School of Engineering Science • Burnaby, BC • V5A 1S6
http://www.ensc.sfu.ca • iv-ensc@sfu.ca

December 21, 2002

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

Re: *ENSC 340 Process Report for an Intelligent Monitor Power Saving Security System*

Dear Dr. Rawicz:

The attached document, *Process Report for an Intelligent Monitor Power Saving Security System*, details our ENSC 340 project post-mortem. Our mission to develop an intelligent sensor unit integrated with computers to improve power consumption, end user's experience and security was achieved. Our product detects the presence of computer users and activates power management and security protection accordingly.

This process report provides details of our final design, implementation challenges, budget, schedule and group dynamics.

InfraVision consists of six dynamic, innovative and motivated fourth year engineering students: Victor Song, Hans Ting, Harry Chen, Sae-Won Lee, Susan Chiu and Wayne Huang. If you have any questions or concerns with our process report, I can be contacted at 604-291-8498 or by email at iv-ensc@sfu.ca.

Sincerely,

Victor Song

Victor Song
CEO and Chairman
InfraVision

Enclosure: *Process Report for an Intelligent Monitor Power Saving Security System*

School of Engineering Science • Burnaby, BC • V5A 1S6
http://www.ensc.sfu.ca • iv-ensc@sfu.ca

# SenSaver ä

## ENSC 340 Process Report by

## InfraVision


| | |
|---|---|
| Team Members | Hans Ting<br>Harry Chen<br>Sae-Won Lee<br>Susan Chiu<br>Victor Song<br>Wayne Huang |
| Contact | Victor Song<br>vksong@sfu.ca |
| Submitted To: | Dr. Andrew Rawicz<br>Steve Whitmore<br>School of Engineering Science<br>Simon Fraser University |
| Submitted: | December 21, 2002 |

# Table of Contents

# Glossary

| | |
|---|---|
| API | Windows Application Programming Interface, a collection of accessible Windows codes |
| GUI | Graphical User Interface |
| HID | Human Interface Device |
| IR | Acronym for Infrared |
| LCD | Liquid Crystal Display, a technology for producing low-power and low-profile monitors |
| LED | Light Emitting Diode |
| MCU | Microcontroller Unit |
| Proximity Sensor | A sensor that detects how far away an object is from the sensor |
| RF | Radio Frequency |
| Transceiver | A device that transmits and receives signals |
| USB | Universal Serial Bus, a general high-speed connection for linking electronic devices |

# List of Figures and Tables

# 1. Introduction

Over the past four months, InfraVision has transformed a conceptual design into a functional system. The team members of InfraVision have demonstrated diligence and coherence during the process, and have gained considerable experiences in applying the expertise unique to each member. The composition of diverse technical backgrounds from each team member makes us a strong team; having an excellent management from the team leader, we have encountered but overcome many obstacles. We have proudly achieved an overwhelming majority of the goals that we have proposed in the functional and design specifications. This report summarizes the process that we have gone through and the difficulties that we have encountered.

# 2. System Design

## 2.1 Current System Operation

We have implemented the high level system according to the design specifications. Figure 2-1 illustrates the system block diagram of SenSaver™ and SenSaver Pro™.

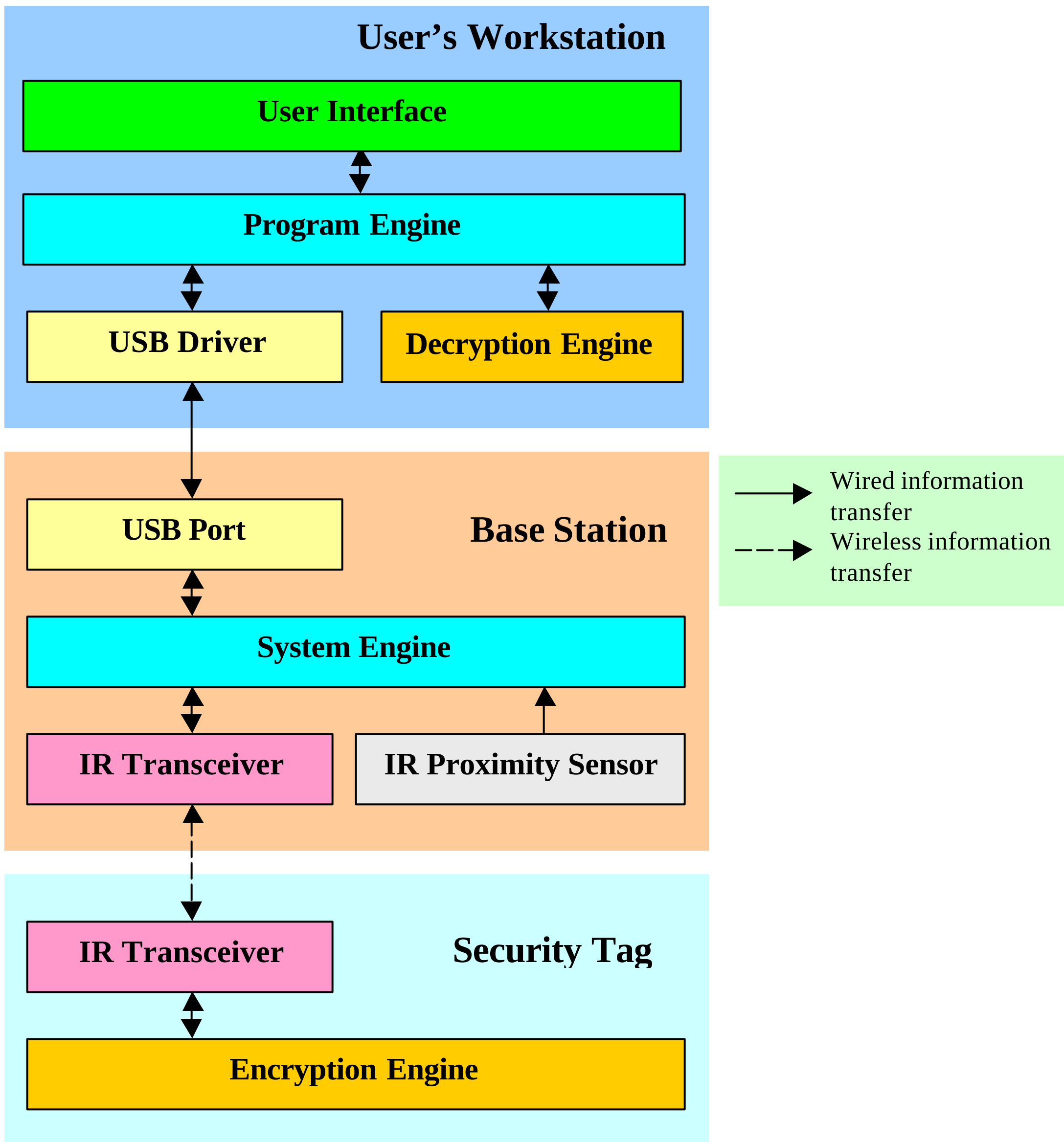


**Figure 2-1. System Block Diagram**

SenSaver™ and SenSaver Pro™ both share the same base station hardware and PC software, while SenSaver Pro™ utilizes additional security tags. When users start their computers, the PC software is launched and runs as a background task. The PC program retrieves the user's distance measured by the

IR proximity sensor through the base station via the USB port. The PC software will determine whether a user is present, and will activate the power management immediately if the user is away. If SenSaver Pro™ is enabled, the program will also lock the workstation immediately when the user leaves the workstation.

When the user returns, the program will send a wireless communication request to the base station. The base station will relay the request to the security tag using the IR transceiver; the base station will then return the reply from the security tag back to the PC program. Further wireless communication will take place, which will determine if the user wearing the tag is authorized to access the computer.

# 3. PC Software

## 3.1 PC Software Current Status

### 3.1.1 PC Program Engine

The program engine constantly polls the base station for the presence of the user. Upon detecting the absence of the user, the program will activate power management and security features. USB HID driver is utilized for the communication between the PC and base station. Note that the security features are only available for SenSaver Pro™.

The program can activate the screensaver and turn off the monitor; theses power management features can be activated after a certain period of time. In order to set the monitor into standby mode, we change Microsoft Windows' settings so that the monitor is turned off on our program request. When the user comes back into the range of the proximity sensor, we simulate a mouse movement to turn the monitor back on. The user can resume work with the original position of mouse. When the user is back, the turn off monitor setting is restored. The same situation applies for the screensaver except that we do not have to modify Windows' settings.

The program will lock the workstation immediately when the user leaves the workstation; a security lock program is launched. If the security tag fails to communicate properly with the base station, the user can log in by typing the password. In addition, unauthorized users cannot bypass the lock program using ctrl-alt-del, alt-tab, and Windows keys. The program will query the security tag and logs the user into the system if the security tag contains the correct tag ID and key.

### 3.1.2 Graphical User Interface

On the final design of our graphical user interface (GUI), we provide two option boxes that allow users to select the version of SenSaver product that they prefer to use. After that, users can set the sensor distance, which determines the detection distance of the device. Two types of power management are offered to the users, turn on the screensaver and turn off monitor (stand-by). The GUI also allows the user to set the launching delays of the power management after he or she leaves the workstation.

For SenSaver Pro™, the GUI provides the features of adding and removing tag ID's. A table will be shown in GUI to inform the user which tag ID's are authorized by the system. The GUI will ask for proper authorization by requesting username and password before any changes can be made in the tag ID's list.

All the values set by the user will be updated upon the clicking the OK button. A help document is also accessible to the user through the Help button. The following picture presents a vivid image of the product's GUI.

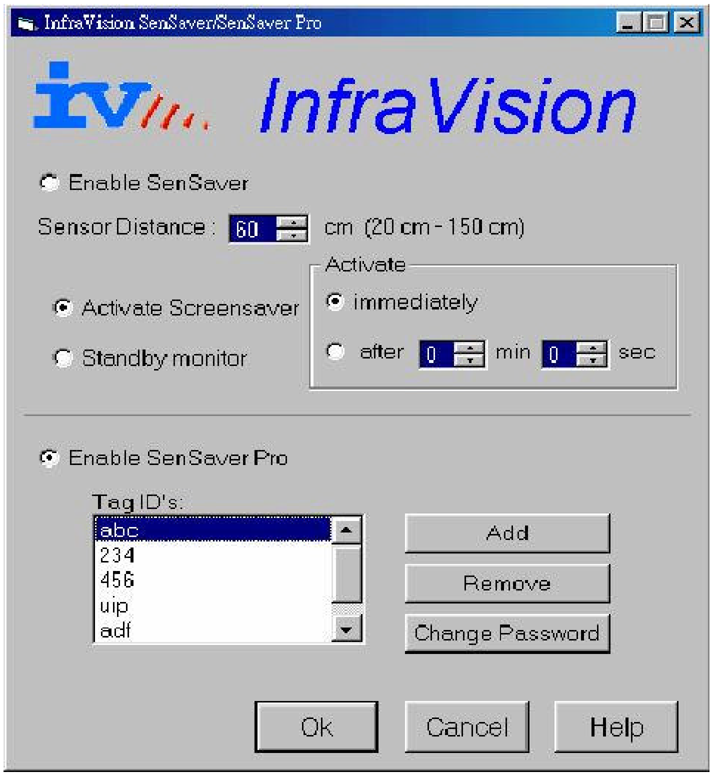


**Figure 3-1 Screenshot of the Graphical User Interface**

## 3.2 PC Software Deviation from Functional Specifications

The design parameters that do not match our specifications are summarized in Table 3-1.

**Table 3-1** Functional Deviations

| Parameter in Functional Specifications | Parameter in Final Prototype Design |
|---|---|
| "The UI must allow users to enable or disable the device." | We did not implement an enable/disable feature due to the tight project timeline. Currently, the GUI only allows users to enable either SenSaver or SenSaver Pro version of the product. |

## 3.3 PC Software Deviation from Design Specifications

### 3.3.1 Program Engine Deviations

The key deviation from the design specifications is that the main PC program does not run in the background (i.e. it is hidden from the user's view). The PC program currently appears as a blank DOS window. We were not able to find any functionality built into Microsoft DOS that would allow us to run the program engine in the background. This constraint is a sharp contrast to UNIX based operating system where the user can run program in the background by supplying a few parameters when calling the program. However, we can hide our program in the background by writing our program as a Microsoft Windows based program. Due to time constraints, we were not able to port our DOS based program to a Windows based program with the ability to run in the background. One bug was also discovered while testing; the timers for the activation of power management only worked for the first time it was set. Subsequent activation will fail to utilize the timers. This bug is due to the fact that the timer variable is not updated properly.

An additional feature that was not put into the functional and design specifications was the ability to change the settings while the main program is running.  We added this feature because it makes the program user-friendly. Since we added this feature quite late in the development cycle, it caused the bug described above to creep in.

For the USB driver, a Human Interface Device (HID) driver was used instead of the proposed Jungo WinDriver development tool. **W**e were worried about the PC driver for our device since no one in our group had prior experience with Windows API's (Application Programmable Interface). However, we avoided writing the Windows USB driver by utilizing the Windows HID driver, which is available in every Windows operating system. The reason we set our device as a HID device is that our product has the common characteristics as other HID devices; they all use low data transferring rate. As a result, we do not have to specifically design a driver for our product.

### 3.3.2 Graphical User Interface Deviations

Initially in the design specifications, we planned to gray-out the tag ID portion of the GUI and "request for an administrator password" when SenSaver Pro is disabled. Nevertheless, due to technical barrier and time constraint of the project, we were not able to implement these GUI features. Before we started this project, no one in our group had experience with Visual Basic. Despite the fact that we have technical difficulty programming in Visual Basic, we still choose to create GUI using Visual Basic because Visual Basic is a dedicated program for building GUI based application. Creating the same graphical interface in C or C++ would take a much longer time.

# 4. Communication Protocols

## 4.1 Communication Protocols Current Status

We are currently utilizing two proprietary communication protocols for communication between the PC, base station and security tag.

### 4.1.1 PC to Base Station Protocol

For the communication between the PC and the base station, a USB channel is utilized. A 5 bytes packet is used to transmit information. The 1st byte contains the header information, which describes the purpose of the packet such as requesting proximity data and generating an error flag with the necessary error information. The last 4 bytes contain the data.

### 4.1.2 Base Station to Security Tag Protocol

The wireless protocol is illustrated in Figure 4-1. The overall handshake protocol has not changed from the design specifications.

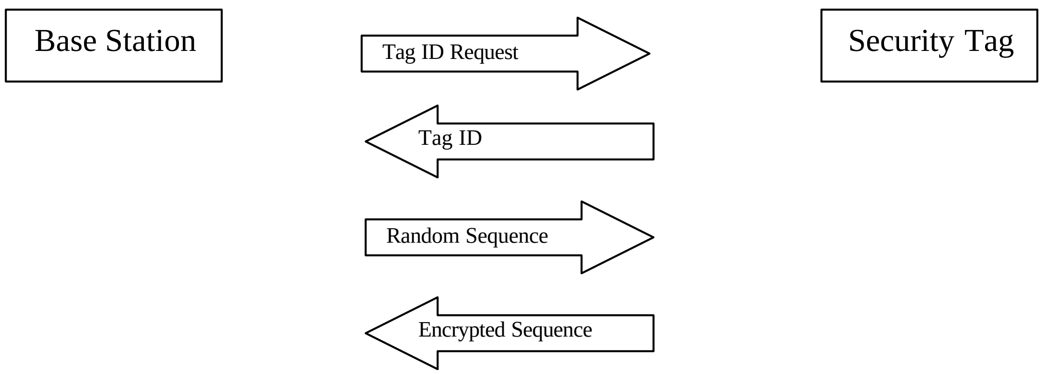


**Figure 4-1 Handshake Protocol**

In addition, the IR frame is illustrated in Figure 4-2. The final design transmits the same IR frame as devised in the design specifications.

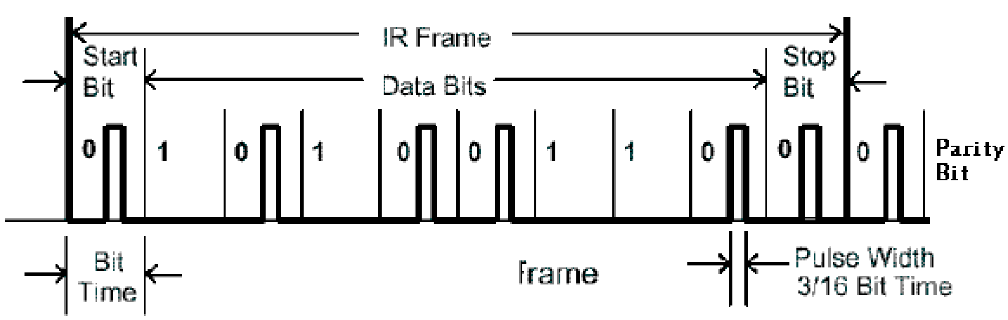


**Figure 4-2 IR Frame Layout**

## 4.2 Deviation from Functional Specifications

There are no deviations from the functional specifications.

## 4.3 Deviation from Design Specifications

In terms of the design specifications, the wireless protocol has differed in terms of error checking. The parity is still generated and transmitted for each IR frame but the parity is not used to determine the integrity of the data. This limitation is due to the fact that we had problems with capturing the parity bit correctly. The data, start and stop bits were usually captured correctly except for the parity. Therefore, we decided to drop parity checking. Due to time constraints, we decided not to add the error handling capability, where after transmitting an IR frame, the transmitting device (base station or security tag) will transmit an error message if it fails to receive data correctly. Furthermore, it was found that the tag would sometime miss the first bit for the first IR frame. Therefore, we decided to transmit 5 bytes where the first byte acts as a training byte for both the security tag and base station to properly synchronize the data receive. The deviations above did not affect the overall communication between the tag and the base station because it works flawlessly for a majority of the time.

# 5. Base Station

## 5.1 Base Station Hardware

### 5.1.1 Current Status

For the current base station hardware setup, a PIC16C745 microcontroller, a Sharp GP2W0002YP IR transceiver and a Sharp GP2Y0A02YK IR proximity sensor are utilized with supporting components. The operating ranges of the transceiver and proximity sensor meet the functional specifications, and they provide reliable performance.

#### 5.1.1.1 Microchip PIC16C745 Microcontroller

We use Microchip PIC16C745 microcontroller for our base station. This microncontroller has powerful built-in features that will speed up our design process. By choosing this chip, our base station design is a one-chip solution, which is more cost-effective and means one less interface connection between a microcontroller and USB chip.

PIC16C745 has 5 A/D channels, and we utilize one of the channels for the proximity sensor, which outputs certain voltage value depending on the distance between the object and sensor. We use the PWM feature to generate desired IR signals through the transceiver, and employ the input capture function to receive the IR signals. We purchased a low cost Microchip's PIC16C745 evaluation board containing examples codes, which shortened our development time.

However PIC16C745 has its disadvantages. It uses EPROM, which means we have to utilize three different tools to reprogram and test the chip (UV eraser, programmer and EVB). This process became tedious during the code debugging process. In addition, we had to spend extra money on the UV eraser and more chips for the programming and erasing iterations. From our research, Microchip will have an EEPROM version of USB microcontroller, which will be a very good option for future development of this project.

### 5.1.2 Deviation from Functional Specifications

There are no deviations from the functional specifications.

### 5.1.3 Deviation from Design Specifications

There are no deviations from the design specifications.

### 5.1.4 Issues Encountered

We have encountered some problems with the IR transceiver modules. Some only worked for transmitting but not for receiving, and some only worked in reverse way. We later realized that the data sheet recommends 230°C and less than 5 seconds of soldering time. Since they are surface mounted modules, prolonged soldering time might have damaged some of our transceiver modules. This problem caused us a lot of time soldering new transceiver modules, which takes approximately 4~5 man hours (including testing).

## 5.2 Base Station Firmware

### 5.2.1 Current Status

The system engine for the base station co-ordinates the following tasks

- Sample the proximity sensor
- PWM (Pulse Width Modulation) for transmitting IR data via IR transceiver
- Input capture for receiving IR data via IR transceiver
- USB communication

Refer to the appendix for the firmware code.

#### 5.2.1.1 Input Capture

In order for the base station to capture the IR data sent by the tag, we employ the PIC16C745 input capture feature. The program starts the whole timing sequence based upon the detection of the starting bit. An array is utilized to store the timer values whenever a falling edge on receiving pin is detected. An impulse at the receiving pin indicates a “0” in a data. A “1” does not pull the pin high or low. Thus all timer values store in the array indicates the time when a “0” bit occurs. Taking that data and our wireless protocol data transfer rate, we are able to reassemble data received from the tag by inserting proper number of "1" between two zero bits.

#### 5.2.1.2 PWM (Pulse Width Modulation)

To output IR signals, the transceiver modules require a duty cycle of 3/16 of the period for transmitting a "0" and a duty cycle of zero for transmitting a "1". Based on this transmission scheme of the IR transceiver, we use the PWM module supported by PIC16C745, with duty cycle 3/16 of the period.

#### 5.2.1.3 USB Firmware

We have encountered major obstacles with USB interfacing. None of our members had any prior experiences with USB. Furthermore, the documentation available lacks the details required for our firmware because our device is very specific to a certain extent. With trial and error and reading through hundreds pages of documents, we finally figured out how to setup a communication channel between the device (our base station) and the host (PC).

In order to establish the USB communication channel, the USB device must have *descriptors* properly written. A descriptor is a data structure that has a defined format and tells the host about the device attributes. There are four descriptors that are required to setup a USB device, which includes device descriptor, configuration descriptor, interface descriptor and endpoint descriptor. These descriptors will setup important attributes such as the device name, maximum current supplied from the USB port, data format in each packet and etc.

We setup the base station as a USB HID device to ease the troublesome Windows driver development. To declare our base station as a HID device, the base station must have an additional *report descriptor* that contains the necessary information for the PC to recognize the base station as a HID device. The interface descriptor set the device class as HID class and the report descriptor contains information about the HID device's data structure and usage.

### 5.2.2 Deviation from Functional Specifications

There is no functional specifications deviation.

### 5.2.3 Deviation from Design Specifications

There is no functional specifications deviation.

# 6. Security Tag

## 6.1 Current Status

The security tag consists of a PIC16F84A microcontroller, a Sharp GP2W0002YP IR transceiver, and a Panasonic CR2477 3V battery. The security tag transmits and receives IR signals via IR transceiver and the firmware will process the wireless handshaking protocol as defined in section 4.1.2 and the data encryption.

### 6.1.1 Hardware

The security tag needs to be able to communicate with base station via IR signals, handle the wireless handshaking protocol correctly, encrypt the random number sequence that the base station sends, and last for a month on one battery.

#### 6.1.1.1 Microchip PIC16F84A Microcontroller

Based on the requirements for the security tag, the PIC16F84A microcontroller from Microchip is used in our security tag. This particular PIC microcontroller provides more than enough functionality for the tag, and its low cost and low power consumption makes it an attractive choice for us.

#### 6.1.1.2 IR Transceiver

The IR transceiver in the security tag is identical to the one used in the base station for ease of design and manufacturing.

#### 6.1.1.3 Battery

The battery in the security tag should have a high capacity to guarantee long continuous operations. The Panasonic CR2477 3V battery is chosen because of its 1000mAh capacity and a substantially low cost. Based on average power consumption calculations and under the assumption that the tag operates continuously, the tag is capable of lasting for a month. With a power-switching button implemented, the tag will last much longer than a month.

#### 6.1.1.4 Deviation from Functional Specifications

The design parameters that do not match our specifications are summarized in Table 6-1.

**Table 6-1** Functional Deviations

| Parameter in Functional Specifications | Parameter in Final Prototype Design |
|---|---|
| "The tag should utilize an on/off switch or a triggering button to save power consumption when not being used." | We did not implement a power-switching button due to the tight project timeline. However, it is an easy feature to implement and it will be for the commercial product. |

#### 6.1.1.5 Deviation from Design Specifications

There are no deviations from the design specifications.

### 6.1.2 Firmware

#### 6.1.2.1 Generation of IR Signals

In order for the IR transceiver to transmit a "0", a duty cycle of 3/16 of the period is required and a duty cycle of 0% is required to transmit a "1". Since PIC16F84A does not support PWM module, we have to pull the output pin high for required time and pull the pin low afterwards, which is equivalent to PWM with duty cycle of 3/16 of the period.

#### 6.1.2.2 Receive of IR Signals

Since PIC16F84 does not support input capture functionality, interrupt on pin change and the associated interrupt flag are used to determine the IR signals received by the transceiver. When the IR transceiver receives a pulse, which represents a "0", the interrupt flag will become high, and based on the time difference between these interrupt flags, we can resemble the data in a frame.

#### 6.1.2.3 Deviation from Functional Specifications

There are no deviations from the functional specifications.

#### 6.1.2.4 Deviation from Design Specifications

##### *6.1.2.4.1 Microchip PIC12C508A Microcontroller*

We had a critical issue with the microcontroller selection. Originally the PIC12C508A was chosen in our design specifications because it is the simplest chip available from Microchip that meets the proposed requirements.

However, during the development process we have discovered that the PIC12C508A is unable to process the data received from the IR transceiver. Furthermore, 25 bytes of RAM, 512 words of ROM, and 2-level hardware stack are inadequate for our firmware.

Therefore another chip that has either a higher clock speed or an interrupt on pin change needs to be used. With less than 3 weeks from our proposed project demonstration date, we decided to replace PIC12C508A with PIC16F84A because of the following reasons:

1) More resources (68 bytes of RAM, 1024 words of ROM, 8-level hardware stack)
2) Supports interrupt-driven data receive
3) CMOS-based EEPROM/Flash device (eliminates the need for a UV Eraser)
4) 10 MHz clock (instead of 4 MHz)

One thing we would have done differently if we had the chance to do it again is to realize that there is a trade off between size and functionality. Because we were so reluctant to trade size with functionality, weeks of development were spent optimizing the codes and debugging. Our firmware development would have been much easier if the PIC12C508A supported interrupts.

#### *6.1.2.4.2 Data Encryption*

In our design specifications we stated that the MCU would be utilizing the TEA (Tiny Encryption Algorithm) for encryption because of its robustness and low hardware requirements. However, due to the tight time constraint and exam conflicts, a simpler data encryption scheme, bit-wise-XOR, is implemented instead. Since the prototype is just a proof of concept, as long as we can demonstrate that a secure date encryption scheme can prevent unauthorized user from logging-in the workstation, we can considered our goal is achieved.

## 7. Printed Circuit Board (PCB)

We have designed a PCB for both of the base station and security tag. We used a PCB development kit, which constraint us to a single layer board. We could not get a single layer layout for our base station. So we decided to use jumper wires on the PCB to make the circuit completed. The following figures show our base station and tag PCB layouts.

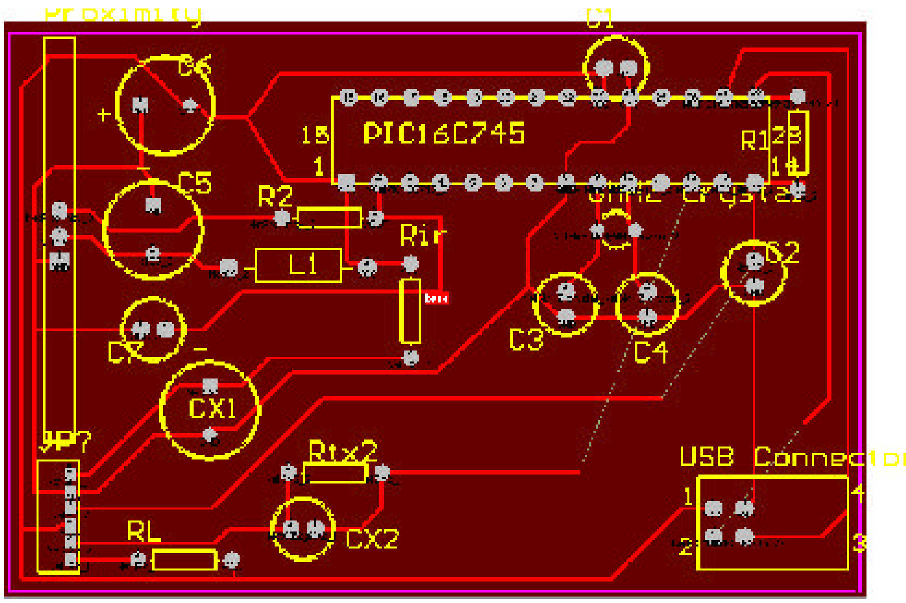


**Figure 7-1 Base Station PCB Layout**

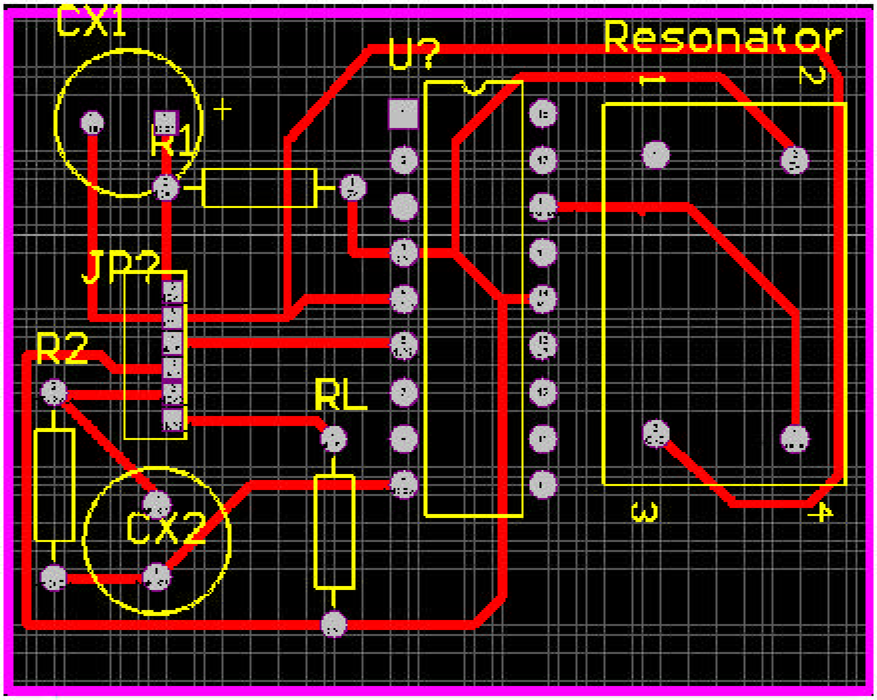


**Figure 7-2 Tag PCB Layout**

After we had the layout, we used the kit to develop the PCB. It took some time to learn about the manufacturing process. Because our team members do not have any experiences in PCB manufacturing, we failed to make a working PCB. We believe that if we have more boards for us to try out the UV and etching processes, we would have successfully built a working PCB-based hardware subsystems.

# 8. Budget and Scheduling

## 8.1 Budget

Table 8-1 summarizes our estimated versus actual costs of our project to date.

**Table 8-1 Budget of Estimated Vs Actual Costs**

| Items | Estimated Costs | Actual Costs |
|---|---|---|
| Analog IR Sensor | $100 | Sample |
| IR Transceiver | $50 | Sample |
| Microcontrollers and EVB | $620 | $414 |
| USB Connector and Cable | $50 | $14 |
| Encryption Encoder and Decoder | $50 | Not used |
| Perf Boards | $50 | From Fred |
| Reference Books | $100 | Not used |
| Miscellaneous Electronic Components | $50 | $32 |
| PCB Printing | $150 | $25 |
| UV Eraser | Not included | $64 |
| **Subtotal** | **$1220** | **$549** |
| 15% Contingency Fund | $180 | $83 |
| **Total Budget** | **$1400** | **$632** |

The actual cost is much lower than the estimated cost because we were able to sample a lot of key components from distributors and manufacturers. Furthermore, we were able to attain parts from Fred Heep.

## 8.2 Scheduling

Figure 8-1 and Figure 8-2 illustrate the projected and actual schedule and milestones, respectively.

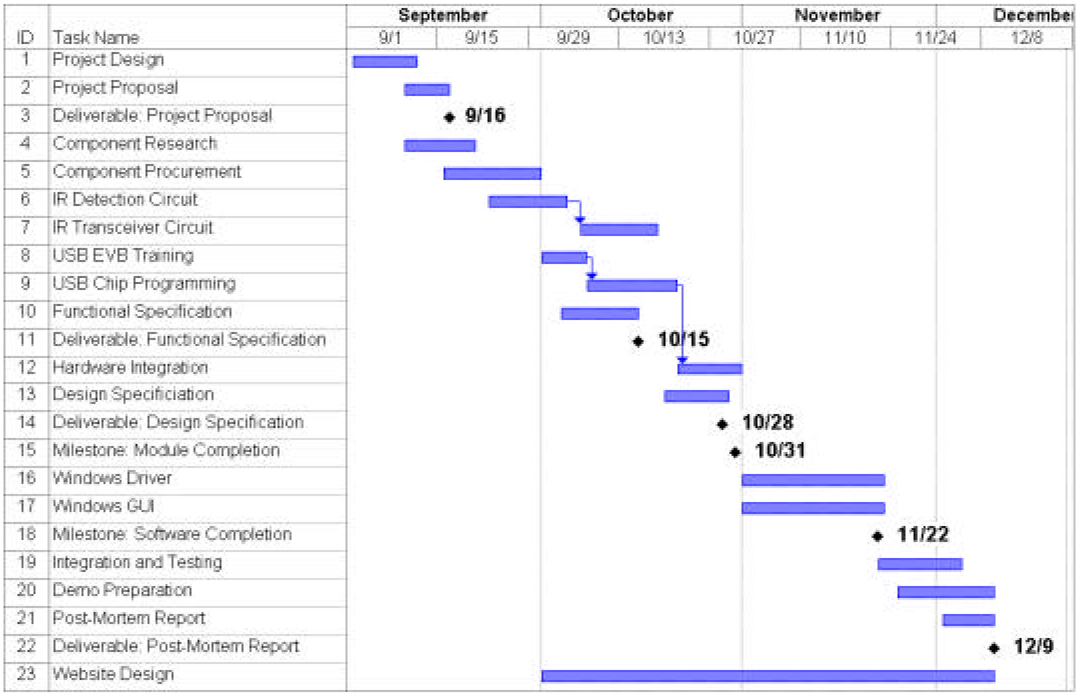


**Figure 8-1 Tentative Project Schedule and Milestones**

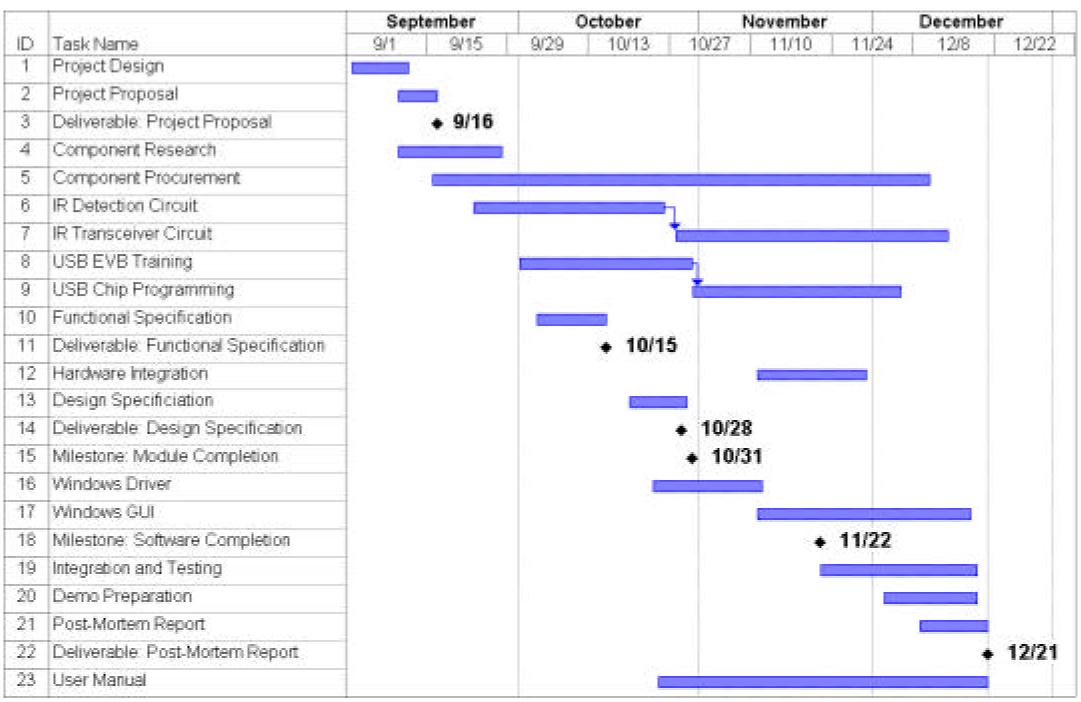


**Figure 8-2 Actual Project Schedule and Milestones**

There are some differences between the projected and actual schedules because of difficulties with USB development, security tag microcontroller change, IR transceiver firmware and circuitry. Heavy course load and exam schedules also slowed the progress of the project implementation. In the future, we will utilize the following rule. Take each task and double their time requirements and then sum up all the time required for all the tasks and double that time again. Especially time consuming are the project integration and firmware debugging.

## 9. Future Improvements

Currently the encryption scheme is used as a proof of concept. However, we are planning to use much more secure commercial grade encryption standards in the future.

Before heading into the market, one of our major goals is to integrate the base station into a monitor to reduce the inconvenience of having another attachment to a PC. Also, if a consumer is buying a new monitor, he/she would be more likely to invest an additional $20 dollar on top of a $300 monitors for the additional features.

Before we bring our product into the market, we would like to add another new feature to our product, such as a heat sensor. We haven't made our decision to include the feature for our real product because adding the additional feature would increase the production cost. At the initial stage we were thinking about adding this feature. However, we thought that when a user leaves the computer, usually the chair would be far enough such that our product would detect the absence of the user. So we had to make a tradeoff between the cost and effectiveness of our product. Our priority was cost, so we did not include the heat sensor as our functional requirement. Other sensors such as a motion detector might be used.

## 10. Future Plans

We are going to work closely with UILO to bring our product into the market. UILO will help us patent our product and look for a monitor manufacturer to license our idea. We will also try to look for regulation bodies so that our product could become a mandatory feature for all computer users. As a matter of fact, we will demo our product to UILO in the coming weeks.

# 11. Group Dynamics

For our group, we utilized a matrix organization. For a matrix organization, each team member has a main task and several subtasks. Usually for a specific component, there will be two people working on it, where one person is mainly responsible for it while the other is there to help. This setup allowed all the group members to get exposure to hardware, firmware and software implementation and brings diverse problem solving skills to each task. Furthermore, we had weekly meetings to keep all members updated on progress and utilized the meeting as a forum to discuss key issues and problems. Moreover, each team member was given leadership roles in the meetings with a new chair and meeting minutes recorder for each meeting.

Lastly, since the group consists of 6 people with different courses, we utilized the web extensively to keep the group synchronized. On our web page, we had a section where we can submit URL on our research and other group members are updated on the status of our research on components or algorithms. We also had a “Bill of Material” section, which would allow members to add components that they want to acquire with the cost information plus pros and cons. Once the part is acquired, it can be checked off. We can say that after this project we have become very good friends, much closer than when we started.

# 12. Team Member Impressions

## Victor Song

My contribution to the team encompasses the PC, base station and security tag. For the PC side, I was responsible for designing the program engine that works with the GUI and USB. For the base station side, I was involved with building the IR proximity sensor circuitry and writing the A/D conversion code. Furthermore, I wrote the system engine for the base station. For the tag side, I was involved with debugging the main program. Lastly I was responsible for the IR frame layout and the wireless protocol design. In addition to the technical contributions, I was involved with organizing the team and directing the design flow.

From this project, I have learned many new technical skills in hardware, firmware and software. For hardware, I have learnt to utilize filters extensively for processing signal and how to use key peripherals in PIC microcontrollers. For firmware, I have learnt new debugging techniques to streamline firmware development. In addition, for software, I have learnt the importance of getting users feedback for designing PC software. Lastly, I have learnt that when leading the team, it is important to allow group members to innovate and come up with creative ideas to problems instead of providing too much constraint on how the design should go.

## Susan Chiu

For the past four months, I have come to realized that how much effort it requires to accomplish a decent project. Nevertheless, the knowledge and experience I gained is priceless.

As of the technical aspects, I was responsible for the IR wireless communication between the base station and the tag. The communication system is a real-time process. Though the wireless protocol we have in our project is a simple two-stage hand-shaking algorithm, I still faced a lot of challenges just to figure out when the whole communication starts and how to synchronize the process on both the receiving and sending sides of the system. The timing is crucial. Missing by one millisecond could result in a fatal failure in the whole process. Once the hand shaking is broken, no correct action can be made by the processor. I realized that it takes not only a good algorithm to implement the communication process, but also a careful reading on the limitations and functions available on the chip. Knows what resources I have in hands, and use them well to accomplish my goals.

On the other hand, I was also responsible for the implementation of the graphical user interface. The greatest challenge I faced was the technical barrier of programming in Visual Basic. Before I finish this project, I had absolutely no experience on Visual Basic. I had to do a lot of readings on Visual Basic reference books/files, Internet research and consulting with other Visual Basic programmers before I could start. I also faced a dilemma of choosing ways to implement the functions on a language that I had never used. Nevertheless, I found out that a good use of help files and consulting with others usually directs me to right path in the shortest time.

Despite the technical aspects that I gained from the project, I also realized that it takes not only personal input, but far more importantly, it takes the co-operation and support from every member in the group to accomplish a respectable project like this. During the past four months, I had been exposed to a group who had been able to standby each other when difficulties arrive and compromise each other when disagreements hit. The team has developed a very good listening skill and a democratic system. We listened to each other's opinions and provided feedback. Then we decided the route as a team by casting a vote. We had a matrix system to divide up the tasks. We could sign up for the tasks that we wished to take on. Most tasks have a group of two people working on, and each person undertakes more than two tasks at the same time. This matrix may divide up the concentration of one person on one task, but it gives everybody a chance to gain different aspects of the project and the knowledge that we wish to learn the most. Moreover, we have a partner while we working on the project. That way we have someone to consult with when problems encounter and the progress wouldn't solely depend on one person.

After this project, I learned the flow of creating a product starting from scratch, and how much sacrifice has to be made to accomplish a goal. I also realized the importance of teamwork. No one is perfect, but a team could be perfect.

## Hans Ting

After three years of engineering, I have finally proved that I have the ability to work in a team and contribute my skills to develop a product from the ground point. Our team has a great group dynamics and democracy. We cast votes when there are important decisions needed to be made. We have weekly meetings are matrix work distribution. Overall it was a great group working experience and I think that I have enjoyed a lot.

On the technical aspect, I was responsible for the communication between PC and USB device. I have read through hundreds pages of documents to understand this complicated, but user-friendly standard. I think that our group was very wise to choose USB as our communication protocol, which gives us more understandings about the communication protocol other than RS-232. It was a risky decision, but after all, we could have just fall back to RS-232 if we failed. Fred Heep actually told us, "The easier for the user, the harder for the designer." He actually suggested us to fall back to RS232, because he thought we could not have the USB device working in merely four months. However we managed to have our device working as a USB device, which gives our product more competitive in the market.

At the end of the project, I worked mostly on the IR communication between the bast station and the tag. The main technical obstacle for the IR communication is the *synchronization.* It is difficult to get two different devices in the same state when the timing is crucial for the wireless communication. Although our device is working fine with the current algorithm, I believe that there are some other ways of designing a better communication algorithm.

In next year, we will demonstrate our project to UILO and hopefully we can patent our device and license to monitor manufacturer. I think that our project is a complete success and I have gained a lot of valuable experience from it.

## Sae-won Lee

When we decided what our group was going to do for the project, I thought that the project idea was very simple and straightforward. However, I figured that the development process would be not as simple, and my expectation came true. One of the reasons was that we had not previously had any Windows programming experience or USB development skill. My first task was getting information about making Windows drivers that could be used for our project. Since we knew that we would have to use a driver for our hardware and did not know how to make one, I had to figure out how we would be able to make drivers. When we learned more about USB, we realize that our device could be categorized into a HID device whose drivers are included in Microsoft Windows. Therefore, all the research and the preparation were thrown away. This was a problem for doing something totally inexperienced with.

The next part I worked on was to write a program for computers to communicate with our device through USB and to determine actions to be taken depending on situations. At the same time, I helped other people to debug their parts of the project. However, the end of the semester was approaching, and we realized that we needed to have a special meeting to come up with a better plan so that we can finish our project on time and to do well in other courses. I almost finished my parts and someone had to move on to figure out how to activate and deactivate Windows screensaver and power management. So, I took the parts. We had no idea how to write a Windows program or how to use Windows API's, and it was difficult to find any information about what we wanted to do. I learned many aspects of Windows programming, but the process was very stressful because what we do was not used by people often and thus was not easy to find information about it. Finding out ways to implement the desired functions took more than three days. Also, I had to learn Visual Basic to implement functions. At the same time, I helped in implementing the graphic user interface (GUI). Since the person who worked on the GUI also did not have previous Visual Basic experience, we helped each other.

On the presentation day when we came to the lab to do the final test, nothing seemed to work. A panic took over us. We figured out the problem, but for the moment we had the most terrible experience.

During this project I learned the difficulties that come with developing a new technology and realized that I should be prepared for any worst case scenarios.

## Wayne Huang

From the technical perspective, I worked primarily on the wireless IR communication between the base station and the security tag. Although it might not seem a very difficult task, since the base station and the tag basically just need to send and receive data properly, it has been one of the most frustrating debugging process ever. Since all IR communication between the base station and the tag are real-time signal processing, a miss by a factor of milliseconds, even microseconds, could cause us missing an entire frame of data. One thing I have learned from this frustrating experience is that we should never get too hung up just to meet our minimum requirement when doing a prototype design. When a chip does not have enough resources for our design, make the decision of switching to another chip fast, size is not that important for a prototype design since we can always find a better alternative chip in the future. Spending too much time just on one firmware component, causing delays on the entire project development, will just come back and haunt us in the crunch time. Failing to realize that made us technically camp in the lab for the 3, 4 days before the demonstration date

One of most important attributes I learned from this course is that planning ahead is critical in succeeding in any project. One thing we could have done better is to make better planning so that we work on the project more before all the final exams and labs cramp together, which idled our development for a while.

One thing I think we have done well is we had regular meetings. Meetings allow each team member to update the team with his/her most recent progress and thus make it easier to integrate the project since everybody has a clear idea what other members are doing. Also, one important attribute I have gained from the meetings is leadership skills. Since we rotated our roles being meeting chair and minutes, each person had to take the leadership role to make sure the goals for each meeting are achieved. But one thing we could have done better is to stick to the agenda and never get too caught up in the problems. Long meetings sometimes are just inefficient and get us nowhere where we want to achieve for the meetings.

Listening and patience are also vital when working in a group, especially a big group of six. They allow each member to understand what others have to say, thus allow each member to help each other. I think we all have done this very well from this perspective. Also, our special work distribution "matrix" allows everyone to do more than the one main task he/she is assigned to, thus everyone can always get help from other team members when he/she runs into any problems.

## Harry Chen

From start of the project, I have participated in making the tag. I first came up with an infrared communication encoding scheme for secure wireless transmission. First, I tried to use public and private key for the security feature. However, we decided to modify the scheme little bit to suit our project better. At the same time, I was responsible for ordering all the necessary parts that are needed for the project. I kept the record of the acquired parts by updating the bill of materials section of our web site, so that every other group member would know easily what parts are available. Our group found a need for a summary of links so that other members of the group can also access all the information needed for the project. Therefore, I created a group web page, which can add useful links into categorized subsections.

When all the necessary parts were acquired and built, I was responsible for producing a firmware for our tags that allowed the user to access computer without typing the password each time. After spending a large amount of time, the microcontroller we had, was not powerful enough to communicate with the base station. We then had to look for another chip. I learned a lot about Microchip's product portfolio as a result.

# 13. Conclusion

Over the past four months, every member of InfraVision has toiled long hours to reach our mission. We are proud to say we have succeeded in developing an intelligent sensor unit integrated with computers, which improves power consumption, end user's experience and security. We will take these skills and experiences in tackling greater challenges ahead.