임베디드 시스템 설계
FINAL PROJECT

2019310272 이명지

Index

I. Project Design

II. Project configuration

- JNI: Dotmatrix, Led, Fullcolorled, Textlcd, Piezo

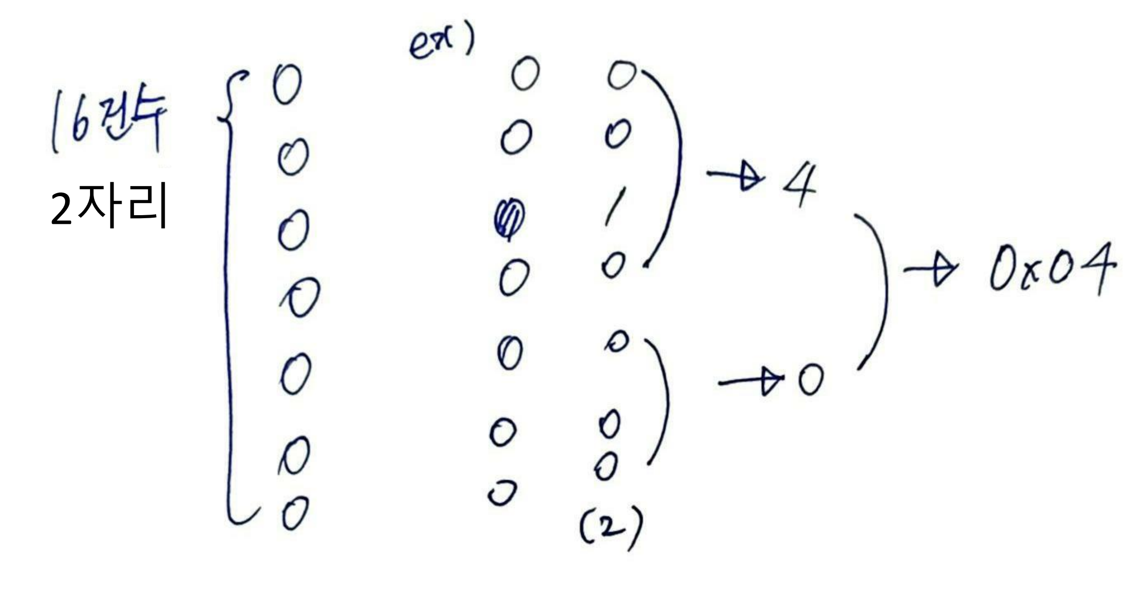
III. MainActivity.java

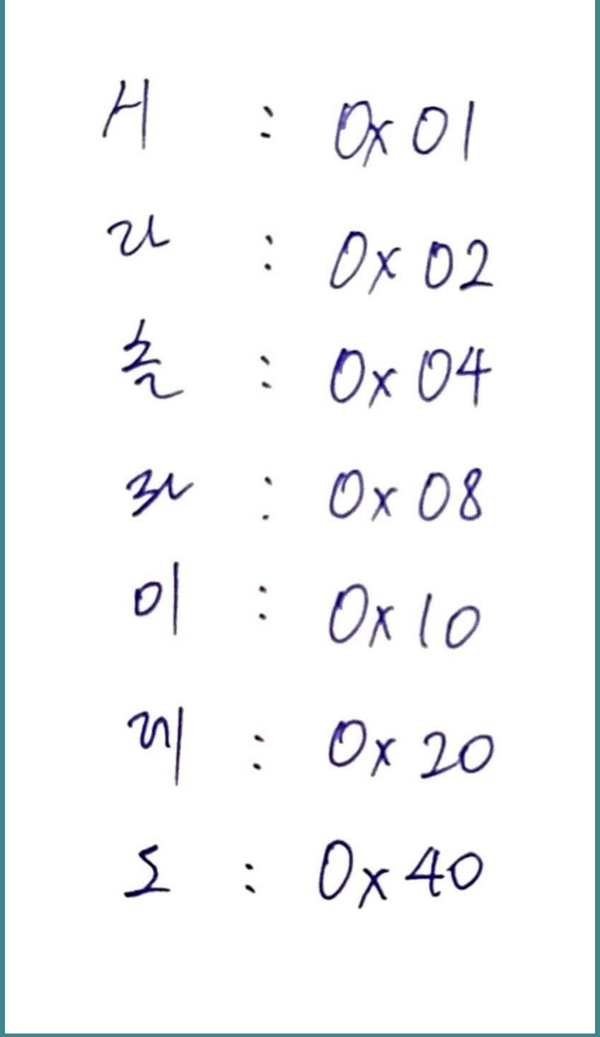
IV. Demonstration Video

V. To Improve…

Project Design

<리듬 게임>



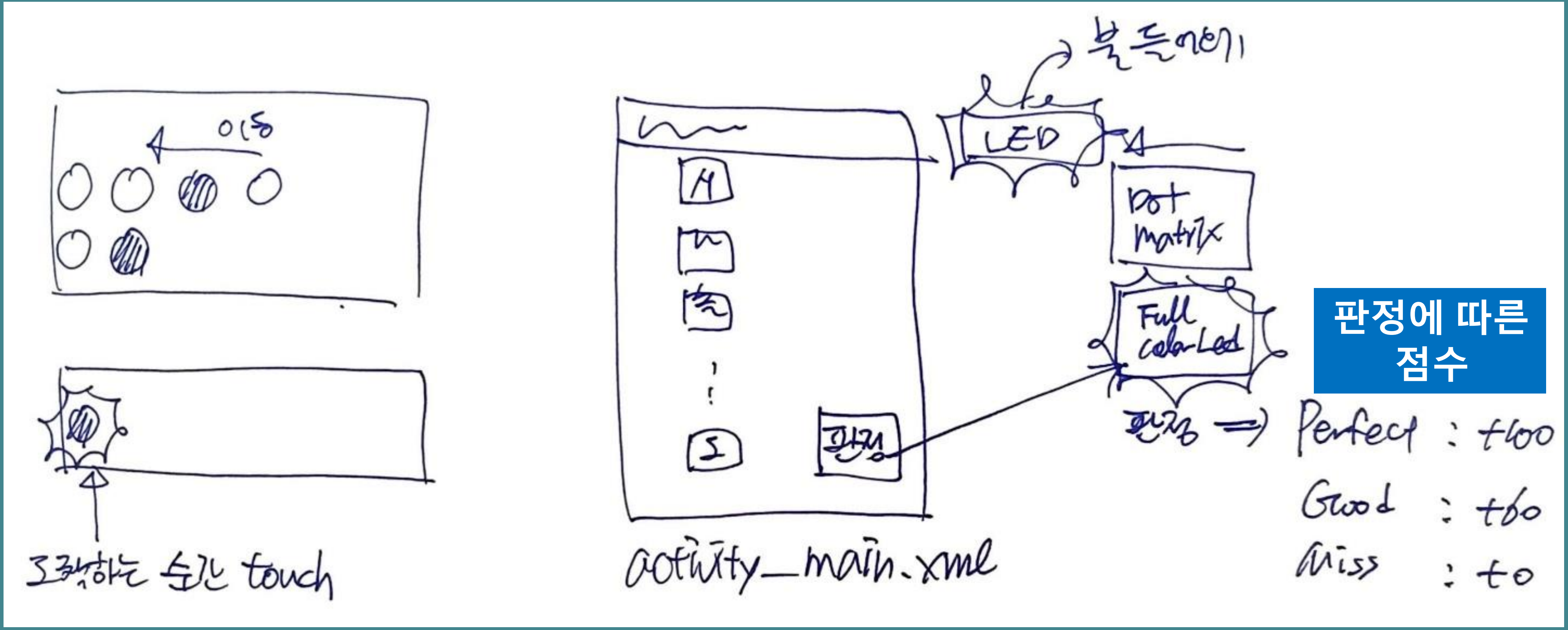


시	:	0x01
라	:	0x02
솔	:	0x04
파	:	0x08
미	:	0x10
레	:	0x20
도	:	0x40

DotMatrix의 입력: 16진수(string)

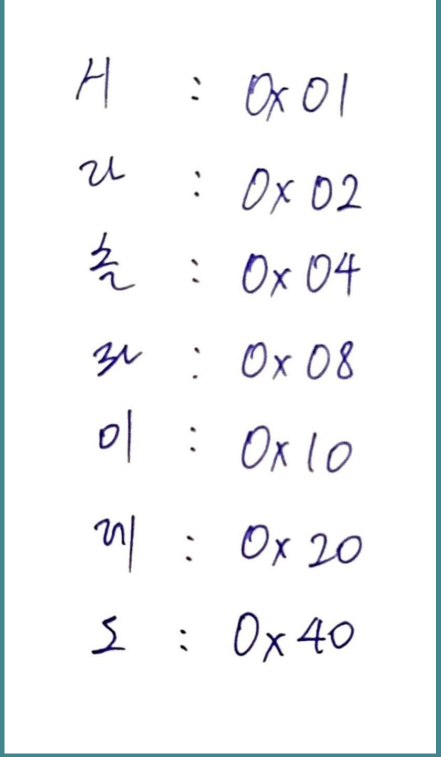
DotMatrix의 row: 7개 -> 도, 레, 미, 파, 솔, 라, 시 에 대응

Project Design



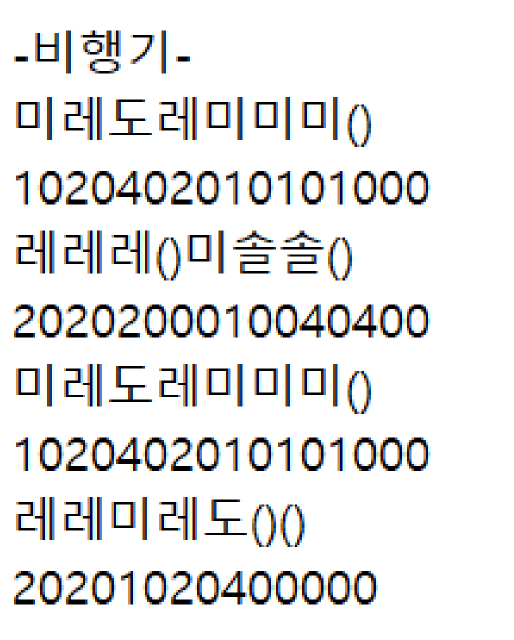
판정에 따른 점수

Project Design

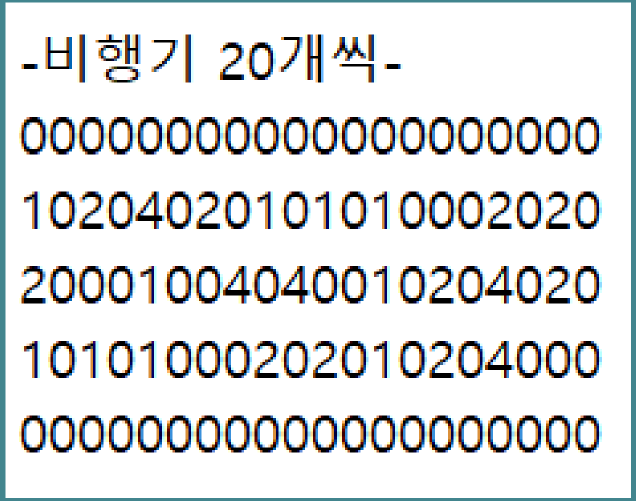


시	:	0x01
라	:	0x02
솔	:	0x04
파	:	0x08
미	:	0x10
레	:	0x20
도	:	0x40

악보 만들기

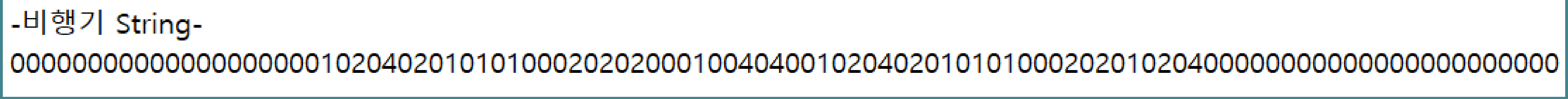


```
-비행기-
미레도레미미미()
1020402010101000
레레레()미솔솔()
2020200010040400
미레도레미미미()
1020402010101000
레레미레도()()
20201020400000
```



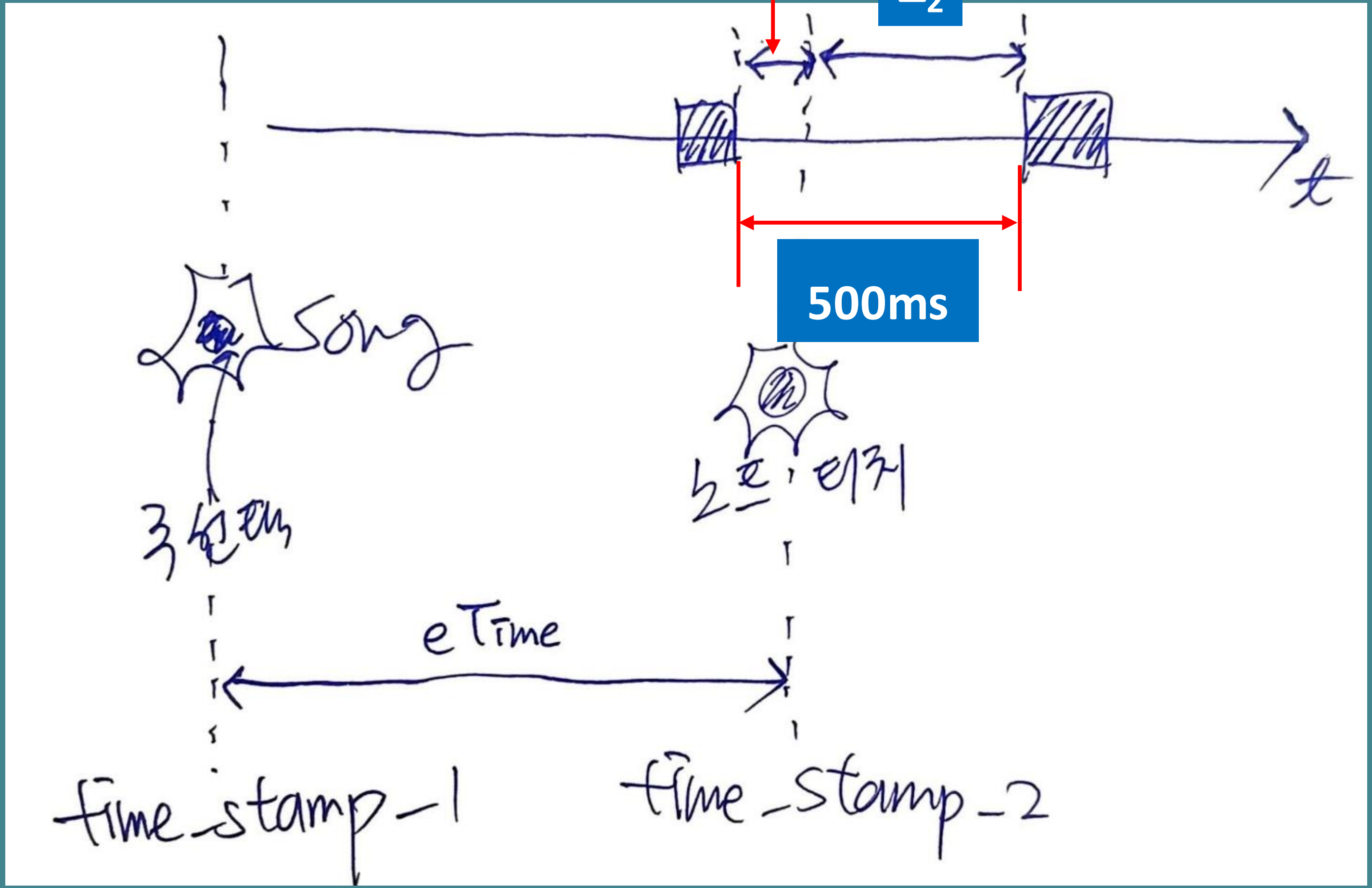
```
-비행기 20개씩-
00000000000000000000
10204020101010002020
20001004040010204020
10101000202010204000
00000000000000000000
```

최종 악보



```
-비행기 String-
00000000000000000000102040201010100020202000100404001020402010101000202010204000000000000000000000000
```

Project Design

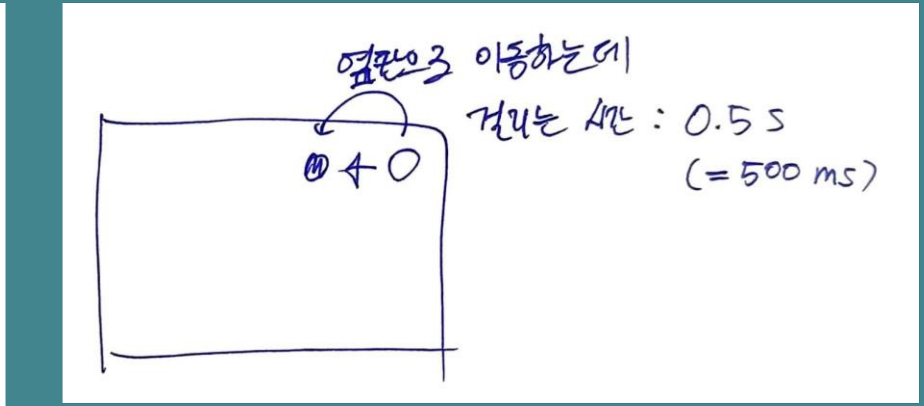


Δ_1

Δ_2

-> 더 작은 Δ_1선택

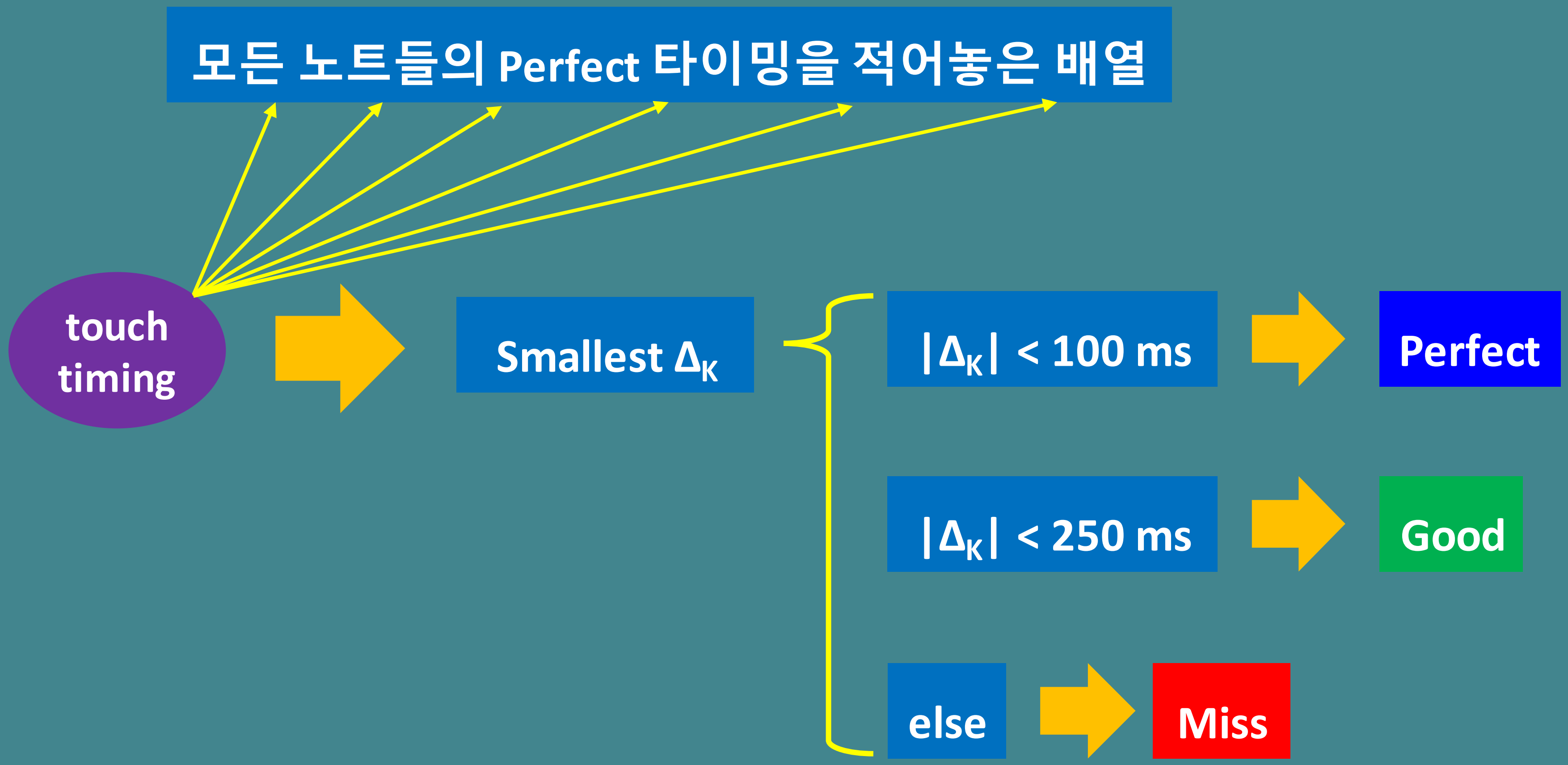
500ms



노트가 옆으로 한 칸 이동하는 시간이

500ms임을 이용하여 판정

Project Design



모든 노트들의 Perfect 타이밍을 적어놓은 배열

touch timing

Smallest Δ_K

$|\Delta_K|$ < 100 ms

Perfect

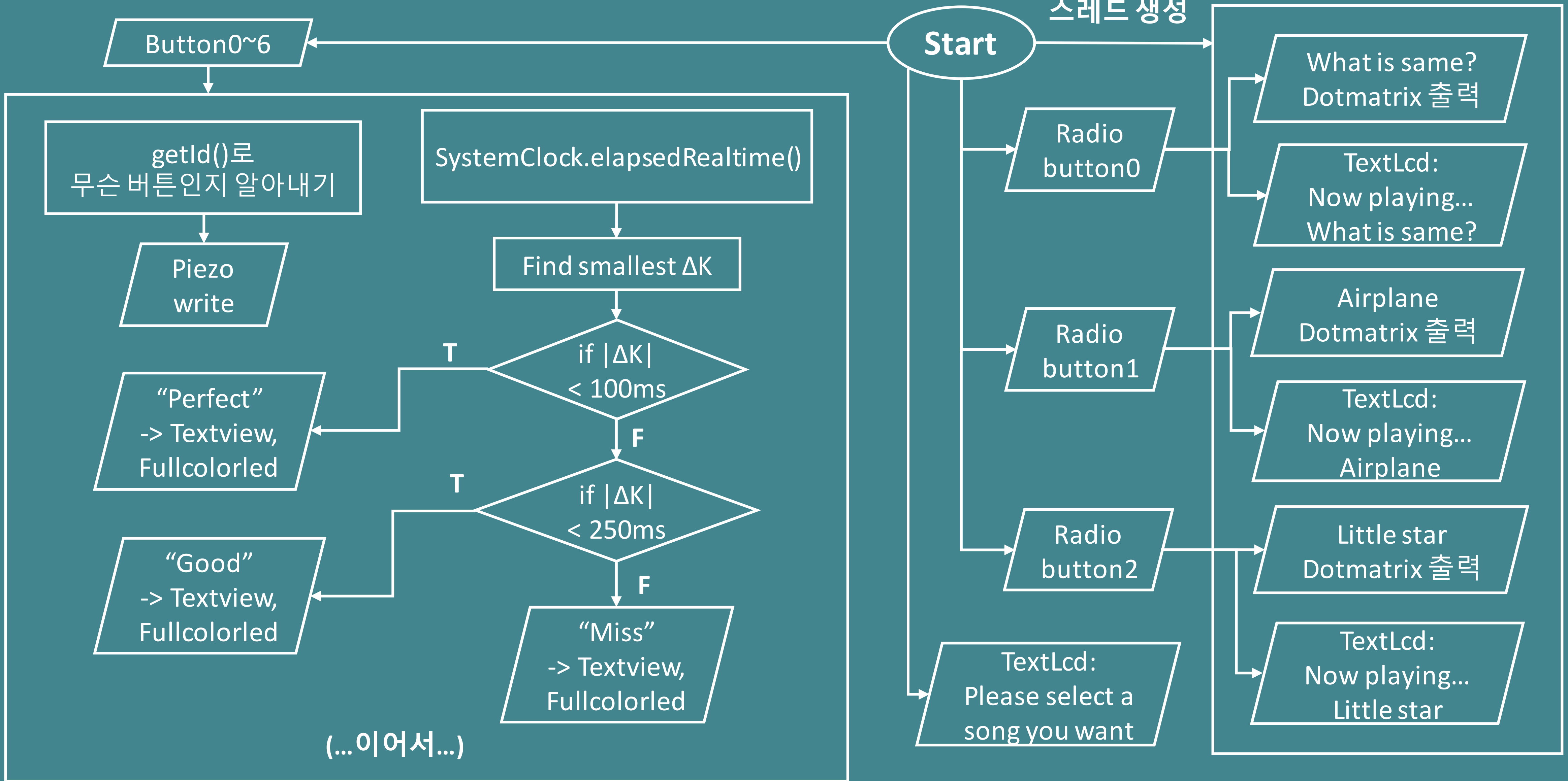
$|\Delta_K|$ < 250 ms

Good

else

Miss

Project Design – Block Diagram (Flow Chart)



Start

스레드 생성

Button0~6

getId()로
무슨 버튼인지 알아내기

Piezo
write

SystemClock.elapsedRealtime()

Find smallest ΔK

if |ΔK|
< 100ms

T

“Perfect”
-> Textview,
Fullcolorled

F

if |ΔK|
< 250ms

T

“Good”
-> Textview,
Fullcolorled

F

“Miss”
-> Textview,
Fullcolorled

(…이어서…)

Radio
button0

What is same?
Dotmatrix 출력

TextLcd:
Now playing…
What is same?

Radio
button1

Airplane
Dotmatrix 출력

TextLcd:
Now playing…
Airplane

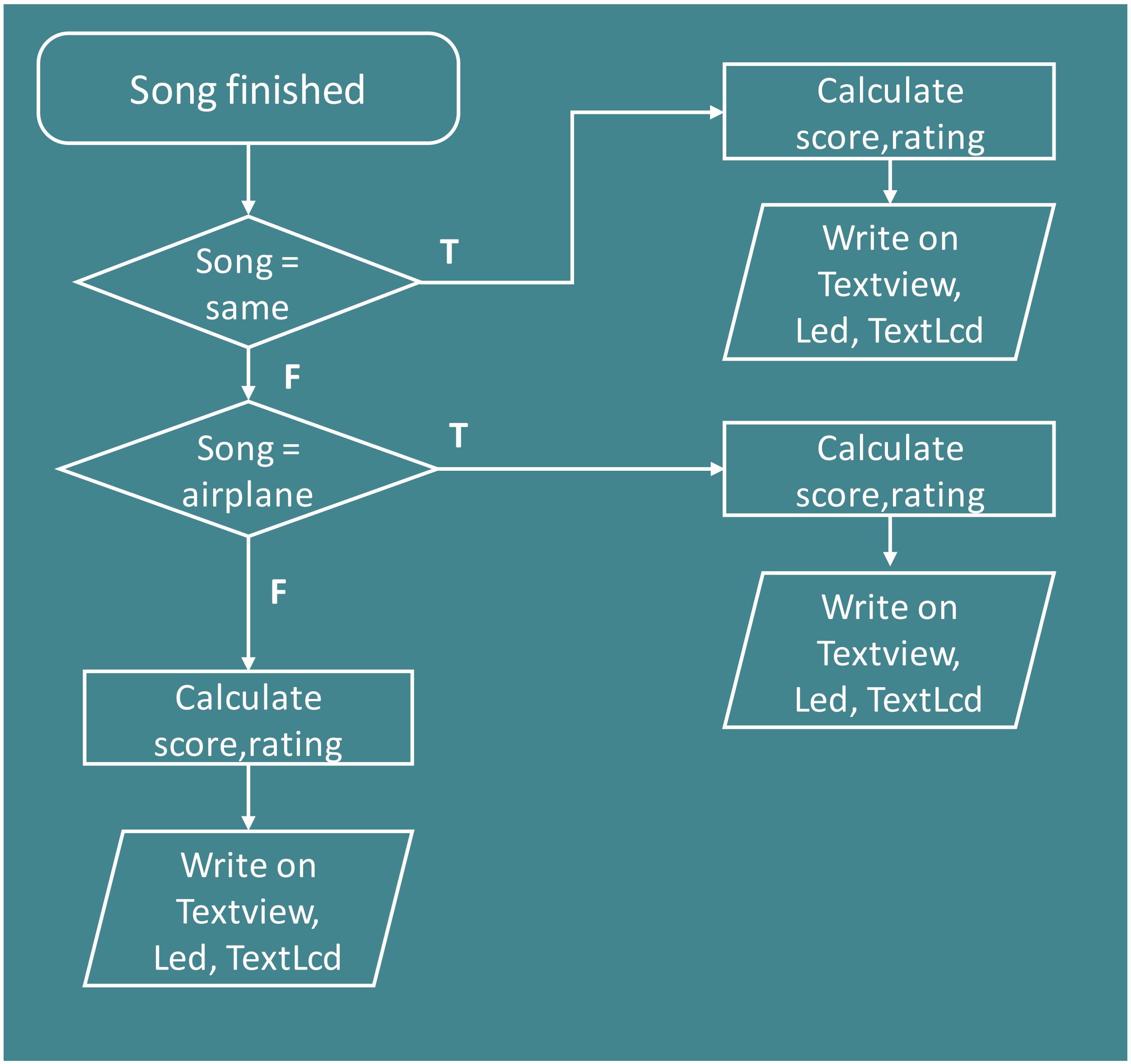
Radio
button2

Little star
Dotmatrix 출력

TextLcd:
Now playing…
Little star

TextLcd:
Please select a
song you want

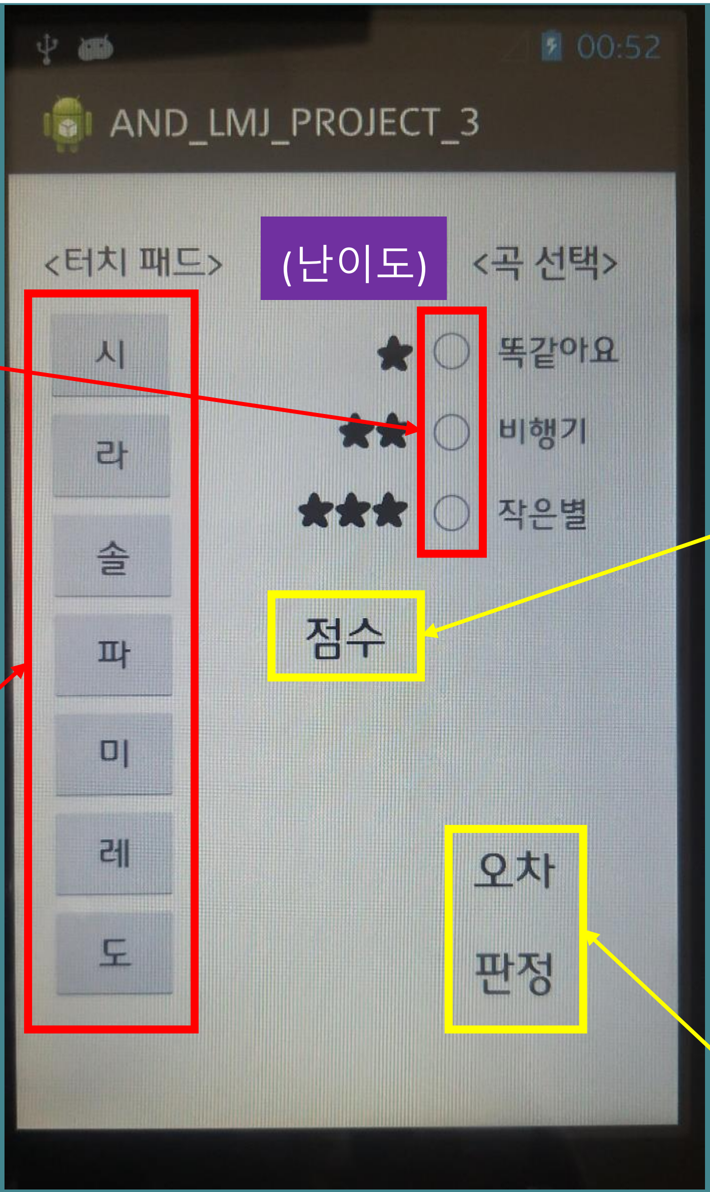
Project Design – Block Diagram (Flow Chart)



Project Configuration

곡 선택

노트 입력



00:52

AND_LMJ_PROJECT_3

<터치 패드> (난이도) <곡 선택>

시

라

솔

파

미

레

도

★ ○ 똑같아요

★★ ○ 비행기

★★★ ○ 작은별

점수

오차

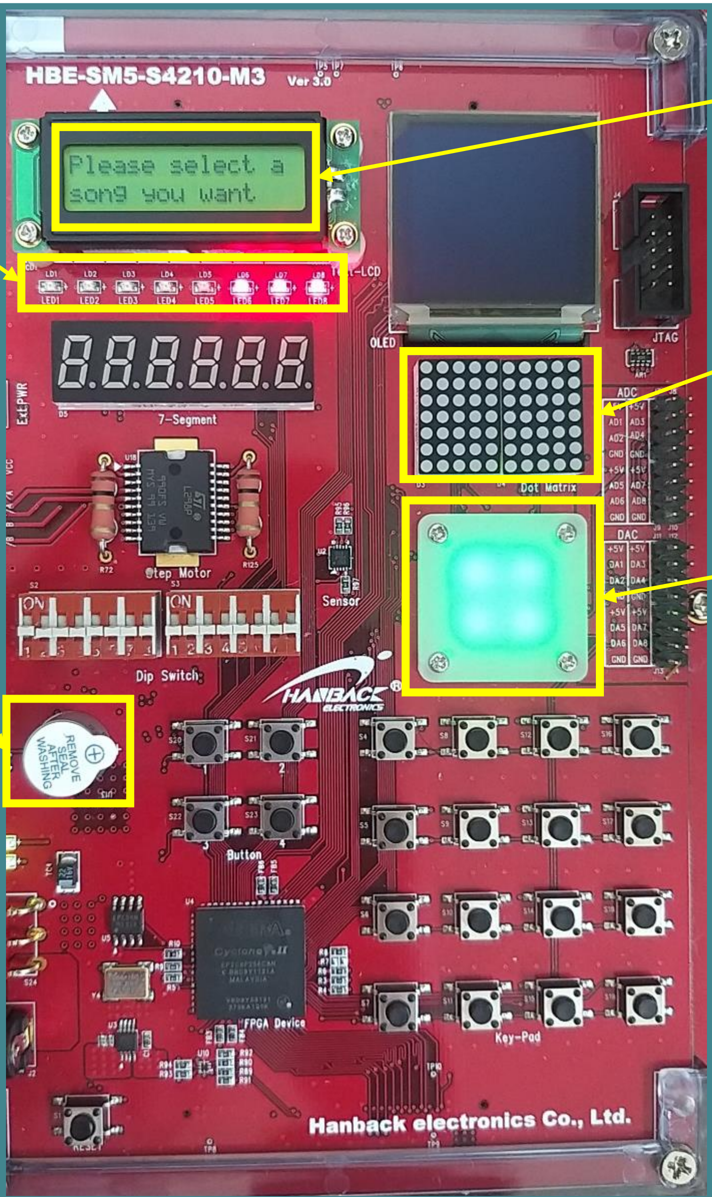
판정

Rating 출력

점수 출력

노트 음 출력

오차, 판정 출력

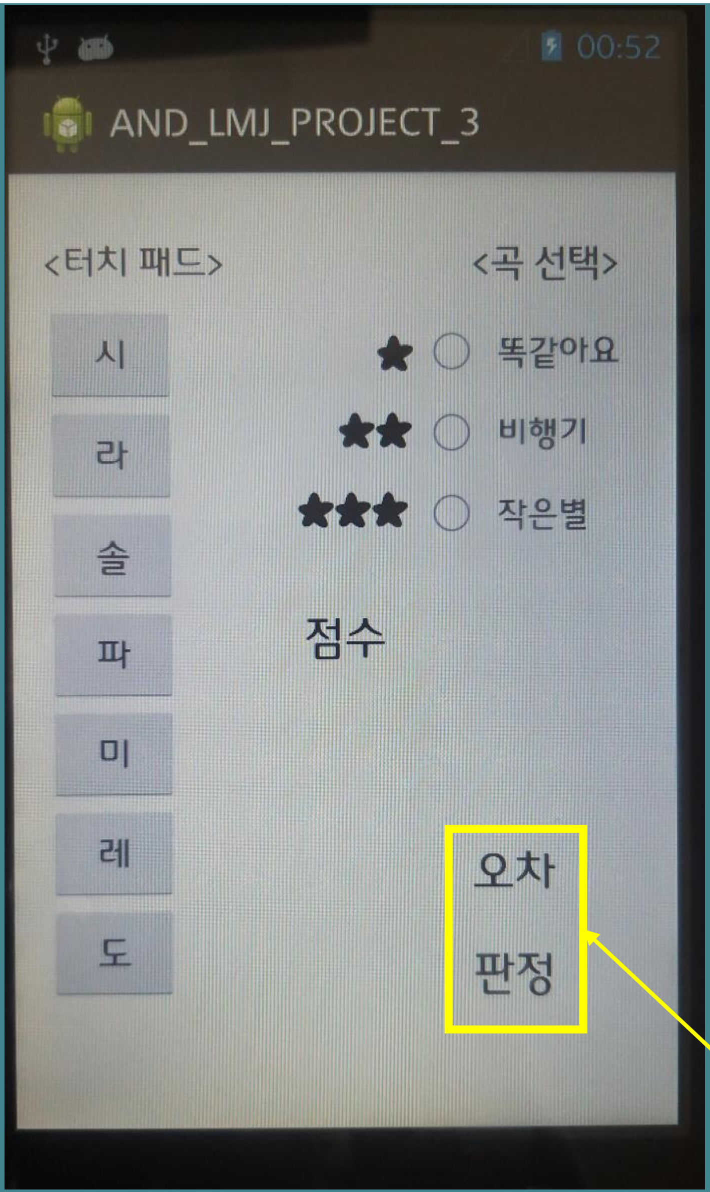


곡 정보 출력
점수 출력

노트 출력

판정 출력

Project Configuration



00:52

AND_LMJ_PROJECT_3

<터치 패드>

<곡 선택>

★ ○ 똑같아요

★★ ○ 비행기

★★★ ○ 작은별

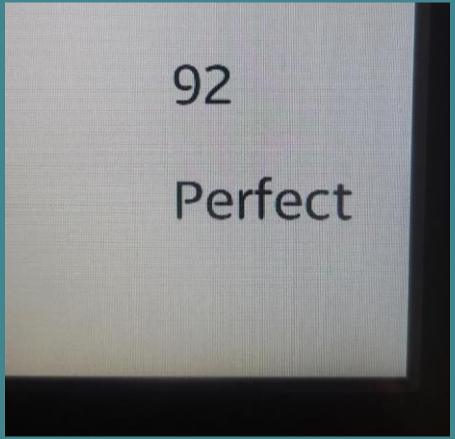
점수

오차

판정

오차, 판정 출력

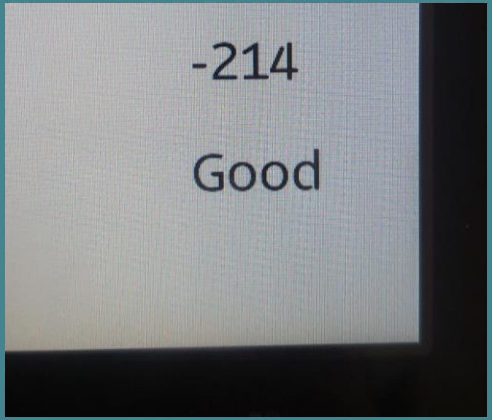
Perfect



92

Perfect

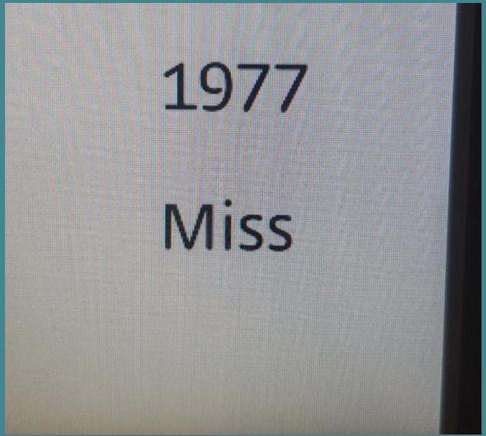
Good



-214

Good

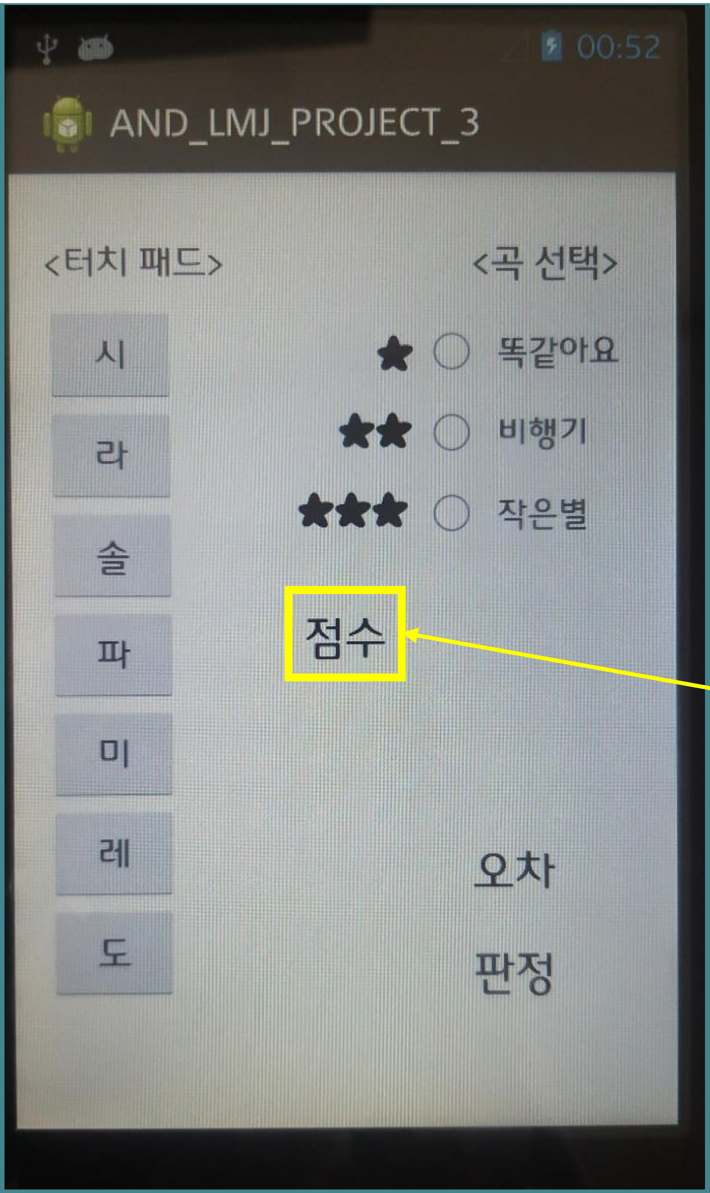
Miss



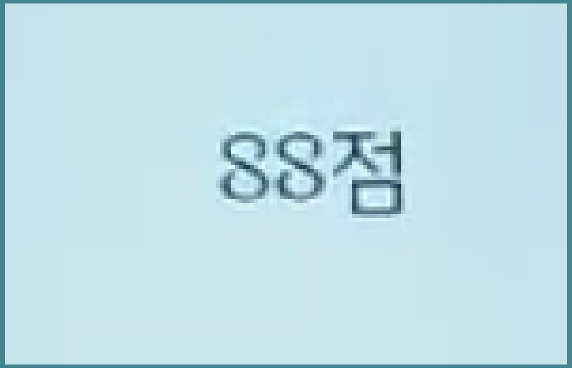
1977

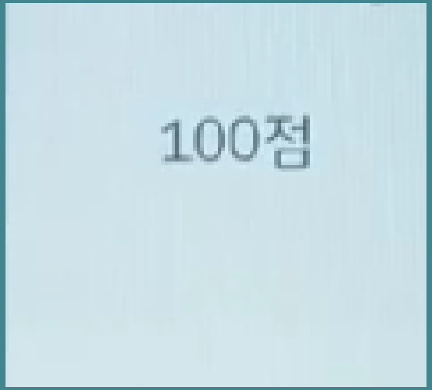
Miss

Project Configuration

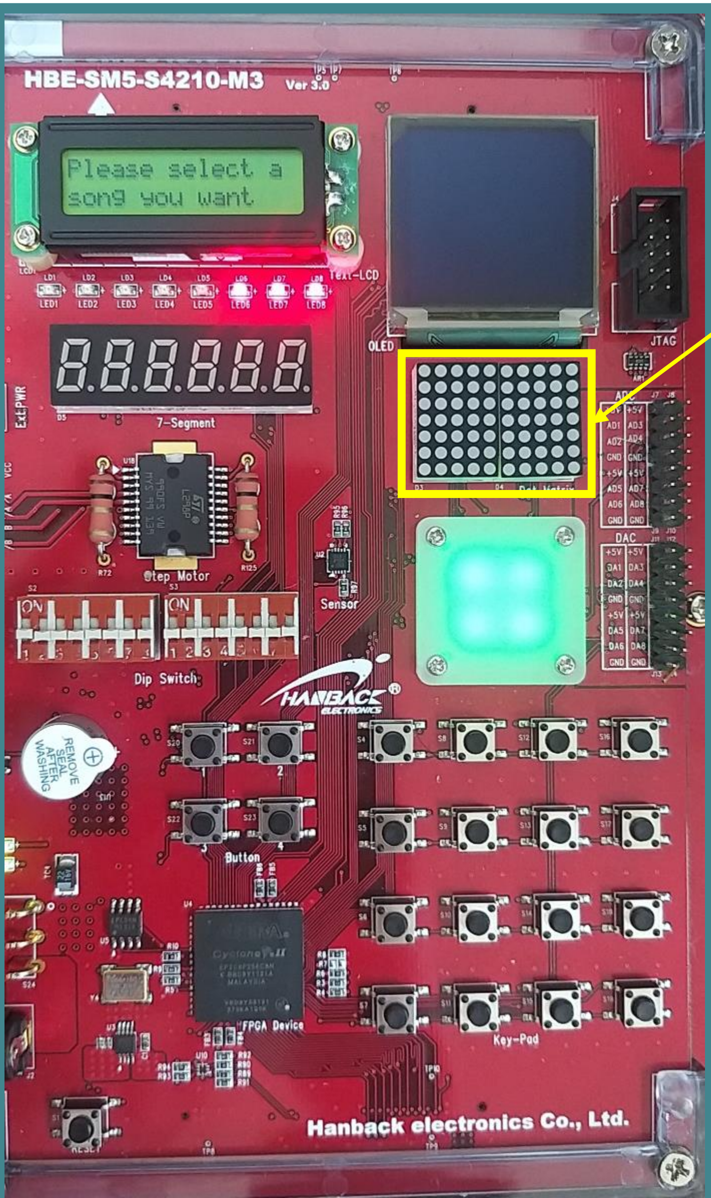


점수 출력

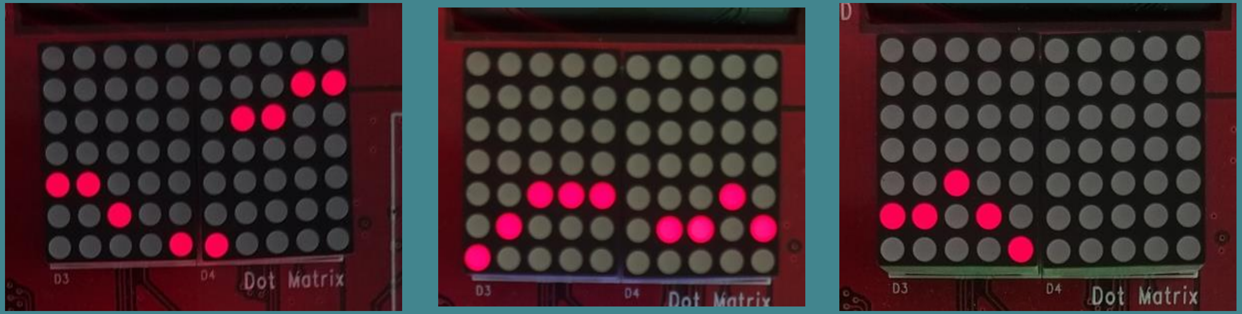




Project Configuration

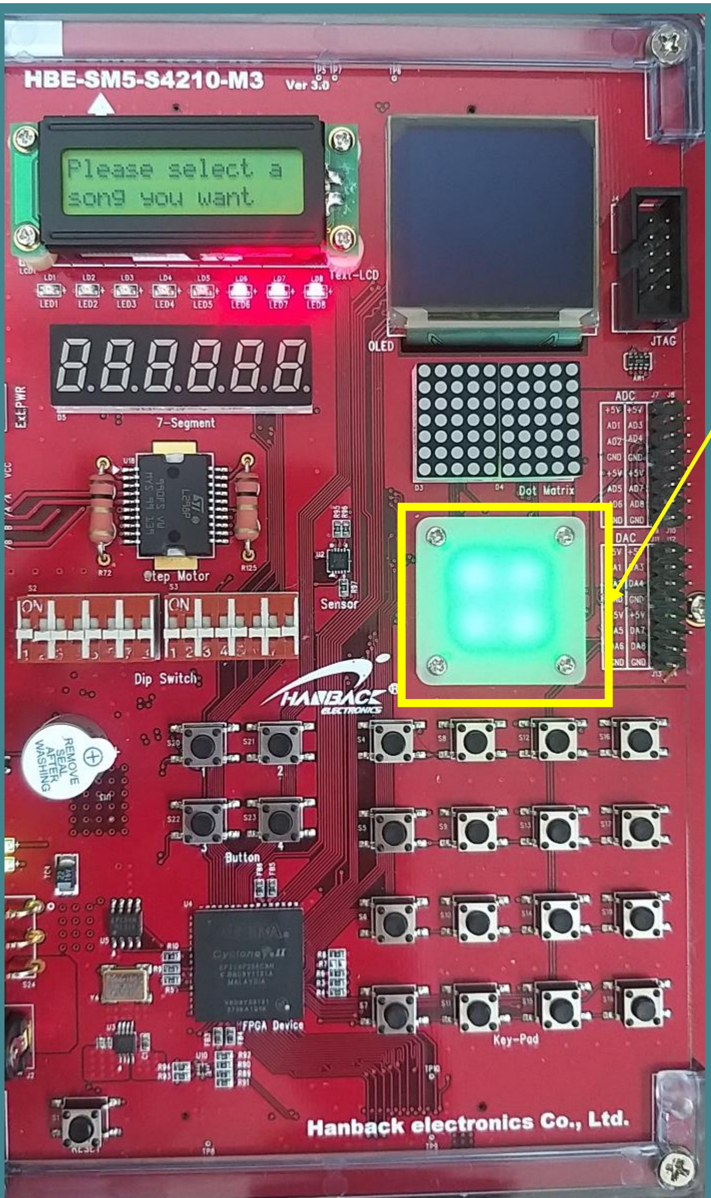


노트 출력



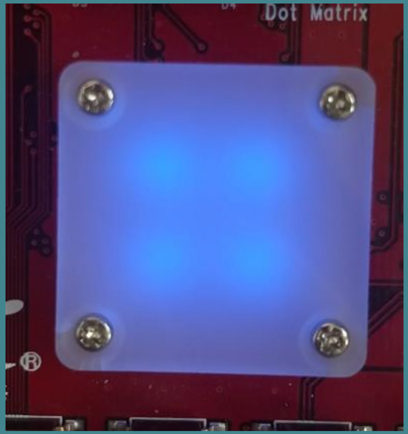
시간이 흐름에 따라
오른쪽에서 왼쪽으로
이동하는 노트들

Project Configuration

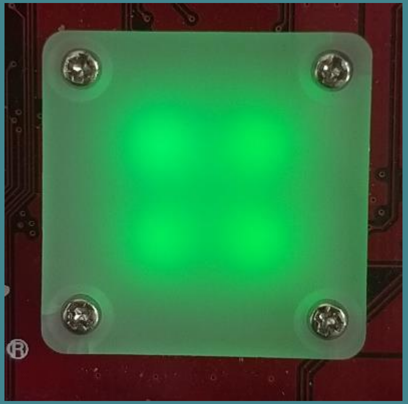


판정 출력

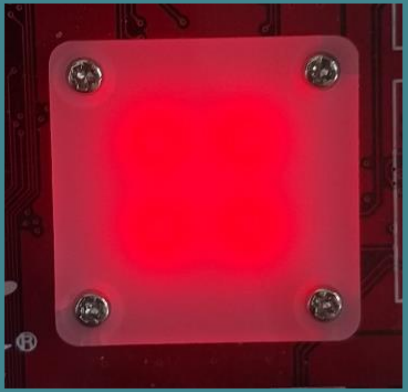
Perfect



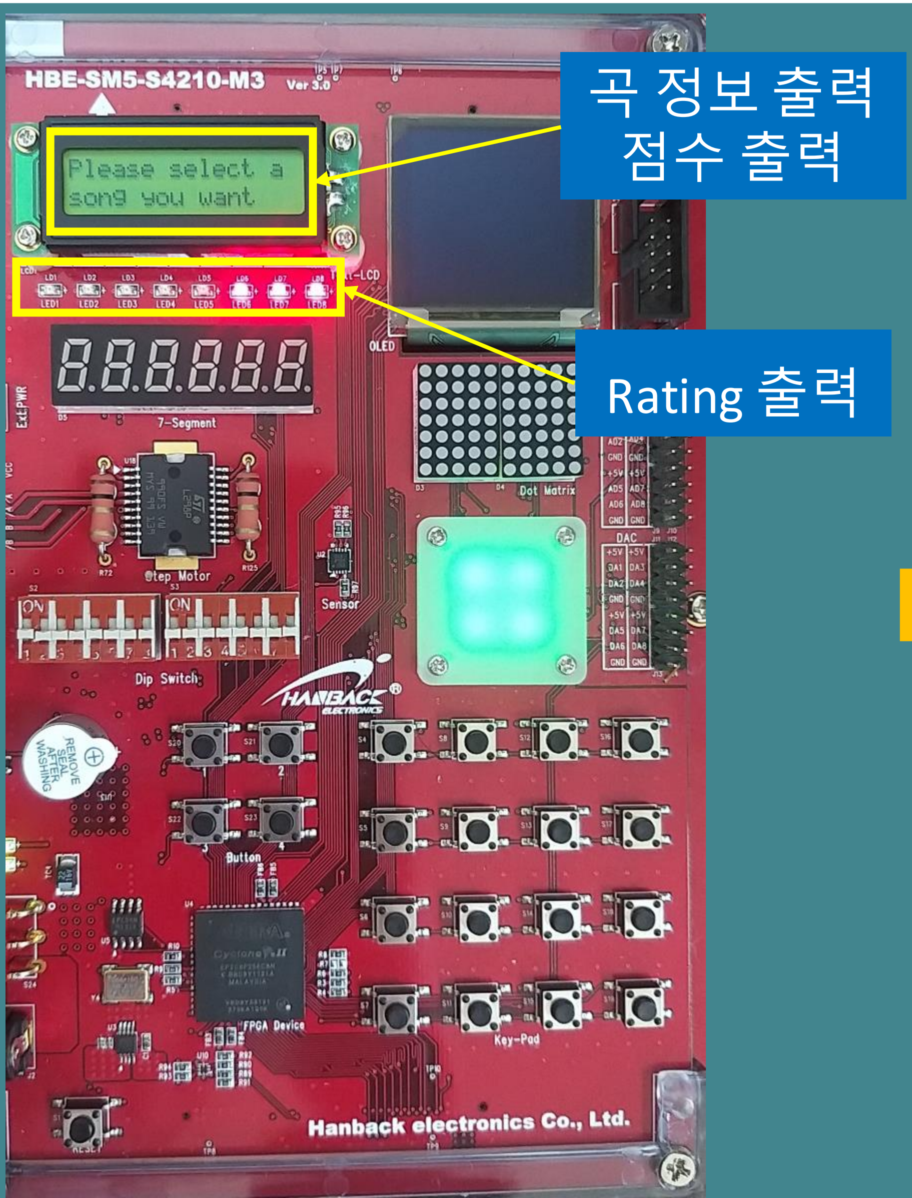
Good



Miss

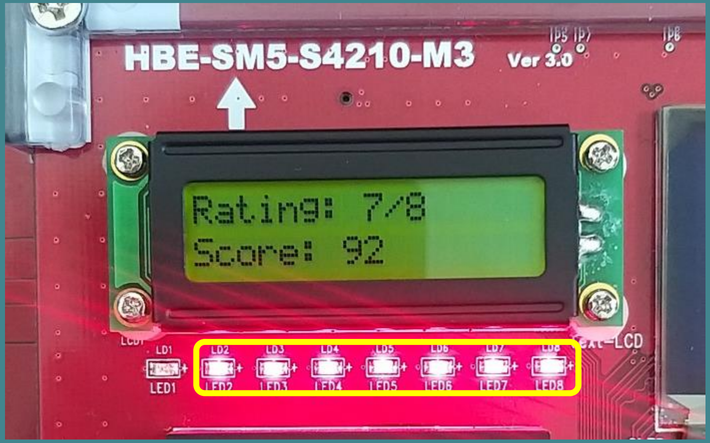


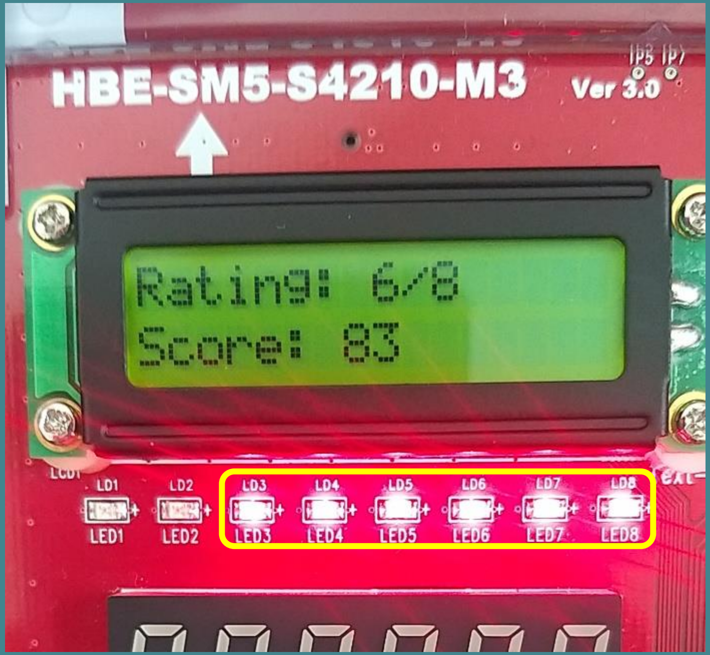
Project Configuration



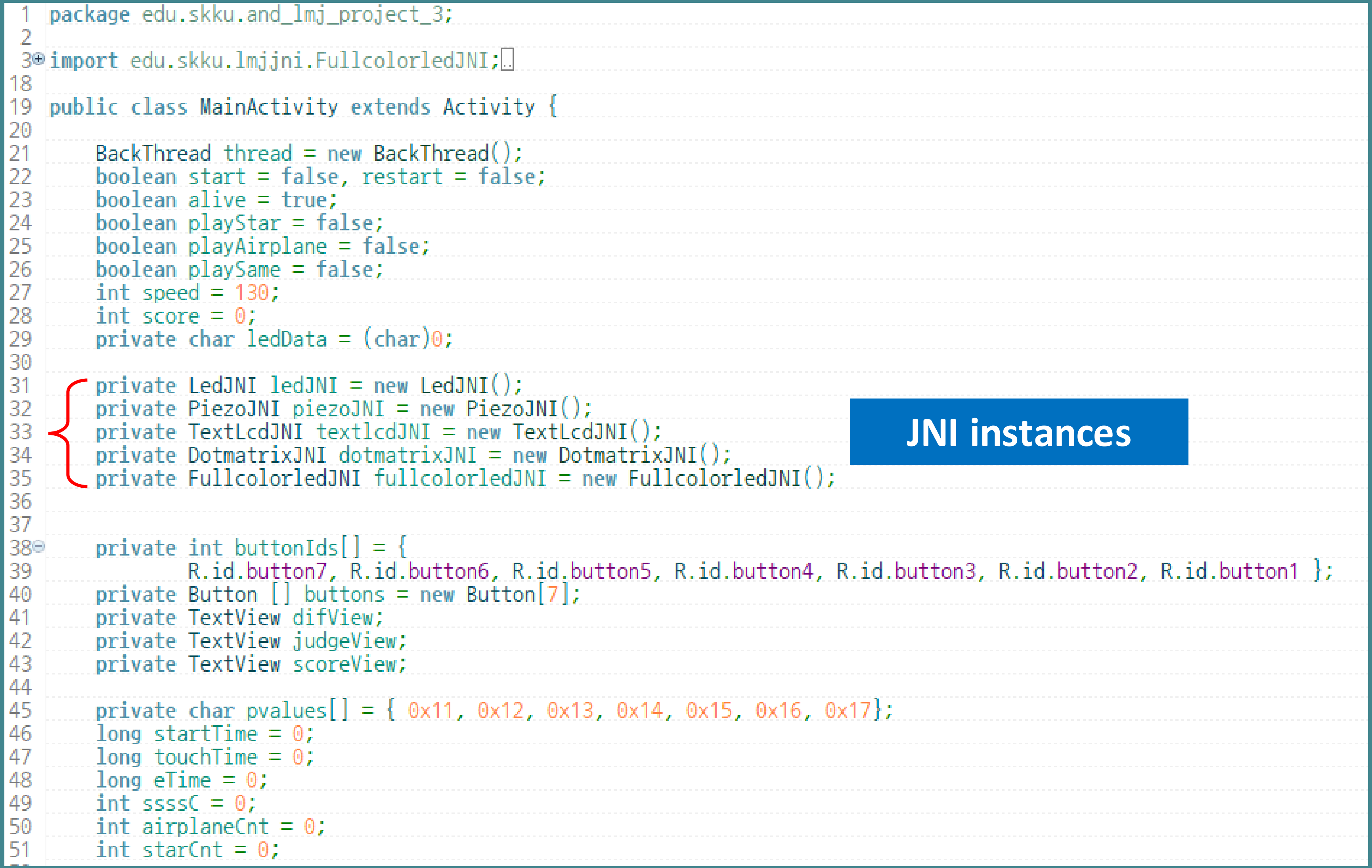
곡 정보 출력
점수 출력

Rating 출력





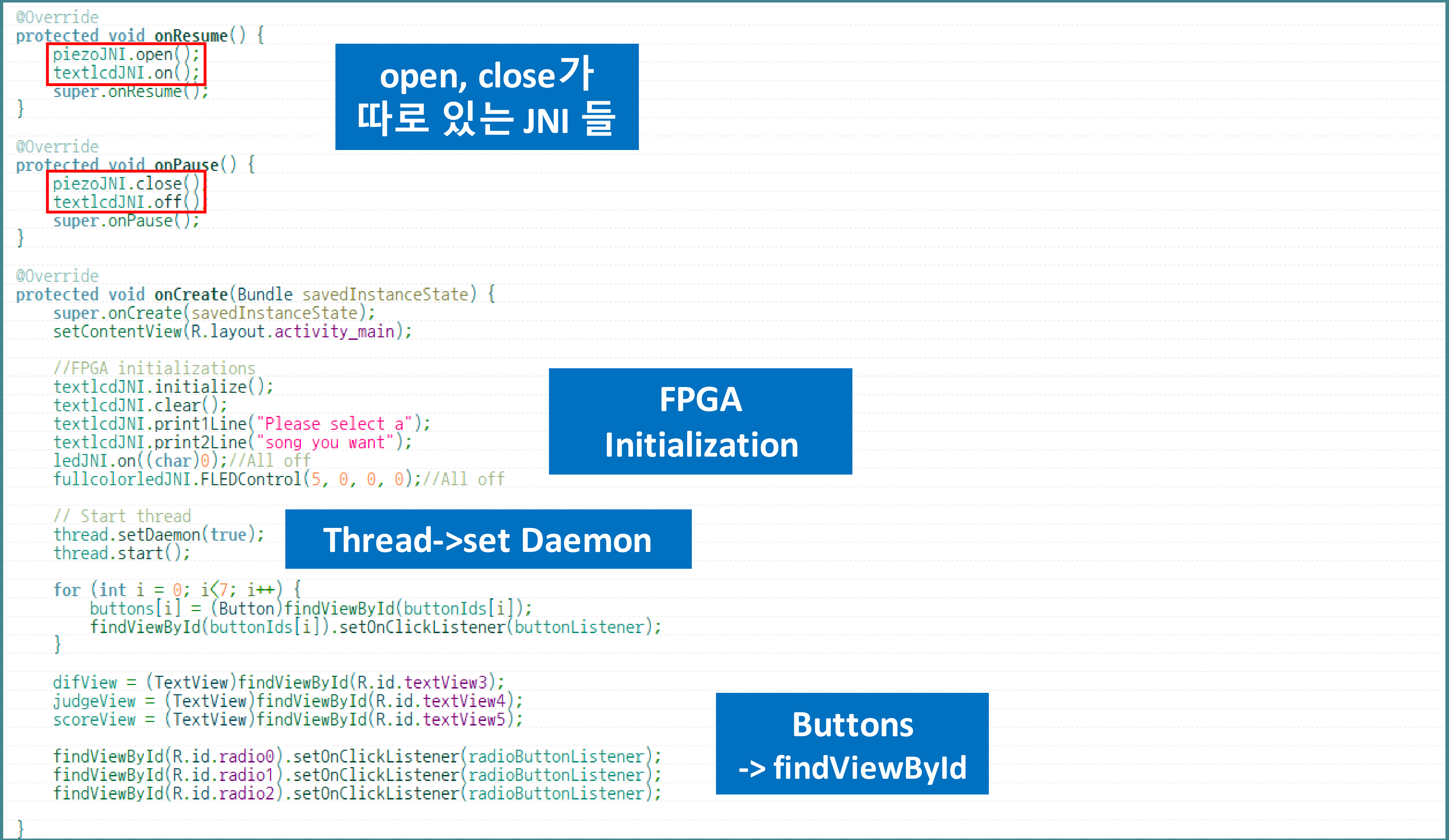
MainActivity.java



```java
package edu.skku.and_lmj_project_3;

import edu.skku.lmjjni.FullcolorledJNI;

public class MainActivity extends Activity {

    BackThread thread = new BackThread();
    boolean start = false, restart = false;
    boolean alive = true;
    boolean playStar = false;
    boolean playAirplane = false;
    boolean playSame = false;
    int speed = 130;
    int score = 0;
    private char ledData = (char)0;

    private LedJNI ledJNI = new LedJNI();
    private PiezoJNI piezoJNI = new PiezoJNI();
    private TextLcdJNI textlcdJNI = new TextLcdJNI();
    private DotmatrixJNI dotmatrixJNI = new DotmatrixJNI();
    private FullcolorledJNI fullcolorledJNI = new FullcolorledJNI();


    private int buttonIds[] = {
            R.id.button7, R.id.button6, R.id.button5, R.id.button4, R.id.button3, R.id.button2, R.id.button1 };
    private Button [] buttons = new Button[7];
    private TextView difView;
    private TextView judgeView;
    private TextView scoreView;

    private char pvalues[] = { 0x11, 0x12, 0x13, 0x14, 0x15, 0x16, 0x17};
    long startTime = 0;
    long touchTime = 0;
    long eTime = 0;
    int ssssC = 0;
    int airplaneCnt = 0;
    int starCnt = 0;
```

JNI instances

MainActivity.java



```java
@Override
protected void onResume() {
    piezoJNI.open();
    textlcdJNI.on();
    super.onResume();
}

@Override
protected void onPause() {
    piezoJNI.close();
    textlcdJNI.off();
    super.onPause();
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    //FPGA initializations
    textlcdJNI.initialize();
    textlcdJNI.clear();
    textlcdJNI.print1Line("Please select a");
    textlcdJNI.print2Line("song you want");
    ledJNI.on((char)0);//All off
    fullcolorledJNI.FLEDControl(5, 0, 0, 0);//All off

    // Start thread
    thread.setDaemon(true);
    thread.start();

    for (int i = 0; i<7; i++) {
        buttons[i] = (Button)findViewById(buttonIds[i]);
        findViewById(buttonIds[i]).setOnClickListener(buttonListener);
    }

    difView = (TextView)findViewById(R.id.textView3);
    judgeView = (TextView)findViewById(R.id.textView4);
    scoreView = (TextView)findViewById(R.id.textView5);

    findViewById(R.id.radio0).setOnClickListener(radioButtonListener);
    findViewById(R.id.radio1).setOnClickListener(radioButtonListener);
    findViewById(R.id.radio2).setOnClickListener(radioButtonListener);
}
```

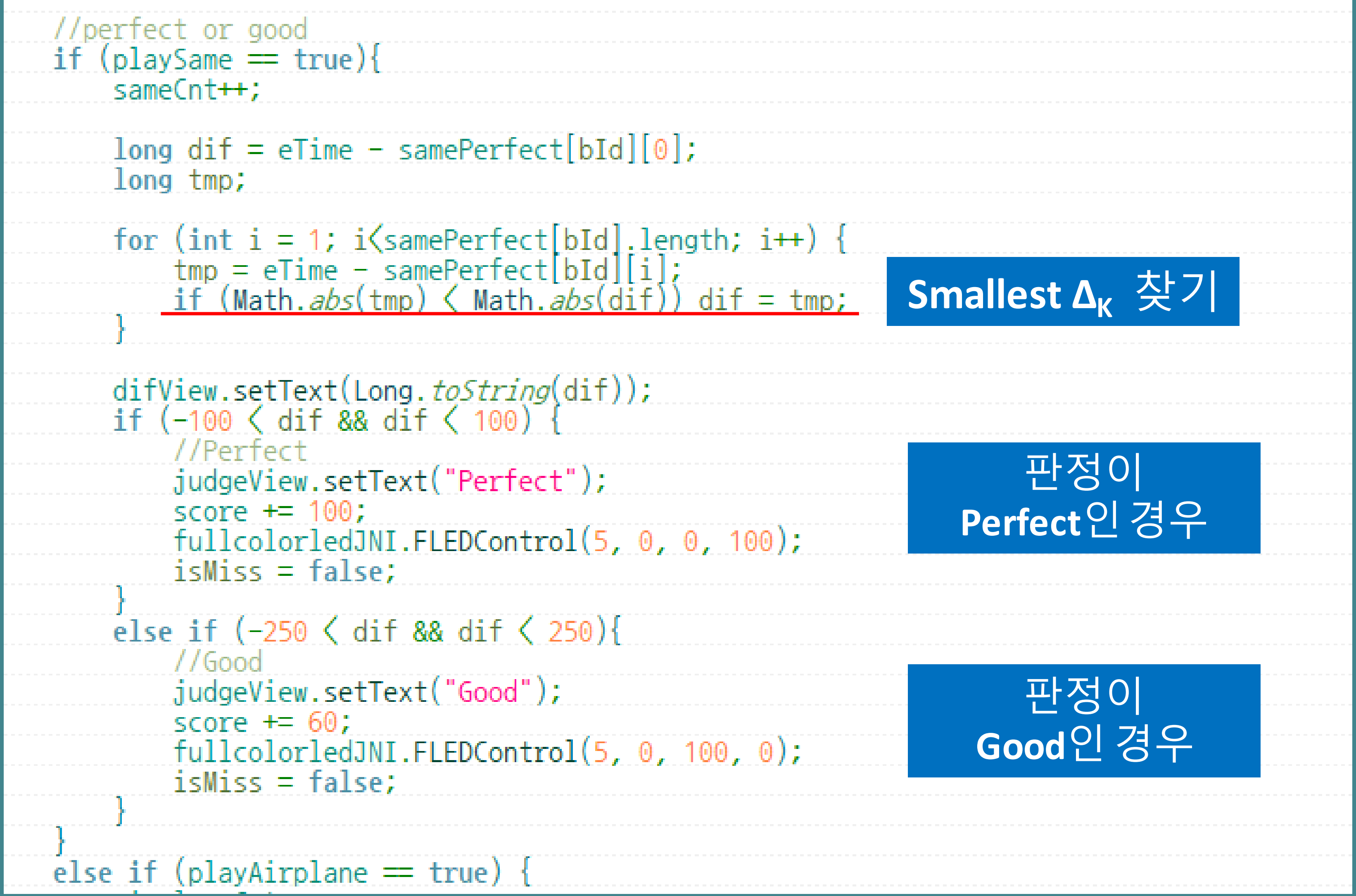
open, close가
따로 있는 JNI 들

FPGA
Initialization

Thread->set Daemon

Buttons
-> findViewById

MainActivity.java



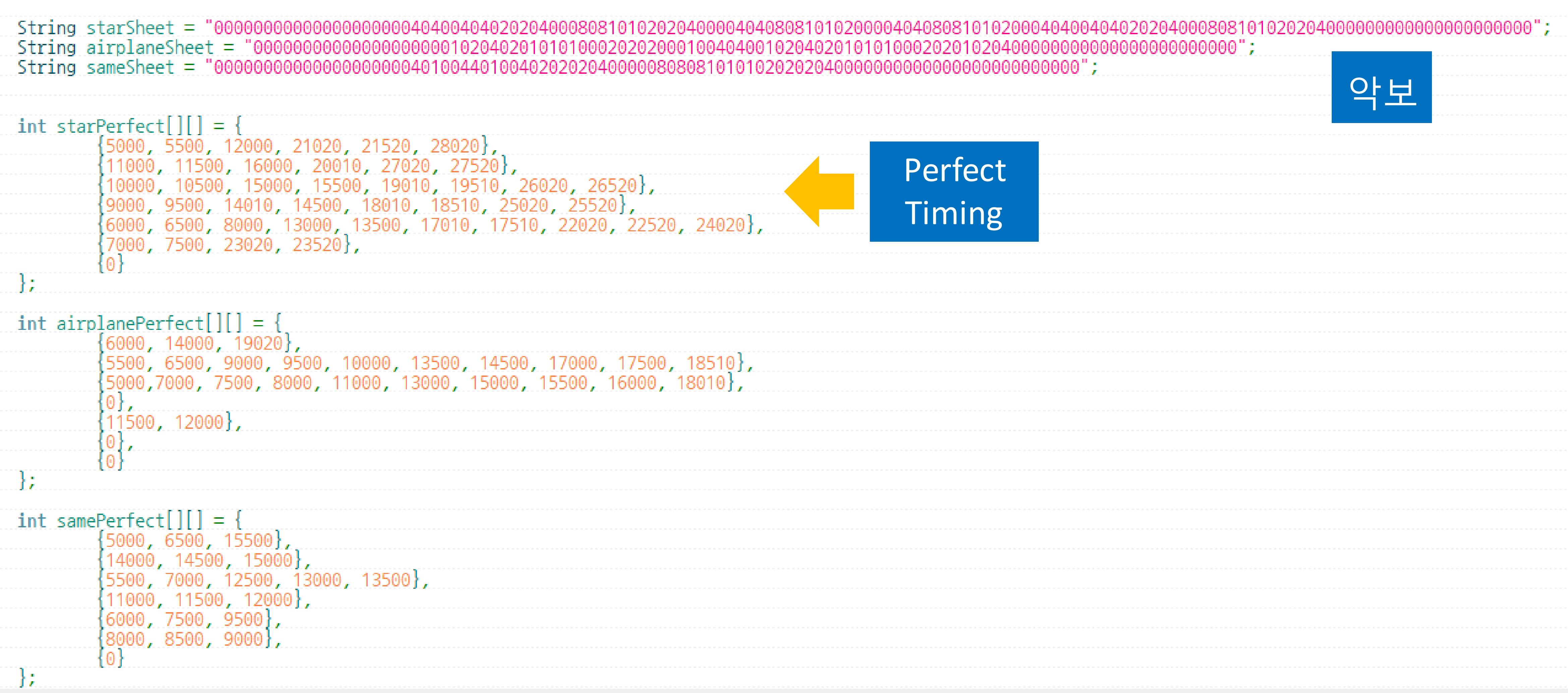
```
//perfect or good
if (playSame == true){
    sameCnt++;

    long dif = eTime - samePerfect[bId][0];
    long tmp;

    for (int i = 1; i<samePerfect[bId].length; i++) {
        tmp = eTime - samePerfect[bId][i];
        if (Math.abs(tmp) < Math.abs(dif)) dif = tmp;
    }

    difView.setText(Long.toString(dif));
    if (-100 < dif && dif < 100) {
        //Perfect
        judgeView.setText("Perfect");
        score += 100;
        fullcolorledJNI.FLEDControl(5, 0, 0, 100);
        isMiss = false;
    }
    else if (-250 < dif && dif < 250){
        //Good
        judgeView.setText("Good");
        score += 60;
        fullcolorledJNI.FLEDControl(5, 0, 100, 0);
        isMiss = false;
    }
}
else if (playAirplane == true) {
```

Smallest Δ_K 찾기

판정이
Perfect인 경우

판정이
Good인 경우

MainActivity.java



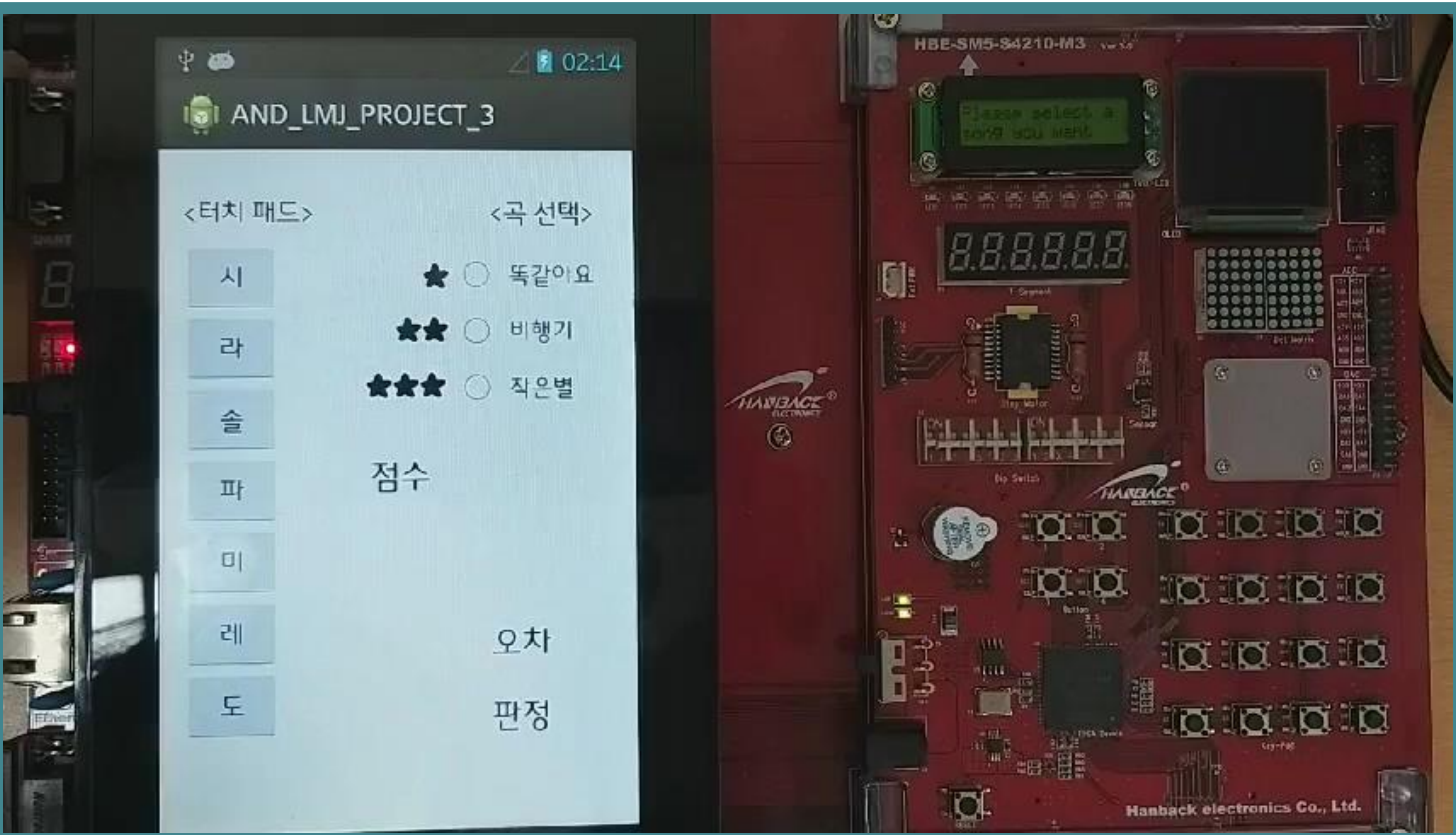
```java
String starSheet = "00000000000000000000404004040202040008081010202040000404080810102000040408081010200040400404020204000808101020204000000000000000000000";
String airplaneSheet = "0000000000000000000010204020101010002020200010040400102040201010100020201020400000000000000000000000";
String sameSheet = "0000000000000000000040100440100402020204000008080810101020202040000000000000000000000000";


int starPerfect[][] = {
        {5000, 5500, 12000, 21020, 21520, 28020},
        {11000, 11500, 16000, 20010, 27020, 27520},
        {10000, 10500, 15000, 15500, 19010, 19510, 26020, 26520},
        {9000, 9500, 14010, 14500, 18010, 18510, 25020, 25520},
        {6000, 6500, 8000, 13000, 13500, 17010, 17510, 22020, 22520, 24020},
        {7000, 7500, 23020, 23520},
        {0}
};

int airplanePerfect[][] = {
        {6000, 14000, 19020},
        {5500, 6500, 9000, 9500, 10000, 13500, 14500, 17000, 17500, 18510},
        {5000,7000, 7500, 8000, 11000, 13000, 15000, 15500, 16000, 18010},
        {0},
        {11500, 12000},
        {0},
        {0}
};

int samePerfect[][] = {
        {5000, 6500, 15500},
        {14000, 14500, 15000},
        {5500, 7000, 12500, 13000, 13500},
        {11000, 11500, 12000},
        {6000, 7500, 9500},
        {8000, 8500, 9000},
        {0}
};
```

악보

Perfect Timing

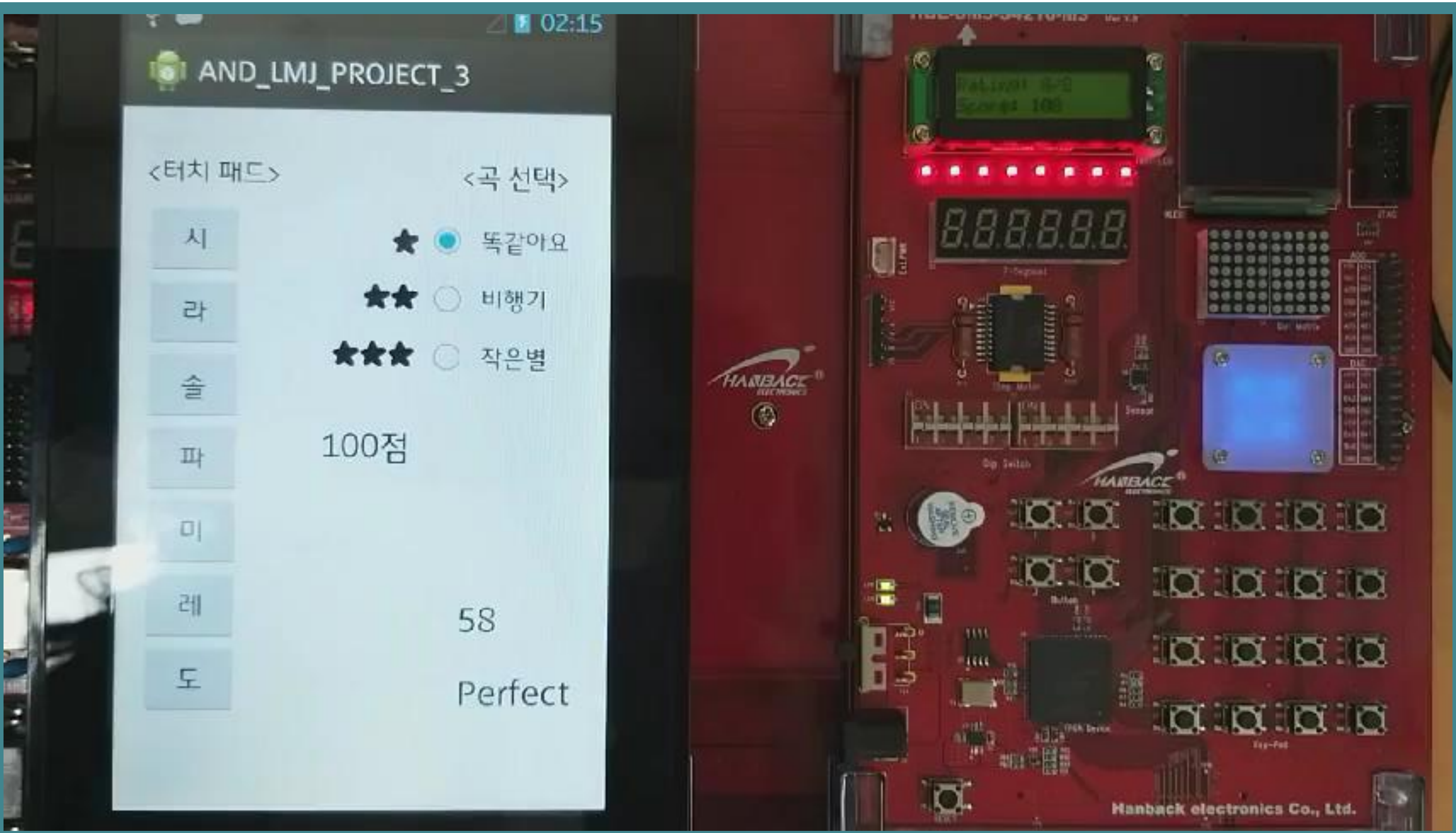
Demonstration Video

All perfect video



Demonstration Video

w/ Good, Miss Notes



To Improve…

- 스레드 간의 정보 주고받기
  - Java -> Runnable
- 판정의 정확도 높이기
- GUI(activity_main.xml)에서 노트 시각화
- 다양한 음역대
- 게임의 난이도 조절 (판정 좁히기)
- 노트의 속도 조절