Android Open Accessory Device Configuration

1. Establish a UsbManager variable

```
manager = (UsbManager) getSystemService(Context.USB_SERVICE);
```

2. Detect connected accessories
   a. Option 1: Intent filter – Automatically queries user for a compatible app to handle the intent, which grants permission until the accessory is detached, unless the app is set to launch by default.

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

<meta-data
    android:name="android.hardware.usb.action.USB_ACCESSORY_ATTACHED"
    android:resource="@xml/accessory_filter" />
```

Snippet from AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <usb-accessory vendor-id="6353" product-id="11521"/>
</resources>
```

Snippet from accessory filter.xml

   b. Option 2: Enumerate connected accessories – Explicitly request permission for accessories

```
UsbAccessory[] accessories = manager.getAccessoryList();
PendingIntent permissionIntent = PendingIntent.getBroadcast( context: this, requestCode: 0,
        new Intent( action: "com.android.example.USB_PERMISSION"), flags: 0);
for(UsbAccessory accessory : accessories) {
    manager.requestPermission(accessory, permissionIntent);
}
```

3. Open accessory and setup input/output streams

```
fileDescriptor = manager.openAccessory(accessory);
if ( fileDescriptor != null ) {
    this.accessory = accessory;
    FileDescriptor fd = fileDescriptor.getFileDescriptor();
    inputStream = new FileInputStream(fd);
    outputStream = new FileOutputStream(fd);
}
```