

## Inhalt

1.	Allgemeines .....	2
2.	Installation .....	3
3.	Bedienung .....	4
3.1	Liste aller Android Geräten in der Umgebung .....	5
3.2	Anzeigen der Messdaten.....	6
3.3	Fehlermeldung.....	7
4.	Technische Details .....	8
5.	Ablaufdiagramm .....	9
6.	Pseudo Code (Teil) .....	10

## 1. Allgemeines

An der Fachhochschule Nordwest Schweiz werden unter Leitung von Prof. Michael Stanimirov Forschungsarbeiten mit der Impedanzspektroskopie durchgeführt. Im Rahmen dieser Forschungsarbeiten werden unter anderen Impedanz Messungen am Biomolekül Vitamin B12 durchgeführt. Im Versuchsaufbau wird als Messelektrode DS DROPSENS verwendet (siehe Elektrischer Impedanz-Fingerabdruck von Biomolekülen von Prof. M. Stanimirov).

Aus dem Forschungsprojekt wurde das Projekt Visualisierung der Messwerte in Echtzeit über Bluetooth auf Endgeräte mit Android Betriebssystemen wie Tablet / Natel oder GEAR Watch entwickelt. Die Motivation oder Aufgabenstellung war, dass die Messgeräte, als Sender mit Bluetooth ausgestattet werden und als Datenschleuder dienen können. Die Messdaten werden von den Endgeräten mit Android über Bluetooth empfangen und permanent angezeigt.

Der Code für das Projekt wurde in Java (Android 4.4.2 / API 19) geschrieben und der Objektcode in ein APK kompiliert. Das APK heisst "BIO Z" und kann über ein Filemanager auf die Android-Geräte direkt installiert werden.

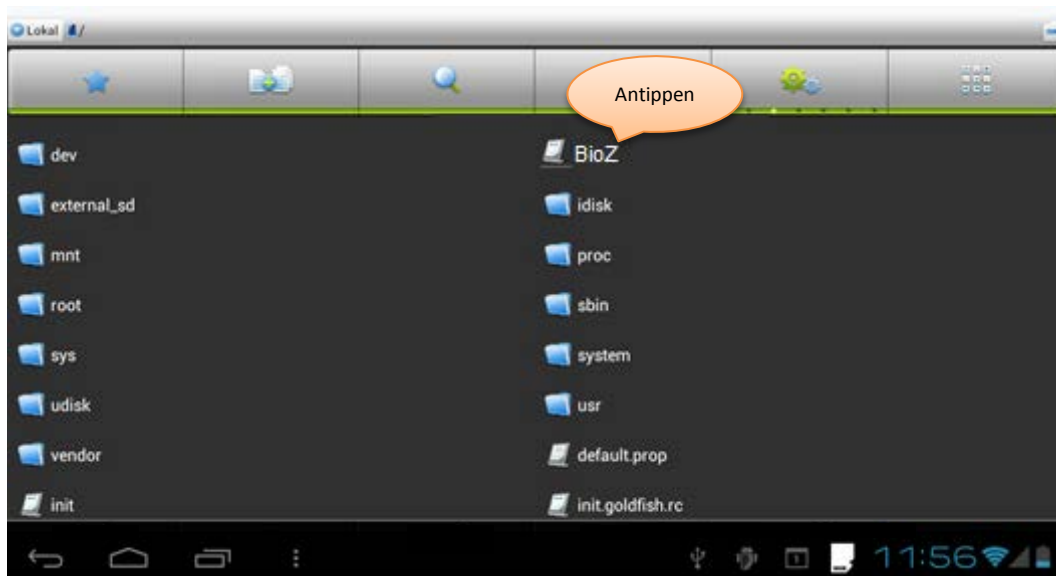
Die Messelektrode DS DROPSENS muss mit einem Bluetooth taugliches Gerät verbunden sein. Das Bluetooth taugliches Gerät kann auf die Grösse eines Bluetooth Sticks reduziert sein in das dann die Messelektrode via Interface Stecker gesteckt werden kann.



## 2. Installation

Das APK wird bei jedem Testlauf im Emulator- oder Debugmodus, (**Run As (Android Application)**) neu erstellt und im bin - Verzeichnis des Projekts abgelegt. Es kann von dort aus in jedes mit USB angeschlossenen Tablet oder Smartphone mit Android System kopiert werden. Gilt nicht für die Gear Watch. Für die Gear Watch braucht man das entsprechende Smartphone und das kostenpflichtige Wondershare go for Android.

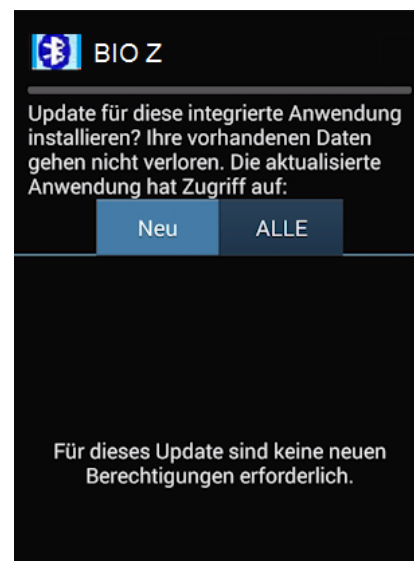
Für die Installation auf dem Tablet und auf dem Smartphone fiel die Entscheidung auf den ES Datei Explorer. Das APK wurde in ein privates Verzeichnis unter dem vorhandenen Download Verzeichnis auf das Tablet kopiert.



Mit dem ES Datei Explorer muss man zu der zu installierende APK navigieren und nach dem Antippen der APK wird der rechte Installationsdialog geöffnet.

Durch das bestätigen auf den Button "ALLE" wird die APK installiert.

(APK = App)



### 3. Bedienung

Nach der Installation des APK "BIO Z" ist die Bedienung ziemlich einfach.  
Nach ISO Standard, sollte ein System mit drei Mausklicks oder max. dreimal Tippen die gewünschten Daten anzeigen.

Wir benötigen insgesamt nur zwei Tipps, um die Messdaten vom entsprechenden Datensender anzeigen zu lassen.



Durch antippen des Icon "BIO Z" wir die APK gestartet und in einen neuen Bildschirm gewechselt.  
Dort werden alle Geräte aus der Umgebung erfasst, aufgelistet und angezeigt.

### 3.1 Liste aller Android Geräten in der Umgebung

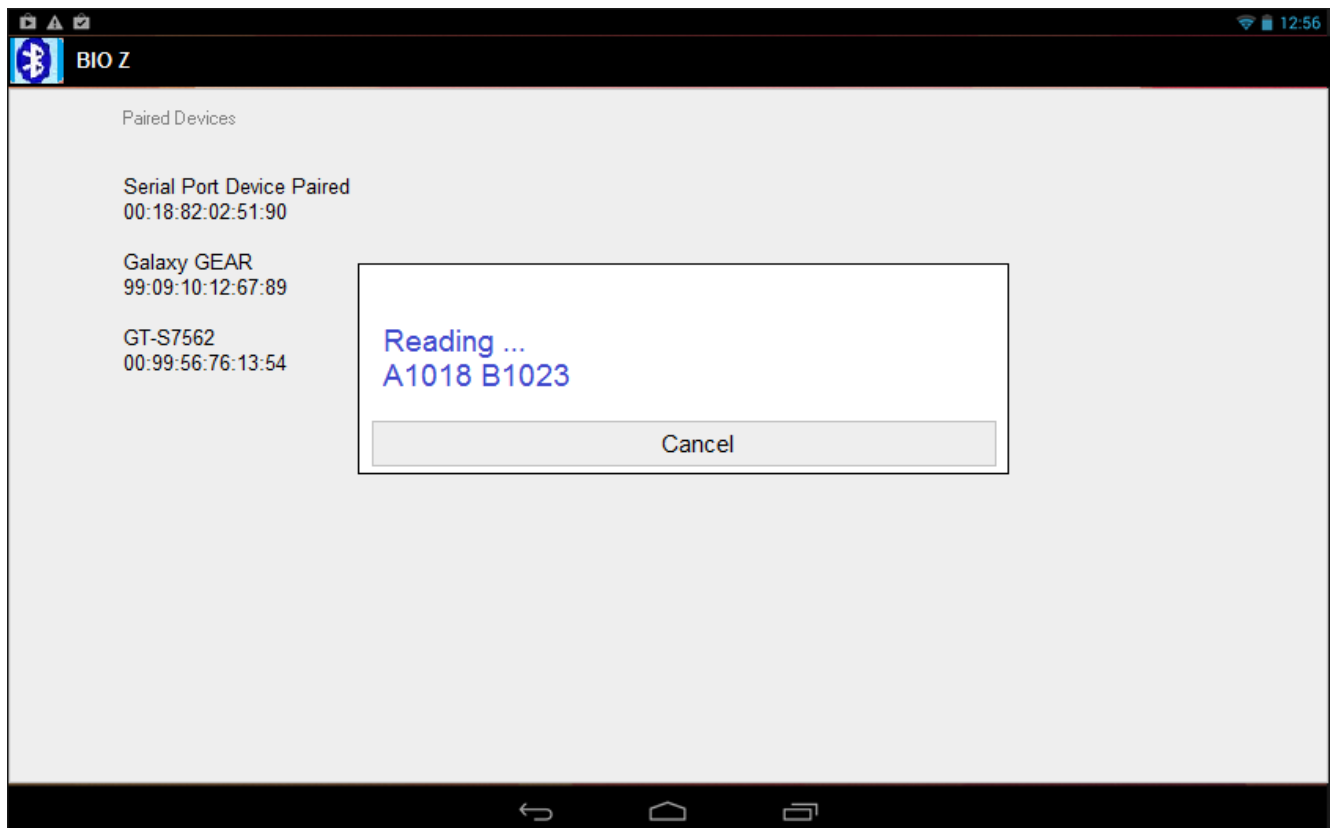


Es wird die Gerätebezeichnung und die eindeutige, zwölfstellige Identifikationsnummer von jedem erfassten Gerät ausgelesen und angezeigt.

Hinter der Gerätebezeichnung wird mit "Paired" angezeigt, ob das Gerät gekoppelt ist, oder noch gekoppelt werden muss.

Unser Datensender ist das "Serial Port Device". und es ist schon gekoppelt (Paired).  
Durch antippen wird ein Dialog geöffnet in welchem die Daten angezeigt werden.

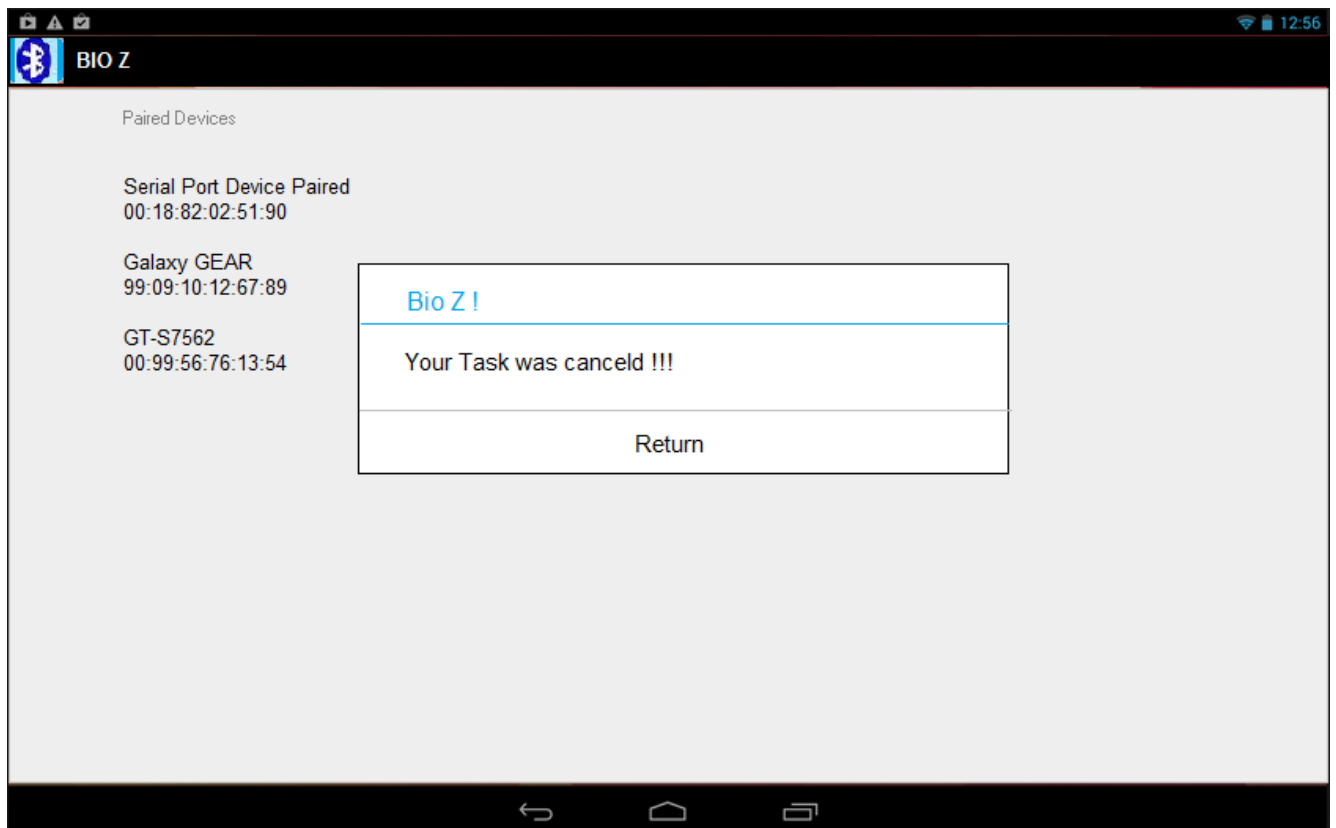
### 3.2 Anzeigen der Messdaten



Die Messdaten werden in der Dialogbox auf dem Bildschirm angezeigt.  
Mit Cancel wird die Anzeigebox geschlossen und kann durch antippen auf die Gerätebezeichnung wieder geöffnet werden.

Wir rufen jede Sekunde einen Wert über Bluetooth ab und zeigen ihn an.  
Die Grösse ist dynamisch eingestellt und passt sich der Bildschirmgröße des Geräts an.

### 3.3 Fehlermeldung



Es gibt nur zwei Gründe für die Fehlermeldung.

Die Fehlermeldung wird angezeigt, wenn Sie mit dem Tablet oder dem Natel sich zu weit von dem Datensender entfernen. (im Test, über 10 m). Ab dieser Entfernung kann das Gerät keine Daten mehr empfangen.

Oder wenn das Datensendegerät einseitig abgestellt wird.

Mit Return wird die Meldungsbox geschlossen und Sie kommen wieder zurück auf die Liste der verfügbaren Geräte.

## 4. Technische Details

### Verbindungsaufbau mit Bluetooth über SPP

Bluetooth ist ein Funk Protokoll für lokale Kommunikation.

#### Auffinden (Discovery)

Die Geräte müssen freigegeben, also gekoppelt werden (paired).

Die Kommunikation mit einem Bluetooth-Gerät muss festgelegt werden.  
Es muss freigeschalten werden. (Also auf sichtbar gesetzt)

#### Protokolle

Mehrere Protokolle sind möglich.

RFCOMM Ist Verbindungsorientiert.

BTSP Bluetooth Serial Port Profile.

#### Adresse

Jedes Gerät verfügt über eine 12-stellige hexadezimale Adresse z.B. E4:32:CB:84:29:FF.

Es ist die MAC Adresse des Gerätes von 48 Bit Länge.

#### Port

Ist der Kanal, auf dem das Remote-Gerät Daten empfangen kann.

z.B. Port 3

Eine komplette Spezifikation wäre

btsp:// E4:32:CB:84:29:FF: 3

#### UUID

Services werden eindeutig durch eine **Universell Unique Identifier (UUID)** identifiziert.

Es ist eine 128-Bit-Zahl, in Hexadezimal

wie z. B. BAE0D0C0B0A00095570605040302011.

Oder

**SPP:** UID = "00001101-0000-1000-8000-00805F9B34FB"

#### Service Discovery Protocol (SDP).

Über das SD Protokoll wird die URL einer Service - UUID bestimmt.

#### Server

Empfängt Daten über die UUID und zeigt sie an.

#### Client

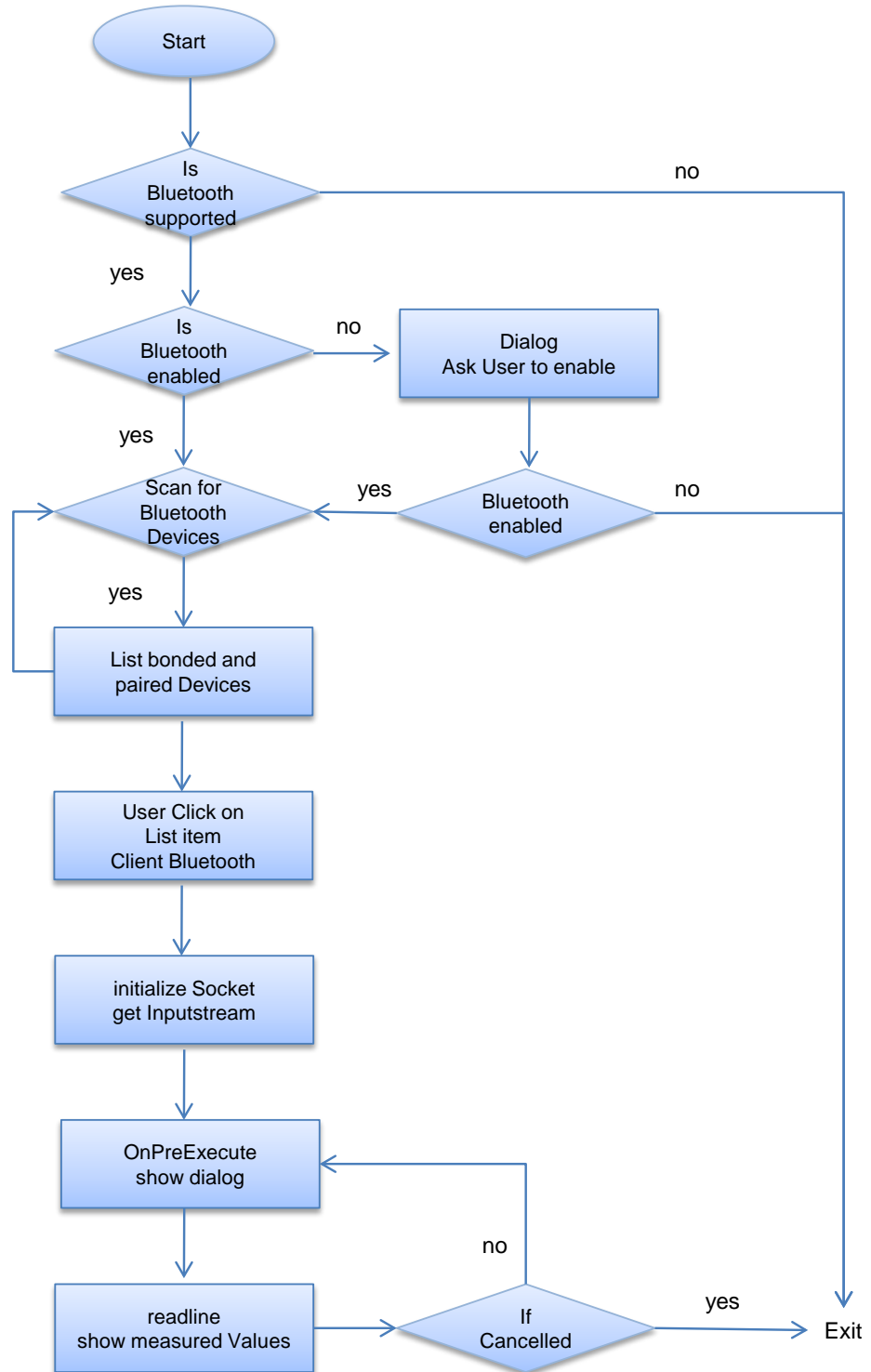
Führt einen Verbindungsaufbau, für die angegebene UUID aus und sendet die Daten.

Server verbindet sich mit der angegebenen URL und fängt die Daten auf.

Da RFCOMM ein Verbindungsorientiertes Protokoll ist, fungiert es als Server, welcher Verbindungen annehmen kann. Während der Client als Auftraggeber über die Verbindung, die Daten zu Verfügung stellt.



5. Ablaufdiagramm



## 6. Pseudo Code (Teil)

```

package ch.fhnw.readbioz;
import Part....

class ReceiveData extends Activity implements OnItemClickListener

Declaration part for global variables

public void onCreate(Bundle savedInstanceState)
super.onCreate(savedInstanceState);
setContentView(R.layout.main);
init();
If BluetoothAdapter == NULL
    Message
    finish();
else{
    if(!btAdapter.isEnabled())
        turnOnBT();
    getPairedDevices();
    startDiscovery();

private void startDiscovery()
    btAdapter.cancelDiscovery();
    btAdapter.startDiscovery();

private void turnOnBT()
    Intent intent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
    startActivityForResult(intent, 1);

private void getPairedDevices()
    devicesArray = btAdapter.getBondedDevices();
    if(devicesArray.size() > 0)
        for(BluetoothDevice device:devicesArray)
            pairedDevices.add(device.getName() + "\n" + device.getAddress());

protected void Pause()
    super.onPause();
    unregisterReceiver(receiver);

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data)
    super.onActivityResult(requestCode, resultCode, data);
    strMessage = "You have to activate Bluetooth !";
    if(resultCode == RESULT_CANCELED){
        Toast.makeText(getApplicationContext(), strMessage, Toast.LENGTH_LONG).show();
        finish();

@Override
public void onBackPressed()
    moveTaskToBack(true);
    bReturn = true;

@Override
public boolean onKeyDown(int keyCode, KeyEvent
    if (keyCode == KeyEvent.KEYCODE_BACK)
        bReturn = true;
        moveTaskToBack(true);
        return true;
    return super.onKeyDown(keyCode, event);

private void init()
public void onItemClick(AdapterView<?> arg0, View arg1, int arg2, long arg3) .
class newTask extends AsyncTask<String, String, String> :

```