

# Terabee SDK for Android – Quick Start Setup

V1.2.0

## Overview

Terabee SDK provides an easy way to communicate with Terabee sensors on an Android device. The SDK supports the following sensor models:

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

The SDK also supports auto-detecting of sensor type, allowing connection to any Terabee sensor without specific selection.

## Setting up the SDK

The SDK setup consists of a few simple steps outlined below:

1. Open Android Studio and place the SDK file (Terabee\_SDK\_{VERSION\_NUMBER}.aar) to the libraries folder of your project. Normally it would 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. Add the following device\_filter.xml file 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.xml in the following way:

```
<?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_DE-
VICE_ATTACHED" />
        </intent-filter>

        <meta-data
            android:name="android.hardware.usb.action.USB_DEVICE_AT-
TACHED"
            android:resource="@xml/device_filter" />
        </activity>
    </application>

</manifest>

```

## Initializing 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
    TerabeeSdk.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 device-
Type) {
    }

    @Override
    public void disconnected() {
    }

    @Override
    public void permission(boolean granted) {
    }
}, TerabeeSdk.DeviceType.EVO_60M);

```

In the process of connecting you will receive callbacks: whether connection established successfully, when device gets disconnected plus results of 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 register the data receiver according to the sensor type.

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

```

private final TerabeeSdk.DataDistanceCallback mDataDistanceCallback = new Ter-
abeeSdk.DataOnePixelCallback() {
    @Override
    public void onDistanceReceived(int distance, int dataBandwidth, int data-
Speed) {
        // 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 completing of using the sensor you need to unregister the data receiver:

```
TerabeeSdk.getInstance().unregisterDataReceive(mDataDistanceCallback);
```

To receive data from a different sensor it's necessary to register a different data receiver according to the sensor type. The SDK supports the following data receivers:

DataDistanceCallback – for Evo 3M and Evo 60M sensors, 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 64px sensors, allow to receive matrix of distances.

DataThermalMatrixCallback - for Evo Thermal sensors, allow to receive matrix of temperatures.

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 above callbacks.

**Done! You are ready to use the Terabee SDK.**