NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

MOBILE PHONES COUPLED WITH REMOTE SENSORS FOR SURVEILLANCE

by

Bradley J Williford

March 2012

Thesis Co-Advisors:	Gurminder Singh John Gibson
Second Reader:	Neil Rowe

Approved for public release; distribution is unlimited

REPORT DOCUMENTATION PAGE *Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE March 2012

3. REPORT TYPE AND DATES COVERED Master’s Thesis

4. TITLE AND SUBTITLE Mobile Phones Coupled with Remote Sensors for Surveillance

5. FUNDING NUMBERS

6. AUTHOR(S) Bradley John Williford

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number: N/A.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE
A

13. ABSTRACT (maximum 200 words)

Highly mobile, maneuver units require the ability to rapidly provide perimeter defense for their assets. Remote sensors, combined with wireless networks and Smartphones offer a means to reduce manpower impacts of perimeter surveillance. The unit can deploy sensors around their perimeter and/or key locations, and use the Smartphone to monitor them. These sensors can be used to detect personnel and vehicles depending upon the sensors’ capabilities.

To demonstrate this, a Smartphone running the Android 2.3 OS and various sensors manufactured by Phidgets, Inc., are used to develop a real-time surveillance system. The system capabilities include wireless transmission of data and detection of vibration, movement, infrared motion, and sound. The limitations of our study are that Phidgets sensors rely on external power, are not weather-resistant, and have to be plugged into a control board to operate. A fully functional system designed to support the needs of maneuver units in virtually any operating environment would enhance the unit’s capabilities and security.

14. SUBJECT TERMS Android, sensors, Phidgets, Wi-Fi, wireless, surveillance

15. NUMBER OF PAGES
69

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

MOBILE PHONES COUPLED WITH REMOTE SENSORS FOR SURVEILLANCE

Bradley J. Williford
Lieutenant, United States Navy
B.S., University of West Florida, 2005

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
March 2012

Author: Bradley J. Williford

Approved by: Gurminder Singh
Thesis Co-Advisor

John Gibson
Thesis Co-Advisor

Neil Rowe
Second Reader

Peter J. Denning
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Highly mobile, maneuver units require the ability to rapidly provide perimeter defense for their assets. Remote sensors, combined with wireless networks and Smartphones offer a means to reduce manpower impacts of perimeter surveillance. The unit can deploy sensors around their perimeter and/or key locations, and use the Smartphone to monitor them. These sensors can be used to detect personnel and vehicles depending upon the sensors' capabilities.

To demonstrate this, a Smartphone running the Android 2.3 OS and various sensors manufactured by Phidgets, Inc., are used to develop a real-time surveillance system. The system capabilities include wireless transmission of data and detection of vibration, movement, infrared motion, and sound. The limitations of our study are that Phidgets sensors rely on external power, are not weather-resistant, and have to be plugged into a control board to operate. A fully functional system designed to support the needs of maneuver units in virtually any operating environment would enhance the unit's capabilities and security.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Small deployed units need portable, disposable, and simple ways to provide them with real-time warnings of nearby entities. Smartphones and inexpensive wireless sensors can accomplish this task. The unit can setup the sensors around their perimeter and/or key locations, and use the Smartphone to monitor them. When the unit relocates, they can collect or leave the sensors in place since they are disposable, and deploy new ones in the future still utilizing the same Smartphone. These sensors can be used to detect personnel, vehicles, and/or illicit activity around where the sensors are deployed.

Potential problems with this study are numerous. The sensors’ detection ranges could be too small. The sensors could be too sensitive or insensitive which would affect their reliability. Their wireless transmission range could be very limited. Their power supply might not last long enough. Environmental factors could hinder their performance. This study was undertaken to determine the viability of implementing a low-cost personnel detection system and assessing the impact of these limitations on that capability.

This was conducted using an Android Smartphone and sensors produced by Phidgets, Inc., because they are inexpensive and can provide some initial answers to some of the questions surrounding this research. The experiments were conducted in controlled environments indoors and outdoors. A baseline of the sensor behavior was established, followed by various experiments to determine the detection capabilities of the sensors.

A. OBJECTIVES

Our primary objective in this research area is to determine the feasibility of using a Smartphone to wirelessly communicate with remote sensors. Our secondary objective is to determine the capabilities of various sensors produced by Phidgets, Inc. This work will show the usefulness of this type of sensor system. Our objective is not to endorse any particular type of sensor or Smartphone. It is to show that a Smartphone can be used to display real-time remote sensor information to a user, and that low cost sensors have surveillance applications.

B. ORGANIZATION

Chapter I provides an introduction to the possible use of a remote sensor system utilizing a Smartphone to display information. Smartphones are commonly used and their operation is fairly easy. This chapter also discusses the primary objective of this research and the possible applications of this sensor system.

Chapter II describes the Phidgets control board and its capabilities. The various sensors, and their capabilities, are also discussed. Initial testing of the sensors and some previous research involving some of the sensors is discussed. The Google Nexus One is described in this chapter.

Chapter III describes the overall system design and data flow from the sensors to the Smartphone. The sensor control board and phone settings to allow wireless communication are also described. The implementation of the Android application is discussed. The application consists of a single “class” with multiple “inner classes” and utilized the Android API supplied by Phidgets, Inc.

Chapter IV discusses the deployment of the sensor system. The experiment methodologies are described, and the baseline sensor readings are discussed. The results of the experiments are presents in graph form. The experiments were conducted in indoor and outdoor environments with a person walking by the sensor field and creating more forceful ground impacts by stomping his/her foot. It concludes with conclusions regarding sensor performance and applications.

Chapter V is a summary of the thesis with conclusions about the sensor system. It also provides possible avenues for enhancement to the system. It describes this author’s thoughts about the future developments this system will need before it can be utilized in any real-world applications.

THIS PAGE INTENTIONALLY LEFT BLANK

II. SENSORS AND SMARTPHONE

This chapter provides an overview of the technology used in this research. We discuss the specifications of the specific technology used so as to give a frame of reference by which to compare other sensors used in related research. The characteristics of the sensors used in our research are also discussed to provide a baseline of sensor detection performance. We also discuss the specifications of the phone used in this research to establish a baseline of testing platforms.

A. EXTERNAL SENSORS AND SENSORS CONTROL

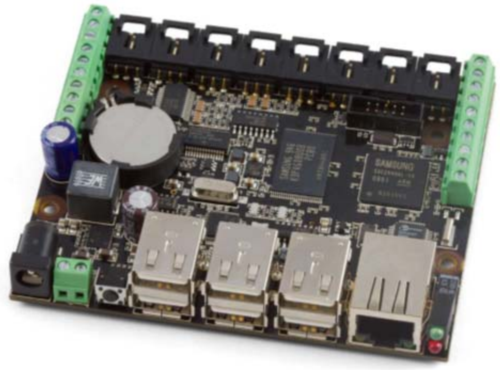


Figure 1. 1072 PhidgetSBC2. (From[4]).

We used the 1072 PhidgetSBC2 sensor control board made by Phidgets, Inc., in our research. It is their most developed model, supporting Ethernet and Wi-Fi interfaces. The PhidgetSBC2 is a Single Board Computer with an integrated PhidgetInterfaceKit, an Ethernet port, and 6 USB ports. The USB ports allow the use of more advanced Phidget sensors to pass information over a network [4]. The network connection gives any connected sensors a substantially increased stand-off range from the user. The PhidgetSBC2 also supports a Wi-Fi adapter, which greatly increases the utility of Phidgets sensors and is a significant reason for its selection for our research.

The PhidgetSBC2 has the following capabilities, as listed in the product manual [4]:

- Provides easy-to-use interface for running custom applications.
- Operates autonomously without a graphical interface or remote connection.
- Operates as an embedded computer running Debian GNU/Linux and provides full shell access via a built-in SSH server.
- Provides access to the full Debian package and all the standard command line tools found in a Linux system.
- Provides integrated PhidgetInterfaceKit 8/8/8 allows connection to devices using any of the 8 analog inputs, 8 digital inputs, and 8 digital outputs.
- Provides a generic way to interface a PC with an assortment of sensors, and operates the same way as an external PhidgetInterfaceKit.
- Provides analog inputs used to measure continuous quantities, such as position, pressure, temperature, etc.
- Provides digital inputs used for detection states of push buttons, switches, relays, logic levels, etc.
- Provides digital outputs can be used to drive LEDs and control devices.

The on-board operating system for the PhidgetSBC2 is a Custom Linux Distribution of Debian, created by using Buildroot, that supports C/C++ and Java programming languages. The use of the configuration GUI is supported by an internet browser.

For remote operation, the following operating systems and languages are supported: Windows 2000/XP/Vista/7, Windows CE, Linux, Mac OS X, VB6, VB.NET, C#.NET, C++, Flash 9, Flex, Java, LabVIEW, Python, Max/MSP, and Cocoa [4].

The following are the specifications of the 1072 PhidgetSBC2 from the product manual:

CPU:	Samsung S3C2440
Core:	ARM920T
Speed:	400MHz
Flash Memory:	512MB
SDRAM:	64MB
Boot time:	30 Seconds
Ethernet:	10/100baseT
USB:	6-Port Full Speed
Operating Temperature:	0 - 70°C
Power Input:	6-15VDC
Power Consumption:	1.2 watt base /w Ethernet
Per additional USB device:	2.5 watt Max

Table 1. 1072 Specifications. (After[4]).

B. 1056 PHIDGETSPATIAL 3/3/3

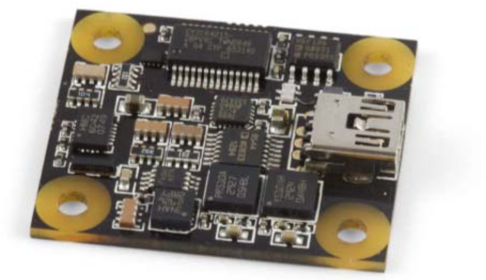


Figure 2. 1056 PhidgetSpatial 3/3/3. (From[5]).

The Phidgets 1056 PhidgetSpatial 3/3/3 is 3 sensors in one. It combines a 3-axis accelerometer, 3-axis gyroscope, and 3-axis magnetometer into one unit. The 1056 Phidget-Spatial has the following capabilities as listed in the product manual [5]:

- Accelerometer can measure up to ±5 gravitational units for dynamic and static acceleration, which is change in velocity and the gravity vector, respectively.
- Gyroscope can measure angular rotation up to ±400° per second.
- Magnetometer, or compass, measures the magnetic field up to ±4 Gauss. It reports the sum of all magnetic fields that are acting on it, not just the Earth's magnetic field.

For this research it will not be a concern, but every 2 minutes the magnetic field data is unavailable for 28ms while the compass performs an internal calibration [4]. All 3 of the measurements were internally calibrated at the factory when it was built, so no calibration was required on our part.

The following are the 1056 PhidgetSpatial specifications as described in the product manual:

Compass	
Resolution:	400µG Minimum
Offset (°) from North:	2° Typical
Gyroscope	
Measurement Range:	±400 °/s
Resolution:	0.02 °/s
Drift / minute:	4° Typical
Typical error over rotation @ 1g:	2mg
Accelerometer	
Acceleration Bandwidth @ 1ms sample rate:	110 Hz
Measurement Range (XYZ Axis):	±5g (49 m/s^2)
Axis 0 Noise Level (X Axis):	300µg standard deviation (σ)at 128 samples/second
Axis 1 Noise Level (Y Axis):	300µg standard deviation (σ)at 128 samples/second
Axis 2 Noise Level (Z Axis):	500µg standard deviation (σ)at 128 samples/second
Acceleration Resolution:	230µg
PhidgetSpatial 3/3/3 board	
Data Rate:	4ms to 1000ms per sample 16ms to 1000ms over the Webservice
Min/Max USB Voltage:	4.75 - 5.25 VDC
USB Current Specification:	45mA max
USB Speed:	Full Speed (12Mbit)
Operating Temperature:	0 - 70°C

Table 2. 1056 Specifications. (After[5]).

Initial testing verified that the accelerometer is very sensitive. While sitting on a desk, it can detect the tapping of a finger or the slightest disturbance of the desk. The gyroscope is very sensitive, as well, as indicated by the specification; the slightest change in orientation is easily detected. The magnetometer is fairly sensitive, but only at close distances; it can detect a pen being moved around it within 5 inches.

At distances greater than 1-2 feet, it would require a substantial magnetic field emission for the sensor to detect.

Below is a figure that illustrates the values for each axis of the accelerometer with the given orientation, as shown in the product manual:

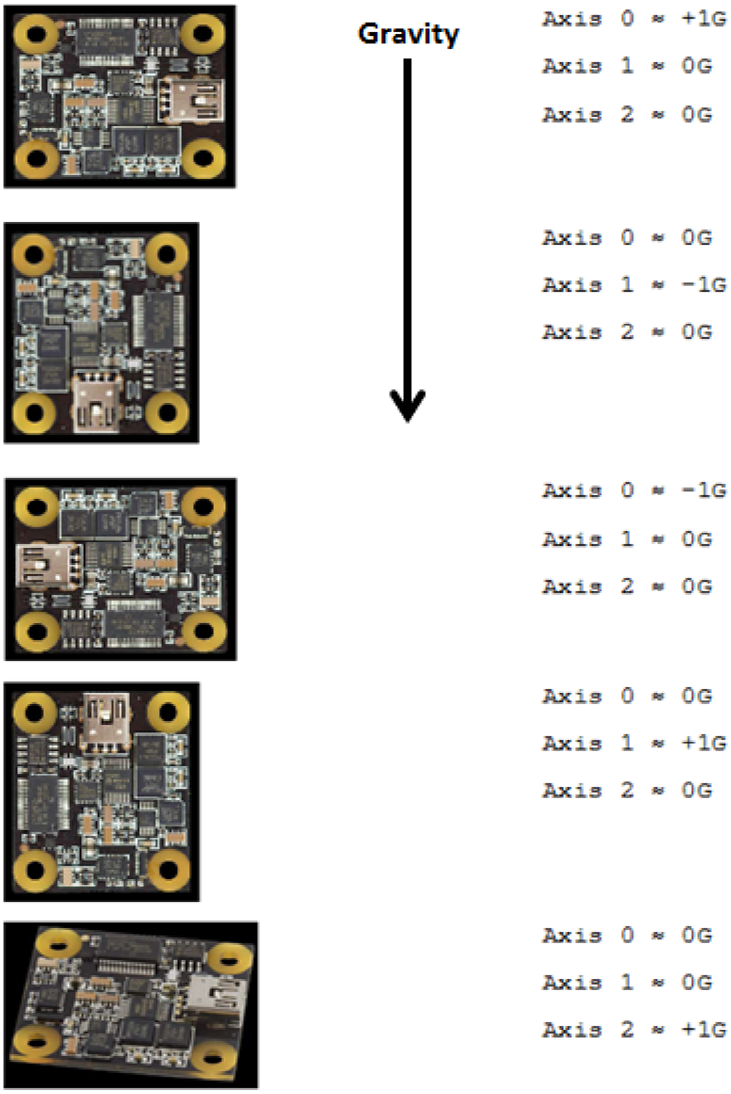


Figure 3. 1056 axis orientation. (After[5]).

C. 1133 SOUND SENSOR

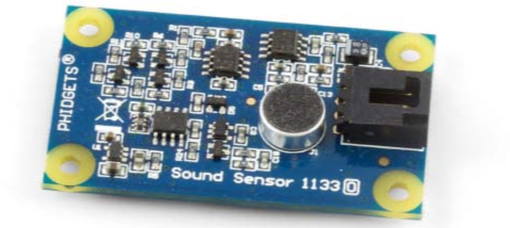


Figure 4. 1133 Sound Sensor. (From[6]).

The Phidgets 1133 Sound Sensor is exactly what its title states, it is not a microphone as it measures sound pressure, in decibels (dB), from 50dB to 100dB in 100Hz to 8 kHz frequency range (human speech is in the frequency range of 60Hz to 7 kHz); it does not generate electronic signals that mimic the detected sound, as would be done by a microphone. It can be used to approximate how loud a detected sound is and connects directly to one of the analog inputs built into the PhidgetSBC2 [6].

Measuring sound pressure is very complex and depends on numerous factors, which is beyond the scope of this research. However, the following formula, from the product manual, is used to translate the sensor value into a sound pressure level:

$$1\,kHz\ \text{tone (db)} = 16.801 * \ln(SensorValue) + 9.872 \quad \text{(After[6])}.$$

This formula is only truly accurate for a 1 kHz pure tone. The outputted value can vary up to ±20 raw Sensor-Value in a stable pressure environment. To compensate for this, the average of sensor readings was used for the detection data.

According to the product manual, the response time of the sensor is 1.40ms when the sound source is 30cm away. After the sound source has stopped emitting, the sensor output will return to normal, but not immediately. Below are the specifications of the 1133 Sound Sensor, taken from the product manual:

Current Consumption:	8.5 mA
Resolution:	30mV/dB
Input Sound Range:	50 to 100 dB
Error (@ 1000Hz):	±3dB
Input Frequency Range:	100Hz to 8 kHz

Table 3. 1133 Specifications. (After[6]).

Initial testing showed that mechanical movement that generates sound is easily detected, as is human speech. Music was not detected very well unless its relative volume was turned up very loud. As expected, lower frequency sounds create the most change in sound pressure and are the easiest to detect with the sensor. The gradual return to a “resting” output after a loud sound pressure was detected, as described above, was observed.

D. 1111 MOTION SENSOR

Figure 5. 1111 Motion Sensor. (From[7]).

The Phidgets 1111 Motion Sensor uses Infrared to detect motion. It detects changes in infrared radiation across its detection zone, i.e., objects of a different temperature from the background are detected as they move across the sensor's field of view. Because people emit their own body heat, which is generally different than their surroundings, this sensor was ideal for detecting people walking by it [7]. Below is a graphic from the product manual that shows the concept of how detection works:

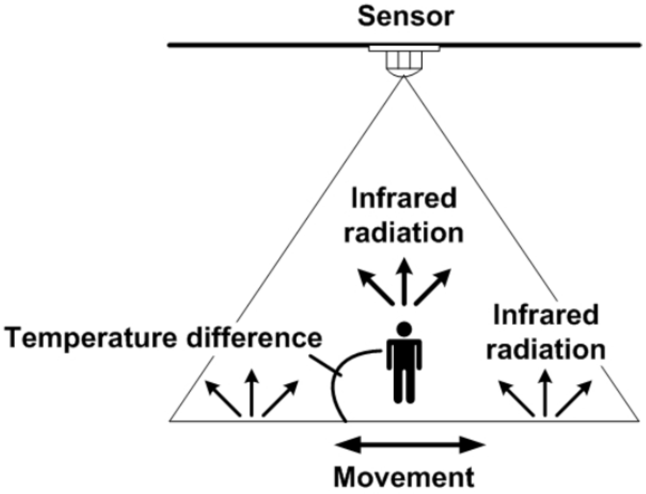


Figure 6. Illustration of 1111 detection field. (From[7]).

These are the device specifications from the product manual:

Current Consumption:	15µA
Output Impedance:	1K ohms
Supply Voltage:	4.75VDC to 5.25VDC
Motion Sensing Module:	Panasonic AMN23111
Horizontal detection cone:	38°
Vertical detection cone:	22°
Rated detection distance:	5 meter - for a human body (size: 700mm x 25mm) moving between 0.5 to 1.5 meter/second
Operating temperature:	-20°C to 85°C

Table 4. 1111 Specifications. (After[7]).

Previous test results by Ahren Reed, of California Polytechnic State University provide the following:

> Experiments show that at rest, the raw values are around 2000. When the sensor picks up motion the raw values drop to as low as 99, which indicates the most amount of motion. The sensor value can also increase to indicate movement, up to around 3000. Therefore, a threshold of +/- 1000 from the "at rest" measurement of 2000 is a safe bet.
>
> Analysis: A threshold value must be selected when polling the sensors. This may be between 500 and 1000 depending on what accuracy the user chooses. The higher the threshold the closer an object must be in order to trigger a detection event. The PhidgetInterfaceKit can be set up to detect interrupts from the sensors; however this must also be used in conjunction with a user programmable threshold. Otherwise the device may report numerous false positives if a low threshold is used.
>
> Ranges: Tests were performed with the sensor lying on the ground (cone pointing up), and walking nearby. When a human subject walks directly toward and then away from the sensor, the range is only 1 yard. When someone is walking

in a circle around the sensor, it has an accurate range of 3 yards. Detection past 3 yards depends on the threshold value that is used.

The sensor can detect someone at 4 yards if the threshold is set at 500. If the threshold is set to 1000 (meaning the Raw Values must change by over 1000 points) there is no detection of subjects at this range. The sensor is not able to detect someone who is standing still because it operates by comparing changes between its detection sectors. If someone stands still, there is no visual change detected. [10]

E. 1104 VIBRATION SENSOR

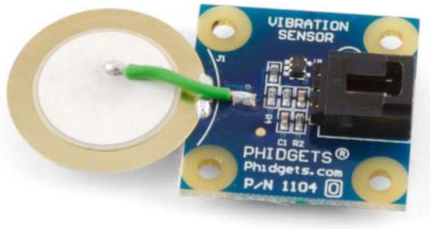


Figure 7. 1104 Vibration Sensor. (From[8]).

The 1104 Vibration Sensor functions by buffering a piezoelectric transducer. Voltages are generated when the piezoelectric element is strained from bending, which displaces the transducer from its normal state. The product manual states that if it is suspended from its mounting points, it will vibrate "in free space." It is not meant to measure precise values of vibration or acceleration, only to detect an impulse of movement or that there is vibrations present [8].

The follow are the 1104 Vibration Sensor specifications from the product manual:

Current Consumption:	400µA
Output Impedance:	1K ohms

Table 5. 1104 Specifications. (After[8]).

Initial testing with the sensor mounted in “free space” confirmed that the vibration detection is not very good, an observation consistent with the Ahren Reed’s tests. However, if the sensor disk itself is mounted to an object, it detects the slightest vibration. During field testing, better results might be gained by attaching the sensor disk to something placed on the ground where people would walk.

F. 1040 PHIDGETGPS

Figure 8. 1040 PhidgetGPS. (From[9]).

The 1040 PhidgetGPS is just a basic GPS unit that comes with an antenna. It is actually the same physical dimensions as the 1056 PhidgetSpatial 3/3/3 for the purpose of mounting it on top of the 1056. The following are characteristics of the PhidgetGPS from the product manual [9]:

- Good GPS signal required to calculate its current position
- From a dead battery start-up, or "cold start", it can take up to 5 minutes to acquire a position fix
- With a charged battery, a position fix is generally less than 5 seconds depending on distance traveled or elapsed time since the last position fix
- Has a 28dB active GPS antenna with a 50cm RG-174 cable
- Antenna shell has a magnet inside for attachment to metallic surfaces

The product manual states that being near other electronics, like a Wi-Fi antenna, can reduce the performance of the GPS antenna. This is important to note, since this research used a Wi-Fi link for the Control Board to phone communication, as well as the 1056 having a magnetometer that may be thrown off by the antenna's magnetic mount. Also of note is that the position accuracy listed in the specifications, and included below, will be affected by tall buildings, electronic interference, and weather conditions [9].

Accuracy	
Position (best case):	2.5m CEP
Velocity:	0.1m/s
Timing:	300ns
Position Updates per second:	10
Max Altitude and Velocity:	18,000m @ >515m/s
GPS Sensitivity:	Tracking -161dBm
Re-acquisition (hot start):	< 1s
USB Voltage:	4.35-5.25VDC
Battery Backup run-time:	1 month
Battery charge time:	24 hours
USB Current Consumption:	50mA
Operating Temperature:	0 - 70°C

Table 6. 1040 Specifications. (After[9]).

Initial testing shows that it performs at the same level as other Commercial Off the Shelf (COTS) handheld GPS units.

G. GOOGLE NEXUS ONE

Figure 9. Google Nexus One. (From[1]).

The Smartphone we used to convey the information that the sensors collected was the Google Nexus One, made by HTC specifically for Google. It runs the Android 2.3.6 operating system, which was updated from Android 2.2. The only phone-relative requirements for this research were that it use the Android 2.3 operating system and be Wi-Fi capable. After exploring the Nexus One, we discovered that all Android devices currently produced do not support Ad-Hoc Wi-Fi connections in their default setting. They can be "rooted" and modified to do so, but we decided to just use a wireless router to bridge the connection between the Phidgets and the phone. See the Appendix for full list of specifications for the Google Nexus One [1].

H. PREVIOUS RESEARCH

Previous work by Peter Young explored the idea of utilizing one or more iPhones to deploy a distributed sensor grid for team operations [12]. His work was a success in that it showed the phone was capable of detecting footsteps using the internal accelerometer and microphone. He also showed that when two phones were used, and a person passed between them, the direction and velocity could be approximated. His research was slightly different, in that he was using the internal phone sensors to detect activity instead of using the phone to display information from remote sensors.

Previous work by Neil Rowe, Ahren Reed, and Jose Flores explored the idea of utilizing non-imaging sensors to detect suspicious behavior [11]. They utilized infrared motion and sound sensors in their research, which was also a success. They were able to determine changes in speed and direction by applying mathematical calculations to data from deployed sensors. The infrared motion and sound sensors they used are from the same type as the ones used in this research. Again, their research was slightly different, in that they were not relaying the sensor data to a mobile device for display. Their findings, along with Peter Young's, are very applicable to this research.

Unattended ground sensor systems traditionally rely on seismic, acoustic, and non-imaging sensors. Work done by Peter Boettcher and Gary Shaw utilized multiple acoustic sensors to determine a bearing to the "target" using time-difference of arrival algorithms [13]. This could also be applied to the 1133 sound sensor discussed above. Shih, Wu, and Chen showed that an automated wireless surveillance systems using cameras could be deployed successfully [14]. Though this thesis focuses on non-imaging sensors, the automation approach is another avenue that could be implemented in future implementations. The Department of

Defense did some work involving the Internet and Web-centric fusion sensor information [14]. Their initiative described large arrays of local multi-sensor systems transmitting data over the Internet to provide real-time imagery, environmental, targeting, and mission planning information. The goals of their initiative are beyond the scope of this thesis, but the utilization of web services is very applicable since the 1072 control-board can transmit sensor data via a wired or wireless network.

I. CHAPTER SUMMARY

This chapter discussed the hardware used in this thesis. The 1072 PhidgetSBC2 is a very capable device that has multiple applications. The 1056 PhidgetSpatial 3/3/3 has a three axis accelerometer, three axis gyroscope, and magnetometer; we are able to detect very small amounts of movement or vibration using it. The 1133 Sound Sensor detects lower frequency sounds very well, and the 1111 Motion Sensor detected movement out to a range of four yards. We also learned that Android devices do not support Ad-Hoc Wi-Fi in their default setup, therefore a wireless router was needed to support communication between the PhidgetSBC2 and the Google Nexus One.

The next chapter discusses how the phone and sensor system was setup, and the implementation of the phone and sensors to communicate data across the wireless link to the user in a usable way.

III. SYSTEM IMPLEMENTATION

This chapter provides an overview of the system that we built for our research. The physical connections of the sensors and control board, the software settings of the control board, and the software settings of the phone are discussed in this chapter. The programming structure of the Android application is also discussed.

A. SYSTEM DESIGN

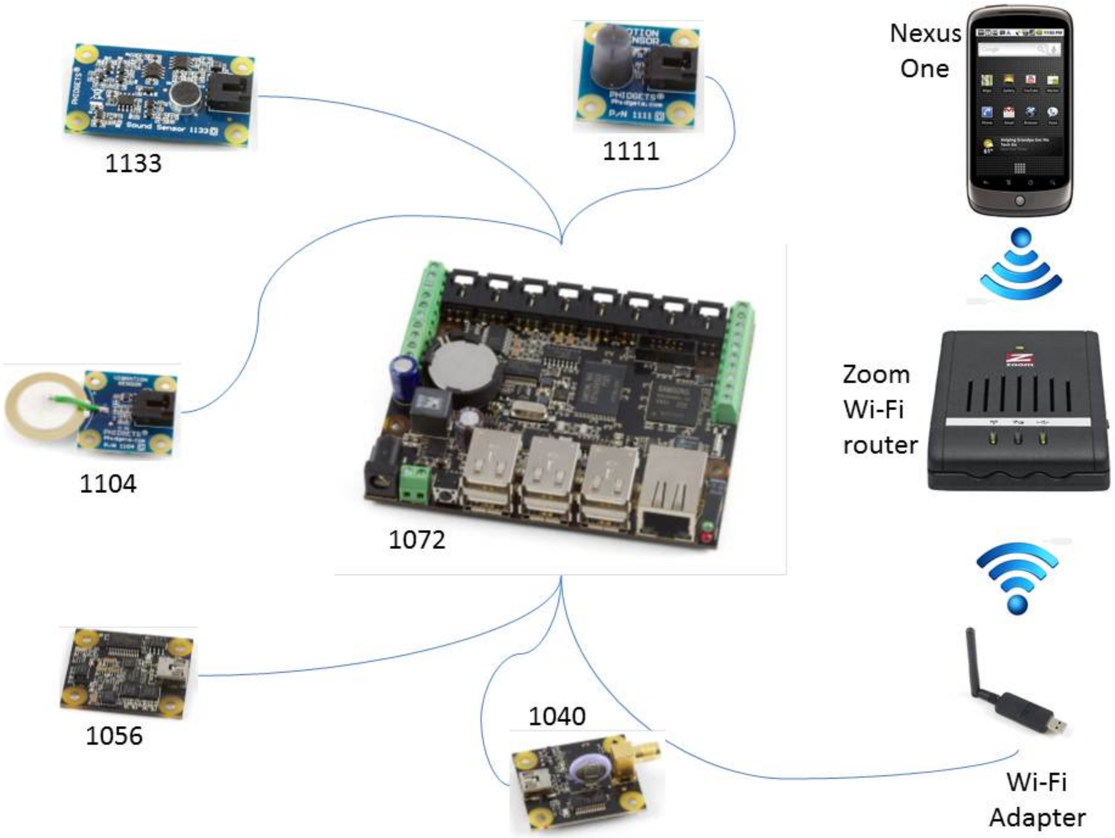


Figure 10. System Design. (After [4–9]).

Figure 10 shows the overall design of the sensor system. The PhidgetSBC2 1072 Control Board is the heart of the system. It allows various types of sensors to be plugged into it. An important characteristic of the system is that it is not weather resistant, in its current form, in any way. As shown in Figure 10, the control and all the sensors are exposed circuit boards. The 1056 PhidgetSpatial 3/3/3, 1040 PhidgetGPS, and Wi-Fi Adapter are plugged into its USB ports. The 1111 IR Motion, 1133 Sound, and 1104 Vibration sensors are plugged into its analog input ports. The Zoom 3G Wireless-N Travel Router serves as a Wi-Fi network bridge between the Nexus One phone and the Wi-Fi Adapter. Sensor data is passed from the sensor to the control board, packaged in the appropriate object, and then passed over the network connection to the device for display to the user. Figure 11 illustrates the data flow.

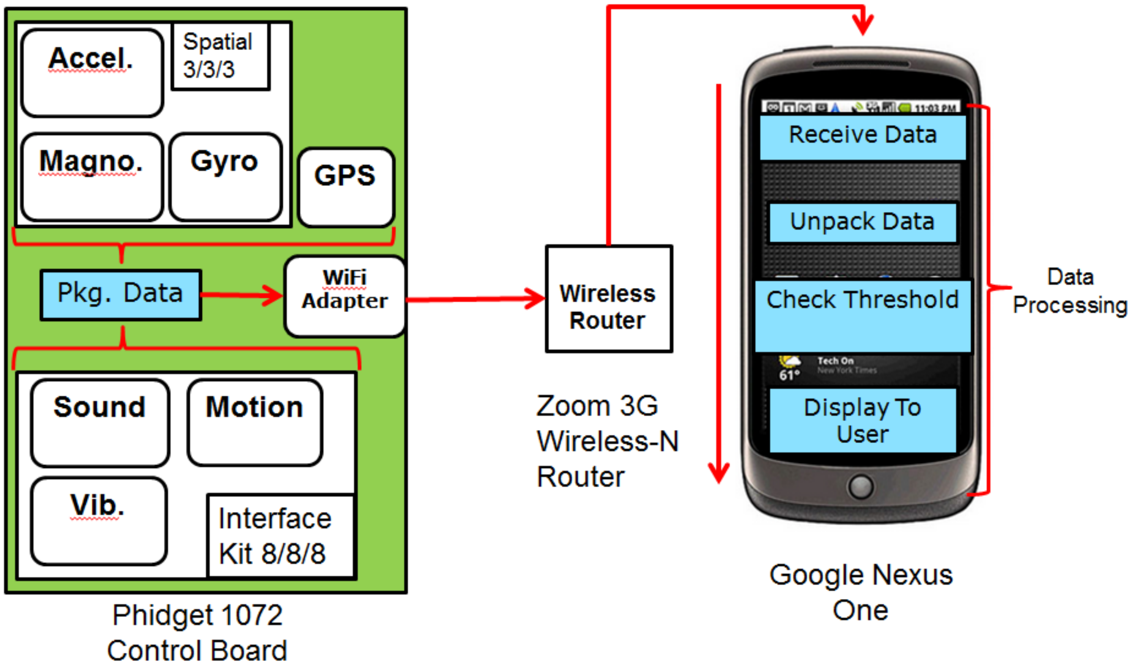


Figure 11. Data flow.

The 1056, 1040, and Wi-Fi adapter can be plugged into any of the USB ports. Our Android application does not depend on them being plugged into the same port each time.

That is, however, not the case for the 1133, 1104, and 1111 sensors. They must be plugged into the same analog ports every time or the data displayed in the application will not match up with the labeled data fields. The port numbering starts at zero from the side which has the Ethernet port. 1104 is plugged into port 0, 1111 is plugged into port 1, and 1133 is plugged into port 2.

The orientation of the sensors is also important for all the sensors except the 1040 GPS, which has an external antenna. The following list outlines the desired orientation.

- 1111 IR Motion – Plastic cylinder facing up with the circuit board parallel to the ground.
- 1133 Sound – Speaker facing up or toward the desired direction of sound detection.
- 1104 Vibration – Metal disk making contact with or attached to object that would cause a vibration when stepped on or moved.
- 1056 Spatial 3/3/3 – If resting on the ground or buried, it should be parallel to the ground or perpendicular to the pull of gravity, but not required. Orientation does not matter if attached to an object as movement would cause drastic changes in acceleration and physical orientation.

The Zoom 3G Wireless-N Travel Router was used after discovering that Android devices do not support ad hoc network connections in their factory configuration. The Zoom is just a basic wireless router that has an internal battery and 3G-client capability if a SIM card with cellular subscription is inserted. For our research, the 3G capability was not required, only the wireless router capability. Experimentation with signal reception determined the router to have decent signal strength for about 200 feet for line-of-sight. However, the signal strength dropped drastically when a concrete wall was between it and the Nexus One phone, which reduced range to about 20-25 feet. For this research, the overall range between the phone and the sensors is greater than that of an ad hoc network connection between the sensors and the phone. An ad hoc network connection between the phone and the sensors is the desired system design to reduce the amount of equipment needed for the system to work.

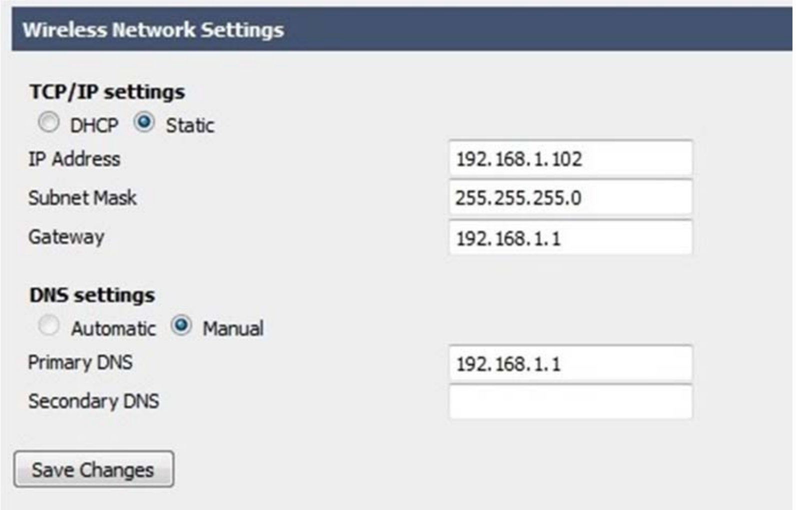
B. CONTROL BOARD/PHONE SETTINGS

The initial setup instructions for the 1072 control board are listed in the 1072 product manual [4]. In this section we discuss the required steps to enable the 1072 to pass information to the Nexus One phone.

Initially, a laptop computer was required to communicate with the 1072 control board to change some of the settings for our purposes. The USB drivers for the Nexus One were also installed on the laptop so the application could be installed on the phone. The Phidget21 Installer was installed, which is available on the Phidgets webpage (http://www.phidgets.com/drivers.php) under the "Drivers" tab. It contained all the necessary drivers and software, specifically the Phidgets Control Panel, for our laptop to communicate with the 1072 control board. After it was installed, we were able to open the Phidget Control panel, which provides information pertaining to the PhidgetSBC2 control board and the integrated PhidgetInterfaceKit 8/8/8. "Double-clicking" the PhidgetSBC2 device name in the PhidgetSBC tab opened a web browser corresponding to the IP address of the control board and loaded the login page to gain access to the control board settings. The first time the login page was accessed, it displayed a “Set System Password” window, which I set to “PhidgetsThesis” for the “admin” username.

The control-board settings windows are very similar to the windows typically found in routers for changing the router settings. There are tabs at the top for all the different categories of settings. Of interest to us were the System and Network tabs. The product manual stated to update the 1072 to the latest firmware and install or update any available development packages, which in our case was the Java Support package. These options were located under the Systems -> Packages tab. To do this, we connected the 1072 to the Naval Postgraduate School (NPS) Wi-Fi network to gain access to the Internet, after setting net necessary configurations under the Network tab. After updating the firmware and installing the Java Support package, we then continued additional configurations associated with the Network tab.

The Network -> Wireless tab was used to configure the control board to access the PhidgetsNet wireless network to communicate with the laptop and phone. Connecting to a Wi-Fi network was just like most any other wireless host computer. We selected the appropriate network from the Detected Networks window, entered the associated password, and clicked the “Add this Network” button. To access the NPS wireless intra-net all the settings were set to automatic. After updating the 1072, we deleted that network from the saved networks list so it would not reconnect to it by default. We then connected the 1072 to the PhidgetsNet network using the following settings:



○ DHCP ● Static

IP Address 192.168.1.102

Subnet Mask 255.255.255.0

Gateway 192.168.1.1

○ Automatic ● Manual

Primary DNS 192.168.1.1

Secondary DNS

Figure 12. 1072 Wireless Network Settings.

This made the 1072 IP address static, which is required for the phone application to receive data from the sensors. After this step, there were no additional changes needed to the settings of the 1072. The IP address for the phone was setup as static, also. This was done by navigating through the following phone menus:

1. Settings
2. Wireless & Networks
3. Wi-Fi Settings
4. Silk screen Menu button
5. Advanced

The following screen shot shows the settings we entered:

Advanced

IP address
192.168.1.104

IP settings

Use static IP ☑

IP address
192.168.1.104

Gateway
192.168.1.1

Netmask
255.255.255.0

DNS 1
192.168.1.1

DNS 2

Figure 13. Google Nexus One Wireless Settings.

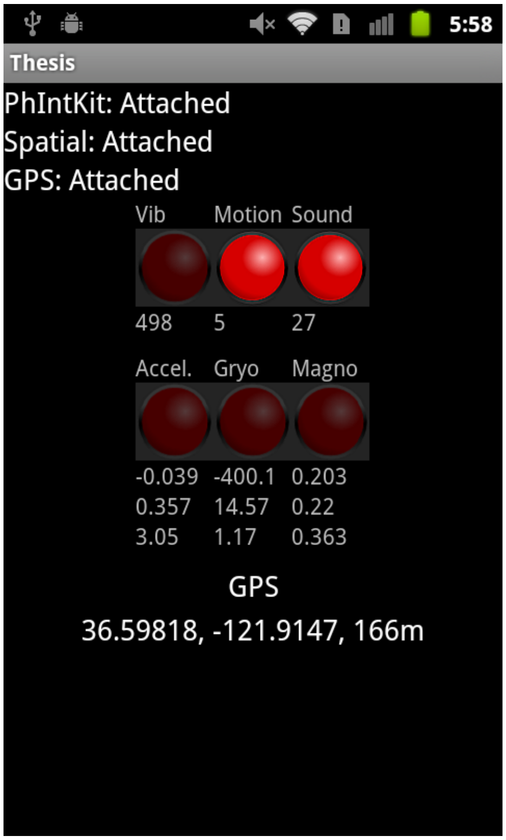
After this step, we connected to the PhidgetsNet network and no further changes were needed to the phone settings. The phone and sensors were then ready to pass information across the network.

C. APPLICATION IMPLEMENTATION

The first step in developing the Android application was to develop a suitable User Interface design to effectively indicate to the user any detection of people or activity in the sensor field. After reviewing the Android application examples provided by Phidgets, Inc., we decided to use a similar appearance for our application [2]. We maintained the connection status at the top, for each sensor, to let the user know if a connection to the phone had been established or not. This information is displayed as follows:

- PhIntKit – "Attached" or "Detached".
- Spatial – "Attached" or "Detached".
- GPS– "Attached" or "Detached".

We then labeled each sensor output display along with a simple graphic of a red light being ON or OFF, with ON representing a detection by the corresponding sensor. Under each red light graphic we placed the raw output information from each sensor. For the GPS information, we placed a label and displayed the position under it. Below is a screenshot of the application running on the Google Nexus One:



PhIntKit: Attached
Spatial: Attached
GPS: Attached

Vib	Motion	Sound
498	5	27

Accel.	Gryo	Magno
-0.039	-400.1	0.203
0.357	14.57	0.22
3.05	1.17	0.363

GPS
36.59818, -121.9147, 166m

Figure 14. Screenshot of the application.

When building our application, we started with the InterfaceKitExample application provided by Phidgets, Inc., as indicated above and modified it to meet our needs [2]. We removed all the objects and functionality we did not want and then used the Phidgets, Inc., Android API documentation to determine what we needed to add to give the application the desired functionality. The application is made up from one class and 6 inner-classes, which serve as handlers of events. The main class contains the code for creating all the objects, linking them to the interface, sending the events to the handlers, and establishing the network connection with the control board. The handler classes implement the actions for the following:

- Attach/detach of the PhidgetInterfaceKit
- Attach/detach of the PhidgetSpatial 1056
- Attach/detach of the PhidgetGPS 1040
- Sensor detections by the 1104, 1111, and 1133
- Sensor detections by the 1056
- Position changes of the GPS sensor

When the PhidgetInterfaceKit attaches or detaches, the event is handed off to the AttachDetachIntKitRunnable inner-class, which updates the text fields displayed on the screen with the appropriate information. When the PhidgetSpatial sensor attaches, the inner-class, AttachDetachSpatialRunnable, sets the sampling interval to 256 milliseconds. The default rate of 16 milliseconds, a limitation while using the Webservice, was too fast and just created excess data that was not useful. This rate can be set to any multiple of 8, starting at 4 milliseconds up to 1000 milliseconds.

The SensorChangeRunnable inner-class handles SensorChangeEvents, using the sensor index and the raw value output by the respective sensor. The outputted value is of type “integer”, and we used it as is. Index 0 is the 1104 vibration sensor, index 1 is the 1111 IR motion sensor, and index 2 is the 1133 sound sensor. On account of the sensitivity of the sensors, we implemented a threshold to filter out a majority of the false-positive SensorChangeEvents. If a SensorChangeEvent met the desired threshold, the sensor output was displayed in the corresponding text field and the light was changed to ON. We used the following function to create the threshold for all 6 sensor outputs, with x being the desired amount of change to meet the threshold:

$$if(|sensorValue - previousValue| > x)$$

The SpatialChangeRunnable inner-class handles the information in a similar way, but the information is passed as an array of SpatialEventData objects instead of simple integers. To extract information from these objects, the following functions were used, returning a corresponding array of type “double:”

- getAcceleration(), for accelerometer data.
- getAngularRate(), for gyroscope data.
- getMagneticField(), for magnetometer data.

The threshold function was implemented on this data as well, but the change was in the thousandths of units, instead of whole numbers as was used for the analog sensors. The PhidgetSpatial 1056 sensors fluctuated faster, but in much smaller increments unless there was an actual detection.

The GPSPositionChangeRunnable inner-class handles the GPSPositionChangeEvents, similar to the SpatialEventData. It uses get() functions to access the Latitude, Longitude, and Altitude values. The latitude and longitude are returned in decimal format. Altitude is returned in meters. All the GPS get() functions return a double value, so we converted the altitude value into an integer before displaying it to the user.

D. CHAPTER SUMMARY

This chapter discussed the overall system design, the settings of the control board and the phone, and the application implementation to display the data to the user. The system design is very simple, with both the control board and the phone requiring very minimal setup. The system is not weather resistant in any way, so attention must be paid to the weather during use until a more rugged design can be implemented. The desired direct network connection between the phone and control board was not possible due to the lack of ad hoc capability with unmodified Android devices. To overcome this, a basic wireless router was used to bridge the network connection. This chapter also discussed a high-level view of the application implementation to display the sensor data on the phone and alert the user of a detection event.

The next chapter discusses the deployment of the sensors, along with a comparison and contrast of the sensors’ performance. The experiment results are also discussed in the next chapter.

THIS PAGE INTENTIONALLY LEFT BLANK

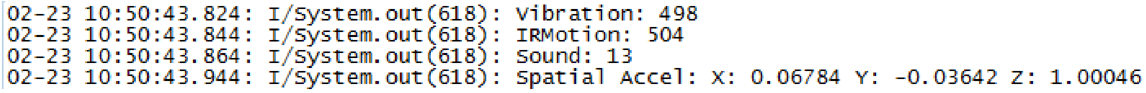
IV. TESTING OF SENSOR SYSTEM

This chapter discusses the sensor deployment and the types of terrain in which the experiments were performed. The results of baseline testing and of the experiments are also discussed in this chapter.

A. SENSOR DEPLOYMENT AND EXPERIMENTATION

We deployed our sensor system in two different environments, one indoors and one outdoors. The indoor environment was in the Naval Postgraduate School (NPS) Network Lab. The lab has a raised floor to support power and network connections, which should provide good vibration levels through the floor to the sensors for testing before moving the sensors outdoors. The outdoor environment was on a hard-packed dirt surface. We believe hard-packed dirt will be a good conductor of vibrations. All experiments were controlled to eliminate unwanted sensor interference. If some form of interference did appear, the experiment was restarted. In the experiments, we were interested in the IR motion, sound, accelerometer, and vibration sensor readings. The gyroscope and magnetometer data is displayed on the phone, but we decided not to use it since the gyroscope and accelerometer both indicate movement. The magnetometer was not useful unless a strong magnetic field was being generated by the passing object. The setup of each experiment was documented with photographs, and the distances were all measured with a standard measuring tape and recorded.

Earlier, we discussed the sensors relaying data to the phone, evaluating that data with a threshold, and then providing the user with a graphical output. While the mobile phone scenario represents the actual use scenario for our experiments, we decided to use a laptop running Eclipse Indigo and an Android 2.3 emulator which enabled easier data-logging and analysis. The laptop was connected using the Ethernet port to the router. For data-logging, each sensor value sent by the control board was displayed in the LogCat window in Eclipse using System.out.println() statements. LogCat is included with the Android add-on package for Eclipse. We created a filter in LogCat to display only System.out logs. Each log contained a date/time stamp, the name of the sensor, and the sensor value. The logs were saved as text files after each experiment and imported into Microsoft Excel using the “space” character (ASKII %20) as a delimiter. After importing into Excel, we sorted the logs by sensor to group each sensor’s data together. A sample of the text file data-logs is shown in Figure 15.



```
02-23 10:50:43.824: I/System.out(618): Vibration: 498
02-23 10:50:43.844: I/System.out(618): IRMotion: 504
02-23 10:50:43.864: I/System.out(618): Sound: 13
02-23 10:50:43.944: I/System.out(618): Spatial Accel: X: 0.06784 Y: -0.03642 Z: 1.00046
```

Figure 15. Sample of data-logs.

For the indoor testing and experiments, we used the building’s power outlets to power the control board and sensors. The sensors were set on the floor in the Network Lab. Once the sensors were positioned we established a baseline for the sensor values. We started the applications and collected logs for four minutes to give us a good baseline of sensor output with no external influences.

The sensor values remained stable within their respective thresholds, and the application did not indicate any movement around the sensors.

For the first indoor experiment, the sensors were flat on the floor two yards from the walkway through the Network Lab. We started the application and allowed approximately 10-15 seconds to pass before a person started to walk through the lab toward the door. The person walked by the sensors with the closest distance of two yards, opened the door, allowed the door to close, and then walked by the sensors again at a distance of one yard when returning to the starting point.

The second indoor experiment was essentially the same, but the 1056 sensor was firmly attached to a twelve inch flathead screwdriver using a zip-tie. The screwdriver was then inserted into a crease in the floor. The theory for this was that the vibrations in the floor would travel up the screwdriver shaft intensifying the amount of vibration at the end where the sensor was attached. Figure 16 shows the attachment of the sensor to the screwdriver.

Figure 16. 1056 sensor attached to screwdriver.

The third and fourth experiments were the same as the first two for setup respectively, but instead of just walking by the sensor, the person walked up to the sensors and stomped the floor a series of times at different distances. After each single stomp, the 1056 sensor was radically moved to mark reference points between each foot stomp to aid in later analysis. First the person stomped at one yard, then at two feet, then one foot, and then moved back to two yards.

The outdoor testing procedures were similar to the indoor tests; however, a 12-volt car battery and power inverter provided power to the equipment. First, all the sensors were flat on the ground and a person walked by at a distance of two yards, paused for three-five seconds, and then walked back at a distance of two feet. The person then walked up and stomped the ground at a distance of one yard and then two yards. These same actions were done with the 1056 sensor attached to the screwdriver and stabbed into the ground approximately four-five inches. Then, all tests were performed again at a distance of three yards.

B. BASELINE TESTING

After importing the data-logs into Microsoft Excel and sorting by sensor type, scatter plots were used to show a graphical representation of the data. All the following sensor measurements are in SensorValue units, which is the raw output by the sensors. In the baseline testing the IR Motion sensor had an average reading of 502 and a standard deviation of 5.7. The Sound sensor had no change and had a reading of nine the entire test. The accelerometer “X” and “Y” axes had average readings of 0.02617 and 0.01777, and both had standard deviations of .00017. The "Z" axis had an average reading of 1.0035 and a standard deviation of 0.00075. When displayed graphically in a scatter plots, the "X" and "Y" axes have fairly stable noise readings. The "Z" axis readings fluctuated much more, as shown in Figure 17.

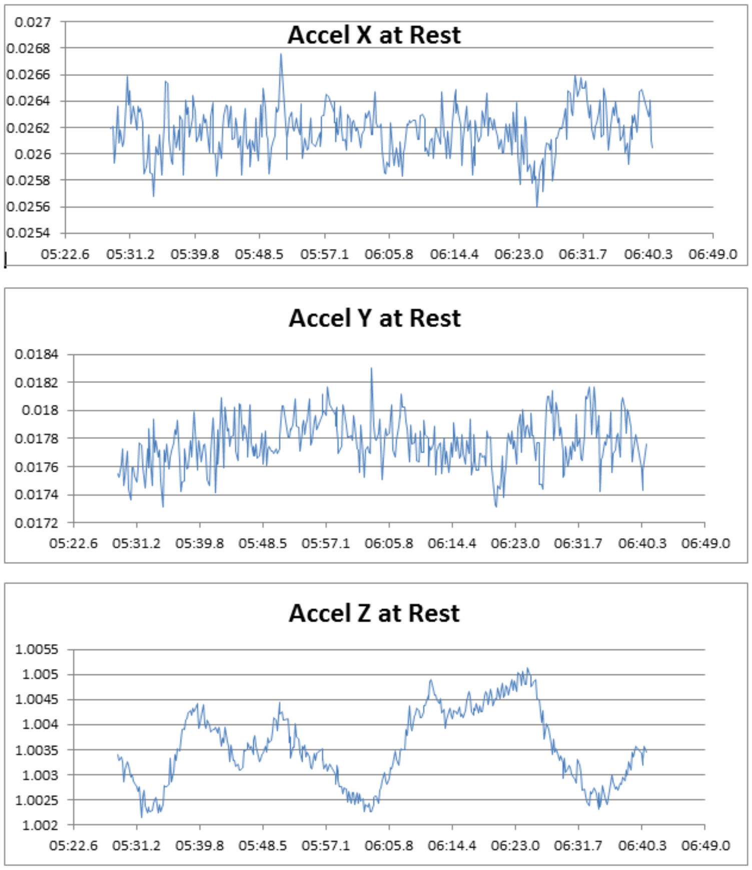


Figure 17. Graphical display of accelerometer data.

After seeing this fluctuation in the "Z" axis, we decided to collect four minutes of sensor readings with no movement to see if there was a pattern. When displayed graphically, it appeared to have a "dirty" sinusoidal curve and is shown in Figure 18. Further investigation within the Network Lab brought our attention to the server that was running, and located about two yards from the sensors.

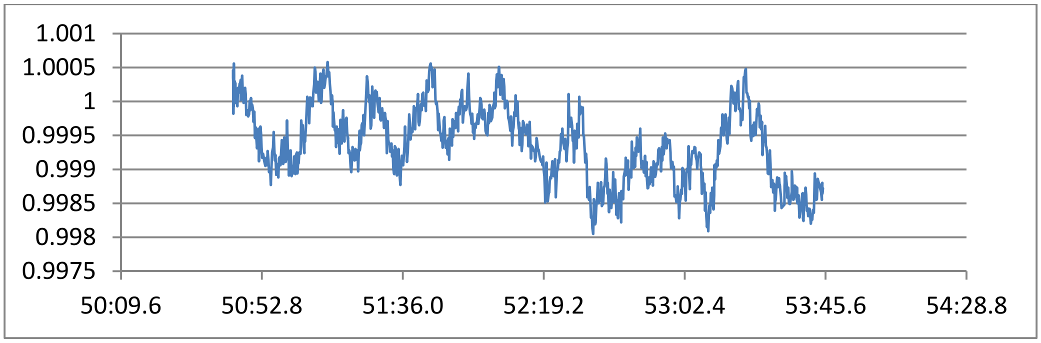


Figure 18. Possible sinusoidal curve.

We then took the sensors outside away from any interference to get a baseline test for comparison. We collected and graphed almost four minutes of sensor readings. The average reading of the "Z" axis on this data set was 0.99624, with a standard deviation of 0.00073. The graph, shown in Figure 19, did not have the appearance of a sinusoidal curve anymore. However, it still had much more fluctuation than the "X" and "Y" axes.

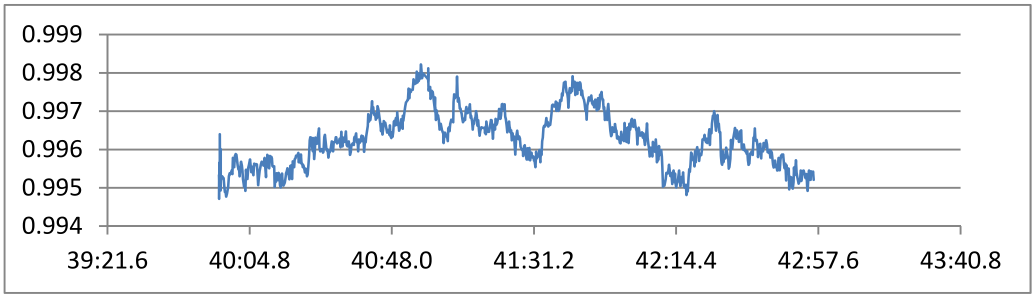


Figure 19. Baseline testing outdoors.

We contacted Phidgets, Inc., to inquire about the disparity in the axes of the sensor. They replied that the "Z" axis sensor is a different type than that of the "X" and "Y" axes, it is normal based on our graphs, and actually performing better than the specifications state.

C. EXPERIMENT RESULTS

The first indoor experiment showed a good detection of movement by the IR motion and sound sensors, but no detection by the accelerometer. No abrupt changes were seen beyond the normal noise level of the accelerometer. The sensor readings for the IR motion and sound are much higher at a distance of one yard, which was expected. These results are shown in Figure 20. The second indoor experiment, which was with the accelerometer attached for the screwdriver, yielded the same results.

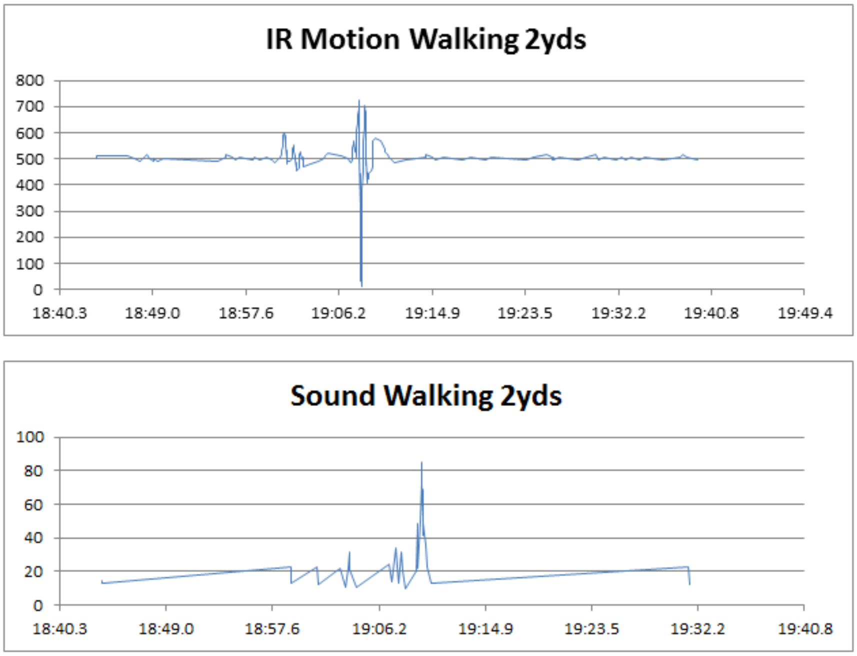


Figure 20. Experiment 1 graph of IR Motion and Sound.

The third and fourth indoor experiments, which had foot stomps at different distances, showed more promising results. All three sensor readings had abrupt changes at all four distances. The IR motion and sound sensor were very distinct with each foot stomp, which is shown in Figure 21. Figure 22 shows the sensor readings of the “Z” axis of the accelerometer during a foot stomp at a distance of two yards while the sensor was flat on the floor. The SensorValue range of readings for this test was 0.00282. Figure 23 shows the “X” axis readings of the accelerometer during a foot stomp at two yards while the sensor was attached to the screwdriver. The SensorValue range of readings for this test was 0.0055. The vibration intensity increased as the distance was reduced to the sensors, as expected. This indicated that a person just walking by did not cause enough vibration for this particular accelerometer to detect. It required a much higher amount of force to distinguish the sensor readings from the normal sensor noise.

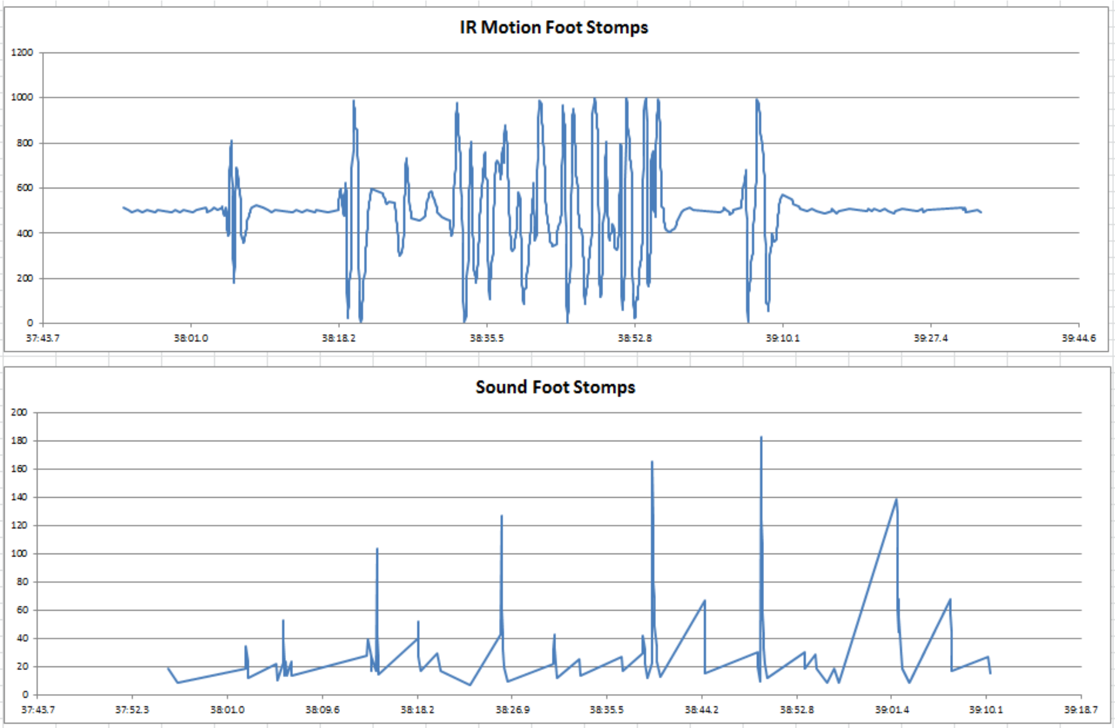


Figure 21. IR motion and sound sensor readings during foot stomps.

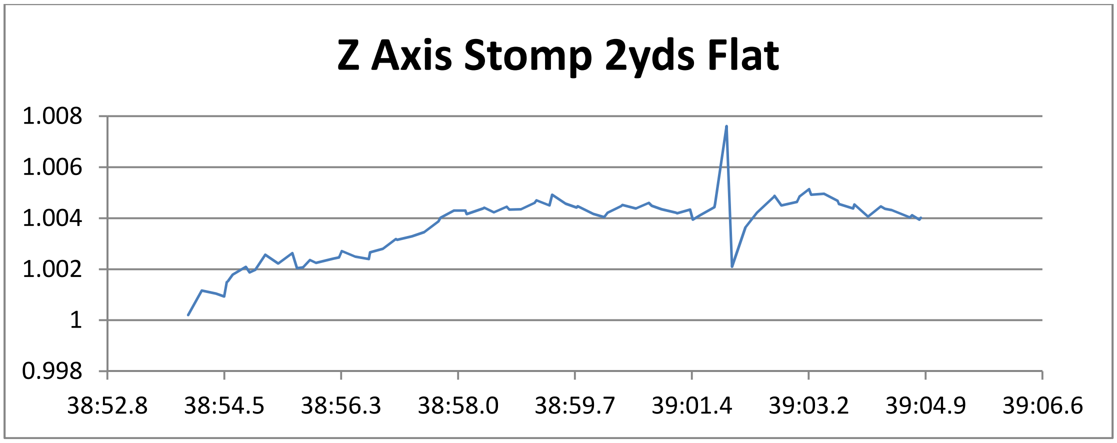


Figure 22. Foot stomp while flat on floor at two yards.

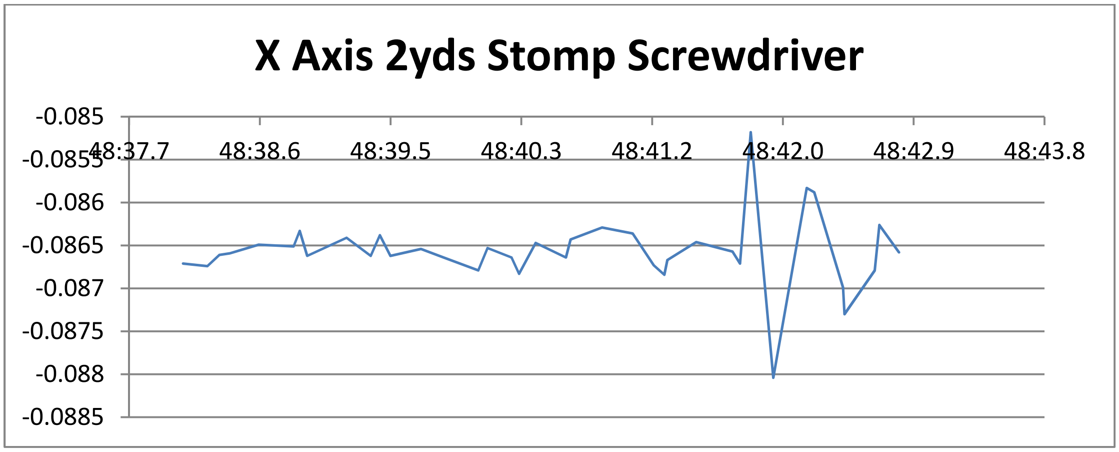


Figure 23. Foot stomp while attached to screwdriver at two yards.

The first outside test was with the sensors lying flat on the ground and a person walked by at a distance two yards, and then walked back at a distance of one yard. The IR Motion sensor had good detection of the person walking by, but the sound and accelerometer readings did not indicate any movement. The same test was performed with the accelerometer attached to the screwdriver, and had the same result. Figure 24 shows the IR Motion sensor readings for the two and three yards walking distances. The biggest difference between the graphs is the SensorValue range of the readings. When walking by at two yards, the SensorValue range is 190. The SensorValue range when walking by at three yards is only 64. This shows that the 1111 IR Motion sensor has a detection range of approximately three yards before the readings would get obscured by noise.

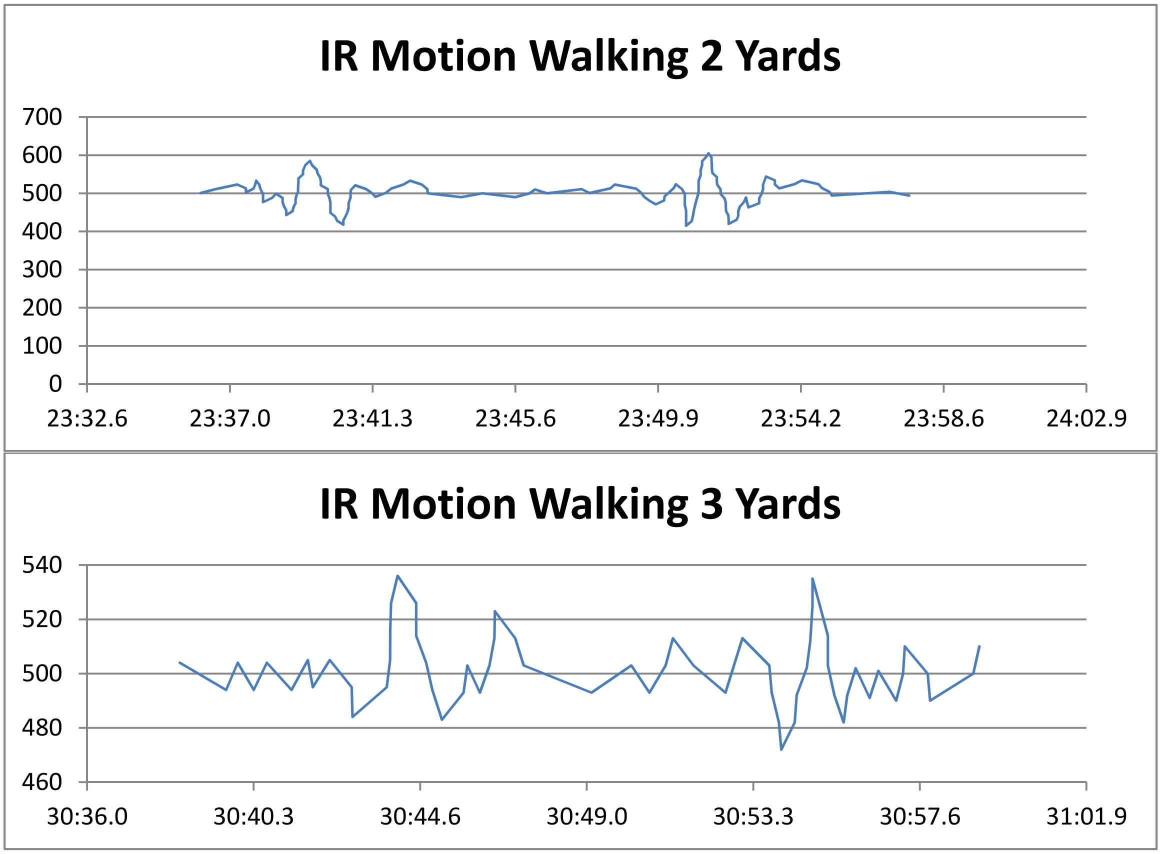


Figure 24. IR Motion comparison for walking at two and three yards.

The results for the outside tests with stomping were similar to the inside tests, but the SensorValue range in the readings decreased. The SensorValue range of readings with a foot stomp at two yards with the sensor flat on the ground was 0.00237, and the SensorValue range while attached to the screwdriver was 0.00501. The graphs of these tests are shown in Figure 25. The accelerometer readings increased as the distance from the sensor decreased, just like the indoor tests. These tests, along with the indoor tests, show that the accelerometer's detection ability is increased when attached to the screwdriver. They also show that the raised floor in the Network Lab is a better conductor of vibration than hard-packed dirt, which was expected. The sound sensor detected the foot stomps with similar results to the indoor tests at two yards and three yards, and the sensor readings remained fairly constant despite the distance change. The graph of the sound sensor readings at two yards is shown in Figure 26.

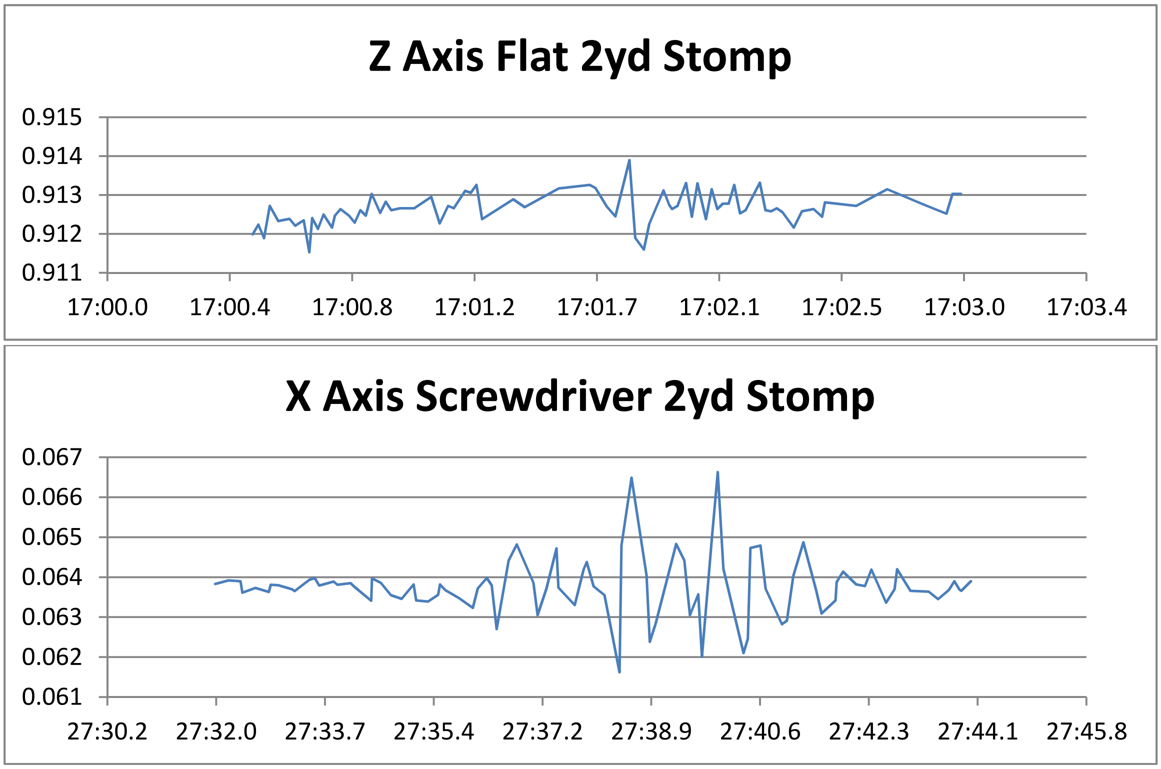


Figure 25. Foot stomp at two yards while flat on ground and attached to screwdriver.

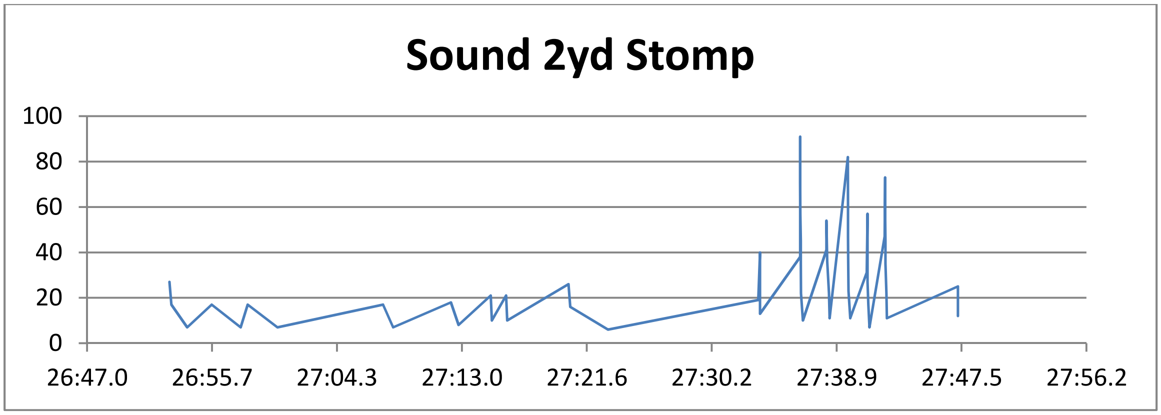


Figure 26. Sound sensor during foot stomp at two yards.

D. CHAPTER SUMMARY

We described the experiment set-up and the results were discussed in this chapter. We reached the following conclusions:

- The "Z" axis of the accelerometer fluctuates much more than the "X" or "Y" axes. The manufacturer, Phidgets, Inc., stated this is normal and that a difference sensor is used for the "Z" axis.
- The 1056 accelerometer is not suited for detecting normal footsteps indoors or outdoors.
- The 1056 can detect impacts on the ground, such as foot stomps or digging with a shovel, with an effective range of approximately two yards.
- The 1056 accelerometer capabilities were improved by attaching it to a metal rod, which was stabbed into the ground.
- The 1111 IR Motion sensor has an effective range of three yards, and detects movement equally indoors and outdoors.
- The 1133 Sound sensor has a substantially longer range than the IR Motion or accelerometer sensors, depending on the frequency and loudness of a sound.

In comparison to previous work by Peter Young, the PhidgetSpatial 1056 sensor does not perform as well as the iPhone's internal accelerometer for detecting footsteps

[12]. The iPhone’s accelerometer had a lower noise level which enabled the detection of smaller disturbances. When compared to previous work by Ahren Reed, the Phidgets, Inc., sensors of the same type performed at the same level he concluded in his research [10].

The next chapter discusses the overall findings of the research and lessons learned about the devices and methodology of our tests. Ideas for future work are also discussed in the next chapter.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSION

A. FINDINGS OF RESEARCH

We have discussed the use of an Android Smartphone to view information from deployed sensors. In this research, we successfully passed sensor information across a wireless network and displayed it to the user using a native Android application on a Google Nexus One and Commercial-Off-The-Shelf sensors from Phidgets, Inc. We discovered that the accelerometer used in this research, the Phidgets 1056 PhidgetSpatial 3/3/3, is not suited for detecting normal footsteps of a person due to noise produced by the sensors. However, it is capable of detecting more forceful impacts on the ground such as a person running, digging, or a heavy object falling, out to an approximate range of two yards. We also learned that the Phidgets 1111 IR Motion and 1133 Sound sensors perform fairly well for their capabilities. The IR Motion sensor is well suited for detecting movement out to an approximate range of three yards, and the Sound sensor is capable of detecting sounds at a much further range depending on the frequency and loudness of the sound.

The limiting factors of the system we used are the reliance on external power, susceptibility to weather and the environment, the limited range of the sensors, and the noise produced by the accelerometer. If these limiting factors are eliminated, or reduced, combining these three sensors into a single system could yield a very useful tool for surveillance. There are numerous types of accelerometers and motion sensors with longer detection ranges available. Better detection results could be obtained by using an accelerometer that produces less noise, and/or a motion sensor that has a longer range. Power is always a limiting factor when using mobile devices and cordless systems. Battery and solar technology is constantly being developed, which could enable sensor systems to run on internal power for a substantial amount of time.

Smartphone and sensor technology is constantly improving. More advanced phones could enable even more capabilities when coupled with advanced sensors designed for specific purposes. The capabilities of applications are limited by the devices running them. As phone and sensor technology is enhanced, so may be the capabilities of the applications created to run on them. The application we created for this research was very basic, and did not utilize the full capabilities of the phone for processing, analysis, and display of data. Much more robust applications can be created to utilize the sensor data to better convey information to the user. The controlled testing we conducted yielded good results, but not at the levels of a fully functional sensor system. The system would need improvements in various areas, and real-world testing would need to be conducted.

B. FUTURE WORK

The power requirements, lack of ruggedness, and size are some very limiting characteristics of our sensor system. A fully function sensor system should have internal power and be immune to weather and the environment. An internal battery with solar recharge ability would provide much more mobility for the system. Creating a smaller sensor package, completely self-contained, and able to operate in any weather condition or land environment would be key to designing a robust sensor system. Sensors for military application generally should not be noticeable by an adversary, so making the system small enough to easily conceal it would be an important step toward a deployable system.

Developing a robust application would greatly enhance the system. Utilizing the GPS data to place alerts on a map within the application would be very useful, especially when multiple sensors are deployed and communicating with the same Smartphone. The challenge of how, and when, to indicate to the user that movement was detected is also important. Our application used only visual alerts based on a simple threshold of the sensor data. Audible and vibrating alerts are also an option for alerting the user, depending on the requirements of the mission. Eliminating false positives and negatives is an ongoing challenge for sensor systems. The application needs to be able to process and analyze the sensor data accurately and quickly when used as a real-time information system. The user being able to adjust the sensitivity of the sensor “on the fly” would also be a useful capability.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] GSM Arena.com. HTC Google Nexus One. < http://www.gsmarena.com/htc_google_nexus_one-3069.php >. Accessed 09 SEP 2011

[2] Phidgets Inc. Android Code Sample. 2011. < http://www.phidgets.com/downloads/examples/android-examples_2.1.8.20120131.tar.gz >. Accessed 07 NOV 2011

[3] Phidgets Inc. Android API Reference. 2011. < http://www.phidgets.com/documentation/JavaDoc.zip >. Accessed 07 NOV 2011

[4] Phidgets Inc. Phidgets 1072 – Product Manual. 2011. < http://www.phidgets.com/documentation/Phidgets/1072_0_Product_Manual.pdf >. Accessed 01 SEP 2011

[5] Phidgets Inc. Phidgets 1056 – Product Manual. 2010. < http://www.phidgets.com/documentation/Phidgets/1056_0_Product_Manual.pdf >. Accessed 01 SEP 2011

[6] Phidgets Inc. Phidgets 1133 – Product Manual. 2010. < http://www.phidgets.com/documentation/Phidgets/1133_0_Product_Manual.pdf >. Accessed 02 SEP 2011

[7] Phidgets Inc. Phidgets 1111 – Product Manual. 2009. < http://www.phidgets.com/documentation/Phidgets/1111_0_Product_Manual.pdf >. Accessed 02 SEP 2011

[8] Phidgets Inc. Phidgets 1104 – Product Manual. 2009. < http://www.phidgets.com/documentation/Phidgets/1104_0_Product_Manual.pdf >. Accessed 02 SEP 2011

[9] Phidgets Inc. Phidgets 1040 – Product Manual. 2011. < http://www.phidgets.com/documentation/Phidgets/1040_0_Product_Manual.pdf >. Accessed 03 SEP 2011

[10] Reed, A. "Detecting Suspicious Behavior With Low-cost Wireless Sensor Networks." California Polytechnic State University, 2011.

[11] Rowe, N. "Detecting Suspicious Motion With Nonimaging Sensors." Naval Postgraduate School, 2010.

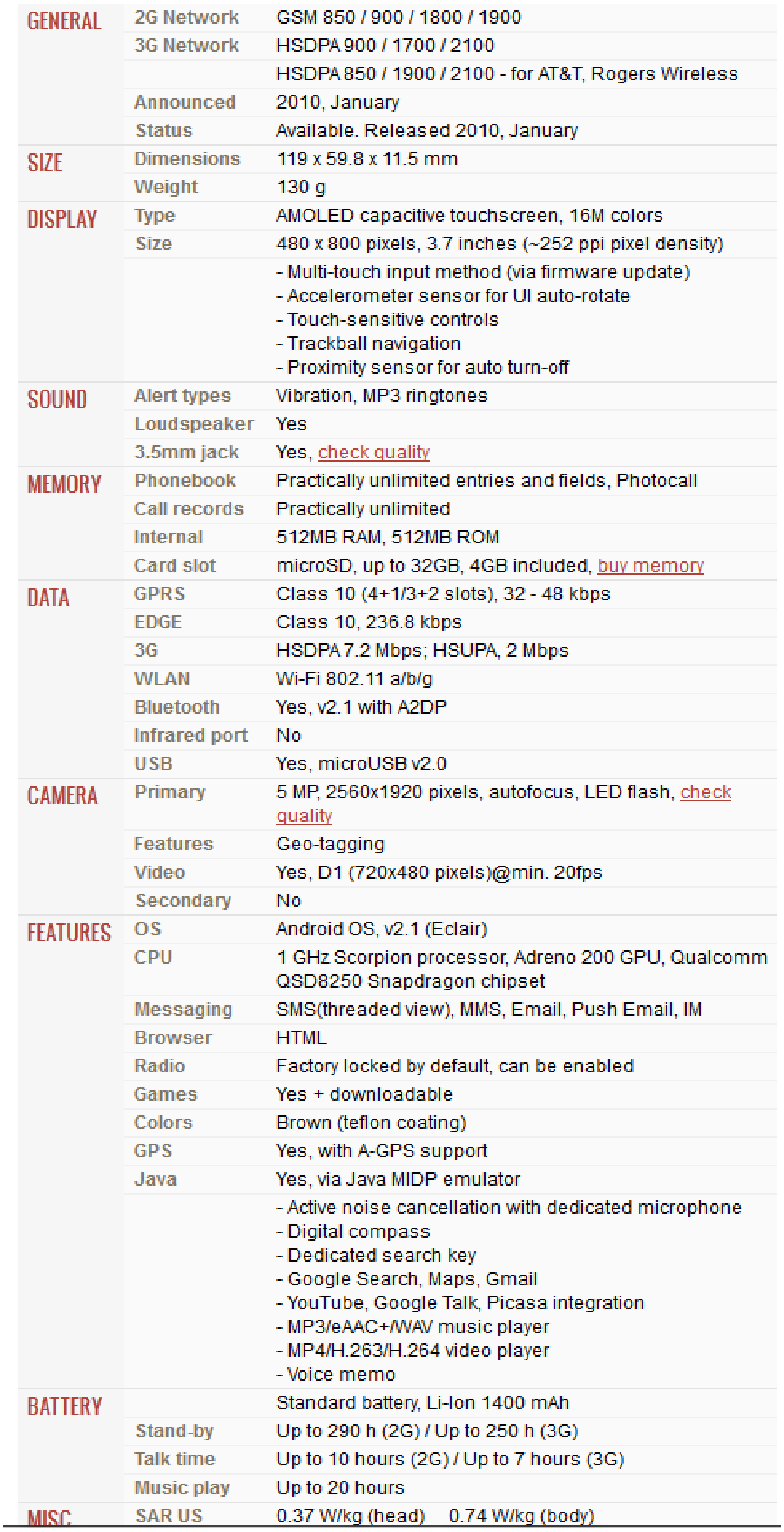
[12] Young, P. “A Mobile Phone-Based Sensor Grid for Distributed Team Operations.” Naval Postgraduate School, 2010.

[13] Boettcher P. and Shaw G. “Energy-Constrained Collaborative Processing for Target Detection, Tracking, and Geolocation.” MIT Lincoln Laboratory, 2003.

[14] Shih, F., Wu, Y., and Chen, J. “A Smart Sensor Network for Object Detection, Classification and Recognition.” New Jersey Institute of Technology, National Dong Hwa University, 2005.

[15] Paul, J. “Web-Based Exploitation of Sensor Fusion for Visualization of the Tactical Battlefield.” Office of the Deputy Under Secretary of Defense, 2001.

APPENDIX



GENERAL	2G Network	GSM 850 / 900 / 1800 / 1900
	3G Network	HSDPA 900 / 1700 / 2100
		HSDPA 850 / 1900 / 2100 - for AT&T, Rogers Wireless
	Announced	2010, January
	Status	Available. Released 2010, January
SIZE	Dimensions	119 x 59.8 x 11.5 mm
	Weight	130 g
DISPLAY	Type	AMOLED capacitive touchscreen, 16M colors
	Size	480 x 800 pixels, 3.7 inches (~252 ppi pixel density)
		- Multi-touch input method (via firmware update) - Accelerometer sensor for UI auto-rotate - Touch-sensitive controls - Trackball navigation - Proximity sensor for auto turn-off
SOUND	Alert types	Vibration, MP3 ringtones
	Loudspeaker	Yes
	3.5mm jack	Yes, check quality
MEMORY	Phonebook	Practically unlimited entries and fields, Photocall
	Call records	Practically unlimited
	Internal	512MB RAM, 512MB ROM
	Card slot	microSD, up to 32GB, 4GB included, buy memory
DATA	GPRS	Class 10 (4+1/3+2 slots), 32 - 48 kbps
	EDGE	Class 10, 236.8 kbps
	3G	HSDPA 7.2 Mbps; HSUPA, 2 Mbps
	WLAN	Wi-Fi 802.11 a/b/g
	Bluetooth	Yes, v2.1 with A2DP
	Infrared port	No
	USB	Yes, microUSB v2.0
CAMERA	Primary	5 MP, 2560x1920 pixels, autofocus, LED flash, check quality
	Features	Geo-tagging
	Video	Yes, D1 (720x480 pixels)@min. 20fps
	Secondary	No
FEATURES	OS	Android OS, v2.1 (Eclair)
	CPU	1 GHz Scorpion processor, Adreno 200 GPU, Qualcomm QSD8250 Snapdragon chipset
	Messaging	SMS(threaded view), MMS, Email, Push Email, IM
	Browser	HTML
	Radio	Factory locked by default, can be enabled
	Games	Yes + downloadable
	Colors	Brown (teflon coating)
	GPS	Yes, with A-GPS support
	Java	Yes, via Java MIDP emulator
		- Active noise cancellation with dedicated microphone - Digital compass - Dedicated search key - Google Search, Maps, Gmail - YouTube, Google Talk, Picasa integration - MP3/eAAC+/WAV music player - MP4/H.263/H.264 video player - Voice memo
BATTERY		Standard battery, Li-Ion 1400 mAh
	Stand-by	Up to 290 h (2G) / Up to 250 h (3G)
	Talk time	Up to 10 hours (2G) / Up to 7 hours (3G)
	Music play	Up to 20 hours
MISC	SAR US	0.37 W/kg (head) 0.74 W/kg (body)

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Dr. Gurminder Singh
   Naval Postgraduate School
   Monterey, California

4. Professor John Gibson
   Naval Postgraduate School
   Monterey, California

5. Dr. Neil Rowe
   Naval Postgraduate School
   Monterey, California