“Android”

Android

□ **Android**

|  |  |
| --- | --- |
| **3. Android 설정** | |
| 1. USB 드라이버 설치  2. USB 디버깅 항목 체크  3. 핸드폰 연결된 상태에서 디버깅  소스파일    1.MainActivity  2.AndroidManifest  3.activity\_main | |
| EMB000011404bd2 | EMB000011404bce |
| 4. **activity\_main 소스** | |
| <?xml version="1.0" encoding="utf-8"?>  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical" >  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="vertical" >  <EditText  android:id="@+id/ip\_EditText"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:hint="IP주소를입력하세요" />  <EditText  android:id="@+id/port\_EditText"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:hint="PORT주소를입력하세요" />  </LinearLayout>  <RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:gravity="center" >  <Button  android:id="@+id/connect\_Button"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="연결" />  </RelativeLayout>  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:orientation="horizontal" >  <ScrollView  android:layout\_width="match\_parent"  android:layout\_height="match\_parent" >  <TextView  android:id="@+id/showText\_TextView"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:text="" />  </ScrollView>  </LinearLayout>  <LinearLayout  android:layout\_width="fill\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal" >  <EditText  android:id="@+id/editText\_massage"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_weight="0.5"  android:ems="10"  android:hint="request/" />  <Button  android:id="@+id/Button\_send"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_weight="2"  android:text="전송" />  </LinearLayout>  </LinearLayout> | |
| **AndroidManifest 소스** | |
| <?xml version="1.0" encoding="utf-8"?>  <manifest xmlns:android="http://schemas.android.com/apk/res/android"  package="com.example.can.myapplication">  <uses-permission android:name="android.permission.INTERNET"/>  <uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE"/>  <uses-permission android:name="android.permission.ACCESS\_WIFI\_STATE"/>  <application  android:allowBackup="true"  android:icon="@mipmap/ic\_launcher"  android:label="@string/app\_name"  android:supportsRtl="true"  android:theme="@style/AppTheme">  <activity android:name=".MainActivity">  <intent-filter>  <action android:name="android.intent.action.MAIN" />  <category android:name="android.intent.category.LAUNCHER" />  </intent-filter>  </activity>  </application>  </manifest> | |
| **MainActivity 소스** | |
| package com.example.shin.test01;  import android.os.Bundle;  import android.app.Activity;  import android.os.Handler;  import android.os.Message;  import android.util.Log;  import android.view.View;  import android.widget.Button;  import android.widget.EditText;  import android.widget.TextView;  import java.io.DataInputStream;  import java.io.DataOutputStream;  import java.io.IOException;  import java.net.Socket;  import java.util.LinkedList;  public class MainActivity extends Activity {  TextView showText;  //EditText showText;  Button connectBtn;  Button Button\_send;  EditText ip\_EditText;  EditText port\_EditText;  EditText editText\_massage;  Handler msghandler;  SocketClient client; // 서버 접속으 위한 클라이언트 클래스  ReceiveThread receive; // 서버에서 보내온 데이터 안드로이드에서 보이게 하는 거  SendThread send; // 안드로이드에서 임의의 문자 보내는거  Socket socket; // 네트워크  LinkedList<SocketClient> threadList;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);//액티비티 초기화  setContentView(R.layout.activity\_main);//액티비티 레이아웃 설정  //안드로이드 view 소스코드 연동 래이아웃에 정의되어 있는 뷰  ip\_EditText = (EditText) findViewById(R.id.ip\_EditText);  port\_EditText = (EditText) findViewById(R.id.port\_EditText);  connectBtn = (Button) findViewById(R.id.connect\_Button);  showText = (TextView) findViewById(R.id.showText\_TextView);  editText\_massage = (EditText) findViewById(R.id.editText\_massage);  Button\_send = (Button) findViewById(R.id.Button\_send);  threadList = new LinkedList<MainActivity.SocketClient>();  ip\_EditText.setText("192.168.1.104");  port\_EditText.setText("9511");  // ReceiveThread를통해서 받은 메세지를 Handler로 MainThread에서 처리(외부Thread에서는 UI변경이불가)  msghandler = new Handler() {  @Override  public void handleMessage(Message hdmsg) {  if (hdmsg.what == 1111) {  //식별자.  showText.setText(hdmsg.obj.toString() + "\n");//보여줄 객체  }  }  };  // 연결버튼 클릭 이벤트  connectBtn.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View arg0) {  //Client 연결부  client = new SocketClient(ip\_EditText.getText().toString(),  port\_EditText.getText().toString());  threadList.add(client);  client.start();  }  });  //전송 버튼 클릭 이벤트  Button\_send.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View arg0) {  //SendThread 시작  if (editText\_massage.getText().toString() != null) {  send = new SendThread(socket);  send.start();  //시작후 edittext 초기화  editText\_massage.setText("");  }  }  });  }  class SocketClient extends Thread {  boolean threadAlive;  String ip;  String port;  DataOutputStream output = null; //byte 로 보내고 문자열로 읽고  public SocketClient(String ip, String port) {  threadAlive = true;  this.ip = ip;  this.port = port;  }  @Override  public void run() {  try {  // 연결후 바로 ReceiveThread 시작  socket = new Socket(ip, Integer.parseInt(port));  output = new DataOutputStream(socket.getOutputStream());  receive = new ReceiveThread(socket);  receive.start();  } catch (IOException e) {  e.printStackTrace();  }  }  }  class ReceiveThread extends Thread {  private Socket sock = null;  DataInputStream input;  public ReceiveThread(Socket socket) {  this.sock = socket;  try{  input = new DataInputStream(sock.getInputStream());  }catch(Exception e){  }  }  // 메세지 수신후 Handler로 전달  public void run() {  try {  while (input != null) {  String msg;  int count = input.available();  byte[] rcv = new byte[count];  input.read(rcv);  msg = new String(rcv);  if (count > 0) {  Log.d(ACTIVITY\_SERVICE, "test :" +msg);  Message hdmsg = msghandler.obtainMessage();  hdmsg.what = 1111;  hdmsg.obj = msg;  msghandler.sendMessage(hdmsg);  Log.d(ACTIVITY\_SERVICE,hdmsg.obj.toString());  }  }  } catch (IOException e) {  e.printStackTrace();  }  }  }  class SendThread extends Thread {  Socket socket;  String sendmsg = editText\_massage.getText().toString()+"\n";  DataOutputStream output;  public SendThread(Socket socket) {  this.socket = socket;  try {  output = new DataOutputStream(socket.getOutputStream());  } catch (Exception e) {  }  }  public void run() {  try {  // 메세지 전송부  Log.d(ACTIVITY\_SERVICE, "11111");  if (output != null) {  if (sendmsg != null) {  output.write(sendmsg.getBytes());  }  }  } catch (IOException e) {  e.printStackTrace();  } catch (NullPointerException npe) {  npe.printStackTrace();  }  }  }  } | |
| **3. Android 실행** | |
| 서버 ip  포트번호  연결후  Request/ 문자열 입력 후 전송버튼  Request/H 문자열 입력 후 전송버튼 | |
| **0.1.7에서 한 것으로 다른 버전과는 다를 수 있음을 먼저 알립니다.**  **0.1.8 도 비슷한 현상이 있습니다**  **서 빠르게 다음 단계로 이동하게 되는 일시적인 상태들이다. 즉, 처음에 시스템이 onCreate() 함수를 호출하고 이어서 onStart()를 호출한 다음에 이어서 onResume()을 빠르게 순서대로 호출하게 되는 것이다.**  세로모드에서 가로모드로 전환 시 onCreate함수가 다시 호출된다. 만약 전역변수를 설정하고 그 값을 유지하며 항상 사용해야 하는 경우라도 화면이 세로모드에서 가로모드로 변경될 경우 전역변수에 설정한 값이 모두 초기화 된다. 이런 경우 변경된 값을 유지하고 싶다면  savedInstanceState을 이용하는  onCreate(...) 메소드 안에 있는  setContentView(R.layout.activity\_main); 코드에서  ;를 지우고 실행 후, 다시 붙이는 것!  1. 안드로이드는 AndroidManifest.xml에 나와있는 Activity를 실행합니다.  2. MainActivity.java파일은 setContentView(R.layout.activity\_main);코드로 res/layout/activity\_main.xml을 호출합니다.  3. activity\_main.xml파일에서는 android:text="@string/hello\_world"구문으로 res/values/string.xml의 값을 참조 합니다4. hello\_world로 설정된 값을 activity\_main에 준다음, 완성된 화면을 MainActivity.java에 보내줍니다.  5. 마지막으로 기기에 화면을 띄웁니다.  rayList는 데이터들이 순서대로 쭉 늘어선 배열의 형식을 취하고 있는 반면 **LinkedList**는 순서대로 늘어선 것이 아니라 자료의 주소 값으로 서로 연결되어 ...  http://postfiles1.naver.net/20130626_288/theparanbi_1372239440952wIURv_PNG/Snap143.png?type=w2 | |

출처: [http://griper.tistory.com/entry/fillparent-와-wrapcontent-의-차이](http://griper.tistory.com/entry/fillparent-%EC%99%80-wrapcontent-%EC%9D%98-%EC%B0%A8%EC%9D%B4) [127.0.0.1]

|  |  |  |
| --- | --- | --- |
| **1** |  | **SERVER** |

□ **SERVER**

|  |
| --- |
| **1. SERVER1** |
| import java.net.\*;  import java.io.\*;  public class Server {  public static void main(String[] args) {  ServerSocket serversocket = null;  Socket socket = null;  try {  serversocket = new ServerSocket(9511);  socket = serversocket.accept();  System.out.println("ok");  Thread thread2 = new receiverthread(socket);  thread2.start();  } catch (Exception e1) {  } finally {  try {  // System.out.println("�뿰寃� �떎�뙣");  serversocket.close();  } catch (Exception e1) {    }  }  }  }  class receiverthread extends Thread {  Socket socket;  receiverthread(Socket socket) {  this.socket = socket;  }  public void run() {  //A A1 = new A();  try {  BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));  OutputStream out = socket.getOutputStream();  String str1 = in.readLine();  System.out.println(str1);  if(str1.equals("request")) {    String a = "okrequet";  out.write(a.getBytes());  }  while (true) {  String str = in.readLine();  System.out.println(str);  // A1.insert(str);  if (str == null)  break;  }  } catch (Exception e1) {  } finally {  try {  socket.close();  } catch (Exception e1) {  }  }  }  } |
| **2. server2 난수발생** |
| **package** socketServer;  **import** java.net.\*;  **import** java.util.Random;  **import** java.io.\*;  **public** **class** server {  **public** **static** **void** main(String[] args) {    ServerSocket serversocket = **null**;  Socket socket = **null**;    **try** {  serversocket = **new** ServerSocket(9511);  socket = serversocket.accept();  System.***out***.println("ok");  Thread thread2 = **new** receiverthread(socket);  thread2.start();  } **catch** (Exception e1) {  } **finally** {  **try** {  // System.out.println("�뿰寃� �떎�뙣");  serversocket.close();  } **catch** (Exception e1) {    }  }  }  }  **class** receiverthread **extends** Thread {  Socket socket;  receiverthread(Socket socket) {  **this**.socket = socket;  }  **public** **void** run() {  //A A1 = new A();  **try** {  Random rand = **new** Random();  BufferedReader in = **new** BufferedReader(**new** InputStreamReader(socket.getInputStream()));  OutputStream out = socket.getOutputStream();  String str1 = in.readLine();  System.***out***.println(str1);  **if**(str1.equals("aSensor")) {    **int** n = rand.nextInt(100);  String str3 = Integer.*toString*(n);  String a = "asensor\n" + str3;    out.write(a.getBytes());  }**else** **if**(str1.equals("bSensor")) {  **int** p = rand.nextInt(100);  String str3 = Integer.*toString*(p);  String a = "bsensor\n" + str3;  out.write(a.getBytes());  }  **while** (**true**) {  String str = in.readLine();  System.***out***.println(str);  // A1.insert(str);  **if** (str == **null**)  **break**;  }  } **catch** (Exception e1) {  } **finally** {  **try** {  socket.close();  } **catch** (Exception e1) {  }  }  }  } |

fill\_parent : 부모가 가지는 길이를 모두 채울 때 사용한다.

wrap\_content : 해당 뷰가 그려질 수 있게 필요한 길이만 사용한다.

gravity : 뷰의 위치 지정

fill\_parent  : 부모가 가지는 길이를 모두 채울 때 사용한다.   
  
출처: [http://griper.tistory.com/entry/fillparent-와-wrapcontent-의-차이](http://griper.tistory.com/entry/fillparent-%EC%99%80-wrapcontent-%EC%9D%98-%EC%B0%A8%EC%9D%B4) [127.0.0.1]

|  |  |  |
| --- | --- | --- |
| **1** |  | **Layout** |

□ **Layout**

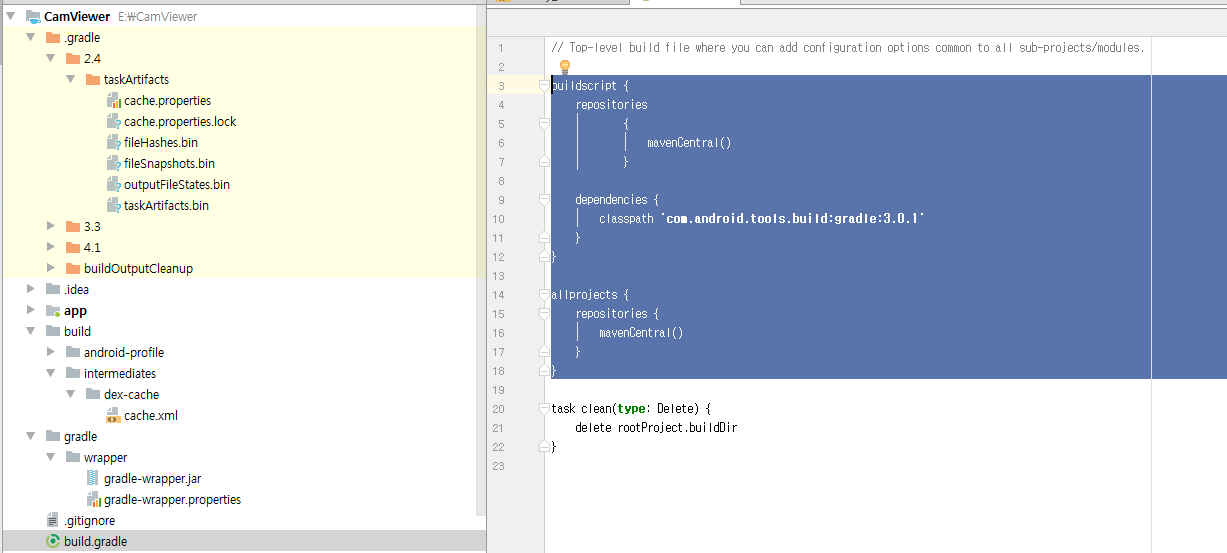
|  |
| --- |
| **1. Linear Layout** |
|  |
| **2. active\_main.xml** |
| <?xml version="1.0" encoding="utf-8"?>  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="fill\_parent"  android:layout\_height="fill\_parent"  android:orientation="vertical" >  <TextView android:id="@+id/text"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Hello, I am a TextView" />  <Button android:id="@+id/button"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Hello, I am a Button" />  </LinearLayout> |
| **2. table layout** |
|  |
| <TableLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_margin="8dp"  android:background="#404040"  app:layout\_constraintLeft\_toLeftOf="parent"  app:layout\_constraintRight\_toRightOf="parent"  app:layout\_constraintTop\_toTopOf="parent"  app:layout\_constraintBottom\_toBottomOf="parent"  android:stretchColumns="\*">  <TextView  android:layout\_width="wrap\_content"  android:layout\_height="0dp"  android:layout\_weight="4"  android:textSize="60sp"  android:textColor="#FFFFFF"  android:background="#607D8B"  android:gravity="right|center\_vertical"  android:text="0" />  <TableRow  android:layout\_width="wrap\_content"  android:layout\_height="0dp"  android:layout\_weight="1">  <Button  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:textSize="32sp"  android:text="7" />  <Button  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:textSize="32sp"  android:text="8" />  <Button  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:textSize="32sp"  android:text="9" />  <Button  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:layout\_span="2"  android:textSize="32sp"  android:text="DEL" />  </TableRow>  <TableRow  android:layout\_width="wrap\_content"  android:layout\_height="0dp"  android:layout\_weight="1">  <Button  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:textSize="32sp"  android:text="4" />  <Button  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:textSize="32sp"  android:text="5" />  <Button  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:textSize="32sp"  android:text="6" />  <Button  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:textSize="32sp"  android:text="+" />  <Button  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:textSize="32sp"  android:text="-" />  </TableRow>  <TableRow  android:layout\_width="wrap\_content"  android:layout\_height="0dp"  android:layout\_weight="1">  <Button  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:textSize="32sp"  android:text="1" />  <Button  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:textSize="32sp"  android:text="2" />  <Button  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:textSize="32sp"  android:text="3" />  <Button  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:textSize="32sp"  android:text="\*" />  <Button  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:textSize="32sp"  android:text="/" />  </TableRow>  <TableRow  android:layout\_width="wrap\_content"  android:layout\_height="0dp"  android:layout\_weight="1">  <Button  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:layout\_span="2"  android:textSize="32sp"  android:text="0" />  <Button  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:textSize="32sp"  android:text="." />  <Button  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:layout\_span="2"  android:textSize="32sp"  android:text="=" />  </TableRow>  </TableLayout> |

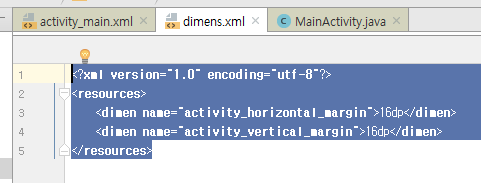
|  |  |  |
| --- | --- | --- |
| **1** |  | **Up down, left, right** |

□ **jdk**

|  |
| --- |
| **1.layout** |
| <?xml version="1.0" encoding="utf-8"?>  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools" android:layout\_width="match\_parent"  android:layout\_height="match\_parent" android:paddingLeft="@dimen/activity\_horizontal\_margin"  android:paddingRight="@dimen/activity\_horizontal\_margin"  android:paddingTop="@dimen/activity\_vertical\_margin"  android:paddingBottom="@dimen/activity\_vertical\_margin" tools:context=".MainActivity"  android:background="#d9d8d8">  <VideoView  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:id="@+id/videoView"  android:layout\_above="@+id/upButton"  android:visibility="invisible" />  <ImageButton  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:id="@+id/upButton"  android:src="@mipmap/b\_up"  android:layout\_above="@+id/rightButton"  android:layout\_centerInParent="true" />  <ImageButton  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:id="@+id/rightButton"  android:src="@mipmap/b\_right"  android:layout\_above="@+id/downButton"  android:layout\_toRightOf="@+id/downButton"  android:layout\_toEndOf="@+id/downButton" />  <ImageButton  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:id="@+id/leftButton"  android:src="@mipmap/b\_left"  android:layout\_below="@+id/upButton"  android:layout\_toLeftOf="@+id/upButton"  android:layout\_toStartOf="@+id/upButton" />  <ImageButton  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:id="@+id/downButton"  android:src="@mipmap/b\_down"  android:layout\_alignParentBottom="true"  android:layout\_alignLeft="@+id/upButton"  android:layout\_alignStart="@+id/upButton"  android:visibility="invisible" />  <ImageButton  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:id="@+id/poweroff"  android:layout\_alignParentRight="false"  android:layout\_alignParentEnd="true"  android:layout\_alignParentBottom="true"  android:background="@mipmap/power"  android:layout\_toEndOf="@+id/rightButton"  android:layout\_alignParentTop="false"  android:layout\_alignWithParentIfMissing="false"  android:layout\_alignParentLeft="false"  android:layout\_alignTop="@+id/downButton"  android:layout\_alignBottom="@+id/config" />  <ImageButton  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:id="@+id/config"  android:layout\_alignParentBottom="false"  android:layout\_toRightOf="@+id/rightButton"  android:layout\_alignParentStart="false"  android:layout\_alignBottom="@+id/rightButton"  android:layout\_alignParentLeft="false"  android:visibility="visible"  android:layout\_alignParentTop="false"  android:background="@mipmap/sense"  android:layout\_alignTop="@+id/rightButton"  android:layout\_toLeftOf="@+id/leftButton"  android:layout\_alignParentRight="true" />  <RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:id="@+id/setting"  android:background="#d9d8d8"  android:visibility="invisible"  android:focusableInTouchMode="false"  android:layout\_alignParentTop="true"  android:layout\_alignParentLeft="true">  <Button  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="SAVE"  android:id="@+id/b\_setting"  android:textSize="20dp"  android:layout\_alignParentBottom="true"  android:layout\_toLeftOf="@+id/b\_cancel"  android:layout\_toStartOf="@+id/b\_cancel"  android:layout\_marginBottom="104dp" />  <EditText  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:id="@+id/t\_address"  android:layout\_above="@+id/textView2"  android:layout\_alignParentRight="true"  android:layout\_alignParentEnd="true"  android:layout\_toRightOf="@+id/textView"  android:layout\_toEndOf="@+id/textView"  android:textColor="#000000"  android:text="192.168.42.1" />  <EditText  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:inputType="numberDecimal"  android:ems="10"  android:id="@+id/t\_port"  android:layout\_alignBottom="@+id/textView2"  android:layout\_toRightOf="@+id/textView2"  android:layout\_alignParentRight="true"  android:layout\_alignParentEnd="true"  android:textColor="#000000"  android:text="9000" />  <TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Address"  android:id="@+id/textView"  android:textColor="#000000"  android:textSize="25dp"  android:textAlignment="center"  android:layout\_marginTop="46dp"  android:layout\_alignParentTop="true"  android:layout\_alignParentLeft="true"  android:layout\_alignParentStart="true" />  <TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Port"  android:id="@+id/textView2"  android:textColor="#000000"  android:textSize="25dp"  android:layout\_below="@+id/textView"  android:layout\_alignParentLeft="true"  android:layout\_alignParentStart="true"  android:layout\_marginTop="36dp"  android:layout\_alignRight="@+id/textView"  android:layout\_alignEnd="@+id/textView"  android:textAlignment="center" />  <Button  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="CANCEL"  android:id="@+id/b\_cancel"  android:textSize="20dp"  android:layout\_alignTop="@+id/b\_setting"  android:layout\_alignParentRight="true"  android:layout\_alignParentEnd="true" />  <TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Angle"  android:id="@+id/textView3"  android:textAlignment="center"  android:textColor="#000000"  android:textSize="25dp"  android:layout\_below="@+id/textView2"  android:layout\_alignParentLeft="true"  android:layout\_alignParentStart="true"  android:layout\_marginTop="30dp"  android:layout\_alignRight="@+id/textView2"  android:layout\_alignEnd="@+id/textView2" />  <EditText  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:inputType="numberDecimal"  android:ems="10"  android:id="@+id/t\_angle"  android:layout\_alignBottom="@+id/textView3"  android:layout\_toRightOf="@+id/textView3"  android:layout\_alignParentRight="true"  android:layout\_alignParentEnd="true"  android:text="30" />  </RelativeLayout>  </RelativeLayout> |
| **2. jdk** |
| <?xml version="1.0" encoding="utf-8"?>  <manifest xmlns:android="http://schemas.android.com/apk/res/android"  package="com.example.cv.camviewer" >  <application  android:allowBackup="true"  android:icon="@mipmap/ic\_launcher"  android:label="@string/app\_name"  android:supportsRtl="true"  android:theme="@style/AppTheme" >  <activity android:name=".MainActivity" >  <intent-filter>  <action android:name="android.intent.action.MAIN" />  <category android:name="android.intent.category.LAUNCHER" />  </intent-filter>  </activity>  </application>  <uses-permission android:name="android.permission.INTERNET" />  </manifest> |
| **2. jdk** |
| package com.example.cv.camviewer;  import android.media.Image;  import android.support.v7.app.AppCompatActivity;  import android.os.Bundle;  import android.view.\*;  import android.widget.Button;  import android.widget.EditText;  import android.widget.ImageButton;  import android.widget.RelativeLayout;  import java.io.IOException;  import java.io.PrintWriter;  import java.net.Socket;  public class MainActivity extends AppCompatActivity {  ImageButton b\_up;  private String ip = "";  private Socket socket;  private PrintWriter socket\_out;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);  b\_up = (ImageButton) findViewById(R.id.upButton);  subThread.start();  }  @Override  public boolean onCreateOptionsMenu(Menu menu) {  menu.add(0, 0, 0, "Connect");  menu.add(0, 1, 0, "Setting");  menu.add(0, 2, 0, "Quit");  return true;  }  @Override  public boolean onOptionsItemSelected(MenuItem item){  switch(item.getItemId()){  case 0:  if(ip != null && port > 0) {  connectThread.start();  sendThread1.start();  sendThread2.start();  sendThread3.start();  }  break;  case 1:  setting.setVisibility(View.VISIBLE);  break;  case 2:  if(socket != null) {  try {  socket\_out.close();  socket.close();  } catch (IOException e) {  e.printStackTrace();  }  finish();  }  else finish();  break;  }  return super.onOptionsItemSelected(item);  }  @Override  protected void onStop() {  if(socket != null) {  try {  socket\_out.close();  socket.close();  } catch (IOException e) {  e.printStackTrace();  }  }  super.onStop();  }  Thread subThread = new Thread() {  public synchronized void run() {  b\_setting.setOnClickListener(new View.OnClickListener() {  public void onClick(View v) {  ip = t\_address.getText().toString();  port = Integer.parseInt(t\_port.getText().toString());  angle = Integer.parseInt(t\_angle.getText().toString());  if(angle > 180 || angle < 0) angle = 30;  setting.setVisibility(View.INVISIBLE);  /\*  if(ip != null && port > 0) {  connectThread.start();  sendThread1.start();  sendThread2.start();  sendThread3.start();  }  \*/  }  });  b\_cancel.setOnClickListener(new View.OnClickListener() {  public void onClick(View v) {  setting.setVisibility(View.INVISIBLE);  }  });  }  };  Thread connectThread = new Thread() {  public synchronized void run() {  try {  socket = new Socket(ip, port);  socket\_out = new PrintWriter(socket.getOutputStream(), true);  } catch (IOException e) {  e.printStackTrace();  }  }  };  Thread sendThread1 = new Thread() {  public synchronized void run() {  b\_up.setOnClickListener(new View.OnClickListener() {  public void onClick(View v) {  if (socket\_out != null) {  socket\_out.println("up" + angle);  }  }  });  b\_down.setOnClickListener(new View.OnClickListener() {  public void onClick(View v) {  if (socket\_out != null) {  socket\_out.println("down" + angle);  }  }  });  }  };  Thread sendThread2 = new Thread() {  public synchronized void run() {  b\_right.setOnClickListener(new View.OnClickListener() {  public void onClick(View v) {  if (socket\_out != null) {  socket\_out.println("right" + angle);  }  }  });  b\_left.setOnClickListener(new View.OnClickListener() {  public void onClick(View v) {  if (socket\_out != null) {  socket\_out.println("left" + angle);  }  }  });  }  };  Thread sendThread3 = new Thread() {  public synchronized void run() {  b\_poweroff.setOnClickListener(new View.OnClickListener() {  public void onClick(View v) {  if (socket\_out != null) {  socket\_out.println("poweroff" + angle);  }  }  });  b\_config.setOnClickListener(new View.OnClickListener() {  public void onClick(View v) {  if (socket\_out != null) {  setting.setVisibility(View.VISIBLE);  socket\_out.println("config" + angle);  }  }  });  }  };  } |

buildscript {  
 repositories  
 {  
 mavenCentral()  
 }  
  
 dependencies {  
 classpath **'com.android.tools.build:gradle:3.0.1'** }  
}  
  
allprojects {  
 repositories {  
 mavenCentral()  
 }  
}





*<?***xml version="1.0" encoding="utf-8"***?>*<**resources**>  
 <**dimen name="activity\_horizontal\_margin"**>16dp</**dimen**>  
 <**dimen name="activity\_vertical\_margin"**>16dp</**dimen**>  
</**resources**>

|  |  |  |
| --- | --- | --- |
| **1** |  | **intent** |

□ **intent**

|  |
| --- |
| **1. Activity Layout** |
| *<?***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="com.example.root.myapplication.MainActivity"**>   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Hello World!"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintLeft\_toLeftOf="parent"  app:layout\_constraintRight\_toRightOf="parent"  app:layout\_constraintTop\_toTopOf="parent"** />   <**LinearLayout  android:layout\_width="368dp"  android:layout\_height="231dp"  android:orientation="vertical"  tools:layout\_editor\_absoluteX="8dp"  tools:layout\_editor\_absoluteY="8dp"**>   <**Button  android:id="@+id/button1"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="first"** />   <**Button  android:id="@+id/button2"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="exit"** />  </**LinearLayout**>  </**android.support.constraint.ConstraintLayout**> |
| **2. intentfirst** |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**android.support.constraint.ConstraintLayout  xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"**>   <**LinearLayout  android:layout\_width="368dp"  android:layout\_height="495dp"  android:orientation="vertical"  tools:layout\_editor\_absoluteX="8dp"  tools:layout\_editor\_absoluteY="8dp"  tools:ignore="MissingConstraints"**>   <**Button  android:id="@+id/fbutton1"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="main으로"** />   <**Button  android:id="@+id/fbutton2"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="종료"** />  </**LinearLayout**> </**android.support.constraint.ConstraintLayout**> |
| **2. Activitymain.java** |
| **package** com.example.root.myapplication;  **import** android.app.Activity; **import** android.content.Intent; **import** android.support.v7.app.AppCompatActivity; **import** android.os.Bundle; **import** android.view.View; **import** android.widget.Button;  **public class** MainActivity **extends** Activity **implements** View.OnClickListener { Button **button1**, **button2**;  @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  **button1** = (Button) findViewById(R.id.***button1***);  **button2** = (Button) findViewById(R.id.***button2***);   **button1**.setOnClickListener((View.OnClickListener) **this**);  **button2**.setOnClickListener((View.OnClickListener) **this**);  }   @Override  **public void** onClick(View view) {  **if**(view.getId()==R.id.***button1***){  Intent intent = **new** Intent(**this**,intentfirst.**class**);  startActivity(intent);  }**else**{  finish();  }  } } |
| **2. intentfirst** |
| **package** com.example.root.myapplication;  **import** android.app.Activity; **import** android.content.DialogInterface; **import** android.content.Intent; **import** android.os.Bundle; **import** android.view.View; **import** android.widget.Button;  */\*\*  \* Created by root on 2018-03-28.  \*/* **public class** intentfirst **extends** Activity **implements** View.OnClickListener {   Button **fbutton1**, **fbutton2**;  **protected void** onCreate(Bundle savedInstanceState) {   **super**.onCreate(savedInstanceState);  setContentView(R.layout.***intentfirst***);  **fbutton1** = (Button)findViewById(R.id.***fbutton1***);  **fbutton2** = (Button)findViewById(R.id.***fbutton2***);  **fbutton1**.setOnClickListener(**this**);  **fbutton2**.setOnClickListener(**this**);    }  @Override  **public void** onClick(View view) {  **if**(view.getId()==R.id.***fbutton1***){  Intent intent = **new** Intent(**this**,MainActivity.**class**);  startActivity(intent);  }**else**{  finish();  }  } } |
| **2. manifests** |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**manifest xmlns:android="http://schemas.android.com/apk/res/android"  package="com.example.root.myapplication"**>   <**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=".MainActivity"**>  <**intent-filter**>  <**action android:name="android.intent.action.MAIN"** />   <**category android:name="android.intent.category.LAUNCHER"** />  </**intent-filter**>  </**activity**>  <**activity android:name=".intentfirst"** />  </**application**>  </**manifest**> |

|  |  |  |
| --- | --- | --- |
| **1** |  | **Listview** |

□ **Listview**

|  |
| --- |
| **1. Activity Layout** |
| <ListView  android:id="@+id/listView1"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_centerHorizontal="true" >  </ListView> |
| **2. java source** |
| package com.hanbit.test11;  import java.util.ArrayList;  import android.os.Bundle;  import android.app.Activity;  import android.view.Menu;  import android.widget.ArrayAdapter;  import android.widget.ListView;  public class MainActivity extends Activity {  ListView listView;  // listView에 들어갈 데이터을 저장하는 객체배열  ArrayList<String> list = new ArrayList<String>();  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);  // 리스트뷰 참조 객체 얻어오기  listView = (ListView)findViewById(R.id.listView1);  // 배열에 데이터를 저장  list.add("홍길동");  list.add("고길동");  list.add("김길동");  list.add("박길동");  list.add("이길동");  // 리스트뷰와 데이터를 연결하기 위한 어탭터 생성  // 배열과 연결할때는 ArrayAdapter를 사용  ArrayAdapter<String> adapter =  new ArrayAdapter<String>(  this, // 컨텐스트 객체  android.R.layout.simple\_list\_item\_1, // 보여줄 레이아웃  list); // 리스트뷰와 연결될 배열객체  // 연결하기  listView.setAdapter(adapter);  }  } |
| **2. Activitymain.java** |
|  |
| **2. intentfirst** |
|  |
| **2. manifests** |
|  |

|  |  |  |
| --- | --- | --- |
| **1** |  | **radio** |

□ **radio**

|  |
| --- |
| **1. Activity Layout** |
| <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:paddingBottom="@dimen/activity\_vertical\_margin"  android:paddingLeft="@dimen/activity\_horizontal\_margin"  android:paddingRight="@dimen/activity\_horizontal\_margin"  android:paddingTop="@dimen/activity\_vertical\_margin"  tools:context=".MainActivity" >  <TextView  android:id="@+id/textView1"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_alignLeft="@+id/checkBox1"  android:layout\_below="@+id/radioGroup1"  android:layout\_marginTop="36dp"  android:text="Large Text"  android:textSize="20dp"  android:textAppearance="?android:attr/textAppearanceLarge" />  <RadioGroup  android:id="@+id/radioGroup1"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_below="@+id/checkBox1"  android:layout\_centerHorizontal="true"  android:layout\_marginTop="42dp"  android:orientation="horizontal" >  <RadioButton  android:id="@+id/radio0"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:checked="true"  android:text="빨강" />  <RadioButton  android:id="@+id/radio1"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:text="파랑" />  <RadioButton  android:id="@+id/radio2"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:text="초록" />  </RadioGroup>  <CheckBox  android:id="@+id/checkBox1"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_alignLeft="@+id/radioGroup1"  android:layout\_alignParentTop="true"  android:layout\_marginTop="53dp"  android:text="TextSize" />  </RelativeLayout> |
| **2. java source** |
| package com.hanbit.test06;  import android.app.Activity;  import android.graphics.Color;  import android.os.Bundle;  import android.widget.CheckBox;  import android.widget.CompoundButton;  import android.widget.RadioGroup;  import android.widget.RadioGroup.OnCheckedChangeListener;  import android.widget.TextView;  public class MainActivity extends Activity {  TextView textView;  RadioGroup radioGroup;  CheckBox checkBox;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);  textView = (TextView) findViewById(R.id.textView1);  radioGroup = (RadioGroup) findViewById(R.id.radioGroup1);  checkBox = (CheckBox) findViewById(R.id.checkBox1);  radioGroup.setOnCheckedChangeListener(new OnCheckedChangeListener() {  @Override  public void onCheckedChanged(RadioGroup group, int checkedId) {  // 체크된 라이오버튼 id가 checkedId에 자동으로 들어간다.  switch (checkedId) {  case R.id.radio0:  textView.setTextColor(Color.RED);  break;  case R.id.radio1:  textView.setTextColor(Color.BLUE);  break;  case R.id.radio2:  textView.setTextColor(Color.GREEN);  break;  }  }  });  checkBox.setOnCheckedChangeListener(  new CompoundButton.OnCheckedChangeListener() {  @Override  public void onCheckedChanged(CompoundButton buttonView,  boolean isChecked) {  // isChecked => 체크상태를 넘겨 받는것(true=>체크된상태)  if(isChecked){ // 체크된 상태  textView.setTextSize(40);  }else{ // 체크가 안된 상태  textView.setTextSize(20);  }  }  });  }  } |
|  |
|  |
|  |
|  |
|  |
|  |

|  |  |  |
| --- | --- | --- |
| **1** |  | **파일** |

□ **파일쓰기**

|  |
| --- |
| **1. Activity Layout** |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**android.support.constraint.ConstraintLayout  xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools"  xmlns:app="http://schemas.android.com/apk/res-auto"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  tools:context="com.example.root.myapplication.MainActivity"**>   <**LinearLayout  android:layout\_width="368dp"  android:layout\_height="495dp"  android:orientation="vertical"  tools:layout\_editor\_absoluteX="8dp"  tools:layout\_editor\_absoluteY="8dp"**>  <**EditText  android:id="@+id/editText1"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:ems="10"** >   <**requestFocus** />  </**EditText**>  <**EditText  android:id="@+id/editText2"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="0.89"  android:ems="10"  android:inputType="textMultiLine"** />  <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"** >   <**Button  android:id="@+id/button1"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="저장"** />   <**Button  android:id="@+id/button2"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="읽기"** />   <**Button  android:id="@+id/button3"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="삭제"** />   <**Button  android:id="@+id/button4"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="읽기2"** />    </**LinearLayout**>     </**LinearLayout**> </**android.support.constraint.ConstraintLayout**> |
| **2. java source** |
| **package** com.example.root.myapplication;  **import** java.io.FileInputStream; **import** java.io.FileOutputStream; **import** java.io.InputStream;  **import** android.app.Activity; **import** android.content.Context; **import** android.os.Bundle; **import** android.util.Log; **import** android.view.View; **import** android.widget.Button; **import** android.widget.EditText; **import** android.widget.Toast; **public class** MainActivity **extends** Activity {    EditText **editText1**, **editText2**;  Button **button1**, **button2**, **button3**, **button4**;    @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  **editText1** = (EditText) findViewById(R.id.***editText1***);  **editText2** = (EditText) findViewById(R.id.***editText2***); // 각 Button의 이벤트 리스너로 onClickListener 지정.   **button1** = (Button) findViewById(R.id.***button1***);  **button2** = (Button) findViewById(R.id.***button2***);  **button3** = (Button) findViewById(R.id.***button3***);  **button4** = (Button) findViewById(R.id.***button4***);  *// 저장* **button1**.setOnClickListener(**new** View.OnClickListener() {   **public void** onClick(View v) {  *// 파일에 저장하기 위한 객체 생성* FileOutputStream fos = **null**;  **try** {  *// 파일 생성 모드  // MODE\_PRIVATE : 혼자만 사용하는배타적 모드 (기본)  // MODE\_APPEND : 추가모든  // MODE\_WORLD\_READABLE : 다른 프로그램이 읽을 수 있도록  // MODE\_WORLD\_WRITABLE : 다른 프로그램이 쓸수 있도록   // 다른 프로그램이 읽기만 가능하도록 파일 생성* fos = openFileOutput(**"test.txt"**,  Context.***MODE\_WORLD\_READABLE***);   *// 입력된 문자열 얻어오기* String s = **editText1**.getText().toString();   *// 파일에 저장하기* fos.write(s.getBytes());   *// 파일 닫기* fos.close();   **editText1**.setText(**""**);  Toast.*makeText*(MainActivity.**this**,  **"저장 OK!"**, 5000).show();  } **catch** (Exception e) {  Log.*i*(**"MyLog"**,**"저장실패"**);  }   }  });   *//읽기* **button2**.setOnClickListener(**new** View.OnClickListener() {   @Override  **public void** onClick(View v) {  FileInputStream fis = **null**;  **try** {  fis = openFileInput(**"test.txt"**);  StringBuffer sb = **new** StringBuffer();  **byte**[] b = **new byte**[1024];  **int** n = 0 ;  **while**((n=fis.read(b)) != -1){  *// 바이트 배열을 String객체로 변환 해서 sb에 append* sb.append(**new** String(b,0,n));  }   *// 에디트 텍스트에 문자열 넣기* **editText2**.setText(sb.toString());   fis.close();   } **catch** (Exception e) {  Log.*i*(**"MyLog"**,**"읽기실패"**);  }   }  });   *//삭제* **button3**.setOnClickListener(**new** View.OnClickListener() {   @Override  **public void** onClick(View v) {  **if**(deleteFile(**"test.txt"**)){ *// deleteFile("test.txt") 파일삭제* Toast.*makeText*(MainActivity.**this**, **"삭제성공!"**, 5000).show();  }**else**{  Toast.*makeText*(MainActivity.**this**, **"삭제실패!"**, 5000).show();  }   }  });   }  } |
|  |
|  |
|  |
|  |
|  |
|  |

|  |  |  |
| --- | --- | --- |
| **1** |  | **진동** |

□ **진동**

|  |
| --- |
| **1. Activity Layout** |
| *<?***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="com.example.root.myapplication.MainActivity"**>    <**LinearLayout  android:layout\_width="368dp"  android:layout\_height="495dp"  android:orientation="vertical"  tools:layout\_editor\_absoluteX="8dp"  tools:layout\_editor\_absoluteY="8dp"**>   <**Button  android:id="@+id/button1"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="진동"** />   <**Button  android:id="@+id/button2"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="계속진동"** />   <**Button  android:id="@+id/button3"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="진동중지"** />  </**LinearLayout**> </**android.support.constraint.ConstraintLayout**> |
| **2. java source** |
| **package** com.example.root.myapplication;  **import** android.support.v7.app.AppCompatActivity; **import** android.os.Bundle; **import** android.os.Vibrator; **import** android.app.Activity; **import** android.content.Context; **import** android.view.Menu; **import** android.view.View; **import** android.view.View.OnClickListener;  **public class** MainActivity **extends** Activity **implements** OnClickListener {  Vibrator **mVib**;  @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);   **mVib** = (Vibrator) getSystemService(Context.***VIBRATOR\_SERVICE***);   findViewById(R.id.***button1***).setOnClickListener(**this**);  findViewById(R.id.***button2***).setOnClickListener(**this**);  findViewById(R.id.***button3***).setOnClickListener(**this**);   }    **public void** onClick(View v) {  **switch** (v.getId()) {  **case** R.id.***button1***:  **mVib**.vibrate(500);  **break**;  **case** R.id.***button2***:  **mVib**.vibrate(**new long**[]{100,50,200,50}, 0);  **break**;  **case** R.id.***button3***:  **mVib**.cancel();  **break**;   }   }   @Override  **protected void** onDestroy() {  **super**.onDestroy();  **mVib**.cancel();  }  } |
| **3,manifest.xml** |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**manifest xmlns:android="http://schemas.android.com/apk/res/android"  package="com.example.root.myapplication"**>   <**uses-permission android:name="android.permission.VIBRATE"** />  <**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=".MainActivity"**>  <**intent-filter**>  <**action android:name="android.intent.action.MAIN"** />   <**category android:name="android.intent.category.LAUNCHER"** />  </**intent-filter**>  </**activity**>  </**application**>  </**manifest**> |
|  |
|  |
|  |
|  |

|  |  |  |
| --- | --- | --- |
| **1** |  | **버튼 이벤트** |

□ **진동**

|  |
| --- |
| **1. Activity Layout** |
| **package** com.example.root.myapplication; *//import android.R;* **import** android.support.v7.app.AppCompatActivity; **import** android.os.Bundle; **import** android.view.View; **import** android.app.Activity; **import** android.widget.\*; **import** java.io.DataInputStream; **import** java.io.DataOutputStream; **import** java.io.IOException; **import** java.net.Socket; **import** java.util.LinkedList; *//Activity : 앱을 실행할때 보이는 화면이 Activity /\* Activity 생명주기 onCreate() : 액티비티 최초생성 onStart(); : 액티비티가 실행되고 사용자 눈에 보이는 시점 onRestart() : 액티비티가 중단되었다가 재시작될때 호출 되는 메소드 onResume() : 사용자와 상호작용하는 메서드 onPause() : 다른 액티비티가 호출되는 메서드 onStop() : 액티비티가 더이상 보이지 않을때 호출 되는 메서드 onDestroy() : 액티비티가 소멸될 때 호출되는 메소드 정상종료 Bundle) -> onStart() -> onResume() -> onPause() -> onStop() -> onDestory()   \*/* **public class** MainActivity **extends** Activity {   Button **button2**; *//뷰 인스턴스 선언* @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***); *//정의된 레이아웃를 호출  // button2 = (Button)findViewById(R.id.button2);* **button2** = (Button) findViewById(R.id.***button2***); *//레이아웃 뷰와 코드내인스턴스를 연결, 뷰 인스턴스 초기화 //인스턴스  //class = 뿡어빵틀 Object=뿡어빵Instance = 각각의 뿡어빵  //인스턴스화한다. = 뿡어빵을 굽다.* **button2**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View view) {  Toast.*makeText*(MainActivity.**this**,**"aaaa"**, Toast.***LENGTH\_LONG***).show();  }  });    } } |
| **2. java source** |
|  |
| **3,manifest.xml** |
|  |
|  |
|  |
|  |
|  |

**package** com.example.root.myapplication;  
*//import android.R;***import** android.os.Handler;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.app.Activity;  
**import** android.widget.\*;  
**import** java.io.DataInputStream;  
**import** java.io.DataOutputStream;  
**import** java.io.IOException;  
**import** java.net.Socket;  
**import** java.net.UnknownHostException;  
**import** java.util.LinkedList;  
**import** android.os.Handler;  
**import** java.net.Socket;  
*//Activity : 앱을 실행할때 보이는 화면이 Activity  
/\*  
Activity 생명주기  
onCreate() : 액티비티 최초생성  
onStart(); : 액티비티가 실행되고 사용자 눈에 보이는 시점  
onRestart() : 액티비티가 중단되었다가 재시작될때 호출 되는 메소드  
onResume() : 사용자와 상호작용하는 메서드  
onPause() : 다른 액티비티가 호출되는 메서드  
onStop() : 액티비티가 더이상 보이지 않을때 호출 되는 메서드  
onDestroy() : 액티비티가 소멸될 때 호출되는 메소드  
정상종료  
Bundle) -> onStart() -> onResume() -> onPause() -> onStop() -> onDestory()  
Handler : 메인 스레드와 서브쓰레드 간에 handler를 통해 메시지를 전달하여 메시지 큐에  
저장하는 방식 \*/***public class** MainActivity **extends** Activity {  
  
 Button **button2**; *//뷰 인스턴스 선언* EditText **editText4**;  
 EditText **editText5**;  
  
 Handler **msghandler**;  
 SocketClient **client**;  
*// ReceiveThread receive;  
// SendThread send;* Socket **socket**;  
  
 LinkedList<SocketClient>**threadLista**; *//객체 생성시 데이터 저장 영역이 생기지 않으며, 서로 인접 데이터를 가리킨다.  
 //ArrayList 객체 생성과 함계 데이터를 저장할 수있는 영역이 존재함  
//LinkedList저장 영역에 SocketClient ip port 가져와 영역생성* @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***); *//정의된 레이아웃를 호출  
  
// button2 = (Button)findViewById(R.id.button2);* **button2** = (Button) findViewById(R.id.***button2***); *//레이아웃 뷰와 코드내인스턴스를 연결, 뷰 인스턴스 초기화* **editText4** = (EditText) findViewById(R.id.***editText4***);  
 **editText5** = (EditText) findViewById(R.id.***editText5***);  
  
 **editText4**.setText(**"192.168.0.20"**);  
 **editText5**.setText(**"5001"**);  
  
*//인스턴스  
 //class = 뿡어빵틀 Object=뿡어빵Instance = 각각의 뿡어빵  
 //인스턴스화한다. = 뿡어빵을 굽다.* **button2**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 Toast.*makeText*(MainActivity.**this**,**"aaaa"**, Toast.***LENGTH\_LONG***).show();  
 **client** = **new** SocketClient(**editText4**.getText().toString(),**editText5**.getText().toString());  
 **threadLista**.add(**client**);  
 **client**.start();  
  
 }  
 });  
  
  
 }  
}  
  
**class** SocketClient **extends** Thread{  
 **boolean threadAlive**; *//thread를 연결이나 종료 여부 상태를 체크* String **ip**;  
 String **port**;  
 DataOutputStream **output** = **null**;  
 Socket **socket**;  
 **public** SocketClient(String ip, String port){  
 **threadAlive** = **true**;  
 **this**.**ip** = ip;  
 **this**.**port** = port;  
 }  
 **public void** run(){  
 **try** {  
 **socket** = **new** Socket(**ip**,Integer.*parseInt*(**port**));  
 **output** = **new** DataOutputStream((**socket**.getOutputStream()));  
  
 } **catch** (UnknownHostException e) {  
 e.printStackTrace();  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
 }  
}

|  |  |  |
| --- | --- | --- |
| **1** |  | **SQL Lite** |

□ **SQL Lite**

|  |
| --- |
| **1.** |
| SQLiteOpenHelper 클래스를 상속한 DBHelper 클래스를 생성  public class DBHelper extends SQLiteOpenHelper {  // DBHelper 생성자로 관리할 DB 이름과 버전 정보를 받음  public DBHelper(Context context, String name, SQLiteDatabase.CursorFactory factory, int version) {  super(context, name, factory, version);  }  // DB를 새로 생성할 때 호출되는 함수  @Override  public void onCreate(SQLiteDatabase db) {  // 새로운 테이블 생성  /\* 이름은 MONEYBOOK이고, 자동으로 값이 증가하는 \_id 정수형 기본키 컬럼과  item 문자열 컬럼, price 정수형 컬럼, create\_at 문자열 컬럼으로 구성된 테이블을 생성. \*/  db.execSQL("CREATE TABLE MONEYBOOK (\_id INTEGER PRIMARY KEY AUTOINCREMENT, item TEXT, price INTEGER, create\_at TEXT);");  }  // DB 업그레이드를 위해 버전이 변경될 때 호출되는 함수  @Override  public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  }  public void insert(String create\_at, String item, int price) {  // 읽고 쓰기가 가능하게 DB 열기  SQLiteDatabase db = getWritableDatabase();  // DB에 입력한 값으로 행 추가  db.execSQL("INSERT INTO MONEYBOOK VALUES(null, '" + item + "', " + price + ", '" + create\_at + "');");  db.close();  }  public void update(String item, int price) {  SQLiteDatabase db = getWritableDatabase();  // 입력한 항목과 일치하는 행의 가격 정보 수정  db.execSQL("UPDATE MONEYBOOK SET price=" + price + " WHERE item='" + item + "';");  db.close();  }  public void delete(String item) {  SQLiteDatabase db = getWritableDatabase();  // 입력한 항목과 일치하는 행 삭제  db.execSQL("DELETE FROM MONEYBOOK WHERE item='" + item + "';");  db.close();  }  public String getResult() {  // 읽기가 가능하게 DB 열기  SQLiteDatabase db = getReadableDatabase();  String result = "";  // DB에 있는 데이터를 쉽게 처리하기 위해 Cursor를 사용하여 테이블에 있는 모든 데이터 출력  Cursor cursor = db.rawQuery("SELECT \* FROM MONEYBOOK", null);  while (cursor.moveToNext()) {  result += cursor.getString(0)  + " : "  + cursor.getString(1)  + " | "  + cursor.getInt(2)  + "원 "  + cursor.getString(3)  + "\n";  }  return result;  }  }  [출처] Android (안드로이드) - SQLite 사용 방법 및 예제|작성자 워늬 |
| **2. java source** |
| *public* *class* MainActivity *extends* AppCompatActivity {  *@Override*  *protected* *void* onCreate(*Bundle* savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(*R*.layout.activity\_main);  *final* *DBHelper* dbHelper = new *DBHelper*(getApplicationContext(), "MoneyBook.db", **null**, **1**);  *// 테이블에 있는 모든 데이터 출력*  *final* *TextView* result = (*TextView*) findViewById(*R*.id.result);  *final* *EditText* etDate = (*EditText*) findViewById(*R*.id.date);  *final* *EditText* etItem = (*EditText*) findViewById(*R*.id.item);  *final* *EditText* etPrice = (*EditText*) findViewById(*R*.id.price);  *// 날짜는 현재 날짜로 고정*  *// 현재 시간 구하기*  *long* now = *System*.currentTimeMillis();  *Date* date = new *Date*(now);  *// 출력될 포맷 설정*  *SimpleDateFormat* simpleDateFormat = new *SimpleDateFormat*("yyyy년 MM월 dd일");  etDate.setText(simpleDateFormat.format(date));  *// DB에 데이터 추가*  *Button* insert = (*Button*) findViewById(*R*.id.insert);  insert.setOnClickListener(new *View*.*OnClickListener*() {  *@Override*  *public* *void* onClick(*View* v) {  *String* date = etDate.getText().toString();  *String* item = etItem.getText().toString();  *int* price = *Integer*.parseInt(etPrice.getText().toString());  dbHelper.insert(date, item, price);  result.setText(dbHelper.getResult());  }  });  *// DB에 있는 데이터 수정*  *Button* update = (*Button*) findViewById(*R*.id.update);  update.setOnClickListener(new *View*.*OnClickListener*() {  *@Override*  *public* *void* onClick(*View* v) {  *String* item = etItem.getText().toString();  *int* price = *Integer*.parseInt(etPrice.getText().toString());  dbHelper.update(item, price);  result.setText(dbHelper.getResult());  }  });  *// DB에 있는 데이터 삭제*  *Button* delete = (*Button*) findViewById(*R*.id.delete);  delete.setOnClickListener(new *View*.*OnClickListener*() {  *@Override*  *public* *void* onClick(*View* v) {  *String* item = etItem.getText().toString();  dbHelper.delete(item);  result.setText(dbHelper.getResult());  }  });  *// DB에 있는 데이터 조회*  *Button* select = (*Button*) findViewById(*R*.id.select);  select.setOnClickListener(new *View*.*OnClickListener*() {  *@Override*  *public* *void* onClick(*View* v) {  result.setText(dbHelper.getResult());  }  });  }  } |
| **3,manifest.xml** |
|  |
| **Activity Layout** |
| <?xml version="1.0" encoding="utf-8"?>  <*LinearLayout* xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"  android:padding="16dp">  <*LinearLayout*  android:layout\_width="match\_parent"  android:layout\_height="40dp"  android:gravity="center\_vertical">  <*TextView*  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="날짜" />  <*EditText*  android:id="@+id/date"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="16dp"  android:clickable="false"  android:focusable="false" />  </*LinearLayout*>  <*LinearLayout*  android:layout\_width="match\_parent"  android:layout\_height="40dp"  android:gravity="center\_vertical">  <*TextView*  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="항목" />  <*EditText*  android:id="@+id/item"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="16dp" />  </*LinearLayout*>  <*LinearLayout*  android:layout\_width="match\_parent"  android:layout\_height="40dp"  android:gravity="center\_vertical">  <*TextView*  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="가격" />  <*EditText*  android:id="@+id/price"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="16dp"  android:inputType="number" />  </*LinearLayout*>  <*LinearLayout*  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="8dp">  <*Button*  android:id="@+id/insert"  android:layout\_width="0dp"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:text="추가" />  <*Button*  android:id="@+id/delete"  android:layout\_width="0dp"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:text="삭제" />  <*Button*  android:id="@+id/update"  android:layout\_width="0dp"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:text="수정" />  <*Button*  android:id="@+id/select"  android:layout\_width="0dp"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:text="조회" />  </*LinearLayout*>  <*ScrollView*  android:layout\_marginTop="16dp"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent">  <*TextView*  android:id="@+id/result"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"/>  </*ScrollView*>  </*LinearLayout*> |
| **SQLITE** |
| package com.hanbit.test22;  import android.app.Activity;  import android.content.Context;  import android.database.Cursor;  import android.database.sqlite.SQLiteDatabase;  import android.database.sqlite.SQLiteOpenHelper;  import android.os.Bundle;  import android.view.View;  import android.view.View.OnClickListener;  import android.widget.Button;  import android.widget.EditText;  // SQLite  // 안드로이드 운영체제는 SQLite라이브러리를 사용함  // 용량이 작고 속도가 빠르다  // 단순한 파일 형태로 저장  // DB작업 때 openCreateDataBase, SQLiteOpenhelper 중 하나 사용  public class MainActivity extends Activity {  EditText edtName;  EditText edtPhone;  EditText edtAddr;  EditText edtResult;    Button  Button btnDelete;  Button btnSelect;  Button btnUpdate;    MyDBHelper helper;    @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);    edtName = (EditText)findViewById(R.id.edtName);  edtPhone = (EditText)findViewById(R.id.edtPhone);  edtAddr = (EditText)findViewById(R.id.edtAddr);  edtResult = (EditText)findViewById(R.id.edtResult);    btnAdd = (Button)findViewById(R.id.btnAdd);  btnDelete = (Button)findViewById(R.id.btnDel);  btnSelect = (Button)findViewById(R.id.btnSelect);  btnUpdate = (Button)findViewById(R.id.btnUpdate);    btnAdd.setOnClickListener(new OnClickListener() {  @Override  public void onClick(View arg0) {  // 방법1  // 데이터를 저장하기위해서 객체을 얻어옴  SQLiteDatabase db = helper.getWritableDatabase();  String sql ="insert into phone values(null,?,?,?)";    String args[] = {edtName.getText().toString(),  edtPhone.getText().toString(),  edtAddr.getText().toString()};  // execSQL(sql, 오브젝트배열);  db.execSQL(sql, args);  helper.close(); // 접속해제  edtResult.setText("Insert OK");      // 방법2  /\*SQLiteDatabase db2 = helper.getWritableDatabase();  String name = edtName.getText().toString();  String phone = edtPhone.getText().toString();  String addr = edtAddr.getText().toString();    // 추가된 정보를 담기  ContentValues values = new ContentValues();    values.put("name", name);  values.put("phone", phone);  values.put("addr", addr);    // db에 추가  db.insert("phone", null, values);  helper.close();  edtResult.setText("Insert OK");\*/    }  });  btnDelete.setOnClickListener(new OnClickListener() {  @Override  public void onClick(View arg0) {  SQLiteDatabase db = helper.getWritableDatabase();  String name = edtName.getText().toString();  String sql = "delete from phone where name=?";  db.execSQL(sql,new String[] {name});  edtResult.setText("Delete OK");  }  });  btnSelect.setOnClickListener(new OnClickListener() {  @Override  public void onClick(View arg0) {  SQLiteDatabase db = helper.getWritableDatabase();  String sql = "select \* from phone";  // select만 rawQuery()을 사용해야 됨  // delete, insert, update는 execSQl() 사용    Cursor cursor = db.rawQuery(sql, null);    // select로 조회된 데이터를 문자열로 연결할 객체 생성  StringBuffer sb = new StringBuffer();    while(cursor.moveToNext()){  int num = cursor.getInt(0);  String name = cursor.getString(1);  String phone = cursor.getString(2);  String addr = cursor.getString(3);    sb.append(num+" , "+name+" , "+phone+" , "+addr+"\r\n");  }  edtResult.setText(sb);  cursor.close();  helper.close();  }  });  btnUpdate.setOnClickListener(new OnClickListener() {    @Override  public void onClick(View arg0) {  SQLiteDatabase db = helper.getWritableDatabase();  String name = edtName.getText().toString();  String phone = edtPhone.getText().toString();  String addr = edtAddr.getText().toString();    String sql = "update phone set name =? , phone=?, addr=? where name =?";  String args[] ={name, phone,addr,name};    db.execSQL(sql,args);  edtResult.setText("update ok");  helper.close();  }  });  helper = new MyDBHelper(this);  }    // SQLiteOpenhelper 만들기  class MyDBHelper extends SQLiteOpenHelper{  public MyDBHelper(Context context) {  // super(컨텍스트, "데이터베이스이름",커스템 커서, 버전);  // 데이터베이스이름이 없으면 생성 onCrate호출  // 있으면 onCreate호출되지 않는다.  // db 이름이 같으나 버전이 다를때는 onUpgrade호출  super(context, "phone.db", null, 1);  }    @Override  public void onCreate(SQLiteDatabase db) {  // 테이블생성 sql  String sql = "create table phone(" +  "num integer primary key autoincrement, " +  "name text, phone text, addr text)";    // 명령실행  db.execSQL(sql);    sql = "insert into phone " +  "values(null,'jongmun','01097329110','seoul')";  // 명령실행  db.execSQL(sql);  }    // 기존 데이터베이가 존재하고 버전이 다를때 자동 호출된다.  @Override  public void onUpgrade(SQLiteDatabase db,  int oldVersion,  int newVersion) {  // db이름이 phone 이라면 삭제해 주세요  String sql = "drop table if exists phone";  db.execSQL(sql);    onCreate(db);    }  }  }  Sql manager down  NULL  INTEGER : 1,2,3,4,6,8bytes의 정수값  REAL : 8bytes의 부동소수점값  TEXT : UTF-8, UTF-16BE, UTE-16LE 인코딩의 문자열  BLOB : 입력된 그대로 저    <http://sqlitebrowser.org/> |

|  |  |  |
| --- | --- | --- |
| **1** |  | **Socket thread msghandler** |

□ **mainactivity.java**

|  |
| --- |
| **1.** |
| package com.example.can.myapplication;  import android.os.Bundle;  import android.app.Activity;  import android.os.Handler;  import android.os.Message;  import android.util.Log;  import android.view.View;  import android.widget.Button;  import android.widget.EditText;  import android.widget.TextView;  import java.io.DataInputStream;  import java.io.DataOutputStream;  import java.io.IOException;  import java.net.Socket;  import java.util.LinkedList;  Main thread : ui변경 가능함  sub thread : ui변경 불가능  handler : 다른 객체가 보낸 메시지 수신  서브 쓰레드가 보낸 메시지 수신하여 ui변경  thread.start()  sendEmptymessage() 숫자 전달 msg.what  sendMessage() msg.what,msg.arg1, msg.arg2, msg.obj  Handler(핸들러)  public void handleMessage(Message msg){  참조방법(자료형)  msg.what(int) FIFO(First in First Out) 자료처리 형태  msg.arg1(int)  msg.arg2(int)  msg.obj(Object)  }  Handler에서 메시지를 받으면 값을 확인후 ui변경관련된  코드 작성  public class MainActivity extends Activity {  TextView showText;  Button connectBtn;  Button Button\_send;  EditText ip\_EditText;  EditText port\_EditText;  EditText editText\_massage;  Handler msghandler;  SocketClient client; //서버접속을 위한 클라이언트 클래스  ReceiveThread receive; //  SendThread send;  Socket socket;  LinkedList<SocketClient> threadList;  /\*  //LinkedList<SocketClient>threadLista; //객체 생성시 데이터 저장 영역이 생기지 않으며, 서로 인접 데이터를 가리킨다.  //ArrayList 객체 생성과 함계 데이터를 저장할 수있는 영역이 존재함  //LinkedList저장 영역에 SocketClient ip port 가져와 영역생성  //Handler handler = new Handler() {  // public void handlerMessage(Message msg) {  // switch(msg.what) { 여러개의 메시지를 구분하기위한 메시지 핸들러  핸드러를 구분하기 위한 구분자  // case 0: // <--- 0 의 의미는?  // msg.obj 사용 객체 전달값을 가져오기위해서 사용  // UI 처리 코드  // break;  //  // case 1: // <-- 새로운 조건 추가될 때 마다 case 비교 구문 추가  // UI 처리 코드  ..  1.Handler 핸들러를 이용해서 쓰레드간 메시지 전달  2. client = new SocketClient ip,port를 이용ㅇ한 연결  class  2-1. SocketClient 접속과 동시에 receivethread를 동작  2-2. ReceiveThread  3. send = new SendThread(socket) 메시지 전송  \*/  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);// // 액티비티 초기화  setContentView(R.layout.activity\_main);// 액티비티의 레이아웃 설정  //안드로이드 view 소스코드 연동 레이아웃에 정의되어 있는 뷰  ip\_EditText = (EditText) findViewById(R.id.ip\_EditText);  port\_EditText = (EditText) findViewById(R.id.port\_EditText);  connectBtn = (Button) findViewById(R.id.connect\_Button);  showText = (TextView) findViewById(R.id.showText\_TextView);  editText\_massage = (EditText) findViewById(R.id.editText\_massage);  Button\_send = (Button) findViewById(R.id.Button\_send);  threadList = new LinkedList<MainActivity.SocketClient>();  ip\_EditText.setText("127.0.0.1");  port\_EditText.setText("5001");  // ReceiveThread를통해서 받은 메세지를 Handler로 MainThread에서 처리(외부Thread에서는 UI변경이불가)  msghandler = new Handler() {//메인에서 생성한 핸들러  @Override  public void handleMessage(Message hdmsg) {  if (hdmsg.what == 1111) {  showText.setText(hdmsg.obj.toString() + "\n");  }  }  };  // 연결버튼 클릭 이벤트  connectBtn.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View arg0) {  //Client 연결부  client = new SocketClient(ip\_EditText.getText().toString(),  port\_EditText.getText().toString());  threadList.add(client);  client.start();  }  });  //전송 버튼 클릭 이벤트  Button\_send.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View arg0) {  //SendThread 시작  if (editText\_massage.getText().toString() != null) {  send = new SendThread(socket);  send.start();  //시작후 edittext 초기화  editText\_massage.setText("");  }  }  });  }  //sleep메소드가 호출될경우 interruptedExcepton 발생  class SocketClient extends Thread {  boolean threadAlive;  String ip;  String port;  DataOutputStream output = null;  public SocketClient(String ip, String port) {  threadAlive = true; ///현재 쓰레드가 살아있으면  this.ip = ip;  this.port = port;  }  @Override  public void run() {  try {  // 연결후 바로 ReceiveThread 시작  socket = new Socket(ip, Integer.parseInt(port));  output = new DataOutputStream(socket.getOutputStream());  receive = new ReceiveThread(socket);  receive.start();  } catch (IOException e) {  e.printStackTrace();  }  }  }  class ReceiveThread extends Thread {  private Socket sock = null;  DataInputStream input;  public ReceiveThread(Socket socket) {  this.sock = socket;  try{  input = new DataInputStream(sock.getInputStream());  }catch(Exception e){  }  }  // 메세지 수신후 Handler로 전달  public void run() {  try {  while (input != null) {  String msg;  int count = input.available();  byte[] rcv = new byte[count]; // 문자열 크기만큼의 버퍼 생성  input.read(rcv);  msg = new String(rcv);  if (count > 0) {  Log.d(ACTIVITY\_SERVICE, "test :" +msg);  i는 Info(정보)  d는 Debug(디버그)  w는 Warning(경고)  e는 Error(오류)입니다.  현재까지 실행이 잘되었는지 확인하는 방법으로 프로그램 확인  Message hdmsg = msghandler.obtainMessage();  hdmsg.what = 1111;  hdmsg.obj = msg;  msghandler.sendMessage(hdmsg);  Log.d(ACTIVITY\_SERVICE,hdmsg.obj.toString());  }  }  } catch (IOException e) {  e.printStackTrace();  }  }  }  class SendThread extends Thread {  Socket socket;  String sendmsg = editText\_massage.getText().toString();  DataOutputStream output;  public SendThread(Socket socket) {  this.socket = socket;  try {  output = new DataOutputStream(socket.getOutputStream());  } catch (Exception e) {  }  }  public void run() {  try {  // 메세지 전송부  Log.d(ACTIVITY\_SERVICE, "11111");  if (output != null) {  if (sendmsg != null) {  output.write(sendmsg.getBytes());  }  }  } catch (IOException e) {  e.printStackTrace();  } catch (NullPointerException npe) {  npe.printStackTrace();  }  }  }  } |
| **2. java source** |
|  |
| **3,manifest.xml** |
|  |
| **Activity Layout** |
|  |
| **SQLITE** |
|  |

“임계값 알람”

Android

|  |  |  |
| --- | --- | --- |
| **1** |  | **임계값 알람** |

□ **mainactivity.java**

|  |
| --- |
| **1.** |
| package com.example.user.myapplication;  import android.support.v7.app.AppCompatActivity;  import android.os.Bundle;  import android.content.Context;  import android.os.Vibrator;  import android.os.Handler; //쓰레드 간에 메시지를 주고니 받거니 우리가 핸들러 쓰레드 데이터 주고  import android.os.Message;  import android.util.Log; //디버그 메시지  import android.view.View;//레이아웃 뷰  import android.widget.Button;  import android.widget.EditText;  import android.widget.TextView;  import java.io.DataInputStream; //전송된 바이트를 입력데이터를 버퍼에 넣은후 문자열 출력  import java.io.DataOutputStream; //입력된 내용을 버퍼에 저장후 바이트로 전송  import java.io.IOException;  import java.net.Socket;  import java.util.LinkedList;  public class MainActivity extends AppCompatActivity {  TextView showText;  Button connectBtn;  Button Button\_send;  EditText ip\_EditText;  EditText port\_EditText;  EditText editText\_massage;  Handler msghandler;  SocketClient client; //서버접속을 위한 클라이언트 클라스  ReceiveThread receive;//서버에서 보내온 데이터 안드로이드에서 보이게  SendThread send; //안드로이드에서 임의의 문자 보내는것  Socket socket;//네트워크  Vibrator mVib;  LinkedList<SocketClient> threadList;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);//액티비티 레이아웃 초기화  setContentView(R.layout.activity\_main);//액티비티 레이아웃 설정  //안드로이드 view 소스 코드 연동 레이아웃에 정의 되어있는 뷰  ip\_EditText = (EditText) findViewById(R.id.ip\_EditText);  port\_EditText = (EditText) findViewById(R.id.port\_EditText);  connectBtn = (Button) findViewById(R.id.connect\_Button);  showText = (TextView) findViewById(R.id.showText\_TextView);  editText\_massage = (EditText) findViewById(R.id.editText\_massage);  Button\_send = (Button) findViewById(R.id.Button\_send);  threadList = new LinkedList<MainActivity.SocketClient>();  mVib = (Vibrator) getSystemService(Context.VIBRATOR\_SERVICE);  ip\_EditText.setText("192.168.35.51"); // 접속할 서버 ip  port\_EditText.setText("5001");  // ReceiveThread를통해서 받은 메세지를 Handler로 MainThread에서 처리(외부Thread에서는 UI변경이불가)  msghandler = new Handler() {  @Override  public void handleMessage(Message hdmsg) { //생성자 클래스 정의한것처럼  if (hdmsg.what == 1111)  if(hdmsg.obj.toString().length() > 0)  {  String[] data =hdmsg.obj.toString().split(",");  int t = Integer.parseInt(data[0]);  int h = Integer.parseInt(data[1]);  String temp\_msg = "";  if(t < 10)  {temp\_msg = "나쁨";}  else if(t >= 20 && t < 30)  {temp\_msg = "보통";}  else if(t >= 30 && t < 35)  {temp\_msg = "중대경보";}  else if (t > 40)  {temp\_msg = "나쁨";}  showText.setText("온도 :" + data[0] + " " + temp\_msg + "\n습도 : " + data[1]);  }  }  };  // 연결버튼 클릭 이벤트  connectBtn.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View arg0) {  //Client 연결부  client = new SocketClient(ip\_EditText.getText().toString(),  port\_EditText.getText().toString());  threadList.add(client);  client.start();  }  });  //전송 버튼 클릭 이벤트  Button\_send.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View arg0) {  //SendThread 시작  if (editText\_massage.getText().toString() != null) {  send = new SendThread(socket);  send.start();  //시작후 edittext 초기화  editText\_massage.setText("");  }  }  });  }  class SocketClient extends Thread {  boolean threadAlive;  String ip;  String port;  DataOutputStream output = null;  public SocketClient(String ip, String port) {  threadAlive = true; //현재 쓰레드가 살아있으면 존제하면  this.ip = ip;  this.port = port;  }  @Override  public void run() {  try {  // 연결후 바로 ReceiveThread 시작  socket = new Socket(ip, Integer.parseInt(port)); //형변환 숫자  output = new DataOutputStream(socket.getOutputStream());  receive = new ReceiveThread(socket);  receive.start();  } catch (IOException e) {  e.printStackTrace();  }  }  }  class ReceiveThread extends Thread {  private Socket sock = null;  DataInputStream input;  public ReceiveThread(Socket socket) {  this.sock = socket;  try {  input = new DataInputStream(sock.getInputStream());  } catch (Exception e) {  }  }  // 메세지 수신후 Handler로 전달  public void run() {  try {  while (input != null) {  String msg;  int count = input.available();  byte[] rcv = new byte[count];  input.read(rcv);  msg = new String(rcv);  //Log.i info  //Log.d 디버그  //Log.w warnig  //Log.e 오류  if (count > 0) {  Log.d(ACTIVITY\_SERVICE, "test :" + msg);  Message hdmsg = msghandler.obtainMessage();  hdmsg.what = 1111;  hdmsg.obj = msg;  msghandler.sendMessage(hdmsg);  Log.d(ACTIVITY\_SERVICE, hdmsg.obj.toString());  }  }  } catch (IOException e) {  e.printStackTrace();  }  }  }  class SendThread extends Thread {  Socket socket;  String sendmsg = editText\_massage.getText().toString() + "\n";  DataOutputStream output;  public SendThread(Socket socket) {  this.socket = socket;  try {  output = new DataOutputStream(socket.getOutputStream());  } catch (Exception e) {  }  }  public void run() {  try {  // 메세지 전송부  Log.d(ACTIVITY\_SERVICE, "11111");  if (output != null) {  if (sendmsg != null) {  output.write(sendmsg.getBytes());  }  }  } catch (IOException e) {  e.printStackTrace();  } catch (NullPointerException npe) {  npe.printStackTrace();  }  }  }  } |
| **1. server** |
| package bluesky;  import java.net.\*;  import java.util.Random;  import java.io.\*;  public class server {  public static void main(String[] args) {    ServerSocket serversocket = null;  Socket socket = null;    try {  serversocket = new ServerSocket(5001);  socket = serversocket.accept();  System.out.println("연결 성공");  Thread thread2 = new receiverthread(socket);  thread2.start();  } catch (Exception e1) {  } finally {  try {  // System.out.println("연결 실패");  serversocket.close();  } catch (Exception e1) {    }  }  }  }  class receiverthread extends Thread {  Socket socket;  receiverthread(Socket socket) {  this.socket = socket;  }  public void run() {  //A A1 = new A();  try {  Random rand = new Random();  BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));  OutputStream out = socket.getOutputStream();      // String str1 = in.readLine();  //System.out.println(str1);    while (true) {  String str = in.readLine();  System.out.println(str);  // A1.insert(str);  if(str.equals("a")) {    int n = rand.nextInt(100);  String str3 = Integer.toString(n);  String a = str3;    out.write(a.getBytes());          out.write(a.getBytes());  }else if(str.equals("bSensor")) {    int p = rand.nextInt(100);  String str3 = Integer.toString(p);  String a = str3;  out.write(a.getBytes());  }else if(str.equals("cSensor")) {  int temperature= rand.nextInt(100);  int humidity = rand.nextInt(100);  String t\_str = Integer.toString(temperature);    String h\_str = Integer.toString(humidity);    String result = t\_str + "," + h\_str;  out.write(result.getBytes());    }  if (str == null)  break;  }  } catch (Exception e1) {  } finally {  try {  socket.close();  } catch (Exception e1) {  }  }  }  } |

“센서값”

Android

|  |  |  |
| --- | --- | --- |
| **1** |  | **Sensor Value** |

□  **Sensor**

|  |
| --- |
| **1. 동작과정** |
| private void AccSensorEventListener implements SensorEventListener {  public void onSensorChanged() {  ...  }  public void onAccuracyChanged() {  ...  }  }  SensorEventListener 오브젝트에는 onSensorChanged()와 onAccuracyChanged() 메소드를 구현해 줘야 한다.  센서 값이 바뀔때마다 리스너 오브젝트의 onSensorChanged() 메소드가 호출된다. |
| **1.** |
| *<?***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="com.example.root.myapplication3333.MainActivity"**>   <**LinearLayout  android:layout\_width="378dp"  android:layout\_height="505dp"  android:orientation="vertical"  tools:layout\_editor\_absoluteX="5dp"  tools:layout\_editor\_absoluteY="2dp"**>   <**TextView  android:id="@+id/acc\_x"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="TextView"** />   <**TextView  android:id="@+id/acc\_y"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="TextView"** />   <**TextView  android:id="@+id/acc\_z"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="TextView"** />   <**TextView  android:id="@+id/ori\_x"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="TextView"** />   <**TextView  android:id="@+id/ori\_y"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="TextView"** />   <**TextView  android:id="@+id/ori\_z"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="TextView"** />  </**LinearLayout**>  </**android.support.constraint.ConstraintLayout**> |
| **1.** |
| **package** com.example.root.myapplication3333;  **import** android.hardware.Sensor; **import** android.hardware.SensorEvent; **import** android.hardware.SensorEventListener; **import** android.hardware.SensorManager; **import** android.support.v7.app.AppCompatActivity; **import** android.os.Bundle; **import** android.util.Log; **import** android.widget.TextView;  **public class** MainActivity **extends** AppCompatActivity {  SensorManager **sm**;  SensorEventListener **accL**;  SensorEventListener **oriL**;  Sensor **oriSensor**;  Sensor **accSensor**;  TextView **ax**, **ay**, **az**;  TextView **ox**, **oy**, **oz**;   @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);   **sm** = (SensorManager)getSystemService(***SENSOR\_SERVICE***); *// SensorManager 인스턴스를 가져옴* **oriSensor** = **sm**.getDefaultSensor(Sensor.***TYPE\_ORIENTATION***); *// 방향 센서* **accSensor** = **sm**.getDefaultSensor(Sensor.***TYPE\_ACCELEROMETER***); *// 가속도 센서* **oriL** = **new** oriListener(); *// 방향 센서 리스너 인스턴스* **accL** = **new** accListener(); *// 가속도 센서 리스너 인스턴스* **ax** = (TextView)findViewById(R.id.***acc\_x***);  **ay** = (TextView)findViewById(R.id.***acc\_y***);  **az** = (TextView)findViewById(R.id.***acc\_z***);  **ox** = (TextView)findViewById(R.id.ori\_x);  **oy** = (TextView)findViewById(R.id.***ori\_y***);  **oz** = (TextView)findViewById(R.id.***ori\_z***);   } @Override  **public void** onResume() {*//자가 일시 중지된 상태에서 액티비티로 돌아오면 시스템은 액티비티를 재개하고 onResume() 메서드를 호출합니다.* **super**.onResume();   **sm**.registerListener(**accL**, **accSensor**, SensorManager.***SENSOR\_DELAY\_NORMAL***); *// 가속도 센서 리스너 오브젝트를 등록* **sm**.registerListener(**oriL**, **oriSensor**, SensorManager.***SENSOR\_DELAY\_NORMAL***); *// 방향 센서 리스너 오브젝트를 등록* }   @Override  **public void** onPause() { *//액티비티가 일시 중지했을때 호출* **super**.onPause();   **sm**.unregisterListener(**oriL**); *// unregister acceleration listener* **sm**.unregisterListener(**accL**); *// unregister orientation listener* }    **private class** accListener **implements** SensorEventListener {  **public void** onSensorChanged(SensorEvent event) { *// 가속도 센서 값이 바뀔때마다 호출됨* **ax**.setText(Float.*toString*(event.**values**[0]));  **ay**.setText(Float.*toString*(event.**values**[1]));  **az**.setText(Float.*toString*(event.**values**[2]));  Log.*i*(**"SENSOR"**, **"Acceleration changed."**);  Log.*i*(**"SENSOR"**, **" Acceleration X: "** + event.**values**[0]  + **", Acceleration Y: "** + event.**values**[1]  + **", Acceleration Z: "** + event.**values**[2]);  }   **public void** onAccuracyChanged(Sensor sensor, **int** accuracy) {  }  }   **private class** oriListener **implements** SensorEventListener {  **public void** onSensorChanged(SensorEvent event) { *// 방향 센서 값이 바뀔때마다 호출됨* **ox**.setText(Float.*toString*(event.**values**[0]));  **oy**.setText(Float.*toString*(event.**values**[1]));  **oz**.setText(Float.*toString*(event.**values**[2]));  Log.*i*(**"SENSOR"**, **"Orientation changed."**);  Log.*i*(**"SENSOR"**, **" Orientation X: "** + event.**values**[0]  + **", Orientation Y: "** + event.**values**[1]  + **", Orientation Z: "** + event.**values**[2]);  }   **public void** onAccuracyChanged(Sensor sensor, **int** accuracy) {   }  } } |

|  |  |  |
| --- | --- | --- |
| **1** |  | **가속도 걸음 재기** |

□  **Sensor**

|  |
| --- |
| **1. 동작과정** |
| **package** com.example.root.myapplication3333;  **import** android.hardware.Sensor; **import** android.hardware.SensorEvent; **import** android.hardware.SensorEventListener; **import** android.hardware.SensorManager; **import** android.support.v7.app.AppCompatActivity; **import** android.os.Bundle; **import** android.util.Log; **import** android.widget.TextView;  **public class** MainActivity **extends** AppCompatActivity {  SensorManager **sm**;  SensorEventListener **accL**;  SensorEventListener **oriL**;  Sensor **oriSensor**;  Sensor **accSensor**;  TextView **ax**, **ay**, **az**;  TextView **ox**, **oy**, **oz**;  TextView **mSteps**;  TextView **mDistance**;  TextView **mCal**;  **private long lastTime**;  **private float speed**;   **private float lastX**;  **private float lastY**;  **private float lastZ**;     **private float x**, **y**, **z**;  **private static final int *SHAKE\_THRESHOLD*** = 800;    @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);   **sm** = (SensorManager)getSystemService(***SENSOR\_SERVICE***); *// SensorManager 인스턴스를 가져옴* **oriSensor** = **sm**.getDefaultSensor(Sensor.***TYPE\_ORIENTATION***); *// 방향 센서* **accSensor** = **sm**.getDefaultSensor(Sensor.***TYPE\_ACCELEROMETER***); *// 가속도 센서* **oriL** = **new** oriListener(); *// 방향 센서 리스너 인스턴스* **accL** = **new** accListener(); *// 가속도 센서 리스너 인스턴스* **ax** = (TextView)findViewById(R.id.***acc\_x***);  **ay** = (TextView)findViewById(R.id.***acc\_y***);  **az** = (TextView)findViewById(R.id.***acc\_z***);  **ox** = (TextView)findViewById(R.id.***ori\_x***);  **oy** = (TextView)findViewById(R.id.***ori\_y***);  **oz** = (TextView)findViewById(R.id.***ori\_z***);   **mSteps** = (TextView)findViewById(R.id.***mSteps***);  **mDistance** = (TextView)findViewById(R.id.***mDistance***);  **mCal** = (TextView)findViewById(R.id.***mCal***);   } @Override  **public void** onResume() {*//자가 일시 중지된 상태에서 액티비티로 돌아오면 시스템은 액티비티를 재개하고 onResume() 메서드를 호출합니다.* **super**.onResume();   **sm**.registerListener(**accL**, **accSensor**, SensorManager.***SENSOR\_DELAY\_NORMAL***); *// 가속도 센서 리스너 오브젝트를 등록* **sm**.registerListener(**oriL**, **oriSensor**, SensorManager.***SENSOR\_DELAY\_NORMAL***); *// 방향 센서 리스너 오브젝트를 등록* }   @Override  **public void** onPause() { *//액티비티가 일시 중지했을때 호출* **super**.onPause();   **sm**.unregisterListener(**oriL**); *// unregister acceleration listener* **sm**.unregisterListener(**accL**); *// unregister orientation listener* }   **private class** accListener **implements** SensorEventListener {  **public void** onSensorChanged(SensorEvent event) { *// 가속도 센서 값이 바뀔때마다 호출됨* **long** currentTime = System.*currentTimeMillis*();  **long** gabOfTime = (currentTime - **lastTime**);  **int** count=0;  **long** cal;  String str;  String str3 = String.*format*(**"%d"**, currentTime) + **" "** +String.*format*(**"%d"**, **lastTime**) ;  **mDistance**.setText(str3);   **if** (gabOfTime > 100) {  **lastTime** = currentTime;  **x** = event.**values**[0];  **y** = event.**values**[1];  **z** = event.**values**[2];   **speed** = Math.*abs*(**x** + **y** + **z** - **lastX** - **lastY** - **lastZ**) / gabOfTime \* 10000;   **if** (**speed** > ***SHAKE\_THRESHOLD***) {  *// 이벤트발생!!* count++;  str = String.*format*(**"%d"**,count);  **mSteps**.setText(str+**"걸음"**); *// cal=5.5\*70\*count/1000; // mCal.setText(cal+"Kcal");* }   **lastX** = event.**values**[0];  **lastY** = event.**values**[1];  **lastZ** = event.**values**[2];  }   **ax**.setText(Float.*toString*(event.**values**[0]));  **ay**.setText(Float.*toString*(event.**values**[1]));  **az**.setText(Float.*toString*(event.**values**[2]));  Log.*i*(**"SENSOR"**, **"Acceleration changed."**);  Log.*i*(**"SENSOR"**, **" Acceleration X: "** + event.**values**[0]  + **", Acceleration Y: "** + event.**values**[1]  + **", Acceleration Z: "** + event.**values**[2]);  }   **public void** onAccuracyChanged(Sensor sensor, **int** accuracy) {  }  }   **private class** oriListener **implements** SensorEventListener {  **public void** onSensorChanged(SensorEvent event) { *// 방향 센서 값이 바뀔때마다 호출됨* **ox**.setText(Float.*toString*(event.**values**[0]));  **oy**.setText(Float.*toString*(event.**values**[1]));  **oz**.setText(Float.*toString*(event.**values**[2]));  Log.*i*(**"SENSOR"**, **"Orientation changed."**);  Log.*i*(**"SENSOR"**, **" Orientation X: "** + event.**values**[0]  + **", Orientation Y: "** + event.**values**[1]  + **", Orientation Z: "** + event.**values**[2]);  }   **public void** onAccuracyChanged(Sensor sensor, **int** accuracy) {   }  } } |
| **1.** |
| *<?***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="com.example.root.myapplication3333.MainActivity"**>   <**LinearLayout  android:layout\_width="375dp"  android:layout\_height="507dp"  android:orientation="vertical"  tools:layout\_editor\_absoluteX="5dp"  tools:layout\_editor\_absoluteY="2dp"**>  <**TextView  android:id="@+id/textView2"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="가속도 센서"  android:textSize="24sp"** />   <**TextView  android:id="@+id/acc\_z"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="TextView"  android:textSize="24sp"** />   <**TextView  android:id="@+id/acc\_x"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="TextView"  android:textSize="24sp"** />   <**TextView  android:id="@+id/acc\_y"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="TextView"  android:textSize="24sp"** />   <**TextView  android:id="@+id/textView5"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="자이로 센서"  android:textSize="24sp"** />   <**TextView  android:id="@+id/ori\_x"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="TextView"  android:textSize="24sp"** />   <**TextView  android:id="@+id/ori\_y"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="TextView"  android:textSize="24sp"** />   <**TextView  android:id="@+id/ori\_z"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="TextView"  android:textSize="24sp"** />   <**TextView  android:id="@+id/mDistance"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="TextView"  android:textSize="24sp"** />   <**TextView  android:id="@+id/mSteps"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="TextView"  android:textSize="24sp"** />   <**TextView  android:id="@+id/mCal"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="TextView"  android:textSize="24sp"** />   </**LinearLayout**>  </**android.support.constraint.ConstraintLayout**> |
| **1.** |
|  |

“GPS1”

Android

|  |  |  |
| --- | --- | --- |
| **1** |  | **Sensor Value** |

□  **Sensor**

|  |
| --- |
| **1. MAINACTIVITY** |
| **package** com.example.root.myapplicationgps;  **import** android.support.v7.app.AppCompatActivity; **import** android.os.Bundle; **import** android.view.View; **import** android.widget.Button; **import** android.widget.Toast;  **public class** MainActivity **extends** AppCompatActivity {  Button **btnShowLocation**;    GPSTracker **gps**;   @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);   **btnShowLocation** = (Button) findViewById(R.id.***btnShowLocation***);   **btnShowLocation**.setOnClickListener(**new** View.OnClickListener() {   @Override  **public void** onClick(View arg0) {   **gps** = **new** GPSTracker(MainActivity.**this**);    **if** (**gps**.canGetLocation()) {   **double** latitude = **gps**.getLatitude();  **double** longitude = **gps**.getLongitude();    Toast.*makeText*(getApplicationContext(), **"Your Location is - \nLat: "** + latitude + **"\nLong: "** + longitude, Toast.***LENGTH\_LONG***).show();  } **else** {    **gps**.showSettingsAlert();  }   }  });   } } |
| **1. GPSTRACKET.JAVA** |
| **package** com.example.root.myapplicationgps;  **import** android.Manifest; **import** android.content.Context; **import** android.content.DialogInterface; **import** android.content.Intent; **import** android.content.pm.PackageManager; **import** android.location.Location; **import** android.location.LocationListener; **import** android.location.LocationManager; **import** android.app.Service; **import** android.os.Bundle; **import** android.os.IBinder; **import** android.provider.Settings; **import** android.support.v7.app.AlertDialog; **import** android.util.Log; **import** android.widget.Toast;   */\*\*  \* Created by Duke on 9/7/2015.  \*/* **public class** GPSTracker **extends** Service **implements** LocationListener {  **private final** Context **mContext**;    **boolean isGPSEnabled** = **false**;    **boolean isNetworkEnabled** = **false**;    **boolean canGetLocation** = **false**;   Location **location**;  **double latitude**;  **double longitude**;    **private static final long *MIN\_DISTANCE\_CHANGE\_FOR\_UPDATES*** = 10;    **private static final long *MIN\_TIME\_BW\_UPDATES*** = 1000 \* 60 \* 1;    **protected** LocationManager **locationManager**;   **public** GPSTracker(Context context) {  **this**.**mContext** = context;  getLocation();  }   **public** Location getLocation() {  **try** {  **locationManager** = (LocationManager) **mContext** .getSystemService(***LOCATION\_SERVICE***);    **isGPSEnabled** = **locationManager** .isProviderEnabled(LocationManager.***GPS\_PROVIDER***);    **isNetworkEnabled** = **locationManager** .isProviderEnabled(LocationManager.***NETWORK\_PROVIDER***);   **if** (!**isGPSEnabled** && !**isNetworkEnabled**) {   } **else** {  **this**.**canGetLocation** = **true**;   **if** (**isNetworkEnabled**) {  **locationManager**.requestLocationUpdates(LocationManager.***NETWORK\_PROVIDER***, ***MIN\_TIME\_BW\_UPDATES***, ***MIN\_DISTANCE\_CHANGE\_FOR\_UPDATES***, **this**);  *// Log.d("Network", "Network");* **if** (**locationManager** != **null**) {  **location** = **locationManager** .getLastKnownLocation(LocationManager.***NETWORK\_PROVIDER***);  **if** (**location** != **null**) {  **latitude** = **location**.getLatitude();  **longitude** = **location**.getLongitude();  }  }  }   **if** (**isGPSEnabled**) {  **if** (**location** == **null**) {  **locationManager**.requestLocationUpdates(  LocationManager.***GPS\_PROVIDER***,  ***MIN\_TIME\_BW\_UPDATES***,  ***MIN\_DISTANCE\_CHANGE\_FOR\_UPDATES***, **this**);  Log.*d*(**"GPS Enabled"**, **"GPS Enabled"**);  **if** (**locationManager** != **null**) {  **location** = **locationManager** .getLastKnownLocation(LocationManager.***GPS\_PROVIDER***);  **if** (**location** != **null**) {  **latitude** = **location**.getLatitude();  **longitude** = **location**.getLongitude();  }  }  }  }  }   } **catch** (Exception e) {  e.printStackTrace();  }   **return location**;  }    **public void** stopUsingGPS() {  **if** (**locationManager** != **null**) {  **if** (checkSelfPermission(Manifest.permission.***ACCESS\_FINE\_LOCATION***) != PackageManager.***PERMISSION\_GRANTED*** && checkSelfPermission(Manifest.permission.***ACCESS\_COARSE\_LOCATION***) != PackageManager.***PERMISSION\_GRANTED***) {  *//* ***TODO: Consider calling*** *// public void requestPermissions(@NonNull String[] permissions, int requestCode)  // here to request the missing permissions, and then overriding  // public void onRequestPermissionsResult(int requestCode, String[] permissions,  // int[] grantResults)  // to handle the case where the user grants the permission. See the documentation  // for Activity#requestPermissions for more details.* **return**;  }  **locationManager**.removeUpdates(GPSTracker.**this**);  }  }    **public double** getLatitude() {  **if** (**location** != **null**) {  **latitude** = **location**.getLatitude();  }   *// return latitude* **return latitude**;  }    **public double** getLongitude() {  **if** (**location** != **null**) {  **longitude** = **location**.getLongitude();  }    **return longitude**;  }    **public boolean** canGetLocation() {  **return this**.**canGetLocation**;  }   **public void** showSettingsAlert() {  AlertDialog.Builder alertDialog = **new** AlertDialog.Builder(**mContext**);    alertDialog.setTitle(**"Warning"**);    alertDialog.setMessage(**"Please enabled GPS from settings"**);    alertDialog.setPositiveButton(**"Settings"**, **new** DialogInterface.OnClickListener() {  **public void** onClick(DialogInterface dialog, **int** which) {  Intent intent = **new** Intent(Settings.***ACTION\_LOCATION\_SOURCE\_SETTINGS***);  **mContext**.startActivity(intent);  }  });    alertDialog.setNegativeButton(**"Cancel"**, **new** DialogInterface.OnClickListener() {  **public void** onClick(DialogInterface dialog, **int** which) {  dialog.cancel();  }  });    alertDialog.show();  }   @Override  **public void** onLocationChanged(Location location) {   **double** lat = location.getLatitude();  **double** longi = location.getLongitude();  Toast.*makeText*(getApplicationContext(), **"My Location is \n"** + lat + **"\n"** + longi, Toast.***LENGTH\_SHORT***);  getLocation();   }   @Override  **public void** onProviderDisabled(String provider) {  }   @Override  **public void** onProviderEnabled(String provider) {  }   @Override  **public void** onStatusChanged(String provider, **int** status, Bundle extras) {  }   @Override  **public** IBinder onBind(Intent arg0) {  **return null**;  }  } |
| **1. 동작과정** |
|  |
| **1. 동작과정** |
|  |

|  |
| --- |
| **1. 동작과정** |
| <**uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION"**/> |
| **1. 동작과정** |
| *<?***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="com.example.root.myapplicationgps.MainActivity"**>   <**LinearLayout  android:layout\_width="368dp"  android:layout\_height="495dp"  android:orientation="vertical"  tools:layout\_editor\_absoluteX="8dp"  tools:layout\_editor\_absoluteY="8dp"**>   <**Button  android:id="@+id/btnShowLocation"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="Button"** />  </**LinearLayout**> </**android.support.constraint.ConstraintLayout**> |

“GPS2”

Android

|  |  |  |
| --- | --- | --- |
| **1** |  | **Sensor Value** |

□  **Sensor**

|  |
| --- |
| **1. MAINACTIVITY** |
| <**uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION"**/> |
| **1. MAINACTIVITY.JAVA** |
| **package** com.example.root.myapplicationgps3333;  **import** android.content.Context; **import** android.location.Location; **import** android.location.LocationListener; **import** android.location.LocationManager; **import** android.support.v7.app.AppCompatActivity; **import** android.os.Bundle; **import** android.util.Log; **import** android.view.View; **import** android.widget.TextView; **import** android.widget.ToggleButton;  **public class** MainActivity **extends** AppCompatActivity {  TextView **tv**;  ToggleButton **tb**;    @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  **tv** = (TextView) findViewById(R.id.***textView2***);  **tv**.setText(**"위치정보 미수신중"**);   **tb** = (ToggleButton)findViewById(R.id.***toggle1***);   *// LocationManager 객체를 얻어온다* **final** LocationManager lm = (LocationManager) getSystemService(Context.***LOCATION\_SERVICE***);    **tb**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  **try**{  **if**(**tb**.isChecked()){  **tv**.setText(**"수신중.."**);  *// GPS 제공자의 정보가 바뀌면 콜백하도록 리스너 등록하기~!!!* lm.requestLocationUpdates(LocationManager.***GPS\_PROVIDER***, *// 등록할 위치제공자* 100, *// 통지사이의 최소 시간간격 (miliSecond)* 1, *// 통지사이의 최소 변경거리 (m)* **mLocationListener**);  lm.requestLocationUpdates(LocationManager.***NETWORK\_PROVIDER***, *// 등록할 위치제공자* 100, *// 통지사이의 최소 시간간격 (miliSecond)* 1, *// 통지사이의 최소 변경거리 (m)* **mLocationListener**);  }**else**{  **tv**.setText(**"위치정보 미수신중"**);  lm.removeUpdates(**mLocationListener**); *// 미수신할때는 반드시 자원해체를 해주어야 한다.* }  }**catch**(SecurityException ex){  }  }  });  } *// end of onCreate* **private final** LocationListener **mLocationListener** = **new** LocationListener() {  **public void** onLocationChanged(Location location) {  *//여기서 위치값이 갱신되면 이벤트가 발생한다.  //값은 Location 형태로 리턴되며 좌표 출력 방법은 다음과 같다.* Log.*d*(**"test"**, **"onLocationChanged, location:"** + location);  **double** longitude = location.getLongitude(); *//경도* **double** latitude = location.getLatitude(); *//위도* **double** altitude = location.getAltitude(); *//고도* **float** accuracy = location.getAccuracy(); *//정확도* String provider = location.getProvider(); *//위치제공자* **tv**.setText(**"위치정보 : "** + provider + **"\n위도 : "** + longitude + **"\n경도 : "** + latitude  + **"\n고도 : "** + altitude + **"\n정확도 : "** + accuracy);  }  **public void** onProviderDisabled(String provider) {  *// Disabled시* Log.*d*(**"test"**, **"onProviderDisabled, provider:"** + provider);  }   **public void** onProviderEnabled(String provider) {  *// Enabled시* Log.*d*(**"test"**, **"onProviderEnabled, provider:"** + provider);  }   **public void** onStatusChanged(String provider, **int** status, Bundle extras) {  *// 변경시* Log.*d*(**"test"**, **"onStatusChanged, provider:"** + provider + **", status:"** + status + **" ,Bundle:"** + extras);  }  }; } *// end of class* |
| **1. 동작과정** |
| *<?***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="com.example.root.myapplicationgps3333.MainActivity"**>   <**LinearLayout  android:layout\_width="368dp"  android:layout\_height="495dp"  android:orientation="vertical"  tools:layout\_editor\_absoluteX="8dp"  tools:layout\_editor\_absoluteY="8dp"**>   <**TextView  android:id="@+id/textView1"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="GPS 에 수신된 현재위치"  android:textAppearance="?android:attr/textAppearanceLarge"** />   <**TextView  android:id="@+id/textView2"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="결과창"  android:textAppearance="?android:attr/textAppearanceLarge"** />   <**ToggleButton  android:id="@+id/toggle1"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:textOff="위치정보수신시작"  android:textOn="위치정보수신종료"**/>    </**LinearLayout**> </**android.support.constraint.ConstraintLayout**> |
| **1. 동작과정** |
|  |

“Gyro 넘어짐 Android”

Android

|  |  |  |
| --- | --- | --- |
| **1** |  | **Sensor Value** |

□  **Sensor**

|  |
| --- |
| **1. MAINACTIVITY** |
| <**uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION"**/> |
| **1. MAINACTIVITY.JAVA** |
| package com.example.naruhodo627.myapplication;  import android.content.Context;  import android.hardware.Sensor;  import android.hardware.SensorEvent;  import android.hardware.SensorEventListener;  import android.hardware.SensorManager;  import android.os.Vibrator;  import android.support.v7.app.AppCompatActivity;  import android.os.Bundle;  import android.util.Log;  import android.view.View;  import android.widget.Button;  import android.widget.EditText;  import android.widget.TextView;  import android.widget.Toast;  import java.util.ArrayList;  import java.util.Iterator;  public class MainActivity extends AppCompatActivity {  SensorManager sm;  SensorEventListener accL; // 자이로  SensorEventListener oriL; // 가속도  Sensor oriSensor;  Sensor accSensor;  TextView ax, ay, az; // 자이로  TextView ox, oy, oz; // 가속도  TextView mDistance;  TextView mSteps;  EditText editShakeHold;  Button btnReset;  ArrayList<Float> SensorX = new ArrayList<>();  ArrayList<Float> SensorY = new ArrayList<>();  ArrayList<Float> SensorZ = new ArrayList<>();  private long lastTime = 0;  private float speed = 0;  private float lastX = 0;  private float lastY = 0;  private float lastZ = 0;  private float x, y, z;  private int SHAKE\_THRESHOLD = 800;  private int StepCount = 0;  Vibrator mVib;  @Override  protected void onCreate(Bundle savedInstanceState) { // 최초 생성시  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);  sm = (SensorManager)getSystemService(SENSOR\_SERVICE);  oriSensor = sm.getDefaultSensor(Sensor.TYPE\_ORIENTATION);  accSensor = sm.getDefaultSensor(Sensor.TYPE\_ACCELEROMETER);  oriL = new oriListener();  accL = new accListener();  ax = (TextView)findViewById(R.id.acc\_x);  ay = (TextView)findViewById(R.id.acc\_y);  az = (TextView)findViewById(R.id.acc\_z);  ox = (TextView)findViewById(R.id.ori\_x);  oy = (TextView)findViewById(R.id.ori\_y);  oz = (TextView)findViewById(R.id.ori\_z);  mSteps = (TextView)findViewById(R.id.mSteps);  mDistance = (TextView)findViewById(R.id.mDistance);  editShakeHold = (EditText)findViewById(R.id.editShakeHold);  btnReset = (Button)findViewById(R.id.btnReset);  mVib = (Vibrator) getSystemService(Context.VIBRATOR\_SERVICE);  btnReset.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View arg0) {  //Client 연결부  String str;  StepCount = 0;  str = String.format("%d", StepCount);  mSteps.setText(str+"걸음");  }  });  } @Override  public void onResume() { // 일시 중지된 상태에서 액티비티로 다시 onResume() 실행  super.onResume();  sm.registerListener(accL, accSensor, SensorManager.SENSOR\_DELAY\_NORMAL);  sm.registerListener(oriL, oriSensor, SensorManager.SENSOR\_DELAY\_NORMAL);  }  @Override  public void onPause() { // 액티비티가 일시 중지했을 때 호출  super.onPause();  sm.unregisterListener(oriL);  sm.unregisterListener(accL);  }  private class accListener implements SensorEventListener { // 자이로 센서 값이 바뀔떄마다 호출해주는곳  public void onSensorChanged(SensorEvent event) {  ax.setText(Float.toString(event.values[0]));  ay.setText(Float.toString(event.values[1]));  az.setText(Float.toString(event.values[2]));  Log.i("SENSOR", "Acceleration changed.");  Log.i("SENSOR", " Acceleration X: " + event.values[0]  + ", Acceleration Y: " + event.values[1]  + ", Acceleration Z: " + event.values[2]);  try{  SHAKE\_THRESHOLD = Integer.parseInt(editShakeHold.getText().toString());  }  catch (Exception e)  {  return;  }  long currentTime = System.currentTimeMillis();  long gabOfTime = (currentTime - lastTime);  int count;  long cal;  String str;  String str3 = String.format("%d", currentTime) + " " + String.format("%d", lastTime);  mDistance.setText(str3);  if(gabOfTime > 100) {  lastTime = currentTime;  x = event.values[0];  y = event.values[1];  z = event.values[2];  speed = Math.abs(x+y+z - lastX - lastY - lastZ) / gabOfTime \* 10000;  if(speed > SHAKE\_THRESHOLD){  StepCount++;  str = String.format("%d", StepCount);  mSteps.setText(str+"걸음");  }  lastX = event.values[0];  lastY = event.values[1];  lastZ = event.values[2];  }  if(SensorX.isEmpty())  {  SensorX.add(event.values[0]);  SensorY.add(event.values[0]);  SensorZ.add(event.values[0]);  return;  }  int averX = 0;  int averY = 0;  int averZ = 0;  Iterator<Float> iter;  iter = SensorX.iterator();  count = 0;  while (iter.hasNext())  {  averX += iter.next();  count++;  }  averX = averX / count;  iter = SensorY.iterator();  count = 0;  while (iter.hasNext())  {  averY += iter.next();  count++;  }  averY = averY / count;  iter = SensorZ.iterator();  count = 0;  while (iter.hasNext())  {  averZ += iter.next();  count++;  }  averZ = averZ / count;  if(5 < Math.abs(event.values[0] - averX) && 5 < Math.abs(event.values[1] - averY) && 10 < Math.abs(event.values[2] - averZ))  {  mVib.vibrate(1000);  Toast.makeText(MainActivity.this, "넘어짐", Toast.LENGTH\_SHORT).show();  }  SensorX.add(event.values[0]);  SensorY.add(event.values[1]);  SensorZ.add(event.values[2]);  if(SensorX.size() > 50)  SensorX.remove(0);  if(SensorY.size() > 50)  SensorY.remove(0);  if(SensorZ.size() > 50)  SensorZ.remove(0);  // if(event.values[0] > 7)  // Toast.makeText(MainActivity.this, "UP", Toast.LENGTH\_SHORT).show();  }  public void onAccuracyChanged(Sensor sensor, int accuracy) {  }  }  private class oriListener implements SensorEventListener { // 가속도 센서 값이 바뀔때마다 호출해주는곳  public void onSensorChanged(SensorEvent event) {  ox.setText(Float.toString(event.values[0]));  oy.setText(Float.toString(event.values[1]));  oz.setText(Float.toString(event.values[2]));  Log.i("SENSOR", "Orientation changed.");  Log.i("SENSOR", " Orientation X: " + event.values[0]  + ", Orientation Y: " + event.values[1]  + ", Orientation Z: " + event.values[2]);  }  public void onAccuracyChanged(Sensor sensor, int accuracy) {  }  }  } |
| **1. MAINACTIVITY.JAVA** |
| <?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="com.example.naruhodo627.myapplication.MainActivity">  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical">  <TextView  android:id="@+id/gyro\_x"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="자이로 x" />  <TextView  android:id="@+id/acc\_x"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="TextView" />  <TextView  android:id="@+id/gyro\_y"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="자이로 y" />  <TextView  android:id="@+id/acc\_y"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="TextView" />  <TextView  android:id="@+id/gyro\_z"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="자이로 z" />  <TextView  android:id="@+id/acc\_z"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="TextView" />  <TextView  android:id="@+id/accr\_x"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="가속도 x" />  <TextView  android:id="@+id/ori\_x"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="TextView" />  <TextView  android:id="@+id/accr\_y"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="가속도 y" />  <TextView  android:id="@+id/ori\_y"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="TextView" />  <TextView  android:id="@+id/accr\_z"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="가속도 z" />  <TextView  android:id="@+id/ori\_z"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="TextView" />  <TextView  android:id="@+id/mSteps"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="걸음" />  <TextView  android:id="@+id/mDistance"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="거리" />  <EditText  android:id="@+id/editShakeHold"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:ems="10"  android:inputType="textPersonName"  android:text="800" />  <Button  android:id="@+id/btnReset"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="걸음초기화" />  </LinearLayout>  </android.support.constraint.ConstraintLayout> |
| **1. MAINACTIVITY.JAVA** |
| <?xml version="1.0" encoding="utf-8"?>  <manifest xmlns:android="http://schemas.android.com/apk/res/android"  package="com.example.naruhodo627.myapplication">  <uses-permission android:name="android.permission.VIBRATE" />  <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=".MainActivity">  <intent-filter>  <action android:name="android.intent.action.MAIN" />  <category android:name="android.intent.category.LAUNCHER" />  </intent-filter>  </activity>  </application>  </manifest> |

“Gyro 넘어짐 Android java server 연동”

Android

|  |  |  |
| --- | --- | --- |
| **1** |  | **Sensor Value** |

□  **Sensor**

|  |
| --- |
| **1. MAINACTIVITY** |
| <?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="com.example.naruhodo627.myapplication.MainActivity">  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical">  <TextView  android:id="@+id/gyro\_x"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="자이로 x" />  <TextView  android:id="@+id/acc\_x"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="TextView" />  <TextView  android:id="@+id/gyro\_y"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="자이로 y" />  <TextView  android:id="@+id/acc\_y"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="TextView" />  <TextView  android:id="@+id/gyro\_z"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="자이로 z" />  <TextView  android:id="@+id/acc\_z"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="TextView" />  <TextView  android:id="@+id/accr\_x"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="가속도 x" />  <TextView  android:id="@+id/ori\_x"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="TextView" />  <TextView  android:id="@+id/accr\_y"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="가속도 y" />  <TextView  android:id="@+id/ori\_y"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="TextView" />  <TextView  android:id="@+id/accr\_z"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="가속도 z" />  <TextView  android:id="@+id/ori\_z"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="TextView" />  <TextView  android:id="@+id/mSteps"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="걸음" />  <TextView  android:id="@+id/mDistance"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="거리" />  <EditText  android:id="@+id/editShakeHold"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:ems="10"  android:inputType="textPersonName"  android:text="800" />  <Button  android:id="@+id/btnReset"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="걸음초기화" />  <EditText  android:id="@+id/ip\_EditText"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:hint="IP주소를입력하세요" />  <EditText  android:id="@+id/port\_EditText"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:hint="PORT주소를입력하세요" />  <Button  android:id="@+id/connect\_Button"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="연결" />  </LinearLayout>  </android.support.constraint.ConstraintLayout> |
| **1. MAINACTIVITY.JAVA** |
| <?xml version="1.0" encoding="utf-8"?>  <manifest xmlns:android="http://schemas.android.com/apk/res/android"  package="com.example.naruhodo627.myapplication">  <uses-permission android:name="android.permission.INTERNET"/>  <uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE"/>  <uses-permission android:name="android.permission.ACCESS\_WIFI\_STATE"/>  <uses-permission android:name="android.permission.VIBRATE" />  <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=".MainActivity">  <intent-filter>  <action android:name="android.intent.action.MAIN" />  <category android:name="android.intent.category.LAUNCHER" />  </intent-filter>  </activity>  </application>  </manifest> |
| **1. MAINACTIVITY.JAVA** |
| package com.example.naruhodo627.myapplication;  import android.content.Context;  import android.hardware.Sensor;  import android.hardware.SensorEvent;  import android.hardware.SensorEventListener;  import android.hardware.SensorManager;  import android.os.Vibrator;  import android.support.v7.app.AppCompatActivity;  import android.os.Bundle;  import android.util.Log;  import android.view.View;  import android.widget.Button;  import android.widget.EditText;  import android.widget.TextView;  import android.widget.Toast;  import java.io.DataOutputStream;  import java.io.IOException;  import java.io.PrintWriter;  import java.net.Socket;  import java.util.ArrayList;  import java.util.Iterator;  import java.util.LinkedList;  public class MainActivity extends AppCompatActivity {  SensorManager sm;  SensorEventListener accL; // 자이로  SensorEventListener oriL; // 가속도  Sensor oriSensor;  Sensor accSensor;  TextView ax, ay, az; // 자이로  TextView ox, oy, oz; // 가속도  TextView mDistance;  TextView mSteps;  EditText editShakeHold;  Button btnReset;  ArrayList<Float> SensorX = new ArrayList<>();  ArrayList<Float> SensorY = new ArrayList<>();  ArrayList<Float> SensorZ = new ArrayList<>();  Button connectBtn;  EditText ip\_EditText;  EditText port\_EditText;  SocketClient client; // 서버접속을 위한 클라이언트클래스  SendThread send; // 안드로이드에서 임의의 문자 보내는것  Socket socket;  LinkedList<SocketClient> threadList;  private long lastTime = 0;  private float speed = 0;  private float lastX = 0;  private float lastY = 0;  private float lastZ = 0;  private float x, y, z;  private int SHAKE\_THRESHOLD = 800;  private int StepCount = 0;  Vibrator mVib;  @Override  protected void onCreate(Bundle savedInstanceState) { // 최초 생성시  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);  sm = (SensorManager)getSystemService(SENSOR\_SERVICE);  oriSensor = sm.getDefaultSensor(Sensor.TYPE\_ORIENTATION);  accSensor = sm.getDefaultSensor(Sensor.TYPE\_ACCELEROMETER);  oriL = new oriListener();  accL = new accListener();  ax = (TextView)findViewById(R.id.acc\_x);  ay = (TextView)findViewById(R.id.acc\_y);  az = (TextView)findViewById(R.id.acc\_z);  ox = (TextView)findViewById(R.id.ori\_x);  oy = (TextView)findViewById(R.id.ori\_y);  oz = (TextView)findViewById(R.id.ori\_z);  mSteps = (TextView)findViewById(R.id.mSteps);  mDistance = (TextView)findViewById(R.id.mDistance);  editShakeHold = (EditText)findViewById(R.id.editShakeHold);  btnReset = (Button)findViewById(R.id.btnReset);  mVib = (Vibrator) getSystemService(Context.VIBRATOR\_SERVICE);  ip\_EditText = (EditText) findViewById(R.id.ip\_EditText);  port\_EditText = (EditText) findViewById(R.id.port\_EditText);  connectBtn = (Button) findViewById(R.id.connect\_Button);  threadList = new LinkedList<MainActivity.SocketClient>();  // 연결버튼 클릭 이벤트  connectBtn.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View arg0) {  //Client 연결부  client = new SocketClient(ip\_EditText.getText().toString(),  port\_EditText.getText().toString());// 클라이언트 소켓 오픈  threadList.add(client); // 스레드 리스트에 추가  client.start(); // 클라이언트 시작  }  });  btnReset.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View arg0) {  //Client 연결부  String str;  StepCount = 0;  str = String.format("%d", StepCount);  mSteps.setText(str+"걸음");  }  });  } @Override  public void onResume() { // 일시 중지된 상태에서 액티비티로 다시 onResume() 실행  super.onResume();  sm.registerListener(accL, accSensor, SensorManager.SENSOR\_DELAY\_NORMAL);  sm.registerListener(oriL, oriSensor, SensorManager.SENSOR\_DELAY\_NORMAL);  }  @Override  public void onPause() { // 액티비티가 일시 중지했을 때 호출  super.onPause();  sm.unregisterListener(oriL);  sm.unregisterListener(accL);  }  private class accListener implements SensorEventListener { // 자이로 센서 값이 바뀔떄마다 호출해주는곳  public void onSensorChanged(SensorEvent event) {  ax.setText(Float.toString(event.values[0]));  ay.setText(Float.toString(event.values[1]));  az.setText(Float.toString(event.values[2]));  Log.i("SENSOR", "Acceleration changed.");  Log.i("SENSOR", " Acceleration X: " + event.values[0]  + ", Acceleration Y: " + event.values[1]  + ", Acceleration Z: " + event.values[2]);  try{  SHAKE\_THRESHOLD = Integer.parseInt(editShakeHold.getText().toString());  }  catch (Exception e)  {  return;  }  long currentTime = System.currentTimeMillis();  long gabOfTime = (currentTime - lastTime);  int count;  long cal;  String str;  String str3 = String.format("%d", currentTime) + " " + String.format("%d", lastTime);  mDistance.setText(str3);  if(gabOfTime > 100) {  lastTime = currentTime;  x = event.values[0];  y = event.values[1];  z = event.values[2];  speed = Math.abs(x+y+z - lastX - lastY - lastZ) / gabOfTime \* 10000;  if(speed > SHAKE\_THRESHOLD){  StepCount++;  str = String.format("%d", StepCount);  mSteps.setText(str+"걸음");  }  lastX = event.values[0];  lastY = event.values[1];  lastZ = event.values[2];  if(SensorX.isEmpty())  {  SensorX.add(event.values[0]);  SensorY.add(event.values[0]);  SensorZ.add(event.values[0]);  return;  }  int averX = 0;  int averY = 0;  int averZ = 0;  Iterator<Float> iter;  iter = SensorX.iterator();  count = 0;  while (iter.hasNext())  {  averX += iter.next();  count++;  }  averX = averX / count;  iter = SensorY.iterator();  count = 0;  while (iter.hasNext())  {  averY += iter.next();  count++;  }  averY = averY / count;  iter = SensorZ.iterator();  count = 0;  while (iter.hasNext())  {  averZ += iter.next();  count++;  }  averZ = averZ / count;  if(5 < Math.abs(event.values[0] - averX) && 5 < Math.abs(event.values[1] - averY) && 5 < Math.abs(event.values[2] - averZ))  {  mVib.vibrate(1000);  Toast.makeText(MainActivity.this, "넘어짐", Toast.LENGTH\_SHORT).show();  }  SensorX.add(event.values[0]);  SensorY.add(event.values[1]);  SensorZ.add(event.values[2]);  if(SensorX.size() > 10)  SensorX.remove(0);  if(SensorY.size() > 10)  SensorY.remove(0);  if(SensorZ.size() > 10)  SensorZ.remove(0);  }  send = new SendThread(socket); // 전송 스레드 시작  send.start();  // if(event.values[0] > 7)  // Toast.makeText(MainActivity.this, "UP", Toast.LENGTH\_SHORT).show();  }  public void onAccuracyChanged(Sensor sensor, int accuracy) {  }  }  private class oriListener implements SensorEventListener { // 가속도 센서 값이 바뀔때마다 호출해주는곳  public void onSensorChanged(SensorEvent event) {  ox.setText(Float.toString(event.values[0]));  oy.setText(Float.toString(event.values[1]));  oz.setText(Float.toString(event.values[2]));  Log.i("SENSOR", "Orientation changed.");  Log.i("SENSOR", " Orientation X: " + event.values[0]  + ", Orientation Y: " + event.values[1]  + ", Orientation Z: " + event.values[2]);  }  public void onAccuracyChanged(Sensor sensor, int accuracy) {  }  }  class SocketClient extends Thread {  boolean threadAlive;  String ip;  String port;  DataOutputStream output = null;  public SocketClient(String ip, String port) {  threadAlive = true;  this.ip = ip;  this.port = port;  }  @Override  public void run() {  try {  // 연결후 바로 ReceiveThread 시작  socket = new Socket(ip, Integer.parseInt(port)); //ip주소와 port로 연결 (ip는 string, port 는 integer)  output = new DataOutputStream(socket.getOutputStream()); // 출력 스트림 오픈  //receive = new ReceiveThread(socket); // 수신 스레드 시작  //receive.run();  // receive.start();  } catch (IOException e) {  e.printStackTrace();  }  }  }  class SendThread extends Thread {  Socket socket;  // String sendmsg = editText\_massage.getText().toString() + "\n"; // 메세지 내용 가져옴  String sendmsg = Integer.toString((int)lastX) + " " + Integer.toString((int)lastY);  //DataOutputStream output;  PrintWriter socket\_out;  public SendThread(Socket socket) {  this.socket = socket;  try {  // output = new DataOutputStream(socket.getOutputStream()); // 데이터 출력 스트림 오픈  socket\_out = new PrintWriter(socket.getOutputStream(), true);  } catch (Exception e) {  }  }  @Override  public void run() {  try {  // 메세지 전송부  Log.d(ACTIVITY\_SERVICE, "11111");  //if (output != null) {  if (sendmsg != null) {  //sendmsg += "\n";  //output.write(sendmsg.getBytes()); // 바이트를 출력 스트림으로 씀, 즉 전송  socket\_out.println(sendmsg);  }  //}  // } catch (IOException e) {  // e.printStackTrace();  } catch (NullPointerException npe) {  npe.printStackTrace();  }  }  }  } |
| **1. MAINACTIVITY.JAVA** |
| package cosmos;  import java.io.\*; // 입출력  import java.net.\*; // 네트워크 소켓통신  import java.sql.\*; // 데이터베이스 쿼리문  import java.awt.event.\*; // 이벤트  public class server {  static String aname; // 클라이언트 핸드폰 값을 문자열  static String address; // 구분자에 의해서 aname, address 분리하기위한 변수  static int asum;  public static void main(String[] args) {  ServerSocket serverSocket = null; // port, client accept  Socket socket = null;  try {  serverSocket = new ServerSocket(90);  socket = serverSocket.accept();  System.out.println("server connected");  // ServerSocket port client accept  // socket  //InputStream in = socket.getInputStream();  BufferedReader ain = new BufferedReader(new InputStreamReader(socket.getInputStream()));  OutputStream out = socket.getOutputStream();      while(true)  {  String str = ain.readLine(); // android send data receive  String ar[] = str.split(" "); // " " 기준으로 파싱  // for(int i=0; i<ar.length; i++){  aname = ar[0]; // x  address = ar[1]; // y  // android ==> javaserver 데이타는 문자열  int su1 = Integer.parseInt(aname);  int su2 = Integer.parseInt(address);  int sum = 0;  sum = su1 + su2;  System.out.println(ar[0]);  System.out.println(ar[1]);  System.out.println(sum);  asum = sum;  // }  A A1 = new A(); // 데이타베이스 class  A1.insert(); // A 클래스  }  } catch (Exception e) {  e.printStackTrace();  } finally {  try {  socket.close();  } catch (Exception ignored) {  }  try {  serverSocket.close();  } catch (Exception ignored) {  }  }  }  }  class A { //  public void insert() {  try {  Class.forName("org.gjt.mm.mysql.Driver");  } catch (ClassNotFoundException e1) {  System.err.println(e1);  }  Connection conn = null;  PreparedStatement pstmt = null;  String url = "jdbc:mysql://127.0.0.1:3306/cosmos1"; // 디비 설치된 ip ort 디비명  String id = "root"; // mysql id  String pwd = "1234"; // mysql pwd  String query = null; // 쿼리값 확인  System.out.println("driver connected");  try {  conn = DriverManager.getConnection(url, id, pwd);  } catch (SQLException e1) {  System.err.println(e1);  }  while (true) {  String aname = server.aname;  String address = server.address;  int asum = server.asum;  query = "insert into bravo values(?,?,?)";  try {  pstmt = conn.prepareStatement(query);  pstmt.setString(1, aname);  pstmt.setString(2, address);  pstmt.setInt(3, asum);  pstmt.executeUpdate();  pstmt.close();  conn.close();  break;  //System.exit(0);  } catch (SQLException e1) {  System.err.println(e1);  }  }  }  } |