HIKARI SYSTEMS CORP.

Sept 7, 2001

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

Re: ENSC 340 Indoor Positioning System Process Report

Dear Dr. Rawicz,

Based on our experiences, we have developed the attached *Process Report for the Indoor Positioning System*. We describe how our project has progressed, the current technical state, the issues we faced, and our future plans.

Our team has enjoyed taking EnSc 340 in the past months. Thank you again for offering an extra session of the course to accommodate our needs. We hope the project will continue to develop, and your input will always be welcomed. For questions and comments, please contact me at gwfung@sfu.ca.

Sincerely,

Gregory Fung
Project Leader
Hikari Systems Corporation

Enclosure: *Process Report for the Indoor Positioning System*

HIKARI SYSTEMS CORP.

Process Report for the Indoor Positioning System

Team Members:	Gregory Fung Ivan Ho Ming Jiang Aguilar Lam Sherman Yin
Submitted to:	Dr. Andrew Rawicz, Steve Whitmore, School of Engineering Science Simon Fraser University

Sept 7, 2001
Version: 1.0

Executive Summary

Between May and September of 2001, our team of students who make up Hikari Systems have taken a conceptual idea to a workable system prototype, demonstrating the feasibility of the Indoor Positioning System (IPS) concept. We have achieved roughly 75% of the functionality we set out for, albeit over a longer time frame due to management inexperience and lack of time from team members.

We encountered many issues during development, mostly due to the unfamiliar PalmOS platform and development tools, as well as limitations of the OS itself. We found solutions and workarounds for nearly every issue, after some effort and perseverance.

Each of the team members had a positive experience, gaining technical experience in their specialisation while building their team work and organisational skills. We were almost exactly on budget, pro-rated against the functionality we have achieved. In the future, we plan on continuing to develop the project based on further marketing studies, seeking a field trial late in 2002.

Table of Contents

Glossary

AAA	Common consumer alkaline battery size, also know as UM-4
CE	Certification mark from European Commission for consumer and industrial products
CSA	International non-profit organisation developing standards for product certification
FCC	Federal Communications Commission of the USA, who regulates use electromagnetic spectrum and noise emissions
Graffiti	Handwriting style recognisable by the PalmOS software
IR	Infra-red radiation, used short-rage for line-of-sight digital communication
PalmOS	Operation system for handheld computers by Palm Inc.
PDA	Personal digital assistant, also know as handheld computers
Springboard	Springboard expansion slot standard by Handspring Inc.
Stylus	Plastic-tipped stick used to tap and write on the screen of pen-based handheld computers
UL	Underwriters Laboratories, an independent non-profit product safety testing and certification organisation

1 Introduction

Looking back at this point at the end of the project, our Hikari Systems team has come a long way. Our idea has materialised in a working prototype. Four of us have become familiar with visual programming with PalmOS. We have hand-soldered more than ever before. Our friendships have grown. All this has happened over four months of our lifes known as EnSc 340.

In this conclusion of our journey, we revisit our earlier thoughts expressed in the proposal and specification documents, to see how we did and discover lessons to be learned. First we begin with technical issues, then on to managing issues and obstacles, then our visions for the project, then our personal reflections on this trying process.

We hope you enjoy this review and learn from our triumphs and pitfalls. Then again, some issues are merely first-hand accounts of common problems found in life, in which case you should already know them and will find our experiences amusing; that's ok as well.

2 System Design and Deviations

2.1 System Operation

We implemented our proposed system design without any unexpected problems. It performed mostly as expected, requiring the user to direct the handheld's IR port towards the beacon. Otherwise, the software performed flawlessly with 90% of the proposed functionality implemented and working well. The major change was halting development of the hardware module, which contains the orientation device and audio output circuitry due to time constraints. Figure 1 shows our system layout.

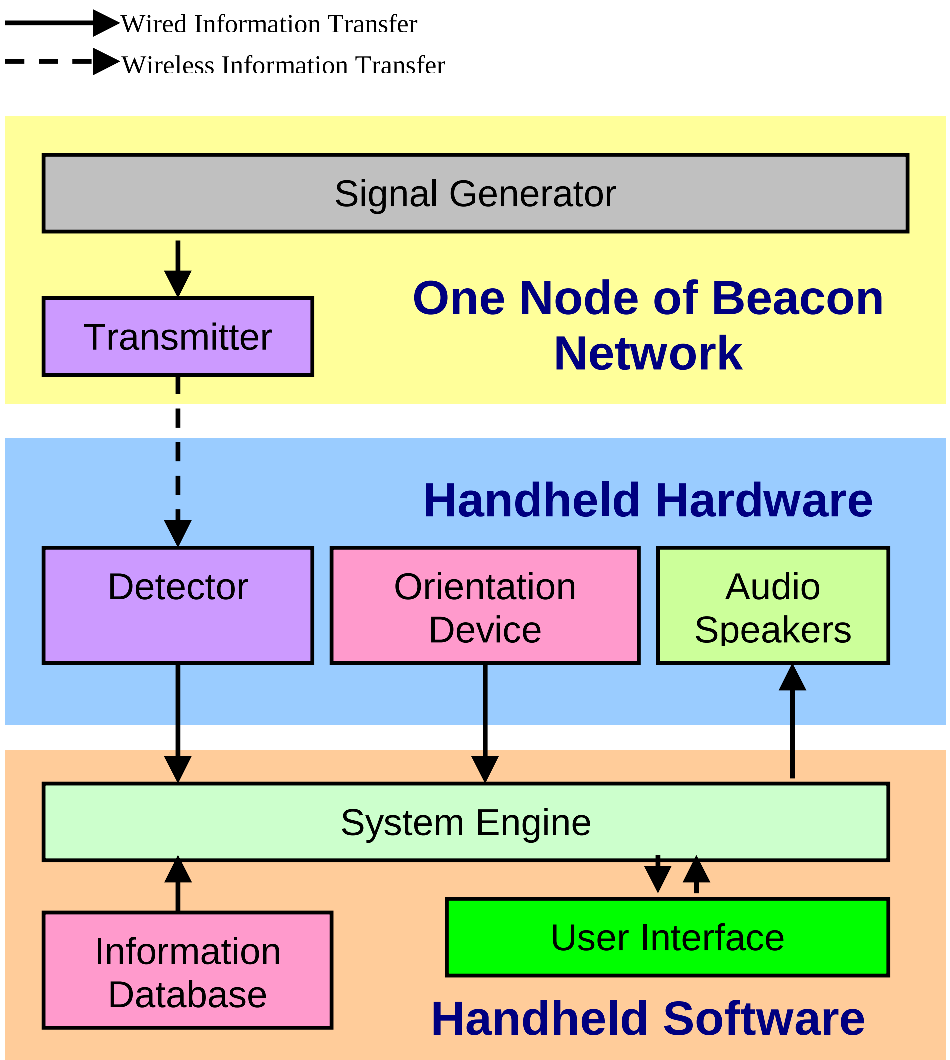

Figure 1. System Block Diagram

2.2 Beacon Network and Transmission

2.2.1 Transmission Format

Shown below in Figure 2 and Figure 3 are the IrDA byte transmission and packet layout that we proposed to use in our design and functional specifications. Note that the 16 bit FCS field is the CRC error checking bits. The packet layout follows the IrDA standard for Palm devices.

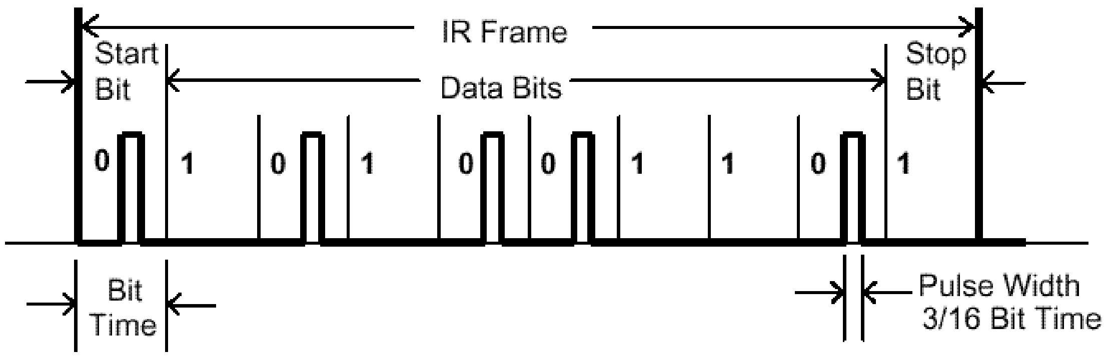

Figure 2. IrDA byte transmission timing example

XBOF	BOF	ADR	Sender	Receiver	Data	FCS	EOF

Figure 3. Packet/Frame layout of transmission signal

The reason we chose to follow the packet layout in Figure 3 is because we thought we would not be able to bypass the receiver protocol on the palm IR receiver. During the production of our prototype, we were able to devise a way to bypass the IRDA standards so that we could simplify our packet content. Now the packet content only contains the data field, which is only one byte in size compared to the original 24 byte layout. Rather than use the CRC form of error checking we used parity error checking which is much simpler and just as effective. CRC error checking involves a much more complicated algorithm as well as an extra 16 bit output while parity error checking is extremely simple and requires only 1 extra bit.

We still followed the IrDA physical standard shown in Figure 2. To provide extra error checking from the receiver side, we only recognise the packet to be valid if we receive four of the same packet in a row without error.

2.2.2 Transmitter

The final transmitters were similar from our initial design, with only component substitutions. We decided to use vector boards rather than using printed circuit board, to save the time and effort of learning the lithography and etching processes. Another slight change we made is the power supply. We decided on using a regulator with a 9V battery rather than multiple AAA batteries. This simplified and lowered the cost of packaging, without requiring more expensive switching regulators.

The final change made a large impact on performance: using a enhancement-mode MOSFET instead of a small-signal BJT we originally selected allowed us to switch much larger current pulses through the IR diodes efficiently. Previously, the 2N3904 transistors used did not yield enough gain, limiting our transmission power output. As this flaw was discovered very late in the process, we could not obtain enough MOSFETs to build all 17 transmitters with them. Using MOSFETS generated much more switching noise too compared to BJTs, necessitating more power supply decoupling.

An issue we had overlooked was the peak wavelength specification of the IrDA physical standard at 850 – 900nm. We discovered this specification just before demonstrating, when 940nm-peak IR emitters were already installed (having a 40nm bandwidth). This means that our transmitter was wasting power, broadcasting most of the energy at the wrong wavelength for the IrDA complient Palm devices. This explains why our range was so poor for such high currents (300mA –1A pulses).

2.3 Handheld Hardware

2.3.1 Add-on Hardware Module

As laid out in the design spec, the hardware module is an optional add-on for IPS users that gave the unit the ability to determine its heading. It also has a sound output module to play short instructions at allow visually impaired users to navigate themselves once the unit is set. However, because of time concerns, we decided to put this part on hold to complete development of the core system first.

Before development stopped, the basic parts of the hardware module prototype were ready for testing. Brief tests showed slight problems with lead inductance because of the high speed the module operates at. Also, the flash memory chip could not be identified for programming by the PalmOS debugger.

Driver software for the module was also ready for in-circuit tests. Further development needed the proper hardware interactions that the emulator could not provide. Since we couldn’t install this software on the flash memory, the hardware held up the progress on this end. After these problems are resolved, we expect to fully test the module software and take readings from the compass circuitry.

2.4 Handheld Software

2.4.1 System Engine

The system engine provides other parts of the software a set of functions that can be called to perform key tasks. It has an event driven structure that receives input from the IR port as well as the Springboard module. Due to time constraints, development of the Springboard module and its support was halted.

The system engine is able to perform the following tasks:

- Determine current node
- Shortest route look-up
- Location information retrieval
- Map locations on site-map with location information

2.4.2 Information Database

The final implementation of the database matches our design with only a few exceptions. To our first design, we made no significant changes, but only added a few tables to hold information that we have omitted in our first design. The successful database design was the result of elementary database design knowledge acquired from previous computer science courses.

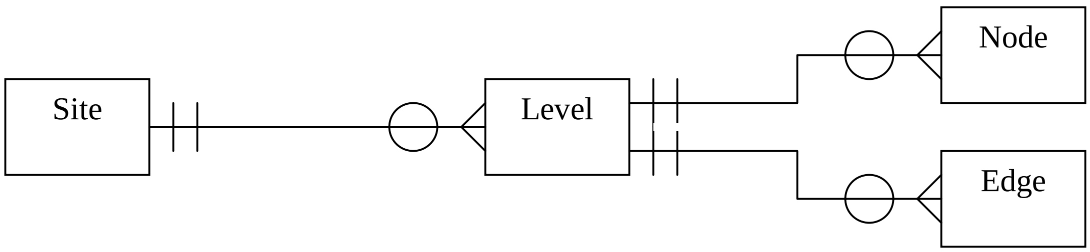

Figure 4: Database structure

2.4.3 User Interface Design

The implementation of the UI is more involved than expected, mostly because of unfamiliarity of the PalmOS API. However, we are able to meet most of the requirements listed in the Functional Specification. There are a few changes from the original plan, due to design improvements and some technical difficulties. Also, since the hardware module is unavailable, we did not implement the compass or any UI elements related to that module. The UI of the final version of IPS has nevertheless the look and feel of what the design specification describes, complete with a splash screen, menus and help screens.

2.4.4 Application File Format

Although we aimed to cleanly separate the main IPS program and any site data into separate files, we managed to only partially achieve that goal. The current program has many site-related size limitations embedded within it, such as the maximum number of beacons, maximum number of locations, and length of text strings. These limitations came unexpectedly as a consequence of the PalmOS database support routines and the C programming language. As we refine the software further, we should be able to remove most of these limitations, making our system engine truly site-independent.

3 Issues and Solutions

3.1 Technical

3.1.1 PalmOS Learning Curve

The Palm OS programming environment proved to be one of the most difficult challenges and caused the most problems in developing the IPS system. Problems with development tools such as emulators and the trial CodeWarrior Lite IDE compounded poor documentation on earlier software development kits (SDK), making programming and debugging quite troublesome. Other problems encountered included unfamiliar Palm OS-specific library functions, dealing with version incompatibilities, unfamiliar database support structure, and puzzling use of strings within the C programming language.

Many of the problems we encountered while programming in the Palm OS involved long tedious hours of debugging. Poor documentation made debugging especially difficult because system functions were not clearly defined and explained to the developer. Often referring system header files and C-language manuals, we were left to induced probable causes of error. Trial and error testing was further handicapped because of periodic communication problems between the emulator and the IDE.

3.1.2 PalmOS Graphical Support

When we were implementing the graphical UI part of the IPS software, we discovered that version 3.5 of the PalmOS has a lot of new functions that will greatly shorten the development time of the software. These functions are not available in the PalmOS 3.1, which was originally in the list of supported PalmOS versions. We would be able to complete the project without the 3.5 new functions, but that would require a much longer development time. We have decided to give up compatibility for the shortened development period due to our tight schedule.

3.1.3 PalmOS IR Interface

There are a number of methods to access the data received from the IR receiver port of a PalmOS supported handheld device. The method that is mostly recommended is through the use of the IR library functions provided by PalmOS 3.0 or above, which enable the user to create an IR connection through the IrDA Link Access Protocol (IrLAP) (http://www.irda.org).

Initially we had intended to employ the protocol as the format of transmission for the IPS beacon signals, but we discovered several issues with the protocol, as well as the library, that forced us to seek another way to access the data received from the IR port of the handheld:

- First of all, the IrDA protocol asks that each data packet must be sent in a 24-byte format including information such as sender/receiver address, data flags, and wearisome error checking data. For the purpose of the IPS prototype, each beacon only requires to send one byte of data as the address of the beacon. Compiling one byte of actual data into a 24-byte packet appears to be wasteful.
- Second, the IR library is a shared library in the PalmOS, which means that the functions can be called by multiple applications during the run time of the device. So if IR signals were received by the device, several applications might be triggered to perform tasks in response to that signal received, which could cause the device to become unstable.
- Finally, each IrLAP packet must include a set of error checking data that has to be pre-calculated before transmission. The calculation method is rather tedious and therefore difficult for the signal beacon controller to perform.

The obvious solution to the above problems was to find a way to access the raw data received at the IR port, so to bypass the IrDA protocol for a simpler data transmission format. Upon in-depth research, we were able to find another library of functions from the PalmOS that would let the user access the raw IR data in bytes. Using the new library, the IPS beacon may broadcast simple data packets to the handheld, and the handheld may then efficiently recognise the data with only rudimentary error checking methods. The new library is not a shared library, which means that one application takes ownership of the library, thus the IR port, so other applications on the device may not access the port at the same time. The physical specifications of the IR signal continue to follow that of IrDA standards, which was part of the design specification of the IPS prototype.

3.1.4 Software Packaging

In order to support multiple sites seamlessly in software, the application core should be a separate entity from the site data. In PalmOS, this is manifested using prc application executable and pdb application database files. Beyond this, however, the application is currently not flexible enough to be completely adaptable for all applications. Many sizing constants are currently being compiled into the executable; in order to maintain reasonable memory usage, these sizes cannot truly accommodate many sites. The best solution is for the executable to allocate only as much memory as needed for each site, a feature to be added later.

3.1.5 PIC Microcontroller Intricacies

The most difficult challenge in designing the transmitters was using the PIC microcontrollers. Due to the high number of transmitters we must use, cost of the transmitters must be kept to a minimum. Using economical PICs lead to problems with limited memory/resources, a limited instruction set, and also a lower number of I/O pins.

To solve the problem with the limited number of I/O pins, we used serial data transfer between the DIP switch and the PIC by using a PISO chip. This will allow 8 bit data transfers using only 3 I/O pins.

Limited instruction set and resources made very easy tasks into very long and complicated code. The PIC could only store 500 lines of code and allow a maximum of 25 eight-bit registers so efficiency was also an issue. To accommodate to these limitations, we were able to simplify the packet content by bypassing the IRDA protocol on the receiver side. In turn, we would be able to avoid CRC error checking which is extremely difficult and long to implement using the PIC. Instead, we used parity error checking which is much simpler and more efficient in terms of code size and resource usage.

3.2 Scheduling

Due to the unexpectedly high workload of documentation, unpredictable work schedules of team members due to coursework, and inexperienced leadership failing to enforce milestones, our original goal of having basic functionality before August 2001 was not met.

On average, between the months of May to early July, team members contributed only 10-15 hours per week, mostly on research, design and documentation. This delayed the bulk of technical implementation into late July, lasting until the end of August around final exams members had.

3.2.1 Decision to simplify project

Roughly halfway in our development work, we found that development of the software core was much slower than anticipated. The work to be done was also larger than we anticipated. Because of this and the fact that we were much behind schedule overall, we decided to suspend development of the add-on hardware module, which is not absolutely required for our project.

3.3 Budget

The following table is a breakdown of the expenditures in building the working prototype of the IPS system. The second column contains the estimated cost for each of the components based on the preliminary system concepts. This was first introduced in the proposal document. It is clear that the final expenditures turned out to be well below our estimated cost. Obtaining free parts such as samples and from other sources, as well as incomplete development of some parts of the system is the main reason for the discrepancy.

Table 1. Budget for System Components

Part	Estimated Cost	Actual Expense	Comment
Transmitters	$10 ea. x 20 = $200	(17 x $9.50) + $25 (misc. experimental parts and supplies = $186.50	Packaging and common components obtained for free through various sources.
Software Development Kit for PalmOS	Freeware	NIL	Free development software from Handspring Palm
Electronic Compass	$30	NIL	Received free as sample.
Micro-controller	$10	NIL	None required
Wires	$5	NIL	From lab and other sources.
Speakers	$10	$15	
Slot Plastic for Handheld	$100	N/A	H/W module development halted
PCB Printing	$200	N/A	H/W module development halted
Misc.	$50	$25(dev. PIC) + $60(other dev. parts) + $85 (books) + $50 (hw module dev) = $220	Developmental parts that are not part of the final product
Contingency fund	$50	N/A	Part of miscellaneous cost.
Total	**$655**	**$421.5**	

3.4 Team Dynamics

3.4.1 Communicating Ideas and Requirements

Beyond the basic documentation required by EnSc 305, our team only used the most basic documentation in our work. This caused some problems when the technical requirements were not well understood by some members. We had several occasions where members worked on one part based on their vision, only to realise later that it was not what was needed. Even small issues such as the semantics of a certain term being differently understood caused confusion. We could have saved some time and frustration by having the responsible person define the exact technical requirements from the onset. Again, without management to define such deliverables, some members did not volunteer such documentation.

3.4.2 Code and Document Revision Control

Our revision control could have been improved to save time and effort. Since the documents are divided among five group members, one person has to merge and edit other members' work to improve the flow of the document. The project leader was always the one in charge of this last step, which became a tedious task for him. A better way is to rotate this role among the members so that everyone gets a chance to learn the process.

Since everyone worked at different pace and encountered different problems, there are situations where a member had to wait for the work of another member before he can continue. This problem can be avoided if we more clearly schedule our work and document individual function requirements.

3.4.3 Working in different sites over internet

Our development was basically divided into software application, with the use of free-ware libraries and development tools provided by Palm Inc.; and hardware modules for IPS beacons and handheld hardware expansion. Since the software application could be developed on any PC, and that testing and debugging could be performed on an emulated version of the PalmOS on the PC, face-to-face meeting between members of the software development was not a necessity except for coordination and design processes. Each member could simply implement and test his own section of the application at home. Communication between members was done online through instant messaging services and e-mails, supplemented by telephone conversations.

Although this method of telecommuting saves vast amount travel time for each member, spreading important information to all member in a short period of time was rather difficult because each member had different working hours on the project. The problem became obvious during software integration when revisions and modifications were often made but the involving members could not work together at the same time. Members were forced to adjust their times so the integration process could go smoothly. We were also forced to rely heavily on network connections and the Internet to pass information amongst members of the project, which were unreliable at times.

One other consequence of infrequent meetings was more difficulty managing the project. Because group meetings were few, management communications relied on the passive nature of emails to request progress updates and assign tasks. Cellure phones and pagers helped for quick responses; but e-mails were sometimes not read because of volume of messages required for the project and in each members other endeavours. This contributed to aforementioned problems with communicating requirements.

4 Future Plans

4.1 Marketing

Over the next year, we will continue to seek support from interested parties to develop our system. The ideal goal would be to find a customer site, where a full field trial can take place. Their patronage or support will be sufficient to fund further development.

To achieve this goal, we will need to invest substantial energy in making ourselves known. Attending local trade shows is a must, such as the BC ASI exchange. Other avenues we can explore are mailings to building managers or advertisements in trade magazines. After the initial customer, our product should be refined and adaptable enough to target many different markets.

4.2 Development

There are unlimited applications for the IPS and the prototype was only a simple version that demonstrated the working of the concept. Many enhancements can be made on the prototype to customise the IPS for different applications. One of our main focus is to tailor the IPS for the visually impaired with voice recognition of commands and real-time audio guidance. Extra hardware modules may be developed to incorporate warnings on obstacles using proximity sensors.

One of the most immediate improvements of the IPS prototype is to add real-time directional guidance on the device for the user. The hardware module described in the design specifications is under development and we plan to fulfil the specifications even after our prototype demonstration.

On the software part of the development, we plan to improve by simplifying the process of site/beacon arrangement with the use of supplementary programs to load maps easily onto the handheld and add/delete beacon nodes in the database during run-time of the application. This improvement provides tools for customers to change the information on the database handily and simplifies the expansion process.

Another important enhancement is to allow a two-way communication between handhelds and beacons. This enhancement requires that the beacons can receive signals from the handheld, which is not a requirement in the design specification of the prototype. With two-way, communication, users may send information to the beacons. For example, in a shopping mall, store-owners may update their corresponding beacons to contain new deals and specials as advertisements to the mall-goers. Moreover, if the beacons were networked to a base station, information from one beacon can then be updated to other beacons.

Finally, having Radio Frequency (RF) as the medium of transmission is more appealing to the users because of the transparency of pointing the handheld at the beacons. With the insertion of Bluetooth technology, we can implement beacon-beacon, beacon-handheld, and handheld-handheld communications to enhance the quality of information transfer.

5 Team Member Impressions

5.1 Gregory Fung

Even though the final result is slightly short of what we had hoped for, I feel we have accomplished much in the last months, turning the idea into a working prototype. If needed, we can release the software to a customer with only minor changes. I am pleased with the effort the team gave, matching our expectations from the beginning of the term. We also learned much in the process, while maintaining and building our friendships.

I found that management can be divided into two levels: organising your resources and facilitating effective work between them; and understanding your team at a personal level to allow them to perform their best. One important task of the first is proactively organising communication by specifying documentation, preventing potential miscommunications and wasted effort.

In the past months, while I felt I worked with the members reasonably well, our team ran into several situations where communication breakdowns resulted in incorrect work and lost time. Fortunately they were correctable without major effort, but the need for better documentation and communication was clearly there. Hopefully, in future opportunities to lead, I will anticipate more of these problems and co-ordinate the team better.

5.2 Ivan Ho

Like any other group projects, I had great expectations on the ultimate product, and what we would achieve by the end of a four-month project. To me, the most interesting part of the project was during the implementation, and at the end of the implementation where the product worked to the way I wanted. However, by the end of the first month, we were occupied with various documentation and design layouts, which delayed our implementation process. Although I was a bit disappointed about the limited time frame in which we could implement our design, I was able to understand the true process of product development, in which a lot of preparation and documentation have to be in place so we can follow the specifications carefully. By the end of the project, I felt that the prototype we implemented was better than what I had anticipated.

I will definitely continue with the project when funding becomes available. I believe that IPS is a marketable idea, which is why we selected this project at the beginning.

5.3 Ming Jiang

After working on this project for four and a half months, I feel things went relatively smoothly for myself and my fellow members. Everyone gave a good effort and helped each other out. The resulting product and documentation is reasonably close to our original intention of what it should be.

I believe the organisation of the group was reasonably flexible to allow each of us to work at our own paces while keeping on tract on our final objectives. Planning was relatively organised and last minute cramming was kept to a minimum.

5.4 Aguilar Lam

After completing on our IPS prototype, I can honestly say that I am proud of what we have accomplished. When we were first designing our product, many people told me that this project was too difficult and complicated to be completed in one semester. Also, all my group members were taking full course loads and I was on a co-op work term, which limited the time we had to spend on this course. I'm glad to say we were able to prove them wrong by completing a functioning and well-packaged prototype.

I was very impressed with the little time it took to do the integration of our parts. Although there were times when miscommunication lead to some wasted time, I believe that it took a lot of good planning and organisation to be able to complete the integration of our separate parts so smoothly.

Overall, this project has been a good learning experience in terms of improving on my technical skills as well as my group organisation and communication skills.

5.5 Sherman Yin

Before taking this course, I thought ENSC340 would be similar to CMPT275 in the sense that the only thing you remember from the course is documentation. Luckily that was not the case. I think the main difference is that in ENSC340, we have the chance to brainstorm and come up with our very own project idea instead of following predefined specifications and requirements. To me, that made all the differences. We got to exercise our imagination, explore our skills, and plan on marketing the idea. That is what makes engineering interesting.

Overall I am quite happy about our achievements in this course. Although we did not have time to implement all the functions we wanted to, I am quite satisfy with what we have done already. Our product is functional and works as we expected without any serious defects. We did not get into any big problems during the development process, and integration was surprisingly smooth.

Near the beginning of implementation, some team members did not completely understand the specific requirements of their parts of the project. The result is that work had to be redone and time wasted because of miscommunication. If we had composed more clear and detailed documents, this problem can probably be avoided. Fortunately, our team saw the problem and corrected it right away. Our team worked more efficiently together since.

Although we gave up our summer holidays to finish the project, I am happy it was worth the time to complete the project. Some team members are looking into bringing our project idea to the next stage, perhaps turning this school project into a marketable product.

6 Conclusion

In the past months we have learned much, regarding teamwork as well as technical issues. Achieving roughly 75% of the functionality we hoped for, we consider the implementation portion of our project a success. What remains appears very achievable in the continuing effort to bring the product to market, which may result in a large field trial installation in 2002.

Just as important are the teamwork and interpersonal skills we gained. We recognise the importance of documentation, communication, thorough consultation and cross checking to a smooth, predictable development process. While we had few external dealings, we are keenly aware of our professionalism during these dealings.

Our thanks go to the Engineering Science Student Endowment Fund for funding our development efforts. Also, our thanks to PNI Corp for supplying us with sample parts of their unique electronic compass technology. Thanks to Seeho Tsang, Wai-Kee Lee, Patrick Leung, members of the Palm development forum and Handspring technical support for their guidance and suggestions during our development.

Works Cited

Handspring. 2001. *Home page*. May 29, 2001. http://www.handspring.com

---. 2001b. *Springboard Development Guide, v1.12*. July 4, 2001. http://www.visor.com/developers/Devkit2/Springboard_Development_Guide_112.pdf

IrDA. 1998. *IrDA Serial Infrared Physical Layer Specification, v1.3*. June 7, 2001. http://www.irda.org/standards/pubs/IrData.zip

PNI Corp. 2001. *Home Page.* July 2, 2001. http://www.pnicorp.com/.