
MiWi™ 软件设计指南

简介

MiWi 是 Microchip 的专有无线网络协议栈，旨在为低速率个人局域网（Low Rate Personal Area Network, LRPAN）提供支持。本指南介绍了基于 SAM 平台（SAMR21 和 SAMR30）中提供的 MiWi 协议实现的 MiWi 应用。

MiWi 支持以下三种网络拓扑：

- 点对点（Peer-to-Peer, P2P）
- 星型
- 网状

特性

在 Microchip 应用程序库（Microchip Libraries for Application, MLA）v2017-03-06 的 MiWi 协议 v5.30 中发布的早期版本 MiWi 的网状网络协议栈（版本 2.10 之前）支持基于库的网状网络协议栈。我们已对该协议栈重新进行了设计，有如下变化：

1. 优化当前 API 以提高简易性。
2. 重新设计 MiWi 网状网络，增加适用于下一代平台的功能。
3. 采用新的调试过程，以更安全的方式将设备加入网络。
4. 在 MiWi 网状网络中动态切换设备类型。
5. 为所有网络消息提供网络安全功能。
6. 支持无线升级，可升级网络中的所有节点。

目录

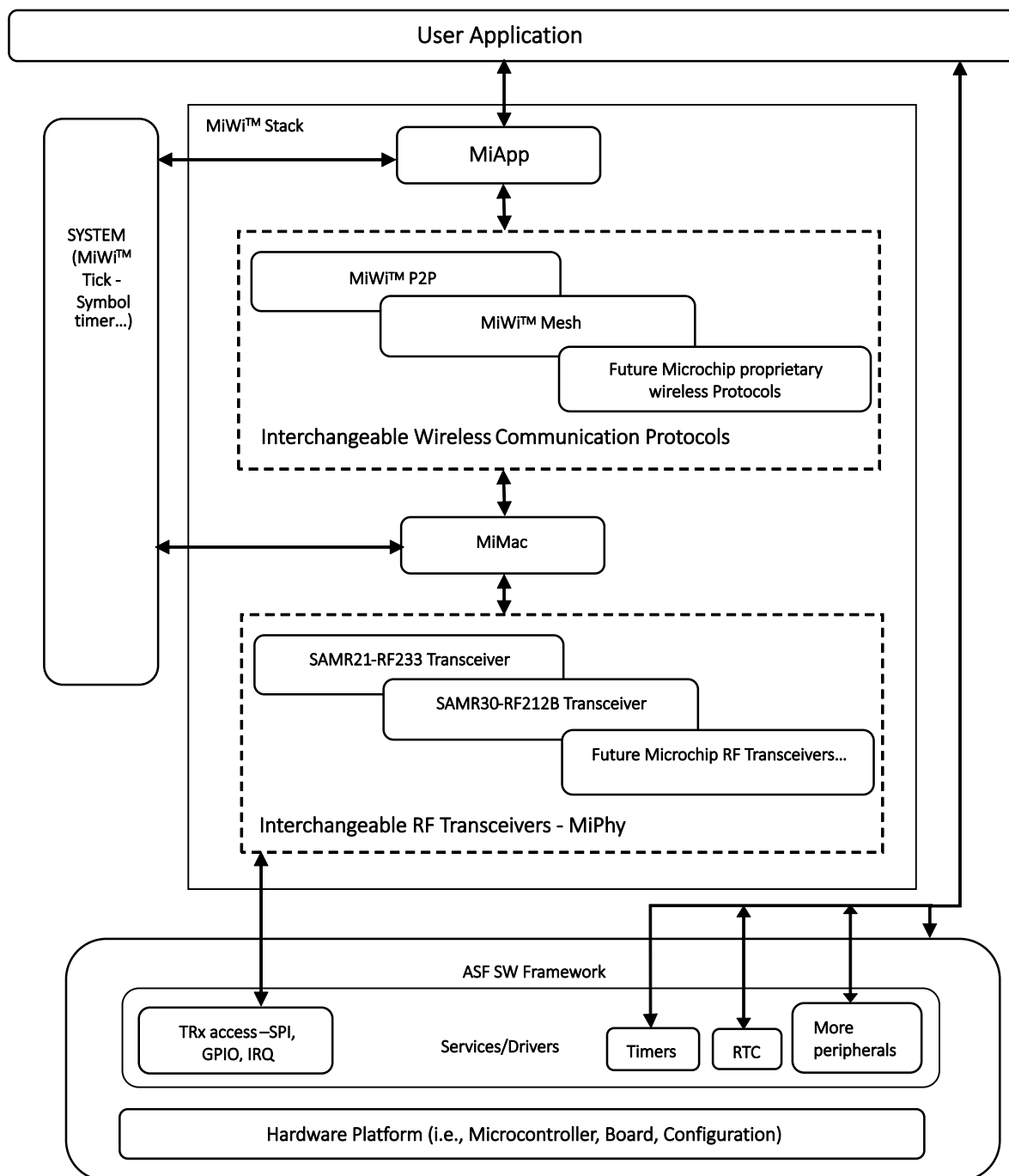
简介.....	1
特性.....	1
1. MiWi 架构.....	4
2. 支持的拓扑.....	5
2.1. 点对点（P2P）拓扑.....	5
2.2. 星型拓扑.....	6
2.3. 网状网络拓扑.....	6
3. MiWi P2P 拓扑.....	8
3.1. 网络寻址.....	8
3.2. 报文格式.....	8
3.3. 发送和接收.....	9
3.4. 握手.....	9
3.5. MiWi P2P 无线协议的定制 MAC 命令.....	10
3.6. 设备空闲时关闭射频.....	12
3.7. 主动扫描.....	13
3.8. 能量扫描.....	14
3.9. 频率捷变.....	14
4. MiWi 星型拓扑.....	15
4.1. MiWi 星型网络无线协议的特性.....	15
4.2. MiWi 星型网络无线协议的定制 MAC 命令.....	15
4.3. MiWi 星型网络无线协议中的握手.....	16
5. MiWi 网状拓扑.....	18
5.1. MiWi 网状网络设备类型.....	18
5.2. MiWi 网状网络帧格式.....	18
5.3. MAC 报头——帧控制字段.....	18
5.4. 网络报头.....	20
5.5. MiWi 网状网络——设备寻址机制.....	21
5.6. MiWi 网状网状——网络.....	22
5.7. MiWi 网状网络的宏.....	23
5.8. 针对宏的建议.....	29
5.9. 延长休眠终端设备的电池寿命.....	30
6. 网络冻结功能.....	31
6.1. 接口.....	31
6.2. 附加说明.....	31
6.3. 默认存储器布局.....	31
7. 休眠模式.....	32
7.1. 接口.....	32

8. 无线更新.....	33
8.1. OTAU 服务器.....	33
8.2. OTAU 客户机.....	33
8.3. OTAU 的域.....	33
8.4. PHY 模式操作.....	35
8.5. 用于 OTAU 的编译器开关.....	35
9. MiApp API.....	37
10. MiApp API 说明.....	39
10.1. MiApp_ProtocolInit.....	39
10.2. MiApp_Set.....	39
10.3. MiApp_StartConnection.....	40
10.4. MiApp_SearchConnection.....	40
10.5. MiApp_EstablishConnection.....	41
10.6. MiApp_RemoveConnection.....	41
10.7. MiApp_ConnectionMode.....	42
10.8. MiApp_SendData.....	42
10.9. MiApp_SubscribeDataIndicationCallback.....	43
10.10. MiApp_NoiseDetection.....	43
10.11. MiApp_TransceiverPowerState.....	44
10.12. MiApp_Get.....	44
10.13. MiApp_RoleUpgradeNotification_Subscribe.....	45
10.14. MiApp_Commissioning_AddNewDevice.....	45
10.15. MiApp_SubscribeReConnectionCallback.....	45
10.16. MiApp_ResetToFactoryNew.....	45
10.17. MiApp_ReadyToSleep.....	46
10.18. MiApp_ManuSpecSendData.....	46
10.19. MiApp_SubscribeManuSpecDataIndicationCallback.....	46
10.20. MiApp_IsConnected.....	47
10.21. MiApp_MeshGetNextHopAddr.....	47
11. 限制.....	48
12. 文档版本历史.....	49
Microchip 网站.....	50
产品变更通知服务.....	50
客户支持.....	50
Microchip 器件代码保护功能.....	50
法律声明.....	50
商标.....	51
质量管理体系.....	51
全球销售及服务网点.....	52

1. MiWi 架构

以下是基于高级软件框架（Advanced Software Framework, ASF）的 MiWi 协议架构，用户可通过“ASF 安装向导”获取所需组件、服务和驱动程序。更多详细信息，请参见“[Atmel 软件框架](#)”网页的“ASF 向导”部分。

图 1-1. MiWi™ 架构



2. 支持的拓扑

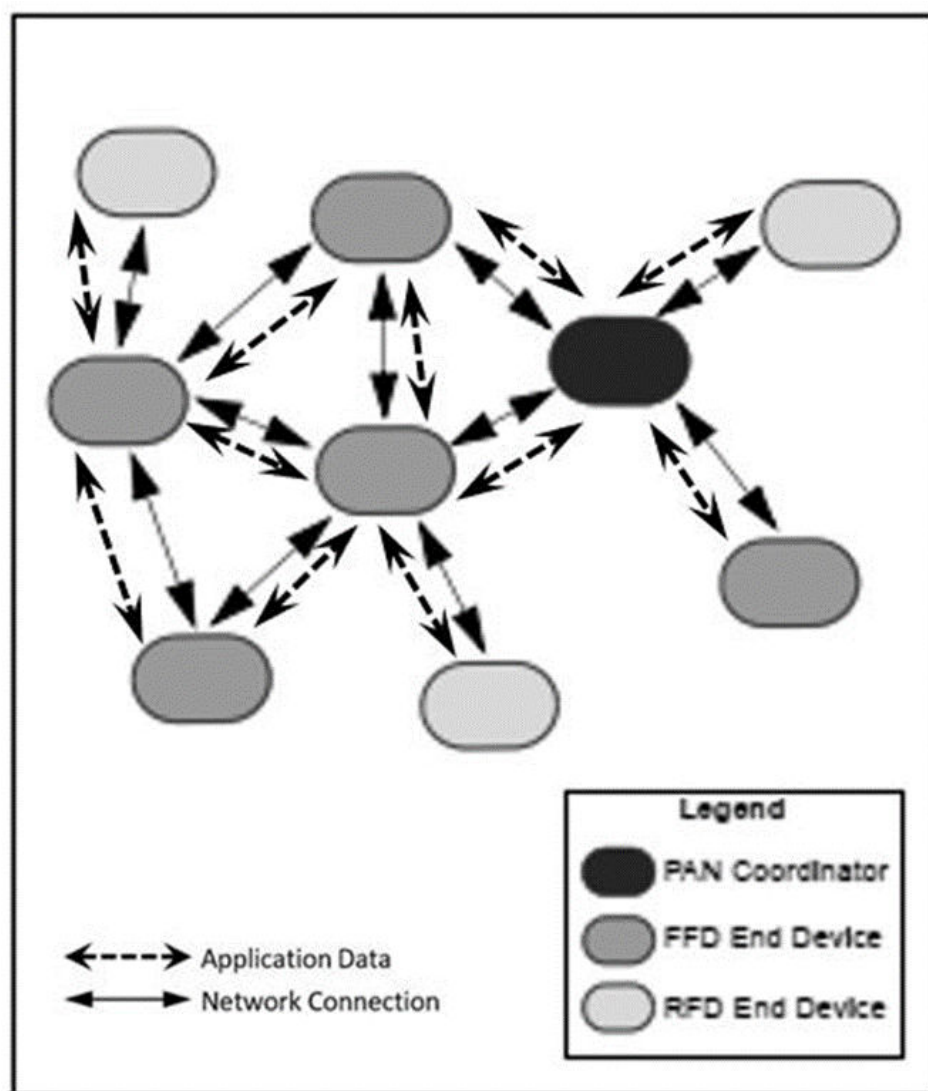
2.1 点对点（P2P）拓扑

下图给出了典型的 P2P 拓扑。从设备角色的角度来看，该拓扑包括一个 PAN 协调器，用于启动与终端设备的通信。不过，在加入网络时，终端设备不必与 PAN 协调器建立连接。

按功能类型划分，PAN 协调器是 FFD（全功能设备），终端设备可以是 FFD 或 RFD（简功能设备）。不过，在该拓扑中，FFD 类型的终端设备可以有多个连接。但是，每个 RFD 类型的终端设备只能连接一个 FFD，不能连接到其他 RFD。

如下图所示，应用数据只能发送到射频范围内的所有设备。

图 2-1. 点对点拓扑



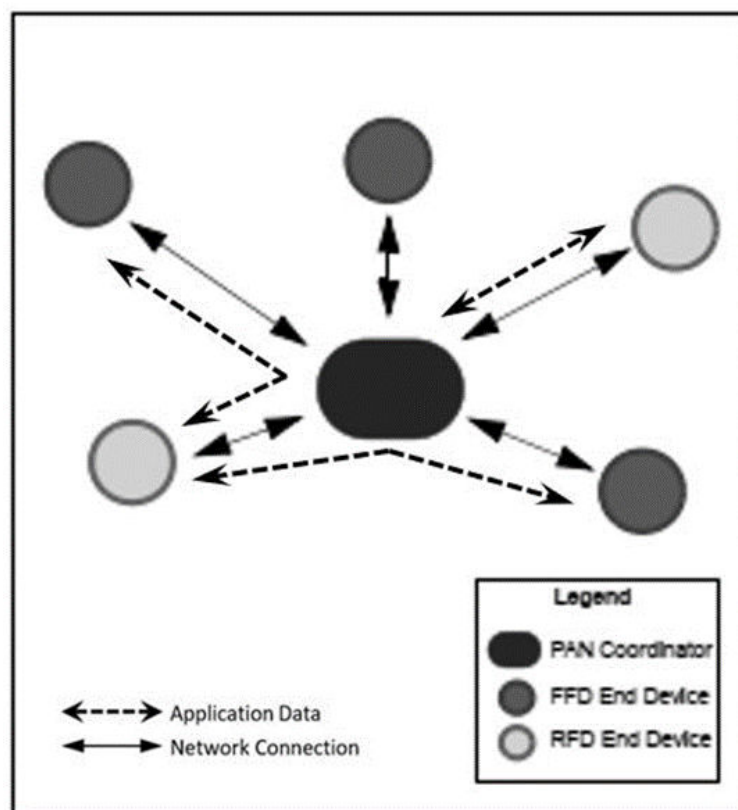
2.2 星型拓扑

下图给出了典型的星型拓扑。从设备角色的角度来看，该拓扑具有一个用于发起通信和接受来自其他设备的连接的个人局域网（PAN）协调器，以及多个加入通信的终端设备。终端设备只能与 PAN 协调器建立连接。

按功能类型划分，星型拓扑的 PAN 协调器是 FFD。终端设备可以是一直开启射频的 FFD，也可以是空闲时关闭射频的 RFD。无论其功能类型如何，终端设备都只能与 PAN 协调器通信。

如下图所示，应用数据可发送到网络中的任意设备，因为 PAN 协调器将数据从一个终端设备转发到另一个终端设备。

图 2-2. 星型拓扑



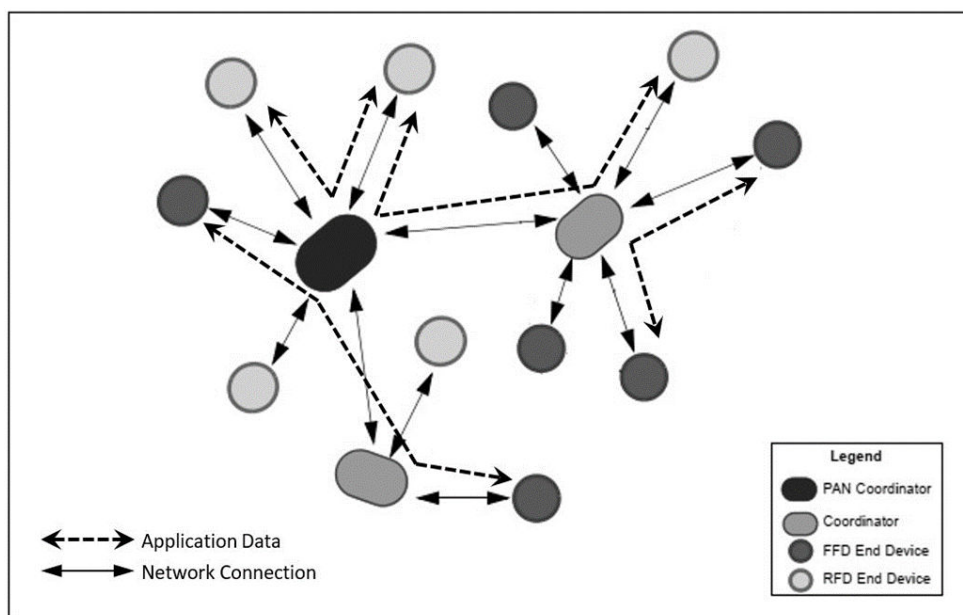
2.3 网状网络拓扑

下图给出了典型的网状网络拓扑。从设备角色的角度来看，此拓扑有一个用于启动网络的 PAN 协调器，所有协调器都会从 PAN 协调器请求一个短地址。加入网络的终端设备将加入附近的协调器。

按功能类型划分，PAN 协调器和其他协调器是 FFD，终端设备可以是 FFD 或 RFD。但是，每个 FFD/RFD 类型的终端设备只能连接一个协调器，不能连接到其他终端设备。

如下图所示，应用数据可通过网状路由从任意设备发送到任意设备。

图 2-3. 网状网络拓扑



3. MiWi P2P 拓扑

3.1 网络寻址

IEEE® 802.15.4 规范定义了两种寻址机制：

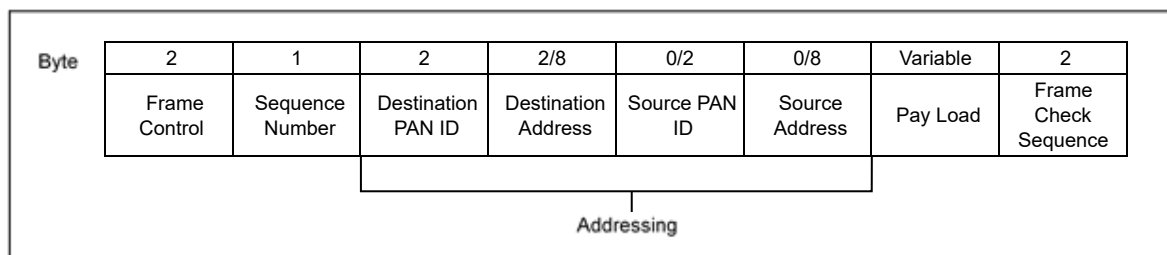
- 长地址或扩展组织唯一标识符（EUI）——一个 8 字节地址，该地址对全球范围内的每个设备都是唯一的。三个高字节由发布产品的公司从 IEEE 购买。五个低字节由设备制造商分配，每个设备的 EUI 必须是唯一的。8 字节唯一地址通常称为无线设备/节点的 MAC 地址，主要与节点硬件相关联。
- 短地址——一个 2 字节地址，当设备加入网络时，由父设备分配给设备。短地址在网络中必须唯一。

MiWi P2P 协议仅支持单路程段（one-hop）通信，因此它通过 EUI（即长地址）发送报文。仅当协议栈发送广播报文时才使用短地址。

3.2 报文格式

MiWi P2P 协议的报文格式是 IEEE 802.15.4 规范的报文格式的子集。下图说明了协议栈及其字段的数据包格式。

图 3-1. MiWi P2P 无线协议数据包格式



3.2.1 帧控制

下图说明了 2 字节帧控制字段的格式。

图 3-2. 帧控制

Bit	3	1	1	1	1	3	2	2	2
	Frame Type	Security Enabled	Frame Pending	Acknowledgement Request	Intra PAN	(Reserved)	Destination Address Mode	(Reserved)	Source Address Mode

3 位帧类型字段使用以下值定义数据包的类型：

- 数据帧 = 001
- 应答 = 010
- 命令帧 = 011

安全使能位指示当前数据包是否已加密。如果使用了加密，还会有一个额外的安全报头。更多信息，请参见 [2.3 网状网络拓扑](#)。

帧待处理位仅用于应答数据包，该数据包由 MRF24J40 射频硬件处理。该位指示在从 RFD 终端设备接收到数据请求数据包并应答后，后续是否还有其他数据包。

PAN 内部位指示报文是否在当前 PAN 内。如果该位设置为 1，寻址字段中的源 PAN ID 字段将被省略。在协议栈中，该位始终设置为 1，但也可以设置为 0 以使能 PAN 之间的通信。如有必要，可以在应用层将该位复位为 0。

目标地址模式可以是 16 位短地址模式 = 10，也可以是 64 位长地址模式

= 11

在 MiWi P2P 协议中，目标地址模式通常设置为长地址模式。短地址模式仅用于广播报文。对于广播报文，寻址字段中的目标地址字段固定为 0xFFFF。

MiWi P2P 协议的源地址模式只能是 64 位长地址模式。

3.2.2 序列号

序列号为 8 位长，以随机数开头，每次发送数据或命令包时都递增 1。应答数据包中使用序列号标识原始数据包。原始数据包和应答数据包的序列号必须相同。

3.2.3 目标 PAN ID

这是目标设备的 PAN 标识符。如果 PAN 标识符未知或不是必需的，则使用广播 PAN 标识符（0xFFFF）。

3.2.4 目标地址

目标地址可以是 64 位长地址或 16 位短地址。目标地址必须与帧控制字段中定义的目标地址模式一致。如果使用 16 位短地址，则该地址必须是广播地址 0xFFFF。

3.2.5 源 PAN ID

源 PAN 标识符是源设备的 PAN 标识符，必须与帧控制字段中的 PAN 内部位的定义匹配。仅当 PAN 内部位的值为 0 时，数据包中才存在源 PAN ID。

在当前 MiWi P2P 协议实现中，所有通信都是 PAN 内部通信。因此，所有数据包都没有源 PAN ID 字段。

但协议栈保留了应用层在 PAN 之间发送报文的能力。如果报文需要在 PAN 之间发送，则使用源 PAN ID。

3.2.6 源地址

源地址字段固定为使用源设备的 64 位扩展地址。

3.3 发送和接收

3.3.1 发送报文

有两种方法可发送报文：广播和单播。

广播数据包将射频范围内的所有设备作为其目标。IEEE 802.15.4 定义了一个作为广播地址的特定短地址，但是未定义长地址。因此，对于符合 IEEE 802.15.4 标准的收发器，只有在广播报文时 MiWi P2P 协议栈才使用短地址。

广播报文时没有任何应答。单播发送只有一个目标，使用长地址作为目标地址。MiWi P2P 协议要求应答所有单播报文。

如果接收设备中至少有一个在空闲时关闭了射频，则发送设备可将报文保存在 RAM 中，等待休眠设备被唤醒并请求报文。这种数据发送过程称为*间接报文传递*。

如果休眠设备无法获取间接报文，则报文将超时并被丢弃。通常，间接报文超时需要长于休眠设备的轮询间隔。

3.3.2 接收报文

在 MiWi P2P 协议中，只有接收报文的设备会通过射频收到通知。如果接收报文的设备在空闲时关闭了射频，则只能接收与之连接的设备发送的报文。

关闭射频的空闲设备如果要接收报文，必须向对等连接方发送数据请求命令。随后，它将获取间接报文（如果存在）。

在星型拓扑中，只有 PAN 协调器具备连接能力，所有终端设备（FFD/RFD）都连接到 PAN 协调器。因此，星型拓扑中的终端设备都有独立的连接。

3.4 握手

握手是一个加入网络的复杂过程。每个设备只能连接一个设备作为父节点，因此，初始握手实际上是选择父节点的过程。

选择父节点需要执行以下步骤：

1. 列出所有可能的父节点
2. 选择合适的设备作为其父节点

MiWi P2P 协议专为在星型网络和 P2P 通信拓扑中实现简单的直接连接而设计。某些 IEEE 802.15.4 要求会增加设计难度：

- 五步握手过程加上两个超时，需要更复杂的协议栈。
- 关联过程使用单连接通信，而不是点对点拓扑的多连接概念。

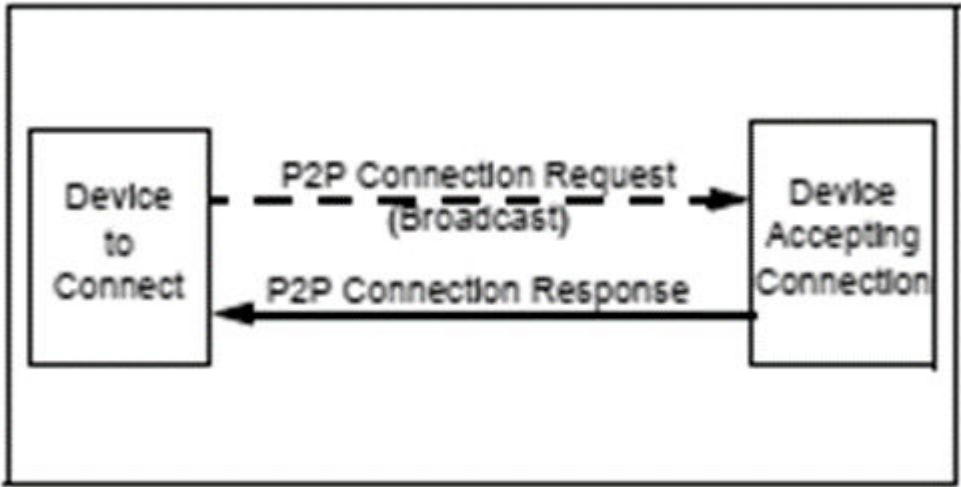
由于上述原因，MiWi P2P 协议使用自己的两步握手过程，如下图所示：

1. 发起设备发出 P2P 连接请求命令。
2. 射频范围内的所有设备均以可建立连接的 P2P 连接响应命令作为响应。

这是一个一对多的过程，可建立多个连接，在可能的情况下还会建立点对点拓扑。由于该握手过程使用 MAC 层命令，因此每次发送时都应使用 CSMA-CA，以降低数据包冲突的可能性。

RFD 可能会收到来自多个 FFD 的连接请求命令，但只能连接到一个 FFD。RFD 选择其接收到的第一个 P2P 连接响应的 FFD 作为对等连接方。

图 3-3. MiWi P2P 无线协议的握手过程



3.5 MiWi P2P 无线协议的定制 MAC 命令

MiWi P2P 协议利用定制的 MAC 命令来移除两个设备之间的连接，从而扩展了 IEEE 802.15.4 规范的功能。下表列出了该协议的所有定制 MAC 命令。

表 3-1. MiWi P2P 无线协议的定制 MAC 命令

命令标识符	命令名称	说明
0x81	P2P 连接请求	请求建立 P2P 连接。上电后通常通过广播方式请求 P2P 连接。也可通过单播方式请求单个连接。
0x82	P2P 连接移除请求	移除与另一终端设备的 P2P 连接。
0x83	P2P 数据请求	类似于 IEEE 802.15.4 规范的数据请求命令（0x04），如果本地节点的射频关闭，则从 P2P 连接的另一端请求数据。该命令供之前休眠的设备用来请求其他节点发送未收到的报文（间接报文传递）。
0x84	信道跳跃	请求将工作信道切换为其他信道。通常用于频率捷变功能。

..... (续)		
命令标识符	命令名称	说明
0x87	主动扫描请求	检查当前和可访问信道中的可用节点。
0x91	P2P 连接响应	对 P2P 连接请求的响应。也可以用于主动扫描过程。
0x92	P2P 连接移除响应	对 P2P 连接移除请求的响应。
0x97	主动扫描响应	响应将返回节点信息（包括信道、PAN ID 和节点 ID）。

注：
有关主动扫描请求和主动扫描响应的详细信息，请参见 3.7 主动扫描。

3.5.1 P2P 连接请求

可在上电后广播 P2P 连接请求（0x81），以和其他设备建立 P2P 连接。也可以将该请求单播到特定设备以建立单个连接。

当发送设备从另一端接收到 P2P 连接响应（0x91）时，将建立 P2P 连接。

P2P 连接请求定制命令还可以启动主动扫描，以确定附近有哪些设备可用。

如果发送 P2P 连接请求命令的目的是主动扫描，就不会附加功能信息和可选有效负载。接收设备根据是否有附加的功能信息和可选有效负载来确定该命令是请求建立连接还是主动扫描。

MiWi P2P 协议可使能或禁止某个设备与其他设备建立连接。禁止设备建立连接后，除以下情况外，所有新的 P2P 连接请求都会被丢弃：

- P2P 连接请求来自接收端已建立连接的设备。
- P2P 连接请求是主动扫描。

下图给出了 P2P 连接请求命令帧的格式。

图 3-4. P2P 连接请求命令格式

Octet	15/21	1	1	1 (Optional)	Various (Optional)
	MAC Header	Command Identifier (0x81)	Operating Channel	Capability Information	Optional payload to identify the node. It is not required for the stack, but may be useful for applications.

图 3-5. 功能信息格式

Bit	0	1	2	3	4-7
	Receiver ON when Idle	Request Data on Wake-up	Need Time Synchronization (Reserved)	Security Capable	(Reserved)

工作信道用于回避可能来自另一信道的次谐波的影响。它能够避免与不同信道上的设备建立虚假连接。图 3-4 中的功能信息字节使用如上图所示的格式。

P2P 连接请求的可选有效负载针对具体应用提供。设备可能需要使用额外的信息来标识自己，可以是惟一标识符，也可以是有关它在应用中所起功能的信息。如果使用可选有效负载，建立连接后就不需要使用额外的数据包来介绍或标识设备。协议栈中未使用可选有效负载。

3.5.2 P2P 连接移除请求

P2P 连接移除请求（0x82）发送到连接的另一端来移除 P2P 连接。下图给出了请求的格式。

图 3-6. P2P 连接移除请求格式

Octet	15/21	1
	MAC Header: Send to the other end of the P2P connection to cut the communication	Command Identifier (0x82)

3.5.3 数据请求

数据请求 (0x83) 命令与 IEEE 802.15.4 规范的数据请求 (0x04) 命令相同。下图给出了请求的格式。

如果 P2P 连接节点的一端能够在空闲时进入休眠状态，并且该节点可以在休眠状态下接收报文，则活动的一端必须将报文存储在其 RAM 中。当休眠设备被唤醒并请求报文时，活动端将传递报文。

如果应用中有这样的情况，则需要激活 ENABLE_INDIRECT_MESSAGE 功能。休眠节点被唤醒后，必须发送数据请求命令。

图 3-7. 数据请求格式

Octet	21	1
	MAC Header: Unicast from extended source address to extended destination address	Command Identifier (0x83 or 0x04)

3.5.4 信道跳跃

信道跳跃命令 (0x84) 请求目标设备将工作信道切换为另一个信道。下图给出了命令的格式。

该命令通常由频率捷变发起器发出，它决定何时切换信道以及选择哪个信道。

通常会广播此命令以通知所有设备（空闲时开启射频）切换信道。为了确保每个设备都接收到该报文，频率捷变发起器将执行 3 次广播，而所有 FFD 设备会执行重新广播。

当执行信道跳跃序列并且所有 FFD 跳到新信道时，RFD 必须执行重新同步以恢复与对应的 FFD 对等设备之间的连接。

图 3-8. 信道跳跃格式

Octet	15/21	1	1	1
	MAC Header: Broadcast or unicast from the Frequency Agility Starter	Command Identifier (0x84)	Current Operating Channel	Destination Channel to Jump to

3.5.5 P2P 连接响应

P2P 连接响应 (0x91) 命令用于响应 P2P 连接请求。下图给出了命令的格式。

P2P 连接响应命令可用于建立连接。此外，设备还可以使用此命令响应主动扫描，将自身标识为附近的活动设备。

图 3-9. P2P 连接响应格式

Octet	21	1	1	1 (Optional)	Various (Optional)
	MAC Header: Unicast from extended source address to extended destination address.	Command Identifier (0x91)	Status. 0x00 means successful. All other values are error codes.	Capability Information	Optional payload to identify the node. Not required for the stack, but possibly useful for applications.

3.5.6 P2P 连接移除响应

P2P 连接移除响应命令 (0x92) 用于响应 P2P 连接移除请求。它通知 P2P 连接的另一端已接收到 P2P 连接移除请求以及连接是否已移除。下图给出了命令的格式。

图 3-10. P2P 连接移除响应格式

Octet	21	1	1
	MAC Header: Unicast from extended source address to extended destination address	Command Identifier (0x92)	Status. • 0x00 means successful. • All other values are error codes

3.6 设备空闲时关闭射频

对于使用电池供电的设备，降低功耗至关重要。实现的方法是让设备在不发送数据时关闭其射频。MiWi P2P 协议具有使射频进入休眠模式以及稍后唤醒设备的功能。

要激活此功能，必须在文件 `miwi_config.h` 中定义“`ENABLE_SLEEP_FEATURE`”。

设备何时进入休眠模式可由具体应用决定。可能的触发条件包括：

- 射频空闲的时间长度
- 从已连接的 FFD 接收到数据包，请求设备进入休眠模式

唤醒设备的条件可由具体应用决定。可能的触发条件包括：

- 外部事件，如按下按钮
- 预定义的定时器超时

设备休眠期间，其对等设备可能需要发送报文。如果不发送报文，对等设备就不需要使能其他特性。

如果对等设备向休眠设备发送一条报文，必须将该报文存储到其易失性存储器中，直到休眠设备被唤醒并获取该报文。由于报文没有直接发送到休眠设备，因此该过程称为*间接报文传递*。

要传送间接报文，休眠节点的对等设备必须在 `miwi_config.h` 文件中定义“`ENABLE_INDIRECT_MESSAGE`”。

如果使能了间接报文传递，就必须指定易失性存储器中可以存储的间接报文数。最大报文大小取决于对等设备中的可用 RAM 存储空间以及连接到同一父 FFD 的 RFD 数量。

最大间接报文数由 `miwi_config_p2p.h` 文件中的“`INDIRECT_MESSAGE_SIZE`”定义。对于间接报文传递，还需要定义间接报文的超时周期。如果未定义超时周期且 RFD 设备处于非活动状态或不可见，则间接报文将永久保留在易失性存储器中。

间接报文超时周期由 `miwi_config_p2p.h` 文件中的“`INDIRECT_MESSAGE_TIMEOUT`”定义（以秒为单位）。

3.7 主动扫描

主动扫描是获取有关本地 PAN 的信息的过程。活动扫描用于确定：

- 设备的工作信道
- 设备在 PAN 中的信号强度
- 符合 IEEE 802.15.4 标准的收发器的 PAN 标识码

主动扫描特别适合本地设备没有预定义信道或 PAN ID 的情况。

在协议栈中，`ACTIVE_SCAN_RESULT_SIZE` 定义了主动扫描可获取的最大 PAN 数。

在主动扫描开始之前，先确定扫描的持续时间和需要扫描的信道。

扫描持续时间由 IEEE 802.15.4 规范定义，其时间长度（以符号为单位）由以下公式中给出的公式计算（一秒等于 62,500 个符号）。

公式 3-1. 扫描持续时间

$$\text{扫描周期} \equiv 960 * (2^{\text{ScanDuration}} + 1)$$

注：

扫描持续时间 = 用户指定的扫描输入参数。值为 1 到 14 范围内的整数。

扫描持续时间为 10 时，扫描周期为 61,500 个符号（约为 1 秒）。扫描持续时间为 9 时，扫描周期约为半秒。

扫描信道由位图定义，每个信道编号用双字中对应的位来表示。信道 11 即 b'0000 0000 0000 0000 0000 1000 0000 0000。

2.4 GHz 频谱中支持的信道 11 至 26 可以表示为 b'0000 0111 1111 1111 1111 1000 0000 0000 或 0x07FFF800。

当主动扫描广播 P2P 连接请求命令时，它要求射频范围内的所有设备均通过 P2P 连接响应命令进行应答。主动扫描仅确定附近有哪些 PAN 可用，但无法确定有多少个可建立新连接的设备。每个设备都会响应扫描，包括不支持新连接的设备。

要调用主动扫描功能，必须在 `miwi_config.h` 文件中定义“`ENABLE_ACTIVE_SCAN`”。

3.8 能量扫描

每个频带上可能有多个信道，但一个 PAN 只能在一个信道上工作。最好使用能量或噪声最小的信道。

能量扫描用于扫描所有可用信道以及确定噪声最小的信道。

扫描持续时间和要扫描的信道是在执行能量扫描之前确定的。

扫描持续时间由 IEEE 802.15.4 规范定义，其时间长度（以符号为单位）由公式 3-1 中给出的公式计算。有关测量的更多信息，请参见 3.7 主动扫描。

扫描完成后，将返回噪声最小的信道标识符。要激活能量扫描功能，必须在 miwi_config.h 文件中定义“ENABLE_ED_SCAN”。

3.9 频率捷变

频率捷变允许 MiWi P2P 协议 PAN 根据工作条件的需要切换到其他信道。

实现该功能后，受影响的设备将充当以下两个角色之一：

- 频率捷变发起器——这些设备负责确定是否需要信道跳跃，以及跳跃到哪个新信道。
- 频率捷变跟随器——这些设备在得到指令后切换到其他信道。

3.9.1 频率捷变发起器

每个 PAN 可以有一个或多个设备作为频率捷变发起器。发起器必须是 FFD。每个发起器必须使能量扫描功能以确定要跳跃的最佳信道。发起器向 PAN 上的其他设备广播信道跳跃命令。

3.9.2 频率捷变跟随器

频率捷变跟随器可以是 FFD 或 RFD 设备。

FFD 通过以下步骤之一实现信道跳跃：

- 从发起器接收信道跳跃命令。
- 在数据发送连续失败时重新同步连接。

RFD 设备使用重新同步方法（即，在通信失败时重新连接 PAN）实现信道跳跃。

3.9.3 使能频率捷变功能

何时执行频率捷变操作由应用决定。频率捷变通常由 CCA 失败或未收到应答引起的连续发送失败触发。

要激活频率捷变功能，必须在 miwi_config.h 文件中定义“ENABLE_FREQUENCY_AGILITY”。

4. MiWi 星型拓扑

MiWi 星型网络协议是 Microchip 定义的 MiWi P2P 协议的扩展。从设备角色的角度来看，这种拓扑由一个用于发起通信和允许其他设备连接的 PAN 协调器，以及多个加入通信的终端设备组成。终端设备只能与 PAN 协调器建立连接。按功能类型划分，星型拓扑的 PAN 协调器是 FFD。终端设备可以是一直开启射频的 FFD，也可以是空闲时关闭射频的 RFD。无论其功能类型如何，终端设备都只能与 PAN 协调器通信。

4.1 MiWi 星型网络无线协议的特性

MiWi P2P 协议栈支持的星型拓扑提供点对点拓扑的所有功能。除此之外，它还能根据设备角色实现更多功能。

PAN 协调器支持以下功能：

- 与所有对等设备共享对等设备连接（FFD 和 RFD）信息
- 将数据包从一个终端设备转发到另一个终端设备
- 定期检查网络运行状况（可选）
- 将数据包发送到终端设备
- 处理休眠终端设备（RFD）的间接报文
- 通过软件应答来表明数据发送成功

FFD（终端设备）或 RFD（休眠终端设备）支持以下功能：

- 链路状态
- 离开网络命令

4.2 MiWi 星型网络无线协议的定制 MAC 命令

MiWi 星型网络协议利用定制的 MAC 命令来移除两个设备之间的连接，从而扩展了 IEEE 802.15.4 规范的功能。下表列出了该协议的所有定制 MAC 命令。

表 4-1. MiWi 星型网络无线协议的定制 Mac 命令

命令标识符	命令名称	说明
0xCC	具有有效负载的转发数据包命令	0XCC（1 字节）命令。目标终端设备地址（3 字节）。数据有效负载。
0xDA	对终端设备的软件应答	N/A
0x7A	链路状态	N/A
0x77	连接表广播命令	0x77（1 字节）命令。网络中的终端设备总数。

下图给出了星型网络协议中修改后的连接模式详细信息。

图 4-1. 星型网络协议中修改后的连接模式详细信息

Node Capacity	Sleep	Reserved	Security Enabled	Connection Mode	Reserved
Bit	0	1-2	3	4-5	6-7

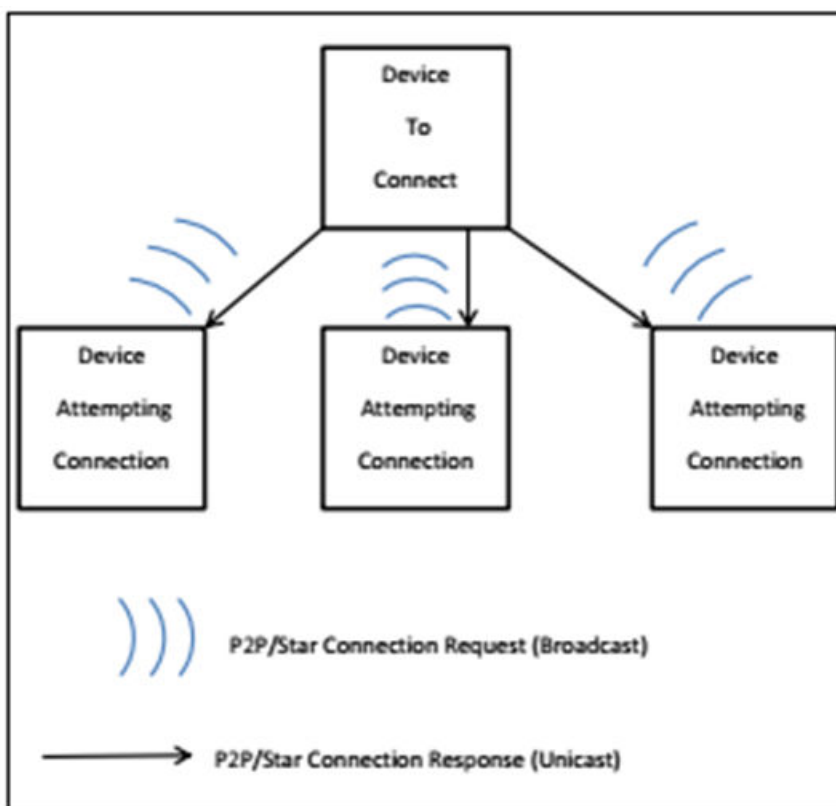
Value	Connection Mode
00	Enable All Connections
01	Enable Previous Connections
10	Enable Active Scan Response
11	Disable All Connections

4.3 MiWi 星型网络无线协议中的握手

4.3.1 MiWi 星型路由

如下图所示，MiWi 星型网络由两种类型的设备（PAN 协调器和终端设备——FFD 或 RFD）组成。PAN 协调器创建网络，而终端设备加入 PAN 协调器。PAN 协调器只需一个路程段就可以将报文发送到网络中的所有任何终端设备。如果一个终端设备要与附近或远处的另一个终端设备通信，则源终端设备必须先将数据包发送到 PAN 协调器，然后由 PAN 协调器将该数据包转发到目标终端设备（2 个路程段）。

图 4-2. MiWi 星型路由

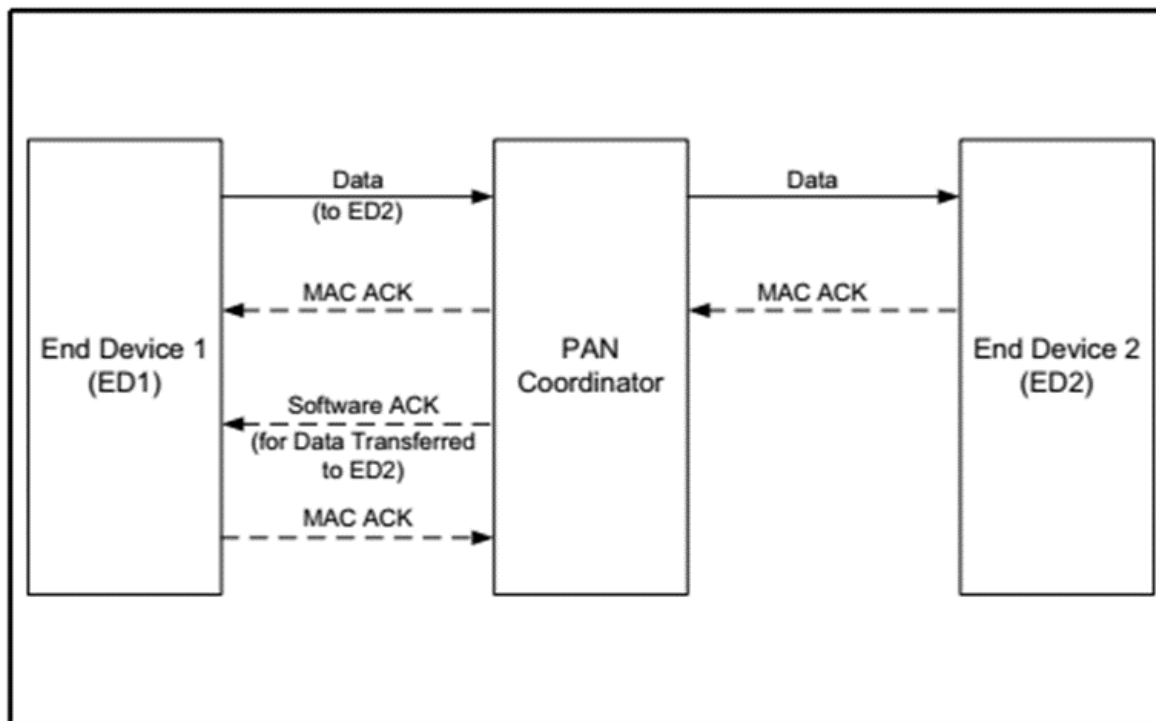


在 MiWi 星型网络中，PAN 协调器负责共享对等连接（终端设备地址）。这样，网络中的所有终端设备都可以知晓网络中其他设备的存在。当一个终端设备要向另一个终端设备发送报文时，源终端设备将目标终端设备的地址包含在有效负载中。源终端设备的有效负载由数据包类型（0xCC）、目标终端设备地址（仅前 3 个字节）和数据有效负载组成。当 PAN 协调器接收到该数据包时，该数据包指示其目标为另一个终端设备，因此，PAN 协调器会将该数据包转发到目标终端设备。

4.3.2 MiWi 星型数据传输

节点之间的连接请求和响应与 P2P 相似。但在 MiWi 星型网络中，PAN 协调器负责构成网络，连接终端设备，并为终端设备之间的通信（通过 PAN 协调器）提供支持。下图给出了 MiWi 星型网络中终端设备之间的数据传输简图。

图 4-3. MiWi 星型网络中终端设备之间的数据传输



5. MiWi 网状拓扑

5.1 MiWi 网状网络设备类型

MiWi 网状网络协议支持以下设备类型：

1. PAN 协调器
 - a. 启动网络
 - b. 分配和维护协调器及其终端设备地址
 - c. 用作路由帧的协调器
 - d. 通过调试控制可加入网络的设备
2. 协调器
 - a. 作为终端设备加入网络
 - b. 请求 PAN 协调器进行角色升级，以成为一个协调器
 - c. 支持在网络中路由帧
 - d. 存储来自 PAN 协调器的调试信息，只允许经调试的设备加入网络
 - e. 维护其终端设备及其地址
 - f. 维护休眠终端设备的数据
3. 终端设备
 - a. 通过可用协调器加入网络
 - b. 支持用于电池供电设备的 Rx-On 终端设备和休眠终端设备
 - c. 支持在 Rx-On 和休眠终端设备之间动态切换

5.2 MiWi 网状网络帧格式

MiWi 网状网络的网头和应用有效负载封装在标准 IEEE 802.15.4 数据帧有效负载内，但协议栈不遵守该标准。因此，MiWi 网状网络不接收和处理 IEEE 802.15.4 命令帧。下图说明了由 IEEE 802.15.4 MAC 报头、网络报头、应用有效负载、可选报文完整性代码（Message Integrity Code, MIC）和校验和（CRC）组成的通用帧格式。

图 5-1. 通用 MiWi 帧格式

2	1	2	2/8	2	0/2/8	1	1	1	0/2	0/2	0/2/8	0/5	Variable	0/4	2
Frame Control	Sequence number	Dest. PANID	Dest. Address	Source PANID	Source Address	Hops	Frame Control	Sequence number	Dest. PANID	Dest. Address	Source Address	Auxiliary Security Header	Payload	MIC	CRC
MAC Header						Network Header							Payload	Network Footer	

5.3 MAC 报头——帧控制字段

下图说明了 MAC 报头的帧控制字段。

图 5-2. MAC 报头——帧控制字段

Bits:0-2	3	4	5	6	7:9	10:11	12:13	14:15
Frame Type	Security Enabled	Frame Pending	Ack. Request	PAN ID Compression	Reserved	Dest. Address Mode	Frame Version	Source Address Mode

以下为 MiWi 网状网络中使用的固定 MAC 帧控制字段设置。下表列出了 MAC 报头的帧控制字段所用的设置。

表 5-1. MAC 帧控制字段设置

字段名称	设置
帧类型	数据
使能安全功能	False
帧待处理	如果待处理数据可用于休眠终端设备，则为 True，否则为 False
应答请求	单播帧为 True，广播帧为 False
PAN ID 压缩	True
目标寻址模式	0 表示无地址字段，2 表示 16 位短地址，3 表示 64 位扩展地址
帧版本	0
源寻址模式	0 表示无地址字段，2 表示 16 位短地址，3 表示 64 位扩展地址

5.3.1 帧类型

帧类型字段的长度为 3 位，应设置为 001——数据。

5.3.2 使能安全功能

使能安全功能字段的长度为 1 位，如果帧受 MAC 子层保护，则应设置为 1，否则设置为 0。仅当使能安全功能子字段设置为 1 时，MHR 的辅助安全报头字段才会显示。

5.3.3 帧待处理

帧待处理字段的长度为 1 位，如果发送帧的设备有更多数据要发送给接收方，则应设置为 1。否则，该子字段应设置为 0。

帧待处理子字段只能用于信标帧、在 CAP 期间由使能信标模式的 PAN 中运行的设备发送的帧，或在任何时候由使能非信标模式的 PAN 中运行的设备发送的帧。在所有其他时间，该字段在发送时应设置为 0，在接收时被忽略。

5.3.4 应答请求

应答请求字段的长度为 1 位，指定接收设备在接收到数据或 MAC 命令帧时是否需要应答。如果将该子字段设置为 1，则只有在接收时帧通过第三级过滤的情况下，接收设备才会发送应答帧。如果将该子字段设置为 0，则接收设备不会发送应答帧。

5.3.5 PAN ID 压缩

PAN ID 压缩字段的长度为 1 位，指定当源地址和目标地址都存在时，要发送的 MAC 帧是否仅包含其中一个 PAN 标识符字段。如果该子字段设置为 1 并且源地址和目标地址都存在，则帧中应只包含目标 PAN 标识符字段，并且假定源 PAN 标识符字段等于目标标识符的字段。如果该子字段设置为 0 并且源地址和目标地址都存在，则帧中应同时包含源 PAN 标识符和目标 PAN 标识符字段。如果只有一个地址存在，则该子字段应设置为零，并且帧中应只包含与该地址对应的 PAN 标识符字段。如果两个地址都不存在，则该子字段应设置为零，并且帧中不能包含任何 PAN 标识符字段。

5.3.6 目标寻址模式

目标寻址模式字段的长度为 2 位，应设置为下表中列出的非保留值之一。如果该子字段为零且帧类型子字段未指定该帧是应答帧还是信标帧，则源寻址模式子字段应为非零值，这意味着该帧定向到 PAN 协调器，相应的 PAN 标识符在源 PAN 标识符字段中指定。

表 5-2. 目标寻址模式和源寻址模式子字段的可能值

寻址模式值 b ₁ b ₀	说明
00	PAN 标识符和地址字段不存在
01	保留
10	地址字段包含一个 16 位短地址
11	地址字段包含一个 64 位扩展地址

5.3.7 帧版本

帧版本子字段的长度为 2 位，指定与帧对应的版本号。该子字段设置为 0x00 表示与 IEEE 标准 802.15.4-2003 兼容的帧，设置为 0x01 表示 IEEE 802.15.4 帧。所有其他子字段值均保留以备将来使用。

5.3.8 源寻址模式

源寻址模式子字段的长度为 2 位，应设置为表 5-2 中列出的非保留值之一。如果该子字段为零且帧类型子字段未指定该帧是应答帧，则目标寻址模式子字段应非零值，这意味着该帧源自 PAN 协调器，相应的 PAN 标识符在目标 PAN 标识符字段中指定。

5.4 网络报头

5.4.1 路程段字段

路程段字段可设置重新发送数据包的路程段数。例如，00h 表示不重新发送数据包。可能的路程段最大值为 0xFF。

5.4.2 帧控制字段

帧控制字段是一个位图，其定义了数据包的行为，如下图所示。

图 5-3. 网络报头——帧控制字段

Bits:0-1	2	3	4	5	6-7
Frame Type	Security Enabled	Infra Cluster	Ack. Request	Address same as MAC	Reserved

下表详细介绍了网络报头的帧控制字段。

表 5-3. 网络报头帧控制字段说明

位编号	字段名称	说明
6-7	保留	对于本实现，将该位设置为 0。
5	地址与 MAC 相同	MAC 地址字段与网络地址字段相同时，该位置 1。该设置适用于休眠终端设备轮询父设备获取数据的情况，因为对于网络层的单一路程段，无线传输的字节数相对较少。
4	应答请求	源设备请求来自目标设备的网络层接收应答时，该位置 1。
3	集群内部	在本实现中保留。将该位设置为 1。
2	使能安全功能	数据包在应用程序级加密时，该位置 1。

..... (续)		
位编号	字段名称	说明
0-1	帧类型	这些位的含义如下： • 00——数据 • 01——命令 • 10——制造商特定 • 11——保留

5.4.3 序列号字段

序列号字段的长度为 1 个字节，指定帧的序列标识符。对于源自节点的每个传出帧，序列号字段应每次递增 1，对于路由的帧则必须保持不变。

5.4.4 目标 PANID 字段

目标 PANID 字段的长度为 2 个字节，指定帧的预期接收方的 PAN 标识符。仅当地址与设置为 0 的 MAC 位相同时，才会出现此字段。

5.4.5 源地址字段

源地址字段的长度为 2 个字节，指定原始帧对应的节点的网络地址。

5.4.6 目标地址字段

目标地址字段的长度为 2 个字节，指定目标节点的网络地址。对于除单播到节点之外的其他帧，可以按照下表设置目标地址字段。不支持使用长地址进行数据发送。

表 5-4. 网络报头目标地址字段说明

目标地址值	说明
0xFFFF	向每个设备广播
0xFFFE	向所有 FFD 多播
0xFFFD	向所有协调器多播

5.4.7 辅助安全报头字段

辅助安全报头字段指定了安全处理所需的信息，其中包括如何保护帧（安全级别）以及帧计数器的相关信息。仅当帧控制字段中使能安全功能的子字段设置为 1 时，该字段才会显示。

表 5-5. 辅助安全报头字段

字节：1	4	8
安全级别	帧计数器	源长地址

5.4.7.1 安全级别

支持的安全级别如下：

- 0（无安全性）
- 1（身份验证——4 字节 MIC）
- 4（仅加密）
- 5（通过身份验证进行加密——4 字节 MIC）

5.5 MiWi 网状网络——设备寻址机制

在网络上执行路由时，MiWi 网状网络使用 2 字节短地址指定网络中的节点。在加入过程中分配地址。使用低字节标识终端设备。使用高字节标识协调器。

Bit 15:8	Bit 7	Bit 6:0
协调器标识符	RxOnWhenIdle	终端设备标识符

5.6 MiWi 网状网络——网络

MiWi 网状网络功能分类如下：

1. 网络调试
2. 启动和加入网络
3. 在网络中路由

5.6.1 网络调试

网络调试功能可控制加入网络的设备。

1. PAN 协调器上的应用程序从一台或多台设备读取 IEEE 地址（例如，改进后从条形码读取）。
2. PAN 协调器使用读取的信息计算 64 位布隆过滤器值。
3. 计算出的布隆过滤器值被发送到网络中的所有协调器。
4. 协调器仅向在布隆过滤器中存在 IEEE 地址的设备提供信标。

默认情况下，PAN 协调器允许任何设备加入，因为 BLOOM_AUTO_JOIN 已使能，这意味着不会过滤 IEEE 地址。

如果用户想要根据通过 MiApp_Commissioning_AddNewDevice 设置的 IEEE 地址过滤设备，需要使用 MiApp_Set API 将 BLOOM_AUTO_JOIN 设置为 0。

用户也可以在运行时使用 MiApp_Commissioning_AddNewDevice 添加新设备。网络将允许这些新添加的设备加入。

5.6.2 启动和加入网络

1. 只有 PAN 协调器才能启动网络。
2. 加入设备会发送信标请求以获取有关其个人操作空间中可用网络的信息。
3. PAN 协调器或协调器使用布隆过滤器值解析特定的 IEEE 地址，以此来评估信标请求。如果找到，则会发送一个带有信标有效负载的信标帧，其中包括 PAN 协调器路程段数和布隆过滤器值（64 字节）。如果未找到，则会丢弃数据包。
4. 接收到信标帧后，加入设备对其进行解析，并基于布隆过滤器值检查自身的地址，然后根据接收到的信标的关联许可、子容量和链路质量指示（Link Quality Indicator, LQI）来决定其父项。选择父项后，会向所选父项单播网状网络连接请求数据包（包括其容量和 JoinWish 字段）。
JoinWish 字段有 2 位用于 C 和 ED，其余位保留。
 - 如果在 JoinWish 中这两个位均置 1，如果网络中的协调器数量已满，特定设备将作为终端设备加入。
 - 如果仅 C 位置 1，则设备仅作为协调器加入。
 - 如果仅 ED 位置 1，则设备仅作为终端设备加入。
5. 如果父项是 PAN 协调器并且 JoinWish 字段的 C 和 ED 均置 1 或仅 C 置 1，则 PAN 协调器会检查是否有新的协调器地址。如果有，则会发送网状网络连接响应，并将设备地址作为新分配的协调器地址。如果地址不可用或 JoinWish 字段只有 ED 置 1，则会分配终端设备地址并发送网状网络连接响应。
6. 如果父项是协调器，则会分配终端设备地址并发送网状网络连接响应（其中设备地址作为分配的终端设备地址）。
7. 加入设备解析网状网络连接响应，并使用接收到的网络地址和接收到的网络密钥在网络中进一步通信。
8. 具有协调器功能的加入设备会接收终端设备地址，并根据角色升级超时设置（可配置），向 PAN 协调器发送角色升级请求数据包，以便将其角色从终端设备升级到协调器。
9. PAN 协调器接收到角色升级请求时，会检查协调器地址是否可用。如果地址可用，则会分配一个新的协调器地址并发送角色升级响应，其中包含分配的地址且状态为成功。如果地址不可用，则会发送状态为失败的角色升级响应。

5.6.3 在网络中路由

1. 在加入过程和角色升级期间，所有协调器中的路由表都会更新。
2. 协调器中的路由表用于将数据包路由到目标设备。
3. 当设备没有目标的下一路程段地址时，会触发一个广播请求路由到目标设备。
4. 与 AODV 路由协议中的传统路由请求不同，应答路由表中具有下一路程段信息的任意节点生成。
5. 源设备（发起路由请求）根据较少的路程段和最佳 LQI 为目标选择路由回复。
6. 为定期建立和同步网络，会根据预先配置的时间间隔将路由表更新广播到单一路程段。
7. 这可确保网络中的协调器与其邻近设备共享邻近设备的信息。

5.7 MiWi 网状网络的宏

本节介绍了 MiWi 网状网络的宏。

5.7.1 CHANNEL_MAP

说明	信道映射是一种位图，用于选择合适的信道以在网络中启动或建立连接。
默认值	• SAMR21——（1<<25） • SAMR30——（1<<2）
范围	基于物理层的位图。以下范围内的任意位置 1 或清零均有效。 • 2.4 GHz（SAMR21）——0x07FFF800 • SubGHz（SAMR30）——0x000007FF
存储器使用情况	无
可配置位置	miwi_config_mesh.h
备注	对于 SAMR30，只能使用此宏更改信道 1-10。要使用信道 0，必须根据欧洲频段数据手册的建议修改 PHY_Init()，以加入 TX 功率和 PHY 模式设置。

5.7.2 KEEP_ALIVE_COORDINATOR_SEND_INTERVAL

说明	具有协调器功能的设备向 PAN 协调器发送保活帧的时间间隔（以秒为单位）。接收到此帧后，PAN 协调器会刷新该协调器的超时时间。
默认值	120
范围	1 - 65535
存储器使用情况	无
可配置位置	miwi_config_mesh.h
备注	KEEP_ALIVE_COORDINATOR_TIMEOUT_IN_SEC 基于此值。

5.7.3 KEEP_ALIVE_COORDINATOR_TIMEOUT_IN_SEC

说明	PAN 协调器维护协调器条目以保存其地址的超时时间（以秒为单位）。每个协调器应在此超时时间内至少向 PANC 发送一个保活帧。
默认值	KEEP_ALIVE_COORDINATOR_SEND_INTERVAL *10 KEEP_ALIVE_COORDINATOR_SEND_INTERVAL 设为 120 时，默认值为 1200。
范围	1 - 65535
存储器使用情况	无

可配置位置	miwi_config_mesh.h
备注	无

5.7.4 KEEP_ALIVE_RXONENDDEVICE_SEND_INTERVAL

说明	终端设备向其协调器发送保活帧的时间间隔（以秒为单位）。接收到此帧后，协调器会刷新该终端设备的超时时间。
默认值	120
范围	1 - 65535
存储器使用情况	无
可配置位置	miwi_config_mesh.h
备注	KEEP_ALIVE_RXONENDDEVICE_TIMEOUT_IN_SEC 基于此值。

5.7.5 KEEP_ALIVE_RXONENDDEVICE_TIMEOUT_IN_SEC

说明	协调器维护终端设备条目以保存其地址的超时时间（以秒为单位）。每个终端设备应在此超时时间内至少向协调器发送一个保活帧。
默认值	KEEP_ALIVE_RXONENDDEVICE_SEND_INTERVAL * 10（即为 1200） KEEP_ALIVE_COORDINATOR_SEND_INTERVAL 设为 120 时，默认值为 1200。
范围	1 - 65535
存储器使用情况	无
可配置位置	miwi_config_mesh.h
备注	无

5.7.6 DATA_REQUEST_SEND_INTERVAL

说明	休眠终端设备向其协调器发送数据请求帧的时间间隔（以秒为单位）。接收到此帧后，协调器会刷新该休眠终端设备的超时时间，并发送在间接队列中缓存的数据。
默认值	3
范围	1 - 254
存储器使用情况	无
可配置位置	miwi_config_mesh.h
备注	RXOFF_DEVICE_TIMEOUT_IN_SEC 和 MAXIMUM_DATA_REQUEST_SEND_INTERVAL 基于此值

5.7.7 RXOFF_DEVICE_TIMEOUT_IN_SEC

说明	协调器维护休眠终端设备条目以保存其地址的超时时间（以秒为单位）。每个休眠终端设备应在此超时时间内至少向协调器发送一个数据请求。
默认值	DATA_REQUEST_SEND_INTERVAL * 20（即，60） DATA_REQUEST_SEND_INTERVAL 设为 3 时，默认值为 60。
范围	1 - 65535
存储器使用情况	无

可配置位置	miwi_config_mesh.h
备注	无

5.7.8 MAXIMUM_DATA_REQUEST_SEND_INTERVAL

说明	网络中终端设备数据请求的最大时间间隔（单位为秒）。
默认值	DATA_REQUEST_SEND_INTERVAL * 2（即，6）
范围	1 - 254
存储器使用情况	无
可配置位置	miwi_config_mesh.h
备注	无

5.7.9 MAX_NUMBER_OF_DEVICES_IN_NETWORK

说明	该宏用于配置要存储以供调试的设备 IEEE 地址的数量。
默认值	32
范围	1 - 255
存储器使用情况	256 字节 RAM，用于 32 个条目，即，每个条目 8 个字节
可配置位置	miwi_config_mesh.h
备注	无

5.7.10 JOIN_WISH

说明	配置设备，使其按定义的角色加入网络。更多信息，请参见 5.6.2 启动和加入网络 。
默认值	- 协调器——JOINWISH_ANY - 终端设备——JOINWISH_ENDEVICE
范围	- JOINWISH_ENDEVICE——0x01 - JOINWISH_COORD_ALONE——0x02 - JOINWISH_ANY——0x03
存储器使用情况	无
可配置位置	miwi_config_mesh.h
备注	无

5.7.11 ROLE_UPGRADE_INTERVAL_IN_SEC

说明	具有协调器功能的终端设备请求 PANC 将其角色升级为协调器的时间间隔（单位为秒）。有关角色升级的更多信息，请参见 5.6.2 启动和加入网络 。
默认值	25
范围	1 - 254
存储器使用情况	无
可配置位置	miwi_config_mesh.h
备注	无

5.7.12 CONNECTION_RESPONSE_WAIT_IN_SEC

说明	将连接请求发送到网络中的任意协调器后等待连接响应的时间间隔（单位为秒）。
默认值	5
范围	1 - 254
存储器使用情况	无
可配置位置	miwi_config_mesh.h
备注	无

5.7.13 NUM_OF_COORDINATORS

说明	该宏用于配置网络中的协调器数量。还用于分配 PANC 上的协调器表，以维护每个协调器的 IEEE 地址和超时。
默认值	64
范围	1 - 200
存储器使用情况	768 字节 RAM，用于 64 个条目，即，每个条目 12 个字节
可配置位置	miwi_config_mesh.h
备注	无

5.7.14 NUM_OF_NONSLEEPING_ENDDEVICES

说明	该宏用于配置网络中的非休眠终端设备数量。还用于分配每个协调器上的设备表，以维护每个非休眠终端设备的 IEEE 地址和超时。
默认值	5
范围	1 - 127
存储器使用情况	80 字节 RAM，用于 5 个条目，即，每个条目 16 个字节
可配置位置	miwi_config_mesh.h
备注	无

5.7.15 NUM_OF_SLEEPING_ENDDEVICES

说明	该宏用于配置网络中的休眠终端设备数量。还用于分配每个协调器上的休眠设备表，以维护每个休眠终端设备的 IEEE 地址和超时。
默认值	5
范围	1 - 128
存储器使用情况	100 字节 RAM，用于 5 个条目，即，每个条目 20 个字节
可配置位置	miwi_config_mesh.h
备注	无

5.7.16 ROUTE_UPDATE_INTERVAL

说明	加入网络后为相邻设备发送路由更新的周期时间间隔（单位为秒）。
默认值	60

范围	1 - 254
存储器使用情况	无
可配置位置	miwi_config_mesh.h
备注	无

5.7.17 ROUTE_REQ_WAIT_INTERVAL

说明	在发送路由请求以发现特定协调器的路由后，等待路由回复的超时时间（单位为秒）。
默认值	5
范围	1 - 254
存储器使用情况	无
可配置位置	miwi_config_mesh.h
备注	无

5.7.18 INDIRECT_DATA_WAIT_INTERVAL

说明	将间接数据保存到休眠终端设备的超时时间（单位为秒）。该值至少须保持在数据请求间隔的两倍以上，这样才能确保可靠的数据传输。
默认值	25
范围	1 - 254
存储器使用情况	无
可配置位置	miwi_config_mesh.h
备注	无

5.7.19 ED_LINK_FAILURE_ATTEMPTS

说明	在确认链路故障之前，通过父设备对终端设备连续尝试连接的次数。
默认值	15
范围	1 - 254
存储器使用情况	无
可配置位置	miwi_config_mesh.h
备注	无

5.7.20 FRAME_RETRY

说明	用于定义连接目标失败后重新尝试的次数。
默认值	3
范围	0 - 254
存储器使用情况	无
可配置位置	miwi_config_mesh.h
备注	此重试配置与 MAC 层上的重试次数无关。 注： 从设备传出的帧会在 MAC 层重试 3 次。

5.7.21 REBROADCAST_TABLE_SIZE

说明	此宏用于配置要存储的条目数，以避免网络中的每条广播都被重复广播。
默认值	10
范围	1 - 255
存储器使用情况	40 字节 RAM，可存储 10 个条目，即每个条目 4 个字节。
可配置位置	miwi_config_mesh.h
备注	无

5.7.22 REBROADCAST_TIMEOUT

说明	将广播数据保存在重新广播表中以避免再次重新广播的超时时间（单位为秒）。
默认值	5
范围	1 - 254
存储器使用情况	无
可配置位置	miwi_config_mesh.h
备注	无

5.7.23 DUPLICATE_REJECTION_TABLE_SIZE

说明	设置重复拒绝表的大小，该表用于避免对应用程序进行多次数据指示。
默认值	10
范围	1 - 255
存储器使用情况	40 字节 RAM，可存储 10 个条目，即每个条目 4 个字节。
可配置位置	miwi_config_mesh.h
备注	无

5.7.24 MAX_BEACON_RESULTS

说明	在主动扫描期间分配用于接收信标响应的条目数。
默认值	5
范围	1 - 255
存储器使用情况	90 字节 RAM，可存储 5 个条目，即每个条目 18 个字节。
可配置位置	miwi_config_mesh.h
备注	无

5.7.25 MESH_SECURITY_LEVEL

说明	IEEE 802.15.4 中定义的 CCM* 安全级别。
默认值	5
范围	0 - 7
存储器使用情况	无

可配置位置	miwi_config_mesh.h
备注	无

5.7.26 PUBLIC_KEY_DEFAULT

说明	公钥是存储在所有设备中的初始密钥，首次通信时使用此密钥，直到获得网络密钥。
默认值	{0x00,0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A, 0x0B, 0x0C, 0x0D, 0x0E, 0x0F}
范围	任意 16 字节值
存储器使用情况	无
可配置位置	miwi_config_mesh.h
备注	无

5.7.27 NETWORK_KEY_DEFAULT

说明	成功加入网络后用于处理事务的网络密钥。
默认值	{0x00,0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77, 0x88, 0x99, 0xAA, 0xBB, 0xCC, 0xDD, 0xEE, 0xFF}
范围	任意 16 字节值
存储器使用情况	无
可配置位置	miwi_config_mesh.h
备注	无

5.8 针对宏的建议

假设使用默认测试网络（带 WSN 演示应用程序），应注意以下事项。

1. 网络规模为 1 个 PAN 协调器、50 个协调器和 30 个终端设备。
2. 网络中的所有设备必须定期向 PAN 协调器报告。
3. 数据流大多是单向的（即上行链路需要更大的带宽）。
4. 来自所有设备的报告均使用 WSNMonitor 进行监视。

此网络使用默认值（见 5.7 MiWi 网状网络的宏）进行编程，经测试可工作 48 小时以上。

默认情况下，在 WSN 演示应用程序中，每个终端设备根据可容纳的设备数加入协调器。由于协调器数多于终端设备，因此大多数终端设备都会加入完全空闲的协调器。

必须根据网络中协调器数量的变化按比例修改以下宏（即，当协调器数量增加时，必须增加配置值，反之亦然）。

1. NUM_OF_COORDINATORS
2. KEEP_ALIVE_COORDINATOR_SEND_INTERVAL
3. KEEP_ALIVE_COORDINATOR_TIMEOUT_IN_SEC
4. ROUTE_UPDATE_INTERVAL
5. ROLE_UPGRADE_INTERVAL_IN_SEC

必须根据网络中终端设备数量的变化按比例修改以下宏（即，当协调器数量增加时，必须增加配置值，反之亦然）。

- 对于休眠终端设备：
 - a. NUM_OF_SLEEPING_ENDDEVICES
 - b. DATA_REQUEST_SEND_INTERVAL
 - c. RXOFF_DEVICE_TIMEOUT_IN_SEC

-
- d. MAXIMUM_DATA_REQUEST_SEND_INTERVAL
 - e. INDIRECT_DATA_WAIT_INTERVAL
 - 对于非休眠终端设备（即，RXON 终端设备）：
 - a. NUM_OF_NONSLEEPING_ENDDEVICES
 - b. KEEP_ALIVE_RXONENDDEVICE_SEND_INTERVAL
 - c. KEEP_ALIVE_RXONENDDEVICE_TIMEOUT_IN_SEC

注： 以上宏基于 SAMR21（即 2.4 GHz）进行了测试；因此，对于基于 SAMR30（即 Sub-GHz）的相同规模的网络，必须增加相应值以补偿数据速率的下降。

5.9 延长休眠终端设备的电池寿命

除了控制器支持的休眠模式外，DATA_REQUEST_SEND_INTERVAL 配置也会直接影响从休眠模式唤醒的频率和电池功耗。

6. 网络冻结功能

网络冻结功能将关键网络信息保存在非易失性存储器（Nonvolatile Memory, NVM）中，并在掉电再上电后进行恢复。这样应用程序就能支持掉电再上电，网络可以在掉电再上电后恢复到之前的状态，无需进行大量报文交换。

此外，耗损均衡实现减少了“备份-擦除-重写”周期的数量，从而提高了闪存寿命。请参见 miwi_mesh_pds.c 文件，此文件指定了 NVM 中存储的参数的相关信息。

6.1 接口

通过在应用程序项目的配置文件中定义 ENABLE_NETWORK_FREEZER API 来使能网络冻结功能。通过调用 MiApp 函数 MiApp_ProtocolInit 来调用该功能。在应用程序中使能网络冻结功能后，网络信息从 NVM 中恢复；否则，无线节点会从初始阶段启动。如果禁止网络冻结功能，则节点会始终以出厂设置启动。

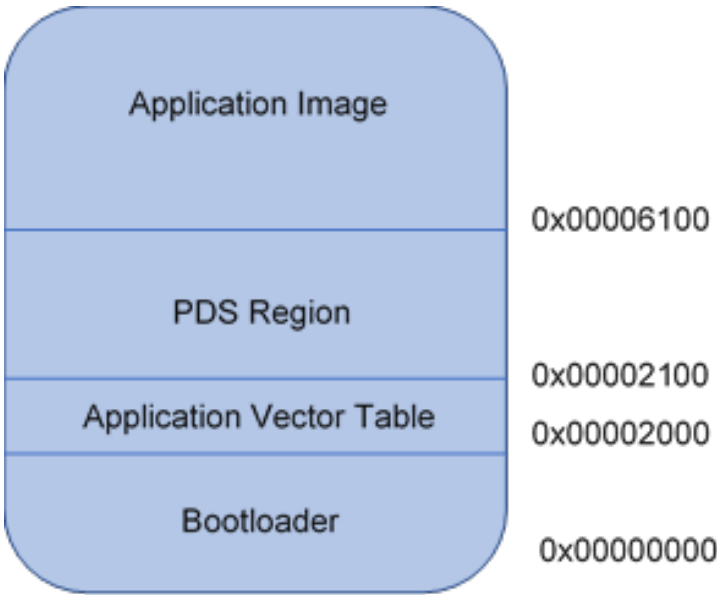
6.2 附加说明

网络冻结功能需要 NVM 来存储关键网络信息。用于此实现的 NVM 为内部闪存。

6.3 默认存储器布局

使能网络冻结功能（PDS——永久性数据存储）的存储器布局详细信息如下所示。用户可通过相应更改代码和链接器文件中的 D_NV_SECTOR_SIZE 来更改扇区的大小。

图 6-1. 默认存储器布局



7. 休眠模式

提高休眠设备的续航时间对于大多数应用都很重要。通常设备有工作模式和休眠模式两种模式。上电后，节点始终以工作模式启动，其 MCU 完全开启。应用可使用 `ENABLE_SLEEP_FEATURE API` 检查协议栈是否允许其进入休眠模式。如果允许，则应用可以进入休眠模式，最长时间由协议栈根据正常工作的要求来决定。

在休眠模式下，RF 芯片和 MCU 处于低功耗状态，只有 MCU 唤醒所需的功能保持活动状态。因此，在休眠模式下，应用程序无法执行任何射频 Tx/Rx 操作或与外设通信。

工作模式期间会产生大量功耗，用于在占空比内请求和发送数据。因此，设备是否处于工作状态取决于其轮询周期。这可使用配置选项进行控制。在所有节点中，只有终端设备可以休眠。

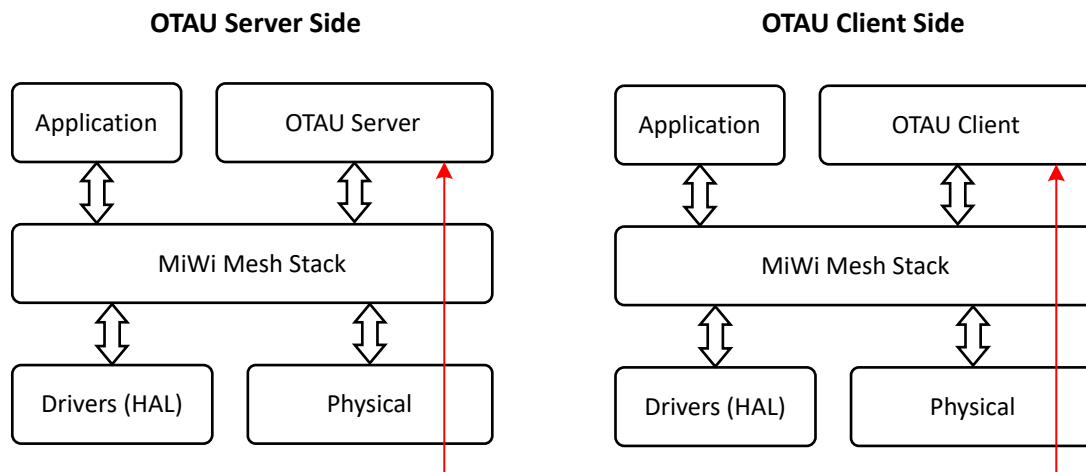
7.1 接口

可以通过在应用程序项目的配置文件中定义 `ENABLE_SLEEP_FEATURE API` 来使能休眠模式。有关更多详细信息，请参见 [MiApp API 说明](#)。

8. 无线更新

下图给出了无线更新（Over-The-Air Update, OTAU）的固件架构。

图 8-1. OTAU 固件架构



8.1 OTAU 服务器

OTAU 服务器通过 UART 或 USB 从 PC 接收命令或向其发送命令。要进行升级，通过 MiWi 网状网络协议栈层发送所需帧，以连接客户机。服务器充当客户机和 PC 上的 OTAU 工具之间的桥梁，也就是说，服务器端的 OTAU 模块没有额外的智能功能。

8.2 OTAU 客户机

OTAU 客户机通过无线方式接收或发送专有命令以便与 OTAU 服务器通信。

8.3 OTAU 的域

以下部分介绍了 OTAU 服务器与客户机的通知和升级域。

8.3.1 通知域

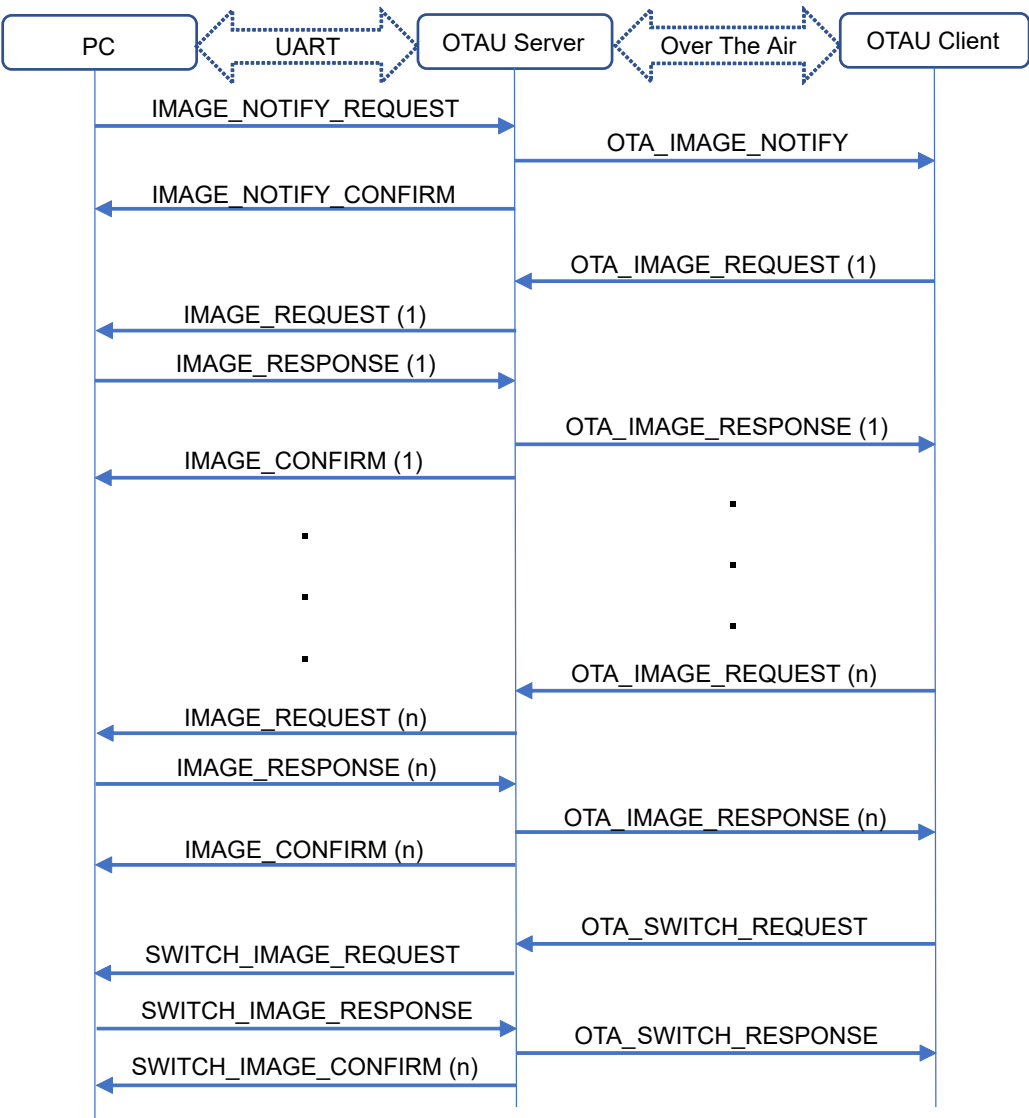
1. 提供有关客户机的基本信息（例如 IEEE 地址、短地址和下一路程序地址），以连接 OTAU 服务器，进而绘制网络拓扑。
2. 发送在客户机上开启 LED 的命令，方便在大型网络上识别客户机。
3. 供用户获取与应用程序相关的其他信息，例如固件名称、固件版本、电路板名称和电路板版本。

8.3.2 升级域

- 支持通过专有协议交换对每个客户机进行 OTA 升级。
- 可在服务器上配置请求映像块的周期。
- 支持在升级所有节点时单独切换到新映像。

8.3.3 升级——顺序流程图

图 8-2. 升级——顺序流程图

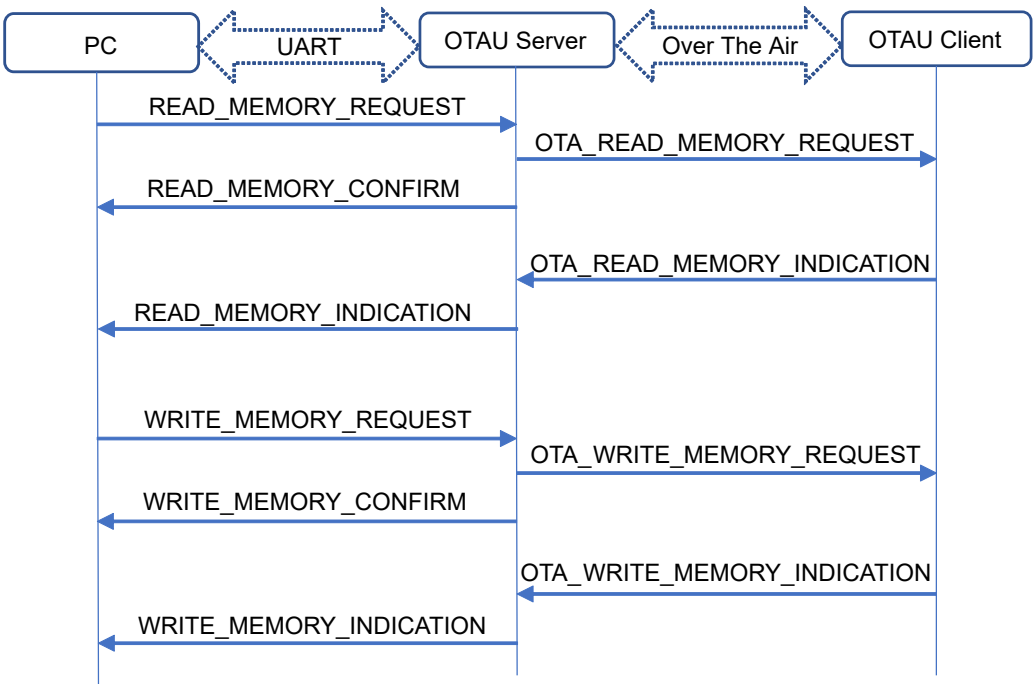


8.3.4 调试

- 支持用于读取/写入存储器变量的观察窗口。
- 支持定期提供记录以跟踪函数调用的序列。
- 支持跟踪功能以跟踪出现问题之前的最后一次函数调用。

8.3.5 调试——顺序流程图

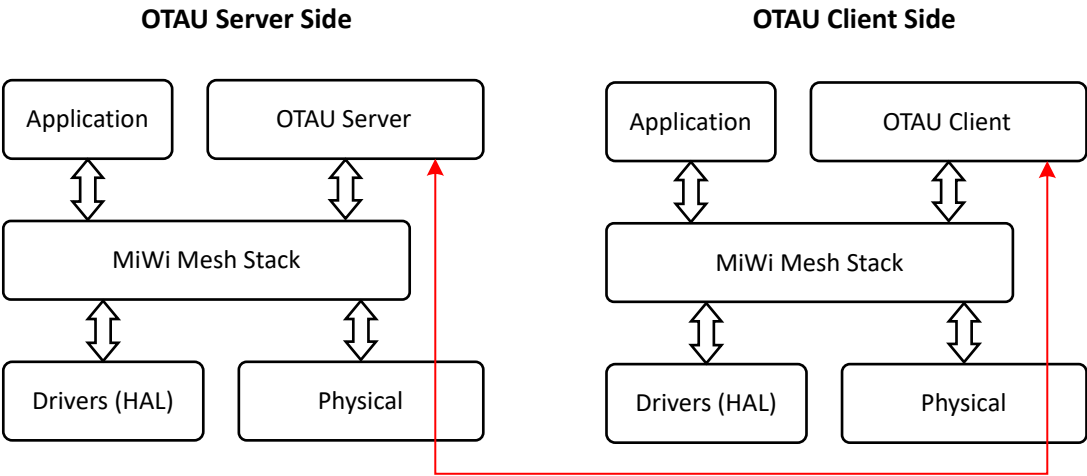
图 8-3. 调试——顺序流程图



8.4 PHY 模式操作

下图给出了使用 PHY 模式时无线更新（OTAU）的固件架构。

图 8-4. PHY 模式的 OTAU 固件架构



PHY 模式提供了一种配置，不需要网络层，只使用物理层在服务器与客户机之间建立通信。服务器通过 IEEE 地址与客户机通信，来自网络层的路由会被旁路，因此服务器必须处于 RF 范围内。

8.5 用于 OTAU 的编译器开关

OTAU_ENABLED

必须在项目符号中添加 OTAU_ENABLED 开关，以启用升级支持。

OTAU_SERVER

在项目符号中启用 OTAU_SERVER 开关后，节点用作 OTAU 服务器。如果未启用，则节点用作 OTAU 客户机。

OTAU_PHY_MODE

项目符号中默认未启用 OTAU_PHY_MODE。如果启用，则 OTAU 模块可在 PHY 模式下进行升级或调试。

9. MiApp API

下表列出了支持的 API。

表 9-1. MiApp API

序号	支持的 API	支持的拓扑
1	miwi_status_t MiApp_ProtocolInit (defaultParametersRomOrRam_t *defaultRomOrRamParams, defaultParametersRamOnly_t *defaultRamOnlyParams)	P2P/星型/网状拓扑
2	bool MiApp_Set(enum id, uint8_t value)	P2P/星型/网状拓扑
3	bool MiApp_StartNetwork(uint8_t Mode, uint8_t ScanDuration, uint32_t ChannelMap, FUNC ConfCallback)	P2P/星型/网状拓扑
4	uint8_t MiApp_SearchConnection(uint8_t ScanDuration, uint32_t ChannelMap, FUNC ConfCallback)	P2P/星型/网状拓扑
5	uint8_t MiApp_EstablishConnection(uint8_t Channel, uint8_t addr_len, uint8_t addr, uint8_t Capability_info, FUNC ConfCallback)	P2P/星型/网状拓扑
6	void MiApp_RemoveConnection(uint8_t ConnectionIndex)	P2P/星型/网状拓扑
7	void MiApp_ConnectionMode(uint8_t Mode)	P2P/星型/网状拓扑
8	MiApp_SendData(uint8_t addr_len, uint8_t addr, uint8_t len, uint8_t pointer, FUNC ConfCallback)	P2P/星型/网状拓扑
9	MiApp_SubscribeDataIndicationCallback(FUNC callback)	P2P/星型/网状拓扑
10	uint8_t MiApp_NoiseDetection(uint32_t ChannelMap, uint8_t ScanDuration, uint8_t DetectionMode, OUTPUT uint8_t NoiseLevel)	P2P/星型/网状拓扑
11	uint8_t MiApp_TransceiverPowerState(uint8_t Mode)	P2P/星型/网状拓扑
12	bool MiApp_InitChannelHopping(uint32_t ChannelMap)	P2P/星型/网状拓扑
13	bool MiApp_ResyncConnection(uint8_t ConnectionIndex, uint32_t ChannelMap)	P2P/星型/网状拓扑
14	uint8_t Total_Connections(void)	P2P/星型拓扑
15	void MiApp_BroadcastConnectionTable()	星型拓扑
16	bool MiApp_Set(enum id, uint8_t value)	网状拓扑
17	bool MiApp_IsMemberOfNetwork(void)	网状拓扑
18	bool MiApp_Get(enum id, uint8_t value)	网状拓扑
19	bool MiApp_Set(enum id, uint8_t value)	网状拓扑
20	bool MiApp_SubscribeReConnectionCallback(ReconnectionCallback_t callback)	P2P/星型/网状拓扑
21	bool MiApp_ResetToFactoryNew(void)	P2P/星型/网状拓扑
22	bool MiApp_ReadyToSleep(uint32_t* sleepTime)	网状拓扑
23	bool MiApp_ManuSpecSendData(uint8_t addr_len, uint8_t *addr, uint8_t msglen, uint8_t *msgpointer, uint8_t msghandle, bool ackReq, DataConf_callback_t ConfCallback)	网状拓扑

..... (续)		
序号	支持的 API	支持的拓扑
24	bool MiApp_SubscribeManuSpecDataIndicationCallback(PacketIndCal lback_t callback)	网状拓扑
25	bool MiApp_IsConnected(void)	网状拓扑
26	uint16_t MiApp_MeshGetNextHopAddr(uint16_t destAddress)	网状拓扑

10. MiApp API 说明

本节介绍 MiApp API。

10.1 MiApp_ProtocolInit

API	<code>miwi_status_t MiApp_ProtocolInit(defaultParametersRomOrRam_t *defaultRomOrRamParams, defaultParametersRamOnly_t *defaultRamOnlyParams)</code>
说明	这是用于初始化 Microchip 专无线协议的用户接口主函数，由应用程序层选择。通常，此函数必须在硬件初始化后调用，然后才能调用其他 MiApp 接口。
前提条件	必须进行硬件初始化。
参数	- defaultParametersRomOrRam_t defaultRomOrRamParams——MiWi™ 网状网络的默认参数。 - defaultParametersRamOnly_t defaultRamOnlyParams——MiWi 网状网络的默认参数。 - 在 P2P/星型拓扑条件下被忽略
返回	初始化状态
示例	<pre><code> HardwareInit(); MiApp_ProtocolInit(); </code></pre>
备注	如果接收到 RECONNECTION_IN_PROGRESS 状态，则应用程序需要等待重新连接回调函数，然后才能继续调用更多 MiApp API。

10.2 MiApp_Set

API	<code>bool MiApp_Set(set_params id, uint8_t *value)</code>
说明	这是用于在 MiWi™ 协议栈中设置不同值的用户接口主函数。
前提条件	必须进行协议初始化。
参数	- set_params id——要设置的值的标识符 - value——要设置的值
返回	指示设置操作是否成功执行的布尔值。
示例	<pre><code> if(true == MiApp_Set(CHANNEL, 15)) { // 信道更改成功 } </code></pre>
备注	无

10.3 MiApp_StartConnection

API	<code>bool MiApp_StartConnection(uint8_t Mode, uint8_t ScanDuration, uint32_t ChannelMap, connectionConf_callback_t ConfCallback)</code>
说明	这是应用程序层用于启动 PAN 的用户接口主函数。通常，此函数由位于 PAN 中的第一个 PAN 协调器调用。如果在输入模式下指定，则 PAN 协调器可在噪声扫描后启动 PAN。
前提条件	必须进行协议初始化。
参数	<code>uint8_t Mode</code>——是否在噪声扫描后启动 PAN。可选模式如下。 <code>START_CONN_DIRECT</code>——不进行噪声扫描，直接启动 PAN。 <code>START_CONN_ENERGY_SCN</code>——先执行能量扫描，然后在噪声最低的信道上启动 PAN。 <code>START_CONN_CS_SCN</code>——先执行载波侦听扫描，然后在噪声最低的信道上启动 PAN。 <code>uint8_t ScanDuration</code>——在单一信道上执行扫描的最长时间。该值介于 5 到 14 之间。可以根据 IEEE 802.15.4 规范规定的以下公式计算执行扫描的实际时间： $960 \times (2^{\text{ScanDuration}} + 1) \times 10^{(-6)}$ 秒 如果连接模式为 <code>START_CONN_DIRECT</code>，则会丢弃 <code>ScanDuration</code>。 <code>uint32_t ChannelMap</code>——用于执行噪声扫描的信道位图。32 位双字参数用一位表示 0 到 31 的相应信道。例如，<code>0x00000003</code> 表示扫描信道 0 和信道 1。如果连接模式为 <code>START_CONN_DIRECT</code>，则会丢弃 <code>ChannelMap</code>。 <code>connectionConf_callback_t ConfCallback</code>——执行启动的连接过程时调用的回调例程。
返回	指示 PAN 是否成功启动的布尔值。
示例	<pre><code> // 在扫描所有可能的信道后，在噪声最低的信道上启动 PAN。 MiApp_StartConnection(START_CONN_ENERGY_SCN, 10, 0x07FFF800, callback); </code></pre>
备注	无

10.4 MiApp_SearchConnection

API	<code>uint8_t MiApp_SearchConnection(uint8_t ScanDuration, uint32_t ChannelMap, SearchConnectionConf_callback_t ConfCallback)</code>
说明	这是应用程序层用于执行主动扫描的用户接口主函数。在此函数调用后，所有主动扫描响应均以 <code>ACTIVE_SCAN_RESULT</code> 结构的格式存储在全局变量 <code>ActiveScanResults</code> 中。返回值表示主动扫描结果数组中有效主动扫描响应的总数。
前提条件	已完成协议初始化。
参数	<code>uint8_t ScanDuration</code>——在单一信道上执行扫描的最长时间。该值介于 5 到 14 之间。可以根据 IEEE 802.15.4 规范规定的以下公式计算执行扫描的实际时间： $960 \times (2^{\text{ScanDuration}} + 1) \times 10^{(-6)}$ 秒。 <code>uint32_t ChannelMap</code>——用于执行噪声扫描的信道位图。32 位双字参数用一位表示 0 到 31 的相应信道。例如，<code>0x00000003</code> 表示扫描信道 0 和信道 1。 <code>SearchConnectionConf_callback_t ConfCallback</code>——执行启动的连接过程时调用的回调例程。

返回	存储在全局变量 <code>ActiveScanResults</code> 中的有效主动扫描响应的数量。
示例	<pre><code> // 对所有可能的信道执行主动扫描 NumOfActiveScanResponse = MiApp_SearchConnection(10, 0xFFFFFFFF, callback); </code></pre>
备注	无

10.5 MiApp_EstablishConnection

API	<pre>uint8_t MiApp_EstablishConnection(uint8_t Channel, uint8_t addr_len, uint8_t *addr, uint8_t Capability_info, connectionConf_callback_t ConfCallback)</pre>
说明	这是应用程序层用于启动基于现有 PAN 的通信的用户接口主函数。对于 P2P 协议，此函数调用可建立一个或多个连接。对于网络协议，该函数可用于加入网络，或与射频范围外的节点建立虚拟套接字连接。
前提条件	已完成协议初始化。如果仅与预定义的设备建立连接，则必须先进行主动扫描，并且必须保存有效的主动扫描结果。
参数	<code>uint8_t channel</code>——用于调用加入程序的所选信道。 <code>uint8_t addr_len</code>——地址长度 <code>uint8_t *addr</code>——父项的地址 <code>uint8_t Capability_info</code>——设备的功能信息 <code>connectionConf_callback_t ConfCallback</code>——执行启动的连接过程时调用的回调例程
返回	连接表中对等设备的索引。
示例	<pre><code> // 与任意设备建立一个或多个连接 PeerIndex = MiApp_EstablishConnection(14, 8, 0x12345678901234567, 0x80, callback); </code></pre>
备注	如果通过此函数调用建立了多个连接，则返回值会指向其中一个对等设备的索引。

10.6 MiApp_RemoveConnection

API	<pre>void MiApp_RemoveConnection(uint8_t ConnectionIndex)</pre>
说明	这是用于断开连接的用户接口主函数。对于 P2P 协议，会移除连接。对于网络协议，如果输入参数所指的设备是调用此函数的设备的父设备，则调用设备会随其子设备一起退出网络。如果输入参数所指的设备是调用此函数的设备的子设备，则目标设备会退出网络。
前提条件	收发器已初始化。节点会建立一个或多个连接。
参数	<code>uint8_t ConnectionIndex</code> ——连接表中要移除的连接的索引。
返回	无
示例	<pre><code> MiApp_RemoveConnection(0x00); </code></pre>
备注	无

10.7 MiApp_ConnectionMode

API	void MiApp_ConnectionMode(uint8_t Mode)
说明	这是应用程序层用于配置主设备如何响应连接请求的用户接口主函数。
前提条件	已完成协议初始化。
参数	uint8_t Mode——连接请求的响应模式。这些模式的权限按照定义逐渐降低。权限较高的模式具备权限较低的模式的所有权限。 可选模式如下： • ENABLE_ALL_CONN——响应所有连接请求 • ENABLE_PREV_CONN——响应连接表中的设备的连接请求 • ENABLE_ACTIVE_SCAN_RSP——只响应主动扫描 • DISABLE_ALL_CONN——禁止响应连接请求，包括主动扫描请求
返回	无
示例	<pre><code> // 响应所有连接请求 MiApp_ConnectionMode(ENABLE_ALL_CONN); </code></pre>
备注	无

10.8 MiApp_SendData

API	bool MiApp_SendData(uint8_t addr_len, uint8_t *addr, uint8_t msglen, uint8_t *msgpointer, uint8_t msghandle, bool ackReq, DataConf_callback_t ConfCallback)
说明	这是应用程序层用于单播报文的用户接口主函数之一。目标设备由输入参数 DestinationAddress 指定。应用有效负载使用 msgpointer 进行填充。
前提条件	已完成协议初始化。
参数	• uint8_t addr_len——目标地址长度 • uint8_t *addr——目标地址 • uint8_t msglen——报文长度 • uint8_t *msgpointer——报文/帧指针 • uint8_t msghandle——报文句柄 • bool ackReq——设置为接收网络级应答 注： 如果是广播数据，此参数将被丢弃。 • DataConf_callback_t ConfCallback——执行启动的数据过程时调用的回调例程。
返回	指示单播过程是否成功的布尔值。
示例	<pre><code> // 保护存储在 msgpointer 中的报文，然后将其广播到 // 输入参数中指定的永久地址。 MiApp_SendData(SHORT_ADDR_LEN, 0x0004, 5, "hello", 1, callback); </code></pre>

备注	无
----	---

10.9 MiApp_SubscribeDataIndicationCallback

API	<code>bool MiApp_SubscribeDataIndicationCallback(PacketIndCallback_t callback)</code>
说明	这是应用程序层用于调用 Microchip 专有协议栈以向应用程序注册报文指示回调函数的用户接口主函数。此函数调用协议栈状态机来保持协议栈运行。
前提条件	已完成协议初始化。
参数	无
返回	指示订阅操作是否成功的布尔值。
示例	<pre><code> if(true == MiApp_SubscribeDataIndicationCallback(ind)) { } </code></pre>
备注	无

10.10 MiApp_NoiseDetection

API	<code>uint8_t MiApp_NoiseDetection(uint32_t ChannelMap, uint8_t ScanDuration, uint8_t DetectionMode, uint8_t NoiseLevel)</code>
说明	这是应用程序层用于对多个信道执行噪声检测的用户接口主函数。
前提条件	已完成协议初始化。
参数	<code>uint32_t ChannelMap</code>——用于执行噪声扫描的信道位图。32 位双字参数用一位表示 0 到 31 的相应信道。例如，0x00000003 表示扫描信道 0 和信道 1。 <code>uint8_t ScanDuration</code>——在单一信道上执行扫描的最长时间。有效值介于 5 到 14 之间。执行扫描的实际时间可以根据 IEEE 802.15.4 规范规定的以下公式进行计算： $960 \times (2^{\text{ScanDuration}} + 1) \times 10^{(-6)} \text{秒}$ <code>uint8_t DetectionMode</code>——用于执行扫描的噪声检测模式。两个可选的扫描模式如下。 - NOISE_DETECT_ENERGY——能量检测扫描模式 - NOISE_DETECT_CS——载波侦听检测扫描模式 <code>uint8_t NoiseLevel</code>——噪声级别最低的信道的噪声级别
返回	噪声级别最低的信道。
示例	<pre><code> uint8_t NoiseLevel; OptimalChannel = MiApp_NoiseDetection(0xFFFFFFFF, 10, NOISE_DETECT_ENERGY, &NoiseLevel); </code></pre>
备注	无

10.11 MiApp_TransceiverPowerState

API	<code>uint8_t MiApp_TransceiverPowerState(uint8_t Mode)</code>
说明	这是应用程序层用于将 RF 收发器设置为休眠或唤醒模式的用户接口主函数。此函数仅适用于必须禁止收发器以节省电池电量的无线节点。
前提条件	已完成协议初始化。
参数	<code>uint8_t Mode</code>——要设置的 RF 收发器的电源状态模式。可选的电源状态如下： <code>POWER_STATE_SLEEP</code>——RF 收发器进入深度休眠模式 <code>POWER_STATE_WAKEUP</code>——RF 收发器进入唤醒状态或工作状态 <code>POWER_STATE_WAKEUP_DR</code>——将设备设置为唤醒模式并将数据请求发送到设备的关联设备
返回	操作状态。可能的状态如下： <code>SUCCESS</code>——操作成功。 <code>ERR_TRX_FAIL</code>——收发器未能进入休眠或唤醒模式。 <code>ERR_TX_FAIL</code>——数据请求命令发送失败。仅适用于输入模式为 <code>POWER_STATE_WAKEUP_DR</code> 的情况。 <code>ERR_RX_FAIL</code>——未收到对数据请求命令的任何响应。仅适用于输入模式为 <code>POWER_STATE_WAKEUP_DR</code> 的情况。 <code>ERR_INVLAID_INPUT</code>——无效输入模式。
示例	<pre><code> // 将 RF 收发器置于休眠模式 MiApp_TransceiverPowerState(POWER_STATE_SLEEP; // 将 MCU 置于休眠模式 Sleep(); // 通过 WDT、外部中断或任何其他方式唤醒 MCU // 确保 RF 收发器唤醒并发出数据请求 MiApp_TransceiverPowerState(POWER_STATE_WAKEUP_DR); </code></pre>
备注	无

10.12 MiApp_Get

API	<code>bool MiApp_Get(set_params id, uint8_t *value)</code>
说明	这是用于在 MiWi™ 协议栈中获取不同值的用户接口主函数
前提条件	已完成协议初始化
参数	<code>get_params id</code> ——要设置的值的标识符
返回	指示获取操作是否成功执行的布尔值
示例	<pre><code> value = MiApp_get(CHANNEL) </code></pre>
备注	无

10.13 MiApp_RoleUpgradeNotification_Subscribe

API	<code>bool MiApp_RoleUpgradeNotification_Subscribe (roleUpgrade_callback_t callback)</code> 仅适用于协调器。
说明	此 API 订阅角色升级通知。成功完成角色升级后，使用新的短地址调用回调函数。
前提条件	已完成协议初始化。
参数	<code>roleUpgrade_callback_t callback</code> ——完成角色升级后调用的回调例程
返回	指示订阅是否成功的布尔值

10.14 MiApp_Commissioning_AddNewDevice

API	<code>bool MiApp_Commissioning_AddNewDevice(uint64_t joinerAddress, bool triggerBloomUpdate)</code>
说明	用于向 PAN 协调器上的布隆过滤器添加设备。此函数仅适用于 PAN 协调器。
前提条件	已完成协议初始化。
参数	<code>uint8_t joinerAddress</code>——要添加的 IEEE 地址 <code>bool triggerBloomUpdate</code>——如果设置为 <code>true</code>，则会发送布隆更新
返回	如果成功添加，则为 <code>true</code> ；否则为 <code>false</code> 。

10.15 MiApp_SubscribeReConnectionCallback

API	<code>bool MiApp_SubscribeReConnectionCallback(ReconnectionCallback_t callback)</code>
说明	此 API 订阅当设备掉电再上电前如果处于网络中，则会在上电后重新连接的通知。重新连接设备时，会调用此回调函数。
前提条件	已完成协议初始化。
参数	<code>ReconnectionCallback_t callback</code> ——重新连接时调用的回调例程
返回	指示订阅是否成功的布尔值

10.16 MiApp_ResetToFactoryNew

API	<code>bool MiApp_ResetToFactoryNew(void)</code>
说明	此 API 会擦除 NVM 中的所有永久项并复位系统
前提条件	无
参数	无
返回	指示操作是否成功的布尔值

10.17 MiApp_ReadyToSleep

API	<code>bool MiApp_ReadyToSleep(uint32_t* sleepTime)</code>
说明	此 API 帮助了解协议栈是否准备好进入休眠状态，以及协议栈在就绪时允许休眠的时间
前提条件	无
参数	<code>uint32_t* sleep Time</code> ——指向休眠时间的指针，如果协议栈准备好进入休眠状态，则将用休眠时间填充
返回	指示协议栈是否准备好进入休眠状态的布尔值

10.18 MiApp_ManuSpecSendData

API	<code>bool MiApp_ManuSpecSendData(uint8_t addr_len, uint8_t *addr, uint8_t msglen, uint8_t *msgpointer, uint8_t msghandle, bool ackReq, DataConf_callback_t ConfCallback)</code>
说明	这是制造商特定数据的接口函数。目标设备由输入参数 <code>DestinationAddress</code> 指定。OTAU 模块使用此 API 实现升级支持。
前提条件	已完成协议初始化。
参数	<code>uint8_t addr_len</code>——目标地址长度 <code>uint8_t *addr</code>——目标地址 <code>uint8_t msglen</code>——报文长度 <code>uint8_t *msgpointer</code>——报文/帧指针 <code>uint8_t msghandle</code>——报文句柄 <code>bool ackReq</code>——设置为接收网络级应答 注：如果是广播数据，此参数将被丢弃。 <code>DataConf_callback_t ConfCallback</code>——执行启动的数据过程时调用的回调程序。
返回	指示单播过程是否成功的布尔值。
示例	<pre><code> // 保护存储在 msgpointer 中的报文，然后将其广播到 // 输入参数中指定的永久地址。 MiApp_ManuSpecSendData(SHORT_ADDR_LEN, 0x0004, 5, "hello", 1, callback); </code></pre>
备注	无

10.19 MiApp_SubscribeManuSpecDataIndicationCallback

API	<code>bool MiApp_SubscribeManuSpecDataIndicationCallback (PacketIndCallback_t callback)</code>
说明	这是 OTAU 模块用于注册制造商特定指示回调函数的用户接口主函数。
前提条件	已完成协议初始化。
参数	无

返回	指示订阅操作是否成功的布尔值。
示例	<pre><code> if(true == MiApp_SubscribeManuSpecDataIndicationCallback (ind)) { } </code></pre>
备注	无

10.20 MiApp_IsConnected

API	bool MiApp_IsConnected(void)
说明	用于检查 MiWi™ 网状拓扑与网络的连接。
前提条件	无
参数	无
返回	布尔值为 true 表示节点已连接到网络。

10.21 MiApp_MeshGetNextHopAddr

API	uint16_t MiApp_MeshGetNextHopAddr(uint16_t destAddress)
说明	用于获取下一路段的地址以访问 destAddress。
前提条件	无
参数	uint16_t destAddress——需要下一路段的设备的目标地址。
返回	用于访问 destAddress 的下一路段的地址。

11. 限制

已知限制如下：

1. 连续运行 1 天或 2 天后，某些返回休眠模式的 SAMR30 器件会出现随机行为，导致无法唤醒。
2. 如果启动过快，可能会丢失 P2P/星型拓扑中数据的确认回调。建议在客户应用程序中采取适当的去抖措施。
3. OTAU——CPU 工作在 48 MHz 频率（由带外部 32 KHz 晶振/时钟源的 DFLL 产生）下时，器件（随机）无法从休眠模式唤醒。

12. 文档版本历史

版本	日期	章节	说明
C	2020 年 10 月	8. 无线更新	更新了此章节
B	2019 年 8 月	3. MiWi P2P 拓扑 4. MiWi 星型拓扑 6.3 默认存储器布局	增加了新章节
A	2019 年 2 月	文档	初始版本

Microchip 网站

Microchip 网站 (www.microchip.com/) 为客户提供在线支持。客户可通过该网站方便地获取文件和信息。我们的网站提供以下内容:

- **产品支持**——数据手册和勘误表、应用笔记和示例程序、设计资源、用户指南以及硬件支持文档、最新的软件版本以及归档软件
- **一般技术支持**——常见问题解答 (FAQ)、技术支持请求、在线讨论组以及 Microchip 设计伙伴计划成员名单
- **Microchip 业务**——产品选型和订购指南、最新 Microchip 新闻稿、研讨会和活动安排表、Microchip 销售办事处、代理商以及工厂代表列表

产品变更通知服务

Microchip 的产品变更通知服务有助于客户了解 Microchip 产品的最新信息。注册客户可在他们感兴趣的某个产品系列或开发工具发生变更、更新、发布新版本或勘误表时, 收到电子邮件通知。

欲注册, 请访问 www.microchip.com/pcn, 然后按照注册说明进行操作。

客户支持

Microchip 产品的用户可通过以下渠道获得帮助:

- 代理商或代表
- 当地销售办事处
- 应用工程师 (ESE)
- 技术支持

客户应联系其代理商、代表或 ESE 寻求支持。当地销售办事处也可为客户提供帮助。本文档后附有销售办事处的联系方式。

也可通过 www.microchip.com/support 获得网上技术支持。

Microchip 器件代码保护功能

请注意以下有关 Microchip 产品代码保护功能的要点:

- Microchip 的产品均达到 Microchip 数据手册中所述的技术规范。
- Microchip 确信: 在正常使用且符合工作规范的情况下, Microchip 系列产品非常安全。
- Microchip 注重并积极保护其知识产权。严禁任何试图破坏 Microchip 产品代码保护功能的行为, 这种行为可能会违反《数字千年版权法案》(Digital Millennium Copyright Act)。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是“牢不可破”的。代码保护功能处于持续发展。Microchip 承诺将不断改进产品的代码保护功能。

法律声明

提供本文档的中文版本仅为了便于理解。请勿忽视文档中包含的英文部分, 因为其中提供了有关 Microchip 产品性能和使用情况的有用信息。Microchip Technology Inc. 及其分公司和相关公司、各级主管与员工及事务代理机构对译文中可能存在的任何差错不承担任何责任。建议参考 Microchip Technology Inc. 的英文原版文档。

本出版物及其提供的信息仅适用于 Microchip 产品, 包括设计、测试以及将 Microchip 产品集成到您的应用中。以其他方式使用这些信息都将被视为违反条款。本出版物中的器件应用信息仅为您提供便利, 将来可能会发生更新。如需额外的支持, 请联系当地的 Microchip 销售办事处, 或访问 <https://www.microchip.com/en-us/support/design-help/client-supportservices>。

Microchip “按原样”提供这些信息。Microchip 对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对非侵权性、适销性和特定用途的适用性的暗示担保，或针对其使用情况、质量或性能的担保。

在任何情况下，对于因这些信息或使用这些信息而产生的任何间接的、特殊的、惩罚性的、偶然的或间接的损失、损害或任何类型的开销，Microchip 概不承担任何责任，即使 Microchip 已被告知可能发生损害或损害可以预见。在法律允许的最大范围内，对于因这些信息或使用这些信息而产生的所有索赔，Microchip 在任何情况下所承担的全部责任均不超出您为获得这些信息向 Microchip 直接支付的金额（如有）。如果将 Microchip 器件用于生命维持和/或生命安全应用，一切风险由买方自负。买方同意在由此引发任何一切损害、索赔、诉讼或费用时，会维护和保障 Microchip 免于承担法律责任。除非另外声明，在 Microchip 知识产权保护下，不得暗或以其他方式转让任何许可证。

商标

Microchip 的名称和徽标组合、Microchip 徽标、Adaptec、AnyRate、AVR、AVR 徽标、AVR Freaks、BesTime、BitCloud、CryptoMemory、CryptoRF、dsPIC、flexPWR、HELDO、IGLOO、JukeBlox、KeeLoq、Kleer、LANCheck、LinkMD、maXStylus、maXTouch、MediaLB、megaAVR、Microsemi、Microsemi 徽标、MOST、MOST 徽标、MPLAB、OptoLyzer、PIC、picoPower、PICSTART、PIC32 徽标、PolarFire、Prochip Designer、QTouch、SAM-BA、SenGenuity、SpyNIC、SST、SST 徽标、SuperFlash、Symmetricom、SyncServer、Tachyon、TimeSource、tinyAVR、UNI/O、Vectron 及 XMEGA 均为 Microchip Technology Incorporated 在美国和其他国家或地区的注册商标。

AgileSwitch、APT、ClockWorks、The Embedded Control Solutions Company、EtherSynch、Flashtec、Hyper Speed Control、HyperLight Load、IntelliMOS、Libero、motorBench、mTouch、Powermite 3、Precision Edge、ProASIC、ProASIC Plus、ProASIC Plus 徽标、Quiet-Wire、SmartFusion、SyncWorld、Temux、TimeCesium、TimeHub、TimePictra、TimeProvider、TrueTime、WinPath 和 ZL 均为 Microchip Technology Incorporated 在美国的注册商标。

Adjacent Key Suppression、AKS、Analog-for-the-Digital Age、Any Capacitor、AnyIn、AnyOut、Augmented Switching、BlueSky、BodyCom、CodeGuard、CryptoAuthentication、CryptoAutomotive、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、Espresso T1S、EtherGREEN、GridTime、IdealBridge、In-Circuit Serial Programming、ICSP、INICnet、Intelligent Paralleling、Inter-Chip Connectivity、JitterBlocker、Knob-on-Display、maxCrypto、maxView、memBrain、Mindi、MiWi、MPASM、MPF、MPLAB Certified 徽标、MPLIB、MPLINK、MultiTRAK、NetDetach、NVM Express、NVMe、Omniscient Code Generation、PICDEM、PICDEM.net、PICKit、PICtail、PowerSmart、PureSilicon、QMatrix、REAL ICE、Ripple Blocker、RTAX、RTG4、SAM-ICE、Serial Quad I/O、simpleMAP、SimpliPHY、SmartBuffer、SmartHLS、SMART-I.S.、storClad、SQL、SuperSwitcher、SuperSwitcher II、Switchtec、SynchroPHY、Total Endurance、TSHARC、USBCheck、VariSense、VectorBlox、VeriPHY、ViewSpan、WiperLock、XpressConnect 和 ZENA 均为 Microchip Technology Incorporated 在美国和其他国家或地区的商标。

SQTP 为 Microchip Technology Incorporated 在美国的服务标记。

Adaptec 徽标、Frequency on Demand、Silicon Storage Technology、Symmcom 和 Trusted Time 均为 Microchip Technology Inc. 在除美国外的国家或地区的注册商标。

GestIC 为 Microchip Technology Inc. 的子公司 Microchip Technology Germany II GmbH & Co. KG 在除美国外的国家或地区的注册商标。

在此提及的所有其他商标均为各持有公司所有。

© 2021, Microchip Technology Incorporated 及其子公司版权所有。

ISBN: 978-1-5224-8731-9

质量管理体系

有关 Microchip 的质量管理体系的信息，请访问 www.microchip.com/quality。

全球销售及服务中心

美洲	亚太地区	亚太地区	欧洲
公司总部 2355 West Chandler Blvd. Chandler, AZ 85224-6199 电话: 480-792-7200 传真: 480-792-7277 技术支持: www.microchip.com/support 网址: www.microchip.com 亚特兰大 德卢斯, 佐治亚州 电话: 678-957-9614 传真: 678-957-1455 奥斯汀, 德克萨斯州 电话: 512-257-3370 波士顿 韦斯特伯鲁, 马萨诸塞州 电话: 774-760-0087 传真: 774-760-0088 芝加哥 艾塔斯卡, 伊利诺伊州 电话: 630-285-0071 传真: 630-285-0075 达拉斯 阿迪森, 德克萨斯州 电话: 972-818-7423 传真: 972-818-2924 底特律 诺维, 密歇根州 电话: 248-848-4000 休斯顿, 德克萨斯州 电话: 281-894-5983 印第安纳波利斯 诺布尔斯维尔, 印第安纳州 电话: 317-773-8323 传真: 317-773-5453 电话: 317-536-2380 洛杉矶 米慎维荷, 加利福尼亚州 电话: 949-462-9523 传真: 949-462-9608 电话: 951-273-7800 罗利, 北卡罗来纳州 电话: 919-844-7510 纽约, 纽约州 电话: 631-435-6000 圣何塞, 加利福尼亚州 电话: 408-735-9110 电话: 408-436-4270 加拿大 - 多伦多 电话: 905-695-1980 传真: 905-695-2078	澳大利亚 - 悉尼 电话: 61-2-9868-6733 中国 - 北京 电话: 86-10-8569-7000 中国 - 成都 电话: 86-28-8665-5511 中国 - 重庆 电话: 86-23-8980-9588 中国 - 东莞 电话: 86-769-8702-9880 中国 - 广州 电话: 86-20-8755-8029 中国 - 杭州 电话: 86-571-8792-8115 中国 - 香港特别行政区 电话: 852-2943-5100 中国 - 南京 电话: 86-25-8473-2460 中国 - 青岛 电话: 86-532-8502-7355 中国 - 上海 电话: 86-21-3326-8000 中国 - 沈阳 电话: 86-24-2334-2829 中国 - 深圳 电话: 86-755-8864-2200 中国 - 苏州 电话: 86-186-6233-1526 中国 - 武汉 电话: 86-27-5980-5300 中国 - 西安 电话: 86-29-8833-7252 中国 - 厦门 电话: 86-592-2388138 中国 - 珠海 电话: 86-756-3210040	印度 - 班加罗尔 电话: 91-80-3090-4444 印度 - 新德里 电话: 91-11-4160-8631 印度 - 浦那 电话: 91-20-4121-0141 日本 - 大阪 电话: 81-6-6152-7160 日本 - 东京 电话: 81-3-6880-3770 韩国 - 大邱 电话: 82-53-744-4301 韩国 - 首尔 电话: 82-2-554-7200 马来西亚 - 吉隆坡 电话: 60-3-7651-7906 马来西亚 - 槟榔屿 电话: 60-4-227-8870 菲律宾 - 马尼拉 电话: 63-2-634-9065 新加坡 电话: 65-6334-8870 台湾地区 - 新竹 电话: 886-3-577-8366 台湾地区 - 高雄 电话: 886-7-213-7830 台湾地区 - 台北 电话: 886-2-2508-8600 泰国 - 曼谷 电话: 66-2-694-1351 越南 - 胡志明市 电话: 84-28-5448-2100	奥地利 - 韦尔斯 电话: 43-7242-2244-39 传真: 43-7242-2244-393 丹麦 - 哥本哈根 电话: 45-4485-5910 传真: 45-4485-2829 芬兰 - 埃斯波 电话: 358-9-4520-820 法国 - 巴黎 电话: 33-1-69-53-63-20 传真: 33-1-69-30-90-79 德国 - 加兴 电话: 49-8931-9700 德国 - 哈恩 电话: 49-2129-3766400 德国 - 海布隆 电话: 49-7131-72400 德国 - 卡尔斯鲁厄 电话: 49-721-625370 德国 - 慕尼黑 电话: 49-89-627-144-0 传真: 49-89-627-144-44 德国 - 罗森海姆 电话: 49-8031-354-560 以色列 - 若那那市 电话: 972-9-744-7705 意大利 - 米兰 电话: 39-0331-742611 传真: 39-0331-466781 意大利 - 帕多瓦 电话: 39-049-7625286 荷兰 - 德卢内市 电话: 31-416-690399 传真: 31-416-690340 挪威 - 特隆赫姆 电话: 47-72884388 波兰 - 华沙 电话: 48-22-3325737 罗马尼亚 - 布加勒斯特 电话: 40-21-407-87-50 西班牙 - 马德里 电话: 34-91-708-08-90 传真: 34-91-708-08-91 瑞典 - 哥德堡 电话: 46-31-704-60-40 瑞典 - 斯德哥尔摩 电话: 46-8-5090-4654 英国 - 沃金厄姆 电话: 44-118-921-5800 传真: 44-118-921-5820