2928 平台 RDA5875Y 蓝牙模块调试心得

Authe:李东汉

Email:ldh@rock-chips.com

Date:2012-12-19

目录

2928 平台 RDA5875Y 蓝牙模块调试心得 1

一、 代码移植 1

1) 补丁<RDA5876_patch_11_22>的目录结构如下 1

2) 移植注意点 2

二、 软件简单流程 2

三、 硬件调试 4

四、 软件调试方法 5

五、 总结 6

一、 代码移植

1) 补丁<RDA5876_patch_11_22>的目录结构如下

```
├<device>
|   ├<rockchip>
|   |   ├<rk29sdk>
|   |   |   └init.rk29board.rc
├<external>
|   ├<bluetooth>
|   |   ├<bluez>
|   |   |   ├<tools>
|   |   |   |   ├Android.mk
|   |   |   |   ├hciattach.c
|   |   |   |   └hciattach_rda5875_5870E.c
├<frameworks>
|   ├<base>
|   |   ├<core>
|   |   |   ├<java>
|   |   |   |   ├<android>
|   |   |   |   |   ├<server>
|   |   |   |   |   |   └BluetoothAdapterStateMachine.java
|   |   ├<data>
|   |   |   ├<etc>
|   |   |   |   └handheld_core_hardware.xml
|   |   ├<services>
|   |   |   ├<java>
|   |   |   |   ├<com>
|   |   |   |   |   ├<android>
|   |   |   |   |   |   ├<server>
|   |   |   |   |   |   |   └SystemServer.java
├<kernel>
|   ├<drivers>
|   |   ├<bluetooth>
|   |   |   └hci_ldisc.c
|   |   ├<misc>
|   |   |   ├Kconfig
|   |   |   ├Makefile
|   |   |   └tcc_bt_dev.c
|   ├<net>
|   |   ├<bluetooth>
```

```
|   |   |   └hci_core.c
├<system>
|   ├<bluetooth>
|   |   ├<bluedroid>
|   |   |   ├Android.mk
|   |   |   └bluetooth.c
```

2) 移植注意点

可阅读附带的 readme.txt。另我在 init.rk2928board.rc 中 chmod 0777 /dev/tcc_bt_dev 后发现无效，随后移到 ueventd.rc 中去后 OK。

二、 软件简单流程

调试一个设备或者功能模块最重要的，莫过于清楚其走过的每一行代码，当然很多的系统调用没有必要去深究，也不会对特定模块的调试有任何影响，但必须清楚其实现的功能，输入输出数据含义。

App（应用层）

⇩

Framework（系统框架层）

⇩

HAL 层，再下面就是设备驱动层

上面的应用层和框架层公共的东西暂且不需理会，重点看下 HAL 层。通过 APP，framwork 下来启动的函数为 bluetooth.c 中的 int bt_enable()，其中 static int set_bluetooth_power(int on)为上下电操作函数，通过设备节点，最终会调用到 kernel 层 tcc_bt_dev.c 中的 int tcc_bt_power_control(int on_off)来对 IO 进行设置。

property_set("ctl.start", "hciattach")启动蓝牙服务。随后转到

Hciattach.c 中的 int main(int argc, char *argv[])中，首先解析 init.rc 中定义的服务字段参数:即解析 init.rk2928board.rc

service hciattach /system/bin/hciattach -n -s 115200 /dev/ttyS0 rda 1500000 flow

user bluetooth

disabled

oneshot

中的-n -s 115200 /dev/ttyS0 rda 1500000 flow

-s 后为设备初始速度

/dev/ttyS0 为设备节点

rda 相当于设备识别码

1500000 为设定的设备速度

flow 为使用流控

获得一些重要的参数后调用设备的初始化函数 init_uart(dev, u, send_break, raw),再通过 if (u->init && u->init(fd, u, &ti) < 0)调用 static int rda_init(int fd, struct uart_t *u, struct termios *ti)—> int RDABT_core_Intialization(int fd)向设备写入厂商提供的初始化数据。

如果到此没有出现错误,说明设备初始化成功。回到 bluetooth.c 的 bt_enable()中，接下的代码是一个 for 循环，先建立一个 bluetooth 的套接字，然后通过 ioctl 来和 bluez 的代码来打开蓝牙设备，可以重试 1000 次。接下来的代码就要跑到内核的 BlueZ 了。Kernel/net/bluetooth/hci_sock.c 中的 static int hci_sock_ioctl(struct socket *sock, unsigned int cmd, unsigned long arg)

```
    ……
{
    switch (cmd) {
    case HCIDEVUP:
        if (!capable(CAP_NET_ADMIN))
            return -EACCES;
```

```
        return hci_dev_open(arg);
……
}
……
}
```

hci_dev_open 位于 Kernel/net/bluetooth/hci_core.c 中，其调用的 hdev->open(hdev) 即我们在注册设备驱动时的 open 函数，位于 driver/Bluetooth/Hci_ldisc.c 中的 static int hci_uart_open(struct hci_dev *hdev)。代码到此，蓝牙算是真正打开了。

三、 硬件调试

1).确认连接是否正确。UART 的 TX,RX 是否接对，BT_REG_ON 是否拉高? VBATT 是否供电正常？26MHZ 时钟还有 32K 时钟是否供给?

2). 用独立 RTC IC 输出来的 CLK 因为是开漏输出所以要加 10K 上拉，否则无法驱动 5875Y。PMU 是 IO 口给出来的所以不用上拉。（硬件曾峥解释）

刚开始看了原厂给的文档说 32K 时钟可以不用，结果正是因为这个原因，调了差不多半个月。

3).只能发送不能接收。将蓝牙中断输出拉高，两个串口拉高

4). 流控主要是针对接收/发射缓冲而言,只要你 AP 的 UART 内部硬 FIFO size 小于 480bytes,就需要接 UART Flow control。

接法：

BT_CTS 接到 AP_RTS.

AP_CTS 接到地，BT_RTS 悬空。

注：流控低有效，两个可直接拉低先调试.

四、 软件调试方法

BT 手动测试命令：

1. 停止以下服务:

setprop ctl.stop bluetoothd

setprop ctl.stop hciattach

2. 关闭 BT 电源

echo 0 > /sys/class/rfkill/rfkill0/state

3. 打开 BT 电源

echo 1 > /sys/class/rfkill/rfkill0/state

4. 运行 hciattach 服务

setprop ctl.start hciattach

5. 打开 hci0 接口

hciconfig hci0 up

6. hciconfig –a

获取蓝牙设备的一些重要信息，如：蓝牙的物理地址，总线类型，协议类型等

7. 扫描

hcitool scan

以上都是标准的蓝牙系统调用接口，具体的设备上下电的节点可能不一样，RDA5875Y 的上下电方式为

关闭电源：echo 0 > dev/tcc_bt_dev

打开电源：echo 1 > dev/tcc_bt_dev

上面的命令是在调试的时候直接屏蔽 app 及 framwork 层，如同直接在终端直接调用 HAL 层的函数进行操作。以此排除 framwork 层带来的影响。所以这些命令都可以在 HAL 中找到对应的代码打开关闭服务即 property_set("ctl.start","hciattach")和 property_set("ctl.stop", hciattach)，上下电操作：set_bluetooth_power，……

五、 总结

这个 RDA 蓝牙调了将近两周，所以有很深刻的体会。希望把一些调试过程中遇到的问题和调试的方法写出来，免得以后再有人调试时少走弯路。由于水平有限，难免有不当之处，若有发现请指出。一起进步！