



USBPIPE 使用说明

Cogobuy Only For ShenZhen FuShi ChanJing Industrial Technology Co., Ltd.

文档版本	00B02
发布日期	2019-01-15

版权所有 © 上海海思技术有限公司 2019。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HISILICON、海思和其他海思商标均为海思技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受海思公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，海思公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

上海海思技术有限公司

地址：深圳市龙岗区坂田华为总部办公楼 邮编：518129

网址：<http://www.hisilicon.com>

客户服务邮箱：support@hisilicon.com



前言

概述

本文档为指导使用 USBPIPE 作为 AI 协处理器与主控传输方案的相关开发程序员而写，目的是供您在开发过程中了解相关接口与操作流程。

产品版本

与本文档相对应的产品版本如下。

产品名称	产品版本
Hi3516C	V500
Hi3516D	V300
Hi3516A	V300


读者对象

本文档（本指南）主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师

符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

符号	说明
 危险	用于警示紧急的危险情形，若不可避免，将会导致人员死亡或严重的人身伤害。



符号	说明
	用于警示潜在的危险情形，若不可避免，可能会导致人员死亡或严重的人身伤害。
	用于警示潜在的危险情形，若不可避免，可能会导致中度或轻微的人身伤害。
	用于传递设备或环境安全警示信息，若不可避免，可能会导致设备损坏、数据丢失、设备性能降低或其它不可预知的结果。 不带安全警示符号的“注意”不涉及人身伤害。
	用于突出重要/关键信息、最佳实践和小窍门等。 “说明”不是安全警示信息，不涉及人身、设备及环境伤害信息。

修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

日期	版本	修改描述
2019-01-15	00B02	第 2 次临时版本发布 新增 1.2 小节，修改第 4 章
2018-11-20	00B01	第 1 次临时版本发布



目 录

前 言.....	i
1 USBPIPE 方案概述	1
1.1 总体介绍.....	1
1.2 芯片型号与内核版本说明	1
2 HOST 端说明.....	2
2.1 用户态接口	2
2.2 操作流程.....	2
3 Device 端说明	4
3.1 用户态接口	4
3.2 操作流程.....	4
4 sample 操作说明	7

Cogobuy Only For ShenZhen FuShi ChanJing Industrial Technology Co., Ltd.



1 USBPIPE 方案概述

1.1 总体介绍

USBPIPE 作为 AI 协处理器通过 USB 连接到主控的一个传输方案，主要解决两个关键问题：

- 把主控的数据通过 USB 传输到 AI 协处理器，由协处理器完成处理，并且返回结果，要保证传输处理时延和传输性能。
- 传统的 IPC，NVR，DVR 等设备的主控由于原有设计规格限制，CPU 和总线资源受限，需要尽量减少主控上的资源消耗，如减少内存拷贝操作，减少 CPU 占用。

USBPIPE demo 方案主要是由 4 个部分构成：usb pipe gadget driver, usb pipe class driver, host sample app, device sample app，方案特点如下：

- 采用 USB bulk 传输，有两个 bulk 传输通路，一个 bulk IN，一个 bulk OUT。
- 目前只支持单线程传输，传输模型更适合 host 发送大量数据，接受少量数据。
- HOST 发送数据基于连续物理地址的 buffer，可以实现 0 拷贝发送，节省 CPU 和带宽开销。

总结：USBPIPE 是专门针对 AI 协处理器的数据流量模型设计，特别对大量数据发送，少量接收，HOST CPU 和带宽资源受限等场景做了优化设计，更适合 AI 协处理器的方案。

1.2 芯片型号与内核版本说明

- device 侧功能适用芯片平台为 Hi3516CV500、Hi3516DV300，内核版本为 Linux4.9。
- host 侧功能对芯片平台无要求，适用内核版本为 Linux3.10、3.18、4.9。



2 HOST 端说明

2.1 用户态接口

用户态只有一个 IOCTL 接口，APP 通过 IOCTL 接口发送或者接收 USB 数据。

命令设计如下：

- #define IOCTL_OPEN_USBPIPE 0x1 /* 打开 usbpipe */
- #define IOCTL_CLOSE_USBPIPE 0x2 /* 关闭 usbpipe */
- #define IOCTL_STREAM_ON 0x3 /* reserved */
- #define IOCTL_STREAM_OFF 0x4 /* reserved */
- #define IOCTL_QUEUE_PHY_BUFF 0x5 /* 发送数据 */
- #define IOCTL_DEQUEUE_RESULT 0x6 /* 接收数据 */

参数：

```
struct usbpipe_ioctl_arg {  
    unsigned long addr; /* 对于IOCTL_QUEUE_PHY_BUFF命令，是物理地址，对于  
IOCTL_DEQUEUE_RESULT，是虚拟地址*/  
    unsigned int len; /* 对于IOCTL_QUEUE_PHY_BUFF命令，这个成员是发送数据长  
度，对于IOCTL_DEQUEUE_RESULT，这个成员返回读取数据的长度 */  
};
```

2.2 操作流程

步骤 1 编译生成 usbpipe.ko。

在 menuconfig 选上 usbpipe

位置如下：

```
Device Drivers --->  
[*] USB support ---->
```



<M> USB Pipe driver

```
< > NXP ISP 1760/1761 support
*** USB port drivers ***
< > USB Serial Converter support ----
*** USB Miscellaneous drivers ***
< > EMI 6|2m USB Audio interface support
< > EMI 2|6 USB Audio interface support
< > ADU devices from Ontrak Control Systems
< > USB 7-Segment LED Display
< > USB Diamond Rio500 support
< > USB Lego Infrared Tower support
< > USB LCD driver support
< > Cypress CY7C63xxx USB driver support
< > Cypress USB thermometer driver support
< > Siemens ID USB Mouse Fingerprint sensor support
< > Elan PCMCIA CardBus Adapter USB Client
< > Apple Cinema Display support
< > USB LD driver
< > PlayStation 2 Trance Vibrator driver support
< > IO Warrior driver support
< > USB testing driver
< > USB EHSET Test Fixture driver
< > iSight firmware loading support
< > USB YUREX driver support
< > Functions for loading firmware on EZUSB chips
< > USB3503 HSIC to USB20 Driver
< > USB4604 HSIC to USB20 Driver
< > USB Link Layer Test driver
< > USB Pipe driver
      USB Physical Layer drivers --->
<*> USB Gadget Support --->
< > USB ULPI PHY interface support
```

步骤 2 insmod usbpipe.ko。（位置：drivers/usb/misc/usbpipe.ko）

步骤 3 如果对端配置正确，USB 连接正常的话，就会生成/dev/usbpipe0 设备节点。

步骤 4 运行 usbpipe-sample 通过控制/dev/usbpipe0 节点完成自己的 host 端传输功能。



说明

可以通过 debugfs（mount debugfs 之后，cat /sys/kernel/debug/usbpipe/queues）查看发送接收数据包的调试信息。

-----结束-----



3 Device 端说明

3.1 用户态接口

在设备驱动安装，设备连接 ready 之后，会有一个/dev/usbpipe0 的设备节点产生，用户程序通过操作这个设备节点来控制 usbpipe 数据传输。主要实现了 3 个接口：

- Poll-----查询是否可以读写数据。
- Read----读数据。
- Write---写数据。

3.2 操作流程

步骤 1 修改 dtsi，设置 usb 端为 device 模式

```
#define USB_HOST 0
#if USB_HOST
    xhci_0@0x100e0000 {
        compatible = "generic-xhci";
        reg = <0x100e0000 0x10000>;
        interrupts = <0 27 4>;
        usb2-lpm-disable;
    };
#else
    hidwc3_0@0x100e0000 {
        compatible = "snps,dwc3";
        reg = <0x100e0000 0x10000>;
        interrupts = <0 27 4>;
        interrupt-names = "peripheral";
        maximum-speed = "high-speed";
        dr_mode = "peripheral";
    };
};
```

步骤 2 编译生成 libcomposite.ko、g_usbpipe.ko。

在 menuconfig 选上 usbpipe



位置如下:

```
Device Drivers --->
[*] USB support --->
    <*> DesignWare USB3 DRD Core Support
        DWC3 Mode Selection (Gadget only mode) --->
    <*> USB Gadget Support --->
        <M> USBPIPE Gadget
```

```
--- USB Gadget Support
[ ] Debugging messages (DEVELOPMENT)
[ ] Debugging information files (DEVELOPMENT)
[ ] Debugging information files in debugfs (DEVELOPMENT)
(2) Maximum VBUS Power usage (2-500 mA)
(2) Number of storage pipeline buffers
USB Peripheral Controller --->
< > USB functions configurable through configs
<M> USB Gadget Drivers
    < > Gadget Zero (DEVELOPMENT)
    < > Ethernet Gadget (with CDC Ethernet support)
    < > Network Control Model (NCM) support
    < > Gadget Filesystem
    < > Function Filesystem
    < > Mass Storage Gadget
    < > Serial Gadget (with CDC ACM and CDC OBEX support)
    < > Printer Gadget
    < > CDC Composite Device (Ethernet and ACM)
    < > CDC Composite Device (ACM and mass storage)
    < > Multifunction Composite Gadget
    < > HID Gadget
    <M> USBPIPE Gadget
    < > EHCI Debug Device Gadget
    < > USB Webcam Gadget
```

步骤 3 insmod libcomposite.ko (位置: drivers/usb/gadget/libcomposite.ko)

步骤 4 insmod g_usbpipe.ko (位置: drivers/usb/gadget/legacy/ g_usbpipe.ko)。

步骤 5 如果对端配置正确, USB 连接正常的话, 就会生成/dev/usbpipe0 设备节点。

步骤 6 运行 usbpipe-sample, 通过控制/dev/usbpipe0 节点完成自己的 host 端传输功能。



说明

可以通过 debugfs (mount debugfs 之后, cat /sys/kernel/debug/usbpipe/queues) 查看驱动发送接收队列的详细信息。

----结束



注意

- 为了简化设计
 - (1) 不支持 USB 插拔。
 - (2) 不支持 ko 的动态卸载。
- 只支持读写各一个通道。
- HOST 端的 IOCTL 目前的 device-sample 只支持了 close，在 host sample 异常退出之后 device-sample 也回收到错误码。

Cogobuy Only For ShenZhen FuShi ChanJing Industrial Technology Co., Ltd.



4 sample 操作说明

sample 操作步骤如下:

步骤 1 HOST DEVICE 之间通过 USB 连接好 (XVR 场景都是板级连接的)

步骤 2 准备好 HOST 端驱动和 sample

- 获取 usbpipe class 驱动代码编译为 ko, 或者直接获取 ko 二进制
- 编译 sample: arm-hixxxx-linux-gcc -Wall -Wextra -g -o usbpipe-sample usbpipe-sample.c -lpthread (arm-hixxxx-linux-gcc 为具体使用的编译器)。
- insmod usbpipe.ko (依赖 usb host 驱动需要先安装) 或者 modprobe usbpipe.ko

步骤 3 准备好 DEVICE 端驱动和 sample

- 获取 usbpipe gadget 驱动代码编译为 g_usbpipe.ko
- 编译 sample: arm-hixxxx-linux-gcc usbpipe-gadget.c -g -o usbpipe-gadget (arm-hixxxx-linux-gcc 为具体使用的编译器)。
- insmod g_usbpipe.ko (依赖的 usb device 驱动需要先安装) 或者 modprobe g_usbpipe.ko

步骤 4 执行 sample 程序

- Device 端, 执行 ./usbpipe-gadget
- Host 端, 执行 ./usbpipe-sample

在 HOST 端会出现如下的打印展示测试正常:



```
usbpipe-sample: info: Speed is 38.5649 MB/s
usbpipe-sample: info: Speed is 38.4109 MB/s
usbpipe-sample: info: RTT information:
usbpipe-sample: info: rounds: 256
usbpipe-sample: info: cur_rtt:26.119 ms
usbpipe-sample: info: avg_rtt:26.146 ms
usbpipe-sample: info: Speed is 38.3421 MB/s
usbpipe-sample: info: Speed is 38.31 MB/s
usbpipe-sample: info: RTT information:
usbpipe-sample: info: rounds: 512
usbpipe-sample: info: cur_rtt:26.087 ms
usbpipe-sample: info: avg_rtt:26.154 ms
usbpipe-sample: info: Speed is 38.3028 MB/s
usbpipe-sample: info: Speed is 38.2933 MB/s
```

注意

在 usbpipe-sample.c 文件的 data_thread 函数中，地址参数 data.addr 需要使用者根据环境分配内存自行设置，代码中亦有注释说明。usbpipe-sample, usbpipe-gadget 代码仅供参考，使用者可以根据情况自行修改以实现自己的需求。

----结束