P442 - Final Project
Trail Counter project
Nick Palumbo (npalumbo)
Isaac Fisch (ifisch)
Concept Design Report

Introduction - High level overview of the project

The trail counter to be implemented will be used to track how many people walk on a trail. The trail counter will be placed on a bridge at a local or state park. The trail counter will count activity through vibrations using an accelerometer and save a head count when it determines people are walking by. Along with a headcount of people walking by the device, the device will also save a timestamp when the activity happens so that when the data is pulled, the owner can determine when the activity occurred. This data collection process will help determine when the most active times are on the trail and which trails are the most active if there are multiple devices deployed. If anyone is coming around at night, the trail counter will know.

The trail counter is not very efficient but can roughly last a few months once deployed. The trail counter can run for a good amount of time because of the efficient low power processes that are running on the device. The owner of the device will be able to collect traffic data from the device using an app on their smartphone that has Bluetooth enabled. The Bluetooth capability allows for convenience so the owner does not have to plug anything into the device to be able to collect the data. The Bluetooth connection will be secured so that there is nobody else who can connect to the device except for the owner and those the owner trusts with the authorization passcode set. The secure connection allows for ease of mind when it comes today's integrity, someone tampering with the data, or using the device for malicious use.

Background - Research on other trail/people counters or similar devices

There are several trail counters that are deployed that we have done research on. The one trail counter that was of most interest was the Trafx trail counter. This trail counter has the capability to count different types of traffic. The different traffic types ranges from people walking to people in cars. The trail counter has a memory capacity that can store more than 400 million counts. The battery life is greater than 1 year. When the trail counter is in vehicle mode it is able to track he disruptions in the earth's magnetic field so that it can count the vehicles that pass. With the bike counter the device is able to use its magnetometer to count bicycles that pass.

The company who makes the trail counter makes different variations and one interesting device has infrared so that it can count anything that passes through the beam given off from the infrared. The device comes with software so that the data can be seen visually. An interesting part is that there is no mention of security for the device. There is no thought into

making sure the device is not tampered with or the device is not used for malicious activity.
- https://www.trafx.net/products.htm

Discussion about the approach.

Our approach to the trail counter will not have as many features as the Trafx. However, we plan to implement the efficiencies that involve memory, energy, and convenience for the owner. We plan to implement the ability to track people walking on the trail. Our first step is to get the counter as accurate as possible so that the basic functionality works. Then we will move into the Bluetooth implementation and develop functionality to retrieve data from the device. After we are able to retrieve data over Bluetooth we will begin to add layers of security for the trail counter in order to make a secure Bluetooth connection. If we have time we will add better algorithms for accuracy and temperature monitoring to account for time loss.

Features - key properties of your system

1. Target counter accuracy.

We strive to reach an accuracy that is greater than 60% accurate. In order to get this accuracy, we have to implement an algorithm to closely monitor the vibration feedback and determine if there are one or more people walking, driving, or running by. This will take a good amount of data in order to determine the average vibrations made from each of the test cases. The accuracy is a big part in the trail counter because it is the main data to be collected.

2. Data collection process - how does the user get data from the counter.

The data collection process will first be done with a cord plugged into the device. however, we will be developing an app that will be able to collect the data over the air with Bluetooth. In order to do this, we will have to read through the documentation on how to use Bluetooth with our device. Then it is a matter of how to connect to the device and make sure there are only connections to the device that we allow. We do not want anyone to be able to connect to the device because this would lead to further tinkering from people and we could end up with null or invalid data on the counter.

3. Expected battery life.

The expected battery life of the trail counter is roughly 5 months. This battery life is given the batter that is used and the processes that will consistently be running. The trail counter will need an amount of energy for when it is sleeping mode and more energy when it is not sleeping. When the device is sleeping it will have certain functionality off so that the device can save energy because there will not always be constant traffic around it. However, when there is traffic, the device will wake up and begin collecting data until a certain time or threshold is met where the device will return to sleep mode.

4. Bluetooth

We spent most of our time trying to implement Bluetooth functionality without any luck. Bluetooth had to be moved from the blue Microsystems configurations and work with the chibi os configuration. The trouble with Bluetooth came when trying to import the needed files from the Bluetooth header and c files because once one file was imported there were 10 more that needed to be. If we had Bluetooth working then we could use the given STM32 BLE application on either android or iOS so that we could begin sending and receiving test data. The app would of given us a good test space so that we could see the data being sent. In the future we could implement our own android or iOS app to receive the trail counter data in a more specific way tailored to the needs of the consumer using the trail counter.

5. Security

The security functionality is in the concept stages still because we could not get Bluetooth to work. The security features relied on Bluetooth because Bluetooth itself is not secure. The concepts we came up with are making the trail counter undiscoverable so that anyone with a bluetooth device cannot find the trail counter's discoverable signal. The discoverable signal is helpful when someone wants to pair with a bluetooth device.

Another security feature would be to make the trail counter unpairable so that even if someone had the bluetooth device address they could not pair with the device. We would make it so that the trail counter is pairable by the device owner and after that the owner can make the device unpairable so that nobody else could try to pair with the trali counter. If someone else could pair with the trail counter then they would be able to use the device much like the owner. This is where a password or authentication signature would need to come into place.

We thought about implementing a password but that could be tiresome to have to type in the password each time the consumer wants to get data from the trail counter. In order to make the data collection seamless for the consumer we wanted to implement a signature file that could be held on the consumer's device and the trail counter would check for the signature file when connecting which would authenticate the user. The signature method makes typing in a password unnecessary.

Lastly, we wanted to somehow upgrade the encryption on the bluetooth signal so that the traffic going to and from the device cannot be captured and read by someone other than the consumer of the trail counter. Bluetooth comes with weak encryption along with weak security overall. The functionality we planned to implement would have made the trail counter more secure and there would be a better chance for data integrity and confidentiality.

6. Counter deployment process - what is needed to deploy a counter

Given the size of the device, the device will be able to be hidden under a bridge or somewhere where it is not very visible. In order to deploy the device, the device will need a power source so that it is able to run without being connected to a computer which would defeat its purpose. The trail counter will need little configuration in order for it to be deployed. The configurations needed are the current time and unique name or id set. The current time must be set because when the device is turned on it is unable to determine what time of day it is. The unique name or id will be used in case an owner has multiple trail counters that they own. The name makes the trail counter unique so that the data from the trail counters can be separated so there is no confusion where the data came from. Before getting into multiple deployments we plan to work out the kinks with one device. After being able to implement a secure Bluetooth connection on one device we will move towards an easy way to keep a single passcode for all devices deployed under a given owner.

Theory of Operation

Vision of the software for the system.

The system will be able to count people walking on the trail with an accuracy above 50% and will be energy efficient where the counter can be deployed for up to 5 months. We want to make the basic trail counter work and move towards a more secure and robust device that could be implemented in the field. Security is always a main priority when it comes to IoT devices and a simple trail counter could easily get overlooked and be taken advantage of. This software should be able to make secure Bluetooth connections with user authentication so that the owner of the device knows the device is inaccessible to anyone trying to connect. The trail counter's main objective is to collect data and if the data is removed by a malicious person then the trail counter is worthless.

The trail counter lacks Bluetooth because of problems importing the necessary configuration files but there is still a basic trail counter implementation. The trail counter runs on 2 threads that work together to collect data. The first thread is the stepCounter thread which will constantly check if the accelerometer data is above a threshold set by us. If the accelerometer data is above the threshold then a timestamp would be taken. The second thread is the timerThread. The timerThread will sleep for a given amount of time, in our case 10 seconds for testing purposes and longer for deployment, and then save the count data if it is above 0. We did not want to save data when nothing occurs because of our limited memory.

Schedule/Milestones

* Milestone 1: (Due 2/24)
- Trigger the counter based on a fixed accelerometer threshold.
- Time stamp the number of counts each minute and store in processor RAM.
- Create a ChibiOS command, display_counts that will print the time-stamps and counts.

* Milestone 2: (Due 3/3)

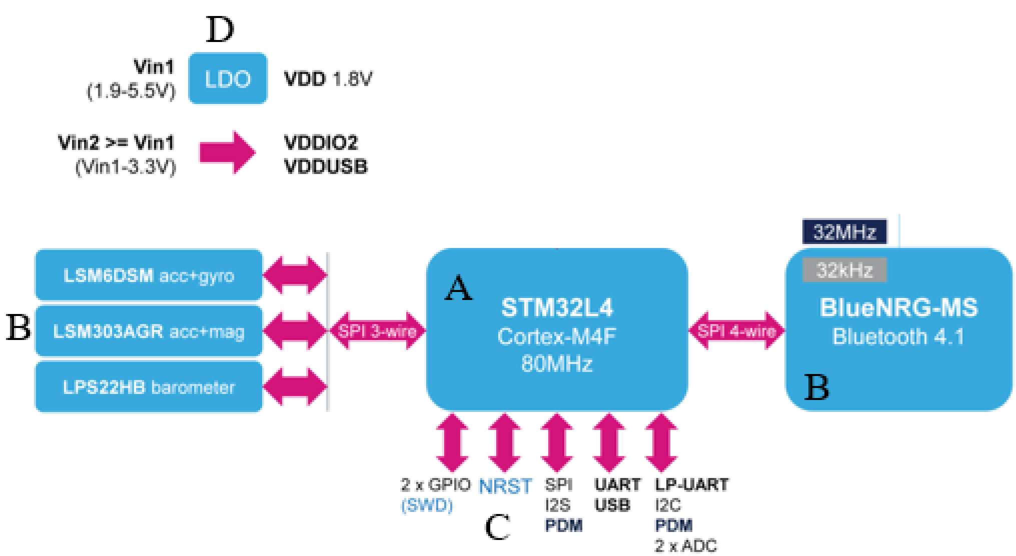
- Store the timestamps/people counts on the SD card in CSV format.
- Determine the accuracy of your counter by mounting it to a campus footbridge for 30 minutes.

* Milestone 3: (Due 3/24)
1. Utilize Bluetooth to collect data from the device (iPhone application hopefully)
2. Implement sleep and stop modes and quantify power savings.

* Milestone 4: (Due 4/21) (x)
1. Implement Secure Data Collection with Bluetooth
2. Develop an approach for the customer update the firmware on the device

* Poster Presentation (4/28) (x)
1. Try to be able to visually display/count people coming to our poster so they get a live demo
2. Make the app available if made so that people can download our app and use it live when they come to our poster

Block Diagram



A: Processor
B: Sensors to be used
C: Memory Storage
D: Power Supply