Bluetooth를 이용한 동영상 전송 시스템 구현

조 경 연, 이 승 은, 최 종 찬
전자부품연구원 시스템IC연구센터
전화 : 031-610-4054 / 핸드폰 : 011-9037-1880

Implementation Of Moving Picture Transfer System Using Bluetooth

Kyoung-Youn Cho, Seung-Eun Lee, J Chan Choi
System IC Research Center, Korea Electronics Technology Institute
E-mail : choky@nuri.keti.re.kr

Abstract

In this paper we implement moving picture transfer system using bluetooth Development Kit (DK). To reduce the size of the image data, we use M-JPEG compression. We use bluetooth Synchronous Connection-Oriented (SCO) link to transfer voice data. Server receive image data from camera and compress the image data in M-JPEG format, and then transmit the image data to client using bluetooth Asynchronous connection-less (ACL) link. Client receive image data from bluetooth ACL link and decode the compressed image and then display the image to screen. Sever and Client can transmit and receive voice data simultaneously using bluetooth SCO link. In this paper bluetooth HCI commands and events generated by host controller to return the results of HCI commands are explained and the flow of bluetooth connection procedure is presented.

Ⅰ. Introduction

Bluetooth is one of the most famous technologies for short range communication between personal computer and potable devices in home or office environment. Bluetooth uses Industrial, Scientific, and Medical (ISM) frequency band at 2.4 GHz which is license-free and available globally. The frequency band ranges from 2.4 to 2.4835 GHz[1]. Bluetooth channels use a frequency hopping (FH) time division duplex(TDD). To cope with unpredictable sources of interference, bluetooth radios use FH spread spectrum. Although there exist IrDA technology for short range communication, IrDA has a limited range (1-2m) and require direct line-of-sight and it can communicate between only two devices[2]. In case of bluetooth, the communication range is 10-100m and communication is possible without direct line-of-sight. And 8 bluetooth devices can compose piconet. In this paper we implement bluetooth application which transfer moving picture and voice between a server and a client. In Section Ⅱ, we present bluetooth HCI command, and in Section Ⅲ, we present the transfer method of image data. In Section Ⅳ, the system environment used in this paper is presented. The GUI implementation is given in Section Ⅴ and Section Ⅵ concludes this paper.

Ⅱ. Bluetooth HCI command

Bluetooth host controller interface (HCI) is

common interface between bluetooth host and bluetooth hardware. In general, USB, PCMCIA, RS232 interface is used as HCI. When HCI command is transferred from host to bluetooth hardware, the bluetooth hardware executes the command and returns the result of the command to host. The results of the command are transferred from host controller to host as event packets. For most HCI commands, bluetooth hardware generates events to report the results of the command[1][3][4]. In this section we present the bluetooth HCI commands and the events which is generated by host controller to return the results of bluetooth HCI commands. Fig. 1 shows the connection procedure between master and slave.

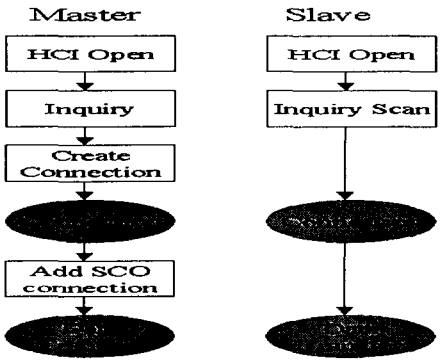

Fig. 1 Bluetooth connection procedure

The operation to receive event packet and ACL data packet is asynchronous. In order to cope with this situation, host must register callback functions which will be executed when event packet or ACL data packet is transferred from host controller.

2.1 Open

Open command sets HCI interface, and sends Reset, Read_Buffer_Size, Read_Local_Name, Read_BD_ADDR command to bluetooth hardware. Reset command resets host controller, link manager, and RF module. Read_Buffer_Size command reads the maximum buffer size of ACL and SOC data which can be transmitted from host to host controller. Read_Local_Name command asks bluetooth hardware to return the stored user-friendly name for the bluetooth device. Read_BD_ADDR command asks bluetooth hardware to return the 48-bit bluetooth device address. For each command host controller generates Command Complete event. The first byte is the number of HCI command packets which are allowed to be sent to the host controller from host. And the second two bytes are the opcode of the command, the followings are return parameters, and the format of return parameters depends on the command.

2.2 Inquiry

Inquiry command makes bluetooth device operate in inquiry mode. Inquiry mode is used to discover other nearby bluetooth devices. Inquiry command uses low address part (LAP), and the range of LAP is 0x9E8B00-0x9E8B3F. In this paper we used 0x9E8B33. As a result of inquiry, master gets information of slave such as bluetooth device address of slave and the clock offset between slave and master. Host controller generates command status event after it receives inquiry command. The command status event packet indicates whether the received command can be executed or not. And then bluetooth hardware starts to inquire nearby bluetooth devices. When bluetooth hardware receives the responses from other nearby bluetooth devices, host controller generates Inquiry Result event and this event contains the BD_ADDR, Page_Scan_Repetition_Mode, Page_Scan_Period_Mode, Page_Scan_Mode, Class_of_Device, and the Clock_Offset. After inquiry is finished, host controller generates Inquiry Complete event which contains the status of inquiry command.

2.3 Inquiry scan and page scan

In inquiry scan mode, bluetooth device responds to the inquiry requests from other bluetooth devices, and in page scan mode, bluetooth device responds to the connection requests from other bluetooth devices. Write_Inquiry_Scan_Activity sets the inquiry scan interval and inquiry scan window. Inquiry scan interval is the amount of time between consecutive inquiry scans, and inquiry scan window is the

amount of time for the duration of the inquiry scan. In this paper, we set inquiry scan interval to 0x0800 (1.28 sec, 0x0800 * 0.625 ms), and set inquiry scan window to 0x0012 (11.25 ms, 0x0012 * 0.625 ms). Write_Page_Scan_Activity sets the page scan interval and page scan window. Page scan interval is the amount of time between consecutive page scans, and page scan window is the amount of time for the duration of the page scan. In this paper, we set page scan interval to 0x1000 (2.56 s), and set page scan window to 0x0900 (1.44 s).
Write_Scan_Enable controls whether or not the bluetooth device will periodically scan for inquiry and page attempts. When the scan enable argument is 0x01, inquiry scan is enabled, and when the argument is 0x02, page scan is enabled. In this paper we assigned 0x03 to activate inquiry scan and page scan. Host controller generates command complete event for each HCI command.

2.4 Create_Connection

Create_Connection command causes link manager to create ACL connection to the bluetooth device with the BD_ADDR which is the result of inquiry procedure. The host controller of master generates Command Status event when it receives Create_Connection command. The status event is used to indicate that the command has been received, and that host controller is currently performing the task for that command. After connection is established, both of the host controllers generate Connection Complete event to indicate the establishment of connection. Connection Complete event contains Status, Connection_Handle, BD_ADDR of other bluetooth device forming the connection, Link_Type, and Encryption_Mode. In this case Link_Type is 0x01, which indicates ACL connection.

2.5 Add_SCO_Connection

Add_SCO_Connection command causes link manager to create SCO connection using the ACL connection specified by the Connection_Handle which is the result of Create_Connection. Using SCO connection master and slave can communicate voice data. In this paper we used HV3 packet type. When host controller receives Add_SCO_Connection command, it generates Command Status event. After connection is established, both of the host controllers generate Connection Complete event to indicate the establishment of SCO link. In this case Link_Type is 0x00, which indicates SCO connection.

Ⅲ. Image data transfer

The image from camera is compressed using M-JPEG compression. M-JPEG is the time sequence of JPEG images, so the compression rate is the same with JPEG. The resolution of the image used in this paper is 320x240 and the color is true color (24bits). The compression rate is 30:1 and the transfer rate of image data is 2 frames / sec. One image frame is transmitted in several ACL packets. To indicate the start of one frame, we used the first byte of every ACL packet as an indicator. If the first byte of received ACL packet is 0x00, this means that this is the start of a new frame. If the first byte of received ACL packet is 0x01, this means that this is the continuing frame. When new frame is received, client displays the previous image. And when continuing frame is received, client saves the image to buffer.

Ⅳ. SYSTEM ENVIRONMENT

Fig. 2 shows the system environment used in this paper.

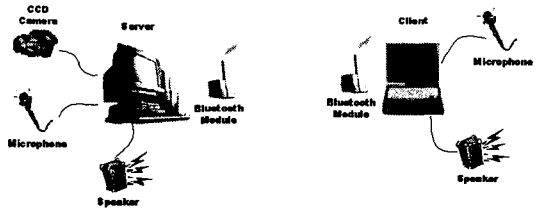

Fig. 2. System environment

The server compresses the image data received from camera using M-JPEG, and send the image data through ACL link of bluetooth, The client decompresses the M-JPEG data received from

bluetooth and display the image to screen. The voice data can be exchanged simultaneously between server and client. And the server can receive hexadecimal number from the client and can output the number to parallel port. We used bluetooth protocol analyzer to check the packets between master and slave. Fig. 3 shows the output of bluetooth protocol analyzer.

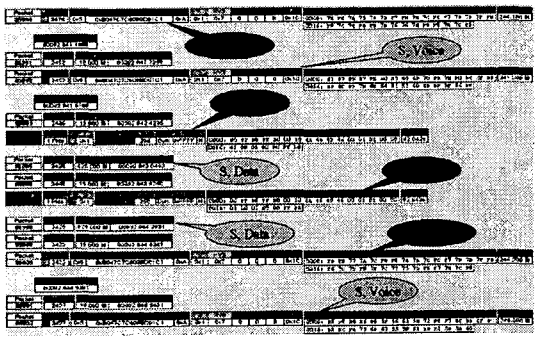

Fig. 3. Output of bluetooth protocol analyzer

The output of bluetooth protocol analyzer shows the voice data between master and slave, and the image data from master to slave. Fig. 4 shows the analysis of the output of bluetooth protocol analyzer in compact form. In the figure, M means that master sends data, and S means that slave sends data. In this case master sends voice data through SCO link and image data through ACL link, and slave sends voice data through SCO link. The figure also shows the hopping frequency of bluetooth(GHz).

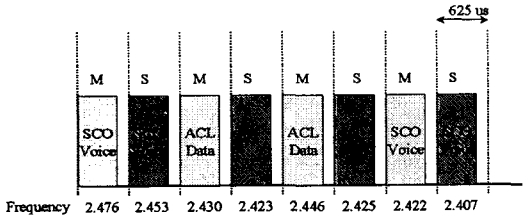

Fig. 4. Analysis of protocol analyzer

V. GUI Implementation

Fig. 5 shows the GUI implementation. The application program is implemented using Visual C++ in Windows environment. The application program uses dynamic link library (DLL) supplied by ATMEL.

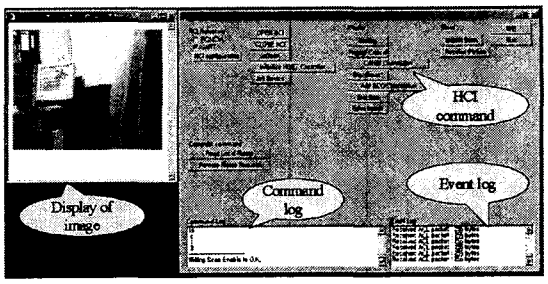

Fig. 5. GUI implementation

All the HCI commands can be executed by clicking buttons. And the command logs and event logs are displayed in respective windows. In case of client, the image transferred from server is displayed on independent window.

VI. Conclusion

In this paper, we implemented bluetooth application which transfer image from server to client and exchange voice between server and client. And we presented bluetooth HCI commands and events generated by host controller to return the result of HCI commands. We set up the bluetooth connection procedure. In other words, our bluetooth application inquires other bluetooth devices, makes connection, and communicates voice and data. The system can be applicable to home security systems and remote control systems.

References

[1] "Bluetooth SIG Technical Standard Draft V1.1", October. 19. 2000.

[2] Jaap C. Haartsen, Sven Mattisson, "Bluetooth-A New Low-Power Radio Interface Providing Short-Range Connectivity", IEEE, Vol. 88, No. 10, Oct. 2000.

[3] "Bluetooth Development Kit Technical Overview", ATMEL, 2000.

[4] "Bluetooth HCI", ATMEL, 2000.