MNEMOSYNE Technologies
School of Engineering Science
Burnaby, BC
V5A 1S6
ensc440-spa@sfu.ca

April 21st , 2005

Dr. Lakshman One
School of Engineering Science
Simon Fraser University
Burnaby, British Columbia
V5A 1S6

Re: ENSC 440 Post Mortem for an anti-theft/reminder transmitter

Dear Dr. One,

The following document, *Post Mortem for an anti-theft/reminder transmitter*, provides an overview of the final implementation of our project, the SafeGuard. The SafeGuard is an RF based, wireless tether which allows users to tag their personal items with RF transmitters. When these tagged items exit a predefined radius from the receiver, the receiver which is worn by the user, will notify the user which object was left behind.

The post mortem will briefly outline the finalized basic components of our project, and our variances in the final product between our functional specifications and design specifications. A comparison between our original projected budget and timeline and the actual budget and timeline will also be included. A review of what we could have done differently in generating the project will also be included. We will also cover potential future improvements to our project.

The design specifications will be used in the future to help us in the construction of our project.

Regards,

Carson Hammoser
Chief Technical Officer (CTO)
Mnemosyne Technologies

Enclosure: *Design Specifications for an anti-theft/reminder transmitter*,

MNEMOSYNE Technologies

POST MORTEM

Project Team: Carson Hammoser
Peter Lin
Albert Uang
Samuel Wong

Contact Person: Peter Lin
ensc440-spa@sfu.ca

Submitted to: Lakshman One
Michael Sjoerdsma
Scott Logie
Tony Ottaviani
Amir Niroumand
Steve Whitmore
School of Engineering Science
Simon Fraser University

Issue date: April 21st, 2005

Executive Summary

Have you ever forgotten something or left something behind? Have you ever accidentally dropped something important? From keys and wallets, to glasses cases and cell phones, everyday people lose track of things. Items can be forgotten at work or home, or objects can be accidentally dropped or lost. Locating these items can often be frustrating as well as time consuming. Replacing them can be costly. Preventing these losses in the first place can eliminate the stress and aggravation associated with looking for these lost items. These annoying situations can be avoided by simply utilizing a small tracker which notifies you when you've left an object behind.

The application of a short range tracking system extends beyond helping you recover your lost items into the realm of security, protecting you from having your items stolen; a simple solution presents itself in terms of a range based security device, ensuring that your valuables remain close by at all times. This device can even be applied at a larger level in order to ensure that office equipment is not misused or removed from a specific location. Alternatively attaching tags to children and animals will allow you to keep track of your children and pets without having to attach chains or other physical tethers. This device will allow you to make sure that your kids stay close when you're in a busy shopping mall, or it will allow you to know when your dog escapes from the back yard.

Mnemosyne Technologies is proposing a project, the SafeGuard, where we will design and construct a multi-object tracking device. Utilizing our SafeGuard tags you can keep track of your personal effects. A receiver in the form of a watch or a keychain will track a specified number of tags, and notify you if you walk away from your tagged items. This warning will allow you to find out what you have left behind, since you will still be relatively close to the lost object.

The following document will focus on the construction of the final product and the deviations from the design specifications. Additionally, a review of our design process, budget constraints, and potential improvements will be discussed as well.

Table of Contents

Executive Summary ii
Table of Contents iii
Acronyms: iv
1. Introduction 1
    1.1. Scope: 1
2. Design Deviations 1
    2.1 Tag Unit 1
        2.1.1 Antenna 1
        2.1.2 Microcontroller 3
        2.1.3 Power Supply 5
        2.1.4 Packaging 5
    2.2 Receiver 6
        2.2.1 Receiver Chip 6
        2.2.2 Antennae 7
        2.2.3 Microcontroller 7
        2.2.4 Interface 9
        2.2.5 Power Supply 9
        2.2.6 Packaging 10
    2.3 Problems and Resolutions 10
3. Budget and Timeline Comparison 12
    3.1 Budget Comparison 12
    3.2 Timeline Comparison 12
4. What We Would Do Differently 15
5. Future Improvements 15
    5.1 Tag 15
    5.2 Receiver 16
6. Personal Reflections 16
7. Conclusion: 17

Acronyms:

CDMA: Code Division Multiple Access

DDRAM: Display Data Random Access Memory

ISP: In System Programming

LCD: Liquid Crystal Display

LED: Light Emitting Diode

PCB: Printed Circuit Board

RFID: Radio Frequency IDentification

RSSI: Received Signal Strength Indicator

SRAM: Static Random Access Memory

1. Introduction

One of the many facts of life is that people lose things. People forget items or leave objects behind. Statistics show that people lose hundreds of thousands of objects every year. The SafeGuard is designed to help prevent a user from misplacing these objects, by reminding him or her that an item has been left behind while he or she is still relatively close to the item.

1.1. Scope:

The following document will focus on the construction of the final product and the deviations from the design specifications. Additionally, a review of our design process, budget constraints, and potential improvements will be discussed as well.

2. Design Deviations

2.1 Tag Unit

The tag unit will consist of a small RF transmitter which will be attached to a variety of personal affects. In our original design we intended to transmit a unique five digit code which the receiver would use to identify the tag. However, due to the complexity involved in implementing such a scheme, and taking into consideration noise and interference, this implementation was laid aside for future consideration. Additionally, we originally intended to utilize one carrier frequency for all tags; however, this implementation would require a CDMA scheme be used to ensure the multiple tag transmissions would not interfere with each other. Thus, in our prototype implementation, each tag has a unique carrier frequency and continually transmits a simple square wave pulse at a predefined frequency. The following sections outline some of the more significant deviations in our prototype from our design specifications and functional specifications.

2.1.1 Antenna

During our initial prototyping stage, we utilized a “home made” transmitter tag unit which placed the receiver chip and all required external components on a PCB board we constructed from scratch ourselves. Using this unit, we attempted to prototype the transmitter and receiver plugging the units into breadboards and connecting them to the microcontrollers. These homemade transmitters had significant problems with noise and body effects. Our final prototype which utilized fabricated boards did not experience the same problems and could transmit a relatively undistorted signal at distances slightly exceeding our range limit without an antenna. During testing we did test two antenna types which proved significantly resilient to noise: a trapezoidal patch, and a slotted antenna (Figure 1: Trapezoidal Patch Antennae & Figure 2: Slotted Antennae). Since at present, we are utilizing the transmission strength to limit the range of the tags, our prototype did not utilize antennas. However, if future implementations utilize an RSSI in order to determine range, a trapezoidal patch antenna would most likely be utilized do to its compact size and resistance to noise and body effects. [1]

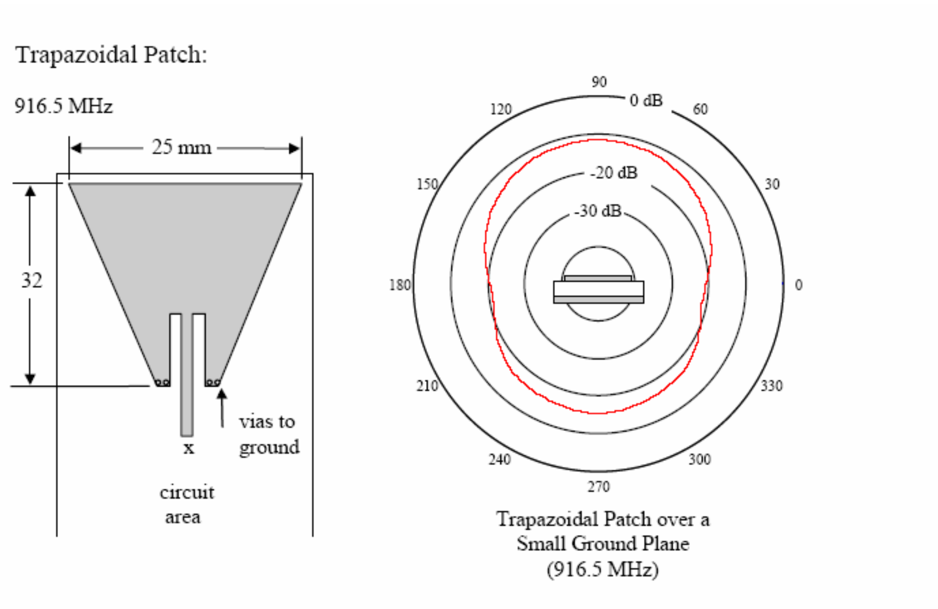

Figure 1: Trapezoidal Patch Antennae

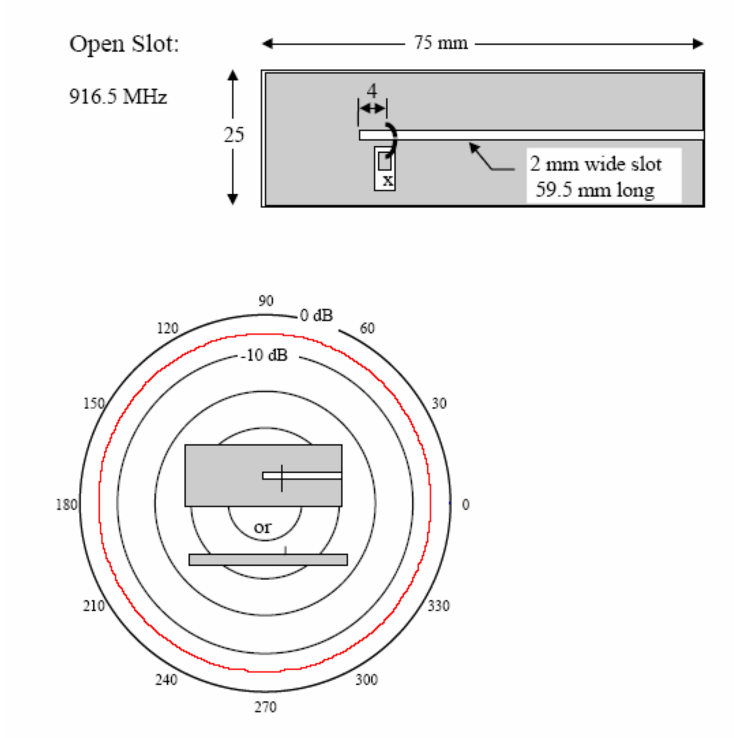

Figure 2: Slotted Antennae

2.1.2 Microcontroller

2.1.2.1. Hardware

One of the most significant changes made in the transmitter tag was the microcontroller used in the tag. Initially we intended to use the ATTiny12, however due to its limited code size we decided to change microcontrollers to the ATTiny2313 (Figure 3: ATTiny2313 Package). The ATTiny2313 not only increased the available amount of memory from 1K to 2K but also allowed us to program in C, as the ATTiny2313 had SRAM, which is required by the compiler. One detriment to using the ATTiny2313 is the extra physical space required. [2]

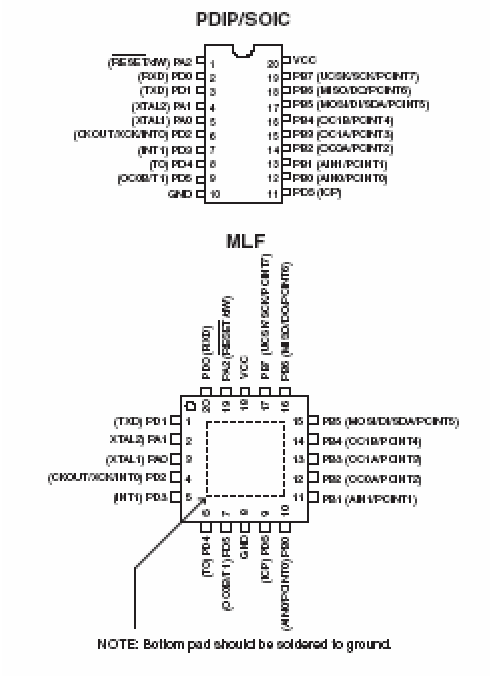

Figure 3: ATTiny2313 Package

2.1.2.2. Software

Another area where we deviated from our design specifications was the software component of the tag. As previously outlined the transmitter does not broadcast a fixed tag ID, but rather transmits a constant digital pulse at a specific frequency. This greatly simplifies the software component as all that is required of the microcontroller is to initialize the transmitter chip and then periodically send the transmitter a pulse. Due to time limitations we were unable to implement a sleep and wake function, which would effectively power down the CC1000 and microcontroller. Figure 4: Transmitter Software Flow Chart below illustrates the transmitter algorithm followed.

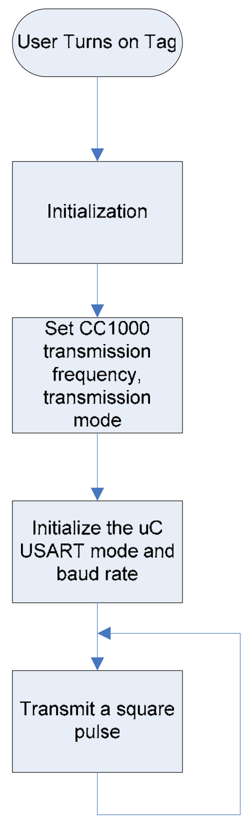

Figure 4: Transmitter Software Flow Chart

2.1.3 Power Supply

Our original choice for a battery was the BR 2477A, but in the end we switched to the CR 2032 (Figure 5: CR2032) as the power supply of the transmitter. One of the primary reasons for switching supplies was the size constraints. The CR 2032 is significantly smaller than the BR 2477As. The CR 2032 is still capable of supplying 3 Vs, which meets our voltage requirement. Additionally, the CR 2032's voltage drop off curve (Figure 6: Characteristics of CR2032) illustrates a more consistent voltage which is closer to 3V at room temperature, whereas the BR 2477A's supply curve requires very high temperatures around 100 °C in order to provide an ideal voltage drop-off. [3]

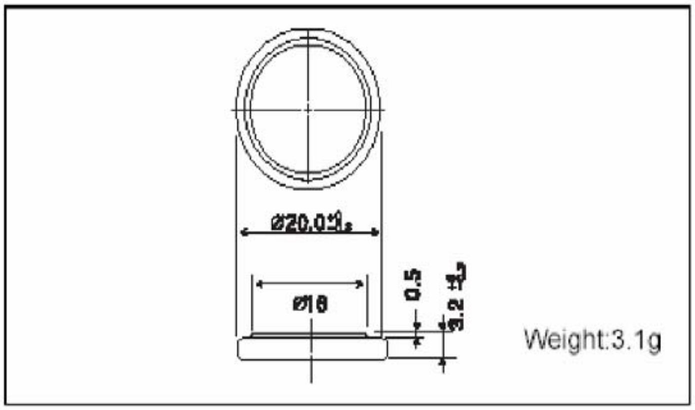

Figure 5: CR2032

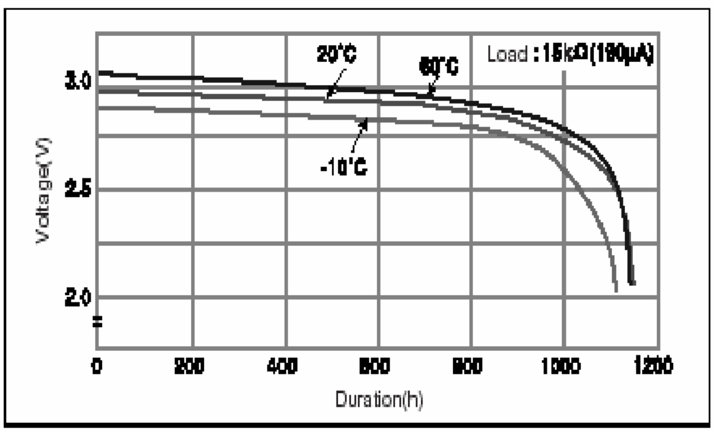

Figure 6: Characteristics of CR2032

2.1.4 Packaging

For convenience as well as due to sizing constraints, we were selected a different package, the Hammond 1551 (Figure 7: Hammond Package Used), for the transmitter tags. However, due to certain limitations in terms of our ability machine the PCB boards we could not accommodate the transmitters into the newly selected package. Additionally, since this implementation was a prototype, we opted to make a slightly larger board which would allow for more flexibility in selecting and interchanging parts. Future implementations should be able to utilize a significantly smaller package. [4]

Figure 7: Hammond Package Used

2.2 Receiver

The receiver unit consists of a reasonably small RF receiver which a user will preferably wear. Users will be able to program in an 8 character description for each of the five tags, using a five button joystick, and an LCD display. There are only two major deviations in our receiver prototype from our design specifications: the receiver chip we use and the software implementation.

2.2.1 Receiver Chip

Due to time constraints and difficulties experienced with initializing the CC1000 into receiver mode, we opted for a different receiver chip, the Linx HP-II (Figure 8: RXM - 900-HPII). Our problems with using the CC1000 as a receiver could potentially have stemmed from flaws in our “home-made” proto-boards; however, due to time constraints we were unable to test this hypothesis. The HP-II proved to be a much simpler chip to use, requiring no pre-configuration, while providing a very simple interface. The interfaced utilized three control lines to select which channel (carrier frequency) the receiver would be tuned to. This simple interface also proved to be one of the more drawbacks of the HPII; the HPII is only capable of receiving on 7 predefined channels, limiting the flexibility of the receiver.

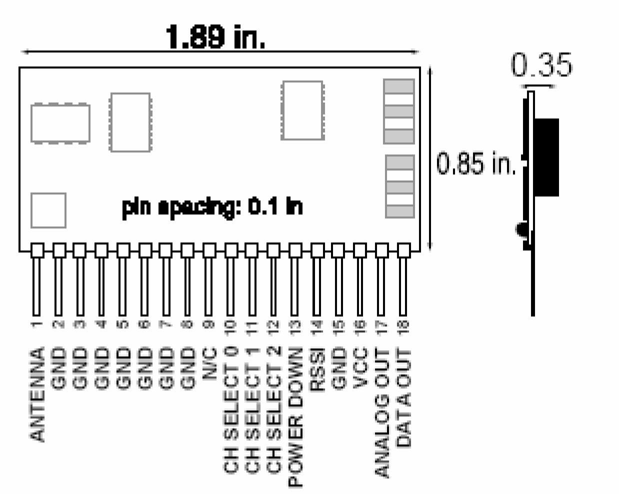

Figure 8: RXM -900-HPII

Additional features that the HPII provided but which we were unable to utilize due to time limitations are an RSSI and a Power Down feature. Future implementations would hopefully utilize a CC1000, as it provides a wider spectrum of frequencies to use, while being significantly smaller than the HPII. However regardless of which receiver chip is used, a power down function to conserve receiver power, as well as an RSSI function for more accurate ranging should be implemented. [5]

2.2.2 Antennae

Like the transmitters, our initial attempts to use the homemade CC1000 proto-boards proved quite problematic in terms of body-effects and noise. When we switched receivers to the HPII, the received signals seemed to be relatively clear without any antenna. Again, since our range limitation is enforced by the signal strength as opposed to the RSSI, the receiver does not utilize an antenna. Tests using the slotted antenna and trapezoidal patch antenna illustrate that both antennas do improve the quality of the received signal. Future implementations where ranging is accomplished using the RSSI would most likely utilize a patch antenna because of its small size.

2.2.3 Microcontroller

2.2.3.1. Software

Due to the variation in the transmitter software where instead of sending a unique 5 digit ID, a constant pulse is sent, the receiver software must be altered in order to decode and detect the correct signal. Additionally, since the receiver chip no longer requires any setup or configuration on the part of the microcontroller, that component can be removed from the software. The new implementation involves periodically switching receiver frequencies and checking to see that the constant pulse is being received. Utilizing the input capture function, we connect the input capture pin to the output of the receiver (the decoded received signal). On each rising edge we check the width of each pulse, and if the width matches a predefined value (i.e. the pulse occurs at the correct frequency within a reasonable degree of error), we increment a counter. If enough pulses of correct width are detected upon the change of channels, the tag is considered present, it the number of pulses detected are insufficient, the tag is noted as missing and the alarm is set off. Figure 9: Receiver Software High Level Design below outlines the operation of the receiver.

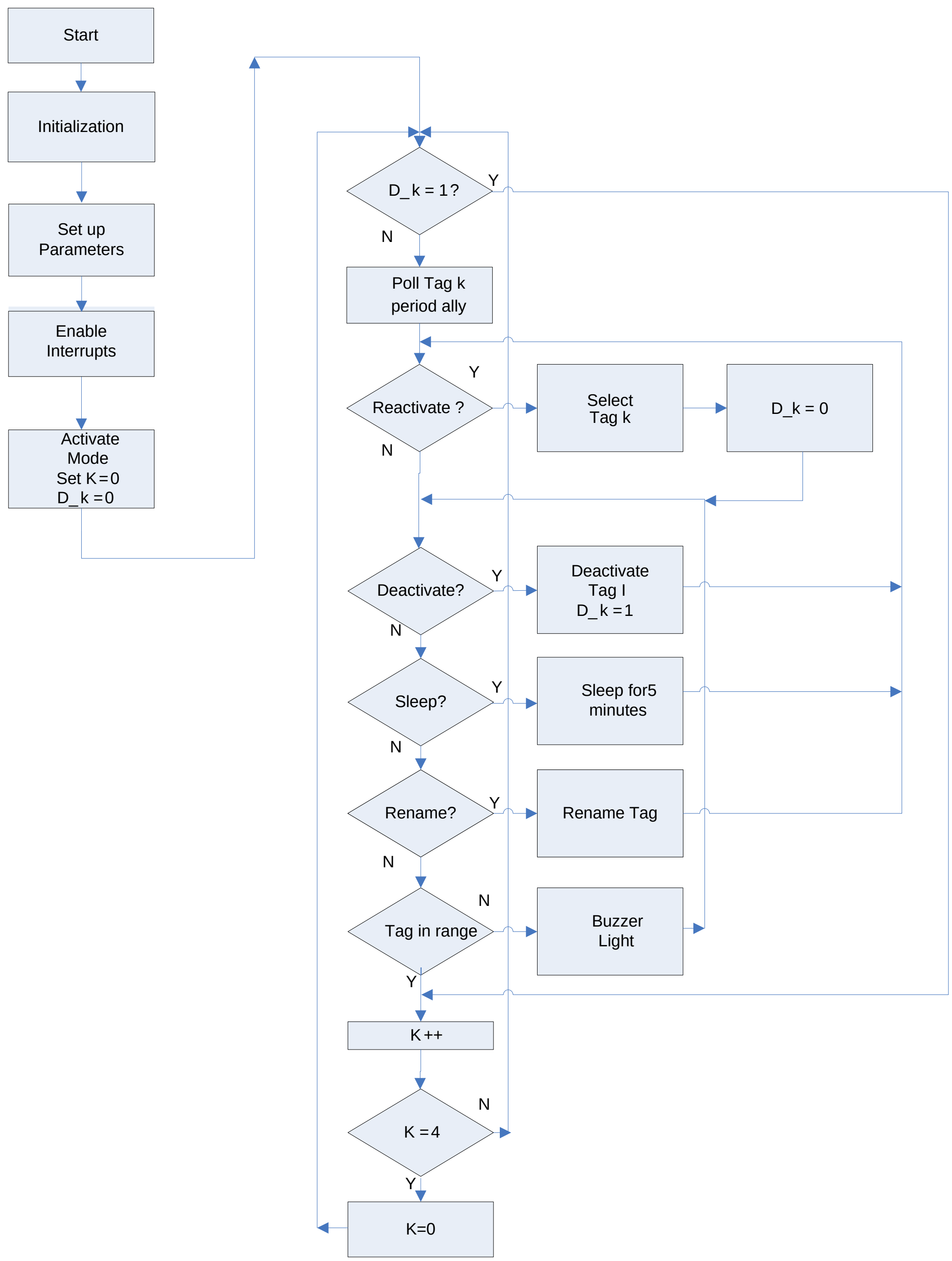

Figure 9: Receiver Software High Level Design

2.2.4 Interface

In terms of interface, we used all the specified devices outlined in the design specifications with the exception of the LED. Originally, we intended to use an LED as a visual, silent indicator that a tag was missing. Instead of using an LED, we opted to attach a small motor which vibrates when set off. Despite the added power required, the motor better serves as an indicator as the receiver does not have to be visible in order to alter the user that a tag is missing.

2.2.5 Power Supply

Our original choice for a power supply was the BR 2477A, but in the end we chose the CR 2477(Figure 10: Size of CR2477). The BR 2477A and the CR 2477 are the same size and voltage; however, the continuous current drain of BR 2477A proved to be too low (0.04mA) for the receiver. The continuous current drain of CR 2477 is sufficient (0.2mA) for the receiver (Figure 11: Characteristics of CR2477). Therefore, we switched to CR 2477. [6]

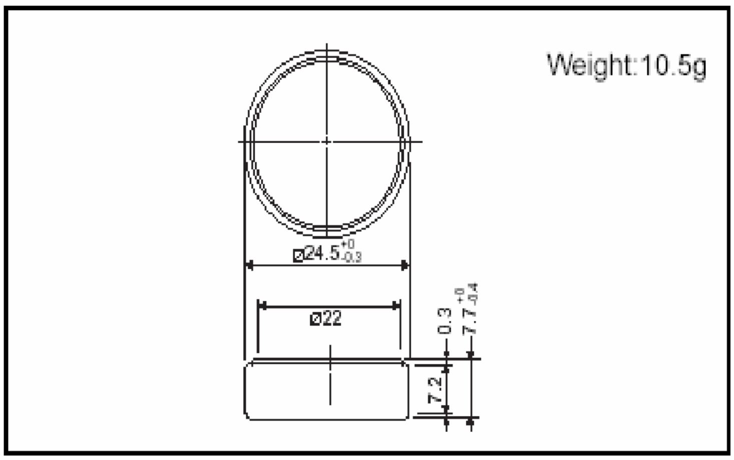

Figure 10: Size of CR2477

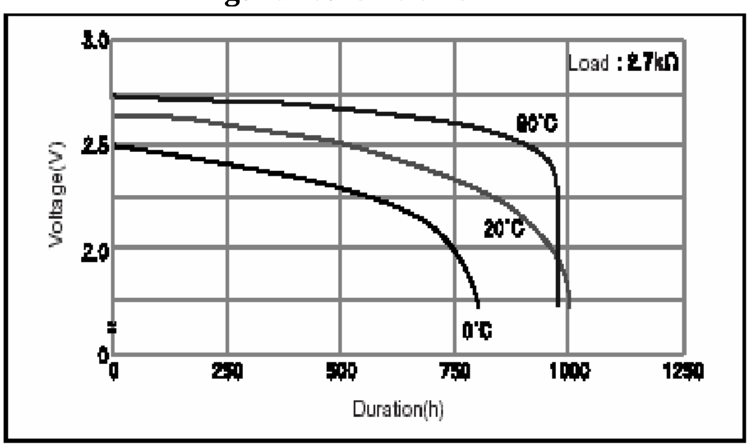

Figure 11: Characteristics of CR2477

2.2.6 Packaging

Like the packaging for the transmitter tags, we switched cases due to time constraints and convenience. We chose the Hammond 1593 (Figure 12: 1593 Package) due to its construction and shape. However, due to certain restrictions in our PCB, this Hammond case also proved unsuitable. Our current prototype PCB utilizes headers for programming and port selection in order to provide greater flexibility in terms of parts. In order to choose a suitable case a finalized PCB is required. [7]

Figure 12: 1593 Package

2.3 Problems and Resolutions

Throughout this project we experienced a wide variety of problems and generated a large number of solutions. Listing all these problems would fall well beyond the scope of this report. Listed below are the more significant problems and their resolutions:

Noise and Body Effects:

Among the most major and worst problems involved with this project was attempting to acquire a clean signal using the transmitters we fabricated. Using these home-made surface mount adapters, we managed to transmit and receive simple signals; however, these signals were very sensitive to noise and body effects. Initially we believed that this problem could be solved using properly tuned antennas. We were correct in this assumption, as properly fabricated antennas did significantly improve the results. However, we did later find out that the real source of the problem was our home-made transmitters. The final conclusion was that the transmitter and receiver could function relatively well without any antennas for the range we were dealing with. Though these systems were still susceptible to body effects and noise they were significantly more resilient compared to our original home-made boards.

LCD Initialization and Reading from the LCD:

In using the LCD, we faced several challenges in terms of ensuring that the LCD could be initialized properly through software as well as from reading characters from the LCD.

The LCD can be initialized through hardware, by providing a sufficient supply voltage at startup; however, since the LCD is being powered from a battery, it cannot be guaranteed that the startup voltage required could be met. In order to initialize the LCD by software, certain values must be written to the driver chip's DDRAM and then held for a time. Finding the correct timing sequence proved to be quite troublesome despite the well outlined instructions. Finally through trial and error, we successfully initialized the LCD.

In reading from the LCD, we face a slightly more interesting problem. Initially having written the code, we managed to read from the LCD without any problems. However, when integrating the code with a working framework, the original code fragment seemed to stop working. This problem we found was linked to the processor speed of the internal oscillator we used to clock the microcontroller. We later found that when running the microcontroller at slower speeds we were leaving sufficient time for the driver controller to complete all it operations. At faster speeds we needed to make the microcontroller wait for the LCD driver chip to complete its operations, thus solving our problem.

Input Capture Problems:

In using the input capture we faced a variety of problems ranging from unexpected resets to unchanging values. When initially using the input capture interrupt, we found that on the rising edge, the input capture would unexpectedly reset the microcontroller. We later found that this problem was caused by enabling the input capture interrupt while not providing an interrupt subroutine to accompany the enabled interrupt.

Another feature we had difficulty with was the automatic clocking feature of the interrupt. The input capture feature of our microcontroller automatically logs how long a pulse is detected for. We initially had difficulty determining the value of this variable, as it appeared as if it were unchanging. We later discovered that this was a simple coding error on our part, where an improper variable declaration caused the value to be unalterable.

3. Budget and Timeline Comparison

3.1 Budget Comparison

The following table is a rough comparison between the estimated budget prior to the project and the actual amount we spent at the end. Major differences between the two are our expenditure on the IC's/parts, printed circuit boards and the evaluation kits. The cost for parts well exceeded our estimation because we ended up ordering several spare micro-controllers. We also did not anticipate the extra surface-mount parts required for the CC1000, which often have to be ordered in bulk. The amounts of component also significantly increase as we decided not to obtain a RF evaluation kit. Cost for PCB is also lowered because of the discovery of a company that can get them cheaply made.

Table 1: Cost breakdown comparison

Item Description	Estimated Cost	Actual Cost
Printed Circuit Board	$ 150	$ 95
RF Evaluation Kit	$ 300	None
IC's Passive Components	$ 75	$ 203
RF transceiver module	$ 50	$ 54
LCD display	$ 40	$ 34
Case	None	$ 20
Shipping	$ 25	$ 74
Tax	N/A	$ 20
Total	**$ 640**	**$500**

3.2 Timeline Comparison

In our original project proposal we provided an outline of the basic timeline for the project, with goals and accomplishments we hoped to achieve by certain days. As was expected, there were certain difficulties and unforeseen problems associated with our project. In Figure 13: Original Proposed Timeline is our original projected timeline. In Figure 14: Actual Timeline is our actual timeline.

ID	Task Name	Start	Finish
1	Research RFID evaluation kits	1/1/2005	1/28/2005
2	Research RFID tags	1/3/2005	1/28/2005
3	Research RFID transponder	1/5/2005	1/28/2005
4	Research power supply	1/7/2005	1/28/2005
5	Research antenna	1/9/2005	1/28/2005
6	Research microcontroller	1/11/2005	1/28/2005
7	Research LCD display	1/13/2005	1/28/2005
8	Research usability (question)	1/29/2005	2/21/2005
9	Research usability (survey)	1/29/2005	2/21/2005
10	Funding	1/21/2005	2/7/2005
11	Proposal	1/10/2005	1/26/2005
12	Functional specification	1/24/2005	2/7/2005
13	Design specification	2/3/2005	2/17/2005
14	Ordering time	2/1/2005	2/14/2005
15	Development	2/14/2005	3/31/2005
16	Assembly/Programming	2/14/2005	3/31/2005
17	Integration transponder&microcontroller	2/25/2005	3/31/2005
18	Integration LCD/LED	2/25/2005	3/31/2005
19	Integration buttons	2/25/2005	3/31/2005
20	Integration interface	3/25/2005	3/31/2005
21	PCB design	2/14/2005	2/28/2005
22	Testing/Debug/Resign	3/10/2005	4/1/2005
23	Documentation	1/10/2005	4/1/2005
24	Post Mortem	4/1/2005	4/8/2005

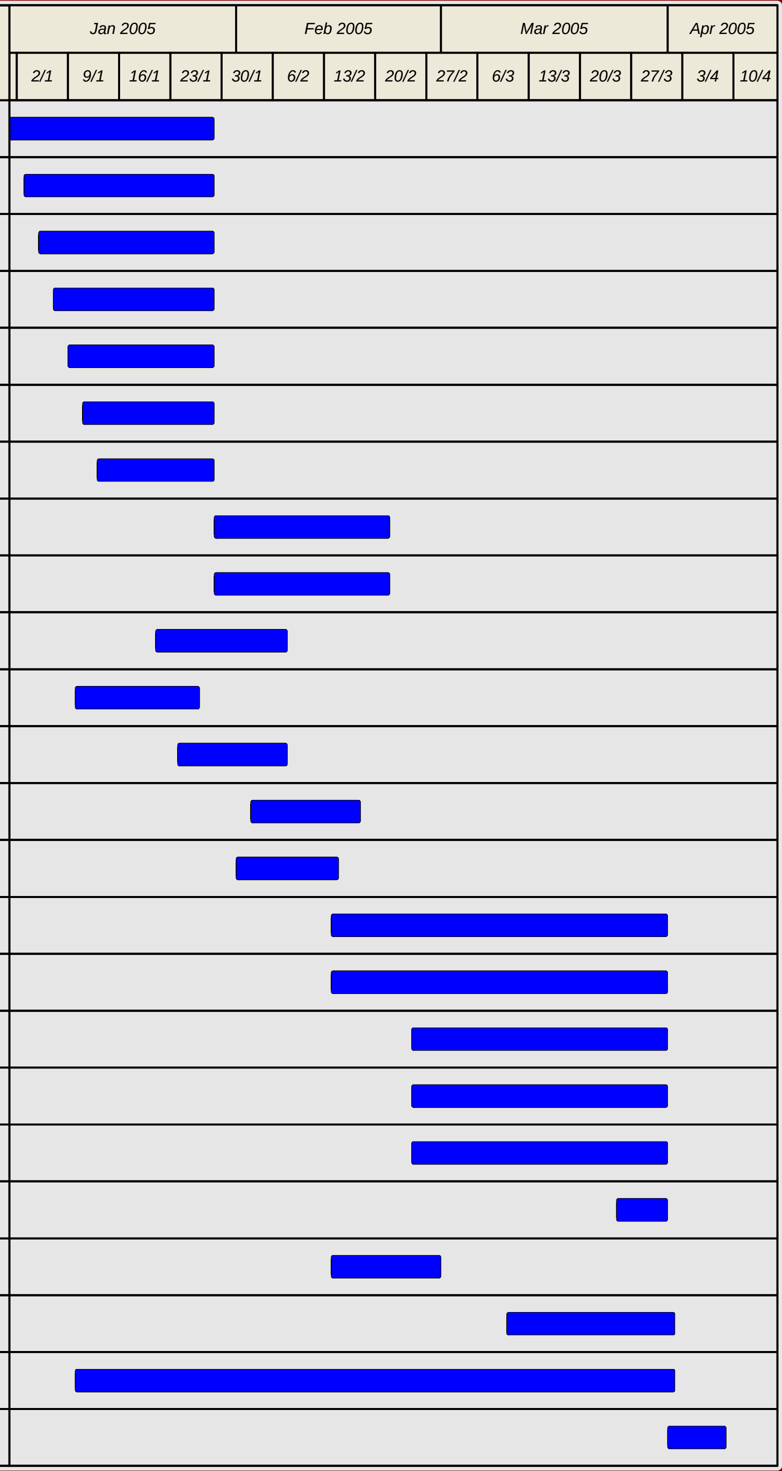

Figure 13: Original Proposed Timeline

ID	Task Name	Start	Finish
1	Research RF evaluation kits	1/1/2005	2/2/2005
2	Research RF TX/RX Pair	2/2/2005	2/7/2005
3	Research RFID transponder	1/5/2005	2/7/2005
4	Research power supply	1/7/2005	2/7/2005
5	Research antenna	1/9/2005	2/7/2005
6	Research microcontroller	1/11/2005	2/7/2005
7	Research LCD display	1/13/2005	1/30/2005
8	Research usability (question)	2/7/2005	3/2/2005
9	Research usability (survey)	2/7/2005	3/2/2005
10	Funding	1/21/2005	2/7/2005
11	Proposal	1/10/2005	1/26/2005
12	Functional specification	1/30/2005	2/13/2005
13	Design specification	2/8/2005	2/22/2005
14	Ordering time	2/4/2005	4/12/2005
15	Development	2/14/2005	3/13/2005
16	Assembly/Programming	2/14/2005	4/13/2005
17	Integration transponder&microcontroller	2/25/2005	4/12/2005
18	Integration LCD/LED	2/25/2005	3/31/2005
19	Integration buttons	2/25/2005	3/31/2005
20	Integration interface	3/25/2005	3/31/2005
21	PCB design	4/8/2005	4/10/2005
22	Testing/Debug/Resign	4/2/2005	5/6/2005
23	Documentation	1/10/2005	4/1/2005
24	Post Mortem	4/14/2005	4/21/2005

Figure 14: Actual Timeline

We can see that obviously there is a significant contrast between the two timelines. Some of the major differences in this timeline were caused primarily by misjudgments on our part for the amount of time required to deal with certain problems. One of our worst judgment calls was in terms of the fabrication of the PCB boards. We initially believed that fabricating the boards ourselves would prove to be relatively hassle-free; however, we discovered near the latter parts of the timeline that getting boards fabricated requires a significantly amount of work. Additionally, our original estimate surmised that we would have our parts fully ordered early on in the project. As the actual timeline shows, we were ordering parts right up to the presentation date of April 14th. These parts were unexpected components required for generating a prototype which functioned off a battery supply, as well as some additional parts for the CC1000. Another good majority of our setbacks stemmed from the cyclical dependencies of our system. In order to properly test the system, we required PCB boards to be made, but in order to make boards we required a relatively functional prototype which involved proper testing. Because of these problems we ran into several major issues and were forced to compromise where we could, causing several major time delays.

4. What We Would Do Differently

The one most prominent issue which we would do differently if we were to start the project from scratch would be to fabricate boards for the CC1000 at the beginning of the project as opposed to at the end. Near the end of our project it became evidently clear that if we had fabricated the transmitter and receiver boards at the very beginning of the project we would have had a much easier time in designing and developing both hardware and software.

In terms of other hardware and software choices, we believe that the majority of our choices were relatively viable and produced relatively useable results. However, in terms of timeline, we believe that we should have put more work into the initial stages of the project in order to avoid the large wave of work near the end.

5. Future Improvements

5.1 Tag

There are many improvements which can be made to our project. For the Tag (transmitter) units, we could make them smaller by incorporating the CC1000 circuitry together with the ATtiny2313 microcontroller. We would also change the firmware in the microcontroller so that it would only send an identifying code, rather than continuously transmitting a square wave. Each of the Tag units would then transmit on the same frequency at roughly a two second interval. While not transmitting, the transmitter chip would be put into low power mode, and the microcontroller would be set to sleep. The microcontroller would then wakeup after its watchdog timer timed out, and send another short transmission.

Initially we built the Tag unit in two parts--the microcontroller part and the CC1000 transmitter, and found that we were unable to flash program the ATtiny2313 via the ISP (In System Programming) header when the CC1000 transmitter board was connected to the microcontroller.

For our next board revision we would either put programming jumpers on the pins which connected to both the programming header and the tansmitter or change the pin configuration of the microcontroller.

Additionally, since there is brown-out circuitry built into the ATtiny2313, if a brown-out (low battery) condition was sensed, it could also transmit this information to the Receiver unit. The Receiver unit would then inform the user that the Tag needs a new battery.

The Tag unit would also use a small compact antenna with uniform gain. The Tag unit would then be less affected by environmental and body effects, simplifying the distance calibration in the Receiver unit. Finally, by integrating the antenna with the transmitter PCB we should be able to make the Tag units quite small.

5.2 Receiver

For the Receiver unit, we would use the Chipcon CC1000 transceiver, rather than the Linx receiver since it is physically a lot smaller.

The circuitry also requires some small modifications. Since we experienced problems with running the LCD controller at 5.0V and the rest of the circuitry at 3.0V we would either use a 3.6V battery or use level translator chips between the microcontroller and the LCD unit.

For the antenna, we would either use the trapezoidal patch antenna, or possibly a simple PCB stub antenna with a small inductor depending which fit the case better.

For the case we would need to do some more research to find an ergonomic case which is small enough, and which would allow the user to easily replace the battery without having to unscrew the unit.

6. Personal Reflections

Carson Hammoser
No matter how well you try to plan a project, there seems to be an unwritten rule that 'Delays are Inevitable'. At the start of the project I thought we would be able to finish on or very close to schedule, but that was not the case.

I learned how difficult it is to stay on schedule when unexpected delays arise. However, I think the most important thing I learned (the hard way) is that 'Time Is Money'. Unless of course you are a student, and then lack of money is a LOT more time. Had we not tried to save money at the beginning of the project by manufacturing our own PCBs for the transmitters we easily would have finished on time, but probably a bit over budget. However, if this situation arose in a company, the extra expense for hardware would easily have been saved by requiring less manhours to troubleshoot and debug hardware problems.

Peter Lin
From this project I learned and experienced the engineering process of design, prototype and testing. I realize that no project is too easy for this course because even tasks that seem trivial such as programming a micro-controller often turn to be a handful from lack of experience and planning. Due to the nature of our project, I also learn a great deal in terms of wireless transmission quality.

Albert Uang
Through this experience I gained a great deal of experience in programming with others and the problems involved in combining code fragments and integrating components. I also discovered that often the simplest mistakes absorb the most amount of time, and that meticulous methodology is the best basis for avoiding these mistakes. Through this experience I gained a great deal of appreciation for sample code, the methods of debugging and group dynamics.

Samuel Wong:
Through this project I gained a great deal of both social and practical experience. In terms of group dynamics, I have gained a true taste of working with others for an extended period of time, including the ups and downs associated with project problems.

7. Conclusion:

People lose items everyday. Lost and founds are everywhere you go; each is filled with items that people never reclaim and often have to replace at significant cost. As the market penetration of cell phones, pocket PCs, and other small electronic equipment increases, the number of items which are lost also increases. The SafeGuard provides a solution for the forgetful and absentminded, preventing the loss of their items. The goal of Mnemosyne Technologies was to generate a product with all the preceding specifications by the end of April. We successfully achieved this goal to a reasonable extent considering the tight timeline and limited budget. Overall, the project members found this to be a useful and educational experience.

References

[1] Smith, Kent (RFM). “Antennas for Low Power Applications”. Feb 8, 2005.
www.rfm.com/corp/appdata/antenna.pdf

[2] “ATTiny2313 Datasheet” March 2005.
www.atmel.com

[3] “2032 Datasheet”. March 2005
http://rocky.digikey.com/WebLib/Panasonic/Web%20data/CR2032,%20CR2320.pdf

[4] “Plastic Enclosure”. March 2005
http://www.hammondmfg.com/scpg.htm#plastic

[5] “HP Series-II Receiver Module Design Guide”. March 2005.
http://rocky.digikey.com/WebLib/Linx%20Technologies/Web%20Data/RXM-900-HP-II.pdf

[6] “CR2477 Manganese Dioxide Lithium Coin Batteries”. March 2005
http://rocky.digikey.com/WebLib/Panasonic/Web%20data/CR2477,%20CR3032.pdf

[7] “Plastic Enclosure”. March 2005
http://www.hammondmfg.com/scpg.htm#plastic