

R6/R16/R18/ R40/R58

Tina softap API v1.0

文档履历

版本号	日期	制/修订人	制/修订记录
V1.0	2017-11-16	Kevin	建立初始版本

Yllwinberte

目 录

1. 概述.....	4
1.1. 编写目的.....	4
1.2. 使用范围.....	4
1.3. 目标读者.....	4
2. softap 相关说明.....	5
2.1. sdk 代码目录.....	5
2.2. 编译配置.....	5
2.2.1. 内核配置.....	5
2.2.2. Tina 配置.....	5
2.3. APP 编写说明.....	6
2.3.1. 导入接口文件.....	6
2.3.2. 链接动态库.....	6
2.3.3. 示例代码.....	6
2.4. wifi 打开和关闭.....	6
2.4.1. wifi 打开.....	6
2.4.2. wifi 服务关闭.....	6
3. API 说明.....	7
3.1. softAP 初始化和配置.....	7
3.1.1. Wifi firmware 切换.....	7
3.1.2. softAP 初始化.....	7
3.1.3. softAP 反初始化.....	7
3.1.4. 配置 softAP.....	8
3.1.5. 保存配置.....	8
3.2. 建立 softAP 热点.....	8
3.2.1. 启动 softap.....	8
3.2.2. 设置 IP 和子网掩码.....	8
3.2.3. 启动 udhcpd 和 dns 服务.....	9
3.2.4. 使能数据转发.....	9
3.3. 关闭 softAP.....	9
3.3.1. 关闭 softap.....	9
3.4. 获取 softAP 状态.....	9
3.4.1. 获取 softAP 状态.....	9
4. 使用说明.....	10
4.1. 关于 station 和 softAP 共存模式的说明.....	10
4.2. Tina Softap 中 firmware 参数设置.....	10
5. Demo.....	12
5.1. softAP 启动.....	12
5.2. softAP 长时耐久性测试.....	14
6. Declaration.....	18

1. 概述

1.1. 编写目的

介绍 Tina wifi softap 管理开发接口和 Demo 代码。

1.2. 使用范围

硬件平台：AW R6、R16、R18、R58 及 R40

系统版本：Tina v2.0 及以上版本

1.3. 目标读者

Tina app 开发人员。

Allwinner

2. softap 相关说明

softap 部分代码为 Tina 平台管理 wifi softap 模式的模块。主要功能包括打开/配置/启动/关闭 softap，获取 softap 的状态等。

2.1. sdk 代码目录

sdk 中 softap 相关代码目录为 package\allwinner\softap。包括源码和 demo 程序。

2.2. 编译配置

2.2.1. 内核配置

Tina 根目录下，输入：

```
make kernel_menuconfig
```

选择如下 softap 所需要的内核组件：

```
[*] Networking support --->
  Networking options --->
    [*] Network packet filtering framework (Netfilter) --->
      [*] Advanced netfilter configuration
      Core Netfilter Configuration --->
        <*> Netfilter connection tracking support
          [*] Connection mark tracking support
          [*] Connection tracking security mark support
          [*] Connection tracking events
          [*] Connection tracking timeout
          [*] Connection tracking timestamping
          <M> Connection tracking netlink interface
          <M> Connection tracking timeout tuning via Netlink
          -* Netfilter Xtables support (required for ip_tables)
            <*> "conntrack" connection tracking match support
            <*> "state" match support
          IP: Netfilter Configuration --->
            <*> IPv4 connection tracking support (required for NAT)
            <*> IP tables support (required for filtering/masq/NAT)
            <*> Packet filtering
            <*> IPv4 NAT
            <*> MASQUERADE target support
            <*> NETMAP target support
            <*> REDIRECT target support
            <*> Packet mangling
```

2.2.2. Tina 配置

Softap 需要用到的应用包括：hostapd、iptables、dnsmasq，在编译之前需要配置并选中这些功能。在 Tina 根目录下，输入：

```
make menuconfig
```

配置选中 hostapd

```
Network --->
```

```
<*> hostapd..... IEEE 802.1x Authenticator (full)
```

```
  -* hostapd-common..... hostapd/wpa_supplicant common support files
```

配置选中 iptables：

```
Network --->
```

```
  Firewall --->
```

```
    <*> iptables..... IP firewall administration tool --->
```

配置选中 dnsmasq：

```
Base system --->
```

```
<*> dnsmasq..... DNS and DHCP server
```

```
softap sdk 相关 menuconfig 配置如下:
```

```
Allwinner --->
```

```
  <*> softap..... Tina softap manager  --->
```

```
    <*> softap-demo..... Tina softap app demo
```

注意:如果要参考 softap app demo 代码, 需要先选择

```
<*> softap..... Tina softap manager  --->
```

再选择 softap-demo 包, 表示 softap app demo 程序。

2.3. APP 编写说明

2.3.1. 导入接口文件

```
#include <aw_softap_intf.h>
```

2.3.2. 链接动态库

```
libsoftap.so
```

2.3.3. 示例代码

softap app demo 代码目录为:

```
package/allwinner/softap/demo。
```

2.4. wifi 打开和关闭

2.4.1. wifi 打开

wifi 打开主要完成如下工作:

1. 启动 wpa_supplicant 服务(如果没有启动);
2. 连接 wpa_supplicant (wifi driver 由系统启动时完成加载, wpa_supplicant 服务可以在系统启动过程中启动)

2.4.2. wifi 服务关闭

wifi 关闭主要完成如下工作:

1. 断开与 wpa_supplicant 的连接
2. kill 掉 wpa_supplicant 服务
3. disable wlan0 网口, wifi 不再可用。

3. API 说明

Tina 平台 softap 包括初始化 softAP，配置 softAP，打开/关闭 softAP，获取 softAP 的状态信息等。

3.1. softAP 初始化和配置

3.1.1. Wifi firmware 切换

函数原型	int aw_softap_reload_firmware(char *ap_sta);
参数说明	ap_sta: 需要切换的 firmware 对应的 WIFI 模式。可以输入的参数有“STA”、“AP”、“P2P”，代表目前支持 WIFI 的三种模式： a、“STA”——station 模式； b、“AP”——softAP 模式； c、“P2P”——P2P 模式； 启动 softap 需指定参数为“AP”；退出 softap 可指定其他模式参数，或者保持不变。
返回说明	0: 切换指定 firmware 成功。 非 0: 切换失败。
功能描述	切换 WIFI 模式对应的 firmware。。
注意事项	①需要注意的是：切换 firmware 并不会立即生效让 WIFI 处于对应的模式，而是在调用 aw_softap_enable()时生效。 ②目前 Tina SDK 支持的 WIFI 模组中，broadcom AP6212/AP6212A 等系列模组需要切换 firmware；而其他厂家（例如 realtek）的模组则无需/无法切换 firmware。具体情况请咨询使用的模组原厂技术支持人员或代理商。 ③firmware 的参数设置详见 4.2 部分。

3.1.2. softAP 初始化

函数原型	int aw_softap_init();
参数说明	无
返回说明	0: 初始化成功。 非 0: 初始化失败。
功能描述	初始化 softap 内部数据结构和默认基本配置。 默认的基本配置如下： a、使用 wlan0 接口启动 softap； b、ssid 为 Smart-AW-HOSTAPD； c、psk 为 wifl1111; (详见注意事项①) d、AP 可见 (broadcast)； e、使用通道 6; (详见注意事项②) f、加密方式为 wpa2-psk。
注意事项	①softap 支持的加密方式只有三种：SOFTAP_NONE、SOFTAP_WPA_PSK、SOFTAP_WPA2_PSK。 ②配置的通道不一定生效。例如，对于 broadcom 的模组，如果是同一模组同时开启了 station 和 softap 模式，则 softap 使用的通道随 station 变动而变动。 ③关于以上两点更多说明见 aw_softap_config() API 的介绍。

3.1.3. softAP 反初始化

函数原型	int aw_softap_deinit();
参数说明	无
返回说明	0: 反初始化成功。 非 0: 反初始化失败。
功能描述	释放 softap 内部数据结构。
注意事项	退出 softap 相关进程时，必须要调用此接口释放内部数据结构。根据使用场景的不同，分为两种场景： ①启动 softAP 以后不关闭 softAP。退出进程前调用此接口释放 softap 内部数据结构，并不

	会影响已经保存或生效的 softAP 相关配置。 ②启动 softAP 以后关闭 softAP。执行完 softap 关闭的所有动作后，退出进程前调用此接口释放 softap 内部数据结构。
--	--

3.1.4. 配置 softAP

函数原型	int aw_softap_config(char *ssid, char *psk, tSOFTAP_KEY_MGMT key_mgmt, char *interface, char *channel, char *broadcast_hidden);
参数说明	ssid: 设置 softAP 的 ssid。 psk: 设置 softAP 的 psk。 (详见注意事项①) key_mgmt: 设置 softAP 的加密方式。softap 支持的加密方式只有三种: SOFTAP_NONE (无密)、SOFTAP_WPA_PSK、SOFTAP_WPA2_PSK。 (详见注意事项①) interface: 设置 softAP 使用的网络接口。 channel: 设置 softAP 使用的通信信道。 (详见注意事项②) broadcast_hidden: 设置 softAP 是否隐藏。不隐藏: 参数为字符串“broadcast”; 隐藏: 参数为字符串“hidden”。
返回说明	0:成功; 非 0:失败;
功能描述	配置 softAP 的各项参数。
注意事项	①如果设置加密方式为 SOFTAP_NONE, 即使 psk 带入的参数不为空字符串, 则仍认为该 AP 设置为开放 (无密) 的 AP。 ②配置的通道不一定生效。例如, 对于 broadcom 的模组, 如果是同一模组同时开启了 station 和 softap 模式, 则 softap 使用的通道随 station 变动而变动。

3.1.5. 保存配置

函数原型	int aw_softap_save_config();
参数说明	无
返回说明	0:成功; 非 0:失败;
功能描述	保存 softAP 的配置到配置文件。
注意事项	设备 softAP 的相关配置保存在/etc/wifi/hostapd.conf, 禁止直接操作该文件。

3.2. 建立 softAP 热点

3.2.1. 启动 softap

函数原型	int aw_softap_enable();
参数说明	无
返回说明	0:成功; 非 0:失败。
功能描述	应用保存到配置文件的配置, 启动 softAP。
注意事项	如果是使用 broadcom 的 WIFI 模组, 设置 interface 为 wlan1, 开启 station 和 softAP 兼容模式 (wlan0 作为 staion, wlan1 作为 softAP), 需要在 aw_softap_intf.h 中将#define IW_UP_BROADCOM_WLAN1 0 修改为#define IW_UP_BROADCOM_WLAN1 1。

3.2.2. 设置 IP 和子网掩码

函数原型	int aw_softap_router_config(char *ip, char *netmask)
参数说明	ip: IP 地址。 netmask: 子网掩码。
返回说明	0:成功; 非 0:失败。
功能描述	设置建立的 softAP 的 IP 和子网掩码。

3.2.3. 启动 udhcpd 和 dns 服务

函数原型	int aw_softap_start_udhcp_dns_server();
参数说明	无
返回说明	0:成功; 非 0:失败。
功能描述	启动 udhcpd 和 dns 中转、缓存服务。
注意事项	只有启动了 udhcpd 和 dns 服务, 其他设备才能正常连接至该 softAP 并自动获取 IP 地址。

3.2.4. 使能数据转发

函数原型	int aw_softap_enable_data_forward(char *interface);
参数说明	interface: 需要转发数据的网络接口。 (详见注意事项①)
返回说明	0:成功; 非 0:失败。
功能描述	使能数据转发。如果设备带有以太网卡等网络接口, 想实现外部网络接入和数据互通, 可以使用此调用开启数据转发。
注意事项	①常见的网络接口如以太网 eth0。

3.3. 关闭 softAP

3.3.1. 关闭 softap

函数原型	int aw_softap_disable();
参数说明	无
返回说明	0:成功; 非 0:失败。
功能描述	关闭 softAP。
注意事项	如果是使用 broadcom 的 WIFI 模组, 设置 interface 为 wlan1, 开启 station 和 softAP 兼容模式 (wlan0 作为 staion, wlan1 作为 softAP), 需要在 aw_softap_intf.h 中将#define IW_UP_BROADCOM_WLAN1 0 修改为#define IW_UP_BROADCOM_WLAN1 1。

3.4. 获取 softAP 状态

3.4.1. 获取 softAP 状态

函数原型	int aw_is_softap_started();
参数说明	无
返回说明	0:未建立; 1:已建立。
功能描述	用于获取 softAP 状态, 从返回值获知 softAP 是否建立。
注意事项	成功调用 aw_softap_enable()后, 调用此接口即返回 1。

4. 使用说明

4.1. 关于 station 和 softAP 共存模式的说明

如果使用的模组（如 realtek 某些型号的模组）驱动能同时生成两个虚拟接口，两个接口相互独立，可以使用一个接口作为 station(如 wlan0)，使用另一个接口作为 softAP。具体详情请咨询模组原厂或代理商相关技术人员。

如使用的为 broadcom 支持 station 和 AP 共存的模组，需要在 wlan0 启动的基础上添加 wlan1 接口。原则上，常见的所有 broadcom 模组均支持 station 和 AP 模式共存，但需要使用 broadcom 原生工具 dhd_priv 添加新的 interface 并建立 softAP。但目前只有部分模组支持使用 hostapd 在 wlan1 建立 softAP。这些 broadcom 模组有 AP6255、AP6356S 等，且驱动需升级为 1.363.59.144.10 以上版本。wlan1 的启动必须先 ifconfig wlan0 up (Tina softap 内部已经进行相应处理)。Tina softap 目前仅支持基于 hostapd 进行 broadcom 模组 station 和 softAP 共存，使用时需要将 package/allwinner/softap/src/include/aw_softap_intf.h 中：

```
#define IW_UP_BROADCOM_WLAN1 0
```

改为

```
#define IW_UP_BROADCOM_WLAN1 1
```

如使用的模组不支持基于 hostapd 在 wlan1 上建立 softAP，可以自行使用命令行添加 wlan1，建立热点（由于未使用 hostapd 标准组件，因此未将此方式兼容进 Tina softap）：

1、启动 wlan0:

```
ifconfig wlan0 up
```

2、添加 wlan1:

```
dhd_priv iapsta_init mode apsta ifname wlan1
```

3、设置 softAP 参数:

```
dhd_priv iapsta_config ifname wlan1 ssid tttv chan 6 amode [open/wpa2psk] emode [none/aes] key 12345678
```

设置 softAP 参数项为:

ssid: 示例为 tttv

channel: softAP 通道，示例为 6

amode: 加密方式，open(无密)/wpa2psk

emode: 加密算法，none/aes

key: 密钥，示例为 12345678

4、启动 softAP:

```
dhd_priv iapsta_enable ifname wlan1
```

5、使用 wlan0 连接其他 AP

6、关闭 softAP:

```
dhd_priv iapsta_disable ifname wlan1
```

4.2. Tina Softap 中 firmware 参数设置

如果使用的模组不需要加载 firmware(如 realtek 的常见模组)，则无需修改。

如果使用的模组在启动时需要加载 firmware，可能需要修改 firmware 的相关参数。目前 Tina softap 默认支持的模组为 AP6212，如果使用的为其他模组，需要修改 package/allwinner/softap/src/include/wifi.h 中如下宏：

```
/*path of firmware for WIFI in different mode*/
```

```
#ifndef WIFI_DRIVER_FW_PATH_STA
```

```
#define WIFI_DRIVER_FW_PATH_STA "/lib/firmware/fw_bcm43455c0_ag.bin"
```

```
#endif
```

```
#ifndef WIFI_DRIVER_FW_PATH_AP
```

```
#define WIFI_DRIVER_FW_PATH_AP "/lib/firmware/fw_bcm43455c0_ag_apsta.bin"
```

```
#endif
```

```
#ifndef WIFI_DRIVER_FW_PATH_P2P
```

```
#define WIFI_DRIVER_FW_PATH_P2P "/lib/firmware/fw_bcm43438a0_p2p.bin"
```

```
#endif
```

```
#ifndef WIFI_DRIVER_FW_PATH_PARAM
```

```
#define WIFI_DRIVER_FW_PATH_PARAM "/sys/module/bcmdhd/parameters/firmware_path"  
#endif
```

其中，WIFI_DRIVER_FW_PATH_STA、WIFI_DRIVER_FW_PATH_AP、WIFI_DRIVER_FW_PATH_P2P 分别为 WIFI 模组 station 模式、softAP 模式、P2P 模式使用的模组在设备上的放置路径；WIFI_DRIVER_FW_PATH_PARAM 为 firmware 参数设置节点，由内核 WIFI 驱动指定。

Allwinner

5. Demo

Demo 部分，softAP 启动部分示例了 softAP 启动的主要流程；softAP 长时耐久性测试，作为内部测试用例，一般不为用户所使用。但它作为一个完整的 softAP 启动、关闭的流程，可以作为读者编程的重要参考。

5.1. softAP 启动

```
#include <stdlib.h>
#include <string.h>
#include <unistd.h>

#include <aw_softap_intf.h>

/*FOR COMPLETE PROCESS for up and dow softap, please refer to the demo:
"softap_up_down_longtime_test".
*/
int main(int argc, char **argv)
{
    int i = 0;
    int len = 0;
    tSOFTAP_KEY_MGMT key_mgmt = SOFTAP_WPA2_PSK;
    int ret = 0;
    char *psk = NULL;
    char *broadcast_hidden = NULL;

    /*No arguments or only the one who is for help*/
    if((argc == 1) || (argc == 2 && (!strcmp(argv[1], "-help") || !strcmp(argv[1], "-h"))))
    {
        printf("*****\n");
        printf("softap_up -help/-h: for usage details\n");
        printf("Usage:\n");
        printf("softap_up \"ssid\" \"psk/open\" \"broadcast/hidden\"\n");
        printf("*****\n");
        return 0;
    }

    ret = aw_softap_init();
    if(0 != ret)
    {
        printf("init softap failed!\n");
        return -1;
    }

    printf("*****\n");
    printf("Start hostapd test!\n");
    printf("*****\n");

    ret = aw_softap_reload_firmware("AP");
    if(0 != ret)
    {
        printf("reload firmware failed!\n");
        return -1;
    }

    /*To keep consistent with the older version demo,
    this demo do no set <interface> <channel>, but
    application of the user's can set them correctly by
    themselves*/
```

```

if((argc >= 3) && (argv[2][0] != '\0') && strcmp(argv[2], "open"))
{
    psk = (char *)argv[2];
    key_mgmt = SOFTAP_WPA2_PSK;
    printf("wpa2-psk!\n");
}
else
{
    key_mgmt = SOFTAP_NONE;
    printf("none key_mgmt!\n");
}

if(argc >= 4)
    broadcast_hidden = (char *)argv[3];
else
    broadcast_hidden = (char *)"broadcast"; //default set

//此 demo 默认使用 wlan0 作为 softAP interface, 使用通道 6
ret = aw_softap_config(argv[1], psk, key_mgmt, "wlan0", "6", broadcast_hidden);

if(0 != ret)
{
    printf("config softap failed!\n");
    return -1;
}

ret = aw_softap_save_config();
if(0 != ret)
{
    printf("save config failed!\n");
    return -1;
}

ret = aw_softap_enable();
if(0 != ret)
{
    printf("enable softap failed!\n");
    return -1;
}

ret = aw_softap_router_config("192.168.5.1", "255.255.255.0");
if(0 != ret)
{
    printf("config IP and maskcode failed!\n");
    return -1;
}

// system("udhcpd /etc/wifi/udhcpd.conf");
// usleep(500000);
ret = aw_softap_start_udhcp_dns_server();
if(0 != ret)
{
    printf("start udhcp and dns server failed!\n");
    return -1;
}

ret = aw_softap_enable_data_forward("eth0");
if(0 != ret)
{
    printf("enable data forward failed!\n");
    return -1;
}

```

```

printf("*****\n");
printf("Hostapd test succeeded!\n");
printf("*****\n");

ret = aw_softap_deinit();
if(0 != ret)
{
    printf("deinit failed!\n");
    return -1;
}
return 0;
}

```

5.2. softAP 长时耐久性测试

```

#include <stdlib.h>
#include <string.h>
#include <unistd.h>

#include <aw_softap_intf.h>

#ifndef DEFAULT_SOFTAP_LONGTIME_TEST_TIMES
#define DEFAULT_SOFTAP_LONGTIME_TEST_TIMES 10
#endif

int main(int argc, char **argv)
{
    int i = 0;
    tSOFTAP_KEY_MGMT key_mgmt = SOFTAP_WPA2_PSK;
    int ret = 0;
    int test_times = 0;
    char *psk = NULL;
    char *broadcast_hidden = NULL;
    int wait_times = 0;

    /*No arguments or only the one who is for help*/
    if((argc == 1) || (argc == 2 && (!strcmp(argv[1], "-help") || !strcmp(argv[1], "-h"))))
    {
        printf("*****\n");
        printf("softap_longtime_test -h/-help: for usage details\n");
        printf("Usage:\n");
        printf("softap_longtime_test \"ssid\" \"psk/open\" \"broadcast/hidden\" \"test_times\"\n");
        printf("*****\n");
        return 0;
    }

    if(argc >= 5)
        test_times = atoi(argv[4]);
    else
    {
        printf("No test times input, use default set: %d",
DEFAULT_SOFTAP_LONGTIME_TEST_TIMES);
        test_times = DEFAULT_SOFTAP_LONGTIME_TEST_TIMES;
    }

    printf("*****\n");
    printf("Start softap longtime test!\n");
    printf("*****\n");

```

```

for(i=0; i<test_times; i++)
{
    ret = aw_softap_init();
    if(0 != ret)
    {
        printf("init softap failed!\n");
        return -1;
    }

    printf("=====\n");
    printf("Start to up softap!\n");
    printf("Test time: %d!\n", i+1);
    printf("=====\n");

    ret = aw_softap_reload_firmware("AP");
    if(0 != ret)
    {
        printf("reload firmware failed!\n");
        return -1;
    }

/*To keep consistent with the older version demo,
this demo do no set <interface> <channel>, but
application of the user's can set them correctly by
themselves*/
    if((argc >= 3) && (argv[2][0] != '\0') && strcmp(argv[2], "open"))
    {
        psk = (char *)argv[2];
        key_mgmt = SOFTAP_WPA2_PSK;
        printf("wpa2-psk!\n");
    }
    else
    {
        key_mgmt = SOFTAP_NONE;
        printf("none key_mgmt!\n");
    }

if(argc >= 4)
    broadcast_hidden = (char *)argv[3];
else
    broadcast_hidden = (char *)"broadcast";

ret = aw_softap_config(argv[1], psk, key_mgmt, "wlan0", "6", broadcast_hidden);
if(0 != ret)
{
    printf("config softap failed!\n");
    return -1;
}

ret = aw_softap_save_config();
if(0 != ret)
{
    printf("save config failed!\n");
    return -1;
}

ret = aw_softap_enable();
if(0 != ret)
{
    printf("enable softap failed!\n");
    return -1;
}

```

```

    }

    ret = aw_softap_router_config("192.168.5.1", "255.255.255.0");
    if(0 != ret)
    {
        printf("config IP and maskcode failed!\n");
        return -1;
    }
// system("udhcpd /etc/wifi/udhcpd.conf");
// usleep(500000);
    ret = aw_softap_start_udhcp_dns_server();
    if(0 != ret)
    {
        printf("start udhcp and dns server failed!\n");
        return -1;
    }

    ret = aw_softap_enable_data_forward("eth0");
    if(0 != ret)
    {
        printf("enable data forward failed!\n");
        return -1;
    }

    printf("=====\n");
    if(aw_is_softap_started() == 1)
        printf("UP SOFTAP succeeded!\n");
    else
        printf("UP SOFTAP failed!\n");
    printf("=====\n");
    usleep(10000000);

    ret = aw_softap_reload_firmware("STA");
    if(0 != ret)
    {
        printf("reload firmware for station failed!\n");
        return -1;
    }

    ret = aw_softap_disable();
    if(0 != ret)
    {
        printf("disable softap failed!\n");
        return -1;
    }

    wait_times = 0;
    while(aw_is_softap_started() == 1 && wait_times <= 10)
    {
        usleep(2000000);
        wait_times++;
    }

    ret = aw_softap_deinit();
    if(0 != ret)
    {
        printf("deinit failed!\n");
        return -1;
    }

    printf("=====\n");

```



```
if(aw_is_softap_started() == 0)
    printf("SHUTDOWN SOFTAP succeeded!\n");
else
    printf("SHUTDOWN SOFTAP failed!\n");
    printf("=====\\n");
    usleep(4000000);

} //end of for(;;)
printf("*****\\n");
printf("Softap longtime test succeeded!\n");
printf("*****\\n");
return 0;
}
```

Allwinner

