

接口函数说明

版本：1.3

<http://wch.cn>

概述

本文档是针对于CH934X的安卓接口库的说明文档

1、getInstance

```
public static CH934XManager getInstance()
```

用于获取全局唯一实例

返回	返回全局唯一实例
----	----------

2、init

```
public void init(android.app.Application application)
```

初始化上下文，注册动态广播监听设备状态变化

参数	application - 全局上下文
----	---------------------

3、enumDevice

```
public java.util.ArrayList<android.hardware.usb.UsbDevice> enumDevice()
```

```
throws cn.wch.ch934xlib.exception.UartLibException
```

枚举当前的 CH934X 设备

返回	设备列表
抛出	cn.wch.ch934xlib.exception.UartLibException

4、getChipType

```
public cn.wch.ch934xlib.chip.ChipType getChipType(@NonNull
```

```
android.hardware.usb.UsbDevice usbDevice)
```

根据 UsbDevice 获取芯片类型

参数	usbDevice - USB 设备
返回	如果为 null 则表示参数 USB 设备并不是 CH934X

5、openDevice

```
public boolean openDevice(@NonNull
```

```
android.hardware.usb.UsbDevice usbDevice)
```

```
throws cn.wch.ch934xlib.exception.UartLibException,
```

cn.wch.ch934xlib.exception.NoPermissionException,

cn.wch.ch934xlib.exception.ChipException

打开设备

参数	usbDevice - USB 设备
返回	true 成功; false 失败
抛出	cn.wch.ch934xlib.exception.UartLibException cn.wch.ch934xlib.exception.NoPermissionException cn.wch.ch934xlib.exception.ChipException

6、requestPermission

```
public void requestPermission(@NonNull  
                             android.content.Context context,  
                             @NonNull  
                             android.hardware.usb.UsbDevice usbDevice)  
    throws cn.wch.ch934xlib.exception.UartLibException
```

请求设备的打开权限

参数	context - 上下文 usbDevice - USB 设备
抛出	cn.wch.ch934xlib.exception.UartLibException

7、setUsbStateListener

```
public void setUsbStateListener(@NonNull  
                                cn.wch.ch934xlib.callback.IUsbStateChange  
                                usbStateListener)
```

监听设备的状态变化

参数	usbStateListener - 设备状态监听回调
----	-----------------------------

8、getSerialCount

```
public int getSerialCount(@NonNull  
                           android.hardware.usb.UsbDevice usbDevice)
```

获取设备的串口数目

参数	usbDevice - USB 设备
返回	返回串口数目; 如果为负, 说明获取失败

9、setSerialParameter

```
public boolean setSerialParameter(@NonNull  
                                  android.hardware.usb.UsbDevice usbDevice,  
                                  int serialNumber,
```

```
        int baud,  
        int dataBit,  
        int stopBit,  
        int parityBit,  
        boolean flow)  
    throws cn.wch.ch934xlib.exception.UartLibException,  
           cn.wch.ch934xlib.exception.ChipException
```

设置串口参数

参数	usbDevice - USB 设备 serialNumber - 串口号 baud - 波特率 dataBit - 数据位 stopBit - 停止位 parityBit - 校验位 flow - 流控
返回	true 设置成功;false 设置失败
抛出	cn.wch.ch934xlib.exception.UartLibException cn.wch.ch934xlib.exception.ChipException

10、writeData

```
public int writeData(@NonNull  
    android.hardware.usb.UsbDevice usbDevice,  
    int serialNumber,  
    byte[] data,  
    int length,  
    int timeout)  
    throws cn.wch.ch934xlib.exception.UartLibException,  
           cn.wch.ch934xlib.exception.ChipException
```

发送数据

参数	usbDevice - USB 设备 serialNumber - 串口号 data - 待发送的数据 length - 待发送的数据的长度 timeout - 超时时间
返回	发送成功的数据的长度
抛出	cn.wch.ch934xlib.exception.UartLibException cn.wch.ch934xlib.exception.ChipException

11、readData

```
public byte[] readData(@NonNull  
    android.hardware.usb.UsbDevice usbDevice,
```

```
int serialNumber)
```

```
throws cn.wch.ch934xlib.exception.ChipException
```

读取数据

参数	usbDevice - USB 设备 serialNumber - 串口号
返回	读取到的数据
抛出	cn.wch.ch934xlib.exception.ChipException

12、isConnected

```
public boolean isConnected(@NonNull
```

```
android.hardware.usb.UsbDevice usbDevice)
```

设备是否已经被打开

参数	usbDevice - USB 设备
返回	true 已经被打开;false 没有被打开

13、getConnectedDevices

```
public java.util.ArrayList<android.hardware.usb.UsbDevice>  
getConnectedDevices()
```

获取当前已经被打开的设备列表

返回	已经被打开的设备列表
----	------------

14、disconnect

```
public void disconnect(@NonNull
```

```
android.hardware.usb.UsbDevice usbDevice)
```

断开连接

参数	usbDevice - USB 设备
----	--------------------

15、close

```
public void close(@NonNull Context context)
```

释放资源。断开所有连接设备, 注销广播

参数	context - 上下文
----	---------------

16、getCurrentMode

```
public Mode getCurrentMode(UsbDevice usbDevice, int serialNumber)
```

获取当前串口的模式。初始默认状态为普通模式

参数	usbDevice - USB 设备
----	--------------------

	serialNumber-串口序号
返回	Mode.NORMAL 普通模式; Mode.HARDFLOW 硬件流控模式; Mode.GPIO GPIO 模式;
抛出	cn.wch.ch934xlib.exception.UartLibException

17、enableGPIO

```
public boolean enableGPIO(@NonNull UsbDevice usbDevice, int  
serialNumber, boolean enabled)
```

使能串口的 GPIO 功能，如果当前模式为硬件流控模式，需要先退出。

参数	usbDevice - USB 设备 serialNumber-串口序号 enabled-打开或者关闭
返回	true 操作成功;false 操作失败
抛出	cn.wch.ch934xlib.exception.UartLibException

18、setGPIONDir

```
public boolean setGPIONDir(@NonNull UsbDevice usbDevice, int gpioNumber,  
@NonNull GPIO_DIR dir)
```

设置 GPIO 口的方向，成功后会同步至缓存

参数	usbDevice - USB 设备 gpioNumber -GPIO 序号 dir -方向
返回	true 操作成功;false 操作失败
抛出	cn.wch.ch934xlib.exception.UartLibException

19、queryGPIONDirFromCache

```
public GPIO_DIR queryGPIONDirFromCache(@NonNull UsbDevice usbDevice, int  
gpioNumber)
```

从缓存中获取某个 GPIO 的方向。初始每个 GPIO 初始默认方向是 GPIO_DIR.IN

参数	usbDevice - USB 设备 gpioNumber -GPIO 序号
返回	GPIO_DIR.IN IN 方向;GPIO_DIR.OUT OUT 方向
抛出	cn.wch.ch934xlib.exception.UartLibException

20、setGPIOValue

```
public boolean setGPIOValue(@NonNull UsbDevice usbDevice, int gpioNumber,  
@NonNull GPIO_VALUE value)
```

设置 GPIO 口的值，成功后会同步至缓存

参数	usbDevice - USB 设备 gpioNumber -GPIO 序号 value -值
返回	true 操作成功;false 操作失败
抛出	cn.wch.ch934xlib.exception.UartLibException

21、getGPIOValue

```
public boolean getGPIOValue(@NonNull UsbDevice usbDevice)
```

从硬件获取 GPIO 值,成功后会刷新缓存

参数	usbDevice - USB 设备
返回	true 操作成功;false 操作失败
抛出	cn.wch.ch934xlib.exception.UartLibException

22、queryGPIOValueFromCache

```
public GPIO_VALUE queryGPIOValueFromCache(@NonNull UsbDevice usbDevice, int  
gpioNumber)
```

从缓存中获取某个 GPIO 值。使用此方法前需要先成功使用 getGPIOValue() 方法,刷新缓存。初始每个 GPIO 初始默认值是 GPIO_VALUE. LOW

参数	usbDevice - USB 设备 gpioNumber -GPIO 序号
返回	GPIO_VALUE. HIGH 高电平;GPIO_VALUE. LOW 低电平
抛出	cn.wch.ch934xlib.exception.UartLibException