



XR829 Bluetooth Porting Guide(Android O)

Revision 1.0

Jun 26, 2018

Declaration

THIS DOCUMENTATION IS THE ORIGINAL WORK AND COPYRIGHTED PROPERTY OF XRADIO TECHNOLOGY ("XRADIO"). REPRODUCTION IN WHOLE OR IN PART MUST OBTAIN THE WRITTEN APPROVAL OF XRADIO AND GIVE CLEAR ACKNOWLEDGEMENT TO THE COPYRIGHT OWNER.

THE INFORMATION FURNISHED BY XRADIO IS BELIEVED TO BE ACCURATE AND RELIABLE. XRADIO RESERVES THE RIGHT TO MAKE CHANGES IN CIRCUIT DESIGN AND/OR SPECIFICATIONS AT ANY TIME WITHOUT NOTICE. XRADIO DOES NOT ASSUME ANY RESPONSIBILITY AND LIABILITY FOR ITS USE. NOR FOR ANY INFRINGEMENTS OF PATENTS OR OTHER RIGHTS OF THE THIRD PARTIES WHICH MAY RESULT FROM ITS USE. NO LICENSE IS GRANTED BY IMPLICATION OR OTHERWISE UNDER ANY PATENT OR PATENT RIGHTS OF XRADIO. THIS DATASHEET NEITHER STATES NOR IMPLIES WARRANTY OF ANY KIND, INCLUDING FITNESS FOR ANY PARTICULAR APPLICATION.

THIRD PARTY LICENCES MAY BE REQUIRED TO IMPLEMENT THE SOLUTION/PRODUCT. CUSTOMERS SHALL BE SOLELY RESPONSIBLE TO OBTAIN ALL APPROPRIATELY REQUIRED THIRD PARTY LICENCES. XRADIO SHALL NOT BE LIABLE FOR ANY LICENCE FEE OR ROYALTY DUE IN RESPECT OF ANY REQUIRED THIRD PARTY LICENCE. XRADIO SHALL HAVE NO WARRANTY, INDEMNITY OR OTHER OBLIGATIONS WITH RESPECT TO MATTERS COVERED UNDER ANY REQUIRED THIRD PARTY LICENCE.

Revision History

Version	Date	Summary of Changes
1.0	2018-6-26	Initial Version

表 0-1 Revision History

目 录

Declaration.....	2
Revision History.....	3
目 录.....	4
表.....	6
图.....	7
1 XRadio Bluetooth 软件框架.....	8
2 Linux kernel 移植.....	9
2.1 添加休眠唤醒与 FDI 模块.....	9
2.2 修改上电控制模块.....	10
2.3 修改板级包配置.....	12
2.3.1 配置供电信息.....	12
2.3.2 配置 UART 接口.....	13
2.3.3 配置上电控制模块信息.....	14
2.3.4 配置休眠唤醒模块信息.....	14
3 Android 移植.....	15
3.1 平台配置.....	15
3.1.1 配置 init.rc.....	15
3.1.2 配置蓝牙模组信息.....	16
3.1.3 配置厂商库信息.....	18
3.2 厂商库添加.....	19
3.3 HCI Interface 修改.....	19
3.3.1 添加 XRADIO_BLUETOOTH 和 BUILD_VERSION 宏开关.....	19
3.3.2 添加 bt hci-impl 版本信息.....	19
3.3.3 添加 FDI 调试接口.....	20
3.3.4 添加 xradio bt 功能的移植修改.....	20
3.4 bluedroid 修改.....	20

4 移植后检查点.....	22
4.1 检查驱动加载.....	22
4.2 检查蓝牙固件路径.....	22
4.3 检查软件版本号.....	22
5 移植 FAQ.....	23
5.1 如何单独编译模块.....	23
5.2 蓝牙打开失败.....	23
5.3 打开蓝牙时不断重启.....	24
5.4 蓝牙使用过程中重启.....	24

表

表 0-1	Revision History	3
--------------	-------------------------	----------



图 1-1	XRADIO Bluetooth 软件框架.....	8
图 2-1	Bluetooth driver 内核编译配置.....	10
图 2-2	Rfkill driver 内核编译配置.....	12
图 2-3	PCB 原理图中的 POWER TREE.....	12
图 2-4	PCB 原理图中 UART 的配置.....	13
图 2-5	PCB 原理图中休眠唤醒的配置.....	14
图 3-1	bluetooth 修改 patch 列表.....	20
图 4-1	检查蓝牙固件是否存在.....	22
图 5-1	查看是否配置了 power_sply.....	23

1 XRadio Bluetooth 软件框架

本文以 A50 平台移植为例（其它平台移植可以参考此文档），Android 版本为 Android8.1，Linux 版本为 Linux4.9。图 1-1 是 Android8.1 系统上 XRadio Bluetooth 软件框架，用户需要移植部分主要包括 BT Power 模块、BT Sleep 模块、Vendor Lib 模块、HCI Interface 模块，如图中标黄色的背景所示。

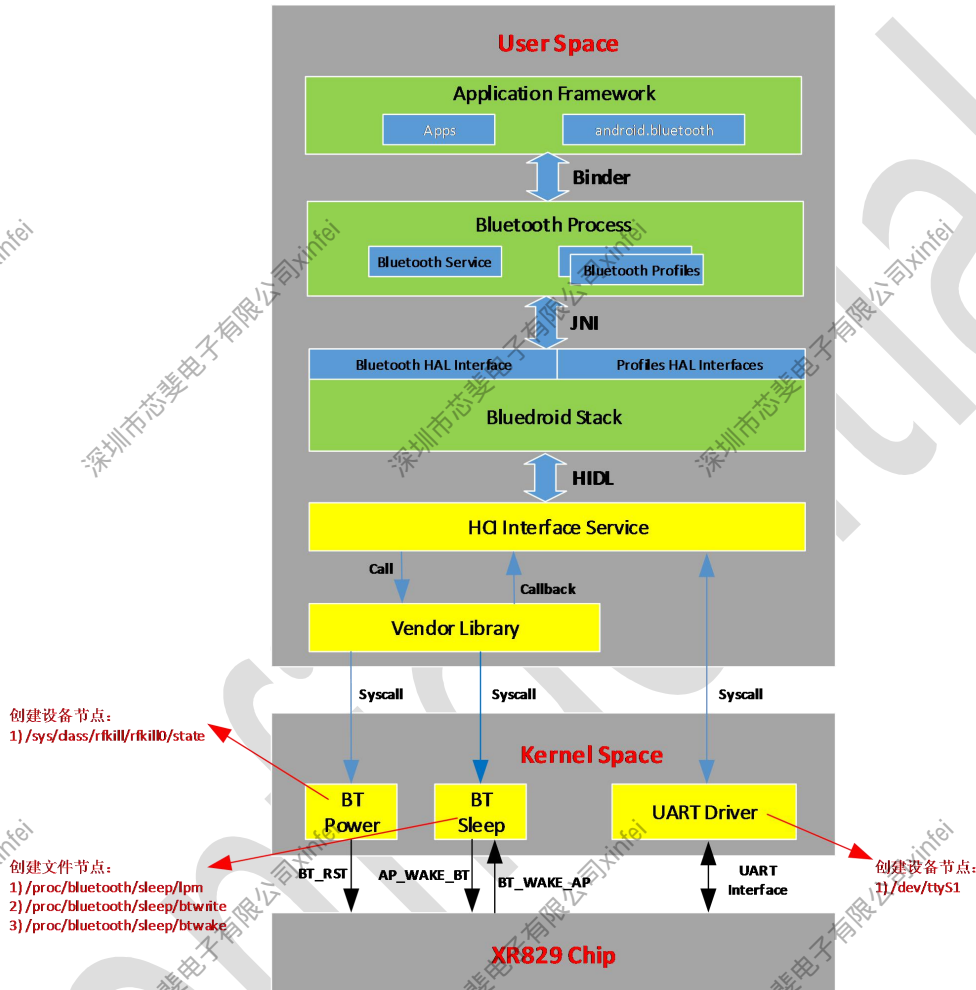


图 1-1 XRadio Bluetooth 软件框架

Bluetooth 软件框架有两个进程，一个进程是 com.android.bluetooth，负责蓝牙协议实现，如图中背景为绿色区域；一个进程是 android.hardware.bluetooth@1.0-service，负责硬件驱动实现，如图中背景为黄色区域。

2 Linux kernel 移植

XRadio Bluetooth Linux 移植部分主要工作包括上电控制模块、休眠唤醒模块、FDI 调试模块、板级包配置，移植的代码目录都在 lichee 目录下。下面以 32 位 Linux4.9 内核版本进行蓝牙内核移植过程的详细说明，文件目录结构各个内核版本可能有所区别，请注意区分。

2.1 添加休眠唤醒与 FDI 模块

Bluetooth 休眠唤醒驱动与 FDI 调试模块代码路径在 lichee/linux-4.9/drivers/bluetooth/目录下，移植步骤如下：

- (1) 拷贝 xradio_btfdi.c 和 xradio_btldm.c 文件至对应路径下。
- (2) 修改 Makefile 文件，添加如下信息。

```
obj-$(CONFIG_XR_BT_LPM) += xradio_btldm.o  
obj-$(CONFIG_XR_BT_FDI) += xradio_btfdi.o
```

- (3) 修改 Kconfig 文件，添加信息如下图所示。

```
config XR_BT_LPM  
    tristate "Xradio Bluetooth sleep driver support"  
    help  
        Bluetooth Bluetooth sleep Driver.  
        This driver provides the dynamic active power saving mechanism for  
        bluetooth radio devices.  
  
        Say Y here to compile support for bluesleep support into the kernel  
        or say M to compile it as module (bluesleep).  
  
config XR_BT_FDI  
    tristate "Xradio Bluetooth farmware debug interface support"  
    help  
        Bluetooth Bluetooth sleep Driver.  
        This driver provides the dynamic active power saving mechanism for  
        bluetooth radio devices.  
  
        Say Y here to compile support for bluesleep support into the kernel  
        or say M to compile it as module (bluesleep).
```

- (4) 配置 menuconfig，编译休眠唤醒模块。在 linux4.9 根目录下输入 make ARCH=arm menuconfig 命令，选择 “Networking support -> Bluetooth subsystem support -> Bluetooth device drivers”，页面配置如下图所示：

```
< > HCI USB driver
< > RTK HCI USB driver
< > HCI SDIO driver
< * > HCI UART driver
[*]   UART (H4) protocol support
[*]   BCSP protocol support
[ ]   Atheros AR300x serial support
[ ]   HCILL protocol support
[*]   Three-wire UART (H5) protocol support
[ ]   Intel protocol support
[ ]   Broadcom protocol support
[ ]   Qualcomm Atheros protocol support
[ ]   Intel AG6XX protocol support
[ ]   Marvell protocol support
< > HCI BCM203x USB driver
< > HCI BPA10x USB driver
< > HCI BlueFRTZ! USB driver
< > HCI VHCI (Virtual HCI device) driver
< > Broadcom Bluetooth Low Power Manager Support
< M > Realtek Bluesleep driver support
< M > Xradio Bluetooth sleep driver support
< M > Xradio Bluetooth firmware debug interface support
< > Marvell Bluetooth driver support
```

图 2-1 Bluetooth driver 内核编译配置

2.2 修改上电控制模块

Bluetooth 上电控制驱动代码路径在 `lichee/linux-4.9/drivers/misc/sunxi-rf/` 目录下，移植步骤如下：

(1) 修改 `sunxi-bluetooth.c` 文件，添加 `sunxi_bluetooth_set_power()` 函数接口并导入到系统符号表中，如下图所示：

```
static struct sunxi_bt_platdata *bluetooth_data = NULL;
static int sunxi_bt_on(struct sunxi_bt_platdata *data, bool on_off);
static DEFINE_MUTEX(sunxi_bluetooth_mutex);

void sunxi_bluetooth_set_power(bool on_off)
{
    struct platform_device *pdev;
    int ret = 0;
    if(!bluetooth_data)
        return;

    pdev = bluetooth_data->pdev;
    mutex_lock(&sunxi_bluetooth_mutex);
    if(on_off != bluetooth_data->power_state){
        ret = sunxi_bt_on(bluetooth_data, on_off);
        if(ret)
            dev_err(&pdev->dev, "set power failed\n");
    }
    mutex_unlock(&sunxi_bluetooth_mutex);
}
EXPORT_SYMBOL_GPL(sunxi_bluetooth_set_power);
```

(2) 由于需要使用 sunxi_bt_platdata 成员信息，定义一个静态全局指针变量 bluetooth_data 保持信息，在 sunxi_bt_probe() 函数中赋值，在 sunxi_bt_remove() 函数中清空，如下图所示：

```
static int sunxi_bt_probe(struct platform_device *pdev)
{
    struct device_node *np = pdev->dev.of_node;
    struct device *dev = &pdev->dev;
    struct sunxi_bt_platdata *data;
    struct gpio_config config;
    const char *power,*io_regulator;
    int ret = 0;

    data = devm_kzalloc(dev, sizeof(*data), GFP_KERNEL);
    if (!dev)
        return -ENOMEM;

    data->pdev = pdev;
    bluetooth_data = data;
    ...
}
```

```
static int sunxi_bt_remove(struct platform_device *pdev)
{
    struct sunxi_bt_platdata *data = platform_get_drvdata(pdev);
    struct rfkill *rfk = data->rfkill;

    platform_set_drvdata(pdev, NULL);

    if(rfk){
        rfkill_unregister(rfk);
        rfkill_destroy(rfk);
    }

    if (!IS_ERR_OR_NULL(data->lpo))
        clk_disable_unprepare(data->lpo);

    bluetooth_data = NULL;

    return 0;
}
```

(3) 修改 lichee/linux4.9/drivers/misc/Makefile 文件，添加以下信息：

```
obj-$(CONFIG_SUNXI_RFKILL) += sunxi-rf/
```

(4) 修改 lichee/linux4.9/drivers/misc/Kconfig 文件，添加以下信息：

```
source "drivers/misc/sunxi-rf/Kconfig"
```

(5) 配置内核编译：

在 lichee/linux4.9 目录下输入 make ARCH=arm menuconfig 命令，依次选择 “Device Drivers -> Misc devices”，

选中“Allwinner rkfill driver”，页面配置如下：

```
< > Analog Devices Digital Potentiometers
< > Dummy IRQ handler
< > Integrated Circuits ICS932S401
< > Enclosure Services
< > Medfield Avago APDS9802 ALS Sensor module
< > Intersil ISL29003 ambient light sensor
< > Intersil ISL29020 ambient light sensor
< > Taos TSL2550 ambient light sensor
< > BHL770GLC / SFH7770 combined ALS - Proximity sensor
< > APDS990X combined als and proximity sensors
< > Honeywell HMC6352 compass
< > Dallas DS1682 Total Elapsed Time Recorder with Alarm
< > Texas Instruments DAC7512
< > FSA9480 USB Switch
< > Lattice ECP3 FPGA bitstream configuration via SPI
[ ] Generic on-chip SRAM driver
[*] Per-UID statistics
[ ] Per-TASK statistics
<*> Memory freq/bandwidth time statistics
< > Silicon Labs C2 port support ----
EEPROM support ---->
Texas Instruments shared transport line discipline ---->
< > STMicroelectronics LIS3LV02Dx three-axis digital accelerometer (SPI)
< > STMicroelectronics LIS3LV02Dx three-axis digital accelerometer (I2C)
*** Altera FPGA firmware download module ***
< > Altera FPGA firmware download module
*** Intel MIC Bus Driver ***
*** SCIF Bus Driver ***
*** VOP Bus Driver ***
*** Intel MIC Host Driver ***
*** Intel MIC Card Driver ***
*** SCIF Driver ***
*** Intel MIC Coprocessor State Management (COSM) Drivers ***
*** VOP Driver ***
< > Line Echo Cancellor support
[*] Allwinner rkfill driver
```

图 2-2 Rkfill driver 内核编译配置

2.3 修改板级包配置

需要结合具体的 PCB 原理图进行硬件板级包配置。下面以 AllWinner A50-a3 平台为例，给出相关配置。配置文件路径为 **lichee/tools/pack/chips/sun8iw15p1/configs/a3/sys_config.fex**。

2.3.1 配置供电信息

(1) 查看 PCB 硬件原理图的 POWER TREE：

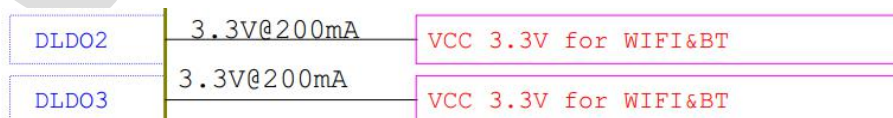


图 2-3 PCB 原理图中的 POWER TREE

(2) 修改 sys_config.fex 文件，在[power_sply]元素中添加 dldo2_vol 和 dldo3_vol 的配置：

```

;-----
; power setting
;note:
;dc1sw_vol=1003300, mean set dc1sw to 3300mV and enable this dc
;dc1sw_vol=3300, mean set dc1sw to 3300mV but disable this dc
;-----

[power_sply]
dc1sw_vol      = 3300
;aldo1_vol     = 3300
;clldo3_vol    = 1800
dc1sw2_vol     = 100900
dc1sw3_vol     = 100900
dc1sw5_vol     = 1001500
dldo2_vol      = 3300
dldo3_vol      = 3300
;aldo5_vol     = 2500
    
```

2.3.2 配置 UART 接口

(1) 查看 PCB 原理图中 UART 的相关配置：

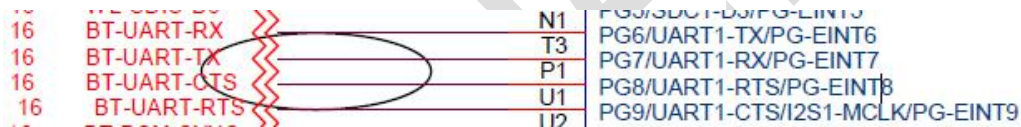


图 2-4 PCB 原理图中 UART 的配置

(2) 进行 uart1 的相关配置：

```

[uart1]
uart1_used      = 1
uart1_port      = 1
uart1_type      = 4
uart1_tx        = port:PG6<2><1><default><default>
uart1_rx        = port:PG7<2><1><default><default>
uart1_rts       = port:PG8<2><1><default><default>
uart1_cts       = port:PG9<2><1><default><default>
uart1_regulator = "vcc-io"
    
```

2.3.3 配置上电控制模块信息

```

;-----
;bluetooth configuration
;bt_used          bluetooth used (0- no used, 1- used)
;bt_power         bluetooth module power supply
;bt_io_regulator   bluetooth io power supply
;bt_rst_n         bt power on
;-----

[bt]
compatible        = "allwinner,sunxi-bt"
bt_used           = 1
bt_power          = "vcc-wifi"
bt_io_regulator    = "vcc-wifi-io"
bt_rst_n          = port:PL02<1><default><default><0>

```

2.3.4 配置休眠唤醒模块信息

(1) 查看 PCB 原理图:

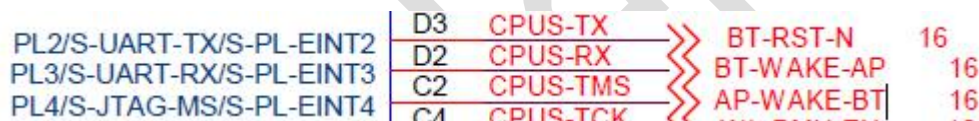


图 2-5 PCB 原理图中休眠唤醒的配置

(2) 进行相应配置:

```

;-----
;bluetooth lpm configuration
;btlpm_used       bluetooth lpm used (0- no used, 1- used)
;uart_index       bluetooth uart num
;bt_wake          ap wake bt, output io port for ap
;bt_hostwake      bt wake ap, input io port for ap
;-----

[btlpm]
compatible        = "allwinner,sunxi-btlpm"
btlpm_used        = 1
uart_index        = 1
bt_wake           = port:PL04<1><default><default><1>
bt_hostwake       = port:PL03<6><default><default><0>
bt_host_wake_invert = 0
bt_wake_invert    = 0

```


3 Android 移植

XRADIO Bluetooth Android 移植部分主要工作包括模块平台配置、厂商库驱动以及 bluebird 协议栈的适配补丁,移植的目录都在 android 目录下。下面基于 Android8.1 版本进行蓝牙 android 部分移植说明,请注意 android 版本带来的文件目录结构区别。

3.1 平台配置

Bluetooth 平台配置文件路径在 android/device/softwinner/目录下,移植步骤如下:

3.1.1 配置 init.rc

修改 android/device/softwinner/common/init.wireless.xradio.rc 文件

- (1) 添加 xradio_btspm 和 xradio_btfdi 开机启动加载脚本
- (2) 修改 bluetooth 相关文件节点权限

on fs

```
# xradio driver
```

```
insmod /system/vendor/modules/xradio_mac.ko
```

```
insmod /system/vendor/modules/xradio_core.ko
```

```
insmod /system/vendor/modules/xradio_btspm.ko
```

```
# bluetooth xradio firmware debugfs driver
```

```
insmod /system/vendor/modules/xradio_btfdi.ko
```

```
# UART device
```

```
chmod 0660 /dev/ttyS1
```

```
chown bluetooth net_bt_admin /dev/ttyS1
```

```
# bluetooth power up/down interface
```

```
chmod 0660 /sys/class/rfkill/rfkill0/state
```

```
chmod 0660 /sys/class/rfkill/rfkill0/type
```

```
chown bluetooth net_bt_admin /sys/class/rfkill/rfkill0/state
```

```
chown bluetooth net_bt_admin /sys/class/rfkill/rfkill0/type
```

```
write /sys/class/rfkill/rfkill0/state 1
```

```
write /sys/class/rfkill/rfkill0/state 0
```

```
# bluetooth LPM
```

```
chmod 0220 /proc/bluetooth/sleep/lpm
```

```
chmod 0220 /proc/bluetooth/sleep/btwrite
```

```
chmod 0220 /proc/bluetooth/sleep/btwake
```

```
chown bluetooth net_bt_admin /proc/bluetooth/sleep/lpm
```

```
chown bluetooth net_bt_admin /proc/bluetooth/sleep/btwrite
```

```
chown bluetooth net_bt_admin /proc/bluetooth/sleep/btwake
```

```
# bluetooth MAC address programming
```

```
mkdir /data/misc/bluedroid 0770 bluetooth net_bt_admin
```

```
chown bluetooth net_bt_admin /system/etc/bluetooth
```

```
chown bluetooth net_bt_admin /data/misc/bluetooth
```

```
.....
```

3.1.2 配置蓝牙模组信息

修改 android/device/softwinner/venus-a3/BoradConfig.mk 文件:

- (1) 选择使用 xradio 蓝牙模组
- (2) 添加 xradio bt 配置


```

...
# 2. Bluetooth Configuration

# make sure BOARD_HAVE_BLUETOOTH is true for every bt vendor

BOARD_BLUETOOTH_VENDOR := xradio

# 2.1 broadcom bt configuration

# BOARD_HAVE_BLUETOOTH_NAME: ap6210/ap6212/ap6330/ap6335
ifeq ($(BOARD_BLUETOOTH_VENDOR), broadcom)
    BOARD_HAVE_BLUETOOTH := true
    BOARD_HAVE_BLUETOOTH_BCM := true
    BOARD_HAVE_BLUETOOTH_NAME := ap6255
    BOARD_BLUETOOTH_BDROID_BUILDCFG_INCLUDE_DIR :=
$(TOP_DIR)device/softwinner/$(basename $(TARGET_DEVICE))/configs/bluetooth/
endif

# 2.2 realtek bt configuration
ifeq ($(BOARD_BLUETOOTH_VENDOR), realtek)
    BOARD_HAVE_BLUETOOTH := true
    BOARD_HAVE_BLUETOOTH_RTK := true
    BOARD_HAVE_BLUETOOTH_RTK_COEX := true
    BOARD_HAVE_BLUETOOTH_NAME := rtl8723cs
    BOARD_BLUETOOTH_BDROID_BUILDCFG_INCLUDE_DIR :=
$(TOP_DIR)device/softwinner/$(basename $(TARGET_DEVICE))/configs/bluetooth/
    include hardware/realtek/bluetooth/firmware/rtlbtfw_cfg.mk
endif

# 2.3 xradio bt configuration
ifeq ($(BOARD_BLUETOOTH_VENDOR), xradio)

```

```
BOARD_HAVE_BLUETOOTH := true

#XRADIO_BT Configuration

BOARD_HAVE_BLUETOOTH_XRADIO := true

BOARD_BLUETOOTH_BDROID_BUILDCFG_INCLUDE_DIR :=
$(TOP_DIR)device/softwinner/$(basename $(TARGET_DEVICE))/configs/bluetooth/

TARGET_USE_BOOSTUP_OPZ := true

include hardware/xradio/bt/firmware/xradio-bt.mk

endif
```

3.1.3 配置厂商库信息

拷贝蓝牙厂商库配置信息文件 vnd_venus-a3.txt 至 android/device/softwinner/venus-a3/configs/bluetooth/ 目录下，文件内容如下：

```
#Set baudrate to 1500000

UART_TARGET_BAUD_RATE = 1500000

BLUETOOTH_UART_DEVICE_PORT = "/dev/ttyS1"

FW_PATCHFILE_LOCATION = "/system/vendor/etc/firmware/"

LPM_SLEEP_MODE = TRUE

BT_WAKE_VIA_PROC = TRUE

USE_CONTROLLER_BDADDR = FALSE

BTVND_DBG = TRUE

BTCMD_DBG = TRUE

BTHW_DBG = TRUE

UPIO_DBG = TRUE

VNDUSERIAL_DBG = TRUE

#SCO_PCM_ROUTING = 0x00

#SCO_PCM_IF_CLOCK_RATE = 0x04

#SCO_PCM_IF_FRAME_TYPE = 0x00
```

```
#SCO_PCM_IF_SYNC_MODE = 0x00  
#SCO_PCM_IF_CLOCK_MODE = 0x00  
#PCM_DATA_FMT_SHIFT_MODE = 0x00  
#PCM_DATA_FMT_FILL_BITS = 0x00  
#PCM_DATA_FMT_FILL_METHOD = 0x00  
#PCM_DATA_FMT_FILL_NUM = 0x03  
#PCM_DATA_FMT_JUSTIFY_MODE = 0x0
```

【备注】：LPM_SLEEP_MODE 是使能休眠唤醒功能的宏开关，用户根据方案需求自行修改。

3.2 厂商库添加

Bluetooth 厂商库驱动代码在 android/hardware/xradio/bt/目录下，由于比较独立，直接拷贝 bt 文件夹至 xradio 目录下即可，然后使用“mm”命令进行编译测试是否可以通过。

【备注】：libbt-vendor 目录下 vnd_buildcfg.mk 使用 gen-buildcfg.sh 脚本对 vnd_venus-a3.txt 文件进行解析，生成 vnd_buildcfg.h 文件。

3.3 HCI Interface 修改

Bluetooth HCI Interface 代码在 android/hardware/interfaces/bluetooth/目录下，这是 Android8.1 蓝牙子系统变动较大的一块，将原有 HCI 驱动是静态库方式变成一个独立的服务运行在后台，通过 HIDL 通信机制与 bluedroid 协议栈进行交互，所有的移植代码都加上 XRADIO_BLUETOOTH 宏控制，并以 patch 的形式给出。具体的移植步骤如下：

3.3.1 添加 XRADIO_BLUETOOTH 和 BUILD_VERSION 宏开关

打上以下补丁：

android/hardware/interfaces/0001-Bluetooth-Get-BOARD_BLUETOOTH_VENDOR-from-BoardConfi.patch

3.3.2 添加 bt hci-impl 版本信息

打上此补丁：android/hardware/interfaces/0001-V1.0.1.patch

3.3.3 添加 FDI 调试接口

添加对 xradio 蓝牙固件调试接口 (bt-fdi) 的支持，打上以下补丁：

android/hardware/interfaces/0001-V1.0.2.patch

3.3.4 添加 xradio bt 功能的移植修改

打上以下补丁：android/hardware/interfaces/0001-V1.0.3.patch

主要修改点：

- (1) 设置 hci_read 线程启动时机
- (2) 设置 MAC 地址随机产生方式，默认由 HOST 随机数产生 MAC 地址

3.4 bluebird 修改

bluebird 是蓝牙协议栈实现，代码路径在 android/system/bt/目录下。所有的移植修改代码都加上 XRADIO_BLUETOOTH 宏开关控制，并以 patch 的形式给出：

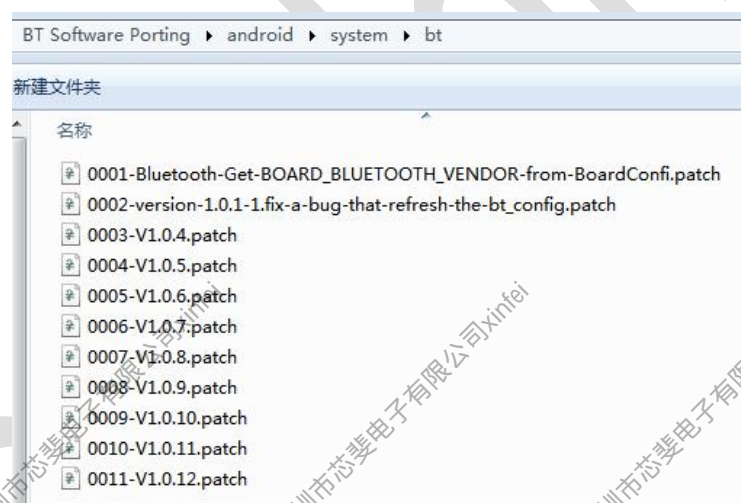


图 3-1 bluebird 修改 patch 列表

请依次打上上述 patch。

上述 patch 的主要内容如下：

- (1) 添加 XRADIO_BLUETOOTH 宏和 BUILD_VERSION 开关宏。修改的文件是 build/fluoride.go。
- (2) 屏蔽不支持的 HCI 指令。主要包括保存 firmware log 命令、codec 设置命令、BLE 厂商命令，主要修改是 hci_layer.cc、controller.cc、btm_ble_adv_filter.cc 文件
- (3) 修改 command timeout 超时时间，默认是 2s，根据芯片性能这里设置为 8s，修改的文件是 hci_layer.cc。
- (4) 添加 bluebird 版本和 bt 版本管理信息。在 btif_core.cc 文件的 btif_enable_bluetooth_evt()函数中添加

相关信息。

4 移植后检查点

4.1 检查驱动加载

(1) 上电驱动模块文件节点，系统启动自动生成，路径目录如下。

```
cat /sys/class/rfkill/rfkilll0/type
```

(2) 休眠唤醒与 FDI 驱动，系统启动自动加载，通过 lsmod 可以查看。

```
xradio_btfdi      19550  0
xradio_btldpm     17494  0
```

4.2 检查蓝牙固件路径

蓝牙固件路径为：/vendor/etc/firmware/fw_xr829_bt.bin，通过“ls -l”命令可以查看：

```
venus-a3:/vendor/etc/firmware # ls -al
ls -al
total 556
drwxr-xr-x  2 root shell    4096 2018-06-28 15:32 .
drwxr-xr-x 10 root shell    4096 2018-06-19 20:13 ..
-rw-r--r--  1 root root     2184 2018-06-19 20:13 boot_xr829.bin
-rw-r--r--  1 root root    73004 2018-06-19 20:13 etf_xr829.bin
-rw-r--r--  1 root root   173488 2018-06-15 16:22 fw_xr829.bin
-rw-r--r--  1 root root   300592 2018-06-27 11:28 fw_xr829_bt.bin
-rw-r--r--  1 root root     744 2018-06-19 20:13 sdd_xr829.bin
```

图 4-1 检查蓝牙固件是否存在

4.3 检查软件版本号

软件版本号包括厂商库版本、HCI Interface 版本、Bluedroid 版本、Firmware 版本，通过“adb shell getprop”命令可以查看。

名称	方法	版本号
Vendor	adb shell getprop bluetooth.driver.version	1.1.13
Hci Interface	adb shell getprop persist.bluetooth.bthci.version	1.0.3
Bluedroid	adb shell getprop persist.service.bdroid.version	1.0.12
FirmWare	adb shell getprop persist.service.bdroid.btfwv	7.10.91

5 移植 FAQ

5.1 如何单独编译模块

模块编译先分别到 `android/hardware/xradio/bt`、`android/hardware/interfaces`、`android/system/bt` 目录下使用“`touch *`”命令更新目录下所有文件的时间戳，然后在对应目录下使用“`mm`”命令进行编译，再把生成 `libbt-vendor.so`、`android.hardware.bluetooth@1.0-impl.so`、`bluetooth.default.so` 推入到目标机器目录，最后更改执行权限并重启。以 32 位系统为例：

```
adb push libbt-vendor.so /vendor/lib/libbt-vendor.so
adb push android.hardware.bluetooth@1.0-impl.so /vendor/lib/hw/
adb push bluetooth.default.so /system/lib/hw/
adb shell chmod 777 /vendor/lib/libbt-vendor.so
adb shell chmod 777 /vendor/lib/hw/android.hardware.bluetooth@1.0-impl.so
adb shell chmod 777 /system/lib/hw/bluetooth.default.so
```

5.2 蓝牙打开失败

蓝牙开启过程中首先会进行 `brom sync` 波特率步骤，如果出现 `sync` 失败，蓝牙就会打开失败，可能产生的原因有以下几个方面：

（1）芯片没有上电

查看 `sys_config.fex` 文件[`power_sply`]字段是否设置，通过“`cat /sys/class/regulator/dump`”命令查看，如下图所示。

```
2|venus-a3:/sys/class/regulator # cat dump
cat dump
axp22x_dc5ldo : enabled 0 900000 supply_name:
axp22x_dc1sw : enabled 1 3300000 supply_name: vcc-ldc
axp22x_ldoi01 : disabled 0 3300000 supply_name:
axp22x_ldoi00 : enabled 1 3300000 supply_name: vcc-ctp
axp22x_eldo3 : disabled 0 1800000 supply_name:
axp22x_eldo2 : disabled 0 2800000 supply_name:
axp22x_eldo1 : enabled 1 1800000 supply_name: avcc
axp22x_dldo4 : disabled 0 1800000 supply_name:
axp22x_dldo3 : enabled 1 3300000 supply_name: bt-vcc-wifi
axp22x_dldo2 : enabled 1 3300000 supply_name: bt-vcc-wifi-io
axp22x_dldo1 : enabled 1 1800000 supply_name: vcc-cpvin
axp22x_aldo3 : enabled 0 3300000 supply_name:
axp22x_aldo2 : enabled 0 2500000 supply_name:
axp22x_aldo1 : enabled 0 1800000 supply_name:
axp22x_rtc : enabled 0 3000000 supply_name:
axp22x_dcdc5 : enabled 0 1500000 supply_name:
axp22x_dcdc4 : enabled 0 900000 supply_name:
axp22x_dcdc3 : enabled 0 940000 supply_name:
axp22x_dcdc2 : enabled 1 900000 supply_name: vdd-sys
axp22x_dcdc1 : enabled 9 3300000 supply_name: vcc-io vcc-io vcc-io vcc-io vcc-io vcc-nand vcc-pd vcc-io vcc-io
```

图 5-1 查看是否配置了 `power_sply`

(2) 芯片 core 没有进行 reset

通过 “cat /sys/class/rfkill/rfkill0/state” 命令查看 BT_RST 电平是否为 1

(3) 芯片没有 wakeup

通过 “echo 1 > /proc/bluetooth/sleep/btwake” 命令进行测试。

5.3 打开蓝牙时不断重启

固件下载成功后会发送 hci_reset 指令进行软复位，如果出现失败，会导致蓝牙一直重启尝试重新复位，hci_reset 失败可能的原因有以下几个方面：

(1) uart 配置不合理，默认发送第 1 条 hci_reset 指令 uart 配置为 115200 波特率并且打开硬件流控，可以通过两次使用 “cat /sys/bus/platform/drivers/uart/uart1/ctrl_info” 命令，查看波特率是否正确和 rx 数值是否有增长。

(2) 32K 时钟没有提供，默认情况下 32K 由 AP 提供，可以通过万用表量对应 PIN 脚是否有时钟输出。

5.4 蓝牙使用过程中重启

HCI Cmd Timeout 会导致蓝牙协议栈重启，产生的原因是 host 端在规定时间内（默认为 8s）没有收到 device 端回复的 status event 或 complete event，常用的调试定位方式如下：

(1) 确保 bt_hci 的 Trace_level 已调整，使用 “adb logcat -v time -s bt_hci” 命令查看定位是哪个 opcode cmd 产生 cmd timeout。

(2) 导出 hcilog 文件，分析该条 hci cmd 格式和参数是否符合协议规范结合实际场景的 Logcat 定位相关问题。