

SDK 接入指南

文档修改记录

序号	版本号	修改内容	修改者	评审者	修改日期
1	v1.0.0	初始版本	邱贤	林德春	2026.04.11

1. 接入包作用

提供适用于Android平台接口，使客户能够通过调用我方提供的Android API将我们的打印机接入到软件系统中。通过调用我们提供的接口，客户可以实现打印机与软件系统的无缝对接。

2. 机型支持范围

编号	机型指令集	固件版本
01	ESC	所有版本
02	CPCL	所有版本
03	TSPL	所有版本
04	ZPL	所有版本

注意：其他机器指令集或支持型号范围外固件版本的设备暂不支持。

3. SDK使用评估

3.1 操作环境

Android 版本

- Android 5.0 及以上版本

Android 设备

- Android 系统手机
- Android 系统平板
- Android 系统开发板

应用环境

- JDK：版本 1.6 及以上
- Android Studio：版本 2.3.2 及以上
- Android SDK：版本 1.6 及以上

3.2 SDK Demo 体验流程

如果您需要进行接入评估可先行体验SDK Demo，体验流程如下：

1. 打开 Android Studio 编辑器
2. 导入 SDK Demo
3. 进行编译
4. 真机编译
5. 体验 SDK Demo

4. SDK接入

4.1 开发接入指南

如果您是开发人员，以下步骤可帮助您轻松开始接入SDK：

1. **查看 SDK DEMO**：使用 Android Studio 打开 SDK DEMO，参考 SDK DEMO 中的调用示例，以了解如何正确使用SDK。
2. **查看接口文档**：查看 SDK 接口说明文档，以获取详细的接口调用说明。
3. **查阅使用说明**：在着手开发之前，请务必仔细查阅接口文档中"使用说明"，以便了解工程配置、权限配置及手机设置说明。
4. **SDK 调用流程**：接口文档样例演示部分包含了关于 Android 端的 SDK 调用流程，请务必仔细研究，这对于顺利接入非常重要。

4.2 相关注意事项

- 蓝牙相关的问题可以参考注意事项文档
- 如果您需要进一步的帮助或有任何疑问，请随时联系我们的支持团队

5. 蓝牙使用注意事项

5.1 开启蓝牙

使用蓝牙功能前，请确保设备蓝牙已开启。

5.2 开启 GPS

Android 6.0 及以上版本，蓝牙扫描需要开启 GPS 定位功能。

5.3 权限配置

在 `AndroidManifest.xml` 中配置以下权限：

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
<uses-permission android:name="android.permission.BLUETOOTH_CONNECT" />
<uses-permission android:name="android.permission.BLUETOOTH_ADVERTISE" />
<uses-permission android:name="android.permission.BLUETOOTH_SCAN" />
```

5.4 动态权限申请

在代码中动态配置权限（使用 RxPermissions）：

```
RxPermissions(this).request(
    Manifest.permission.BLUETOOTH,
    Manifest.permission.ACCESS_FINE_LOCATION,
    Manifest.permission.ACCESS_COARSE_LOCATION,
    Manifest.permission.BLUETOOTH_SCAN,
    Manifest.permission.BLUETOOTH_CONNECT
)
    .subscribe {
        if (it) {
            // 权限授予成功
        }
    }
}
```

6. 工程配置

6.1 Gradle 依赖配置

在项目的 `build.gradle` 中添加依赖：

```
implementation files('libs/UxPrint-sdk-v1.xx.xx.aar')
implementation files('libs/lzo-sdk-v1.xx.xx.aar')
```

6.2 ProGuard 混淆配置

在 `proguard-rules.pro` 中添加：

```
-keep class com.printer.sdk.** { *; }
-keep interface com.printer.sdk.** { *; }
```

7. 权限配置

根据使用的连接方式，在 `AndroidManifest.xml` 中添加相应权限：

7.1 蓝牙连接权限

```
<!-- 蓝牙基础权限 -->
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />

<!-- Android 12+ 蓝牙权限 -->
<uses-permission android:name="android.permission.BLUETOOTH_CONNECT" />
<uses-permission android:name="android.permission.BLUETOOTH_SCAN" />

<!-- 位置权限（蓝牙扫描需要） -->
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

7.2 WiFi连接权限

```
<!-- WiFi网络权限 -->
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
```

7.3 USB连接权限

```
<!-- USB权限 -->
<uses-feature android:name="android.hardware.usb.host" />
```

USB设备过滤器配置（可选）：

```
<activity android:name=".UsbPermissionActivity">
    <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>
```

8. SDK初始化

在 Application 中初始化 SDK:

8.1 基础初始化

```
class MyApplication : Application() {  
    override fun onCreate() {  
        super.onCreate()  
        PrinterSDK.initialize(this)  
    }  
}
```

8.2 带日志功能的初始化

```
class MyApplication : Application() {  
    override fun onCreate() {  
        super.onCreate()  
        PrinterSDK.initialize(  
            applicationContext = this,  
            enableIoLog = true,  
            ioLogConfig = PrinterIoLogConfig(  
                enableReadLog = true,  
                enableWriteLog = true,  
                logAsHexString = true  
            )  
        )  
    }  
}
```

日志文件路径:

- 文件名: SDK_log.txt
- 优先路径: /storage/emulated/0/Android/data/<包名>/files/Documents/SDK_log.txt
- 备用路径: /data/data/<包名>/files/SDK_log.txt

9. 快速开始

9.1 蓝牙连接示例

```
// 创建蓝牙连接  
val connection = BluetoothConnection()  
  
// 连接打印机  
lifecycleScope.launch {  
    val result = connection.connect(  
        ConnectionParams.Bluetooth(address = "00:11:22:33:44:55")  
    )  
}
```

```

        when (result) {
            is Result.Success -> {
                // 连接成功, 发送数据
                connection.write(printData)
            }
            is Result.Failure -> {
                Log.e("Bluetooth", "Connection failed: ${result.error.message}")
            }
        }
    }
}

```

9.2 WiFi连接示例

```

// 创建WiFi连接
val connection = WiFiConnection()

// 连接打印机
lifecycleScope.launch {
    val result = connection.connect(
        ConnectionParams.WiFi(
            ip = "192.168.1.100",
            port = 9100,
            timeout = 5000
        )
    )
    when (result) {
        is Result.Success -> {
            // 连接成功, 发送数据
            connection.write(printData)
        }
        is Result.Failure -> {
            Log.e("WiFi", "Connection failed: ${result.error.message}")
        }
    }
}
}

```

9.3 USB连接示例

```

// 获取USB设备
val usbManager = getSystemService(Context.USB_SERVICE) as UsbManager
val deviceList = usbManager.deviceList

if (deviceList.isNotEmpty()) {
    val device = deviceList.values.first()

    // 请求USB权限
    val permissionIntent = PendingIntent.getBroadcast(

```

```
        this, 0, Intent(ACTION_USB_PERMISSION), PendingIntent.FLAG_IMMUTABLE
    )
    usbManager.requestPermission(device, permissionIntent)

    // 创建USB连接
    val connection = USBConnection(this)

    // 连接打印机
    lifecycleScope.launch {
        val result = connection.connect(ConnectionParams.USB(device))
        when (result) {
            is Result.Success -> {
                // 连接成功, 发送数据
                connection.write(printData)
            }
            is Result.Failure -> {
                Log.e("USB", "Connection failed: ${result.error.message}")
            }
        }
    }
}
```

10. 常见问题

Q1: 蓝牙扫描不到设备?

解决方案:

1. 确保蓝牙已开启
2. Android 6.0+ 确保GPS已开启
3. 确保已授予位置权限
4. 确保已授予蓝牙相关权限

Q2: 蓝牙连接失败?

解决方案:

1. 确保打印机已开机且电量充足
2. 确保打印机未被其他设备连接
3. 尝试重新配对蓝牙设备
4. 检查蓝牙地址是否正确

Q3: WiFi连接超时?

解决方案:

1. 确保手机和打印机在同一网络
2. 检查打印机IP地址是否正确

3. 检查端口是否正确（默认9100）
4. 尝试ping打印机IP地址

Q4: USB连接无反应?

解决方案:

1. 确保USB线缆连接正常
2. 确保已授予USB权限
3. 检查设备是否支持USB Host模式

11. 技术支持

如果您需要进一步的帮助或有任何疑问，请随时联系我们的支持团队，他们将竭诚为您提供支持和指导。