creative commons

COMMONS DEED

저작자표시-비영리-동일조건변경허락 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.
- 이차적 저작물을 작성할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

동일조건변경허락. 귀하가 이 저작물을 개작, 변형 또는 가공했을 경우에는, 이 저작물과 동일한 이용허락조건하에서만 배포할 수 있습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

[UCI]I804:24011-200000263700

2 0 1 3년 2월
석사학위논문

스마트폰 착신번호 자동송신을 위한 애플리케이션 구현

조 선 대 학 교 대 학 원

정보통신공학과

김 서 궁

스마트폰 착신번호 자동송신을 위한 애플리케이션 구현

Implementation of application for automatic transmission of incoming phone numbers on smart phones

2013년 2월 25일

조 선 대 학 교 대 학 원

정보통신공학과

김 서 궁

스마트폰 착신번호 자동송신을 위한 애플리케이션 구현

지도교수 박 종 안

이 논문을 공학석사학위신청 논문으로 제출함

2012년 10월

조 선 대 학 교 대 학 원

정보통신공학과

김 서 궁

김서궁의 석사학위 논문을 인준함

위원장 조선대학교 교수 변 재 영 (인)

위 원 조선대학교 교수 최 광 석 (인)

위 원 조선대학교 교수 박 종 안 (인)

2012년 11월

조 선 대 학 교 대 학 원

목 차

목 차 ····· i
그 림 차 례 ····· iii
표 차 례 ····· v
ABSTRACT ····· vi

1. 서 론 ····· 1

1.1 연구 목적 ····· 1
1.2 연구 동향 ····· 2
1.3 연구 배경 ····· 2
1.4 연구 내용 ····· 3

2. 안드로이드 플랫폼 ····· 5

2.1 안드로이드(Android) ····· 5

2.1.1 안드로이드 아키텍쳐 ····· 6
2.1.2 안드로이드 구성 및 특징 ····· 8
2.1.3 안드로이드 점유율 ····· 10

2.2 안드로이드 발전과정 ····· 11

2.2.1 안드로이드 SDK 1.x 버전 ····· 11
2.2.2 안드로이드 SDK 2.x 버전 ····· 12
2.2.3 안드로이드 SDK 3.x 버전 ····· 13

2.2.4 안드로이드 SDK 4.x 버전 ····· 13

2.3 안드로이드 컴포넌트 ····· 14

2.3.1 액티비티(Activity) ····· 14
2.3.2 서비스(Service) ····· 17
2.3.3 컨텐트 프로바이더(Content Provider) ····· 18
2.3.4 브로드캐스트 리시버(Broadcast Receiver) ····· 19

3. 착신번호 자동송신 시스템 설계 및 제작 ····· 21

3.1 착신번호 자동송신 시스템의 필요성 및 기본구조 ····· 21
3.2 착신번호 자동송신 애플리케이션 ····· 21

3.2.1 스마트폰 착신정보 자동전송 ····· 23
3.2.2 안드로이드 블루투스 데이터 통신 ····· 24

3.3 중계기 ····· 28

3.3.1 중계기 필요성 ····· 28
3.3.2 중계기 설계 ····· 29

4. 제안하는 시스템 구현 및 평가 ····· 31

4.1 애플리케이션 구현 ····· 31
4.2 중계기 ····· 34
4.3 착신번호 자동송신 시스템 구현 ····· 36

4.4 성능평가 결과 ····· 38

5. 결론 ····· 43

참고문헌 ····· 45

그 림 목 차

그림 2-1 안드로이드 아키텍처 ····· 6
그림 2-2 안드로이드 애플리케이션 프레임워크 ····· 7
그림 2-3 안드로이드 라이브러리와 런타임 ····· 7
그림 2-4 안드로이드 리눅스 커널 ····· 8
그림 2-5 안드로이드 액티비티 생명주기 ····· 15
그림 2-6 안드로이드 액티비티 상태변화 ····· 16
그림 2-7 안드로이드 서비스 생명주기 ····· 17
그림 2-8 안드로이드 컨텐트 프로바이더 ····· 19
그림 2-9 안드로이드 브로드캐스트 리시버 ····· 20
그림 3-1 알리미 애플리케이션 생명주기 ····· 22
그림 3-2 알리미 애플리케이션 플랫폼 순서도 ····· 23
그림 3-3 PhoneStateListener 상태구별 소스코드 ····· 24
그림 3-4 안드로이드 블루투스 아키텍쳐 ····· 25
그림 3-5 블루투스 퍼미션 선언 ····· 26
그림 3-6 중계기 블록도 ····· 29
그림 4-1 블루투스 기능 활성화 ····· 31
그림 4-2 디바이스 검색 ····· 32
그림 4-3 중계기 연결 ····· 33
그림 4-4 착신전화 정보전송 ····· 34
그림 4-5 블루투스 모듈 ····· 35
그림 4-6 지그비 모듈 ····· 35
그림 4-7 중계기 인터페이스 보드 ····· 36
그림 4-8 중계기 ····· 36
그림 4-9 디스플레이 ····· 37
그림 4-10 블루투스 기반 착신번호 자동전송 시스템 ····· 37
그림 4-11 블루투스 통신과 중계기 통신비교 실험 ····· 42

표 목 차

표 2-1 안드로이드 구성 및 특징 ····· 9
표 2-2 안드로이드 SDK 1.x 버전 ····· 11
표 2-3 안드로이드 SDK 2.x 버전 ····· 12
표 2-4 안드로이드 SDK 3.x 버전 ····· 13
표 2-5 안드로이드 SDK 4.x 버전 ····· 13
표 4-1 블루투스 모듈 규격 ····· 39
표 4-2 지그비 모듈 규격 ····· 39
표 4-3 스마트폰 수납공간 재질에 따른 통신가능여부 및 거리 실험 ····· 40
표 4-4 블루투스와 중계기 통신비교 실험 ····· 41

ABSTRACT

Implementation of application for automatic transmission of incoming phone numbers on smart phones

SeoGung Kim

Advisor : Prof. Jong-An Park Ph.D.

Department of Information and communications Engineering

Graduate School of Chosun University

This thesis is intended to explain Bluetooth-based incoming numbers automatic transmission system by implementing an application, which automatically extracts the number of an incoming call as a smart phone, with Android OS, receives a call. It then automatically transmits the incoming number by means of the built-in Bluetooth and relay station, which converts Bluetooth signals into ZigBee signals. This helps to solve the weak points of Bluetooth and makes it possible to communicate with up to 255 devices and expand its possible communication range.

The application in this work was developed using Android development tools in order for Android smart phones to extract the incoming number from incoming information and automatically transmit the number. After the application is executed it is in the ready state until any incoming call occurs. When any incoming information is receieved, the application extracts its incoming number from the information and then automatically transmits the number to a relay station through Bluetooth communication.

To solve the weak points of Bluetooth, such as a short communication distance, the limited number of communication devices (up to seven), the sensitivity to surroundings, the relay station converts the Bluetooth signals to ZigBee signals. It is therefore possible to communicate with up to 255 devices and expand its communication range because the converted signals are less sensitive to surroundings.

As described earlier, this research implemented and verified Bluetooth-based

incoming number automatic transmission system that consists of an application to automatically handle the incoming calls received by smart phones, and a relay station to convert Bluetooth signals into ZigBee signals.

1. 서론

1.1 연구 목적

현대의 통신기기 발달로 가장 큰 목적이 전화 통화 및 메시지 전송에 불과했던 휴대폰은 전화통화와 메시지 전송뿐만 아니라 인터넷, 게임, 미디어 매체 감상, 블루투스 통신 등의 다양한 기능이 내장된 스마트폰이라는 활용성이 높은 이동통신기기로 급부상하고 있다. 기존의 휴대폰을 스마트폰으로 교체하여 사용하는 사용자가 늘어나고 스마트폰뿐만 아니라 태블릿PC와 같은 스마트기기의 사용자가 급격히 증가하면서 스마트폰과 더불어 다양한 스마트기기와 관련한 사회적인 관심과 경제적 발전가능성에 대한 기대로 세계적인 기업들과 개발자들의 관심 및 연구가 이루어지고 있다.

급격히 발달하고 있는 다양한 모바일 플랫폼을 적용한 스마트폰들은 전 세계적으로 증가하고 있는 추세이며 컴퓨터를 사용해야 이용할 수 있던 인터넷 통신과 정보검색 등이 가능해지고 사용자가 원하는 기능을 지닌 애플리케이션을 설치 할 수 있는 특징을 지니고 있어 사용자들은 마치 간단한 컴퓨터처럼 기기를 사용할 수 있다 [1].

모바일 플랫폼 종류에는 iOS, Android, Windows, Symbian, Black Berry 등이 있다. 이와 관련된 스마트폰 제조사들은 다양한 단말기를 출시하고 있으며 각 회사에서 제조하는 단말기의 최적화된 플랫폼 개발과 많은 연구로 타 회사들의 단말기와 사용하는 플랫폼에 경쟁하고 있다. 특히 타 플랫폼에 비하여 많은 장점을 지닌 안드로이드는 구글(Google)에서 오픈 플랫폼으로 무료로 배포하고 있어 라이선스의 제약 없이 사용가능하다 [2]. 그리고 휴대전화뿐만 아니라 특수 목적으로 개발 가능한 임베디드 환경에서도 적용이 가능하도록 안드로이드 응용프로그램을 동작 시키는 Dalvik VM은 저수준의 임베디드 환경을 고려하여 개발되었기 때문에 적은 메모리를 가지고 있어 폭넓고 다양한 분야에서 사용 가능할 것으로 기대하고 있다 [3].

1.2 연구 동향

스마트폰 애플리케이션의 여러 분야에서 개발 가능하다는 장점으로 안드로이드를 기반으로 하는 대표적인 근거리 통신인 블루투스를 이용하는 다양한 애플리케이션이 개발되고 있다. 블루투스 통신은 케이블을 제거하여 무선으로 사용하면서 저전력과 저가의 근거리 통신 규약으로 탄생되었다 [4].

무선으로 저전력과 저가의 근거리 통신이라는 강한 장점으로 블루투스 통신을 이용한 다양한 애플리케이션이 개발되고 있다. 예를 들어 스마트폰에 설치한 애플리케이션을 통한 RC카 제어, 기존 청진기의 단점인 짧은 거리제약과 심박음의 높은 주파수 대역의 소리를 놓치는 문제점을 개선한 무선청진기기와 청진기에서 나오는 소리를 듣고, 저장과 전송까지 가능한 블루투스 통신기반 스마트폰 무선청진기 애플리케이션 등 다양한 분야에서 활용 가능한 기술들이 있다 [5-6].

또한 요즘 시대에는 정보통신을 기반으로 언제 어디서나 무선으로 네트워크 연결이 가능하고 이용이 가능하고, 사용자에게 편의성을 제공하는 유비쿼터스(ubiquitous) 환경을 지향하고 있다. 근거리 통신으로 사용되는 블루투스 통신방법을 이용하여 획득한 정보를 무선으로 전송, 수신할 수 있는 유비쿼터스에 입각한 애플리케이션 및 기술들의 증가는 앞으로도 여러 분야로 개발 가능성이 기대가 되므로 이에 대한 연구를 필요로 한다.

1.3 연구 배경

근거리 통신 기술인 블루투스 통신은 저전력과 저가로 이용 가능하며, 통신기기 간 케이블 연결 없이 무선으로 2.4GHz대의 ISM 밴드를 이용하여 수신데이터를 송수신하면서 10m 이내의 근거리에 있는 둘 이상의 통신장치 사이에서 데이터를 전송하기 위한 기술적 세부 사항을 명시한 라이선스가 없는 개방 표준 규격이다. 블루투스 통신이 가

능한 기기와 스마트폰에 내장된 블루투스 통신 기능을 제어하는 애플리케이션을 설치하고 스마트폰을 이용하면 사용자의 편의성을 크게 향상 시킬 수 있다.

블루투스 통신을 이용한 애플리케이션으로는 RC카 제어로 시작하여 실제 자동차와의 통신으로 제어를 할 수 있는 가능성을 보이며, 사용자에게 있어서 특정 환자 혹은 진찰을 받고자 하는 일반 환자들에게 보다 정확한 진찰과 편한 진료를 가능하도록 개발한 무선청진기 애플리케이션 등 스마트폰을 사용하는 사용자들에게 편리함을 제공할 수 있는 애플리케이션 개발이 필요하다.

거리적 제한으로 스마트폰의 착신정보를 확인 할 수 없는 상황이 발생하는 사물함과 같은 곳에 스마트폰을 보관해야 하는 경우 사용자는 스마트폰을 소지하지 않고도 스마트폰으로 수신되는 정보를 확인할 수 있는 시스템을 개발 하면 사용자는 스마트폰을 소지하지 않아도 스마트폰의 수신되는 정보를 디스플레이를 통해 확인하고 짧은 시간에 스마트폰 착신정보로 인한 문제를 해결할 수 있다.

따라서, 본 연구는 스마트폰 사용자가 블루투스 통신을 이용하여 착신정보를 자동으로 데이터화 하여 전송할 수 있는 기능의 안드로이드 운영체제기반 애플리케이션을 설계하고, 블루투스 통신의 단점인 거리적 제한과 7대까지 통신이 제한되는 것을 지그비 통신으로 통신거리를 확장하며, 255대까지 통신이 가능하게 하고, 블루투스 보다 저전력과 저가의 지그비 통신을 이용할 수 있도록 블루투스 통신과 지그비 통신의 신호변환 중계기를 설계한다. 스마트폰에 애플리케이션을 설치하고 애플리케이션이 실행하면 착신정보를 데이터화하여 블루투스 통신을 이용해 중계기로 데이터를 보내고 중계기는 블루투스 통신을 지그비 통신으로 변환하여 디스플레이 장치에서 수신정보를 확인할 수 있도록 설계하였다.

1.4 연구 내용

본 연구의 구성은 다음과 같다. 제1장의 서론에 이어 제2장에서는 안드로이드 기본 내용과 안드로이드 프레임워크의 발전해온 과정, 안드로이드의 아키텍쳐, 컴포넌트를

통한 애플리케이션을 설계하기 위한 기본 내용을 살펴본다. 제3장은 블루투스 기반 착신번호 자동송신 시스템의 설계 및 제작을 설명하였다. 제4장에서는 설계한 시스템의 시뮬레이션 구현 및 성능을 제시하였고, 마지막 제5장에서 결론을 기술하였다.

2. 안드로이드 플랫폼

본 장에서는 설계하고자 하는 블루투스 통신을 이용한 스마트폰 착신정보 자동전송 애플리케이션 연구의 기반이 되는 안드로이드 프레임워크를 알아본다. 안드로이드 SDK 버전의 발전해온 과정과 안드로이드란 무엇인지 살펴보고, 안드로이드 애플리케이션을 개발하기 위한 안드로이드 컴퍼넌트의 구성요소 4가지인 액티비티(Activity), 서비스(Service), 컨텐트 프로바이더(Content Provider), 브로드캐스트 리시버(Broadcast Receiver)를 알아본다.

2.1 안드로이드(Android)

안드로이드는 스마트폰과 태블릿 PC 등의 휴대용 스마트기기를 사용하기 위해 설치하는 운영체제(OS)와 미들웨어, 사용자 인터페이스, 웹 브라우저, 이메일 클라이언트, 단문 메시지 서비스(SMS), 멀티미디어 메시지 서비스(MMS) 등을 포함하고 있는 모바일 운영체제이다. 자바 프로그래밍 언어를 사용하여 응용 프로그램을 작성할 수 있는 안드로이드는 컴파일된 바이트코드를 구동할 수 있는 런타임 라이브러리를 제공한다. 또한 SDK(Software Development Kit)를 통해 각종 도구들과 API(Application Programming Interface)를 제공하여 응용프로그램을 개발할 수 있다.

리눅스 커널을 기반으로 동작하는 안드로이드는 C/C++ 라이브러리를 포함하고 있다. 기존의 자바 가상머신과는 다른 달빅(Dalvik) 가상머신을 사용하여 자바 프로그래밍으로 개발된 응용 프로그램을 별도의 프로세스에서 실행하는 구조로 되어 있다.

2005년 안드로이드 사를 구글에서 인수한 후 2007년 11월 안드로이드 플랫폼을 휴대용 장치 운영체제로서 무료공개로 발표했고, 그 후 48개의 하드웨어, 소프트웨어, 통신회사가 모여 만든 OHA(Open Handset Aliance)에서 공개 표준을 위해 개발하고 있다. 구글은 기업이나 사용자가 안드로이드 프로그램을 독자적으로 개발을 해서 탑재할 수 있도록 모든 소스코드를 오픈 소스라이선스인 아파치 v2 라이선스로 배포하고 있으며,

제조사 혹은 통신사별 응용 프로그램 마켓이 함께 운용되고 있고, 유료 및 무료 응용 프로그램이 마켓에서 제공되고 있다 [7].

2.1.1 안드로이드 아키텍처

안드로이드는 모바일 장치를 위해 제안하는 운영체제와 미들웨어, 핵심 애플리케이션을 포함하고 있는 모바일 디바이스를 위한 소프트웨어 스택구조로 구글과 OHA가 함께 제안하였다.

안드로이드 전체적인 플랫폼은 그림 2-1과 같이 구성 된다 [8].

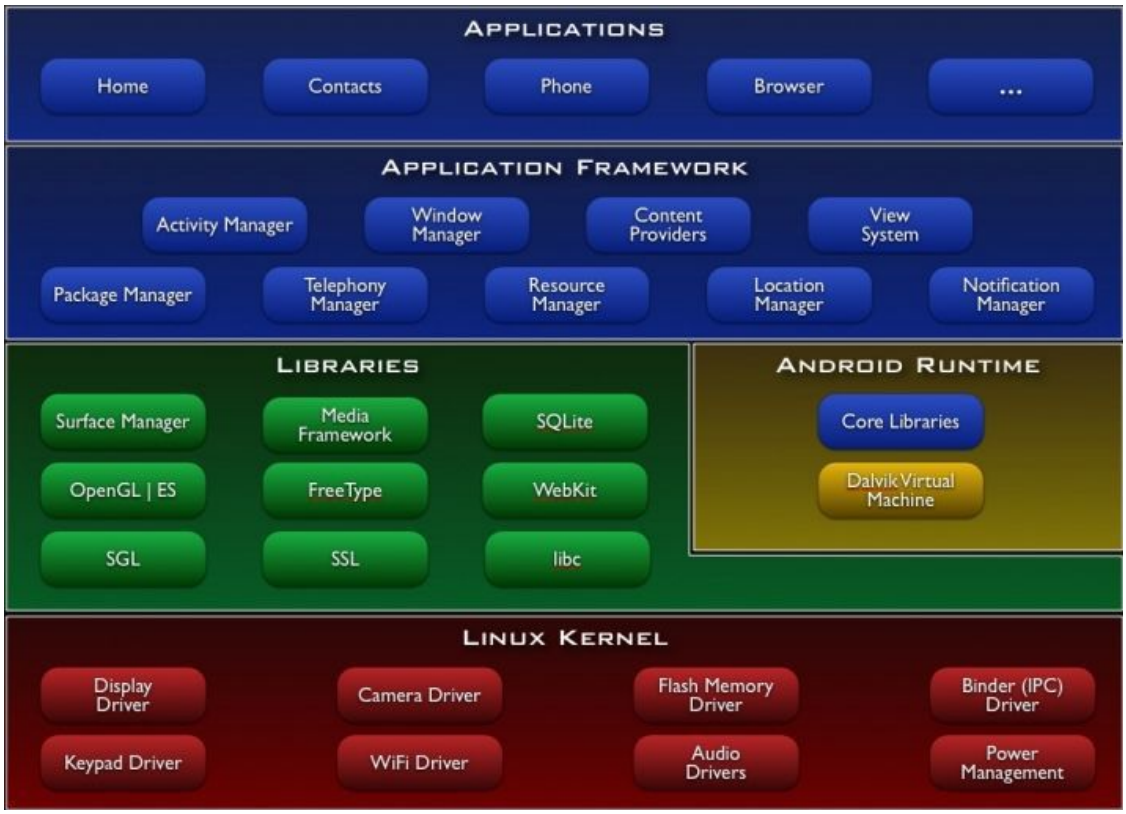

그림 2-1. 안드로이드 아키텍처 [8]

그림 2-2의 애플리케이션 프레임워크는 각 매니저들로 구성된 애플리케이션을 실행한다. 각 애플리케이션은 다음과 같다.

그림 2-2. 안드로이드 애플리케이션 프레임워크 [8]

- Activity Manager : 애플리케이션의 생명주기를 제어
- Content Provider : 애플리케이션 간의 공유되는 정보를 요약
- Resource Manager : 리소스는 프로그램에서 코드를 제외한 모든 부분
- Location Manager : 안드로이드 폰은 항상 자신의 위치를 파악
- Notification Manager : 메시지 도착 등이 사용자에게 방해되지 않도록 전달

그림 2-3의 데이터 저장을 위한 SQLite, 2D 및 3D 그래픽 레이어 들을 관리하는 Surface, 오디오와 비디오 포맷 관리 및 지원하는 Media Framework, SGL, 3D 지원을 위한 OpenGL ES, FreeType, libc으로 구성된 라이브러리와 애플리케이션을 실행하고 자바 프로그래밍 언어의 핵심 라이브러리의 사용가능한 기능을 포함하는 핵심라이브러리를 구성하는 런타임으로 이루어졌다.

그림 2-3. 안드로이드 라이브러리와 런타임 [8]

- Libraries : 그림 4와 같이 안드로이드 플랫폼이 적용된 핸드폰에 탑재
- Dalvik VM : 구글에서 만든 낮은 메모리에 최적화된 자바 VM

다음으로 그림 2-4와 같이 리눅스 커널은 리눅스 버전 2.6 이상에 의존하여 메모리 관리, 프로세스 관리, 네트워크 스택, 그리고 드라이버 모델과 통신전용 블루투스와 와이파이 등의 서비스를 담당한다. 또한 하드웨어와 안드로이드 플랫폼 스택의 나머지 계층 간의 추상화 된 계층 역할을 리눅스 커널에서 담당한다.

그림 2-4. 안드로이드 리눅스 커널 [8]

2.1.2 안드로이드 구성 및 특징

안드로이드는 리눅스커널을 사용하고 있으며, 오픈 GL의 휴대용기기용 버전인 GL ES를 그래픽 엔진으로 채택하였다. 웹 렌더링은 웹키트 엔진을 사용하고, SSL 보안모듈을 사용하고 있다. 개발환경은 이클립스와 이클립스에 통합된 플러그인 ADT(Android Development Tools) 사용을 권장하고 있다 [9].

표 2-1. 안드로이드 구성 및 특징 [9]

구성 요소	구성 내용
핸드셋 레이아웃	▪ VGA, 2D 그래픽스 라이브러리, OpenGL ES 1.0에 기반 한 3D 그래픽스 라이브러리를 확장하기에 적응적인 플랫폼
저장소	▪ SQLite 데이터베이스 소프트웨어 사용
통신	▪ GSM/EDGE, CDMA/ EV-DO, UMTS, 블루투스, 와이파이를 포함한 커넥션 기술 지원
메시징	▪ SMS, MMS 사용가능
웹 브라우저	▪ 오픈 소스인 웹키트 응용 프로그램 프레임워크 기반의 브라우저 지원
자바 지원	▪ 자바로 작성된 소프트웨어는 달빅 가상 머신에서 실행 가능한 코드로 컴파일
미디어 지원	▪ H.263, H.264(3GP 또는 MP4 컨테이너), MPEG-4 SP, AMR, AMR-WB(3GP 컨테이너), AAC, HE-AAC(MP4 또는 3GP 컨테이너), MP3, OGG Vorbis, WAV, JPEG, PNG, GIF, BMP 지원
하드웨어 지원	▪ 카메라, 터치스크린, GPS, 가속도 센서, 나침반 센서, 트랙볼 2D 그래픽 가속, 3D 그래픽 가속 활용 가능
개발 환경	▪ 에뮬레이터, 디버깅 도구, 이클립스 IDE 플러그인(ADT), 플랫폼 개발 키트(PDK)
마켓	▪ 무선으로 대상기기 하드웨어로 다운로드와 설치 가능한 응용프로그램 목록을 제공하는 구글플레이 ▪ 유료 애플리케이션 제공으로 별도의 라이선싱, 애플리케이션 내에서의 유료 구매를 위한 SDK 제공
멀티 터치	▪ 네이티브로 멀티 터치 지원
블루투스	▪ 블루투스 통신을 이용한 파일 전송은 버전 2.0부터 추가
영상통화	▪ UI 및 프로그램을 추가하여 사용 가능 ▪ HTC 자체의 센스(Sense) UI를 사용하여 영상통화 지원 ▪ 허니콤부터 GTalk를 통한 영상통화를 지원

표 2-1은 안드로이드의 구성 및 특징에 대하여 각각의 구성 요소에 설명하고 있다.

2.1.3 안드로이드 점유율

2011년 2분기부터 2012년 2분기까지 기준 안드로이드의 점유율은 2012년 2분기 기준으로 출하된 스마트폰 중 68%가 안드로이드 계열 스마트폰이었으며, 안드로이드 스마트폰 중 44%가 국내 삼성사의 제품이 차지했다. 안드로이드는 iOS의 새로운 스마트폰이 출시되는 시기에 따라 약간의 변동이 있을 뿐 스마트폰에 사용되는 운영체제 중 가장 큰 비중을 차지하고 있다 [9].

2.2 안드로이드 발전과정

안드로이드는 애플리케이션 개발을 하기 위해 자바 프로그래밍 언어를 사용하며 개발에 필요한 도구들과 API를 제공한다. 자바 프로그래밍의 기술은 크게 세 가지로 나뉘며, 자바 애플리케이션을 위한 J2SE(Java to Standard Edition)과 기업 솔루션을 위한 J2EE(Java to Enterprise Edition) 기술, 임베디드 환경을 위한 J2ME(Java to Micro Edition) 기술이다. J2SE와 J2EE의 기술은 호환되는 반면 J2ME는 호환이 되지 않는다. 이러한 조건으로 J2SE 기반의 기술 제공으로 안드로이드 애플리케이션 개발이 가능하다.

안드로이드 운영체제 버전역사는 2007년 11월의 안드로이드 1.0 베타 버전으로 시작하여 구글(Google)과 OHA(Open Handset Alliance)가 개발한 모바일 운영체제로 최초 배포판 이후로도 꾸준히 업데이트 되고 있다. 2009년 4월부터 배포되는 안드로이드 버전의 각각의 코드네임은 디저트 이름을 바탕으로 개발 되고 있다. 안드로이드 시험판 버전으로 나왔던 애스트로(Astro)와 벤더(Bender)는 상표권 문제로 사용되지 않았으며, 그 이후 나온 버전들은 알파벳 순서대로 컵케이크(Cupcake), 도넛(Donut), 이클레어(Eclair), 프로요(Proyo), 진저브레드(Gingerbread), 허니콤(Honeycomb), 아이스크림 샌드위치(Ice Cream Sandwich)로 이름을 붙였다 [10].

2.2.1 안드로이드 SDK 1.x 버전

표 2-1은 안드로이드 OS의 첫 버전인 애플파이로 시작하여 1.6 버전의 도넛까지의 안드로이드 초기 개발 과정이다. 초기 1.1, 1.2 버전은 HTC사의 G1을 개발하는데 사용되며 수정되어 왔다. 2009년 4월을 기점으로 안드로이드 사업이 활성화 되면서 1.5 버전인 컵케이크를 최초 정식 버전으로 인정하며 발표 하였고 1.2 버전에서 1.5 버전을 거치며 가상 키보드, 음성인식, 풀 스크린 위젯, 홈 스크린 등이 추가되며 한국어 지원도 시작되었다. 2009년 9월 1.6 버전인 도넛이 출시되면서 다양한 스크린 사이즈, 제스처 API, 통합검색기능, 카메라와 동영상 UI의 업데이트와 CDMA가 지원되기 시작하며, 국내 LG U+사에서도 안드로이드 폰을 출시했다.

표 2-2. 안드로이드 SDK 1.x 버전 [10]

버전	이름	리눅스커널	버전별 특징
1.0 API Level 1	애플파이 (Apple Pie)		▪ 안드로이드 OS의 첫 버전
1.1 API Level 2	바나나 브레드 (Banana Bread)		▪ 구글 지도에 상세정보 및 리뷰 추가 ▪ MMS 메시지의 첨부파일 저장지원
1.5 API Level 3	컵케이크 (Cupcake)	2.6.27	▪ 캠코더 모드(camcoder mode)를 통한 동영상 녹화 및 재생 ▪ 기기를 통한 동영상 업로드 및 그림 업로드 가능 ▪ 문자 예측 기능을 제공하는 새로운 소프트 키보드 추가 ▪ 블루투스 A2DP 및 AVRCP 지원 ▪ 일정거리 안에서 블루투스 헤드셋으로의 자동접속
1.6 API Level 4	도넛 (Donut)	2.6.29	▪ 카메라, 캠코더, 갤러리 인트페이스 내장 ▪ CDMA, EVDO, 802.1x, VPNs, 제스처, 문자열 음성변환 엔진기술지원 업데이트

2.2.2 안드로이드 SDK 2.x 버전

표 2-3과 같이 2009년 10월 2.0, 2.1 버전인 이클레어가 출시되면서 블루투스 2.1과 멀티터치를 지원하게 되었다. 또한 HTML5 지원과 멀티미디어 재생기능을 향상되었으며, 2010년 5월 2.2 버전인 프로요가 나오면서 애플리케이션 실행속도가 2~5배 정도 증가 하고, 메모리 회수기능의 개선으로 애플리케이션간의 전환 속도가 빨라졌다. 프로요 버전은 애플리케이션을 외장 메모리에 설치가 가능하게 되었고, 핫스팟 및 테더링을 지원하게 되었다. 그리고 2010년 12월 스마트폰의 안드로이드 OS 버전 중 가장 많은 점유율을 가진 진저브레드 버전이 출시되면서 NFC지원, 자이로스코프 지원 등의 업그레이드가 진행 되었다.

표 2-3. 안드로이드 SDK 2.x 버전 [10]

버전	이름	리눅스커널	버전별 특징
2.0 API Level 5 2.0.1 API Level 6 2.1 API Level 7	이클레어 (Eclair)	2.6.29	▪ 블루투스 2.1지원 ▪ 멀티터치 지원
2.2 2.2.1 2.2.2 API Level 8	프로요 (Froyo)	2.6.32	▪ USB 테더링 및 와이파이 핫스팟 기능 추가 ▪ 블루투스를 이용한 음성통화 및 연락처 공유기능 추가 ▪ 확장메모리에 응용 프로그램 설치 가능 ▪ Adobe Flash Player 10.1 지원
2.3 API Level 9 2.3.2 2.3.3 2.3.4 2.3.5 2.3.6 2.3.7 API Level 10	진저브레드 (Gingerbread)	2.6.35	▪ 인터넷 전화 지원(SIP) ▪ NFC(near-field Communication) 태그를 식별기능 지원 ▪ VP8 및 WebM, AAC, AMR 코덱지원 ▪ 자이로스코프, 회전 벡터, 선형 가속, 중력, 기압계 센서에 대한 정보처리기능 추가

2.2.3 안드로이드 SDK 3.x 버전

표 2-4에서 표기된 3.0 버전의 허니콤은 스마트폰 전용이 아닌 태블릿PC 전용으로 출시되었다. 태블릿PC 전용에 맞춘 UI 및 위젯 등이 개선되었지만, 호환성의 문제로 성과가 미약한 버전이다.

표 2-4. 안드로이드 SDK 3.x 버전 [10]

버전	이름	리눅스커널	버전별 특징
3.0 API Level 11 3.1 API Level 12 3.2 API Level 13	허니콤 (Honeycomb)	2.6.36	▪ 태블릿 PC에 최적화된 UI 지원 ▪ 외장 키보드 지원 ▪ 연결된 각각의 와이파이 액세스 포인트(AP)에 대한 HTTP 프록시 접근지원 ▪ 마이크로 SD카드 사용가능

2.2.4 안드로이드 SDK 4.x 버전

표 2-5는 현재 출고되는 스마트폰에 설치되는 버전인 아이스크림 샌드위치로 2011년 10월에 4.0.1이 발표가 되었으며, 스마트폰뿐만이 아니라 태블릿PC 전용 OS 버전인 허니콤 버전의 호환성 문제를 해결할 수 있는 버전이다. 2012년 6월 4.1버전인 젤리 빈이 출시되어 현재 사용되고 있다.

표 2-5. 안드로이드 SDK 4.x 버전 [10]

버전	이름	리눅스커널	버전별 특징
4.0 4.0.1 4.0.2 API Level 14 4.0.3 4.0.4 API Level 15	아이스크림 샌드위치 (Ice-Cream Sandwich)	3.0.1 (4.0.1) 3.0.8 (4.0.2,4.0.3,4.0.4)	▪ 음성을 문자로 변환하여 입력하는 음성인식 입력 기능추가 ▪ NFC 기능을 통한 웹사이트, 연락처, 위치 등을 교환 가능한 안드로이드 빔 추가

4.1 API Level 16	젤리 빈 (Jelly Bean)	3.0.31 ge8de0e2 (4.1)	▪ USB 오디오 출력과 멀티채널 오디오 지원 ▪ Adobe Flash Player 지원안함

2.3 안드로이드 컴포넌트

안드로이드 애플리케이션을 개발하는데 있어 가장 핵심이 되는 것이 바로 애플리케이션 컴포넌트이다. 안드로이드는 액티비티(Activity), 서비스(Service), 콘텐츠 제공자(Content Provider), 브로드캐스트 리시버(Broadcast Receiver) 컴포넌트를 제공하며, 제공되는 컴포넌트를 상속받아 안드로이드 애플리케이션을 작성한다. 애플리케이션은 자신의 컴포넌트를 다른 애플리케이션에서 사용할 수 있도록 허가한 경우 재사용이 가능하고, 애플리케이션은 객체지향의 특징이 잘 적용된 프레임워크에 의해 객체가 관리된다. 안드로이드 애플리케이션에서 사용되는 컴포넌트는 AndroidManifest.xml 파일에 선언 된다 [11].

2.3.1 액티비티(Activity)

안드로이드 컴포넌트에서 가장 기본적인 빌딩블록으로 하나의 사용자 인터페이스를 나타내는 단위를 액티비티라고 한다. 하나의 애플리케이션에는 하나 이상의 액티비티가 존재할 수 있으며, 이들 액티비티들은 하나의 UI를 형상하기 위해 함께 사용되지만, 액티비티들은 각각 독립적으로 존재하며, 액티비티 클래스의 하위 클래스로 액티비티 클래스에서 상속받아 구현된다.

액티비티는 활성, 일시정지, 정지, 종료 상태로 나누어진다. 백그라운드 프로세스가 아니면서 액티비티가 실행중이면 활성, 알림 메시지 등에 의해 화면이 잠시 가려진 상태를 일시정지, 액티비티가 실행중이지만 다른 액티비티가 활성화 되어 뒤로 밀려나 백그라운드에 대기상태를 정지, 액티비티가 활성화 상태에서 활성화 되지 않는 상태가 되면 종료 상태, 이렇게 네 가지로 나누어진다. 각각 독립적인 액티비티들은 화면의

전환이 이루어지게 되면 새로운 액티비티가 실행되어 처리하게 되고, 액티비티 간 서로 데이터를 주고받는 것 또한 가능하다. 또한 액티비티는 각각의 스택을 통하여 관리되며 현재 액티비티를 종료하면 그 이전 화면으로 전환된다.

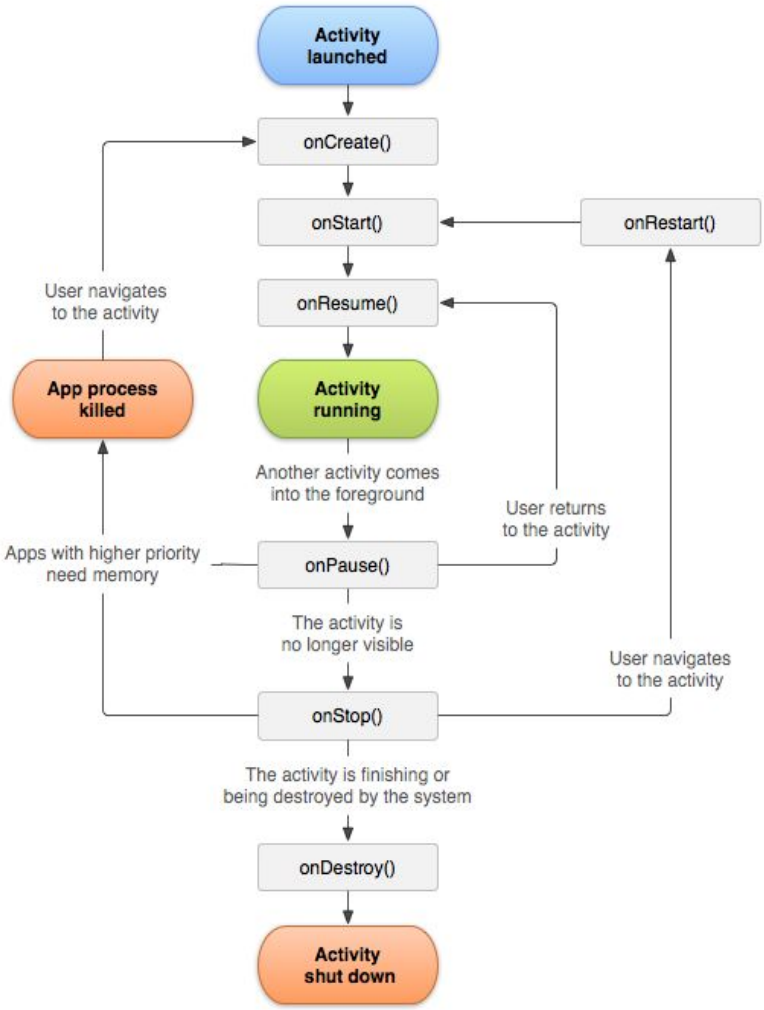

그림 2-5. 안드로이드 액티비티 생명주기 [12]

그림 2-5는 안드로이드 컴포넌트 중 액티비티의 생명주기를 나타내고 있으며, 액티비티가 실행되어 종료되기까지를 설명한다.

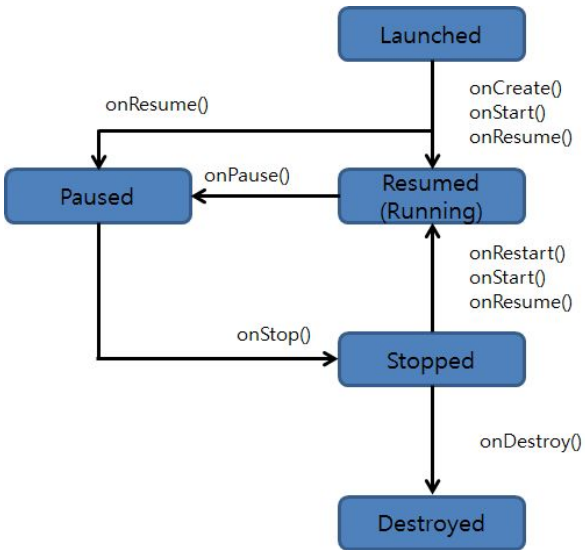

그림 2-6. 안드로이드 액티비티 상태변화 [12]

그림 2-6은 액티비티가 상태를 변화하는 것을 설명하고 있다. 각 메소드들의 특징은 다음과 같다.

- onCreate() : 애플리케이션을 실행하면 액티비티가 생성되는 시점에 안드로이드에서 실행하는 메소드
- onDestroy() : 활성화 된 액티비티를 종료하기 위해 안드로이드에서 호출하는 메소드
- onStart() : 액티비티가 사용자 화면에 보일 때 호출되는 메소드
- onRestart() : 액티비티가 Paused나 Stopped 상태에서 액티비티를 다시 시작할 때 안드로이드에서 호출하는 메소드
- onStop() : 액티비티 상태가 활성화 상태에서 다른 액티비티의 활성화로 밀려 Stopped 상태로 전환 될 때 안드로이드에서 호출하는 메소드
- onResume() : 액티비티를 사용자가 사용할 수 있을 때 안드로이드에서 호출하는 메소드로 사용자가 액티비티를 사용할 수 있음을 의미
- onPause() : 사용자가 다른 액티비티로 전환하는 순간 액티비티 상태가 Resumed에서 Paused로 전환될 때 안드로이드가 호출하는 메소드

2.3.2 서비스(Service)

서비스는 사용자가 사용하는 화면에 나타나지 않고 백그라운드(background)에서 활성화 되어있는 컴포넌트를 뜻한다. 백그라운드에 활성화 되어 있다는 것은 사용자가 현재 사용하는 애플리케이션 또는 다른 작업 중에도 음악을 재생하거나 데이터를 처리하는 등의 역할을 하며, 서비스 자체에서 요청하는 작업을 처리하여 해당하는 액티비티로 제공이 가능하다. 서비스의 생명주기 메소드는 void onCreate(), void onStart(Intent intent), void onDestroy() 이렇게 세 가지로 구성되며, 그림 2-7은 서비스의 생명주기를 설명한다.

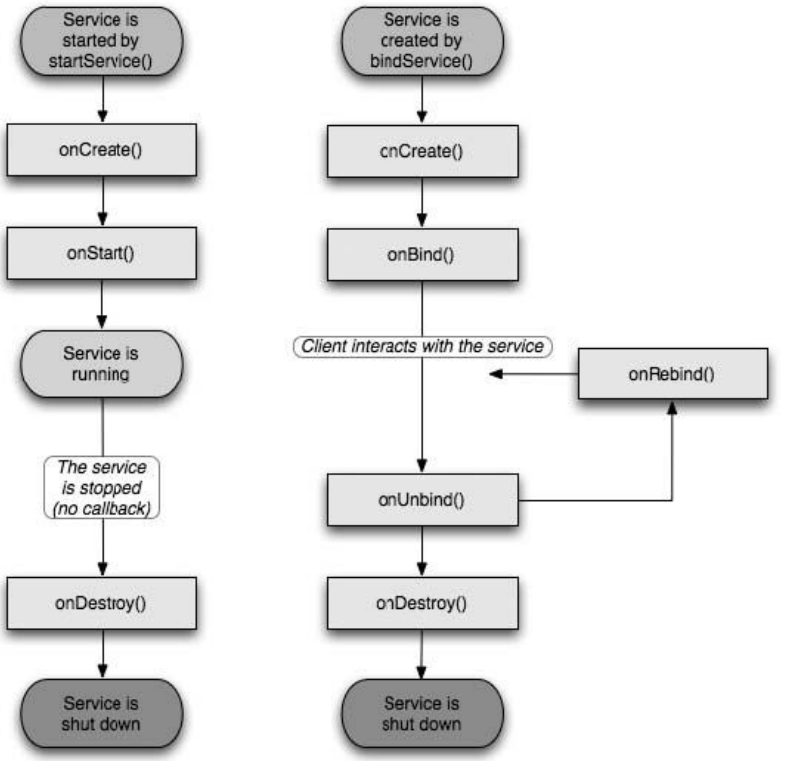

그림 2-7. 안드로이드 서비스 생명주기 [13]

서비스는 Context.startService()에서 시작되고 Context.stopService()에서 중지된다. 그리고 스스로 중지를 할 경우는 ServiceSelf() 또는 Service.stopselfResult()를

호출하여 중지한다. 서비스는 인터페이스로 사용을 원할 경우 서비스 객체에 커넥션을 만들고 서비스 내부를 호출해야 한다. Context.bindService()를 호출하여 커넥션을 만들고, Context.unbindService()를 호출하여 연결을 끊는다. 그리고 서비스는 실행이 되면 잠재적으로 클라이언트가 백그라운드에서 활성화 하는 것을 허용할 수 있기 때문에 onBind(), onUnbind()를 호출 받을 수 있다 [13].

2.3.3 컨텐트 프로바이더(Content Provider)

컨텐트 프로바이더는 데이터를 공급해주어서 애플리케이션들이 접근하여 데이터를 저장하고 가져와서 사용할 수 있도록 해주는 컴포넌트이다. 컨텐트 프로바이더는 안드로이드에 모든 패키지가 접근하여 데이터를 저장할 수 있는 공동의 저장공간이 없기 때문에 애플리케이션 간의 데이터를 공유할 수 있는 방법이다.

모든 컨텐트 프로바이더를 사용하기 위해서는 컨텐트 리졸버(ContentResolver) 객체가 필요하다. 사용하고자 하는 컨텐트 프로바이더와 통신하기 위해서는 각 메소드로 전달되는 URI에 의해 결정된다. 시스템은 URI로부터 컨텐트 프로바이더를 찾아 프로바이더를 로드하고 프로바이더의 메서드를 호출한다. URI 형태는 content://authority/path/id 로 구성되며 content://는 프로바이더 접근임을 명시하므로 반드시 들어가고, authority는 접근하는 프로바이더를 명시하는 것으로 다른 프로바이더들과 중복으로 사용할 수 없다. path는 정보를 지정하는 가상경로를 뜻하며, id는 구체적인 정보를 뜻한다. 그림 2-8은 컨텐트 프로바이더의 구조를 설명 한다 [14].

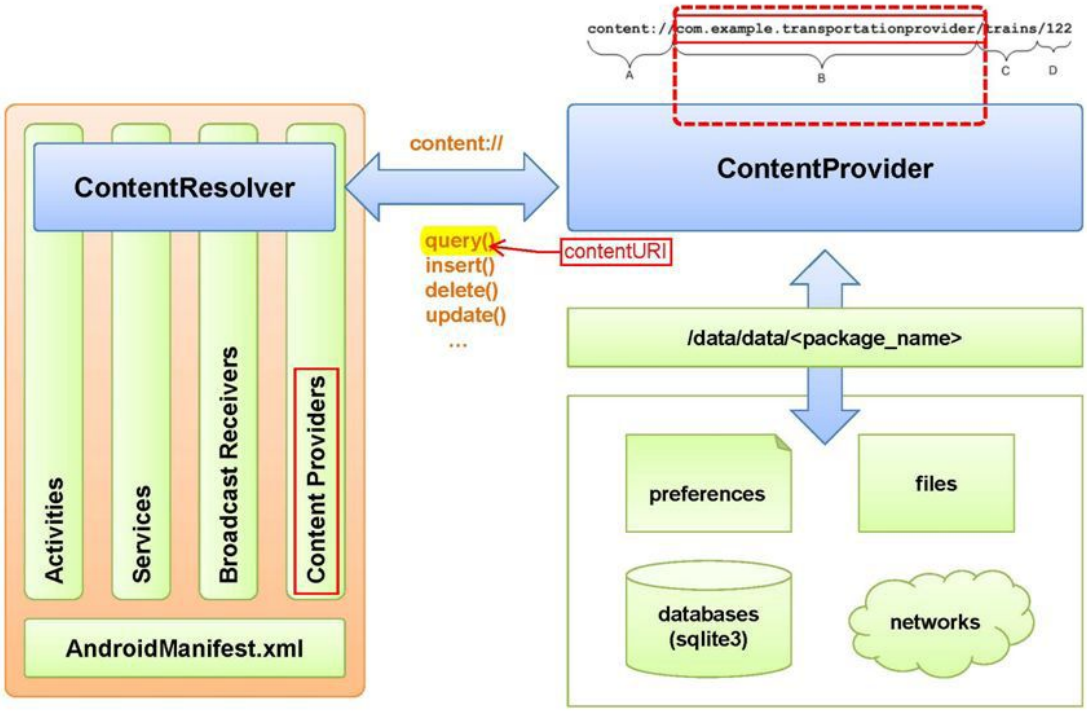

그림 2-8. 안드로이드 컨텐트 프로바이더 [14]

2.3.4 브로드캐스트 리시버(Broadcast Receiver)

브로드캐스트 리시버는 서비스 컴포넌트와 같이 화면을 제공하지 않는 상태로 애플리케이션에서 사용하는 정보를 받는 역할을 한다. 브로드 캐스트는 일반적으로 시간변화, 배터리 소모 상태 변화, 전화 및 메시지 수신 상태를 감지하여 알려주거나 데이터 다운로드 완료 등의 정보를 받는 역할을 한다.

브로드캐스트 리시버는 브로드캐스트 공지(announcement)를 수신하고 응답하는 역할만 수행한다. 애플리케이션은 공지가 발생하면 발생하는 공지에 따라 그 수만큼 브로드캐스트 리시버를 가질 수 있고, 그림 2-9와 같이 모든 리시버는 브로드캐스트 리시버 클래스를 상속 한다 [15].

public abstract class

BroadcastReceiver

extends Object

Summary: Nested Classes | Ctors | Methods | Inherited Methods | [Expand All]

Since: API Level 1

java.lang.Object

↳android.content.BroadcastReceiver

▸ Known Direct Subclasses

AppWidgetProvider, DeviceAdminReceiver

그림 2-9. 안드로이드 브로드캐스트 리시버 [15]

브로드캐스트는 수신한 정보를 처리하기 위해 액티비티를 실행 가능하며, 노티피케이션 매니저(Notification Manager)를 이용하여 사용자에게 알려줄 수 있다. 브로드캐스트 리시버 컴포넌트를 구현 하려면 프레임워크에서 제공하는 BroadcastReceiver 클래스를 상속받아 onRecevie 메소드를 새롭게 정의하고 Andoroidmanifest.xml 파일에 등록 한다 [15].

3. 착신번호 자동송신 시스템 설계 및 제작

3.1 착신번호 자동송신 시스템의 필요성 및 기본구조

본 논문에서 제안하는 착신번호 자동송신 시스템은 스마트폰을 소지하지 못하는 곳이나, 불가피하게 스마트폰을 특정 장소에 보관해야할 경우 거리적 제한을 해결하여 스마트폰을 확인하지 않고 스마트폰으로 착신되는 전화정보를 디스플레이를 통하여 확인할 수 있는 시스템이다. 블루투스 통신이 갖는 장점을 이용하여 무선 통신으로 데이터를 송신/수신할 수 있고, 블루투스 통신의 단점인 통신가능 범위의 거리적 제한과 통신가능 단말기가 7대 한정적인 문제를 해결할 수 있다. 지그비 모듈을 이용하여 블루투스 통신 신호를 지그비 통신 신호로 변환하는 기능을 갖춘 중계기로 스마트폰에서 착신정보를 자동으로 블루투스 통신을 이용하여 데이터를 송신/수신할 수 있는 애플리케이션과, 블루투스 통신의 단점을 해결할 수 있는 중계기 설계가 필요하다.

착신번호 자동송신 시스템은 크게 세 가지로 나뉜다. 스마트폰의 착신정보 데이터를 블루투스 통신을 이용하여 중계기로 자동으로 전송하고, 중계기에서 송신하는 데이터를 수신할 수 있는 애플리케이션과 스마트폰에 전송된 데이터를 수신하여 신호변환으로 변환된 데이터를 지그비 통신으로 전송할 수 있는 중계기, 그리고 중계기에서 보내진 데이터를 사용자가 확인할 수 있는 지그비 모듈을 탑재한 디스플레이장치이다.

3.2 착신번호 자동송신 애플리케이션

본 논문에서 제안하는 착신번호 자동송신 시스템은 스마트폰을 소지할 수 없는 제한적인 공간에서 사용 가능하다. 스마트폰에 착신정보가 발생했을 때 스마트폰에 설치된 애플리케이션을 실행하여 자동으로 착신정보를 데이터화하고 블루투스 통신을 이용하여 중계기로 전송한다. 착신번호 자동송신 애플리케이션은 착신정보 전송을 위한 블루

투스 기능 부분과 스마트폰 착신정보를 감지하고 데이터화하는 부분으로 구성된다. 착신번호 자동송신 애플리케이션의 생명주기는 그림 3-1과 같이 실행된다. 그림 3-2는 안드로이드 아키텍쳐에서 착신번호 자동송신 애플리케이션이 사용하는 플랫폼 순서도를 설명한다.

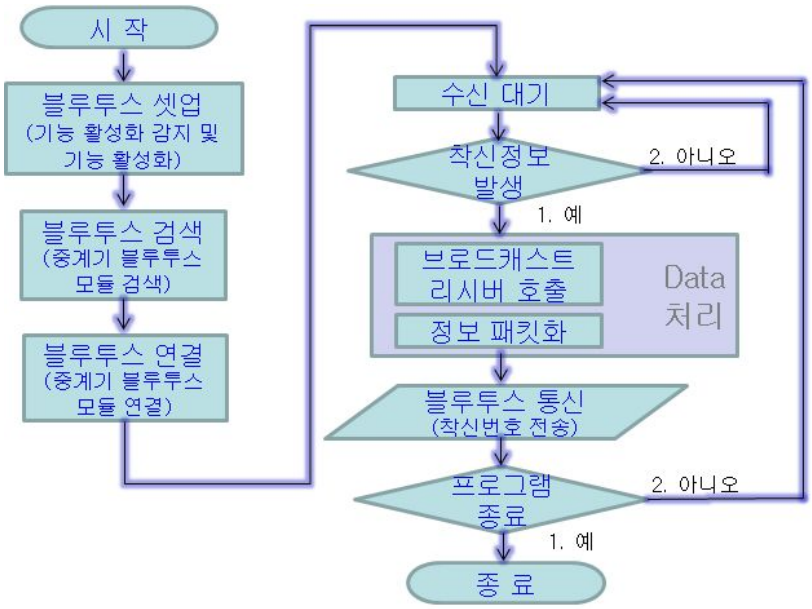

그림 3-1. 착신번호 자동송신 애플리케이션 생명주기

그림 3-1은 착신번호를 자동으로 송신하는 기능을 수행하는 애플리케이션의 생명주기와 같이 애플리케이션이 진행된다. 애플리케이션을 실행하면 블루투스 셋업기능으로 블루투스 활성화상태 감지를 하고 블루투스 기능이 비활성화 상태일 경우 블루투스 기능을 활성화 하도록 해준다. 그리고 블루투스 검색을 하여 연결하고자 하는 중계기 블루투스 모듈을 검색하고 해당 블루투스 모듈과 연결한다. 중계기와 블루투스 통신이 가능해진 애플리케이션은 착신정보가 발생할 때까지 수신대기에 들어가게 되고 착신정보가 발생하면 사용자가 전화를 받지 않을 경우 착신정보를 추출하기위해 브로드캐스트 리시버를 호출하여 전화상태 감지 및 착신번호를 추출하고 착신번호를 블루투스 통신으로 전송하기위해 정보를 패킷화 하여 블루투스 데이터 통신으로 중계기에 착신번

호를 전송하게 된다. 이후 사용자가 확인하고 애플리케이션을 종료하기 전까지 다시 수신대기 상태로 돌아가 착신정보 발생을 감지하며, 사용자가 애플리케이션을 종료하면 애플리케이션이 종료된다.

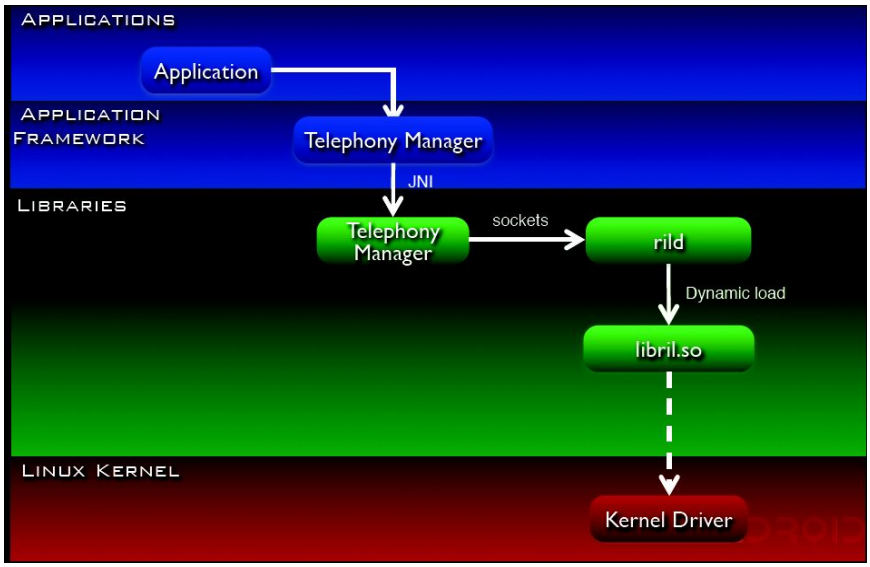

그림 3-2. 착신번호 자동송신 애플리케이션 플랫폼 순서도

3.2.1 스마트폰 착신정보 자동전송

본 논문에서 제시하는 착신번호 자동송신 시스템 중 착신번호 자동송신 애플리케이션에서 스마트폰에 착신정보가 발생하면 착신정보가 발생한 것을 확인하고 발신번호를 추출해야 한다. 착신번호 추출을 위해서는 안드로이드 운영체제를 기반으로 하는 스마트폰에 착신전화가 발생 했을 때 텔레포니 매니저(Telephony manager)에서 스마트폰 전화 상태를 감지하는 PhoneStateListener.LISTEN_CALL_STATE를 그림 3-3와 같이 코딩하여 사용한다.

```
public void onCallStateChanged(int state, String incomingNumber){
    switch (state) {
    case TelephonyManager.CALL_STATE_IDLE:
      // 스마트폰 상태가 울리거나 통화중이 아님
      break;

    case TelephonyManager.CALL_STATE_RINGING:
      // 스마트폰 상태가 울림 (발신자 번호는 incomingNumber로 받음)
        String InNum = incomingNumber;
      break;

    case TelephonyManager.CALL_STATE_OFFHOOK:
        txtPhoneNo.setText(incomingNumber);
      // 폰이 현재 통화 중.
      break;

    default:
      break;
    }
}
```

그림 3-3. PhoneStateListener 상태구별 소스코드

스마트폰의 상태설명은 다음과 같다.

- CALL_STATE_IDLE : 스마트폰이 울리거나 통화중이 아닌 상태
- CALL_STATE_RINGING : 스마트폰이 착신전화로 인하여 벨이 울리는 상태
- CALL_STATE_OFFHOOK : 스마트폰이 통화중인 상태

위 소스코드 작성으로 스마트폰 전화 상태를 감지하여 착신전화가 발생하면 incomingNumber를 이용하여 발신자 번호를 임의의 정적 변수에 저장한다.

incomingNumber의 경우 PhoneStateListener가 착신전화가 발생하는 도중에만 사용할 수 있기 때문에 착신전화가 끊어지기 전인 PhoneStateListener가 활성화 중인 상태에서 미리 선언해 둔 정적 변수에 저장을 한다. 이렇게 추출되어 저장된 발신자 번호는 블루투스 통신을 통하여 중계기로 전송한다.

3.2.2 안드로이드 블루투스 데이터 통신

본 논문에서 제안하는 착신번호 자동송신 시스템은 착신정보를 중계기로 전송하기 위해 안드로이드에서 제공하는 블루투스 API를 이용하여 블루투스 통신으로 데이터를 전송하여야 한다. 안드로이드 운영체제는 블루투스 프로토콜 스택을 포함하고 있어 블루투스 디바이스들과 무선 데이터 통신이 가능하다. 안드로이드 블루투스 아키텍쳐는 아래의 그림 3-4에서 설명한다 [16].

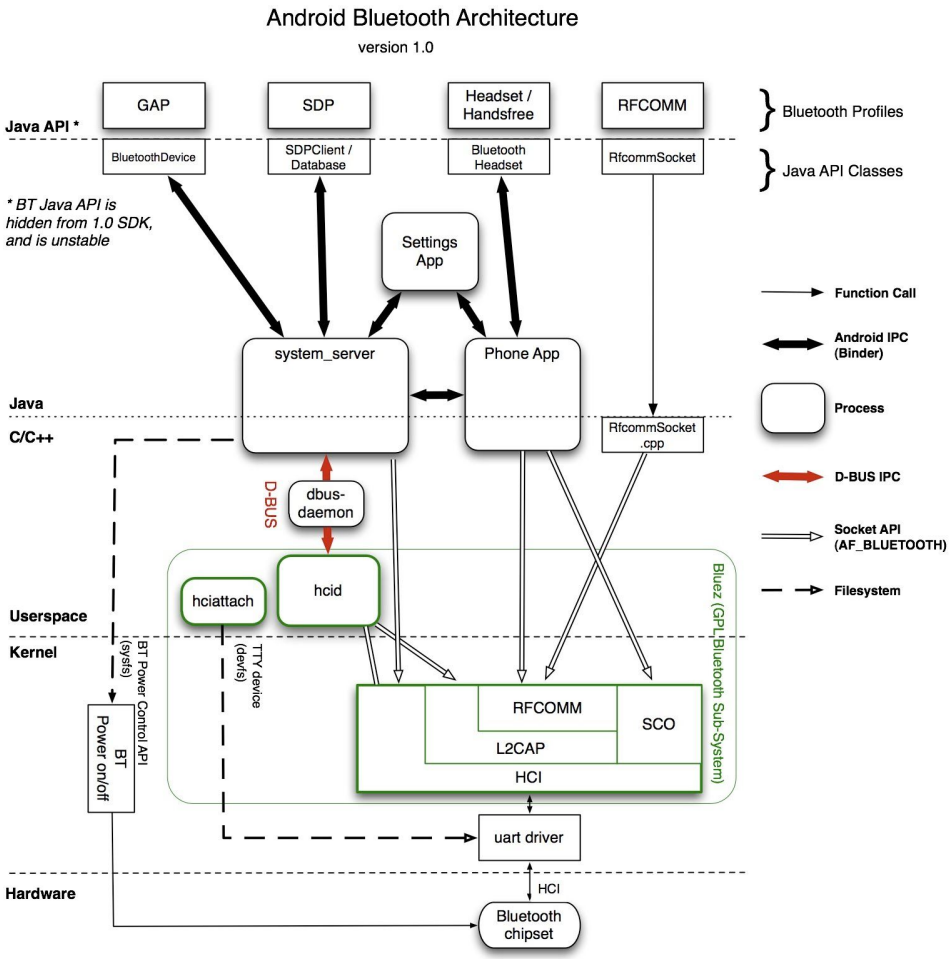

그림 3-4. 안드로이드 블루투스 아키텍쳐 [16]

블루투스를 사용하여 통신을 하기위해 안드로이드 운영체제의 네 가지 주요 태스크

가 사용된다. 네 가지 주요태스크는 다음과 같다.

- 블루투스 셋업 : 블루투스 상태 확인 후 비활성화 시 블루투스 기능 활성화 설정
- 디바이스 검색 : 블루투스 통신가능 범위 안에 통신대기 또는 페어링 된 기기검색
- 디바이스 연결 : 디바이스 검색으로 찾은 디바이스 목록 중 해당 디바이스와 연결
- 디바이스 간 데이터전송 : 연결된 디바이스간의 데이터 송신/수신

착신번호 자동송신 애플리케이션의 블루투스 기능을 사용하려면 BLUETOOTH 혹은 BLUETOOTH_ADMIN 둘 중에 하나를 애플리케이션 내의 퍼미션을 AndroidManifest.xml 파일에 선언해야 사용할 수 있다. 커넥션 요구와 커넥션 에셉트(Accept), 데이터 전송 기능은 BLUETOOTH 퍼미션을 선언하고, 블루투스 디스커버리를 시작하거나 블루투스 설정을 조작하기 위해서는 BLUETOOTH_ADMIN 퍼미션을 선언해야 한다. 그리고 BLUETOOTH_ADMIN 퍼미션을 선언하고 사용하려면 BLUETOOTH 퍼미션을 함께 선언해야 한다. 퍼미션 선언 방법은 그림 3-5의 다음과 같이 설정한다.

```
<uses-permission android:name=
    "android.permission.BLUETOOTH_ADMIN" />
<uses-permission android:name=
    "android.permission.BLUETOOTH" />
```

그림 3-5. 블루투스 퍼미션 선언

블루투스 퍼미션을 선언 이후에는 모든 블루투스 액티비티를 위해서 BluetoothAdepter가 필요하고 이를 사용하기 위해서는 스태틱 메소드인 getDefaultAdepter()를 호출하여 BluetoothAdepter를 얻는다.

다음 블루투스 기능의 활성화 상태감지를 하기위해 isEnabled()를 호출해서 상태감지 하고 비활성화 상태일 경우 ACTION_REQUEST_ENABLE 인텐트를 이용하여 startActivityForResult()를 호출하면 블루투스 상태를 활성화 상태로 전환시킬 수 있다. 블루투스 상태가 활성화 상태로 전환이 되면 스마트폰에서 데이터 통신을 하고자 하는 중계기의 블루투스 모듈을 검색하기 위해서 BlutoothAdapter를 이용하여 디바이스 디스커버리 또는 페어링 된 디바이스 목록을 쿼리하고, 중계기의 블루투스 디바이

스를 선택하여 커넥션 한다. 커넥션을 하기 전에 디바이스를 선택하고 최초로 연결을 했을 경우 페어링을 할 것인지 확인하게 되고, 페어링이 이루어지면 페어링 된 디바이스의 이름, 클래스, MAC 주소 등의 디바이스 정보가 저장되고 그 내용들은 블루투스 API를 통해 읽어올 수 있다. 이렇게 저장된 페어링이 이루어진 디바이스 정보들은 디바이스 디스커버리를 수행하기 전에 확인할 수 있으며, 확인 하기위해서 getBondedDevices()를 호출하면 페어링 된 디바이스들의 집합인 BluetoothDevices를 돌려준다. ArrayAdapter를 통하여 페어링 된 디바이스들의 이름을 볼 수 있고, 해당 디바이스와 연결을 시작하려면 MAC address만 있으면 된다. 디스커버리 과정은 보통 12초간의 인쿼리 스캔(inquiry scan)후 발견된 디바이스의 이름을 가져오기 위한 페이지 스캔(page scan)으로 이루어진다. 애플리케이션은 발견된 디바이스에 대한 정보를 ACTION_FOUND 인텐트를 위한 브로드캐스트 리시버를 등록해야만 받을 수 있고, ACTION_FOUND 인텐트는 BluetoothDevice와 BluetoothClass가 들어있는 EXTRA_DEVICE와 EXTRA_CLASS 필드를 전달한다.

다음으로, 중계기 디바이스를 커넥션 하기위해서는 안드로이드 블루투스 통신 방법 중 디바이스 연결을 기다리는 서버기능의 방법과, 접근을 대기 중인 디바이스에 연결을 시도하는 클라이언트 방법이 있다. 중계기 디바이스에 커넥션 하는 애플리케이션은 클라이언트 방법의 블루투스 통신방법을 사용한다. 중계기 디바이스와 커넥션을 시작하려면 중계기 디바이스의 BluetoothDevice 객체를 얻고, BluetoothSocket을 얻은 뒤 커넥션을 시작한다. Bluetoothsocket을 얻기 위해서는 BluetoothDevice를 사용해서 createRfcommSocketToServiceRecord(UUID)를 호출해서 얻는다.

UUID(Universally Unique IDentidfier)는 정보를 식별하기 위한 128비트 포맷의 표준화된 문자열로 구성된 유일한 정보이다. 애플리케이션의 블루투스 서비스를 식별하는데 사용되며, 블루투스 디바이스마다 고유 프로토콜 UUID 값을 UUID.fromString()에 지정을 해주어야 서비스를 이용하고자 하는 디바이스와 커넥션을 할 수 있고, 데이터 통신이 가능하다. 커넥션을 원하는 디바이스의 프로토콜 UUID 값이 아닌 다른 프로토콜 UUID 값이나 다른 값을 저장할 경우 페어링이나 디바이스 디스커버리 쿼리는 되지만, 데이터 통신이 불가능하다. 그러므로 연결 하고자하는 디바이스의 정확한 프로토콜 UUID 값을 저장하고 connect()를 호출하여 연결을 시작한다. connect()의 경우 블

록킹 호출이기 때문에 이 절차는 항상 메인엑티비티 스레드와는 독립된 별개의 스레드에서 수행되어야 한다.

착신번호 자동송신 애플리케이션에서 스마트폰 디바이스와 중계기 디바이스 사이에 커넥션이 성공하게 되면 BluetoothSocket을 가지게 되고, 이 소켓을 통하여 데이터 통신이 가능하다. 데이터 통신은 getInputStream()과 getOutputStream()을 사용하며, 소켓을 통한 전송을 처리해줄 InputStream과 OutputStream을 얻은 후에 read(byte[])와 write(byte[])를 사용해서 통신 내용의 데이터를 읽고 쓴다.

이렇게 안드로이드 운영체제에서 제공하는 블루투스 통신 기능을 이용하여 스마트폰과 중계기 사이에 데이터 통신이 가능하도록 애플리케이션을 설계한다.

3.3 중계기

3.3.1 중계기 필요성

본 논문에서 제안하는 블루투스 기반 착신번호 자동송신 시스템은 안드로이드 운영체제를 이용하는 스마트폰의 블루투스 기능을 이용하여 스마트폰에서 2.4GHz의 ISM 대역의 신호로 데이터 통신이 가능하도록 설계가 된다. 블루투스 통신은 높은 속도로 저전력, 저가의 서비스를 제공하지만, 근거리통신의 특성상 제한거리가 약 10m정도에서 최대 100m라는 통신거리에 대한 단점과 7대까지 한정으로 통신할 수 있는 제한점으로 사용자가 이용하는 노트북, 스마트폰, 블루투스 내장 헤드셋 등의 단말기 통신에 적합하다. 하지만 착신번호 자동송신 시스템은 용량이 큰 데이터 통신을 하지 않아도 되며, 블루투스 보다 저가와 저전력으로 운용이 가능한 지그비 통신이 적합하다. 지그비 통신은 같은 ISM 대역임에 불구하고 보다 넓은 통신거리를 제공하고, 최대 255대까지 통신이 가능하여 1:1로 스마트폰과 중계기가 아닌 1:N의 통신에서 유리하다. 이를 실현하기 위해 중계기에서 지그비 통신신호로 전환하여 통신할 수 있고, 스마트폰에서 전송하는 데이터를 송수신 하면서 전송된 데이터를 보여주는 디스플레이 기기까지의 통신가능범위를 확장한다 [16].

3.3.2 중계기 설계 [16]

중계기는 스마트폰에서 실행된 애플리케이션에서 착신정보가 발생하였을 때 추출된 발신번호 데이터를 패킷 화하여 전송하는 데이터를 수신하고 디스플레이에서 수신여부를 알리는 수신 확인메시지를 스마트폰으로 전송할 수 있는 장치이다. 블루투스 데이터 송수신이 가능하고, 스마트폰으로부터 수신된 데이터를 블루투스 통신신호에서 지그비 통신신호로 변환하여 디스플레이까지 전송이 가능하도록 지그비 모듈을 탑재하여 지그비 통신이 가능하다. 그림 3-6에서 중계기의 블록도로 중계기의 구조와 각 부분의 역할을 설명한다 [16].

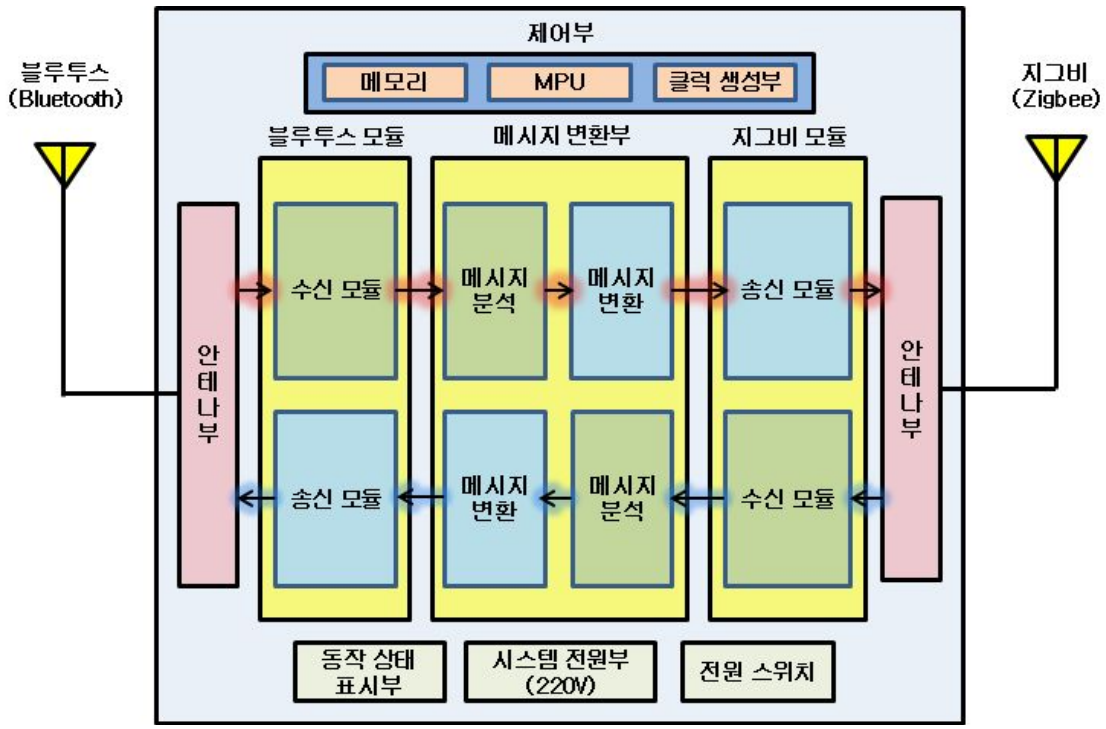

그림 3-6. 중계기 블록도 [16]

그림 3-6에서 보는 바와 같이 안테나부, 블루투스 모듈부, 메시지 변환부, 지그비 모듈부, 전원관리부, 제어부가 포함된다. 안테나부는 스마트폰의 NFC 시스템들이 사용하는 2GHz ISM 대역으로 해당 대역의 신호를 송수신할 수 있는 안테나, 그리고 디스플

레이와 통신할 수 있도록 지그비 통신신호를 송수신하는 안테나로 구성된다.

블루투스 모듈부는 스마트폰이 전송하는 메시지 데이터를 수신하는 모듈과 디스플레이에서 수신여부를 확인해서 데이터수신 확인메시지를 스마트폰으로 전송하는 송신기능 모듈로 구성된다.

메시지 변환부는 스마트폰으로부터 수신된 블루투스 통신신호를 지그비 통신신호로 변환하면서 디스플레이에서 착신정보 데이터 수신여부 확인메시지를 지그비 통신신호에서 블루투스 데이터로 변환하는 기능을 포함한다.

지그비 모듈부는 메시지 변환부로부터 수신한 데이터를 디스플레이로 전송하거나 디스플레이로부터 수신된 수신 여부 확인메시지를 메시지 변환부로 전송하는 기능을 포함한다.

전원부는 중계기의 동작 상태를 표시하는 표시부, 시스템 전원부, 전원 관리용 전원 스위치로 구성되며, 제어부는 메모리, MPU, 클럭 생성부로 구성되고, 데이터 변환을 위한 버퍼기능과 시스템 제어기능, 시스템 클럭 생성 등의 기능을 포함한다.

본 논문에서 제안하는 중계기는 위와 같은 구성으로 스마트폰과 디스플레이간 데이터 통신을 위한 서로 다른 통신신호를 변환하여 스마트폰에서 디스플레이까지 통신이 가능하도록 하였다. 블루투스가 지닌 단점을 극복하기위해 지그비 통신 신호로 변환하여 통신범위가 확장되고, 7대만 통신 가능한 제한을 극복하고, 블루투스보다 저전력, 저가로 시스템을 구현하여 문제를 개선한다.

4. 제안하는 시스템 구현 및 평가

4.1 애플리케이션 구현

본 논문에서 제안하는 블루투스 기반 착신번호 자동송신 시스템 중 착신번호 자동송신 애플리케이션은 안드로이드 운영체제를 이용하기 때문에, 안드로이드 어플리케이션 개발 전용 툴인 이클립스(Eclipse)를 사용한다. 이클립스는 3.6.1버전의 Helios를 사용하고, 안드로이드 2.2버전의 프로요(Froyo)가 2.1버전과의 호환성의 문제로 인해 API Level 8의 2.2 프로요 버전을 기반으로 구현한다.

착신번호 자동송신 애플리케이션의 기본 틀은 안드로이드 개발자를 위한 개발자 전용 오픈소스의 BluetoothChat을 응용하여 설계하고, 착신정보 자동전송 기능을 추가하여 착신번호 자동송신 애플리케이션을 구현한다. 애플리케이션의 구현은 다음에 나오는 그림 4-1에서부터 그림 4-4까지 그림순서와 같이 진행된다.

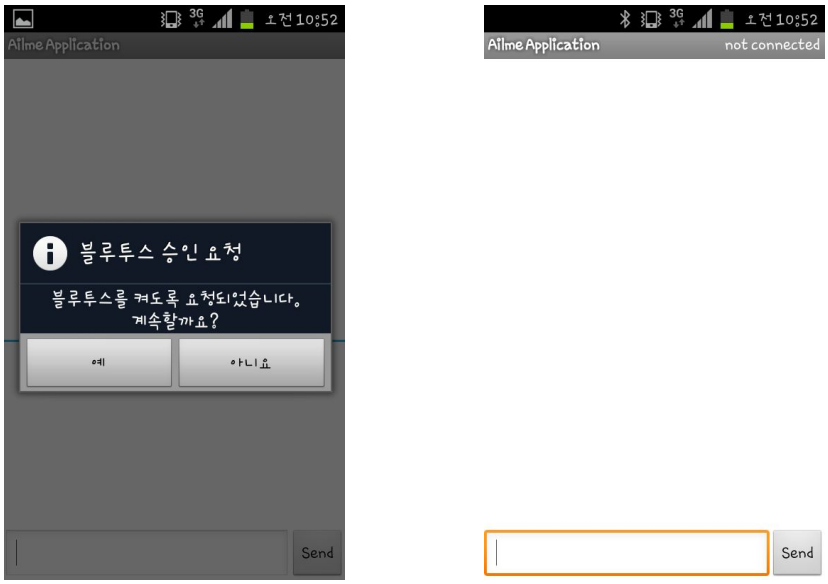

그림 4-1. 블루투스 기능 활성화

그림 4-1은 착신번호 자동송신 애플리케이션을 실행했을 때 애플리케이션이 블루투스 기능의 활성화 상태를 확인하고, 블루투스 기능 활성화가 되지 않은 상태의 경우 블루투스 기능을 활성화 하기위해 블루투스 셋업 기능을 이용하여 블루투스 기능 활성화 요청의 팝업창으로 블루투스 기능 활성화 승인 요청을 한다. 사용자는 블루투스 기능 활성화 승인요청을 수락하면 블루투스 기능이 활성화 되어 애플리케이션이 블루투스 기능을 사용할 수 있게 된다. 만약 블루투스 기능이 애플리케이션 실행 전 이미 활성화 상태일 경우 블루투스 활성화 요청은 생략된다.

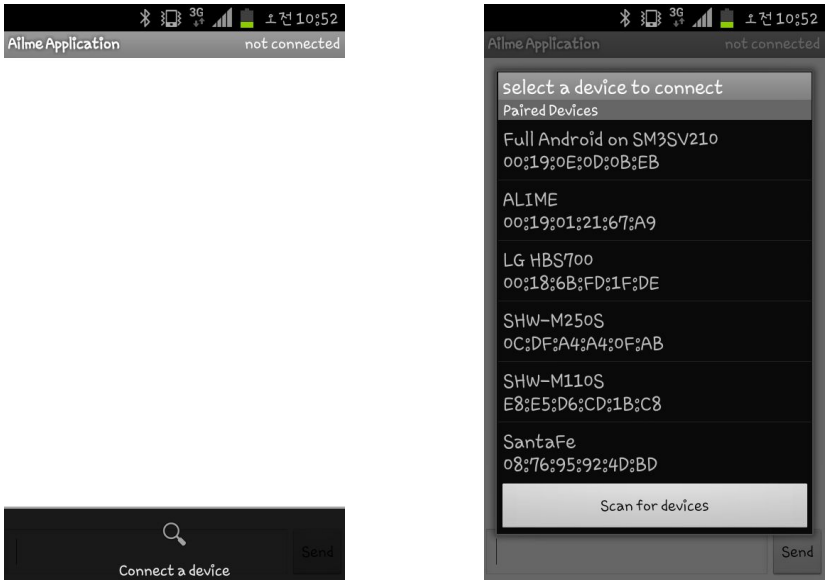

그림 4-2. 디바이스 검색

그림 4-2는 스마트폰이 중계기의 블루투스 모듈과 데이터 통신이 가능하기위해 중계기의 블루투스 모듈을 찾아서 연결을 해야 한다. 중계기의 블루투스 모듈을 찾기 위해서 디바이스 검색기능을 사용하여 주변에 블루투스 통신이 가능한 블루투스 모듈을 검색하고 페어링 된 적 있는 디바이스 목록을 보여주면 해당 디바이스의 정보를 확인하고 중계기 블루투스 모듈을 검색할 수 있다.

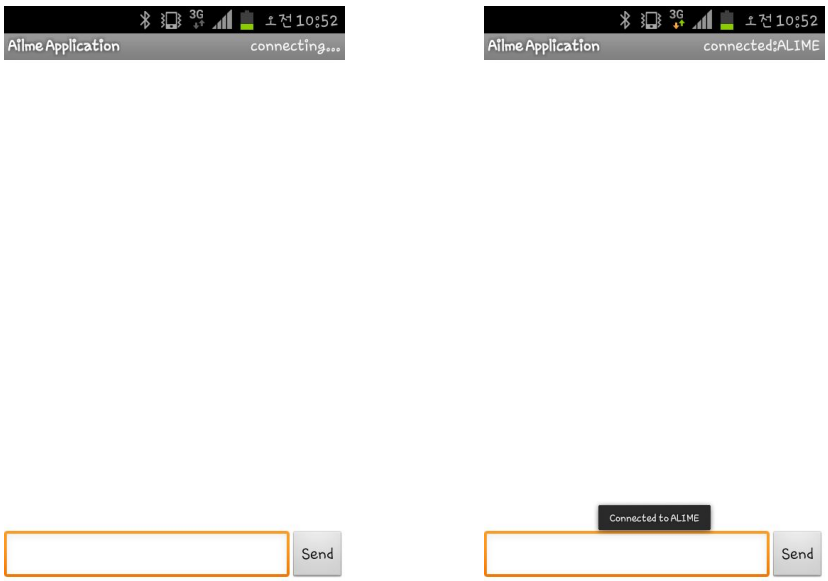

그림 4-3. 중계기 연결

그림 4-3은 스마트폰에서 블루투스 데이터 통신하고자 하는 중계기의 블루투스 모듈을 선택하여 연결 하는 것을 나타낸다. 구현하는 애플리케이션은 중계기의 블루투스 모듈과 연결해야 하므로 디바이스 이름을 검색하고 중계기와 연결한다. 연결이 정상적으로 이루어지면 연결이 됐다는 메시지 팝업창과 애플리케이션 우측 상단에 연결된 디바이스의 이름이 나타나고 연결 상태가 확인 가능하다.

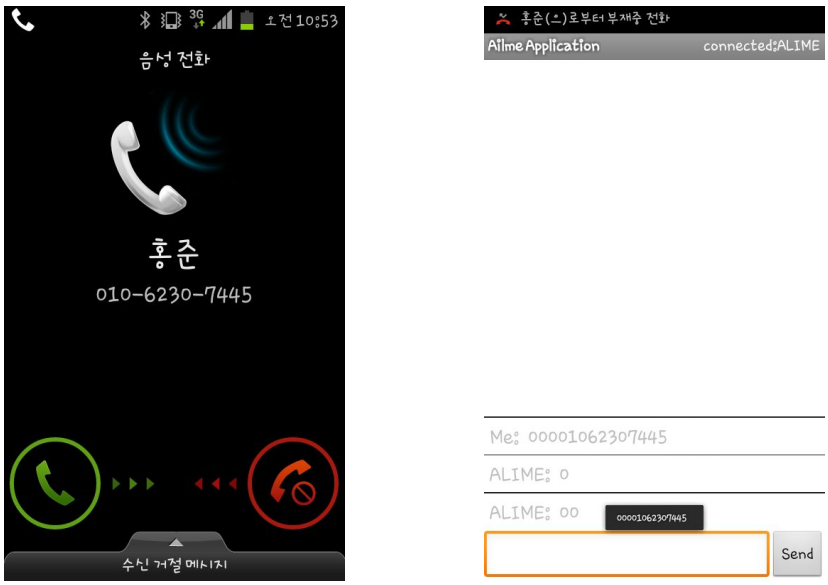

그림 4-4. 착신전화 정보전송

그림 4-4는 구현된 애플리케이션이 실행 중 착신전화가 발생하면 착신전화를 사용자가 받게 된 경우 애플리케이션은 착신정보를 전송하지 않아도 되므로 중계기와 데이터 통신을 하지 않는다. 하지만 착신전화를 받지 않고 착신전화가 종료가 되면, 브로드캐스트 리시버를 호출하여 착신정보 중 착신번호를 추출하여 애플리케이션 내에서 정적 변수에 저장하고 착신전화번호와 디스플레이기기에서 착신번호의 수신을 확인하기 위해서 확인메시지 숫자 3자리와 함께 블루투스 통신을 통하여 중계기로 전송하게 된다. 전송이 된 데이터는 우측 그림과 같이 중계기와의 데이터 통신 내용을 보여주는 텍스트 박스에 데이터 통신내용이 저장되어 착신되어진 정보 확인 및 디스플레이기기와의 데이터 통신 가능유무를 확인 가능하다.

4.2 중계기

블루투스 기반 착신번호 자동송신 시스템 중 중계기는 2.4GHz ISM 대역을 사용하는 블루투스 모듈과 지그비 모듈이 중계기 인터페이스 보드에 장착되며, 블루투스 통신신

호를 지그비 통신신호로 변환하여 데이터를 전송하거나 수신하고, 지그비 통신 신호를 블루투스 통신신호로 역으로 변환하여 전송한다. 중계기 구성은 다음 그림 4-5, 그림 4-6, 그림 4-7로 블루투스 모듈, 지그비 모듈, 인터페이스보드로 구성된다.

그림 4-5. 블루투스 모듈 [17]

중계기에서 사용되는 블루투스 임베디드 모듈은 그림 4-5와 같은 모델을 사용하였으며, 펌테크(Firmtech)사의 블루투스 통신전용 모듈로 총 8개의 핀으로 구성되며 4-5의 그림과 같이 설명된다 [17].

그림 4-6. 지그비 모듈

지그비 모듈은 그림 4-6과 같으며 (주)칩센 사의 제품으로 사용하였다.

그림 4-7. 중계기 인터페이스 보드

위와 같이 블루투스 모듈, 지그비 모듈, 인터페이스 보드로 구성된 중계기는 다음 그림 4-8에서 보는바와 같이 인터페이스 보드에 블루투스와 지그비의 모듈이 장착되고 블루투스 신호를 받아 지그비 신호로 변환해주는 기능을 수행한다.

그림 4-8. 중계기

4.3 착신번호 자동송신 시스템 구현

블루투스 기반 착신번호 자동송신 시스템은 목차 4-1과 4-2에서 제안된 애플리케이션이 설치된 스마트폰 SHW-M250S 단말기와 중계기를 포함하여, 지그비 모듈이 탑

재된 숫자 전용 디스플레이와 함께 전체적인 시스템이 구현된다. 데이터를 전송받아 통신 상태를 확인 가능한 디스플레이는 그림 4-9와 같다.

그림 4-9. 디스플레이

디스플레이는 총 14자리 숫자를 표현하며, 앞단 3자리는 수신여부 확인메시지로 숫자 3자리와 나머지 11자리 착신번호를 전송하여 착신정보를 출력한다.

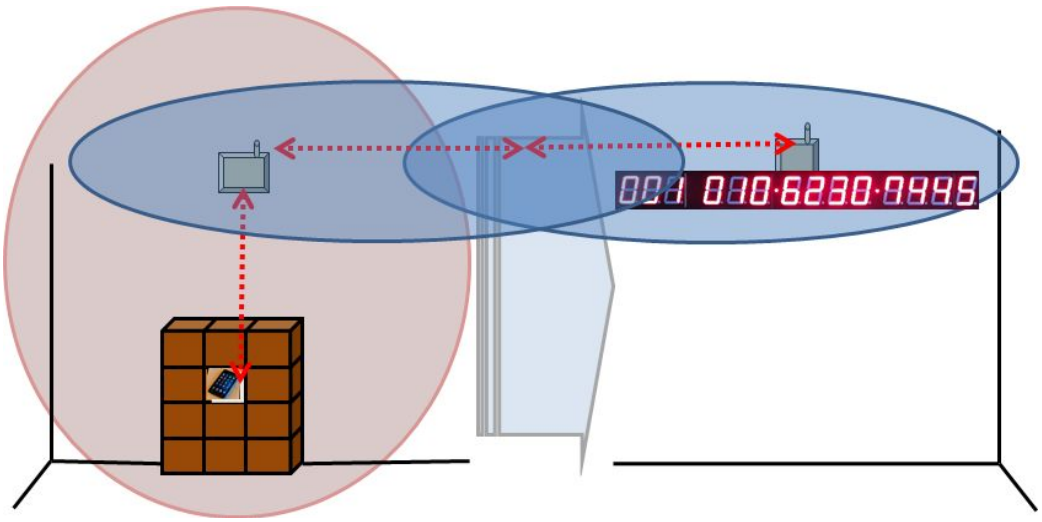

그림 4-10. 블루투스 기반 착신번호 자동전송 시스템

구현된 시스템은 그림 4-10에서 보는바와 같이 적용할 수 있다.

4.4 성능평가 결과

본 논문에서 제안하는 블루투스 기반으로 하는 착신번호 자동송신 시스템은 안드로이드 운영체제를 사용하는 스마트폰의 블루투스 통신을 이용하여 스마트폰에 착신정보가 발생했을 때 착신번호를 추출하여 자동으로 중계기에 데이터를 송신할 수 있는 애플리케이션 개발로 실현하였다. 개발된 애플리케이션의 성능 평가 결과 애플리케이션이 실행 중 착신정보가 발생했을 때 착신정보를 수신하고 중계기로 데이터를 전송하는 기능을 구현한 애플리케이션은 스마트폰의 블루투스와 중계기의 블루투스의 통신으로 착신정보를 전송하였다.

제안하는 시스템은 스마트폰을 소지하지 못하는 장소나 특정 공간에 보관해야 할 경우에 스마트폰을 소지하지 않아도 착신정보를 확인가능 하도록 설계했으므로, 테스트 환경은 스마트폰을 임의의 장소에 보관했다는 가정으로 각각 다른 재질의 저장장소를 설정하고 스마트폰 저장장소에서 블루투스 통신으로 통신이 가능하도록 설계한 중계기와의 통신가능거리 측정한다.

또한, 애플리케이션에서 전송되는 데이터를 받아 블루투스 통신의 신호를 지그비 통신이 사용하는 신호로 변환해주는 중계기는 블루투스 통신이 취약한 통신 가능한 거리적 제한 및 통신가능한 단말장치가 7대라는 제한과 주변 환경영향을 받는 단점을 보완하고자 지그비 통신신호로 변환해주는 역할을 한다. 중계기를 사용했을 때 블루투스 통신 보다 높은 성능을 입증하고자 테스트 환경을 설정하고 실험하였다.

실험에 사용되는 안드로이드 운영체제를 탑재한 단말기로 SHW-M250S(SWB-B42)를 사용하였고, 중계기에서 사용되는 블루투스 임베디드 모듈(FB155BC)과 지그비 모듈(EZB-M100)로 각각 통신모듈의 규격은 다음 표 4-1, 표 4-2와 같다.

표 4-1. 블루투스 모듈 규격 [17][18]

모델명	SHW-M250S (SWB-B42)	FB155BC
블루투스 규격	Bluetooth Specification V3.0 + HS	Bluetooth Specification V2.0 + EDR
통신 거리	최대 40m (Class1)	10m 이상 (Class2)
통신 속도	최대 24Mbps	1,200~230,400bps
동작 온도	-20℃ ~ 50℃	-10℃ ~ 50℃

표 4-2. 지그비 모듈 규격 [19]

모델명	통신 거리	통신 속도	동작 온도
EZB-M100	500m	1,200~115,200 bps (4800제외)	-25℃ ~ 80℃

위와 같이 각각의 통신모듈로 통신을 하기 전 블루투스 간 최대 50m와 중계기를 이용하였을 경우 최대 54m가 될 수 있다는 이론적 통신범위 확장을 입증하기 위해 거리적인 환경과 블루투스 통신의 취약한 주변 환경을 만들고 블루투스 간 통신과 중계기를 이용한 통신의 실험환경은 동일한 조건에서 최대한 변화가 없도록 실험을 진행하였다.

실험환경은 중계기를 사용하여 블루투스 통신만을 이용했을 때보다 성능향상 검증에 중점을 두어 구성하였으며, 중계기의 사용으로 블루투스 통신의 이용성 향상과 통신범위 확장 가능성을 실험을 통하여 데이터를 얻어 분석하였다.

스마트폰을 보관할 수 있는 공간을 구성하는 재질에 따라 스마트폰과 중계기의 블루투스 통신가능 범위가 달라질 수 있으므로 여러 재질을 이용하여 실험을 진행한다. 표

4-3은 각 테스트환경에서 스마트폰과 블루투스 통신 가능여부와 통신가능거리를 측정하였다.

표 4-3. 스마트폰 수납공간 재질에 따른 통신가능여부 및 거리 실험

실험환경	통신가능여부	통신가능거리
플라스틱	가능	18.9m
알루미늄	가능	11.3m
나무	가능	14.4m
유리	가능	18.4m
종이	가능	15.7m

표 4-3에서의 결과와 같이 스마트폰을 수납공간 재질에 따라서 중계기와의 통신이 가능한 거리는 짧게는 11.3m에서 길게는 18.9m까지 통신거리가 차이가 있었다. 이와 같은 결과는 스마트폰의 수납공간이 어떠한 재질이냐에 따라서 중계기의 설치위치를 설정하는데 기준이 되고, 스마트폰을 수납하는 공간이 설치되는 장소에 따라서 중계기의 설치하는 개수를 설정할 수 있다.

스마트폰에서 중계기까지의 통신성능 테스트는 스마트폰을 보관하는 장소의 재질이나 중계기 설치위치를 설정할 수 있는 기준을 설정하였고, 표 4-4에서 나타나는 블루투스 통신과 중계기 통신을 비교하는 실험은 블루투스만으로 중계기 없이 통신이 가능하지만, 중계기에서 블루투스 신호를 지그비 신호로 변환하여 지그비 통신으로 중계기에서 디스플레이까지 통신하는 방법이 블루투스만으로 통신하는 방법보다 성능이 좋아지는 것을 증명하고자 하였다.

표 4-4. 블루투스와 중계기 통신비교 실험

실험환경	재질	통신방법	통신가능여부	통신가능거리
장애물 없는 공터		블루투스 통신	가능	26.47m
		중계기 통신	가능	47.32m
복도/내실	벽: 콘크리트 문: 철문	블루투스 통신	불가능	
		중계기 통신	가능	10.90m
복도/내실	벽: 콘크리트 문: 나무	블루투스 통신	가능	2.84m
		중계기 통신	가능	10.35m
복도/내실	벽: 철판 문: 유리	블루투스 통신	가능	1.05m
		중계기 통신	가능	17.14m
복도층간	벽: 콘크리트	블루투스 통신	가능	2층 (약 6m)
		중계기 통신	가능	2층 (약 6m)
실내/실외	벽: 콘크리트 (폭 16 m 내실을 포함하는 복도)	블루투스 통신	불가능	
		중계기 통신	가능	2.56m
건물외곽	벽: 콘크리트 (건물 폭 약 35 m)	블루투스 통신	불가능	
		중계기 통신	가능	0.5m

표 4-4에서 나타나는 블루투스 통신과 중계기를 이용한 통신 비교는 그림 4-11의 차트와 같다.

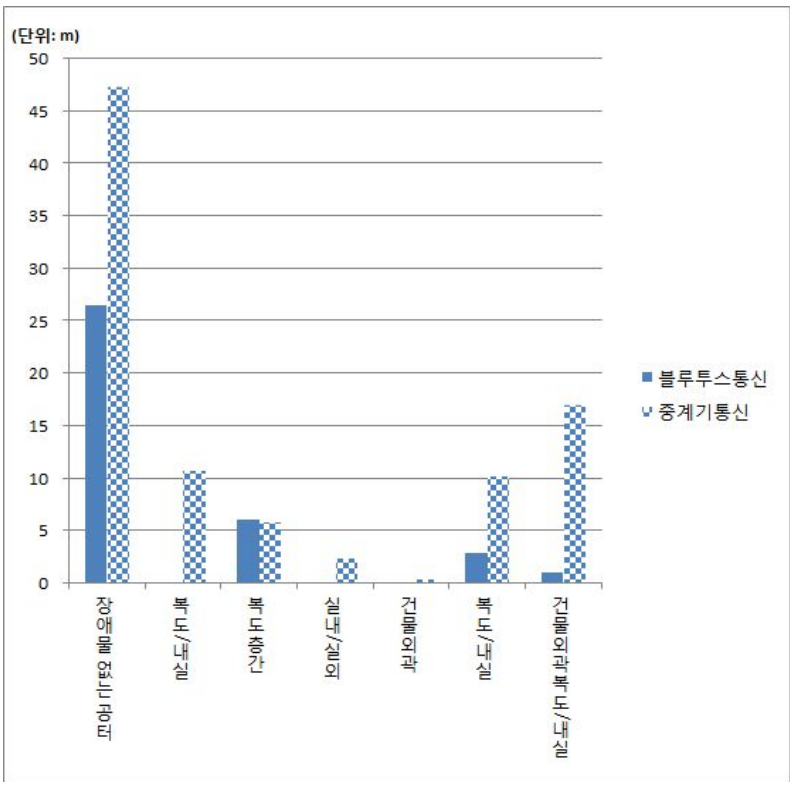

그림 4-11. 블루투스 통신과 중계기 통신비교 실험

표 4-4와 그림 4-11에서 보는바와 같이 실험결과는 같은 2.4GHz ISM 대역의 신호로 통신을 하지만, 블루투스로 통신했을 때 보다 중계기를 이용하여 지그비 통신을 이용했을 때 장애물이 없는 장소에서 최대 20.85m의 통신거리가 길어지는 것을 확인할 수 있었으며, 복도층간 통신을 이용할 경우에는 블루투스와 중계기가 차이가 없는 것을 확인할 수 있었다.

5. 결론

본 연구에서는 스마트폰을 소지할 수 없는 제한된 공간에서 스마트폰에 착신되는 착신번호를 확인할 수 있도록 스마트폰 착신 자동송신 애플리케이션을 개발하였다. 그리고 이를 보다 광범위한 영역에서 구현하기 위하여 중계기와 디스플레이를 제작하여 주변 환경에 따른 제한 특성을 실험하여 다음과 같은 결과를 확인하였다.

실험은 크게 두 가지방향으로 실시하였고, 첫 번째로 스마트폰을 소지하지 못하는 장소에서 스마트폰을 보관하는 공간재질에 특성별로 스마트폰과 중계기의 블루투스 통신 가능범위를 설정할 수 있는 통신가능유무 및 통신가능거리 실험으로 총 5가지의 플라스틱, 알루미늄, 나무, 유리, 종이 재질로 보관함을 구성하여 실험을 진행하였으며, 5가지 종류 모두 통신은 가능하였다. 통신가능 거리는 알루미늄 재질의 실험으로 11.3m가 되었으며, 플라스틱 재질이 18.9m의 가능거리로 가장 먼 거리까지 통신이 가능하였다. 이와 같은 실험결과는 최소한 10m의 범위 안에 중계기가 설치할 경우 스마트폰과 중계기의 블루투스 통신이 가능함을 증명하였다.

두 번째 실험으로는 블루투스 통신과 중계기의 지그비 통신의 주변 환경에 따른 실험으로 통신가능확인 및 통신가능거리를 실험으로 중계기 사용이 블루투스 통신보다 적합함을 증명하기 위하여 실시하였다. 실험환경 장소의 설정은 총 7개의 장소이고 중계기를 이용한 통신은 전부 통신이 가능하였으나, 블루투스는 3개의 장소에서 통신이 불가능하였다. 또한 장애물이 없는 장소에서 최대 20.85m의 통신가능 거리가 확장되는 것이 중계기를 이용한 통신이 적합함을 증명하였다.

제안하는 시스템은 근거리 통신에서 블루투스 통신은 지속적인 발전으로 많은 분야에서 이용될 것이며, 그로인한 블루투스 통신의 사용량 증가로 제한적인 범위에 그쳐있는 사용가능한 범위를 확장할 수 있는 방법으로 중계기가 블루투스 통신의 단점을 극복하는 하나의 시스템이 될 수 있을 것이다.

향후에는 애플리케이션이 다양한곳에서 활용될 수 있도록 완성도를 높이고, 착신번호 뿐만 아니라 메시지 및 보다 큰 정보를 전송할 수 있는 기능과 사용자들이 편하게 사용할 수 있도록 UI를 구성하여야한다. 또한 중계기는 통신거리 확장과 주변 환경의

통신장애를 해결하기 위해서 통신가능거리가 길고, 금속이나 습도, 주변 환경의 통신장애 요소에 대해 영향을 적게 받는 통신방법 연구와 중계기 하나와의 통신이 아닌 건물전체를 통신가능 범위로 설정할 수 있는 하나 이상의 중계기를 사용하여 시스템을 설계할 수 있도록 지속적인 응용과 연구가 필요할 것이다.

참고 문헌

[1] 장윤정, 김철우, “스마트폰 시장의 진화와 안드로이드의 영향,” 한국정보과학회, 정보과학학회지 제28권 제5호, pp. 48-56, 2010.5

[2] “모바일 개방형 플랫폼 - 안드로이드(Adroid)” 정보통신 진흥원, [IITA] 정보통신 연구진흥원 학술정보 인터넷 이슈리포트 2007권 12호, 2010. 6

[3] 고석훈, “안드로이드 플랫폼 동향,” 한국콘텐츠학회, 한국콘텐츠 학회지 제8권 제2호, pp. 45-49, 2010. 6

[4] “Specification of the Bluetooth System,” version 1.1, 2001

[5] 김학윤, “스마트폰 어플리케이션을 이용한 RC 카 제어,” 한국정보기술학회, 2012년도 빅[Big]데이터 시대의 정보기술전략을 위한 워크숍, pp. 450-453, 2012. 5

[6] 김서궁외 3명, “원격진단을 위한 U-Health Care 어플리케이션 개발,” 한국정보기술학회, 2012년도 빅[Big]데이터 시대의 정보기술전략을 위한 워크숍, pp. 53-57, 2012. 5

[7] (2012) the Wikipedia website. [Online]. http://ko.wikipedia.org/wiki/안드로이드_(운영_체제)/

[8] (2012) the Google Android Developers Official website. [Online]. http://developer.android.com/guide/basics/what-is-android.html/

[9] (2012) the IDC website. [Online]. http://www.idc.com/getdoc.jsp?containerId=prUS23638712/

[10] (2012) the Google Android Developers Official website. [Online]. http://developer.android.com/tools/index.html/

[11] (2012) the Google Android Developers Official website. [Online]. http://developer.android.com/guide/topics/fundamentals.html/

[12] (2012) the Google Android Developers Official website. [Online]. http://developer.android.com/reference/android/app/Activity.html/

[13] (2012) the Google Android Developers Official website. [Online]. http://developer.android.com/reference/android/app/Service.html/

[14] (2012) the Google Android Developers Official website. [Online]. http://developer.android.com/reference/android/content/ContentProvider.html/

[15] (2012) the Google Android Developers Official website. [Online]. http://developer.android.com/reference/android/content/BroadCastReceiver.html/

[16] 박종안외 5명, “휴대폰 수신신호 알림을 위한 디스플레이 시스템 설계,” 한국ITS학회, 학술대회, pp. 15-17, 2011. 5

[17] (2012) the Firmtech website. [Online]. http://www.firmtech.co.kr/02download/main_kor.php?index=4/

[18] (2012) the Samsung website. [Online]. http://www.samsung.com/sec/support/model/SHW-M250SLKSC-downloads/

[19] (2012) the Firmtech website. [Online]. http://chipsen.com/index.html?TPL=zg4.tpl/

저작물 이용 허락서

학 과	정보통신공학과	학 번	20117307	과 정	석사
성 명	한글 김 서 궁	한문 金 서 궁		영문 SeoGung Kim	
주 소	광주광역시 동구 서석동 375 조선대학교 전자정보공과대학				
연락처	e-mail : forevertopstar@nate.com				
논문 제목	한글 : 스마트폰 착신번호 자동송신을 위한 애플리케이션 구현				
	영문 : Implementation of application for automatic transmission of incoming phone numbers on smart phones				

본인이 저작한 위의 저작물에 대하여 다음과 같은 조건 아래 조선대학교가 저작물을 이용할 수 있도록 허락하고 동의합니다.

- 다 음 -

1. 저작물의 DB구축 및 인터넷을 포함한 정보통신망에의 공개를 위한 저작물의 복제, 기억장치에의 저장, 전송 등을 허락함
2. 위의 목적을 위하여 필요한 범위 내에서의 편집과 형식상의 변경을 허락함 (다만, 저작물의 내용변경은 금지함)
3. 배포・전송된 저작물의 영리적 목적을 위한 복제, 저장, 전송 등은 금지함
4. 저작물에 대한 이용기간은 5년으로 하고, 기간종료 3개월 이내에 별도의 의사 표시가 없을 경우에는 저작물의 이용기간을 계속 연장함
5. 해당 저작물의 저작권을 타인에게 양도하거나 출판을 허락을 하였을 경우에는 1개월 이내에 대학에 이를 통보함
6. 조선대학교는 저작물 이용의 허락 이후 해당 저작물로 인하여 발생하는 타인에 의한 권리 침해에 대하여 일체의 법적 책임을 지지 않음
7. 소속 대학의 협정기관에 저작물의 제공 및 인터넷 등 정보통신망을 이용한 저작물의 전송・출력을 허락함

동의여부 : 동의 (○) 반대 ( )

2012 년 12 월 10 일

저작자 : 김 서 궁 (인)

조선대학교 총장 귀하