



Melis RTOS Wi-Fi 开发指南

版本号: 1.1
发布日期: 2020-10-22

版本历史

版本号	日期	制/修订人	内容描述
1.1	2020.08.14	AW1381	1. 建立初始版本.



目 录

1 前言	1
1.1 文档简介	1
1.2 目标读者	1
1.3 适用范围	1
2 Wi-Fi 简介	2
2.1 Wi-Fi 工作模式	2
2.2 Wi-Fi 支持列表	2
2.3 代码路径	2
3 Wi-Fi 配置	3
3.1 驱动配置	3
3.2 协议栈配置	3
3.3 应用配置	4
4 Wi-Fi 使用	5
4.1 基础功能	5
4.2 应用测试	5
4.2.1 回环测试	6
4.2.2 扫描测试	7
4.2.3 联网测试	7
4.2.4 断开连接	10
4.2.5 Ping 测试	11
4.2.6 WGET 下载测试	12
4.2.7 HTTP 测试	13
4.2.8 时间同步测试	14
4.2.9 ifconfig 测试	15

1 前言

1.1 文档简介

介绍 Allwinner 平台 Melis 系统 Wi-Fi 驱动移植, 介绍的 Wi-Fi 的配置, 使用以及 Wi-Fi 常见问题。

1.2 目标读者

适用于 Melis 平台的广大客户和对 Melis Wi-Fi 感兴趣的同事。

1.3 适用范围

Allwinner 软件平台 Melis 3.0 版本

Allwinner 硬件平台 V 系列 (V833)

2 Wi-Fi 简介

2.1 Wi-Fi 工作模式

目前 Melis 平台上的 Wi-Fi 一般可处于 2 种工作模式, 分别是 STATION,AP。

- STATION: 连接无线网络的终端, 大部分无线网卡默认都处于该模式, 也是常用的一种模式。
- AP: 无线接入点, 常称热点, 比如路由器功能。

2.2 Wi-Fi 支持列表

厂商	产品名称	备注
正基	AP6203	已支持
全志	XR829	支持中

2.3 代码路径

```
驱动:melis-v3.0/source/ekernel/drivers/drv/source/net  
协议栈:melis-v3.0/source/ekernel/subsys/net/rt-thread/lwip  
应用:source/ekernel/subsys/net/rt-thread/net_tools/netutils-master
```

3 Wi-Fi 配置

Melis 下 Wi-Fi 的配置主要分为三部分：驱动，协议栈，应用；部分应用附带对应的测试用例。

在 melis-v3.0/source 目录下执行 make menuconfig。

3.1 驱动配置

```
> Kernel Setup > Drivers Setup > net drivers
  [ ] xr829 wifi driver
  [*] brcm ap6203 wifi driver
```

3.2 协议栈配置

```
> Kernel Setup > Subsystem support > Network > Network protocol
  *- lwip
  [ ] smtp
  [*] mdns
  [ ] mqtt
  [ ] altcp_tls
  [ ] snmp
  [ ] netbiosns
  [ ] websocket
  [*] mbedtls --->
      [ ] using hardware crypto module for aes.
      [ ] mbedtls test
  [ ] cyassl(wolfssl) support
  [*] http
```

3.3 应用配置

```
> Kernel Setup > Subsystem support > Network > Network tools
[*] ping
[ ] wget
[*] iperf
[ ] netio
[*] ntp
[ ] tcpdump
[ ] telnet
[ ] tftp
libcurl --->
    [*] libcurl-7.69.1
    [*] libcurl test case
libcurl2rtt --->
[ ] httpclient
[ ] webclient ----
```



4 Wi-Fi 使用

该文档暂时以 V833(AP6203 模组) 为例。

4.1 基础功能

```
bcm_sdio init //sdio初始化
bcm_init //网卡和协议栈初始化
bcm_sta命令集
    msh />bcm_sta --help
    argc=2
    sta scan
    sta scan_ext <1,2,3>
    sta scanresults
    sta connect ssid security [password]
    sta connect_ext ssid security [password] channel bssid
    sta disconnect
    sta state
    sta connection
    sta reconnect
    sta mac
bcm_sta scan //扫描,注意:该命令只会显示ssid数
bcm_sta scanresults //扫描,注意:执行该命令前必须先执行bcm_sta scan
bcm_sta connect ssid security passwd //连接
注意:这里的security通过bcm_sta scanresults命令查看,然后对应如下数组的元素位置。
char* security_name[] = {
    "open", //0
    "wpa_psk_aes", //1
    "wpa2_psk_aes", //2
    "wep-open", //3
    "wep-shared", //4
    "wpa_psk_tkip", //5
    "wpa2_psk_tkip", //6
    "wpa2_psk_mix"}; //7
暂时不支持动态识别加密方式,可以通过bcm_sta scanresults扫描后查看具体SSID的加密方式,然后联网。
例如:bcm_sta connect fly2.4g 7 22224444
```

4.2 应用测试

用例编号	用例名称	测试结果	备注
1	回环测试	PASS	
2	扫描测试	PASS	
3	联网测试	PASS	
4	断网测试	PASS	
5	Ping 压力测试	PASS	
6	Wget 下载测试	PASS	
7	http 测试	PASS	加密需配置 TLS
8	时间同步测试	PASS	
9	ifconfig 测试	PASS	

4.2.1 回环测试

用例名称:回环测试
 功能说明:利用回环地址独立测试协议栈
 前置条件:硬件贴有ap6203模组
 操作步骤:
 1.初始化: tcpip_init
 2.启动server: loop_server
 3.启动client: loop_client
 预期结果:正常收发数据
 验证情况:pass

测试 log:

```
msh />tcpip_init
msh />loop_server
Creating loop_if_server_main task
server waiting
msh />loop_client
Creating loop_if_client_main task
[loop_if_client,88],enter
=====loop_if_client 1] send total: 64=====
[loop_if_server,51],enter
=====loop_if_server=====
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F
30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
=====
msh />
=====loop_if_client 2] send total: 64=====
=====loop_if_server=====
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F
30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
```

4.2.2 扫描测试

用例名称:扫描
功能说明:扫描网络
前置条件:硬件贴有ap6203模组
操作步骤
1.初始化: bcm_sdio init;bcm_init
2.扫描: bcm_sta scan; bcm_sta scanresults
预期结果:扫描出附近ap
验证情况:PASS

测试 log:

```
msh />bcm_sta scan
argc=2
[MKD SCAN] START
[MKD SCAN] COMPLETED_SUCCESSFULLY
[MKD SCAN] END 2820 ms
[WiFi] Scan finished. AP Valid: 20. (Total: 44, Duplicated: 24)
msh />bcm_sta scanresults
argc=2

[001]
    SSID      : AW-PDC-PD4-316Test_5G
    BSSID     : 88:D7:F6:88:8B:C4
    RSSI      : -69dBm
    Security  : WPA2 PSK AES
    Channel   : 149
    Country   : CN

[002]
    SSID      : AW-PDC-PD2-hello_5G
    BSSID     : C0:61:18:F8:0B:11
    RSSI      : -51dBm
    Security  : WPA2 PSK AES
    Channel   : 149
    Country   : CN
```

4.2.3 联网测试

用例名称:联网
功能说明:连接指定网络ap
前置条件:硬件贴有ap6203模组,能正常扫描到需要连接的ap
操作步骤:
1.初始化: bcm_sdio init;bcm_init
2.联网: bcm_sta connect ssid secure passwd (如: bcm_sta connect fly2.4g 7 22224444,若要切换网络必须先断开再连接其他网络)

预期结果：正常联网
验证情况：PASS

测试 log:

```
msh />bcm_sdio init
argc=2
SDC:HAL_SDC_Create host:c28bf0f8 id:1
SDC:Not implement __mci_restore_io,820
SDC:Not imp HAL_SDC_Init, 2213
SDC:SDC Host Capability:0x3820f 0cr avail:0x3f0000
SDC:SDC cd_mode:3 present_val:0
SDC:SDC id:1 dma_use:1 present:1
  driver version SD/MMC/SDIO Host Controller Driver(v0.24 2020-7-20 14:32) init ok.
card id is 1
SD:mmc_card_create card:c2f42028 id:1
SDC:Not implement __mci_restore_io,820
SDC:SDC clock=400000 kHz,src:0, n:1, m:14
SDC:SDC clock=400000 kHz,src:0, n:1, m:14
[ERR] SDC: __mci_irq_handler,904 raw_int:100 err!
[ERR] SDC:SDC err, cmd 52,  RT0
[ERR] SDC:sd 687 abnormal status: RespErr
[ERR] SDC: __mci_irq_handler,904 raw_int:100 err!
[ERR] SDC:SDC err, cmd 52,  RT0
[ERR] SDC:sd 687 abnormal status: RespErr
[ERR] SDC: __mci_irq_handler,904 raw_int:100 err!
[ERR] SDC:SDC err, cmd 8,  RT0
[ERR] SDC:sd 687 abnormal status: RespErr
SD:sd1.0 or mmc
SD:***** Try sdio *****
HAL_GetDevClock,85 Warning Use fix value 1200000000
HAL_GetDevClock,85 Warning Use fix value 1200000000
SDC:SDC clock=25000000 kHz,src:1, n:1, m:11
[WRN] SD:cis_tpl_parse: queuing unknown CIS tuple 0x80 (2 bytes)
[WRN] SD:cis_tpl_parse: queuing unknown CIS tuple 0x80 (3 bytes)
[WRN] SD:cis_tpl_parse: queuing unknown CIS tuple 0x80 (3 bytes)
[WRN] SD:cis_tpl_parse: queuing unknown CIS tuple 0x80 (7 bytes)
[WRN] SD:cis_tpl_parse: queuing unknown CIS tuple 0x81 (9 bytes)
SD:sdio highspeed
HAL_GetDevClock,85 Warning Use fix value 1200000000
HAL_GetDevClock,85 Warning Use fix value 1200000000
SDC:SDC clock=50000000 kHz,src:1, n:0, m:11
SD:mmc_sdio_init_card bus width type:2
[WRN] SD:cis_tpl_parse: queuing unknown CIS tuple 0x91 (3 bytes)
[os E] OS_SemaphoreCreateBinary():68, OS_SemaphoreCreateBinary() NOT SUPPORT!
SD:
===== card information =====
SD:Card Type      : SDIO
SD:Card Spec Ver  : 1.0
SD:Card RCA       : 0x0001
SD:Card OCR       : 0xb0ffff00
SD:  vol_window   : 0x00ffff00
SD:  to_lv8_acpt  : 0
SD:  high_capac   : 0
SD:Card CSD      :
SD:  speed        : 50000 KHz
SD:  cmd class    : 0x0
```

```
SD: capacity : 0MB
SD:Card CUR_STA :
SD: speed_mode : DS: 25 MHz
SD: bus_width : 2
SD: speed_class : 0
SD:=====
SD:***** sdio init ok *****
Initial card success
msh />bcm_init
argc=1
208.67.222.222
8.8.8.8
202.216.228.18
dns_init: initializing

[WiFi] Starting initialization...
[WiFi] SET F2 blkksz 512
[WiFi] image_size[1949] 00
[WiFi] Wlan RAM size: configured[a0000], calculated[2000]
[WiFi] Time for WiFi FW/NV download: 20 ms
[WiFi] HT CLK Actived [70]
[WiFi] SR Disabled
[WiFi] Wlan Bus Up
[WiFi] mhdTask priority[10] stack[0x0] stack_size[4096]
[WiFi] call mhd_bus_download_clm_blob
[MKD IOCTL] max_time=10
[WiFi] CMD Init
[WiFi] Turn off TX Glomming
[WiFi] Turn on APSTA
[WiFi] Turn on AMPDU TX
[WiFi] Set country code
[MKD IOCTL] max_time=50
[WiFi] Set wl UP
[WiFi] Set GMode
[WiFi] ARPOE Enabled (ver:20120910)
[WiFi] Initialization done !

[WiFi] Chip: a804/2
[WiFi] MHD: Version 1.0.2 B014 disable sdio sleep (ad4b1fe) CC:4B:73:BF:76:C0 Aug 7 2020
10:05:39
[WiFi] wl0: Jul 8 2020 13:44:39 version 18.35.389.47.xy_t1 FWID 01-f4886cf
<<Module Init>> [490 ms]
[WiFi] Register network link up/down
msh />bcm_sta connect AW-PDC-PD2-fly2.4g 7 22224444
argc=5
Connect to -
  SSID : AW-PDC-PD2-fly2.4g
  Password : 22224444
  Security: WPA2-PSK TKIP
[WiFi] Dhcp SET enable successfully!
[WiFi] Joining AW-PDC-PD2-fly2.4g
  [SCAN_START] [127220]
[WiFi] Looking for - AW-PDC-PD2-fly2.4g
WIFI scan status [1]
[WiFi] Find - AW-PDC-PD2-fly2.4g, retry 4
  [SCAN_DONE ] [129880]
[WiFi] AP Found -
  SSID : AW-PDC-PD2-fly2.4g
  BSSID : 50:D2:F5:F1:B7:08
  Channel : 1
```

```

Security : WPA2 PSK MIXED
[JOIN_START] [129880]
[WiFi] Dhcp GET enable successfully!
[JOIN_DONE ] [132480]
Connected to -
  SSID      : AW-PDC-PD2-fly2.4g
  BSSID     : 50:d2:f5:f1:b7:08
  Channel   : 1
  Password  : 22224444
[WiFi] Join successfully !
<<Associate AP>> [5260 ms]
<<Connect AP>> [5260 ms]
[WiFi] Enable ARPOE mode (Agent + Peer Auto-Reply)
[WiFi] Bring up network interface
[WiFi] Dhcp GET config successfully!
[DHCP_DONE ] [132490]
[WiFi] Network ready IP: 192.168.31.29
<<DHCP>> [10 ms]

```

4.2.4 断开连接

用例名称: 断网
 功能说明: 断开连接的网络ap
 前置条件: 硬件贴有ap6203模组, 已经成功连接了某个ap.
 操作步骤:
 1. 先参看联网测试联网.
 2. bcm_sta disconnect
 预期结果: 正常断开连接的ap
 验证情况: PASS

测试 log:

```

msh />bcm_sta connect AW-PDC-PD2-fly2.4g 7 22224444
argc=5
Connect to -
  SSID      : AW-PDC-PD2-fly2.4g
  Password  : 22224444
  Security: WPA2-PSK TKIP
[WiFi] Dhcp SET enable successfully!
[WiFi] Joining AW-PDC-PD2-fly2.4g
[SCAN_START] [127220]
[WiFi] Looking for - AW-PDC-PD2-fly2.4g
WIFI scan status [1]
[WiFi] Find - AW-PDC-PD2-fly2.4g, retry 4
[SCAN_DONE ] [129880]
[WiFi] AP Found -
  SSID      : AW-PDC-PD2-fly2.4g
  BSSID     : 50:D2:F5:F1:B7:08
  Channel   : 1
  Security  : WPA2 PSK MIXED
[JOIN_START] [129880]

```

```
[WiFi] Dhcp GET enable successfully!
[JOIN_DONE ] [132480]
Connected to -
  SSID      : AW-PDC-PD2-fly2.4g
  BSSID     : 50:d2:f5:f1:b7:08
  Channel   : 1
  Password  : 22224444
[WiFi] Join successfully !
<<Associate AP>> [5260 ms]
<<Connect AP>> [5260 ms]
[WiFi] Enable ARPOE mode (Agent + Peer Auto-Reply)
[WiFi] Bring up network interface
[WiFi] Dhcp GET config successfully!
[DHCP_DONE ] [132490]
[WiFi] Network ready IP: 192.168.31.29
<<DHCP>> [10 ms]
msh />bcm_sta disconnect
argc=2
[WiFi] Leave ap
[WiFi] AOE Cleared all IP in dongle arp_hostip
[WiFi] Remove network interface
[WiFi] Disable ARPOE mode
```

4.2.5 Ping 测试

用例名称: Ping压力测试
功能说明: Ping 百度压力测试
前置条件: 硬件贴有ap6203模组, 已经成功连接了某个可以连接外网的ap
操作步骤:
1. 执行联网操作.
2. ping www.baidu.com
预期结果: 正常ping通, 测试过程无异常打印
验证情况: PASS

测试 log:

```
msh />ping www.baidu.com
dns_enqueue: "www.baidu.com": use DNS entry 0
dns_enqueue: "www.baidu.com": use DNS pcb 0
dns_send: dns_servers[0] "www.baidu.com": request
sending DNS request ID 23988 for name "www.baidu.com" to dns_server[0]:208.67.222.222
dns_send: dns_servers[0] "www.baidu.com": request
sending DNS request ID 23988 for name "www.baidu.com" to dns_server[0]:208.67.222.222
dns_rcv: "www.baidu.com": response = 183.232.231.174
60 bytes from 183.232.231.174 icmp_seq=0 ttl=55 time=40 ms
60 bytes from 183.232.231.174 icmp_seq=1 ttl=55 time=40 ms
60 bytes from 183.232.231.174 icmp_seq=2 ttl=55 time=20 ms
60 bytes from 183.232.231.174 icmp_seq=3 ttl=55 time=40 ms
```

4.2.6 WGET 下载测试

用例名称:Wgets多线程下载测试
功能说明:实现网络小文件下载,测试http协议
前置条件:硬件贴有ap6203模组,已经成功连接了某个可以连接外网的ap
操作步骤:
1.执行联网操作
2.wgets https://img.ivsky.com/img/tupian/pre/201312/04/nelumbo_nucifera-009.jpg (下载)
3.ls /data (查看是否有文件)
预期结果:正常下载文件并可以查看
验证情况:PASS

测试 log:

```
msh />wget http://img.ivsky.com/img/tupian/pre/201312/04/nelumbo_nucifera-009.jpg
1: Parsing url...
start:7
url:http://img.ivsky.com/img/tupian/pre/201312/04/nelumbo_nucifera-009.jpg
newurl:img/tupian/pre/201312/04/nelumbo_nucifera-009.jpg
url:http://img.ivsky.com/img/tupian/pre/201312/04/nelumbo_nucifera-009.jpg
newurl:img/tupian/pre/201312/04/nelumbo_nucifera-009.jpg
2: Get ip address...
domain:img.ivsky.com
>>>>Detail<<<<
URL: img/tupian/pre/201312/04/nelumbo_nucifera-009.jpg
DOMAIN: img.ivsky.com
IP: 122.228.204.72
PORT: 80
FILENAME: nelumbo_nucifera-009.jpg
GET img/tupian/pre/201312/04/nelumbo_nucifera-009.jpg HTTP/1.1
Accept:text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
User-Agent:Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537(KHTML, like Gecko) Chrome
/47.0.2526Safari/537.36
Host:img.ivsky.com
Connection:close
296
3: Connect server...
4: Send request...
>>>>Response header:<<<<
HTTP/1.1 400 Bad Request
Server: nginx
Date: Fri, 17 Apr 2020 02:45:00 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 166
Connection: close
5: Start thread to download...
rtos_file_name: /data/nelumbo_nucifera-009.jpg ^_^
100.00% [=====] 0.00/0.00MB
Download 100.00%
Download successful ^_^
msh />ls /data
-rwxrwxrwx root root 1970-01-01 00:00:00 0000000050B .gitignore
drwxrwxrwx root root 1970-01-01 00:00:00 0000000000B apps
drwxrwxrwx root root 1970-01-01 00:00:00 0000000000B etc
```

```
-rwxrwxrwx root root 1970-01-01 00:00:00 0000000166B nelumbo_nucifera-009.jpg
drwxrwxrwx root root 1970-01-01 00:00:00 0000000000B res
```

4.2.7 HTTP 测试

用例名称: http测试

功能说明: 基于http协议, 返回访问的http链接的html页面

前置条件: 硬件贴有ap6203模组, 已经成功连接了某个可以连接外网的ap

操作步骤:

1. 执行联网操作
2. fork https http://14.215.177.39/

预期结果: 正常返回百度的html链接

验证情况: PASS

测试 log:

```
msh />fork https http://www.baidu.com/
hostent.h_name      == www.baidu.com
hostent.h_aliases   == c257327c
hostent.h_addrtype  == 2
hostent.h_length    == 4
hostent.h_addr_list == c257339c
hostent.h_addr_list[0] == c2573294
hostent.h_addr_list[0]-> == 183.232.231.172
<!DOCTYPE html><!--STATUS OK-->
<html>
<head>
  <meta http-equiv="content-type" content="text/html;charset=utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=Edge">
  <link rel="dns-prefetch" href="//s1.bdstatic.com"/>
  <link rel="dns-prefetch" href="//t1.baidu.com"/>
  <link rel="dns-prefetch" href="//t2.baidu.com"/>
  <link rel="dns-prefetch" href="//t3.baidu.com"/>
  <link rel="dns-prefetch" href="//t10.baidu.com"/>
  <link rel="dns-prefetch" href="//t11.baidu.com"/>
  <link rel="dns-prefetch" href="//t12.baidu.com"/>
  <link rel="dns-prefetch" href="//b1.bdstatic.com"/>
  <title>百度一下, 你就知道</title>
  <link href="http://s1.bdstatic.com/r/www/cache/static/home/css/index.css" rel="
stylesheet" type="text/css" />
  <!--[if lte IE 8]><style index="index" >#content{height:480px\9}#m{top:260px\9}</
style><![endif]-->
  <!--[if IE 8]><style index="index" >#u1 a.mnav,#u1 a.mnav:visited{font-family:
simsun}</style><![endif]-->
  <script>var hashMatch = document.location.href.match(/#+(. *wd=[^&].+));if (
hashMatch && hashMatch[0] && hashMatch[1]) {document.location.replace("http://"+
location.host+"/s?" +hashMatch[1]);}var ns_c = function({});</script>
  <script>function h(obj){obj.style.behavior='url(#default#homepage)';var a = obj.
setHomePage('/www.baidu.com/');}</script>
  <noscript><meta http-equiv="refresh" content="0; url=/baidu.html?from=noscript"/></
noscript>
```



```
<script>>window._ASYNC_START=new Date().getTime();</script>
</head>
```

4.2.8 时间同步测试

用例名称: 时间同步测试
功能说明: 同步网络时间到本地时间
前置条件: 硬件贴有ap6203模组, 已经成功连接了某个ap
操作步骤:
1. 执行联网操作.
2. ntp_sync //同步时间
3. date //查看时间
预期结果: 正常获取时间
验证情况: PASS

测试 log:

```
msh />date
1970-01-01 00:16:12.937
msh />ntp_
ntp_sync
msh />ntp_sync
=====
Default NTP Server: ntp1.aliyun.com
You can use ntp_sync argv[1] replace
=====
dns_enqueue: "ntp1.aliyun.com": use DNS entry 1
dns_enqueue: "ntp1.aliyun.com": use DNS pcb 0
dns_send: dns_servers[0] "ntp1.aliyun.com": request

sending DNS request ID 54335 for name "ntp1.aliyun.com" to dns_server[0]:208.67.222.222
dns_rcv: "ntp1.aliyun.com": response = 120.25.115.20
hostent.h_name      == ntp1.aliyun.com
hostent.h_aliases   == c257327c
hostent.h_addrtype  == 2
hostent.h_length    == 4
hostent.h_addr_list == c257339c
hostent.h_addr_list[0] == c2573294
hostent.h_addr_list[0]-> == 120.25.115.20
Get local time from NTP server: Fri Aug 14 11:24:11 2020
The system time is updated. Timezone is 8.
msh />date
2020-08-14 11:24:14.175
```

4.2.9 ifconfig 测试

用例名称: ifconfig测试
功能说明: ifconfig查看获取的IP地址
前置条件: 硬件贴有ap6203模组, 已经成功连接了某个ap
操作步骤:
1. 执行联网操作
2. ifconfig //查看IP信息
预期结果: 正常获取IP信息
验证情况: PASS

测试 log:

```
msh />ifconfig  
  
mac address: cc-4b-73-bf-76-c0  
[WiFi] Dhcp GET ip address successfully!  
ip address: 192.168.43.127  
[WiFi] Dhcp GET gateway ip successfully!  
ip dns: 192.168.43.1  
[WiFi] Dhcp GET server ip successfully!  
ip server: 192.168.43.1  
[WiFi] Dhcp GET gateway ip successfully!  
ip gateway: 192.168.43.1
```




著作权声明

版权所有 © 2020 珠海全志科技股份有限公司。保留一切权利。

本档及内容受著作权法保护，其著作权由珠海全志科技股份有限公司（“全志”）拥有并保留一切权利。

本档是全志的原创作品和版权财产，未经全志书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本档内容的部分或全部，且不得以任何形式传播。

商标声明

、 **全志科技** （不完全列举）均为珠海全志科技股份有限公司的商标或者注册商标。在本档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

免责声明

您购买的产品、服务或特性应受您与珠海全志科技股份有限公司（“全志”）之间签署的商业合同和条款的约束。本档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，全志概不负责。

本档作为使用指导仅供参考。由于产品版本升级或其他原因，本档内容有可能修改，如有变更，恕不另行通知。全志尽全力在本档中提供准确的信息，但并不确保内容完全没有错误，因使用本档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，全志概不负责。本档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本档未以明示或暗示或其他方式授予全志的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。全志不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。全志不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。