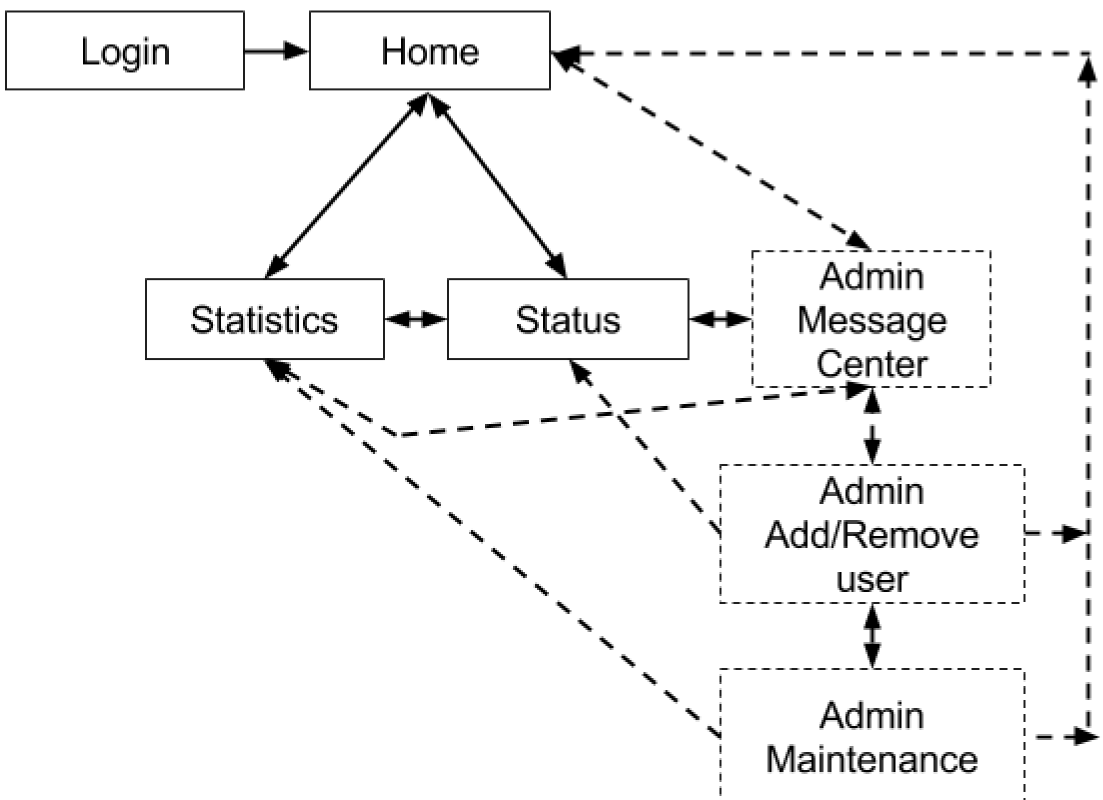

Figure 19 - User Account Panes and Transitions

Figure 19 shows a rudimentary user web site flowchart that includes all of the desired panes available to the user and admin. The dotted line signifies a transition that is only reachable for people with an admin account. Solid lines are reachable by all accounts.

When users and admins login, they are redirected to the home page. From the homepage, users and admins can navigate to the statistics page, which shows the trail statistics, and the status page, which shows the status of each node on the trail. Admins can go to three admin pages. The first admin page is the message center, which shows any important messages admins should know. From the message center, admins can then move to the add/remove users page, and the maintenance page. The home page, stats page, status page, logout, and admin message center (for admins only) is accessible from all pages.

Figures 20 - 25 are rough outlines of what the website will look like:

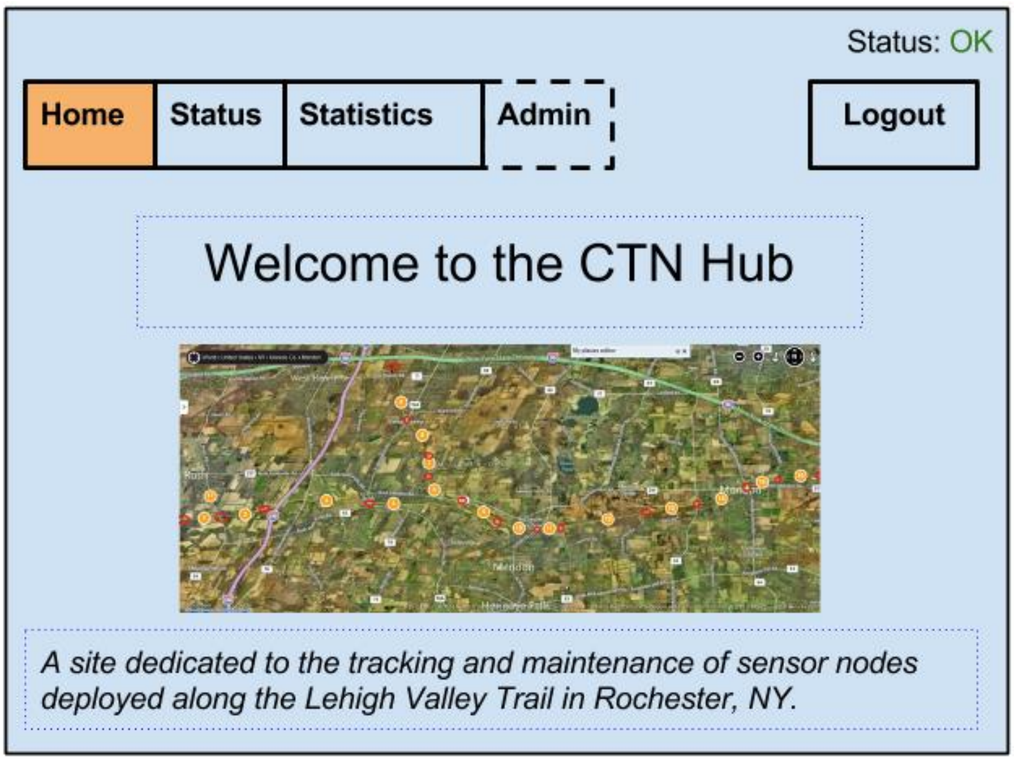

Figure 20 - Home page

Figure 20 shows the homepage of the website. It will only show a welcome message, and have buttons that go to the other pages.

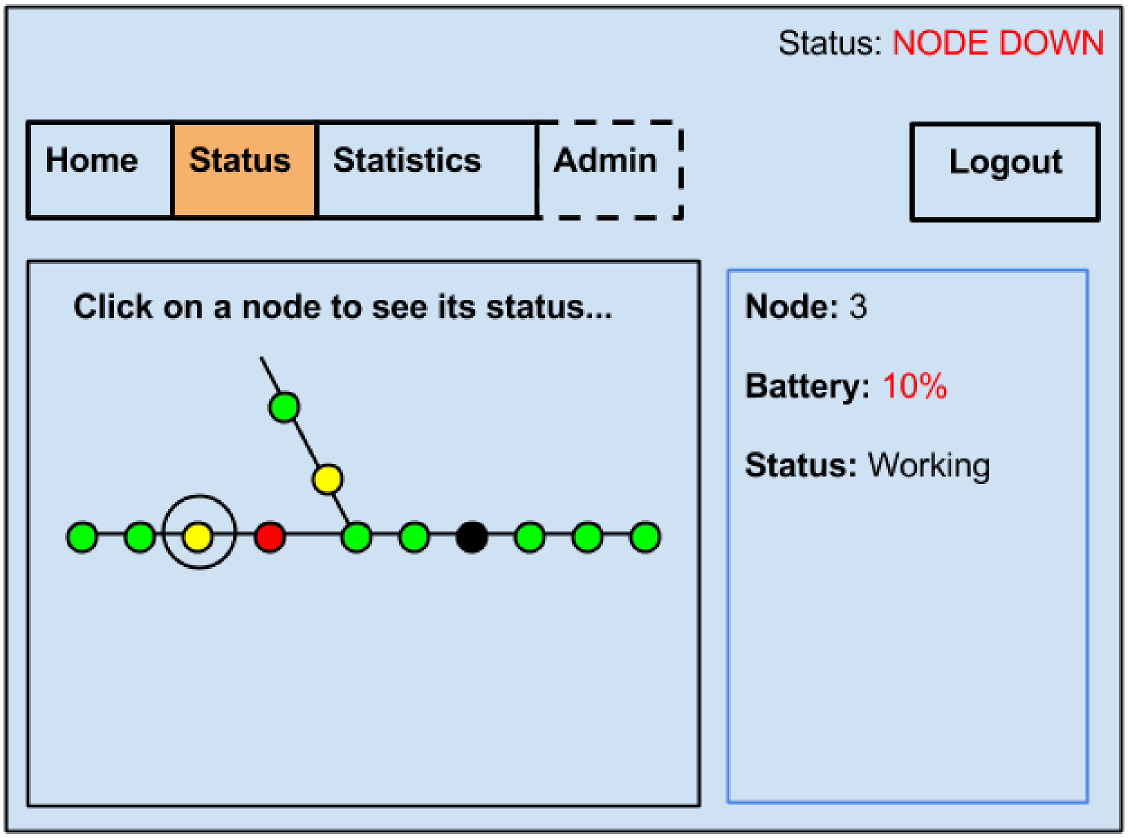

Figure 21 - Status page

Figure 21 shows the status of each of the nodes on the trail, with each node being a green, yellow, red, or black circle on the right. A green circle means the node is functioning correctly. A yellow circle means the node will need maintenance soon, such as in the case of a low battery. A red circle means the node cannot be communicated with, and immediate maintenance is needed. A black circle means the node has been purposefully turned off. Clicking on one of these circles will produce more specific information of the node in the right panel. For example, in the above figure, node 3 is selected, and on the right it shows some statistics such as the battery remaining, and if it's working.

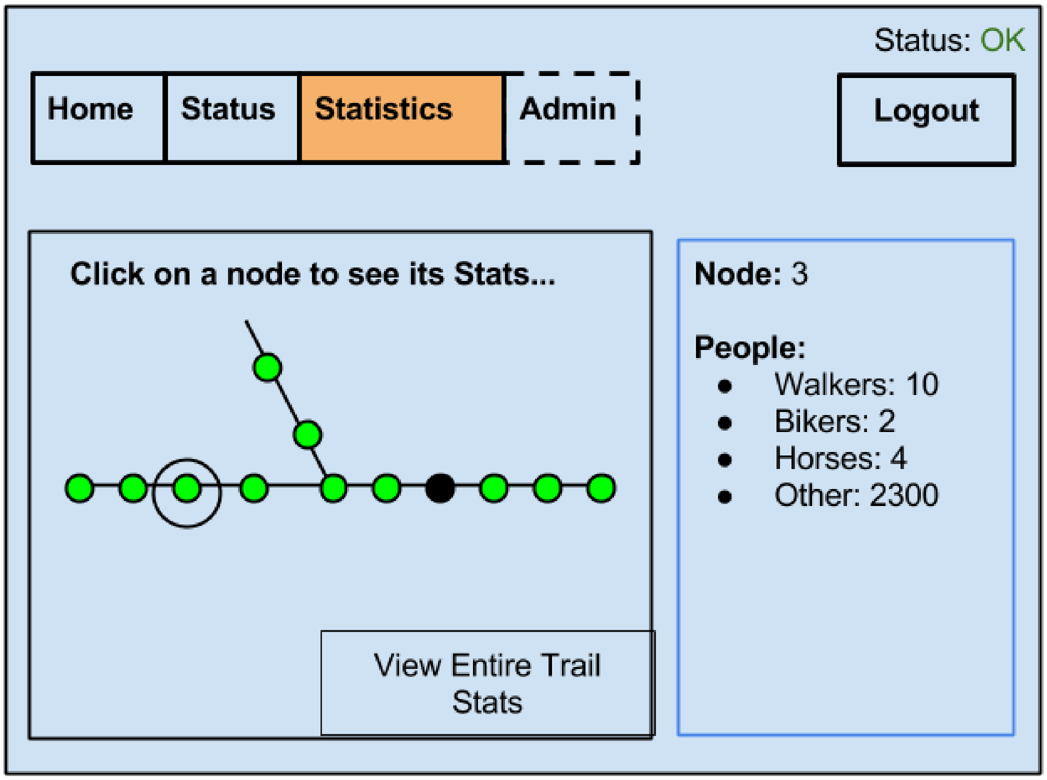

Figure 22 - Statistics Page

Figure 22 is the page the user will go to get the usage data. Like in the status view (Figure 21), each node will be color-coordinated based on their status. In the above view, all the nodes are either working, or turned off on purpose. The user can click on any node to get specific stats of a node, or click on the "View Entire Trail Stats" button to view stats of the entire trail. All stats will appear in the right pane.

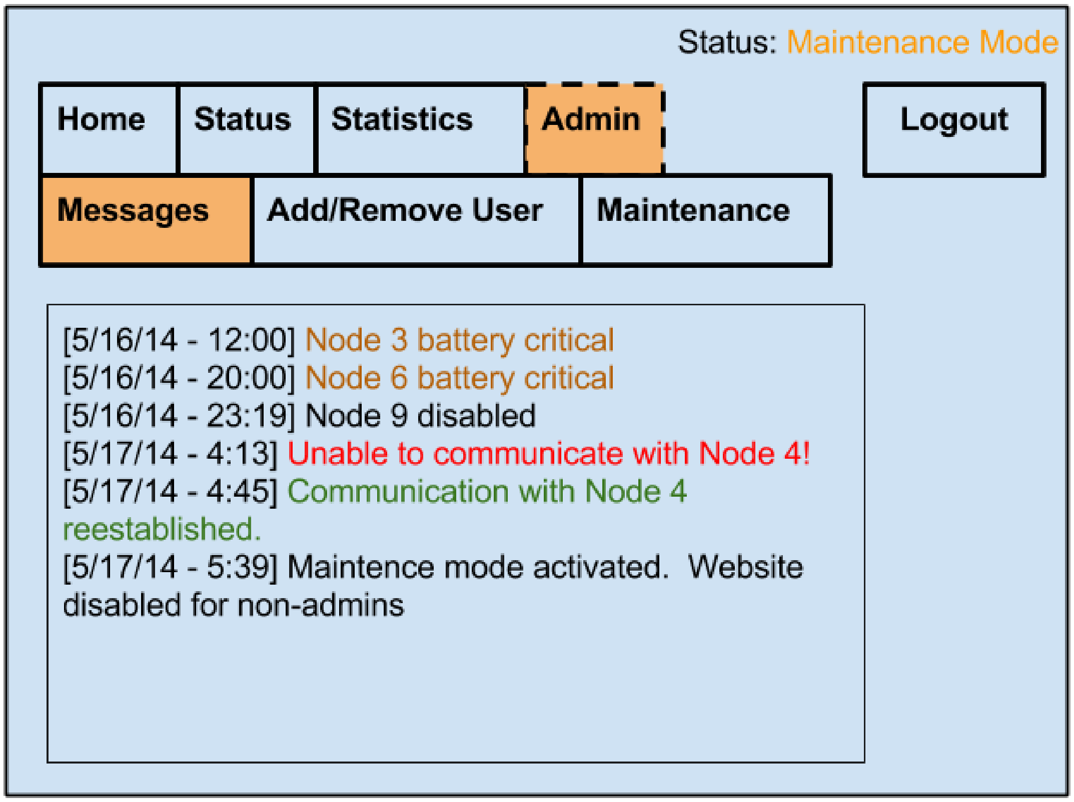

Figure 23 - Admin message center

Figure 23 shows the admin message center. This is only viewable to admins. It shows all important messages that developers or maintainers need to know.

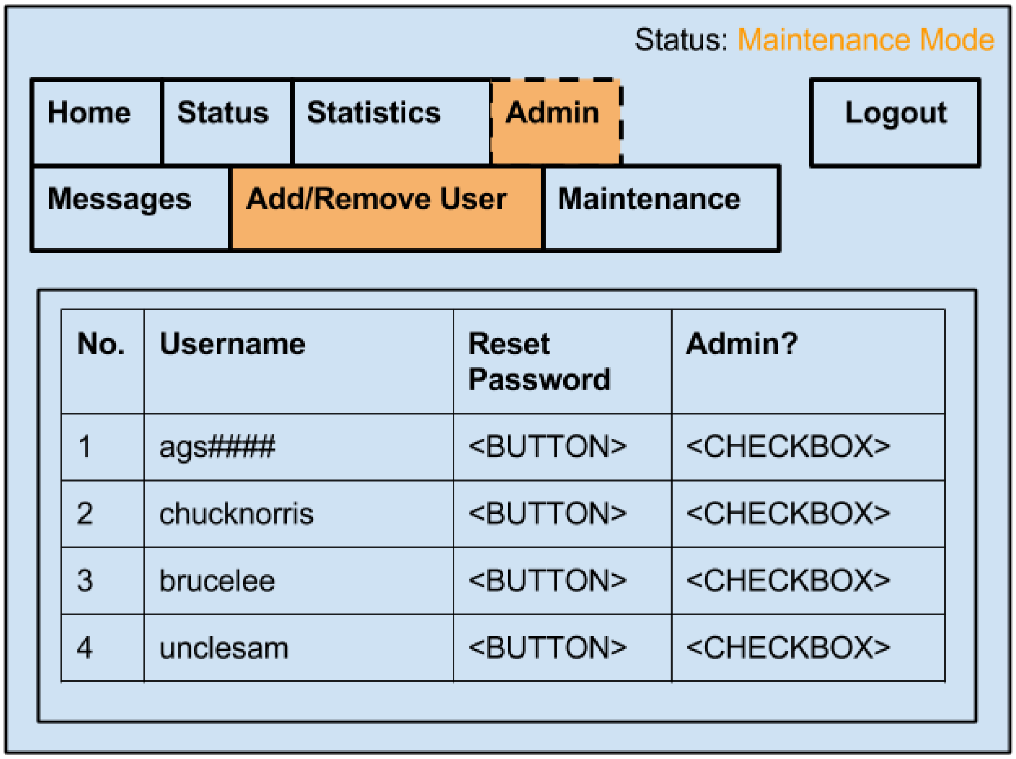

Figure 24 - Admin Add/Remove User View

Figure 24 shows the view associated with the Admin user account that allows an administrator to reset the password or change the user account privileges for any stored account.

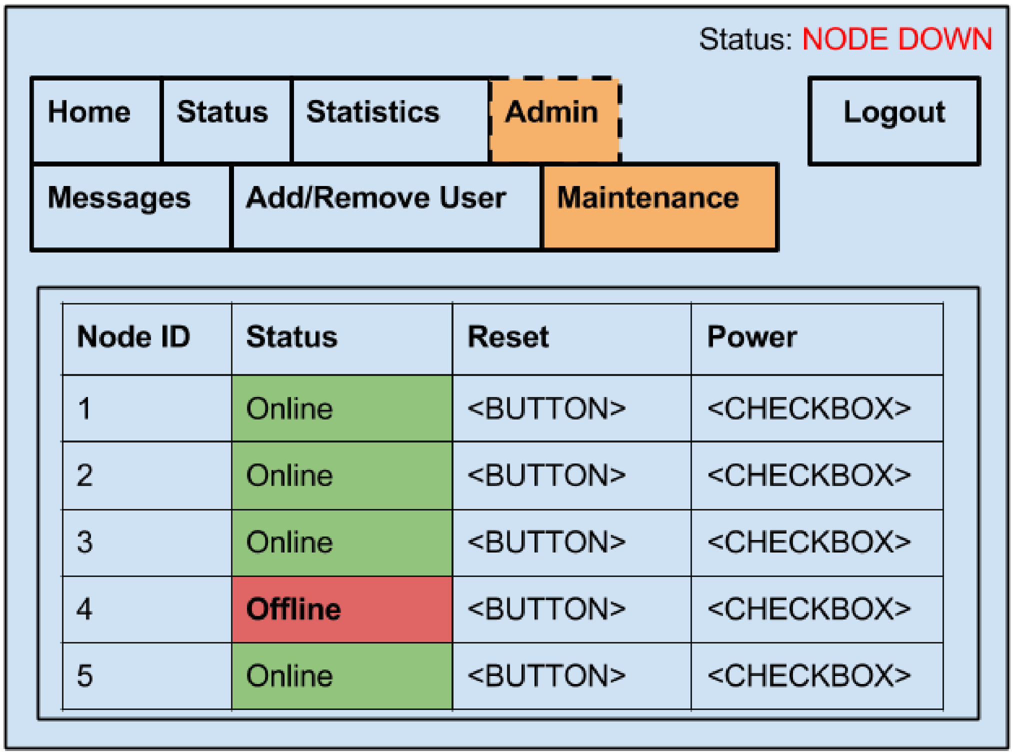

Figure 25 - Admin Maintenance View

Figure 25 shows the maintenance view associated with the admin user group. Administrators will be able to access this view to view whether a node is online or offline, reset any given node, and power a node on or off.

Engineering Standards

- C/C++
    - Both C and C++ are standardized by the International Standards Organization. C and C++ will be used to program the Pixy Cameras, and can also be used in the gateway node to read in data from the trail nodes, and write it to the database.
- Hypertext Transfer Protocol (HTTP)
    - When a client tries the access the website, they will be using HTTP “get” and “post” requests. Django will handle the HTTP the requests, and return a webpage to the user.
- Portable Operating System Interface (POSIX)
    - The Raspberry Pi used for the server and gateway will be running Linux, which is a “mostly-POSIX compliant” computing platform.
- ZigBee
    - ZigBee is a communication protocol for radio modules. The Digi XBee is based off on the ZigBee specification.
- Universal Asynchronous Receiver/Transmitter (UART)
    - UART is used for transmitting serial data. UART will be used to send or receive data between the MCUs and the XBee radio modules that need to be sent out or are received from the mesh network.

There are currently no industry or engineering standards for vision systems, and algorithms may be changed as desired to increase overall system performance.

Multidisciplinary Aspects

- Electrical Engineering
    - Needed to design and build the hardware associated with the signal conditioning circuit, and to interface AC with DC.
- Computer Engineering
    - Needed to program digital components such as the image processor,

image sensor, and boost converter.

- Software Engineering / Computer Science
    - Write software for recognizing the mode of transportation of the person who walked in front of the node.
    - Write software to save all data to a database.
    - Write software so the nodes can communicate with each other.
- Mechanical Engineering
    - Build a housing to protect the nodes from the weather.
    - Design and build the windbelt for the trickle-charger.

Background

For this project, several classes offered at RIT will be of use.

- Data Communications
    - Gives knowledge of networking, which may be important for setting up the nodes to be in a network.
- Interface and Digital Electronics
    - Gives knowledge on how to use analog filter, which will be used in the windbelt.
- Computer Science 1 and 2
    - Gives knowledge on computer programming, design patterns and data structures with Python.
- Computer Science 4, Applied Programming
    - Gives knowledge on how to write code in C and C++, which is how the MCU will be programmed.
- Web I
    - Gives knowledge on how to design a static website using HTML and CSS.
- Intro to Software Engineering
    - Gives knowledge on how to build a website using the web framework, Django.
    - Gives knowledge on how to communicate with a database through a

website.

Outside Contributors

Dr. Jeff Wagner is a professor in the RIT Department of Economics. The requirements specification were based to support Dr. Wagner's research in the area of benefits transfer analysis.

Constraints / Considerations

Sustainability - Since the nodes will be operating far away from any potential power source, a low current source must be provided through use of wind energy. Nodes will be pre-charged, and the trickle charge provided to them by the windbelts will merely be a way in which the pre-existing charge will be maintained for a longer period of time. Therefore, the nodes will all be operating within a variable time constraint, and the LiPo batteries will require a manual recharge when this period expires. The time constraint will fluctuate dependent on the amount of wind that is converted to energy within a given time period. During windier periods, the nodes will remain in deployment for longer, and during periods of less wind nodes will need to be manually recharged more frequently. Power management algorithms will also be employed on the device to minimize the amount of power being used by the microcontroller and associated peripherals during periods of low charge. Ideally, the windbelts combined with effective power conditioning and management should significantly lengthen the deployment period of the devices.

Ethical/Privacy - The nodes will be recording the people walking by them. This makes the privacy of the people on the trail a consideration. With this in mind, the nodes will not save any photos of the trail users unless explicit approval of the trail owners has been provided. Nodes will function solely to determine the person's mode of transportation, and send statistical data through the gateway node to the cloud server.

Property Rights/Political - Another consideration is whether or not the network can actually be deployed on the trail. The owners of the trail will need to be notified, and proper permissions will need to be obtained to perform research within the 15 mile stretch of trail.

Ethical/Privacy - It should also be considered as to who has access to the collected data. It could be a confined group of people performing research and development for debugging and statistical use, town personnel could also have access to it, or all the information could be available to the public. Access level should also be changed throughout the course of development to deployment.

Extensibility - The modules should be able to be repurposed for a variety of analog tasks. The basic platform is being adapted to the current application of collecting trail usage data for the Lehigh Valley Trail, but a variety of analogous applications can be imagined: wildlife data collection, agricultural data collection, etc. The modules need to be developed with this kind of flexibility in mind. The same needs to be considered for the database backend; it too will need to be flexible so as to be able to adapt it to other applications.

Manufacturability - The modules are fairly complicated in both the number of systems and their integration. This was one of the considerations that caused us to choose the Pixy sensor over developing our own unit. Manufacturing should be off loaded onto specialist partners as much as possible. We think we have done that to a high efficiency with our component selection.

Reliability - This is a strong consideration as we are using the network for data collection. We need to show that the modules will reliably capture the required information and that that data will make it to the database backend. We hope to have full redundancy for the wireless network across the nodes to increase reliability.

Environmental - There is a small chance that the battery may explode, causing a fire. The enclosure for the nodes should be designed to prevent the fire from spreading by making it fire retardant.

Cost Estimates

The table below shows the estimated cost for the project.

Table 4 - Bill of Materials

Part	Part Name	Price	Quantity	Total Price	Ship Time	Source
Li-Po Battery	**Tenergy Li-Ion 3.7V 15600mAh**	$40.49	16	$560	Immediate	All-Battery
Schottky Diodes	1SS389(TL 3,F,D) High Speed 0.23V VF	$0.16	16	$2.40	Immediate	Mouser
Resistors and Capacitors	--	~$0.10	32	$3.20	Immediate	CE Department
Boost Converter	bq25504	$6.17	32	$197.44	In stock.	Mouser
Windbelt Frame	--	$1.00	16	$16.00	Immediate	3D Printed
Magnet	--	$1.00	16	$16.00	Immediate	Reuse old hard drive magnets
Windbelt Coil	--	$2.00	16	$36.00	Immediate	Hand Coil
PCB	--	$5.00	16	$80.00	Two weeks	Designed and Fabricated
Radio Module	XBP9B-DMST-002	$39.00	17	$714.00	Immediate	Mouser
Camera Board	Pixy Board	$75.00	16	$1200.00	Ready in April	Charmed Labs

Antenna	ANT-ELE-S01-005	$3.15	17	$53.55	Immediate	Mouser
Web server / Gateway	Raspberry Pi	$35.00	1	$35.00	N/A	Group Members
Wired Router	-	$30.00	1	$30.00	N/A	Group Members
Ethernet Cables	-	$6.00	2	$12.00	N/A	Group Members
XBee Breakout	992-XBEE-USB XBee-USB	$27.92	1	$27.92	Immediate	Mouser
Infraed Sensor	ZEPIR0Ax S02MODG	$10.80	16	172.80	Immediate	DigiKey
			Total Price:	$3159.31		

The total price of the project comes out to just over $3150 for 16 trail nodes and a gateway node. A grant for $1000 has been applied for to offset the cost. Team members own some of the components already, which further decreases the cost. The department of Computer Engineering will provide $300 ($100 per person in the group) as well. This means the total cost, out of pocket, will be about $1600, or roughly $530 per group member.

Due to the high cost of the project a scaled proof-of-concept will be deployed first to prove our methodology before additional modules are built and deployed. We will initially limit the size of the network to five nodes. By limiting ourselves to this number, the cost will be reduced to $1067 for everything. Subtracting out the supplied grant of $1000, and components the team already has leaves $20.77 unaccounted for, which the Computer Engineering department can help cover with the $300 offered to us.

Testing Strategy

Signal Acquisition and Conditioning

Testing of the power acquisition and conditioning circuit will involve five phases as shown in Table 1. Phase 1, early testing, will involve varying the length of the windbelt to maximize AC current production and minimize length. Subsequent tests of the early stage involve varying fan speed and the angle at which wind is applied.

(***Warning: All Lithium-Polymer battery testing MUST be accomplished with the battery in plain site prior to remotely testing. Li-Po batteries can become highly unstable if charging conditions are less than nominal.***)

Table 5 - Signal Acquisition Test Descriptors

Test ID	**Phase - Component**	**Description**	**Pass Condition**
P.1	Early - Windbelt Component	Vary length of the belt and measure AC voltage peak-peak using an oscilloscope.	AC current production is at a maximum, size of windbelt is at a minimum.
P.2	Early - Windbelt Component	Vary fan speed and measure AC voltage using oscilloscope.	AC current production is maximized.
P.3	Early - Windbelt Component	Vary the angle at which the optimal wind speed is applied and measure AC voltage.	AC current production is maximized.
P.4	Intermediate I - AC/DC Rectification Component	Connect optimal design from early testing phase to half-wave rectifier circuit. *(Requires completion of windbelt.)*	Smoothing levels are adequate, and levels are held at 80% or higher of 3.3V with minimum fall time.
P.5	Intermediate I - AC/DC Rectification Component	Connect optimal design from early testing phase to full-wave rectifier circuit. *(Requires completion of windbelt.)*	Smoothing levels are adequate, and levels are held at 85% or higher of 3.3V with minimum fall time.
P.6	Intermediate II - Primary Boost Converter Component	Connect output of rectifier to boost converter at VIN_DC port (pin 2). Measure output of boost controller at VSTOR (pin 15) using an oscilloscope (*Requires completion of custom PCB*	Measure voltage levels at output over elongated periods of time. Levels should not exceed those allowable by the battery (4.2V). Observe and track the DC output signal as it

		power management module.) When running without MPPT, reference Figure 4 in the bq25504 technical document for the proper port configuration	fluctuates over time to determine whether MPPT is necessary. Verify that the boost controller remains on during harvesting periods.
P.7	Intermediate II - Lithium Polymer Battery Component	Manually charge the Li-Po battery at the specified voltage for a nominal charge period. Measure the output voltage with a multimeter. *(May be done in an automated fashion as well.)*	Measure voltage levels of the battery periodically until it is determined that the battery has fully discharged. Record the time it took for the battery to discharge.
P.8	Intermediate II - Voltage Regulation Component	Connect fully charged Li-Po battery to voltage regulator on custom PCB. Test DC output with multimeter at both outputs. *(Should be completed after battery testing)*	Measure voltage levels at both ports to ensure that 3.3V is being provided to the XBee and 3.7V is being provided to the second boost converter (6-10V out).
P.9	Intermediate II - Secondary Boost Converter Component	Connect 3.7V pin to secondary boost converter and measure voltage at the output using a multimeter. Do not configure MPPT yet. *(Not sure which boost converter will be used yet)*	Measure voltage levels at the output of the boost converter. Output should be between 6 and 10 V to adequately supply the Cortex processors. Observe the output over time to determine whether MPPT is required at the secondary boost converter level.
P.10	Advanced I - Primary Boost Converter Component	Connect the battery to the boost converter as shown in Figure 4 of the bq25504 technical document. Measure duration and level of charge without MPPT enabled across the poles of the battery and at pin 15. *(Should be completed after battery testing. Do not leave the battery unattended!)*	Battery duration, charge level and output should be within 20% of the measured discharge period of the battery when measured separately. Compare discharge rate with those of the battery without the sustainable energy source.
P.11	Advanced I - Primary Boost Converter	Configure the boost converter as shown in Figure 2 of the bq25504 technical document. Measure duration	Battery duration, charge level and output should be within 20% as above for elongated periods.

	Component	and level of charge with MPPT enabled for a solar energy application with the Li-Po battery attached. *(Should be completed after battery testing. Do not leave the battery unattended!)*	Compare discharge rates with those of the battery without the sustainable energy source, and with the source but without MPPT.
P.12	Advanced I - Secondary Boost Converter Component	Configure the boost converter for use with MPPT and measure the output of the converter using a multimeter. *(Should be completed after non-MPPT testing if required)*	Measure voltage at the output over elongated periods to ensure that the signal provided is within 6-10V, and that adding MPPT further stabilizes the signal for use with the MCU.
P.13	Advanced II - Secondary Boost Converter Component	Configure the boost converter for use with MPPT and connect the M0 and M4 processors. Measure voltage at the output using a multimeter. *(Should be completed after MPPT testing without load.)*	Measure voltage at the output over elongated periods to ensure that the signal provided is within 6-10V.
P.14	Advanced II - Integration with Focus on Signal Acquisition	Deploy a single node in a controlled environment and monitor power conditions closely. *(Should be completed after all other component testing.)*	Measure voltage levels, battery duration and charge throughout the elongated period, and confirm that CV algorithms, networking, and all power dependent subsystems are fully-functional.
P.15	Advanced II - Integration with Focus on Signal Acquisition	Deploy a single node in an outdoor environment. Allow node to operate without intervention and monitor conditions remotely. *(Should be completed after controlled integration testing.)*	Measure voltage levels, battery duration and charge throughout the elongated period, and confirm that CV algorithms, networking, and all power dependent subsystems are fully-functional.
P.16	Advanced III - Acceptance with Focus on Power Management	Deploy multiple (3-4) nodes in an outdoor environment. Allow node to operate without intervention and monitor conditions remotely.	Measure voltage levels, battery duration and charge throughout the elongated period, and confirm that CV

	Requirements	*(Should be completed after controlled and single-node integration testing.)*	algorithms, networking, and all power dependent subsystems are fully-functional.

Intermediate testing focuses on continuing testing with the optimized design chosen from phase 1. The rectifier circuit is connected to the windbelt, and smoothing and fall time are optimized. For the second intermediate phase, the focus is on providing safe levels at the output of the boost converter with the MCU connected to a non-battery power source.

Advanced testing allows for connection of the battery and MCU to the boost converter and deployment in an outdoor environment. Battery voltage levels, duration, and charge levels will be measured for as long as the battery maintains its charge. Once components are proven fully functional, the signal acquisition system will be tested as a whole in a controlled testing environment. Acceptance testing will be performed in the field, most likely with less distance between nodes for easy maintenance and monitoring prior to actual deployment on the Lehigh Valley Trail. Tests will be run to ensure that all power-dependent requirements are satisfied after integration level testing has completed.

Computer Vision Testing

For the software side of the nodes, unit testing can be used for all software written while development is underway. Unit testing prevents bugs from escaping into the field, as it ensures all functions and branches work as intended separate from all other modules.

To test the software algorithm that determines what walks by a node, the nodes can be set up in a controlled environment, such as inside of a lab. A person can then walk by with or without a bike, and the node should be able to identify if the person is walking or riding a bike. The subject should walk past the node at various speeds, and the node should still be able to make a capture. The node should be able to pass captured information to the rest of the nodes, and to the gateway so it can be recorded by the database.

Various lighting scenarios should be tested as well. In a lab environment, the lighting should be adjusted so it gets dimmer until the node is unable to make accurate readings. This will be the minimum light needed for operation, and the node might not require frames to be captured at times like this in the field. During the lab test, a light should be positioned at various angles, such as behind the node shining on the target, behind the target shining on the node, overhead, and to the left and right of the node. Regardless of where the light is positioned, the node should still operate accurately.

When the node is built, and the software written, a stress test that can be performed is to deploy the nodes on the RIT quarter mile, which gets a lot of foot traffic. This test will also show just how accurate and fast a node is. If the node just cannot keep up with the traffic, some redesigning might need to occur.

Table 6 - Computer Vision Test Descriptors

Test ID	Phase	Description	Pass Condition
V.1	Early - Image Sensor Component	Verify that the image sensor is correctly capturing images by connecting the Pixy Cam via USB, loading the GUI and observing.	Images in front of the lens are rendered correctly in the Pixy Cam GUI window. It can be assumed that the M0 is functioning correctly.
V.2	Early - Image Sensor Component	Verify that the image sensor is correctly identifying colors by running the included color identification algorithms with the board connected via USB	The chosen color is amplified in processed images, as shown in the Pixy Cam GUI window. It can be assumed that the M0 is functioning correctly.
V.3	Intermediate - Infrared Component	Connect the board via USB and connect an Infrared sensor to the camera via GPIO or UART. Dim the room of all light sources. Verify that the sensor data is correctly displayed in the Pixy Cam GUI window by walking in front of the	The image should be correctly represented with more orange to red colors showing higher temperatures. The background should contain blue and purple colors for lower temperatures.

		camera and observing..	
V.4	Intermediate - CV Algorithm Component	Compile and load the CV software onto the Pixy Cam. Walk by the Pixy Cam to verify whether the algorithm is functioning properly. *(To be tested with and without Infrared sensor connected.)*	The image should be identified and the correct CV size capture, speed and direction determinations are made.
V.5	Intermediate - CV Algorithm Component	Vary speeds at which the subject walks by the Pixy Cam to verify that the algorithm still functions properly. *(To be tested with and without Infrared sensor connected.)*	The image should be identified and the correct CV size capture, speed and direction determinations are made.
V.6	Advanced - CV Algorithm Component	Walk by the Pixy Cam with a bike or other large object to verify that the algorithm still functions properly. *(To be tested with and without Infrared sensor connected.)*	The CV algorithm is able to correctly discern between a walking and biking commuter based on speed of the individual, and the correct output displays in the GUI.
V.7	Advanced - CV Algorithm and Infrared Component Integration	Compile and load the CV software onto the Pixy Cam. Dim the lights in the room. Walk by the Pixy Cam to verify whether the CV algorithm is functioning properly. *(To be tested after CV and IR component tests pass.)*	The algorithm should be able to use the Infrared data with the lights dimmed to accurately detect size, direction and speed of an individual.
V.8	Advanced - Computer Vision and Networking System Level Integration	Compile and load the CV software onto the Pixy Cam. Connect the cam via USB. Connect the CMUCam to the XBee as well via GPIO or UART header and provide power to it via pre-charged Li-Po battery connected to custom voltage regulation unit. Dim the lights in the room. Walk by the Pixy Cam to verify whether the CV algorithm is functioning	The algorithm should be able to use the Infrared data with the lights dimmed to accurately detect size, direction and speed of an individual. The algorithm should also work with lights on. This data should be broadcasted to a central node, server, and pushed to a website to view for confirmation.

		properly. Turn on the lights and perform the same tests. *(To be tested after CV, IR, voltage regulation and GPIO component tests pass. The website GUI should also be operational, but data can alternatively be viewed through local GUI.)*	
V.9	Advanced - Computer Vision and Power Management System Level Integration	Compile and load the CV software onto the Pixy Cam. Connect the CMUCam to the secondary boost converter. *(Warning: Do not perform this step until component level testing of the secondary boost converter is complete.)* Connect the CMUCam to the XBee as well via GPIO or UART header and provide power to it via pre-charged Li-Po battery connected to custom voltage regulation unit. Leave out everything prior to the battery in the Signal Conditioning circuit. Walk by the Pixy Cam to verify whether the CV algorithm is functioning properly. Turn on the lights and perform the same tests.	The algorithm should be able to use the Infrared data with the lights dimmed to accurately detect size, direction and speed of an individual. The algorithm should also work with lights on. This data should be broadcasted to a central node, server, and pushed to a website to view for confirmation.
V.10	Advanced - Computer Vision Acceptance Testing	Compile and load the CV software onto the Pixy Cam. Connect the CMUCam to the secondary boost converter. *(Warning: Do not perform this step until component level testing of the secondary boost converter is complete.)* Connect the CMUCam to the XBee as well via GPIO or UART header and provide power to it via pre-charged Li-Po battery connected to	The algorithm should be able to use the Infrared data with the lights dimmed to accurately detect size, direction and speed of an individual. The algorithm should also work with lights on. This data should be broadcasted to a central node, server, and pushed to a website to view for confirmation.

		custom voltage regulation unit and add signal acquisition and conditioning circuit including primary boost converter, rectifier, and windbelt. Walk by the Pixy Cam to verify whether the CV algorithm is functioning properly. Turn on the lights and perform the same tests. Vary size of CV subjects. *(Perform this test after System level integration is complete for each system involved.)*	Extended testing should be performed at this level once all system level and integration is complete. The node should be able to process CV algorithms for an extended period with sustainable energy available.

Enclosure testing

The node will be deployed outside in Rochester's infamously bad weather. Therefore, a test to ensure the node remains intact in extreme weather needs to occur once the weather-proof chamber is completed.

Table 7 - Enclosure Test Descriptors

Test #	Test Name	Description	Reason	Pass Condition
E.1	Temperature Test	The nodes should be placed in a cold environment to simulate Rochester's winter, and a hot environment to simulate Rochester's summer.	Rochester has a variety of weather conditions, which can range to several degrees below zero, and up to almost 100 degrees. The nodes will be outside during these times, and should be able to survive in	If all the node's hardware still works after being in the extreme environments, the test passes.

			them.	
E.2	Rain Test	Water can be sprayed on the enclosure to simulate rain, and a powerful fan can be used to simulate consistent wind. For these tests, paper towels or some kind of other material that reacts to water should be placed inside the enclosure instead of the expensive hardware. This way, if water does manage to get in, no hardware is damaged.	Rochester can have days of rain in a row. The housing needs to be able to keep the water out at all costs, otherwise the hardware might get fried.	If no water gets into the enclosure, the test passes.
E.3	Drop Test	There should be a drop test of 5-6 feet to ensure that the trail node's enclosure is sturdy enough to protect the hardware inside. This can be done by placing a raw egg, or something else cheap and easily breakable, in the enclosure instead of the expensive hardware.	It is possible that while deploying the nodes, they could be dropped. The enclosure should protect the hardware from drops.	If the egg cracks when being dropped, then the enclosure is not sturdy enough, but if the egg is intact, the test passes.
E.4	Fire test	Pack flammable materials inside of the enclosure that resembles the battery and set fire to it (with a fire extinguisher near by).	There is a chance that the battery can catch fire. To prevent a forest fire, the enclosure should be able to contain any fire that may occur from the inside.	If the fire stays contained within the enclosure, the test passes.

Network Architecture Testing

It is essential that all the nodes are able to communicate with each other over a long distance. To ensure that all the nodes will be communicating with each other through the XBees, the following tests can be done:

1. The first test that should be performed is a small scale test to ensure that the XBees are able to be configured through UART and send messages via the DigiMesh protocol. This test should be performed once the XBees are acquired.
    a. Acquire 4 XBees.
    b. Attach the 4 XBees to either Raspberry Pis or the Pixy Cam through UART. A mixture of Pis and Pixy Cams is preferred, as the Pixy Cam will be the MCU for the trail nodes, and the Pi will eventually act as the gateway node.
    c. Ensure the XBees can be configured through UART. If they are unable to be configured, then the XBee Development kit might need to be purchased in order to program the XBees.
    d. Ensure the XBees can send and receive messages from each other. If they are able to, then the XBee Development Kit is not needed.
    e. For this test, the XBees can be in close proximity with each other. Test 3 increases the distance.
2. The second test that should be performed is a test to see how far away the XBees are able to communicate with the selected antenna. This should be done once the previous test passes.
    a. Create 2 nodes by connecting an XBee to either a Pi or a Pixy Cam through UART.
    b. Make one node stationary, and take the other node and go as far away from the stationary node as possible before communication cuts out.
    c. The biggest distance between two nodes on the trail is about 1.75 miles, so the XBees should be able to communicate with each other from at least 2 miles apart.
    d. If this test fails, then a more powerful antenna needs to be purchased.

   e. Preferably, the test should be performed on the trail itself, as that's where the nodes will eventually be placed.

3. The final test is a small scale trail test. This is where a subset of the trail has nodes deployed to it. If this test passes, then it is probably safe to purchase more nodes and cover the entire trail. This test can be performed when the last test passes.
   a. Attach 4 XBees to either Pis or Pixy Cams.
   b. Deploy 4 nodes are positions 17, 2, 3, and 4 (referring to Figure 2). These four positions are the most spread apart spots on the trail.
   c. If all the nodes can communicate with each other, then the test passes, and it is probably safe to up the scale to cover the whole trail.
   d. If the test fails, then some of the nodes might need a more powerful antenna.

Server Security Testing

Once the Raspberry-Pi server and all security implementation is in place, an attempt will be made to take control of the server from an outside source using DDOS, SQL injection and brute forcing. If this attack succeeds, depending on the amount of time and difficulty it took to penetrate server security, additional security measures may be taken.

Table 8 - Server Security Test Descriptors

Test #	Test Name	Description	Reason	Pass Condition
S.1	Ping test	The router should not be pingable from an outside source.	When many people try to ping a server, it could create a DDOS attack. Therefore, pinging should be removed to prevent this from	Trying to ping the router from outside the network should not work.

			happening.	
S.2	Disable Root Login	While SSHing into the server, users should not be able to login as root	Root is all knowing in Linux. If someone were to somehow gain access to root, they could wreak havoc on the server.	Trying to login as root results in an access denied.
S.3	SSH non-standard port test	While trying to SSH into the server using the default SSH port, the result should be not being able to connect.	By moving the SSH port to a non-standard port, it makes it difficult for some hackers to find the SSH port, and try brute forcing.	Trying to login to port 22 via SSH will fail. Trying to login to the non-standard port will succeed.
S.4	Disabled Password Test	Trying to login with SSH will have an access denied without a SSH key. SSHing will not be allowed without a password.	SSHing with a user name and a password is not safe for a server. A hacker can brute force their way in.	SSHing into a server without a key will result in an access denied. SSHing will not ask for a password.
S.5	White hat hacker test	Borrow one of the many security majors the group knows, give them the IP address, and have them attempt to take the server down.	Having a friendly hacker attempt to break into the server will emulate a more sinister hacker trying to get in. If any information is compromised by the friendly hacker, they will not steal it.	The hackers should not be able to take the server down.

Website Testing

The website is the gateway to the data. However, it can also be a gateway for a hacker wreak havoc. These tests will confirm that anyone with dark intentions will not be able to compromise the server from the website.

Table 9 - Website Test Descriptors

Test #	Test Name	Description	Reason	Pass Condition
W.1	Sanitation Test	Data from the website is sent to the server via http get and post requests. Some commands sent that could break the server include "; && rm -rf /" and "drop table *;". These commands should of break the server.	If the data is not "sanitized" correctly, it is possible for a user to drop a table from the SQL database, or execute shell commands.	Posting and getting from the server using common commands that break a server should not break the server.
W.2	Password protection test	Connect Wireshark, and see if user names / passwords are able to be seen leaving the client and going to the server.	If the website is meant to have a limited set of users, then a user name and a password is needed. The passwords need to be protected.	The password leaving the client should be hashed and salted so a hacker cannot steal them.
W.3	User login test	Ensure valid users can log in and invalid users cannot log in	The data needs to be available to a limited set of users. Therefore, a user name and a password is needed to gain access to the website.	Valid users can log in, invalid users are rejected.
W.4	White-hat hacker test	Borrow one of the many	Having a friendly hacker attempt	If the friendly-

		security majors the group knows, tell them the website, and see if they can break in.	to break into the website will emulate a more sinister hacker trying to get in. If any information is compromised by the friendly hacker, they will not steal it.	hacker cannot break in, the test passes.

Risks

Theft and Weather-proofing

One risk is the nodes might get damaged or stolen. The nodes will need to be weather-proofed to prevent damage from the weather, and ruggedized in case they are dropped. To minimize the risk of the nodes being stolen, they might need to be hidden, such as in a fake rock or camouflaged. Another option would be to place nodes at a vantage point by raising them up with a pole or extending device. This would not only deter theft, but also would allow for the field of view necessary for the CV algorithms that will be used.

Use of Low-Power Modes

There is a risk of the nodes using so much power that they cannot be deployed for very long. Some things that can be done to extend the power of the node could be to put them to sleep at night, and to ensure there is not a lot of CPU processing when the trail is empty by forcing a low power state during these periods. Although these techniques, along with the trickle charger attached to the nodes, can extend the battery's charge, it will probably not be enough to sustain the nodes indefinitely. Therefore, there should be some kind of notification sent to the frontend when a node gets close to or runs out of power, so that the batteries can be replaced.

Battery Damage and Fire

Regarding the battery, there is risk of fire if it is not properly charged with a regulated current at a specified rate, as with all Lithium based batteries. The boost converter should mitigate these risks, but the notification system should once again be used to alert the user and the system when the battery is in a bad state. This would occur after the boost converter starts exhibiting signals at the output that are in excess of what is allowed by the battery, and should result in shutdown of the device, or the signal acquisition portion of the device. Overcharge and undercharge are both a concern with this type of battery as well, and similar notifications should be sent when battery life is at 95% and less than 5% to alert the user that the node's battery is in a critical state. DC signal harvesting should not be active in either case, which can be programmed through the boost converter as well.

CV Procedural Difficulties

The most difficult part of the project is probably going to be identifying the specimens that walk or ride within the line of sight of the node. The node's software needs to determine if the person is walking, riding a bike, or riding a horse, but other factors might come into play, such as lighting, small animals, how close or how far the person is away from the node, how many people are in the shot, and other unknowns. It is going to take some trial and error to get the algorithm right. Use of the Pixy cam will mitigate these risks somewhat, due to its ability to utilize robust versions of modern-day CV algorithms, and due to all of the support and example material that is available through online forums.

High-Risk Investigations

Several components were identified as high-risk, and a detailed investigation has been conducted for each of them, including the choice of sustainable energy used for each node, the MCU, and the sensor node network architecture.

Windbelt

The windbelt can be considered a high-risk component simply due to its nature as a newly innovated sustainable energy solution. Because of this, there are very few example cases on which to conduct research. Furthermore, the results of those cases are vastly diverse due to custom construction of the windbelt and application in widely different settings. Because of this, original research will be needed in identifying an optimal variant of the design and applying that design in the most effective manner.

Other risk considerations are whether or not the windbelt can provide a large enough signal to the amplification stage of the circuit, and whether it can do so consistently. To mitigate this risk, an optimized version of the windbelt will be used, where dimensions, belt length, magnet and coil sizes and distances have been carefully considered and tested to maximize AC current production and minimize size. AC rectification will function not only to convert AC to DC, but will also provide limited reverse-current protection.

The majority of risk mitigation will be handled by the boost converter, which will allow for the use of MPPT algorithms. MPPT will track current production over time, calculate an average, and then adjust internal potentiometer values to zero in on that average during over-efficient and under-efficient periods. MPPT will greatly reduce the risk of battery damage, plating, and fire for this reason. Extensive testing will be conducted in a controlled environment prior to exposing the signal harvesting system to the outdoors. It is possible that no wind will be available in the area. During extended periods of little to no harvesting, the admin must once again be made aware through use of a notification system that no charge is being generated.

Sensor-Node Architecture

The risks associated with the sensor nodes and network communications mainly relate to whether or not range of the nodes will be sufficient. The XBee documentation claims that the communication range of the XBees is 4 miles with a 2.1dB antenna. However, documentation regarding range is typically inaccurate. If the range of the XBees is not long enough, adding a more powerful antenna to the XBee radio modules should suffice.

Milestone Chart

Table 10 - Project Milestones

Milestone	Team member in charge	Schedule Completion Date
Contact Monroe County ● Discuss deployment options for sensor nodes.	Jared	6/1/14
Networking Architecture Configuration and Testing ● Configure XBees for DigiMesh and have them communicating in close proximity ● Range Test ● Small-scale trail deployment	Seth	6/15/14
Windbelt power module design ● Breadboard prototyping ● PCB design ● Ship design for stamping	Alex	6/18/14
Windbelt power module construction and testing ● Solder on components ● Continuity tests	Alex	6/30/14
Server/Gateway setup ● Install software (Django, Apache, etc.) ● Interface XBee with Pi ● Install fail2ban	Seth	7/1/14
Server/Gateway testing ● See Table 8 for specifics	Seth	7/1/14
Sensor hardware testing and integration ● Being playing with Pixy Cam in USB tethered mode ● Interface Pixy Cam with an XBee ● Integrate with existing power module	Jared	7/11/14

Sensor Enclosure Design / Testing • Use CAD tools to design sensor enclosure • Use 3D printer to print the enclosures • See Table 7 for testing specifics	Jared	8/7/14
Windbelt Testing • See Table 5 for specifics	Alex	9/14/14
Sensor Software - Identify targets • Train camera for identifying walkers, bikers, and horses • Train camera to figure out what direction the target is going	Seth	9/1/14
Database Creation • Create MySQL or MariaDB database so data from trail can be saved to it	Jared	9/14/14
Website Creation • Create front end • Link to database • Create status webpage, hosted somewhere else	Alex	9/28/14
Website Testing • See Table 9 for testing details	Seth	10/4/14
Target Data Communication • Sensors communicate target data with each other • Sensors can communicate and write target data to database	Seth	10/5/14
Computer Vision Testing • See Table 6 for testing specifics	Alex	10/28/14
Deployment • Deploy nodes on trail • Activate website	Team	11/9/14