

STM32WB 系列微控制器上的动态并发模式 BLE/Zigbee®入门指南

引言

本文档概述了 STM32WB 系列微控制器上的动态并发模式 Bluetooth®低功耗（BLE）/Zigbee®。

STM32WB 系列微控制器支持 Bluetooth® 5.0 和 IEEE 802.15.4 无线标准。

有些用例需要动态并发模式，以便通过 Bluetooth®低功耗设备控制 Zigbee®或 Thread®网络。动态设备必须能够随时使用射频分时方案处理两个协议。

1 概述

本文档适用于 STM32WB 系列基于双核 Arm®的微处理器。

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。



1.1 术语表

表 1. 术语表

缩略语	定义
API	应用编程接口
APS	应用支持子层
BDB	基本设备特性
BLE	Bluetooth®低功耗
CI	连接间隔
GAP	通用应用配置文件
GATT	通用属性配置文件
HAL	硬件抽象层
IAS	入侵报警系统
IoT	物联网
IPCC	处理器间通信控制器
MAC	介质访问控制
NVM	非易失性存储器
PAN	个人局域网
RTSM	射频分时管理器
SED	休眠终端设备
ZCL	Zigbee®集群库
ZDO	Zigbee®设备对象

1.2 参考文档

表 2. 参考文档

参考	文档名称
[1]	使用 STM32WB 微控制器系列构建无线应用 (AN5289)
[2]	STM32WB 系列的 Zigbee®永久数据管理非易失性存储器 (AN5492)
[3]	创建 STM32WB 的制造商 Zigbee®集群 (AN5491)
[4]	使用 STM32WB 的 Zigbee®集群模板 (AN5498)
[5]	面向 STM32WB 上的 Zigbee®的 ZSDK API (AN5500)
[6]	STM32WB Zigbee®入门指南 (AN5506)
[7]	STM32CubeWB Nucleo 演示固件 (UM2551)
[8]	STM32WB Bluetooth®低功耗无线接口 (AN5270)
[9]	stm32wb-ble-stack-programming-guidelines-stmicroelectronics.pdf (PM071)

2

动态模式简介

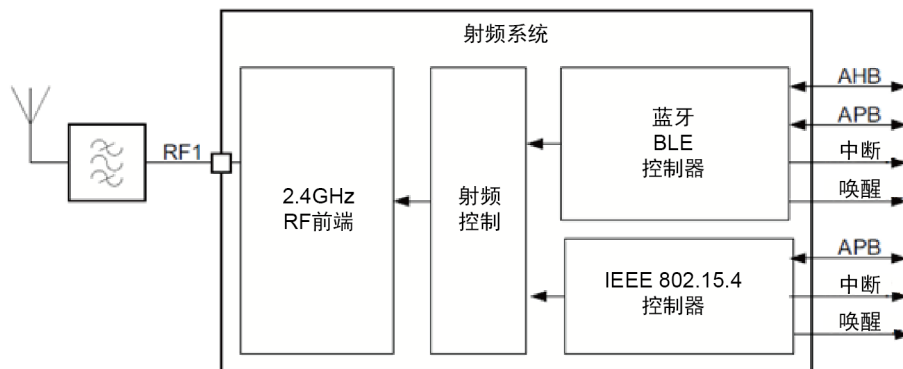
STM32WB 系列微控制器是一种双核多协议无线微控制器，基于运行于 64 MHz 的 Arm® Cortex®-M4 内核（应用处理器），以及运行于 32 MHz 的 Arm® Cortex®-M0+内核（网络处理器）。

该微控制器支持 BLE 网络和多个配置文件，并可以灵活地集成专有 BLE 协议栈。

通用的 IEEE 802.15.4 MAC 层确保 STM32WB 系列可以运行专有协议或协议栈，包括 ZigBee®和 Thread®低功耗 Mesh 网络协议，为设计人员提供更多将设备连接到物联网（IoT）的选项。

为了支持这些特性，每个协议（BLE 或 802.15.4）会在内部结构中共享相同的射频外设，如下图所示。射频外设将在 802.15.4 模式或 BLE 模式下随时进行动态配置。

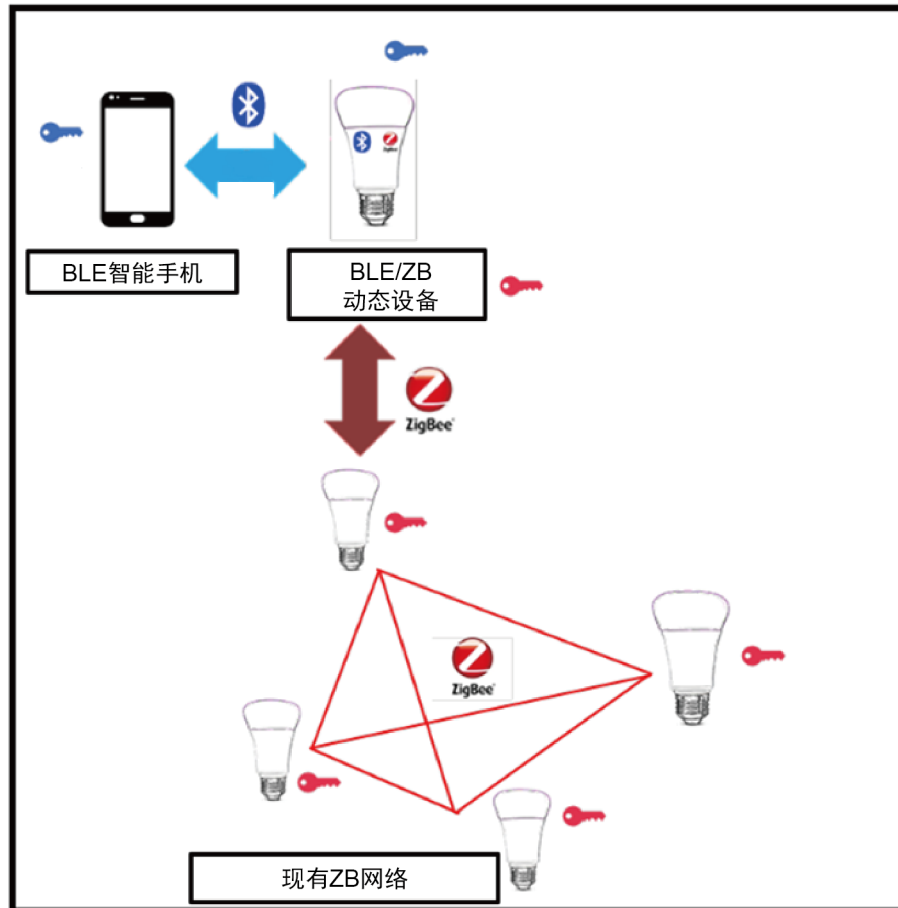
图 1. 射频外设



2.1 动态模式用例

有些用例需要动态模式，以便通过 BLE 设备控制其 Zigbee®或 Thread®网络，因此需要能够随时使用射频分时方案处理两个协议的动态设备。

图 2. 动态模式用例



3

RTSM 架构

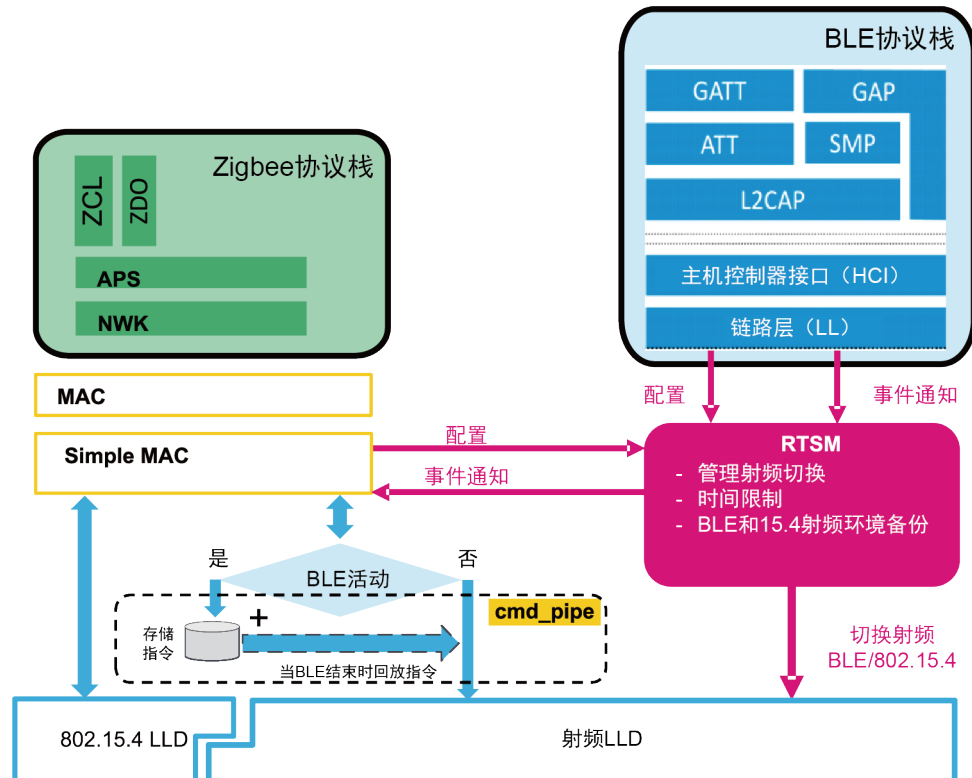
3.1

RTSM 框图

实施 RTSM 可实现 BLE 与 802.15.4 之间的射频切换，因此 BLE 和 Zigbee®协议栈可以在相同的射频上并行运行。实施情况如下图所示。

Zigbee®和 BLE 协议栈说明参见第 4 节。

图 3. RTSM 框图



3.2

RTSM 描述

RTSM 基于 Cortex®-M0+ 实现，以支持 BLE 与 802.15.4 之间的射频切换，因此可以使用相同的射频并行运行 BLE 和 Zigbee®协议栈。

- 在 Zigbee®端，RTSM 集成在 SimpleMAC 层级，该层连接 MAC 层与 802.15.4/射频 LLD。
- 在 BLE 端，RTSM 集成在链路层与射频 LLD 之间。

3.2.1

RTSM 功能

RTSM 实现了动态模式的关键特性：

- 管理 BLE 与 802.15.4 模式之间的射频切换
- 管理 BLE 连接要求的严格时间限制，尤其是连接间隔，以保持连接处于活跃状态。
- 备份和恢复与 BLE 和 802.15.4 相关的射频环境
- 当 BLE 需要高带宽时，使 BLE 的优先级完全高于 802.15.4。

3.2.2 RTSM 规划

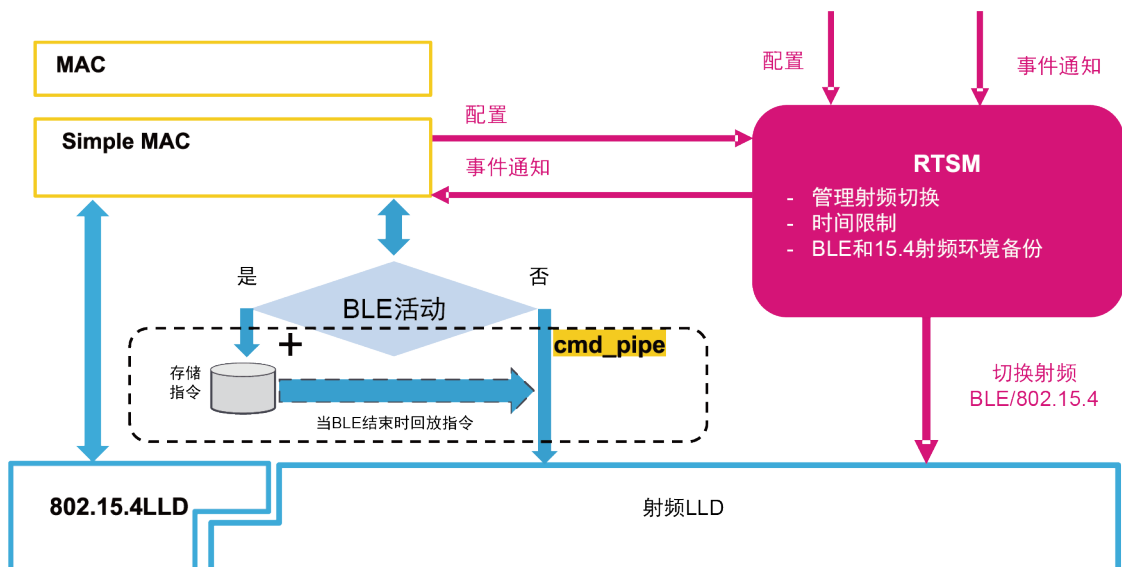
通过 BLE 连接情况规划 RTSM 行为：

- BLE 会在每个连接间隔编写 RTSM 中断程序（以前 1 ms）
 - 此中断通过“CompC Sftimer”中断进行管理
 - 当收到此中断时，RTSM 会保存当前的 802.15.4 射频上下文和 SPI 寄存器，然后切换到 BLE 射频上下文
 - BLE 此时成为射频所有者
- BLE 连接事件结束时，BLE 会通知 RTSM 连接事件已结束，并编写下一连接间隔的时间
 - RTSM 相应编写下一 CompC 中断程序
 - RTSM 将射频和 SPI 寄存器切换回 802.15.4 上下文
 - 802.15.4 此时成为射频所有者，直至收到下次 RTSM 中断
- RTSM 还管理每 3 分钟发生一次的“Sftimer”回绕

3.2.3 Cmd_pipe 模块

与 RTSM 模块一起，还实施了称为 cmd_piped 的新模块（参见下图），以根据射频状态管理 MAC 至射频的接口：

图 4. Cmd_pipe 模块



- MAC 指令直接发送到 802.15.4 控制器（通过 LLD）或发送到射频外设。“cmd_pipe”仅处理“MAC 至射频”指令。
- 如果 802.15.4（Zigbee®模式）拥有射频所有权，则所有“MAC 至射频”指令直接发送到射频。
- 如果 BLE 拥有射频所有权，则所有“MAC 至射频”指令暂时存储到 cmd_pipe 缓存区，等待射频再次分配到 802.15.4。待射频所有权切换至 802.15.4 后，再安全地将这些指令发送到射频。正常执行不直接处理射频功能的其他 MAC 指令。
- 其中大多数指令是通常在 Zigbee®协议栈启动过程中使用的配置指令（SetChannel、SetPower）。

- 运行时可以使用的指令如下表所示。

表 3. Zigbee®运行时指令

指令	执行详情
发送	当 802.15.4 可以使用射频时执行
睡眠	当 802.15.4 可以使用射频时执行
唤醒	当 802.15.4 可以使用射频时执行
能量检测	如果射频不可用，此指令将返回错误代码

- 有些指令可能会返回错误代码，有些指令则不会：
 - 如果射频不可用，则非无效指令将返回错误代码。
 - 在特定发送指令情况下，不返回错误，然后当射频再次可用时执行 Tx。
- 当前大小的 cmd_pipe 支持 30 个待处理的指令。
 - 在调试阶段，主要在 Zigbee®初始化和加入过程中，在 cmd-pipe 中最多可以看到 8 个待处理命令。
 - 如果发生 cmd_pipe 溢出，则将刷新并重新启动 cmd_pipe，且所有待处理指令丢失。

3.2.4 动态模式下的 BLE & Zigbee®协议栈特性

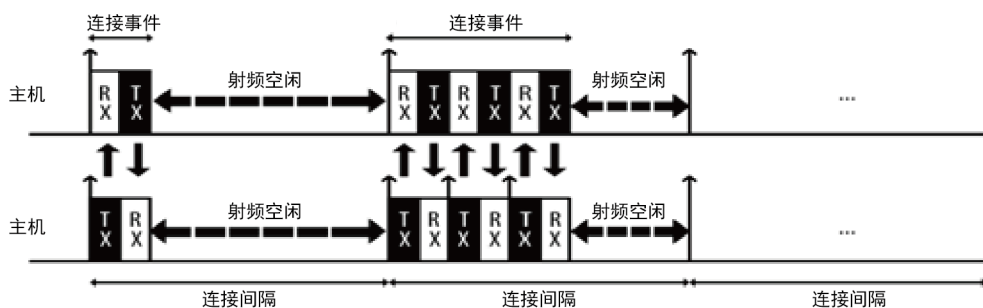
BLE 和 Zigbee®协议栈不了解射频切换机制，认为射频完全专门用于各自的协议栈。

BLE 的时间限制比 Zigbee®更加严格。因此，BLE 的射频访问优先级必须高于基于 802.15.4 的协议。

BLE 协议栈

在 BLE 端，必须以预期连接间隔（CI）精准规划连接事件。RTSM 负责此精确时间。连接间隔和连接事件定义如下表和下图所示：

图 5. BLE 连接事件和连接间隔



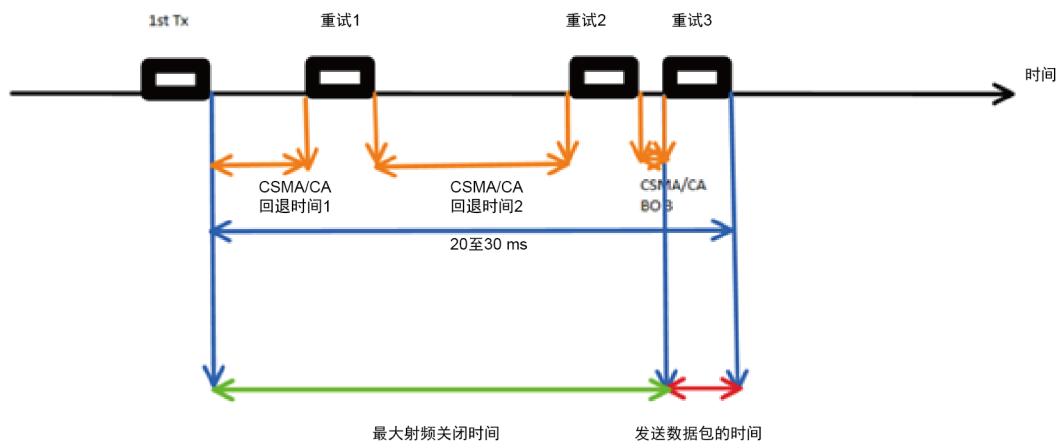
- 连接间隔：两个连接事件之间的时间（7.5 ms - 4 s）。
- 连接事件：主从设备之间在连接间隔内的连续 Rx/Tx 切换。
- RTSM 会将以上面定义的“射频空闲”间隙时间分配到 802.15.4，从而允许 Zigbee®运行。

Zigbee®协议栈

在 Zigbee®端，根据本地和远程 Zigbee®设备需求随时发生 Tx 和 Rx。

- 对于 Zigbee®/MAC Tx 请求：
 - 如果将射频授予 802.15.4，则正常发送。
 - 如果将射频授予 BLE，则将所有射频指令（包括 Tx）存储到 cmd_pipe 中。
 - 当射频切回到 802.15.4 时，发送所有待处理指令（包括 Tx）。
- 对于 Zigbee®/MAC Rx 事件：
 - 所有 Rx 事件通过 Rx 中断进行管理。当射频处于 BLE 模式时，则禁用此类中断。由于 CSMA/CA 重试机制（最多重试 3 次），将在几毫秒后重新规划这些 Rx 事件。
 - 内部调查表明，没有射频（分配给 BLE）的最大时间必须小于 16 ms，以防止丢包。

图 6. Zigbee® Rx/Tx 特性



4 动态 NVM 功能

4.1 动态 NVM 概述

BLE/Zigbee®动态模式还提供 NVM 功能。此模式能够将动态设备的 Zigbee®状态保存到 Flash 中并从中恢复。当 BLE 运行时，NVM 功能支持所有主要的 Flash 操作（读取/写入/擦除）。

为了在发生 Flash 操作时（例如，执行写入或擦除操作时）保持 BLE 特性的完整性，Cortex®-M4 能够在 Cortex®-M0 上执行两个指令（托管 BLE/ Zigbee®协议栈和 RTSM）。这些指令直接由 RTSM 处理。这是因为 Flash 会被阻止，且无法执行获取指令，定义如下表所示：

- 请求下次 BLE 事件前的可用时间（ μs ）。如果在 BLE 模式下，则返回时间'0'。
- 当发生下次 802.15.4 事件时通知。

4.2 BLE/Zigbee®动态模式 Flash 操作

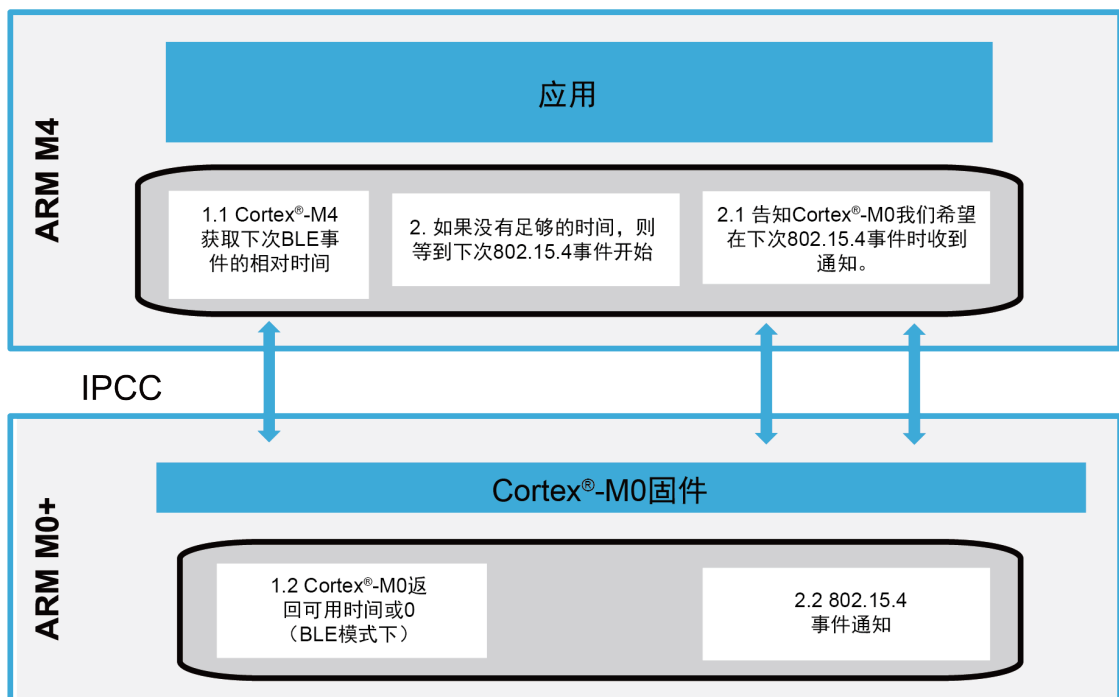
下表概述了 Flash 操作特性：

- Flash 写入持续 5 μs
- 整个页面的 Flash 擦除操作持续 20 ms。

Cortex®-M4 软件基于这些 IPCC 指令组织 Flash 操作序列。动态模式 Flash 驱动器可用于 BLE/Zigbee®动态 NVM 应用（参见第 8.4.4 节）。

有关动态 Flash 操作图例，请参见下图。

图 7. Flash 操作过程图例



5 Zigbee®架构

5.1 Zigbee®概述

Zigbee®是一种基于 IEEE 802.15.4 的通信协议，用于创建无线个人局域网（WPAN）。它意图在低功耗和低带宽限制条件下，提供简单的网络层以及用于创建可互操作解决方案的标准应用配置文件。

该协议涉及：

- 家庭自动化
- 工业控制系统
- 建筑物自动化、HVAC 控制
- 医疗数据收集和监测
- 无线传感器网络。

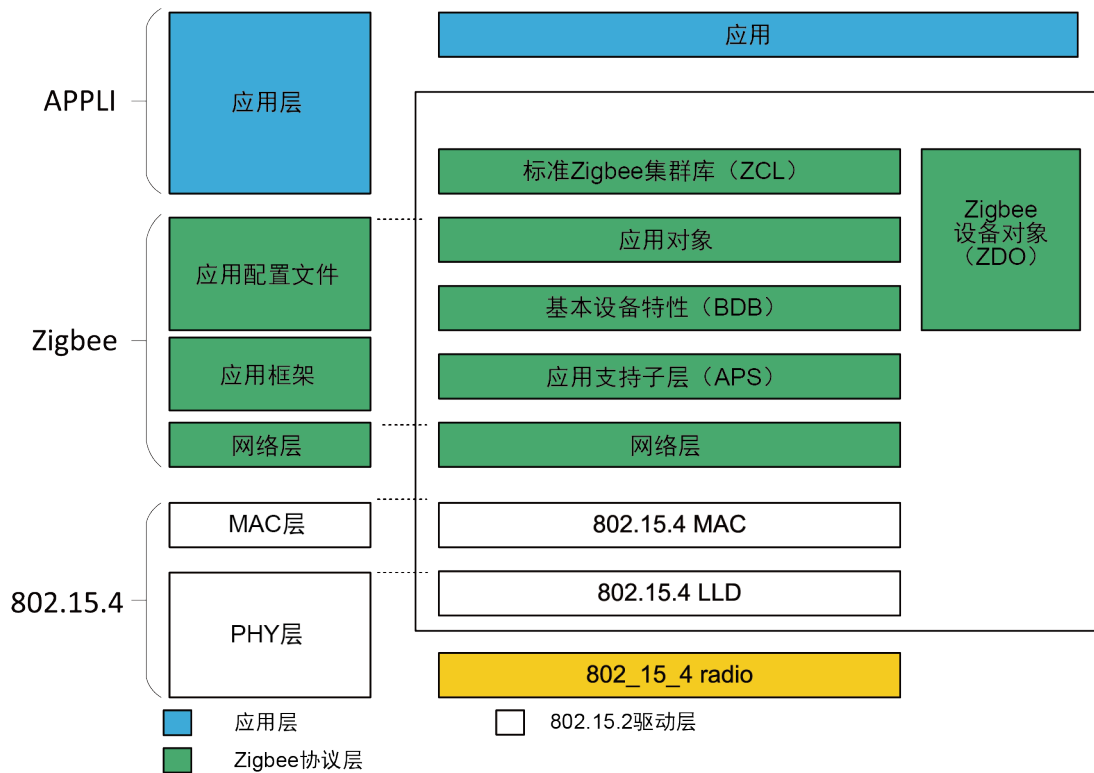
在 10-20 米的典型范围内，2.4 GHz 频率下的吞吐量为 250 kbps。

5.2 Zigbee®协议栈层

如上文所述，Zigbee®基于 IEEE 802.15.4 标准接口。Zigbee®为基于数据包的射频协议提供路由和多跳功能。构建在基于 802.15.4 指定的两层：物理（PHY）层和 MAC 层。

下图描述了 Zigbee®协议栈（绿框）的主要组件，及其与 IEEE 802.15.4 和通用应用层的交互方式。

图 8. Zigbee®协议栈说明



6 BLE 架构

6.1 BLE 概述

STM32WB 系列 BLE 架构将 BLE 配置文件和应用分开：即在 Cortex®-M4 上运行应用，而 BLE 协议栈位于 Cortex®-M0+ 中。

BLE 协议栈处理链路层、通用属性配置文件（GATT）层和通用访问配置文件（GAP）层。链路层直接连接物理 2.4 GHz 射频。

基于所选配置和 10 米的典型范围，将达到 1 或 2 Mbps 的理论 BLE 5.0 模式吞吐量值。

6.2 BLE 协议栈层

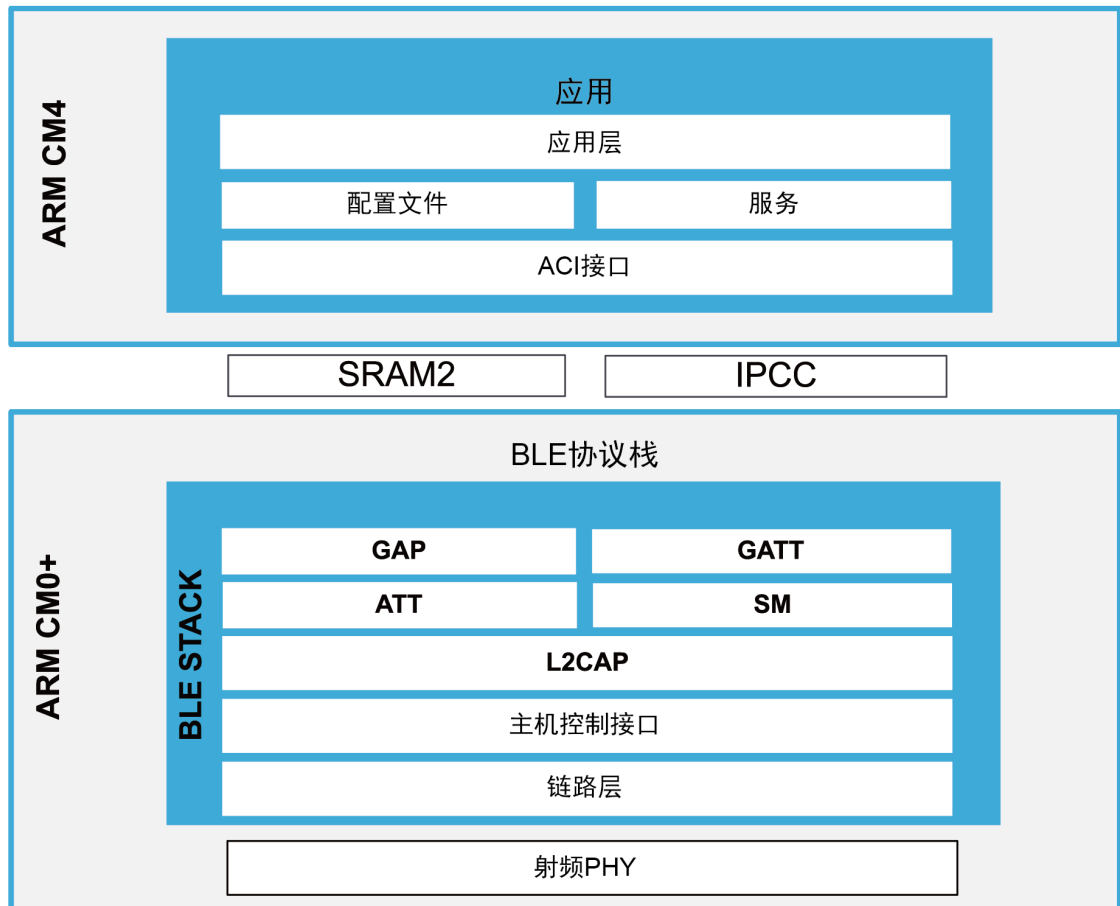
应用由 Cortex®-M4 内核进行管理，详情如下：

- 收集并计算将通过 BLE 传输的数据。
- 如需传输数据，使用 BLE 协议栈服务和功能。

BLE 协议栈由 Cortex®-M0+ 内核进行管理，详情如下：

- 借助 GATT 配置文件与应用层进行通信
- 通过 HCI 协议实现 LE 控制器与 LE 主机
- “链路层”管理所有射频物理层交互

图 9. BLE 协议栈说明



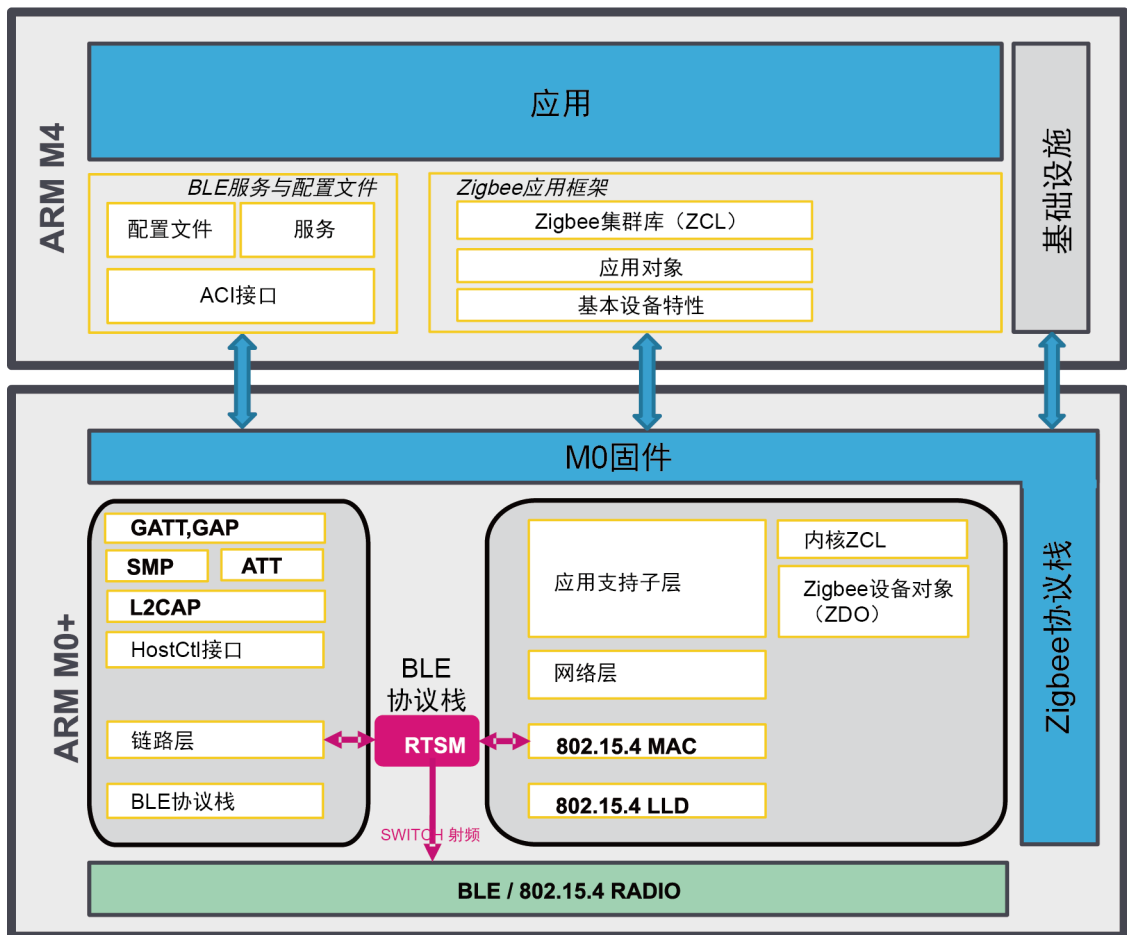
7 STM32WB 系列上的 BLE/Zigbee®动态模式

7.1 架构概述

下图概述了总体架构。图中尤其展示了 Cortex®-M4 与 Cortex®-M0 内核之间的分割。Cortex®-M0 上运行的所有代码均以二进制库形式提供（更多详情参见第 7.2 节）。

客户仅访问 Cortex®-M4 内核，而将 Cortex®-M0 内核上运行的固件视为黑盒。框架会隐藏 Cortex®-M4 与 Cortex®-M0 之间的所有通信。分配用于 Zigbee®和 BLE 的专门 IPCC 通道。

图 10. BLE/Zigbee®动态模式架构



- Zigbee®协议栈在 802.15.4 MAC 层上运行，该层本身会使用负责控制射频的 802.15.4 LLD（低级驱动）提供的服务。
- BLE 协议栈在控制射频的 BLE LLD 上运行。
- RTSM 负责连接两个协议栈并管理两个射频之间的切换。

7.2 支持的动态固件

在动态模式下，BLE 和 Zigbee®协议栈并行运行，从而可以同时使用 BLE 和 Zigbee®应用。

动态模式固件包括 BLE 和 Zigbee®协议栈：

- BLE 协议栈已获得 BLE 5.0 认证。
- **STM32WB 系列**器件支持两种 Zigbee®协议栈：FFD（全功能设备）和 RFD（精简功能设备）。这些协议栈已获得 Zigbee® PRO 2017（第 22 版）认证，详情如下表所示。

表 4. 协议栈固件关联

支持的无线协议栈	相关的固件
Zigbee® FFD + Bluetooth®低功耗 5.0	stm32wb5x_BLE_ZigBee_FFD_dynamic_fw.bin
Zigbee® RFD + Bluetooth®低功耗 5.0	stm32wb5x_BLE_ZigBee_RFD_dynamic_fw.bin

- FFD 可以在网络中扮演任何角色：
 - 路由器
 - 协调器
 - 终端设备。
- RFD 仅支持终端设备角色。RFD 比 FFD 的占用空间小。当构建作为“休眠终端设备”的应用，以实现最佳低功耗时，必须使用 Zigbee® RFD 协议栈构建应用。

这些二进制固件用于动态并发模式应用。此类应用的示例参见：

Projects\P-NUCLEO-WB55.Nucleo\Applications\BLE_ZigBee 目录。

重要：

在 **STM32WB 系列**上运行任何 BLE/Zigbee®应用前，应检查是否将正确固件下载到 Cortex®-M0。如果未下载正确固件，则使用 STM32CubeProgrammer (**STM32CubeProg**) 加载适当的二进制固件。

所有可用的 BLE/Zigbee®二进制固件位于：/Projects/STM32WB_Copro_Wireless_Binaries/STM32WB5x.

有关如何更改无线协同处理器二进制固件的详细步骤，请参考

/Projects/STM32WB_Copro_Wireless_Binaries/STM32WB5x/Release_Notes.html。

7.3 支持的 Zigbee®集群

STM32WB 系列上可用的 Zigbee®生态系统支持 Zigbee® 3.0。

Zigbee® 3.0 集群符合 ZCL 7 标准。

该集群包括基本设备特性（BDB）、Zigbee® Green Power 以及下列特定 ZCL 集群：

表 5. 支持的 Zigbee®集群

Nb	集群 ID	集群名称
1	0x0000	基本集群
2	0x0001	电源配置集群
3	0x0003	识别集群
4	0x0004	群组集群
5	0x0005	场景集群
6	0x0006	开/关集群
7	0x0008	级别控制
8	0x000a	时间集群
9	0x0019	空中升级集群
10	0x0020	轮询控制集群
11	0x0021	节能代理
12	0x0102	窗口覆盖集群
13	0x0202	风扇控制集群

Nb	集群 ID	集群名称
14	0x0204	恒温器用户接口集群
15	0x0300	颜色控制集群
16	0x0301	镇流器配置集群
17	0x0400	照度测量集群
18	0x0402	温度测量集群
19	0x0406	占用检测集群
20	0x0502	IAS WD 集群
21	0x0b05	诊断集群
22	0x1000	Touchlink 集群
23	0x0002	设备温度配置集群
24	0x0007	开/关开关配置集群
25	0x0009	报警集群
26	0x000b	RSSI 位置集群
27	0x0015	配置集群
28	0x001a	电源配置文件集群
29	0x0024	最近网关集群
30	0x0101	门锁集群
31	0x0200	泵配置和控制集群
32	0x0201	恒温器集群
33	0x0203	除湿控制集群
34	0x0401	照度水平检测集群
35	0x0403	压力测量集群
36	0x0405	相对湿度测量
37	0x0500	IAS 区域集群
38	0x0501	IAS ACE 集群
39	0x0700	价格集群
40	0x0701	需求响应和负载控制集群
41	0x0702	计量集群
42	0x0703	消息集群
43	0x0704	智能能源隧道（复杂计量）
44	0x0800	密钥建立
45	0x0904	Voice over Zigbee®集群
46	0x0b01	仪表识别集群
47	0x0b04	电气测量集群

- 通过 STM32_WPAN 中间件可以使用所有这些 47 个集群。此中间件对于 BLE 和 Thread®通用。对于特定需求，客户可以在需要时创建“专有”集群。更多详细信息，请参见[3]。
- 与这些集群相对应的 API 位于以下目录中：
\\Middlewares\\ST\\STM32_WPAN\\Zigbee\\stack\\include。
- 默认情况下，所有集群均以单个库的形式提供。尽管如此，也可根据请求访问源代码。

8 STM32WB 系列动态应用设计

8.1 BLE/Zigbee®动态应用框架

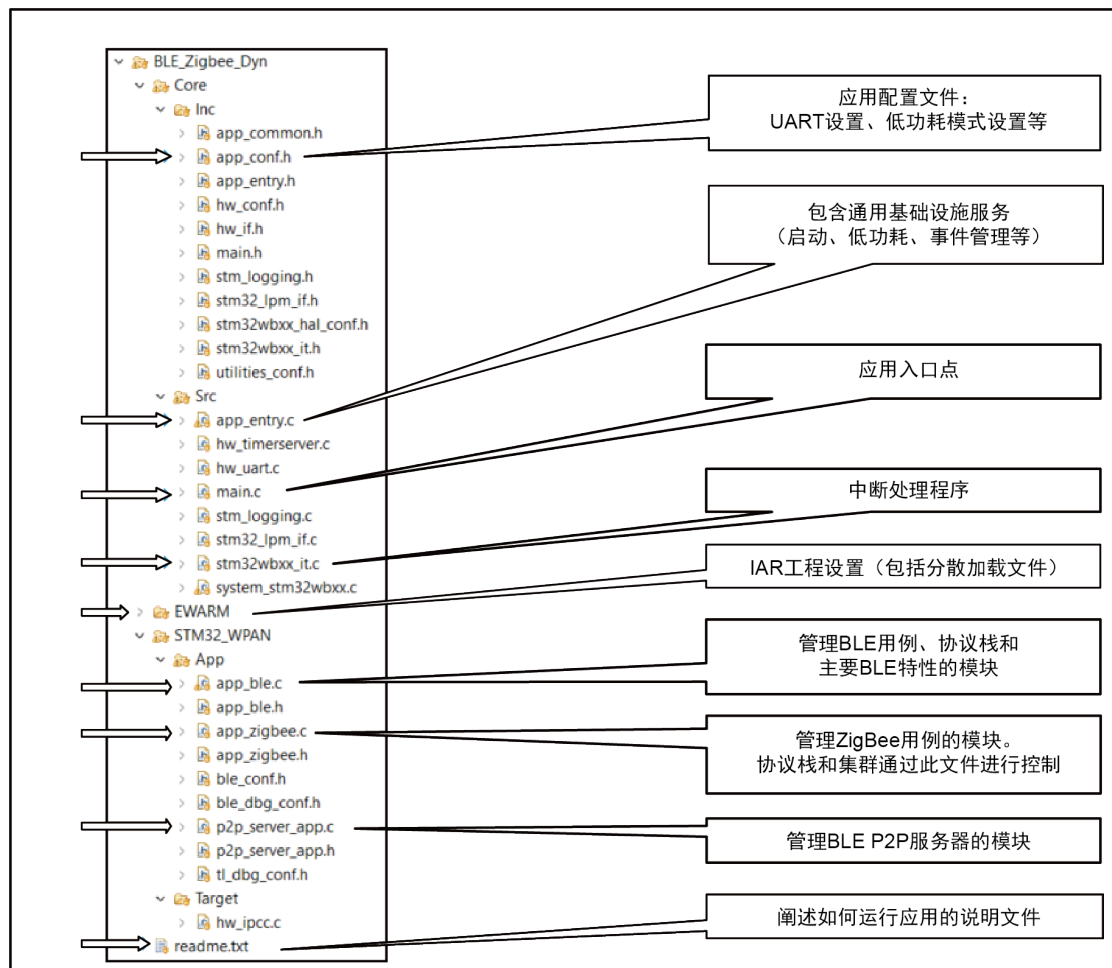
所有工程均采用相同的框架构建。应用的主要功能定义位于：

Projects\Board_X\Applications\BLE_ZigBee\BLE_ZigBee_Dyn\STM32_WPAN\App.

- Zigbee®用例在 app_zigbee.c 文件中定义并实现。
- BLE 用例在 app_ble.c 文件中定义并实现。
- BLE P2P 服务器在 p2p_server_app.c 文件中定义并实现。

应用工程中的所有其他文件主要用于全局基础设施管理（中断管理、IPCC 封装器、系统启动和配置等）

图 11. BLE/Zigbee®动态应用



8.2 Zigbee®应用框架

有关 Zigbee®应用框架和架构的更多详情，请参考[6]。

8.3 BLE 应用架构

有关 BLE 应用架构的更多详情，请参考[1]。

8.4 可用的动态应用

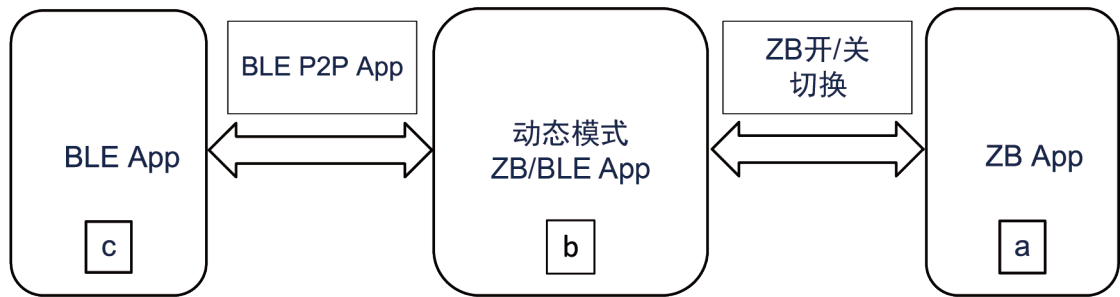
已实现以下四个动态应用：

- BLE/Zigbee 动态应用
- BLE/Zigbee 动态 SED 应用
- BLE/Zigbee 动态 BLE 吞吐量应用
- BLE/Zigbee 动态 NVM 应用

8.4.1 BLE/Zigbee®动态应用

此应用阐述了如何在同一设备上同时连接 BLE 和 Zigbee®，从而在同一设备上同时运行 BLE P2P 应用执行和 Zigbee®切换开/关应用。

图 12. BLE/Zigbee®动态应用图例



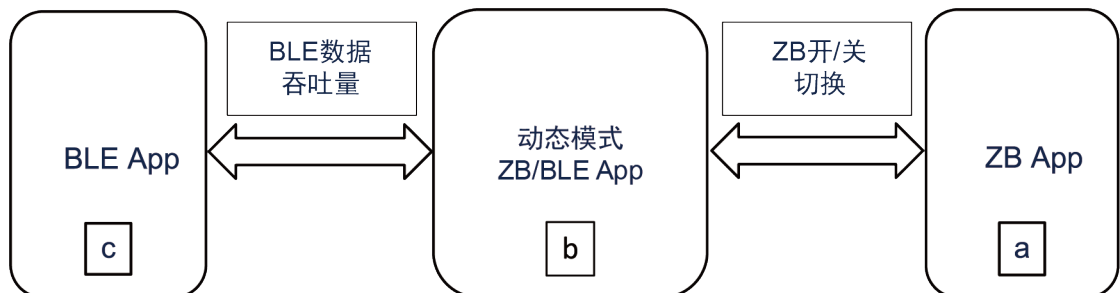
8.4.2 BLE/Zigbee®动态 SED 应用

与上一应用相同，但是经过优化，适合 SED 设备（休眠终端设备）和低功耗模式。

8.4.3 BLE/Zigbee®动态 BLE 吞吐量应用

将 BLE 吞吐量应用移植到动态环境，以便在使用高 BLE 带宽时检查 BLE/Zigbee®共存情况。

图 13. BLE/Zigbee®动态 BLE 吞吐量应用图例



8.4.4 BLE/Zigbee®动态 NVM 应用

此应用阐述了如何在同一设备上同时连接 BLE 和 Zigbee®，从而在同一设备上同时运行 BLE P2P 应用执行和 Zigbee®切换开/关应用。

当 BLE 运行时，利用 Zigbee®上的永久数据保存/恢复 Zigbee®状态。

此应用如图 12 所示。

8.5 BLE/Zigbee®动态应用

8.5.1 固件和软件要求

为了运行 BLE/Zigbee®动态应用，必须安装以下二进制固件和/或软件应用：

表 6. 设备固件规范

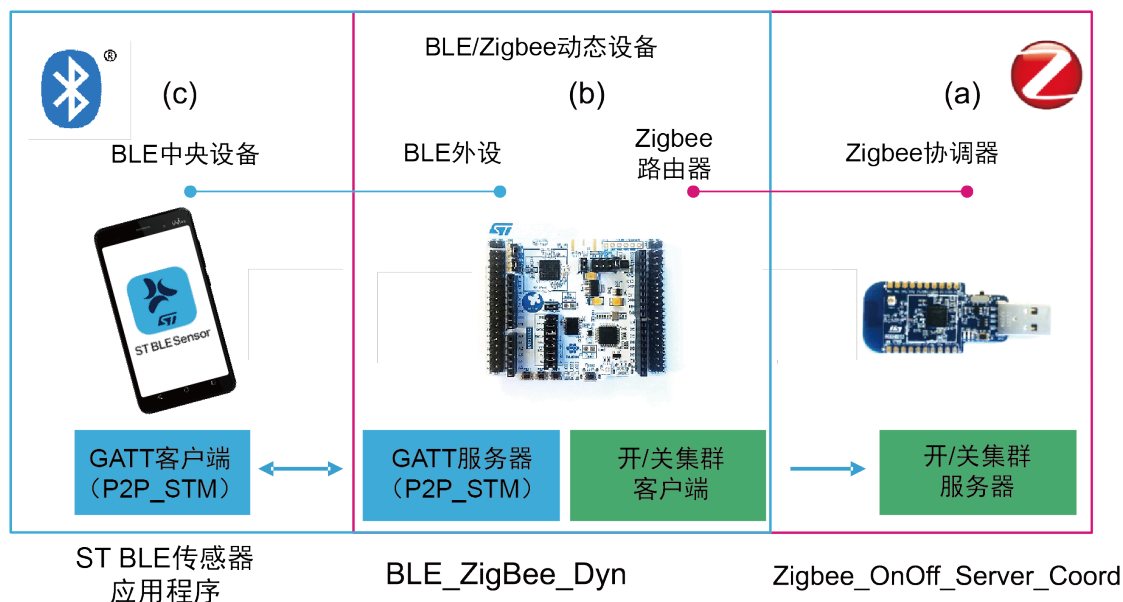
角色	ID	器件	Cortex®-M0 固件/Cortex®-M4 应用
Zigbee®协调器	(a)	Nucleo 板	Cortex®-M0 stm32wb5x_ZigBee_FFD_fw.bin Cortex®-M4 ZigBee_OnOff_Coord.bin
动态设备	(b)	Nucleo 板	Cortex®-M0 stm32wb5x_BLE_ZigBee_FFD_dynamic_fw.bin Cortex®-M4 ble_zigbee_Dyn.bin
BLE 设备	(c)	智能手机 (Android™/iOS™)	“ST BLE Sensor” 手机应用，可在应用商店和 Google Play 获取
		Nucleo 板	Cortex®-M0 stm32wb55xx_ble_full_host_stack_cut2.1.bin Cortex®-M4 BLE_p2pClient.out

8.5.2 动态应用说明

下图所示的 BLE/Zigbee®动态应用的主要组件包括：

- 运行开/关集群服务器的 Zigbee®协调器 (a)。
- 配置为运行开/关集群客户端的 Zigbee®路由器以及运行 P2P 服务器应用的 BLE 外设的 BLE/Zigbee®动态设备 (b)。
- 运行 P2P 客户端应用 7 的 BLE 设备 (c)。

图 14. 动态应用概述



8.5.3

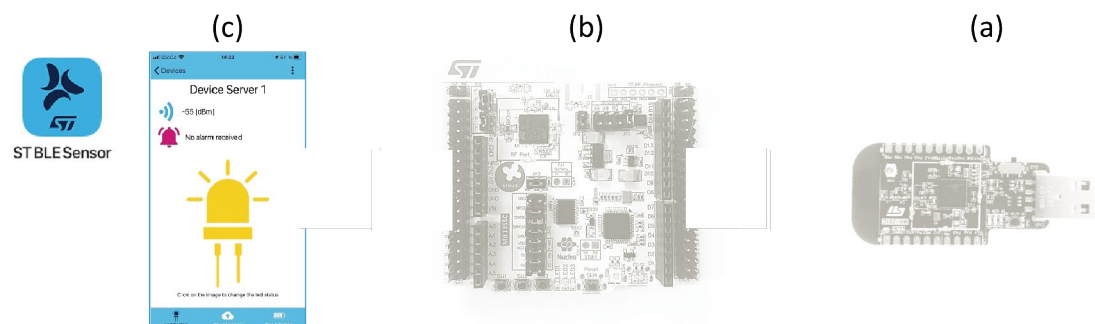
运行动态应用

本演示阐述了如何同时连接 BLE 和 Zigbee 的情况，动态设备 (b) 上的红色 LED 切换反映 BLE 活动，Zigbee®协调器 (a) 上的红色 LED 切换反映 Zigbee®活动。

动态设备 (b) 在 BLE 和 Zigbee®模式均激活时启动。

BLE 示例采用 P2P 组件实现点对点通信。

图 15. 动态应用特性



1. 蓝色 LED 指示网络已形成 (Zigbee®协调器 (a)) 并已加入网络 (动态设备 (b))。
2. 开/关集群客户端 (动态设备 (b)) 每秒将切换指令发送到开/关集群服务器 (Zigbee®协调器 (a)) 一次，然后 Zigbee®协调器 (a) 上的红色 LED 切换。
3. 接着通过“ST BLE Sensor”应用 (c) 将动态设备 (b) 连接到 BLE 并通过此应用发送 BLE 切换指令，同时通过动态设备 (b) 上的红色 LED 状态反映。

动态应用详细特性

1. 动态设备 (b) (P2P 服务器) 启动 BLE 广播 (绿色 LED 切换)，同时开始加入 Zigbee®网络。

2. 节点然后连接现有的 Zigbee®网络（运行 ZigBee_OnOff_Coord.bin 应用的 Zigbee®协调器（a））。当 Zigbee®加入成功时，动态设备（b）蓝色 LED 点亮。
3. 当建立 Zigbee®连接时，Zigbee®路由器（b）每秒将 Zigbee® ON_OFF 切换指令发送到 Zigbee®协调器（a）一次。Zigbee®协调器（a）上的红色 LED 切换。
4. 根据所连接的设备，BLE 连接按照以下步骤启动。在本例中，BLE 设备是智能手机或第三个 Nucleo 板：
 - a. 对于智能手机，选择“P2PZBSRV1”条目，“ST BLE Sensor”（智能手机应用）将扫描并连接到 P2P 服务器。过程如下：
 - i. 连接后，应用开始搜索 P2P 服务和特征。
 - ii. 此阶段会发现 LED 按钮服务、LED 特征和按钮特征（仅意法半导体应用提供此特定服务）。
 - iii. 按下应用上的 LED 按钮将点亮或关闭动态设备（b）的红色 LED。
 - b. 当使用第三个 Nucleo 板时，运行 BLE_p2pClient.out 应用，然后按下 SW1 按钮，启动扫描：
 - o BLE 设备（c）上的蓝色 LED 亮起。
 - o 扫描完成后，BLE 设备（c）连接到动态设备（b）动态节点上的 P2P 服务器，并在按下 SW1 时发送 BLE 切换指令。
 - o 因此，每当按下 BLE 设备（c）上的 SW1 按钮时，动态设备（b）上的红色 LED 就会切换。

当 BLE 断开连接时，动态设备（b）则显示闪烁的绿色 LED，重新启动广播。

8.5.4 动态 Zigbee®路由问题和应对方案

在测试阶段，发现有些情况下，动态设备的 Zigbee®端可能会停滞一段时间（1 - 5 分钟），然后恢复正常执行。本节分析了问题并阐述了应对方案。

Zigbee®路由问题说明

发现有些情况下，动态设备的 Zigbee®端可能会停滞，且 Zigbee®在一段时间内（1 - 5 分钟）停止切换，然后再恢复正常执行。

根据 Zigbee®和 BLE 时间，随时会发生此问题。

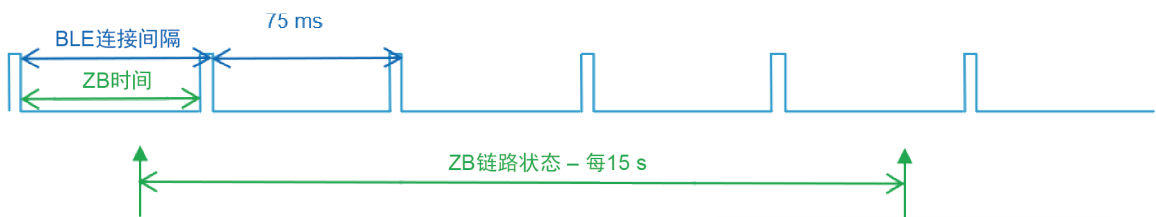
Zigbee®路由问题分析

此问题是由于 Zigbee®和 BLE 之间的时序对齐所致，当在 BLE 时间内收到每 15 s 发送的 Zigbee®链路状态时就会发生此问题。下图展示了在成功和失败用例期间会发生的情况。

成功用例：

这种情况下，在 Zigbee 时间内发生链路状态且一切正常工作。如图 16 所示。

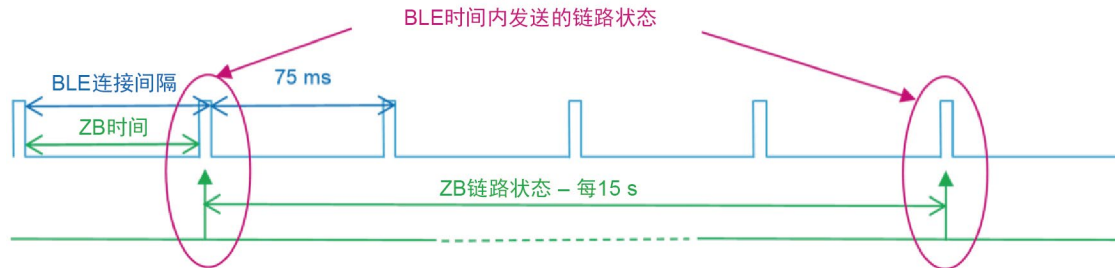
图 16. Zigbee®成功动态路由



- 失败用例：

这种情况下，在 BLE 时间内发生链路状态，但是动态模式下的路由器未检测到。由于“链路状态”消息未重试，所以视为此消息丢失。此应用如图 17 所示。

图 17. Zigbee®动态路由故障



Zigbee®协调器每 15 秒发送一次 Zigbee®链路状态，以通知 Zigbee®链路正确（15 秒是 Zigbee®规范定义的默认时间）。

有些情况下，由于 BLE/Zigbee®动态设备处于 BLE 模式，可能会未发现此通知，并且无法检测到任何 802.15.4 流量。

由于此时间（15 s）是 BLE 连接间隔（75 ms）的整数倍，并且“链路状态”消息未重试，所以视为此消息丢失。链路状态连续丢失 4 次后，Zigbee®链路视为无效，并进入锁定状态。

Zigbee®锁定状态可能保持 3 或 5 分钟，因为 Zigbee®和 BLE 时序保持一致，此时间是两个时钟缓慢漂移所需的时间，最终可使用户能在 Zigbee®时间内看到 Zigbee®链路状态。这样 Zigbee®即可恢复。

当使用需要高带宽的 BLE 应用时，可能会更频繁地发生这种问题。这种情况下，BLE 时间会延长，而 Zigbee®带宽会减小。

当开发动态 BLE/Zigbee®应用时，在选择和调整 BLE 参数（连接间隔和 CEmax）时必须特别注意，以使 Zigbee®和 BLE 应用均具有良好的性能。

Zigbee®路由问题应对方案

实施了动态 BLE/Zigbee®设备请求更改连接间隔时的应对方案，概述如下：

- 动态设备每 15s 监测一次 Zigbee®连接的 AGE。
- 如果 AGE 等于 4（意味着链路状态连续丢失 4 次），则 BLE/Zigbee®设备将请求更新 BLE 连接间隔（CI）。
- 远程 BLE 设备（主机）接受并更新连接间隔。
- 新 CI 会导致时钟缓慢漂移，最终达到在 Zigbee®时间内收到“链路状态”的时间。
- 之后 Zigbee®链路恢复正常执行。

当动态 BLE/Zigbee®设备是 BLE 从机设备时，将由 BLE 主机决定接受还是拒绝此更改。使用 Android™和 iPhone®设备进行的测试表明，两个设备均接受连接更新请求：

- Android™设备通常接受请求的准确连接更新值。
- iPhone®设备通常在调整请求值后接受连接更新。

实施此应对方案后，此问题的负面影响会大幅降低，当收到下一“连接状态”时，Zigbee®链路不再受到阻止，通常在 15 秒内恢复。

8.6 BLE/Zigbee®动态 SED 应用

8.6.1 固件和软件要求

要运行 BLE/Zigbee®动态 SED 应用，必须安装以下二进制固件和/或应用：

表 7. BLE/Zigbee®动态 SED 应用

角色	Id	器件	M0 固件/M4 应用
Zigbee®协调器	(a)	Nucleo 板	Cortex®-M0 stm32wb5x_ZigBee_FFD_fw.bin Cortex®-M4 ZigBee_OnOff_Coord.bin
动态设备	(b)	Nucleo 板	Cortex®-M0 stm32wb5x_BLE_ZigBee_RFD_dynamic_fw.bin Cortex®-M4 ble_zigbee_SED_Dyn.bin
BLE 设备	(c)	智能手机 (Android™/iOS™)	“ST BLE Sensor” 手机应用，可在应用商店和 Google Play 获取
		Nucleo 板	Cortex®-M0 stm32wb5xx_ble_full_host_stack_cut2.1.bin Cortex®-M4 BLE_p2pClient.out

8.6.2 动态 SED 应用说明

BLE/Zigbee®动态 SED 应用的特性基本上与第 8.5.2 节所述的 BLE/Zigbee®动态应用相同。

BLE/Zigbee®动态 SED 应用的主要组件包括：

- 运行开/关集群服务器的 Zigbee®协调器 (a)。
- 配置为运行开/关集群客户端的 Zigbee®路由器以及运行 P2P 服务器应用的 BLE 外设的 BLE/Zigbee®动态设备 (b)。
- 运行 P2P 客户端应用的 BLE 设备 (c)。

固件差异如下所列（仅适于动态设备）：

- Cortex®-M0 固件内置 RFD（精简功能设备）代替 FFD。
- Cortex®-M4 固件内置 SED 配置（休眠终端设备）。

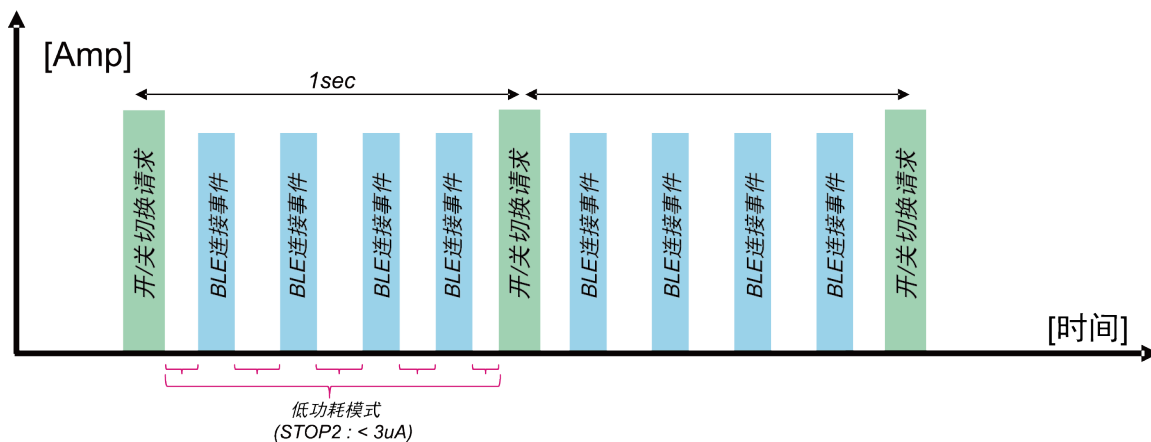
功能性区别如下所列：

- 低功耗模式激活
- LED 和按钮禁用
- 调试模式和调试跟踪禁用。

为了在 SED 端实现最低功耗，编译此应用时默认将标志 CFG_FULL_LOW_POWER 设为 1（位于 app_conf.h 文件中）。在此配置下，LED 和按钮不再可用，且 Cortex®-M4 内核的调试访问也禁用。

在此配置下，通过使用功耗评估板，可以检查既没有 BLE 也没有 Zigbee®活动的时间，并且 SED 能够达到 STOP2 低功耗模式，功耗降低至 3 µA。

图 18. 动态 SED 应用低功耗

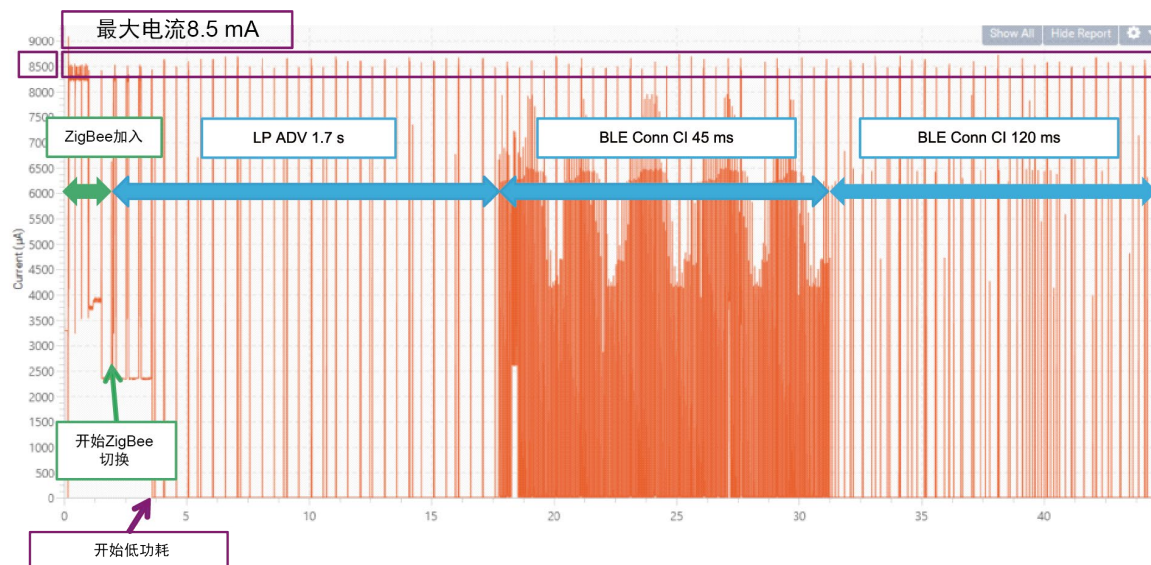


8.6.3 动态 SED 功耗图

总功耗图

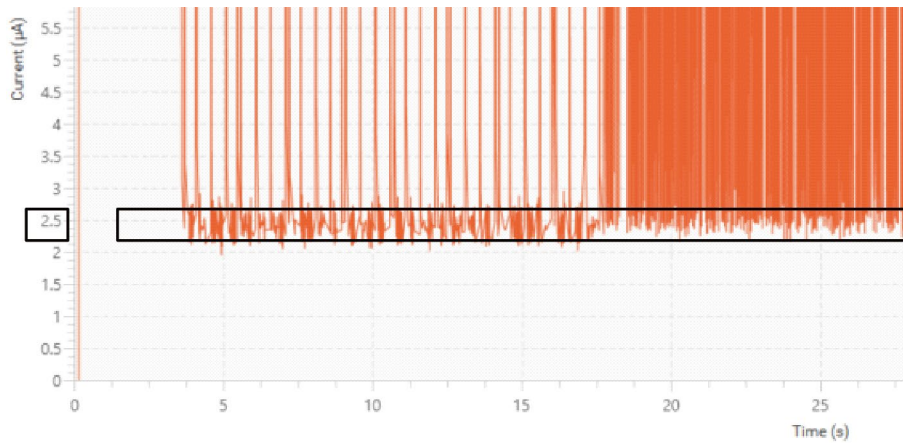
图 19 展示了从启动到 BLE 连接状态的不同阶段期间的总功耗图。

图 19. 动态 SED 总功耗图



1. 启动后，动态设备立即启动“ZB JOIN”过程。
2. 同时，动态设备每 1.7 秒启动一次 BLE 广播。
3. Zigbee®加入完成后，动态设备开始每秒发送一次 ZB 开/关切换指令。
4. 此时，低功耗模式启用（时间约 5 s）。
5. 约 17 秒后，建立 BLE 连接（连接间隔 50 ms）。
6. 9 秒后，BLE 连接更新将新连接间隔设为 120 ms。
7. 如图 19 所示，SMPS 启用时的最大电流为 8.5 mA。
8. 如下图 20 所示，SMPS 启用时的低功耗边缘刚好为 2.5 μ A。

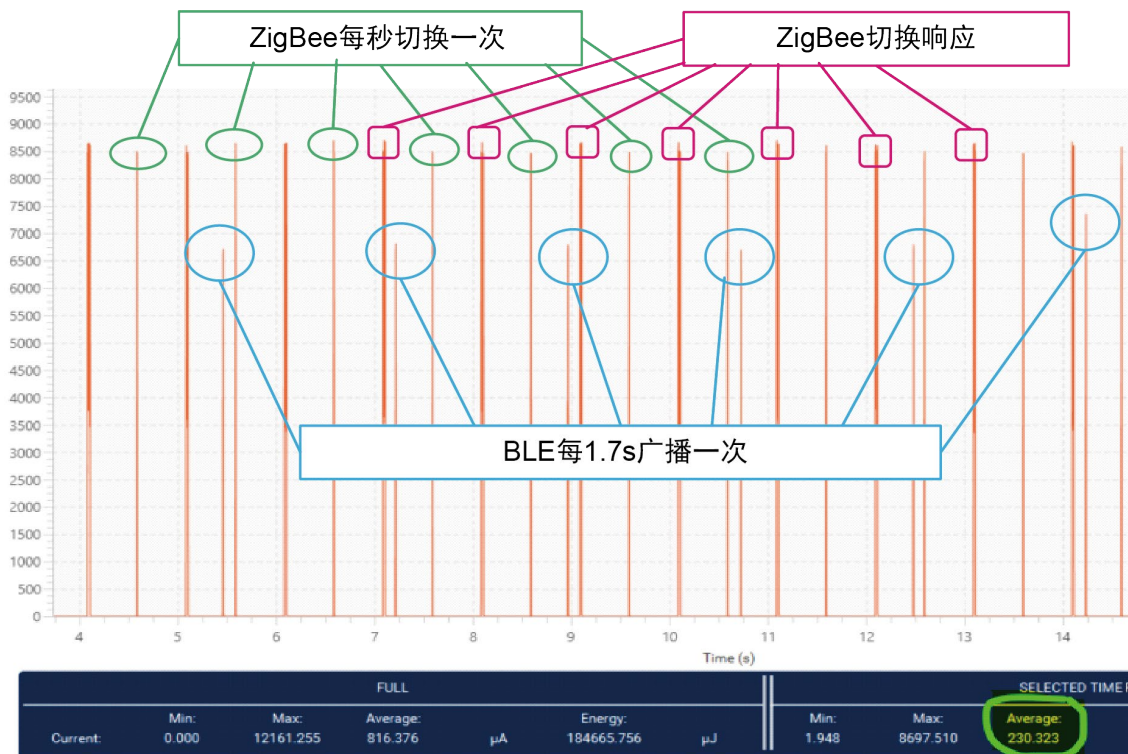
图 20. SMPS 启用时的低功耗边缘电流



BLE 广播功耗图

图 21 展示了 BLE 广播阶段期间的功耗图。无 SMPS 时的平均功耗为 460 μA ，SMPS 启用时的平均功耗为 230 μA 。

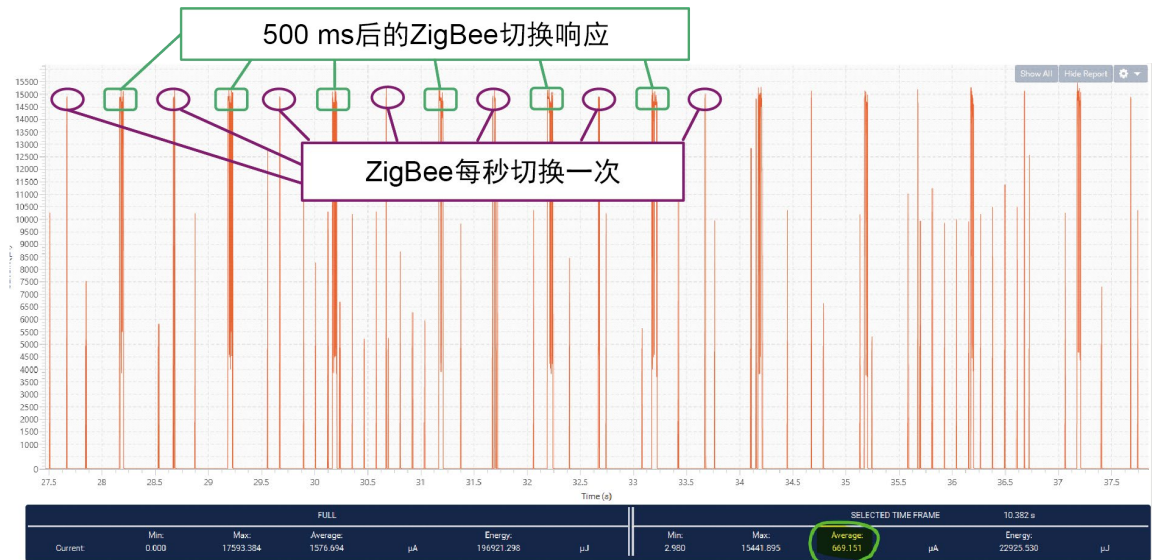
图 21. BLE 广播功耗图



BLE 连接功耗图

图 22 展示了 BLE 连接状态期间的功耗图。

图 22. BLE 连接功耗图



当以 120 ms 连接间隔连接 BLE 时，无 SMPS 时的平均功耗为 660 µA，SMPS 启用时的平均功耗为 280 µA。更详细观察，在 CI 分别设为 120 ms 和 45 ms 的图 23 和图 24 中可以清楚看到 Zigbee® 切换事件和 BLE 连接事件。

图 23. Zigbee® 切换 & BLE 连接状态 (CI = 120)

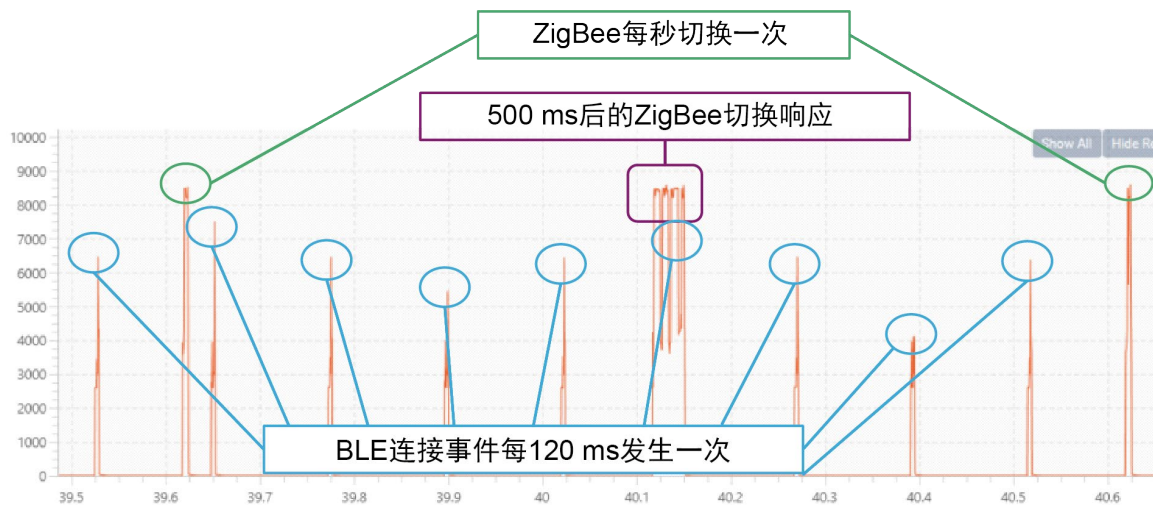
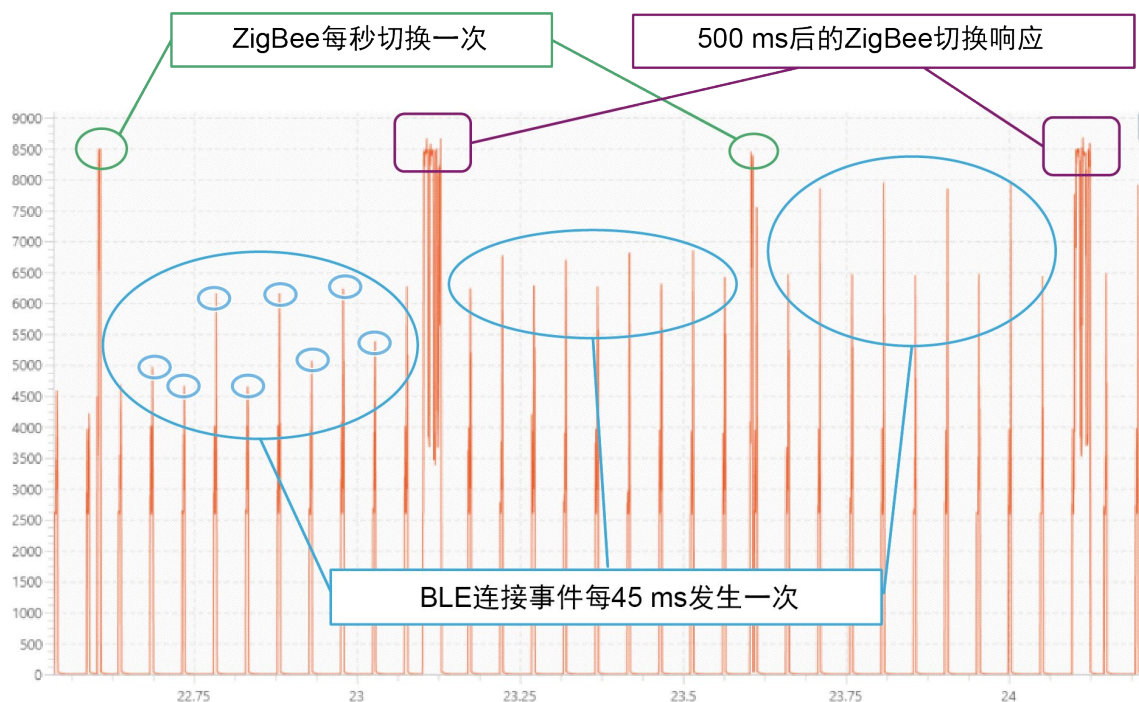


图 24. Zigbee®切换 & BLE 连接状态 (CI = 45)



动态 SED 功耗测量总结

表 8 提供了在三种不同 SMPS 配置下（SMPS 禁用、SMPS 启用 – 1V4、SMPS 启用 – 1V7）在不同 BLE 状态下的功耗总结。

Zigbee®活动始终相同，每秒发送一次 Zigbee®切换指令，并在 500 ms 后收到切换响应。

这表明启用 SMPS 时可以节省大约 50%能源。

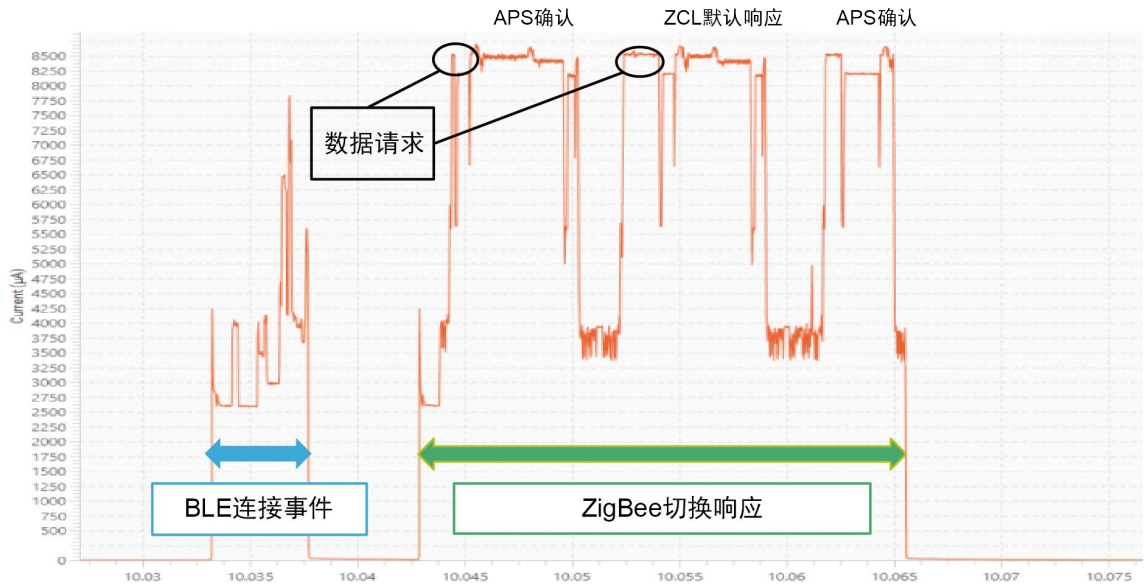
表 8. 动态 SED 功耗测量

状态	SMPS 禁用	SMPS 启用 – 1V4	SMPS 启用 – 1V7
广播 1.7 s	460 μ A	230 μ A	273 μ A
连接 CI=120	660 μ A	280 μ A	321 μ A
连接 CI=45	1.14 mA	579 μ A	703 μ A
连接 CI=15	NA	1.54 mA	1.72 mA
最大功率	15 mA	8.5 mA	10.2 mA
最小功耗	3 μ A	2.5 μ A	2.5 μ A

BLE/Zigbee®功耗图详情

图 25 展示了 BLE 连接事件和 Zigbee®切换响应的详情。

图 25. BLE 连接事件 & Zigbee®切换详情



以下是相应的 Zigbee®跟踪，结合图 26 中的切换跟踪一起展示了 ZB 切换响应的不同步骤：

- 数据请求
- APS 确认
- ZCL 默认响应
- APS 确认

图 26. Zigbee®切换跟踪

No.	Time	Time delta	Source	Source Ext	Destination	Src EP	Protocol	Seq Nb	Length	Frame Pend	Info
123	8.392670	0.484...	0x1746	0x1746	0x0000	17	ZigBee HA	65	62	False	ZCL OnOff: Toggle, Seq: 6
124	8.393246	0.000...					IEEE 802.15.4	65	19	False	Ack
125	8.891974	0.498...	0x1746	0x1746	0x0000		IEEE 802.15.4	66	26	False	Data Request
126	8.892399	0.000...					IEEE 802.15.4	66	19	True	Ack
127	8.895411	0.003...	0x0000	0x0000	0x1746	17	ZigBee	163	59	False	APS: Ack, Dst Endpt: 17, Src Endpt: 17
128	8.895761	0.000...					IEEE 802.15.4	163	19	False	Ack
129	8.903863	0.006...	0x1746	0x1746	0x0000		IEEE 802.15.4	67	26	False	Data Request
130	8.904620	0.000...					IEEE 802.15.4	67	19	True	Ack
131	8.906758	0.002...	0x0000	0x0000	0x1746	17	ZigBee HA	164	64	False	ZCL: Default Response, Seq: 6
132	8.907119	0.000...					IEEE 802.15.4	164	19	False	Ack
133	8.913437	0.006...	0x1746	0x1746	0x0000	17	ZigBee	68	59	False	APS: Ack, Dst Endpt: 17, Src Endpt: 17
134	8.913694	0.000...					IEEE 802.15.4	68	19	False	Ack
135	9.394359	0.480...	0x1746	0x1746	0x0000	17	ZigBee HA	69	62	False	ZCL OnOff: Toggle, Seq: 7
136	9.394742	0.000...					IEEE 802.15.4	69	19	False	Ack
137	9.890334	0.495...	0x1746	0x1746	0x0000		IEEE 802.15.4	70	26	False	Data Request
138	9.890736	0.000...					IEEE 802.15.4	70	19	True	Ack
139	9.894125	0.003...	0x0000	0x0000	0x1746	17	ZigBee	165	59	False	APS: Ack, Dst Endpt: 17, Src Endpt: 17
140	9.894543	0.000...					IEEE 802.15.4	165	19	False	Ack
141	9.897833	0.003...	0x1746	0x1746	0x0000		IEEE 802.15.4	71	26	False	Data Request
142	9.898306	0.000...					IEEE 802.15.4	71	19	True	Ack
143	9.903159	0.004...	0x0000	0x0000	0x1746	17	ZigBee HA	166	64	False	ZCL: Default Response, Seq: 7
144	9.903530	0.000...					IEEE 802.15.4	166	19	False	Ack
145	9.908891	0.005...	0x1746	0x1746	0x0000	17	ZigBee	72	59	False	APS: Ack, Dst Endpt: 17, Src Endpt: 17
146	9.909261	0.000...					IEEE 802.15.4	72	19	False	Ack
147	10.393970	0.484...	0x1746	0x1746	0x0000	17	ZigBee HA	73	62	False	ZCL OnOff: Toggle, Seq: 8
148	10.394328	0.000...					IEEE 802.15.4	73	19	False	Ack

8.7 BLE/Zigbee®动态 BLE 吞吐量应用

将 BLE 吞吐量应用移植到动态环境，以便在使用高 BLE 带宽时检查 BLE/Zigbee®共存情况。

8.7.1 固件和软件要求

为了运行 BLE/Zigbee®动态数据吞吐量应用，必须安装下表中列出的以下二进制固件和/或软件应用：

表 9. 固件和软件要求

角色	Id	器件	Cortex®-M0 固件/Cortex®-M4 应用
Zigbee®协调器	(a)	Nucleo 板	Cortex®-M0 stm32wb5x_ZigBee_FFD_fw.bin Cortex®-M4 ZigBee_OnOff_Coord.bin
动态设备	(b)	Nucleo 板	Cortex®-M0 stm32wb5x_BLE_ZigBee_FFD_dynamic_fw.bin Cortex®-M4 BLE_DataThroughput.out
BLE 设备	(c)	智能手机 (Android™/iOS™)	“STM32WB Toolbox” 手机应用，采用数据吞吐量测试，可在应用商店和 Google Play 获取
		Nucleo 板	Cortex®-M0 stm32wb55xx_ble_full_host_stack_cut2.1.bin Cortex®-M4 BLE_DataThroughput.out

8.7.2 动态 BLE 吞吐量应用概述

两种不同配置的 BLE 吞吐量应用移植到动态环境：

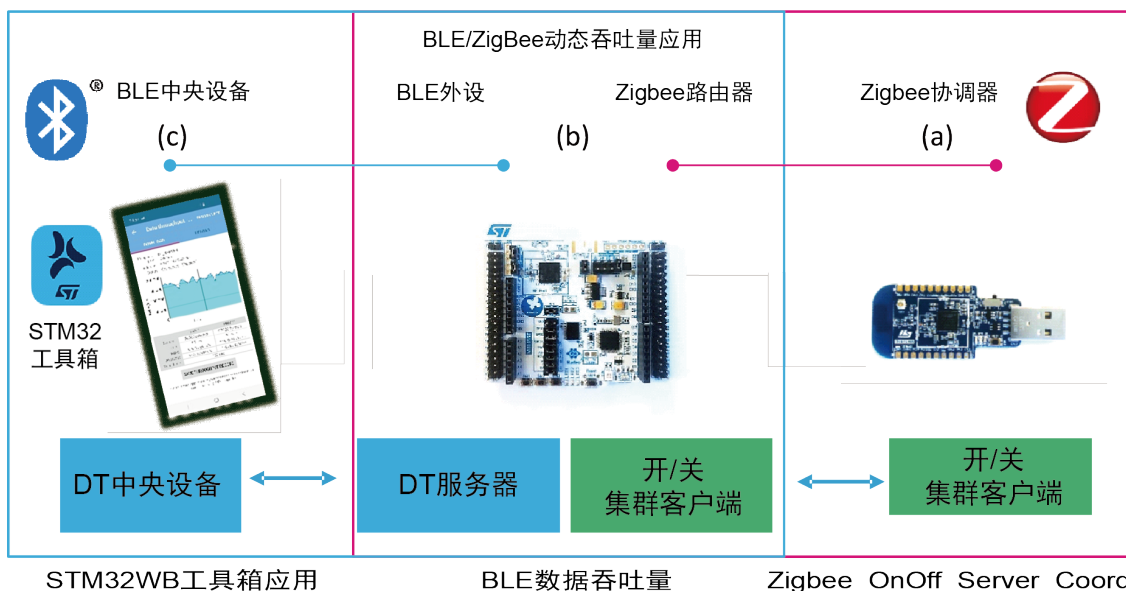
- 配置 1：两个 Nucleo 板结合手机上的 BLE 吞吐量应用
 - Zigbee®协调器 (a) 和动态设备 (b) 位于 Nucleo 板上
 - BLE 设备 (c) 位于智能手机上
- 配置 2：三个 Nucleo 板，其中一个 Nucleo 板上的 BLE 吞吐量应用如图 12 所示
 - Zigbee®协调器 (a) 位于板 (a) 上
 - 动态设备 (b) 位于板 (b) 上
 - BLE 设备 (c) 位于板 (c) 上。

8.7.3 动态 BLE 吞吐量应用说明

动态 BLE 吞吐量应用的主要组件包括：

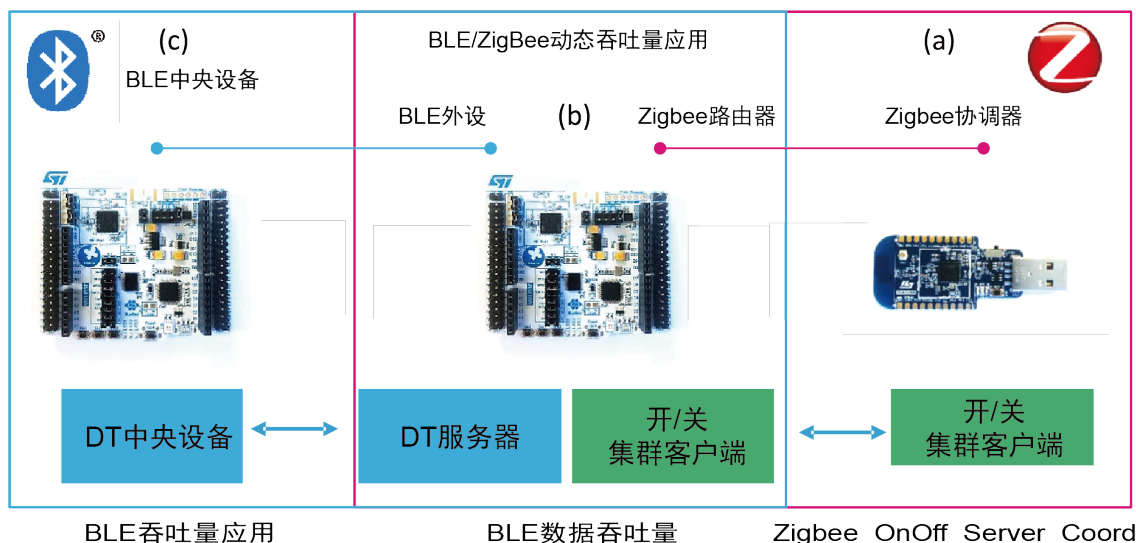
- 配置为运行开/关集群客户端的 Zigbee®路由器以及运行 BLE 吞吐量应用的 BLE 外设的动态设备（b）。
- 运行 BLE 吞吐量应用的 BLE 设备（c）（智能手机或 Nucleo 板）。
- 运行开/关集群的 Zigbee®协调器（a）。

图 27. 采用 BLE 智能手机的动态 BLE 吞吐量应用



BLE 设备（c）可以是图 27 中的智能手机或者图 28 中运行 BLE 吞吐量应用的 Nucleo 板。最后一个配置可以更加灵活地调整 BLE 连接参数。

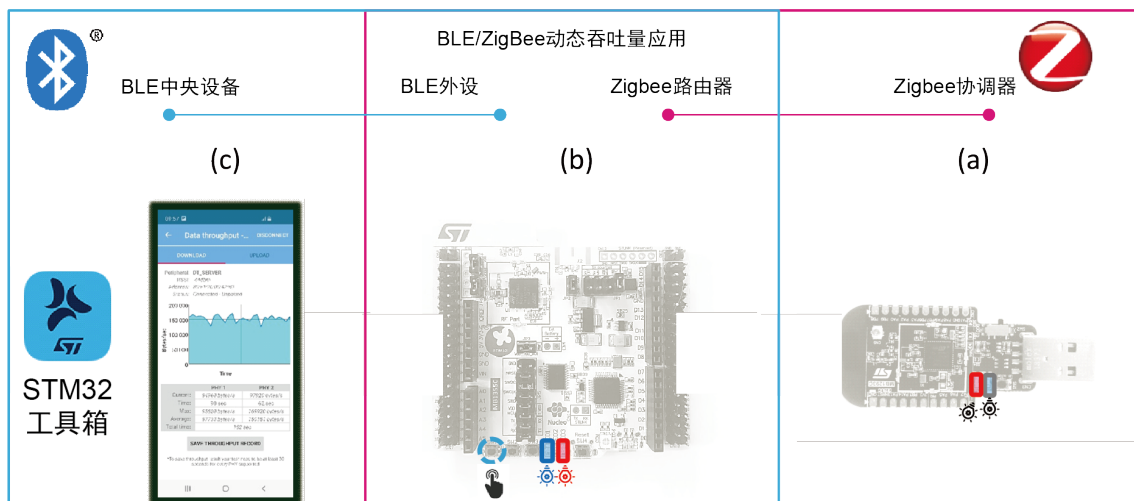
图 28. 采用 Nucleo 板载 BLE 的动态 BLE 吞吐量应用



8.7.4 运行动态 BLE 吞吐量应用

动态 BLE 吞吐量应用可以在运行 Zigbee®路由器时借助 Zigbee®协调器客户端测量 BLE 吞吐量，从而在使用高 BLE 带宽时检查 BLE/Zigbee®共存情况。此应用如图 29 所示。

图 29. 动态 BLE 吞吐量应用特性



动态设备 (b) 在 BLE 和 Zigbee®模式均激活时启动。此应用实现了 BLE 吞吐量应用以及 Zigbee®切换 Zigbee®协调器。

蓝色 LED 指示 Zigbee®协调器 (a) 上已建立网络，并且已加入动态 Zigbee®路由器 (b) 上的网络。

开/关集群 (动态设备 (b)) 每 500 ms 将切换指令发送到开/关集群服务器 (Zigbee®协调器 (a)) 一次，然后 Zigbee®协调器 (a) 上的红色 LED 切换。

动态设备 (b) 广播为 “DT_SERVER” (绿色 LED 闪烁)

如需使用智能手机 (c) 获取 BLE 吞吐量测量值，则通过 STM32WB Toolbox 应用 (c) 进行连接。

如果使用运行 BLE 吞吐量应用 (DT 中央设备) 的 Nucleo 板 (c)，则该板会开始扫描并连接到 “DT_SERVER” 动态设备 (b)。

BLE 吞吐量测量通过使用 1M 或 2M PHY 完成。此设置可通过动态设备 (b) 上的 SW2 进行选择。

BLE 吞吐量应用详细特性

1. 动态设备 (b) (DT 服务器) 启动 BLE 广播 (绿色 LED 切换)，同时开始加入 Zigbee®网络。
2. 节点连接到现有的 Zigbee®网络 (Zigbee®协调器 (a)，运行 ZigBee_OnOff_Coord.bin 应用)。
当 Zigbee®加入成功时，动态设备 (b) 的蓝色 LED 亮起。
3. 当建立 Zigbee®连接时，Zigbee®路由器 (b) 每 500 ms 将 ZB ON_OFF 切换指令发送到 Zigbee®协调器 (a) 一次。
Zigbee®协调器 (a) 上的红色 LED 切换。
4. 根据 BLE 设备是智能手机还是第三块 Nucleo 板，BLE 连接按照以下方式启动：
 - 对于智能手机，使用 STM32WB Toolbox 应用 (选择列表中的 “数据吞吐量”)，通过选择 “DT_SERVER” 条目扫描并连接到 DT 服务器：
 - a. 在下一界面中，选择 “下行链路” 或 “上行链路” 测试。
 - b. 要运行 “下行链路” 测试，连接，并按下动态设备 (b) 上的 SW1 按钮以启动传输。结果显示在智能手机上。物理链路默认设为 1M：
 - 再按一次 SW1 按钮，停止测试。
 - 要将 PHY 改为 2M，按下 SW2 按钮。再次使用 SW1 按钮重新开始和停止 2M 设置下的测试。
 - c. 要运行 “上行链路” 测试，连接，并按下智能手机应用上的 SW1 按钮以启动传输。传输结果显示在智能手机上。物理链路默认设为 1M。
 - 按下应用上的 SW1 按钮，停止测试。
 - 要将 PHY 改为 2M，按下 BLE/Zigbee®设备 (b) 上的 SW2 按钮。再次使用 SW1 按钮重新开始和停止 2M 设置下的测试。
 - 对于 (运行 BLE 吞吐量应用) 的 Nucleo 板 (c)，BLE 会自动开始扫描，然后连接到 “DT_SERVER” 动态设备 (b)。
 - o 要启动 “下行链路” 测试，按下动态设备 (b) 上的 SW1 按钮，结果会显示在 BLE 设备 (c) 串行控制台。物理链路默认设为 1M。
 - o 再按一次 SW1 按钮，停止测试。
 - o 要将 PHY 改为 2M，按下 SW2 按钮。再次使用 SW1 按钮重新开始和停止 2M 设置下的测试。
 - o 按下动态设备 (b) 上的 SW3 按钮将改变具有一些预定义值 (当前 23、43、63、83、105、124、153、209、305、405 ms) 的 BLE 设备 (c) 的连接间隔，然后重新开始吞吐量测试。
5. 在传输过程中以及传输后，Zigbee®协调器 (a) 上的红色 LED 会不停切换。
当 BLE 断开连接时，动态设备 (b) 会重新启动广播 (绿色 LED 闪烁)。

8.7.5 测试结果

使用三星 Galaxy S10e 时的测试配置 1

表 10 提供了以上配置下的结果。

表 10. 三星 Galaxy S10e 测试结果

配置	下载 1M	下载 2M	上传 1M	上传 2M
连接间隔	48.75 ms		11.25 ms	
仅 BLE	90.8 kB/s	161 kB/s	42.5 kB/s	75.6 kB/s
动态模式	87.7 kB/s	150.2 kB/s(1)	41.7 kB/s	75.6 kB/s ⁽¹⁾
ZB 切换	是, ZB 切换工作, 但是非常不稳定且缓慢, 测试失败			

1. 在一段时间后的 BLE 2M 传输过程中, 有时会发现 Zigbee®致命错误, 并且 BLE 保持工作。

下载结果比仅 BLE 配置下降低 3 - 10%。上传结果相似, 但是 Zigbee®活动在测试过程中缓慢且不稳定, 并因 BLE 占用大多数带宽而发生切换错误, 且无法调整 BLE 参数。

测试结束后, Zigbee®活动恢复。

使用 Nucleo 板载 BLE 应用时的测试配置 2

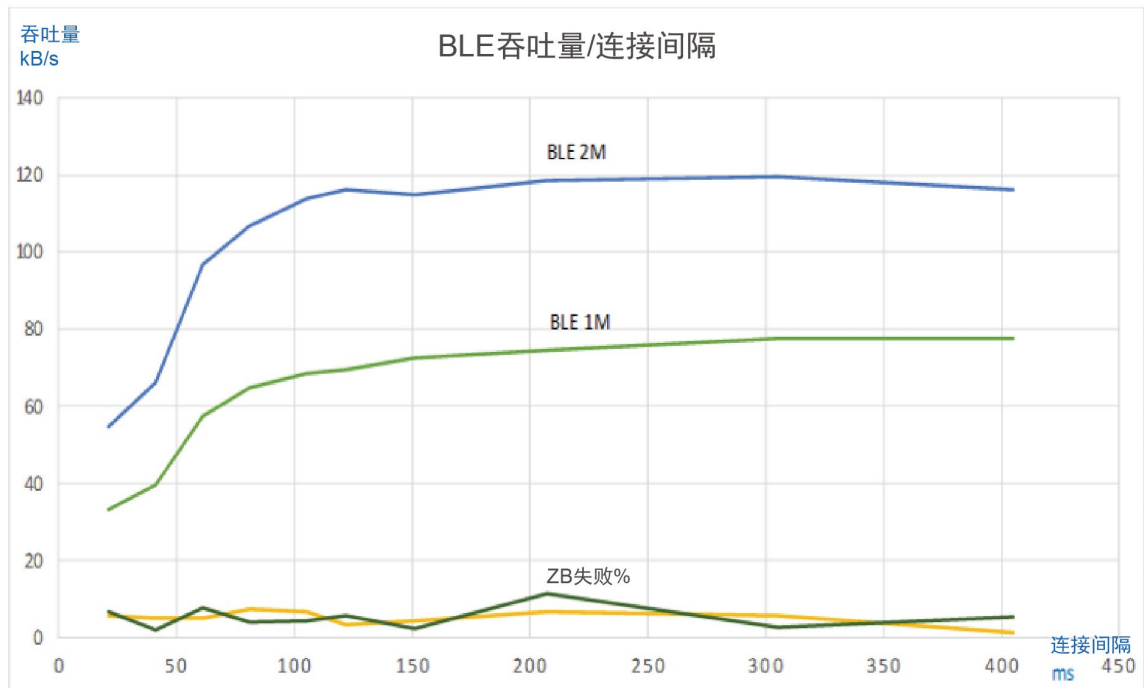
BLE 参数全部可在 Nucleo 板上进行配置。在不同连接配置和 CEmax 参数下运行测试。

表 11 和图 30 中的结果是在 500 ms 的 Zigbee®切换频率下获得的。

表 11. 动态 BLE 吞吐量结果

Cfg	连接间隔	CE _{max}	Dnl 1M kB/s	切换 ZB	失败 ZB	失败 %	Dnl 2M kB/s	切换 ZB	失败 ZB	失败 %
默认	400	625	78.1	不稳定且非常慢	13/159	8.2%	123.8	不稳定且缓慢	15/242	6.2%
2	405	385	77.4	非常稳定	5/410	1.2%	116.3	稳定	19/363	5.4%
3	305	285	77.5	稳定	21/385	5.5%	119.6	稳定	11/448	2.5%
4	207	188	74.4	稳定	27/401	6.7%	118.7	稳定	41/360	11.3%
5	151	132	72.5	稳定	17/396	4.3%	114.9	稳定	11/498	2.2%
6	122	103	69.4	稳定	14/444	3.1%	116.2	稳定	21/374	5.6%
7	105	85	68.6	稳定	26/400	6.5%	113.8	稳定	20/452	4.4%
8	81	62	64.8	稳定	30/411	7.3%	106.9	稳定	17/445	3.8%
9	61	42	57.4	稳定	22/448	4.9%	96.6	稳定	31/405	7.6%
10	41	22	39.4	稳定	20/508	4.9%	66.1	非常稳定	10/525	1.9%
11	21	12	33.3	稳定	23/420	5.5%	54.6	很稳定	25/371	6.7%

图 30. 动态 BLE 吞吐量图



8.7.6 注释

最大 CE 值设为比连接间隔值低 20 ms，以便 Zigbee®在每个连接间隔期间具有至少 20 ms 可用时间。

例如，连接间隔为 105 ms 时，CEmax 值即设为 85 ms，这样即可为 BLE 分配 85 ms，为 Zigbee®分配 20 ms。

由于 Zigbee®协议的特定路由管理问题（参见第 8.5.4 节），必须避免连接间隔值是 15 ms 的直接子倍数，以防 Zigbee®机制死锁。

因此，连接间隔采用以下值：21、41、61、81 等代替 20、40、60、80 等，后者会妨碍稳定的 Zigbee®特性。

基于经验将连接间隔值调整为确保 Zigbee®特性稳定的最佳值，例如，122、151、207 等。

当动态设备请求新的连接间隔时，将由 BLE 主机决定是否接受此值，因此，最终的 CI 值可能并非请求的 CI 值：

- 请求的可能值列表为：23、43、63、83、105、124、153、209、305、405 ms，获取的实际值为：21、41、61、81、105、122、151、207、305、405 ms。
- 这些最终值始终是 1.25 ms 的倍数（例如，81.25 而不是 81）。

Zigbee®失败率始终低于 12%。但是，有些数据包会丢失，Zigbee®切换有时会停滞 2 - 5 秒，然后恢复正常执行，即每 500 ms 切换一次。

8.8 BLE/Zigbee®动态 NVM 应用

8.8.1 固件和软件要求

要运行 BLE/Zigbee®动态 NVM 应用，必须安装以下二进制固件和/或软件应用：

表 12. Zigbee®动态 NVM 应用固件

角色	ID	器件	M0 固件/M4 应用
Zigbee®协调器	(a)	Nucleo 板	Cortex®-M0 stm32wb5x_ZigBee_FFD_fw.bin Cortex®-M4 ZigBee_OnOff_Coord.bin
动态设备	(b)	Nucleo 板	Cortex®-M0 stm32wb5x_BLE_ZigBee_FFD_dynamic_fw.bin M4 ble_zigbee_Dyn_NVM.bin
BLE 设备	(c)	智能手机 (Android™/iOS™)	“ST BLE Sensor” 手机应用，可在应用商店和 Google Play 获取
		Nucleo 板	Cortex®-M0 stm32wb55xx_ble_full_host_stack_cut2.1.bin Cortex®-M4 BLE_p2pClient.out

8.8.2 动态 NVM 应用说明

BLE/Zigbee®动态 NVM 应用的主要组件包括：

- 运行开/关集群服务器的 Zigbee®协调器（a）。
- 配置为运行开/关集群客户端的 Zigbee®路由器以及运行 P2P 服务器应用的 BLE 外设的动态设备（b）。
- 运行 P2P 客户端应用的 BLE 设备（c）。如图 14 所示。

8.8.3 运行动态 NVM 应用

本演示阐述了如何同时连接 BLE 和 Zigbee®的情况，动态设备（b）上的红色 LED 切换反映 BLE 活动，Zigbee®协调器（a）上的红色 LED 切换反映 Zigbee®活动。利用 NVM 永久数据保存/恢复 Zigbee®设备状态。

动态设备（b）在 BLE 和 Zigbee®模式均激活时启动。

BLE 示例采用 P2P 组件实现点到点通信。

此应用如图 15 所示。

Zigbee®协调器上的蓝色 LED 指示网络已形成并与动态设备（b）一起加入网络。

开/关集群客户端（动态设备（b））每 1 秒将切换指令发送到开/关集群服务器（动态设备（a））一次，然后 Zigbee®协调器（a）上的红色 LED 切换。

接着通过“ST BLE Sensor”应用（c）将动态设备（b）连接到 BLE 并通过此应用发送 BLE 切换指令，同时通过动态设备（b）上的红色 LED 状态展示。

8.9 动态 NVM 应用详细特性

- 动态设备（b）（P2P 服务器）启动 BLE 广播（绿色 LED 切换），同时开始加入 Zigbee®网络。
- 借助 Zigbee® NVM 功能，动态设备（b）尝试通过永久数据启动，这样即可在以前的配置下重新启动。此时将有两个选项：
 - 当从 NVM 读取的永久数据有效时：路由器撤销其在网络中的作用。参见第 4 点。
 - 如果未发现永久数据或者数据损坏时，则执行重新启动。参见以下第 3 点。

3. 节点连接到现有的 Zigbee®网络（Zigbee®协调器（a），运行 ZigBee_OnOff_Server_Coord 应用）。
 - a. 当 Zigbee®成功加入网络时，动态设备（b）的蓝色 LED 亮起，表示 Zigbee®已成功加入网络。
4. 当建立 Zigbee®连接时，动态设备（b）每秒将 ZB_ON_OFF 切换指令发送到 Zigbee®协调器（a）一次：
 - a. Zigbee®协调器（a）上的红色 LED 切换。
5. BLE 连接启动取决于使用的 BLE 设备：智能手机还是第三块 Nucleo 板：
 - a. 使用智能手机时，选择“P2PZBSRV1”条目，“ST BLE Sensor”应用（智能手机应用）将扫描并连接到 P2P 服务器。
 - i. 连接后，应用开始搜索 P2P 服务和特征。
 - ii. 此阶段会发现 LED 按钮服务、LED 特征和按钮特征（此服务为意法半导体专有）。
 - iii. 按下应用上的 LED 按钮会点亮或关闭动态设备（b）上的红色 LED。
 - b. 使用 Nucleo 板（运行 BLE_p2pClient 应用（c））时，按下 SW1 按钮将启动扫描过程：
 - i. Bluetooth®低功耗设备（c）上的蓝色 LED 亮起。
 - ii. 扫描完成后，Bluetooth®低功耗设备（c）连接到动态设备（b）动态节点上的 P2P 服务器，并在按下 SW1 时发送 BLE 切换指令。
 - iii. 因此，每当按下 BLE 设备（c）上的 SW1 按钮时，动态设备（b）上的红色 LED 就会切换。
6. 当需要时，将自动更新 NVM 中的 Zigbee®永久数据。
7. 当用户重启动态设备（b）时，将读回永久数据并恢复协议栈配置。
8. 按钮 SW2 用于删除 Zigbee® NVM（下次启动时重新开始）。
当 BLE 断开连接时，动态设备（b）会重新启动广播且绿色 LED 闪烁。

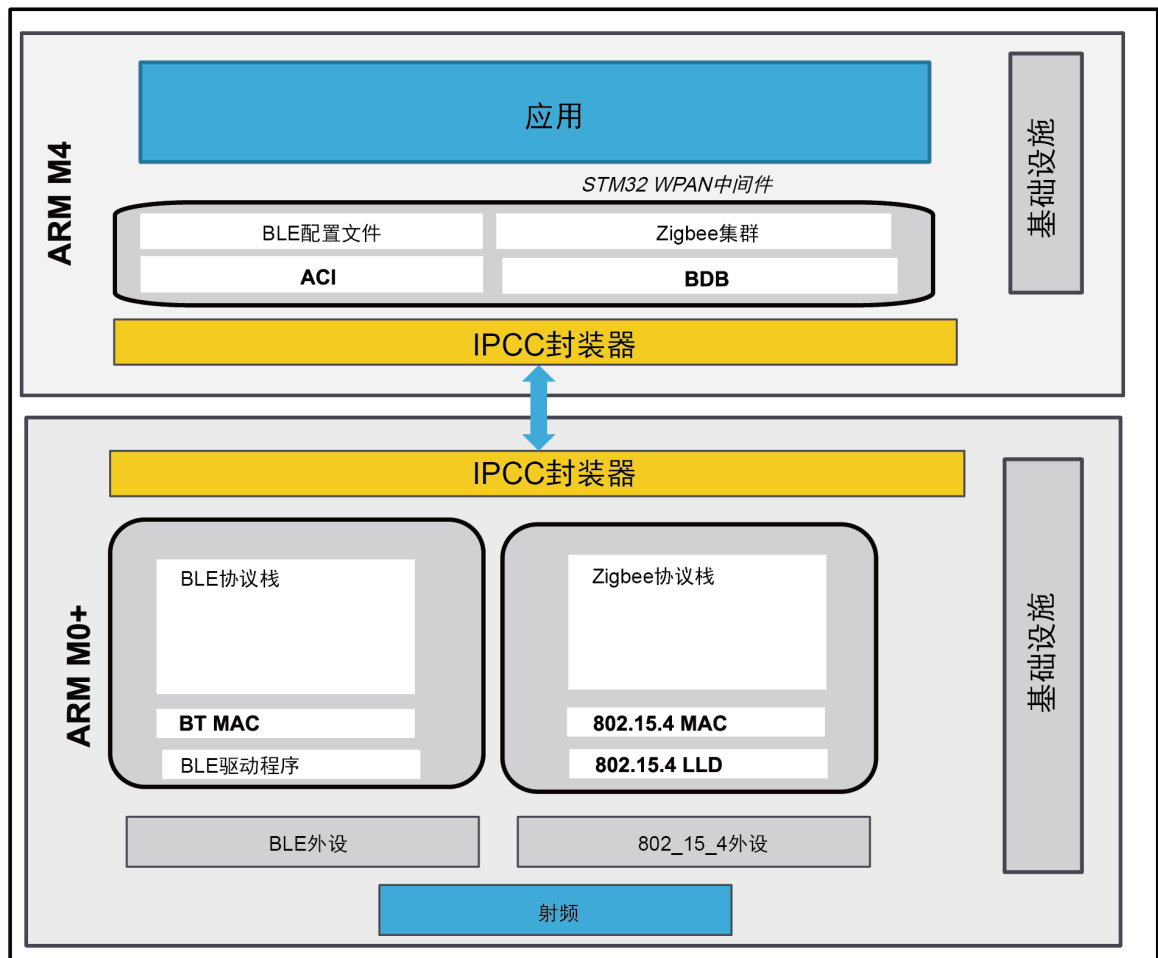
9 BLE/Zigbee®静态并发模式

9.1 静态模式概述

STM32WB 固件包中提供了静态并发模式（BLE/Zigbee®）示例。此应用位于：Projects\P-NUCLEO-WB55.Nucleo\Applications\BLE_ZigBee 目录。

静态模式是动态模式的一个子集，在此模式下，可以根据用户请求从 BLE 切换至 Zigbee®，反之亦然。此设备通过 BLE 连接到运行“ST BLE Sensor”应用的智能手机，一旦 BLE 活动停止，就会加入 Zigbee®网络。接着，当 Zigbee®应用完全停止后，即可再次连接到 BLE。此应用如下图所示。

图 31. STM32WB 系列上的静态并发模式



9.2 支持的静态固件

静态模式

静态模式是动态模式的一个子集，在此模式下，可以根据用户请求从 BLE 切换至 Zigbee®，反之亦然。当 BLE 协议运行时，Zigbee®协议栈则不运行。当 BLE 停止时，系统会切回到 Zigbee®。这种情况下，将完全重新初始化 Zigbee®协议栈。

静态模式固件包括 BLE 和 Zigbee®协议栈：

- BLE 协议栈已获得 Bluetooth®低功耗 5.0 认证。
- **STM32WB** 系列器件支持两种 Zigbee®协议栈：FFD（全功能设备）和 RFD（精简功能设备）。这些协议栈均已获得 Zigbee® PRO 2017（第 22 版）认证。

表 13. 静态协议栈

支持的无线协议栈	相关的固件
Zigbee® FFD + Bluetooth®低功耗 5.0	stm32wb5x_BLE_ZigBee_FFD_static_fw.bin
Zigbee® RFD + Bluetooth®低功耗 5.0	stm32wb5x_BLE_ZigBee_RFD_static_fw.bin

- FFD 可以在网络中扮演以下角色：
 - 路由器
 - 协调器
 - 终端设备。
- RFD 仅支持终端设备角色。RFD 比 FFD 的占用空间小。当构建作为“休眠终端设备”的应用，以实现最佳低功耗时，必须使用 Zigbee® RFD 协议栈构建此应用。

这些二进制固件用于静态并发模式应用。此类应用的示例参见：

Projects\P-NUCLEO-WB55.Nucleo\Applications\BLE_ZigBee 目录。

重要：

在 **STM32WB** 系列上运行任何 BLE/Zigbee®应用前，应确保将正确固件下载到 Cortex®-M0。如果未下载正确固件，则使用 STM32CubeProgrammer 加载适当的二进制固件。

所有可用的 BLE/Zigbee®二进制固件位于：

/Projects/STM32WB_Copro_Wireless_Binaries/STM32WB5x

请参见：

/Projects/STM32WB_Copro_Wireless_Binaries/STM32WB5x/Release_Notes.html，了解如何更改无线协同处理器二进制固件的详细步骤

9.3 静态模式与动态模式

在本节中，将通过以下几点讨论静态模式与动态模式之间的差异：

- 动态模式提供比静态模式更多的功能。
- 静态模式提供更优化的存储器使用和更高效的存储器管理。例如，如果 Zigbee®停止，当我们切回到 BLE 时，将完全释放 Cortex®-M0 存储器（但会分配到 Cortex®-M4 端）。但这并不适用于 Cortex®-M4 存储器。
- 与动态模式相比，静态模式可提供 7 kB 增益。
- 与动态模式相比，静态模式下的吞吐量提高约 5 - 10%。
- 由于每次仅使用一次通信协议，即可减少潜在冲突，因而能够提高总体性能稳健性。
- 实施静态模式可能是特定客户要求。产品可能仅适于一种协议，可能根本不需要其他协议。
- 在以最大速率发送 Zigbee®切换指令的压力测试中，静态模式性能略优于动态模式性能。由于将更多带宽分配到 802.15.4 射频，因此性能提高约 5%。但是对于标准“使用”案例，这可能没有影响。

系统设计在很大程度上取决于产品规范和所需模式。

版本历史

表 14. 文档版本历史

日期	版本	变更
2021 年 6 月 22 日	1	初始版本。

目录

1	概述	2
1.1	术语表	2
1.2	参考文档	3
2	动态模式简介	4
2.1	动态模式用例	5
3	RTSM 架构	6
3.1	RTSM 框图	6
3.2	RTSM 描述	6
3.2.1	RTSM 功能	6
3.2.2	RTSM 规划	7
3.2.3	Cmd_pipe 模块	7
3.2.4	动态模式下的 BLE & Zigbee®协议栈特性	8
4	动态 NVM 功能	10
4.1	动态 NVM 概述	10
4.2	BLE/Zigbee®动态模式 Flash 操作	10
5	Zigbee®架构	11
5.1	Zigbee®概述	11
5.2	Zigbee®协议栈层	11
6	BLE 架构	12
6.1	BLE 概述	12
6.2	BLE 协议栈层	12
7	STM32WB 系列上的 BLE/Zigbee®动态模式	13
7.1	架构概述	13
7.2	支持的动态固件	14
7.3	支持的 Zigbee®集群	14
8	STM32WB 系列动态应用设计	17
8.1	BLE/Zigbee®动态应用框架	17
8.2	Zigbee®应用框架	17
8.3	BLE 应用架构	18
8.4	可用的动态应用	18
8.4.1	BLE/Zigbee®动态应用	18
8.4.2	BLE/Zigbee®动态 SED 应用	18
8.4.3	BLE/Zigbee®动态 BLE 吞吐量应用	18
8.4.4	BLE/Zigbee®动态 NVM 应用	19
8.5	BLE/Zigbee®动态应用	19
8.5.1	固件和软件要求	19
8.5.2	动态应用说明	19
8.5.3	运行动态应用	20
8.5.4	动态 Zigbee®路由问题和应对方案	21
8.6	BLE/Zigbee®动态 SED 应用	23
8.6.1	固件和软件要求	23

8.6.2	动态 SED 应用说明	23
8.6.3	动态 SED 功耗图	24
8.7	BLE/Zigbee®动态 BLE 吞吐量应用	29
8.7.1	固件和软件要求	29
8.7.2	动态 BLE 吞吐量应用概述	29
8.7.3	动态 BLE 吞吐量应用说明	30
8.7.4	运行动态 BLE 吞吐量应用	31
8.7.5	测试结果	33
8.7.6	注释	34
8.8	BLE/Zigbee®动态 NVM 应用	35
8.8.1	固件和软件要求	35
8.8.2	动态 NVM 应用说明	35
8.8.3	运行动态 NVM 应用	35
8.9	动态 NVM 应用详细特性	35
9	BLE/Zigbee®静态并发模式	37
9.1	静态模式概述	37
9.2	支持的静态固件	38
9.3	静态模式与动态模式	39
版本历史		40

图片目录

图 1.	射频外设.....	4
图 2.	动态模式用例.....	5
图 3.	RTSM 框图.....	6
图 4.	Cmd_pipe 模块.....	7
图 5.	BLE 连接事件和连接间隔.....	8
图 6.	Zigbee® Rx/Tx 特性.....	9
图 7.	Flash 操作过程图例.....	10
图 8.	Zigbee®协议栈说明.....	11
图 9.	BLE 协议栈说明.....	12
图 10.	BLE/Zigbee®动态模式架构.....	13
图 11.	BLE/Zigbee®动态应用.....	17
图 12.	BLE/Zigbee®动态应用图例.....	18
图 13.	BLE/Zigbee®动态 BLE 吞吐量应用图例.....	18
图 14.	动态应用概述.....	20
图 15.	动态应用特性.....	20
图 16.	Zigbee®成功动态路由.....	21
图 17.	Zigbee®动态路由故障.....	22
图 18.	动态 SED 应用低功耗.....	24
图 19.	动态 SED 总功耗图.....	24
图 20.	SMPS 启用时的低功耗边缘电流.....	25
图 21.	BLE 广播功耗图.....	25
图 22.	BLE 连接功耗图.....	26
图 23.	Zigbee®切换 & BLE 连接状态 (CI = 120).....	26
图 24.	Zigbee®切换 & BLE 连接状态 (CI = 45).....	27
图 25.	BLE 连接事件 & Zigbee®切换详情.....	28
图 26.	Zigbee®切换跟踪.....	28
图 27.	采用 BLE 智能手机的动态 BLE 吞吐量应用.....	30
图 28.	采用 Nucleo 板载 BLE 的动态 BLE 吞吐量应用.....	31
图 29.	动态 BLE 吞吐量应用特性.....	31
图 30.	动态 BLE 吞吐量图.....	34
图 31.	STM32WB 系列上的静态并发模式.....	37

表格索引

表 1.	术语表.....	2
表 2.	参考文档.....	3
表 3.	Zigbee®运行时指令	8
表 4.	协议栈固件关联	14
表 5.	支持的 Zigbee®集群.....	14
表 6.	设备固件规范.....	19
表 7.	BLE/Zigbee®动态 SED 应用	23
表 8.	动态 SED 功耗测量.....	27
表 9.	固件和软件要求	29
表 10.	三星 Galaxy S10e 测试结果	33
表 11.	动态 BLE 吞吐量结果.....	33
表 12.	Zigbee®动态 NVM 应用固件	35
表 13.	静态协议栈	38
表 14.	文档版本历史.....	40

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 及 ST 标识是意法半导体公司的商标。若需意法半导体商标的更多信息，请参考 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。

© 2023 STMicroelectronics - 保留所有权利