CH34x serial port Android program development instructions 1  
CH34x series chip serial port function Android program development instructions  
Version: 1.3  
Introduction  
The CH34x series chip is a USB bus adapter chip, mainly including CH340, CH341, CH345, and provides different via USB bus.  
Step serial port, print port, parallel port, MIDI and common 2-wire and 4-wire interfaces.  
This document mainly introduces the USB to asynchronous serial port function of CH340/CH341 (hereinafter referred to as CH34x UART) and Android.  
How to use the APK to operate the CH34x for serial communication. This feature is based on the Android USB Host protocol and can be called by the user.  
The interface API is implemented to communicate with Android devices.  
The relationship between Android Host, USB Device and serial device is as follows.  
The Android interface provided by the CH34x serial port needs to be based on Android 3.1 and above, using CH34x serial port Android  
Driving conditions:  
1, need to be based on Android 3.1 and above system  
2. Android device has USB Host or OTG interface  
This document will focus on the communication interface API of the Android USB Host and Device and the operating instructions of the test program.  
For instructions on the Android USB Host protocol, you can refer to the official Google documentation.  
Android  
USB Host  
(OTG)  
(3.1 or  
Higher)  
USB channel  
CH34x  
Transfer  
chip  
UART  
PC/MCU/  
Other belt  
Serial port  
device  
USB HOST USB Device  
CH34x serial port Android program development instructions 2  
1, Android Host  
The sample programs described in this document are all written under Android 3.1 and above. Android application  
The dynamic parameters are the product-id and vendor-id defined in the device\_filter.xml file. Based on the CH34x UART  
The Android application is mainly divided into two parts, as shown below:  
2, Android USB To Uart Demo  
2.1 UART  
EnumerateDevice, OpenDevice, UartInit, SetConfig, WriteData are provided for the operation of the CH34x UART.  
And the ReadData method and the WriteTimeOutMillis and ReadTimeOutMillis properties, implemented with the CH34x UART function module  
Block communication. At the same time, the CloseDevice interface is provided to close the UART Device and the isConnected interface to determine whether the device is connected.  
Note the following points when programming:  
λ Create the object of the CH34xUARTDriver class provided by our company under the Application class to ensure that there are multiple applications.  
Serial port can be sent and received when the activity is switched.  
λ The operation flow is: ResumeUsbList (or OpenDevice after EnumerateDevice), UartInit,  
SetConfig, after the above process is executed, the serial port can be sent and received.  
For specific implementation, please refer to the Demo program provided by our company.  
2.2 UART User-Layout  
EnumerateDevice: Enumerate CH34x devices  
Prototype: public UsbDevice EnumerateDevice()  
Returns the device of the CH34x enumerated, or null if there is no device  
OpenDevice: Turn on the CH34x device  
Prototype: public void OpenDevice(UsbDevice mDevice)  
mDevice : CH34x device that needs to be turned on  
ResumeUsbList: Enumerate and open the CH34x device. This function contains EnumerateDevice, OpenDevice operation.  
Prototype: public int ResumeUsbList()  
Return 0 to succeed, otherwise fail  
UartInit: Set initialization CH34x chip  
Prototype: public boolean UartInit()  
If the initialization fails, it returns false and returns true.  
Android Host  
CH34x UART  
Applications  
User Layout  
CH34xDriver.jar(lib)  
CH34x serial port Android program development instructions 3  
SetConfig: Set the baud rate, data bits, stop bits, parity bits, and flow control of the UART interface.  
Prototype: public boolean SetConfig(int baudRate, byte dataBit, byte stopBit, byte  
Parity, byte flowControl)  
baudRate: baud rate: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200,  
230400, 460800, 921600, default: 9600  
dataBits: 5 data bits, 6 data bits, 7 data bits, 8 data bits, default: 8 data bits  
stopBits : 0: 1 stop bits, 1: 2 stop bits, default: 1 stop bit  
Parity :0:none,1:add,2:even,3:mark and 4:space, default: none  
flowControl :0:none,1:cts/rts,default:none  
If the setting fails, it returns false and returns true.  
WriteData: send data  
Prototype: public int WriteData(byte[] buf, int length)  
Buf : send buffer  
Length : the number of bytes sent  
The return value is the number of bytes successfully written.  
ReadData: read data  
Prototype: public int ReadData(char[] data, int length)  
Data : receive buffer, data type is char  
Length : the number of bytes read  
Returns the number of bytes actually read  
Prototype: public int ReadData(byte[] data, int length)  
Data : receive buffer  
Length : the number of bytes read  
Returns the number of bytes actually read  
CloseDevice: Close the serial port.  
Prototype: public void CloseDevice()  
isConnected: Determines if the device is connected to the Android system  
Prototype: public boolean isConnected()  
Returning to false means the device is not connected to the system, true means the device is connected  
In addition to the interface APIs provided above, users can also set read and write timeouts based on their own devices:  
Prototype: public boolean SetTimeOut(int WriteTimeOut, int ReadTimeOut)  
WriteTimeOut: set the write timeout period, the default is 10000ms  
ReadTimeOut : Set the read timeout period, the default is 10000ms  
3, test software operation instructions  
The user installs the test software provided by our company (ie CH34xUARTDemo.apk) on an Android device with an OTG interface. if  
The first time you install and use the software, the system will automatically pop up the permission request window after inserting the CH34x UART function module.  
Click "Use by default for this USB device", select the operation, and then use this module will not pop up this  
Permissions request window; if you do not select "Use by default for this USB device" and directly determine the operation, soft  
The item will pop up a no permission dialog and request to exit.  
After entering the software, the ResumeUsbList will be executed during the process of opening the device to complete the enumeration of the USB device, open the device, and obtain the device.  
Steps such as resource information (or OpenDevice can be turned on after EnumerateDevice), then the user sets the wave as needed  
Parameters such as rate, data bit, stop bit, parity bit, and flow control, click the configuration button to complete the UART configuration, and then  
CH34x serial port Android program development instructions 4  
Read and write operations can be performed.