

안드로이드 기반의 블루투스 디바이스 응용 설계 및 구현

The Android-based Bluetooth Device Application Design and Implementation

조 효 성*
(Hyosung Cho)

이 혁 준**
(Hyukjoon Lee)

요 약

현재, 차량 내에서 사용하는 대부분의 블루투스 핸드프리 디바이스는 음성 통화, 발신자 식별번호 표시 및 SMS 메시지 표시 등의 전화 서비스 기능을 제공하고 있지만 인터넷 기반의 텍스트 데이터 디스플레이 기능은 지원하지 않고 있다. 최근 들어 차량에서도 SNS와 같은 인터넷 서비스 이용에 대한 요구가 증가하고 있어, 기존의 핸드프리 기능 이외에도 인터넷 기반의 텍스트 데이터를 표시하는 기능의 개발이 필요하다. 제안하는 블루투스 디바이스 응용은 SNS 메시지 및 이메일 도착 알림 표시 등의 향상된 기능을 포함하며, 구현 플랫폼으로는 대부분의 SNS 응용 프로그램의 동작이 가능하고 소형 임베디드 디바이스에 탑재가 용이한 안드로이드 플랫폼을 사용한다. 제안하는 블루투스 디바이스와 연결되는 스마트폰 또는 태블릿 PC 역시 안드로이드 플랫폼 기반의 단말기이며, 단말 탑재 기능 모듈인 오디오-텍스트 게이트웨이는 안드로이드 앱 형태로 설계된다. 오디오-텍스트 게이트웨이 앱은 SNS 텍스트 데이터 수신시, 메시지 헤더 정보에서 제목, 발신자 정보 등을 추출하여 텍스트 데이터 형태로 ACL(Asynchronous Connection-Oriented) 링크를 통해 블루투스 디바이스로 전송하여 화면에 표시되도록 한다. 일반적으로 안드로이드 플랫폼에 탑재된 블루투스 핸드프리 또는 헤드셋 프로파일은 오디오 게이트웨이의 기능만을 포함하고 있기 때문에 안드로이드 기반의 블루투스 디바이스에서는 음성 데이터의 스피커 출력이 불가능하다. 따라서 제안하는 블루투스 디바이스 응용은 블루투스 헤드셋 프로파일에서 일반적으로 사용되는 SCO(Synchronous Connection-Oriented) 링크를 통한 음성 데이터 전송 방식 대신에 ACL 링크를 통한 스트리밍 방식을 적용한다.

Abstract

Today, although most bluetooth hands-free devices within a vehicle provide telephone service functions such as voice communication, caller id display and SMS message display and so on, they do not provide a function that displays Internet-based text data. We need to develop a scheme that displays the internet-based text data including existing hands-free function because the request for using the Internet service is increasing within a vehicle recently. The proposed bluetooth device application includes advanced function such as SNS message arrival notification, the message display function and we chose Android as the implementation mobile platform giving consideration to the fact that most SNS applications operate on Android and the platform is easily embedded into small embedded device. Smartphone or tablet PC connected with the proposed bluetooth device is an Android-based device and we designed a form of Android app for the function implementation of the devices. When the audio-text gateway app receives SNS text data, it extracts title and sender information from the message header information in a form of text data and sends them via ACL (Asynchronous Connection-Oriented) link to the bluetooth device showing the data on the screen. Android-based bluetooth devices are not possible to play voice through speaker because the bluetooth hands-free or headset profile ported within Android platform normally only includes audio gateway's function. The proposed bluetooth device application, therefore, applies the streaming scheme that sends data via ACL link instead of the way that sending them via SCO (Synchronous Connection-Oriented) link.

Key words : Android, Bluetooth, ACL, SCO, Audio Gateway, Audio streaming

† 이 논문은 2011년도 광운대학교 교내 학술 연구비 지원에 의해 연구되었음.

* 주저자 : 광운대학교 임베디드 소프트웨어공학과 석사과정

** 공저자 및 교신저자 : 광운대학교 컴퓨터공학과 교수

† 논문접수일 : 2012년 2월 14일

† 논문심사일 : 2012년 2월 22일

† 게재확정일 : 2012년 2월 27일

1. 서 론

최근 들어, 운전 중 전화통화의 안전 문제가 심각해짐에 따라 블루투스 핸즈프리 장치의 사용이 확산되고 있다. 모바일 단말과 블루투스로 연결하여 사용할 수 있는 차량용 핸즈프리 장치는 주로 차량 내장형 또는 외장형 내비게이터의 주요 기능 중 하나로 포함되고 있으며, 대부분 음성통화 기능, 발신자 식별 번호 표시 기능, SMS 메시지를 표시 기능 등을 제공한다. 블루투스 헤드셋 및 핸즈프리 디바이스의 주목적은 음성 통화시 사용자의 손이 자유롭지 못한 상황이거나 모바일 단말의 접근이 용이하지 못할 경우에 편리성을 제공하기 위한 것으로, 최근에는 SNS의 수요가 증가하면서 SNS 메시지 표시기능에 대한 요구도 높아지고 있다[1]. 또한, 갤럭시탭과 같이 음성통신 기능을 포함하는 태블릿 PC의 경우, 주로 전화통화시 블루투스 헤드셋을 이용하는 것이 편리하지만 발신자 식별번호, 메시지 등의 텍스트 표시 및 확인이 불가능하여 편리성에 많은 제약이 따르고 있다.

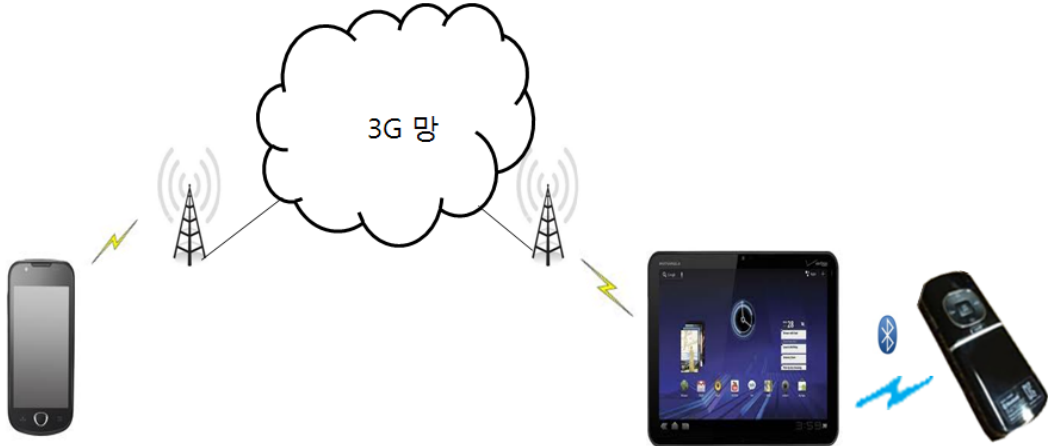
본 논문에서는 음성 출력 기능과 텍스트 표시 장치를 탑재한 블루투스 핸즈프리 및 헤드셋 디바이스 상에서 음성통화와 SMS 및 SNS, 이메일 텍스트 정보 표시가 가능하도록 하는 응용 시스템의 설계 및 구현 방법에 대해 설명한다. 특히, 빠르게 진화하고 있는 스마트 단말 및 응용 서비스의 변화를 수용하기 위하여, 목표 플랫폼으로 안드로이드 플랫폼을 고려한다. 안드로이드는 리눅스를 기반으로 하는 모바일 플랫폼으로서 새로운 응용을 다운로드 받아 간단한 설치과정을 거쳐 동작시킬 수 있고, 응용이 하드웨어 기능을 사용할 수 있도록 다양한 프레임워크를 제공한다. 제안하는 응용 시스템의 경우, 스마트폰 또는 태블릿 PC와 블루투스 디바이스 양쪽에 모두 응용 프로그램의 형태로 설치되어 동작한다. 이 중, 스마트폰 또는 태블릿 PC에서 동작하는 응용은 오디오-텍스트 게이트웨이로서 음성통화 수신 알림을 블루투스 디바이스로 전송하는 기능 이외에도 SNS, 이메일 등

의 텍스트 기반 데이터의 추출 및 전송 기능을 담당한다.

블루투스 디바이스를 안드로이드 플랫폼 기반으로 구현할 경우, 안드로이드 플랫폼에 탑재된 블루투스 핸즈프리 및 헤드셋 프로파일이 오디오 게이트웨이 기능만을 포함하고 있기 때문에 음성출력 장치를 통한 음성신호 발생이 불가능 하다. 이것은 안드로이드에 탑재된 블루투스의 오디오 게이트웨이가 SCO 링크를 사용하여 음성 데이터를 전송할 수는 있지만 스피커를 통해 출력하지는 못하기 때문이다. 본 논문에서는 기존의 핸즈프리 및 헤드셋 프로파일 기반의 음성 데이터 전송 방식 대신에 ACL 링크를 통한 음성 데이터 스트리밍 및 재생 방식을 적용하여 이 문제를 해결한다.

본 논문에서 제안하는 응용과 관련된 제품으로 최근 들어 영국 알파리사에서 출시된 안드로이드 기반의 손목시계 형 블루투스 디바이스인 아이엠 워치(I'm Watch)와 소니 에릭슨사의 라이브뷰 등을 예로 들 수 있다. 아이엠 워치는 전화 수신, 전화번호부 검색, SMS 메시지 확인, 음악 듣기, 주식 확인 기능 등을 제공하여 사용자에게 많은 관심을 끌고 있다. 또한 일본 소니 에릭슨사에서 출시된 라이브뷰는 아이엠 워치와 같은 손목시계 형태로서 전화 수신과 SMS 메시지, 페이스북과 트위터의 업데이트 메시지를 표시하는 기능을 제공하여 큰 관심을 끌고 있다[2]. 이 두 제품들은 제안하는 블루투스 응용과 같이 안드로이드 플랫폼을 기반으로 하고 있으나 안드로이드의 블루투스 프레임워크의 특징 때문에 음성신호 출력이 거의 불가능하다.

<그림 1, 2>는 본 논문에서 제안하는 응용의 유스케이스에 대한 개념을 도식화 한 그림이다. 본 논문의 구성은 다음과 같다. 2장에서는 제안하는 연구를 위한 배경 기술에 대해 설명하고, 3장에서는 블루투스 디바이스 응용 시스템 설계에 대하여 설명한다. 4장에서는 개발 환경과 파라미터 값을 적용한 응용의 구현 및 기능 검증을 설명하고, 5장에서 본 연구의 결과를 간단히 정리하고 향후 과제에 대해서 논의한다.



〈그림 1〉 승객을 위한 블루투스 디바이스 유스케이스
 (Fig. 1) Use case of the bluetooth device for passengers



〈그림 2〉 운전자를 위한 블루투스 디바이스 유스케이스
 (Fig. 2) Use case of the bluetooth device for drivers

II. 선행 연구

1. 안드로이드

안드로이드는 운영체제, 미들웨어, 키(key) 애플리케이션들을 포함하여 모바일 디바이스에서 사용하는 모바일 플랫폼이다[3]. 모바일 플랫폼의 여러 가지 특징 중에서 안드로이드는 다음과 같은 차별적인 장점들을 가지고 있다. 첫째, 안드로이드의 커널은 리눅스 커널을 기반으로 하여 임베디드 환경에 최적화되었다. 리눅스는 오랫동안 개발된 커널로서 정교한 메모리 관리, 안정적인 멀티 스레드, 보안 등의 고급 기능을 무료로 사용할 수 있다. 둘째, 고급 언어인 자바를 사용하여 생산성을 높이며

하드웨어 추상 계층을 제공하기 때문에 전문 지식 없이 응용의 개발이 용이하다. 반면 고급 언어이기 때문에 하드웨어에 최적화된 성능을 위한 구현이 다소 어려울 수 있으나 NDK(Native Development Kit)를 통해 그 문제점을 해결할 수 있다. 셋째, 검증된 많은 라이브러리들을 포함하고 있기 때문에 응용 개발에 필요한 대부분의 기능 구현은 외부 라이브러리를 사용하지 않는다. 넷째, 플랫폼에 내장된 빌트인(built-in) 응용과 사용자가 만든 응용이 동일한 API를 사용하므로 모든 응용은 우선순위가 평등하다. 따라서 사용자의 의도에 따라 기본 응용들을 교체할 수 있고 플랫폼을 구성하는 요소들을 자유롭게 선택할 수 있다. 마지막으로 안드로이드는 개방형 소스에 걸맞게 개발 툴과 관련 문서들이 모



〈그림 3〉 안드로이드의 구조
(Fig. 3) The structure of Android

두 무료로 제공된다. 더욱이 운영체제의 핵심 소스 코드까지 대부분 공개되어 있다.

안드로이드의 구조는 <그림 3>과 같이 애플리케이션, 애플리케이션 프레임워크, 라이브러리, 안드로이드 런타임, 리눅스 커널로 구성되어 있다. 안드로이드 런타임은 달빅 가상 머신과 코어 라이브러리를 포함한다. 다른 모바일 플랫폼과는 달리 안드로이드 환경에서 동작하는 모든 응용 프로그램은 달빅 가상 머신에서 동작한다. 자바 언어로 응용을 구성하는 안드로이드가 자바 가상 머신을 사용한다면 Sun 사의 라이선스 정책으로 인해 안드로이드를 무료로 배포할 수 없기 때문에 구글이 달빅 가상머신을 채택했다. 응용 프로그램을 구성할 수 있는 컴포넌트는 액티비티, 서비스, 브로드캐스트 리시버, 콘텐츠 프로바이더가 있고 각 컴포넌트 간에는 메시지 형태의 인텐트(Intent)를 주고받으며 통신한다[4]. 그 컴포넌트들은 안드로이드가 제공하는 다양한 프레임워크를 사용하여 응용에 필요한 기능을 구현할 수 있다. 다음은 제안하는 시스템의 구현을 위해 위에서 언급한 세 가지 프레임워크에 대한 자세한 설명이다.

1) 블루투스 프레임워크

안드로이드는 블루투스 통신을 위해 안드로이드 블루투스 API를 사용할 수 있는 프레임워크를 제공한다. 블루투스 프레임워크는 안드로이드 기반의

디바이스가 블루투스 연결 및 연결 해제, 디바이스 탐색 및 응답, 데이터 전송 기능을 사용할 수 있도록 프로파일 구현 모듈과 블루투스 하드웨어 인터페이스를 제공한다. 안드로이드 블루투스 프레임워크는 범용 액세스 프로파일(Generic Access Profile), 서비스 탐색 프로파일(Service Discovery Profile), 헤드셋 프로파일, 핸드프리 프로파일, RFCOMM(Radio Frequency Communication) 프로파일 구현 모듈을 포함한다. 서비스 탐색 프로파일은 블루투스 서비스 탐색 기능 및 그 서비스의 데이터베이스 기능 등을 제공하고, RFCOMM 프로파일은 유저 데이터 전송을 위한 RFCOMM 소켓을 제공한다[5]. RFCOMM 소켓은 패킷 기반의 데이터를 ACL 링크를 통해 전송할 수 있는 채널을 사용한다. 블루투스 프레임워크는 SDK를 이용한 프로그램 개발 레벨에서 RFCOMM 소켓을 이용한 데이터 전송은 가능하지만 SCO 소켓을 이용한 음성 데이터 전송 기능은 현재까지 제공하지 않는다. 따라서 안드로이드 기반의 디바이스에서 SCO 소켓을 통해 음성 데이터를 처리하는 블루투스 응용 프로그램을 작성하는 것은 불가능하므로 RFCOMM 소켓을 이용하는 방법으로 적용 되어야 한다.

2) 오디오 프레임워크

오디오 데이터의 저장과 플레이는 오디오 프레임워크가 담당한다. 따라서 응용은 오디오 프레임워크를 사용하여 오디오 데이터를 PCM 포맷으로 저장 및 플레이 할 수 있다. 커뮤니케이션 프로세서를 통해 디코딩된 전화 음성 데이터는 서론에서 설명한 안드로이드 오디오 데이터의 저장은 안드로이드 미디어 패키지의 오디오 레코드 클래스를 사용한다. 이 클래스는 모바일 플랫폼의 오디오 입력 하드웨어에서 나온 오디오 데이터의 입력 소스를 결정하고 디코딩에 필요한 파라미터 값을 설정하여 데이터 버퍼에 저장하는 메소드를 제공한다. 파라미터의 종류는 샘플링 주파수, 채널, 인코딩, 버퍼 크기 값이 있는데 버퍼의 최소 크기는 이 클래스의 객체를 생성할 때 미리 결정된다. 오디오 트랙 클래스는 오디오 데이터를 플레이하기 위한 메소드를 제공한다.

3) 텔레포니 프레임워크

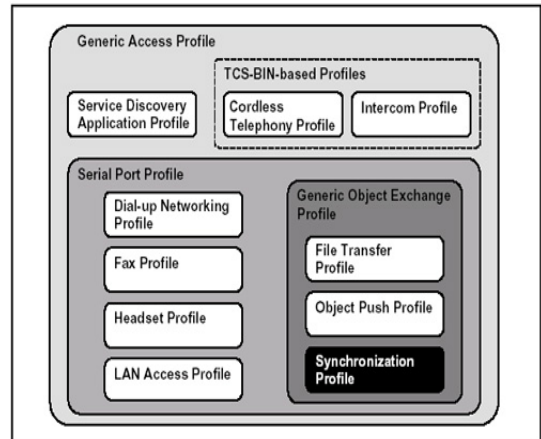
안드로이드 텔레포니 프레임워크는 상위 계층의 텔레포니 서비스와 하위 계층의 라디오 인터페이스 계층을 포함하여 전화 수신, 전화 끊기, SMS 메시지 전송 기능 등을 제공한다. 이 프레임워크는 텔레포니 서비스를 위해 모바일 단말의 위치와 콜 상태 값을 모니터링하는 기능, 네트워크 타입(GSM, CDMA, LTE 등) 저장 기능과 이웃 베이스 스테이션 위치 정보 저장 기능 등을 포함한다. 또한 SMS 메시지 전송 기능을 포함하여 일반 안드로이드 응용 프로그램에 SMS 메시지를 수신하는 기능을 추가할 수 있다. 본 연구에서는 이 프레임워크에서 SMS 메시지 수신, 발신자 식별번호 표시, 콜 상태 값 확인 기능과 전화 수신 기능을 사용한다.

2. 블루투스

블루투스는 2.4 GHz 대역의 ISM(Industrial Scientific Medical) 밴드를 사용하는 근 거리의 블루투스 디바이스를 연결하여 데이터를 전송하는 통신방식이다. 블루투스는 저전력 소비와 PAN(Personal Area Networking) 영역에서의 무선 데이터 전송 기능을 지원하기 때문에 사용자들에게 많은 관심을 받아왔다. 블루투스 통신에서 마스터는 최대 7개까지의 슬레이브와 연결하여 피코넷을 형성한다. 슬레이브는 마스터와 연결되면 마스터의 주파수 호핑(Frequency Hopping) 스케줄링에 의해 마스터와 데이터를 교환한다[6]. 블루투스 디바이스 간에 데이터를 전송하는 링크의 종류는 재전송이 가능한 패킷 기반의 데이터와 제어 명령어를 전송할 수 있는 ACL 링크와 재전송을 하지 않는 서킷 기반의 음성 데이터를 전송할 수 있는 SCO 링크가 있다. 가용한 데이터 링크는 디바이스에 탑재되는 블루투스 프로파일 모듈에 따라 달라질 수 있으나 ACL 링크는 모든 프로파일에서 기본적으로 제공되므로 대부분의 블루투스 통신에서 사용할 수 있다.

블루투스 프로파일은 블루투스 응용 소프트웨어를 구현할 때 필요한 프로토콜의 종류, 구조 및 사용 방법을 규정한 것이다[7]. <그림 4>는 프로파일

의 상관관계를 나타낸 것으로서 범용 액세스 프로파일은 모든 프로파일을 포함하며 블루투스 통신에 필요한 디바이스 탐색, 응답, 연결, 보안 기능 등을 제공한다. 나머지는 프로파일에서 프로토콜 스택의 상위 계층이 어떤 프로토콜 계층을 사용하느냐에 따라 3가지로 분류된다. 본 연구에서는 범용 액세스 프로파일과 핸드프리 프로파일에 대해 설명한다.



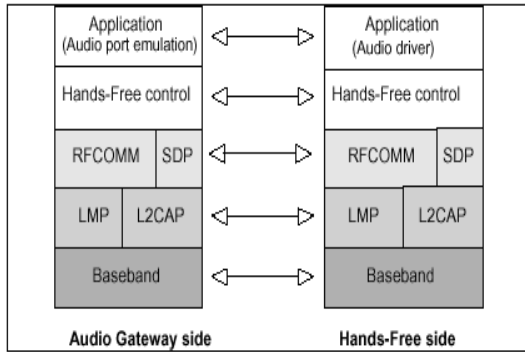
<그림 4> 블루투스 프로파일
<Fig. 4> Bluetooth profile

1) 범용 액세스 프로파일

범용 액세스 프로파일은 블루투스 통신을 위한 기본적인 프로파일로서 다음의 사항 등을 서술하고 있다. 이 프로파일은 전송 프로파일(Transport Profile)과 응용 프로파일(Application Profile)에서 사용하는 동작 모드와 디바이스 액세스 절차에 관련된 정의, 권장 사항, 공통 사항 등을 정의하고 있다. 그리고 링크와 채널이 항상 연결 성공할 수 있도록 보장하기 위해 스탠바이 상태와 연결 상태에서 디바이스가 동작하는 방법을 설명한다. 디바이스에 범용 액세스 프로파일이 포팅되어 있다면 ACL 링크를 사용할 수 있으므로 다른 프로파일이 없어도 주변 블루투스 디바이스에 일반 데이터와 명령어를 전송할 수 있다.

2) 핸드프리 프로파일

핸드프리 프로파일은 모바일 단말과 핸드프리 디바이스간의 통신을 위해 제공하는 최소한의 기능



〈그림 5〉 블루투스 핸드프리 프로토콜 스택 (a)오디오 게이트웨이 (b)핸즈프리 유닛
 〈Fig. 5〉 Bluetooth hands-free protocol stack (a)Audio gateway (b)Hands-free unit

집합들을 정의하고 있다. 이 프로파일에서 디바이스의 역할은 오디오 게이트웨이와 핸드프리 유닛으로 분류된다. <그림 5>는 오디오 게이트웨이와 핸드프리 유닛의 프로토콜 스택 구조와 각 계층 간의 연결을 나타낸 것이다. 두 프로토콜 스택의 모든 계층은 응용 계층을 제외하고 같은 역할을 수행하여 통신에 필요한 데이터를 교환하지만 데이터 전송 순서에 대해서는 마스터와 슬레이브 역할을 구분하지 않는다. 오디오 게이트웨이의 응용 계층은 음성 데이터를 전송하는 오디오 포트 에뮬레이션 기능을 지원하고, 핸드프리 유닛의 응용 계층은 음성 데이터를 전송과 출력하는 오디오 드라이버 기능을 제공한다. 핸드프리 프로파일은 다양한 모바일 디바이스와 운영체제 등에 포팅될 수 있지만 안드로이드와 같은 모바일 플랫폼에 탑재되면 오디오 게이트웨이 프로토콜 스택으로 구현된다. 따라서 모바일 플랫폼이 포팅된 모바일 단말은 일반적으로 핸드프리 디바이스로 사용될 수 없다.

Ⅲ. 블루투스 디바이스 응용 설계

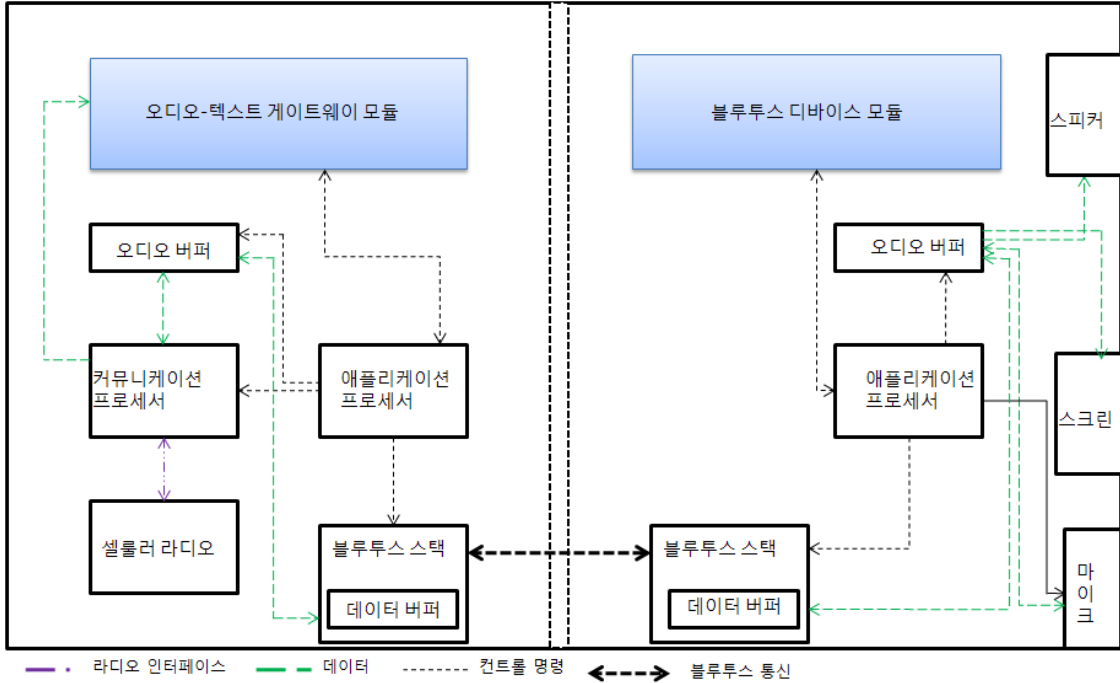
1. 오디오-텍스트 게이트웨이와 블루투스 디바이스 유닛

본 절에서는 3G 망을 통해 음성 통화, 발신자 식별번호, SMS 수신 기능과 네트워크 망을 통해 인터넷

넷 기반의 SNS 메시지 수신, 이메일의 제목 표시 기능을 제공하는 스마트폰 혹은 태블릿 PC 탑재 오디오-텍스트 게이트웨이 및 안드로이드 기반 소형 디바이스 탑재 블루투스 디바이스 유닛으로 구성되는 응용 시스템의 설계에 대해 설명한다. <그림 6>은 오디오-텍스트 게이트웨이와 블루투스 디바이스의 각 컴포넌트를 설명한 블록도이며, 오디오-텍스트 게이트웨이 모듈과 블루투스 디바이스 모듈 이외의 모듈들은 기존의 안드로이드 플랫폼에 포함된 S/W 및 H/W 모듈들이므로 설명에서 제외한다.

오디오-텍스트 게이트웨이 모듈은 모바일 단말의 콜 상태를 확인하는 콜 상태 확인부, 음성 데이터와 SMS 메시지 및 인터넷 기반의 텍스트 데이터를 임시 저장하는 음성과 텍스트 데이터 처리부, 블루투스 데이터를 전송하는 블루투스 데이터 전송부, 블루투스 장치 탐색 및 연결 기능을 제공하는 블루투스 연결부로 구성된다. 블루투스 디바이스 모듈은 오디오-텍스트 게이트웨이 모듈과 비슷한 구성을 가지고 있지만 사용자 이벤트 처리부를 통해 화면 UI를 제공한다. <그림 7>은 <그림 6>에 포함된 오디오-텍스트 게이트웨이 모듈과 블루투스 디바이스 모듈의 내부 구조를 구성 요소 간의 연결로 나타낸 블록도이다. <표 1>은 각 컴포넌트가 상호작용하여 오디오-텍스트 게이트웨이 모듈과 블루투스 디바이스 모듈의 기능별 동작내용을 요약한 것이다.

오디오-텍스트 게이트웨이가 인터넷 기반의 텍스트 데이터를 수신하면 콜 상태 확인부의 상태 값에 상관없이 안드로이드 시스템은 패킷의 포트 번호를 확인하여 패킷의 최종 목적지가 SNS 응용 혹은 이메일 응용인지를 판단한다. 오디오-텍스트 게이트웨이가 설정한 특정 응용 포트 번호를 가진 패킷이 확인되면 커널 영역에서 메모리 공간을 할당하고, 인터넷 패킷 복사함수를 통해 패킷 복사를 수행한다. 복사된 패킷은 헤더 정보를 붙여 데이터 처리부로 전달되어 블루투스 디바이스 유닛으로 전송된다. 오디오-텍스트 게이트웨이 모듈에서 콜 상태 확인부의 상태 값이 IDLE에서 RINGING으로 천이되면 음성과 텍스트 데이터 저장부는 안드로이드의 Phone 응용에서 수신된 발신자 식별번호를 임시 저장한 후



〈그림 6〉 블루투스 디바이스 응용 시스템의 아키텍처 (a) 오디오-텍스트 게이트웨이 (b)블루투스 디바이스
 (Fig. 6) Architecture of bluetooth device application system (a)Audio-text gateway (b)Bluetooth device

“CALLERID” 헤더를 붙여 블루투스 데이터 처리부로 보내 블루투스 디바이스 유닛으로 전송한다. 다음으로 콜 상태가 RINGING에서 OFFHOOK으로 천이되면 음성과 텍스트 데이터 저장부는 오디오 저장 인터페이스를 통해 음성 데이터를 오디오 버퍼에 임시 저장한 후, “VOICE” 헤더를 붙여 블루투스 데이터 처리부로 전송한다. 이때, 음성 데이터는 블루투스 ACL 링크를 통해 블루투스 디바이스 유닛으로 전송되며 세부 처리과정은 1) 절에서 자세히 설명한다. 이와 반대로 블루투스 디바이스 유닛으로부터 수신한 마이크 음성 데이터는 “VOICE” 헤더를 제거된 후 오디오 플레이 인터페이스를 통해 오디오 하드웨어의 버퍼에 임시 저장된다. 이 과정이 끝나면 마이크 음성 데이터는 커뮤니케이션 프로세싱 과정을 거쳐 3G망으로 송출된다.

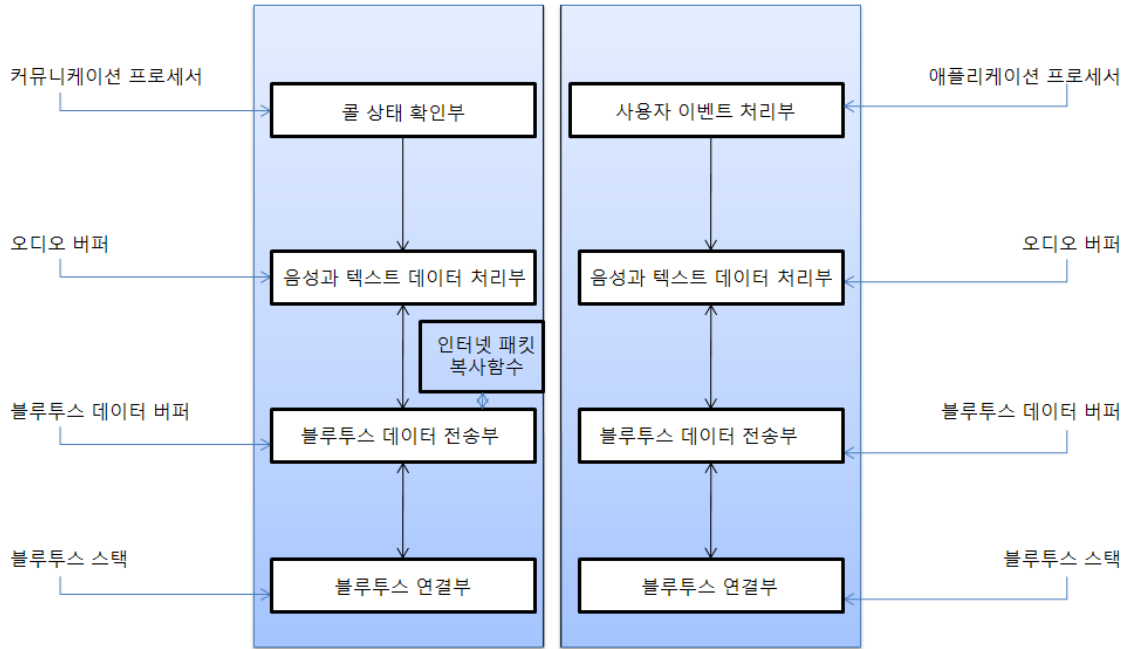
블루투스 디바이스 모듈은 사용자와 직접적인 상호 작용을 수행한다. 오디오-텍스트 게이트웨이를 통해 콜이 수신되면 블루투스 디바이스 모듈은 사용자 이벤트 처리부를 통해 발신자 식별번호와

함께 대화상자를 띄워 통화 수락을 할 수 있도록 UI를 제공한다. 음성 통화가 시작되면 오디오-텍스트 게이트웨이에서 수신된 음성 데이터는 이 디바이스의 블루투스 데이터 처리부로 전송되고 오디오 플레이 인터페이스를 통해 오디오 버퍼를 이와 동시에 블루투스 디바이스 모듈이 마이크 음성 데이터를 ACL 링크를 통해 오디오-텍스트 게이트웨이로 전송한다. 그리고 인터넷 기반의 텍스트 데이터를 수신할 때, SNS, 이메일의 종류를 분류하여 각 응용의 이름, 발신자 정보, 수신된 메시지 및 이메일 제목 등을 모두 팝업창에 표시한다. 오디오-텍스트 게이트웨이에서 이 데이터를 저장하지만 사용자의 요청에 따라 블루투스 디바이스에서도 데이터를 저장할 수 있다.

1) 오디오-텍스트 게이트웨이 모듈의 구성

(1) 콜 상태 확인부

안드로이드 텔레포니 프레임워크에서 사용되는 콜 상태 값은 세 가지 상태로 구분되고 수신 콜인 경우



〈그림 7〉 오디오-텍스트 게이트웨이 모듈(좌)과 블루투스 디바이스 모듈(우)
 (Fig. 7) Audio-text gateway module(left) and bluetooth device module(right)

〈표 1〉 기능별 오디오-텍스트 게이트웨이와 블루투스 디바이스의 동작 과정
 (Table 1) Operations of audio gateway and bluetooth device

	텍스트 데이터 수신	음성 통화	SMS 메시지 수신
오디오-텍스트 게이트웨이	(1) 안드로이드 시스템으로 수신되는 인터넷 패킷에서 포트 번호를 식별하여 SNS 응용으로 수신 되는 특정 패킷을 복사 (2) 복사된 패킷에서 애플리케이션 계층을 위한 페이로드에서 SNS 메시지 혹은 이메일 제목 부분을 추출 (3) 추출된 데이터를 바이트 스트림으로 변환하여 ACL 링크를 통해 블루투스 디바이스로 전송	(1) 오디오-텍스트 게이트웨이는 3G 망에서 콜을 수신하고 발신자 식별번호를 블루투스 디바이스로 전송 (2) 사용자가 전화를 수락하면 커뮤니케이션 프로세스가 디코딩한 상대 음성 데이터를 PCM 포맷으로 변환하여 블루투스 디바이스로 전송하는 동시에 그 디바이스에서 수신한 사용자 음성 데이터를 인코딩 과정을 거쳐 3G망으로 송출	(1) 발신자 식별번호와 SMS 메시지 몸체 부분을 데이터 버퍼에 임시 저장 (2) 그 데이터를 바이트 스트림으로 변환하여 ACL 링크를 통해 블루투스 디바이스로 전송
블루투스 디바이스	(1) 오디오-텍스트 게이트웨이에서 수신된 데이터를 데이터 버퍼에 임시저장 (2) 오디오-텍스트 게이트웨이에 설치된 응용의 종류를 구분하여 블루투스 디바이스 유닛의 스크린에 응용 이름과 SNS 메시지, 이메일 제목 등을 화면표시	(1) 오디오-텍스트 게이트웨이에서 PCM 포맷으로 수신된 음성 데이터를 데이터 버퍼에 임시저장 (2) 음성 데이터를 출력하는 인터페이스를 사용하여 상대방 음성을 스피커 출력 (3) 사용자 음성 데이터를 PCM 포맷으로 변환하여 ACL 링크를 통해 전송	(1) 오디오-텍스트 게이트웨이에서 수신된 SMS 메시지를 데이터 버퍼에 임시 저장 (2) 팝업창과 함께 발신자 식별번호와 메시지를 화면에 표시

는 세 가지 상태가 모두 사용된다. 상태 정보는 모바일 단말의 현재 콜 상태에 따라 천이된다. 콜 상태 정보에서 IDLE은 전화 통화를 하지 않는 일반 상태이고, RINGING은 상대방이 전화를 시도하여 폰이 울리는 상태이고, OFFHOOK는 전화 음성 통화가 시작되는 상태를 의미한다. 안드로이드 시스템은 이 상태 정보들을 인텐트 형식으로 브로드캐스트하거나 사용자에게 noti피케이션 형태로 전달된다.

(2) 음성 및 텍스트 데이터 처리부

오디오-텍스트 게이트웨이가 인터넷 기반의 텍스트 데이터를 수신하면 인터넷 패킷 복사 함수는 패킷의 포트번호를 확인하여 SNS 응용 및 이메일 응용을 구분한다. 인터넷 패킷 복사 함수는 특정 응용으로 수신되는 인터넷 패킷을 복사하여 데이터 버퍼에 임시 저장한 후 애플리케이션 페이로드만을 추출하여 SNS 메시지 혹은 이메일 제목 등을 블루투스 디바이스에 전송한다.

3G 망을 통해서 오디오-텍스트게이트웨이로 수신된 전화통화 음성 데이터를 ACL 링크를 통해 블루투스 디바이스로 전송하기 위해서는 오디오 프레임 워크의 오디오레코드 클래스를 이용하여 우선 버퍼에 저장한다. 이때 버퍼에 저장되는 데이터 타입은 PCM 포맷으로 3G망에서 사용되는 CVSD(continuously variable slope delta) 코덱으로부터 트랜스코딩이 이루어진 상태이며, 이 데이터를 그대로 ACL 링크를 통해 전송한다. 오디오 버퍼를 사용하기 위해서는 우선 오디오 버퍼 공간을 할당받아야 하며, <표 2>와 같은 파라미터를 통해 최소한의 오디오 버퍼 크기와 오디오 소스 타입을 결정한다. 하드웨어 조건에 따른 오디오 파라미터 설정 값은 각 단말마다 다르기 때문에 응용을 구현할 때 그 값들을 체크해야 한다. 스트림 타입은 처리되는 오디오 스트림 타입을 결정하고, 샘플링율은 음성 데이터를 인코딩/디코딩할 때 사용하는 샘플링 주파수를 나타낸다. 채널 설정은 데이터 방향(ex. 입력, 출력)을 표시하고 모노와 스테레오를 구분한다. 오디오 포맷은 음성 데이터가 인코딩 될 때 쓰이는 단위 비트 수를 나타낸다. 그리고 오디오 소스는 음성 데이터를 지

장할 때 마이크 음성 데이터인지 상대방의 음성 데이터인지를 구분해준다. VOICE_CALL은 사용자 음성과 상대방 음성 모두를 선택할 수 있지만 안드로이드의 내부 구현상 버그로 인해 사용될 수 없다.

<표 2> 음성 데이터를 저장할 때와 출력할 때 버퍼 크기를 결정하기 위한 파라미터
<Table 2> Parameters for deciding audio buffer size for storing and rendering voice data

파라미터	값
Stream Type	STREAM_VOICE_CALL, STREAM_SYSTEM, STREAM_RING, STREAM_MUSIC, STREAM_ALARM
Sampling Rate(Hz)	44,100, 22,050, 11,025
Channel Configuration	CHANNEL_IN_MONO CHANEELE_IN_STEREO CHANNEL_OUT_MONO CHANEELE_OUT_STEREO
Audio Format	ENCODING_PCM_16BIT ENCODING_PCM_8BIT
Audio Source	VOICE_UPLINK, MIC VOICE_DOWNLINK VOICE_CALL(system bug)

(3) 블루투스 데이터 전송부

블루투스 마스터와 슬레이브가 소켓을 통해 연결이 되면 데이터를 전송할 수 있는데 입력 버퍼와 출력 버퍼의 크기를 정한 후에 블루투스 소켓의 입력 스트림과 출력 스트림의 객체를 통해 데이터를 전송한다. 입력 스트림의 read 메소드를 사용하여 원격 디바이스로부터 데이터를 수신하고 출력 스트림의 write 메소드를 사용하여 원격 디바이스로 데이터를 송신한다. 사용자의 정의에 따라 블루투스 데이터의 버퍼 크기를 정할 수 있지만 사용자의 음성 데이터를 수신 및 출력을 할 때 정해지는 최소한의 음성 데이터 버퍼의 크기를 반영하여 블루투스 송신 및 수신 버퍼의 크기를 결정한다.

(4) 블루투스 연결부

블루투스 통신을 수행할 때 각 디바이스는 서버와 클라이언트로 나누어지고 연결 요청과 연결 수락 동작을 한다. 필요에 따라 서버와 클라이언트는 모두 연결을 요청할 수 있기 때문이다[5]. 블루투스 디바

이스 클래스는 블루투스 장치의 UUID(Universally unique identifier)를 사용하여 블루투스 소켓을 생성할 수 있으므로 블루투스 데이터 통신을 수행하려는 디바이스는 블루투스 소켓을 생성하고 서버 역할을 하는 블루투스 장치에게 연결을 요청한다. 블루투스 어댑터 클래스는 서버 소켓을 생성할 수 있으므로 블루투스 장치의 UUID를 사용하여 블루투스 서버 소켓을 생성할 수 있다. 서버 역할을 하는 블루투스 장치는 원격 디바이스의 연결 요청이 들어올 때까지 대기하고 있다가 요청이 들어오면 연결을 수락하고 원격 디바이스의 정보를 저장한다.

2) 블루투스 디바이스 모듈의 구성

(1) 사용자 이벤트 처리부

이 처리부는 사용자에게 SMS, SNS 메시지를 표시하거나 오디오-텍스트 게이트웨이로부터 음성 통화 요청이 수신되었을 때 발신자 식별번호를 팝업창에 표시하면서 통화 수락을 위한 UI를 제공한다.

(2) 음성과 텍스트 데이터 처리부

음성과 텍스트 데이터 처리부는 오디오-텍스트 게이트웨이 모듈에서 수신한 음성 데이터와 텍스트 데이터를 사용자가 들을 수 있거나 볼 수 있게 스피커로 출력하거나 스크린에 표시한다. 음성 데이터의 경우, 음성 데이터를 스피커에 출력하기 위해 최소한의 데이터 이상이 오디오 버퍼에 버퍼링이 되어야 한다. 블루투스 디바이스 모듈의 오디오 저장과 출력의 파라미터 값은 오디오-텍스트 게이트웨이와 동기를 맞추기 위해 같은 값을 사용한다.

(3) 블루투스 데이터 전송부

블루투스 디바이스 모듈의 블루투스 데이터 전송부는 사용자 음성 데이터를 PCM 포맷으로 오디오-텍스트 게이트웨이로 송신하며 구체적인 처리과정은 오디오-텍스트 게이트모듈의 음성데이터 전송과정과 같다.

(4) 블루투스 연결부

오디오-텍스트 게이트웨이가 블루투스 연결 요청을 할 때 연결 수락을 하거나 블루투스 디바이스가 오디오-텍스트 게이트웨이로 연결 요청을 수행한다.

IV. 모듈 구현 및 기능 검증

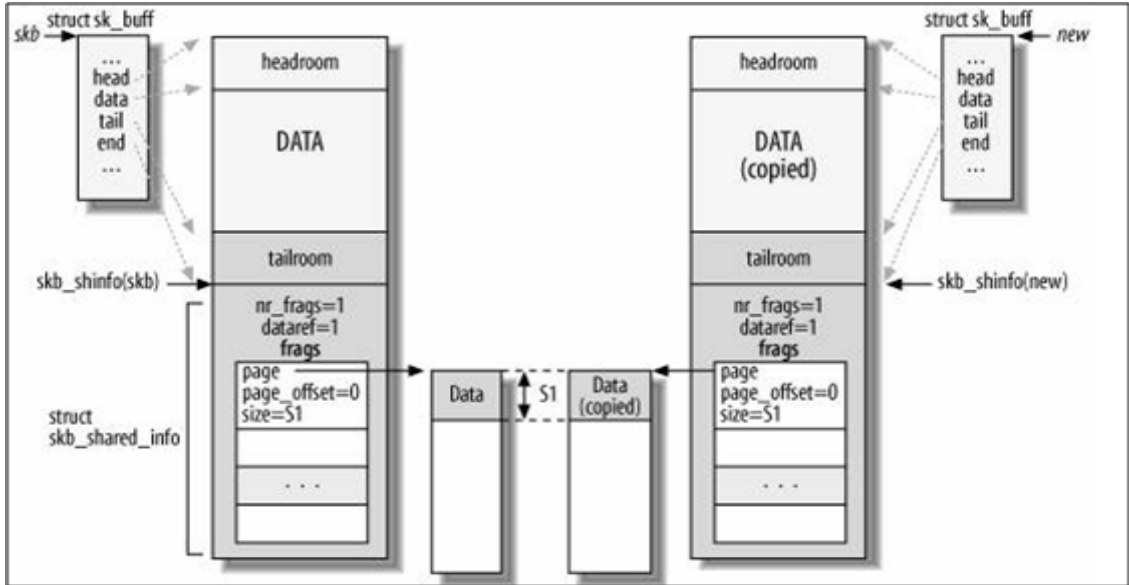
1. 모듈 구현

1) 음성 통화 및 SMS 수신 기능

안드로이드 기반의 블루투스 디바이스 응용은 오디오-텍스트 게이트웨이 모듈과 블루투스 디바이스 모듈로 구현하였다. 응용의 각 모듈은 사용자 인터페이스를 위한 XML 파일, 기능 동작을 위한 자바 파일, 안드로이드 컴포넌트들과 하드웨어 사용 퍼미션을 표시하는 안드로이드 매니페스트 파일로 구성한다[8]. 오디오-텍스트 게이트웨이 모듈과 블루투스 디바이스 모듈은 이클립스(버전 3.6 Helios) 환경에서 구현하였다. 오디오-텍스트 게이트웨이 모듈은 안드로이드 버전 2.3을 선택했고, 블루투스 디바이스 모듈은 안드로이드 버전 2.1을 선택하여 구현했다.

오디오-텍스트 게이트웨이와 블루투스 디바이스 간에 음성 데이터 전송을 위해 ACL 링크를 사용하는 목적은 오디오-텍스트 게이트웨이가 SCO 링크를 통해 음성 데이터를 블루투스 디바이스로 전송하지만 그 데이터의 스피커 출력을 할 수 없기 때문이다. 따라서 전화 음성 데이터를 PCM 포맷으로 변환하여 ACL 링크를 통해 전송한 후 블루투스 디바이스의 오디오 프레임 워크를 통해 출력하도록 한다.

오디오-텍스트 게이트웨이 모듈은 전화 이벤트 및 음성 저장, 오디오 하드웨어 셋팅, 블루투스 사용, SMS 사용을 위한 퍼미션을 포함한다. 두 디바이스 간에 블루투스 연결이 되어 있는지를 사용자에게 보여주는 사용자 인터페이스를 구현하였다. 디바이스의 내부 이벤트와 3G 망을 통해 수신된 음성 데이터를 임시 저장하는 기능과 임시 저장된 음성 데이터를 3G망으로 전송하기 위해 디바이스의 오디오 하드웨어 버퍼에 전달하는 기능을 구현하였다. 오디오 데이터를 저장할 때 모바일 단말의 오디오 하드웨어가 지원하는 설정 값에 따라 16bit PCM 인코딩 오디오 포맷, 모노 채널, 그리고 샘플링 주파수는 44,100Hz를 선택했다. 또한 3G 망을 통해 수신된 음성 데이터를 처리하기 때문에 VOICE_DOWNLINK를 오디오 소스로 선택하였다. 사용자의 명시적 이벤트에 따라 전화 이벤트



<그림 8> 소켓 버퍼 구조체를 이용한 패킷 복사
 (Fig. 8) Packet copy using socket buffer structure

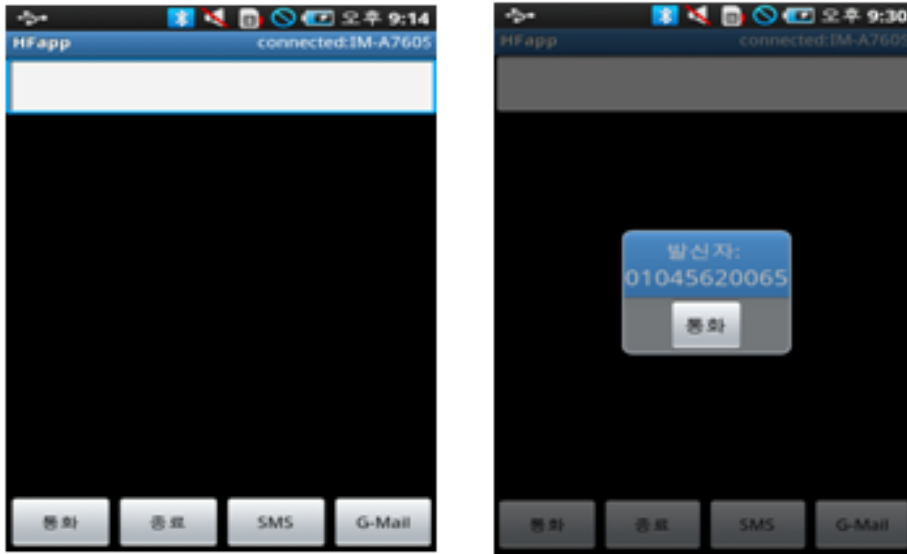
캡처, 발신자 식별번호 전송, SMS 데이터 전송을 구현하였다. 오디오-텍스트 게이트웨이가 인터넷 기반의 텍스트 데이터를 수신하기 위해 NDK에서 제공하는 리눅스 소켓 버퍼의 skbuff 구조체를 이용하여 네트워크 패킷 복사 모듈을 설계하였다. 그 모듈을 컴파일 하면 라이브러리 파일(.so 파일)이 생성되는데, 이 파일을 안드로이드 프로젝트에 연결하여 패킷 복사 기능을 사용하여 지메일의 제목을 임시 저장하여 블루투스 디바이스 모듈로 전송한다.

블루투스 디바이스 모듈은 음성 저장, 오디오 하드웨어 사용 권한 셋팅, 블루투스 사용을 위한 퍼미션을 포함한다. 블루투스 디바이스 모듈은 사용자와 직접적인 상호작용을 수행하기 때문에 두 디바이스간의 연결요청 및 수락과 연결 상태 표시, 전화 받기, 수신한 SMS 메시지 표시, SMS 작성 후 전송, 전화 걸기, 지메일 작성의 기능을 위한 사용자 인터페이스를 구현하였다. 또한 오디오-텍스트 게이트웨이로부터 수신된 음성 데이터를 스피커로 출력하는 기능, 마이크 음성을 임시 저장한 후 오디오-텍스트 게이트웨이 모듈로 전송하는 기능을 구현하였다. 오디오 버퍼 할당을 위한 파라미터 값으로는 오디오 데이터를 저장할 때 모바일 단말의 오디오 하

드웨어가 지원하는 설정 값에 따라 16bit PCM 인코딩 오디오 포맷, 모노 채널, 44,100Hz를 선택했다. 또한 마이크를 통해 전달된 음성 데이터를 처리하기 때문에 마이크를 오디오 소스로 선택한다. 수신된 SMS 메시지 표시 기능을 포함한다. 최소 음성 데이터 버퍼 크기는 오디오 프레임워크가 제공하는 getMinBufferSize 메소드에 의해 오디오-텍스트 게이트웨이 모듈 버퍼는 3,480 바이트, 블루투스 디바이스 모듈 버퍼는 16,384 바이트로 설정되었다.

2) 인터넷 기반의 패킷 복사

안드로이드의 리눅스 커널에서 네트워크 패킷을 캡처하는 방법은 다양하게 있지만 pcap 혹은 넷필터와 같이 구현의 복잡도가 높거나 안드로이드 커널 컴파일이 필요한 방법과는 달리 소켓버퍼를 사용하여 네트워크 패킷을 복사하는 방법이 있다. 소켓 버퍼는 skbuff(소켓버퍼) 구조체를 사용하여 각 계층의 헤더 정보와 데이터를 저장하며, 소켓 버퍼의 구조체들은 연결 리스트로 연결되어 있다. 본 연구의 인터넷 기반의 텍스트 데이터 수신 기능을 위해 <그림 8>과 같이 소켓 버퍼에서 사용하는 skbuff 구조체를 이용하고, 구현을 위해 NDK를 이용한 인터



<그림 9> 블루투스 디바이스의 UI (좌)오디오-텍스트 게이트웨이와 연결된 상태 표시 (우)음성 전화 수신 UI
 <Fig. 9> UI of bluetooth device (a)Display for connecting with audio-text gateway (b)UI of incoming call

넷 패킷 복사 라이브러리를 작성한다. 오디오-텍스트 게이트웨이의 인터넷 패킷 복사 함수가 이 라이브러리를 JNI(Java Native Interface)를 통해 호출한다.

안드로이드는 NDK를 통해 안드로이드 API 기능 보충 및 고급 기능을 제공한다. 관련 연구에서 설명한 네트워크 패킷을 복사하는 기능은 안드로이드 SDK를 통해 제공되지 않지만 NDK를 사용하면 리눅스 커널 코드의 skbuff 구조체에 접근하여 네트워크 패킷을 복사할 수 있으므로 인터넷 기반의 텍스트 데이터를 수신할 수 있다. 그러나 NDK의 skbuff.h는 skb_alloc()와 skb_copy()함수를 제공하지 않기 때문에 skbuff 구조체를 이용하여 이 함수들을 직접 구현해야 한다. skb_alloc() 함수는 패킷 복사를 위해 커널 메모리 공간을 할당하고, skb_copy() 함수는 skbuff 구조체의 각 필드 값을 통해 인터넷 패킷을 복사할 수 있다[8]. 커널 영역의 메모리 할당 함수는kmem_cache_alloc(), SKB_DATA_ALIGN(), kmalloc() 등이 사용된다[9]. skbuff 구조체를 위한 메모리 공간이 할당되면 시스템은 패킷의 포트 번호를 확인하여 특정 응용 프로그램으로 수신되는 패킷의 데이터 블록을 <그림 8>과 같이 복사할 수 있다. 패킷 복사를 위해 소켓 버퍼를 통한 패킷 클

론을 할 수 있지만, 본 연구에서는 수신된 패킷을 변형해야 되기 때문에 패킷 복사 기능을 사용한다. 마지막으로 네트워크를 통한 패킷 수신이 완료되면 반드시 skbuff 구조체를 위한 메모리를 해제해야 한다. 오디오-텍스트 게이트웨이는 수신된 패킷의 이메일 제목을 텍스트 데이터 형식으로 변환하여 ACL 링크를 통해 블루투스 디바이스로 전송한다.

2. 기능 검증

본 절에서는 안드로이드 환경에서 구현한 오디오-텍스트 게이트웨이와 블루투스 디바이스의 동작을 검증하기 위해 상대 전화가 오디오-텍스트 게이트웨이에게 전화를 걸었을 때와 SMS 메시지를 송신했을 때, 오디오-텍스트 게이트웨이를 통해 수신된 데이터를 블루투스 디바이스의 화면에 표시하는 것을 <그림 9>와 <그림 10>에서 보여주고 있다. 또한 지메일을 수신했을 때 noti피케이션 형태로 수신된 메일의 제목을 표시하는 것을 <그림 11>에서 나타낸다.

<그림 9(우)>는 상대방이 음성통화를 요청하면 오디오-텍스트 게이트웨이의 콜 상태가 RINGING으로 천이하면서 발신자 식별번호를 ACL 링크를 통



〈그림 10〉 SMS 메시지 수신 (a)블루투스 디바이스의 메시지 수신 (b)오디오-텍스트 게이트웨이 SMS 수신
 〈Fig. 10〉 SMS message receiving (a)Bluetooth device' message receiving (b)Audio-text gateway SMS receiving



〈그림 11〉 지메일 제목 표시
 〈Fig. 11〉 Displaying gmail's subject

해 블루투스 디바이스로 전송한 후 스크린에 팝업 창 형식으로 발신자 식별번호와 통화 버튼을 표시하도록 하는 것을 나타낸다. 사용자는 이 번호를 보고 전화 통화 할지를 판단한다. <그림 10>은 오디오-텍스트 게이트웨이가 SMS를 수신하면 블루투스 디바이스의 스크린에 오디오-텍스트가 수신한 SMS를 발신자 식별번호와 같이 표시한다. 오디오-텍스

트 게이트웨이가 지메일을 수신하면 인터넷 패킷 복사함수 동작을 통해 패킷의 애플리케이션 페이로드를 복사한 후 메일의 제목 부분과 발신자 ID를 <그림 11>과 같이 블루투스 디바이스의 화면에 팝업 창 형식으로 표시한다.

V. 결론 및 향후 계획

본 연구에서는 음성 통화 기능을 위주로 제공하는 블루투스 헤드셋과 핸드프리와는 달리 차량에서 운전자와 승객을 위해 인터넷 기반의 텍스트 데이터 화면 표시 기능을 강화한 안드로이드 기반의 블루투스 디바이스 응용 시스템을 제안한다. 제안하는 응용 시스템을 위해 이클립스를 사용하여 안드로이드 기반의 디바이스에 응용 소프트웨어를 오디오-텍스트 게이트웨이 모듈과 블루투스 디바이스 모듈로 설계 및 구현하였다. 기능 검증을 위해 안드로이드 기반의 스마트폰에 제안하는 응용 시스템을 안드로이드 앱 형태로 설치하여 음성 통화 기능, SMS 메시지 수신 기능 및 지메일 제목표시 기능을 그림으로 보여줬다. 전화와 SMS를 수신할 때 오디오-텍스트 게이트웨이 모듈은 3G망을 통해 수신한

음성 데이터 및 SMS를 ACL 링크를 통해 전송하면 블루투스 디바이스는 발신자 식별번호 및 SMS와 같은 텍스트 데이터를 화면 표시하거나 음성 데이터를 스피커로 출력한다. 또한 오디오-텍스트 게이트웨이가 인터넷 패킷 복사 함수를 통해 지메일을 수신했을 때 수신된 메일의 제목을 블루투스 디바이스의 화면에 표시했다.

두 모듈을 구현하여 기능을 검증한 결과 인터넷 기반의 텍스트 데이터와 음성 데이터를 전송할 때 지연 시간이 발생하지만 음성 통화 기능과 텍스트 데이터 수신 및 화면 표시 기능을 모두 제공할 수 있다. 기능을 검증해 본 결과 본 연구는 디스플레이가 탑재된 안드로이드 기반의 디바이스를 차량에서 운전자와 승객들을 위한 텍스트 데이터의 화면 표시 기능이 강화된 블루투스 디바이스로 사용할 수 있다. 이 블루투스 디바이스를 통해 운전자와 승객들은 차량 이동 중에도 안전하고 편리하게 음성 통화 기능과 텍스트 데이터 표시 기능을 사용할 수 있다. 추후 과제는 다양한 SNS 응용을 제안하는 블루투스 디바이스에 설치하여 사용자가 원하는 텍스트 데이터를 차량 이동 중에 화면 표시하는 것이다.

참 고 문 헌

- [1] 고덕영, 정동규, “블루투스 기술을 이용한 핸드즈 프리용 헤드셋과 베이스 개발,” *한국정보기술학 회논문지*, vol. 6 no. 3, pp.8-14, 2008. 6.
- [2] Kelly Montgomery, “I’m Watch gives you Android on your wrist, connects to your smartphone,” *Digital Trends*, 2011. 11.
- [3] 김정훈, “구글의 안드로이드 프로그래밍” 성안당, 2009.
- [4] Frank Bellosa, Jan Stob and Konrad Miller, “Analysis of the Android Architecture,” pp.7-16. 2010. 06.
- [5] Bruce Hopkins, “Bluetooth for Java,” Apress, 2003.
- [6] 임채권, “Adhoc Networking을 위한 블루투스 스캐터넷 형성 프로토콜,” 석사학위논문, 서울대학교, 2003.
- [7] Bluetooth Specification, “Specification of the bluetooth system,” Bluetooth SIG group, 2004. 11.
- [8] 한동훈, 원일용, 하홍준, “리눅스 커널 프로그래밍,” 한빛미디어, 2010. 09.
- [9] Christian Benvenuti, “Understanding Linux Network Internals,” O’Reilly, 2005.

저자소개



조 효 성 (Cho, Hyosung)

2008년 : 광운대학교 전자공학과(학사)

2012년 : 광운대학교 대학원 임베디드S/W학과(석사과정)



이 혁 준 (Lee, Hyukjoon)

1987년 : University of Michigan, Computer Science(학사)

1989년 : Syracuse University, Computer Science(석사)

1993년 : Syracuse University, Computer Science(박사)

1994년 ~ 1996년 : 삼성전자(주) 멀티미디어 연구소 선임연구원

1996년 ~ 현 재 : 광운대학교 컴퓨터공학과 교수