



摘要

以下用户指南是 [WiLink8 Linux Wi-Fi 驱动程序 R8.8 版用户指南](#) 的续篇，详细演示了 STA、AP、MESH 和 STA+AP 等基本 Wi-Fi® 模式。本用户指南介绍了如何针对高级用例配置 WiLink™8 驱动器，例如多 BSSID、AP DFS 相关配置、AP 增强型低功耗、P2P (Wi-Fi Direct)、AP 和 STA 无线局域网唤醒 (WoWLAN) 模式。

内容

1 引言.....	3
1.1 本文档中使用的首字母缩写词.....	3
2 对等 (P2P) 模式.....	4
2.1 P2P 设备.....	4
2.2 PSP 客户端.....	5
2.3 P2P GO.....	5
2.4 P2P 命令.....	6
2.5 P2P 用例.....	9
3 接入点 (AP) 和对等 (P2P) 多角色.....	20
3.1 错误/限制.....	20
4 多 BSSID (mBSSID).....	20
4.1 设置和配置.....	20
4.2 用户指南和示例.....	21
5 接入点 (AP) 增强型低功耗 (ELP) 模式.....	22
5.1 设置和配置.....	22
5.2 用户指南和示例.....	22
5.3 错误/限制.....	22
6 WiLink8 的 WLAN 唤醒 (WoWLAN) 特性.....	23
6.1 运行模式.....	23
6.2 为 AM335x EVM 添加了“暂停/恢复” WoW 模式.....	23
6.3 WoWLAN (WLAN 唤醒) 模式启用过程.....	23
6.4 WoWLAN - 魔术包.....	26
6.5 区块确认 (BA) 过滤器设置.....	27
6.6 AM335x EVM 的硬件修改工程变更命令 (ECO) 请求.....	27
7 WiLink8 暂停恢复模式.....	27
7.1 AM437x SDK 的暂停恢复示例.....	27
8 接入点 (AP) 动态频率选择 (DFS) 主设备支持.....	29
8.1 设置和配置.....	29
8.2 用户指南和示例.....	30
8.3 错误/限制.....	30
9 站点模式 - 介绍了使用 iw 命令的备选方法.....	31
9.1 第 1 步 - 检查 wlan0 接口是否已在运行.....	31
9.2 第 2 步 - 启动 wlan0 接口 (如果未运行).....	31
9.3 第 3 步 - 将设备连接到可用的接入点.....	31
10 参考文献.....	32

商标

WiLink™ is a trademark of Texas Instruments.

Wi-Fi® is a registered trademark of Wi-Fi Alliance.

Bluetooth® is a registered trademark of Bluetooth SIG, Inc and used by Motorola, Inc. under license.

所有商标均为其各自所有者的财产。

1 引言

WiLink8 器件支持除基本 Wi-Fi 模式 (如 STA、AP 和 MESH) 外的其他特性。[表 1-1](#) 详细介绍了该器件支持的特性，并对这些特性进行了简要描述。后续各节提供了实现这些特性的详细信息。其中一些特性在其他用户指南中进行了详细介绍，并在特性描述中提供了相关链接。此外，[WiLink™ 8 WLAN 特性用户指南](#) 详细介绍了每种特性的性能指标。

表 1-1. WiLink8 高级特性列表

特性	说明
对等 (P2P) 模式	
多 BSSID (mBSSID)	在同一器件上同时运行 2 个 BSS
AP DFS 主设备	支持 AP DFS 主设备功能 (FCC/ETSI/TELEC)
AP ELP	软件 AP 的节能机制，可减少功耗
AP + P2P	AP + P2P GO/CL 多角色支持
WoWOW 支持	WiLink8 的 WLAN 唤醒
暂停/恢复	暂停和恢复 WLAN 以优化能耗
WL18xx 5GHZ 天线分集	5GHz 频带的天线分集
通过 WLAN 进行精确时间同步	在 WL8 Wi-Fi 器件之间实现精确时间同步
WiLink™ 8 WLAN 软件 - 802.11s Mesh	在 Mesh 拓扑中连接 WL18xx 器件

1.1 本文档中使用的首字母缩写词

首字母缩写词	定义
AP	接入点
DFS	动态频率切换
ELP	增强型低功耗
P2P	对等模式
GO/CL	P2P 组所有者和客户端模式
STA	站点
WoW	无线唤醒
DPDT	双极双投
GPIO	通用输入/输出
CLI	命令行界面

2 对等 (P2P) 模式

P2P 的作用是在两台设备之间直接建立 WLAN 连接，不使用路由器或 AP。P2P 在 Android 设备中被称为 Wi-Fi Direct，可从 Wi-Fi 菜单中运行。通常，P2P 角色会与 WLAN 站点角色同时存在，导致 Android 设备在 WLAN 多角色状态下运行。

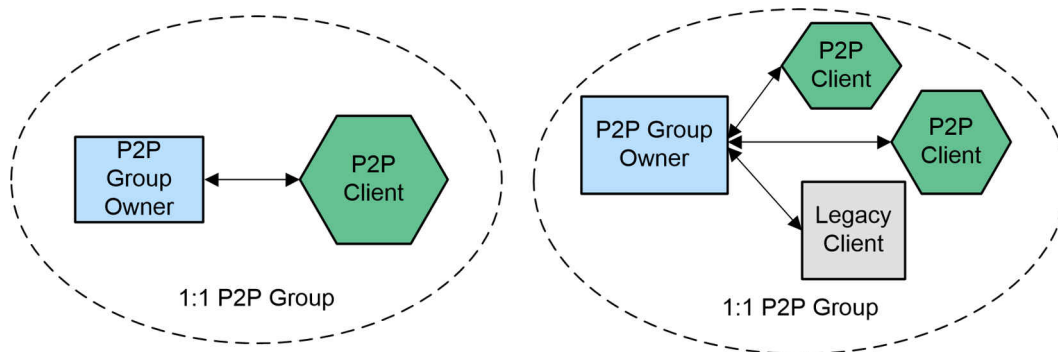


图 2-1. P2P 网络拓扑

P2P 是一种 WLAN 角色，与站点或 AP WLAN 角色的存续时长（从启用至用户明确终止使用）相比，通常具有较短的寿命。P2P 角色主要通过 Android OS 中的 Miracast 功能使用，例如，用于将 Android 智能手机屏幕镜像到具有 WLAN 模块且支持 Miracast 功能的智能 TV 或智能手机等其他设备上。

P2P 可能有三种状态：设备、客户端和 GO。连接后的 P2P 功能与 WLAN 站点和 AP 功能类似。但 P2P 具有有别于标准站点和 AP 运行的补充功能行为，允许使用不同的服务。P2P 也有自己的节能行为，支持附加电池。

2.1 P2P 设备

P2P 设备是遵循 Wi-Fi P2P 规范的 Wi-Fi 认证设备。启用 Wi-Fi Direct 或 Miracast 功能就可启用 P2P，它最初为设备状态。该状态可寻找打印机或智能 TV 等具体服务，用于发现其他可供连接的 P2P 设备。P2P 处于设备状态时会建立 P2P 连接。连接后，P2P 设备作为客户端或 GO 运行，具体取决于两台 P2P 设备间进行协商后做出的决策。P2P 连接流程包含三个步骤：搜索、协商和组构建。

2.1.1 搜索阶段

在搜索阶段，P2P 设备会发现支持 P2P 功能的所有设备，也能被其他 P2P 设备发现，以进行进一步连接。如果一些特定应用使用 P2P 功能，例如 Miracast，则只有支持 Miracast 功能的设备才会出现在待连接的设备列表中。可以进行这样的筛选是因为 P2P 设备中具有服务发现功能。P2P 设备没有可被检测的静态运行信道。因此搜索阶段包含两个阶段：扫描和监听。在扫描阶段，P2P 设备扫描 2.4GHz 和 5GHz（如果支持 5GHz）的所有 WLAN 信道，并等待支持 P2P 功能的设备响应。在监听阶段，P2P 设备留在特定信道中，这些信道被称为社会信道。在允许检测的时间段内，P2P 设备保持监听状态。搜索阶段的 P2P 检测具有统计学意义，取决于扫描和监听阶段的适当结合。检测到 P2P 设备后，它们会出现在 P2P 设备列表中。

2.1.2 协商

检测到 P2P 设备后，可以建立连接。在大多数情况下，建立连接的过程包含两个步骤：在一台 P2P 设备上的设备列表中选择目标设备，并允许此设备连接到第二台 P2P 设备。初始化的顺序并不影响连接后分配的角色。在某些情况下，例如使用 Miracast 连接对设备显示进行镜像处理，建立连接只需要在源设备上选择目标设备。调用连接流程时，连接的第一个阶段是使用 0 到 15 之间的值（通常在每台设备中提前定义）协商哪台设备在此连接中作为 GO。值较高的设备作为 GO，另一台设备作为客户端。

2.1.3 组构建

在这一阶段会建立 Wi-Fi 连接。在协商阶段，一个 P2P 设备被选作 GO。这个 GO 设备开始在工作信道上传输信标，并等待来自第二个 P2P 设备的连接。第二个 P2P 设备在搜索阶段获知该 GO 工作信道，从而能够在协商阶段过后立即开始连接。连接过程与标准 WPS 连接过程类似，包含两个阶段：创建安全密钥并使用该密钥连接，就像 WPA2-PSK 身份验证一样。连接建立后，设备的工作方式与正常 Wi-Fi 站点和 AP 类似，还有额外的 P2P 功能和节能功能（根据需要）。

2.2 PSP 客户端

设备经协商成为客户端后，会像标准 Wi-Fi 站点一样运行。其他 P2P 设备不能与它连接。P2P 客户端受 GO 指令控制，例如开始节能期。客户端设备可终止 P2P 连接（方式与 GO 设备类似）并返回设备状态，而 GO 设备如果只与此客户端连接，将在预定义的阶段内（通常为两分钟）继续作为 GO。

如果设备希望重新建立连接，它们必须完成从协商开始、包含 WPS 配置和 WPA2 连接的整个流程。但如果 P2P 设备具有特殊的 P2P “持续”功能，可以忽略较长的 WPS 段，立即使用 WPA2-PSK 身份验证进行连接。

2.3 P2P GO

在连接前的协商阶段成为了组所有者 (GO) 的设备是该组的协调器，它具有 P2P 的特殊功能和 AP 的标准功能，可以连接其他 P2P 设备，还可以连接传统 Wi-Fi 站点，例如笔记本电脑、智能手机等（如果已知晓提前共享的连接安全密钥）。加入组可以为 GO 连接其他 P2P 设备，无需协商，因为此设备已作为 GO，在此次连接中不会改变其角色。

由于 GO 的表现类似 AP，必须定期传输信标，大部分情况下处于有效状态，需要消耗更高的电流。但与 AP 在进入节能模式时的限制不同，GO 可以一次或定期调用节能模式，这样可以省电。出于节省电池电量的考虑，在 P2P 连接期间，使用电池工作的设备通常会成为客户端。

GO 和一般 P2P 的有效期会延续到某个对等设备终止连接为止。如果对等设备断开连接，另一个对等设备会停止作为 P2P 设备运行。

2.4 P2P 命令

以下部分详细介绍了用于发现、连接、对等管理、授权的基本 p2p 命令以及其他命令。各子小节还介绍了 P2P 命令的详情，以及通过执行这些命令生成的事件。

表 2-1. P2P 基本命令 - 发现、连接

命令	说明
p2p_find [timeout (seconds)] [type <social \ progressive>]	启用发现 - 开始发送探测请求帧
p2p_stop_find	停止发现，或您正在执行的任何操作（监听模式、连接流程等）
p2p_connect <device address> <PBC \ PIN> [GO_intent=<0-15> \ auth \ join]	GO_intent - 启动与另一设备的连接（使用输入的组 Intent） Auth - WPS 对传入连接授权 Join - 连接到一个现有 GO No 输入 - 使用默认 GO Intent 启动连接
p2p_listen [timeout (seconds)]	启用监听模式
p2p_group_remove <interface>	从组中移除设备，如果作为 GO 或自主 GO 则恢复设备模式
p2p_group_add	成为自主 GO

表 2-2. 管理已发现的设备的 P2P 命令

命令	说明
p2p_peers [discovered]	展示已发现的设备列表（“discovered” 仅展示完全发现的设备）
p2p_peer <address>	展示已发现的设备的详细信息
p2p_flush	清除 p2p_state，清除已发现的设备列表

表 2-3. P2P GO WPS 授权命令

命令	说明
wps_pbc	按下按钮以接受传入的连接
wps_pin <PIN> <UUID>	启用 WPS 注册者 PIN

表 2-4. P2P 其他命令

命令	说明
p2p_prov_disc <address> <method>	请求设置发现
p2p_serv_disc_req <addr> <TLVs>	发送服务发现请求
p2p_serv_disc_cancel_req <id>	取消服务发现请求
p2p_serv_disc_resp <freq> <addr> <dialog token> <TLVs>	服务发现响应
p2p_service_add <bonjour\upnp> <query\version> <response\service>	添加本地服务
p2p_service_del <bonjour\upnp> <query\version> [service]	删除本地服务
p2p_invite <cmd> [address]	向设备发送邀请
Reconfigure	Wpa_supplicant 将重新读取配置文件
Ping	Ping wpa_supplicant (仅限)

2.4.1 P2P 命令详细信息

以下各节介绍了 P2P CLI 命令的详情。

2.4.1.1 p2p_find

开始 P2P 设备发现。可使用可选参数来指定发现的持续时长，并以秒为单位 ("P2P_FIND 5")。如果未指定持续时间，发现的持续时间将是无限的，直到 P2P_STOP_FIND 或 P2P_CONNECT 将其终止 (用发现的对等设备开始创建组)。默认搜索类型是首先对所有信道运行全面扫描，然后仅继续扫描社会信道 (1、6、11)。指定另一个搜索类型可改变此行为：**social** ("P2P_FIND 5 type=social") 将跳过最初的全面扫描，仅搜索社会信道；**progressive** ("P2P_FIND type=progressive") 将首先进行全面扫描，然后在社会信道扫描时逐步搜索所有信道，每次一个信道。逐步发现设备可用于在一段时间内寻找新组 (以及在最初的扫描中未找到的组，例如，是 GO 进入睡眠状态造成的)，而不会为每轮搜索状态添加过多的额外延迟。

2.4.1.2 p2p_connect

利用发现的 P2P 对等设备开始创建 P2P 组。其中包括组所有者协商、组接口设置、配置和建立数据连接。P2P_CONNECT <peer device address> <pbcc|pin|PIN#> [label|display|keypad] [persistent] [join|auth] [go_intent=<0..15>] [freq=<in MHz>] :

- <pbcc|pin|PIN#> 参数指定了 WPS 配置方法。
- “pbcc” 字符串启动了按钮方法
- “pin” 字符串启动了 PIN 方法 (使用自动生成的 PIN，将作为命令返回代码返回)
- “PIN#” 表示可以使用提前选择的 PIN (12345670)。
- [label|display|keypad] 与 PIN 方法配合使用，指定了使用哪种 PIN
- “label” = PIN 来自本地标签
- “display” = 动态生成的随机 PIN，本地显示
- “keypad” = PIN 根据对等设备标签或显示输入)。
- “persistent” 参数可用于请求创建长期组。
- “join” 指示可通过此命令作为客户端加入现有组。会跳过 GO 协商部分。
- “auth” 指示了 WPS 参数为对等设备授权，但不会开始 GO 协商 (期待对等设备发起 GO 协商)。主要用于测试目的。
- 可选的 “go_intent” 参数可用于覆盖默认的 GO Intent 值。

2.4.1.3 p2p_listen

启动仅监听状态。使用可选参数来指定监听操作的持续时长，以秒为单位。在正常运行期间此命令可能不起什么作用，它主要用于测试。它还可用于将设备保持在可发现状态，而无需维护一个组。

2.4.1.4 p2p_group_add

手动设置 P2P 组所有者 (不与特定对等设备进行组所有者协商)。这也被称为自主 GO。P2P_GROUP_ADD [persistent] [freq=<in MHz>] :

- 使用可选的 “persistent” = <network id> 可指定长期组的重启。
- 可选的 “freq” 是该组的信道频率 (MHz)。

2.4.1.5 p2p_group_remove

终止一个 P2P 组。如果该组使用了新的虚拟网络接口，会将其删除。组接口的网络接口名将用作此命令的参数。P2P_GROUP_REMOVE <ifname> :

- <ifname> 是组接口的网络接口名，也可使用 “*” 全部删除。

2.4.1.6 p2p_peer

获取已发现对等设备的信息。展示已知 P2P 对等设备的信息。P2P_PEER [addr] [FIRST] [NEXT-<P2P Device Address>]。此命令可通过一个参数指定要选择哪个对等设备：对等设备的 P2P 设备地址，“FIRST” 指示列表中的第一个对等设备，“NEXT-<P2P Device Address>” 指示指定对等设备之后的条目 (以能够在整个列表中迭代)。

2.4.1.7 p2p_invite

邀请对等设备加入组，或（重新）启动长期组。P2P_INVITE <cmd> [peer=addr] <cmd> 格式为：
 <persistent=id> [peer=addr] 可重新启动长期组，其中 id 是唯一网络标识符 <group= > <peer=addr>
 [go_dev_addr=addr] 可邀请对等设备加入一个激活组。

2.4.2 P2P 事件详情

以下事件可以通过控制接口从 wpa_supplicant 接收。这些事件在 hostap/src/common/wpa_ctrl 中指定。

表 2-5. P2P 事件详情

P2P 事件	详细信息
P2P_EVENT_DEVICE_FOUND	指示发现了 P2P 设备，并提供有关该设备的信息。例如：P2P-DEVICE-FOUND 02:b5:64:63:30:63 p2p_dev_addr=02:b5:64:63:30:63 pri_dev_type=1-0050f204-1 name='Wireless Client' config_methods=0x84 dev_capab=0x21 group_capab=0x0
P2P_EVENT_GO_NEG_REQUEST	一台 P2P 设备已请求进行 GO 协商，但我们还没有做好协商准备。例如：P2P-GO-NEG-REQUEST 02:40:61:c2:f3:b7 dev_passwd_id=4
P2P_EVENT_GO_NEG_SUCCESS	指示成功完成了组所有者协商。P2P-GO-NEG-SUCCESS
P2P_EVENT_GO_NEG_FAILURE	指示组所有者协商失败。附加参数是 status。P2P-GO-NEG-FAILURE status=1
P2P_EVENT_GROUP_FORMATION_SUCCESS	指示成功完成了 P2P 组构建。P2P-GROUP-FORMATION-SUCCESS
P2P_EVENT_GROUP_FORMATION_FAILURE	指示 P2P 组构建失败（由于配置失败或超时）。P2P-GROUP-FORMATION-FAILURE
P2P_EVENT_GROUP_STARTED	指示构建了新的 P2P 组。附加参数：组的网络接口名、角色（GO/客户端）、SSID、频率。组中使用的已知口令也在此处提供（适用于 GO）或 PSK（适用于客户端）。对于长期组，还包含指明长期组的标志。例如：P2P-GROUP-STARTED p2p0-p2p-0 GO ssid="DIRECT-3F Testing" freq=2412 passphrase="12345678" go_dev_addr=02:40:61:c2:f3:b7 [PERSISTENT]
P2P_EVENT_GROUP_REMOVED	指示已删除了 P2P 组。附加参数：组的网络接口名、角色（GO/客户端）。例如：P2P-GROUP-REMOVED p2p0-p2p-0 GO
P2P_EVENT_CROSS_CONNECT_ENABLE	指示允许 P2P 与上行链路网络接口交叉连接。附加参数：组的网络接口名、交叉连接上行链路网络接口名。例如：P2P-CROSS-CONNECT-ENABLE p2p0-p2p-0 wan0
P2P_EVENT_CROSS_CONNECT_DISABLE	指示禁用 P2P 与上行链路网络接口交叉连接。附加参数：组的网络接口名、交叉连接上行链路网络接口名。例如：P2P-CROSS-CONNECT-DISABLE p2p0-p2p-0 wan0
P2P_EVENT_PROV_DISC_SHOW_PIN	对等设备请求本地 p2p 设备显示 PIN（将在对设备上输入）。将在事件前缀后包含以下参数：peer_address、PIN。PIN 是为连接生成的随机 PIN。P2P_CONNECT 命令可用于接受请求（使用为此连接配置的同一 PIN）。例如：P2P-PROV-DISC-SHOW-PIN 02:40:61:c2:f3:b7 12345670 p2p_dev_addr=02:40:61:c2:f3:b7 pri_dev_type=1-0050F204-1 name='Test' config_methods=0x188 dev_capab=0x21 group_capab=0x0
P2P_EVENT_PROV_DISC_ENTER_PIN	对等设备请求本地 p2p 设备输入对设备上显示的 PIN。将在事件前缀后包含以下参数：对等地址、设备类型、名称和功能。例如：P2P-PROV-DISC-ENTER-PIN 02:40:61:c2:f3:b7 p2p_dev_addr=02:40:61:c2:f3:b7 pri_dev_type=1-0050F204-1 name='Test' config_methods=0x188 dev_capab=0x21 group_capab=0x0
P2P_EVENT_PROV_DISC_PBC_REQ	对等设备请求本地 p2p 设备使用 PBC 进行连接。将在事件前缀后包含以下参数：peer_address、设备类型、名称和功能。可使用 P2P_CONNECT 命令接受请求。例如：P2P-PROV-DISC-PBC-REQ 02:40:61:c2:f3:b7 p2p_dev_addr=02:40:61:c2:f3:b7 pri_dev_type=1-0050F204-1 name='Test' config_methods=0x188 dev_capab=0x21 group_capab=0x0
P2P_EVENT_PROV_DISC_PBC_RESP	对等设备接受了我们使用 PBC 进行连接的配置发现请求。将在事件前缀后包含以下参数：peer_address。可使用 P2P_CONNECT 命令在此之后开始 GO 协商。例如：P2P-PROV-DISC-PBC-RESP 02:40:61:c2:f3:b7
P2P_EVENT_SERV_DISC_REQ	指示接受 P2P 服务发现请求。将在事件前缀后包含以下参数：频率（MHz）、源地址、对话令牌、服务更新指示器、服务查询 TLV（hexdump）。例如：P2P-SERV-DISC-REQ 2412 02:40:61:c2:f3:b7 0 0 02000001
P2P_EVENT_SERV_DISC_RESP	指示接受 P2P 服务发现响应。将在事件前缀后包含以下参数：源地址、对话令牌、服务更新指示器、服务响应 TLV（hexdump）。例如：P2P-SERV-DISC-RESP 02:40:61:c2:f3:b7 0 0 0300000101
P2P_EVENT_INVITATION_RECEIVED	指示接受 P2P 邀请请求。对于长期组，事件前缀后的参数指示哪个网络块包含长期组数据。例如：P2P-INVITATION-RECEIVED sa=02:40:61:c2:f3:b7 persistent=0

表 2-5. P2P 事件详情 (continued)

P2P 事件	详细信息
P2P_EVENT_INVITATION_RESULT	指示使用 P2P_INVITE 命令请求的 P2P 邀请的结果。参数 status=<value> 展示了对等设备返回的状态代码 (或 -1, 表示本地故障或超时)。例如: P2P-INVITATION-RESULT status=1

2.5 P2P 用例

以下部分详细介绍了如何使用 WiLink8 器件实现 P2P 用例。演示使用的硬件为 AM335x EVM 与 WL1837MODCOM8I 模块。演示将使用这两个硬件进行 P2P 连接。典型的硬件设置如图 2-2 中所示。



图 2-2. P2P 硬件设置

每种用例均使用相同的脚本启动 P2P 模式。安装 TI WiLink8 驱动程序后, 这些脚本位于 AM335x EVM 上的 /usr/share/wl18xx/。脚本详情和典型输出如下所示:

- p2p_start.sh : 启动 p2p 接口
- p2p_stop.sh : 停止 p2p 接口
- p2p_cli.sh : 进入 P2P 命令行界面 (CLI) 模式

各步骤详情和脚本典型输出如下所示:

1. 将目录更改为 “/usr/share/wl18xx/”

```
cd /usr/share/wl18xx/
```

2. 运行 p2p_start 脚本。此脚本将启动 p2p_mode。

```
root@am437x-evm:/usr/share/wl18xx# ./p2p_start.sh
Successfully initialized wpa_supplicant
[ 76.303654] wlcore: PHY firmware version: Rev 8.2.0.0.245
[ 76.392303] wlcore: firmware booted (Rev 8.9.0.0.86)
[ 76.413164] IPv6: ADDRCONF(NETDEV UP): wlan0: link is not ready wlan0: CTRL-EVENT-REGDOM-CHANGE init=USER type=COUNTRY alpha2=US
[ 76.771490] wlcore: down
root@am437x-evm:/usr/share/wl18xx#
```

3. 运行 p2p_cli 脚本。此脚本将启用 cli 模式。

```
root@am437x-evm:/usr/share/wl18xx# ./p2p_cli.sh
wpa_cli v2.9-R8.8+
Copyright (c) 2004-2019, Jouni Malinen <j@w1.fi> and contributors

This software may be distributed under the terms of the BSD license. See README for more details.

Interactive mode
>
```

4. 运行 p2p_stop.sh 脚本, 以停止 p2p 接口。以下是典型输出。

```
root@am437x-evm:/usr/share/wl18xx# ./p2p_stop.sh
OK
root@am437x-evm:/usr/share/wl18xx# n180211: deinit ifname=p2p-dev-wlan0 disabled_11b_rates=0
[ 133.968657] wlcore: down
p2p-dev-wlan0: CTRL-EVENT-TERMINATING
n180211: deinit ifname=wlan0 disabled_11b_rates=0
wlan0: CTRL-EVENT-TERMINATING
```

备注

- 请确保加入 WiFi Direct 网络的所有 EVM 均有独特的 MAC 地址，所有 IP 地址均在同一子网中。
- 默认情况下 P2P 使用 5G 频带。如果某个 EVM (例如 BeagleBone) 不支持 5G，请在 p2p 连接命令中使用附加参数 “freq=2412”。

2.5.1 PBC (按钮控制) 下的 P2P 连接

表 2-6 中列出的命令介绍了如何使用按钮控制法创建 1:1 P2P 组。图 2-2 介绍了硬件设置。需要在每个 EVM 上运行的命令在不同的列中提供，注释内容为系统将执行的操作。

表 2-6. 在 PBC 模式下创建 P2P 的命令

步骤编号	EVM #1	EVM #2	说明
1	运行 : p2p_start.sh	运行 : p2p_start.sh	cd /usr/share/wl18xx/ ./p2p_start.sh
2	运行 : p2p_cli.sh	运行 : p2p_cli.sh	./p2p_cli.sh
3	p2p_find	p2p_find	
4	p2p_peers	p2p_peers	验证 p2p 候选 MAC 地址
5	p2p_connect EVM#2_MAC_ADDRESS pbc go_intent=7		go_intent=7 表示两个 EVM 有相同的机会成为 GO ; go_intent= 15 表示该 EVM 将成为 GO ; go_intent= 0 表示该 EVM 将成为客户端
6		p2p_connect EVM#1_MAC_ADDRESS pbc	
7	使用 Ping 流量验证连接		
8	运行 : p2p_stop.sh	运行 : p2p_stop.sh	./p2p_stop.sh

以下是执行第 3 步后 EVM#1 终端的典型输出：

```
> p2p_find
OK
<3>CTRL-EVENT-SCAN-STARTED
<3>CTRL-EVENT-SCAN-STARTED
<3>CTRL-EVENT-SCAN-STARTED
> P2P-DEVICE-FOUND 54:4a:16:3a:c6:29 p2p_dev_addr=54:4a:16:3a:c6:29 pri_dev_type=0-00000000-0
name='Sitara' config_methods=0x188 dev_capab=0x25 group_capab=0x0 new=1
<3>P2P-DEVICE-FOUND 54:4a:16:3a:c6:29 p2p_dev_addr=54:4a:16:3a:c6:29 pri_dev_type=0-00000000-0
name='Sitara' config_methods=0x188 dev_capab=0x25 group_capab=0x0 new=1
<3>CTRL-EVENT-SCAN-STARTED
```

以下是执行第 3 步后 EVM#2 终端的典型输出：

```
> p2p_find
OK
<3>CTRL-EVENT-SCAN-STARTED
<3>CTRL-EVENT-SCAN-STARTED
> P2P-DEVICE-FOUND 78:a5:04:26:97:3e p2p_dev_addr=78:a5:04:26:97:3e pri_dev_type=0-00000000-0
name='Sitara' config_methods=0x188 dev_capab=0x25 group_capab=0x0 new=1
<3>P2P-DEVICE-FOUND 78:a5:04:26:97:3e p2p_dev_addr=78:a5:04:26:97:3e pri_dev_type=0-00000000-0
name='Sitara' config_methods=0x188 dev_capab=0x25 group_capab=0x0 new=1
<3>CTRL-EVENT-SCAN-STARTED
<3>CTRL-EVENT-SCAN-STARTED
> p2p_peers
78:a5:04:26:97:3e
```

以下是执行第 4 步 (EVM#1 上为 p2p_peers , EVM#2 上为 p2p_connect) 后的典型输出：

```
> p2p_peers
54:4a:16:3a:c6:29
> p2p_connect 54:4a:16:3a:c6:29 pbc auth go_intent=7
OK
<3>CTRL-EVENT-SCAN-STARTED
> P2P-FIND-STOPPED
<3>P2P-FIND-STOPPED
> P2P-GO-NEG-SUCCESS role=GO freq=5785 ht40=1 peer_dev=54:4a:16:3a:c6:29
peer_iface=56:4a:16:3a:c6:28 wps_method=PBC
<3>P2P-GO-NEG-SUCCESS role=GO freq=5785 ht40=1 peer_dev=54:4a:16:3a:c6:29
peer_iface=56:4a:16:3a:c6:28 wps_method=PBC
> rfkill: Cannot[ 163.485189] IPv6: ADDRCONF(NETDEV_UP): p2p-wlan0-0: link is not ready
open RFKILL control device
[ 163.696518] wlcore: down
p2p-wlan0-0: interface state UNINITIALIZED->HT_SCAN
Using interface p2p-wlan0-0 with hwaddr 7a:a5:04:26:97:3d and ssid "DIRECT-HS"
[ 163.964447] IPv6: ADDRCONF(NETDEV_CHANGE): p2p-wlan0-0: link becomes ready
p2p-wlan0-0: interface state HT_SCAN->ENABLED
p2p-wlan0-0: AP-ENABLED
p2p-wlan0-0: CTRL-EVENT-CONNECTED - Connection to 7a:a5:04:26:97:3d completed [id=0 id_str=]
p2p-wlan0-0: WPS-PBC-ACTIVE
<3>CTRL-EVENT-SCAN-RESULTS
> p2p-wlan0-0: CTRL-EVENT-EAP-STARTED 56:4a:16:3a:c6:28
p2p-wlan0-0: CTRL-EVENT-EAP-PROPOSED-METHOD vendor=0 method=1
p2p-wlan0-0: CTRL-EVENT-EAP-PROPOSED-METHOD vendor=14122 method=254
p2p-wlan0-0: WPS-REG-SUCCESS 56:4a:16:3a:c6:28 78efc791-3214-54b1-8c7e-5be0d1f274e2
P2P-GROUP-FORMATION-SUCCESS
<3>P2P-GROUP-FORMATION-SUCCESS
<3>P2P-GROUP-STARTED p2p-wlan0-0 GO ssid="DIRECT-HS" freq=5785 passphrase="hf300VFN"
go_dev_addr=78:a5:04:26:97:3e
> P2P-GROUP-STARTED p2p-wlan0-0 GO ssid="DIRECT-HS" freq=5785 go_dev_addr=78:a5:04:26:97:3e
p2p-wlan0-0: WPS-PBC-DISABLE
p2p-wlan0-0: WPS-SUCCESS
p2p-wlan0-0: CTRL-EVENT-EAP-FAILURE 56:4a:16:3a:c6:28
p2p-wlan0-0: AP-STA-CONNECTED 56:4a:16:3a:c6:28 p2p_dev_addr=54:4a:16:3a:c6:29
AP-STA-CONNECTED 56:4a:16:3a:c6:28 p2p_dev_addr=54:4a:16:3a:c6:29
<3>AP-STA-CONNECTED 56:4a:16:3a:c6:28 p2p_dev_addr=54:4a:16:3a:c6:29

> p2p_connect 78:a5:04:26:97:3e pbc
P2P-FIND-STOPPED
OK
<3>P2P-FIND-STOPPED
> P2P-GO-NEG-SUCCESS role=client freq=5785 ht40=1 peer_dev=78:a5:04:26:97:3e
peer_iface=7a:a5:04:26:97:3d wps_method=PBC
<3>P2P-GO-NEG-SUCCESS role=client freq=5785 ht40=1 peer_dev=78:a5:04:26:97:3e [ 152.100796] IPv6:
```

```

ADDRCONF(NETDEV_UP): p2p-wlan0-0: link is not ready
:26:97:3e peer_iface=7a:a5:04:26:97:3d wps_method=PBC
> rfkill: Cannot open RFKILL control device
p2p-wlan0-0: SME: Trying to authenticate with 7a:a5:04:26:97:3d [ 153.031799] p2p-wlan0-0:
authenticate with 7a:a5:04:26:97:3d
(SSID='DIRECT-HS' freq=5785 MHz)
[ 153.048411] p2p-wlan0-0: send auth to 7a:a5:04:26:97:3d (try 1/3)
[ 153.106117] p2p-wlan0-0: authenticated
p2p-wlan0-0: Trying to associate with 7a:a5:04:26:97:3d (SSID='D[ 153.112919] p2p-wlan0-0:
associate with 7a:a5:04:26:97:3d (try 1/3)
IRECT-HS' freq=5785 MHz)
[ 153.137937] p2p-wlan0-0: RX AssocResp from 7a:a5:04:26:97:3d (capab=0x11 status=0 aid=1)
[ 153.160889] IPv6: ADDRCONF(NETDEV_CHANGE): p2p-wlan0-0: link becomes ready
[ 153.168129] p2p-wlan0-0: associated
p2p-wlan0-0: Associated with 7a:a5:04:26:97:3d
p2p-wlan0-0: CTRL-EVENT-EAP-STARTED EAP authentication started
p2p-wlan0-0: CTRL-EVENT-EAP-PROPOSED-METHOD vendor=14122 method=1
p2p-wlan0-0: CTRL-EVENT-EAP-METHOD EAP vendor 14122 method 1 (WSC) selected
p2p-wlan0-0: WPS-CRED-RECEIVED
p2p-wlan0-0: WPS-SUCCESS
P2P-GROUP-FORMATION-SUCCESS
<3>P2P-GROUP-FORMATION-SUCCESS
> p2p-wlan0-0: CTRL-EVENT-EAP-FAILURE EAP authentication failed[ 153.587483] p2p-wlan0-0:
deauthenticating from 7a:a5:04:26:97:3d by local choice (Reason: 3=DEAUTH_LEAVING)

[ 153.641061] cfg80211: Calling CRDA to update world regulatory domain
p2p-wlan0-0: CTRL-EVENT-DISCONNECTED bssid=7a:a5:04:26:97:3d reason=3 locally_generated=1
p2p-wlan0-0: SME: Trying to authenticate with 7a:a5:04:26:97:3d [ 153.662373] p2p-wlan0-0:
authenticate with 7a:a5:04:26:97:3d
(SSID='DIRECT-HS' freq=5785 MHz)
[ 153.693043] p2p-wlan0-0: send auth to 7a:a5:04:26:97:3d (try 1/3)
[ 153.886367] p2p-wlan0-0: authenticated
p2p-wlan0-0: Trying to associate with 7a:a5:04:26:97:3d (SSID='D[ 153.890652] cfg80211: World
regulatory domain updated:
IRECT-HS' freq=5785 MHz)
[ 153.901788] cfg80211: DFS Master region: unset
[ 153.908091] cfg80211: (start_freq - end_freq @ bandwidth), (max_antenna_gain, max_eirp),
(dfs_cac_time)
[ 153.917925] cfg80211: (2402000 KHz - 2472000 KHz @ 40000 KHz), (N/A, 2000 mBm), (N/A)
[ 153.926348] cfg80211: (2457000 KHz - 2482000 KHz @ 40000 KHz), (N/A, 2000 mBm), (N/A)
[ 153.934436] cfg80211: (2474000 KHz - 2494000 KHz @ 20000 KHz), (N/A, 2000 mBm), (N/A)
[ 153.942746] cfg80211: (5170000 KHz - 5250000 KHz @ 80000 KHz, 160000 KHz AUTO), (N/A, 2000
mBm), (N/A)
[ 153.953124] cfg80211: (5250000 KHz - 5330000 KHz @ 80000 KHz, 160000 KHz AUTO), (N/A, 2000
mBm), (0 s)
[ 153.962716] cfg80211: (5490000 KHz - 5730000 KHz @ 160000 KHz), (N/A, 2000 mBm), (0 s)
[ 153.970848] cfg80211: (5735000 KHz - 5835000 KHz @ 80000 KHz), (N/A, 2000 mBm), (N/A)
[ 153.979253] cfg80211: (57240000 KHz - 63720000 KHz @ 2160000 KHz), (N/A, 0 mBm), (N/A)
[ 153.987494] cfg80211: Calling CRDA for country: US
p2p-wlan0-0: CTRL-EVENT-REGDOM-CHANGE init=CORE type=WORLD[ 154.002636] p2p-wlan0-0: associate
with 7a:a5:04:26:97:3d (try 1/3)
[ 154.122766] cfg80211: Regulatory domain changed to country: US
[ 154.129005] p2p-wlan0-0: RX AssocResp from 7a:a5:04:26:97:3d (capab=0x11 status=0 aid=1)
[ 154.137984] cfg80211: DFS Master region: FCC
[ 154.142205] cfg80211: (start_freq - end_freq @ bandwidth), (max_antenna_gain, max_eirp),
(dfs_cac_time)
[ 154.153611] cfg80211: (2402000 KHz - 2472000 KHz @ 40000 KHz), (N/A, 3000 mBm), (N/A)
[ 154.161787] cfg80211: (5170000 KHz - 5250000 KHz @ 80000 KHz, 160000 KHz AUTO), (N/A, 1700
mBm), (N/A)
[ 154.172011] cfg80211: (5250000 KHz - 5330000 KHz @ 80000 KHz, 160000 KHz AUTO), (N/A, 2300
mBm), (0 s)
[ 154.181774] cfg80211: (5735000 KHz - 5835000 KHz @ 80000 KHz), (N/A, 3000 mBm), (N/A)
[ 154.191168] cfg80211: (57240000 KHz - 63720000 KHz @ 2160000 KHz), (N/A, 4000 mBm), (N/A)
p2p-wlan0-0: CTRL-EVENT-REGDOM-CHANGE init=USER type=COUNTRY alpha2=US
[ 154.212782] p2p-wlan0-0: associated
p2p-wlan0-0: Associated with 7a:a5:04:26:97:3d
p2p-wlan0-0: WPA: Key negotiation completed with 7a:a5:04:26:97:[ 154.291945] wlcore: Association
completed.
3d [PTK=CCMP GTK=CCMP]
p2p-wlan0-0: CTRL-EVENT-CONNECTED - Connection to 7a:a5:04:26:97:3d completed [id=0 id_str=]
<3>P2P-GROUP-STARTED p2p-wlan0-0 client ssid="DIRECT-HS" freq=5785
psk=08dd88ac41f0a0d4321fd33de19e35d7a54fb827e8a5359193d8487880e15704 go_dev_addr=78:a5:04:26:97:3e
> P2P-GROUP-STARTED p2p-wlan0-0 client ssid="DIRECT-HS" freq=5785 go_dev_addr=78:a5:04:26:97:3e
    
```

在两个 EVM 上点击 “q” 退出请求程序 CLI。

> q

在两个 EVM 上将 IP 地址分配给 P2P 接口，如下所示：

```

root@am437x-evm:/usr/share/wl18xx# ifconfig p2p-wlan0-0 192.168.3.3
root@am437x-evm:/usr/share/wl18xx# ifconfig -a p2p-wlan0-0

p2p-wlan0-0 Link encap:Ethernet HWaddr 7A:A5:04:26:97:3D
  inet addr:192.168.3.3 Bcast:192.168.3.255 Mask:255.255.255.0
  inet6 addr: fe80::78a5:4ff:fe26:973d/64 Scope:Link
  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
  RX packets:33 errors:0 dropped:2 overruns:0 frame:0
  TX packets:36 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:1000
  RX bytes:2919 (2.8 KiB) TX bytes:4457 (4.3 KiB)

root@am335x-evm:/usr/share/wl18xx# ifconfig p2p-wlan0-0 192.168.3.4
root@am335x-evm:/usr/share/wl18xx# ifconfig -a p2p-wlan0-0

p2p-wlan0-0 Link encap:Ethernet HWaddr 56:4A:16:3A:C6:28
  inet addr:192.168.3.4 Bcast:192.168.3.255 Mask:255.255.255.0
  inet6 addr: fe80::544a:16ff:fe3a:c628/64 Scope:Link
  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
  RX packets:26 errors:0 dropped:0 overruns:0 frame:0
  TX packets:38 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:1000
  RX bytes:3023 (2.9 KiB) TX bytes:4041 (3.9 KiB)

```

执行第 7 步中指定的 ping 测试，以验证连接。

表 2-7. P2P 模式下的 Ping 响应

EVM#1 上的 Ping 响应	EVM#2 上的 Ping 响应
<pre> root@am437x-evm:/usr/share/wl18xx# ping 192.168.3.4 PING 192.168.3.4 (192.168.3.4): 56 data bytes 64 bytes from 192.168.3.4: seq=0 ttl=64 time=193.529 ms 64 bytes from 192.168.3.4: seq=1 ttl=64 time=1032.839 ms 64 bytes from 192.168.3.4: seq=2 ttl=64 time=32.006 ms </pre>	<pre> root@am335x-evm:/usr/share/wl18xx# ping 192.168.3.3 PING 192.168.3.3 (192.168.3.3): 56 data bytes 64 bytes from 192.168.3.3: seq=0 ttl=64 time=229.453 ms 64 bytes from 192.168.3.3: seq=1 ttl=64 time=9.430 ms 64 bytes from 192.168.3.3: seq=2 ttl=64 time=10.999 ms </pre>

若要退出 p2p 测试，在两个 EVM 上运行 p2p_stop.sh :

EVM#1 上的 p2p_stop 响应	EVM#2 上的 p2p_stop 响应
<pre> root@am437x-evm:/usr/share/wl18xx# ./p2p_stop.sh P2P-GROUP-REMOVED p2p-wlan0-0 GO reason=REQUESTED p2p-wlan0-0: interface state ENABLED->DISABLED p2p-wlan0-0: AP-STA-DISCONNECTED 56:4a:16:3a:c6:28 p2p_dev_addr=54:4a:16:3a:c6:29 AP-STA-DISCONNECTED 56:4a:16:3a:c6:28 p2p_dev_addr=54:4a:16:3a:c6:29 p2p-wlan0-0: AP-DISABLED p2p-wlan0-0: CTRL-EVENT-DISCONNECTED bssid=7a:a5:04:26:97:3d reason=3 locally_generated=1 nl80211: deinit ifname=p2p-wlan0-0 disabled_11b_rates=1 [1135.058889] wlc core: down P2P-DEVICE-LOST p2p_dev_addr=54:4a:16:3a:c6:29 nl80211: deinit ifname=p2p-dev-wlan0 disabled_11b_rates=0 OK root@am437x-evm:/usr/share/wl18xx# [1135.173022] wlc core: down p2p-dev-wlan0: CTRL-EVENT-TERMINATING nl80211: deinit ifname=wlan0 disabled_11b_rates=0 wlan0: CTRL-EVENT-TERMINATING </pre>	<pre> root@am335x-evm:/usr/share/wl18xx# ./ p2p_stop.shnl80211: deinit ifname=p2p-dev-wlan0 disabled_11b_rates=0OK[1169.243405] wlc core: downroot@am335x-evm:/usr/share/wl18xx# p2p-dev- wlan0: CTRL-EVENT-TERMINATINGnl80211: deinit ifname=wlan0 disabled_11b_rates=0wlan0: CTRL- EVENT-TERMINATING </pre>

2.5.2 创建自主 1:2 P2P 组 (按钮控制)

以下部分详细介绍了使用图 2-2 中的硬件设置创建自主 1:2 P2P 组 (按钮控制) 网络的详细信息。

表 2-8. 创建自主 1:2 P2P 组 (按钮控制)

步骤编号	EVM #1	EVM #2	说明
1	运行 : p2p_start.sh	运行 : p2p_start.sh	cd /usr/share/wl18xx/ ./p2p_start.sh
2	运行 : p2p_cli.sh	运行 : p2p_cli.sh	./p2p_cli.sh
3	p2p_find	p2p_find	
4	p2p_peers	p2p_peers	验证 p2p 候选 MAC 地址
5	p2p_group_add		将 EVM#1 定义为组所有者 (GO)
6	在 EVM#1 上退出 wpa_cli		键入 : quit
7	使用新创建的“p2p-wlan0-0”接口重新进入 wpa_cli		wpa_cli -ip2p-wlan0-0
8	wps_pbc		在按钮模式下工作
9		p2p_connect EVM#1_MAC_ADDRESS pbc join	
10	退出 wpa_cli	退出 wpa_cli	键入 : quit
11	获取 p2p-wlan0-0 的 IP 地址	获取 p2p-wlan0-0 的 IP 地址	EVM#1: ifconfig p2p-wlan0-0 192.168.3.3 EVM#2: ifconfig p2p-wlan0-0 192.168.3.4
12	使用 ping 验证连接		EVM#1: ping 192.168.3.4 EVM#2: ping 192.168.3.3

有两种方法可以配置组所有者和客户端的 IP 地址 :

- 静态 IP - 在上一节中进行了介绍。
- 在 GO EVM 上启用 DHCP 服务器，使用 DHCP 会话为客户端提供 IP 地址。下一节将介绍这部分内容。

2.5.2.1 使用 DHCP 方法在 P2P 模式下分配 IP 地址

以下各节详细介绍了如何在组所有者 EVM 上使用 DHCP 方法为客户端分配 IP 地址，步骤如下：

- 配置 udhcpd 配置文件
- 运行 dhcp 服务器
- 定义 iptables 以启用 NAT (网络地址转换)

配置 udhcpd.conf

首先，我们要调用以下命令，备份现有 `udhcpd.conf` 文件：

```
mv /etc/udhcpd.conf /etc/udhcpd.conf.bak
```

用以下内容新建 `udhcpd.conf`：

```
# Sample udhcpd configuration file (/etc/udhcpd.conf)
# The start and end of the IP lease block
start 192.168.0.20 #default: 192.168.0.20
end 192.168.0.254 #default: 192.168.0.254
# The interface that udhcpd will use
Interface p2p-wlan0-0 #default: eth0
#Examples
opt dns 8.8.8.8 8.8.4.4 # public google dns servers
option subnet 255.255.255.0
opt router 192.168.0.1
option lease 864000 # 10 days of of lease
```

建立连接后，执行以下步骤

- 启用 IPv4 转发
- 分配 IP 地址
- 启动 `udhcpd`，以运行 DHCP 服务器
- 定义 `iptables`，以启用网络地址转换 (NAT)

执行以上操作的命令为：

```
root@am335x-evm:~# echo 1 > /proc/sys/net/ipv4/ip_forward
root@am335x-evm:~# ifconfig p2p-wlan0-0 192.168.0.1
root@am335x-evm:~# udhcpd /etc/udhcpd.conf
```

备注

为“`p2p-wlan0-0`”接口分配的 IP 地址 (`ifconfig p2p-wlan0-0 192.168.0.1`) 应与 `udhcpd.conf` 文件 (在上文中编辑) 中“`router`”字段的 IP 匹配。

在客户端要使用 DHCP 方法获得 IP 地址，运行以下代码：

```
root@am335x-evm:~# udhcpc -i p2p-wlan0-0
udhcpc (v1.20.2) started
Sending discover...
Sending select for 192.168.0.20...
Lease of 192.168.0.20 obtained, lease time 864000
/etc/udhcpc.d/50default: Adding DNS 8.8.8.8
/etc/udhcpc.d/50default: Adding DNS 8.8.4.4
root@am335x-evm:~#
```

表 2-9 展示了 DHCP 用例。

表 2-9. 使用 DHCP 方法配置 P2P IP 地址

步骤编号	EVM #1	EVM #2	说明
1	运行 : p2p_start.sh	运行 : p2p_start.sh	cd /usr/share/wl18xx/ ./ p2p_start.sh
2	运行 : p2p_cli.sh	运行 : p2p_cli.sh	./p2p_cli.sh
3	p2p_find	p2p_find	
4	p2p_peers	p2p_peers	验证 p2p 候选 MAC 地址
5	p2p_group_add		将 EVM#1 定义为组所有者 (GO)
6	在 EVM#1 上退出 wpa_cli		键入 : quit
7	使用新创建的“p2p-wlan0-0”接口重新进入 wpa_cli		wpa_cli -ip2p-wlan0-0
8	wps_pbc		在按钮模式下工作
9		p2p_connect EVM#1_MAC_ADDRESS pbc join	
10	退出 wpa_cli	退出 wpa_cli	键入 : quit
11	echo 1 > /proc/sys/net/ipv4/ip_forward		启用 IP 转发
12	ifconfig p2p-wlan0-0 192.168.0.1		在 EVM#1 上获得 p2p-wlan0-0 的 IP (必须与 DHCP 配置文件中的 router 字段相同)
13	udhcpd /etc/udhcpd.conf		启动 DHCP 服务器

2.5.2.2 创建 1:2 P2P 组 - 利用 PIN 码连接

以下部分详细介绍了如何使用 PIN 码创建 P2P 组。

步骤编号	EVM #1	EVM #2	说明
1	运行 : p2p_start.sh	运行 : p2p_start.sh	cd /usr/share/wl18xx/ ./ p2p_start.sh
2	运行 : p2p_cli.sh	运行 : p2p_cli.sh	./p2p_cli.sh
3	p2p_find	p2p_find	使用 wpa_cli 前缀运行 wpa cli 命令 (而不是进入 wpa_cli 实用程序)
4	p2p_peers	p2p_peers	验证 p2p 候选 MAC 地址
5	p2p_connect EVM#2_MAC_ADDRESS pin		该命令将在命令的下一行输出 EVM#1 的 PIN 码
6		p2p_connect EVM#2_MAC_ADDRESS EVM#1_PIN_CODE	从上一条 EVM#1 命令获得 EVM#1_PIN_CODE
7	定义 IP 地址	定义 IP 地址	EVM#1: ifconfig p2p-wlan0-0 192.168.3.3 EVM#2: ifconfig p2p-wlan0-0 192.168.3.4
8	使用 ping 验证连接		EVM#1: ping 192.168.3.4 EVM#2: ping 192.168.3.3

以下是上面列出的每条命令的输出详情。响应 p2p_cli.sh、p2p_find、p2p_peers 的命令相同，已在之前的章节中详细介绍。EVM 找到对等设备之后，就使用 PIN 建立连接，如下所示：

```
> p2p_peers
54:4a:16:3a:c6:29
```


EVM#1 在连接命令中定义 **EVM#2** MAC 地址，该命令的响应可得到 **EVM#1** Pin 码 (在以下示例中为 **04194996**)。**EVM#2** 尝试利用 **EVM#1** MAC 地址和 Pin 码连接 **EVM#1**。

```
> p2p_connect 78:a5:04:26:97:3e 04194996
OK
> P2P-GO-NEG-SUCCESS role=client freq=5745 ht40=1 peer_dev=78:a5:04:26:97:3e
peer_iface=7a:a5:04:26:97:3d wps_method=Keypad
<3>P2P-GO-NEG-SUCCESS role=client freq=5745 ht40=1 peer_dev=78:a5[ 1954.766093] IPv6:
ADDRCONF(NETDEV_UP): p2p-wlan0-0: link is not ready
:04:26:97:3e peer_iface=7a:a5:04:26:97:3d wps_method=Keypad
> rfkill: Cannot Open RFKILL control device
<3>CTRL-EVENT-SCAN-RESULTS
<3>CTRL-EVENT-SCAN-RESULTS
> p2p-wlan0-0: SME: Trying to authenticate with 7a:a5:04:26:97:3d [ 1955.517183] p2p-wlan0-0:
authenticate with 7a:a5:04:26:97:3d
(SSID='DIRECT-Ka' freq=5745 MHz)
[ 1955.534399] p2p-wlan0-0: send auth to 7a:a5:04:26:97:3d (try 1/3)
[ 1955.602583] p2p-wlan0-0: authenticated
p2p-wlan0-0: Trying to associate with 7a:a5:04:26:97:3d (SSID='DIRECT-Ka' freq=5745 MHz)
[ 1955.613139] p2p-wlan0-0: associate with 7a:a5:04:26:97:3d (try 1/3)
[ 1955.643432] p2p-wlan0-0: RX AssocResp from 7a:a5:04:26:97:3d (capab=0x11 status=0 aid=1)
[ 1955.665736] IPv6: ADDRCONF(NETDEV_CHANGE): p2p-wlan0-0: link becomes ready
[ 1955.672755] p2p-wlan0-0: associated
p2p-wlan0-0: Associated with 7a:a5:04:26:97:3d
p2p-wlan0-0: CTRL-EVENT-EAP-STARTED EAP authentication started
p2p-wlan0-0: CTRL-EVENT-EAP-PROPOSED-METHOD vendor=14122 method=1
p2p-wlan0-0: CTRL-EVENT-EAP-METHOD EAP vendor 14122 method 1 (WSC) selected
p2p-wlan0-0: WPS-CRED-RECEIVED
p2p-wlan0-0: WPS-SUCCESS
P2P-GROUP-FORMATION-SUCCESS
<3>P2P-GROUP-FORMATION-SUCCESS
> p2p-wlan0-0: CTRL-EVENT-EAP-FAILURE EAP authentication failed[ 1956.259741] p2p-wlan0-0:
deauthenticating from 7a:a5:04:26:97:3d by local choice (Reason: 3=DEAUTH_LEAVING)

[ 1956.314091] cfg80211: Calling CRDA to update world regulatory domain
p2p-wlan0-0: CTRL-EVENT-DISCONNECTED bssid=7a:a5:04:26:97:3d reason=3 locally_generated=1
p2p-wlan0-0: SME: Trying to authenticate with 7a:a5:04:26:97:3d [ 1956.338506] p2p-wlan0-0:
authenticate with 7a:a5:04:26:97:3d
(SSID='DIRECT-Ka' freq=5745 MHz)
[ 1956.366373] p2p-wlan0-0: send auth to 7a:a5:04:26:97:3d (try 1/3)
[ 1956.523895] p2p-wlan0-0: authenticated
p2p-wlan0-0: Trying to associate with 7a:a5:04:26:97:3d (SSID='D[ 1956.528187] cfg80211: World
regulatory domain updated:
IRECT-Ka' freq=5745 MHz)
[ 1956.539313] cfg80211: DFS Master region: unset
[ 1956.545556] cfg80211: (start_freq - end_freq @ bandwidth), (max_antenna_gain, max_eirp),
(dfs_cac_time)
[ 1956.555800] cfg80211: (2402000 KHz - 2472000 KHz @ 40000 KHz), (N/A, 2000 mBm), (N/A)
[ 1956.564298] cfg80211: (2457000 KHz - 2482000 KHz @ 40000 KHz), (N/A, 2000 mBm), (N/A)
[ 1956.572350] cfg80211: (2474000 KHz - 2494000 KHz @ 20000 KHz), (N/A, 2000 mBm), (N/A)
[ 1956.580685] cfg80211: (5170000 KHz - 5250000 KHz @ 80000 KHz, 160000 KHz AUTO), (N/A, 2000
mBm), (N/A)
[ 1956.590251] cfg80211: (5250000 KHz - 5330000 KHz @ 80000 KHz, 160000 KHz AUTO), (N/A, 2000
mBm), (0 s)
[ 1956.600097] cfg80211: (5490000 KHz - 5730000 KHz @ 160000 KHz), (N/A, 2000 mBm), (0 s)
[ 1956.608334] cfg80211: (5735000 KHz - 5835000 KHz @ 80000 KHz), (N/A, 2000 mBm), (N/A)
[ 1956.616665] cfg80211: (57240000 KHz - 63720000 KHz @ 2160000 KHz), (N/A, 0 mBm), (N/A)
[ 1956.624925] cfg80211: Calling CRDA for country: US
p2p-wlan0-0: CTRL-EVENT-REGDOM-CHANGE init=CORE type=WORLD
[ 1956.642652] p2p-wlan0-0: associate with 7a:a5:04:26:97:3d (try 1/3)
[ 1956.667467] p2p-wlan0-0: RX AssocResp from 7a:a5:04:26:97:3d (capab=0x11 status=0 aid=1)
[ 1956.768148] cfg80211: Regulatory domain changed to country: US
[ 1956.774906] cfg80211: DFS Master region: FCC
[ 1956.779752] cfg80211: (start_freq - end_freq @ bandwidth), (max_antenna_gain, max_eirp),
(dfs_cac_time)
[ 1956.790725] cfg80211: (2402000 KHz - 2472000 KHz @ 40000 KHz), (N/A, 3000 mBm), (N/A)
[ 1956.800254] cfg80211: (5170000 KHz - 5250000 KHz @ 80000 KHz, 160000 KHz AUTO), (N/A, 1700
mBm), (N/A)
[ 1956.810834] cfg80211: (5250000 KHz - 5330000 KHz @ 80000 KHz, 160000 KHz AUTO), (N/A, 2300
mBm), (0 s)
[ 1956.821088] cfg80211: (5735000 KHz - 5835000 KHz @ 80000 KHz), (N/A, 3000 mBm), (N/A)
[ 1956.830259] cfg80211: (57240000 KHz - 63720000 KHz @ 2160000 KHz), (N/A, 4000 mBm), (N/A)
p2p-wlan0-0: CTRL-EVENT-REGDOM-CHANGE init=USER type=COUNTRY alpha2=US
[ 1956.847737] p2p-wlan0-0: associated
p2p-wlan0-0: Associated with 7a:a5:04:26:97:3d
p2p-wlan0-0: WPA: Key negotiation completed with 7a:a5:04:26:97:[ 1956.867484] wlcore: Association
completed.
3d [PTK=CCMP GTK=CCMP]
```

```
p2p-wlan0-0: CTRL-EVENT-CONNECTED - Connection to 7a:a5:04:26:97:3d completed [id=0 id_str=]
<3>P2P-GROUP-STARTED p2p-wlan0-0 client ssid="DIRECT-Ka" freq=5745
psk=b7e653e4fef3ddc385497ea2df892b3c520c438cc763e91064bc1d92b6fcf37c go_dev addr=78:a5:04:26:97:3e
> P2P-GROUP-STARTED p2p-wlan0-0 client ssid="DIRECT-Ka" freq=5745 go_dev_addr=78:a5:04:26:97:3e
```

EVM#1 的典型响应如下所示：

```
> P2P-GO-NEG-SUCCESS role=GO freq=5745 ht40=1 peer_dev=54:4a:16:3a:c6:29
peer_iface=56:4a:16:3a:c6:28 wps_method=Display
<3>P2P-GO-NEG-SUCCESS role=GO freq=5745 ht40=1 peer_dev=54:4a:16:3a:c6:29
peer_iface=56:4a:16:3a:c6:28 wps_method=Display
> rfkill: Cannot open RFKILL control device
[ 1966.108421] IPv6: ADDRCONF(NETDEV_UP): p2p-wlan0-0: link is not ready
[ 1966.265558] wlcore: down
p2p-wlan0-0: interface state UNINITIALIZED->HT_SCAN
Using interface p2p-wlan0-0 with hwaddr 7a:a5:04:26:97:3d and ssid "DIRECT-Ka"
[ 1966.524889] IPv6: ADDRCONF(NETDEV_CHANGE): p2p-wlan0-0: link becomes ready
p2p-wlan0-0: interface state HT_SCAN->ENABLED
p2p-wlan0-0: AP-ENABLED
p2p-wlan0-0: CTRL-EVENT-CONNECTED - Connection to 7a:a5:04:26:97:3d completed [id=0 id_str=]
<3>CTRL-EVENT-SCAN-RESULTS
> p2p-wlan0-0: CTRL-EVENT-EAP-STARTED 56:4a:16:3a:c6:28
p2p-wlan0-0: CTRL-EVENT-EAP-PROPOSED-METHOD vendor=0 method=1
p2p-wlan0-0: CTRL-EVENT-EAP-PROPOSED-METHOD vendor=14122 method=254
p2p-wlan0-0: WPS-REG-SUCCESS 56:4a:16:3a:c6:28 78efc791-3214-54b1-8c7e-5be0d1f274e2
P2P-GROUP-FORMATION-SUCCESS
<3>P2P-GROUP-FORMATION-SUCCESS
<3>P2P-GROUP-STARTED p2p-wlan0-0 GO ssid="DIRECT-Ka" freq=5745 passphrase="VRtsc4Nz"
go_dev_addr=78:a5:04:26:97:3e
> P2P-GROUP-STARTED p2p-wlan0-0 GO ssid="DIRECT-Ka" freq=5745 go_dev_addr=78:a5:04:26:97:3e
p2p-wlan0-0: WPS-SUCCESS
p2p-wlan0-0: CTRL-EVENT-EAP-FAILURE 56:4a:16:3a:c6:28
p2p-wlan0-0: AP-STA-CONNECTED 56:4a:16:3a:c6:28 p2p_dev_addr=54:4a:16:3a:c6:29
AP-STA-CONNECTED 56:4a:16:3a:c6:28 p2p_dev_addr=54:4a:16:3a:c6:29
<3>AP-STA-CONNECTED 56:4a:16:3a:c6:28 p2p_dev_addr=54:4a:16:3a:c6:29
```

这时，通过点击控制台中的“q”，两个 EVM 均可退出 CLI 界面。

下一步是指定静态 IP 地址并验证 P2P 客户端可以使用该 IP 地址通信。使用 ping 序列进行验证。

EVM#1	EVM#2
<pre>root@am437x-evm:/usr/share/wl18xx# ifconfig p2p-wlan0-0 192.168.3.3 root@am437x-evm:/usr/share/wl18xx# ifconfig -a p2p-wlan0 p2p-wlan0-0 Link encap:Ethernet HWaddr 7A:A5:04:26:97:3D inet addr:192.168.3.3 Bcast:192.168.3.255 Mask:255.255.255.0 inet6 addr: fe80::78a5:4ff:fe26:973d/64 Scope:Link UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1 RX packets:8 errors:0 dropped:2 overruns:0 frame:0 TX packets:18 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:1253 (1.2 KiB) TX bytes:2503 (2.4 KiB)</pre>	<pre>root@am335x-evm:/usr/share/wl18xx# ifconfig p2p-wlan0-0 192.168.3.4 root@am335x-evm:/usr/share/wl18xx# ifconfig -a p2p-wlan0 p2p-wlan0-0 Link encap:Ethernet HWaddr 56:4A:16:3A:C6:28 inet addr:192.168.3.4 Bcast:192.168.3.255 Mask:255.255.255.0 inet6 addr: fe80::544a:16ff:fe3a:c628/64 Scope:Link UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1 RX packets:10 errors:0 dropped:0 overruns:0 frame:0 TX packets:13 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:1511 (1.4 KiB) TX bytes:1903 (1.8 KiB)</pre>

指定 IP 地址后可以验证连接，如下所示：

EVM#1	EVM#2

```

root@am437x-evm:/usr/share/wl18xx# ping
192.168.3.4
PING 192.168.3.4 (192.168.3.4): 56 data bytes
64 bytes from 192.168.3.4: seq=0 ttl=64
time=1384.869 ms
64 bytes from 192.168.3.4: seq=1 ttl=64
time=383.960 ms
64 bytes from 192.168.3.4: seq=2 ttl=64
time=97.437 ms
64 bytes from 192.168.3.4: seq=3 ttl=64
time=122.172 ms

```

```

root@am335x-evm:/usr/share/wl18xx# ping
192.168.3.3
PING 192.168.3.3 (192.168.3.3): 56 data bytes
64 bytes from 192.168.3.3: seq=0 ttl=64
time=11.911 ms
64 bytes from 192.168.3.3: seq=1 ttl=64
time=1115.697 ms
64 bytes from 192.168.3.3: seq=2 ttl=64
time=115.208 ms
64 bytes from 192.168.3.3: seq=3 ttl=64
time=9.148 ms

```

2.5.2.3 P2P 邀请过程 - 创建自主 1:2 P2P 组 (按钮控制)

以下部分详细介绍了如何使用 PIN 方法来连接 P2P。以下是通用步骤：

1. P2P 组所有者或该 P2P 组中的一个 P2P 客户端传输 P2P 邀请请求帧。
2. 接收 P2P 邀请请求后，支持 P2P 邀请过程信令机制的 P2P 设备会传输一个 P2P 邀请响应帧。
3. 接受邀请的决定由受邀 P2P 设备做出

以下命令流展示了 EVM#1 为 GO 时邀请 EVM#2 的情况。

表 2-10. P2P 邀请过程 - 创建自主 1:2 P2P 组 (按钮控制) 命令序列

步骤编号	EVM #1	EVM #2	说明
1	运行：p2p_start.sh	运行：p2p_start.sh	cd /usr/share/wl18xx/ ./p2p_start.sh
2	运行：p2p_cli.sh	运行：p2p_cli.sh	./p2p_cli.sh
3	p2p_group_add freq=2412		
4	退出 wpa_cli (在 EVM#1 上)		键入：quit
5	在 EVM#1 上使用新创建的“p2p-wlan0-0”接口重新进入 wpa_cli		wpa_cli -ip2p-wlan0-0
6	wps_pbc		
7	等待，直到远程“p2p_find”之后		
8		p2p_find	
9	p2p_invite group=p2p-wlan0-0 peer=EVM#2_MAC_ADDRESS		
10		p2p_connect EVM#1_MAC_ADDRESS pbc join	
11	退出 wpa_cli	退出 wpa_cli	键入：quit
12	获取 p2p-wlan0-0 的 IP 地址	获取 p2p-wlan0-0 的 IP 地址	EVM#1: ifconfig p2p-wlan0-0 192.168.3.3 EVM#2: ifconfig p2p-wlan0-0 192.168.3.4
13	使用 ping 验证连接		EVM#1: ping 192.168.3.4 EVM#2: ping 192.168.3.3

3 接入点 (AP) 和对等 (P2P) 多角色

在同一信道上，可同时使用 WiLink8 AP 模式与对等 (P2P) GO 模式。不需要进行任何特殊配置。但要注意以下事项：

- 只要执行 P2P 连接时没有明确设置信道，同步就会自动进行。
- AP 已在某信道上运行后明确设置另一 P2P-GO 信道，将导致 P2P 连接失败。
- AP 可在相同/不同信道上与 P2P 客户端同时工作。

使用“ap_start.sh”和“ap_stop”脚本可分别加载和取消 AP 角色。使用“p2p_start.sh”和“p2p_stop”脚本可分别加载和取消 P2P 角色。

备注

在单一角色场景中使用 p2p_start.sh 会影响电流消耗，对于单一角色场景请使用“sta_start.sh”和“sta_stop.sh”脚本。

3.1 错误/限制

AP 在与 P2P GO 同时工作时不能加载到 DFS 信道（在 hostapd 中实现了雷达检测机制，wpa_supplicant 中未实现）。

4 多 BSSID (mBSSID)

WiLink8 器件可在同一实体器件上支持两个虚拟 AP。这两个虚拟 AP 在同一射频信道上工作。在 mBSSID 模式下，WLAN 元件将创建两个 SoftAP 网络接口。

- 每个 AP 有独立的 BSSID 和 SSID。
- 每个 AP 将分别传输信标。
- 每个虚拟 AP 将分别响应探测请求。
- 两个 BSS 总共支持多达 10 个流量链接（10 个互联 STA）。
- WiFi 隐私/加密
 - 两个 AP 均可分别启用或禁用隐私，无论另一个 AP 如何设置。
 - 支持各种加密类型的所有组合。

4.1 设置和配置

mBSSID 解决方案在两个 BSS 上均运行一个“hostapd”实例。hostapd 的配置参数位于配置文件“hostapd.conf”中。此文件已提供，并可根据产品要求进行修改。此外，还提供了两个“udhcp.conf”文件（见下），用于控制每个 BSS 中提供的 IP 地址。

请确保以下文件位于 /usr/share/wl18xx/

- hostapd.conf
- udhcp.conf
- udhcp2.conf

在 hostapd.conf 文件末尾添加必要属性（这样可用与第一个 AP 相同的属性配置第二个 AP，SSID 除外。bss=wlan2 以下的所有参数用于配置第二个 bss）：

```
bss=wlan2
ssid=BSS2
use_driver_iface_addr=1
```

备注

在此文件中，可以添加其他属性，例如安全性和 WPS，与为第一个 BSS 添加属性的方法相同。

```
wpa=2
wpa_key_mgmt=WPA-PSK
wpa_pairwise=CCMP
wpa_passphrase=12345678
```

```
wps_independent=1
wps_state=2
device_name=Sitara
device_type=1-0050F204-1
manufacturer=TexasInstruments
model_name=TI_Connectivity_module
model_number=w118xx
serial_number=12345
uuid=12345678-9abc-def0-1234-56789abcdef1
eap_server=1
config_methods=virtual_display virtual_push_button push_button keypad
```

备注

不支持添加不同信道和频带。

每个 AP 应配置为不同子网的 IP。udhcp.conf 文件中 IP 租借区块的开始和结束应进行相应更新，包括“opt router”（默认网关）参数。

例如：如果 AP1 配置为 IP 192.168.43.1，AP2 配置为 IP 192.168.53.1（掩码均为 255.255.255.0）则开始、结束和 opt router 应如下所示：

表 4-1. IP 地址范围

属性/文件	udhcp.conf	udhcp2.conf
start	192.168.43.20	192.168.53.20
end	192.168.43.254	192.168.53.254
opt router	192.168.43.1	192.168.53.1

4.2 用户指南和示例

在 mBSSID 模式期间，每个 AP 都有自己的网络接口。第一个 AP 网络接口被称作“wlan1”，第二个 AP 网络接口通常被称作“wlan2”。请注意，这些名称必须在 hostapd.conf 文件中进行相应配置（请参阅上文的设置和配置）。

每个 AP 都分配了唯一 MAC 地址。此 MAC 地址来自芯片的基本 MAC 地址，并依次递增。

例如，如果基本 MAC 是 E0:C7:9D:2D:AA:CC，则为第一个 AP 分配 E0:C7:9D:2D:AA:CD，并为第二个 AP 分配 E0:C7:9D:2D:AA:CE。

以下脚本用于启动/停止两个 BSS：

- 启动两个 AP：ap_start.sh
- 停止两个 AP：ap_stop.sh

备注

mBSSID 模式需要静态启动或停止。

5 接入点 (AP) 增强型低功耗 (ELP) 模式

与传统的 AP 不同，不能假定实现软件 AP 特性的便携式设备是与电源连接的，而且软件 AP 角色的要求比传统 STA 角色要高得多。因此必然要求减少设备功耗，同时在不显著影响性能的情况下履行软件 AP 角色的职责。这就要求软件 AP 具有独立的节能机制，无需向 STA 发送显式消息即可减少此设备的功耗。

5.1 设置和配置

AP ELP 模式需要修改 wl18xx-conf ini 文件的 AP ELP 属性。目前仅支持以下属性：

wl18xx.ap_sleep.idle_duty_cycle：处于闲置模式时，AP 唤醒部分的占空比 (0-99%)。默认为 0x00，表示 AP 始终处于唤醒状态。

目前不支持以下 AP ELP 属性：

wl18xx.ap_sleep.connected_duty_cycle：存在连接站点时，AP 唤醒部分的占空比 (0-99%)。默认为 0x00，表示 AP 始终处于唤醒状态。**wl18xx.ap_sleep.max_stations_thresh**：在禁用此特性之前，允许连接到 AP 的站点数量上限 (255：无限制)。默认为 0x04。**wl18xx.ap_sleep.idle_conn_thresh**：流量停止后启用 AP 睡眠状态超时。单位为 0.1 秒。默认为 0x08。

改变 default.conf 文件中的值并从头创建 bin 文件可完成此操作。

```
$ cd /usr/sbin/wlconf
$ vi default.conf
```

将 **wl18xx.ap_sleep** 属性设为适当值，退出并保存。

```
$ rm wlconf-18xx-conf.bin
$ ./wlconf -D (a new wlconf-18xx-conf.bin will be created)
$ cp wlconf-18xx-conf.bin <conf-bin> (Where <conf-bin> is usually /lib/firmware/ti-connectivity/
wl18xx-conf.bin)
```

或者也可以通过以下方式编辑现有 bin 文件来完成此操作：

```
$ cd /lib/firmware/ti-connectivity
$ ./wlconf -i wl18xx_conf_bin -o wl18xx_conf_bin --set wl18xx.ap_sleep.idle_duty_cycle=0x14
```

这样可将 AP 设为在闲置模式下有 20% 的时间处于睡眠状态。

5.2 用户指南和示例

设备根据以上介绍进行配置后，将在 AP ELP 模式下工作。无需额外操作。

5.3 错误/限制

使用 *Bluetooth*® 时会禁用此特性。

多角色场景不支持此特性

6 Wilink8 的 WLAN 唤醒 (WoWLAN) 特性

无线唤醒 (WoW) 特性允许系统进入低功耗状态，而无线 NIC 保持有效，为主机执行各种操作，例如，连接到 AP 或搜索网络。如需更多信息，请参阅 kernel.org。

本节讨论了如何启用 WL18xx 器件和 AM335x 主机处理器的“暂停恢复” WoW 模式。也可在其他主机处理器上对硬件和软件进行如下修改来启用 WoW。

6.1 运行模式

在暂停状态下，WL18xx 芯片仍会连接到 Wi-Fi 网络，等待 Wi-Fi 媒体的特定“魔法”帧。在接收到该帧之后芯片才会唤醒主机。支持两种运行模式：

- **暂停状态**：芯片将处于关闭模式，主机使 WLAN_Enable 信号保持为 OFF，以禁用 WLAN 部分，在该模式下 WL18xx 消耗的电流最少。
- **WOWLAN 状态**：主机关闭但 WLAN 芯片处于激活状态 (WLAN 启用引脚保持有效)。这使得 WL18xx 固件在暂停期间保持有效，如果系统配置为允许 WLAN 中断唤醒主机，则可在 WL18xx 固件接收到特定数据包时触发系统恢复。

在进入暂停模式之前，用户需要声明哪种事件可触发唤醒。在本例中，触发事件设为“any”（任意），表示任何事件（控制台活动、魔法帧等）都将唤醒电路板：

从暂停模式恢复有两种选项：

- 串行控制台活动或配置的其他唤醒源（键盘、触摸屏）将触发恢复。
- 从其他站点向处于暂停模式的站点发送魔法帧。

6.2 为 AM335x EVM 添加了“暂停/恢复” WoW 模式

若要在 PSP 中启用暂停/恢复，需要添加以下内容：

- **IRQ 线路** - 选择能够将主机从暂停状态唤醒的 GPIO，例如 GPIO0 组。
- **WLAN 启用** - 对于 SR1.0 器件，适用 AM335x Advisory 1.0.14 与 GMII_SEL 和 CPSW 相关的引脚控制寄存器：在 PD_PER 转换期间这些寄存器的上下文丢失。通常情况下，所选引脚在 AM335x 复位过程中会被拉低，例如本 EVM 上使用的 GPIO1_16 (GPMC_A0)。在上电期间需要将它拉低，以满足电源时序要求。但考虑到 WoWLAN，WL_EN 引脚必须被拉高。对于 AM335x SR1.0，GPIO1_16 的此种错误导致寄存器在暂停序列中总是恢复为默认值。因此，SR 1.0 器件必须使用不会受此影响的另一引脚，例如 GPIO3_16。
- **SDIO MMC 保持活动** - 使 SDIO 在从暂停模式恢复后能够避免重新枚举，因为假设 SDIO 在主机暂停模式期间持续处于激活模式。在 Linux 中，这可在器件树中使用“keep-power-in-suspend”指示符作为 MMC 配置的一部分来实现。

6.2.1 补丁描述

以下补丁对 am335x-evm 进行了以下硬件修改，用以支持 WoW 模式：

- **WLAN_IRQ**：从 GPIO3_17 迁移到 GPIO0_19（通过 WOWLAN 事件将 SA 从暂停状态唤醒）
- **WLAN_EN**：从 GPIO1_16 迁移到 GPIO3_16，其原因是 GPIO1_16 在暂停期间会丢失上下文 (AM335x SR1.0 Advisory 1.0.14)。
- **Enable MMC_PM_KEEP_POWER** 适用于 mmc2
- **激活 padmux 外部上拉** 主机暂停时使 WLAN_EN 引脚保持高电平

6.3 WoWLAN (WLAN 唤醒) 模式启用过程

以下步骤展示了启用 WoWLAN 的过程：

1. 将站点 (EVM) 连接到 AP。
2. 使 EVM 进入 WoWLAN 暂停模式（[任何] 或其他模式，示例如下）。

```
iw phy0 wowlan enable any
```

3. 进入暂停模式。

```
echo mem > /sys/power/state
```

4. 向设备发送 ping。ping 会使 EVM 从暂停状态唤醒。

6.3.1 命令和预期输出

WoW 使用 iw 命令。下面列出了与 WoW 相关的 iw 工具命令：

```
iw <phname> wowlan enable [any] [disconnect] [magic-packet] [gtk-rekey-failure] [eap-identity-  
request] [4way-handshake] [rfkill-release] [patterns <pattern>*] iw <iwname> wowlan disable iw  
<phname> wowlan show
```

进入 WoW 暂停模式

调用以下命令可进入 WoW 暂停模式：

```
echo mem > /sys/power/state
```

以下屏幕截图展示了主机如何进入暂停模式，WL18xx 也已暂停

```
[16277.797393] PM: Syncing filesystems ... done.  
[16277.929870] PM: Preparing system for mem sleep  
[16277.934509] Freezing user space processes ...(elapsed 0.02 seconds) done.  
[16277.963470] Freezing remaining freezable tasks ...(elapsed 0.02 seconds) done.  
[16277.994689] PM: Entering mem sleep  
[16277.998565] Suspending console(s) (use no_console_suspend to debug)  
[16278.242462] PM: suspend of devices complete after 236.060 msecs  
[16278.243408] PM: late suspend of devices complete after 0.945 msecs  
[16279.748199] Successfully put all powerdomains to target state
```

从暂停模式恢复

系统可通过 ping 从暂停模式恢复。以下屏幕截图展示了主机如何从暂停模式恢复，WL18xx 也重新连接到 Wi-Fi 网络（通过 WPA 请求），流量也已恢复。

```
[16279.748840] PM: early resume of devices complete after 0.518 msecs  
[16279.749237] platform iva.0: omap_voltage_scale: Already at the requested rate 800000000  
[16279.749237] platform mpu.0: omap_voltage_scale: Already at the requested rate 1000000000  
[16280.107391] PM: resume of devices complete after 358.428 msecs  
[16280.157989] PM: Finishing wakeup.  
[16280.161499] Restarting tasks ... done.
```


6.3.2 Rx 过滤器配置

启用后，芯片固件会在每个传入的数据包中搜索 WoWLAN 模式。检测到模式后，芯片将主机唤醒。可使用 iw 实用工具配置 WoWLAN 模式。这些模式始于 802.3 (以太网) 报头以及正确的源/目标 MAC (不是无线传输的实际 802.11 报头)。

在暂停或唤醒状态下均可启用 Rx 过滤器。

表 6-1. 802.3 MAC 头模式

参数	字节数
目标 MAC 地址	6
源 MAC 地址	6
EtherType	2
有效载荷	46-1500
CRC	4

有效载荷可以是任何上层协议。出于过滤目的，我们将仅显示 IP 报头。这是因为过滤其他协议不需要解析 MAC 有效载荷。如果为 ARP 或 EAPOL 配置过滤器，则不需要在 ARP/EAPOL 报头中配置模式，因为要全部传递。

以下是启用了 IP 过滤的数据包的模式/模板。

```
AA:AA:AA:AA:AA:AA:BB:BB:BB:BB:BB:BB:CC:CC:DD:-:-:-:-:-:-:-:-:EE:-:-:FF:FF:FF:FF:GG:GG:GG:GG:HH:HH:II:II
```

- A** 以太网目标地址
- B** 以太网源地址
- C** 以太网协议类型
- D** IP 报头 VER+Hlen，使用 : 0x45 (4 - 用于 ver 4，5 用于 len 20)
- E** IP 协议
- F** IP 源地址 (192.168.0.4: C0:A8:00:2C)
- G** IP 目标地址 (192.168.0.4: C0:A8:00:2C)
- H** 源端口 (1024: 04:00)
- I** 目标端口 (1024: 04:00)

备注

- 在任何时间只能激活 7 种模式
- 上限为 81 个字节
- 每种模式不超过 7 个段，如果模式跨越层报头，计为两个段 (每个段的开销为 4 字节)。

6.3.2.1 Rx 过滤器配置示例

表 6-2 展示了一些用例场景，以及如何设置数据包过滤的示例：

表 6-2. Rx 过滤器配置示例

用例	iw 命令
根据发送到 MAC 00:44:44:44:44:44 的任何数据包唤醒	iw phy0 wowlan enable patterns 00:44:44:44:44:44
根据发送到 MAC 00:44:44:44:44:44 的任何 TCP 数据包唤醒 IP 192.168.1.4 来自 TCP 端口 5001	iw phy0 wowlan enable patterns 00:44:44:44:44:44:-:-:-:-:-:08:00:45:-:-:-:-:-:06:-:-:-:-:-:c0:a8:01: 04:13:89
根据到达 MAC 00:44:44:44:44:44 的任何 EAPOL 流量唤醒	iw phy0 wowlan enable patterns 00:44:44:44:44:44:-:-:-:-:-:88:8e
根据任何单播、广播和多播唤醒	iw phy0 wowlan enable patterns 00:44:44:44:44:44 ff:ff:ff:ff:ff:ff 01:00:5e
设置多种模式	iw phy0 wowlan enable patterns PATTERN1 PATTERN2 PATTERN3
查看配置的模式	iw phy0 wowlan show

备注

有关 iw 命令的注意事项：

- “:-” 指示不匹配的通配符字节
- phy0 标识物理设备。最好使用 “iw list | grep wiphy” 验证设备

有关 iw 命令的更多详情，请参阅 <https://wireless.wiki.kernel.org/en/users/documentation/iw>。

备注

您应配置所有单播模式，以接收发送给您的流量。对于更具体的模式，请务必包含一个捕获 EAPOL 的模式，因为它们用于 AP 组密钥更新（定期进行）。如果不包含该模式，在密钥更新事件发生后，站点将无法接收多播和广播流量，因为没有解密的正确密钥（这一切只与密钥会更新的 WPA2 相关）。

6.4 WoWLAN - 魔术包

魔术包是一个广播帧，其有效载荷的 6 字节均为 255（十六进制 FF FF FF FF FF FF），接下来将目标计算机的 48 位 MAC 地址重复十六次，总计 102 字节。

由于魔术包只扫描以上字符串，完整的协议栈实际上并不会对它进行解析，它可作为任何网络和传输层协议发送，不过它通常作为 UDP 数据报发送到端口 7 或 9，或直接作为 EtherType 0x0842 通过以太网发送。

wl18xx 固件不支持扫描整个以太网帧寻找魔术包。此外，如前所述，模式的大小上限为 81 字节。

但可以定义模式来解剖魔术包的实现，采用的方法是比较实际魔术包的子集。下方显示了示例：

利用 EtherType = 0x0842 或以下 UDP 端口号唤醒魔术包：9

```
iw phy0 wowlan enable patterns
01:02:03:04:05:06:-:-:-:-:-:08:00:45:-:-:-:-:-:11:-:-:-:-:-:C0:A8:01:04:-:-:-00:09:-:-:-:-:-:
ff:ff:ff:ff:ff:ff:01:02:03:04:05:06:01:02:03:04:05:06
01:02:03:04:05:06:-:-:-:-:-:08:42:ff:ff:ff:ff:ff:ff:01:02:03:04:05:06:01:02:03:04:05:06
```

其中，

01:02:03:04:05:06 -> 目标 MAC ID C0:A8:01:04 -> 目标 IP 地址 (192.168.1.4)

设置上述模式后，实用工具 “etherwake” 或 “wakeonlan” (Linux) 或 “Wol Wake on Lan Wan” (Android) 可用于唤醒 WL18xx 主机。

在远程 (Ubuntu) 计算机上运行的命令：

```
# etherwake -iwlan2 -D 01:02:03:04:05:06
OR
# wakeonlan -i 192.168.1.4 01:02:03:04:05:06
```

6.5 区块确认 (BA) 过滤器设置

默认情况下启用 WL18xx 固件，以向主机发送任何 BA 动作帧，即使主机暂停，数据过滤器也不会阻止它。若要停止发送 BA 请求，可在 WLCONF 中配置 BA 过滤器（暂停时将不接收 BA 动作帧）。

```
core.conn.suspend_rx_ba_activity = 0x01
```

有关 wlconf 的更多信息，请参阅“[WiLink™ 8 解决方案 WiLink8 - wlconf](#)”。

6.6 AM335x EVM 的硬件修改工程变更命令 (ECO) 请求

在 AM335x 上启用 WoWLAN 特性需要对硬件进行以下更改。

板名称：AM335x 15x15 基板修订版本：B1 项目名称：Sitara

Sitara 平台上的 WOW 支持（AM335x 15x15 基板）

表 6-3. AM335x WoWLAN 支持需要 ECO

组件	旧值	新值	说明
R221, R349, R328	0R 0402 RES-100004R	NC	移除电阻器（用于 WLAN_IRQ 修改）
R350, R347	0R 0402 RES-100004R	NC	移除电阻器（用于 WLAN_EN 修改）

连线/切断

- 涉及的元件：TP31、R328 - 在 TP31 和 R328 之间焊接导线 (WLAN_IRQ)
- 涉及的元件：R350、R347 - 在 R350 和 R347 之间焊接导线 (WLAN_EN)

7 WiLink8 暂停恢复模式

对于低功耗模式，WL18xx 芯片可以保持在暂停/关断模式下，该模式使用的电流最少，可延长电池寿命/最大限度地减少功耗。这些命令可在任何受支持的平台上运行，包括但不限于 AM335x、AM437x、AM57x 和 BeagleBone。WL18xx 模块还支持在这些平台上执行 WLAN 唤醒 (WoWLAN)。

关断模式下的暂停恢复 在暂停状态，WL18xx 芯片将处于关断模式，主机会将 WLAN_Enable 信号保持为 OFF，禁用 WLAN 部分，尽量减少 WL18xx 芯片的电流消耗。

如何进行暂停恢复 若要使处理器进入暂停状态，请执行以下操作：

将“debugfs”文件系统（文件类型 debugfs）放在“tmp”目录下：

```
mount -t debugfs debugfs /tmp/
```

使系统达到 OFF 状态：默认情况下，保持模式是可以尝试达到的最深睡眠状态。若要使电源域转换为关闭模式：

```
echo 1 > /tmp/pm_debug/enable_off_mode
```

若要进入暂停模式：

```
echo mem > /sys/power/state
```

若要从暂停模式恢复：

串行控制台活动或配置的其他唤醒源（键盘、触摸屏）将触发恢复

7.1 AM437x SDK 的暂停恢复示例

以下部分详细介绍了 AM437x SDK 的暂停/恢复示例。以下为通用过程：

1. 加载 wlcore 模块。
2. 启动站点模式。

3. 连接到 AP。
4. 进入暂停模式。
5. 一段时间后，从暂停模式恢复。

利用以下命令，首先加载 `wlcore` 模块，启动站点并连接到 AP：

```
cd /usr/share/wl18xx
./load_wlcore.sh
./sta_start.sh
./sta_connect-ex.sh OpenSSID
```

使用以下命令验证 EVM 是否已连接，输出将为 AP 设置和 RSSI：

```
iw wlan0 link
```

调用以下命令可进入“暂停”模式：

```
echo mem > /sys/power/state
```

预期结果如下所示，EVM 屏幕会关闭：

```
[ 344.900146] PM: Syncing filesystems ... done.
[ 347.304138] Freezing user space processes ...(elapsed 0.01 seconds) done.
[ 347.324859] Freezing remaining freezable tasks ...(elapsed 0.01 seconds) done.
[ 347.344879] Suspending console(s) (use no_console_suspend to debug)
[ 347.359039] wl12xx: down
[ 347.492248] PM: suspend of devices complete after 139.892 msecs
[ 347.494049] PM: late suspend of devices complete after 1.708 msecs
[ 354.081787] GFX domain entered low power state
[ 354.081848] Successfully transitioned all domains to low power state
[ 1599.585723] PM: Syncing filesystems ... done.
[ 1602.515808] Freezing user space processes ...(elapsed 0.01 seconds) done.
[ 1602.539642] Freezing remaining freezable tasks ...(elapsed 0.01 seconds) done.
[ 1602.559600] Suspending console(s) (use no_console_suspend to debug)
[ 1602.567535] wl12xx: down
[ 1602.706909] PM: suspend of devices complete after 139.770 msecs
[ 1602.708557] PM: late suspend of devices complete after 1.556 msecs
[ 1634.825073] GFX domain entered low power state
```

备注

最后 4 行只有在恢复后才会显示（因为暂停是在打印任务完成之前发生的）。

点击终端控制台后（为了从暂停模式恢复）将出现以下日志，EVM 屏幕将再次激活：

```
[ 354.290771] PM: early resume of devices complete after 208.465 msecs
[ 354.606018] net eth0: CPSW phy found : id is : 0x4dd074
[ 354.610931] wl12xx: state: 0
[ 355.069427] wl12xx: PHY firmware version: Rev 8.2.0.0.245
[ 355.069488] wl12xx: firmware booted (Rev 8.9.0.0.86)
[ 355.130493] wl12xx: Association completed.
[ 355.153411] PM: resume of devices complete after 861.928 msecs
[ 355.216430] Restarting tasks ... done.
```

在暂停之后或空闲一段时间后，根据电源域转换状态来检查实际上是否正在转换到关闭模式：

```
root@am335x-evm:~# cat /tmp/pm_debug/count
[ 611.886413] pwrldm state mismatch(cefuse_pwrldm) 3 != 0
cefuse_pwrldm (ON),OFF:0,RET:0,INA:0,ON:1,RET-LOGIC-OFF:0
mpu_pwrldm (ON),OFF:0,RET:0,INA:0,ON:1,RET-LOGIC-OFF:0,RET-MEMBANK1-OFF:0,RET-MEMBANK2-OFF:0,RET-MEMBANK3-OFF:0
per_pwrldm (ON),OFF:0,RET:0,INA:0,ON:1,RET-LOGIC-OFF:7,RET-MEMBANK1-OFF:0,RET-MEMBANK2-OFF:0,RET-MEMBANK3-OFF:0
rtc_pwrldm (OFF),OFF:1,RET:0,INA:0,ON:0,RET-LOGIC-OFF:0
gfx_pwrldm (ON),OFF:0,RET:0,INA:0,ON:1,RET-LOGIC-OFF:0,RET-MEMBANK1-OFF:0
l4_cefuse_clkdm->cefuse_pwrldm (0)
gfx_l4ls_gfx_clkdm->gfx_pwrldm (0)
gfx_l3_clkdm->gfx_pwrldm (1)
l4_rtc_clkdm->rtc_pwrldm (1)
```

```
mpu_clkdm->mpu_pwrnm (1)
l4_wkup_aon_clkdm->wkup_pwrnm (3)
l3_aon_clkdm->wkup_pwrnm (1)
l4_wkup_clkdm->wkup_pwrnm (5)
clk_24mhz_clkdm->per_pwrnm (1)
lcdc_clkdm->per_pwrnm (1)
cpsw_125mhz_clkdm->per_pwrnm (2)
pruss_ocp_clkdm->per_pwrnm (0)
ocpwp_l3_clkdm->per_pwrnm (0)
l4hs_clkdm->per_pwrnm (1)
l3_clkdm->per_pwrnm (10)
l4fw_clkdm->per_pwrnm (2)
l3s_clkdm->per_pwrnm (4)
l41s_clkdm->per_pwrnm (19)
```

每个暂停恢复周期计数器会增加：per_pwrnm (ON),OFF:0,RET:0,INA:0,ON:1,RET-LOGIC-OFF:7,RET-MEM BANK1-OFF:0,RET-MEM BANK2-OFF:0,RET-MEM BANK3-OFF:0。

8 接入点 (AP) 动态频率选择 (DFS) 主设备支持

从全球范围来看，雷达系统大部分使用 5GHz 频带的频率。该频带 (或部分) 分配给了无许可证的 WLAN 设备。重复使用这一频带时，要求采用一种被称为动态频率选择 (DFS) 的方法。根据每个 DFS 标准中介绍的法规要求，需要 DFS 的系统一定要能够避免对雷达系统的干扰。

WiLink8.0 在所有三个监管领域均具有 DFS 主设备功能，这些领域为：TELEC、FCC 和 ETSI。有效领域是根据“Region”参数配置的。根据这些监管领域，WiLink8 AP 可以在 DFS 信道上工作。在 DFS 信道上工作之前，AP 能够执行信道可用性检查。在 DFS 信道上工作时，AP 能够检测雷达。检测到雷达后 AP 会移动至一个随机选择的信道。与 DFS 信道上的 AP 相连的 STA 将在 AP 信标中接收 CSA。

8.1 设置和配置

在 DFS 信道上运行 AP 是通过 TI 提供的“user-scripts” (用户脚本) 完成的。这些脚本可用作运行 AP DFS 主设备特性的参考，用法如下所示：

备注

这些脚本也用作在非 DFS 信道上运行 AP。

hostapd 的配置参数位于配置文件 “hostapd.conf” 中。此文件已由 TI 提供，并可根据产品要求进行修改。但有一些参数需要修改，以支持 AP DFS 主设备（表 8-1 中提及），必须设置这些参数，该特性才能正常发挥作用。

表 8-1. 支持 DFS 的 Hostapd.conf 文件参数

属性/文件	hostapd.conf	说明
hw_mode	a	DFS 仅在 A 频带运行
country_code	选择国家/地区	例如：US
信道	根据所选的硬件模式和地区代码挑选信道	所选国家/地区必须支持该信道
ieee802.11d	1	
ieee802.11h	1	
basic_rates	60 120 240	默认值为：10、20、55、110、60、120、240。在 5G 频带运行需要进行更改。

8.2 用户指南和示例

以下是在 DFS 信道上运行的脚本和流程：

1. 启动 AP：ap_start.sh。
2. 等待 60 秒，完成信道可用性检查并启用 AP。
3. 如果在 AP 信道上检测到雷达：
 - a. 会随机选择新信道
 - b. AP 执行信道切换

雷达仿真：有一个调试选项可模拟特定 DFS 信道上的雷达检测：示例：

```
echo 60 > /sys/kernel/debug/ieee80211/phy0/wlcore/wl18xx/radar_detection
```

更改非占用周期 (NOP)：若要编辑 “NOP” 时间，加载 cfg80211 模块时使用所需参数：dfs_nop_time_ms - 非占用周期，单位为 ms，默认为 1,800,000。dfs_cac_time_ms - CAC 时间，单位为 ms，默认为 60,000。示例：

```
modprobe cfg80211.ko dfs_nop_time_ms=60000 dfs_cac_time_ms=10000
```

备注

需要重新加载驱动程序。

启用/禁用测试模式：允许测试雷达探测功能，无需切换信道。

```
echo 8 > /proc/sys/kernel/printk Disable: echo 0 > /sys/kernel/debug/ieee80211/phy0/wlcore/wl18xx/radar_debug_mode Enable: echo 1 > /sys/kernel/debug/ieee80211/phy0/wlcore/wl18xx/radar_debug_mode
```

8.3 错误/限制

DFS 主设备不支持多角色。

9 站点模式 - 介绍了使用 iw 命令的备选方法

“[WiLink8 Linux Wi-Fi 驱动程序 R8.8 版用户指南](#)”中介绍了如何使用 WiLink8 驱动程序提供的脚本来启动站点模式。以下部分使用 iw 命令启动站点模式，并在不使用 WiLink8 驱动程序脚本的情况下连接到接入点。



图 9-1. Wi-Fi 站点硬件设置

以下是需要遵循的一般步骤：

1. 检查 wlan0 接口是否已可用。
2. 如果接口不可用，启动该接口。
3. 扫描可用的接入点 (AP)。选择接入点并与其连接

9.1 第 1 步 - 检查 wlan0 接口是否已在运行

可使用“ifconfig”命令完成这一步。该命令的典型输出如下所示：

```
root@am335x-evm:~# ifconfig wlan0
wlan0      Link encap:Ethernet  HWaddr 78:A5:04:26:97:3D
           inet addr:192.168.1.4  Bcast:192.168.1.255  Mask:255.255.255.0
           UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
           RX packets:3 errors:0 dropped:0 overruns:0 frame:0
           TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:1240 (1.2 KiB)  TX bytes:1552 (1.5 KiB)
```

如果 wlan0 出现在列表中则固件已加载，可跳过第 2 步。

9.2 第 2 步 - 启动 wlan0 接口 (如果未运行)

如果 wlan0 接口不可用，则可以使用以下命令使其运行。输出如下所示：

```
root@am335x-evm:~# ifconfig wlan0 up
wlc core: PHY firmware version: Rev 8.2.0.0.245
wlc core: firmware booted (Rev 8.9.0.0.86)
```

现在该接口已可正常运行。

9.3 第 3 步 - 将设备连接到可用的接入点

扫描该设备的可用接入点，典型输出如下所示：`(iw wlan0 scan)`。选择一个特定接入点，使用 iw 命令进行连接，典型输出如下所示：`(iw wlan0 connect <SSID>)`。也可使用 iw 命令验证链接：`(iw wlan0 link)`。最后，ping 测试可确保 WiLink8 设备连接到接入点。

```
root@am335x-evm:~# iw wlan0 scan | grep SSID
      SSID: IOP_035
      SSID: Demo_24
      SSID: abc
      SSID: mobile
root@am335x-evm:~# iw wlan0 connect abc
wlan0: authenticate with b8:a3:86:20:93:32
wlc core: is_sta: 1 old:0 new:1
wlan0: send_auth to b8:a3:86:20:93:32 (try 1/3)
wlan0: authenticated
wlc core: is_sta: 1 old:1 new:2
wlan0: associate with b8:a3:86:20:93:32 (try 1/3)
wlan0: RX AssocResp from b8:a3:86:20:93:32 (capab=0xc01 status=0 aid=1)
wlc core: is_sta: 1 old:2 new:3
wlc core: is_sta: 1 old:3 new:4
wlc core: Association completed.
```

```
wlan0: associated
cfg80211: Calling CRDA for country: RU
cfg80211: Regulatory domain changed to country: RU
cfg80211: (start_freq - end_freq @ bandwidth), (max_antenna_gain, max_eirp)
cfg80211: (2402000 KHz - 2482000 KHz @ 40000 KHz), (N/A, 2000 mBm)
cfg80211: (5735000 KHz - 5835000 KHz @ 20000 KHz), (N/A, 3000 mBm)
root@am335x-evm:~# iw wlan0 link
Connected to b8:a3:86:20:93:32 (on wlan0)
    SSID: algranati
    freq: 2437
    RX: 1360 bytes (12 packets)
    TX: 122 bytes (2 packets)
    signal: -60 dBm
    tx bitrate: 1.0 MBit/s
    bss flags:        short-slot-time
    dtim period:    1
    beacon int:     100
root@am335x-evm:~# udhcpc -i wlan0
udhcpc (v1.23.1) started
Sending discover...
Sending select for 192.168.1.4...
Lease of 192.168.1.4 obtained, lease time 172800
/etc/udhcpc.d/50default: Adding DNS 192.168.1.1

root@am335x-evm:~# ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1): 56 data bytes
64 bytes from 192.168.1.1: seq=0 ttl=64 time=1003.369 ms
64 bytes from 192.168.1.1: seq=1 ttl=64 time=2.526 ms
64 bytes from 192.168.1.1: seq=2 ttl=64 time=7.931 ms
```

10 参考文献

- 德州仪器 (TI) : [WiLink8 Linux Wi-Fi 驱动程序 R8.8 版用户指南](#)
- 德州仪器 (TI) : [WiLink™ 8 WLAN 特性用户指南](#)
- 德州仪器 (TI) : [WL18xx 5GHZ 天线分集](#)
- 德州仪器 (TI) : [通过 WLAN 进行精确时间同步](#)
- 德州仪器 (TI) : [WiLink™ 8 WLAN 软件 - 802.11s Mesh](#)

重要声明和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Copyright © 2022，德州仪器 (TI) 公司