**Final lab**

**for**

**<씽쏭씽쏭>**

**<Hyeongil>**

**< June 14.2016>**

모바일 프로그래밍 II/모바일시스템 종합설계

32112186 고영일

32112219 황형언

유시환 교수님

**목차**

1. **소개 3**
   1. 선정배경
   2. 목표
2. **개발 4**
   1. 타임라인
   2. 개발 개괄
      1. 개발 내용
      2. 개발 환경
      3. 개발 언어
      4. 개발 도구
   3. 개발 내용
      1. 클래스 다이어그램
      2. 클래스 개요
      3. 클래스 세부내용
      4. 코드 내용
3. **결과 13**
4. **결론 15**
5. **참고문헌 16**
6. **후기**
7. **전체 코드**
8. **소개**

**1.1 선정 배경**

이번 실습 장비인 HBE-SM5-S4210에는 accelerometer sensor, gyro sensor, proximity sensor, 4x4 keypad, textlcd, dip switch 등이 장착되어있다.

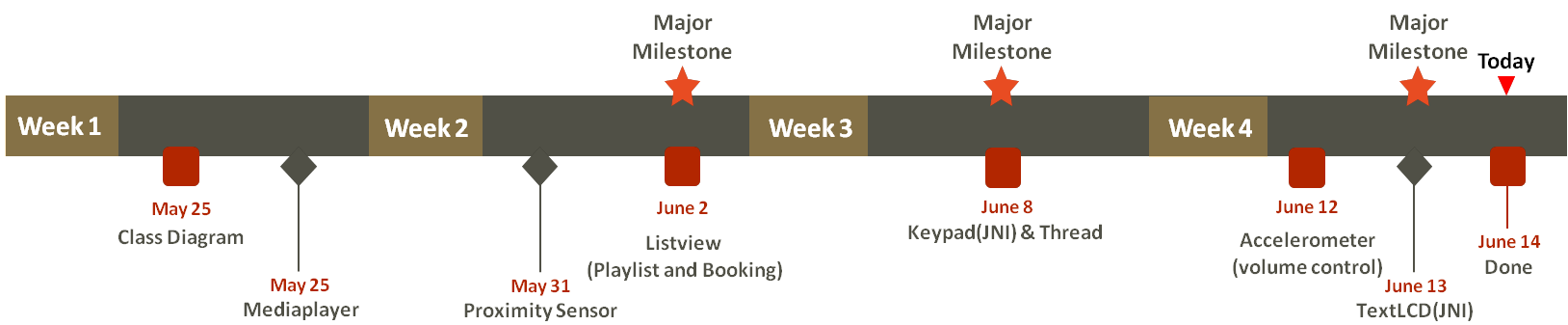
이것들을 최대한 활용할 수 있는 어플리케이션을 생각하다가 4x4 keypad, textlcd, proximity sensor, accelerometer sensor를 이용한 노래방 어플리케이션을 제작하려고했다.

* 1. **목표**

사용자는 어플리케이션 안에 저장된 노래를 예약 버튼을 눌러서 예약리스트에 담을 수 있다. 예약 확인 버튼을 눌러서 예약 리스트를 확인할 수 있고 시작 버튼을 눌러서 예약 리스트에 담긴 노래들을 차례로 재생할 수 있다. 예약과 시작 기능 등을 키패드에 있는 버튼을 이용하여 노래방의 리모콘 같은 기능을 수행하도록 한다. 초기에 오른쪽 위에 있는 Textlcd 1행, 2행에 각각 title, singer라는 문자열이 나오고 노래가 재생되면 노래에 맞는 제목과 가수가 나오게 된다. Proximity sensor를 이용하여 노래를 재생/정지 시킬 수 있다. 장비를 기울여 Accelerometer sensor로 볼륨을 올리거나 내릴 수 있다.

1. **개발**

**2.1 타임라인**

****

**2.2 개발 개괄**

**2.2.1 개발 내용**

* 노래방 어플리케이션 개발 및 커널 수정

|  |  |
| --- | --- |
| CPU | Intel ® Core™ i7-2600 CPU @ 3.40GHz |
| RAM | 8GB |
| OS | Windows 7 32bit |
| Android-SDK | 2.3.3 |
| Android-API | 10 |

**2.2.2. 개발 환경**

* PC :
* HBE-SM5-S4210

|  |  |
| --- | --- |
| Processor | Samsung S5PC210  (ARM Cortex ™ - A9 Dual Core) |
| RAM | 1GM LPDDR2 |
| System Software | Uboot 1.3.4, Linux Kernel 2.6.35.7,  Android 2.3.5(Gingerbread) |

**2.2.3. 개발 언어**

- Java, C(NDK build)

**2.2.4 개발 도구**

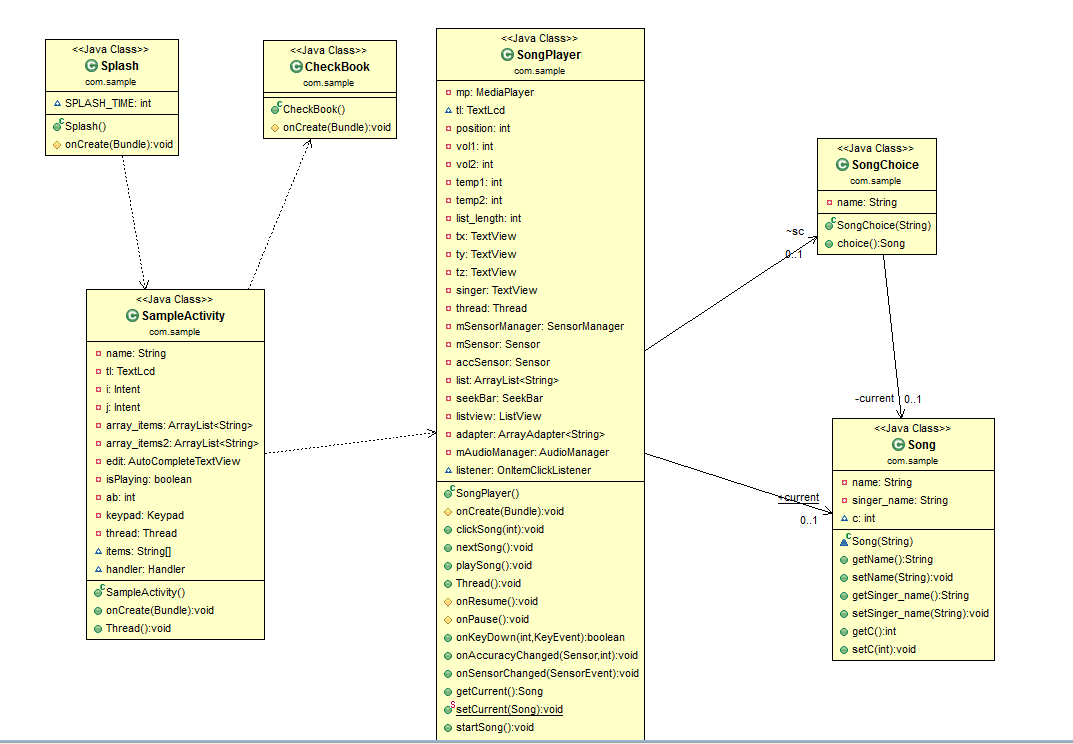
-Eclipse Indigo

- Visual Studio

- Putty

**2.3 개발 내용**

**2.3.1 클래스 다이어그램**



**2.3.2 클래스 개요**

* SampleActivity 클래스
  + SampleActivity : 어플리케이션의 메인 액티비티 역할을 한다. CheckBook클래스와 SongPlayer 사이에 놓여서 이어주는 역할을 한다.
  + Thread : 키패드를 입력 받기 위해 쓰레드를 실행시키고 노래가 시작되며 액티비티가 전환될 때 쓰레드를 종료시킨다.
* SongPlayer 클래스
  + SongPlayer : 음원이 재생되는 클래스로서 Proximity sensor, accelerometer sensor 등의 장치들에 의하여 컨트롤 되는 곳이며 이 어플리케이션에서 가장 중요하게 작용한다.
  + clickSong : 리스트의 노래를 클릭했을 때 실행되는 메서드
  + nextSong : 리스트의 다음 노래를 지정하기 위한 메서드
  + playSong : 노래를 재생시키기 위한 메서드
  + MediaPlayer의 onCompletion : setOnCompletionListener안에서 노래가 끝났을 때 끝난 노래를 리스트에서 삭제한다. 그리고 nextSong 메서드를 호출하여 다음 노래로 넘어간다. 만약 리스트가 모두 삭제되었을 때는 액티비티를 종료한다.
  + Thread: seekbar를 조작했을 때 노래가 그 위치에 맞게 재생되도록 한다.
  + onKeyDown: 이전 액티비티의 기록을 안남기기 위하여 FLAG\_ACTIVITY\_NO\_HISTORY 매크로를 사용하고 재생된 노래는 예약 리스트에서 삭제한다.
  + onSensorChanged: Sensor의 값들을 읽어온다. Accelerometer sensor를 이용하여 볼륨 조절을 하고, Proximity sensor를 이용하여 음악을 재생/정지시킨다.
* Song 클래스
  + Song : 노래에 대한 정보를 담고 있는 클래스
* SongChoice 클래스
  + SongChoice : 음원 파일과 필드의 이름을 매칭시켜주는 클래스
* CheckBook 클래스
  + CheckBook : 노래 예약 리스트에 대한 클래스
* Splash 클래스
  + Splash : 어플리케이션 로딩 화면에 대한 클래스

**2.3.3 클래스 세부내용**

* SampleActivity

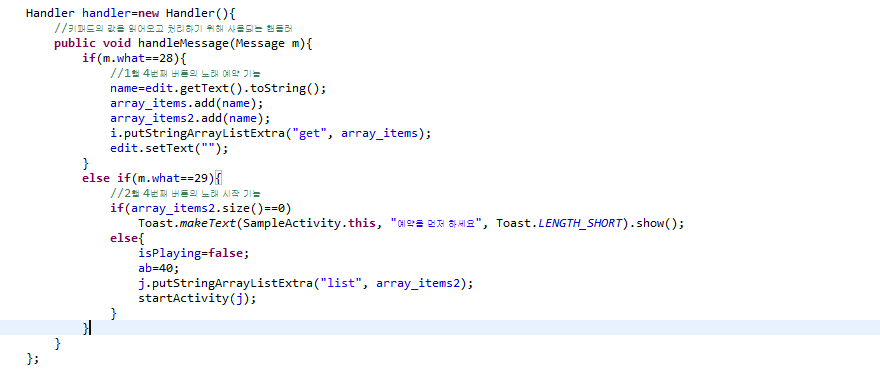
|  |  |  |
| --- | --- | --- |
| 멤버 변수 | 타입 | 설명 |
| t1 | TextLcd | 최초에 textlcd 1행에 “title”, 2행에 “singer”라는 문자열을 출력 |
| i, j | Intent | ArrayList의 노래를 i는 CheckBook클래스로 j는 SongPlayer 클래스로 넘겨줌 |
| keypad | Keypad | Keypad의 입력값을 받음 |
| items | String[] | 모든 노래 목록을 가지고 있음 |
| handler | Handler | 키패드의 값을 읽어오고 처리하기 위해 사용 |

* SongPlayer

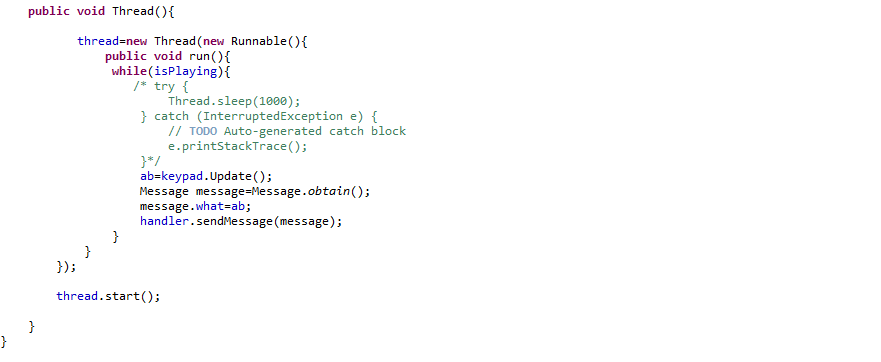
|  |  |  |
| --- | --- | --- |
| 멤버 변수 | 타입 | 설명 |
| mp | MediaPlayer | 노래를 저장 |
| t1 | TextLcd | 재생 중인 노래의 제목과 가수를 저장 |
| adapter | ArrayAdapter<String> | 노래 목록을 위한 어댑터 |
| accSensor | Sensor | 가속도 센서 변수 |
| mAudioManager | AudioManager | 볼륨 컨트롤을 위한 변수 |
| mSensorManager | SensorManager | 센서 컨트롤을 위한 변수 |

**2.3.4 코드 내용**

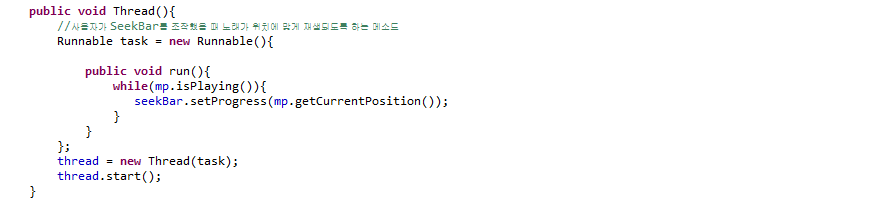
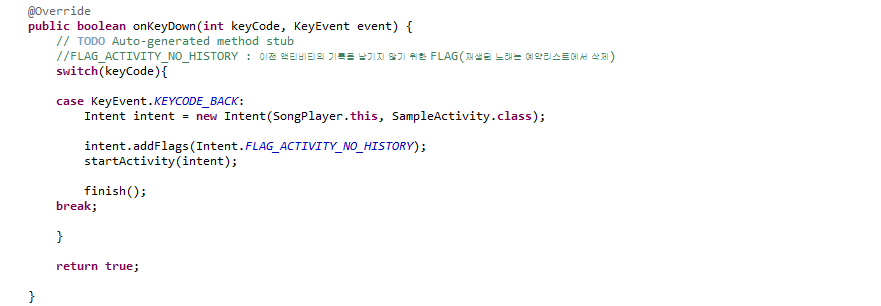
* SampleActivity.java



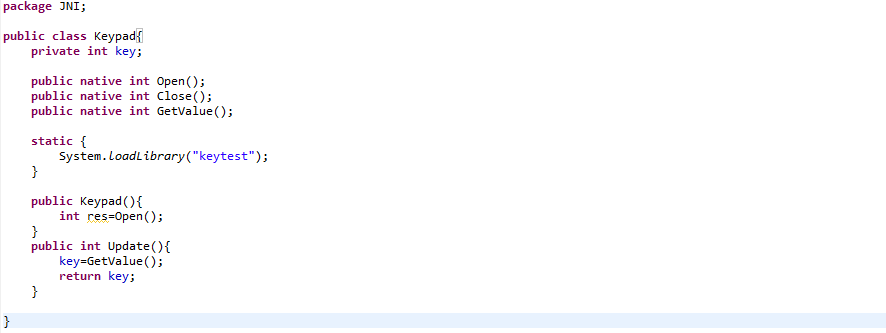
* SongPlayer.java

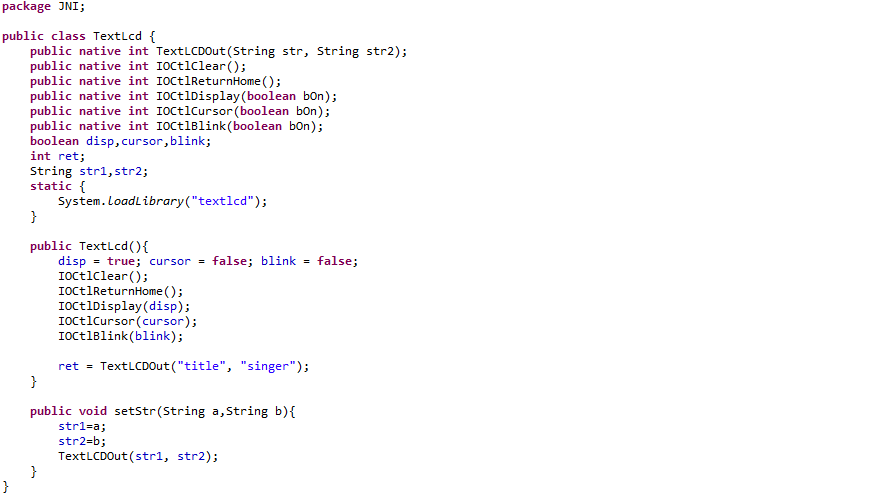
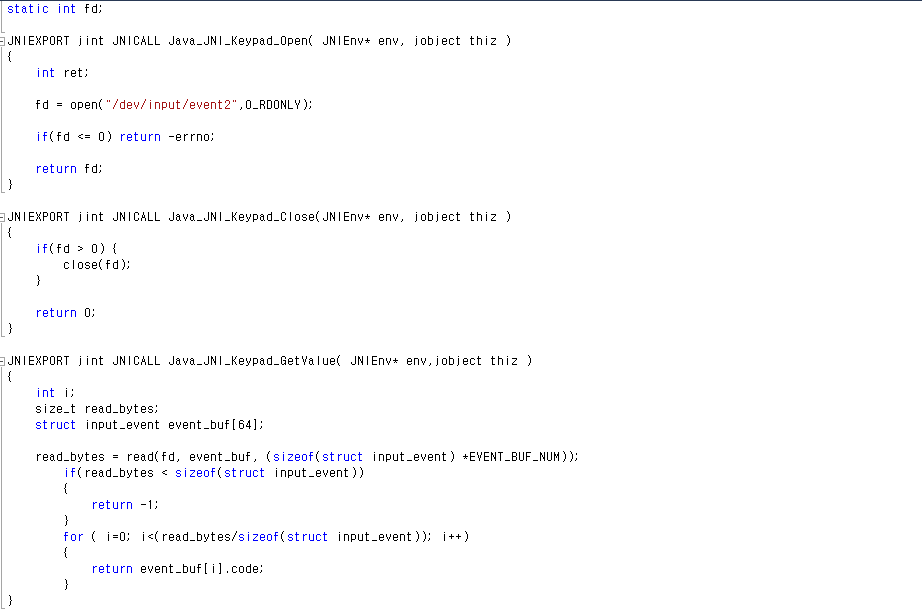


* JNI package



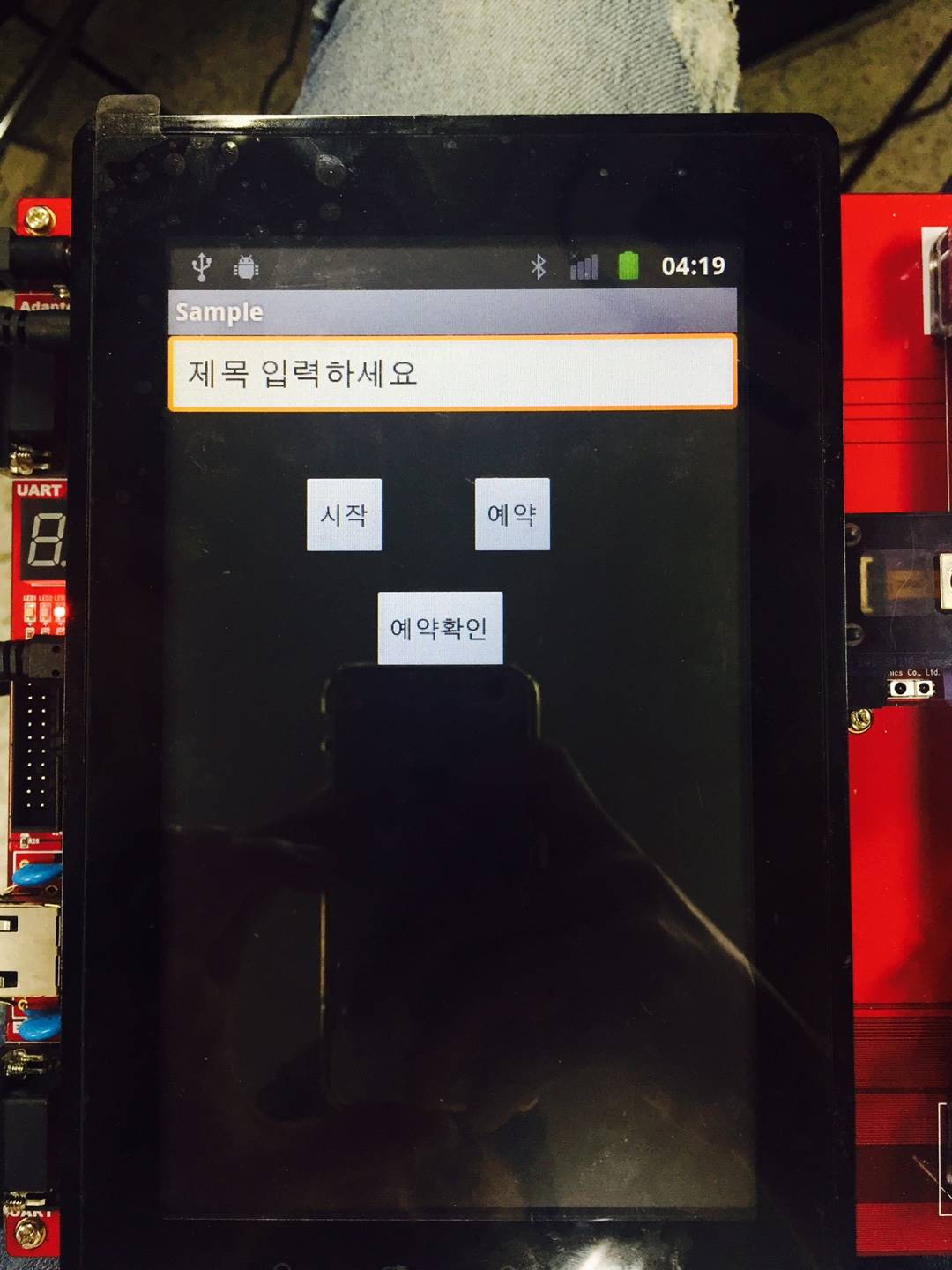
* + Keypad.java



* TextLcd.java
* Keytest.c

1. **결과**

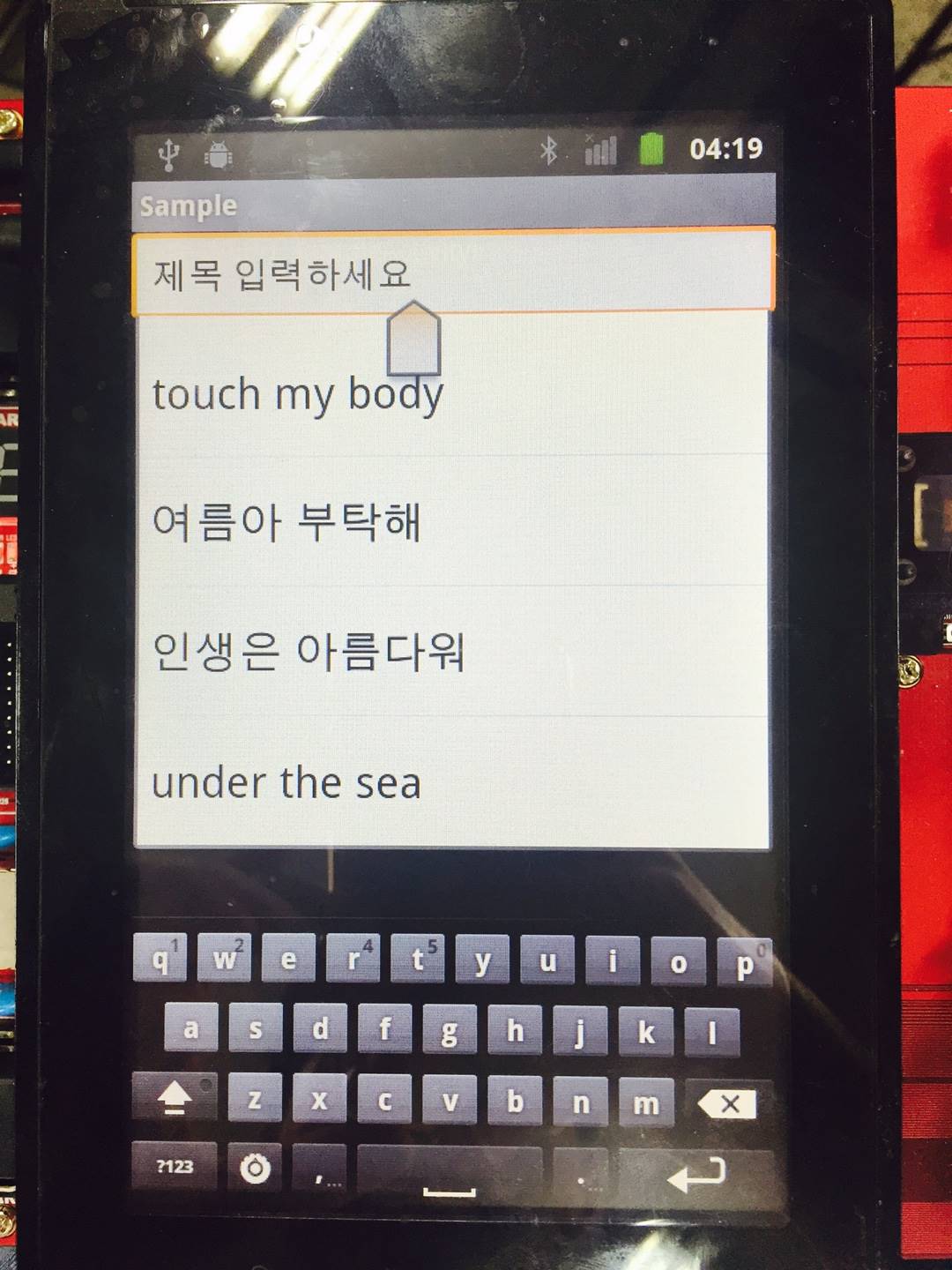
노래의 제목 입력



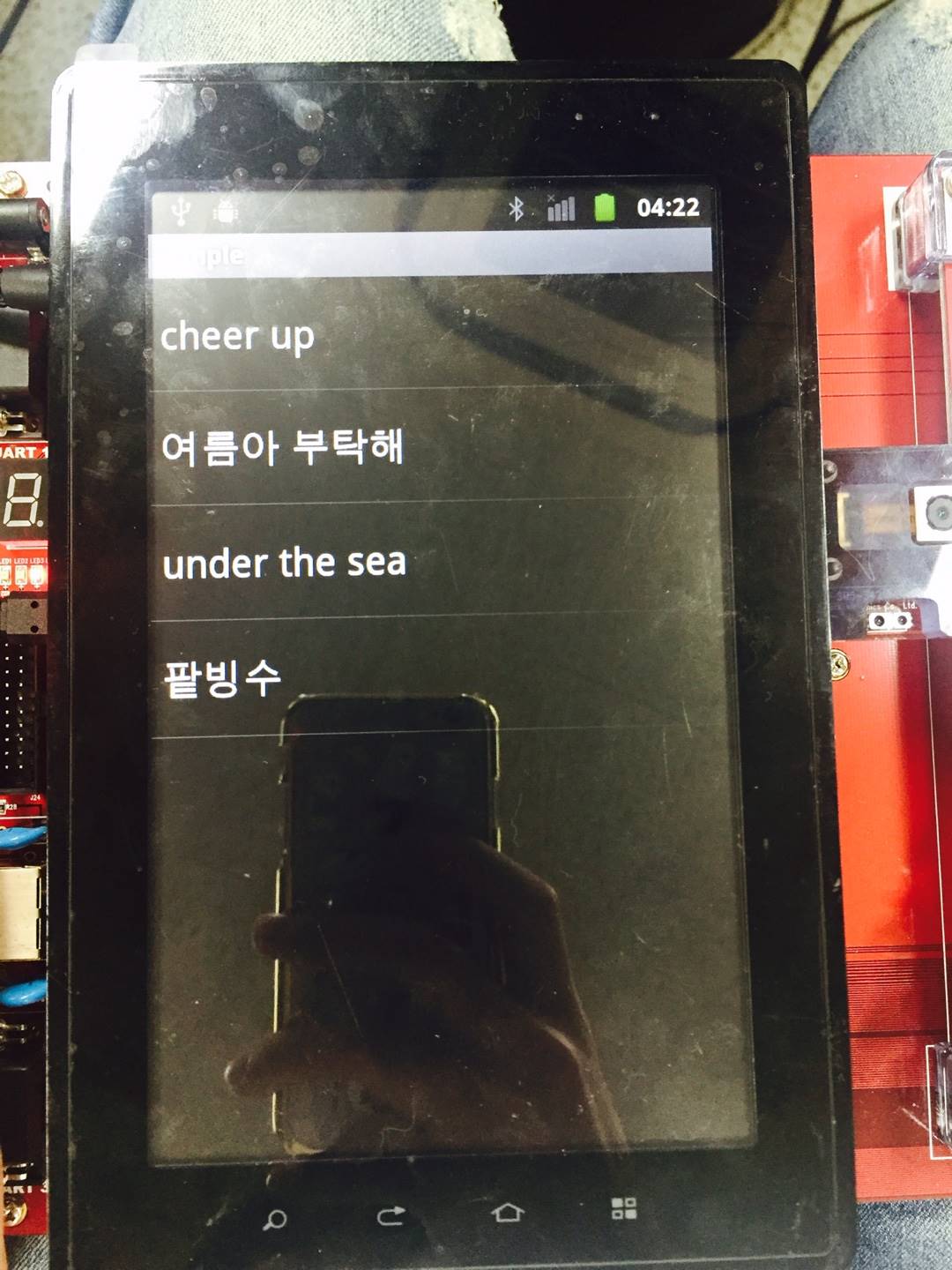
시작 : 노래를 바로 재생

예약 : 노래를 예약 리스트에 저장

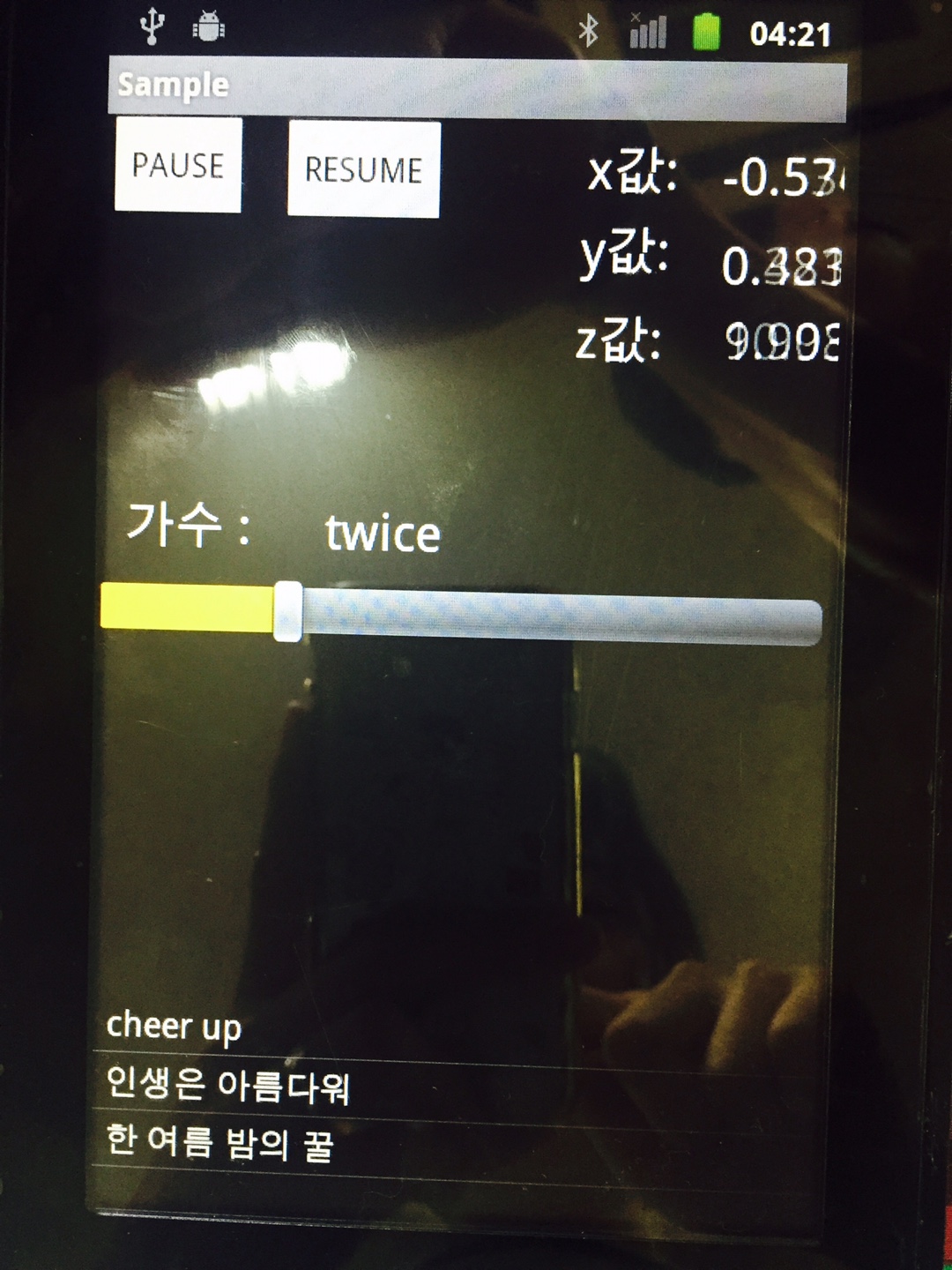
예약 확인 : 예약 리스트를 직접 확인



노래 검색 결과



노래 예약 리스트



노래 예약 리스트

현재 재생 곡



현재 재생 곡

1. **결론**

기본적인 기능을 어플리케이션에서 수행하고 커널 하드웨어 장치들을 최대한 많이 이용하려고 했다. 그리고 어플리케이션에서 장치로 혹은 장치에서 어플리케이션으로 요청되는 서비스를 수행하는 것이 기본 구조이다. 안드로이드 어플리케이션 개발은 Java 언어를 사용하기 때문에 처음에는 어렵지 않을 것이라 생각했다. 하지만 커널과 어플리케이션 사이에 있는 JNI에서 난관이 많았다. Keypad으로부터 원하는 값을 받을 때 값이 자꾸 틀려서 커널의 키 매크로 값을 바꿔봤지만 별 소득이 없었다. adb shell chmod 777를 안해줘서 반영이 되지 않았고 Keypad의 입력값을 바꿀 수 있었다. 게다가 장치의 API가 현재 나와있는 API에 비해서 꽤 낮은 버전이어서 개발하는 데 있어서 제한 사항이 종종 존재했다. 일례로 accelerometer sensor를 사용하는 함수에서 쓰이는 파라미터 중에 변화하는 가속도 값을 얼마나 자주 받는지에 대한 delay가 있었는데 API가 낮은 관계로 오직 4개의 매크로만 쓸 수 있었다. API가 높았더라면 delay를 원하는 값으로 넣을 수 있었다.

1. **참고문헌**

- HBE-SM5-S4210으로 배우는 안드로이드 개정판 – ㈜ 한백전자 기술연구소

- 안드로이드 개발자 홈페이지 <http://developer.android.com/index.html>

- <http://stackoverflow.com/> 에서 참고한 많은 답변들

1. **후기**

주제를 나열하고 선택하는데 시간이 오래 걸려 프로젝트를 다소 늦게 시작하게 되었다. 안드로이드 어플리케이션은 능숙하게 사용하고 있었지만 안드로이드 어플리케이션 개발 경험은 거의 없었다. 그러나 이번 프로젝트를 시작하면서 막연하게 생각하였던 Java와 d안드로이도 개발 능력이 크게 향상되었다는 것을 느꼈다. 특히 Java와 Eclipse는 거의 처음 접 하는 상황이라 시행착오도 많았지만, 전공 과목으로 수강하였던 모바일 프로그래밍 I 과목이 많은 도움이 되었다. 학부 생활 동안 이렇게 한 프로그램을 오랫동안 작성해본 적이 없었다. 그래서 처 음에는 부담도 많이 되었지만, 이제는 큰 프로젝트를 접하더라도 막연한 두려움을 갖지 않게 될 것 같다

1. **전체 코드**

<https://github.com/ever0101/singsongsingsong>