AN1283

Microchip 无线 （MiWi™）介质访问控制器 ——MiMAC

作者： *Yifeng Yang*
Microchip Technology Inc.

简介

无线通信协议的主要功能是在两个节点之间发送和 / 或接收信息。介质访问控制器（Media Access Controller，MAC）层提供基本的信道访问、寻址和数据发送 / 接收功能，处于处理原始数据的物理 （PHY）层之上。在标准的开放系统互连 （Open Systems Interconnection，OSI）模型中，它用作数据链路层 （Data Link Layer，DLL）。由于 PHY 层的可能实现方式多种多样，所以MAC是可在软件中针对通信协议进行标准化的最低层。

本应用笔记定义了 Microchip 的 MAC 层 MiMAC，用于通过 Microchip 支持的通信协议和收发器实现短距离、低数据速率的低功耗无线应用。

实现 MiMAC 可以为无线应用开发人员带来多方面的好处：

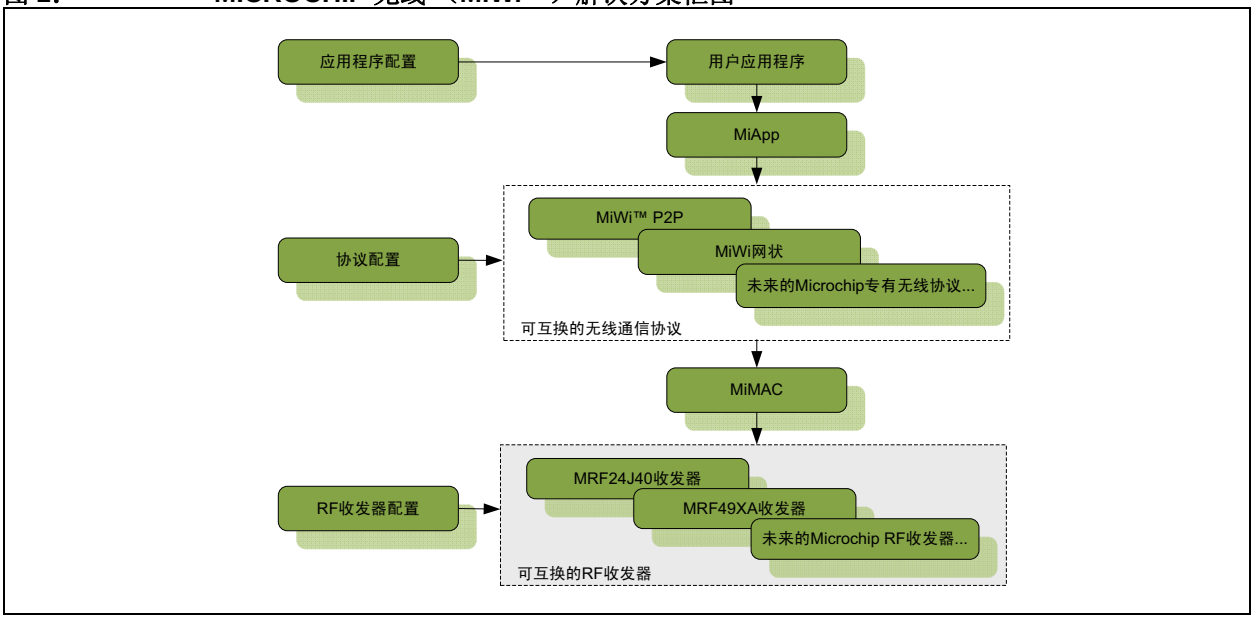
- 传统上，无线通信协议栈的实现很复杂，且难以使用。借助新定义的 MiMAC，可以让协议栈用于广泛的不同 RF 收发器。
- MiMAC 的学习曲线可以变得平缓，并且可以应用于不同频段和调制方式的所有 Microchip 收发器。它使最终用户可以在软件开发的任意阶段更改为使用不同的收发器，从而显著降低无线应用开发人员的开发风险。在固件中通过修改配置参数选择收发器是一个对客户透明的过程。

MiMAC 特性

MiMAC 实现了以下特性：

- 易于学习、实现和支持。
- 在 Microchip 的单片机 （MCU）和 RF 收发器上实现时具有足够的灵活性。
- 功能强大，足以应对大多数短距离、低数据速率应用。
- 简单但强大的安全模块，其安全模式可用于不具有硬件安全引擎的收发器。
- MiMAC 和所有 Microchip 专有无线通信协议之间简明却功能强大的编程接口。
- 对固件占用空间的影响最低。

图 1： MICROCHIP 无线 （MiWi™）解决方案框图



Microchip 应用程序编程接口 （MiApp）

除了在 MiMAC 层中进行标准化之外，Microchip 的另一个目标是标准化应用层中的接口。应用层中的标准接口称为 Microchip 无线应用程序编程接口 （Application Programming Interface， API）或 MiApp。 MiApp 的定义使所有 Microchip 专有无线协议可以互换，而无需或几乎无需更改软件应用程序代码。关于 MiApp 的详细信息，请参见 AN1284，《Microchip 无线 （MiWi™）应用程序编程接口 ——MiApp》。

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 使所有 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。

MiMAC 和 MiApp 使无线应用开发人员获得最大程度的灵活性，可以在软件开发的任意阶段选择 RF 收发器和无线通信协议，从而将开发风险降至最低。

Microchip 无线配置

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 / 或网络中的角色，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。
- “ 协议栈配置 ” 可以微调协议栈的行为。协议栈级别的大部分配置用于设置协议栈的时序，指定路由机制等方面。
- “ 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。

协议栈和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。但应用程序配置往往需要进行更改，以适应不同无线应用的需求。

图 1 显示了 Microchip 无线 （MiWi™）解决方案。

MiMAC 概述

MiMAC 层包含三个独立但又紧密相关的主要部分。在这三个主要部分中，第一个和第二个部分是针对在MAC 层中具有有限硬件支持的 Microchip 专有 RF 收发器定义的。第三个部分是针对所有 Microchip RF 收发器定义的。这三个部分为：

1. MiMAC 帧格式

   帧格式定义数据包在空气中的传播形式。基本上，MiMAC 帧格式决定 MiMAC 规范的功能和效率。它用作 MiMAC 架构中其他两部分的基础。

2. MiMAC 安全模块

   对于所有无线通信，报文均通过空气发送。与有线通信相比，无线通信的信息相对更容易进行拦截。因此，安全性是许多应用程序需要认真考虑的问题。MiMAC 安全模块定义了具有高安全强度的低成本分块加密算法。MiMAC 安全模块还定义了多种安全模式来配合分块加密算法，满足应用程序的不同要求。

3. MiMAC 通用编程接口

   MiMAC 通用编程接口用作所有 Microchip RF 收发器和 Microchip 专有无线通信协议之间的驱动程序。编程接口使 Microchip RF 收发器可以在任意 Microchip 专有无线协议下工作；它们还使所有 Microchip 专有无线通信协议可以使用 Microchip RF 收发器。

Microchip 支持的收发器在功能方面会存在很大不同。一些收发器具有明确定义的硬件 MAC 层，包括帧格式和 / 或安全引擎。为了满足规范，可能会将一些硬件功能内置到收发器中。 Microchip MRF24J40 是此类收发器的一个很好示例；它满足 IEEE 802.15.4™ 规范。如果收发器硬件中已实现了帧格式和 / 或安全引擎，则 MiMAC 不会调控它们，因为以往的经验证明硬件功能通常速度更快、消耗的系统资源更少。

对于在帧格式和 / 或安全引擎方面具有内置硬件支持的收发器，建议使用收发器上的硬件实现和 MiMAC 编程接口。

对于其他专有 RF 收发器，硬件中定义的 MAC 层极其有限或几乎没有。对于这些类型的收发器，建议使用 MiMAC 规范的所有三个主要部分。通过在软件中采用功能强大的 MiMAC 定义，Microchip 使那些简单 RF 收发器可以在软件中实现与复杂性更高的同级硅片几乎相同的通信或联网功能。

后面几节介绍了 MiMAC 规范三个主要部分中的每个部分。

MiMAC 帧格式

MiMAC 帧格式定义确保它对于无线应用开发人员易于学习和易于支持。作为一种副产品，通用数据包格式可以简化嗅探器实现 —— 可以仅实现一个运行于 PC 上的嗅探器软件，同时使用其他硬件收发器来嗅探空气，并将数据包发送到 PC 来进行解释。由于所有数据包在 MiMAC 帧格式定义中具有相同的格式，MiMAC 层中的数据插补在 Microchip 的所有 RF 收发器之间是相同的。

帧格式的评估标准是其功能与效率。 IEEE 802.15.4 是短距离、低数据速率的低功耗无线 PAN 的实际行业标准，与它相比， MiMAC 帧格式提供了本质上相同的功能，同时具有更高的效率。作为一种对比，典型的最小 IEEE 802.15.4 帧在 MAC 报头中为 9 字节，而 MiMAC 单播最短可以为 2 字节。

图 2 显示了 MiMAC 帧格式的详细信息。

图 2： MiMAC 帧格式

层	PHY报头			MAC报头		
名称	前导符	SFD	数据包长度	MAC报头	MAC有效载荷	CRC
字节	各种长度	各种长度	0 - 1	2 - 21	各种长度	0 - 2

名称	帧控制	额外控制	序列号	目标地址	源地址
字节	1	0 - 3	1	0 - 8	0 - 8

名称	数据包类型	广播	安全性	转发	应答	目标地址存在	源地址存在
位	2	1	1	1	1	1	1

RF 收发器的数据包格式在顶层至少包含两个部分：

1. PHY 层
2. MAC 层

PHY 层

收发器使用 PHY 层来同步通信，并确保通信的可靠性。PHY 层中的各个字段的功能为：

a) 前导符用于同步通信。对于不同的收发器，前导符的长度可能不同，内容也可能不同。一些收发器可以配置前导符的长度和内容。如果前导符是可配置的，只需根据 RF 收发器数据手册中所述的建议尝试配置前导符即可。 MiMAC 帧格式不会调控前导符字段。

b) 帧起始定界符（Start-of-Frame Delimiter，SFD）通常与前导符配合使用，用于确保通信的同步。一些收发器可以使能 / 禁止 SFD 或配置 SFD 的内容。如果 SFD 是可配置的，则强烈建议使能 SFD 并根据收发器数据手册的建议设置其内容。MiMAC 帧格式不会调控 SFD 字段。

c) 数据包长度字段用于指定 MAC 帧的长度。一些收发器的该模式仅发送固定长度的数据包。在这种情况下，可以省略 PHY 报头中的数据包长度字段。MiMAC 帧格式不会调控数据包长度字段。

MAC 层

MiMAC 帧格式的 MAC 层包含 3 个子层； MiMAC 帧格式会调控所有三个部分：

- MAC 报头
- MAC 有效载荷
- 循环冗余校验 （Cyclic Redundancy Check， CRC）

MAC 报头

MAC 报头字段向数据包接收方提供关于如何解释数据包的关键信息。它包含 5 个子字段：

- 帧控制
- 额外控制
- 序列号
- 目标地址
- 源地址

帧控制

帧控制字段用于解释 MAC 报头。它具有 7 个独立的子字段，用于控制 MAC 层的不同方面。帧控制中每个子字段的详细描述如下：

- 2 位的数据包类型字段指定如何解释数据包，包括其有效载荷。对于不同的数据包类型， MiMAC 层应以不同的方式处理数据包。
  - 对于数据型数据包，数据包类型为 `0b00`。接收数据型数据包时， MiMAC 通常会将 MAC 有效载荷直接传递给上层协议层。数据型数据包可以在上层协议层中处理，也可以直接在应用程序中处理。
  - 对于命令数据包，数据包类型为 `0b01`。在这种情况下，有效 MAC 有效载荷的第一个字节是 MAC 命令，后面跟随可选的命令参数。接收命令数据包时， MiMAC 通常会将 MAC 有效载荷传递给上层协议层，并使用一个标志来指示它是一个命令帧。命令由上层协议层进行解释。命令数据包通常在上层协议层中进行处理。
  - 对于应答数据包，数据包类型为 `0b10`。应答数据包不具有源地址和目标地址。它依靠序列号来标识要应答的数据包。应答数据包由 MiMAC 处理；有时，它仅由收发器硬件处理。 MiMAC 层中的高级功能 （如自动应答和重发）全都依靠应答数据包。应答帧不会被传递给上层协议层。
  - 数据包类型 `0b11` 保留用于一些收发器和 Microchip 专有协议的高级功能。 MiMAC 层会将该数据包类型的接收数据包直接传递给上层协议层。当 MiMAC 层接收到发送此类数据包的请求时，它会在不进行任何修改的情况下发送出数据包。
- 1 位的广播字段用于指定数据包是广播还是单播。当该位设置为 1 时，表示该数据包是无目标地址的广播；否则，清零该位表示它是提供目标地址或可推断出目标地址的单播报文。通过在帧控制中使用 1 位来指定广播， MiMAC 帧格式规范实际上可以避免在目标地址字段中发送特殊的广播地址。

- 1 位的安全字段用于指定在发送过程中是否已加密 MAC 有效载荷。将该位置 1 时，指示 MAC 有效载荷需要通过解密过程来获取原始数据。使能安全性时，会在 MAC 报头之后提供一个附加的辅助安全性报头。关于如何解释辅助安全性报头，请参见 **“MiMAC 安全模块 ”** 一节。
- 1 位的转发字段用于指定数据包是否需要中继器来转发该数据包。该位仅对具有转发功能的器件有用。当该位置 1 时，如果目标地址不是中继器的地址，接收该数据包的中继器将会转发该数据包，以扩大通信覆盖范围。
- 1 位的应答字段用于指定是否需要期待来自接收方的应答数据包。当该位设置为 1 时，发送方需要在预定义周期接收具有相同序列号的应答数据包。应答的超时周期取决于收发器设计。该位与数据包类型 “ 应答 ” 是不同的。应答位用于指示希望收到数据包类型 “ 应答 ”，以确认当前数据包是否送达。而数据包类型 “ 应答 ” 则是对应答位置 1 的数据包的响应。
- 1 位的目标地址存在字段用于确定 MAC 报头中是否存在目标地址。当该位置 1 时， MAC 报头中会提供目标地址 （其长度由收发器或上层通信协议定义）。当该位清零时， MAC 报头中不会提供目标地址。在以下情况下，不会提供目标地址：
  - 在应答数据包中，不存在目标地址。当数据包类型为 `0b10` 时，目标地址存在位必须清零。
  - 在广播数据包中，不存在目标地址。当广播位置 1 时，目标地址存在位必须清零。
  - 如果使用推断目标，则可以省略目标地址。在这种情况下，目标地址存在位必须清零。当使用推断目标模式时，在计算 CRC 时仍然会使用目标地址，但不会发送目标地址。当其他收发器接收到数据包时，它们会将自己的地址添加到数据包的目标地址位置中，并校验 CRC。在这种情况下，如果产生 CRC 错误，则说明发生发送错误或该报文并不是想要发送给该接收节点的。在以上任一条件下，接收节点都会丢弃该数据包。只有预定的目标收发器在使用自己的地址作为目标地址计算 CRC 时不会产生 CRC 错误，因此，只有预定的目标器件才会接受该数据包，并在上层协议层中进行相应的处理。隐藏目标地址不仅可以节省发送这些地址的时间和能量，而且还可以提供最低限度的保护，以避免网络活动完全暴露。

    在发送范围中两个具有不同地址的收发器可能生成相同 CRC 代码的几率非常微小 （对于 2 字节 CRC，约为 0.0015%）。推断目标模式适合于大多数应用程序。对于需要绝对确定目标的应用程序，建议将目标地址存在位置 1。

> **注：** 推断目标地址方法属于 Microchip 的知识产权（Intellectual Property，IP）。该方法的专利申请目前正在审批中。

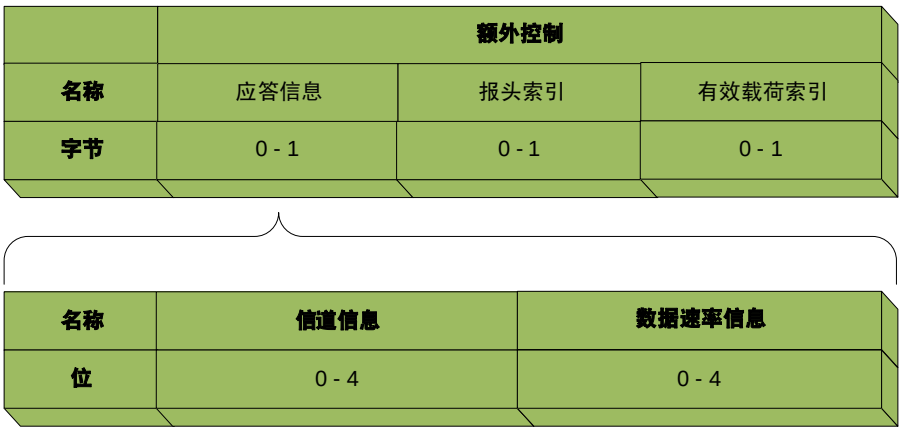
- 1 位的源地址存在字段用于确定 MAC 报头中是否存在源地址。当该位置 1 时， MAC 报头中会提供源地址 （其长度由收发器或联网协议定义）。当该位清零时， MAC 报头中不会提供源地址。在正常数据发送过程中是否提供源地址取决于应用的需求。

额外控制

对于一些具有高级功能（如上层安全模块、自适应数据速率和信道）的收发器，需要更多的信息来解释 MAC 信息。通常情况下，这些字段仅保留用于高端收发器，将由收发器硬件而不是软件使用。

图 3 给出了额外控制字段的定义。

图 3：　　用于高级功能的额外控制字段



额外控制字段包含三个部分：

- 应答信息
- 报头索引
- 有效载荷索引

只有在要求应答，并且自适应信道功能或数据速率功能已开启时，才会提供应答信息。应答信息主要由硬件使用，用以确定数据速率或用于发送回应答的信道。信道信息字段用于自适应信道功能，数据速率信息字段用于自适应数据速率功能。自适应信道功能使收发器可以使用不同的频率发送和接收数据。对于在拥挤的无牌频段工作的大型网络，该功能非常有用。对于大型网络，该功能使每个无线节点可以在噪声最低的频率（信道）进行接收，并根据目标器件以接收频率（信道）进行发送。自适应数据速率功能使收发器可以使用不同数据速率发送和接收数据包。它类似于自适应信道功能，并支持在网络中进行更高效的数据传输。

注：	自适应信道功能属于Microchip的知识产权（IP）。该方法的专利申请正在审批中。

报头索引和有效载荷索引专门用于硬件安全引擎，特别是加密和身份验证过程。如果安全功能不是在MAC层而是在更高协议层执行，则它用于识别身份验证信息和安全信息。只有安全功能已使能，但不是在 MiMAC 层中执行时，才会提供报头索引和有效载荷索引。MiMAC 规范并未定义如何处理不在MiMAC层中执行的安全功能。如何在相应的安全层中使用这些额外控制字段来执行安全操作完全由上层协议层决定。

序列号

序列号用于标识各个发送数据包。所有收发器的序列号都必须以一个随机数开始，然后随每个数据包的传输而递增。序列号通常用于在应答数据包中标识所应答的数据包。通常，应答数据包的序列号必须与所要应答的数据包相同。

当不存在由上层协议层提供的网络层时，序列号用于标识广播报文；因而，如果先前已执行了重播，则无需重播。

目标地址

目标地址定义单播数据包的目标地址。该字段的长度为 0 至 8 字节。MAC 报头中的目标地址由帧控制字段中的目标地址存在标志决定。

如果目标地址字段的长度不为零，则目标地址的长度由收发器寻址机制和应用决定。应用层可以选择 2 至 8 字节的地址长度，这取决于网络规模和具体的应用。

如果目标地址字段的长度为零，可能的情形是：

- 数据包是应答。
- 数据包是广播报文，通过广播位进行指示，该位在帧控制字段中置 1。
- 目标地址通过使用 CRC 进行推断。

源地址

源地址定义发送器件的地址。该字段的长度为 0 至 8 字节。源地址由帧控制字段中的源地址存在标志决定。

如果源地址字段的长度不为零，则源地址的长度由收发器寻址机制和应用决定。应用层可以选择 2 至 8 字节的地址长度，这取决于网络规模和具体的应用。

对于同一网络，源地址和目标地址的长度必须完全相同。如果源地址字段的长度为零，则单播报文的源地址对于该特定应用并非必要。在正常数据包单播期间是否要在 MAC 报头中包含源地址字段，可以在 MiMAC 层中配置。

MAC 有效载荷

MAC 有效载荷是通过 Microchip 专有无线协议或应用层发送的信息。如何解释该信息完全由 Microchip 专有无线协议层或客户应用程序决定。MAC有效载荷将不进行任何修改而通过 MiMAC 编程接口直接传递给 Microchip 专有无线协议层。如果 MAC 有效载荷采取了保护措施，它会先通过 MiMAC 安全模块解除保护。只有 MAC 有效载荷的解密明文会通过 MiMAC 编程接口传递给上层。如果安全检查由于任何原因而失败，将在 MiMAC 层中丢弃整个数据包。对于 MAC 有效载荷，其长度也会被传递给上层协议层。MAC 有效载荷长度根据 PHY 层的数据包长度计算得到，即将其减去 MAC 报头长度和安全模块的可能调整。

MAC CRC

MAC 层中的 CRC 字段用于确保数据包在发送过程中的完整性。一些 RF 收发器提供了硬件 CRC 生成 / 校验功能。对于不具有硬件 CRC 生成 / 校验功能的收发器，将使用 CRC 软件。

使用 CRC 软件时，可以使用循环和查找表 CRC 生成方法。通常，循环 CRC 生成方法相比查找表方法，使用的编程空间大约少 600 字节，但运行速度慢 3-4 倍。这两种方法生成相同的 CRC 值，因此它们可以互换。选择哪种方法取决于各个应用的需求。

在正常情况下，首选 2 字节 CRC，它保持了可靠性和简单性之间的平衡。强烈建议对于所有数据发送使用 CRC。在单播过程中省略目标地址时， CRC 是必需的。**“ 目标地址 ”** 一节介绍了如何省略目标地址。

MiMAC 安全模块

由于无线通信物理方面的原因，所有通信方将可以同样方便地访问通过空气交换的信息内容，无论它是预定还是非预定的侦听者。因此，确保数据包的安全对于一些应用至关重要。 MiMAC 安全模块通过以下方式帮助解决应用的安全需求：

- 如果收发器硬件支持安全模块，包括加密算法和不同的安全模式，则建议直接使用硬件安全引擎。在固件中加密和解密数据包需要消耗相对较高的 MCU 系统资源，因而它会降低吞吐率，并提高收发器主机 MCU 的速度和功耗要求。在这种情况下，不适用 MiMAC 安全规范。
- 如果硬件安全引擎仅提供分块加密算法，但不提供安全模式，则建议使用硬件安全加密算法，但在硬件加密算法之上应用软件安全模式。在这种情况下，不适用 MiMAC 安全加密算法，但适用 MiMAC 安全模式规范。
- 如果收发器硬件不提供任何安全支持，则同时适用 MiMAC 安全规范中的加密算法和安全模式。如果用户倾向于使用非 MiMAC 选择的其它分块加密法，一个备用的 MiMAC 安全模块将会提供一个预定义的接口以调用任何分块加密法。

选择默认的 MiMAC 安全引擎

选择默认的 MiMAC 安全引擎取决于三个条件：

- 安全引擎 IP 问题
- 低成本安全性
- 增强的安全强度

安全引擎 IP 问题

在处于公共域的所有流行安全引擎中，不存在 IP 问题的良好候选者包括：

- 数据加密标准 （Data Encryption Standard，DES/TDES）
- Blowfish/Twofish
- Serpent
- 高级加密标准 （Advanced Encryption Standard， AES）
- 微型加密算法 （Tiny Encryption Algorithm，TEA/XTEA/XXTEA）系列

所有这些安全引擎均可免费获得，具有参考设计，并已在大量实际产品中实现。

低成本安全性

低成本实现确保可以在系统资源和计算速度有限的低成本 MCU 上实现安全模块。

DES/TDES—— 上一代的加密标准；众所周知它们很复杂，相对于它们的安全强度而言需要相对更多的系统资源。

Blowfish/Twofish、 Serpent 和 AES—— 提供更安全的算法，同时其实现比 DES 系列简单。但是，这些加密算法所需的系统资源仍然高于嵌入式系统预期的资源。

请注意，这些加密引擎的典型实现需要至少 4 KB 的编程空间。相反， TEA 系列的典型实现只需几百字节的编程空间，并且执行速度更快。

考虑到嵌入式系统的系统资源，TEA 系列的安全引擎可以满足这种条件的要求。

增强的安全强度

MiMAC 安全规范不推荐使用具有已知弱点的安全引擎。

在 TEA 系列安全引擎中，存在 3 种形式：TEA、XTEA 和 XXTEA。TEA 是原始实现方案，在 1994 年首次发布。它存在等效密钥的已知弱点。利用一对相关密钥，对 TEA 安全引擎进行相关密钥攻击的最好结果需要 2^{32} 个选择明文，时间复杂度为 2^{32}。类似于 XTEA，开发 XXTEA 也是为了提供超过 TEA 的安全强度。它是一种异构的非平衡 Feistel 网络分块加密算法，它不限制分块大小。因此，XXTEA 通常可以更有效地处理较长报文，因为 XXTEA 可以应用于整个报文，而不是逐块进行加密。但是，XXTEA 的一个限制是需要至少 8 字节的加密数据。不修改安全引擎本身的情况下， XXTEA 无法作为现成的选择。

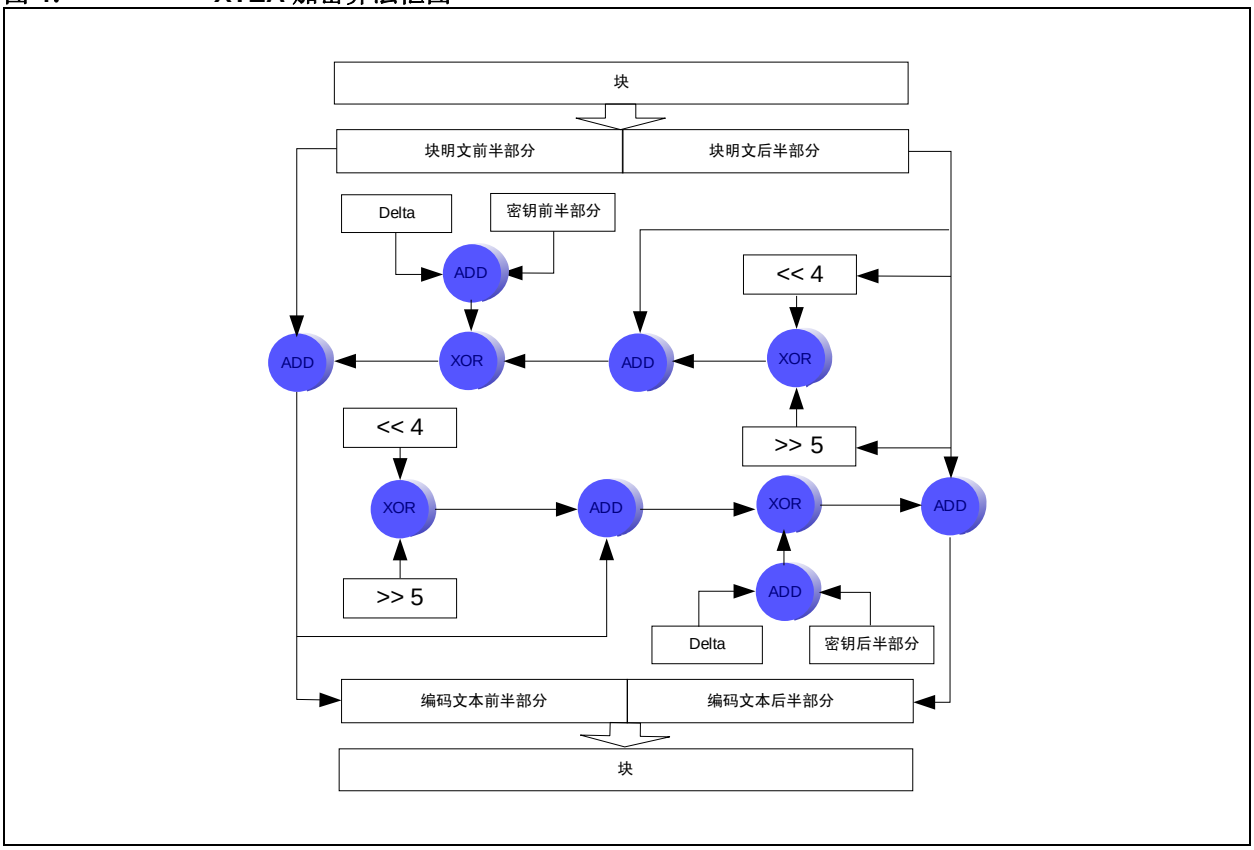
分析选择安全引擎的所有条件之后，只剩下 TEA 系列中的 XTEA 可以作为 MiMAC 安全规范中的默认安全引擎。

XTEA 分块加密算法

XTEA 是使用 128 位密钥的 64 位分块加密算法。它的设计可以绕过 TEA 加密算法中的弱点。它由英国剑桥大学剑桥计算机实验室的 David Wheeler 和 Roger Needham 在 1997 年首次发布，现在已进入公共域。它不受任何专利约束。

图 4 显示了 XTEA 加密算法的工作原理。

图 4： XTEA 加密算法框图



最新的加密算法分析显示，只有在极端条件下，才能通过相关密钥差分攻击破解XTEA。要执行相关密钥差分攻击，攻击者需要观察几个不同密钥下的加密操作，并获取一组已知明文的加密内容。对于 64 轮迭代 XTEA，已知的最好攻击结果为 26 轮，需要 $2^{20.5}$ 个选择明文，时间复杂度为 $2^{115.15}$。（Youngdai Ko、 Seokhie Hong、 Wonil Lee、 Sangjin Lee 和 Jongin Lim。 *"Related Key Differential Attacks on 26 Rounds of XTEA and Full Rounds of GOST"*。FSE 2004 学术会议论文集，Lecture Notes in Computer Science，2004 Springer-Verlag）。这意味着破解 XTEA 的条件和复杂度极其困难。即使满足了每个条件，在 1000 MIPS 计算机上破解 XTEA 的时间也需要 1.46×10^{18} 年！相反，根据最新的估计，宇宙的年龄仅约为 1.4×10^{10} 年。

XTEA 的优点

XTEA 最大的优点之一是加密或解密信息所需的系统资源非常有限。仔细研究 XTEA 算法可以发现，与具有类似强度的其他安全引擎相比， XTEA 的易失性存储器要求极低。因此， XTEA 因其可用于资源很少的嵌入式系统而知名。

XTEA 的另一个优点是算法所需的资源和复杂度可以通过对算法应用不同的轮次进行微调。轮次越少，算法的执行速度就越快，并且复杂度会随轮次线性降低。但是，此时也更容易使用较少的轮次破解算法。对于 MiMAC 服务的无线应用，所需的安全级别和响应时间会有很大差别。可在 XTEA 中简便地调整安全级别和系统资源需求对于配合一系列广泛的应用是非常具有价值的。

修改 XTEA 分块加密算法

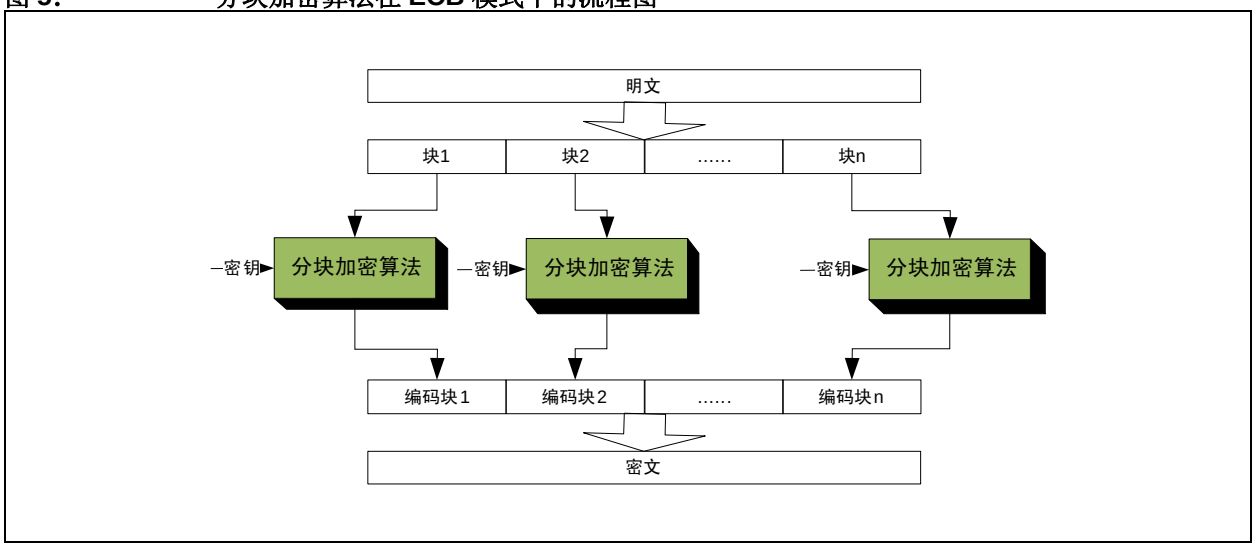
XTEA 加密引擎适合于嵌入式系统的安全需求。但是，XTEA 需要进一步的修改，以最好地满足 MiMAC 安全策略。

安全模式

通常，安全引擎会应用不同的安全模式来确保数据的安全。分块加密算法最简单的安全模式实现是电子码书（Electronic CodeBook，ECB）模式。简单地说，报文划分成多个具有相同大小的块（块大小由加密算法定义），然后将加密算法应用于每个独立数据块，从而对输入数据进行加密。类似地，在使用分块加密算法解码器时，过程将相反，对数据进行解密。

图 5 显示了分块加密算法如何在 ECB 模式下进行编码。

图 5： 分块加密算法在 ECB 模式下的流程图



但是， ECB 模式存在一个缺点 —— 它不会隐藏数据模式。例如，如果明文的所有块是相同的，则输出加密数据也会是相同的，从而为黑客提供了一个关于如何破解安全引擎的重要提示。

为了克服 ECB 模式的缺点，计数器 （Counter， CTR） 模式使用非重复杜撰数据来隐藏明文中的模式。这需要额外的资源，但它可以显著提高输出报文的安全性。

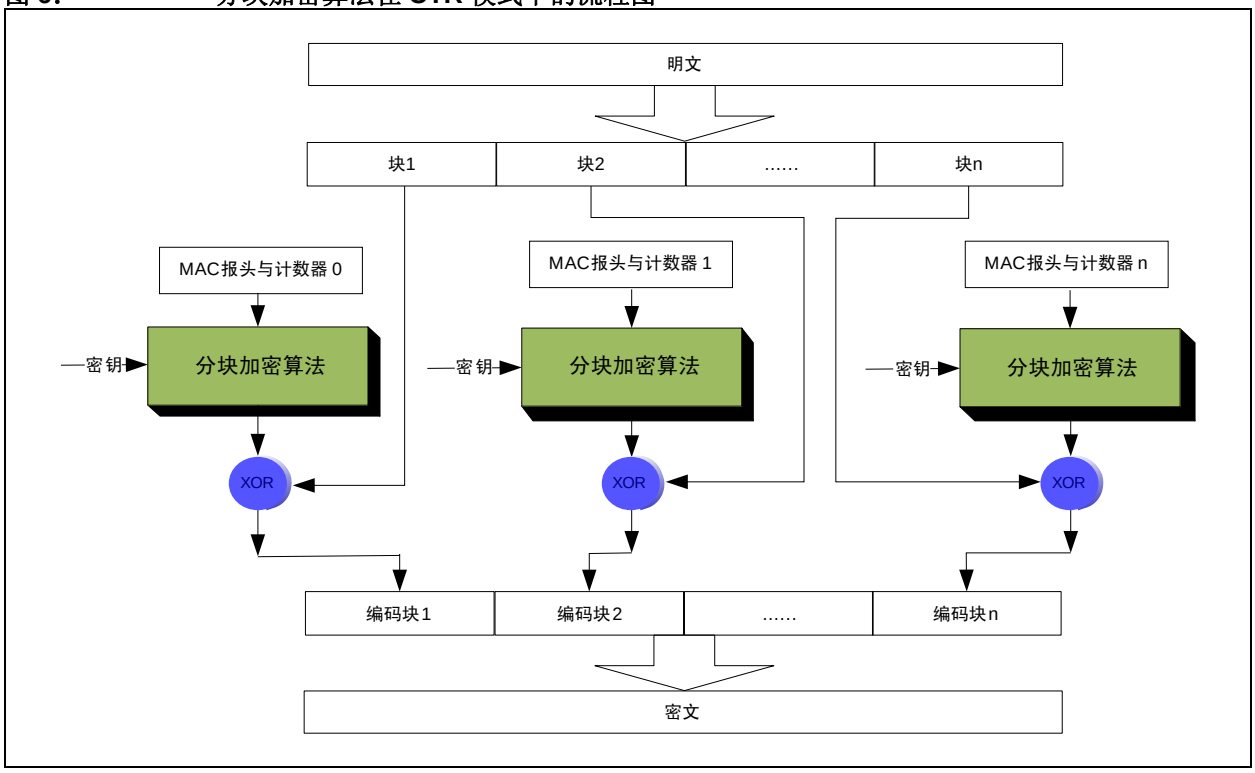
MiMAC 安全模块使用 MAC 报头来指定杜撰数据。

- 如果 MAC 报头的长度大于块大小，使用 MAC 报头填充杜撰数据时，将从使用帧控制字节作为杜撰数据中的最低字节开始，直到达到块大小的限制。
- 如果 MAC 报头长度小于块大小，使用 MAC 报头填充杜撰数据时，将从使用帧控制字节作为杜撰数据中的最低字节开始，并将杜撰数据的其余部分填充为零。

  最后，杜撰数据的最高字节将作为计数器，其起始值为零，并随后续的块自动递增。

图 6 显示了分块加密算法如何在 CTR 模式下工作。

图 6： 分块加密算法在 CTR 模式下的流程图



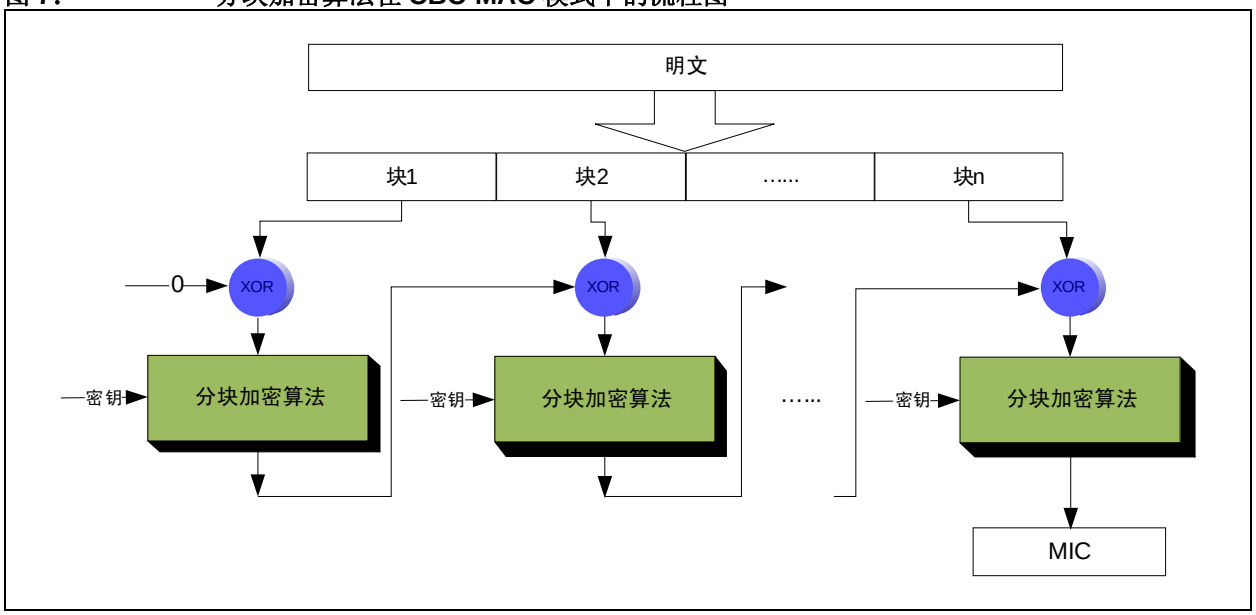
无线通信不仅应防止信息泄漏，而且还应确保信息不会干扰网络的正常工作，无论是有意还是无意。由于以下原因， CTR 模式下的信息加密可以被证明并不足够：

- 使用简单的嗅探器可以轻松地执行重放攻击。它会严重影响某些应用的网络操作。重放攻击通过发送所接收到的相同数据包来执行。在某些应用中，并不希望接收到相同的报文。一个很好的示例就是通过报文来交替点亮 / 关闭一盏灯。
- 解密过程无法检测任何故障，因而传输的任何随机数据在经过解密过程之后都可能可以在网络上进行操作。

除了 CTR 加密模式之外， MiMAC 还需要定义对报文进行认证的工作模式。认证通过检查附加的报文完整性代码（Message Integrity Code，MIC）来确保传输的报文未经过任何形式的更改。对于分块加密算法，标准认证模式是密码块链接报文认证代码（Cipher Block Chaining Message Authentication Code，CBC-MAC）。CBC-MAC 是一种不与任何特定安全引擎关联的工作模式。在 IEEE 802.15.4 规范中， CBC-MAC 模式应用于 AES-128 引擎。在 MiMAC 安全规范中， CBC-MAC 可应用于 XTEA 分块加密算法。在 MiMAC 安全规范中，定义了认证模式 XTEA-CBC-MAC-32 和 XTEA-CBC-MAC-64 来生成 32 位或 64 位 MIC。

图 7 显示了 CBC-MAC 模式的过程。

图 7： 分块加密算法在 CBC-MAC 模式下的流程图



如图 7 所示，XTEA 分块加密算法用作一个哈希函数。为了在 XTEA 中调用 CBC-MAC 模式，报文被分成小块，其块大小由分块加密算法定义。默认情况下，XTEA 定义 64 位的块大小。如果最后一个块只有一部分数据，则使用零填充块的其余部分。第一个块用作具有一个预定义密钥的 XTEA 引擎的输入。经过加密过程之后，XTEA 引擎的输出将与下一个块进行异或运算，并作为 XTEA 分块加密算法的输入。处理完最后一个块之后，XTEA 引擎的最终输出即为 MIC。对于 XTEA-CBC-MAC-64 模式，完整的最终结果将用作 MIC ；对于 XTEA-CBC-MAC-32，只有最终结果的低 32 位将用作 MIC。 MiMAC 会将 MIC 附加到数据包末尾进行发送。在接收方，节点将使用完全相同的过程计算 MIC。然后，接收节点会将计算出的 MIC 与作为原始报文的附件而接收的 MIC 进行比较。如果两个 MIC 完全相同，则会接受整个接收数据包；否则，将会丢弃该数据包。

CBC-MAC 可以用于防止重放攻击。通常情况下，带有 CBC-MAC 认证信息的已发送数据包会在 MAC 报头之后包含具有预定义长度（通常为 4 字节）的帧计数器字段。对于发送的每个数据包，帧计数器都会递增 1。在接收方，只有帧计数器值高于已记录的值时，才会接受数据包。因此，发送重复数据包将视为执行重放攻击，将被丢弃。即使发送方有意地将帧计数器修改为较高值，数据包也无法通过认证检查，因为帧计数器值会用于计算附加到数据包末尾的 MIC。

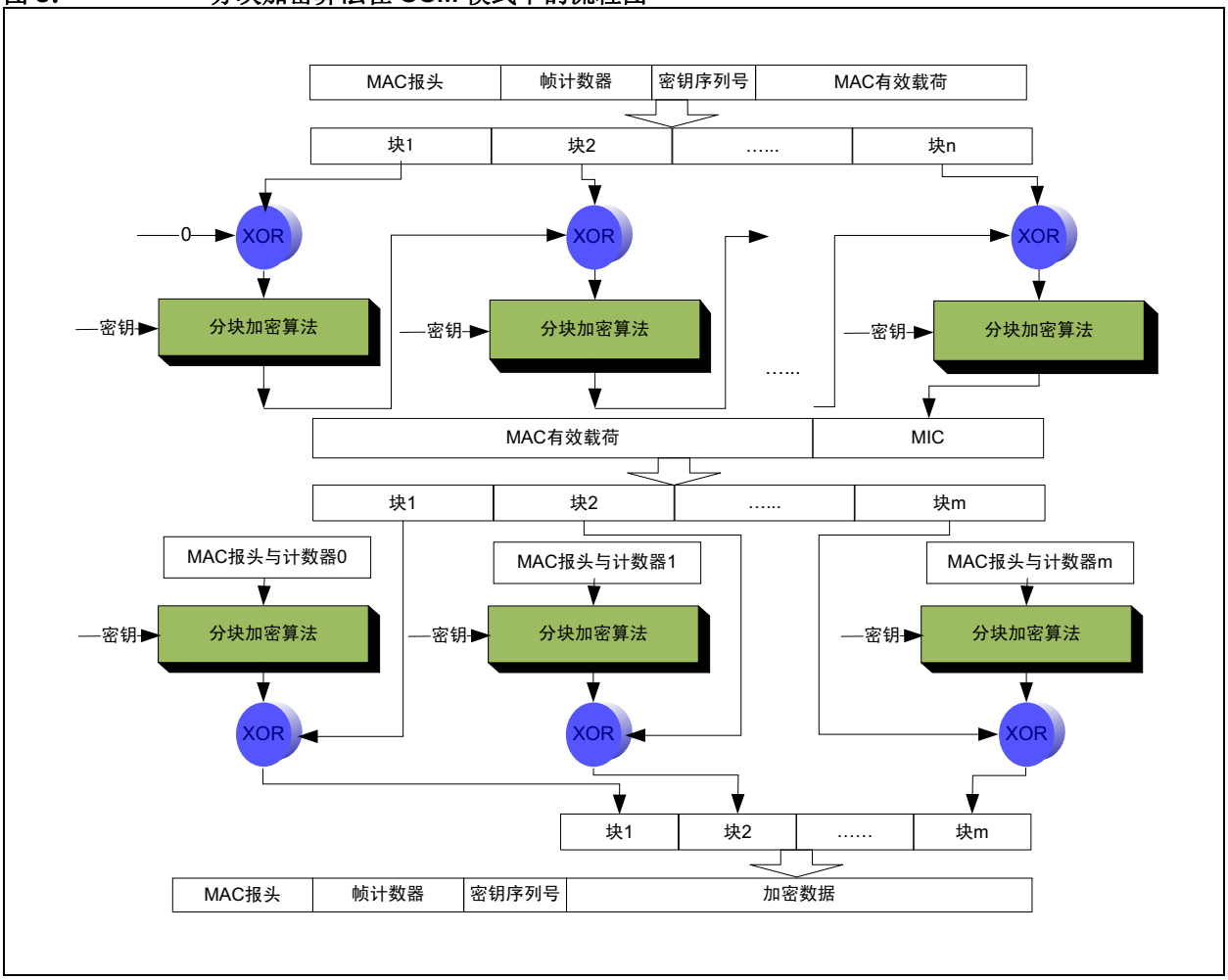
CBC-MAC 模式用于对报文进行认证，但该模式本身不会加密报文。IEEE 802.11i 使用偏移码书（Offset CodeBook，OCB）模式进行认证，并在同一时间加密数据，从而节省对于计算能力的要求。但是，一项专利申请涵盖了 OCB 模式。虽然在 GNU 通用公共许可证（General Public License，GPL）下使用 OCB 模式可以获得特别豁免权，但 MiMAC 安全规范不依赖于存在潜在 IP 问题的 OCB 模式是明智之举。

作为 OCB 模式的一种替代，计数器与 CBC-MAC（CCM）模式将对报文执行 2 遍计算（而不是 OCB 模式下的 1 遍），因而使计算资源需求加倍。CCM 模式基本上就是先应用 CBC-MAC 模式来生成 32 位或 64 位 MIC，随后通过对报文和 MIC 应用 CTR 模式来加密报文。与 CTR 模式或 CBC-MAC 模式相比，CCM 模式需要加倍的计算资源，但可以对通过空气传输的数据提供最完备的保护。根据 MIC 的长度，有两种 CCM 模式可用：CCM-64 和 CCM-32。

当 MiMAC 应用 CCM 模式时，将使用完整的数据包（包括 MAC 报头、安全辅助报头和 MAC 有效载荷）来认证报文，并生成 MIC 来保护整个数据包。只有 MAC 有效载荷和 MIC 将被加密。

图 8 显示了完整的 CCM 模式过程。

图 8：　　分块加密算法在 CCM 模式下的流程图



当无线节点接收到通过 CCM 进行保护的数据包时：

- 首先，它会应用 CTR 模式对 MAC 有效载荷进行解密。

  解密的明文包含原始数据和 MIC。

- 然后，对原始数据应用 CBC-MAC 模式来计算 MIC。
- 最后，将计算出的 MIC 与从原始数据中解密出的 MIC 进行比较。

  如果两个 MIC 不匹配，将丢弃整个数据包。类似于 CBC-MAC 模式，使用 CCM 模式时可以在数据包中包含一个帧计数器。出于上面对于 CBC-MAC 模式所描述的相同原因，数据包中附加的帧计数器字段可以有效地防止重放攻击。

由于在 XTEA 分块加密算法之上使用安全模式，一个小小的好处是只需实现 XTEA 的编码功能。

密钥强度

通常，密钥越长，加密引擎的安全强度就越高。 XTEA 最初开发并发布时是一种使用 128 位对称密钥的 64 位分块加密算法。

但是，美国政府的出口法规要求，出口使用 64 位以上对称密钥的加密引擎需要特别批准。

因此，XTEA 安全引擎需要进行降级，以支持 64 位对称密钥作为一种工作模式。

虽然 XTEA 是作为使用 128 位对称密钥的 64 位分块加密算法开发的，它可以遵循相同的概念，修改为支持使用 64 位对称密钥的 32 位分块加密算法。其差别在于块大小和 DELTA 的定义（来自黄金分割率的常量魔法数字）。由于引入了 32 位块大小和 64 位密钥， XTEA 分块加密算法现在具有两种模式：XTEA-128和XTEA-64。对于 XTEA-64，由于块大小减小， CBC-MAC-64 和 CCM-64 安全模式不再可用。

由于 XTEA 降级，安全过程的速度预计会提高，而安全模块的强度会下降。由于 XTEA-64 加密算法不是标准的安全引擎，所以对该算法未执行过任何加密分析，因而破解该算法的复杂度目前为止是未知的。据认为，XTEA-64 仍然可以为加密引擎的普通用户提供足够的安全性。建议用户增加轮次，从而在一定程度上提高引擎的安全性。对于在安全方面需要更高保密性的客户，只需获得批准而满足美国关于安全引擎的出口法规要求，随时可以采用标准 XTEA 或 XTEA-128 来提供更高的保密性。

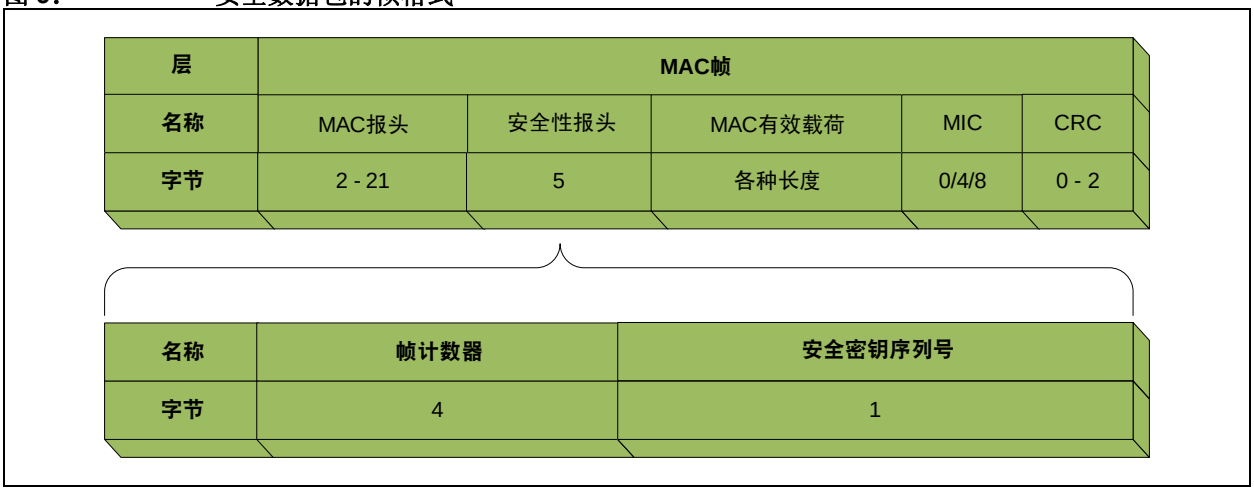
安全数据包格式

当 MiMAC 数据包采取了保护措施时，需要附加的信息来确保成功地处理解密过程。基本上，发送方需要发送除安全密钥本身之外对于解密数据包可能有用的所有必要资料。

有两个部分是安全处理过程所必需的安全资料：帧计数器和安全密钥序列号。它们是辅助安全性报头的一部分，紧跟在 MiMAC 报头之后。

图 9 给出了安全数据包的帧格式。

图 9：　　安全数据包的帧格式



安全数据包的帧格式与非安全数据包的区别在于以下方面：

- 帧控制字节中的 secEn 标志置 1。
- 包含附加的辅助字段，称为安全性报头；它包含两个部分：
  - 4 字节的帧计数器，用于避免重放攻击。**“ 安全模式 ”** 一节详细介绍了如何避免重放攻击。
  - 安全密钥序列号，用于标识要使用哪个安全密钥。
- 附加到 MAC 有效载荷末尾的 MIC。 MIC 的长度取决于所使用的安全模式。

在不同安全模式下，数据包中采取安全措施的部分会不同：

- 对于 CTR 模式，只有 MAC 有效载荷会进行加密；不存在附加的 MIC。
- 对于 MAC-CBC 模式，认证信息从 MAC 报头开始，直到 MAC 有效载荷的末尾。对于 MAC-CBC-32 和 MAC-CBC-64 模式， MIC 长度分别为 4 字节或 8 字节。
- 对于 CCM 模式，认证信息从 MAC 报头开始，直到 MAC 有效载荷的末尾。加密信息从 MAC 有效载荷起始处开始，直到 MIC 的末尾。对于 CCM-32 和 CCM-64 模式， MIC 长度分别为 4 字节或 8 字节。

替代加密引擎支持

XTEA-128 是一个强大的安全引擎，适用于大多数应用程序。但是，对于一些需要特定安全引擎的无线应用开发人员， MiMAC 安全模块提出一种可选方案。对于分块加密算法，定义了一个安全引擎接口。为替代分块加密算法实现该安全接口之后，就可以在 MiMAC 中使用新的安全引擎，无需任何进一步的修改。请注意，三种安全模式 CTR、CBC-MAC 和 CCM 仍将应用于替代安全引擎，以确保对数据进行适当的保护。由于具有这些安全模式，所以只需实现替代分块加密算法的编码过程。

替代安全引擎的接口如下：

```
BOOL encode(BYTE *buffer, BYTE *key);
```

在安全接口函数调用 encode 中，存在两个参数：buffer（缓冲区）和 key （密钥）。

- Buffer

  Buffer 参数是调用该函数时指向明文的指针。当函数返回时，指针所指向的缓冲区将被替换为编码数据。

- Key

  Key 参数是指向分块加密算法的安全密钥的指针。

MiMAC 通用编程接口

如前面所述，在 MiMAC 层中标准化帧格式和安全模块为无线应用开发人员提供了很大的优势。下一步是标准化 MiMAC 和 Microchip 上层专有协议层之间的软件编程接口。

Microchip 支持多种短距离、低数据速率 RF 收发器。如前面所述， MiMAC 通用编程接口：

- 使所有 Microchip RF 收发器在软件开发过程中可以互换。
- 降低无线应用开发人员的软件开发风险。
- 在兼顾不同收发器的潜在特点的同时，提供了一个简单接口和上层协议配合，并且没有显著增加程序空间的占用。

定义了两种类型的接口来配合不同的Microchip RF收发器：配置文件和函数调用。

对于不同的 Microchip RF 收发器，硬件接口、功能和寄存器设置会有很大不同。对于仅适用于一种 RF 收发器的配置，都定义了一个配置文件。这些配置通常不会在应用程序运行时更改。通常情况下，文件中的配置将在初始化过程中设置。

对于所有无线协议，所有函数调用可以分为 4 个主要类别：

- 配置
- 发送数据包
- 接收数据包
- 特殊功能

MiMAC 编程接口为每个函数类别定义了一个或多个函数调用。从上层协议层调用这些编程接口将执行 RF 收发器的几乎所有功能。后面几节按功能类别定义了编程接口。

MiMAC 配置

MAC 层配置接口从上层协议层对 MiMAC 层进行调用，用以配置 RF 收发器的行为。不同于在配置文件中定义的配置参数，这些函数调用中的配置可以在无线应用程序运行时更改。对于收发器硬件支持的功能，MiMAC 层需要设置收发器中相应的寄存器位。对于收发器硬件不支持的功能， MiMAC 层固件需要先对它进行处理，之后再将控制权转移到上层。以下函数用于配置 MAC 层：

- `MiMAC_Init`
- `MiMAC_SetPower`
- `MiMAC_SetChannel`
- `MiMAC_SetAltAddress`

`MiMAC_Init`

`MiMAC_Init` 函数调用用于初始化 MiMAC 层的行为。可以配置以下 MAC 行为：

- 永久地址
- 使能 / 禁止中继器模式

该函数作为 Microchip 上层协议层中的最初步骤之一进行调用。在该函数调用之后，将在 RF 收发器中初始化 MiMAC 层的行为。

为了向 MiMAC 层表示所有可配置信息，定义了以下结构，并用作 `MiMAC_Init` 函数调用的输入参数。

```
typedef struct
{
    union
    {
        BYTE Val;
        struct
        {
            BYTE RepeaterMode:1;
            BYTE PAddrLength:4;
        } bits;
    } actionFlag;

    BYTE *PAddress;
} MACINIT_PARAM;
```

该结构的描述如下：

- **RepeaterMode**

  使能收发器在 MAC 报头中的帧控制字节的重复位置 1 时转发数据包。

- **PAddrLength**

  定义收发器的永久地址的长度。长度可以定义为 2 至 8 字节。

- **PAddress**

  指向收发器永久地址的指针。

该函数调用的完整格式如下所示。返回值指示操作是否成功。

```
BOOL MiMAC_Init(MACINIT_PARAM initValue);
```

MiMAC_SetPower

MiMAC_SetPower 函数调用用于设置收发器的输出功率。它是一个 PHY 函数调用，而不是 MiMAC 层。此处，MiMAC 层只是将设置传递给收发器 PHY 层的中间层。单独定义该接口，而不是将它用作 MiMAC_Init 函数调用中的一个参数，是为了使上层协议层或应用层可以在运行过程中基于应用需求灵活地调整输出功率。

该函数调用的完整格式如下所示。返回值指示操作是否成功。

```
BOOL MiMAC_SetPower(BYTE outputPower);
```

输入参数 outputPower 使用一个字节表示。各种收发器需要负责解释输入值，并设置相应的输出功率。强烈建议上层协议层中的固件或应用层使用收发器可识别的预定义值，该值可以位于目标收发器的定义头文件中。

MiMAC_SetChannel

MiMAC_SetChannel 函数调用用于设置 RF 收发器的工作频率。该函数调用的完整格式如下所示。返回值指示操作是否成功。

```
BOOL MiMAC_SetChannel(BYTE channel,
BYTE offsetFreq);
```

该函数调用具有两个输入参数：信道和偏移频率。

- **信道**

  定义 RF 收发器工作的中心频率。信道编号定义为从 0 至 31。

- **偏移频率**

  它在一些 RF 发送器中用作一个额外配置，用于将中心频率设置为信道未定义的任何频率。通常情况下，偏移频率不能大于相邻信道之间的频隙。

适当地设置信道和偏移频率时，可以将 RF 收发器的中心频率微调为任何可能值。对于在规范中严格定义了固定信道中心频率的收发器，可以丢弃 offsetFreq 参数。

并不是每种收发器和数据速率或频段都支持所有信道。如果 RF 收发器在当前条件下不支持输入参数 channel，则不会更改当前工作信道，并返回值 FALSE 来指示信道更改失败。否则，将在收发器中更改工作信道，并返回值 TRUE。

MiMAC_SetAltAddress

MiMAC_SetAltAddress 函数调用用于在无线节点加入网络之后设置备用网络地址。只有收发器支持多个地址和个人局域网标识符（Personal Area Network Identifier，PANID）的概念时，才使用该函数。PANID 和备用网络地址由 IEEE 802.15.4 规范规定。并不强制要求所有收发器都支持该函数。对于不支持多个地址或 PANID 的收发器，输入参数将被丢弃，返回值将为 FALSE。

该函数调用的完整格式为：

```
BOOL MiMAC_SetAltAddress(BYTE *Address,
BYTE *PANID);
```

该函数调用具有两个输入参数：指向地址和 PAN 标识符的指针。

- **地址**

  对于支持网络地址的收发器它是必需的，用于标识网络中的器件。

- **PANID**

  它用于标识网络本身。

MiMAC RF 收发器配置文件

除了基于所有 RF 收发器中定义的参数从上层协议层配置 RF 收发器的函数调用之外，各种 RF 收发器都具有它们特殊的配置参数，这些配置参数很难统一在同一框架下。对于这些配置，默认值（在 RF 收发器启动并运行之后通常不会修改）在 RF 收发器目录下一个独立的配置文件中进行配置。MiMAC 规范不会调控这些控制变量中的每个设置。要了解并根据需要修改这些配置参数，请参见 RF 收发器数据手册。

通信是 RF 收发器的主要功能。双方进行可靠的数据包通信需要两个步骤：发送和接收。

发送数据包

本节定义用于发送数据包的接口。

为了将收发器配置为以所需方式发送数据包，定义了以下结构，并用作一个输入参数来发送数据包。

```
typedef struct
{
        union
        {
            BYTE  Val;
            struct
            {
                BYTE    packetType: 2;
                BYTE    broadcast: 1;
                BYTE    secEn   : 1;
                BYTE    repeat  : 1;
                BYTE    ackReq: 1;
                BYTE    destPrsnt: 1;
                BYTE    sourcePrsnt: 1;
            } bits;
        } flags;

        #if defined(IEEE_802_15_4)
                BOOL    altDestAddr;
                BOOL    altSrcAddr1;
                BYTE    *DestPANID;
        #endif
        BYTE    *DestAddress;
} MAC_TRANS_PARAM;
```

以下结构对每个变量进行了介绍：

- flags 是用于发送数据包的配置的集合。 flags 参数包含以下配置选项：
  - `packetType` 用于定义数据包类型。在通用 MAC 策略中，数据包类型的定义如下：

0b01	数据型数据包	—
0b10	应答数据包	在 MiMAC 层中进行处理，因而，上层协议层不需要发送此类数据包；不使用它。
0b11	命令数据包	—
0b00	0b00 保留	保留用于某些收发器硬件或协议层的特殊功能。 并非所有收发器都支持它。MiMAC 将直接发送此类数据包，不进行任何进一步的处理。

  - `broadcast` 元素定义数据包是单播还是广播。将该位置 1 会使能当前数据包的广播操作。该位置 1 之后，应清零 `destPrsnt` 位，并且 MAC 报头中的目标地址字段不应存在。
  - `secEn` 元素指示发送数据包是否需要保护 MAC 有效载荷。安全级别和安全密钥应先前已在应用层中进行了定义。来自 Microchip 上层协议层的 MAC 有效载荷应为明文。 MiMAC 层会自动添加安全辅助报头，如 **“MiMAC 安全模块 ”** 一节中所述。MAC 有效载荷将在通过空气进行发送之前进行加密和 / 或认证处理。
  - `repeat` 元素指示该报文在发送方和接收方之间需要一个中继器，以传递该报文。中继器只会转发超出彼此无线电射程之外的节点之间的数据包。这可防止中继器转发过量的报文，即使发送方和接收方可以直接进行通信。

  - 接收方使用 ackReq 元素来发送应答，以确保可靠地传递报文。 MiMAC 层 （硬件或软件）应能够处理接收方发送的应答帧。如果未在预定义时间内接收到应答，发送方 MiMAC 层应能够处理重发。
  - dstPrsnt 元素指示是否在 MAC 报头中包含目标地址。如果数据包为广播或目标地址通过推断获得，则可以省略目标地址。如果使用 2 字节 CRC，则目标地址只能通过推断获得。即使目标地址是通过推断获得的，还是应总是向 MiMAC 层提供一个有效的目标地址，用于计算 CRC。
  - sourcePrsnt 元素指示是否在 MAC 报头中包含源地址。当器件首次尝试建立连接时，需要在 MAC 报头中包含源地址，以便对等节点可以知道与之通信的节点。但对于应用层数据，源地址是可选的，具体取决于应用的需求。
- altDestAddr 是一个布尔值，指示目标地址是备用地址还是永久地址。
- altSrcAddr 是一个布尔值，指示源地址是备用地址还是永久地址。

注：	对于 altDestAddr 和 altSrcAddr，设置为 0 表示源地址是永久地址，而设置为 1 则表示是备用地址。只有在 IEEE 802.15.4 模式下，当 RF 收发器能够通过永久地址或备用网络地址发送数据包时，该字段才有效。

- DestPANID 是指向目标 PANID 的指针。该字段仅在 IEEE 802.15.4 模式下有效，此时， PANID 与目标地址一起用作寻址目标器件的过滤器之一。
- DestAddress 是指向目标地址的指针。在 IEEE 802.15.4 模式下，该地址可以为永久地址或备用网络地址，具体取决于 altDestAddr 的设置。如果广播字段设置为 1，该字段将不会生效。

发送数据包的完整函数格式如下。返回值指示操作是否成功。

```
BOOL MiMAC_SendPacket(MAC_TRANS_PARAM
transParam, BYTE *MACPayload,
BYTE MACPayloadLen);
```

在 MiMAC_SendPacket 函数调用中，transParam 参数用于调控发送选项的所有方面。输入参数 MACPayload 指向要传输的缓冲区。输入参数 MACPayloadLen 指定 MAC 有效载荷的长度。

接收数据包

除了发送数据包之外，收发器另一个最重要的功能是接收数据包。由于在上层协议固件运行时，随时都可能接收到数据包，有两种方法可以处理报文：

- 第一种方法 —— 通过调用回调函数，并让上层处理层立即处理数据包。

  这种方法的响应速度较快，但需要在层之间（一直到应用层）调用一组函数调用，从而会快速填满堆栈空间。

- 第二种方法 —— 通过解释数据包，将它存储在全局变量中，并通过标记某个事件让上层协议层知道某个数据包已可用。

  该选项可以更好地适应通用 Microchip 协议栈的状态机架构。

这两种方法都不错。

为了实现第二种方法，定义了以下类型，用于包含来自数据包的信息：

```
typedef structure
{
        union
        {
            BYTE Val;
            struct
            {
                BYTE    packetType: 2;
                BYTE    broadcast: 1;
                BYTE    secEn   : 1;
                BYTE    repeat  : 1;
                BYTE    ackReq: 1;
                BYTE    destPrsnt: 1;
                BYTE    sourcePrsnt: 1;
            } bits;
        }  flags;

        BYTE *  SourceAddress;
        BYTE *  Payload;
        BYTE    PayloadSize;
        BYTE    RSSI;
        BYTE    LQI;

        #if defined(IEEE_802_15_4)
                BOOL    altSourceAddress;
                BYTE *  SourcePANID
        #endif

} MAC_RECEIVED_PACKET;
```

该 MAC_RECEIVED_PACKET 结构中定义的元素 flags 用于指定所接收数据包的配置。flags 元素的定义与结构 MAC_TRANSMIT_PARAM 中定义的 flags 实际上是相同的。

元素 SourceAddress 是一个指向源地址的指针（如果数据包中存在源地址）。如果 RF 收发器支持 IEEE 802.15.4，则源地址可以为永久地址或备用网络地址，这由 altSourceAddress 元素中的设置决定。此外，只有在 IEEE 802.15.4 模式下，SourcePANID 元素可用于指定发送器的 PAN 标识符。

元素 Payload 是一个指向 MAC 有效载荷缓冲区的指针。MAC 有效载荷大小在 PayloadSize 元素中指定。如果 MAC 有效载荷已经过加密和 / 或认证处理，则传递给 Microchip 上层协议层的有效载荷应已进行解密和 / 或认证检查。此外，还应从传递给上层协议层的有效载荷中删除安全辅助报头。只有 flags 元素中的 secEn 位向上层协议层指示是否对原始数据应用了安全性。

RSSI 和 LQI 元素指示所接收数据包的物理方面。RSSI 代表接收数据包的平均信号强度，而 LQI 表示所接收数据包的信号质量。并不是所有 RF 收发器都支持 RSSI 和 LQI。

接收数据包之后，数据包的内容将被放入该结构，并提供给上层协议层进行处理。

上层协议层需要定期检查是否已接收到数据包。当上层协议层发现已接收到数据包时，它将立即处理该全局结构。上层协议层通过两个函数调用接口来处理接收到的数据包：MiMAC_ReceivedPacket 和 MiMAC_DiscardPacket。

MiMAC_ReceivedPacket

上层协议层会通过调用MiMAC_ReceivedPacket函数调用来定期检查是否已接收到数据包。它没有任何输入参数，并返回一个布尔值，指示是否已接收到数据包。完整的函数格式如下：

```
BOOL MiMAC_ReceivedPacket(void);
```

MiMAC_ReceivedPacket 函数调用会运行 MiMAC 协议栈，并检查是否已接收到数据包。

当该函数调用的返回值为 TRUE 时，说明 MAC_RECEIVED_PACKET 结构的内容已填充，并准备好由上层协议层进行处理。

MiMAC_DiscardPacket

上层协议层通过调用MiMAC_DiscardPacket函数调用来通知 MiMAC 层当前接收到的数据包已处理，可以丢弃。丢弃已处理的数据包非常重要。否则，主机 MCU 可能会很快用尽存储器资源。MiMAC_DiscardPacket 函数调用没有任何输入参数，不返回任何值。完整的函数格式为：

```
void MiMAC_DiscardPacket(void);
```

特殊功能

除了发送或接收报文之外，大多数 RF 收发器还能够执行一些特殊的功能，以确保 RF 收发器能够在最佳条件下工作。两个可能对于所有协议层都非常有价值的常用功能是执行能量扫描和节能。 MiMAC 编程接口中定义了以下函数：

- MiMAC_ChannelAssessment
- MiMAC_PowerState

MiMAC_ChannelAssessment

MiMAC_ChannelAssessment 函数调用会执行信道评估，以确定信道或频率是否适合于可靠的通信。

该操作不应与发送之前使用的 CSMA-CA 空闲信道评估相混淆。

CSMA-CA 操作将在较低 MAC 层中在 MiMAC_SendPacket 函数调用中进行，以避免与邻近对等节点在同一时间发送 RF 数据包。在协议层进行的信道评估操作（有时也称为能量检测扫描）主要用于检查不同频率下的噪声级别，以确定在通信中使用哪种频率。该操作通常由协议层调用，在启动网络之前或频率捷变操作之前调用。

完整的函数格式为：

```
BYTE MiMAC_ChannelAssessment
(BYTE AssessmentMode);
```

MiMAC_ChannelAssessment 函数调用具有一个输入参数，用于指示信道评估模式。

存在两种可能的信道评估模式：

- 能量检测

  测量工作信道中来自所有可能来源的总噪声水平。通常情况下，能量检测评估用于评估来自自然源的噪声、来自其他调制方式的信号，以及来自调制方式相同的邻近无线节点的信号。

- 载波侦听

  它用于检测同类 RF 收发器的通信。它仅测量该 RF 收发器可以接收和解释的通信的信号强度。它通常用来避免在存在几个同类邻近无线节点的频率下运行网络。

输入模式参数指定使用哪种方法进行信道评估。并不是所有 RF 收发器都能够通过其中全部或任意评估模式来执行信道评估。对于不支持任何评估模式的收发器，将不支持该函数。对于仅支持一种评估模式的收发器，其他模式将被丢弃。MiMAC_ChannelAssessment 函数调用会将评估结果返回到上层。返回值越高，表示环境噪声越高。

MiMAC_PowerState

只有可以在空闲时进入休眠状态以节能的节点，才可以使用 MiMAC_PowerState 函数调用。它不是一个 MAC 函数；它是一个直接的 PHY 函数。此处，MiMAC 接口只是将该函数直接传递给 PHY 层。完整的函数格式为：

```
BOOL MiMAC_PowerState(BYTE PowerState);
```

MiMAC_PowerState函数调用具有一个输入参数 PowerState，用于指示收发器所需的功耗状态。定义了两种通用的功耗状态：

- DEEP_SLEEP—— 收发器功耗最低的休眠状态。
- OPERATE—— 收发器的完全工作状态。

0x00 和 0xFF 之间的所有值都可能表示某种功耗状态。详细的定义取决于特定的收发器。通常，希望将靠近深度休眠状态的低功耗休眠状态定义为0x01，并随着休眠模式消耗更高的电流而增大其值。

结论

对于寻求短距离、低数据速率无线解决方案的开发人员，在多种频段、不同数据速率和其他功能之间的选择有很多。Microchip MAC（MiMAC）规范为应用开发人员提供了一个低成本、低复杂度的设计解决方案。它使 Microchip 支持的 RF 收发器可以与任何 Microchip 专有无线协议对接。强烈建议本应用笔记的读者参考另一应用笔记 AN1284，《Microchip 无线（MiWi™）应用程序编程接口 ——MiApp》。

- MiApp 旨在使开发人员可以灵活地使用任何 Microchip 专有无线协议，而无需或几乎无需在应用层中进行修改。
- MiMAC 旨在使开发人员可以利用相同的 Microchip 专有协议层灵活地使用任何 Microchip RF 收发器。

MiMAC 和 MiApp 配合工作，可以为 Microchip 的客户在整个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。

请访问 Microchip 网站 （www.microchip.com）获取以下应用笔记：

- AN1284，《Microchip 无线 （MiWi™）应用程序编程接口 ——MiApp》
- 《MRF24J40 数据手册》 ——IEEE 802.15.4 2.4 GHz RF 收发器 （DS39776B_CN）
- 《MRF49XA 数据手册》 ——ISM 波段 Sub-GHz RF 收发器 （DS70590B_CN）
- AN1066，《MiWi™ 无线网络协议栈》（DS01066A_CN）
- AN1204，《Microchip MiWi™ P2P 无线协议》（DS01204A_CN）

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。

ISBN：978-1-62076-549-4

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。

全球销售及服务网点

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 - Drunen
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

11/29/11