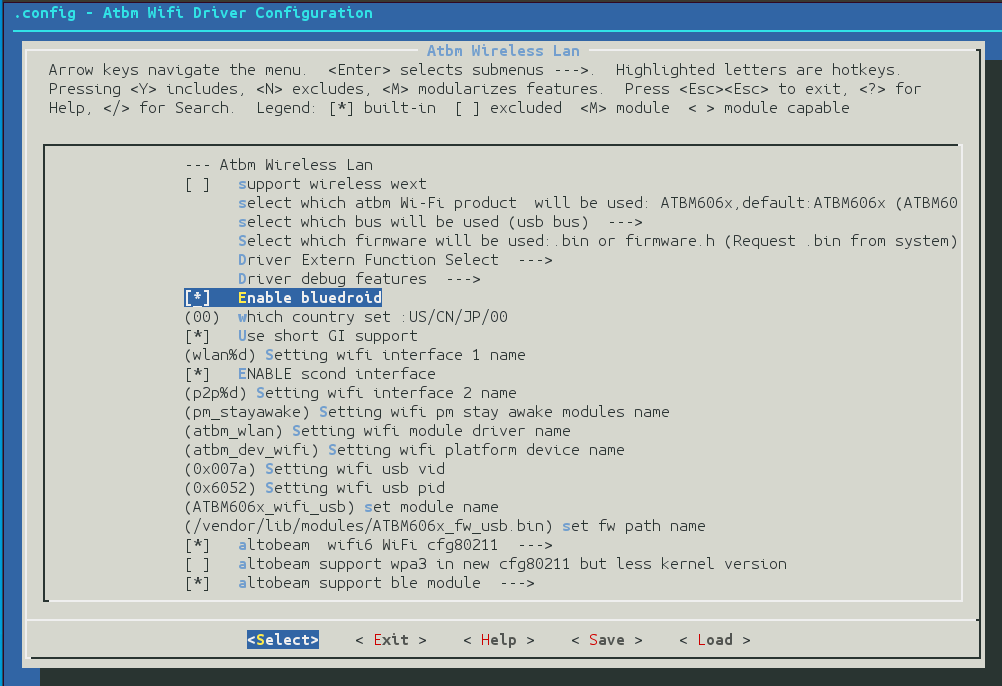
Android BT HAL 库移植

1. Driver

make menuconfig ;选中Enable bluedroid：



1. Hal

将altobeam包拷贝至hardware目录下。

* hardware\altobeam\libbt\atbm.mk文件将复制fw.bin到指定目录，该目录要与kernel配置一致,默认/vendor/etc/firmwar。fw.bin 位置：hardware\altobeam\libbt\conf\firmware。

|  |
| --- |
| CUR\_PATH := hardware/altobeam/libbt  PRODUCT\_COPY\_FILES += \  $(CUR\_PATH)/conf/firmware/fw.bin:$(TARGET\_COPY\_OUT\_VENDOR)/etc/firmware/fw.bin |

* device/rockchip/xxxxx/wifi\_bt.mk添加：

|  |
| --- |
| **diff --git a/device/rockchip/rk3399/wifi\_bt.mk b/device/rockchip/rk3399/wifi\_bt.mk**  **index 6d53792560..559f0e6811 100644**  **--- a/device/rockchip/rk3399/wifi\_bt.mk**  **+++ b/device/rockchip/rk3399/wifi\_bt.mk**  @@ -39,5 +39,8 @@  # rda587x, #like rtl8188+rda587x  # mt6622, #like rtl8188+mt6622  #  -BOARD\_CONNECTIVITY\_VENDOR := Broadcom  -BOARD\_CONNECTIVITY\_MODULE := ap6xxx  +#BOARD\_CONNECTIVITY\_VENDOR := Broadcom  +#BOARD\_CONNECTIVITY\_MODULE := ap6xxx  +  +BOARD\_CONNECTIVITY\_VENDOR := AltoBeam  +BOARD\_CONNECTIVITY\_MODULE := atbm606x |

* device/rockchip/common/wifi\_bt\_commod.mk添加：

|  |
| --- |
| **diff --git a/device/rockchip/common/wifi\_bt\_common.mk b/device/rockchip/common/wifi\_bt\_common.mk**  **index 9e157683b6..c6d5d2a2b7 100644**  **--- a/device/rockchip/common/wifi\_bt\_common.mk**  **+++ b/device/rockchip/common/wifi\_bt\_common.mk**  @@ -37,5 +37,15 @@ BLUETOOTH\_ENABLE\_FM := false  endif  endif  +  +ifeq ($(strip $(BOARD\_CONNECTIVITY\_VENDOR)), Altobeam)  +BOARD\_HAVE\_BLUETOOTH := true  +BOARD\_HAVE\_BLUETOOTH\_ATBM := true  +  +ifeq ($(strip $(BOARD\_CONNECTIVITY\_MODULE)), atbm606x)  +BLUETOOTH\_USE\_BPLUS := true  +BLUETOOTH\_ENABLE\_FM := false  +endif  +endif |

* device/rockchip/common/device.mk 添加：

|  |
| --- |
| **diff --git a/device/rockchip/common/device.mk b/device/rockchip/common/device.mk**  **index 47fe75c4b2..94c3f62cd5 100644**  **--- a/device/rockchip/common/device.mk**  **+++ b/device/rockchip/common/device.mk**  @@ -794,6 +794,10 @@ ifeq ($(strip $(BOARD\_HAVE\_BLUETOOTH\_RTK)), true)  include hardware/realtek/rtkbt/rtkbt.mk  endif  +ifeq ($(strip $(BOARD\_HAVE\_BLUETOOTH\_ATBM)), true)  +include hardware/altobeam/libbt/atbm.mk  +endif  +  ifeq ($(strip $(TARGET\_BOARD\_PLATFORM\_PRODUCT)), box)  include device/rockchip/common/samba/rk31\_samba.mk  PRODUCT\_COPY\_FILES += \ |

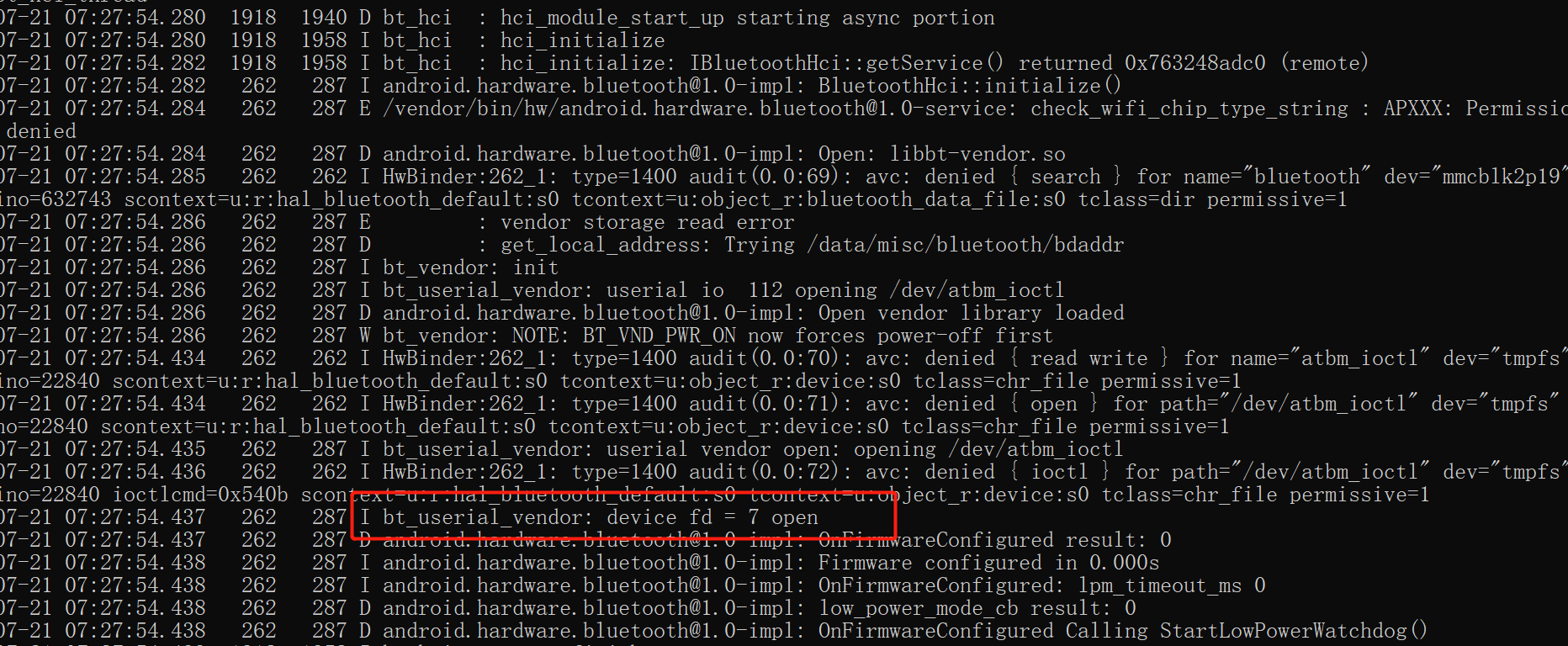
* 修改Android/system/core/rootdir/init.rc ，修改/dev/atbm\_ioctl权限，添加：

|  |
| --- |
| diff --git a/system/core/rootdir/init.rc b/system/core/rootdir/init.rc  index 0fa9febdd3..20f382beab 100644  --- a/system/core/rootdir/init.rc  +++ b/system/core/rootdir/init.rc  @@ -820,6 +820,8 @@ on property:security.perf\_harden=0  on property:security.perf\_harden=1  write /proc/sys/kernel/perf\_event\_paranoid 3    + chmod 777 /dev/atbm\_ioctl  # on shutdown  # In device's init.rc, this trigger can be used to do device-specific actions  # before shutdown. e.g disable watchdog and mask error handling |

* system修改,配置可发现与可连接模式。

|  |
| --- |
| diff --git a/system/bt/btif/src/btif\_core.cc b/system/bt/btif/src/btif\_core.cc  index aa77979acf..036104c2d3 100644  --- a/system/bt/btif/src/btif\_core.cc  +++ b/system/bt/btif/src/btif\_core.cc  @@ -929,7 +929,9 @@ bt\_status\_t btif\_set\_adapter\_property(const bt\_property\_t\* property) {    case BT\_SCAN\_MODE\_CONNECTABLE\_DISCOVERABLE:  disc\_mode = BTA\_DM\_GENERAL\_DISC;  + disc\_mode |= BTA\_DM\_BLE\_GENERAL\_DISCOVERABLE;  conn\_mode = BTA\_DM\_CONN;  + conn\_mode |= BTA\_DM\_BLE\_CONNECTABLE;  break;    default: |

* Libbt-vendor.so库加载日志，用logcat指令查看，设备打开成功：



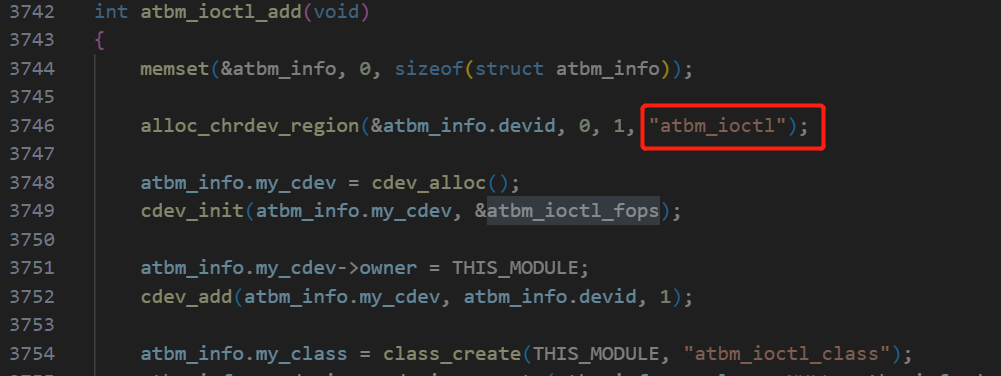
* 驱动修改

驱动部分只修改了字符设备驱动atbm\_ioctl\_fops ，将发到我们协议栈的数据通过文件节点/dev/atbm\_ioctl传给bluedroid.

流程：

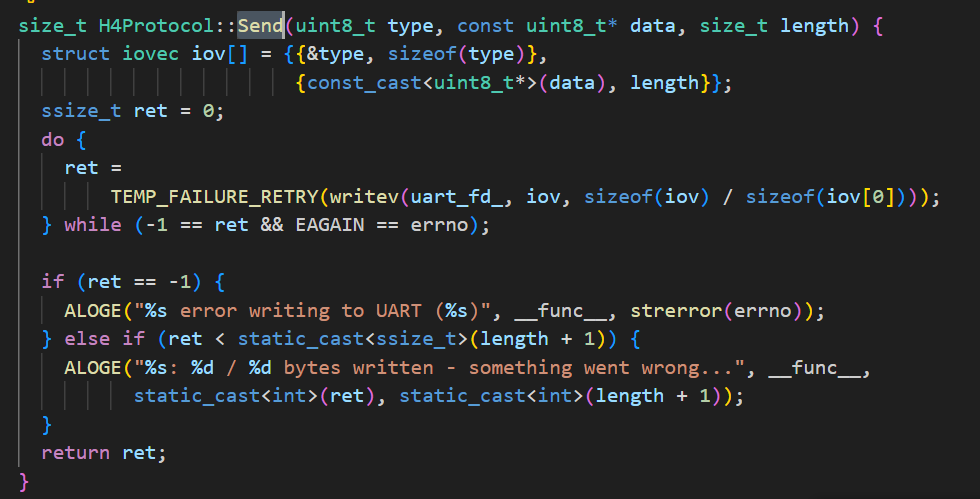
1. 在打开文件节点/dev/atbm\_ioctl时，配置ble()共存。
2. read阻塞。读到数据先放到rx\_skb\_queue 队列中，再通过bt\_read\_wait唤醒read读取队 列里面的数据。
3. 通过atbm\_ioctl\_write发送hci数据.

文件节点的名称修改:

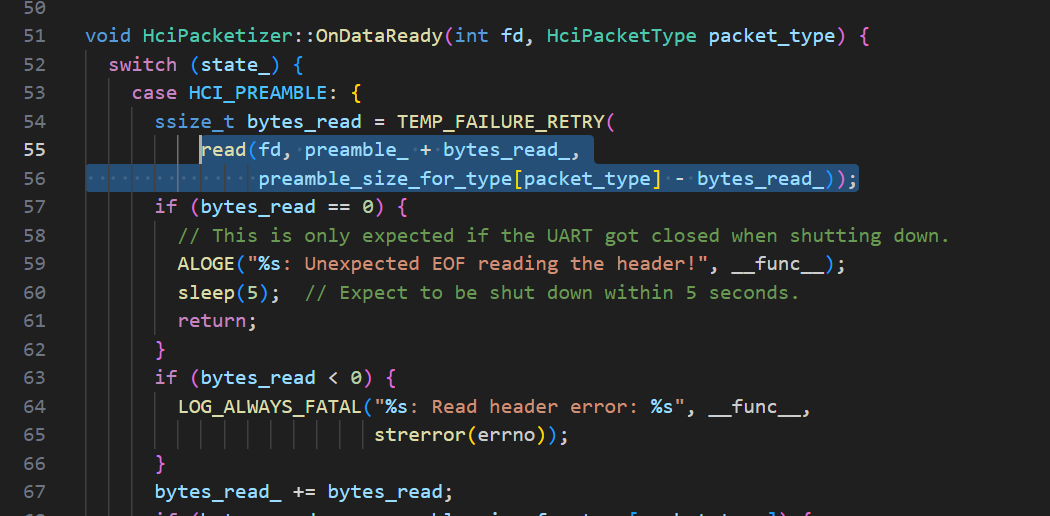


1. libbt-vendor.so通过文件描述符读写hci数据的接口：

Android\hardware\interfaces\bluetooth\1.0\default\h4\_protocol.cc



Android\hardware\interfaces\bluetooth\1.0\default\hci\_packetizer.cc

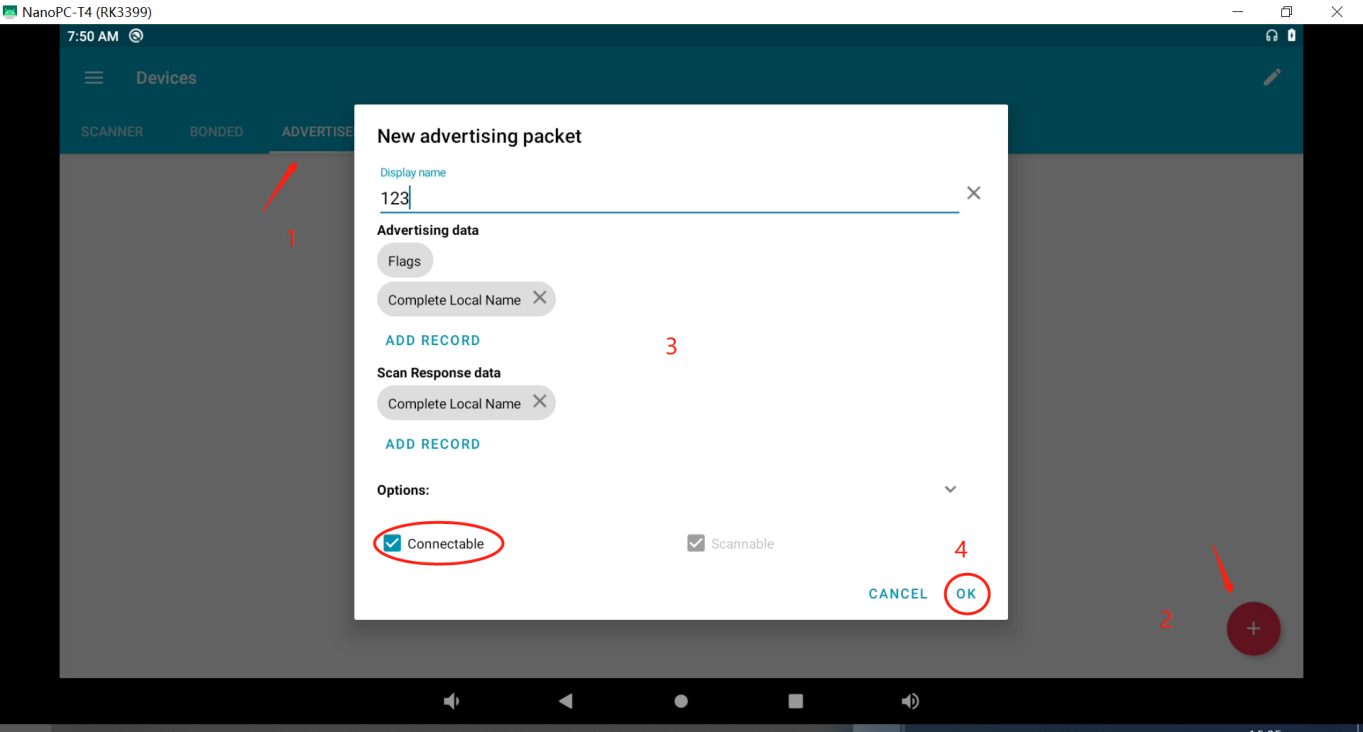


* 自测：

测试设备无显示屏情况下，准备工具：

1. 调试工具adb.exe
2. 安卓投屏工具Scrcpy.exe
3. BLE调试工具nRF.Connect.apk

* 手机端安装并打开nRF.Connect.apk 创建一个广播设备



* 设备端：usb连接设备端的otg口

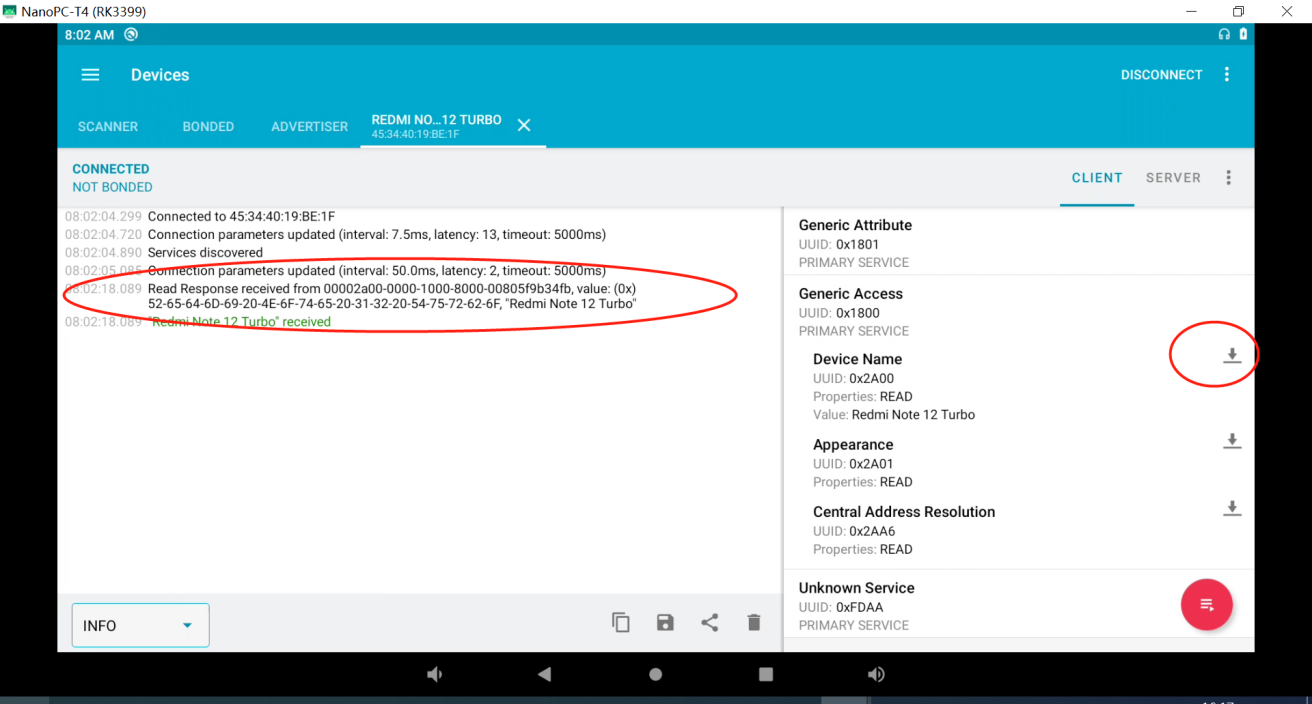
Win+r 输入cmd 打开命令运行窗口，安装nRF.Connect.apk：

|  |
| --- |
| C:\>adb root & adb remount  adbd is already running as root  remount succeeded  C:\>adb install C:\doc\tool\nRF.Connect.4.24.3.apk  Performing Streamed Install  Success |

* 运行Scrcpy.exe投屏，打开nRF.Connect工具，扫描手机设备，连接。



* 连接成功后，获取属性值,测试成功：



* 日志导出

查看Android系统日志：logcat

查看Kernel日志：dmesg 或 cat /dev/kmsg