

## 泰芯 Linux WiFi SMAC 驱动开发指南



保密等级	A	泰芯 Linux WiFi SMAC 驱动开发指南	文件编号	
发行日期	2022-8-4		文件版本	V1.2

修订记录

日期	版本	描 述	修订人
2022/8/4	v1.2	修改蓝牙配网开发说明	DY
2022/7/27	v1.1	新增蓝牙配网自定义协议说明	DY
2022/6/23	v1.0	初始版本	DY

泰芯保密文件

	珠海泰芯半导体有限公司 TaiXin Semiconductor Co., Limited	珠海市高新区港湾一号科创园港 11 栋 3 楼
---	--	-------------------------

版权所有侵权必究 Copyright © 2022 by TaiXin Semiconductor All rights reserved
--

保密等级	A	泰芯 Linux WiFi SMAC 驱动开发指南	文件编号	
发行日期	2022-8-4		文件版本	V1.2

目录

1 概述 .....	1
2 Linux Kernel 编译配置 .....	1
3 WiFi Driver 开发使用说明 .....	2
3.1 hgic_smac .....	2
3.1.1 hgic_smac 驱动文件 .....	2
3.1.2 hgic_smac 编译说明 .....	3
3.1.3 hgic_smac 加载说明 .....	4
3.1.4 hgic_smac proc fs 接口 .....	4
3.1.5 hgic_smac 辅助模块 .....	5
3.1.6 hgic_smac 调试与测试 .....	5
3.1.7 hgic_smac BLE 配网 .....	6
3.2 固件下载 .....	6

泰芯保密文件

## 1 概述

泰芯 SMAC 驱动需要内核支持 MAC80211/CFG80211, WiFi 协议栈运行在主控端。支持 ieee80211 bgn 协议, 支持 AP/STA/P2P 模式。对应的 WiFi 固件为 vx.x.x.1 类型。

## 2 Linux Kernel 编译配置

SMAC 驱动支持 SDIO 和 USB 两种接口, 在编译 kernel 需要打开以下功能:

1. 根据选择使用的接口 (sdio/usb), 打开对应的支持模块

- (1) sdio 接口: 打开 `Device Driver` → `MMC/SD/SDIO card support`, 以及对应的 mmc host driver。
- (2) usb 接口: 打开 `Device Driver` → `USB support`, 以及对应的 usb host driver。

**注:** 编译驱动时如果出现 `mmc_card_disable_cd` 未定义 error, 请打开 `if_sdio.c` 中 `mmc_card_disable_cd` 定义代码, 如下图所示:

```
5:
6: #define FUNC_DEV(f) (&((f)->dev))
7:
8: #define SDIO_CAP_IRQ(func) ((func)->card->host->caps & MMC_CAP_SDIO_IRQ)
9: #define SDIO_CAP_POLL(func) ((func)->card->host->caps & MMC_CAP_NEEDS_POLL)
0: #define HOST_SPI_CRC(func, crc) (func)->card->host->use_spi_crc=crc
1:
2: // #define mmc_card_disable_cd(c) (1)
3: #define hgic_card_disable_cd(func) mmc_card_disable_cd((func)->card)
4: #define hgic_card_set_highspeed(func) mmc_card_set_highspeed((func)->card)
5: #define hgic_host_is_spi(func) mmc_host_is_spi((func)->card->host)
6: #define hgic_card_cccr_widebus(func) ((func)->card->cccr.low_speed && !(func)->card->cccr.high_speed)
7: #define hgic_card_cccr_highspeed(func) ((func)->card->cccr.high_speed)
8: #define hgic_host_highspeed(func) ((func)->card->host->caps & MMC_CAP_SD_HIGHSPEED)
9: #define hgic_host_supp_4bit(func) ((func)->card->host->caps & MMC_CAP_4_BIT_DATA)
0: #define hgic_card_highspeed(func) mmc_card_highspeed((func)->card)
1: #define hgic_func_rca(func) ((func)->card->rca)
2: #define hgic_card_max_clock(func) ((func)->card->cis.max_dtr)
```

## 3 WiFi Driver 开发使用说明

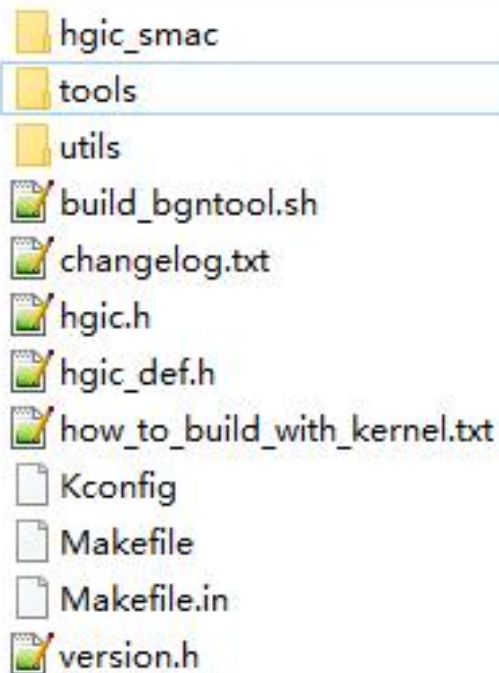
### 3.1 hgic\_smac

SMAC 驱动需要内核支持 MAC80211/CFG80211。AP 模式需要使用 hostapd，STA 模式需要使用 wpa\_supplicant。hostapd/wpa\_supplicant 均可自行从 Internet 下载编译。

hgic\_smac 驱动适用于 TXW801 系列芯片，不支持低功耗模式。可编译成独立 ko 文件，也可以编译进内核。

#### 3.1.1 hgic\_smac 驱动文件

hgic\_smac 驱动以源码形式发布，用户自行编译：



其中 tools 目录提供了常用工具：hostapd-2.9/wpa\_supplicant-2.9/iw-5.0。smac 驱动对 hostapd/wpa\_supplicant 的版本无要求，可以使用其它版本。

### 3.1.2 hgic\_smac 编译说明

编译 smac 驱动需要修改 Makefile 文件添加编译工具链信息。

Makefile 文件已经添加部分主控编译参数，请自行添加方案所使用的主控的编译参数。

```
#Hi3518
#ARCH := arm
#COMPILER := arm-hisiv300-linux-uclibcgnueabi-
#LINUX_KERNEL_PATH := /home/matt/Hi3518/hi3518/linux-3.4.y

#FH8852
#ARCH := arm
#COMPILER := arm-fullhan-linux-uclibcgnueabi-
#LINUX_KERNEL_PATH := $(CURRENT_PATH)/../linux-3.0.8
#CFLAGS += -DFH8852 -DCONFIG_NO_ZERO_PACKET
```

编译 smac 驱动时可以选择支持 sdio 或者 usb 接口。执行 make 命令可以查看编译说明。

```
usage:
make smac      : compile SMAC driver. support sdio/usb interface. generate hgics.ko
make smac_usb  : compile SMAC driver. only support usb interface. generate hgics.ko
make smac_sdio: compile SMAC driver. only support sdio interface. generate hgics.ko
make fmac      : compile FMAC driver. support sdio/usb interface. generate hgicf.ko
make fmac_usb  : compile FMAC driver. only support usb interface. generate hgicf.ko
make fmac_sdio: compile FMAC driver. only support sdio interface. generate hgicf.ko
make clean
```

### 3.1.3 hgic\_smac 加载说明

加载 smac 驱动: insmod hgics.ko。

加载驱动时，可以根据需要指定以下参数：

- **fw\_file**: 指定 WiFi 固件名称。未指定该参数时驱动默认加载 hgics.bin。
  - insmod hgics.ko **fw\_file**=xxxx.bin
 具体说明参见 [3.2 固件下载](#) 章节。

### 3.1.4 hgic\_smac proc fs 接口

除了可以常规使用 hostapd/wpa\_supplicant 程序之外，hgic\_smac 驱动也提供 proc fs 接口。应用程序通过 proc fs 接口可以与驱动/固件进行交互。

proc fs 接口文件存放在 /proc/hgics/ 目录下。

1. **/proc/hgics/fwevent**（只读）  
该文件用于接收驱动消息。在驱动无消息时访问该接口会被阻塞 100ms。  
tools/test\_app/hgics.c 提供了示例代码。
2. **/proc/hgics/status**（只读）  
cat /proc/hgics/status 可以查看驱动运行状态。包括固件信息，驱动数据缓存信息。
3. **/proc/hgics/iwpriv**（读写）  
该文件是泰芯提供的 hgpriv.c/iwpriv.c 工具交互接口，用于驱动/固件参数设置。  
具体参见 3.1.5 章节。

### 3.1.5 hgic\_smac 辅助模块

除了可以正常使用 hostapd/wpa\_supplicant 应用程序外，smac 驱动提供了 /proc/hgics/iwpriv 接口，通过该接口程序可以与 WiFi 驱动/固件进行交互。

为了方便使用 /proc/hgics/iwpriv 接口，smac 驱动在 tools/test\_app 目录下提供了辅助 API 文件：iwpriv.c/hgics.c/hgpriv.c。

应用程序可以集成这些代码，直接通过 API 方式与 WiFi 驱动/固件进行交互。

其中 hgics.c 可以作为 smac 驱动的服务程序，接收处理驱动的 event 数据，与驱动/固件进行交互。

说明：tools/test\_app/iwpriv.c 封装了所有的驱动接口，但是 smac 固件只支持其中部分 API。

### 3.1.6 hgic\_smac 调试与测试

#### 1. 接口测试

hgic\_smac 提供了 sdio/usb 接口测试模式，可对 sdio/usb 接口进行稳定性测试。在



insmod hgics.ko 时指定 if\_test 参数可以启动接口测试。

- insmod hgics.ko if\_test=1 #启动接口单向测试,测试数据从 host 端发送给 WiFi 芯片
- insmod hgics.ko if\_test=2 #启动接口双向测试,测试数据双向传输。

## 2. 调试信息

WiFi 驱动可以导出 WiFi 芯片调试信息输出到主控端打印。

在不方便使用 WiFi 芯片串口时可以使用此方法查看 WiFi 芯片调试信息。

- hgpriv wlan0 set dbginfo=1 //打开 WiFi 调试信息输出
- hgpriv wlan0 set dbginfo=0 //关闭 WiFi 调试信息输出

### 3.1.7 hgic\_smac BLE 配网

请参考文档《泰芯 TXW80x BLE 配网 Linux 开发指南》。

## 3.2 固件下载

WiFi 模块支持通过 SDIO/USB 接口下载固件运行。固件下载功能与 Linux 系统有关,不同版本的 Linux kernel,使用方式可能会有差异。WiFi 固件要放在系统支持的固件加载目录下,通常是/lib/firmware。如果固件加载遇到问题,请检查以下几个方面:

### 1. kernel 编译配置

kernel 需要支持 CONFIG\_FW\_LOADER (Devices Drivers->Generic Driver Options)

```
Generic Driver Options
>. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> module < > module capable

(/sbin/hotplug) path to uevent helper
[ ] Maintain a devtmpfs filesystem to mount at /dev
[*] Select only drivers that don't need compile-time external firmware
[*] Prevent firmware from being built
<*> Userspace firmware loading support
[ ] Include in-kernel firmware blobs in kernel binary
[ ] External firmware blobs to build into the kernel binary
[*] Fallback user-helper invocation for firmware loading
[ ] Driver Core verbose debug messages
[ ] Managed device resources verbose debug messages
```

修改该选项后，WiFi 驱动需要重新编译。

## 2. 查看 kernel 支持的固件目录

查看 firmware\_class.c 文件中的 fw\_path 数组。

```
9:
9: /* direct firmware loading support */
1: static char fw_path_para[256];
2: static const char * const fw_path[] = {
3:     fw_path_para,
4:     "/lib/firmware/updates/" UTS_RELEASE,
5:     "/lib/firmware/updates",
6:     "/lib/firmware/" UTS_RELEASE,
7:     "/lib/firmware"
8: };
9:
```

注：

1. 部分旧版本 kernel 的 firmware\_class.c 可能没有 fw\_path 数组。
2. Android 系统默认的固件目录是 /vendor/firmware 或 /etc/firmware 或 /firmware/image

## 3. busybox 支持 mdev

查看是否存在 /sbin/mdev 文件。

## 4. 查看 /proc/sys/kernel/hotplug 事件处理程序

cat /proc/sys/kernel/hotplug 查看是否指定事件处理程序，如果未指定则需要在系统初始化时执行：echo /sbin/mdev > /proc/sys/kernel/hotplug

## 5. USB 接口发送空包

部分主控的 USB Host 不支持发送空包，会造成固件下载失败。如果遇到如下类似现象，请确认 USB Host 是否支持发送空包，或者尝试在 Makefile 添加 **-DCONFIG\_USB\_ZERO\_PACKET**

```

07] leave hgic_bootd1_download
96] enter hgic_bootd1_download
75] hgic_get_fw_d1_addr:150::hgic_get_fw_d1_addr:check Para:download addr:20000000
18] hgic_get_fw_aes_en:114::hgic_get_fw_aes_en:check Para:aes_en:0
45] hgic_get_fw_crc_en:132::hgic_get_fw_crc_en:check Para:crc_en:1
72] hgic_get_fw_local_crc32:204::hgic_get_fw_local_crc32:check Para:local crc:0
02] hgic_get_fw_code_offset:186::hgic_get_fw_code_offset:check Para:code offset:3072
30] hgic_bootd1_parse_fw:184::firmware hdr len : 3072
51] hgic_bootd1_parse_fw:185::firmware run addr : 20000000 内核已读取固件数据
74] hgic_bootd1_parse_fw:186::firmware size : 134160
96] hgic_bootd1_parse_fw:187::firmware aes_en:0,crc_en:1
29] hgic_bootd1_send_fw:215::send fw data error, no resp!
63] hgic_bootd1_download:412::hgic_bootd1_send_fw error!
88] hgic_bootd1_download:427::Release boot download firmware... 固件下载失败
92] leave hgic_bootd1_download
25] hgic_bootd1_send_cmd_tmo:252::cmd: 0, no response!!

```

```

#FH8852
#ARCH := arm
#COMPILER := arm-fullhan-linux-uclibcgnueabi-
#LINUX_KERNEL_PATH := $(CURRENT_PATH)/../linux-3.0.8
#CFLAGS += -DFH8852 -DCONFIG_USB_ZERO_PACKET

```

## 6. fw\_file 参数不能包含路径