



V316 WiFi 模块使用说明书

文档版本号: SDK-V1.0

发布日期: 2019-03-30

版权所有©珠海全志科技股份有限公司 2019。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



、全志和其他全志商标均为珠海全志科技股份有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受全志公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，全志公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。





文档履历

版本号	日期	制/修订人	内容描述
V1.0	2019-03-30	Allwinner	V316 初始化版本

1	前言	1
1.1	编写目的	1
1.2	相关人员	1
1.3	相关术语	1
2	WiFi 模块框架概述	2
2.1	代码分布	2
2.2	WiFi 框架介绍	2
3	WiFi 模块移植配置说明	3
3.1	已支持 WiFi 模组	3
3.2	sys_config.fex 配置	3
3.3	firmware 移植	4
3.4	驱动移植	4
4	WiFi 模块使用说明	5
4.1	Station 相关使用说明	5
4.2	Softap 相关使用说明	7
5	Declaration	9



表 目 录





图 目 录



1 前言

1.1 编写目的

介绍 WiFi 模组移植配置方法，介绍 SDK 中 WiFi 的 Station 和 Softap 使用说明，目的是让 WiFi 模块的开发和使用人员可以根据该文档完成一些与 WiFi 相关的常规工作，解决常见问题。

1.2 相关人员

WiFi 模块开发和使用人员。

1.3 相关术语

- **Station:** WiFi 的一种工作模式，处于 Station 模式的设备可以去连接无线路由器，并通过 WiFi 上网；
- **Softap:** WiFi 的一种工作模式，处于 Softap 模式的设备作为无线热点，供其他设备连接，实现网络共享；

2

WiFi 模块框架概述

2.1 代码分布

层次	路径
APP	custom_aw/apps/XXX/source/device_model/system/net
framework	system/public/wifi
driver	linux-4.9/drivers/net/wireless

2.2 WiFi 框架介绍

1. APP 层作为与用户交互的窗口，通过调用 Framework 层提供的 API 实现对 WiFi 模块的控制与管理；
2. Framework 层起到承上启下的作用，向上为 APP 层提供 API 接口，向下通过调用 wpa_supplicant 提供的接口实现与 wpa_supplicant 或 hostapd 守护进程的通信；wpa_supplicant 实现了 station 对无线网络进行管理和控制的功能，hostapd 能够使无线网卡切换为 master 模式，模拟 AP（路由器）的功能，负责控制管理 Station 的接入和认证。wpa_supplicant 和 hostapd 都是通过 netlink 机制与驱动进行通信。
3. driver 层实现了 Upper MAC 协议，负责与 WiFi 硬件进行通信。

3 WiFi 模块移植配置说明

3.1 已支持 WiFi 模组

目前 V316 SDK 已支持 XR819 WIFI 模组，以下移植过程都以 XR819 为例进行说明；

3.2 sys_config.fex 配置

sys_config 需参考硬件原理图进行配置，例如，pro 板的配置如下；

```
[wlan]
wlan_used      = 1
wlan_busnum    = 1
wlan_usbnum    = 3
wlan_power     = "vcc33-wifi"
wlan_io_regulator = "vcc18-wifi"
;wlan_en       = port:PG11<1><default><default><0>
wlan_regon     = port:PL12<1><default><default><0>
wlan_hostwake  = port:PL05<6><default><default><0>
```

说明：

1. ”;” 符号起注释作用，如果不需要则将其注释；
2. ”wlan_busnum” 表示 WiFi 所使用的 SDIO 控制器号；
3. ”wlan_usbnum” 表示 WiFi 所使用的 USB 控制器号；
4. ”wlan_power” 表示给 WiFi 模组的 VBAT 供电的 regulator 名称；
5. ”wlan_io_regulator” 表示给 WiFi 模组的 GPIO 供电的 regulator 名称；
6. ”wlan_en” 表示 wlan_power 给 WiFi 模组的 VBAT 供电，wlan_en 控制供电的通断；
7. ”wlan_regon” 表示 WiFi 的功能使用 GPIO；
8. ”wlan_hostwake” 表示 WiFi 唤醒主控的 GPIO；
9. 以上所有项必须参看原理图进行配置，配置与原理图实际使用的资源保持一致；

3.3 firmware 移植

1. 在 system/public/wifi/firmware 目录下建立对应的文件夹如“xr819”，并将 firmware 文件拷贝到该文件夹下；
2. 在 system/public/wifi/build.mk 中添加拷贝 xr819 firmware 的操作：

```
ifeq (${WIFI_CONFIG},xr819)
$(warning "The wifi is xr819")
$(shell rm -rf $(TARGET_OUT)/target/etc/firmware)
$(call copy-files-under, \
  $(TARGET_PATH)/firmware/xr819/*, \
  $(TARGET_OUT)/target/etc/firmware \
)
$(shell mkdir -p $(TARGET_OUT)/target/etc/firmware/xr819)
```

3.4 驱动移植

1. 将 xr819 驱动源码放到 SDK 的 lichee/linux/driver/net/wireless 目录下面，并命名为“xr819”；
2. 修改 lichee/linux-4.9/drivers/net/wireless/Kconfig，添加：

```
source "drivers/net/wireless/xr819/Kconfig"
```

3. 修改 lichee/linux-4.9/drivers/net/wireless/Makefile，添加：

```
obj-$(CONFIG_XR819_WLAN) += xr819/
```

4. 执行 make menuconfig 选中 xr819，随内核编译成.ko。

4 WiFi 模块使用说明

4.1 Station 相关使用说明

WiFi Station 功能的主要 API 函数介绍如下：

`int wifi_sta_init(void)` 作

用：初始化操作，根据模组名加载对应驱动；参

数：无返回值：：操作成功；：操作失败；

0-1

`int wifi_sta_exit(void)` 作

用：卸载驱动；WIFI 参

数：无返回值：：操作成功；：操作失败；

0-1

`int wifi_sta_open(const char *wifi_name)` 作

用：启动进程；wpa_supplicant 参

数：网口名，通常为 wlan0 返回值：：操作成功；：操作失败；

0-1

`int wifi_sta_close(const char *wifi_name)` 作

用：退出进程；wpa_supplicant 参

数：网口名，通常为 wlan0 返回值：：操作成功；：操作失败；

0-1

`int wifi_sta_start(const char *wifi_name)` 作

用：创建监听线程，监听事件；wpa_supplicant 参

数：网口名，通常为 wlan0 返回值：：操作成功；：操作失败；

0-1

`int wifi_sta_stop(const char *wifi_name)` 作

用：销毁监听线程；参

数：网口名，通常为 wlan0 返回值：：操作成功；：操作失败；

0-1

`int wifi_sta_start_scan(const char *wifi_name)`作

用：触发进行扫描；wpa_supplicant参

数：网口名，通常为wlan0返回值：：操作成功；负值：操作失败；

0

`int wifi_sta_get_scan_status(const char *wifi_name, WIFI_STA_STATUS_E *pscan_status)`作

用：获取当前扫描状态；参

数：网口名，扫描状态返回值：：操作成功；：操作失败；

0-1

`int wifi_sta_get_scan_results(const char *wifi_name, WIFI_STA_AP_LIST_S *pap_list)`作

用：获取扫描结果；参

数：网口名，列表AP返回值：：操作成功；：操作失败；

0-1

`int wifi_sta_connect(const char *wifi_name, WIFI_STA_AP_INFO_S *pap_info)`作

用：连接指定；AP参

数：网口名，指定信息AP返回值：：操作成功；：操作失败；

0-1

`int wifi_sta_disconnect(const char *wifi_name)`作

用：断开连接；参

数：网口名，通常为wlan0返回值：：操作成功；：操作失败；

0-1

`int wifi_sta_get_connect_status(const char *wifi_name, WIFI_STA_CONNECT_STATUS_S *pconnect_status)`作

用：获取连接状态；参

数：网口名，连接状态返回值：：操作成功；：操作失败

0-1

`int wifi_sta_do_dhcp(const char *wifi_name)`作

用：启动获取地址udhcpIP;参

数：网口名返回值：：操作成功；：操作失败

0-1

4.2 Softap 相关使用说明

WiFi Softap 功能的主要 API 函数介绍如下：

`int wifi_ap_init(void)` 作

用：初始化操作，根据模组名加载对应驱动；参
数：无返回值：：操作成功；：操作失败；

0-1

`int wifi_ap_exit(void)` 作

用：卸载驱动；WIFI 参
数：无返回值：：操作成功；：操作失败；

0-1

`int wifi_ap_open(const char *wifi_name)` 作

用：更新状态为；OPENED 参
数：网口名返回值：：操作成功；：操作失败；

0-1

`int wifi_ap_close(const char *wifi_name)` 作

用：更新状态为；CLOSED 参
数：网口名返回值：：操作成功；：操作失败；

0-1

`int wifi_ap_start(const char *wifi_name, WIFI_AP_CFG_S *ap_cfg)` 作

用：将配置写入 hostapd，启动进程；confhostapd 参
数：网口名，配置信息返回值：：操作成功；：操作失败；

0-1

`int wifi_ap_stop(const char *wifi_name)` 作

用：退出进程，更新状态为；hostapd STOPPED 参
数：网口名返回值：：操作成功；：操作失败；

0-1

`int wifi_ap_get_config(const char *wifi_name, WIFI_AP_CFG_S *ap_cfg)` 作

用：获取配置信息；AP 参
数：网口名，配置信息返回值：：操作成功；：操作失败；



5

Declaration

This document is the original work and copyrighted property of Allwinner Technology (‘ ‘Allwinner’ ’). Reproduction in whole or in part must obtain the written approval of Allwinner and give clear acknowledgement to the copyright owner.

The information furnished by Allwinner is believed to be accurate and reliable. Allwinner reserves the right to make changes in circuit design and/or specifications at any time without notice. Allwinner 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 Allwinner. This datasheet neither states nor implies warranty of any kind, including fitness for any particular application. tates nor implies warranty of any kind, including fitness for any particular application.

