Terabee SDK for Android
Quick Start Setup **V1.1.1**

Overview

TerabeeSDK provides an easyway to communicate with Terabee sensorson anAndroid device. The SDK supports the following sensor models:

- TeraRanger Evo 3M
- TeraRanger Evo 60M
- TeraRanger Evo 64px
- TeraRanger Multiflex

The SDK support auto-detecting of type of sensor, that allow connect to any Terabee sensor without exactly define its type.

Setting up the SDK

TheSDK setup consists of a few simple stepsoutlined below.

1. Open Android Studioand place theSDK file (Terabee_SDK_{VERSION_NUMBER}.aar) tothe librariesfolder of your project. Normallyitwould be:

```
YouApp/app/libs
```

2. Add a new dependency to the build.gradle :

```
dependencies {
            ...
        implementation fileTree(dir: 'libs', include: ['*.aar'])
        implementation 'com.github.felHR85:UsbSerial:4.5'
            ...
}
```

3. Addthe followingdevice_filter.xmlfile to the resource folder /res/xml:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <usb-device vendor-id="1155" product-id="22336" />
</resources>
```

4. Edit your AndroidManifest.xmlin the followingway:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.terabee.sdkdemo">

    <uses-feature
    android:name="android.hardware.usb.host"
    android:required="false" />

<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"
        tools:ignore="GoogleAppIndexingWarning">
        <activity
              android:name=".MainActivity"
              android:launchMode="singleInstance">
              <intent-filter>
                    <action android:name="android.intent.action.MAIN" />

                    <category android:name="android.intent.category.LAUNCHER" />

              </intent-filter>

              <intent-filter>
                    <action
android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED" />
              </intent-filter>

              <meta-data

android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED"
                    android:resource="@xml/device_filter" />
        </activity>
    </application>
</manifest>
```

Initialiazing the SDK

Before you start to use the SDK, it's required to be initialized. Additionally, after your app finishes using the SDK it needs to release it explicitly. In the context of a typical Android Activity it may look like this:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      ...
      // init Terabee SDK
      TerabeemSdk.getInstance().init(this);
}

@Override
protected void onDestroy() {
      // release Terabee SDK
      TerabeeSdk.getInstance().dispose();
      super.onDestroy();
}
```

Connecting to a sensor

To receive data from a sensor(s) the SDK needs to connect to such first. That should be done in the following way:

```
TerabeeSdk.getInstance().connect(new TerabeeSdk.IUsbConnect() {
      @Override
      public void connected(boolean success, TerabeeSdk.DeviceType deviceType){
      }

      @Override
      public void disconnected() {
      }
      @Override
      public void permission(boolean granted) {
```

```
        }
}, TerabeeSdk.DeviceType.EVO_60M);
```

In the process of connecting you willreceive callbacks: whether connection established successfully, when devicegetsdisconnected plusresultsof permission requests.Also, note that SDK supports connection to sensor with auto-detection of sensor type. For apply auto-detect mode need pass in SDK next type of sensor:

TerabeeSdk.DeviceType.AUTO_DETECT

Receiving data from the sensor

To receive data from the connected sensor, it's necessary to registerthe datareceiver accordingto the sensor-type.

Create instance of a receiver for aEvo 3M and Evo 60M sensor:

```
private final TerabeeSdk.DataDistanceCallbackmDataDistanceCallback= new TerabeeSdk.Data-
OnePixelCallback() {
    @Override
    public void onDistanceReceived(int distance, int dataBandwidth, int dataSpeed)
    {// received distance from the sensor
    }
    @Override
    public void onReceivedData(byte[] bytes, int i, int i1){
        // received raw data from the sensor
    }
};
```

Register the data receiver:

TerabeeSdk.getInstance().registerDataReceive(mDataDistanceCallback);

After completingof usingthesensoryouneed to unregister the data receiver:

TerabeeSdk.getInstance().unregisterDataReceive(mDataDistanceCallback);

To receive data from a different sensor it's necessarytoregister a different data receiver accordingto the sensortype.The SDK supports the followingdata receivers:

DataDistanceCallback–for Evo 3Mand Evo60Msensors, allow to receive distance value from the sensor

DataDistancesCallback –for Multiflex sensors, allow to receive array values of distances from the sensor

DataMatrixCallback –for Evo 64pxsensors, allow to receive matrix of distances.

DataCallback–for all types of Terabee sensors, allow to receive raw data from any sensor

DataSensorCallback–for all supported types of Terabee sensors, allow to receive all parsable data from sensors. In essence is the combination of the abovecallbacks.

Done! You are ready to use the terabee SDK.