

MANIFEST

<**uses-permission android:name="android.permission.BLUETOOTH"** />

**// Main – start service**

**// START SERVICE**

val1 = (EditText) findViewById(R.id.***Alto***);  
val2 = (EditText) findViewById(R.id.***Bajo***);  
telefono = (EditText) findViewById(R.id.***NumTel***);  
  
String Alto = val1.getText().toString();  
String Bajo = val2.getText().toString();  
String NumFrom = telefono.getText().toString();  
  
**try** {  
 Class aClass = Class.*forName*(**"com.example.tdal6392.servicemonitorbattery.ServiceBattery"**);  
 Intent service1 = **new** Intent(getApplicationContext(), aClass);  
 service1.putExtra(**"Alto"**, Alto);  
 service1.putExtra(**"Bajo"**, Bajo);  
 service1.putExtra(**"NumFrom"**, NumFrom);  
  
 startService(service1);  
}  
**catch**( Exception e) {}

**SERVICE**

**import** android.app.Service;  
**import** android.bluetooth.BluetoothAdapter;  
**import** android.bluetooth.BluetoothDevice;  
**import** android.bluetooth.BluetoothSocket;  
**import** android.content.BroadcastReceiver;  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.content.IntentFilter;  
**import** android.os.BatteryManager;  
**import** android.os.Bundle;  
**import** android.os.IBinder;  
**import** android.provider.Telephony;  
**import** android.support.annotation.Nullable;  
**import** android.telephony.SmsManager;  
**import** android.telephony.SmsMessage;  
**import** android.widget.Toast;  
  
**import** java.io.IOException;  
**import** java.io.InputStream;  
**import** java.io.OutputStream;  
**import** java.util.Set;  
**import** java.util.UUID;  
**import** android.os.Handler;

**public class** ServiceBattery **extends** Service {

// parameters to start service  
 **int alto**, **bajo**;  
 String **NumFrom** = **""**;

String **EMail** = **""**;

// variables for bluetooth  
 BluetoothSocket **mmSocket** = **null**;  
 **int BTStatus** = 0;

*// @Override* **public void** onCreate() {  
 **super**.onCreate();  
  
 *// REGISTER BROADCAST RECEIVER* **this**.registerReceiver(**this**.**mBatInfoReceiver**,  
 **new** IntentFilter(Intent.***ACTION\_BATTERY\_CHANGED***));  
  
  
 *// OPEN SOCKET FOR BLUETOOTH* **mmSocket** = GetSocketBT();  
 String mensaje = **"error in blueetooth"**;  
  
 **if** (**mmSocket** != **null**) {  
 ListenForData();  
 mensaje = **"ARD Service Started"**;  
 }  
 **else** {  
 **if** (**BTStatus** ==1) mensaje = **"ARD Error device do not have bluetooth"**;  
 **if** (**BTStatus** ==2) mensaje = **"ARD Error bluetooth not enabled"**;  
 **if** (**BTStatus** ==3) mensaje = **"ARD Error bluettoth device not found"**;  
 }  
  
 SmsManager sms = SmsManager.*getDefault*();  
 sms.sendTextMessage(**NumFrom**, **null**, mensaje, **null**, **null**);

Toast.*makeText*(getApplicationContext(), **"Service Started"**, Toast.***LENGTH\_LONG***).show();  
 }

@Nullable  
 @Override  
 **public** IBinder onBind(Intent intent) {  
  
 Bundle extras = intent.getExtras();

String altoIn = (String) extras.get(**"Alto"**);  
 String bajoIn = (String) extras.get(**"Bajo"**);  
 **NumFrom** = (String) extras.get(**"NumFrom"**);

**EMail** = (String) extras.get(**"EMail"**);  
  
 **alto** = Integer.*parseInt*(altoIn.toString());  
 **bajo** = Integer.*parseInt*(bajoIn.toString());  
  
 **return null**;  
 }

@Override  
 **public void** onDestroy() {  
   
 **try** {  
 **mmSocket**.close();  
 }  
 **catch** (Exception e) {}  
 }

**private** BroadcastReceiver **mBatInfoReceiver** = **new** BroadcastReceiver() {

@Override  
 **public void** onReceive(Context ctxt, Intent intent) {  
  
 *// DETECT BATTERY LEVEL TO STOP RECHARGE OR START RECHARG  
 // ======================================================* **int** level = intent.getIntExtra(BatteryManager.***EXTRA\_LEVEL***, 0);  
  
 **if** (level == **alto**) SendData(**"$"**); *// high, stop charging* **if** (level == **bajo**) SendData(**"#"**); *// low, start charging  
 // batteryTxt.setText(String.valueOf(level) + "%");*

*// DETECT INCOMING SMS FOR ARDUINO  
 //================================* String action = intent.getAction();  
 **if** (action.equals(**"android.provider.Telephony.SMS\_RECEIVED"**)) {  
 **for** (SmsMessage sms1 : Telephony.Sms.Intents.*getMessagesFromIntent*(intent)) {  
  
 **NumFrom** = sms1.getOriginatingAddress();  
 String msgBody = sms1.getMessageBody();  
  
 **if** (!msgBody.contains(**"ARD "**)) {  
 **return**;  
 }  
  
 msgBody = msgBody.replace(**"ARD "**, **""**);  
 SendData(msgBody);  
  
 } *// for* } *// if* }  
 };

***// OPEN BLUETOOTH SOCKET FOR ARDUINO  
 //==================================***

**public** BluetoothSocket GetSocketBT() {  
  
 BluetoothAdapter mBluetoothAdapter = BluetoothAdapter.*getDefaultAdapter*();  
 **BTStatus** = 0;  
  
 **if** (mBluetoothAdapter == **null**) {  
 **BTStatus** = 1;  
 **return null**;  
 }  
  
 **if** (!mBluetoothAdapter.isEnabled()) {  
 Intent enableBluetooth = **new** Intent(BluetoothAdapter.***ACTION\_REQUEST\_ENABLE***);  
 **BTStatus** = 2;  
 **return null**;  
 *//startActivityForResult(enableBluetooth, 0);* }  
  
 BluetoothDevice mmDevice = **null**;  
 Set<BluetoothDevice> pairedDevices = mBluetoothAdapter.getBondedDevices();  
 **if** (pairedDevices.size() > 0) {  
 **for** (BluetoothDevice device : pairedDevices) {  
 **if** (device.getName().equals(**"FireFly-108B"**)) {  
 mmDevice = device;  
 **break**;  
 }  
 }  
 }  
  
 **if** (mmDevice == **null**) {  
 **BTStatus** = 3;  
 **return null**;  
 }  
  
 *// OPEN BLUETOOTH* UUID uuid = UUID.*fromString*(**"00001101-0000-1000-8000-00805f9b34fb"**); *//Standard SerialPortService ID* **try** {  
 **mmSocket** = mmDevice.createRfcommSocketToServiceRecord(uuid);  
 **mmSocket**.connect();  
 } **catch** (Exception e) {  
 }  
  
 *//myLabel.setText("Bluetooth Opened");* **return mmSocket**;  
 }  
  
  
 ***// SEND DATA TO ARDUINO  
 // ====================*****public void** SendData(String mensaje) {  
 **try** {  
 OutputStream mmOutputStream = **mmSocket**.getOutputStream();  
 **for** (**int** i = 0; i < mensaje.length(); i++) {  
 **char** caracter = mensaje.charAt(i);  
 mmOutputStream.write(caracter);  
 }  
  
 mmOutputStream.close();

} **catch** (Exception e) {  
  
 }  
 }  
  
  
 ***// LISTEN TO INCOMING DATA FROM ARDUINO  
 //=====================================*****public void** ListenForData()  
 {  
 **final** Handler handler = **new** Handler();  
 **final byte** delimiter = 10; *// new line ascii* **int** readBufferPosition = 0;  
  
 Thread workerThread = **new** Thread(**new** Runnable() {  
 **public void** run() {  
  
 InputStream mmInputStream = **null**;  
  
 **try** {  
 mmInputStream = **mmSocket**.getInputStream();  
 }  
 **catch** (Exception e) {}  
  
 **byte**[] readBuffer = **new byte**[1024];  
 Boolean stopWorker = **false**;  
  
 **while**(!Thread.*currentThread*().isInterrupted() && !stopWorker) {  
 **try** {  
  
 **int** readBufferPosition = 0;  
 **int** bytesAvailable = mmInputStream.available();  
  
 **if**(bytesAvailable > 0) {  
 **byte**[] packetBytes = **new byte**[bytesAvailable];  
 mmInputStream.read(packetBytes);  
 **for**(**int** i=0;i<bytesAvailable;i++) {  
 **byte** b = packetBytes[i];  
 **if**(b == delimiter) {  
 **byte**[] encodedBytes = **new byte**[readBufferPosition];  
 System.*arraycopy*(readBuffer, 0, encodedBytes, 0, encodedBytes.**length**);  
 **final** String data = **new** String(encodedBytes, **"US-ASCII"**);  
 readBufferPosition = 0;  
  
 handler.post(**new** Runnable() {  
 **public void** run() {

//==============  
  
String mensaje = **"ARD "** + data;  
  
**if** (**NumFrom**!=**""**) {  
  
 SmsManager sms = SmsManager.*getDefault*();  
 sms.sendTextMessage(**NumFrom**, **null**, mensaje, **null**, **null**);  
}  
**if** (**EMail**!=**""**) {  
 Intent emailIntent = **new** Intent(Intent.***ACTION\_SEND***);  
  
 emailIntent.setData(Uri.*parse*(**"mailto:"**));  
 emailIntent.setType(**"text/plain"**);  
  
 emailIntent.putExtra(Intent.***EXTRA\_EMAIL***, **EMail**);  
 emailIntent.putExtra(Intent.***EXTRA\_SUBJECT***, **"Arduino MSG"**);  
 emailIntent.putExtra(Intent.***EXTRA\_TEXT***, mensaje);  
  
 **try** {  
 startActivity(Intent.*createChooser*(emailIntent, **"Send mail..."**));  
 *//finish();* }  
 **catch** (Exception e) {  
 }  
}

//==============  
 }  
 });  
 }  
 **else** {  
 readBuffer[readBufferPosition++] = b;  
 }  
 }  
 }  
 }  
 **catch** (IOException ex) {  
 stopWorker = **true**;  
 }  
 }  
 }  
 });  
  
 workerThread.start();  
 }  
}