

**CH34x 系列芯片串口功能
Android 应用程序开发手册**

版本： V1.01

<http://wch.cn>

CH34x 系列芯片串口功能 Android 应用程序开发手册

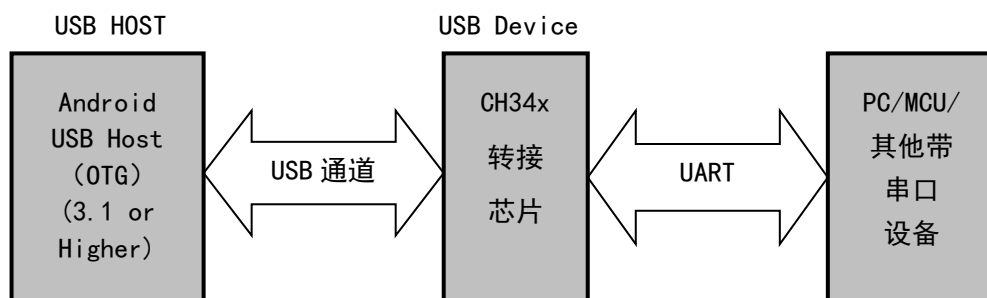
版本： 1.01

简介

CH34x 系列芯片是 USB 总线的转接芯片，主要包含 CH340、CH341、CH345，通过 USB 总线提供异步串口、打印口、并口、MIDI 以及常用的 2 线和 4 线等同步串行接口。

本文档主要介绍其中 CH340/CH341 的 USB (SLAVE) 转异步串口功能（以下简称 CH34x UART），以及 Android 下如何使用 APK 操作 CH34x 实现串口通讯。该功能基于 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、USB Device、串口设备三者关系如下图。



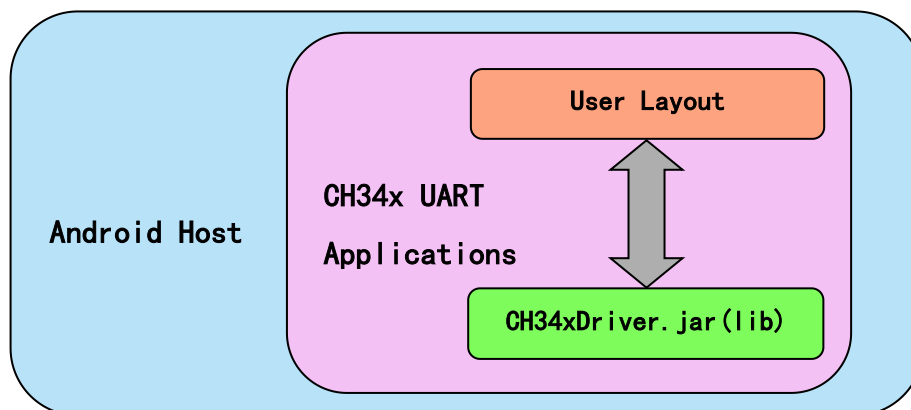
CH34x 串口提供的 Android 接口需要给予 Android 3.1 及以上版本系统，CH34x 串口 Android 驱动条件：

- 1、需要给予 Android 3.1 及以上版本系统；
- 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 以及测试程序的操作说明。关于 Android USB Host 协议说明，可以参考 Google 官方文档。

1、Android Host

本文档所描述的例子程序皆是在 Android 3.1 及以上版本系统下编写的。Android 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 CH34x UART 开发的 Android 应用程序主要分为两个部分，如下图：



2、Android USB To Uart Demo

2.1 UART

针对 CH34x UART 的操作提供了 EnumateDevice、OpenDevice、UartInit、SetConfig、WriteData 和 ReadData 方法以及 WriteTimeOutMillis 和 ReadTimeOutMillis 属性，实现与 CH34x UART 功能模块的通讯。同时提供 CloseDevice 接口来关闭 UART Device，isConnected 接口来判断设备是否连接。

2.2 UART User-Layout

EnumerateDevice: 枚举 CH34x 设备。

函数原型: `public UsbDevice EnumerateDevice()`

返回枚举到的 CH34x 的设备，若无设备则返回 NULL。

OpenDevice: 打开 CH34x 设备。

函数原型: `public synchronized void OpenDevice(UsbDevice mDevice)`

mDevice : 需要打开的 CH34x 设备

打开枚举 USB 设备。

UartInit: 设置初始化 CH34x 芯片。

函数原型: `public boolean UartInit()`

若初始化失败，则返回 false，成功返回 true。

SetConfig: 设置 UART 接口的波特率、数据位、停止位、奇偶校验位以及流控。

函数原型 : `public boolean SetConfig(int baudRate, byte dataBit, byte stopBit, byte parity, byte flowControl)`

baudRate : 波特率: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800, 921600, 默认: 9600。

dataBits : 5 个数据位、6 个数据位、7 个数据位、8 个数据位, 默认: 8 个数据位。

stopBits : 0: 1 个停止位, 1: 2 个停止位, 默认: 1 个停止位。

parity : 0: none, 1: add, 2: even, 3: mark 和 4: space, 默认: none。

flowControl : 0: none, 1: cts/rts, 默认: none。

WriteData: 向 USB Device 写设备。

函数原型 : `public int WriteData(byte[] buf, int length)`

Buf : 发送缓冲区;

Length : 发送的字节数, ($0 < \text{length} \leq 255$ bytes)

返回值为写成功的字节数。

ReadData: 读取数据。

函数原型: `public int ReadData(char[] data, int length)`

Data : 接收缓冲区;

Length : 读取的字节数, 默认设置为 64 字节, 用户也可以根据需要修改此值;

返回实际读取的字节数。

函数原型: `public int ReadData(byte[] data, int length)`

Data : 接收缓冲区;

Length : 读取的字节数, 默认设置为 64 字节, 用户也可以根据需要修改此值;

返回实际读取的字节数。

CloseDevice: 关闭串口。

函数原型 : `public synchronized void CloseDevice()`

isConnected: 判断设备是否已经连接到 Android 系统。

函数原型 : `public boolean isConnected()`

返回为 `false` 时表示未连入系统或者非 CH34x UART 功能芯片, 会导致读写操作的失败。

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 10000ms;

ReadTimeOut : 设置读超时时间, 默认为 10000ms。

3、测试软件操作说明

用户在有 OTG 接口的 Android 设备上安装我司提供的测试软件 (即 WchUsbDriver.apk)。若是第一次安装、使用该软件, 则在插入 CH34x UART 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。

在软件打开过程中, 执行 ResumeUsbList 完成对 USB 设备的枚举、打开设备、获取设备资源信息等步骤 (或可使用 EnumerateDevice 后 OpenDevice 打开设备), 然后用户根据需要设置波特率、数据位、停止位、奇偶校验位以及流控等参数, 点击配置按键, 即完成 UART 的配置, 之后就可以执行读写操作。