
EA6XXX 无线芯片移植指导

Seekwave Confidential

修订记录

版本	修订日期	作者	描述
1.0.0	2023-6-5	Liang.Huang	EA6XXXX 芯片 Android 移植文档初稿
1.0.1	2023-6-21	Liang.Huang	Android 13 移植修改

目录

一、	用户层移植	1
1.	BOARD 配置	1
2.	VENDOR HAL 库移植	1
3.	通过 libwifi_hal 加载驱动	3
4.	STATION 适配	4
5.	HOSTAPD 适配	5
二、	驱动移植	5
1.	配置 defconfig	5
2.	配置 Makefile 及 Kconfig	5
3.	驱动日志等级	6
4.	国家码配置	7
5.	MAC 地址	7
6.	固件及校准文件	8

一、 用户层移植

1. BOARD 配置

1) 打开 11AX 功能

在 device/rockchip/<device-name>/BoardConfig.mk 文件中添加如下配置：

```
WIFI_FEATURE_HOSTAPD_11AX := true
```

使能 HOSTAPD 11ax 功能；

2) 配置 wpa_config.txt

在 device/rockchip/common/wpa_config.txt 文件中添加如下配置：

```
[seekwave]
/vendor/bin/hw/wpa_supplicant
-O/data/vendor/wifi/wpa/sockets
-puse_p2p_group_interface=1
-g@android:wpa_wlan0
```

参考 PATCH： device_rockchip_common.patch

2. VENDOR HAL 库移植

源码路径：

```
hardware/seekwave/wlan/wifi_hal/
```

修改 Android.mk

在 hardware/rockchip/wifi/wifi_hal/common/Android.mk 中添加 libwifi-hal-skw；

```
LOCAL_REQUIRED_MODULES := \
    libwifi-hal-bcm \
    libwifi-hal-rtk \
    libwifi-hal-bes \
    libwifi-hal-aic \
    libwifi-hal-skw
```

添加 hardware/rockchip/wifi/wifi_hal/vendor/seekwave/Android.mk

```
LOCAL_PATH := hardware/seekwave/wlan/wifi_hal/
# Make the HAL library
# =====
include $(CLEAR_VARS)
LOCAL_CFLAGS := -Wall \
                -Werror \
                -Wno-format \
                -no-reorder \
                -Wno-unused-function \
                -Wno-unused-parameter \
                -Wno-unused-private-field \
                -Wno-unused-variable \
                -Wno-unused-parameter

LOCAL_C_INCLUDES := \
                external/libnl/include \
                $(call include-path-for,
libhardware_legacy)/hardware_legacy \
                external/wpa_supplicant_8/src/drivers \
                external/boringssl/include \
                external/boringssl/src/crypto/digest \
                external/boringssl/src/crypto/evp/
LOCAL_HEADER_LIBRARIES := libutils_headers liblog_headers
LOCAL_SHARED_LIBRARIES := \
                librkwifi-ctrl \
                libcrypto \
                libnl \
                libutils \
                libcutils \
                liblog

LOCAL_SRC_FILES := \
                main.cpp \
                wifi_command.cpp

LOCAL_MODULE := libwifi-hal-skw
LOCAL_PROPRIETARY_MODULE := true

include $(BUILD_SHARED_LIBRARY)
```

参考 PATCH: hardware_rockchip_wifi.patch

3. 通过 libwifi_hal 加载驱动

1) 修改 Android.mk

修改 frameworks/opt/net/wifi/libwifi_hal/Android.mk, 添加 libwifi-hal-skw 库。

```
ifeq ($(BOARD_WLAN_DEVICE), seekwave)
    LIB_WIFI_HAL := libwifi-hal-skw
endif
```

2) 添加 EA6621Q 信息

在 frameworks/opt/net/wifi/libwifi_hal/rk_wifi_ctrl.cpp 文件中添加 EA6621Q 芯片信息;

添加 VID & PID:

```
static wifi_device supported_wifi_devices[] = {
    {"BES2600", "be57:2002"},
    {"AIC8800", "5549:0145"},
    {"SKW6160", "0000:0000"},
    {"SKW6160", "0483:5720"},
    {"SKW6160", "0483:5721"},
    {"SKW6160", "3607:6160"},
};
```

添加驱动配置如下:

```
#define SKW6160_DRIVER_MODULE_PATH    WIFI_MODULE_PATH"skw.ko"
#define SKW6160_DRIVER_MODULE_NAME    "skw"
#define SKW_WIFI_HAL                  "libwifi-hal-skw.so"

const wifi_file_name module_list[] =
{
    {"AIC8800", AIC8800_DRIVER_MODULE_NAME,
    AIC8800_DRIVER_MODULE_PATH, UNKOWN_DRIVER_MODULE_ARG, AIC_WIFI_HAL},
    {"SKW6160", SKW6160_DRIVER_MODULE_NAME,
    SKW6160_DRIVER_MODULE_PATH, UNKOWN_DRIVER_MODULE_ARG, SKW_WIFI_HAL},
};
```

参考 PATCH: frameworks_opt_net_wifi.patch

4. STATION 适配

1) WPA_SUPPLICANT 适配

修改 external/wpa_supplicant_8/wpa_supplicant/main.c, 添加 Wi-Fi TYPER 如下:

```
#define SKW_MODULE_NAME "[seekwave]"
int main(int argc, char *argv[])
{
    ...
    if(argc == 2) {
        ...
        if (0 == strncmp(wifi_type, "RTL", 3)) {
            ...
        } else if (0 == strncmp(wifi_type, "SKW", 3)) {
            ret = read_wpa_param_config(SKW_MODULE_NAME,argv[1]);
        } else {
            ...
        }
    }
}
```

参考 PATCH: external_wpa_supplicant_8.patch

2) P2P 接口适配

修改 hardware/interfaces/wifi/1.6/default/wifi_chip.cpp 文件如下:

```
std::string getPredefinedP2pIfaceName() {
    ...
    if ((0 == strncmp(wifi_type, "AP", 2))
        || (0 == strncmp(wifi_type, "SPRDWL", 6))
        || (0 == strncmp(wifi_type, "AIC", 3))
        || (0 == strncmp(wifi_type, "BES2600", 6))
        || (0 == strncmp(wifi_type, "SKW", 3))) {
        property_set("vendor.wifi.direct.interface", "p2p-dev-wlan0");
        property_get("wifi.direct.interface", buffer.data(), "p2p-dev-wlan0");
    }
    ...
}
```

```
std::pair<WifiStatus, std::vector<hidl_string>>
WifiChip::getP2pIfacesInternal() {
    ...
    if ((0 == strncmp(wifi_type, "AP", 2))
        || (0 == strncmp(wifi_type, "SPRDWL", 6))
        || (0 == strncmp(wifi_type, "AIC", 3))
        || (0 == strncmp(wifi_type, "BES2600", 6))
        || (0 == strncmp(wifi_type, "SKW", 3))
    ) {
        property_set("vendor.wifi.direct.interface", "p2p-dev-wlan0");
    }
    ...
}
```

参见 PATCH: hardware_interfaces.patch

5. HOSTAPD 适配

打开 WIFI_FEATURE_HOSTAPD_11AX 宏，使能 hostapd 11ax 功能，见 1.1.1 中修改；

热点支持能力见附件 I

二、 驱动移植

1. 配置 defconfig

在 kernel defconfig 中打开如下宏定义

```
CONFIG_WLAN_VENDOR_SEEKWAVE=m
CONFIG_WEXT_PRIV=y
CONFIG_WIRELESS_EXT=y
```

2. 配置 Makefile 及 Kconfig

修改 kernel-5.10/drivers/net/wireless/rockchip_wlan/目录 Makefile 及 Kconfig

Makefile 修改如下：

```
obj-$(CONFIG_WLAN_VENDOR_SEEKWAVE) += seekwave/
```

Kconfig 修改如下：

```
source "drivers/net/wireless/rockchip_wlan/seekwave/Kconfig"
```

3. 驱动日志等级

3) 日志等级定义

驱动日志等级定义如下：

```
SKW_ERROR  
SKW_WARN  
SKW_INFO  
SKW_DEBUG  
SKW_DETAIL
```

4) 默认日志等级

默认 Wi-Fi 驱动日志等级为 SKW_WARN；可通过修改 Makefile 中宏定义 SKW_LOG_LEVEL 设置默认日志等级；例如：

```
ccflags-y += -DSKW_LOG_LEVEL=SKW_DEBUG
```

5) 运行时日志等级

通过向 /sys/kernel/debug/skwifi/log_level 写入对应日志等级，修改运行时日志级别；例如：

```
echo "DEBUG" > /sys/kernel/debug/skwifi/log_level
```

设置日志等级关键字如下（不区分大小写）：

```
ERROR/WARN/INFO/DEBUG/DETAI
```

6) 查看日志等级

通过读取 `/sys/kernel/debug/skwifi/log_level` 获得当前日志等级，例如：

```
cat /sys/kernel/debug/skwifi/log_level

Log Level: DEBUG      [ERROR|WARN|INFO|DEBUG|DETAIL]

command log: disable
event   log: disable
dump    log: disable
scan    log: disable
timer   log: disable
state   log: enable
work    log: disable
```

4. 国家码配置

EA6XXXX 无线驱动支持两种国家码方案；

方案一

该方案使用外部管制域数据库，即数据库由内核或用户层维护，国家码改变时，驱动处理内核通知完成国家码切换；

方案二

该方案驱动使用内部管制域数据库（打开 `SKW_REGD_SELF_MANAGED` 宏），此时驱动国家码不受全局国家码影响；可通过 `ioctl` 命令修改驱动当前国家码

5. MAC 地址

EA6XXXX 无线驱动通过以下方式获得 MAC 地址，在无法获取 MAC 地址或 MAC 地址无效的情况下，驱动生成随机 MAC 地址；

1. EFUSE MAC 地址

芯片在产测中将 MAC 地址写入到 EFUSE 中，驱动读取 EFUSE 地址作为默认 MAC 地址。

2. 加载驱动时传入 MAC 地址参数

加载驱动时，通过参数传入驱动；

3. NV 获得 MAC 地址

需要客户提供 NV 读写接口，加载驱动时读取 NV 分区 MAC 地址作为默认 MAC 地址

MAC 地址优先顺序如下：

参数 MAC 地址 > NV MAC 地址 > EFUSE MAC 地址 > 随机 MAC 地址

6. 固件及校准文件

EA6XXX 芯片校准文件及固件保存在 vendor/rockchip/common/wifi/firmware/ 目录下；

固件：

```
RAM_RW_KERNEL_DRAM.bin
ROM_EXEC_KERNEL_IRAM.bin
```

校准文件：

```
EA6621Q_SEEKWAVE_R00000.bin
EA6621Q_SEEKWAVE_R00002.bin
EA6621Q_SEEKWAVE_R00005.bin
```

校准文件源文件：

```
EA6621Q_SEEKWAVE_R00000.ini
EA6621Q_SEEKWAVE_R00002.ini
EA6621Q_SEEKWAVE_R00005.ini
```

附录 I SOFTAP 功能列表

序号	功能列表	描述
1	最大连接数	28
2	ACL 功能	支持
3	隐藏 SSID 功能	支持
4	支持带宽	20/40/80 MHz
5	802.11i	WPA/WPA2/WPA3
6	WEP	WEP40/WEP104