

Microchip 无线（MiWi™）应用程序 编程接口 —— MiApp

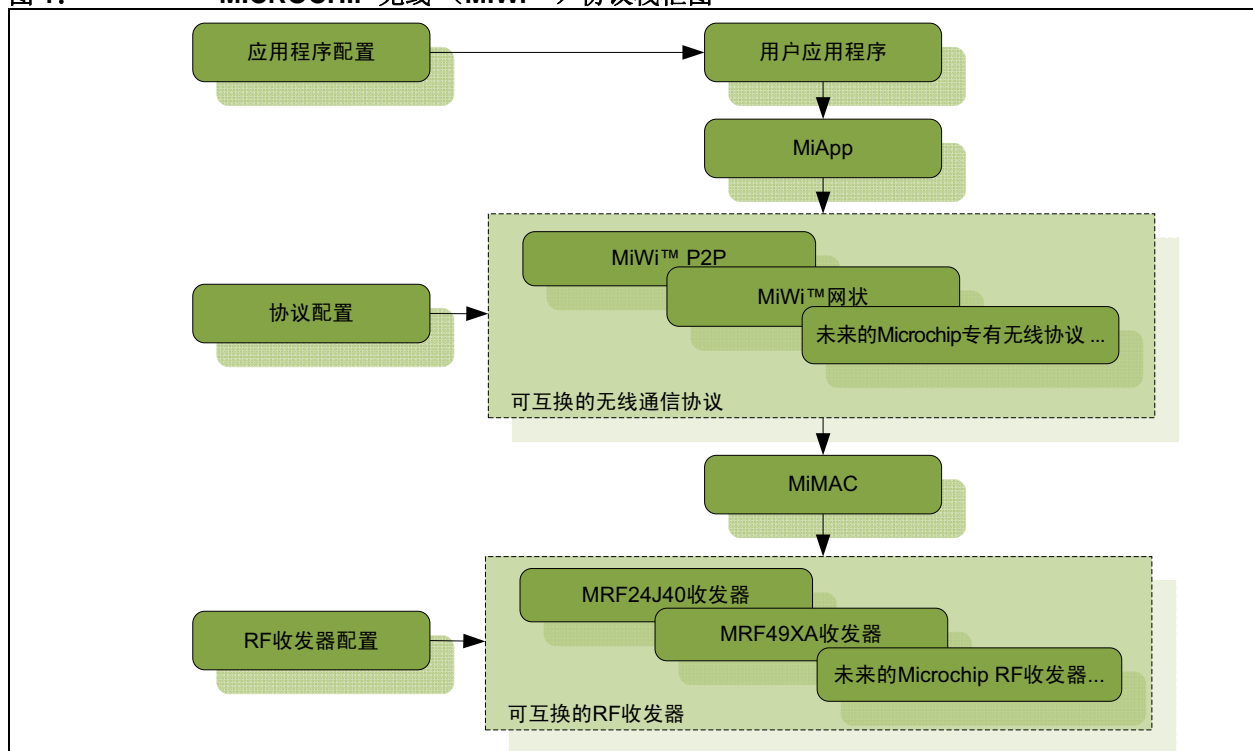
作者: Yifeng Yang
Microchip Technology Inc.

简介

开发短距离、低数据速率的低功耗无线应用并不是一项简单的任务。除了复杂的射频（Radio Frequency, RF）电路设计之外，固件开发过程可能还要求开发人员了解 RF 收发器的细节，以及不同的无线通信协议。Microchip 开发了一种方法来处理困难复杂的 RF 硬件和/或通信协议栈软件开发，使无线开发人员能够专注于开发他们自己的应用程序。这通过应用层中简明却功能强大的通信编程接口实现，它被称为 MiApp，本应用笔记中将对它进行定义。

MiApp 规范定义了应用层和 Microchip 专有无线通信协议之间的编程接口。MiApp 编程接口通过两种方式实现：作为在配置文件中定义的配置参数，以及作为对 Microchip 专有无线协议的一组函数调用。遵从本应用笔记中定义的 MiApp 规范，应用程序可以使用任何 Microchip 专有无线协议。软件开发可以根据应用需求，在针对小型或大型网络的专有 P2P/ 星型拓扑连接协议和全网状专有联网协议之间轻松切换，而在应用层中无需或只需少量修改。

图 1: MICROCHIP 无线（MiWi™）协议栈框图



MiApp 规范以多种方式无线应用开发人员带来好处：

- 无线应用开发可以专注于应用程序本身。复杂的 RF 或协议注意事项则通过 MiApp 编程接口透明地处理。
- MiApp 规范提供了最大程度的灵活性，几乎无需任何额外工作，即可在应用软件开发中的任何阶段选择无线协议，从而极大地降低软件开发的成本。联网功能中的应用需求变化对应用程序开发没有或只有很低的影响。
- MiApp 对 Microchip 无线专有协议使用相同的控制接口。一旦熟悉 MiApp 之后，就可以将这方面的知识应用于另一个应用程序的开发，即使它具有完全不同的联网功能需求。
- 通过与 Microchip 专有协议进行通信，MiApp 可以通过 MiMAC 接口与 Microchip RF 收发器间接进行对话。因此，MiApp 间接使无线应用开发人员可以通过 MiMAC 在 Microchip RF 收发器之间进行切换。这种灵活性又可以进一步降低无线应用项目的开发成本。

特性

MiApp 编程接口具有以下特性：

- 易于学习和使用
- 功能强大的接口，可满足无线应用的大多数要求
- 在 Microchip 专有无线协议之间移植无线应用时，无需或几乎无需额外的工作
- 占用空间很小

注意事项

MiApp 规范旨在支持 Microchip 专有的无线通信协议。通过 MiApp 编程接口实现无线应用之后，也就通过 MiMAC 中的标准化支持 Microchip RF 收发器；MiMAC 是在介质访问控制器（Media Access Controller，MAC）层中定义的模块。

MiMAC 对 Microchip 无线协议和 Microchip RF 收发器之间的接口进行了标准化。MiMAC 使 Microchip RF 收发器可以互换，而无需或几乎无需更改软件应用程序代码。关于 MiMAC 的详细信息，请参见应用笔记 AN1283，《Microchip 无线（MiWi™）介质访问控制器——MiMAC》。

MiMAC 调控 Microchip 专有无线协议的低层接口，而 MiApp 则调控 Microchip 专有无线协议的高层接口。MiMAC 和 MiApp 配合工作，为无线应用开发人员提供了最大程度的灵活性，使其可以在软件开发的任何阶段选择 RF 收发器和无线通信协议，从而进一步将软件开发成本降到最低。图 1 中的框图显示了 Microchip 无线（MiWi™）协议栈所包含的内容。

共存在三层配置：应用程序、协议栈和 RF 收发器。根据器件的硬件设计、器件在应用和网络中的角色，相同应用程序中的应用程序配置在不同器件之间可能会不同。无线应用开发人员通常在应用层中进行大多数的配置。协议配置可以微调协议栈的行为。大多数协议栈配置用于定义所选无线协议的时序和路由机制。收发器配置用于定义 RF 收发器的频段、数据速率和其他 RF 相关功能。协议和 RF 收发器配置的默认设置可以不进行任何修改而用于应用程序。但应用程序配置通常需要进行更改，以适应不同无线应用的需求。

MiApp 概述

如前面所述，MiApp 规范定义了两个部分：

- 在配置文件中定义的配置参数
- Microchip 专有无线通信协议的函数调用的格式

配置文件包含应在编译之前设置的参数。一般来说，配置文件中定义两方面的信息：

- 硬件定义：包括 MCU 硬件资源、外设定义和 RF 收发器控制引脚的定义。对于支持 Microchip RF 收发器的几种 Microchip 标准演示板，已经定义了默认硬件。在这些情况下，这些演示板的定义会自动引入所有硬件定义。
- 软件定义：这些定义控制要编译到固件 hex 文件中的代码段。软件定义包括 Microchip 专有无线协议的选择，以及 Microchip RF 收发器与各种功能的选择。在该类别中进行正确定义可以确保最小的固件占用空间，并提供所需的协议功能。

应用程序编程接口（Application Programming Interface，API）是 Microchip 专有无线通信协议与无线开发人员的应用程序之间的函数调用。一般来说，应用程序接口必须纯净、简明、易于理解，并且功能强大。

API 具有 5 种类别的接口：

- 初始化接口，使无线应用开发人员可以正确初始化已在配置文件中选择的 Microchip 专有无线协议。
- 握手接口，使无线节点可以发现并连接对等端，或加入网络。
- 用于发送报文的接口，使应用开发人员可以通过空气从当前节点向目标受众发送信息。
- 用于接收报文的接口，使应用开发人员可以通过空气从其他器件接收信息。
- 一些特殊功能，通过环境噪声控制和节能确保无线节点的最佳工作条件。

MiApp 配置文件

在 MiApp 配置文件的两种配置中，硬件定义高度依赖于演示板、MCU 和 RF 收发器选择。硬件定义可以分为以下类别：

- 演示板上的 I/O——按钮、LED 和串行端口等
- MCU 系统资源——定时器和中断等
- MCU 和 RF 收发器之间的互连

硬件定义主要与无线应用系统设计的硬件选择相关联。它们更多地依赖于硬件而不是软件，并且会因不同的设计而有所不同。因此，对于这些硬件定义，MiApp 并没有一套标准。

选择性编译配置用于从可用功能列表中选择所需的功能。通过使用选择性编译，应用开发人员能够配置 Microchip 专有无线协议，使用尽可能最少的系统资源执行所需的功能。表 1 介绍了可能的选择性编译配置，以及这些选择的范围、值和功能。

表 1：配置文件中的软件定义

定义示例	功能	限制
#define PROTOCOL_MIWI #define PROTOCOL_P2P	选择要在无线应用中使用的 Microchip 无线协议。	在任意时刻，只能定义一种协议。
#define MRF24J40 #define MRF49XA	选择要在无线应用中使用的 Microchip RF 收发器。	在任意时刻，只能定义一种收发器。
#define TX_BUFFER_SIZE 40	定义要发送的应用程序有效载荷的最大大小（不包括所有协议报头）。	对于可以发送的缓冲区的大小，可能存在 RF 收发器硬件限制。硬件限制包括所有协议报头。

表 1: 配置文件中的软件定义 (续)

定义示例	功能	限制
#define RX_BUFFER_SIZE 40	要接收的应用程序有效载荷的最大大小 (不包括所有协议报头)。	对于可以接收的缓冲区的大小, 可能存在 RF 收发器硬件限制。硬件限制包括所有协议报头。
#define CONNECTION_SIZE 10	连接表的大小。决定节点可以连接的最大器件数量。	取决于可用的 MCU RAM。
#define ADDITIONAL_NODE_ID_SIZE 0	定义在握手过程中附加到数据包的附加信息的大小。主要用于在应用层中识别节点。	附加的节点标识符在 Microchip 专有协议中没有任何作用。但是, 它在应用程序中可能具有重要作用。以简单的灯和开关为例, 两盏灯对彼此之间的互连不感兴趣, 两个开关也是这样。使用附加的节点标识符使应用程序可以识别节点在应用中的作用, 从而使开关仅与灯相连接。
#define ENABLE_PA_LNA	使 RF 收发器可以使用外部功率放大器和 / 或低噪声放大器。	用于可以控制外部 PA 和 / 或 LNA 的 RF 收发器。
#define ENABLE_HAND_SHAKE	使 Microchip 专有无线协议可以自动与对等端建立连接。	握手过程使两个无线节点可以相互知悉。在其他协议中, 该过程也称为“配对”。无握手功能的应用程序仅使用广播来交换报文。
#define ENABLE_SLEEP	使 RF 收发器可以在空闲时进入休眠状态, 以节省电能。	休眠模式取决于 RF 收发器的功能。
#define ENABLE_ED_SCAN	使 Microchip 专有无线协议和 RF 收发器可以执行能量检测扫描。	能量扫描取决于 RF 收发器的功能。
#define ENABLE_ACTIVE_SCAN	使 Microchip 专有无线协议可以执行主动扫描, 以发现邻近的节点和网络。	主动扫描用于搜索邻近的同类现有无线设备。可以使用主动扫描来决定连接到哪个器件。
#define ENABLE_SECURITY	使 Microchip 专有协议可以保护传输的数据包。	安全引擎、安全模式和密钥在 RF 收发器的配置文件中定义, 因为安全性作为 MiMAC 的一部分进行定义。
#define ENABLE_INDIRECT_MESSAGE	使无线节点可以缓存休眠器件的报文, 并在休眠器件唤醒并要求获取报文时传递它们。	只有不进入休眠状态的无线节点可以缓存休眠节点的报文。可以缓存的报文数量取决于可用的 MCU RAM。
#define RFD_WAKEUP_INTERVAL 5	定义 RFD 器件的唤醒时间间隔 (以秒为单位)。	仅在使能间接报文时有效。该定义用于总是处于唤醒状态的器件, 用以跟踪间接报文的超时。休眠器件的休眠时间取决于主机 MCU 的 WDT 设置。

表 1：配置文件中的软件定义（续）

定义示例	功能	限制
#define ENABLE_BROADCAST	使无线节点可以处理休眠器件的广播报文。	只有不进入休眠状态的无线节点可以缓存休眠节点的报文。
#define ENABLE_FREQUENCY_AGILITY	使 Microchip 专有无线协议可以执行频率捷变过程。	N/A
#define HARDWARE_SPI	使 MCU 可以使用硬件 SPI 与收发器进行通信。	定义 HARDWARE_SPI 使 MCU 可以使用硬件 SPI 与收发器进行通信。否则，MCU 可以使用位响应（bit-bang）来模拟与收发器的 SPI 通信。
#define NWK_ROLE_COORDINATOR #define NWK_ROLE_END_DEVICE	定义当前器件在网络中的角色。	该配置仅用于网络协议。P2P 协议（如 MiWi™ P2P）不使用该配置。
#define TARGET_SMALL	最大程度减小 Microchip 专有无线协议的占用空间。	最大程度减小协议占用空间时，可能不支持 Microchip 专有无线协议的某些功能。
#define ENABLE_NETWORK_FREEZER	使 Microchip 专有无线协议可以存储关键的网路参数，即便断电，也能在重启后恢复为原始网络设置。	需要非易失性存储器，可以是 MCU 数据 EEPROM、外部 EEPROM 或编程空间。网络规模和所选无线协议将决定所需非易失性存储器的总容量。

MiApp 函数接口

除了配置文件中的选项之外，应用层还可以使用函数调用与 Microchip 专有无线协议层进行通信，从而间接控制收发器执行无线通信。应用层对协议层的函数调用具有 5 种类别：

- 初始化
- 握手
- 发送报文
- 接收报文
- 特殊功能

以下几节详细介绍了函数接口，以及关联的结构定义。

初始化

要初始化 RF 收发器和协议栈，应用层只需通过调用 *ProtocolInit* 函数来触发初始化过程。完整函数格式为：

```
BOOL MiApp_ProtocolInit(BOOL bNetworkFreezer);
```

初始化只有一个参数。输入布尔值决定在初始化过程中是否执行网络冻结器功能。执行网络冻结器功能时，将会恢复存储在非易失性存储器中的原有网络设置。返回值是一个布尔值，指示操作是否成功。

除了正常初始化过程之外，无线应用可能还需要在工作过程中更改发送或接收频率。MiApp 定义了以下函数，用于根据预定义的信道来更改 RF 收发器的工作频率。每个信道根据规范或不同工作频段下的 RF 收发器设置来定义频率。函数格式为：

```
BOOL MiApp_SetChannel(BYTE Channel);
```

惟一的输入参数是要设置的信道。返回值指示操作是否成功。可能的信道编号为 0 至 31。根据 RF 收发器、频段和数据速率，0 至 31 的所有信道并不是在所有条件下都有效。如果输入信道在当前条件下是无效的，则不会改变工作信道，并返回 FALSE 来表示失败。

握手

除非在制造时进行硬编码，否则在大多数应用中，两个通信端点都需要先进行导引，之后才能在一对无线节点之间单播报文。网络协议的导引有时称为加入网络。对于 P2P 协议，该过程还可以称为配对。由于该策略不关注任何特定拓扑或协议，该过程通常称为握手阶段。如果没有握手过程，无线节点将只能使用广播，而广播会将源无线电射程中的每个无线节点视为受众，以进行相互通信。

以下握手函数调用可供应用层使用：

- MiApp_StartConnection
- MiApp_SearchConnection
- MiApp_RemoveConnection
- MiApp_ConnectionMode

MiApp_StartConnection

函数调用 MiApp_StartConnection 使无线节点可以用不同的方式启动操作。有三种方法可以启动 PAN：直接在某个特定信道上启动 PAN，或在执行两种信道评估中的任意一种之后启动 PAN。完整函数格式为：

```
BOOL StartConnection(BYTE Mode, BYTE
ScanDuration, DWORD ChannelMap,
BYTE *DestAddr);
```

该函数调用的返回值指示操作是否成功。

输入参数 Mode 用于指定启动 PAN 的模式。可能的模式有：

- **START_CONN_DIRECT**：不进行任何信道评估的情况下在当前信道启动连接。
- **START_CONN_ENERGY_SCN**：在能量检测扫描之后启动连接，并且在能量最低的信道启动 PAN。
- **START_CONN_CS_SCN**：在载波侦听扫描之后启动连接，并在检测到的载波侦听信号最低的信道启动 PAN。

对于不支持能量检测和 / 或载波侦听扫描的收发器，这些模式将无效，如果在输入参数中指定了此类模式，函数将在不进行任何信道评估的情况下启动 PAN。

输入参数 ScanDuration 用于指定执行信道评估的最长时间。如果可以执行多次扫描，则应对扫描周期应用最大保持方法。在启动模式未指定信道评估的情况下，该输入参数将被丢弃。输入参数 ScanDuration 的值符合 IEEE 802.15.4™ 规范中的定义。它的范围为 1 至 14。公式 1 给出了用于计算扫描持续时间的公式。

公式 1：扫描持续时间计算

$$\text{扫描时间 (us)} = 960 * (2^{\text{ScanDuration}} + 1)$$

如该公式所示，ScanDuration 为 10 时，时间约为 1 秒。增 1 之后时间大约为两倍，减 1 之后时间大约减少一半。

输入参数 ChannelMap 用于指定要在该过程中扫描的信道。ChannelMap 定义为一个 4 字节双字。它使用位映射来表示信道 0 至信道 31。当双字中的某位置 1 时，意味着相应信道将执行信道评估。例如，如果输入参数 ChannelMap 的 bit 0 置 1，则信道 0 将执行信道评估。要对所有可用信道执行信道评估，输入参数 ChannelMap 需为 0xFFFFFFFF。

MiApp_SearchConnection

函数调用 MiApp_SearchConnection 用于搜索并发现邻近的现有对等无线节点。该过程也称为主动扫描。在某些应用中，该步骤用于通知器件是应该启动 PAN 还是选择要加入的 PAN。如果启动了某个 PAN，则可以使用该过程来决定选择哪个 PAN 标识符。如果器件加入某个 PAN，则该过程用于选择加入哪个 PAN 和哪个器件。

完整函数格式为：

```
BYTE SearchConnection(BYTE ScanDuration,
DWORD ChannelMap);
```

该函数的返回值指示所返回 PAN 的总数。返回 PAN 的结果将以结构 **ACTIVE_SCAN_RESULT** 的格式存储在全局变量中，该结构定义如下：

```
typedef struct
{
    BYTE Channel;
    BYTE Address[];
    WORD_VAL PANID;
    BYTE RSSI;
    BYTE LQI;
    union
    {
        BYTE Val;
        struct
        {
            BYTE Role: 2;
            BYTE Sleep: 1;
            BYTE SecurityEn: 1;
            BYTE RepeatEn: 1;
            BYTE AllowJoin: 1;
            BYTE Direct: 1;
            BYTE altSrcAddr: 1;
        } bits;
    } Capability
} ACTIVE_SCAN_RESULT;
```

在该结构中，元素 **Address** 指示对主动扫描作出响应的器件的地址。

元素 **PANID** 指示 PAN 标识符（如可用）。PAN 标识符用于指定网络 ID。

元素 **RSSI** 和 **LQI** 分别指示响应信号的强度和质量。该信息可能并不是对于所有 RF 收发器都可用。

元素 **Capability** 包含有关送回响应的器件的功能信息。它是功能的位图，以联合体的形式定义。根据在应用层下使用的协议，功能信息可能不可用。

MiApp_RemoveConnection

函数调用 **MiApp_RemoveConnection** 使当前节点可以断开某些连接。完整函数格式为：

```
void MiApp_RemoveConnection(BYTE
ConnectionIndex);
```

该函数没有返回值。输入参数 **ConnectionIndex** 用于指定要删除的对等节点在连接表中的索引。如果 **ConnectionIndex** 为 **0xFF**，器件将会删除所有连接并离开网络。在网络协议中，这也意味着器件的所有子节点都会离开网络。在 **ConnectionIndex** 指向网络协议中某个父节点的情况下，当前节点及其所有子节点都必须离开网络。如果连接索引指向的节点不是当前节点的父节点，则会删除连接，器件将保留在 PAN 中。

MiApp_EstablishConnection

函数调用 **MiApp_EstablishConnection** 用于与一个或多个器件建立连接。完整函数格式为：

```
BYTE EstablishConnection(BYTE
ActiveScanIndex, BYTE Mode);
```

该函数调用将返回一个字节，指示新的对等节点在连接表中的索引。如果返回值为 **0xFF**，则意味着建立连接的过程在尝试预定义重试次数之后发生失败。如果在该过程中建立了多个连接，返回值将为其中一个连接的连接表索引。

参数 **ActiveScanIndex** 是要建立连接的节点在主动扫描结果表中的索引。如果值为 **0xFF**，协议将尝试与所有器件建立连接。因此，在该过程中可能会建立多个连接。

参数 **Mode** 用于指定连接模式。定义了两种模式：

- **MODE_DIRECT**: 该模式直接建立无线电射程中的连接。P2P 协议栈使用该模式来建立连接，而网络协议使用它来与父节点建立连接，从而加入网络。
- **MODE_INDIRECT**: 网络协议使用该模式在网络上建立具有一个或多个跃点的连接。所连接的器件可能处于也可能不处于请求节点的无线电射程内。在 **MiWi 应用笔记（AN1066）** 中，如果输入参数 **ActiveScanIndex** 为 **0xFF**，则这种连接也定义为群集套接字。

MiApp_ConnectionMode

函数调用 **MiApp_ConnectionMode** 用于设置连接模式，而连接模式则调控当前无线节点是否能够接受来自新器件的直接连接。完整函数格式为：

```
void MiApp_ConnectionMode(BYTE Mode);
```

该函数没有返回值。输入参数 “Mode” 用于指示工作模式。可能的工作模式有：

- **ENABLE_ALL_CONN**: 该模式使能任何条件下的连接。这是应用程序启动时的默认模式。
- **ENABLE_PREV_CONN**: 该模式仅使能原有连接。如果连接请求来自已处于连接表中的节点，则允许连接请求。否则，请求将被忽略。
- **ENABLE_ACTIVE_SCAN_RSP**: 该模式使能当前节点响应任何主动扫描请求，以标识其自身。
- **DISABLE_ALL_CONN**: 该模式将禁止所有连接请求，包括主动扫描。

ENABLE_CONN 至 DISABLE_ALL_CONN 的连接权限从高到低。权限较高的模式具有权限较低模式的所有权限。

发送报文

无线节点最重要的功能是进行通信，即发送和接收数据。所有协议都为数据传输保留了缓冲区，其大小大于等于配置文件中定义的 **TX_BUFFER_SIZE**。定义了两个函数来管理堆栈中的 **TX** 缓冲区：

```
void MiApp_FlushTx(void);  
void MiApp_WriteData(BYTE OneByteTxData);
```

函数 **MiApp_FlushTx** 用于复位堆栈中的发送缓冲区的指针。它没有任何参数和返回值。

函数 **MiApp_WriteData** 用于在堆栈中的发送缓冲区中填充一个数据字节。惟一的输入参数是要填充到发送缓冲区中的一个数据字节。

通常情况下，需要先调用 **MiApp_FlushTx** 来复位缓冲区指针。然后，多次调用 **MiApp_WriteData** 来填充发送缓冲区，每次一个字节。

发送缓冲区填满之后，下一步是触发要由协议层发送的报文。有三种方法可以发送报文：

- 广播
- 按节点在连接表中的索引对节点进行单播
- 按节点地址（永久地址或备用网络地址）对节点进行单播

广播面向所有器件的报文，不论其地址如何。下面给出了广播的完整函数格式：

```
BOOL MiApp_BroadcastMessage(BOOL SecEn);
```

该函数调用的返回值指示发送是否成功。惟一的输入参数 **SecEn** 是一个布尔值，指定有效载荷是否需要加以保护。

单播以单个器件作为目标。有两种方法可以单播报文：通过连接表中的索引表示目标，或明确给出目标地址（永久地址或网络地址）。

使用连接表索引作为参数的单播的完整函数格式为：

```
BOOL MiApp_UnicastConnection(BYTE  
ConnectionIndex, BOOL SecEn);
```

该函数调用的返回值指示发送是否成功。输入参数 **ConnectionIndex** 是目标节点在连接表中的索引。输入参数 **SecEn** 是一个布尔值，指示有效载荷是否需要加以保护。

使用目标地址作为参数的单播的完整函数格式如下所示：

```
BOOL MiApp_UnicastAddress(BYTE *Address,  
BOOL PermanentAddr, BOOL SecEn);
```

该函数调用的返回值指示发送是否成功。

输入参数 **Address** 是指向目标地址的指针。

输入布尔值参数 **PermanentAddr** 指示目标地址是永久地址还是备用网络地址。对于星型或 P2P 拓扑协议，仅使用永久地址，因而输入参数 **PermanentAddr** 不起作用。

输入参数 **SecEn** 指示有效载荷是否需要加以保护。

接收报文

收发器的另一个重要功能是接收报文。应用层需要知道何时接收到报文、报文的内容，有时还需要知道如何接收报文。应用层还需要丢弃报文，以便可以释放资源，从而接收并处理新的报文。要使用所述的流程，需要定义两个函数调用和一个结构。

MiApp_MessageAvailable

函数调用 **MiApp_MessageAvailable** 没有任何输入参数，它返回一个布尔值，指示是否已接收到新的报文，以及是否可供在应用层中处理。完整函数格式为：

```
BOOL MessageAvailable(void);
```

接收报文的数据结构

转发到应用层的所有接收报文都存储在使用 **RECEIVED_MESSAGE** 格式定义的全局变量中，如下所示：

```
typedef struct
{
    union
    {
        BYTE Val;
        struct
        {
            BYTE broadcast: 1;
            BYTE ackReq: 1;
            BYTE secEn: 1;
            BYTE repeat: 1;
            BYTE command: 1;
            BYTE srcPrnt: 1;
            BYTE dstPrnt: 1;
            BYTE altSrcAddr: 1;
        } bits
    } flags;

    BYTE *SourceAddress;
    BYTE *Payload;
    BYTE PayloadSize;
    BYTE RSSI;
    BYTE LQI;
} RECEIVED_MESSAGE;
```

根据所使用的收发器和 **Microchip** 专有协议，并不是结构中的所有元素都是有效的。

MiApp_DiscardMessage

函数调用 **MiApp_DiscardMessage** 没有任何输入参数，不返回任何值。应用层通过调用该函数来通知 **Microchip** 专有无线协议层，当前数据包已完成处理，它已准备处理下一个数据包。完整函数格式为：

```
void DiscardMessage(void);
```

特殊功能

一些收发器具有一些特殊的功能，使协议栈可以更稳健、更能够适应环境。

噪声检测扫描

噪声检测扫描使收发器可以检测环境中的噪声级别。在未占用频率下启动新的 **PAN**，以及确定是否需要信道跳跃和跳跃到哪个信道，这些都是极具价值的。

完整函数格式为：

```
BYTE MiApp_NoiseDetection(DWORD ChannelMap,
    BYTE ScanDuration, BYTE DetectionMode, BYTE
    *NoiseLevel);
```

函数调用 **MiApp_NoiseDetection** 会返回噪声最低的信道。该函数具有 4 个函数参数：

- **ChannelMap**：该输入参数用于定义要扫描的信道的位图。对于每个收发器，支持的信道数量是不同的；因此，并不是输入参数 **ChannelMap** 中的所有位图都是有效的。
- **ScanDuration**：该输入参数用于定义每个信道上的噪声检测的总时间。使用最大保持机制来检测每个信道上的噪声级别。输入参数 **ScanDuration** 遵循 **IEEE 802.15.4™** 规范，本应用笔记在前面对它进行了详细介绍，并给出了一个计算实际时间的公式。
- **DetectionMode**：该输入参数用于定义要使用的检测模式：能量检测或载波侦听检测。并不是所有 **RF** 收发器都支持所有检测模式。
- **NoiseLevel**：该输出参数返回最佳信道（即该函数调用的返回值对应的信道）上的噪声级别。该输出参数使应用层可以查看最佳信道上的噪声级别。**NoiseLevel** 参数值越高，说明环境的噪声越高。

收发器电源状态

为了支持通过电池供电的无线节点，需要将无线电收发器设置为不同的电源状态，或将它置为休眠状态并定期唤醒它。定义了函数调用 `MiApp_TransceiverPowerState` 来实现这一目标：

```
BYTE MiApp_TransceiverPowerState  
(BYTE Mode);
```

该函数调用的惟一输入参数是工作模式。预定义的工作模式有：

- **POWER_STATE_SLEEP:** 将收发器置于休眠模式。
- **POWER_STATE_WAKEUP:** 不发送任何数据请求而唤醒收发器。
- **POWER_STATE_WAKEUP_DR:** 唤醒收发器，然后向其主要关联器件发出数据请求，要求提供输入数据。

函数调用 `MiApp_TransceiverPowerState` 返回一个字节，指示操作的状态。预定义的操作状态返回值有：

- **SUCCESS:** 指示每个操作已成功。
- **ERR_TRX_FAIL:** 指示休眠或唤醒收发器的请求失败。
- **ERR_TXFAIL:** 指示发送数据的请求失败。只有工作模式为 `WAKE_DR` 时，该选项才可用。
- **ERR_RXFAIL:** 指示从父节点接收数据的请求失败。只有工作模式为 `WAKE_DR` 时，该选项才可用。

频率捷变

频率捷变是指在工作期间进行信道跳跃，以避开特定频率下的持续性噪声的功能。

并不是所有收发器和协议都支持频率捷变。频率捷变函数对于应用程序接口是可选的。

有两个函数可用于建立频率捷变。一个函数用于启动频率捷变过程。另一个函数用于在以下情况下同步连接：由于在通信的另一端执行频率捷变而丢失通信。

启动频率捷变过程的完整函数格式为：

```
BOOL MiApp_InitChannelHopping  
(DWORD ChannelMap);
```

函数调用 `MiApp_InitChannelHopping` 的返回值指示信道跳跃操作是否成功。`ChannelMap` 输入参数指示要移到的可用信道。`ChannelMap` 参数是可能信道的位图。如果某个信道可用，相应的位（对应信道 `n` 为第 `n` 个位）将置 1；否则，它将会被清零。

`MiApp` 规范未定义何时启动信道跳跃。触发事件可以是连续发送 / 接收失败，也可以只是定期搜索最佳频率来运行无线应用。何时启动信道跳跃过程由无线应用决定。`MiApp` 规范提供了 Microchip 专有无线协议的相应接口，用来执行这些由应用层决定的操作。

完成信道跳跃过程之后，某些无线节点（特别是那些在空闲时处于休眠状态的节点）有可能并不知道网络已移到另一个信道。需要定义一个函数来重新同步连接：

```
BOOL MiApp_ResyncConnection(BYTE  
ConnectionIndex, DWORD ChannelMap);
```

函数 `MiApp_ResyncConnection` 的返回值指示重新同步过程是否成功。它具有两个输入参数：`ConnectionIndex` 和 `ChannelMap`。`ConnectionIndex` 是要同步的器件在连接表中的索引。参数 `ChannelMap` 是可能同步到的信道的位图。

结论

对于正在寻求短距离、低数据速率解决方案的无线应用开发人员，点到点通信和跨越几个跃点传递报文的要求是不同的。Microchip 的 MiApp 规范提供了一个低成本、低复杂度的解决方案，可以满足几乎所有这些应用的需求。它使无线应用开发人员在移植时无需或几乎无需进行修改，即可使用 Microchip 专有无线协议。借助低层的 MiMAC，它可以间接允许开发人员选择 Microchip 支持的任何现有和未来 RF 收发器。强烈建议本应用笔记的读者也阅读应用笔记《Microchip 无线 (MiWi™) 介质访问控制器——MiMAC》(AN1283)，了解 Microchip 针对无线应用提供的整体解决方案。将低层的 MAC 层标准化为 MiMAC，高层的应用层标准化为 MiApp，可以为无线应用开发人员在软件开发过程中提供最大程度的灵活性。

参考资料

IEEE 标准 802.15.4-2003，适用于低速率无线个人局域网 (WPAN) 的无线介质访问控制 (MAC) 和物理层 (PHY) 规范。纽约：IEEE，2003。

《Microchip 无线 (MiWi™) 介质访问控制器——MiMAC》，Yifeng Yang，Microchip Technology Inc. 2009。

MRF24J40 数据手册——《IEEE 802.15.4™ 2.4GHz RF 收发器》(DS39776B_CN)，Microchip Technology Inc. 2008

MRF49XA 数据手册——《ISM 波段 Sub-GHz 射频收发器》(DS70590B_CN)，Microchip Technology Inc. 2009

AN1066，《MiWi™ 无线网络协议栈》(DS01066A_CN)，David Flowers 和 Yifeng Yang，Microchip Technology Inc.，2007

AN1204，《Microchip MiWi™ P2P 无线协议》(DS01204A_CN)，Yifeng Yang，Microchip Technology Inc.，2008

版本历史

版本 A（2009 年 7 月）

- 这是本文档的初始版本。

请注意以下有关 Microchip 器件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指标。
- Microchip 确信：在正常使用的情况下，Microchip 系列产品是当今市场上同类产品中最安全的产品之一。
- 目前，仍存在着恶意、甚至是非法破坏代码保护功能的行为。就我们所知，所有这些行为都不是以 Microchip 数据手册中规定的操作规范来使用 Microchip 产品的。这样做的人极可能侵犯了知识产权。
- Microchip 愿与那些注重代码完整性的客户合作。
- 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中。Microchip 承诺将不断改进产品的代码保护功能。任何试图破坏 Microchip 代码保护功能的行为均可视为违反了《数字器件千年版权法案（Digital Millennium Copyright Act）》。如果这种行为导致他人在未经授权的情况下，能访问您的软件或其他受版权保护的成果，您有权依据该法案提起诉讼，从而制止这种行为。

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。

本出版物中所述的器件应用信息及其他类似内容仅为您提供便利，它们可能由更新之信息所替代。确保应用符合技术规范，是您自身应负的责任。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，包括但不限于针对其使用情况、质量、性能、适销性或特定用途的适用性的声明或担保。Microchip 对因这些信息及使用这些信息而引起的后果不承担任何责任。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伤害、索赔、诉讼或费用时，会维护和保障 Microchip 免于承担法律责任，并加以赔偿。在 Microchip 知识产权保护下，不得暗中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、dsPIC、KEELOQ、KEELOQ 徽标、MPLAB、PIC、PICmicro、PICSTART、PIC³² 徽标、rfPIC 和 UNI/O 均为 Microchip Technology Inc. 在美国和其他国家或地区的注册商标。

FilterLab、Hampshire、HI-TECH C、Linear Active Thermistor、MXDEV、MXLAB、SEEVAL 和 The Embedded Control Solutions Company 均为 Microchip Technology Inc. 在美国的注册商标。

Analog-for-the-Digital Age、Application Maestro、BodyCom、chipKIT、chipKIT 徽标、CodeGuard、dsPICDEM、dsPICDEM.net、dsPICworks、dsSPEAK、ECAN、ECONOMONITOR、FanSense、HI-TIDE、In-Circuit Serial Programming、ICSP、Mindi、MiWi、MPASM、MPLAB Certified 徽标、MPLIB、MPLINK、mTouch、Omniscient Code Generation、PICC、PICC-18、PICDEM、PICDEM.net、PICkit、PICtail、REAL ICE、rfLAB、Select Mode、Total Endurance、TSHARC、UniWinDriver、WiperLock 和 ZENA 均为 Microchip Technology Inc. 在美国和其他国家或地区的商标。

SQTP 是 Microchip Technology Inc. 在美国的服务标记。

在此提及的所有其他商标均为各持有公司所有。

© 2012, Microchip Technology Inc. 版权所有。

ISBN: 978-1-62076-548-7

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
== ISO/TS 16949 ==

Microchip 位于美国亚利桑那州 Chandler 和 Tempe 与位于俄勒冈州 Gresham 的全球总部、设计和晶圆生产厂及位于美国加利福尼亚州和印度的设计中心均通过了 ISO/TS-16949:2009 认证。Microchip 的 PIC[®] MCU 与 dsPIC[®] DSC、KEELOQ[®] 跳码器件、串行 EEPROM、单片机外设、非易失性存储器和模拟产品严格遵守公司的质量体系流程。此外，Microchip 在开发系统的设计和生产方面的质量体系也已通过了 ISO 9001:2000 认证。

全球销售及服务中心

美洲

公司总部 **Corporate Office**
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 1-480-792-7200
Fax: 1-480-792-7277

技术支持:
<http://www.microchip.com/support>

网址: www.microchip.com

亚特兰大 Atlanta
Duluth, GA
Tel: 1-678-957-9614
Fax: 1-678-957-1455

波士顿 Boston
Westborough, MA
Tel: 1-774-760-0087
Fax: 1-774-760-0088

芝加哥 Chicago
Itasca, IL
Tel: 1-630-285-0071
Fax: 1-630-285-0075

克里夫兰 Cleveland
Independence, OH
Tel: 1-216-447-0464
Fax: 1-216-447-0643

达拉斯 Dallas
Addison, TX
Tel: 1-972-818-7423
Fax: 1-972-818-2924

底特律 Detroit
Farmington Hills, MI
Tel: 1-248-538-2250
Fax: 1-248-538-2260

印第安纳波利斯 Indianapolis
Noblesville, IN
Tel: 1-317-773-8323
Fax: 1-317-773-5453

洛杉矶 Los Angeles
Mission Viejo, CA
Tel: 1-949-462-9523
Fax: 1-949-462-9608

圣克拉拉 Santa Clara
Santa Clara, CA
Tel: 1-408-961-6444
Fax: 1-408-961-6445

加拿大多伦多 Toronto
Mississauga, Ontario,
Canada
Tel: 1-905-673-0699
Fax: 1-905-673-6509

亚太地区

亚太总部 **Asia Pacific Office**
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

中国 - 北京
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

中国 - 成都
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

中国 - 重庆
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

中国 - 杭州
Tel: 86-571-2819-3187
Fax: 86-571-2819-3189

中国 - 香港特别行政区
Tel: 852-2401-1200
Fax: 852-2401-3431

中国 - 南京
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

中国 - 青岛
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

中国 - 上海
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

中国 - 沈阳
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

中国 - 深圳
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

中国 - 武汉
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

中国 - 西安
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

中国 - 厦门
Tel: 86-592-238-8138
Fax: 86-592-238-8130

中国 - 珠海
Tel: 86-756-321-0040
Fax: 86-756-321-0049

亚太地区

台湾地区 - 高雄
Tel: 886-7-536-4818
Fax: 886-7-330-9305

台湾地区 - 台北
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

台湾地区 - 新竹
Tel: 886-3-5778-366
Fax: 886-3-5770-955

澳大利亚 Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

印度 India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

印度 India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

印度 India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

日本 Japan - Osaka
Tel: 81-66-152-7160
Fax: 81-66-152-9310

日本 Japan - Yokohama
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

韩国 Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

韩国 Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 或
82-2-558-5934

马来西亚 Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

马来西亚 Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

菲律宾 Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

新加坡 Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

泰国 Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

欧洲

奥地利 Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

丹麦 Denmark-Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

法国 France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

德国 Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

意大利 Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

荷兰 Netherlands - Druenen
Tel: 31-416-690399
Fax: 31-416-690340

西班牙 Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

英国 UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820