Contemporary Engineering Sciences, Vol. 7, 2014, no. 14, 683 - 689
HIKARI Ltd, www.m-hikari.com
http://dx.doi.org/10.12988/ces.2014.4675

Smartphone as a Remote Control Proxy in Automotive Navigation System

Yejin Shin and Soonuk Seol

Department of Electrical, Electronics & Communication Engineering,
Korea University of Technology & Education, Korea

Abstract

In this paper, we design and implement a novel system where a remote person can control the car navigation system on behalf of the driver. In the proposed system, we suggest to utilize the driver's smartphone for allowing sharing the Internet connection of the phone with the navigation system. The required smartphone application and the server are also designed and implemented. In addition, we propose a common framework for the remote control of any embedded devices.

Keywords: Car Navigation System, Remote Control, Smartphone, Application

1 Introduction

Manipulating car navigation system while driving has been one of main reasons of car accidents. A new Road Traffic Law enacted recently in South Korea prohibits drivers from watching or operating any video displaying devices. A driver must stop the car and then can manipulate the navigation system unless there is a fellow passenger. In this paper, we design and implement a novel system where a remote person can control the car navigation system on behalf of the driver. The rest of the paper is organized as follows. Section 2 introduces related work. Section 3 describes our proposed design and its implementation. Section 4 explains a common framework proposed for remote control of any embedded devices. Finally, we conclude the paper in Section 5.

2 Remote Control Technologies

A remote control is a technique or a component for operating an electronics device typically wirelessly. Common applications include a television set, a DVD player, an electric light, and a gate control. Another type of remote control includes a remote diagnostic service. Recently, many people also utilize remote control software to access their own PC in a distance by using their smartphones.

There are different combinations in remote control of PC and smartphone. PC-to-PC is used for several purposes such as accessing documents at home PC, sharing contents, help desks service for technical troubleshooting of the customers' problems. Remote Desktop [1] and TeamViewer [2] are typical software for this. Smartphone applications such as TeamViewer [2] and ZOOK [3] are also widely used for controlling a remote PC. Another type of remote control is to control a smartphone at a PC. Android Screen Monitor (ASM) [4] receives frame buffer continuously on the device and transfer image to the desktop window. Unlike ASM, Droid VNC Server [5] allows a user to control an android device in a remote computer through the Internet possibly in a wireless manner. However, it has a disadvantage in that the device must be rooted.

A device should be connected to the Internet so that a remote controller can find the device and control it. In many cases, however, devices have no Internet connection. Therefore, a local area network (LAN) or a personal area network (PAN) is built first through the technologies such as Ethernet, Wi-Fi, Bluetooth, ZigBee, NFC, and so on. A simple approach shown in [6] is to use Wi-Fi module and the module is connected to the device via a serial cable. A computer server located in the Wi-Fi LAN can control the device. A drawback of this system is that only a dedicated server can control the device. Even though the controlled device would be run as a server mode, it is difficult to be located from the Internet due to an NAT issue, so the AP is configured with a port forwarding function [7]. The limited coverage and mobility in the Wi-Fi network is another drawback. For instance, in case of car navigation system, Wi-Fi cannot be used due to the movement of the device and the lack of seamless Internet connectivity. As another approach to NAT issue, Remote Control Lock System [8] employs a management server where both locking units and smartphone are connected as clients. However, a mobility issue still remains. One may consider using 3G or LTE module for the device's seamless Internet connection, but a monthly rate is a huge obstacle. In this paper, we adapt a similar approach with [8], but our system differs from it in that the wireless-to-wireless connection is applied.

With respect to screen sharing, there are two major approaches to transfer screen information from one device to another via network connections. One approach is to use pixel information or graphic primitives [9,10,11] and the other is to employ text descriptions of widgets [12,13,14]. Issues of the first approach include high bandwidth and responsiveness requirements, and an optimal scaling for mobile devices. This paper adapts the second approach. We focus on text messages exchange between devices. Data transmitted at each moment is called state information as they describe a pending state and possible events from buttons, lists, input forms, and so on.

3 Design and Implementation of Remote Control System

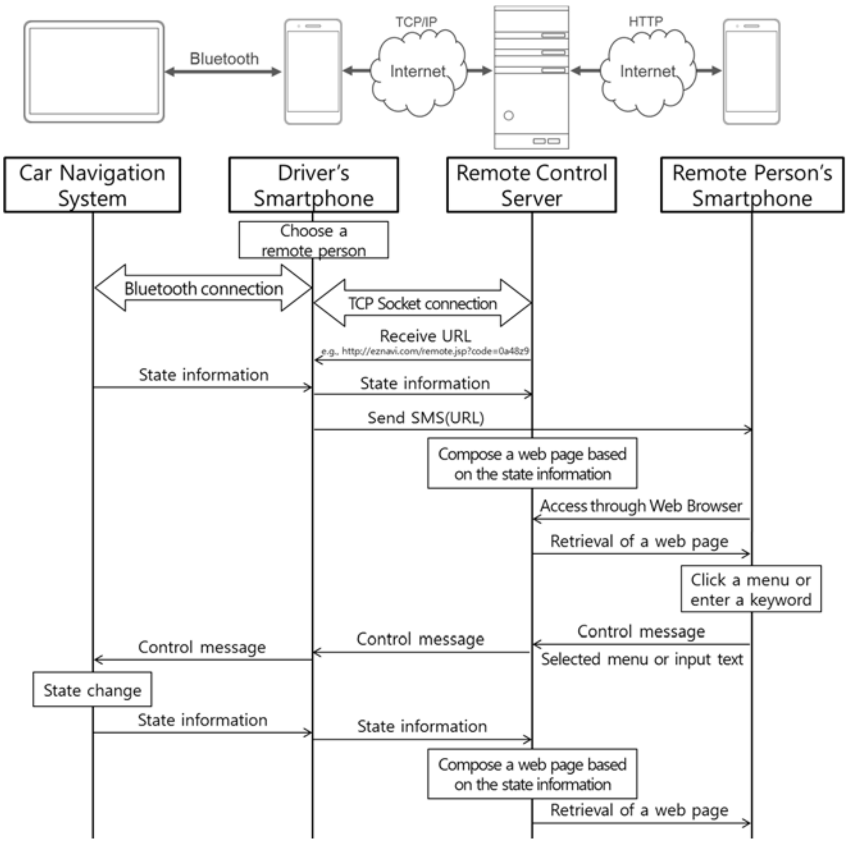

Figure 1. The procedure of remote control for car navigation system

Figure 1 shows architecture and procedure of the proposed remote control system for car navigation system. Our system consists of a car navigation device, driver's smartphone, remote control server, and remote control client. In driver's smartphone, there is an application which maintains a Bluetooth connection with the car navigation system and a TCP connection to the remote control server. After establishing all connections, the application on the driver's smartphone receives an access URL from the server and sends it to a remote person via SMS message. The URL is a one-time link that is valid within this session. Thus, only the designated remote person can control the device at a moment. The application also receives state information from car navigation system and forwards it to the remote control server. Then, the server composes a dynamic web page based on state information. The remote person can then access the web page by just clicking the link given in the SMS message. The selection or text input made in the web page is delivered all the way back to the car navigation system as a control message and it will change the state of the system. This state change will refresh the web page so that the remote person can make the subsequent controls.

By using a smartphone application, a car driver can invite a remote person, e.g., his son or a person who knows a best route. The driver's smartphone works as a proxy for a car navigation system. In order to allow any devices with a web browser to control a device in a distance, we choose HTTP as a communication protocol between the server and the remote devices. However, we employ TCP socket communication between the server and driver's smartphone to provide bidirectional active communication over a persistent connection between them. Thus, there are two communication components in the server. A web server part is implemented based on Java Server Page (JSP) and takes advantage of JQueryMobile to support various mobile terminals. A TCP socket part is implemented as JavaBeans and it works in conjunction with JSP. The web pages also utilize Asynchronous JavaScript and XML (Ajax) to update the web pages upon receiving next state information. Figure 2 shows some web pages displayed in controller's web browser.

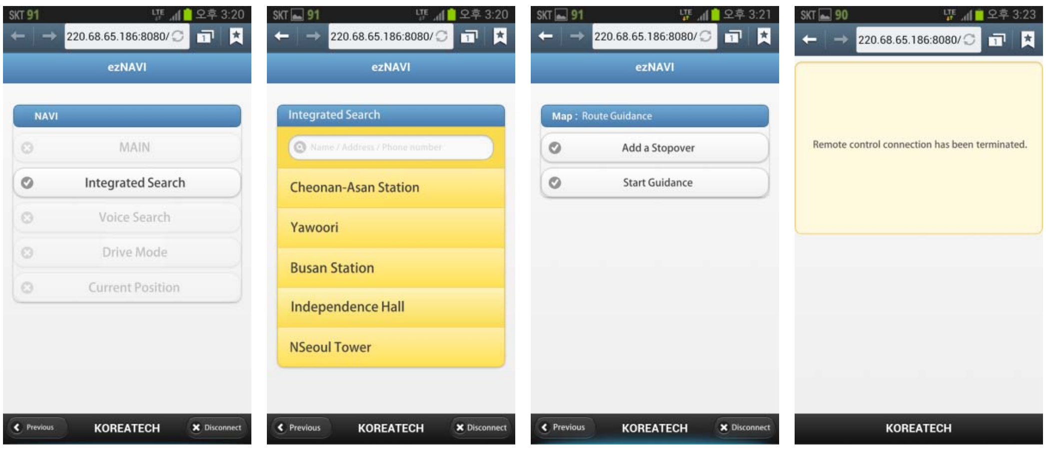

Figure 2. Screenshots of controller's web browser

Note that we do not transfer whole bitmaps of the original screen at the car navigation system. Instead, we interprets each different state of the system and send state information as widget descriptions to the web server so that the server can represent them with buttons, text fields, a list of hyperlinks, and input fields. The remote controller's input, i.e., either a menu selection or text input, becomes a control message and is delivered to the car navigation device.

4 Remote Control Framework for Embedded Devices

In this section, we further extend our design to be a common framework for any embedded devices. By using our proposed remote control APIs, embedded devices with no Internet connection can easily equip a remote control feature. This will easily realize the help desks for technical troubleshooting of embedded devices. Figure 3 depicts our proposed framework. The existing embedded software will utilize the remote control APIs to send state information of the device and to react to the remote control messages received.

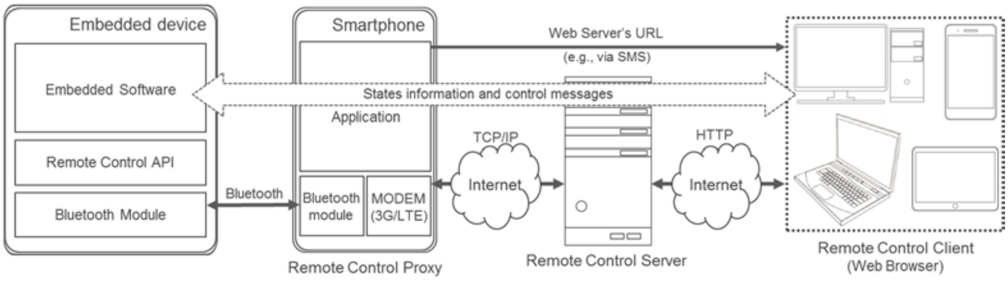

Figure 3. Remote control framework for embedded devices

API usage is shown in Figure 4 for stopwatch example. When there is a state where user's input is required, the programmer just add a short code for transmitting the state information as a form of menu list by using set_state("START", "STOP") function. After setting the state information, we can enable remote control by calling start_remote_control(). When the Bluetooth module receives a remote control message, an Interrupt occurs and will call rc_read() function. Consequently, we get the control message, either 0 or 1 in this example, and will call the corresponding function, doFn1() or doFn2().

```
#include<avr/io.h>
#include<avr/interrupt.h>
#include"rclib.h"

int MENU_SELECTED;
...

int main(){
    ...
  set_state("START", "STOP");
  start_remote_control();

  while(1){
      ...
  } return 0;
}
```

```
void *rc_read(){
      MENU_SELECTED = read_input();
      switch(MENU_SELECTED){
         case 0 :  doFn1();  break;
         case 1 :  doFn2();  break;
      }
}

void doFn1(){
   if(Time_STOP == 0) Time_STOP = 1;
   else               Time_STOP = 0;
}

void doFn2(){
   time = 0;
}
```

Figure 4. A simple example of the operation of the proposed system

5 Conclusion

In this paper we have designed and implemented a novel system where a remote person can control the car navigation system on behalf of the driver. This will be useful for beginner drivers, disabled people, or elders as a remote person may know better directions or is good at using the navigation system. Also, it can reduce car accidents and provide drivers with an easier and safer way of manipulating the navigation system. Our system has several advantages over the existing approaches with respect to remote controls. First, it can be applicable when there is no Internet connectivity since we utilize driver's smartphone as remote control proxy by sharing the Internet connection of the phone. Thus, it also allows mobility of the embedded system while it can be seamlessly controlled by a remote device. Another advantage is that a remote person can use smartphone, laptop, PC, or any other devices with a web browser. Our method also reveals a solution for screen sharing for smartphone-to-smartphone, for a limited bandwidth environment, and desirably for devices with low computing power. We have shown that text based state descriptions is feasible with a real implementation for remote control of car navigation system. In addition, we have also proposed a common framework together with APIs for remote control of any embedded devices. We have shown that easy and fast extension to the existing embedded software can be realized. So, new business opportunities may arise such as help desks for embedded devices. In the future, we plan to apply the proposed common framework to a real embedded device and conduct a performance analysis.

References

[1] Remote Desktop, www.microsoft.com
[2] TeamViewer, www.teamviewer.com
[3] ZOOK, www.zook.co.kr
[4] ASM, https://code.google.com/p/android-screen-monitor
[5] Droid VNC Server, https://github.com/oNaiPs/droid-VNC- server
[6] Hyung-Su Kim, Young-Hwan Han, Remote Control Device Using Wi-Fi Module, In Proceeding of the 2013 Korean Institute of Information Technology Summer Conference, (2013), 389 - 393
[7] S-C. Lo, T-H. Yu, C-C. Tseng, A Remote Control and Media-Sharing System Using Smart Devices, Journal of Systems Architecture, (2014).
[8] S. Park, C. Kim, H. Park, Y. Kim, Implementation of Remote Control Lock System By Combinational Design Method of NFC and UHF Communication, Journal of Korean Institute of Information Technology, **10** (2012), 11 - 18.
[9] T. Richardson, Q. Stafford-Fraser, K. Wood, and A. Hopper, Virtual network computing, IEEE Internet Computing, **2** (2002), 33 - 38.

[10] ITU-T Rec. T.128, Multipoint application sharing, (2008).
[11] R. Bose, J. Brakensiek, and K. Y. Park, Terminal mode: transforming mobile devices into automotive application platforms, Proc. 2nd International Conference on Automotive User Interfaces and Interactive Vehicular Applications, (2010), 148 - 155.
[12] R. G. Wendorft, R. T. Udink, and M. P. Bodlaender, Remote execution of HAVi applications on Internet-enabled devices, IEEE Transactions on Consumer Electronics, **47** (2001), 485 - 495.
[13] R. Verhoeven and W. Dees, Defining services for mobile terminals using remote user interfaces, IEEE Transactions on Consumer Electronics, **50** (2004), 535 - 542.
[14] D. Thommes, A. Gerlicher, Q. Wang and C. Grecos, RemoteUI: A high-performance remote user interface system for mobile consumer electronic devices, IEEE Transactions on Consumer Electronics, **58**(2012), 1094- 1102.

Received: May 1, 2014