**SEYDC-14-02-결과보고서**

IVA

(Intelligent Voice Application)

**2014. 2. 16**

이 름: 안종현

학 교: 강남대학교

연락처: 010-9995-5600

이메일: [markajh@naver.com](mailto:markajh@naver.com)

**목차**

[1. 개요 1](#_Toc380351842)

[1.1 작품 제목 1](#_Toc380351843)

[1.2 부제 1](#_Toc380351844)

[1.3 개요 1](#_Toc380351845)

[2. 작품 설명 2](#_Toc380351846)

[2.1 개발결과물 주요 동작 및 특징(영상 구동 Scenario 포함) 2](#_Toc380351847)

[2.2 전체 시스템 구성(구조 설명 Flow Chart 포함) 4](#_Toc380351848)

[2.3 S/W 구조(개발결과물 Algorithm) 6](#_Toc380351849)

[2.4 개발환경(활용언어, 사용시스템, 사용된 공개 SW 등) 6](#_Toc380351850)

[2.5 해당 개발과제 진행 시 애로 사항(불량 및 해결방법) 6](#_Toc380351851)

[2.6 향후 과제개선 목표 및 보완사항 7](#_Toc380351852)

[2.7 참조 소스 및 출처 7](#_Toc380351853)

[3. 개발결과물 사진 8](#_Toc380351854)

[3.1 전화 받기 & 거절 8](#_Toc380351855)

[3.2 상태바 기능 조절 8](#_Toc380351856)

[3.3 시스템 어플리케이션 컨트롤 9](#_Toc380351857)

[3.4 사용자 어플리케이션 컨트롤 9](#_Toc380351858)

[3.5 하드웨어 기능 조절 9](#_Toc380351859)

[3.6 잠금 화면 기능 10](#_Toc380351860)

[3.7 특정 단어 터치 이벤트 10](#_Toc380351861)

[4. 개발결과물 동영상 11](#_Toc380351862)

1. 개요

# 1.1 작품 제목

IVA로 Intelligent Voice Application의 약자이다.

# 1.2 부제

양손이 사용불가능 일 때, 목소리만으로 스마트 폰을 작동시키거나 제어할 수 있는 어플리케이션이다.

# 1.3 개요

1. 개발배경 및 목적: 인간의 말을 인식해 텍스트로 바꿔주거나 해당 명령을 수행하는 ‘음성인식(speech recognition)’ 기술이 IT업계의 Next Big Thing으로 급 부상 중이다.

2. 시장 현황 분석: 음성인식 기술 관련 세계 시장 규모는 2005년 11억 달러에서 2010년 30억 달러로 성장한 데 이어 2014년에는 60억 달러까지 성장할 것으로 전망된다.

3. 문제점: 음성인식은 사람이 일상생활 속에서 마우스나 키보드 등을 사용하지 않고 목소리를 통해 원하는 기기 및 정보 서비스의 이용을 제어할 수 있는 기술로, 1950년대 등장해 지속적인 연구가 진행되어 왔지만 2000년대 중반까지 낮은 음성 인식률로 대중화되지 못했다.

4. 미래방향: 네트워크와 컴퓨팅 기술의 발달로 음성 인식률이 개선되었을 뿐만 아니라 정보기기가 소형화되고 이동성이 중요시되면서 음성으로 간편하게 제어할 수 있는 음성인식에 대한 수요는 더욱 증가할 것으로 전망된다.

2. 작품 설명

# 2.1 개발결과물 주요 동작 및 특징(영상 구동 Scenario 포함)

**1. 어플리케이션 실행 화면**

- 4개의 탭으로 구성

**2. 음성인식 탭**

- 음성인식 서비스를 끄고 실행할 수 있다.

**3. 어플목록 탭**

- 두 번째 탭은 어플리케이션 목록을 보여준다. 이름이 길거나 발음이 어려운 경우 실행할 어플리케이션 이름 변경이 가능하다.

- 실행방법: “실행” + 어플이름, 또는 어플이름 + “실행”

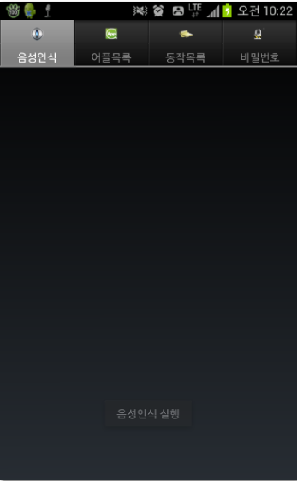
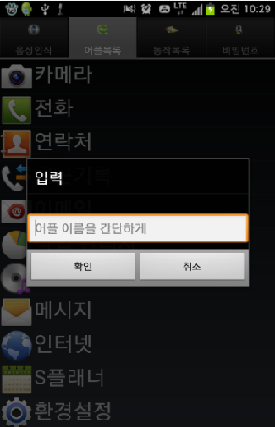
 

Figure . App Tab-1,2

**4. 동작목록 탭**

- 음성으로 동작 가능한 목록을 보여준다.

- 실행방법: “버튼”+목록에 있는 버튼 이름, 목록에 있는 버튼 이름 + “버튼”

**5. 비밀번호 탭**

- 음성 및 숫자로 비밀번호 설정이 가능하다.

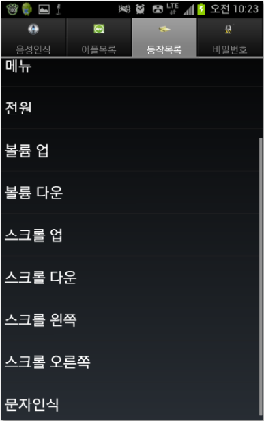
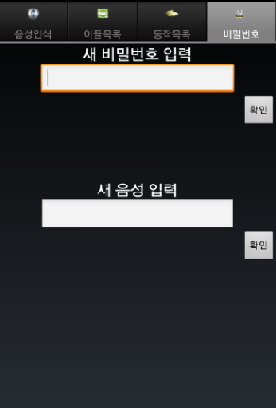
 

Figure . App Tab-3,4

**6. 문자인식**

- Tesseract OCR로 화면을 인식하고 사용자가 말한 글자에 대한 좌표 값을 리턴하여 터치하는 방식으로 구현했다.

- 실행방법: “인식”(Tesseract OCR로 문자 인식) -> “터치” + 화면에 보이는 글씨



Figure 3. Character Recognition

**7. 전화 걸기**

- 실행방법: “통화” + 사람이름

**8. 웹 검색**

- 실행방법: “검색” + 검색할 내용

**9. 문자 보내기**

- 실행방법: “문자” + 사람이름 + 내용

**10. 음악 실행**

- 실행방법: “음악” + 음악파일이름

**11. 전화 받기 및 거절**

- 실행방법: 전화가 왔을 때 “받기” & “거절”

**12. 밸소리/무음/진동/Wifi**

- 실행방법: “벨소리”, “무음”, “진동”, “Wifi”

**13. 락 화면 해제 방법**

- 미리 입력해 놓은 글자에 매칭되는 음성이 들어오면 잠금 기능이 풀리는 방식

# 2.2 전체 시스템 구성(구조 설명 Flow Chart 포함)

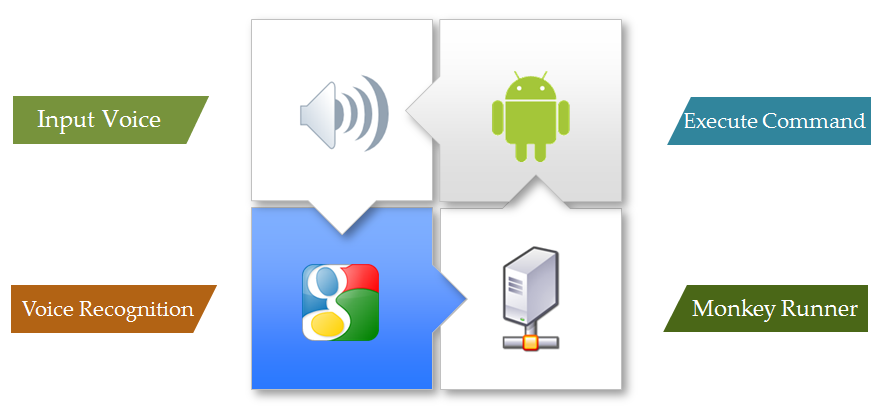


Figure 4. System Configuration

**1. Input Voice**

- 음성 입력

**2. Execute Command**

- 구글 서버로부터 받은 문자열 결과 값을 필터링 처리하여 명령 수행

**3. Voice Recognition**

- Android에서 제공되는 SpeechRecognition Lib를 이용하지 않고 음성을 녹음하여 서버로 전송하고 그 결과값을 받음

- HTTP POST방식으로 구글 서버에 음성 데이터를 전송함

- 반환된 결과값 포맷은 Json방식임, 이를 파싱하여 문자열 확인

**4. MonkeyRunner**

- 캡처, 문자인식, 버튼, 터치



Figure 5. System Flow

1. App은 백그라운드로 실행하며 사용자 음성을 판별한다. 음성일 경우 음성인식 시스템으로 음성파일을 보내고 비 음성일 경우 음성이 들어올 때까지 판별작업을 계속한다.

2. 음성인식 시스템에서는 음성파일을 구글 음성인식 서버로 보낸다. 그리고 반환되는 JSON형식의 데이터 타입을 파싱하여 명령어들을 조합해 하나의 명령어로 생성한다.

3. 하나의 문자열 명령어로 된 데이터를 음성처리 시스템으로 보내서 문자열 파싱을 하며 정해진 방식으로 어플리케이션들을 실행한다.

4. 만약 시스템 권한이 필요한 명령어이면 Monkey Runner를 통해 우회하여 스마트 폰에 실행 명령어를 내린다.

# 2.3 S/W 구조(개발결과물 Algorithm)

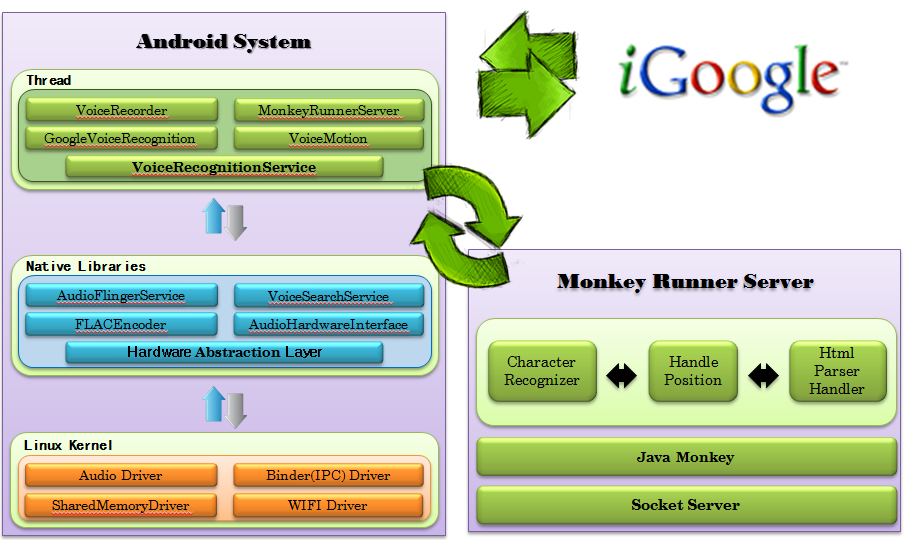


Figure 6. S/W 동작 구조

# 2.4 개발환경(활용언어, 사용시스템, 사용된 공개 SW 등)

1. 활용언어: C, C++, Java, Android SDK, Android NDK

2. 사용시스템: Ubuntu 12.04LTS, Windows7, Eclipse, vim, ctags, source explorer, Android adb, Monkey Runner, Google Voice Recognition

3. 사용된 공개 SW: FLAC Encoder, Tesseract OCR, Android Open Source Project

# 2.5 해당 개발과제 진행 시 애로 사항(불량 및 해결방법)

1. 음성만으로는 정확한 정보 전달이 어려울 수 있으므로 터치, 동장인식 등 다른 입력 방식과 음성인식을 결합해서 활용하는 방안도 모색해야 할 필요가 있다.

2. 발음에 개인차가 있다. 같은 말을 해도 사람에 따라서 발음이 다르기 때문에, 그 음향 특성(주파수 스펙트럼 등)에 차이가 생긴다.

3. 주위 환경에 영향을 많이 받는다. 예를 들어 주변에 소음이나 잡음이 많은 경우 화자의 음성과 주변 잡음이 섞여 인식되기 때문에 인식률이 낮아진다.

4. 개발보드인 ODROID-U2의 음성Input에 잡음이 심해 플랫폼 키를 활용하지 못하고 MonkeyRunner를 통해 우회하는 방법을 사용하였다. 때문에 일반사용자에게 배포할 때는 문자인식 등의 기능을 사용하지 못한다.

5. 문자인식의 시간이 오래 걸린다.

# 2.6 향후 과제개선 목표 및 보완사항

1. 주변 소음과 화자의 음성을 구별하여, 주변 소음은 필터를 하고 화자의 음성에 반응하도록 개선하면 더욱 좋은 서비스를 제공할 수 있다.

2. 화자가 휴대폰을 사용하지 않을 때 Service를 작동 중지시키고 사용할 때 자동 Service를 실행이 되도록 하여 베터리 절약을 할 수 있다.

3. 현재는 MonkeyRunner를 사용하여 문자인식과 Button 이벤트를 처리하기 때문에 서버와 연동이 필요하지만, 플랫폼 키를 획득하면 서버를 통한 간접실행이 아닌 App 자체로 컨트롤 할 수 있다. 이 점을 개선해야 할 것이다.

# 2.7 참조 소스 및 출처

1. 문화기술(CT) 심층리포트: 음성인식 기술의 동향과 전망, 2011.11

2. 신성장동력산업용 대어휘 음성인식 기술 동향 및 응용, ETRI, 전자통신동향분석 제23권 제1호 2008권 2월

3. 제2장 음성정보처리기술

4. GMM 모델링에 기초한 음색의 특징 연구, 이은, 충남대학교 인공지능연구실

5. HMM 음성 인식 알고리즘의 분석, 장순석, 조선대학교 제어계측로봇공학과

6. C#언어를 이용한 음성분석시스템의 구현, 오지형, 한밭대학교 컴퓨터공학과

7. Google Search by Voice: A case study, Johan Schalkwyk, Google Inc.

8. Speaker Recognition, JOSEPH P. CAMPBELL, JR., SENIOR MEMBER, IEEE

9. AudioFlinger, 音频系统

10. 안드로이드 구조와 원리, 이백, 수원 안드로이드 스터디

11. <http://dtmilano.blogspot.kr/>

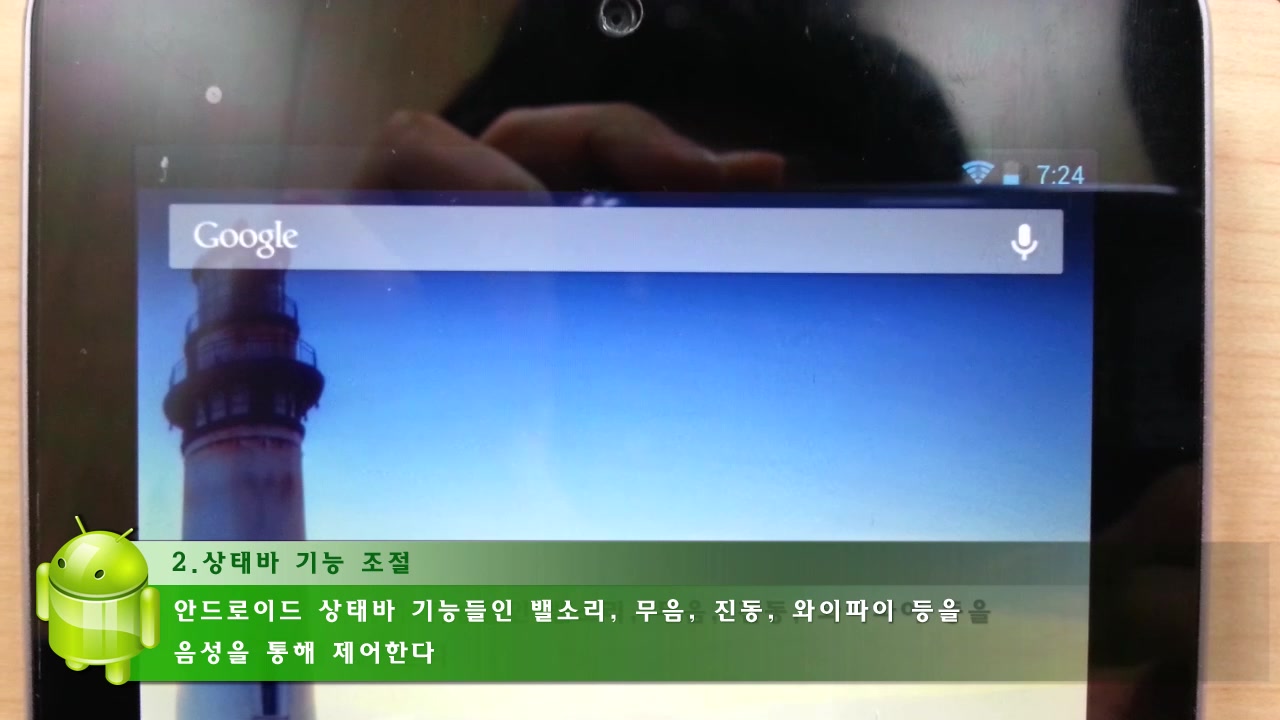
12. <http://research.google.com/pubs/SpeechProcessing.html>

3. 개발결과물 사진

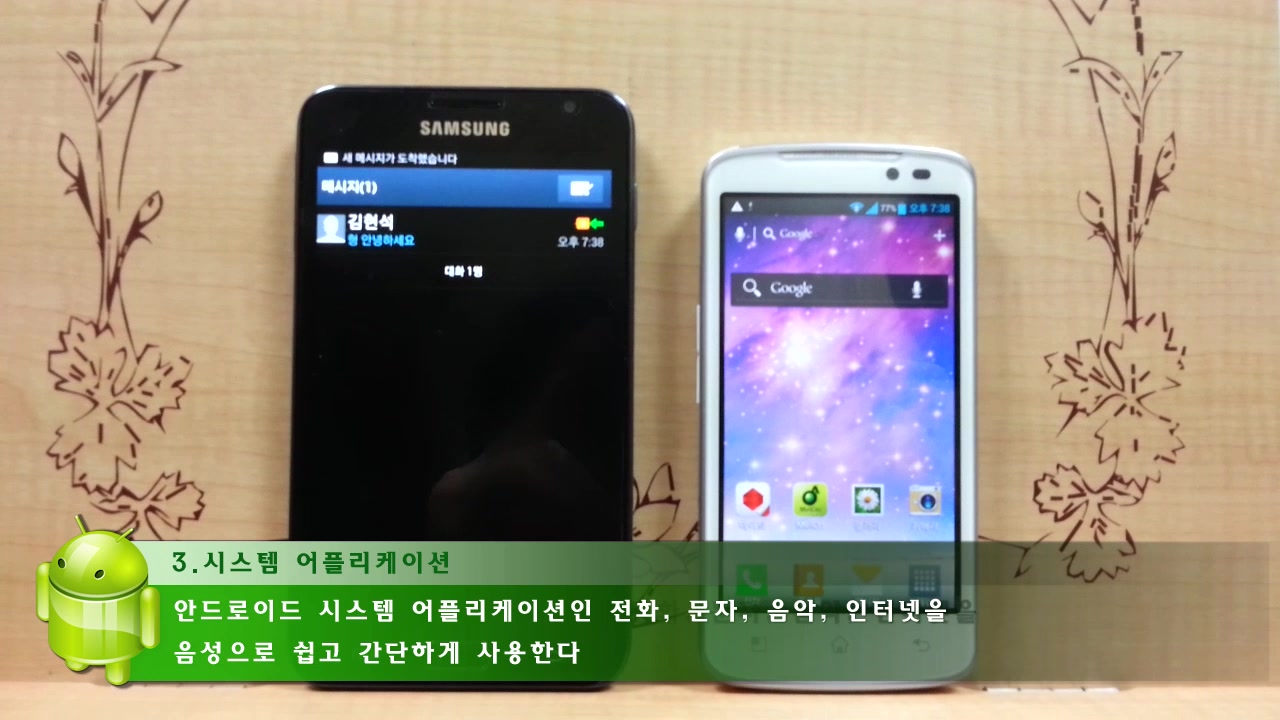
# 3.1 전화 받기 & 거절



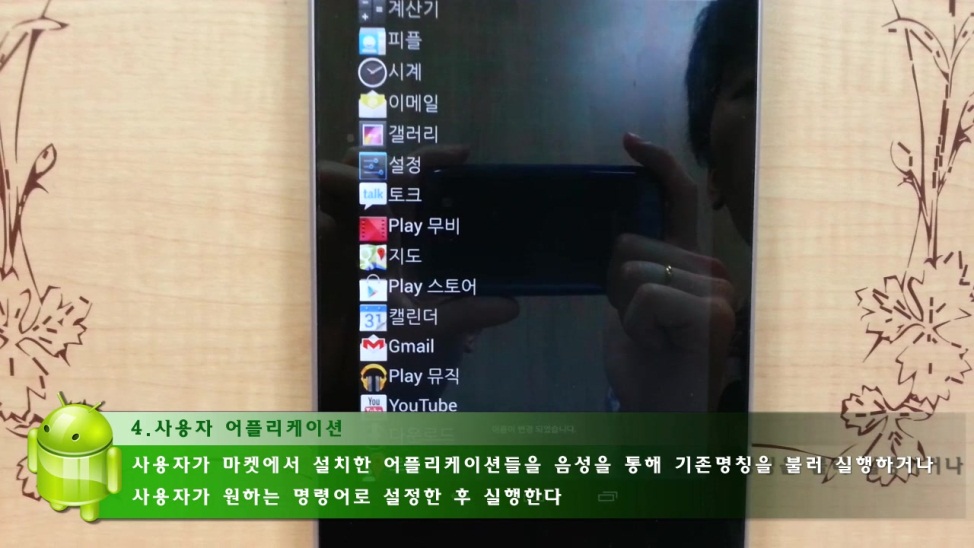
# 3.2 상태바 기능 조절



# 3.3 시스템 어플리케이션 컨트롤



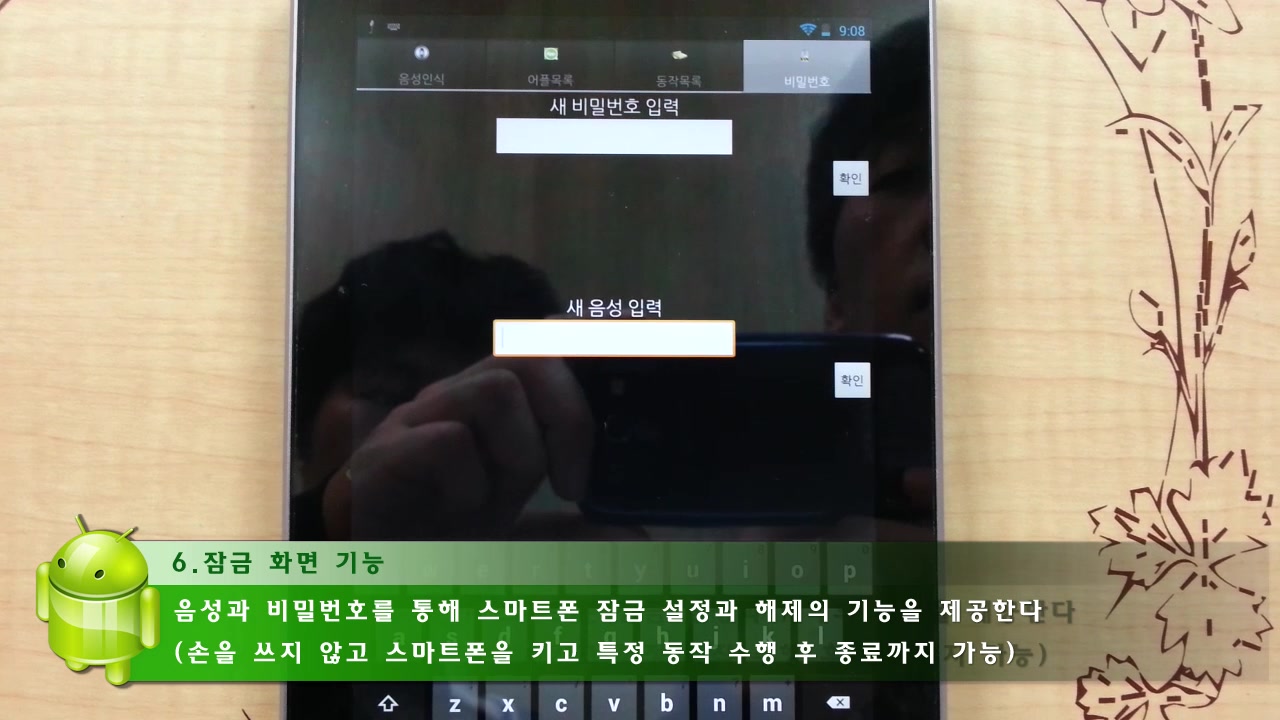
# 3.4 사용자 어플리케이션 컨트롤



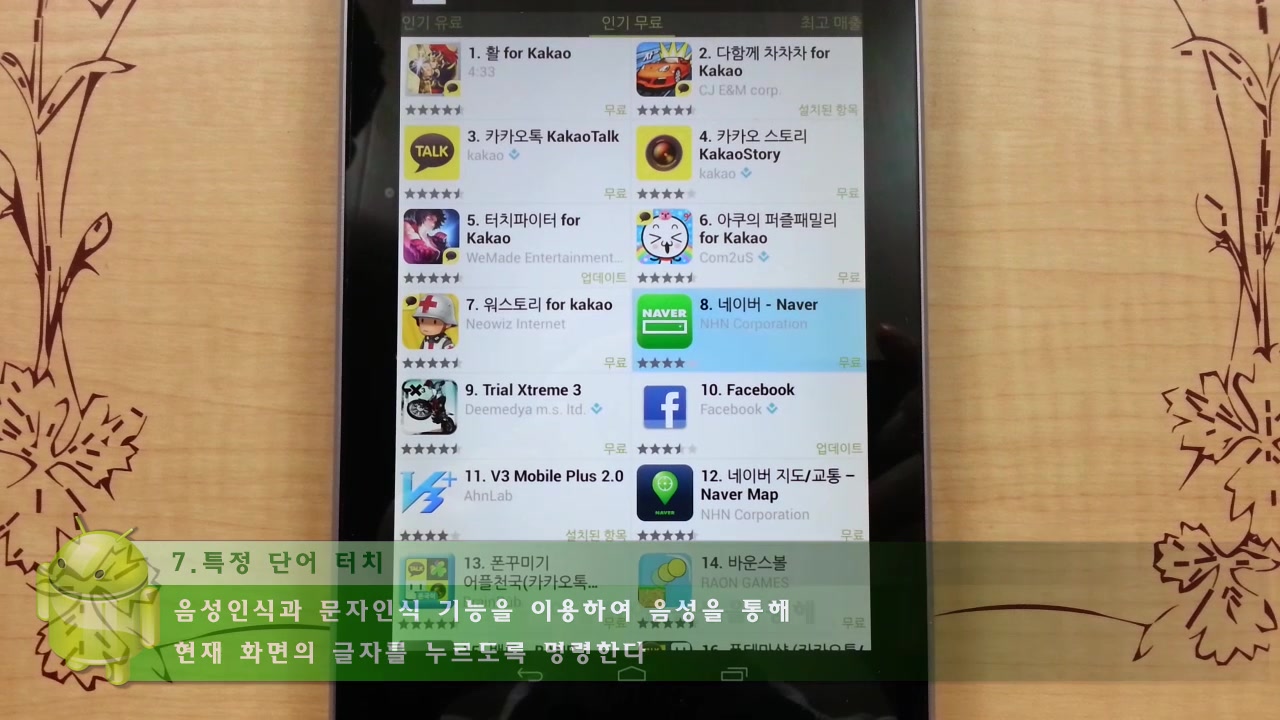
# 3.5 하드웨어 기능 조절



# 3.6 잠금 화면 기능



# 3.7 특정 단어 터치 이벤트



4. 개발결과물 동영상

동영상 파일이 커서 성함과 연락처를 기재하여 [**samsungexynos@all-f.com**](mailto:samsungexynos@all-f.com)로 첨부 등록하여 보냈습니다.