Product Name		
Yuge ASR module		
Produce Version	Total 9 pages 共 9 页	
ALL		

域格ASR模块Android使用手册

V 1.30



上海域格信息技术有限公司

All rights reserved



目录

1.	修改驱动	3
2.	无线接口层介绍	5
3.	ril 库文件的使用	6
4.	调试	7
5.	带 GPS 模块调试	8
6.	注意事项	9



1. 修改驱动

ASR 模块在 Android 下,仅需加载 DIAG 口,AT 口及 modem 口,其他无关端口需过滤。以下提供两种解决方式,可根据实际情况选择处理。

1、kernel 版本支持 RSVD 的,在 option.c (路径一般为drivers/usb/serial/option.c)中的 option_ids 中添加 RSVD,可使驱动在加载时自动跳过 RSVD 指定的 interface。

在文件中找到 option_ids[]的数组,添加域格模块的 VID 和 PID,参照以下方法添加 VID、PID:

#define	ASR_VENDOR_ID	0x1286
#define	YUGA_PRODUCT_JD3	0x4E3C

设置 interface 0、1 不加载 option 驱动,添加内容如下:

/* Yuga products use ASR vendor ID */
{ USB_DEVICE(ASR_VENDOR_ID, YUGA_PRODUCT_JD3),
 .driver_info = RSVD(0) | RSVD(1) },

2、kernel 版本支持 blacklist 的,在 option.c (路径一般为 drivers/usb/serial/option.c)中的 option ids 中添加 blacklist,可使驱动在加载时自动跳过 blacklist 指定的 interface。

先在文件中找到 option_ids[]的数组,参照以下方法添加 VID、PID:

#define	YUGA VENDOR AC3	0x1286
#define	YUGA PRODUCT AC3	0x4E3C

设置 interface 0、1 不加载 option 驱动,添加内容如下:

```
static const struct option_blacklist_info YUGA_VENDOR_AC3_blacklist = {
    .reserved = BIT(0) | BIT(1),
};
```

. .

添加 blacklist 到 option_ids 数组中:

3、对于 kernel 版本不支持在 option_ids 数组中设置 blacklist 的,要先添加模块的 VID 和 PID。

```
static const struct usb_device_id option_ids[] = {
    { USB DEVICE(YUGA VENDOR AC3, YUGA PRODUCT AC3)},
```



```
#define YUGA_VENDOR_AC30x1286#define YUGA_PRODUCT_AC30x4E3C
```

在 option_probe 函数内判断当前 interface 号进行过滤:

编译完成后,通过查询命令: ls /dev/ttyU*,应有 ttyUSB0、ttyUSB1、ttyUSB2 三个端口,如下图 shanghaiyuge@shanghaiyuge-virtual-machine:~/桌面\$ ls /dev/ttyU* /dev/ttyUSB0 /dev/ttyUSB1 /dev/ttyUSB2

ppp 配置:

4G模块需要使用 ppp 协议进行拨号,内核配置中必须把 ppp 相关协议打开。以kernel2.6.7 内核为例, menuconfig 的路径如下

Device Drivers->Network device support->PPP 相关选项全部勾选。

RNDIS配置:

使用RNDIS,需要安装驱动 rndis_host.ko。一般发行版均已经编译好,直接运行即可。

如果嵌入式开发板可以配置内核的 rndis_host 为动态模块, 部分内核需要打开:

cd kernel

make menuconfig

Device Drivers \rightarrow Network device support \rightarrow USB Network Adapters \rightarrow Multi-purpose

USB Networking Framework



2. 无线接口层介绍

当数据卡驱动被内核加载并正确驱动后,该 4G模块才能进行无线通信功能的应用开发。 这在 Android 系统下称为无线接口层——RIL。即该文档所要说明的 RIL 驱动。



android 的 ril 位于应用程序框架与内核之间,分成了两个部分,一个部分是 rild,它负责 socket 与应用程序框架进行通信。另外一个部分是 Vendor RL,这个部分负责向下是通过两种方式与 radio 进行通信,它们是直接与 radio 通信的 AT 指令通道和用于传输数据包的通道,数据通道用于手机 的上网功能。也就是我们所提供的 RIL 驱动,是实现通信业务的核心功能模块。对于 RIL 的 java 框架 部分,也被分成了两个部分,一个是 RIL 模块,这个模块主要用于与下层的 rild 进行通信,另外 一个是 Phone 模块,这个模块直接暴露电话功能接口给应用开发用户,供他们调用以进行电话功 能的实现。这是属于 Android 应用程序的开发部分,简单了解即可。所以 RIL 驱动模块,必需是 针对不同的 4G 模块指令和通信业务功能定制的,以支不同应用需求。 建议开发人员在使用我们 的 RIL 驱动前,先了解下"4G 模块的特点和构造"。



3. ril 库文件的使用

将我们提供的 ril 库文件,修改成系统使用的库文件名后,push 到相应文件夹下。如将 libreference-ril.so 放到 /system/lib: adb push libreference-ril.so /system/lib



4. 调试

使用 adb shell logcat -b radio -v time 来获取 ril 日志。logcat 来获取详细的日志信息。调试如遇 到问题可以把日志反馈给我们。



5. 带 GPS 模块调试

将我们提供的 gps 库文件,修改成系统使用的库文件名后,push 到相应文件夹下。如将 gps.default.so 放到 /system/lib/hw: adb push gps.default.so /system/lib/hw

且需打开RNDIS,使用支持 RNDIS的 ril 库,拨号方式要用RNDIS。



6. 注意事项

要使用模块网络功能,务必将 wifi 和本地网络关闭,否则系统会把无线数据断开。

要使用 PPP 拨号时, persist.ril.data.prefer默认不设置或者设置为空。注意将 RNDIS 拨号去掉,否则 会导致网络冲突,造成频繁掉线。

要使用RNDIS拨号时,需设置属性persist.ril.data.prefer为rndis。