# Aktueller Arbeitsstand

In der App funktioniert soweit alles bis auf die Umwandlung der Bytes in den richtigen Ton.  
Auch die Connections, zu den Services, funktionieren alle ohne Probleme. Um das Problem mit dem Ton zu beheben musste du in der Class [ControlBlinkyActivity](#_ControlBlinkyActivity) und in der Funktion rawToWave() das Schreiben der .wav Datei bearbeiten. Den Fehler konnte ich leider ohne deine Hilfe nicht mehr beheben. Du solltest besser Bescheid wissen als ich was genau in die .wav Datei geschrieben wird. Die erstellte .wav Datei findest du im Internal Storage als vocal.wav. Um diese zu finden habe ich auf meinem Android ein Explorer Datei App installiert.

# Kurz Beschreibung

Die App läuft im generellen von drei Activitys. Die erste Activity ist die [SplashScreenActivity](#_SplashScreenActivity). Diese startet die App und zeigt nur das Logo. Bei dieser Activity muss man nichts mehr verändern da bereits alles funktioniert. Anschliessend wird man automatisch weitergeleitet in die nächste Activity. Die zweite Activity, welche die Connection zum Gerät und den beiden Services herstellt, ist die [MainActivity](#_MainActtivity). In dieser wird eine Liste erstellt und nach allen Geräten in der Umgebung gesucht welche dann in der Liste angezeigt werden. Sobald man auf ein Gerät klickt wird zuerst die Connection zum Gerät aufgebaut und anschliessend zur nächste und letzten Activity weitergeleitet. In der [ControlBlinkyActivity](#_ControlBlinkyActivity) passiert das meiste. Zuerst wird die ganze Anzeige der Buttons und Wave etc. erstellt. Anschliessend werden die Services Classes connected. Danach wird per Eingabe mit dem Gerät kommuniziert.

In diesen drei Activitys wurde das meiste programmiert und geschieht auch das meiste. Du wirst das meiste in diesen Classes programmieren. Die beiden Service Classes sollten nicht verändert werden, die wurden so von Nordic kopiert. Auch die restlichen Classes sollten nicht verändert werden, da auch die meisten von diesen von Nordic stammen und funktionieren.

# Activity Classes

## SplashScreenActivity

onCreate() – Input: Bundle savedInstanceState – Output: keiner  
Hier wird der Start Screen angezeigt und nach ein paar Millisekunden weitergeleitet.

## MainActivity

onCreate() – Input: Bundle savedInstanceState – Output: keiner  
Hier wird eine Liste erstellt in welcher anschliessend die verfügbaren Geräte angezeigt werden.

prepareForScan() – Input: keiner – Output: keiner  
Hier wird die gesuchte UUID herausgelesen und der Scanner vorbereitet.

onItemClick() – Input: AdapterView<?> parent, View view, int position, long id  
Hier wird beim Auswählen eines Gerätes mit diesem connected und die Activity ControlBlinkyActivity wird gestartet.

onStart() – Input: keiner – Output: keiner  
Hier wird die Funktion startLeScan() gestartet.

onStop() – Input: keiner – Output: keiner  
Hier wird die Funktion stopLeScan() gestartet.

startLeScan() – Input: keiner – Output: keiner  
Hier wird zuerst alle Einstellungen übernommen und anschliessend startet der scann.

stopLeScan() – Input: keiner – Output: keiner  
Hier wird das Scannen gestoppt und alle Einstellung zurückgesetzt.

## ControlBlinkyActivity

onCreate() – Input: Bundle savedInstanceState – Output: keiner  
Hier wird zuerst die ganze Darstellung erstellt. Zudem werde alle Listener zugeordnet. Die Connection zum BlinkyService wird hier aufgebaut.

service\_Init(); - Input: keiner – Output: keiner  
Hier wird die Connection mit dem UARTService aufgebaut.

(UARTStatusChangeReceiver) C00593() – Input: keiner – Output: keiner  
In dieser Unterclass wird mit dem UART Service kommuniziert. In der Unterfunktion onReceive() werden alle Daten welche man vom UART Service erhält bearbeitet. Beim Empfangen der Bytes wird die Funktion rawToWave aufgerufen und die Bytes mitgeliefert. Anschliessend erhält man von der Funktion mit einem String mit dem Pfad der .wav Datei. Danach wird diese per MediaPlayer abgespielt.

rawToWave() – Input: byte[] rawArray – Output: String filePath  
Hier werden die Bytes verarbeitet und in eine .wav Datei abgespeichert. Anschliessend wird ein String mit dem Pfad zurückgeschickt.

convert() – Input: byte[] array – Output: short[] converted  
Hier wird der ByteArray in ein Short Array umgewandelt.

BroadcastReceiver() – Input: keiner – Output: keiner  
Hier wird mit dem Blinky Service kommuniziert. Beim Empfangen von verschiedenen Signalen werden die jeweiligen Funktionen ausgeführt.

In den Activitys sind noch einige mehrere Funktionen welche aber nicht benötigt oder verändert werden sollten. Falls du trotzdem bei einer solchen Funktion nicht mehr weiter kommst kannst du mir schreiben oder anrufen.

# Informationsquellen

Die meisten Code Teile fand ich auf dem GitHub von Nordic: <https://github.com/NordicSemiconductor>

Bei allgemeinen Java Fragen findet man auf Stack Overflow viele hilfreiche Infos:  
<https://stackoverflow.com/>

Ansonsten kann man nur das Internet durchforsten für eine Lösung.