PC Control System Using a Smart Phone


* Department of Computer Engineering, Hallym University, Korea
** Department of Ubiquitous Computing, Hallym University, Korea
E-mail: thdwngh@hallym.ac.kr, kaizerad@nate.com, mzon1@naver.com, youngwoong@hallym.ac.kr, sunkim@hallym.ac.kr, cgsong@hallym.ac.kr

Abstract

In this paper we propose a way to control personal computer or tablet computer using a smart phone such as iPhone or Android phone. Any additional hardware is not required in our method but new interface is designed on the screen. The method is suitable for controlling media player like a Window Media Player. Our interface can replace a joystick when we play a game.

Keywords: Remote, PC Control, Smart Phone, Mobile

1. Introduction

A smartphone is a mobile phone that offers more advanced computing ability and connectivity than a contemporary feature phone. A smartphone allows the user to download application programs that are native to the underlying hardware and add or delete them freely. Smartphones run a complete operating system software providing a platform for application developers. Thus, they provide a camera phone, touch screen, GPS, accelerometer, a keyboard and Wi-Fi facility. It is a cellular telephone with built-in applications and Internet access.

Smart phones provide digital voice service as well as text messaging, e-mail, Web browsing, still and video cameras, MP3 player, video viewing, and often video calling. In addition to their built-in functions, smartphones can run a myriad applications, turning the once single-minded cell phone into a mobile computer.

In this work we propose a way to control application programs on PC remotely. The application programs include a media player such as a Windows Media Player and GOM player and the Microsoft Powerpoint also. Through Wi-Fi facility the smartphone controls programs on the PC remotely. The interface to control programs is designed and implemented so that the user can use it easily.

2. Related Works

There have been many advances in controlling PC using a smartphone. The Remote Desktop of MochaSoft and TeamViewer are examples of iPhone applications to control PC remotely. Figure 1 shows a screen shot of TeamViewer. The Touch Mouse of Logitech enables users to control the mouse cursor on a remote server. All these applications are used to control the whole screen. These are not suitable to control a program on a desktop; rather, they are used to control the whole screen.

Figure 1: Screen shot of TeamViewer

Figure 2: GOM Remote Controller

The GOM remote controller in Figure 2 from TG Sambo is designed to control a GOM player remotely. It enables users to control a GOM player easily. But we have to buy extra hardware. And we cannot make additional functions.
3. System Design

Figure 3 shows the system configuration and the flow of data from client to server. Smartphone acts as a client and gets the data and transmits it to PC as a server. The transmitted data is the information about which button is pushed down. As soon as the server receives the data from the client, it sends a message to the corresponding program such as a media player. The message is actually a command to control the program.

![Figure 3: System Architecture](image)

4. System Implementation

4.1 Server

Figure 4 shows the interface of the server. Server gets the data from the client and sends the proper message determined by the data to the application program. The activated application program on the server is selected by choosing a radio button on the right upper corner of the interface. Now only three application programs such as GOM player, GOM audio, and MS Powerpoint can be controlled by a smartphone.

When a client connects with a server successfully, the IP address is displayed in the edit box as shown in Figure 4. In order to help debug a program the useful information is displayed in Debug edit box.

We have used two WinAPI functions such as FindWindow() and PostMessage() to implement the control program on the server. FindWindow() accepts the title name and the class name of the window as arguments and then returns a handle of the window. We have used Microsoft Spy++ to obtain the title name and the class name of the controlled window.

![Figure 5: Microsoft Spy++]()

When we want to get a handle of the window, it is better to use the class name rather than to use the title name since the title name often changes. Once you have a handle of the window, you may call the message by using functions like SendMessage() and PostMessage(). In our work we have used PostMessage() rather than SendMessage(). It takes less time to process the message queue when we use PostMessage() rather than SendMessage().

4.2 Client

Figure 6 shows the snapshot of client program. Before controlling the program, the client should connect with the server through Wi-Fi. The IP address is shown on the window of the client. The middle of figure 6 shows the

![Figure 4: Client Application](image)
input window in which the user types in the IP address used. The right of figure 6 shows the window of the client when the client makes a successful connection with the server. The play/pause button, forward and fast forward, backward and fast backward button and the exit button are shown. The button to maximize the window of the active program on the server is also shown on the right upper part in figure 6.

Figure 6: Client Program – Main window (left), IP Address input window (center), control window (right)

5. Conclusion

In this work we develop an app running on a smartphone which controls the program on the PC remotely. Figure 7 shows two snapshots of an app program on the smartphone and application program on the PC. It is noted that our interface does not require any additional hardware and we can add additional function to it easily.

Now TCP/IP is used to communicate between the client and the server. It is necessary but time consuming to type an IP address in when we try to connect a smartphone to the server. In future work we would like to improve this procedure in such a manner that the user can select or choose one from available IP addresses. We might keep track of previously used IP addresses. And we may use Bluetooth facility for data communication instead of Wi-Fi facility because our work space is short ranged.

Figure 7: The snapshot of the execution.

Acknowledgement

This research was financially supported by the Ministry of Education, Science Technology (MEST) and National Research Foundation of Korea(NRF) through the Human Resource Training Project for Regional Innovation

6. References


