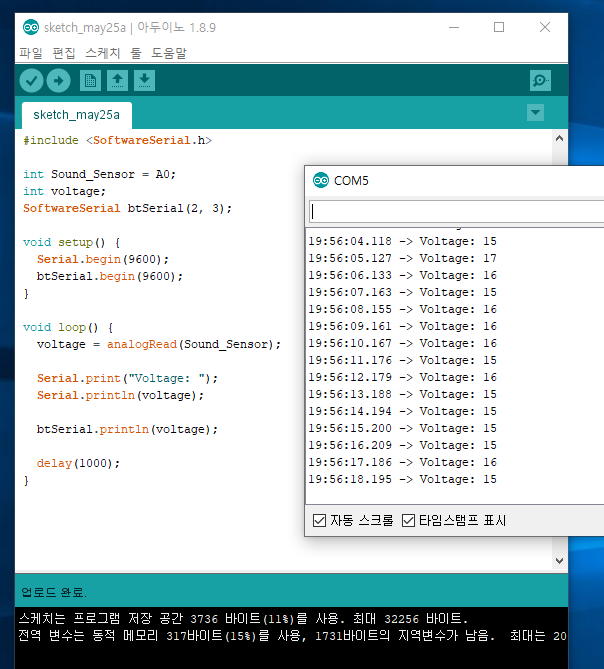
**소스코드**

**> 아두이노**



**> AndroidManifest.xml**

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?> <manifest xmlns:android="http://schemas.android.com/apk/res/android"  package="com.example.smartsoundcontroller">   <uses-permission android:name="android.permission.MODIFY\_AUDIO\_SETTINGS" />  <uses-permission android:name="android.permission.VIBRATE" />  <uses-permission android:name="android.permission.BLUETOOTH" />  <uses-permission android:name="android.permission.BLUETOOTH\_ADMIN" />   <application  android:allowBackup="true"  android:icon="@mipmap/ic\_launcher"  android:label="@string/app\_name"  android:roundIcon="@mipmap/ic\_launcher\_round"  android:supportsRtl="true"  android:theme="@style/AppTheme">  <activity  android:name=".SplashActivity"  android:theme="@style/SplashTheme">  <intent-filter>  <action android:name="android.intent.action.MAIN" />   <category android:name="android.intent.category.LAUNCHER" />  </intent-filter>  </activity>  <activity android:name=".MainActivity" />  <activity android:name=".SettingActivity" />  </application>  </manifest> |

**> MainActivity.java**

|  |
| --- |
| /\*  \* 2019년 1학기 모바일프로그래밍 프로젝트  \* https://github.com/youseokhwan/SmartSoundController  \*  \* SmartSoundController  \* - 아두이노 사운드센서로 주변 소음을 측정하여 미디어 볼륨을 자동으로 조절해주는 어플  \*/  package com.example.smartsoundcontroller;  import android.app.Activity; import android.content.Context; import android.content.Intent; import android.content.SharedPreferences; import android.media.AudioManager; import android.os.Bundle; import android.os.Handler; import android.os.Vibrator; import android.view.KeyEvent; import android.view.View; import android.widget.Button; import android.widget.SeekBar; import android.widget.TextView; import android.widget.Toast;  import android.bluetooth.BluetoothAdapter; import app.akexorcist.bluetotohspp.library.BluetoothSPP; import app.akexorcist.bluetotohspp.library.BluetoothState; import app.akexorcist.bluetotohspp.library.DeviceList;  */\*\*  \* SmartSoundController  \*  \* 아두이노 보드와 사운드 센서(SZH-EK033)와 블루투스 모듈(HC-06)을 이용하며,  \* 센서에서 측정한 전압 값(아마도?)을 블루투스로 기기에 전달합니다. (아두이노 스케치 사용)  \* 전달받은 값과 사용자가 설정한 값으로 수치를 계산하여 자동으로 미디어 볼륨을 조절합니다.  \*/* public class MainActivity extends Activity implements View.OnClickListener {   public SharedPreferences appData;  private Thread soundManager;  private BluetoothSPP bluetoothSPP;  public Vibrator vibrator;   SeekBar currentVolumeSeekBar;  TextView currentDecibelTextView;  Button settingButton, exitButton;  AudioManager audioManager;   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_main*);   appData = getSharedPreferences("appData", *MODE\_PRIVATE*);  vibrator = (Vibrator)getSystemService(Context.*VIBRATOR\_SERVICE*);  bluetoothSPP = new BluetoothSPP(this);  soundManager = new Thread(new Runnable() {  final Handler threadHandler = new Handler();   */\*\*  \* 변경할 볼륨 = 설정한 최소 볼륨 + (주변 소음 - 설정한 기준 소음) / (설정한 단위)  \*/* @Override  public void run() {  while(true) {  try {  soundManager.*sleep*(1000);   threadHandler.post(new Runnable() {  @Override  public void run() {  int distancedB = Integer.*parseInt*(currentDecibelTextView.getText().toString())  - appData.getInt("STD\_DB", 20);   if(distancedB < 0)  distancedB = 0;   int divideUnit = distancedB / appData.getInt("STD\_UNIT", 5);   // 변경할 값이랑 현재 값이랑 다르면 실행  if(currentVolumeSeekBar.getProgress() != appData.getInt("STD\_VOLUME", 3) + divideUnit) {  currentVolumeSeekBar.setProgress(appData.getInt("STD\_VOLUME", 3) + divideUnit);   if(appData.getBoolean("VIB\_CHECK", false)) {  vibrator.vibrate(500);  }  }  }  });  }  catch (InterruptedException e) {  e.printStackTrace();  }  }  }  });  soundManager.start();   currentVolumeSeekBar = (SeekBar)findViewById(R.id.*currentVolumeSeekBar*);  currentDecibelTextView = (TextView)findViewById(R.id.*currentDecibelTextView*);  settingButton = (Button)findViewById(R.id.*settingButton*);  exitButton = (Button)findViewById(R.id.*exitButton*);   // 블루투스 사용 불가할 때 메시지 출력  if(!bluetoothSPP.isBluetoothAvailable()) {  Toast.*makeText*(getApplicationContext(), "블루투스 사용 불가", Toast.*LENGTH\_SHORT*).show();  }   // 블루투스 모듈 선택 화면  Intent intent = new Intent(getApplicationContext(), DeviceList.class);  startActivityForResult(intent, BluetoothState.*REQUEST\_CONNECT\_DEVICE*);   // 데이터 수신  bluetoothSPP.setOnDataReceivedListener(new BluetoothSPP.OnDataReceivedListener() { //데이터 수신  public void onDataReceived(byte[] data, String message) {  // Toast.makeText(MainActivity.this, message, Toast.LENGTH\_SHORT).show();  int temp = (int)(20 \* Math.*log10*(Integer.*parseInt*(message)));   currentDecibelTextView.setText(Integer.*toString*(temp));  }  });   // 설정, 종료 버튼 리스너  settingButton.setOnClickListener((View.OnClickListener)this);  exitButton.setOnClickListener((View.OnClickListener)this);   // 기기의 현재 미디어 볼륨 정보를 가져와서 volumeBar에 적용  audioManager = (AudioManager)getSystemService(Context.*AUDIO\_SERVICE*);  currentDecibelTextView.setText("0");  currentVolumeSeekBar.setMax(audioManager.getStreamMaxVolume(AudioManager.*STREAM\_MUSIC*));  currentVolumeSeekBar.setProgress(audioManager.getStreamVolume(AudioManager.*STREAM\_MUSIC*));   // volumeBar의 progress값 변경 이벤트  currentVolumeSeekBar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {  public void onStartTrackingTouch(SeekBar seekBar) {  // empty  }   public void onStopTrackingTouch(SeekBar seekBar) {  // empty  }   // Progress값 변경 시  public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {  audioManager.setStreamVolume(AudioManager.*STREAM\_MUSIC*, progress, 0);  }  });   }   @Override  public void onStart() {  super.onStart();   // 블루투스 disabled면 실행  if(!bluetoothSPP.isBluetoothEnabled()) {  Intent intent = new Intent(BluetoothAdapter.*ACTION\_REQUEST\_ENABLE*);  startActivityForResult(intent, BluetoothState.*REQUEST\_ENABLE\_BT*);  }  else {  if(!bluetoothSPP.isServiceAvailable()) {  bluetoothSPP.setupService();  bluetoothSPP.startService(BluetoothState.*DEVICE\_OTHER*);  bluetoothSPP.send("Text", true);  }  }  }   public void onActivityResult(int requestCode, int resultCode, Intent data) {  if(requestCode == BluetoothState.*REQUEST\_CONNECT\_DEVICE*) {  if(resultCode == Activity.*RESULT\_OK*) {  bluetoothSPP.connect(data);  }  }  else if(requestCode == BluetoothState.*REQUEST\_ENABLE\_BT*) {  if(resultCode == Activity.*RESULT\_OK*) {  bluetoothSPP.setupService();  bluetoothSPP.startService(BluetoothState.*DEVICE\_OTHER*);  bluetoothSPP.send("Text", true);  }  }  }   @Override  public void onDestroy() {  super.onDestroy();   bluetoothSPP.stopService(); // 앱 종료될 때 블루투스 중지  }    // 버튼 클릭 이벤트  @Override  public void onClick(View v) {  // "설정" 버튼 클릭  if(v.getId() == R.id.*settingButton*) {  Intent intent = new Intent(this, SettingActivity.class);  startActivity(intent);  }  // "종료" 버튼 클릭  else if(v.getId() == R.id.*exitButton*) {  finish();  }  else {  // empty  }  }   // 기기 볼륨 버튼 누를 때 volumeBar에 반영  public boolean onKeyDown(int keycode, KeyEvent event) {  switch(keycode) {  case KeyEvent.*KEYCODE\_VOLUME\_DOWN*:  if(currentVolumeSeekBar.getProgress() > 0)  currentVolumeSeekBar.setProgress(currentVolumeSeekBar.getProgress() - 1);  break;  case KeyEvent.*KEYCODE\_VOLUME\_UP*:  if(currentVolumeSeekBar.getProgress() < currentVolumeSeekBar.getMax())  currentVolumeSeekBar.setProgress(currentVolumeSeekBar.getProgress() + 1);  break;  }   return false; // true면 볼륨 조절 시스템 창 뜨지 않게 설정  }  } |

**> SettingActivity.java**

|  |
| --- |
| package com.example.smartsoundcontroller;  import android.app.Activity; import android.content.Context; import android.content.SharedPreferences; import android.media.AudioManager; import android.os.Bundle; import android.os.Vibrator; import android.view.View; import android.widget.Button; import android.widget.CheckBox; import android.widget.EditText; import android.widget.SeekBar;  public class SettingActivity extends Activity implements View.OnClickListener {   public SharedPreferences appData;   EditText standardDecibelEditText;  SeekBar standardVolumeSeekBar;  EditText standardUnitEditText;  CheckBox vibratorCheckBox;  Button applyButton;   AudioManager audioManager;  Vibrator vibrator;   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_setting*);   standardDecibelEditText = (EditText)findViewById(R.id.*standardDecibelEditText*);  standardVolumeSeekBar = (SeekBar)findViewById(R.id.*standardVolumeSeekBar*);  standardUnitEditText = (EditText)findViewById(R.id.*standardUnitEditText*);  vibratorCheckBox = (CheckBox)findViewById(R.id.*vibratorCheckBox*);  applyButton = (Button)findViewById(R.id.*applyButton*);   // 기기의 볼륨 상태 가져오고 standardVolumeSeekBar의 Max값 지정  audioManager = (AudioManager)getSystemService(Context.*AUDIO\_SERVICE*);  standardVolumeSeekBar.setMax(audioManager.getStreamMaxVolume(AudioManager.*STREAM\_MUSIC*));   // 진동 관련  vibrator = (Vibrator)getSystemService(Context.*VIBRATOR\_SERVICE*);   // 이벤트 리스너 등록  applyButton.setOnClickListener((View.OnClickListener)this);  vibratorCheckBox.setOnClickListener((View.OnClickListener)this);   // 저장된 설정 값 불러오기  appData = getSharedPreferences("appData", *MODE\_PRIVATE*);  standardDecibelEditText.setText(Integer.*toString*(appData.getInt("STD\_DB", 20)));  standardVolumeSeekBar.setProgress(appData.getInt("STD\_VOLUME", 3));  standardUnitEditText.setText(Integer.*toString*(appData.getInt("STD\_UNIT", 5)));  vibratorCheckBox.setChecked(appData.getBoolean("VIB\_CHECK", false));  }   @Override  public void onClick(View v) {  if(v.getId() == R.id.*applyButton*) {  // 적용 버튼 누르면 기준 dB, 기준 볼륨, 기준 단위, 진동 여부 4가지 값 저장하고 해당 레이아웃 종료  SharedPreferences.Editor editor = appData.edit();  editor.putInt("STD\_DB", Integer.*parseInt*(standardDecibelEditText.getText().toString().trim()));  editor.putInt("STD\_VOLUME", standardVolumeSeekBar.getProgress());  editor.putInt("STD\_UNIT", Integer.*parseInt*(standardUnitEditText.getText().toString().trim()));  editor.putBoolean("VIB\_CHECK", vibratorCheckBox.isChecked());  editor.apply();   finish();  }  // 진동 체크박스 클릭  else if(v.getId() == R.id.*vibratorCheckBox*) {  // 체크될 때 잠깐 진동  if(((CheckBox)v).isChecked()) {  vibrator.vibrate(200);  }  }  else {  // empty  }  } } |

**> SplashActivity.java**

|  |
| --- |
| package com.example.smartsoundcontroller;  import android.app.Activity; import android.content.Intent; import android.os.Bundle;  public class SplashActivity extends Activity {   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);   try {  Thread.*sleep*(1500);  Intent intent = new Intent(this, MainActivity.class);  startActivity(intent);  finish();  } catch (InterruptedException e) {  e.printStackTrace();  }  } } |

**> activity\_main.xml**

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?> <android.support.constraint.ConstraintLayout  xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  tools:context=".MainActivity">   <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_marginTop="100dp"  android:layout\_marginBottom="50dp"  android:layout\_marginLeft="50dp"  android:layout\_marginRight="50dp"  android:orientation="vertical">   <!-- Title -->  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:gravity="top"  android:layout\_weight="3"  android:orientation="vertical">   <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal">   <TextView  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:fontFamily="@font/tmontium"  android:text="S"  android:textSize="45dp"  android:textColor="#000000" />   <TextView  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:gravity="bottom"  android:fontFamily="@font/tmontium"  android:text="MART"  android:textSize="35dp"  android:textColor="#000000" />   </LinearLayout>   <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal">   <TextView  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:fontFamily="@font/tmontium"  android:text="S"  android:textSize="45dp"  android:textColor="#000000" />   <TextView  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:gravity="bottom"  android:fontFamily="@font/tmontium"  android:text="OUND "  android:textSize="35dp"  android:textColor="#000000" />   </LinearLayout>   <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal">   <TextView  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:fontFamily="@font/tmontium"  android:text="C"  android:textSize="45dp"  android:textColor="#000000" />   <TextView  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:gravity="bottom"  android:fontFamily="@font/tmontium"  android:text="ONTROLLER"  android:textSize="35dp"  android:textColor="#000000" />   </LinearLayout>   </LinearLayout>   <!-- 현재 미디어 볼륨 레이블과 미디어 볼륨 bar -->  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:orientation="horizontal">   <TextView  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:gravity="center\_vertical"  android:fontFamily="@font/tmonmonsori"  android:textSize="20dp"  android:text="현재\n미디어 볼륨"  android:textColor="#000000" />   <SeekBar  android:id="@+id/currentVolumeSeekBar"  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:layout\_weight="4"  android:gravity="center"  android:max="100"  android:progress="30" />   </LinearLayout>   <!-- 실시간 소음 및 dB값 -->  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:orientation="horizontal">   <TextView  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:gravity="bottom"  android:fontFamily="@font/tmonmonsori"  android:textSize="20dp"  android:text="실시간 소음"  android:textColor="#000000" />   <TextView  android:id="@+id/currentDecibelTextView"  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:layout\_weight="2"  android:gravity="center|right|bottom"  android:fontFamily="@font/tmonmonsori"  android:textSize="20dp"  android:text="75"  android:textColor="#000000" />   <TextView  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:layout\_weight="2"  android:gravity="center|bottom"  android:fontFamily="@font/tmonmonsori"  android:textSize="20dp"  android:text="dB"  android:textColor="#000000" />   </LinearLayout>   <!-- 설정 버튼 -->  <Button  android:id="@+id/settingButton"  android:layout\_marginTop="45dp"  android:layout\_marginBottom="10dp"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:background="@drawable/btn\_bg"  android:fontFamily="@font/tmonmonsori"  android:textSize="20dp"  android:text="설 정"/>   <!-- 종료 버튼 -->  <Button  android:id="@+id/exitButton"  android:layout\_marginTop="0dp"  android:layout\_marginBottom="45dp"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:background="@drawable/btn\_bg"  android:fontFamily="@font/tmonmonsori"  android:textSize="20dp"  android:text="종 료"/>   </LinearLayout>  </android.support.constraint.ConstraintLayout> |

**> activity\_setting.xml**

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?> <android.support.constraint.ConstraintLayout  xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent">   <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_marginTop="70dp"  android:layout\_marginBottom="50dp"  android:layout\_marginLeft="50dp"  android:layout\_marginRight="50dp"  android:orientation="vertical">   <!-- Title -->  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginBottom="50dp"  android:layout\_weight="6"  android:orientation="horizontal">   <TextView  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:gravity="bottom"  android:fontFamily="@font/tmontium"  android:text="S"  android:textSize="45dp"  android:textColor="#000000" />   <TextView  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:gravity="bottom"  android:fontFamily="@font/tmontium"  android:text="ETTING"  android:textSize="35dp"  android:textColor="#000000" />   </LinearLayout>   <!-- 기준 dB -->  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:orientation="horizontal">   <TextView  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:layout\_marginRight="35dp"  android:gravity="center"  android:fontFamily="@font/tmonmonsori"  android:textSize="20dp"  android:text="기준 dB"  android:textColor="#000000" />   <EditText  android:id="@+id/standardDecibelEditText"  android:layout\_width="50dp"  android:layout\_height="match\_parent"  android:fontFamily="@font/tmonmonsori"  android:gravity="center"  android:text="75"  android:textColor="#000000"  android:textSize="20dp" />   </LinearLayout>   <!-- 기준 볼륨 -->  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:orientation="horizontal">   <TextView  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:gravity="center|left"  android:fontFamily="@font/tmonmonsori"  android:textSize="20dp"  android:text="내 최소\n미디어 볼륨"  android:textColor="#000000" />   <SeekBar  android:id="@+id/standardVolumeSeekBar"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:gravity="center"  android:max="100"  android:progress="30" />   </LinearLayout>   <!-- 설명 -->  <TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="5dp"  android:layout\_marginBottom="15dp"  android:fontFamily="@font/tmonmonsori"  android:textSize="13dp"  android:text="가장 조용한 곳에서의 dB과 볼륨을 설정해주세요  \n내 최소 볼륨보다 낮아지지는 않습니다" />   <!-- 단위 dB 설정 -->  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginBottom="10dp"  android:layout\_weight="1"  android:orientation="horizontal">   <EditText  android:id="@+id/standardUnitEditText"  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:gravity="center"  android:fontFamily="@font/tmonmonsori"  android:textSize="20dp"  android:text="5"  android:textColor="#000000" />   <TextView  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:layout\_weight="5"  android:layout\_marginRight="20dp"  android:gravity="center|left"  android:fontFamily="@font/tmonmonsori"  android:textSize="20dp"  android:text="dB마다 볼륨 증가"  android:textColor="#000000" />    </LinearLayout>   <!-- 체크 박스 -->  <CheckBox  android:id="@+id/vibratorCheckBox"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:layout\_marginLeft="-5dp"  android:layout\_marginBottom="40dp"  android:text="볼륨 변경 시 진동"  android:fontFamily="@font/tmonmonsori"  android:textSize="20dp"  android:textColor="#000000" />   <!-- 적용 버튼 -->  <Button  android:id="@+id/applyButton"  android:layout\_marginTop="0dp"  android:layout\_marginBottom="35dp"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:background="@drawable/btn\_bg"  android:fontFamily="@font/tmonmonsori"  android:textSize="20dp"  android:text="적 용"/>   </LinearLayout>  </android.support.constraint.ConstraintLayout> |

**> styles.xml**

|  |
| --- |
| <resources>   <!-- Base application theme. -->  <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">  <item name="windowNoTitle">true</item>  <!-- Customize your theme here. -->  <item name="colorPrimary">@color/colorPrimary</item>  <item name="colorPrimaryDark">@color/colorPrimaryDark</item>  <item name="colorAccent">@color/colorAccent</item>  </style>   <style name="SplashTheme" parent="AppTheme">  <item name="windowActionBar">false</item>  <item name="windowNoTitle">true</item>  <item name="android:windowBackground">@drawable/bg\_splash</item>  </style>  </resources> |

**> build.gradle (Module: app)**

|  |
| --- |
| apply plugin: 'com.android.application'  android {  compileSdkVersion 28  defaultConfig {  applicationId "com.example.smartsoundcontroller"  minSdkVersion 15  targetSdkVersion 28  versionCode 1  versionName "1.0"  testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"  }  buildTypes {  release {  minifyEnabled false  proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'  }  } }  dependencies {  implementation fileTree(dir: 'libs', include: ['\*.jar'])  implementation 'com.android.support:appcompat-v7:28.0.0'  implementation 'com.android.support.constraint:constraint-layout:1.1.3'  testImplementation 'junit:junit:4.12'  androidTestImplementation 'com.android.support.test:runner:1.0.2'  androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'  implementation 'com.akexorcist:bluetoothspp:1.0.0' } |