

# CH37x Android JAR 库接口说明

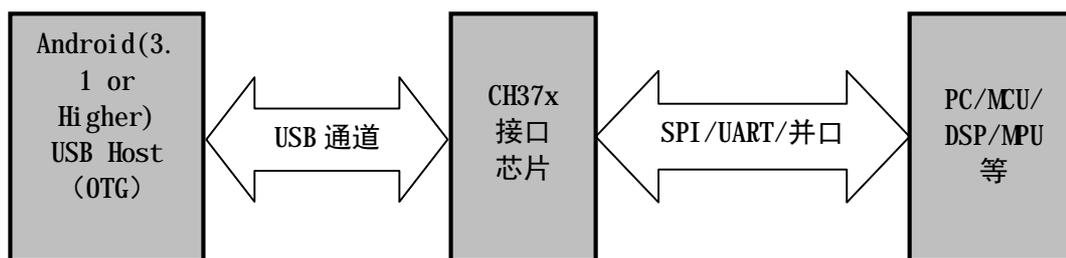
版本： 1.1

## 简介

CH37x 系列芯片是 USB 总线通用接口芯片，主要包含 CH372/CH375、CH374、CH376/CH378 等。

本文档主要介绍 CH37x USB Slave(以下简称 CH37xUSBDev) 模式下设备操作的 Java Driver 库, 以及 Android 下如何使用 APK 操作 CH37x 实现 Android USB 接口通讯。该驱动基于 Android USB Host 协议完成, 用户可调用相关的接口 API 实现与 Android USB Host 的通讯。

Android Host、CH37x 接口芯片、外部设备的关系如下图。



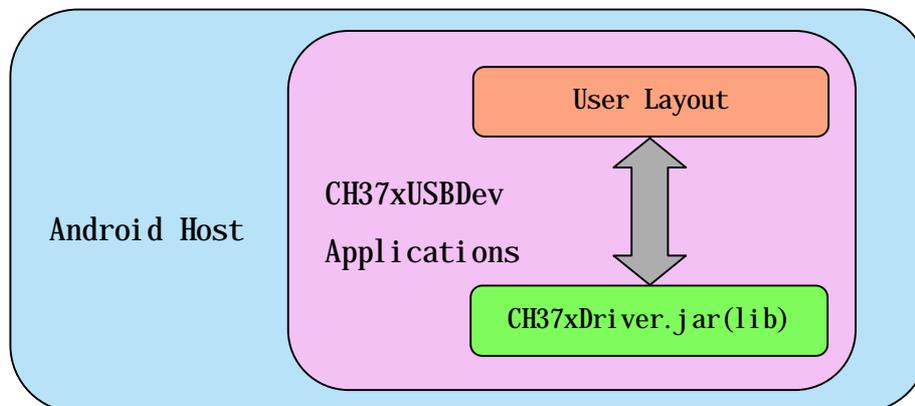
CH37xUSBDev 提供的接口是针对 USB 控制端点、批量端点进行的数据通讯, Android 端需要满足以下两个条件方可使用:

- 1、需要 Android 3.1 及以上系统版本
- 2、Android 设备具有 USB Host 或 OTG 接口

本文档将会重点说明 Android USB Host 与 Device 的通讯接口 API 以及演示程序的操作说明。关于 Android USB Host 协议说明, 可以参考 Google 官方文档。

## 1、Android Host

本文档所描述的演示程序皆是基于 Android 3.1 及以上版本 Android 系统下编写的。应用程序的启动参数是定义在 device\_filter.xml 文件中的 product-id 和 vendor-id。基于 CH37xUSBDev 开发的 Android 应用程序主要分为两个部分，如下图：



## 2、CH37xUSBDev API

### 2.1 概述

针对 CH37xUSBDev 的操作提供了 OpenDevice、WriteData 和 ReadData 等方法以及 WriteTimeoutMillis 和 ReadTimeoutMillis 属性，实现与 CH37xUSBDev 功能模块的通讯。同时提供 CloseDevice 接口来关闭 USB Device，isConnected 接口来判断设备是否连接。

### 2.2 User-Layout

OpenDevice : 打开 USB 设备

函数原型 : `boolean OpenDevice()`

返回 true 则表示打开成功，返回 false 则表示打开失败

WriteData : 向 CH37xUSBDev 的批量 OUT 端点写数据

函数原型 : `public int WriteData (byte[] Buf, int Length)`

Buf : 发送缓冲区

Length : 发送的字节数 (0<length<=255)

返回值为成功写入的字节数

ReadData : 读取 CH37xUSBDev 的批量 IN 端点数据

函数原型 : `public int ReadData (byte[] Data, int Length)`

Data : 接收缓冲区

Length : 读取的字节数

返回实际读取的字节数

CloseDevice : 关闭设备

函数原型 : `public void CloseDevice()`

GetVID : 获取设备 VID

函数原型 : `public int GetVID()`

返回设备的 VID 值

**GetPID** : 获取设备 PID  
函数原型 : `public int GetPID()`  
返回设备的 PID 值

**isConnected** : 判断设备是否已经连接到 Android 系统  
函数原型 : `public boolean isConnected()`  
返回 true 表示设备已经连接, false 表示设备未连接

除了上述提供的接口 API, 用户还可以根据自己的设备来设置读写超时时间:

函数原型 : `public boolean SetTimeOut(int WriteTimeOut, int ReadTimeOut)`  
**WriteTimeOut**: 设置写超时时间, 默认为 1000ms  
**ReadTimeOut** : 设置读超时时间, 默认为 10000ms

### 3、Demo 程序操作说明

用户在有 OTG 接口或 Host 接口的 Android 设备上安装我司提供的测试软件 (即 CH37xDriverDemo.apk)。若是第一次安装、使用该软件, 则在插入 CH37xUSBDev 功能模块以后, 系统会自动弹出权限请求窗口, 点击 “Use by default for this USB device”, 选择确定操作以后, 再使用该模块就不会弹出这个 permissions 请求窗口; 如果不选择 “Use by default for this USB device” 而直接确定操作, 则下次使用该模块时将还会弹出 permissions 请求窗口。打开软件后, 点击 “打开” 按钮后, 如果打开成功会有提示, 界面会显示设备的 VID, PID, 成功打开设备后即可发送接收数据了。

### 4、其他

用户可根据我司提供的 Demo 程序以及开放的函数接口说明, 开发相关产品。若在使用我司提供的测试软件过程中出现问题, 可以查看我司提供的《Android 免驱常见问题及解决手册》。