Virtual Key

Ztronicx Microsystems

Post Mortem Report for *VirtualKey*

Bluetooth Access Control System for Home Appliances

Project Team: Halim Soetanto
May (Mei-Yi) Liu
Rick Liu
Shahin Teymouri

Contact Person: Shahin Teymouri
ensc440-whizkids@sfu.ca

Submitted to: Andrew Rawicz – ENSC 440
Steve Whitmore – ENSC 305
School of Engineering Science
Simon Fraser University

Issued Date: January 11th , 2007

Revision: 2.0

January 11th, 2007

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

Re: ENSC440 Post Mortem Report for VirtualKey
(Bluetooth Access Control System)

Dear Dr. Rawicz,

The attached documents contain Ztronicx Microsystems' Post Mortem Report for VirtualKey (a Bluetooth Access Control System). The purpose of VirtualKey is to lock / disable any home appliances while parents / guardians are out of sight and to give temporary access to guests visiting the house via Bluetooth wireless link by a cell phone.

The attached Post Mortem Report includes the following major information of VirtualKey:

- Introduction of the VirtualKey product
- Current State of the Device
- Deviation of the Device
- Future Plans
- Budgetary and Time Constraints
- Conclusion
- Interpersonal and Technical Experiences

This Post Mortem Report describes the overall features that the current VirtualKey model has and discusses some future development which will further extend the functionalities to meet our design goals. If you have any questions or concerns about VirtualKey, you are welcomed to contact Ztronicx Microsystems by email at ensc440-whizkids@sfu.ca.

Sincerely,

Shahin Teymouri

Shahin Teymouri
President and CEO
Ztronicx Microsystems

Enclosure: Post Mortem Report for VirtualKey

Executive Summary

There have been a lot of cases where problems occur at home while an individual is away such as theft, fire or gas leakage accidents. Virtual Key wireless access control system targets on any unattended or invalid access of home appliances and doors. Adults, especially parents, can prevent their kids from playing with hazardous machines or devices and also if they have caregiver or guests, they could easily give them the access to any door or appliance. The elderly, small children and dementia patients will be safe at home by not triggering any unintentional devices causing an accident. Virtual Key allows the users to fully control lock/unlock and enabling/disabling features over wireless communication by a small plug-and-play module that can be connected to the electrical outlet.

Development process of Virtual Key, a wireless access control over home appliances, is divided into two phases. In the first phase, we are concentrating on proof of concept together with a working prototype which includes the following features:

1. Enable/Disable of home appliances which is determined by detecting the presence of cell phone within range.
2. Provide interface for user to arrange a list of individuals that have access permission by pairing the devices.
3. Encryption and pass-code protection.

The following post mortem describes the overall performance summary of the first phase of VirtualKey access control system. The current model has incorporated the features as outlined in our functional specification except a few minor add-ons which will be discussed in the post mortem as well as future plans and improvements.

A working model of VirtualKey system will be ready by January, 2007 for presentation purpose.

Table of Contents

List of Figures

List of Tables

Ztronicx Microsystems Inc.
Simon Fraser University
Burnaby, BC V5A 1S6
ensc440-whizkids@sfu.ca

Glossary

VirtualKey Hardware Module	The Hardware module that is installed in appliances, power bars, doors that will be controlled with VirtualKey Software
VirtualKey Software	The Java based software that is installed on JSR-82 capable phones.
JSR-82	Bluetooth Java API, a capability that is needed on the cellular phone before VirtualKey Software can run. Almost 90% of cellular phones are JSR-82 capable.
UUID	Universally Unique Identifier which provides communication between VirtualKey software and hardware module.
J2ME	Java 2 Micro Edition is a java platform for mobile application
CLDC	Connected Limited Device Configuration classifies mobile devices that can establish a network connection, and typically don't have a lot of processing power comparing with CDC (connected device configuration)
MIDP	Mobile Information Device Profile specifies the requirement for mobile devices
COD	Class of Device

1 Introduction of the VirtualKey Product

VirtualKey is a system that eliminates carrying of many keys or access cards just to open house, office, or car doors. VirtualKey can also disable dangerous appliances in a house when a person that has access to them is not around so that children will not be able to use them. These two functions of VirtualKey system, acting as a key to all doors and enabling and disabling the appliances, are all integrated into one simple device that can be installed easily in your house.

For the past four months, our Ztronicx Microsystems team has been developing VirtualKey prototype which can be modularized into two parts: Java-based software installed on your cellular phone and a hardware module installed on the electrical plug of your appliances or inside of your door locks. The software installed on your cellular phone will have a user interface that can send different pass codes to the hardware module over a short range wireless system that comes with the cellular phone.

The wireless system can use Infrared, Wi-Fi, WiMAX or Bluetooth technology but for this current VirtualKey System, we are concentrating on Bluetooth wireless for the phone-to-module communication. The pass code will travel from the cell phone to the VirtualKey hardware module over a wireless signal. The VirtualKey hardware module can then verify the pass code and activate the door or the appliances.

1.1 Scope

This document focuses on detailed performance of current working prototype of VirtualKey with a discussion of deviation, overall budget and timeline of the system to the proposed design. Problems encountered during development of first release will also be covered in the following sections as well as future plans and implementations for extended features such as disabling and enabling of door locks. Additionally, a brief summary of each group member’s technical background experience is also attached.

1.2 Intended Audience

Design engineers will use this document as the foundation for building the VirtualKey system.

Project managers will use this document to measure the performance, development objectives and to evaluate whether this project is following the required design.

Marketing team will use this document to create the initial advertisement and promotional content.

2 Current State of the Device

2.1 GUI Software

The software for the cell phone is written in J2ME technology and meets CLDC1.0 and MIDP2.0 requirement. The current version of the VirtualKey software has the following features:

1. **Password login protection**
   After launch the software, user requires to enter password to be able to enter the VirtualKey Search Device Screen. The current maximum password length is 9 digits fixed and case sensitive.

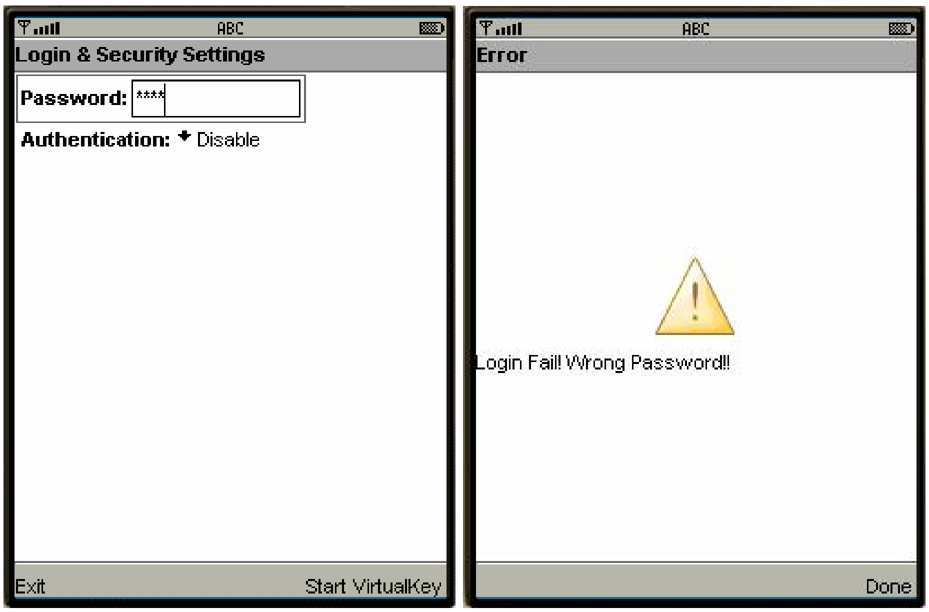

Figure 1: Screen Shot of Password Input and Login Failure

2. **Bluetooth Connection Security Setting**
   Before the user actually login, the user can also choose his security level settings. There are two settings: one is authentication and the other is encryption. However, encryption can be enabled only when authentication is enabled. According to Sun Java JSR 82 API explanation (http://java.sun.com/javame/reference/apis/jsr082/javax/bluetooth/RemoteDevice.html), "authentication is a means of verifying the identity of a remote device. Authentication involves a device-to-device challenge and response scheme that requires a 128-bit common secret link key derived from a PIN code shared by both devices. If either side's PIN code does not match, the authentication process fails"

Figure 2: Screen Shot of Security Settings

3. **Search VirtualKey Devices**

After user successfully login, the user can start to search for the VirtualKey devices by choosing "Search" function inside the menu selection. The search will search all surround Bluetooth devices and then filter out non-VirtualKey Bluetooth devices based on the COD (Class of Device) number:

Major Class = Computer Peripheral = #1280 (0x0500)
Minor Class = Remote Control = #12 (0x000C)
COD = #1280 + #12 = #1292 = #1292 (0x050C)

If there's no VirtualKey devices found, the screen will show an alert.

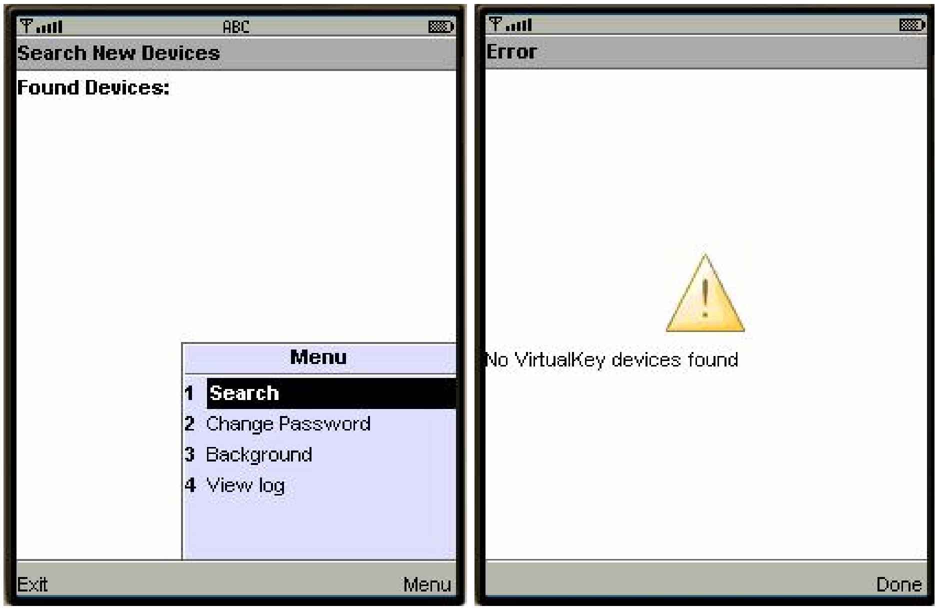

Figure 3: Screen Shot of Search Function and Alert

If there's at least one VirtualKey devices are found, each VirtualKey device will be shown on the screen with its current connection status, Bluetooth Address, and friendly device name. The current connection status is indicated by an image: a red stop sign means no connection and a purple star sign means being connected. The following is an example of screen shots.

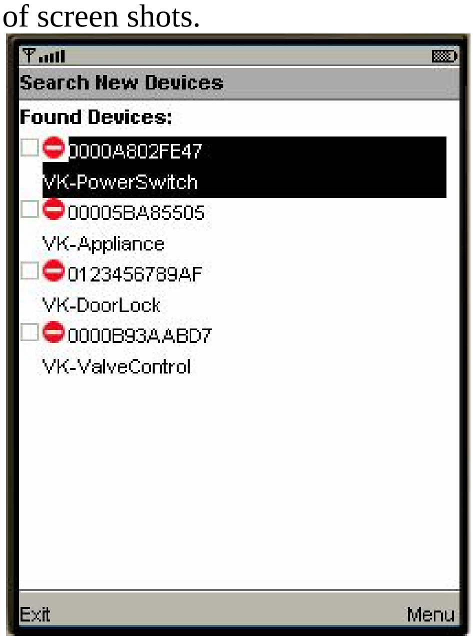

Figure 4: Screen Shot of Found Devices

4. Enable VirtualKey Devices

After the user has found at least one VirtualKey device, he can first select the checkbox of the VirtualKey device and then choose "Enable" function inside the menu selection. After few seconds delay, the current connection status image will be updated to a purple star image, which means the connection is established. The connection status image will be updated in real time. The following is an example of screen shots.

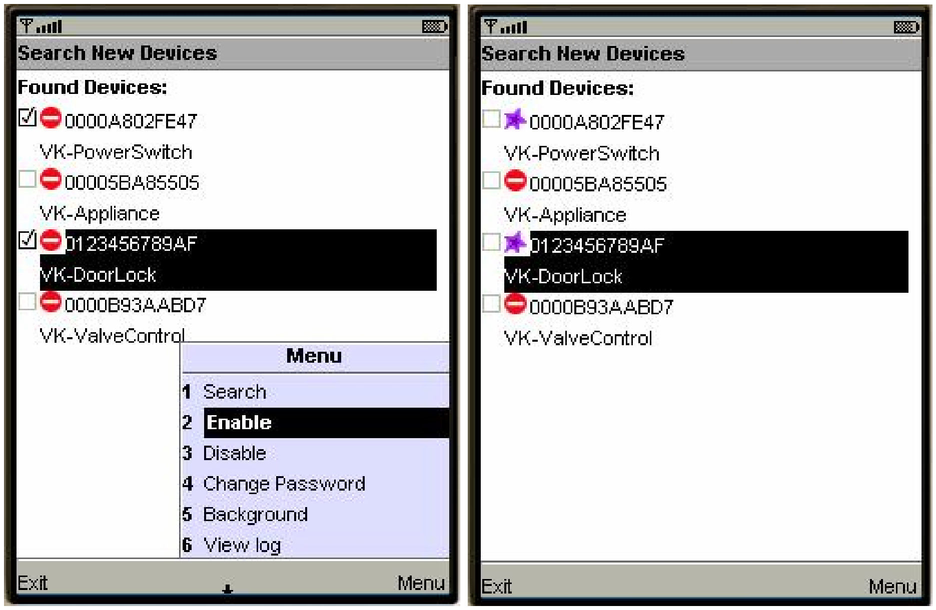

Figure 5: Screen Shot of Enabled Devices

5. Disable VirtualKey Devices

After the user enables the devices, the user can also disable the devices. The user can select the checkbox of the device that he wants to disable, and then choose “Disable” function inside the menu selection. The following is an example of screen shots.

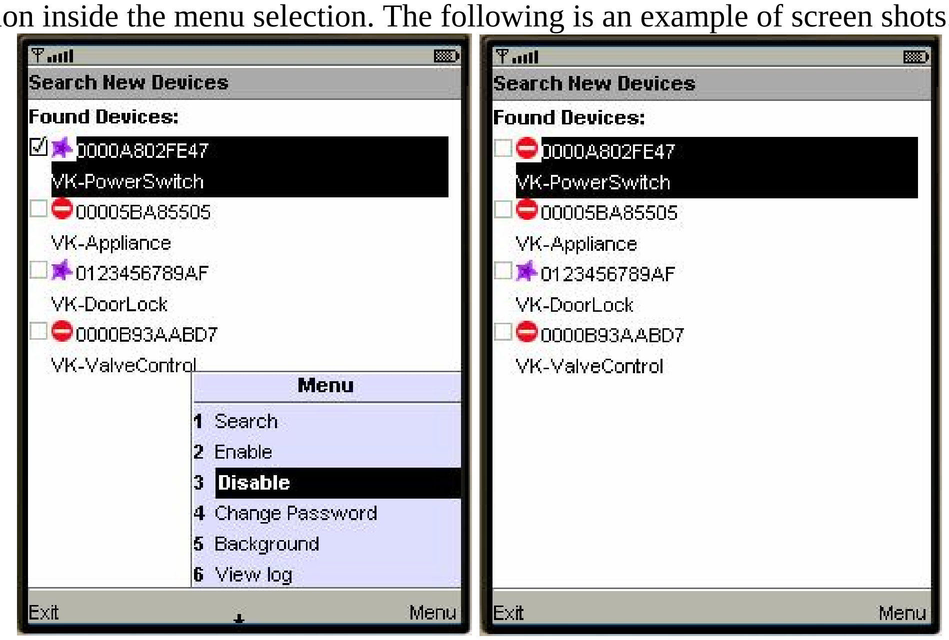

Figure 6: Screen Shot of Disabled Devices

6. Change Password

After the user successfully login, he can also choose to change the current password. After the user chooses “Change Password” inside the menu selection, the screen will be changed to change-password screen. The change-password screen contains two text fields for new password and new password confirmed. Inside the menu selection, there are two selections: “Cancel” and “Ok”. The user can successfully change the password only if two new passwords are matched. The following is an example of screen shots.

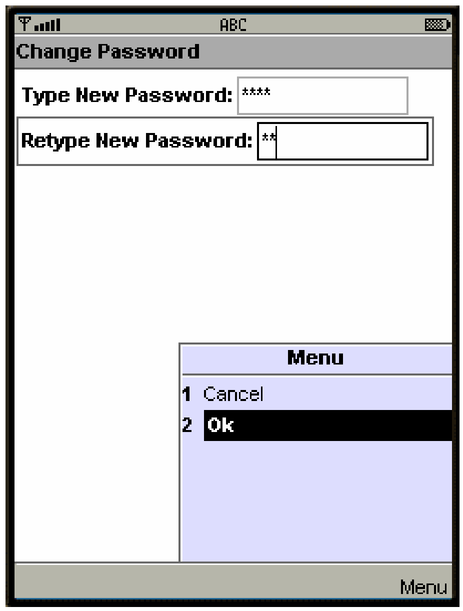

Figure 7: Screen Shot of Change Password

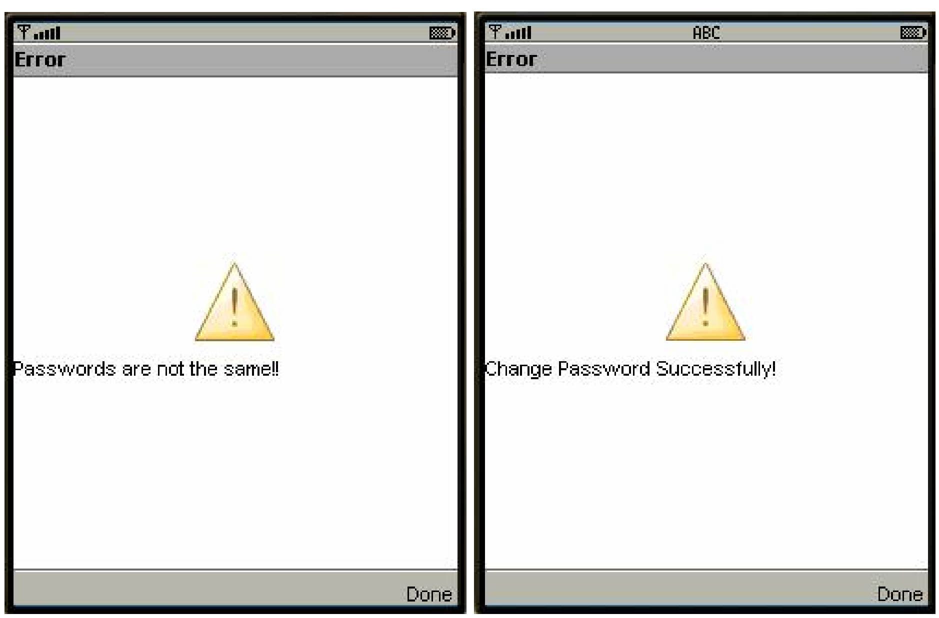

Figure 8: Screen Shot of Change Password Alert

7. **Run in Background**
   When the user executes this function, or when the user receives a phone call, the VirtualKey program will jump to the background and continue to run.

2.2 Hardware

We originally proposed two separate VirtualKey hardware modules, one for appliances and one for doors. However as the project progressed we combined the two modules into one. The current state contains a single VirtualKey Hardware Module that can be installed into any appliance or any door. In this project, we created a single VirtualKey Hardware module and for demonstration purposes, we implemented the module into a power bar. Figure 9, shows the VirtualKey Hardware Module implemented into a power bar.

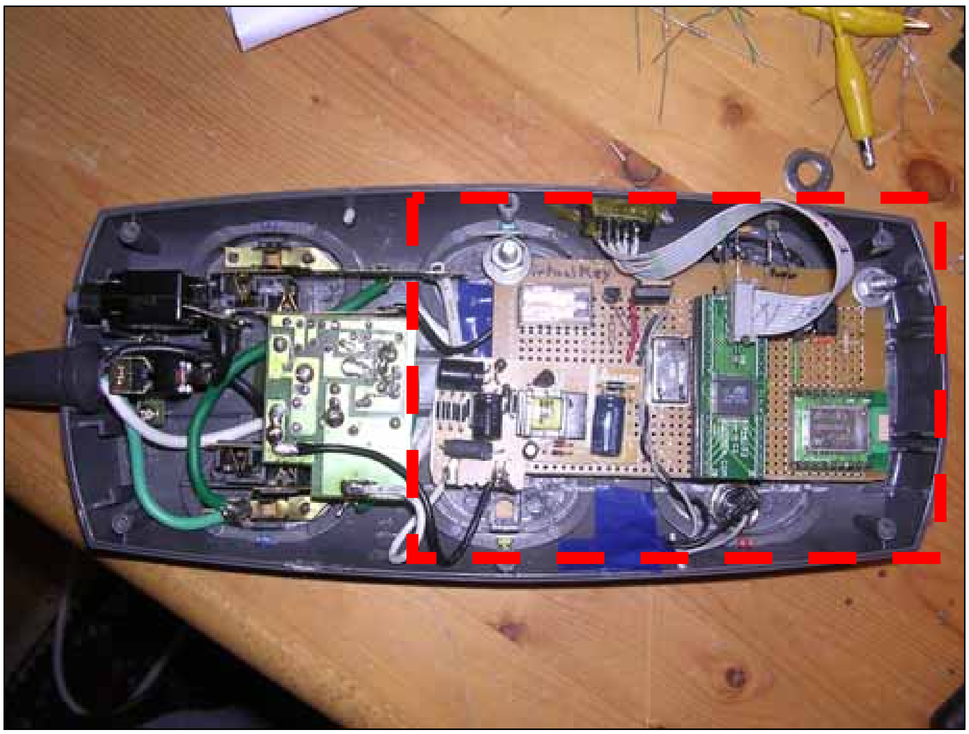

Figure 9: VirtualKey Hardware Module (shown inside dashed lines) integrated into a power bar

The same module can be used inside doors or any appliance with upgrading the relay to handle more current (as needed). Figure 10 show the final product after the VirtualKey Hardware Module is installed inside a 6-plug power bar. The VirtualKey Hardware Module also has a bypass key to enable the module in case of emergency. The bypass key is shown in Figure 10 as well.

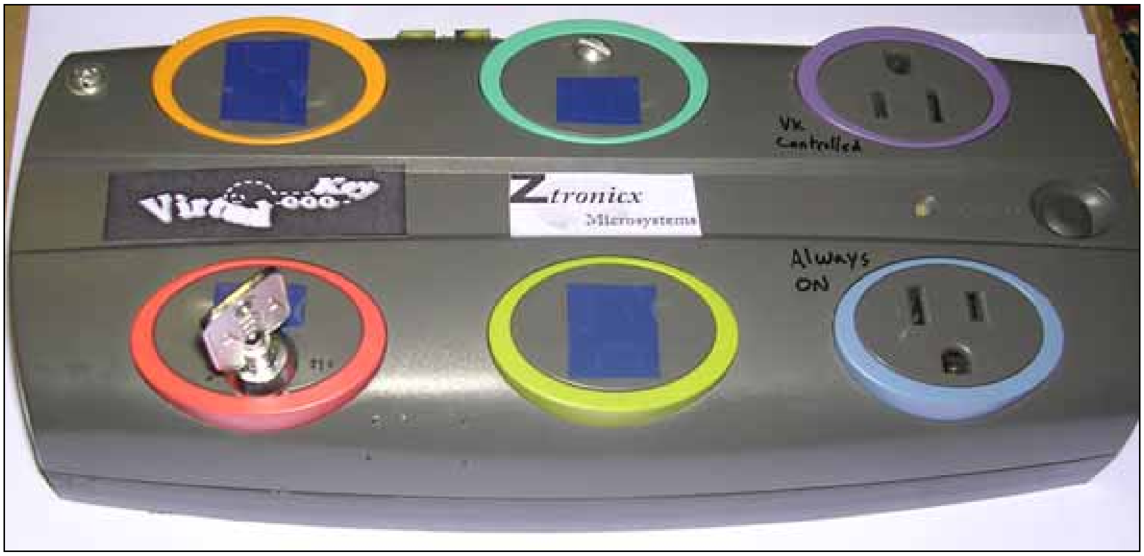

Figure 10: Power bar, after the full integration of VirtualKey Hardware Module as well as bypass key

Figure 11 shows a closer view of the final version of VirtualKey Hardware Module. The Microcontroller, Bluetooth module, power supply and relay are shown. This module can be integrated into doors or appliances.

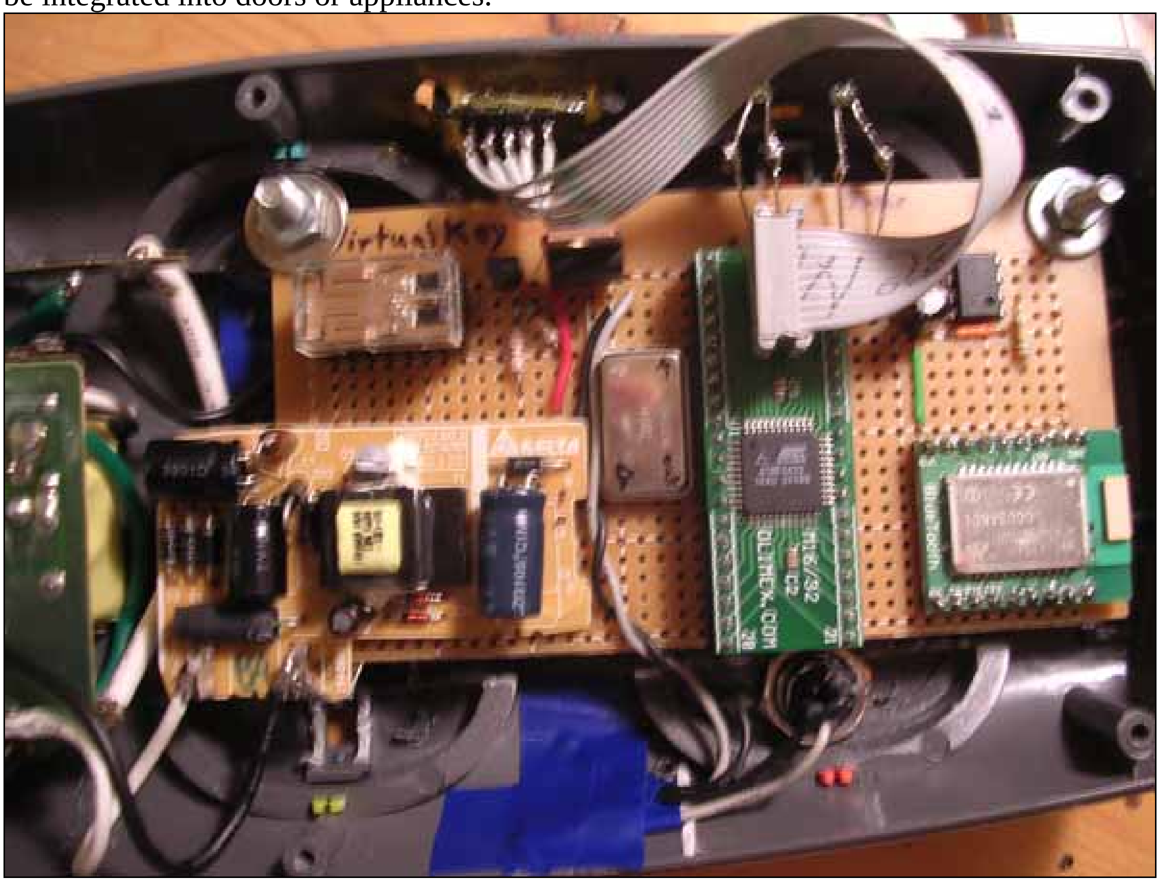

Figure 11: VirtualKey Hardware Module, showing the AVR Microcontroller, Bluetooth module, power supply and relay

3 Deviation of the Device

We originally proposed two separate VirtualKey Hardware Modules, but late in the semester we deviated a little. We combined the two VirtualKey Hardware Modules for doors and appliances into one unit and added a bypass key to the module for emergency situation. For software side, all functions are well implemented as proposed.

4 Budgetary and Time Constraints

4.1 Overall Budget Summary

Equipment	Estimated Cost	Actual Cost
2 Different Bluetooth Enabled Phone (Nokia and SonyEricsson)	$200	$200
3 Bluetooth Modules	$180	$100
2 Microcontroller Dev. Kits	$100	$150
Door Locks and Actuators	$120	$0
High Current Relays and Other Circuit Devices	$200	$50
Total	**$800**	**$500**

Table 1: Equipment Cost Summary

Sources of Funding	Estimated Funding	Actual Funding Received
Components From Fred Heep	$50	$50
Wighton Fund	$450	$0
ESSS Fund	$300	$400 (not confirmed yet as of now)
Total Funding Available	**$800**	**$450**

Table 2: Sources of Funding

The estimated and actual cost of producing a working model of our project is shown in Table 1. We made a small change to the project as the semester progressed, and we made one universal module that can work on door as well as appliances. Therefore, we didn't end up purchasing door lock and actuators since it was costly and unnecessary for demonstrating a prototype of VirtualKey system. High current relays were not used as well since we were able to use a regular relay to prove the concept, and high current relays can be added later on.

As for the sources of funding, we applied for ESSS Fund, and they mentioned we might be able to get around $400 in a few months. The amount is not confirmed, and we haven't received the funding yet as of Jan 10th, 2007. For Wighton Fund, we tried to find the application, but were not successful. Since we were promised some amount by ESSS, we didn't pursue Wighton Fund further.

4.2 Deviation from Estimated Timeline

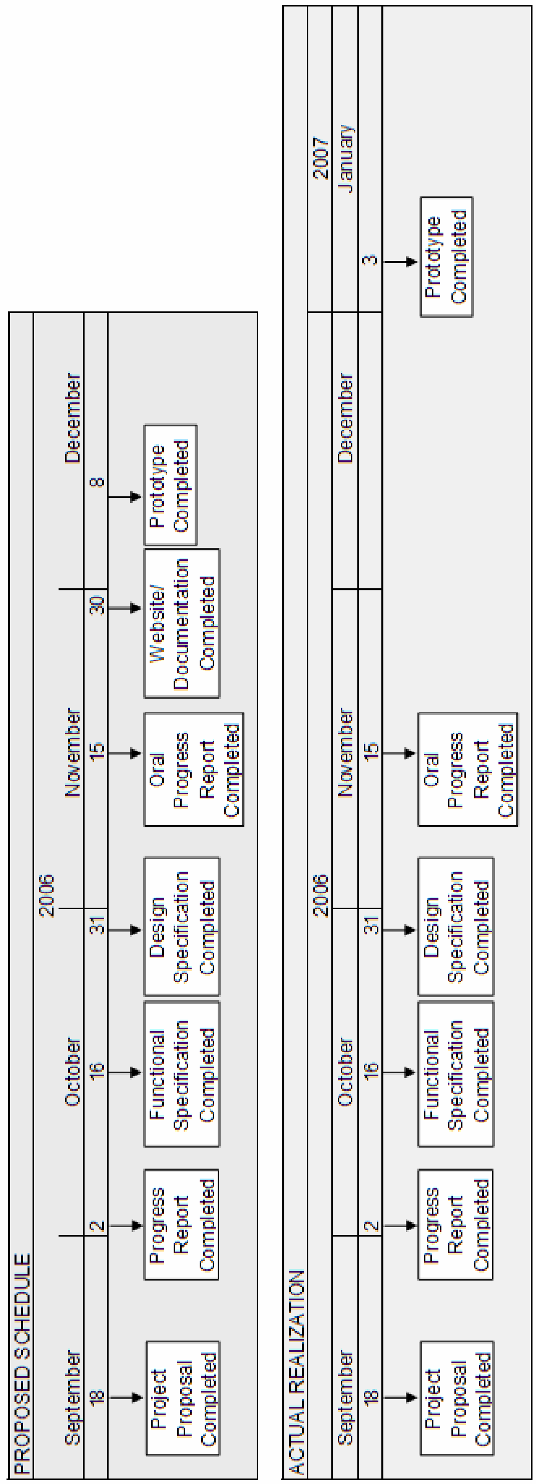

Figure 12: Comparison between Proposed Milestone Chart and Actual Realization

Our VirtualKey system steadily progresses from September to November according to our proposed timeline in Figure 4. Our actual time of starting to build the prototype is in November which is after the research and documentation of Design Specification. This has also changed our Gantt chart such that the whole implementation and testing phase got pushed back to November and through December. The final stage of having a completed prototype has been delayed from December to January, as illustrated in Actual Realization Milestone Chart in Figure 4, due to busy end-of-semester schedules.

5 Future Plans

5.1 Design Improvement of VirtualKey

The first version of VirtualKey system has established the bases of conceptual design model. We originally proposed a separate VirtualKey Hardware Module for doors and appliances, but we then combined them into one device for simplicity. In the future we might just change the firmware on the door or appliance modules for further improvement.

Software Future Development:

- Make the GUI more user-friendly
- Test to see how the system reacts if multiple actions are taken simultaneously
- Implement functions such that user can manually modify hardware parameters, for example, a friendly name for each device.

Hardware Future Development:

- Make Bluetooth module more reliable
- Remove unnecessary Bluetooth stack to bring costs down
- Decrease the size of VirtualKey hardware module
- Improve the power supply

5.2 Market and Commercialization

In order to bring VirtualKey into the market, we would first test its feasibility and usability on a group of individuals, for example, a family with children and elderly people especially those with Alzheimer or dementia disease. These groups of individual tend to be at high risk of inappropriate access to home appliances. For the general public, a secure door lock is beneficial and a cell phone to control all locks is preferred over traditional keys. Currently, the cost of VirtualKey hardware is $110 but if we mass produce it, the cost is estimated to be around $10 each. Also, since the system is simple to use, it will be accepted by the public easily. In addition, we would try to obtain external funding and acquire the patent to start our own company

6 Conclusion

VirtualKey is a cheap and effective way to control appliances and loved ones. Users can control the door locks and appliances with their cell phone. The current hardware cost of the demo model is $110 but if we mass produce the system, the estimated cost would be $10 per module.

For the past four months, we have endeavored to meet our goals within estimated timeline and budget. Everything went smoothly until the actual building of the system. In the software development of the cell phone, none of us have sufficient knowledge in establishing Bluetooth communication between cell phone and the hardware so the implementation phase has been slowed down a little. Also, as discussed in section 4.2 that we begin to build our model in November and that is usually the busiest time of the semester. This causes the delay in our first release of the VirtualKey system. Seeing the end working module, we are satisfied with our great effort to meet our proposed functions of the system.

Throughout the project design and development, we learned AVR Assembly Programming, AT command set, Bluetooth structure and protocol, J2ME programming, Encryption in Java and Bluetooth. These are valuable experiences that we would never learn in academic courses.

7 Interpersonal and Technical Experiences

Shahin Teymouri – CEO (Hardware Design for VirtualKey)

This project taught me many aspects I didn't expect. I learned all the way from hardware implementation and the problems that can occur to team work and documentation.

The hardware, as simple as it looked, when all the parts put together they refused to work as expected. Even though it was professionally assembled and many research was done, the hardware sometimes refused to work as we expected. The wireless system we used in this unit (Bluetooth) was extremely sensitive to other external noises, hence I spent days trying to figure out the problem.

We had good team work and all the members put exceptional effort in and I am very happy. However sometimes some minor conflicts happened and they were mostly due to the extra work from the other courses putting members of the group under pressure.

I never expected to do this amount of documentation for a project like this, but now I know why this is necessary and how it helps all the group member or external member on the same page.

This project was overall an excellent experience for me to come up with a solution to a well known problem and to design hardware from ground up. I also learned how to work with other members that are working on the software part. I learned what it takes to design a product and introduce it to the market, and what standards I have to follow and more importantly how to interact with users for further product improvements.

Halim Soetanto – CTO (Software Design & Development of VirtualKey)

I learned a lot from this project about cell phone application and Bluetooth™ development using Java™ technology. There are a lot of Java™ standards and APIs in order to develop various cell phone applications that work for virtually almost all of the cell phones using Java™ technology.

During the development, I learned that although cell phone emulator comes in handy to test the functionality of our code, I realized that we can't rely solely on emulators since the actual implementation of the application on a cell phone might be different than the emulators after several frustrating hours of tracing and debugging. The main reason I used the emulator a lot - instead of testing it on an actual cell phone - was because there was only one cell phone for our team and we need to take turns among the four members to test each of our code and the hardware on the actual cell phone that has the desired Bluetooth standard.

I gained valuable experience in developing serial communication, cell phone multi tasking, GUI design, and timer functionality. A lot of documentations, references, and tutorials exist in the internet to help me solve questions arose but not all of them were useful. So it took some time for me to sort out which ones were useful and I needed to go through all the references and read them one by one and from page to page before I finally got to the one I wanted.

Rick Liu – VP R&D Project Manager

Rick Liu is a 5th year Engineering Science student at SFU specializing in Electronics and Communication. He has work experience in the wireless industry where he attained valuable skills as a customer service engineer and computer technician. During his scholastic career he has worked on many successful engineering projects and has expertise in network analysis and communication signal decoding.

For this project course, one of the things that impress me is a lot of readings. For other regular courses, I might need to read only one textbook for about hundreds of pages or sometimes plus what professors assigned handouts. However, for this project, I have read at least thousands of pages whether they are journals, tutorials or books. Most of the time, I can't be sure that the document I'm reading has the information I want or not, even after briefly look through tables of contents.

Another experience I had during the development is that I won't know the result unless I actually did the work and tested it. Theoretical result doesn't always guarantee a successful result. A lot of works were try and errors. Moreover, the hardest part for the development is not to find a solution because there often has more than one solution to the problem. The hardest part is to find the most efficient solution among all the solutions. The efficient here means either time efficient, performance efficient, or even both.

For our team, I think there's only one thing need to be improved – the meeting. I feel the meeting time was too long for each of our team meeting. According to the meeting agenda for each meeting, the time might require only 10~15 minutes, but almost each time the meeting length increases up to half hour or an hour. However, one fortunate thing is that Shahin is a really good leader. He would push us to meet the deadline over the emails. I would like to thank him and rest of team members. Each one of them is able to fulfill their assigned tasks.

May (Mei-Yi) Liu – Software Developer & Creative Director

As the project progressed, I practiced my previous Graphical User Interface programming in C++ and basic Java programming skills to become familiar with J2ME programming in cell-phones. I participated in the software development of VirtualKey system specifically in GUI representation.

Throughout the implementation stage, I realized that generating GUI in Java is completely different in Microsoft Visual Studio C++. The concept might be similar but the structure is totally unrelated. I studied several J2ME documents especially on Midlet application and referenced to various samples but sometimes I am still not able to figure out the bugs. The major obstacle that I encountered was the transition between screens and accessing the variables from different screens. Often, a simple feature could be complicated to code; however, I find it easier to program a similar behavior in J2ME than in C++ where fine-tuning is sometimes required. In addition, I am also responsible for all the graphical contents of the system.

8 Reference

HowStuffWorks Inc., “How Bluetooth Works”,
http://electronics.howstuffworks.com/bluetooth1.htm, 2006
Bluetooth Special Interest Group (SIG), “Compare with Other Technologies”,
http://www.bluetooth.com/Bluetooth/Learn/Technology/Compare/, 2006
Palowireless Pty Ltd. (Palowireless Resource Center), “Bluetooth Tutorial – Specifications”
http://www.palowireless.com/infotooth/tutorial.asp, 2006
Wireless Net DesignLine, “The Wireless LAN Rush by Richard A. Quinnell”,
http://www.wirelessnetdesignline.com/showArticle.jhtml?articleID=192200581
Sun microsystems, “Developer Network - The Java APIs for Bluetooth Wireless Technology”,
http://developers.sun.com/techtopics/mobility/midp/articles/bluetooth2/,
Sun microsystems, “Developer Network - Using the Java APIs for Bluetooth”,
http://developers.sun.com/techtopics/mobility/apis/articles/bluetoothcore/index.html,
Sun microsystems, “Developer Network - Using the Java APIs for Bluetooth”,
http://developers.sun.com/techtopics/mobility/apis/articles/bluetoothcore/index.html,
Sun microsystems, “Developer Network - Using the Java APIs for Bluetooth”,
http://developers.sun.com/techtopics/mobility/apis/articles/bluetoothcore/index.html,
AVR RISC Microconttrollers, “ATmega16 Datasheet”, ATMEL Corporation, 2006
Olimex Ltd, ATmega16 Development Board Schematics, http://www.olimex.com/dev/images/avr-m16-sch.gif, 2006.
BlueRadios Inc, BR-C40, http://www.blueradios.com/BR-C40.pdf, 2006
Goodsky Relays, http://kitsrus.com/pdf/rwh_relay.pdf, 2006
Texas Instruments Inc, TPS7233 Voltage Regulator Description, http://www-s.ti.com/sc/ds/tps7233.pdf, 2006
Bruce Hopkins, Ranjith Antony, “Bluetooth for Java,” Berkeley, Calif.: Apress; London: Springer, 2003.
Ortiz, C. Enrique, “Mobile information device profile for Java 2 Micro Edition: professional developer's guide,” New York: Wiley, 2001
Sony Ericsson Mobile Communications, “Developing Applications with the Java APIs for Bluetooth (JSR-82),” http://developer.sonyericsson.com/site/global/home/p_home.jsp
NetBeans Community, “Docs & Support”, http://www.netbeans.org/kb/index.html