



AIC8800D
Wi-Fi6/BT5.0 SoC
蓝牙语音移植手册

Revision: 1.1

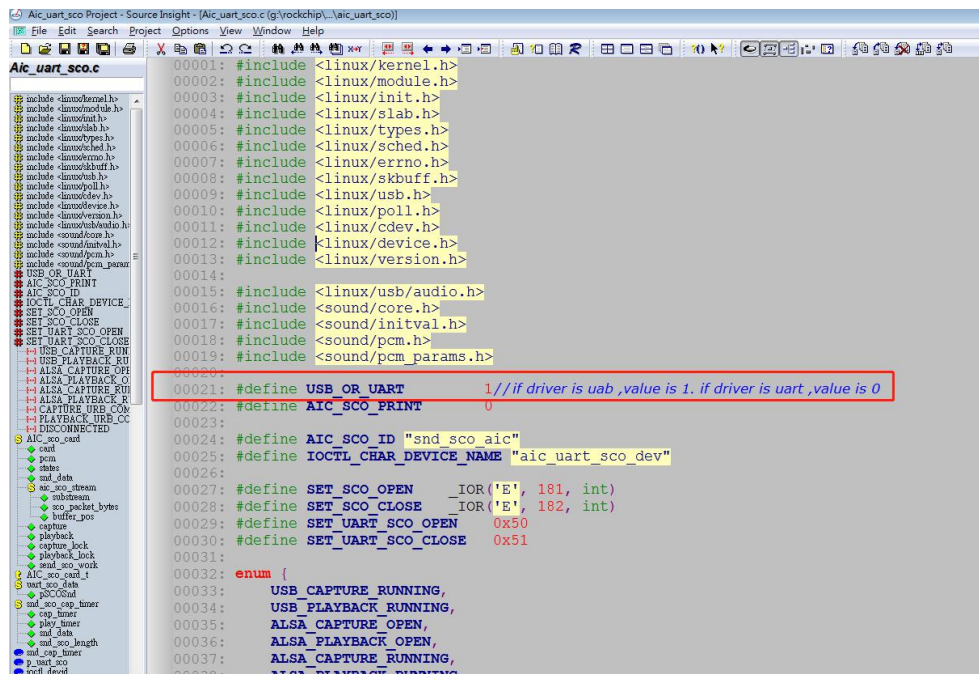
2022/05/26

历史更新记录

时间	修改内容	修订人	版本
2022/05/13	初版	Aiden	1.0
2022/05/26	新增 Q&A 栏位	Aiden	1.1

编译 aic_uart_sco 驱动

选择 USB 或是 Uart 之后编译并且设定开机自动加载



bluedroid 中添加补丁

可对比 bt_stack_patch 中的 orig 和 mod

esco_parameters.cc: android/system/bt/device/src/esco_parameters.cc

btm_sco.cc: android/system/bt/stack/btm/btm_sco.cc

btm_main.cc: android/system/bt/stack/btm/btm_main.cc

注意 bluedroid 中 HFP 和 HSP 的配置！！

需要注意 bdroid_buildcfg.h 中的 BTIF_HF_SERVICES

BTA_HFP_SERVICE_MASK: 支持 HFP

BTA_HSP_SERVICE_MASK: 支持 HSP

这个只能二选一，建议选 BTA_HFP_SERVICE_MASK，因为有些设备不支持 HSP

选择适合的 profile

在 android/packages/apps/Bluetooth/res/values/config.xml 或

在

android/device/rockchip/rk356x_box/rk356x_box/overlay/packages/apps/Bluetooth/res/values/config.xml 修改

(注意先改 overlay 之下的 config.xml)

Source 以下参数为 true:

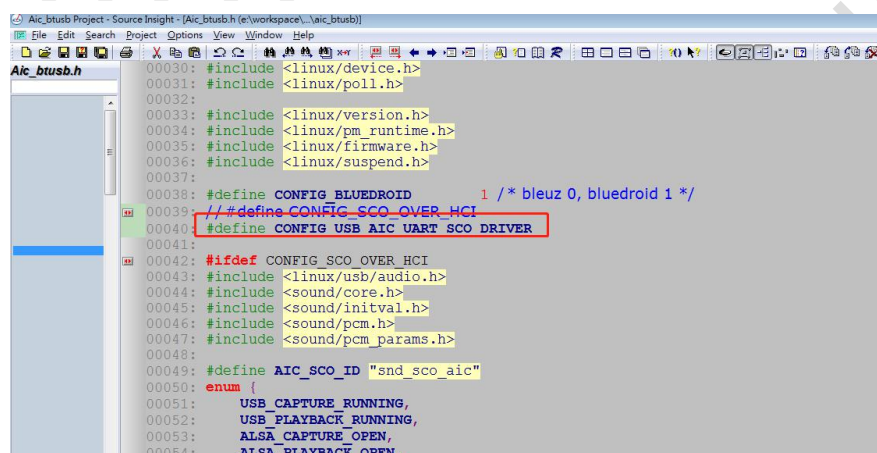
profile_supported_a2dp
profile_supported_hs_hfp

Sink 以下参数为 true:

profile_supported_a2dp_sink
profile_supported_hfpcclient

设定 aic_btusb (如果是 uart 可跳过该步骤)

打开 CONFIG_USB_AIC_UART_SCO_DRIVER



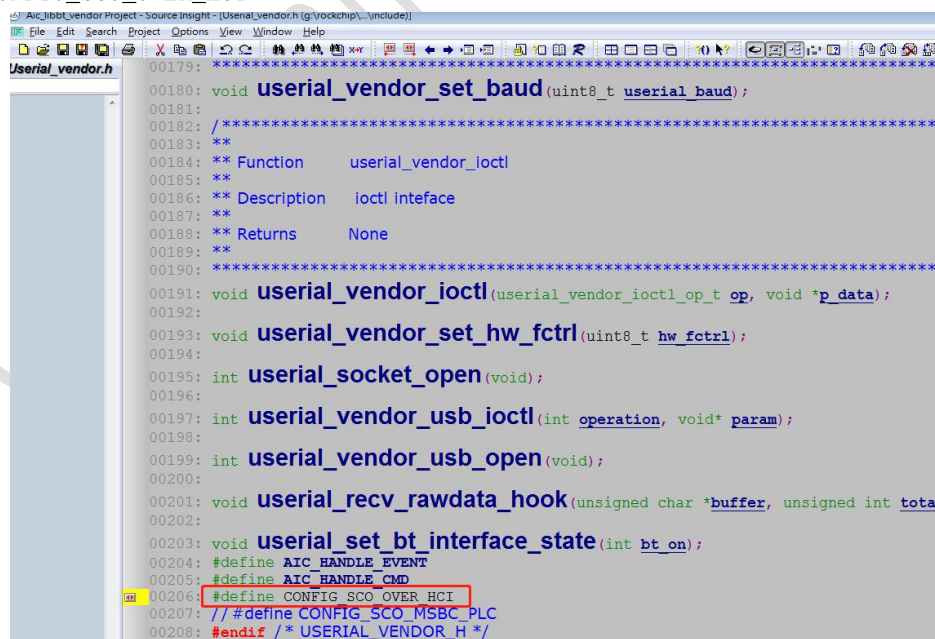
```

00030: #include <linux/device.h>
00031: #include <linux/poll.h>
00032:
00033: #include <linux/version.h>
00034: #include <linux/pm_runtime.h>
00035: #include <linux/firmware.h>
00036: #include <linux/suspend.h>
00037:
00038: #define CONFIG_BLUEROID 1 /* bleuz 0, blueoid 1 */
00039: // #define CONFIG_SCO_OVER_HCI
00040: #define CONFIG_USB_AIC_UART_SCO_DRIVER
00041:
00042: #ifdef CONFIG_SCO_OVER_HCI
00043: #include <linux/usb/audio.h>
00044: #include <sound/core.h>
00045: #include <sound/initval.h>
00046: #include <sound/pcm.h>
00047: #include <sound/pcm_params.h>
00048:
00049: #define AIC_SCO_ID "snd_sco_aic"
00050: enum {
00051:     USB_CAPTURE_RUNNING,
00052:     USB_PLAYBACK_RUNNING,
00053:     ALSA_CAPTURE_OPEN,
00054:     ALSA_PLAYBACK_OPEN,

```

设定 libbt-vendor (如果是 uart 可跳过该步骤)

打开 CONFIG_SCO_OVER_HCI



```

00179: *****
00180: void serial_vendor_set_baud(uint8_t serial_baud);
00181:
00182: /*****
00183: **
00184: ** Function    serial_vendor_ioctl
00185: **
00186: ** Description  ioctl interface
00187: **
00188: ** Returns     None
00189: **
00190: *****/
00191: void serial_vendor_ioctl(serial_vendor_ioctl_op_t op, void *p_data);
00192:
00193: void serial_vendor_set_hw_fctrl(uint8_t hw_fctrl);
00194:
00195: int serial_socket_open(void);
00196:
00197: int serial_vendor_usb_ioctl(int operation, void* param);
00198:
00199: int serial_vendor_usb_open(void);
00200:
00201: void serial_recv_rawdata_hook(unsigned char *buffer, unsigned int total);
00202:
00203: void serial_set_bt_interface_state(int bt_on);
00204: #define AIC_HANDLE_EVENT
00205: #define AIC_HANDLE_CMD
00206: #define CONFIG_SCO_OVER_HCI
00207: // #define CONFIG_SCO_MSBC_PLC
00208: #endif /* SERIAL_VENDOR_H */

```

设定 aic_usb 和 aic_uart_sco 权限

在 init.xxx.rc 当中配置

```
chmod 0660 /dev/aic_uart_sco_dev
chown bluetooth net_bt /dev/aic_uart_sco_dev
```

系统编译

编译 SDK 并且将 img 刷道板子中

打开蓝牙测试进行语音测试

打开蓝牙在板子中的/dev/snd/检查是否新增声卡

Source 测试:

1. 安装可以用蓝牙耳机录音的 app
2. 点击录音
3. tinycap /data/test/test.wav -D 2 -d 0 -c 1 -r 8000 -b 16 -p 384 -n 2 (USB 384,Uart 480)录制声音，露好后退出 tinycap
4. tinyplay /data/test/test.wav -D 2 -d 0 -c 1 -r 8000 -b 16 -p 384 -n 2 (USB 384,Uart 480)播放录制声音

Sink 测试:

1. 拨打电话
2. tinycap /data/test/test.wav -D 2 -d 0 -c 1 -r 8000 -b 16 -p 384 -n 2 (USB 384,Uart 480)录制声音，对端说话后将 tinycap 退出
3. tinyplay /data/test/test.wav -D 2 -d 0 -c 1 -r 8000 -b 16 -p 384 -n 2 (USB 384,Uart 480)播放录制声音，此时对端即可听到刚刚录制的声音

Q&A