

RK USB Application Note

版本	日期	描述	作者	审核
V1.0	2010-12-18	USB 子系统应用文档	杨凯	
V1.1	2011-06-08	添加 OTG 支持, 添加 FAQ	杨凯	
V2.0	2012-11-03	Update for android4.0	杨凯	

目录

概述	3
一 硬件电路及信号说明.....	4
1.1 USB 2.0 控制器相关信号	4
1.2 USB 控制器信号使用	5
二 KERNEL 模块配置说明	6
2.1 USB HOST 相关配置	6
2.1.1 <i>Mass Storage Class (MSC)</i>	7
2.1.2 <i>USB Serial Converter</i>	7
2.1.3 <i>USB 鼠标键盘等HID 设备</i>	8
2.1.4 <i>网络设备</i>	8
2.1.5 <i>USB camera</i>	8
2.1.6 <i>USB audio</i>	8
2.1.7 <i>基于libusb 的用户层驱动</i>	9
2.1.8 <i>USB HUB device 支持</i>	9
2.1.7 <i>其他 USB 设备</i>	9
2.2 USB GADGET 配置	9
2.3 芯片 USB 控制器配置	10
三 SYSFS 系统调用接口	12
3.1 GADGET 连接状态	12
3.2 系统中 VBUS 状态	12
3.3 开机检测 VBUS 状态	12
3.4 设备主动连接 PC 使能.....	12
3.5 ANDROID GADGET 接口	13
3.6 OTG 控制器角色强制切换.....	13
3.7 查询 OTG 控制器当前角色	14
3.8 控制器调试接口.....	14
3.9 控制器使能.....	14
四 USB 充电器检测	15
4.1 采用充电 IC 与标准充电器.....	15
4.2 无充电 IC, 标准充电器.....	15
4.3 无充电 IC, 非标准充电器.....	15
五 设备信息修改	16
5.1 MSC 外设(U 盘)挂载.....	16
5.2 VID/PID 规则及修改	16
5.3 设备 VENDOR, PRODUCT 字符串修改	16
5.4 ANDROID2.3 以前设备信息修改.....	16
六 USB 常见问题 DEBUG	17
6.1 USB 正常工作 LOG	17

6.1.1 OTG 正常切换 log	17
6.1.2 Device 状态 log	17
6.1.3 Host 状态 log.....	18
6.2 USB 常见问题排查步骤	19
6.2.1 软件配置.....	19
6.2.2 硬件电路.....	19
6.2.3 Device 功能异常排查步骤.....	19
6.2.4 Host 功能排查步骤.....	20
6.3 U 盘无法挂载.....	21
6.4 PC 驱动问题.....	22
6.5 USB 问题上报.....	22
七 测试要求	23
7.1 USB DEVICE 测试.....	23
7.2 USB HOST 测试.....	23
7.3 测试报告要求.....	24

本文档适用范围:

RK291X, RK290X, RK3066, RK3066B, RK292X, RK3108。

运行 Android 4.0 及更新的系统

适用对象:

软件工程师, 硬件工程师, 测试工程师

概述

Rockchip SOC 通常内置多个 USB 控制器, 请在芯片 datasheet 中获取详细信息, 并明确方案具体的需求决定需要使用几个控制器, 由于部分控制器有使用限制, 所以请务必明确方案需求及控制器限制后确定使用方案。不同控制器互相独立,

RK2918 和 RK2906 内置 3 个 USB 控制器, 均挂在 RK29 periph_ahb 总线上。一个 USB2.0 OTG 控制器, 支持作为 host 或 device, 一个 USB 2.0 host 控制器, 一个 USB1.1 host 控制器。

USB1.1 host 控制器支持 USB WIFI, USB 3G dongle, 不支持 MSC 设备(U 盘), 内置使用; 使用时 HOST DP/DM 通过 15K Ω 电阻下拉, 不使用时务必在 kernel 配置中去掉对应选项, 否则会导致休眠唤醒出问题;

USB2.0 host 控制器不支持 Periodic OUT 传输(如 audio out 设备), 支持直接接不含 Periodic OUT 传输的各种设备, 如果接 hub, 仅能支持 MSC 设备及 HID 设备;

USB2.0 otg 控制器作为 host 时使用没有限制。

RK3066, RK3066B, RK2928 均内置 2 个 USB 控制器, 包含一个 USB 2.0 OTG 控制器和一个 USB 2.0 HOST 控制器, 使用上没有限制。

SDK 提供的 USB 子系统支持所有 linux 所提供的标准接口, 包括各种传输类型, 协议规范内的休眠唤醒等, 用户支持新的设备只需调试 usbcore 层以上的 class 驱动, 需要注意的是通过 usb_submit_urb 提交请求时, 需要保证 urb 成员 transfer_buffer 为四字节对齐。

Full speed(usb 1.1)bulk 理论最高数据传输速率: 1216K byte/second

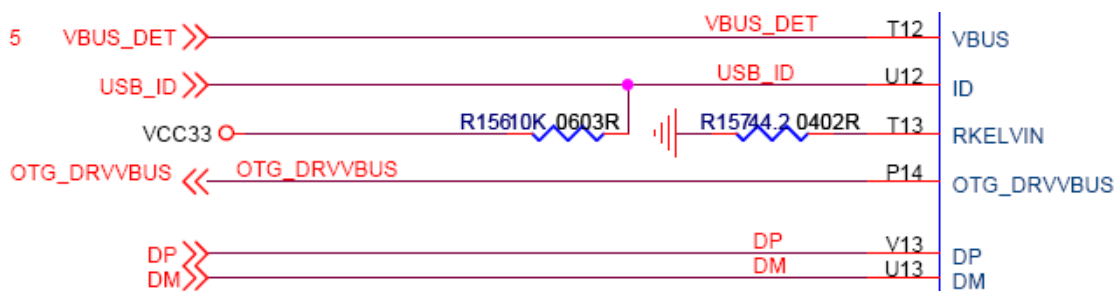
High speed(usb2.0)bulk 理论最高数据传输速率: 53248K byte/second, 实际最高速率约 30MB/S

一 硬件电路及信号说明

USB2.0 的工作时钟高达 480MHz，所以 layout 时需要特别注意，USB 走线宽度为 7-8MIL，做 90R 阻抗差分等长等间距走线，过孔数量不超过 2 个，最好在表层走线并有包地，边上无干扰源，正对的上下层不能有其他信号走线。

1.1 USB 2.0 控制器相关信号

完整的 USB 2.0 OTG 控制器硬件信号如下图：



VBUS_DET: 输入信号，用于 USB DEVICE 检测 VBUS 电平，
0: 低电平约 0V，1: 高电平约 3V。默认无连接时电平为低，连接至 PC 或充电器时电平为高。

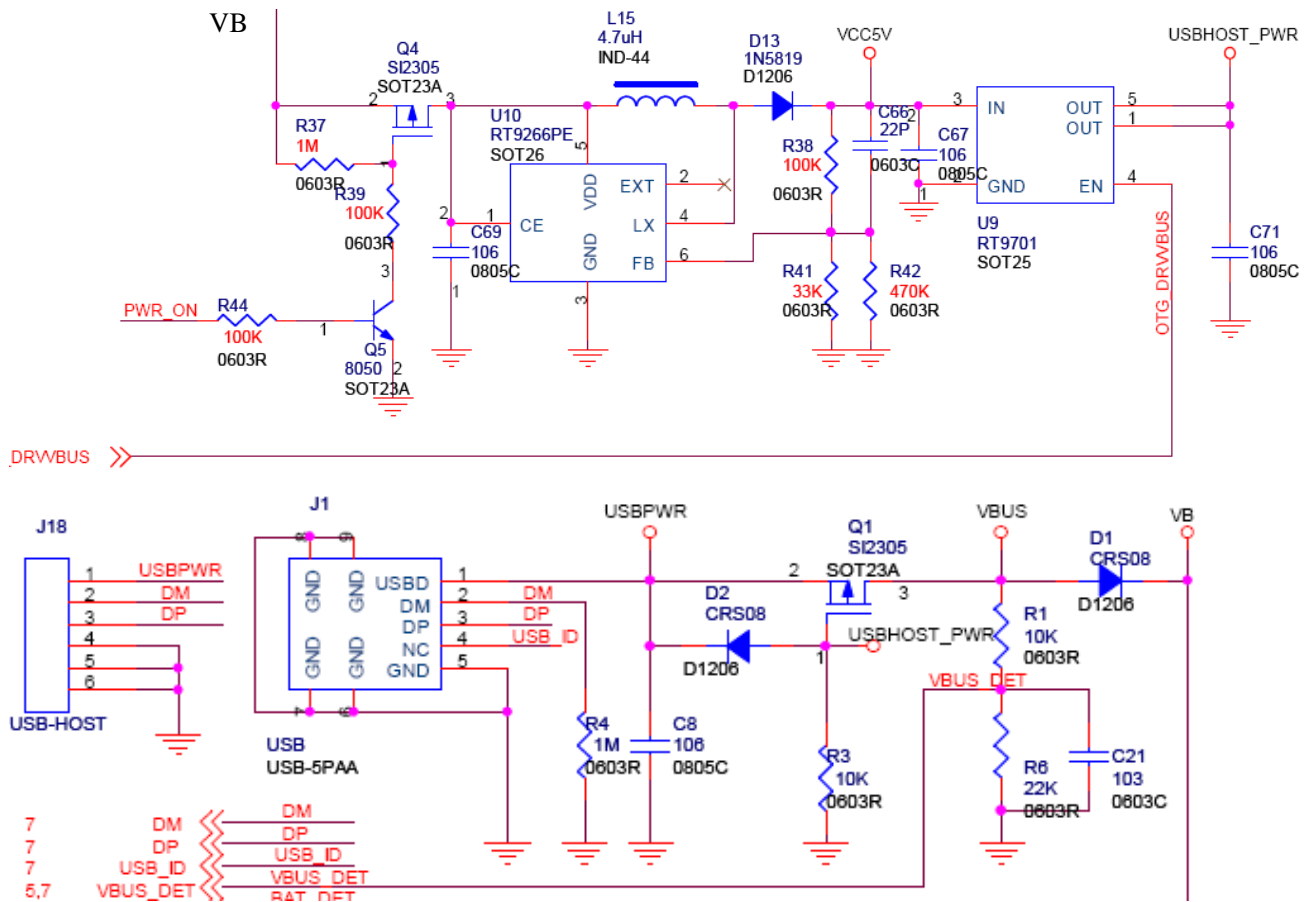
USB_ID: 输入信号，由 USB OTG 协议定义，用于识别 USB 口所接设备的默认角色(host or device)。USB_ID 默认上拉，处于 device 状况，如果要控制器进入 host 状态，需外接 mini-A 口或 micro-A 口将 USB_ID 短接到地。

RKELVIN: 参考电阻默认 44.2 欧到地，可通过调节该电阻阻值来调整 USB 信号质量。

OTG_DRVVBUS: 该信号由 USB 控制器的 host 寄存器控制，硬件上通过该信号来控制 host 所需 5V VBUS 输出，高电平输出有效。

DP/DM: 即 Data+, Data-，USB 的两根差分信号线。

HOST 典型供电电路及信号连接如下图，实际使用可能比下图复杂：



1.2 USB 控制器信号使用

控制器如果只作为 host， 需要信号 DP/DM， DRV_VBUS(可选)。

控制器如果只作为 device， 需要信号 DP/DM, VBUS_DET。

控制器作为 OTG 使用， 需要信号 DP/DM, USB_ID, DRV_VBUS, VBUS_DET。

二 kernel 模块配置说明

使用 `make menuconfig` 命令进行 kernel 配置，在配置选项之前，请使用 `make help` 命令列出目前 kernel 支持的配置，并选用对应芯片配置如 `make rk30_sdk_defconfig`。

Make menuconfig 得到 kernel 配置界面后，USB 模块的配置位于

Device Drivers --->

[*] USB support --->

必须选上 USB support 项后才能支持 USB 模块并进行进一步的配置。后面详细说明每一项的具体配置。

2.1 USB host相关配置

需要支持 USB host，首先需要选上 <>Support for Host-side USB 项，然后会有出现很多 host 相关的配置，我们应根据实际需求来配置。

```

--- USB support
<*> Support for Host-side USB
[ ] USB verbose debug messages
[*] USB announce new devices
*** Miscellaneous USB options ***
[ ] USB device filesystem (DEPRECATED)
[*] USB device class-devices (DEPRECATED)
[ ] Dynamic USB minor allocation
[ ] USB runtime power management (autosuspend) and wakeup
[ ] Rely on OTG Targeted Peripherals List
[ ] Disable external hubs
< > USB Monitor
< > Enable wireless USB extensions (EXPERIMENTAL)
< > Support WUSB Cable Based Association (CBA)
*** USB Host Controller Drivers ***
< > Cypress C67x00 HCD support
< > OXU210HP HCD support
< > ISP116X HCD support
< > ISP 1760 HCD support
< > ISP1362 HCD support
< > SL811HS HCD support
v(+)
-----
<Select> < Exit > < Help >

```

常用 USB 设备有：

U 盘/CDROM /USB 硬盘(USB Mass Storage support)

3G modem (USB Serial Converter support -->)

USB 鼠标键盘(HID)

2.1.1 Mass Storage Class (MSC)

U 盘属于 SCSI 设备，所以在配置 USB 模块之前需要配置 SCSI 选项

Device Drivers --->

SCSI device support --->

<*> SCSI disk support

U 盘属于 SCSI disk 设备，另外有些 U 盘可能有多个盘符，需要注意选上相关选项，如下图

```

----- SCSI device support -----
menu. <Enter> selects submenus --->. Highlighted letters are hotkeys.
Prizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search.
< > module capable

-----
< > RAID Transport Class
<*> SCSI device support
< > SCSI target support
[*] legacy /proc/scsi/ support
    *** SCSI support type (disk, tape, CD-ROM) ***
<*> SCSI disk support
< > SCSI tape support
< > SCSI OnStream SC-x0 tape support
<*> SCSI CDROM support
[ ] Enable vendor-specific extensions (for SCSI CDROM)
<*> SCSI generic support
< > SCSI media changer support
    *** Some SCSI devices (e.g. CD jukebox) support multiple LUNs ***
[*] Probe all LUNs on each SCSI device
[ ] Verbose SCSI error reporting (kernel size +=12K)
[ ] SCSI logging facility
[ ] Asynchronous SCSI scanning
    SCSI Transports --->
[ ] SCSI low-level drivers --->
-----

```

配置完 SCSI device support 后，可以在 USB support 中找到如下选项，选上即可。

```

<*> USB Mass Storage support
[ ] USB Mass Storage verbose debug

```

2.1.2 USB Serial Converter

USB 3G modem 使用的是 USB 转串口，根据实际默许可能使用下面的选项

<> **USB Modem (CDC ACM) support**

<*> **USB Serial Converter support --->**

USB Serial Converter support 的具体配置可能用到下面两个项：

<*> **USB driver for GSM and CDMA modems**

<> **USB USI Serial driver**

2.1.3 USB 鼠标键盘等HID设备

USB 键鼠的配置位于

```
Device Drivers --->
[*] HID Devices --->
    <*> USB Human Interface Device (full HID) support
```

选上后内核即可支持 USB 键盘和鼠标及基于 HID 协议遥控器等设备。

2.1.4 网络设备

USB 蓝牙

```
Device Drivers --->
[*] Networking support --->
    <*> Bluetooth subsystem support --->
        Bluetooth device drivers --->
            <*> HCI USB driver
```

USB WIFI

通常直接使用 vendor 提供的驱动。

USB Ethernet

```
Device Drivers --->
[*] Network device support --->
    USB Network Adapters --->
```

2.1.5 USB camera

```
Device Drivers --->
<*> Multimedia support --->
    [*] Video capture adapters --->
    [*] V4L USB devices --->
    <*> USB Video Class (UVC)
```

2.1.6 USB audio

```
Device Drivers --->
<*> Sound card support --->
    <*> Advanced Linux Sound Architecture --->
    [*] USB sound devices --->
    <*> USB Audio/MIDI driver
```

2.1.7 基于libusb的用户层驱动

部分 USB 设备使用基于 libusb 的用户层驱动，需要 usbfs，libusb，用户层驱动共同完成对设备的支持。

Android 的 libusb 需要在 android 层单独编译，可在网络上搜索实现方法。

```

Usbfs 支持:
Device Drivers --->
  [*] USB support --->
    [*] USB device filesystem (DEPRECATED)
  
```

2.1.8 USB HUB device支持

USB HUB 支持可能默认被 disable 了，下面的选项没有被选上

```

Device Drivers --->
  [*] USB support --->
     Disable external hubs
  
```

2.1.7 其他USB设备

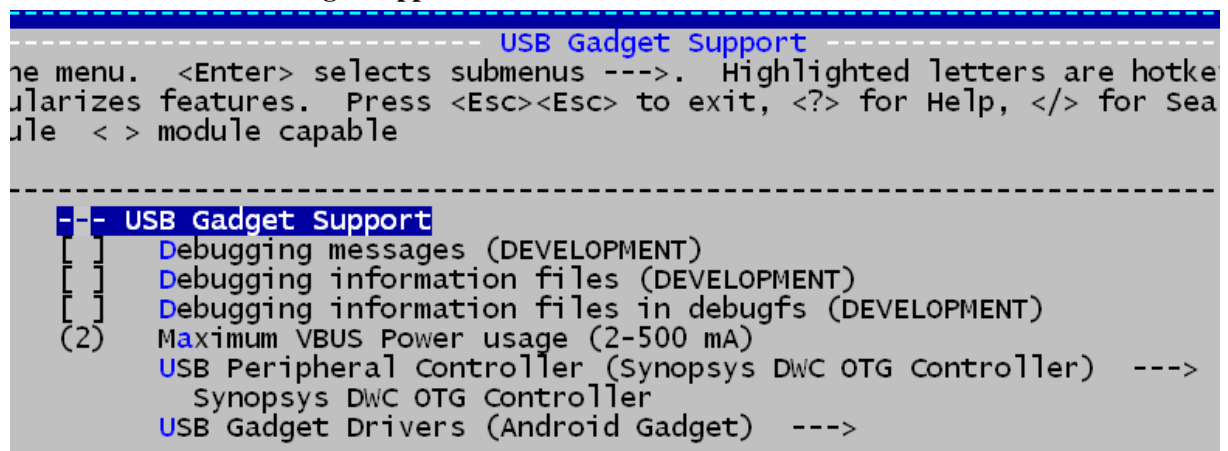
其他有可能用到的 USB 设备还有很多，如 GPS，printer 等，有可能需要 vendor 定制的驱动，也有可能是标准的 class 驱动，如需支持，可直接在网络上搜索 linux 对该设备支持要做的工作，RK 平台并无特殊要求，可直接参考。

2.2 USB gadget配置

控制器作为 device 使用时，需要配置 USB gadget:

```

Device Drivers --->
  [*] USB support --->
    <*> USB Gadget Support --->
  
```



```

----- USB Gadget Support -----
ne menu. <Enter> selects submenus --->. Highlighted letters are hotke
ularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Sea
le < > module capable
-----
-- USB Gadget Support
[ ] Debugging messages (DEVELOPMENT)
[ ] Debugging information files (DEVELOPMENT)
[ ] Debugging information files in debugfs (DEVELOPMENT)
(2) Maximum VBUS Power usage (2-500 mA)
USB Peripheral Controller (Synopsys DWC OTG Controller) --->
    Synopsys DWC OTG Controller
USB Gadget Drivers (Android Gadget) --->
  
```

Android 4.0 以后所有 gadget 类型默认都参与编译，需要使用时通过 android 设备管理进行切换。

2.3 芯片USB控制器配置

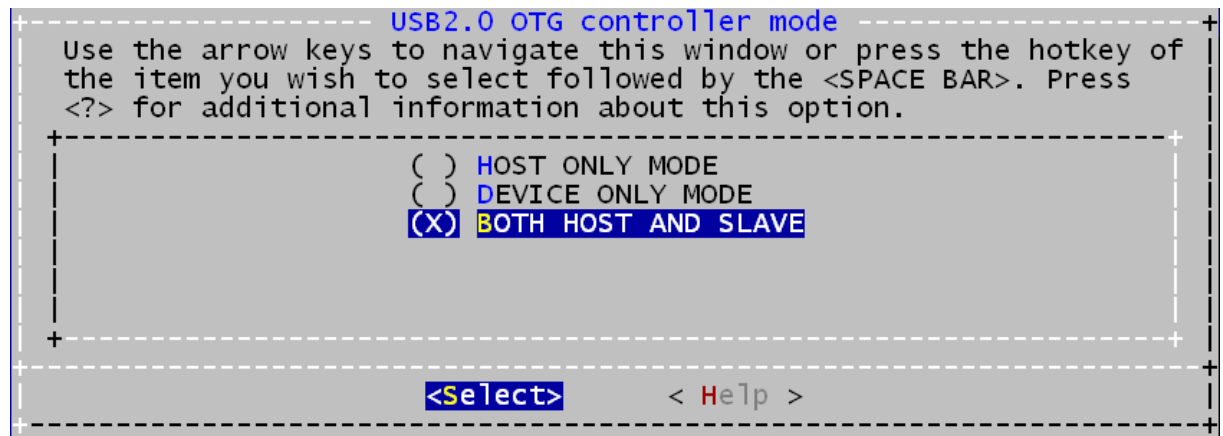
控制器配置位于 USB 选项最下方，三个控制器的默认配置如下：

```
<*>  RockChip USB 1.1 host controller
[*]   ---usb1.1 host controller enable (NEW)
<*>  Rockchip USB 2.0 host controller
[*]   ---usb2.0 host controller enable
<*>  RockChip USB 2.0 OTG controller
      USB2.0 OTG controller mode (BOTH HOST AND SLAVE)
      Controller default status (depends on USB_ID) ---->
[*]   ---connect to PC when vbus detect
[*]   ---usb2.0 otg host controller enable (NEW)
[]    DWC_OTG debug messages
```

其中 host 控制器选项需要完成 2.1 的配置后才能出现。

如果项目中没有使用 USB1.1 host 控制器或者 USB2.0 host 控制器，请在这去掉对相应控制器的支持。

USB2.0 OTG 控制器可选为 HOST ONLY MODE, DEVICE ONLY MODE 或 BOTH HOST AND SLAVE。



[*] ---usb2.0/1.1 host controller enable

该选项用于使能控制器作为 host，默认是否 enable，可利用 sysfs 接口对控制器进行开关。

[] connect to PC when vbus detect

该选项只有在 USB2.0 OTG 工作于 DEVICE ONLY MODE 才会出现，用于确定默认情况当控制器检测到 VBUS 时，设备是否主动进行连接。如果该项没有选上，控制器检测到 VBUS 后并不会主动连接 PC，sysfs 提供选项用于该选项的开关。

三 SYSFS 系统调用接口

Linux kernel 通过 sysfs 接口完成与 android 层的交互，除了 android 自带的接口，其他增加的调试接口可能需要在特定环境才能使用。

3.1 GADGET连接状态

用于查询控制器工作于 device 模式时 gadget 连接 HOST(PC 机)状态，可用于判断外接为充电器还是 PC。在 android2.3 以前的系统，使用 MSC 连接状态，android4.0 开始增加了 MTP/PTP 等其他模式，无法继续使用 MSC 状态。

本接口目前仅支持 kernel 函数调用接口，int get_gadget_connect_flag(void)

返回值为： 0 无 PC 连接连接；
1 设备已经通过 USB 连接到 PC。

3.2 系统中VBUS状态

控制器作为 device 功能时，一直通过 VBUS_DET 信号监控 VBUS 状态，VBUS 如果为高，一般是设备通过 USB 线连接到 PC。Kernel 函数调用接口为 int dwc_vbus_status(void)。

VBUS 状态查询节点：

/sys/bus/platform/drivers/usb20_otg/vbus_status (read only)

vbus_status 可能的值为： 0 VBUS 为低电平(约 0V)；
1 VBUS 为高电平(约 5V)；
2 VBUS 为高，且 DP/DM 信号均为高电平(用于国标充电器判断)。

3.3 开机检测VBUS状态

系统开机时可能会出现低电导致不能正常开机的情况，这时候无法使用系统正常运行的接口，又需要检测是否外接充电器来确定是否充电，所以提供 kernel 函数接口：

int dwc_otg_check_dpdm(void)

返回值与 3.2 一致，请注意本接口只能在系统开机低电不进入系统时使用，如果 usb 模块已完成初始化，请使用 3.2 的接口。

3.4 设备主动连接PC使能

对应 2.3 节的控制器选项，用于使能控制器检测到 VBUS 后连接 PC。

/sys/bus/platform/drivers/usb20_otg/dwc_otg_conn_en (read&write)

dwc_otg_conn_en 可能的值为： 0 设备检测到 VBUS 后不发起 USB 连接；
1 设备检测到 VBUS 后会发起 USB 连接。

3.5 android gadget接口

Android4.0 以后， android gadget 接口所有内容均在以下节点：

```
ls /sys/class/android_usb/android0
bDeviceClass
bDeviceProtocol
bDeviceSubClass
bcdDevice
enable
f_accessory
f_acm
f_adb
f_audio_source
f_mass_storage
f_mtp
f_ptp
f_rndis
functions
iManufacturer
iProduct
iSerial
idProduct
idVendor
power
state
subsystem
Uevent
```

连上 PC 后， flash 用户盘及 SD 卡可以作为 gadget 设备被 PC 访问。实际上 PC 访问的媒介路径是由 android 通过 sysfs 写到 USB 驱动的， 该 sysfs 路径为：

```
/sys/class/android_usb/android0/f_mass_storage/lun/file
/sys/class/android_usb/android0/f_mass_storage/lun1/file
```

3.6 OTG控制器角色强制切换

2.3 节提到 OTG 控制器可以作为 HOST， 也能作为 device， 控制器的角色一般由 USB ID 电平决定(参考 1.1 节 USB_ID 信号)， 也可以由软件进行强制切换， 接口如下：

```
/sys/bus/platform/drivers/usb20_otg/force_usb_mode
```

```
force_usb_mode 可能的值为： 0 depends on USB ID
                               1 force host
                               2 force device
```

3.7 查询OTG控制器当前角色

USB2.0 HOST 和 USB1.1 HOST 都只能作为 host 使用，而 USB2.0 OTG 可以作为 host，也能作为 device 功能使用，查询当前工作的角色接口如下：

```
/sys/devices/platform/usb20_otg/mode
Mode:  0   device mode
       1   host mode
```

3.8 控制器调试接口

可以通过如下接口打印各个控制器当前的寄存器状态：

```
cat /sys/devices/platform/[controller name]/debug
```

其中[controller name]为控制器名称，可以是 usb11_host, usb20_host, usb20_otg 。

3.9 控制器使能

控制器可以通过 SYSFS 接口使能或关闭，控制器 disable 后进入低功耗模式，该接口主要用途在于内置 USB 设备需要动态开关保证整机的最低功耗。接口如下：

```
/sys/devices/platform/[controller name]/enable
```

其中[controller name]为控制器名称，

```
Enable 状态: 0   disable
              1   enable
```

四 USB 充电器检测

USB-IF 发布 battery charge1.2 标准作为移动设备充电检测标准，其他一些组织也有发布类似标准，其中中国工信部标准要求 USB 充电器的 DP,DM 短接。

RK 的 USB 控制器不支持 battery charge1.2 标准，但是可以检测出国标充电器的 DP,DM 短接。

由于 PC 的 USB host 口默认只提供 500mA 的供电，而一般充电器可以达到 800mA 或者 1A 以上的电流充电，所以需要识别并做不同处理。

USB 模块提供的 VBUS 状态和 PC 连接状态变化情况如下，充电模块如需使用请务必了解状态的变化过程：

- 1 无连接，VBUS 状态为 0，gadget 连接状态为 0；
- 2 刚连接到 PC 或充电器，VBUS 状态为 1，gadget 连接状态为 0；
- 3 device 会发起连接，如果是信号较好的 PC，可以很快连上(1,2s 内)，如果信号较差，设备会 retry 3 次，每次时间约为 1S，连上后 VBUS 状态为 1，gadget 连接状态为 1；
- 4 如果外接不是 PC，那么 3 次连接超时后，会给出 DP,DM 的状态，如果是标准充电器 VBUS 状态为 2，gadget 连接状态为 0；如果非标准充电器，VBUS 状态为 1，gadget 连接状态为 0。

4.1 采用充电IC与标准充电器

如果项目采用充电 IC，可以直接使用充电 IC 提供的状态进行判断，需要注意的是充电 IC 会连接到 USB 的 DP,DM 信号，硬件上需要注意进行不能影响 DP,DM 的信号质量。

4.2 无充电IC，标准充电器

可以直接通过 USB 模块提供的标志来判断是否外接充电器，具体见 3.2，3.3。

4.3 无充电IC，非标准充电器

非标准充电器的判定标准，一般为有 VBUS 连接，且长时间无 PC 连接，这实际上是根据 USB 的连接行为做的判定，如果 USB 线较差导致实际上的 PC 连接没有连上，也会被误判为充电器，所以是存在风险的，建议尽量使用标准充电器，即只需要将 DP,DM 短接悬空即可。

五 设备信息修改

Android4.0 后所有的 VID,PID, 设备描述符等信息均由 android 的 vold 代码提供。

5.1 MSC外设(U盘)挂载

已发布的 SDK 默认只支持在 USB2.0 HOST 控制器上挂载 U 盘等 MSC 外设, 配置位于 /device/rockchip/rk30sdk/vold.fstab, 如果需要增加其他控制器的支持, 如 USB2.0 OTG, 需要修改如下代码:

```
dev_mount udisk /mnt/usb_storage auto /devices/platform/usb20_host/usb
```

修改为

```
dev_mount udisk /mnt/usb_storage auto /devices/platform/usb20_host/usb /devices/platform/usb20_otg/usb
```

系统起来后可以使用 `cat /system/etc/vold.fstab` 命令查看实际配置。

对 MSC 外设, 默认支持最多 4 个分区, 并显示第一分区, 如果要支持移动硬盘的多分区, 或者显示同时在多个 USB 口接进来的不同 U 盘, 需要修改的内容有:

1. 去除代码中对于最多分区支持的限制(/system/vold/DirectVolume.h MAX_PARTITIONS 宏定义);
2. 在 init.rc 中增加并在 vold.fstab 中配置挂载点, 上面的代码都是挂载到 /mnt/usb_storage;
3. 在 APK 应用, 如资源管理器中增加对应的入口。

5.2 VID/PID规则及修改

USB gadget 在不同配置下, 必须使用不同的 PID, 否则会导致 PC(windows)驱动识别混乱, 常用的 gadget 为 MSC, MTP 及跟 ADB 的组合。Google 在 2011 年提出 AOA(Android Open Accessory), 在 gadget 中增加一个 accessory 模式, 对于 ID 强制要求为:

```
Accessory: VID/PID = 0x18d1, 0x2d00
```

```
Accessory,adb: VID/PID = 0x18d1, 0x2d01
```

其他 ID 可以使用 vendor 自己的 ID, SDK 默认使用 rockchip 的 VID(0x2207), PID(depends on config), 如果需要使用 rockchip 的 ID, 请使用默认值, 请勿做任何修改, 如果需要使用自己的 VID/PID, 请自行规划。VID/PID 修改位于 /device/rockchip/rk30sdk/init.rk30board.usb.rc 文件

5.3 设备vendor, product字符串修改

Gadget 连接电脑时, 设备的 vendor 和 product 字符串会出现在弹出的 USB 设备和资源管理器中的设备名称。SDK 发布同时带有固件工厂工具用于修改这两个字符串

5.4 android2.3 以前设备信息修改

Android2.3 以前的设备 PID/VID 及 Vendor/product 描述符: arch/arm/mach-rk29/device.c android_usb_pdata 结构体, 带 ADB 与不带 ADB 对应的 product_id 是不同的, 由 usb_products[] 数组中不同的配置 ID 决定。

MSC vendor and product string descriptor: mass_storage_pdata in arch/arm/mach-rk29/device.c


```
[78.629993] DWC_OTG: USB SUSPEND
[80.443976] DWC_OTG: *****soft reconnect*****
[80.623991] DWC_OTG: *****soft connect!!!*****
[80.630007] DWC_OTG: USB SUSPEND
[82.643969] DWC_OTG: *****soft reconnect*****
[82.623989] DWC_OTG: *****soft connect!!!*****
[82.630003] DWC_OTG: USB SUSPEND
```

6.1.3 Host状态log

Host 枚举 USB 设备

LowSpeed device

```
[73.094029] usb 1-1: new low speed USB device using usb20_otg and address 2
[73.310527] usb 1-1: New USB device found, idVendor=04b3, idProduct=3107
[73.314502] usb 1-1: New USB device strings: Mfr=0, Product=0, SerialNumber=0
[73.323506] usb 1-1: configuration #1 chosen from 1 choice
```

HighSpeed device

```
[101.844051] usb 1-1: new high speed USB device using usb20_otg and address 3
[102.047356] usb 1-1: New USB device found, idVendor=0930, idProduct=6545
[102.051301] usb 1-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[102.058516] usb 1-1: Product: DataTraveler G2
[102.062902] usb 1-1: Manufacturer: Kingston
[102.067192] usb 1-1: SerialNumber: 001D92AD7612B91113680066
[102.074682] usb 1-1: configuration #1 chosen from 1 choice
```

MSC 设备磁盘挂载

```
[902.079988] scsi0 : SCSI emulation for USB Mass Storage devices
[907.120400] scsi 0:0:0:0: Direct-Access Kingston DataTraveler G2 PMAP PQ: 0 ANSI: 0 CCS
[907.865209] sd 0:0:0:0: [sda] 31375360 512-byte logical blocks: (16.0 GB/14.9 GiB)
[907.870422] sd 0:0:0:0: [sda] Write Protect is off
[907.874938] sd 0:0:0:0: [sda] Assuming drive cache: write through
[908.641540] sd 0:0:0:0: [sda] Assuming drive cache: write through
[908.644920] sda: sda4
[908.681165] sd 0:0:0:0: [sda] Assuming drive cache: write through
[908.684586] sd 0:0:0:0: [sda] Attached SCSI removable disk
```

HOST 断开设备

```
[112.264068] usb 1-1: USB disconnect, address 3
```

6.2 USB 常见问题排查步骤

6.2.1 软件配置

首先必须明确项目中 USB 控制器是如何分配的，并确保 kernel 的配置是正确的，请参考第二章配置说明，需要根据项目的实际使用情况进行配置。主要注意下面几点：

1 USB1.1 HOST 控制器是否使用。如果不使用，请在配置中去掉对应支持，如果使用，请保证 HOST1.1 控制器的数据线有下拉电阻，否则会影响设备的识别或系统待机唤醒。

2 如果使用 USB2.0 HOST 控制器，请配置上对该控制器的支持，否则不支持。

3 USB2.0 OTG 控制器，明确该控制器作为什么角色并进行配置。如一般带两个 USB 口的 MID 设备选用 DEVICE ONLY 模式，带单个 USB 口的 MID 设备选用 BOTH HOST AND SLAVE 模式，TVBOX 一般采用 HOST ONLY 模式。HOST ONLY 模式必须同时去掉 USB gadget 支持。

4 USB class driver 支持，如 U 盘需要配置 USB Mass Storage 支持，具体见 [2.1 节](#)。

6.2.2 硬件电路

在同时使用多个控制器对应同一个 USB 口，或者一个控制器对应多个 USB 口时，可能会使用电子开关来切换 USB 信号及电源。需要确保不同控制器的电源控制是互相独立的，通过电子开关后，控制器与 USB 口之间的连接是有效的。

Case 1

1 个硬件 USB 口同时支持 HOST 和 device 功能，使用 USB2.0 HOST 控制器作为 host 和 USB2.0 OTG 控制器作为 device，通过硬件电子开关进行切换。

需要保证工作于 host 状态时，USB 信号是切换到 USB2.0 HOST 控制器，而 VBUS 是由 host 供电电路提供，而不影响 device 的 VBUS 电平检测电路。工作于 device 状态时，USB 信号是切换到 USB2.0 OTG 控制器，VBUS 由 PC 通过 USB 线提供。

Case 2

使用一个 USB2.0 OTG 控制器，对应使用两个硬件 USB 口分别是 HOST 和 Device。通过电子开关进行信号切换。

工作于 host 状态时，USB2.0 OTG 的 DP/DM 信号线是切换到 HOST 口，且 HOST 口 VBUS 提供 5V 500MA 的供电；工作于 device 状态时 DP/DM 信号是切换到 device 口，VBUS 电平检测电路只检测 PC 提供的 5V 供电。

6.2.3 Device 功能异常排查步骤

USB device 正常连接至 PC 的现象主要有：

- 1 串口输出正常 log 见 [6.1.2](#);
- 2 PC 出现盘符，但默认不能访问；(windows 7 和 MAC OS 可能只出现在设备管理器)；
- 3 设备 UI 状态栏出现“USB 已连接”标识；
- 4 打开 USB 已连接窗口，选择 打开 USB 存储设备 后，PC 可以访问盘符。

常见异常排查:

1 连接 USB 时串口完全没有 log:

USB 硬件信号连接正确;

USB 控制器确保工作在 device 状态;

测量 USB_DET 信号电压, USB 连接时应该由低到高。

2 连接失败, PC 显示不可识别设备, log 一直重复打印:

```
[36.682587] DWC_OTG: *****soft connect!!!*****
```

```
[36.688603] DWC_OTG: USB SUSPEND
```

```
[36.807373] DWC_OTG: USB RESET
```

但是没有正常 log 中的后面几条信息。

一般为 USB 硬件信号差, 无法完成枚举。

3 连接 PC 后, log 正常, PC 盘符出现无异常, 设备为出现“USB 已连接”标识:

驱动工作正常, android 层异常, 截取 logcat 内容, 有条件的话请负责维护 void 代码的 android 工程师帮忙 debug。

4 连接 PC 正常, 并能正常访问, 拷贝文件过程中提示拷贝失败:

一般为信号差, 可使用 USB 分析仪抓取数据流后分析。

也有可能是 flash/sd 卡读写超时, log 一般为连接 window xp 时约 10S 出现一次重新连接的 log。

5 USB 线拔掉后 UI 状态栏仍然显示“USB 已连接”, 或 USB 线拔掉时的 log 只有下面这句:

```
[25.330017] DWC_OTG: USB SUSPEND
```

而没有下面的 log

```
[25.514407] DWC_OTG: *****session end intr, soft disconnect*****
```

VBUS 异常, 一直为高, 会影响 USB 检测及系统休眠唤醒, 请硬件工程师排查问题。

6.2.4 Host功能排查步骤

USB host 正常工作情况如下:

1 首先 HOST 电路提供 VBUS 为 5V, 至少 500mA 的供电, 其中移动硬盘, 3G dongle 等外设的峰值电流可能大于 500mA, 需要支持的话要提高限流到 800mA 或 1A, 另外部分外设接入 USB 口时会导致 5V 有个较大的电压塌陷, 从而导致设备工作异常, 这点需要示波器确认设备接入瞬间的 VBUS 波形。

2 如果有USB设备连接进来, 串口首先会打印HOST枚举USB设备的log(见[6.1.3](#)), 表明USB设备已经通过HOST的标准设备枚举。

3 class 驱动打印设备扫描、识别信息, 如 4.1.3 中的 MSC 设备磁盘挂载。

4 步骤 3 中 kernel 驱动已完成设备挂载, android 层可以挂载、使用设备资源。

常见异常及排查:

1 HOST 口接入设备后, 串口无任何打印:

首先需要确认通过电子开关后的电路连接正确;

确认控制器工作于 HOST 状态, 并确认供电电路正常。

2 串口有 HOST 枚举 USB 设备内容，但是没有出现 class 驱动的打印信息。

Kernel 没有加载 class 驱动，需要重新配置 kernel，加入对应 class 驱动支持。

3 kernel 打印信息完整(USB 标准枚举信息及 CLASS 驱动信息)，已在 linux 对应位置生成节点，但是 android 层无法使用。

Android 层支持不完善，如 U 盘在 kernel 挂载完成/dev/block/sda 节点后，需要 android 层 vold 程序将可存储介质挂载到/udisk 提供媒体库，资源管理器等访问，同样鼠标键盘等 HID 设备也需要 android 层程序支持。

6.3 U盘无法挂载

U 盘完整的识别和挂载步骤为：

- 1 标准 USB 设备枚举流程；
- 2 MSC 及 SCSI 对 U 盘进行扫描；
- 3 扫描正常在/dev/block 目录生成 sda, sdb...节点，例如 sda 为磁盘设备，sda1 为分区；
- 4 Android 层 vold 通过接收 uevent 得知有磁盘接入，根据 vold.fstab 进行挂载；
- 5 挂载完成后媒体库对新接入磁盘进行媒体扫描，其他应用可以访问 U 盘。

完整出现 6.1.3 的 log 后 U 盘完成/dev/block/sda 节点生成，仍然无法使用，比较有可能是 vold 的挂载过程出现问题。

1 打开 android 层 uevent dump 调试，提取有用信息 debug；

2 对比 uevent 路径和 vold.fstab 中配置的挂载路径(5.1 节)，路径不匹配有可能因为控制器是 usb20_otg，或最后字段为 usb2 而配置里面为 usb1，或 U 盘分区超出到 sda5 等。

修改 system/vold/ NetlinkHandler.cpp 文件:onEvent 函数，打印出设备的相关信息：

```
在 if (!strcmp(subsys, "block")) {
后增加以下语句
    evt->dump();
```

以下是打开 uevent dump 后打印的 log：

```
D/NetlinkEvent( 72): NL param
'DEVPATH=/devices/platform/usb20_host/usb1/1-1/1-1:1.0/host2/target2:0:0/2:0:0:0/block/sda'
D/NetlinkEvent( 72): NL param 'MAJOR=8'
D/NetlinkEvent( 72): NL param 'MINOR=0'
D/NetlinkEvent( 72): NL param 'DEVNAME=sda'
D/NetlinkEvent( 72): NL 不 param 'DEVTYPE=disk'
D/NetlinkEvent( 72): NL param 'NPARTS=1'
D/NetlinkEvent( 72): NL param
'DEVPATH=/devices/platform/usb20_host/usb1/1-1/1-1:1.0/host2/target2:0:0/2:0:0:0/block/sda/sda1'
D/NetlinkEvent( 72): NL param 'MAJOR=8'
D/NetlinkEvent( 72): NL param 'MINOR=1'
D/NetlinkEvent( 72): NL param 'DEVNAME=sda1'
D/NetlinkEvent( 72): NL param 'DEVTYPE=partition'
D/NetlinkEvent( 72): NL param 'PARTN=1'
```

从上面的 log 我们可以得到一些重要信息，如设备的挂载目录，分区数，和分区编号。通过上面的分析可修改 vold.fstab 里正确的挂载路径。

6.4 PC驱动问题

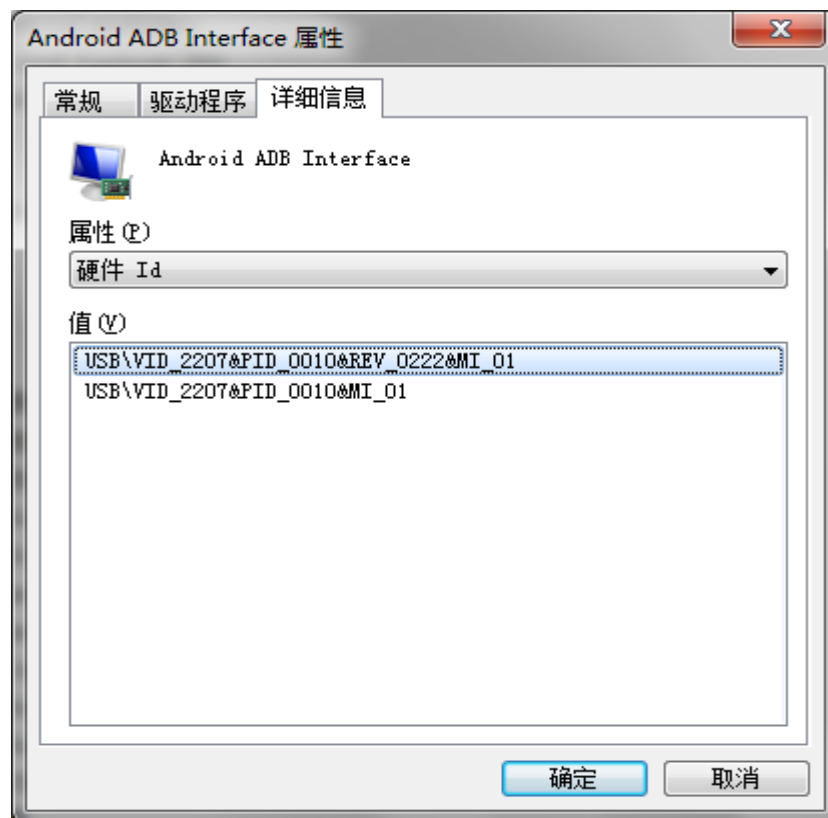
所有 USB 设备要在 PC 上正常工作都是需要驱动的，有些驱动是标准且通用的，而有些驱动是需要额外安装的。对于 RK 的设备连接到 PC 后，需要安装驱动的情况有两种的设备，需要分别选择对应的驱动。

- 1 生成后未烧写的裸片或者进入升级模式后的 RK 设备，会以 rockusb 的模式连接到 PC，需要 PC 安装 rockchip 提供的 rockusb 驱动才能识别；
- 2 Android 系统的设备，在设置里面打开了 USB debugging 选项，会以 ADB 模式连接 PC，rockchip 提供 windows 下的 adb 驱动(<http://developer.android.com/tools/extras/oem-usb.html>)。
- 3 linux 或者 mac os 用户 ADB 驱动请参考 <http://developer.android.com/tools/device.html>

对于完全未安装相关驱动的 PC，找不到驱动，在设备管理器中显示为未知设备。请在设备管理器中找到，右击“未知设备”->点击属性->进入详细信息选项->选择硬件 Id。

Rockchip VID 都是 2207([5.2 节](#));

PID 如果是 29,30,31 等开头,0A 或者 1A 结尾,为 rockusb 设备,如 PID 300A 为 RK30 的 rockusb 设备; PID 为 001x 的设备为 adb 设备,如 PID 为 0010 是 MSC+ADB 设备,0011 是 MTP+ADB。



6.5 USB问题上报

如果使用以上方法后，问题仍无法解决，可以提供下列内容并邮件至 fae@rock-chips.com

- 1 开机后串口输出的完整 kernel log，或可确认的出错前开始 log；
- 2 重现现象所需详细操作步骤；
- 3 电路原理图和 PCB 板图；
- 4 USB 分析仪数据(至少一份，尽量提供多份)。
- 5 调用[3.8 控制器调试接口](#)，如USB2.0 OTG口为cat /sys/devices/platform/usb20_otg/debug，打印控制器寄存器。

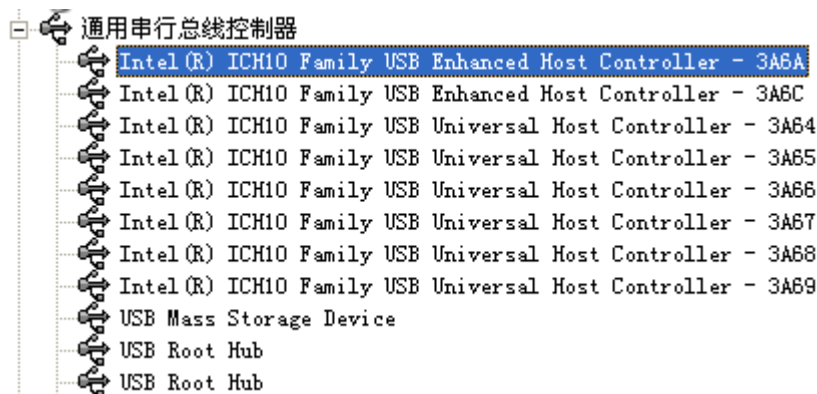
七 测试要求

不同项目在 USB 模块的配置和设备支持上不一致，但是所有项目必须对所有支持的功能进行测试，对应测试项及要求如下。

7.1 USB Device测试

- 1 在中英文，windows 2000，Windows XP，Windows vista 和 Windows 7 系统下连接 USB，安全删除设备时反应是否正常；
- 2 各种系统下 USB 数据传输是否正常(拷贝对比)；
- 3 连接 USB 开机是否能正常识别；
- 4 连接 USB 关机是否正常；
- 5 待机时 USB 拔插(a 连接 PC 后，PC 删除 USB，不拔 USB 线，进入待机后拔 USB 线；b 不插 USB 线，进入待机后连接 USB 线)；
- 6 USB 拔插测试，连续快速拔插和各种情况下随机拔插(待机，系统在各种应用的情况)；
- 7 传输速率测试，需要在 windows 2000，windows XP，windows 7 系统分别进行大文件和小文件的拷贝速率测试；
- 8 在 MTP，PTP，MSC 及其与 ADB 的组合，不同配置进行切换，测试连接电脑是否正常；
- 9 禁用 PC 高速 USB 控制器后，重复测试第 1,2 两项内容。

禁用 PC 高速 USB 控制器方法如下，打开设备管理器，找到如下图中选中的设备(Enhanced Host Controller), 右击选择停用。



7.2 USB HOST测试

USB HOST 支持的 USB 设备有：U 盘，3G 模组，鼠标，键盘，HUB，WIFI 等

测试项

- 1 各种设备识别情况测试；
- 2 连接设备开关机测试；
- 3 各种设备在各种情况下拔插测试；
- 4 各种设备功能测试，如 U 盘读写，3G 模组上网，鼠标键盘使用是否正常等；
- 5 各种设备的性能测试，如 U 盘读写速率。

7.3 测试报告要求

1. 完成的测试项，如有测试流程请记录具体测试流程；
2. 各测试项对应的测试量，如 USB device 拔插次数；
3. 各测试项中包含的性能测试结果；
4. 完整的测试 log(不需要打开 logcat)，可以通过该 log 直接查看 USB 拔插次数，非预期内的异常等；
5. 如果测试到异常，请记录详细出错情况，包括操作流程，出错时的 log，测试用的 USB 设备和 PC 操作系统；
6. 其他测试人员认为需要补充的内容。