Android USB SDK
User Guide

Introduction:

This document describes the prerequisites for using "android-usb-sdk" and "sdk package in the description of the relevant classes and interfaces".Make sure that this device meets the prerequisites mentioned in this document, as well as when programming in the documentation for the sdk package of the description for code editing.

一. Prerequisites:

- Android supports a variety of USB peripherals, through two modes to support: USB peripheral mode and USB host mode.

Android android3.1 and above versions only support android USB host mode.

When the android device is in host mode, it plays the USB host role and powers the bus.This USB SDK Package that is based on the android host model to develop

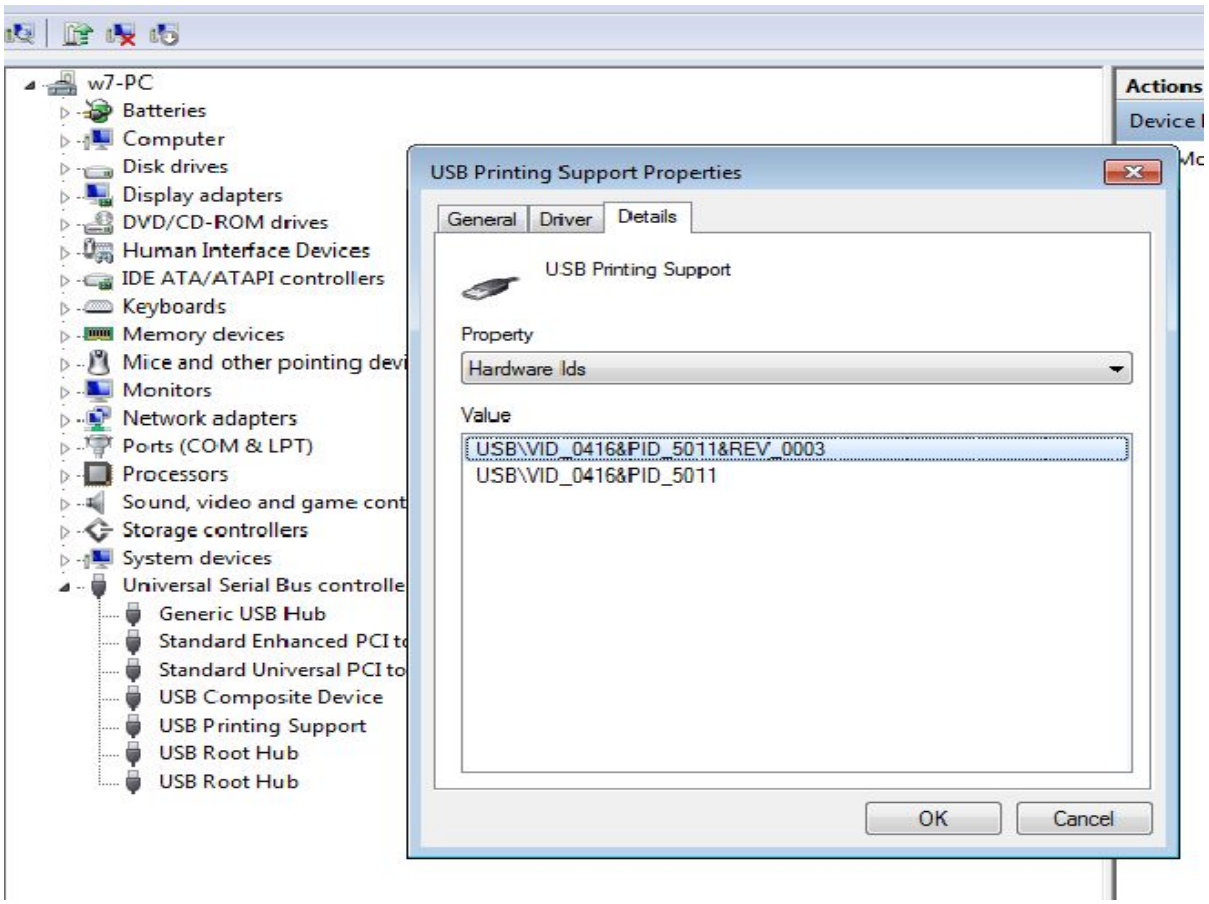
This USB SDK Package is based on the android host model to develop android devices.

- All USB devices have a vendor ID (Vendor ID) and a product identification number (Product ID), the host through a different Vendor ID and Product ID to distinguish between different devices.

The Vendor ID and Product ID can be obtained in the following ways:

- **Windows System :**

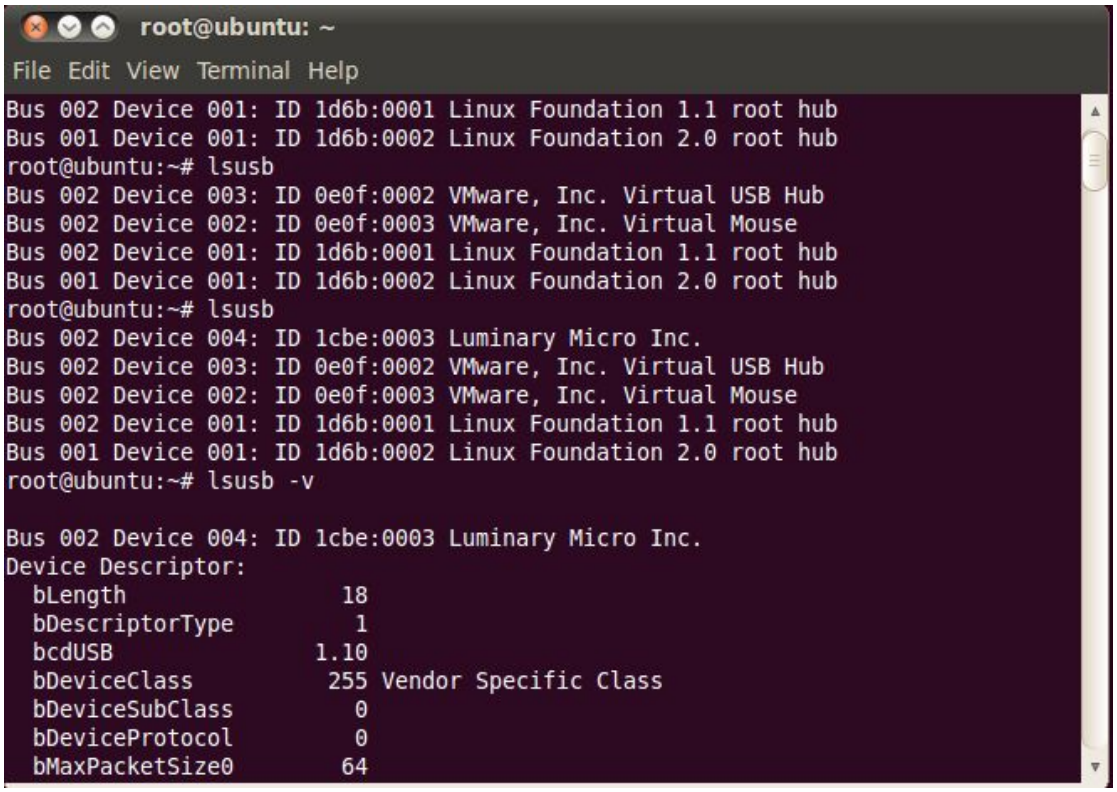
Connect with the USB printer, install the printer driver, click on "My Computer - Management - Device Manager - Universal Serial Bus Controller - USB Printing Support - Properties - Details ", as shown:



As above image show the info "**USB\VID_0416&PID_5011&REV_0003**", The **Vendor ID is** 0x0416, and the **Product ID is** 0x5011

- **Linux System ：**

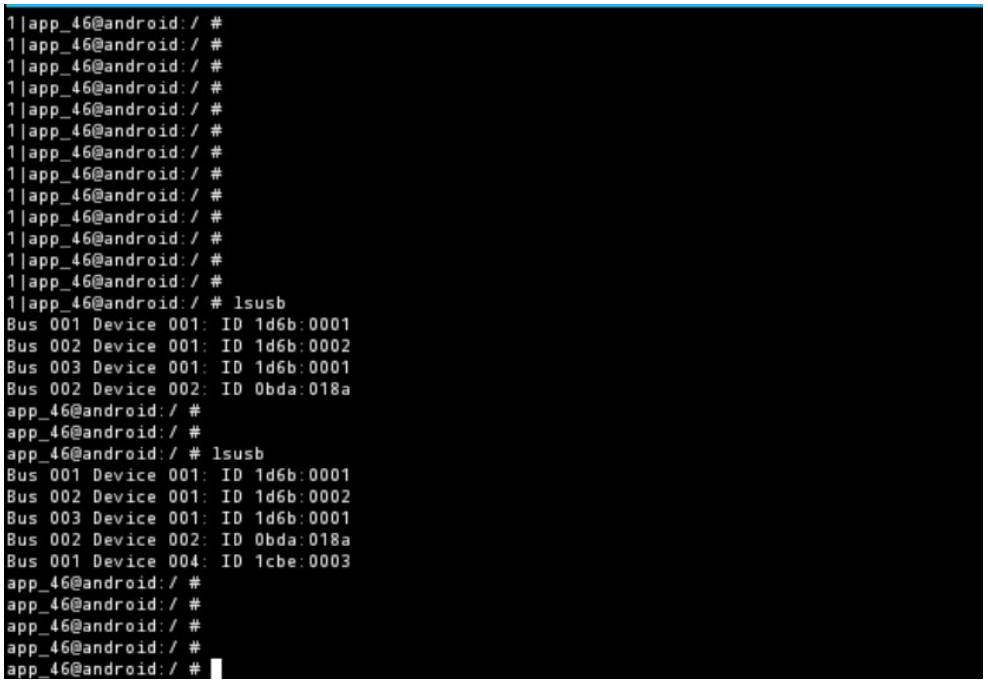
Open the terminal in Linux, enter into “ Isusb” before insert the device, after insert the device, then enter into “Isusb” again, as shown:



The message “**Bus 002 Device 004: ID 1cbe:0003 Luminary Micro Inc**”appears after inserting the device in the figure， where Vendor ID is **0x1CBE**, Product ID is **0x0003**; You can also enter "**lsusb-v**" to view the USB Device details;

- **Android System：**

In the android device, download the HyperTerminal, open the HyperTerminal, do not connect with USB printer, enter a "lsusb" Command, then connect the USB printer, and enter a "lsusb" command again to compare the return information of these two commands, as shown:



Above image shown that before and after connected with the USB printer, the system returns more information in this message "Bus 001 Device 004: ID1cbe: 0003 ", where Vendor ID is 0x1CBE and Product ID is 0x0003

二， Overview:

- This SDK package for the android application to connect to the printer via USB to print, USB-related operations；

- com.zj.usbsdk

UsbController

java.lang.Object
  └com.zj.usbsdk.UsbController

The UsbController class is a USB-related class of operations (getting a USB device, getting a list of connected USB devices, getting usb.

- com.zj.btsdk

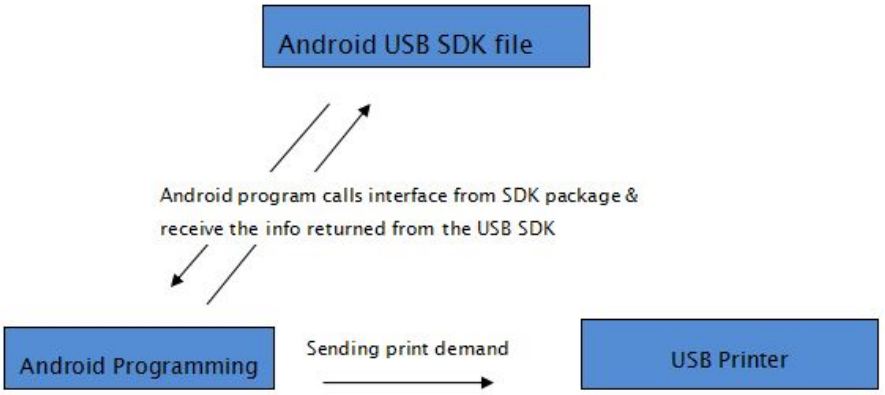
PrintPic

java.lang.Object
  └com.zj.btsdk.PrintPic

PrintPic for the image data processing class (the image can be decomposed into the printer can identify the data stream)

三． Schematic:

- Android application calls android USB SDK package in the corresponding interface, such as access to USB device instance, request USB device access and other operations connected USB printer, and print data sent to the USB printer for Data printing.



四． Interface description:

- UsbController

✧ Description

Field	Description
USB_CONNECTED	Message will be sent to APP. indication USB device connected

✧ Interface description

- Constructors:

public UsbController(Activity parentActivity,Handler handler)

Function Description:

procedure Setup:

Parameter description:

Handler: Inserts a Hander instance that returns a message from the SDk package to the android application

- Ordinary functions:

a. getDev

Prototype: **public synchronized** UsbDevice getDev(**int** vid, **int** pid)

Function Description:

Returns a UsbDevice instance based on the USB device's Vendor ID and Product ID

Parameter description:

Vid: Vendor ID of the USB device

Pid: Product ID of the usb device

Return Value Description:

Returns a U sbDevice instance successfully

Returns null on failure

b. isHasPermission

Prototype : **public synchronized boolean** isHasPermission(UsbDevice dev)

Function Description:

Determine whether to have access to the specified USB device permissions.

Parameter description:

dev: Specifies the USB device

Return Value Description:

True: Already has access to visit this device

False: You do not have permission to access this USB device, you need to request access rights for the USB device.

c. getPermission

Prototype: **public synchronized void** getPermission(UsbDevice dev)

Function Description:

Request access to the USB device

Parameter description:

Specifies the USB device

d. sendMsg

Prototype: **public synchronized void** sendMsg(String msg,String charset,UsbDevice dev)

Function Description:

Converts the string information to a charset encoding (eg "GBK") and sends it to a USB printer for data printing.

Such as the large amount of data transmission, you can create a thread for transmission, so as to avoid interface block.

Parameter description:

msg: Print information string Charset:
Character set encoding (eg "GBK")
dev: Identifies the USB printing
device.

e. sendByte

Prototype: **public synchronized void** sendByte(**byte**[] bits,UsbDevice dev)

Function Description:

Send byte stream data to a USB printer (this function can be used to send printer control instructions) such as the amount of data transferred larger, you can create a thread to transfer, so as to avoid interface block.

Parameter description:

Bits: Byte stream data
Dev: Specifies the USB device

f. cutPaper

Prototype: **public synchronized void** cutPaper(UsbDevice dev,**int** n)

Function Description:

Cut in default mode (half-cut)

Parameter description:

dev:usb device
n: the length of the blank to cut and at the end of the print text, the range of $0 \leq n \leq 255$;

g. catPaperByMode

Prototype: **public synchronized void** catPaperByMode(UsbDevice dev,**int** mode)

Function Description:

Cut in full / half cut mode

Parameter description:

dev:usb device
mode: 0 -- full cut , 1 -- half cut

h. openCashBox

Prototype: **public synchronized void** openCashBox(UsbDevice dev)

Function Description:

Generates a pulse to
open the cash drawer

Parameter description:

dev:USB printer device

i. defaultBuzzer

Prototype: **public synchronized void** defaultBuzzer(UsbDevice dev)

Function Description:

The buzzer defaults to 4 beeps and the beep time is 1 * 50 milliseconds

Parameter description:

dev : USB printer device

j. buzzer

Prototype: **public synchronized void** buzzer(UsbDevice dev,**int** n,**int** time)

Function Description:

n : 1<=n<=9

time: 1<=t<=9

dev: USB device

Parameter description:

dev: USB printer device

k. setBuzzerMode

Prototype: **public synchronized void** setBuzzerMode(UsbDevice dev,**int** n,**int** time,**int** mode)

Function Description:

Set buzzer beep mode

Parameter description:

dev: usb device

n : (1<=n<=20)，refers to the number of alarm light flashes or buzzer calls.

time: (1<=t<=20)，refers to the warning light flashes the interval time is t * 50 ms or the buzzer beats the interval time is (t × 50) milliseconds.

mode: When the mode = 1, the buzzer beeps; when the mode = 2, the alarm light flashes; when the mode = 1, the buzzer will not beep.

L. getUsbList

Prototype: **public synchronized** HashMap<String, UsbDevice>getUsbList()

Function Description:

Returns to the list of connected USB devices

四. Demo Program

Please refer to PrintDemo for bluetooth printer for Details；

Development tool ： Android Developer Tools（Build: v21.0.1-543035）

The testing information of Android device：