

12-2007

Route Validation Using Radio Frequency Identification

William T. Watson
Columbus State University

Follow this and additional works at: https://csuepress.columbusstate.edu/theses_dissertations



Part of the [Computer Sciences Commons](#)

Recommended Citation


Watson, William T., "Route Validation Using Radio Frequency Identification" (2007). *Theses and Dissertations*. 79.

https://csuepress.columbusstate.edu/theses_dissertations/79

This Thesis is brought to you for free and open access by the Student Publications at CSU ePress. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of CSU ePress.

ROUTE VALIDATION USING
RADIO FREQUENCY IDENTIFICATION

William Todd Watson



Digitized by the Internet Archive
in 2012 with funding from
LYRASIS Members and Sloan Foundation

<http://archive.org/details/routevalidationu00wats>

Columbus State University
The College of Science
The Graduate Program in Computer Science

ROUTE VALIDATION USING
RADIO FREQUENCY IDENTIFICATION

A Thesis in
Computer Science
by
WILLIAM TODD WATSON

Submitted in Partial Fulfillment
of the Requirements
for the Degree of
MASTER OF SCIENCE

December 2007

©2007 by William Todd Watson

I have submitted this thesis in partial fulfillment of the requirements for the degree of Master of Science.

12-11-2007

Date



William Todd Watson

We approve the thesis of William Todd Watson as presented here.

12. 11. 07

Date

Lydia Ray

Dr. Lydia Ray, Thesis Advisor,
Assistant Professor of Computer Science

12-11-07

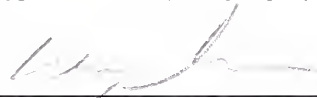
Date



Dr. Glenn Stokes, Acting Dean,
Professor of Environmental Sciences and Biology

12-11-07

Date



Dr. Wayne Summers, Chairman,
Professor of Computer Science

12/11/2007

Date



Dr. Edward Bosworth,
Associate Professor of Computer Science

ABSTRACT

In recent years, radio frequency identification (RFID) has been proposed and implemented in a variety of applications where tracking objects, animals or people is desirable. This paper proposes a novel approach to the application of RFID technology in those applications where it is possible to validate a person or an object's passive contact with a given or arbitrary set of fixed points along a predefined route. The notable departure from the typical application of RFID technology is that in this scheme, the transponders are permanently installed while the interrogator is affixed to a person or object that travels the course. Data are collected by the interrogator and can be examined later to derive path taken, distance, and time. The technology can also be adapted for use in a loosely-connected sensor network in which the whereabouts of the interrogator can be transmitted between nodes. The benefit of this design is the capability to detect and report locations of persons between endpoints in both reasonably remote or local conditions. In a hiking trail application, for example, park authorities can use the information gathered to more quickly locate a missing hiker, reducing safety risks to the person as well as saving time and money associated with search and rescue. We will also illustrate that industrial and security applications are also feasible with this implementation.

TABLE OF CONTENTS

	Page
LIST OF FIGURES	vi
LIST OF TABLES	vii
CHAPTER	
1 INTRODUCTION	1
2 RFID TECHNOLOGY AND LITERATURE REVIEW	3
2.1 TERMINOLOGY	4
2.2 DESCRIPTION OF RFID SUBSYSTEMS	5
2.3 ISSUES AND CONCERNS ASSOCIATED WITH RFID	9
2.4 LITERATURE REVIEW	10
2.5 SUMMARY	14
3 REQUIREMENTS AND SPECIFICATIONS	15
3.1 OVERALL REQUIREMENTS	15
3.2 SOFTWARE REQUIREMENTS	17
3.3 HARDWARE REQUIREMENTS	19
3.4 SOFTWARE DESCRIPTION	23
4 THE INTERROGATOR	25
4.1 SOFTWARE	25
4.2 OPERATIONAL EXAMPLES	26

5	POST PROCESSING	28
5.1	CALCULATIONS	28
5.2	POST PROCESSING EXAMPLE	30
5.3	ANALYSIS	31
6	CONCLUSIONS	33
7	FUTURE WORK	35
	BIBLIOGRAPHY	37
	APPENDIX	
A	INTERROGATOR SOURCE CODE	41

LIST OF FIGURES

2.1	Typical Passive RFID transponder. (Source: [25])	6
2.2	A Phidget TM Transceiver (Source:[22])	8
3.1	EM4102 Transponder Block Diagram (Source: [5])	20
3.2	EM4102 Memory Map. (Source:[5])	21
3.3	Basic Operation of the Transceiver (Source: [5])	22
3.4	Gunstix TM Verdex Computer (Source: [9])	23

LIST OF TABLES

2.1 Properties of Transponder types 7

ACKNOWLEDGMENTS

My sincere thanks are extended to Drs. Lydia Ray and Edward Bosworth for their helpful suggestions and direction during this research. I particularly appreciate Dr. Ray's continued involvement through the process of research and thesis so that the goals were straightforward and not a series of unrelentingly frustrating hurdles. Dr. Bosworth's time and suggestions were invaluable to me during the research and problem analysis phases. My thanks to our Distinguished Chairman, Dr. Wayne Summers, for his leadership and assistance in many ways that he would doubtless consider to be innocuous. However, in my view, they were significant and made a real difference to me. Dr. Glenn Stokes, Acting Dean of the College of Science, pointed out to me that Columbus State University did not have a \LaTeX Thesis style format, but perhaps the Department of Computer Science might benefit from one. Dr. Ray was similarly encouraging with the prospect. To the extent that I know, I am therefore pleased to deliver the first Thesis based upon a style format customized for The Columbus State University College of Science and derived from those used at Stanford, MIT and other institutions. Finally, I appreciate the input provided by numerous students throughout my Master's studies at Columbus State University. There are many very bright Computer Science students here who provide sufficient motivation to be optimistic for the future of our profession.

DEDICATION

This work is lovingly dedicated to my family, who endured countless (and seemingly, endless) hours of my focus on this research.

To my beautiful wife, Roberta: A single superlative is not enough. Your inspiration and encouragement has been unfaltering, even in the face of a project that mysteriously created its own clutter in our home. I am thankful in view of the added demands on my time, you could quietly overlook grass a bit more ragged and cars a bit more dingy than you prefer. I also appreciate your watchful eye and the occasional professional pointer that $\sum \sum_{i=1}^{\infty} x^{y+z} = \frac{p+q+r}{s+t+u+v}$ does not in itself constitute a sentence, despite my insistence to the contrary. I love you.

To my children, Kathryn and William: I encourage you to appreciate learning for learning's sake. As you both transition into adulthood, I pray you will use your God-given talents and skills to make a difference that matters. Seek your own calling in life and consider meaningful things that you love to do - then do them. You have seen firsthand that even an older person can set goals and attain them. Remember the words of Ovid, the Roman poet - *finis coronat opus*.

To my parents: I appreciate your prayers and encouragement throughout this undertaking. I know that you are proud of such an accomplishment as am I in the achievements of my own children. Long ago, I made a promise to you. It is my pleasure to finally keep it.

CHAPTER 1

INTRODUCTION

Radio Frequency Identification (RFID) devices are used in a variety of applications. Regardless of purpose, they are typically implemented using a stationary reader, or interrogator, that accepts data transmitted by transponders affixed to non-stationary objects which are within operational range. The location and presence of transponder-affixed objects are arbitrary with respect to the interrogator.

Implementing an RFID system to record one's progress on a predetermined course is a possible application of the technology. On a hiking trail, for example, mileposts are often installed to indicate the intended path through an area. Hikers will reference mileposts to avoid getting lost in addition to measuring their progress. Facilitators place mileposts to concentrate potentially destructive foot traffic along a designated path.

On the surface, mileposts appear to be ideal locations for the installation of interrogators, with corresponding RFID transponders affixed to each hiker. As hikers pass mileposts, data can be acquired by interrogators as transponders enter their operating range. Such data might be forwarded to a computing facility which can track the approximate whereabouts of individual hikers.

This scheme has significant problems. First, it is not practical to connect interrogators distributed in such a manner to a central data collection facility due to the paucity of available power sources along hiking trails. Second, since most RFID

transponders can be interrogated and read at any time without the knowledge or consent of the bearer, it is possible to track the whereabouts of an individual while on the trail or anywhere the transponder might be taken. The key challenge, therefore, was to develop a means to implement RFID technology in a manner suitable to validate a hiker's route along a trail while mitigating issues of privacy.

To ameliorate these concerns and address the challenge, I implemented a novel scheme in which transponders were placed on the stationary mileposts and interrogators were constructed for portability and carried by the hiker. For the proof of concept design, I chose Class 1 passive RFID devices for their low costs and ease of availability. I created an interrogator by combining a PhidgetTM RFID transceiver with a miniature GumstixTM computer. I developed the necessary software to activate the transceiver, read the data from transponders within range, and subsequently rationalize the received data to a database that contains the coherent data. To complete this project, I integrated elements of computer systems engineering, radio engineering, software engineering, and information assurance. To that end, my design combined modular, off-the-shelf subsystems in such a way to acquire the necessary data while accommodating issues of privacy. I chose these products because they share a compatible interface and satisfy primary requirements of the design.

This thesis is organized in the following manner. The second chapter contains a detailed description of RFID and a review of the literature on this topic. The third chapter describes the operational and functional requirements of the design. The fourth chapter details the functional operation of an interrogator subsystem designed to accommodate the requirements. In the fifth chapter, a description of the post processing and reporting processes are detailed. The remaining chapters conclude this work in addition to offering suggestions where future work might follow.

CHAPTER 2

RFID TECHNOLOGY AND LITERATURE REVIEW

Radio Frequency Identification is not a new technology. Indeed, its first practical use occurred during World War II as a means for identifying Allied airplanes. While the Germans, Japanese and Allied forces were each using RADAR, it only provided information about the presence of any aircraft. It was not possible to determine whether the aircraft was “friendly” or not. Consequently, the Allied forces installed a radio system that would be quiescent until it received a RADAR signal. When the signal was detected, the transmitter on the aircraft was energized and began to transmit a signal identifying the aircraft. This so-called Identity Friend or Foe (IFF) system enabled RADAR technicians to determine whether the incoming aircraft were a likely cause for concern [17].

Despite its early start, RFID technology has not developed at the same pace as other technologies such as telecommunications and computing. In fact, RFID as a technology was essentially dormant until a resurgence of interest that began in the mid-1970’s, but has only become popular again since the mid 1990’s. The variations in the basic technology has enabled it to resurface in a surprising number of applications ranging from inventory tracking and control, electronic entry systems, animal identification and even the identification of the deceased [1]. In this chapter, the components and terminology associated with RFID technology are explained.

2.1 TERMINOLOGY

Active transponder - An RFID transponder with a connected power source. Active transponders generally transmit continuously, irrespective of the presence of transceivers within range.

API - An application programmer interface that allow libraries which support subsystems to be available to other applications.

Collision - A situation in which multiple transponders are attempting to communicate with an interrogator cause interference with each other, resulting in data misalignment at the interrogator. Some transponders are equipped with collision detection or avoidance subsystems.

Far-field - The far field corresponds to an RF source-to-antenna range r great enough that energy radiates from the source only in a radial direction¹.

Interrogator - The device which radiates electromagnetic energy and receives the transmission from a transponder. Also known as a *transceiver* or *reader*.

Near-field - The part of the electromagnetic field that is nearest to the transmitting antenna.

Passive transponder - An RFID transponder that derives power from eletromagnetic inductive coupling or far-field energy harvesting.

Semi-Passive transponder - An RFID transponder that operates using a self-contained power source. It is distinguished from an active transponder by its conditional operation. Semi-passive transponders usually only transmit when a transceiver is within operational range.

¹More formally, at $r = \frac{\lambda}{2\pi}$, where λ represents wavelength, is found the approximate transitional point from near-field to far-field for electrically small antennas, such as those designed for transceivers. The power available to a transponder by inductive coupling is decreased by a proportional distance d from the transceiver at a rate of $\frac{1}{d^2}$.

Tag - See Transponder.

Transceiver - See Interrogator.

Transponder - That RFID device which responds to an activation attempt by an interrogator. Transponders can be active or passive. Commonly referred to as a *tag*.

2.2 DESCRIPTION OF RFID SUBSYSTEMS

The fundamental parts of a Radio Frequency Identification system are a transponder, colloquially referred to as a tag, and an interrogator, or so-called reader. Usually an RFID system will also include software to decode the data received from the tag and a database to manage the encounters with the transponders.

2.2.1 TRANSPONDERS

Transponders are affixed to objects that require identification. The basic component design of a transponder is an integrated circuit, typically of a microchip design, an antenna for communication and some means to connect the two. The memory in the microchip can be read-only or read-write design.

Three basic types of systems have been developed. Active systems are characterized by transponders that contain their own power source. Active transponders typically can be read from relatively long distances, have significant amounts of computational resources, and may also possess limited amounts of storage. They are often physically larger, more expensive and have a limited operational life. Active transponders are often used in high-value implementations because of the flexibility they provide. For certain applications they are superior because they can also communicate with other transponders. Passive transponders such as the example in

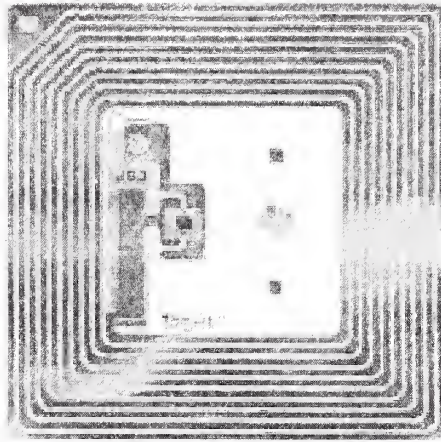


Figure 2.1: Typical Passive RFID transponder. (Source: [25])

Figure 2.1 are powered by electromagnetic inductive coupling or far-field energy harvesting emitted by the interrogator and induced on the attached coil which is rectified and briefly stored in an on-board capacitor. They are generally least expensive, smaller because they have no power source and can be queried a virtually unlimited number of times. They are completely inactive absent an appropriate electromagnetic field required to power them. They suffer from limited range, have little or no local storage, and have few resources to perform any tasks beyond their initial design. A hybrid “semi-passive” transponder is available which effectively increases the operating range of the unit. Semi-passive transponders are powered by their own power source, but usually do not initiate communications. Passive transponders are usually read-only while semi-passive and active transponders can be either read-only or read-write. Because active transponders typically have a large power budget, they can have more computational capability. Consequently, they do not necessarily suffer the same limitations as their passive counterparts. However, because of the capa-

	Power Source	Range	Transmitter
Passive	RF	under 10m	Passive
Semi-Passive	Battery	under 100m	Passive
Active	Battery	under 1000m	Active

Table 2.1: Properties of Transponder types

bility to read active devices from much longer distances is possible, if transmissions are not encrypted, personal information can be divulged at even grater distances. Some active RFID transponders are readable from a distance approaching 1000M.

When in operation, the now-defunct MIT Auto-ID Center developed a classification for transponders in terms of their intended use. The classification system begins with 0 for transponders which contain little or no computational capability, but rather register their presence with the transceiver. These transponders are the ones with which consumers are most familiar, since they are the ones that are typically used in retail inventory control applications. Class 1 transponders are pre-programmed during manufacture to contain unique data within the scope of a particular manufacturer. These passive transponders may also be provided to the end user who programs them prior to use. Class 2 transponders are passive, semi-passive or active. They may have re-writable non-volatile memory. Therefore, they can be used to record certain data, then be erased, re-programmed and re-used. Class 3 transponders are typically active, although may be semi-passive. They can contain sensors to detect environmental conditions, such as temperature or humidity, enabling the purchaser of the attached product to determine if the product encountered unsuitable conditions during shipping. An example of such an application is perishable

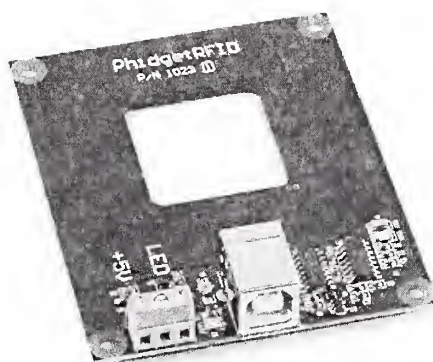


Figure 2.2: A PhidgetTM Transceiver (Source:[22])

food. Class 4 transponders are the most robust and complex of this group. They are capable of establishing wireless networks with other transponders, transceivers, and other equipment.

2.2.2 TRANSCEIVERS

Transceivers or interrogators are manufactured in a number of designs, form factors and sizes that are suitable for their particular application. Handheld units in a “pistol” form factor are built by a number of manufacturers. Many offer the ability to query transponders and forward the query to a database via a wireless connection. Other transceivers are permanently mounted at loading dock doors to interrogate transponders that enter and leave a warehouse. Other designs are less elaborate. The PhidgetTM transceiver shown in Figure 2.2 is an example of a basic transceiver which must be connected to a host computer for operation and subsequent processing of acquired data.

2.2.3 FREQUENCIES

There are four standard frequency ranges for RFID products. Low frequency products operate at approximately 135 KHz. High-frequency products operate at about 13.56 MHz. Ultra-high frequency products operate around 915 MHz. Finally, the highest standard frequency for RFID operation is about 2.4 GHz. Each range is best suited for particular applications. For example, animal and key entry systems operate passively in the low frequency ranges or in related applications where long-range operation is not beneficial. On the other end of the frequency spectrum, the 2.4GHz transponders are usually active, but can also be deployed in such a manner to be compatible with wireless networks[6].

2.3 ISSUES AND CONCERNS ASSOCIATED WITH RFID

2.3.1 PRIVACY

Privacy advocates are concerned with the basic operation of RFID transponders and have created web sites such as Privacyrights.org that seek to educate the public about technologies that may place private citizens at risk. RFID technology is understandably a concern because without the knowledge or consent of the holder, any tags in one's possession can be examined, assuming an appropriate interrogator is available. It is therefore possible that personal information can be derived.

For example, if government regulations required all products to be outfitted with RFID technology, it might be possible for a malfeasant to aggregate information from a number of transponders that an individual is carrying. Such information might reveal past medical procedures (subject had an atrial valve replacement),

clothing choices (subject prefers certain name-brand shirts), and even political or ideological persuasion (subject is reading subversive materials from the library).

2.3.2 CONFIDENTIALITY

Data confidentiality is another concern with RFID technology. Since transponders communicate with interrogators using radio frequencies, the transmitted signals are available to both intended and unintended recipients. Low-cost transponders are purposefully designed to possess minimum computational resources that not only save design costs, but reduce the power requirements. Consequently, luxuries such as cryptographic subsystems must be excluded by necessity [27]. Some manufacturers are addressing such concerns, but the desire to produce transponders at a price of less than \$0.05US is a market force that continues to preclude the incorporation of such technology. Some more expensive transponders are designed with encryption technology. However, it is not yet widely implemented and standards are still being debated.

2.4 LITERATURE REVIEW

A remarkable amount of literature exists regarding the tracking of people and things using Radio Frequency Identification (RFID). There are many research and commercial projects that are focused on various schemes for providing the answer to, "Where is ...?" While on the surface, this seems to be a positive step in asset control, location of personnel who move about a facility, and countless other beneficial schemes, there are also concerns raised by individuals and groups who are interested in privacy issues. This literature review will focus on a number of RFID-related tracking systems that are somewhat related to the types of tracking that has been proposed as

a possible project for identifying one's location and progress in outdoor recreational settings. Products, where applicable, are noted in this review.

Recreational Use - A partnership between Nike and Apple has produced the Nike+iPod product [15]. It incorporates an RFID transponder in a cavity designed in specially-equipped shoes, along with a transceiver that is connected to the data port on an Apple iPod. After calibrating the device, the sensor in the shoe reportedly transmits a block of data to the receiver, which in turn, uploads the data into the iPod. When the user wishes, they can upload the data to the Nike+iPod website, where a record of their exercise activity is maintained.

This product is not without some privacy issues, however. Saponas, Lester, Hartung, and Kohno [23] have performed some reverse-engineering on the Nike+iPod device. Apparently, other Nike devices in the area can also receive the information transmitted by the transponder fitted in the shoe, including rogue devices designed specifically to collect this information from unwitting users of the product. Clearly, the issue is that data can be collected without the user's knowledge or permission. The basis for their paper is to communicate issues of privacy related to the Nike+iPod product.

Emergency Use - A related application could be associated with emergency personnel, such as firefighters, in which the firefighter or the fire is tracked by RFID or sensor networks, so that the location of either can be known by those managing a remote fire. A related technology is advanced by Hefeeda [11], in which RFID sensors forming a network are deployed into a fire zone to determine its scope. However, such applications involve resources which are beyond the practicality and capability of existing technology when widely deployed [26]. The Agilla project [7] details an implementation whereby low-cost sensors are deployed that form an ad-hoc network.

The sensors contain thermistors to sense ambient temperature, enabling this information to be transmitted to the firefighter command post. Firefighters, in turn, could be safely guided to the optimal point to fight the fire. Clearly, this technology could be adapted to manage other types of emergencies in which the area is indeterminate or dynamic, such as radiological. The deployment of wireless sensor networks as a solution to this problem is a logical continuance of this research. Such an application could be used in emergency and tactical situations, such as in a war zone. Each of these applications seem to be closely related, with variations occurring in which equipment is stationary and mobile, and with the incorporation of a sensor network that may track individuals.

Related Configurations - In similar applications, users at one of Europe's largest amusement parks, Legoland, utilized a product developed by Bluesoft to enable parents to track their children who were outfitted with an 802.11b "Aeroscout" tag [3]. The advantage to this technology is that the existing 2.4 GHz system used for their Wi-Fi infrastructure for point-of-sale and other functions was usable by this technology, negating the need to have two wireless network infrastructures operating at incompatible frequencies.

NASA reports that a similar technology will be used at the new Old Faithful Visitor's Center, scheduled to be opened in 2008 [4]. Visitors will be able to have their RFID tag scanned to later retrieve information about the points of interest. Privacy advocates may see issues with this, although the details of the system is not provided in the article.

The Great Wolf Resorts issues RFID wristbands to its quests, enabling keyless room entry and the purchase of goods and services for patrons [14]. In addition, the technology provided by Precision Dynamics can identify the guest wearing the wrist-

band, although the data transmitted between the RFID transponder and transceiver is reportedly encrypted.

The City of Ocean City, New Jersey is planning to use RFID tags as a means for identifying those patrons who have paid to use the municipal beach [18]. The wrist bands are queried by a hand-held device used by the beach staff to ensure that the tags are legitimate. No additional details are available, but the tags could be used to determine who has visited the Ocean City beaches on a particular day, assuming that data is collected from each patron. There are a number of ways to implement this technology which could limit revealing the existence of a particular patron.

Privacy Concerns - An application of RFID in a California high school was widely reported in the press in 2005 [30]. The program was ostensibly to assist teachers with taking attendance in each class. The system could also track the location of any student on campus, enabling administration to quickly find a student should it become necessary. Privacy concerns were raised from a few parents, community residents and the American Civil Liberties Union. Ultimately, the system was discontinued. Such a system, however, could easily be adapted to the proposed outdoor system.

The practices of individual item tagging has been at the center of protests leveled at retailer Wal-Mart in recent years [2]. Because some RFID transponders are not much larger than a grain of sand, they are virtually undetectable. Consequently, they can be placed in virtually any packaging, enabling the circuitry to be reused after the sale by any compatible system.

Similar far-field detection issues have been raised by those concerned with RFID implementations in passports [19]. Consequently, in such a system, managing privacy concerns will be part of the issue. As Weis noted, such issues are partly addressed by the application of appropriate technology and partly addressed by policy [29].

2.5 SUMMARY

Each of the technologies noted have similarities to the ideas proposed. It would seem uncomplicated to adapt existing technologies to the problems proposed. Switching position between a fixed transmitter and mobile transceiver does present additional problems related to data collection and portability, particularly with respect to an adequate power source. There do appear to be portable transmitters that are capable of extended operation on low power, and several devices, including University of California's Mote technology, which are implemented using Intel Motes [13]. Small, lightweight computers, such as those from Gumstix [9] seem to offer some promise for configurable portability for ad-hoc wireless network devices. Their relatively low cost and capable features could be an appropriate platform for portable data gathering and transmission.

CHAPTER 3

REQUIREMENTS AND SPECIFICATIONS

The development of any system will necessarily have some parameters for its design. Such definitions determine the scope of the problem that is to be solved. In this design, there are physical, operational and practical requirements that must be satisfied. Certain aspects of the design have influence on other design decisions, necessitating mutual considerations of each requirement.

This chapter details the overarching requirements of a design to achieve the project goals. We will first detail the desirable attributes in what can be considered a successful design. Then, the products selected for integration are described. A more detailed description of these products follows to explain their features and capabilities, and the rationale for choosing them.

3.1 OVERALL REQUIREMENTS

Designing a portable system has a number of goals and challenges. The following goals and challenges were key considerations in the design of the devices and the software to integrate them.

Size. Ideally, the hardware should be contained within a small package. A 16-bit PC104 form-factor, having the dimensions of about 90mm X 95mm should be considered the maximum size allowed to maintain a compact size [21]. PC104 form factor designs are very robust. While they are characterized as miniature low-power

devices, this is in comparison to standard-sized personal computer mainboards. The design of a PC104 board often includes many unneeded components for our design, such as a video frame buffer, network ports and direct keyboard and mouse I/O.

Low power. Since the application is intended for use where access to a constant supply of power is not necessarily available, all hardware must operate in low-power conditions. Advancements in battery technology are continuing to emerge, so size, mass and capacity were key considerations when choosing the computing and RF components.

Safe. The equipment must be safe to operate within the proximity of humans. Implicit in this criteria is minimal risk to injury or long-term effects associated with near-field contact with RF energy. Open questions remain regarding the safety of operating RF devices in the proximity to human tissue [16]. Prudence dictates minimizing the level of radiated power in the interrogator to limit exposure and conserve power. The human-worn component is ideally a single or integrated part, including the power system, to minimize the potential of entanglement in cables.

Interference. The equipment must not cause interference to any other equipment operating in the area. While these are experimental, it should be noted that any derived commercial product should conform to FCC part 15 for interference [28].

Cost. While cost is not the prime consideration, it is important that the project minimizes costs by choosing the appropriate hardware to meet the objectives. Choices of active vs. passive tags increase costs in each tag, but in fact may minimize overall utility by improving the capabilities of the system.

Complexity. Design should be easy to use and contain a minimum of components. It should be relatively easy for an average person to understand and use. Consequently, the use of pre-programmed tags, for example, are adequate for the design

of this project. Although programming tags is a rather trivial activity, it almost certainly would require additional software and hardware. An off-the-shelf product eliminates this task in implementation and maintenance, such as if a defective tag requires field replacement.

Frequency. Operational frequency is not significant, as long as other goals are met. The frequency chosen must operate properly and safely within the proximity to human tissue (most manufacturers indicate 915MHz is least desirable frequency adjacent to tissue). Implicit considerations in the frequency choice for moving objects is data transfer rate. In this application, even a slow rate of transfer associated with lower frequency devices is not significant since the average walking speed of humans is about 5 km/h, and running speed is about 10km/h. For purposes of this proof of concept design, even low-frequencies satisfy our data transference rate requirements.

Post processing. A host computer will execute programs to extract the data from the portable interrogation computer, compare it with a prepopulated database, and determine if the route is consistent with the expected route. Distance and time calculations will be performed to determine the individual segment and total distances traveled as well as elapsed, total and average rate of travel.

3.2 SOFTWARE REQUIREMENTS

Software selections are often conditional upon the hardware. In the case of an integration project, it is beneficial to attempt to use common systems where possible to maximize reuse of common programs or code. For commercial RFID systems, it is clear that many integrators choose to develop applications for operation on the X86 processor family running some variation of the Microsoft Windows® system. Other

RFID application vendors, particularly those who are based in Europe, tend to favor Linux.

3.2.1 INTERROGATOR SOFTWARE

Since one of the hardware design specifications was a physically small interrogator that consumed a minimal amount of power, the search gradually focused upon a system running a version of embedded Linux. Several manufacturers produce hardware which either support an embedded Linux or is predisposed to Linux in their design philosophy.

3.2.2 TRANSPONDER SOFTWARE

The transponder types chosen for this project are passive designs. Consequently, its programming is read-only. There is no additional software which can possibly be installed to enhance or augment the EM4102's operations.

3.2.3 HOST COMPUTER SOFTWARE

The type of host computer is not a key issue in this project. It is assumed that it has an ANSI-compliant C compiler and a character-based user interface such as a shell. For the purpose of this project, a desktop computer running a Linux 2.6 kernel was used. Software written to read the database and the data acquired from the transponders and subsequently correlate the two does not require large computational resources

3.3 HARDWARE REQUIREMENTS

In keeping with the design goal of minimal hardware complexity, Linux was also the common operating system for some inexpensive RFID interrogator hardware. Consequently, computing hardware was selected that supports Linux. At about 9 grams, the Gumstix [9] embedded computer platform is thus much smaller and has lower mass than could have been possible if a Windows® platform was chosen. It is equipped with a reasonable compliment of resources to accomplish the project goal.

3.3.1 TRANSPONDER HARDWARE

Transponders of the type used are built around an EM4102 Read-Only Contactless Identification Device (Figure 3.1). They operate in the 100 to 150KHz frequency range. The pre-assembled module operates at 125Khz, with an operational range of approximately 10cm from the interrogator used. When power is induced at its electromagnetic field coil, the power is rectified and used to power the EM4102 circuitry. The transponder transmits a 64-bit data string serially. Since no authentication or encryption is performed, EM4102 devices simply continue transmitting the 64-bit string until the electromagnetic field no longer can supply adequate power to energize the circuit.

As detailed in Figure 3.2, the EM4102 circuitry has a memory array that is arranged as 64 bit array, divided into 5 sections. The header is the first 9 bits, each of which is set to 1. There are 10 row parity bits, 4 column parity bits and a single stop bit which is set to 0. The remaining 40 bits are reserved for data. Referring to the figure, bits D00-D03 and D10-D13 are customer specific identification bits. The integrated circuits are programmed with odd parity for P0 and P1 and always with

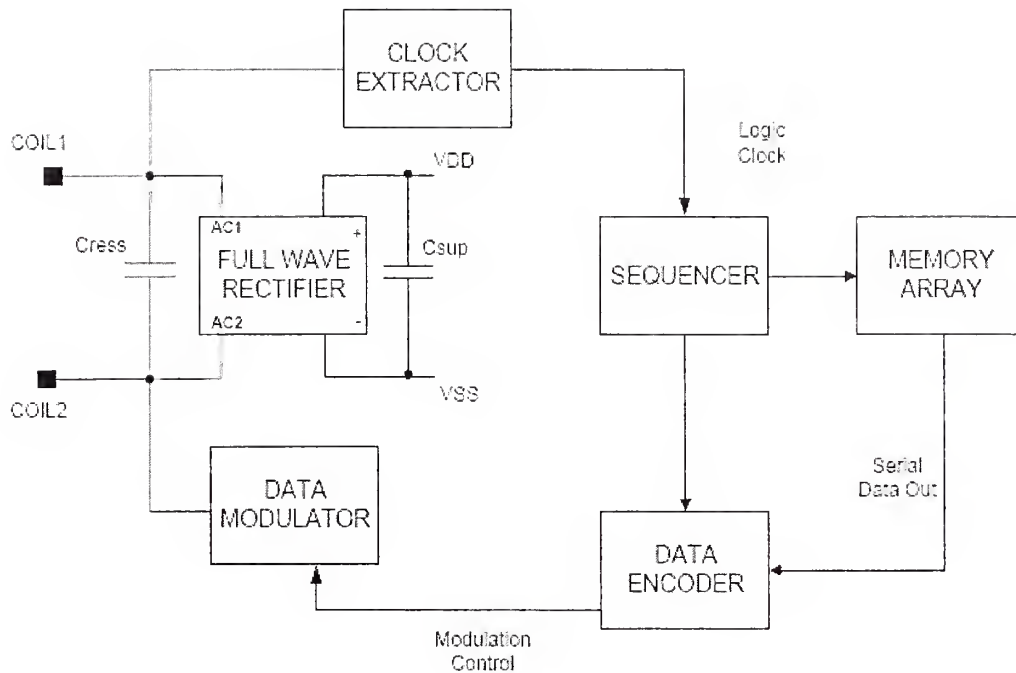


Figure 3.1: EM4102 Transponder Block Diagram (Source: [5])

a logic zero. The parity bits from P2 to P9 are even. The column parity PC0 to PC3 are calculated including the version bits and are even parity bits.

3.3.2 INTERROGATOR HARDWARE

The transceiver was purchased already assembled [22]. A notable benefit to this particular product is the inclusion of a USB-B (slave) port. This enables quick connection and disconnection from the host computer. In addition, it is powered via the USB port. It has an integrated Light Emitting Diode (LED) that can be turned on via software when it receives data from a transponder, and off again when the transponder moves out of range.

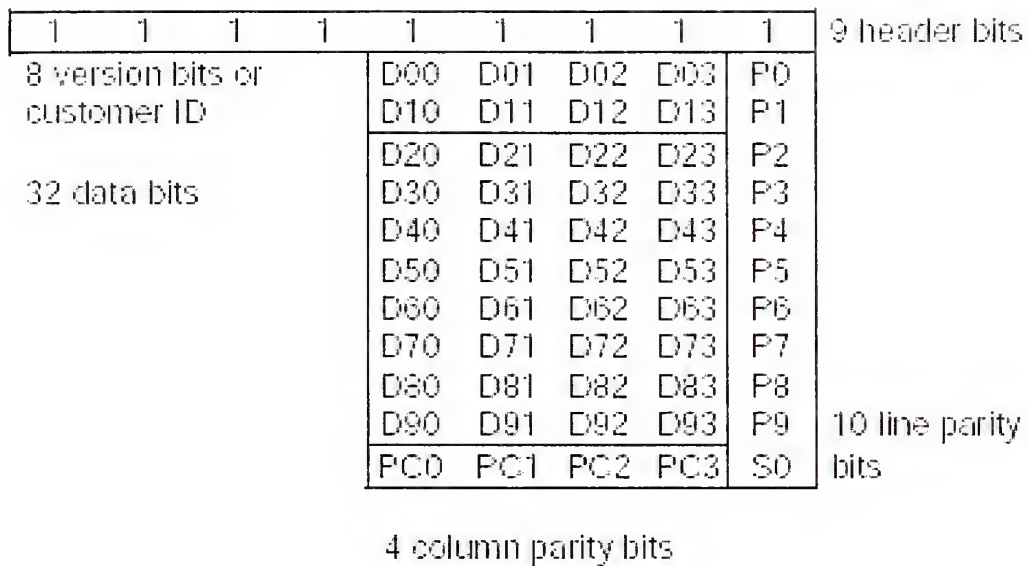


Figure 3.2: EM4102 Memory Map. (Source:[5])

The interrogator is accessible via software written using the programming interface. A basic set of functions, including status, identifier, and control functions are available from this interface. Connecting the transceiver to the portable computer and building control software resulted in a working interrogator.

3.3.3 PORTABLE COMPUTER

The Gumstix™ line of embedded Linux computers are designed around the Intel® XScale PXA270 processor. The one chosen operates at 400 MHz with 64MB of RAM and 16MB flash memory, which is sufficient to operate a small Linux kernel and the software developed for this application [9]. One benefit to the particular model chosen is that it is equipped with USB-A (host) signals and a Bluetooth® wireless subsystem, enabling the data to be conveniently transferred to the host computer

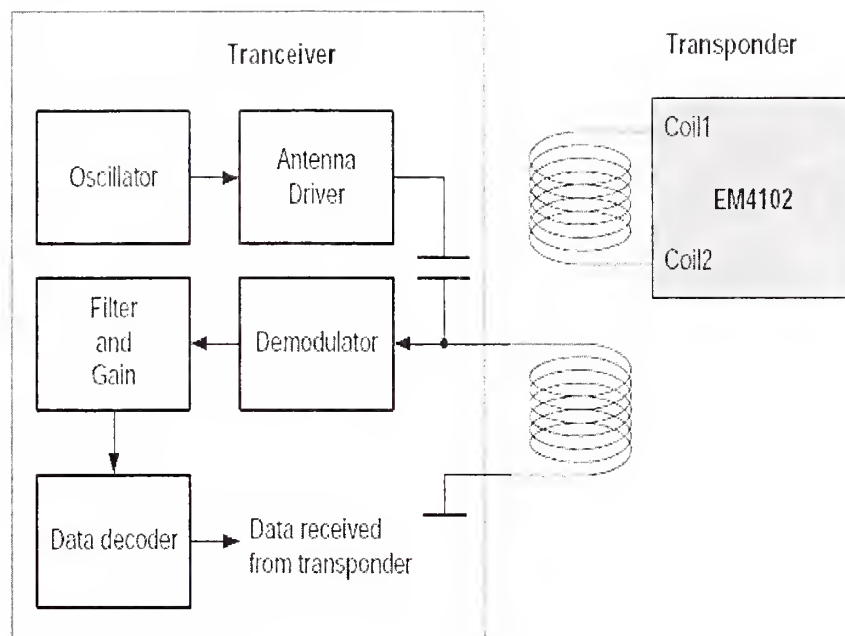


Figure 3.3: Basic Operation of the Transceiver (Source: [5])

using wireless or a wired console. When the Bluetooth® subsystem is disabled, the computer consumes less than 250mA of power at a nominal 5 DC volts [10]. This makes it ideal for a portable platform. It can be operated for several hours on three standard 1.5v type AA or AAA alkaline batteries wired in series. The computer is convection-cooled. Therefore, it is not necessary to expend limited battery power to operate a cooling fan. Figure 3.4 is an example of the approximately 20mm x 80mm Gumstix™ Verdex computer used in this project.



Figure 3.4: Gumstix™ Verdex Computer (Source: [9])

3.4 SOFTWARE DESCRIPTION

There are several software components necessary at each stage of the system. First, the transponder software is pre-programmed. Consequently, this means that any modifications to enable reading or decoding of the data presented must be performed elsewhere in the system. The interrogator's vendor publishes an application programming interface (API) in addition to header files for the developer to use when developing their own applications.

Using the Gumstix™ computer requires applications to be cross-compiled to the PXA270 processor. The computer vendor makes a cross-compiler available to enable applications written on an x86 Linux platform to be cross-compiled for the Intel PXA270 processor. The toolkit provided with the Gumstix™ enables one to quickly move executable software from a desktop platform to the Gumstix™. Since the computer runs a lightweight version of Linux, enabling the USB port on the Gumstix™ is similar to that of a more conventional Linux computer. Once USB support is added, the interrogator can be connected to the computer, detected as a

peripheral, and activated once the interrogator software is instantiated.

CHAPTER 4

THE INTERROGATOR

4.1 SOFTWARE

The interrogator is built from a combination of an RFID transceiver and a connected portable computer. For a proof of concept, a transceiver purchased from Trossen Robotics was chosen. It is connected and powered via the integrated USB connector. The portable computer was built with a small user environment sufficient to enable file transfers to and from a host computer. The vendor-provided libraries enabled software to be written to enable and capture encounters with RFID transponders.

In a conventional RFID application, the interrogator may be placed within the range of numerous transponders. Obviously, they will each attempt to communicate with the interrogator. When this occurs, collisions result. A number of schemes are available to mitigate collisions, involving space, time, frequency and codes [6]. In this conceptual application, collisions should not be a factor since transponders will be spatially distinct. However, collisions could emerge as an issue if the components were replaced with active RFID technology.

The code enables the transceiver to be energized and power its transmitter. When transponders are encountered, they are energized and transmit their data. The software captures a 10-digit hexadecimal number of the transponder along with a timestamp, offset from the time the software was instantiated. As a transponder enters the electromagnetic field and begins to transmit the encounter is recorded.

When the transponder leaves the field, its departure is recorded. Within the scope of this problem, recording such an encounter in this fashion is sufficient. This data is enough information to be analyzed later with the intention of validating route statistics. The appendix contains a listing of the course code for the interrogator program.

The interrogator subsystem is capable of both a wired serial and a wireless connection via Bluetooth®. The connection is created with the host computer using either of these means. Then the record of the encounter is transmitted from the interrogator to the host for post processing.

4.2 OPERATIONAL EXAMPLES

When the RFID transceiver is connected to the computer, the USB device drivers will detect and enable the interrogator. Although three USB ports are detected, there is a single physical USB host port presented on this computer. Therefore, usb1 is used each time. The following sample output is the system message when the driver is instantiated and the transceiver is connected to the computer.

```
pxa27x-ohci pxa27x-ohci: PXA27x OHCI
pxa27x-ohci pxa27x-ohci: new USB bus registered, assigned bus number 1
pxa27x-ohci pxa27x-ohci: irq 3, io mem 0x4c000000
usb usb1: configuration #1 chosen from 1 choice
hub 1-0:1.0: USB hub found
hub 1-0:1.0: 3 ports detected
usb 1-2: new low speed USB device using pxa27x-ohci and address 2
usb 1-2: configuration #1 chosen from 1 choice
starting rfidio...
```

Once the interrogator is connected and functioning, the reader software is instantiated. The interrogator responds with its name and serial number, along with the

configured outputs. The outputs are not connected nor defined in the program. The following output is an example of the data collected from a set of transponders on a given course. For convenience in debugging, a local file was added to correlate the transponder identifier to a known waypoint, causing the “Position n ” to be printed in the example output below. This identifier is not required nor used for route validation, however.

```
start
Device attached
have waited stat=OK
device name is Phidget RFID 4-output
serial number is 37625
2 outputs = (0 = undefined) (1 = undefined)
Acquired: 1100b8fca3 90.443825 Position2
Lost: 1100b8fca3 91.067582
Acquired: 1100a09b8f 168.024005 Position1
Lost: 1100a09b8f 168.652006
Acquired: 0d003b43ca 333.713041 Position3
Lost: 0d003b43ca 334.328907
Acquired: 0f00aeea6e 544.254061 Position4
Lost: 0f00aeea6e 544.878053
Acquired: 11006d0b12 932.572101 Position5
Lost: 11006d0b12 933.196052
Done.
```

There were 5 waypoints passed in this example. The transceiver software detected the acquisition and loss of each transponder by reporting a 10-digit identifier and a timestamp. The timestamp, relative to the time of the software’s instantiation, was sufficient to process segment and aggregate times, used later to derive statistical information. Because connection to transponders are ephemeral, either the time associated with acquisition or loss of the transponder is adequate for calculation provided that one is used consistently at each phase of calculations. The timestamp accuracy exceeds our requirement and is rounded in the final route report.

CHAPTER 5

POST PROCESSING

5.1 CALCULATIONS

When the encounter record is received into the host computer, post processing can begin. During installation of the transponders, each one is cataloged. The key information that will be required for statistical processing are the identifier, the longitude and latitude of the transponder, and the previous and next transponder one would encounter during a prescribed route.

The software will canonically process the encounter record beginning with the first encounter. By using the tag identifier as its record key, the database is queried. If this record in the database contains a null previous pointer, the software will recognize this transponder as a starting point. Each encounter is successively checked against the database for its existence and the correct order. If a transponder is encountered that is not in the database, it is noted but not processed. The assumption is made that the encounter is with a transponder that is not legitimate. The checks continue until no more encounters are in the encounter record. If the database contains a forward link for the last transponder identifier, this exception is noted.

With the results already gathered and processed for route validation, it is then possible to compute the approximate segment and aggregate distances traveled, speed and time. Because of the small distances calculated in our application and

the very small ratio between the sides, the spherical law of cosines [8] is not necessarily well suited. Since the formula uses the inverse of the cosine, rounding errors become untenable for short distances for numbers with few significant digits. Rather, the Haversine formula [24] is used for our short distance calculations because it is not prone to the same error margin. It should be noted that Haversine assumes a spherical earth. Consequently, distance is measured from one point to the next, making no distinctions for variations in terrain. The Haversine formula is based upon the Great Circle Radius. The general formula is as follows.

Given $R = \text{Earth's mean Radius} = 6,371\text{km}$

$$\Delta(\text{lat}) = \text{lat2} - \text{lat1}$$

$$\Delta(\text{long}) = \text{long2} - \text{long1}$$

$$A = \sin^2(\Delta \text{lat}/2) + \cos(\text{lat1}) * \cos(\text{lat2}) * \sin^2(\Delta \text{long}/2)$$

$$C = 2 * \text{atan2}(\sqrt{A}, \sqrt{(1 - A)})$$

$$D = R * C$$

It is straightforward to recreate this formula in a programming language. For example, in JavaScript, the formula can be represented in this fashion. Since JavaScript utilizes IEEE 754 floating point numbers, the accuracy is to 15 digits resulting in accuracy of measurement within approximately one meter.

```
var R = 6371; // km
var dLat = (lat2-lat1).toRad();
var dLon = (lon2-lon1).toRad();
var A = Math.sin(dLat/2) * Math.sin(dLat/2) +
        Math.cos(lat1.toRad()) * Math.cos(lat2.toRad()) *
        Math.sin(dLon/2) * Math.sin(dLon/2);
var C = 2 * Math.atan2(Math.sqrt(A), Math.sqrt(1-A));
var D = R * C;
```

The remaining processing involves computing time to travel each segment. Since the data gathering program on the transceiver records the time as function of the

program's instantiation, the calculations are a matter of applying $distance = rate * time$.

5.2 POST PROCESSING EXAMPLE

While the collection of transponder data is referred to as being stored in a database, no particular preference is given to the type or structure. A text file is sufficient for small examples. Clearly, as the inventory of transponders increases in size, efficient retrieval of the data becomes an issue, justifying the overhead of a relational database. The data are arranged in the database such that each transponder identifier can be the index to a record. Maintaining pointers to the previous and next transponder ensures referential integrity. The fields are supplied as a comment on the first line in the following example.

```
ID, LOCATION, NEXT, PREV, LAT, LONG
1100a09b8f, Position1, 1100b8fca3, , 33.571212334, -83.23254235
1100b8fca3, Position2, 1100a09b8f, 0d003b43ca, 33.571123344, -83.23612324
0d003b43ca, Position3, 1100b8fca3, 0f00aeea6e, 33.573212331, -83.23410251
0f00aeea6e, Position4, 0d003b43ca, 1100bbad79, 33.563545654, -83.23102020
1100bbad79, Position5, 0f00aeea6e, 01038dd604, 33.560147296, -83.20005612
01038dd604, Position6, 1100bbad79, 11006d0b12,
11006d0b12, Position7, 01038dd604, 01066d9e00,
01066d9e00, Position8, 11006d0b12, 041575d516,
041575d516, Position9, 01066d9e00, ,
```

In the acquisition example, an exception from the intended route was noted. The debugging notes imply that the second transponder was acquired prior to the acquisition of the first, assuming that the route taken was canonical. Examining the database confirms that this was the case. The data reporting software noted this exception, but continued with the remaining processing.

Route Reporting

NOTE *** Transponders Acquired out of expected sequence ****

	Distance(km)	Time(s)	Rate(km/h)
*Position2 --> Position1	0.3319	77.5	1.54
*Position1 --> Position3	0.2652	165.6	0.57
Position3 --> Position4	1.1120	210.5	1.90
Position4 --> Position5	2.8940	388.3	2.68
Totals	4.60	2.33h	1.67(av)

5.3 ANALYSIS

The operating system and software on the portable Gumstix™ computer instantiates and enables the transceiver within approximately 3 seconds. Data are acquired from the transponders in less than one second, once the transponder and transceiver are within approximately a 10cm range of each other. Data is acquired for the identifier and a timestamp for each transponder as it enters and exits RF range of the transceiver. The software will not acquire partial data from a misread. Consequently, data that is received can be presumed to be an accurate representation of the data transmitted. Since a separate database maintains the details of each installed transponder, only those which are valid by comparison between the actual read and the data in the database are considered to be legitimate. Any encounter with transponders that are not in the database are considered spurious and are ignored.

The data items compared can offer the examiner an indication of the route, time and rate of speed traveled without a compromise of personally-identifiable information by performing a small number of calculations.

When the portable computer was operating on 3 AAA batteries, the computer and transceiver combination could be operated for a period of about 4.5 hours. A more sophisticated power management system involving managing sleep cycles of

the computer could improve power utilization. Rechargeable batteries are an obvious improvement for repeated reuse as well as an appropriate environmentally-suitable case for housing the computer and transceiver.

CHAPTER 6

CONCLUSIONS

A portable interrogator system that enables the collection and processing of data acquired from fixed RFID transponders using hardware that is commercially available can indeed be developed. Software must be created to mate the computational component to the transceiver component, but results in a very compact device for gathering the data from a set of transponders. This data can be uploaded to a host computer for post processing. On the host computer, software validates the actual route, time, and distance required by a user against the expected route. Exceptions to the intended route can be recorded and noted for reporting. These activities can be performed without divulging confidential information, assuming that the data on the host computer is managed in accordance with standard security principles.

While the inexpensive Class 1 RFID subsystem used was clearly not sufficient to accommodate the desired long-range operations, this was an expected limitation. It was nevertheless sufficient to demonstrate a proof of concept for future work, with the possibility of replacing this subsystem with Class 4 RFID components.

By fixing the locations of transponders and developing a suitable mobile transceiver, privacy issues that involve enumeration of individuals using unique transponder identifiers can be ameliorated. This scheme does not require the custom manufacture or programming of transponders. Rather, inexpensive and

readily-available preprogrammed units are registered by adding them to a database prior to installation and use.

CHAPTER 7

FUTURE WORK

This work is illustrative of the possibility of creating a portable RFID data collection system. There are a number of logical extensions to this work, such as extending read range by using different RFID transceivers and transponders. An ideal actual implementation would enable the user to pass within a few meters of the transponder, yet still acquire its data. Implementation of a collision management system becomes more significant as the portability of additional transceivers and transponders are grouped within the same read range. In addition, discrimination protocols would be necessary to avoid or manage duplicate reads.

An extension of this technology is the combination of RFID technology with sensor networks. A related implementation has been evolving at the Chaos Computer Camp in Finowfurt, Germany for the past two years [20]. Milosch Meriac, Harald Welte and Brita Meriac developed a 2.4GHz active RFID system that was used to track attendees at their meetings. Their OpenBeacon design could be modified somewhat using some of the ideas suggested in this paper to enable a system for tracking hikers of national parks with user consent. When operating as a sensor network, the system could not only track the progress along a trail for the benefit of the user, but could also forward the general location of hikers back to a Ranger station. As a hiker moves along the trail, in addition to the information the hiker

gathers from RFID-equipped mileposts, the active RFID transponders might forward the contact event with the hiker to adjacent sensors. It would be possible to differentially determine the location of a node within the margin of error afforded by the design.

A recent examination of injuries in Hawaii Volcanoes National Park indicated that as many as 59% of the park's users suffered some kind of injury during their visit [12]. In view of the number of hiker injuries and fatalities that occur in the United States National Parks each year, such a device could provide a general location of a missing hiker to authorities, enabling search and rescue teams to more quickly dispatch to the general area where a hiker was last known to have been located. A more efficient response such as one offered by this technology could be instrumental in saving lives. In addition, the costs associated with search and rescue would assuredly be minimized by targeting a more specific search area. It is conceded that certain privacy issues associated with such an implementation exist. In areas where accidents are commonplace, such a system would benefit both the user and the facilitator of the hiking area. If active technologies are employed, it is reasonable to implement nodes that are computationally robust to enable the encryption of data transmitted between them, suggesting a potential means for ameliorating personal privacy concerns.

BIBLIOGRAPHY

- [1] Associated Press. "RFID chips help track Katrina dead.", 1995 Copy available at <http://www.msnbc.msn.com/id/9514138/>
- [2] "CASPIAN Wal-Mart Protest." RFID Today (from a CASPIAN Press Release), October 21, 2005.
<http://rfidtoday.blogspot.com/2005/10/rfid-caspian-wal-mart-protest.html>
- [3] Collins, Jonathan. "Lost and Found in Legoland." RFID Journal. 28 April 2004. <http://rfidjournal.com/article/view/921/1/1/>
- [4] Educational and Public Outreach Report 2005. NASA Ames Research Center.
<http://nai.nasa.gov/team/index.cfm?page=epo&teamID=22&year=7>
- [5] "EM4102 Data Sheet." EM Microelectronic-Marin SA. Marin, Switzerland. 2006.
http://www.emmicroelectronic.com/webfiles/Product/RFID/DS/EM4102_DS.pdf
- [6] Finkenzeller, K. *RFID Handbook*. John Wiley and Sons. West Sussex, England. 2003.
- [7] Fok, Chien-Liang. "Agulla." MobiLab, Washington University, St. Louis. 2006
<http://mobilab.wustl.edu/projects/agilla/>.
- [8] Gellert, W., Gottwald, S., Hellwich, M., Küstner, H., and Kästner, H., *The VNR Concise Encyclopedia of Mathematics, 2nd ed.* Van Nostrand Reinhold: New York, 1989.

- [9] "Gumstix Product Information." Gumstix web site. 2007.
<http://www.gumstix.com/>
- [10] "Gumstix Verdex Document Wiki" Gumstix. 2007
<http://docwiki.gumstix.org/Verdex>
- [11] Hefeeda, M. "Forest Fire Modeling and Early Detection using Wireless Sensor Networks." Simon Fraser University, Canada, 2007.
- [12] Heggie, T. and Heggie, T. "Viewing Lava Safely: An Epidemiology of Hiker Injury and Illness in Hawaii Volcanoes National Park." *Wilderness and Environmental Medicine*: Vol. 15, No. 2, 2001, pp. 77-81.
- [13] "Intel Mote." Intel Research. 2007.
<http://www.intel.com/research/exploratory/motes.htm>
- [14] O'Connor, Mary. "Great Wolf Water Park Launches RFID."
<http://www.rfidjournal.com/article/view/2211/1/1/>
- [15] "The Nike+iPod" NikePlus web site. 2007. <http://nikeplus.nike.com/nikeplus>
- [16] *IEEE Standard for Safety Levels with Respect to Human Exposure to Radio Frequency Electromagnetic Fields, 3 KHz to 300 GHz*, IEEE Standard C95.1-1991, Institute of Electrical and Electronics Engineers, New York, 1992.
- [17] Landt, J. *The History of RFID*. IEEE Potentials, Vol 24, No. 4., 2005
- [18] Layer8. "Sand, sun and RFID?: The high-tech networked beach is coming soon." Network World. 25 July 2007.
<http://www.networkworld.com/community/node/17841>

- [19] Lyman, J. "Hacker Cracks, Clones RFID Passport." TechNewsWorld. August 7, 2006.
- [20] "Open 2.4GHz RFID" Open Beacon Project. 2007.
<http://www.openbeacon.org>.
- [21] "PC/104 Specifications." PC/104 Embedded Consortium.
[http://www.pc104.org/technology/PDF/PC104 Spec v2.5.pdf](http://www.pc104.org/technology/PDF/PC104%20Spec%20v2.5.pdf)
- [22] "Phidgets RFID Reader" Trossen Robotics. 2007.
<http://www.trossenrobotics.com/store/p/3606-PhidgetsRFID-Reader-Only-USB-.aspx>
- [23] Saponas, T.S., Lester, J, Hartung, C., and Kohno, Tadayoshi. *Devices That Tell On You: The Nike+iPod Sport Kit*. Department of Computer Science and Engineering. University of Washington, Seattle, WA, 2006
<http://www.cs.washington.edu/research/systems/privacy.html>
- [24] Sinnott, R.W., "Virtues of the Haversine," Sky and Telescope, vol. 68, no. 2, 1984, p. 159.
- [25] "Tag-It Transponder Inlays," Transponder Inlay Reference Guide, Document #SCBU004. Texas Instruments. December 2005.
- [26] Tanenbaum, A., Gamage, C., and Crispo, B., "Taking Sensor Networks from the Lab to the Jungle," Computer, vol. 39, no. 8, pp. 98-100, Aug., 2006 "
- [27] Thornton, F. *RFID Security*. Syngress. Rockville, MA, 2006.
- [28] *Title 47, FCC Part 15* Federal Communications Commission. 47 CFR 15, as of September 2007. <http://ecfr.gpoaccess.gov>

- [29] Weis, S. "Security and Privacy in Radio Frequency Devices", Master Thesis, Massachusetts Institute of Technology, Cambridge, MA, 2003.
- [30] Zetter, Kim. "School RFID Plan Gets an F." Wired Magazine. 10 February 2005. <http://www.wired.com/politics/security/news/2005/02/66554>

APPENDIX A

INTERROGATOR SOURCE CODE

```

/* rfidio.c */

#include <stdio.h>
#include <unistd.h>
#include <signal.h>
#include <sys/time.h>
#include <time.h>

#include <phidget21.h>
#include "cphidgetconstants.h"

#include "rfchip.h"

struct timeval basetime;

static double mydifftime(struct timeval *tv)
{
    return
        (double)tv->tv_sec - (double)basetime.tv_sec +
        ((double)tv->tv_usec - (double)basetime.tv_usec) / 1e6;
}

static int RFID_Handler(CPhidgetRFIDHandle RFID, void *userptr,
unsigned char *buf)
{
    int stat;
    struct timeval tv;

    //Tag is represented as a 10-digit hex number - array of 5
    // unsigned bytes
    //Here we tell printf to display two characters for each byte
    // - ie 0x00 - 0xFF
    //the '0' in the '%02x' just make it display any leading zeros.

    //fix_tag(RFID, buf);

    gettimeofday(&tv, NULL);

    printf("Got:  %02x%02x%02x%02x%02x %6.6f %s\n",
        buf[0], buf[1], buf[2], buf[3], buf[4],
        mydifftime(&tv),
        search_tab((char *) buf));

    stat = CPhidgetRFID_setLEDOn(RFID, PTRUE);
    if (stat != EPHIDGET_OK)

```

```
        printf("set led on error\n");

    return 0;
}

static int RFIDLost_Handler(CPhidgetRFIDHandle RFID, void *userptr,
unsigned char *buf)
{
    int stat;
    struct timeval tv;

    gettimeofday(&tv, NULL);

    printf("Lost: %02x%02x%02x%02x%02x %6.6f\n",
        buf[0], buf[1], buf[2], buf[3], buf[4],
        mydifftime(&tv));

    stat = CPhidgetRFID_setLEDOn(RFID, PFALSE);

    if (stat != EPHIDGET_OK)
        printf("set led off error\n");

    return 0;
}

static int AttachHandler(CPhidgetHandle ph, void *userptr)
{
    printf("Device attached\n");
    return 0;
}

static int DetachHandler(CPhidgetHandle ph, void *userptr)
{
    printf("Device detached\n");
    return 0;
}

static int ErrorHandler(CPhidgetHandle rfdev, void *userptr,
int errcode, const char *errmsg)
{
    printf("Phidget Error #%d: %s\n", errcode, errmsg);
    return 0;
}
```

```
// use sig INT interrupt (ctrl-c) to shutdown nice and proper

volatile int done = 0;

static void sighandler(int sig)
{
    done = 1;
}

int main(int argc, char **argv)
{
    CPhidgetRFIDHandle rfdev;

    int stat, ser;
    char *nam;
    int i, numout, state;

    {
        struct sigaction s;

        s.sa_handler = sighandler;
        sigemptyset(&s.sa_mask);
        s.sa_flags = 0;
        sigaction(SIGINT, &s, NULL);
    }

    gettimeofday(&basetime, NULL);

    load_tab();
    printf("start\n");
    stat = CPhidgetRFID_create(&rfdev);
    if (stat != EPHIDGET_OK)
        printf("create stat=%d\n", stat);

    CPhidgetRFID_set_OnTag_Handler(rfdev, RFID_Handler, NULL);
    CPhidgetRFID_set_OnTagLost_Handler(rfdev, RFIDLost_Handler
        , NULL);

    CPhidget_set_OnAttach_Handler((CPhidgetHandle) rfdev,
        AttachHandler, NULL);
}
```

```

CPhidget_set_OnDetach_Handler((CPhidgetHandle) rfdev,
DetachHandler, NULL);

CPhidget_set_OnError_Handler((CPhidgetHandle) rfdev,
ErrorHandler, NULL);

stat = CPhidget_open((CPhidgetHandle) rfdev, -1);
if (stat != EPHIDGET_OK) {
    printf("have open stat = %d\n", stat);
    if (getuid())
        printf("This isn't root,"
            " You probably need to use sudo or something\n");
}

stat = CPhidget_waitForAttachment((CPhidgetHandle) rfdev,
1000);
printf("have waited stat=%s\n",
    stat == EPHIDGET_OK ? "OK" :
    stat == EPHIDGET_INVALIDARG ? "error" :
    stat == EPHIDGET_TIMEOUT ? "timeout" : "unknown");

if (stat == EPHIDGET_TIMEOUT)
    if (getuid())
        printf("This isn't root,"
            " You probably need to use sudo or something\n");

if (stat == EPHIDGET_OK) {

    stat =
        CPhidget_getDeviceName((CPhidgetHandle) rfdev,
            (const char **) &nam);
    if (stat != EPHIDGET_OK)
        printf("get device name error\n");

    printf("device name is %s\n", nam);

    stat = CPhidget_getSerialNumber((CPhidgetHandle) rfdev, &ser);
    if (stat != EPHIDGET_OK)
        printf("get serial number error\n");

    printf("serial number is %d\n", ser);

    stat = CPhidgetRFID_getNumOutputs(rfdev, &numout);

```

```

if (stat != EPHIDGET_OK)
    printf("set get num out error\n");

printf("%d outputs = ", numout);

for (i = 0; i < numout; i++) {
    stat = CPhidgetRFID_getOutputState(rfdev, i, &state);

    if (stat != EPHIDGET_OK)
        printf("get output state error\n");
    printf(" (%d = %s)", i,
        (state == PTRUE) ? "true" : (state == PFALSE)
? "false" :
        // (state = PUNKNOWN) ? "unknown" :
        "undefined");
    }
printf("\n");

stat = CPhidgetRFID_setAntennaOn(rfdev, PTRUE);
if (stat != EPHIDGET_OK)
    printf("set antenna on error\n");

{
    char x = '-';
    while (!done) {
        printf("%c\b", x);
        fflush(stdout);
        x = (x == '-') ? '/' : (x == '/') ? '\\' : '-';

        sleep(1);
    }
}

stat = CPhidgetRFID_setAntennaOn(rfdev, PFALSE);
if (stat != EPHIDGET_OK)
    printf("set antenna off error\n");

}

stat = CPhidget_close((CPhidgetHandle) rfdev);
if (stat != EPHIDGET_OK)
    printf("close error\n");
stat = CPhidget_delete((CPhidgetHandle) rfdev);

```

```
    if (stat != EPHIDGET_OK)
        printf("delete error\n");

    printf("Done.\n");

    return 0;
}

/* end of rfidio.c */

-----

/* rfchip.h */

void load_tab(void);

char *search_tab(char id[5]);

/* end of rfchip.h */

-----

/* rfchips.c */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include "rfchip.h"

#define tabnam "transponders.dat"

#define LINELEN 80

typedef struct {
    char id[5];
    char *desc;
} tabentry;

int maxents = 0;
int numents = 0;

tabentry *tab = NULL;
```



```
static char *dyncopy(char *str)
{
    char *tmp = malloc(strlen(str) + 1);
    if (tmp == NULL) {
        fprintf(stderr, "out of memory\n");
        exit(1);
    }
    strcpy(tmp, str);
    return tmp;
}
```

```
int hexval(char c)
{
    switch (c) {
        case '0':
            return 0;
        case '1':
            return 1;
        case '2':
            return 2;
        case '3':
            return 3;
        case '4':
            return 4;
        case '5':
            return 5;
        case '6':
            return 6;
        case '7':
            return 7;
        case '8':
            return 8;
        case '9':
            return 9;
        case 'a':
        case 'A':
            return 10;
        case 'b':
        case 'B':
            return 11;
        case 'c':
        case 'C':
            return 12;
    }
```

```
    case 'd':
    case 'D':
        return 13;
    case 'e':
    case 'E':
        return 14;
    case 'f':
    case 'F':
        return 15;
    }
    return -1;
}

void load_tab(void)
{
    FILE *fp;
    char line[LINELEN];

    fp = fopen(tabnam, "r");
    if (fp == NULL) {
        perror("can't open " tabnam);
        return;
    }

    tab = malloc(sizeof(tabentry) * 32);
    if (tab == NULL) {
        fprintf(stderr, "out of memory\n");
        exit(1);
    }
    maxents = 32;

    while (fgets(line, LINELEN, fp) != NULL) {
        int i;
        char *tmp;

        if (line[strlen(line) - 1] == '\n')
            line[strlen(line) - 1] = 0;

        if (strlen(line) > 6) {

            if (numents == maxents) {
                tmp = realloc(tab, sizeof(tabentry) * maxents * 2);
                if (tmp == NULL) {
```

```
        fprintf(stderr, "out of memory\n");
        exit(1);
    }
    tab = (tabentry *) tmp;
    maxents *= 2;
}

tmp = line;
for (i = 0; i < 5; i++) {
    int x = hexval(*tmp++);
    x = x * 16 + hexval(*tmp++);

    tab[numents].id[i] = x;
}
while (*tmp && isspace(*tmp))
    tmp++;
tab[numents].desc = dyncopy(tmp);
numents++;
}
}

fclose(fp);
}

char *search_tab(char id[5])
{
    int i = 0;

    while (i < numents) {

        if (memcmp(id, tab[i].id, 5) == 0)
            return tab[i].desc;
        i++;
    }

    return "\aUnknown Transponder";
}

/* end of file */
```