Turn Smartphones into Computer Remote Controllers

Yonggao Yang and Lin Li

***Abstract*—This article presents the design and implementation of turning smartphones into computer remote controllers that allow users to wirelessly operate a computer. Smartphones talk to a computer through either wireless Ad Hoc or Wi-Fi access point network. A computer user uses a smartphone as the mouse and keyboard of the computer and operates the computer remotely and wirelessly. Such a system has wide application in many environment settings, such as conference rooms, classrooms, workgroup project environments, and even people's living rooms, where the computer screen is projected onto a big screen, such as a projector screen or a big-screen television.**

***Index Terms*—Computer remote controller, human computer interface, smartphone, smart device.**

I. Introduction

There exist several situations where we want to wirelessly and comfortably operate a computer, where the computer screen is projected onto a big screen through a projector or big-screen television, such as classrooms, conference/meeting rooms, mobile, workgroup project environments and modern office environments, and even living rooms.

Several specifically designed devices are available on the market for the purpose of operating computers remotely and wirelessly. Wireless keyboard, as shown in Fig. 1a, uses either Bluetooth or wireless USB mini-receiver plugged into the USB port of computer for the communication between the keyboard and the computer. Some wireless keyboards have a touchpad for controlling the mouse cursor. Wireless presentation controller, as shown in Fig. 1b, allows user to operate his/her computer remotely for PowerPoint presentation through Bluetooth connection. It usually has several buttons, including mouse-left, mouse-right, next slide and previous slide buttons (for PPT presentation), and even a small size rectangle touchpad for moving mouse cursor.

(a) Wireless keyboard (b) Presentation controller

Fig. 1. Existing wireless computer controllers.

However, all those devices have certain drawbacks. Wireless keyboard has limited flexibility and is not convenient for a presenter to carry it around in the room during the presentation. Presenters usually like to walk around while presenting. Carrying a wireless keyboard is definitely not convenient.

Wireless presentation controller does have good mobility. However, most of such devices do not allow user to have full operation on the computer, such as running a program, moving or closing an application window, etc. Even it has a small touchpad for moving mouse cursor (as shown in Fig. 1b), however it is very difficult for the presenter to use it to move the mouse cursor while he/she is walking around.

On the other hand, a computer is allowed to have one and only one such device controlling it. If an audience wants to operate the computer to facilitate asking questions or provide comments during the presentation session, he or she must walk to the computer to use the wired mouse/keyboard, or must get the wireless keyboard or wireless presentation controller. This imposes inconvenience when a group of people, such as a project team or a company internal meeting attendees, are having discussion and need to interact with each other through the same computer with its screen projected onto a wall.

The widely used and very popular smart devices, such as iPads, smartphones, PDAs, and smart game controllers, can be the excellent alternatives as computer remote controllers if we develop appropriate apps for them. For example, Yang and Li [1] developed a system to use Wii Remote, Wiimote in short, to remotely operate a computer, where Wiimote communicates with the computer through Bluetooth. B. Koo etc. [2], proposed a framework of using smartphones to control consumer electronics. C. Wingrave and others [3] provided an excellent tutorial on how to use a smart device, Nintendo Wii Remote (Wiimote) in 3D User Interface systems. Generally speaking, smart devices [4] are digital, active, computer networked, and user reconfigurable devices that can operate to some extent autonomously. These devices usually have the many unique characteristics, such as

- Small physical size for excellent mobility. They are designed for handholding and carrying around. They are owned and operated by individuals.
- Good Human-Computer-Interface. They have a small touch screen, a build-in physical keyboard or soft keyboard, and run a simplified modern operating system with good graphics user interface.
- Strong network capability, such as Wi-Fi wireless and Bluetooth, allowing Internet access.

These devices are designed for multi-purposes, such as personal office, mobile phone, camera, games console, etc. The list of the most popular smart devices include iPad and iPhone from Apple Company, Android smartphone from Google, Windows mobile smartphone and PDA from Microsoft, and various handhold game systems from Nintendo, Microsoft, Sony, and other. Using these smart

Manuscript received June 2, 2012; revised July 1, 2012.

The authors are with the Department of Computer Science Prairie View A and M University Prairie View, Texas, USA (e-mail: e-mail: yoyang, linli@pvamu.edu).

devices is becoming part of our daily and routine life.

In this article, we discuss and present how to turn smartphones into computer remote controllers. The remaining of this article is organized as follows. Section II describes and presents the system design that turns a smartphone into a computer remote controller. Section III discusses the system implementation, more specifically the software implementation. In section IV, we outline the application of such system. Section V concludes this work.

II. System Design

The system presented below allows multiple smartphones to wirelessly connect to the same computer and users use their smartphone touch screen and keyboard to operate the computer.

A. Smartphones

A smartphone is a cellular telephone with built-in applications and Internet access. Smartphones may be thought of as handheld mobile computers integrated with a mobile telephone, and usually allow the user to install and run more advanced applications that are also called apps.

Smartphones run complete operating system software providing a platform for application developers. The major smartphone operating systems in use are Symbian OS, Android OS, Blackberry OS, iOS, and Windows Mobile OS. These system environments provide platforms for users to develop their own applications for the smartphones. Fig. 2 shows three smartphones running Windows Mobile OS, iOS, and Google Android OS, respectively.

Fig. 2. HTC, iPhone, and Android smartphones.

Smartphone has been widely used as a handheld device to control other devices/equipment [5-7]. In this section, we present the system design of turning a smartphone into a computer remote controller, which involves developing a smartphone application running on the smartphone and software running on the computer. The details are described below.

B. System Diagram and Architecture

A smartphone usually has built-in wireless network function (Wi-Fi) that allows it to connect to the Internet through the wireless access point (WAP). This Wi-Fi capability also allows a smartphone to talk to another smart device (such as a computer or another smartphone) through peer-to-peer Ad Hoc wireless connection or Wi-Fi access point. Using smartphone Wi-Fi capability and developing appropriate software, we can turn a smartphone into a computer remote controller and use it to operate the computer wirelessly. Fig. 3 outlines the structure of the smartphone-based computer remote controller, where the computer screen is projected onto a big-screen, either through a projector or a big-screen television.

The smartphone application, *SMTFController*, running on smartphones captures user’s keyboard and mouse events, and sends them to the computer in real-time through wireless Ad Hoc or Wi-Fi access point network.

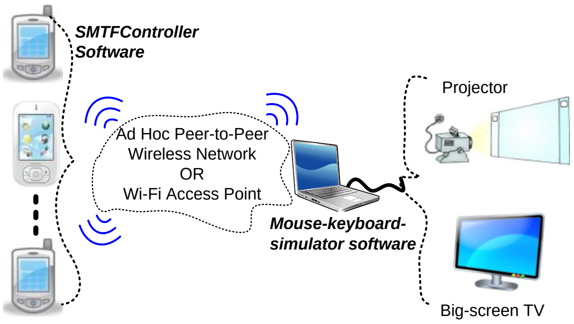

SMTFController Software
Ad Hoc Peer-to-Peer Wireless Network OR Wi-Fi Access Point
Projector
Mouse-keyboard-simulator software
Big-screen TV

Fig. 3. Smartphone-based controller architecture.

The “*Mouse-Keyboard-Simulator*” software running on the computer accepts keyboard and mouse events from the smartphones, and through the operating system running on the computer to simulate the keyboard and/or mouse events to control the computer, including program execution, keyboard input, and mouse actions. Those keyboard/mouse events received from smartphones are simulated in such a way that they seem to be generated by the local physical keyboard and mouse belonging to the computer.

The keyboard/mouse events from different smartphones carry different smartphone IDs, and the computer uses this ID information to decide whether or not it should mimic the received events based on the configuration at computer side. For example, if the “*Mouse-Keyboard-Simulator*” software is configured to allow multiple smartphones to operate the computer simultaneously, then the keyboard/mouse events received from all the smartphones will be simulated through the computer operating system. Otherwise, only the events from a specific smartphone that possesses the control right will be sent to operating system to operate the computer. Under this situation, a mechanism is applied to monitor the computer control right and manage this right transference among the multiple smartphones.

III. System Implementation

The implementation of the above system design involves three major software modules, which are the communication module handling the communication between smartphones and the computer, the smartphone-side software module (*SMTFController*), and the computer-side software module (*Mouse-Keyboard-Simulator*).

A. Communication between Smartphones and Computer

The smartphone-side program (*SMTFController*) talks to the computer-side program (*Mouse-Keyboard-Simulator*) through Ad Hoc peer-to-peer wireless connection or Wi-Fi access point. The communication between the two sides can be in one of the two phases described below.

Connection establishment phase: *SMTFController broadcasts* a connection-request UDP packet carrying its IP address and unique phone ID to the network. The reason for choosing to broadcast mode is that smartphone user does not need to know the computer IP address when he tries to connect his smartphone to the computer. Only the computer running the *Mouse-Keyboard-Simulator* will respond this request by sending an acknowledge UDP packet to the smartphone, carrying the computer IP address and other smartphone-needed information. As soon as the smartphone receives the ACK packet from the computer, the smartphone has the computer IP address and other necessary information. The two sides now know each other's IP address and port numbers, and since then the communication between them is switched from UDP broadcast mode to peer-to-peer UDP mode to avoid wasting network bandwidth, which is the next phase.

Regulation communication phase: during this phase, all the data traffic is transmitted through the peer-to-peer UDP mode. Most of the data traffic is from smartphone to the computer, carrying the key and mouse events, and mouse position change offset. This information is sent from smartphones to the computer in the best effort to achieve smooth operation of the computer.

B. Smartphone-Side Software Implementation: SMTFController

The smartphone-side software, *SMTFController*, is a typical smartphone application, which turns the smartphone keyboard into the computer keyboard, and its touch screen into the computer mouse touchpad, thus allows the user to operate the computer through the smartphone.

Fig. 4 is a snapshot of the *SMTFController*. The whole smartphone screen serves as the virtual touchpad of the computer. User uses his/her finger or stylus to move on the screen or tap the screen. The finger/stylus position and tap events are sent to the computer through peer-to-peer UDP packets in real-time, which are mimicked at the computer side to operate the computer. Since smartphone does not have mouse-left-button, a check-box, *MLBtn Down*, is provided to simulate "mouse-left-button down" event. For instance, user moves a window on the computer by moving his/her finger or stylus on the smartphone screen while the "`MLBtn Down`" box is checked. *SMTFController* can also recognize single-tap and double-tap actions, and send these events to computer to simulate mouse-left-button single or double click. The "keyboard edit area" is used for the user to use the smartphone keyboard to provide keyboard input to the computer. When a key is pressed on the smartphone keyboard, this key is displayed in this edit area, and also is immediately sent to the *Mouse-Keyboard-Simulator* at computer side for the computer to simulate that key stroke.

C. Computer-Side Software Implementation: Mouse-Keyboard-Simulator

The software running on the computer, *Mouse-Keyboard-Simulator*, receives mouse and keyboard events from smartphones, and simulates these events through operating system in a way that those events are treated as they were generated by the physical keyboard and mouse of the computer.

Whenever a keyboard or mouse event is received from a smartphone, it is immediately passed over to the operating system by calling keybd_event(), mouse_event(), or SendInput() functions, so that the operating system treats the event as the one generated by the physical keyboard or mouse of the computer. These events including mouse moving, mouse button single click and double click, key stroke, and so forth.

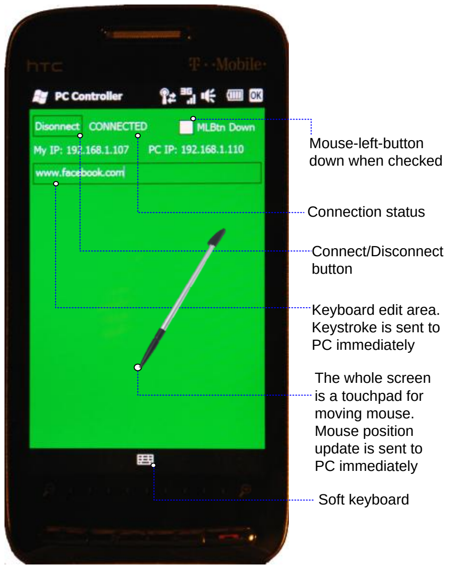

htc
T · Mobile
PC Controller
Disconnect
CONNECTED
MLBtn Down
My IP: 192.168.1.107
PC IP: 192.168.1.110
www.facebook.com
Mouse-left-button down when checked
Connection status
Connect/Disconnect button
Keyboard edit area. Keystroke is sent to PC immediately
The whole screen is a touchpad for moving mouse. Mouse position update is sent to PC immediately
Soft keyboard

Fig. 4. Snapshot of *SMTFController*.

For example, when a mouse-left-button-double-click event comes, the following C++ code snippet calls the operating system functions to simulate the mouse-left-button-double click:

```
mouse_event(MOUSEEVENTF_LEFTDOWN,0,0,0,0);
mouse_event(MOUSEEVENTF_LEFTUP,0,0,0,0);
mouse_event(MOUSEEVENTF_LEFTDOWN,0,0,0,0);
mouse_event(MOUSEEVENTF_LEFTUP,0,0,0,0);
```

The following code snippet simulates the mouse-move event, and moves the mouse cursor to the specific position on the screen（mousePos.*x*, mousePos.*y*）:

```
INPUT MouseClick;
MouseClick.type = INPUT_MOUSE;
MouseClick.mi.dx = deviceRatioX * mousePos.x;
MouseClick.mi.dy = deviceRatioY * mousePos.y;
MouseClick.mi.mouseData = 0;
MouseClick.mi.dwFlags =
                              MOUSEEVENTF_ABSOL
                              UTE |
                              MOUSEEVENTF_MOVE;
MouseClick.mi.time = 0;
MouseClick.mi.dwExtraInfo = (DWORD)0;
SendInput(1,andMouseClick, sizeof(INPUT));
```

When user moves is finger or stylus on the smartphone touch screen, the offset of the finger/stylus position change is sent to *Mouse-Keyboard-Simulator* module at the computer side as a mouse-move-event. *Mouse-Keyboard-Simulator* module uses the above code snippet to move the computer mouse cursor to its new location. This is done in real-time, so that the user sees the computer mouse moving smoothly following its finger/stylus move on the smartphone touch screen.

As for the keystroke simulation, the following two-line code snippet informs operating system that a key with the converted value stored in *n*Char variable is pressed and released. Whenever user hits a key on the smartphone, that key event is sent to *Mouse-Keyboard-Simulator* module, and it uses the following code snippet to generate the same keystroke on the computer.

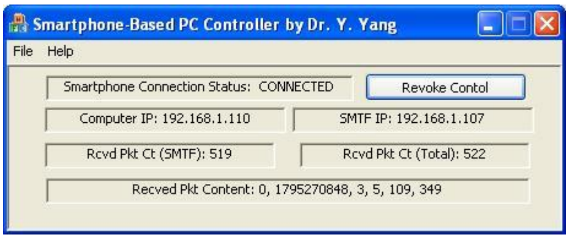
```
    keybd_event(nChar, 0×45,
                KEYEVENTF_EXTENDEDKEY | 0, 0);
    keybd_event(nChar, 0×45,
                KEYEVENTF_EXTENDEDKEY |
                KEYEVENTF_KEYUP, 0);
```

Mouse-Keyboard-Simulator was developed using Microsoft Visual Studio 2008. It has successfully passed the test on MS Windows XP, Vista, and Windows 7. It is usually minimized to the system tray at the right-bottom corner of the computer screen. However, user can click its icon to bring it up as shown in Fig. 5. The "Revoke Control" button is used to close the connections between the computer and all the smartphones. Other information displayed is for monitoring purpose only.

IV. Application

This smartphone-based system can find a wide application in several environment settings.

Classrooms: Instructor has the full mobility in classroom while teaching with a computer. The instructor will be allowed to be away from the instructor computer and still has full control of the instructor computer. Instructor can use his/her smartphone to do PPT presentation, execute program, and have free-hand writing, etc., just like the instructor has the physical mouse and keyboard following him/her (Fig. 6).


Smartphone-Based PC Controller by Dr. Y. Yang

Smartphone Connection Status: CONNECTED

Computer IP: 192.168.1.110

SMTF IP: 192.168.1.107

Rcvd Pkt Ct (SMTF): 519

Rcvd Pkt Ct (Total): 522

Recved Pkt Content: 0, 1795270848, 3, 5, 109, 349

Fig. 5. Snapshot of *Mouse-Keyboard-Simulator*.

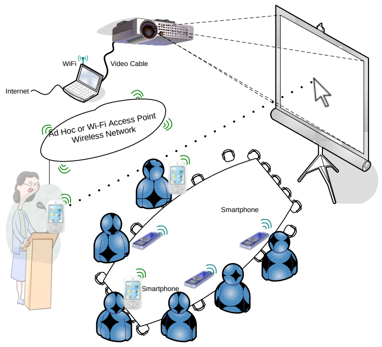


Fig. 6. The typical environment setting of using smartphone-based computer controller system.

Meeting/conference rooms: Meeting attendees can operate the computer for discussion/presentation/free-hand-writing-drawing without leaving their seats. There is no limit on the number of smartphones concurrently operating the same computer in the smartphone-based system. Imagine the scenario that whenever a meeting attendee wants to voice his opinion through the computer, he just needs to pull out his smartphone, run the smartphone-side app, *SMTFController*, connect his smartphone to the computer, and immediately he can operate the computer while comfortably sitting in his chair.

Resident living room: with the big-screen television connected to a computer serving as the monitor, we can sit cozily on the couch to operate the computer using a smartphone, without crouching in front of the computer.

Value-added system to projector products: This system can be bound to projector products and facilitate users to use their computers and projectors.

V. Conclusion

This article describes how to turn smart devices, more specifically smartphones, into computer remote controllers. The system presented above can be widely used in classrooms and meeting/conference rooms for presentation and interactive discussion. It also allows multiple users to use multiple smartphones to operate the same computer, facilitating group discussion or classroom interaction. Currently we are exploring approaches of using smart devices as controllers or operators for other devices.

References

[1] Y. Yang and L. Li, "Turn Nintendo Wiimote into Handheld Computer Mouse, " *IEEE Potentials*, vol. 30, no. 1, pp. 12-16, Jan/Feb. 2011.

[2] B. Koo, T. Ahn, and others, "R-URC: RF4CD-based Universal Remote Control Framework using Smartphone," in *proc. IEEE Conf. Computational Science and Its Applications*, 2010, pp. 311-314.

[3] C. Wingrave, B. Williamson, and others, "The Wiimote and Beyond: Spatially Conveient Devices for 3D User Interfaces," *IEEE Computer Graphics and Applications*, vol. 30, no. 2, 2010, pp. 71-85.

[4] S. Poslad, "Ubiquitous Computing: Smart Devices, Environments and Interactions," Wiley, 1st ed, 2009.

[5] C. M. Spelta, V. Corti, A. Goggi, and A. Savaresi, "Smartphone-Based Vehicle-to-Drive/Environment Interaction System for Motorcycles," *IEEE Embedded Systems Letters*, vol. 2, no. 2, 2010, pp. 39.

[6] V. Nguyen, S. Hao, and J. Szajman, "WiiKey: An Innovative Smartphone Based Wi-Fi Application," in *proc. IEEE Conf. International Multisymposiums on Computer and Computational Science (IMSCCS'08)*, 2008, pp. 91-97.

[7] A. Hammershoj, A. Sapuppo, and R. Tadayoni, "Challenges for Mobile Application Development," in *proc. IEEE Conf. 14th International Conf. on Intelligence in Next Genearation Networks IICIN'10)*, 2010, pp. 1-8.

Dr. Yonggao Yang is an associate professor and the Interim Department Head in the Department of Computer Science at Prairie View AandM University (PVAMU) in Texas. He joined CS Department at PVAMU as an Assistant Professor in August 2002 immediately upon receiving a Ph.D. degree in Information Technology from George Mason University (Fairfax, VA, USA). He worked as a faculty member at Southwest Jiaotong University during 1987 through 1998. He is a member of IEEE. His research interests include computer graphics, scientific visualization, virtual reality, and smart HCI.

Dr. Lin Li is an assistant professor in the Department of Computer Science at Prairie View AandM University (PVAMU) in Texas. His research interests include computer networks, smart energy system, and engineering education. He joined CS Department at PVAMU as an Assistant Professor in August 2006 upon receiving a Ph.D. degree from University of Nebraska Lincoln (Lincoln, NE, USA).