应用笔记

在 STM32WB 系列上开发 Zigbee®睡眠终端设备

引言

本应用笔记介绍了在 STM32 系列微控制器上开发 Zigbee®睡眠终端设备（SED）时的必要考虑因素。

Zigbee®睡眠终端设备是一类特殊的节点，能够在极低功耗模式下运行，并且通常采用电池供电。为了节省电力，它们会以低功耗状态进行功耗管理，并在大部分时间关闭无线网络。这些设备的 RxOnWhenIdle 属性设置为“false”。

根据 IEEE 802.15.4 MAC 标准，这些设备属于 RFD（精简功能设备）。在低功耗无线技术中，Zigbee 传感器被称为睡眠终端设备。

1 概述

本文档适用于 STM32WB 系列基于 Arm®的微控制器。

注意： *Arm® 是 Arm Limited（或其子公司）在美国和/或其他地区的注册商标。*

术语表

表 1. 术语表

术语	定义
APS	应用支持子层
PAN	个人局域网
TCSO	信任中心换出
ZCL	Zigbee®群集库
OTA	无线升级
FFD	全功能设备
RFD	精简功能设备
FP	帧挂起
BO	信标顺序

参考文档

表 2. 参考文档

参考	文档
[1]	05-3474-22《Zigbee 规范》R22(1)
[2]	AN5500：*《ZSDK API 在 STM32WB 系列上实现 ZigBee®》*

1. *该 URL 属于第三方。它在文档发布时处于活动状态，但意法半导体对 URL 或参考材料的任何变更、转移或停用不承担责任。*

2 应用设计

睡眠终端设备经过设置，可在大多数时间处于关闭模式，从而节省功耗。这意味着网络中的其他设备大多无法访问睡眠终端设备。与在网络上收集和推送数据的设备相比，这限制了睡眠终端设备的应用。

关于睡眠终端设备的应用场景，传感器是一个值得注意的例子。传感器设备定期唤醒，获取读数，并将结果推送到常开集中器或网关设备。在该情形下，睡眠终端设备能够正确地工作。因为传感器数据量比较小，并且适合在非常小的唤醒时段内传输。此外，睡眠终端设备能够传输大量数据。

睡眠终端设备可以使用 OTA 固件群集下载整个固件映像。OTA 群集设计灵活，可确保设备对下载进行控制。得益于该设计，即使客户端需要花费数天时间用于完成固件的下载，服务器端也不会有影响。

有些低功耗设备可缓慢地为电容器充电。这些低功耗设备经过专门设计，能够在电容器充电时被唤醒。它们仅在电容器的电量允许时，才会保持开启状态。当电量耗尽时，设备进入睡眠状态。因此，重要的是，应用要确保设计灵活并且由睡眠终端设备驱动。

如果网络上的其他设备需要与睡眠设备通信，则通过双向镜像和通知标志使用智能能源。在智能能源中，常开设备对驻留在睡眠设备上的属性的副本进行维护。该副本被称为镜像。

远程设备当需要从睡眠设备读取数据时，可以从镜像中读取数据。睡眠设备的作用是在处于唤醒状态的短暂间隔内，对镜像上的数据进行维护。双向通信采用双向镜像实现。发送到睡眠设备的数据允许远程设备写入到镜像中。

此外，更改镜像时，会设置通知标志。当睡眠设备唤醒时，它会检查通知标志，以便在必要时获取更新，并在完成后重置通知标志。

下文展示了应用中的特定设计如何为睡眠终端设备提供支持。

3 创建睡眠终端设备

设备的配置通过传递给 ZbStartup()的配置结构完成。

ZbStartupT 是一种具有许多选项的丰富结构。传统上，应用首先使用 ZbStartupConfigGetProDefaults()或 ZbStartupConfigGetProSeDefaults()，针对智能能源网络对结构进行初始化。

作为终端设备，它本身并不能形成网络，而只能加入。ZbStartupT config.startupControl 不可以是 ZbStartType of ZbStartTypeForm，而是在重新加入时必须是 ZbStartTypeJoin 或 ZbStartTypeRejoin。

ZbStartupConfigGetProDefaults()将功能配置为 IEEE 802.15.4 全功能设备，该设备为路由、RxOnWhenIdle（非睡眠）、电源供电设备。

- 清单 1：设备配置样例：

```
/* Reduced Capability Device */
config.capability &= ~MCP_ASSOC_CAP_DEV_TYPE;

/* disable RxOnWhenIdle */
config.capability &= ~MCP_ASSOC_CAP_RXONIDLE;

/* battery powered (i.e. not mains powered) */
config.capability &= ~MCP_ASSOC_CAP_PWR_SRC;
```

终端设备应处于睡眠状态：RxOnWhenIdle=false。这意味着睡眠终端设备和非睡眠终端设备的调试过程类似。非睡眠终端设备很少见，如果设备的无线电始终打开，则使该设备成为路由器更实用。一般，非睡眠终端设备仅用于测试情况。

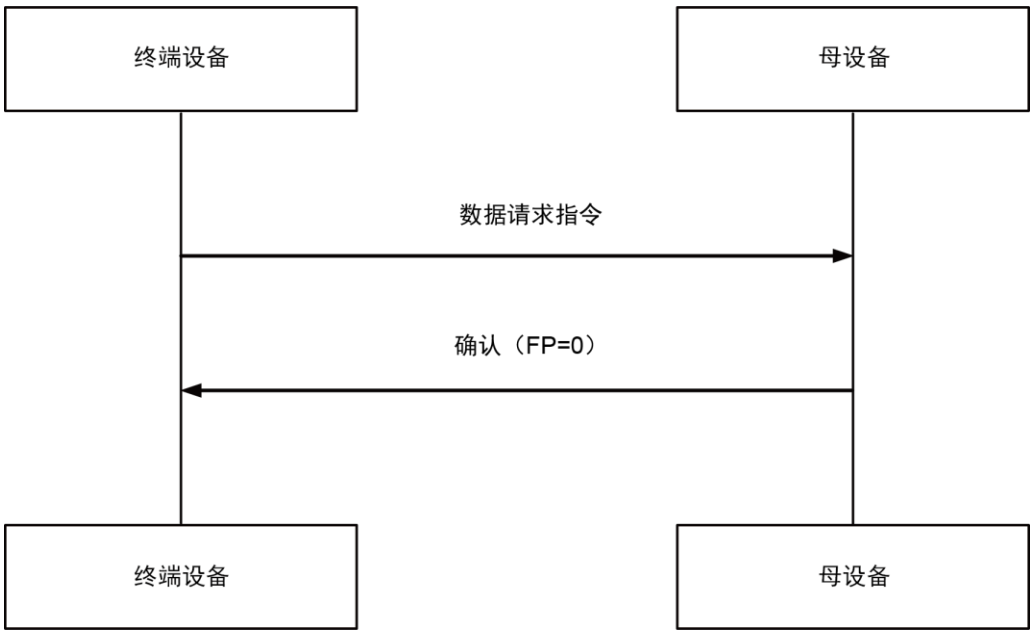
4 睡眠终端设备中的 Zigbee 网络行为

睡眠终端设备必须始终与另一个节点配对。它必须是 IEEE 802.15.4 FFD（全功能设备）。它可以是路由器或协调器，即睡眠终端设备的母设备和睡眠终端设备的子设备。

睡眠终端设备不直接与网络上的任意节点通信。所有通信都通过母设备进行。母设备为每个子设备维护所谓的间接消息队列。在其唤醒周期中，子设备对其母设备执行 MAC 数据轮询。当间接消息队列中没有数据时，母设备使用 MAC Ack 进行响应，并清除 FP（帧挂起）位。为了节省电池，子设备可以选择立即返回睡眠状态。

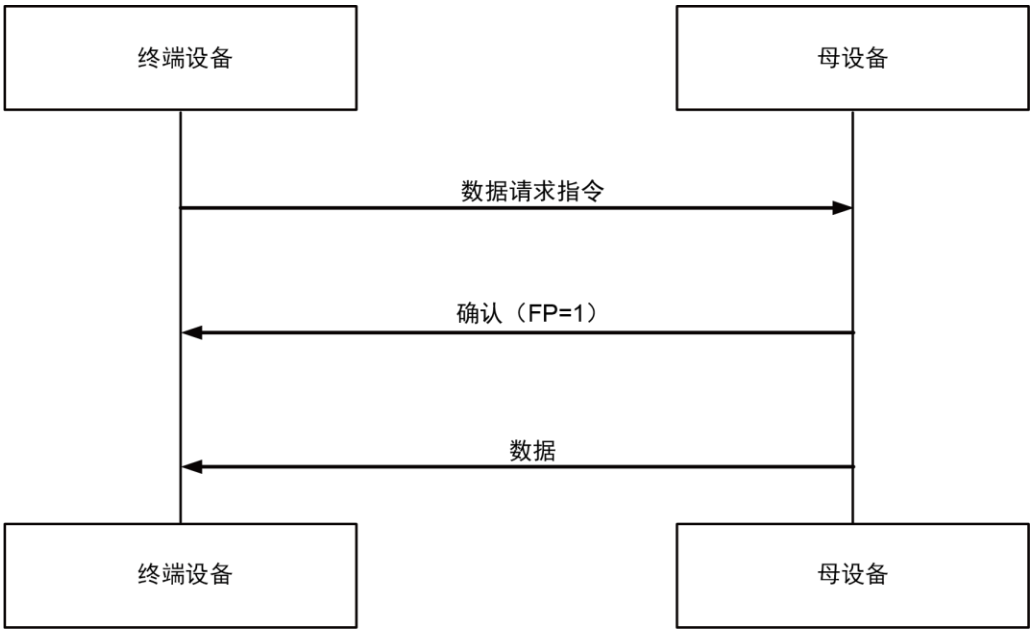
下图展示了在没有帧挂起的情况下请求数据。

图 1. 在不挂起帧时请求数据



下图展示了在帧挂起的情况下请求数据。

图 2. 在挂起帧时请求数据



母设备为睡眠子设备保留消息的时间量由 nwkTransactionPersistenceTime 表示，对于 2.4 GHz O-QPSK PHY，该时间量 7.68 秒。

间接消息队列机制原则上并非用于充当任意网络节点与睡眠终端节点通信的通用机制。其目的是使睡眠终端节点在加入过程中与其母设备进行关联。

通常，睡眠终端设备需要使用具有特定拉动机制的各种应用，确定是否有数据在等待它们，以便其选择何时检索该数据并在考虑能源管理的情况下进行平衡。MAC 数据轮询机制是使用关联时使用的拉动机制。

智能能源使用不同的应用级机制，允许睡眠终端设备拉取数据。智能能源双向镜像允许任意节点随时从镜像中检索任意数据，尤其是在睡眠节点处于睡眠状态时。除此之外，还提供了事件标志。睡眠节点在其开机周期内进行读取，这告知睡眠节点镜像上有数据在等待。睡眠终端节点拉取事件标志并决定拉取镜像数据的理想时机，同时对其功耗进行管理。

双向镜像特定于智能能源。目前尚无通用拉动机制。由应用确定如何向睡眠终端设备广播数据的可用性。然后由睡眠终端设备来处理该问题。

5 轮询快慢速率

MAC 数据轮询有两种速率，慢速和快速。慢速完全由设备决定。可以慢到每 30 分钟一次，甚至每天一次，具体取决于应用本身。从子设备老化的角度来看，当母设备确定子设备不再存在并且可能已老化时，就务必需要了解慢速轮询速率。这与第 7 节“终端设备保持连接”中描述的保持连接机制有关。睡眠终端设备在慢速轮询之间进入睡眠状态。

在快速轮询模式下，终端设备保持唤醒，并且以值小于 nwkTransactionPersistenceTime（7.68 秒）的快速轮询速率进行轮询。通常使用 7.5 秒的快速轮询值。

在以快速轮询速率进行轮询时，终端设备能够接收消息。通常，根据协议栈的需要在内部启用和禁用快速轮询，而无需应用参与。快速轮询的主要用途在于，确保终端设备在加入过程中可以从信任中心接收消息。允许信任中心在几个跃点之外。

当终端设备需要从远程设备接收数据（如拉取数据）时，该终端设备通过调用 ZbNwkFastPollRequest()来启用快速轮询。由设备确定何时启用快速轮询以及启用快速轮询多久。这是根据能源管理考虑因素确定的。对于每个快速轮询请求，当应用调用 ZbNwkFastPollRelease()时，必须释放匹配的快速轮询。

6 轮询控制群集

终端设备支持轮询控制群集。轮询控制群集启用或禁用由其他设备进行的快速轮询。如要使用轮询控制群集，远程应用必须验证要与之通信的设备是否为终端设备。应用必须支持轮询控制客户端群集，并且终端设备必须支持轮询控制服务器。

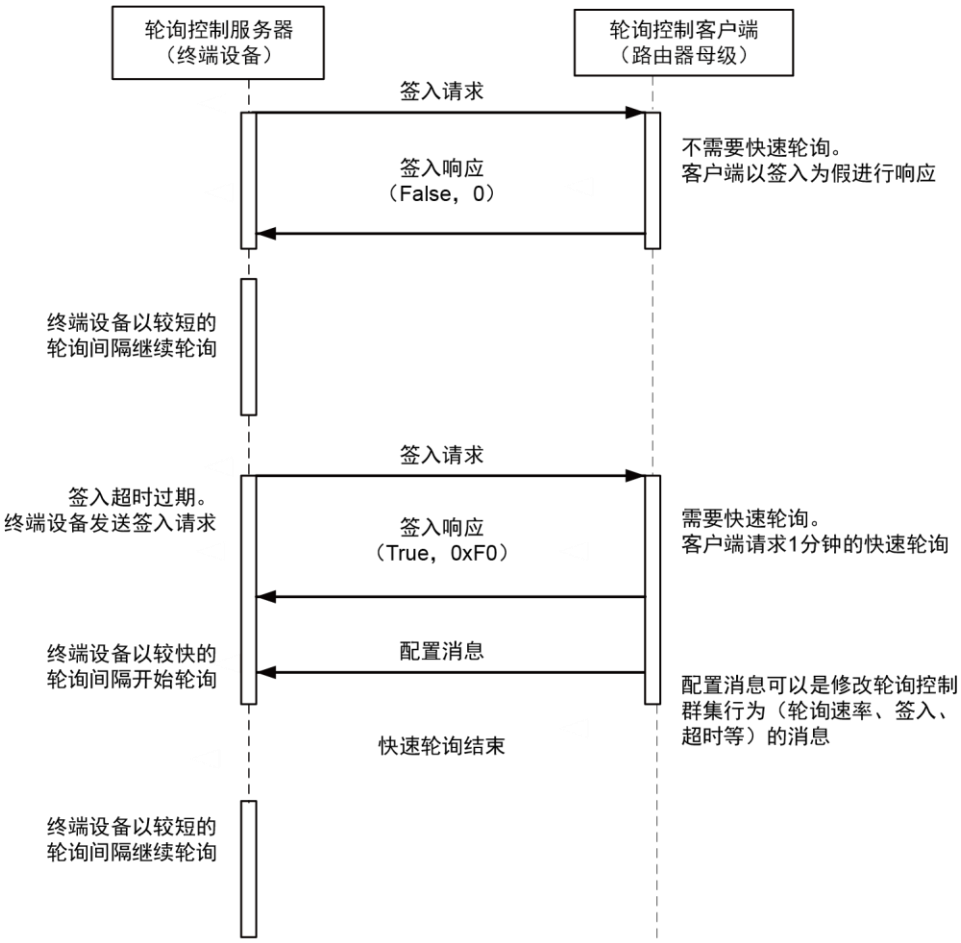
在需要严格控制其电源使用的睡眠终端设备上，不得支持此机制。远程应用未确认此项要求，并且可以通过以过长间隔或过高频率强制终端设备进入快速轮询模式，快速耗尽终端设备电池。当难以甚至无法更换电池时，这一点尤其重要。

为了在启动前支持轮询控制群集，终端设备必须通过调用 zcl_poll_server_alloc()，将轮询控制群集的实例添加到端点。终端设备必须使用 zcl_poll_server_write_long_poll_intvl()设置长轮询间隔。

查找和绑定机制用于在服务器和网络上的所有轮询控制客户端之间建立绑定。必须通过设置 ZB_BDB_CommissioningMode BDB 属性的 BDB_COMMISSION_MODE_FIND_BIND 位（使用 ZbBdbSet()进行设置），启用查找和绑定。当设置在 BDB 调试模式下时，ZbStartup()在加入网络后自动发起查找和绑定。查找和绑定也可以通过调用 ZbStartupFindBindStart()或 ZbStartupFindBindStartEndpoint()在较晚的时间发起，以创建绑定在此终端设备之后加入的客户端。

在操作过程中，睡眠终端设备必须在每次唤醒时调用 zcl_poll_server_send_checkin()，以发送签入请求，从而发起快速轮询模式。

图 3. 轮询控制流程图



7 终端设备保持连接

网络是动态的，路由器必须能够删除已老化设备的条目，即邻设备、路由或其他关联的设备。然而，终端设备依赖于其母设备来访问网络。为了防止具有较长睡眠间隔的睡眠终端设备老化，可采用相关方法使终端设备保持连接。

对终端设备保持连接方法的支持内置于协议栈中。终端设备以由 nwkEndDeviceTimeoutDefault IB 参数确定的时间间隔，将终端设备超时请求消息发送到其母设备。在默认情况下，设置为 8 分钟。终端设备超时请求消息每 nwkEndDeviceTimeoutDefault/4 发送一次，以确保在 nwkEndDeviceTimeoutDefault 时间段内至少发送三条终端设备超时请求消息。nwkEndDeviceTimeoutDefault 是在启动时通过将 ZbStartupT 配置参数 endDeviceTimeout 配置为 n 进行设置的，该参数产生的超时为 2^n 分钟 n=1…14（2、4、8、……16384 分钟）。特殊值 0（10 秒）和 0xFF（禁用）主要用于测试。启动后，通过 ZbNlmeGetReq() 使用 ZB_NWK_NIB_ID_EndDeviceTimeoutDefaul，终端设备超时也可作为 NIB 参数提供给应用。除此之外，也可以复位该值（但并不常见），在这种情况下，对母设备的下一个终端设备超时请求会请求新值。

Zigbee 协议栈还支持路由器母端的终端设备保持连接功能。该功能在路由器端没有需要应用层考虑的因素。

8 ZCL 响应处理

大多数 ZCL 指令对每个请求都有一个响应。然而，当请求没有结果时，某些指令不会定义响应。在这种情况下，定义了通用默认响应。默认响应的主要用途是传达错误状态。为了减少不必要的 OTA 流量，ZCL 指令具有一个在指令成功时对默认响应进行抑制的标志。如果未收到默认响应，则认为该指令成功。

当 ZCL 请求没有定义的默认响应时，大多数睡眠终端设备必须请求发送 ZCL 默认响应。在最坏的情况下，在具有 APS 重试的多跳路径上，如果请求了 APS Ack，则在发送请求后至多可以在 10 秒内收到错误，或者如果未请求 APS Ack，则至多可以在发送后 3 秒内收到错误。请求成功后，设备可能需要等待至多 10 秒（如果没有 APS Ack，则等待 3 秒）才能知道请求成功。

如果未指定且未预期 ZCL 响应，则请求始终发送默认响应（即使在成功时也是如此）可消除该超时延迟。为了实现这一点，当没有定义标准的 ZCL 响应时，应用应将 ZCL ZbZclClusterCommandReqT 的 noDefaultResp 字段设置为 ZCL_NO_DEFAULT_RESPONSE_FALSE 的值。这会导致在大多数情况下始终发送默认响应。这也意味着 ZCL 业务在不到 10 秒或 3 秒的超时时间内结束，对于希望尽快恢复睡眠的睡眠设备而言，尤其重要。

9 持久性和睡眠唤醒循环

对于某些应用设计，设备无法在整个间隔内始终保持唤醒。届时，应用可选择关机。尽管这会导致通信中断和数据丢失，然而在某些情况下，这算得上是可以接受的折衷方案。一般而言，应用在设计时会考虑避免此类情况的发生。

当设备唤醒时，设备必须调用 ZbStartupPersist()，而不是 ZbStartup()。ZbStartupPersist()会在睡眠之前恢复协议栈状态。然而，在使用 ZbStartupPersist()之前，应用必须添加对持久性的支持。

为了启用持久性，应用必须在原始启动和从睡眠模式唤醒时都调用 ZbPersistNotifyRegister()。应用可以提供回调函数，例如：

```
MyPersistenceCallback(structZigBeeT *zb, void*cbarg)
```

在该回调中，应用负责将持久性数据保存到 Flash 存储器中。通过调用 ZbPersistGet()，从 Zigbee 协议栈获得要持久化的数据的当前快照。应用不得更改持久性数据块，必须保留并在以后恢复。

有些 FLASH 内存的系统会受到诸如 FLASH 写入次数等的限制。应用可以通过 BDB 参数 ZB_BDB_PersistTimeoutMs 调整调用回调的频率，这个参数描述了持久性更新动作之间的最小延迟。应用还可以使用 ZbPersistNotifyControl()，暂时禁用或重新启用持久性回调。

版本历史

表 3. 文档版本历史

日期	版本	变更
2022 年 1 月 14 日	1	初始版本。

目录

表格索引

图片目录

重要通知 - 请仔细阅读

意法半导体公司及其子公司（“意法半导体”）保留随时对 ST 产品和/或本文档进行变更、更正、增强、修改和改进的权利，恕不另行通知。买方在订货之前应获取关于意法半导体产品的最新信息。意法半导体产品的销售依照订单确认时的相关意法半导体销售条款。

买方自行负责对意法半导体产品的选择和使用，意法半导体概不承担与应用协助或买方产品设计相关的任何责任。

意法半导体不对任何知识产权进行任何明示或默示的授权或许可。

转售的意法半导体产品如有不同于此处提供的信息的规定，将导致意法半导体针对该产品授予的任何保证失效。

ST 和 ST 标志是意法半导体的商标。关于意法半导体商标的其他信息，请访问 www.st.com/trademarks。其他所有产品或服务名称是其各自所有者的财产。本文档中的信息取代本文档所有早期版本中提供的信息。