

## 引言

本文档介绍了通过低功耗蓝牙连接在 STM32WB 设备上 OTA 固件更新的过程。它讲解如何使用 STM32Cube 固件包中提供的 OTA 应用程序，实现用户应用程序、无线固件和固件升级服务的更新。

# 目录

<b>1</b>	<b>术语表 .....</b>	<b>6</b>
<b>2</b>	<b>参考文档.....</b>	<b>6</b>
<b>3</b>	<b>STM32WB 设备上的 OTA 应用程序.....</b>	<b>7</b>
3.1	OTA 固件更新 .....	7
3.1.1	原理 .....	7
3.1.2	OTA 应用程序 .....	9
3.1.3	存储器架构 .....	9
3.1.4	用户应用程序更新 .....	12
3.1.5	无线固件更新 .....	12
3.2	用于固件更新的低功耗蓝牙服务和特性 .....	13
3.2.1	BLE 应用程序 – 重启请求特性 .....	13
3.2.2	OTA 应用 - 服务和特性 .....	15
3.2.3	广播包 .....	15
3.3	流程说明 .....	16
3.4	无线和 FUS 更新流程.....	17
3.5	更新程序客户端 .....	18
3.5.1	ST BLE Sensor .....	18
3.5.2	STM32CubeMonitor-RF .....	19
<b>4</b>	<b>OTA 分步过程 .....</b>	<b>21</b>
4.1	项目设置 .....	22
4.1.1	用户应用程序 .....	22
4.1.2	OTA 应用程序 .....	24
4.2	使用 ST BLE Sensor 更新固件 .....	25
4.2.1	设备连接 .....	25
4.2.2	用户应用程序更新 .....	28
4.2.3	无线协议栈或 FUS 的更新.....	31
4.3	使用 STM32CubeMonitor-RF 进行固件更新.....	31
4.3.1	透传模式 .....	31
4.3.2	用户应用程序更新 .....	32
<b>5</b>	<b>结论.....</b>	<b>34</b>

6      版本历史.....35

# 表格索引

表 1.	重启请求特性 .....	14
表 2.	OTA 服务和特性声明 .....	15
表 3.	AD 结构 - 主 .....	16
表 4.	AD 结构 - 制造商特定字段 .....	16
表 5.	AD 结构 - 组 B 功能 .....	16
表 6.	示例应用程序的服务和特性.....	22
表 7.	示例（设备蓝牙 MAC 地址 = 80:E1:25:00:50:D6） .....	25
表 8.	文档版本历史 .....	35
表 9.	中文文档版本历史 .....	35



## 图片目录

图 1.	STM32WB 双核 FW 架构.....	7
图 2.	通过低功耗蓝牙连接更新 STM32WB 固件.....	8
图 3.	OTA 流程.....	9
图 4.	应用程序简化的存储器映射.....	10
图 5.	低功耗蓝牙应用程序和无线固件架构.....	11
图 6.	用户应用程序更新.....	12
图 7.	无线固件更新.....	13
图 8.	低功耗蓝牙用户应用程序中添加的 OTA 重启特性.....	14
图 9.	OTA 数据流.....	17
图 10.	ST BLE Sensor 移动应用.....	18
图 11.	STM32CubeMonitor-RF.....	19
图 12.	STM32CubeMonitor RF 使用处于透传模式的 USB 蓝牙模块.....	20
图 13.	相关参考示例.....	21
图 14.	用户应用程序更新 - 用例.....	22
图 15.	点对点服务器设备检测.....	26
图 16.	点对点应用程序/LED 开关.....	27
图 17.	ST BLE Sensor 固件更新面板和检测到的 OTA 服务.....	28
图 18.	新应用程序下载.....	29
图 19.	更新后的心率配置文件.....	30
图 20.	无线协议栈或 FUS 的更新.....	31
图 21.	在 USB DFU 模式下使用 STM32CubeProgrammer 进行 USB 蓝牙模块编程.....	32
图 22.	用户应用程序更新.....	33
图 23.	无线或 FUS 更新.....	33

## 1 术语表

CPU1	Cortex® M4（执行用户应用程序）
CPU2	Cortex® M0+（执行 FUS 和无线固件）
IDE	集成开发环境
IPCC	处理器间通信控制器
FUS	固件更新服务
OTA	OTA 固件更新
SBRSA	选项字节 - 安全备份 RAM 起始地址
SBRV	选项字节 - 安全启动复位向量
SFSA	选项字节 - 安全 Flash 存储器起始地址
SIG	Bluetooth®技术联盟（SIG）
SNBRSA	选项字节 - 安全非备份 RAM 起始地址

## 2 参考文档

- [1] RM0434: 基于多协议无线 32 位 MCU Arm®的 Cortex®-M4 使用 FPU、低功耗蓝牙和 802.15.4 无线电解决方案
- [2] AN5185: STM32WB ST 固件更新服务
- [3] UM2288: 用于无线性能测量的 STM32CubeMonitor-RF 软件工具

所有文档均可从 [www.st.com](http://www.st.com) 获得。

## 3 STM32WB 设备上的 OTA 应用程序

### 3.1 OTA 固件更新

#### 3.1.1 原理

必须在设备生命周期内更新固件，以确保最佳性能、用新功能或更正的补丁更新应用程序并保持最高的安全级别。

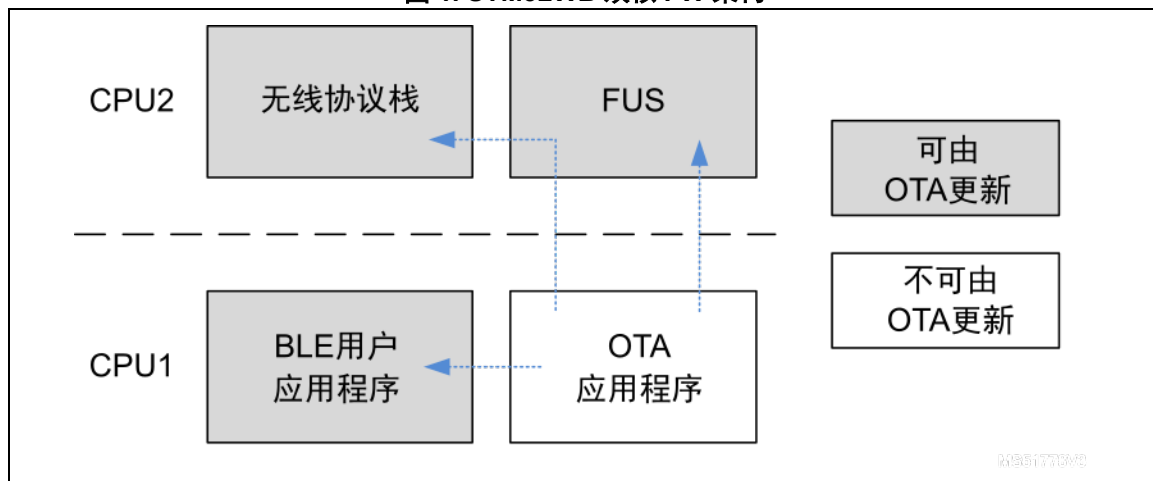
##### 用户应用程序、无线固件和 FUS 更新

STM32WB 系列微控制器基于双 Arm<sup>®(a)</sup>内核。

用户应用程序运行在 CPU1（Cortex<sup>®</sup> M4）上，而无线固件和 FUS 运行在 CPU2（Cortex<sup>®</sup> M0+）上。这三个应用程序可以独立更新。

由于无线和 FUS 以加密格式提供，其更新过程依赖于 STM32WB 安全固件以实现更新内容的解密以及 Flash 存储区内受保护区域的分配。

图 1. STM32WB 双核 FW 架构



#### 无线

可以通过物理链路（如 USB、UART 或 I2C）来完成固件的更新，但此过程需要对设备上特定端口进行物理访问。得益于 OTA，可以使用无线连接来远程更新固件。在本应用笔记中，无线连接使用低功耗蓝牙协议。

arm

a. Arm 是 Arm Limited（或其子公司）在美国和/或其他地区的注册商标。

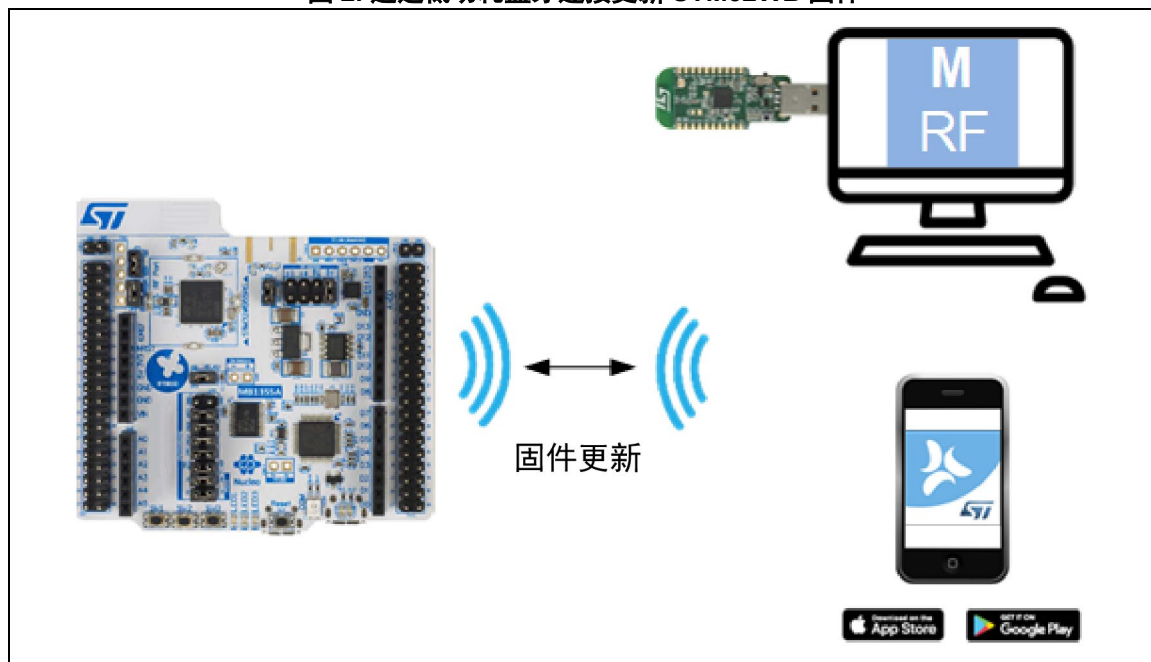
## 更新客户端

更新过程遵循客户端/服务端架构，由外部客户端发起。需更新设备作为服务端，通过响应客户端的请求实现新版本固件的升级。

本文介绍了意法半导体提供的基于低功耗蓝牙协议的两个 OTA 客户端的示例（见图 2）：

- ST BLE Sensor：用于 Android™ 或 iOS 设备的移动应用程序
- STMCubeMonitor-RF：在 HCI 透传模式下配置的与低功耗蓝牙 USB 蓝牙模块通信的 PC 工具

图 2. 通过低功耗蓝牙连接更新 STM32WB 固件





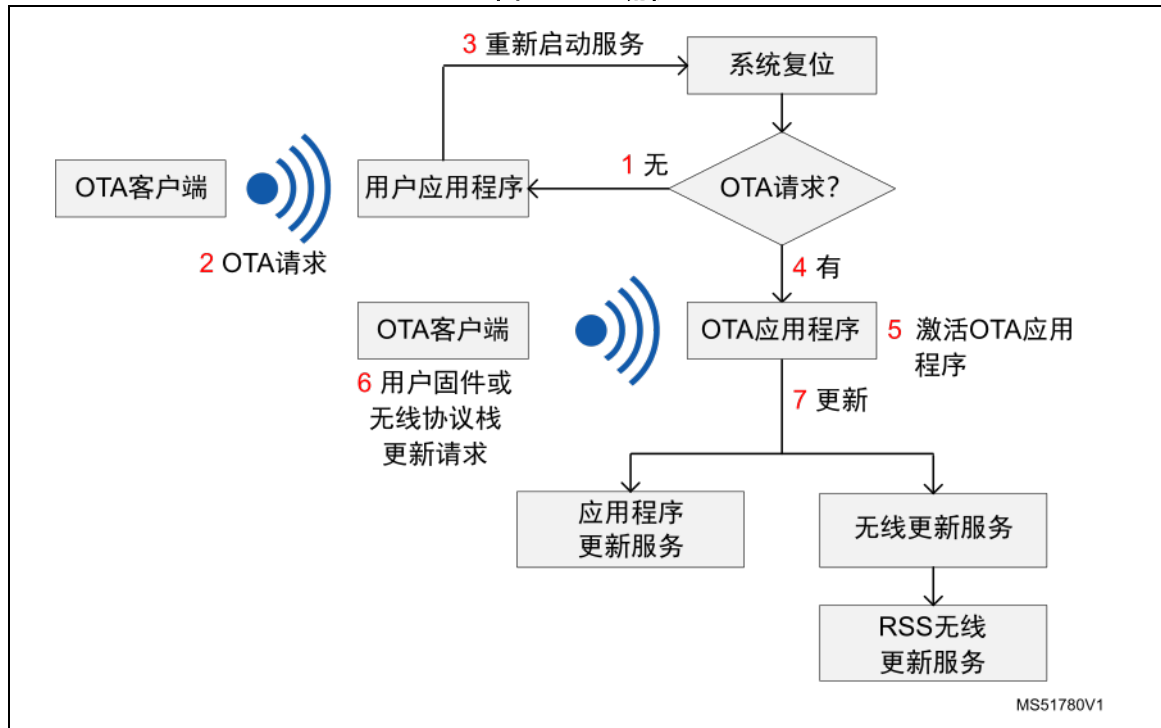
### 3.1.2 OTA 应用程序

OTA 应用程序是一个独立运行的应用程序，集成了用于管理固件传输并将其烧录到 Flash 存储器中准确位置的相关低功耗蓝牙服务。它在用户应用程序接收 OTA 请求后由 CPU1 激活和执行。

流程如中所 [图 3](#) 示，其中红色数字表示不同的处理阶段。

初始状态是用户应用运行（1）的状态。接收到 OTA 请求（2）时，将触发系统复位（3）且 OTA 应用程序被激活（4），从而监听客户端请求（5）。针对无线固件或用户应用程序的更新提供不同服务（6 和 7）。

图 3. OTA 流程



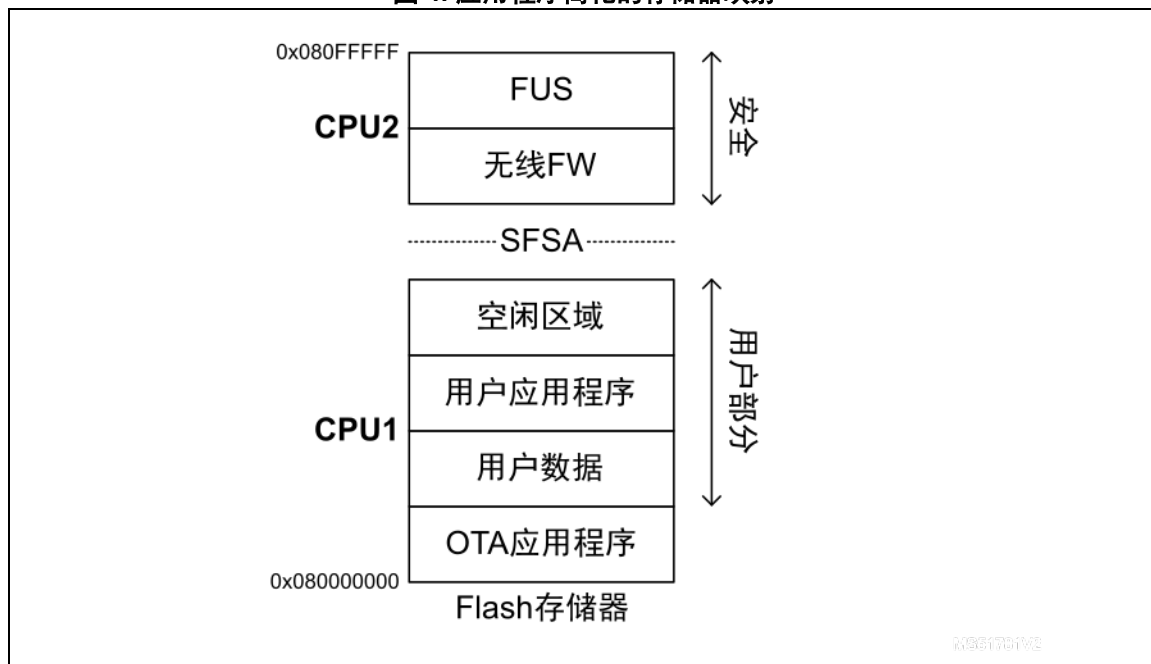
**注意：** 凭借此固件架构，当前用户应用程序负责接收针对用户应用程序和无线固件的 OTA 请求。这是该过程的入口点。然后，OTA 应用程序将与客户端进行通信，以更新其中一个或另一个固件。

### 3.1.3 存储器架构

STM32WBFlash 存储器分为两部分：用户部分和安全部分。用户部分可供 CPU1 访问且包含 OTA 应用程序，及用户应用程序代码和数据。安全部分仅可供 CPU2 访问且包含 FUS 和无线固件。此部分不可供调试端口、用户应用程序或自举程序访问。选项字节 SFSA 提供两个区域之间的边界，上述存储器为安全部分（请参见 [图 4](#)）。SFSA 值取决于无线固件大小，可在安装过程中对其进行调整。

OTA 过程中涉及固件的位置如图 4 中所示。

图 4. 应用程序简化的存储器映射



## FUS

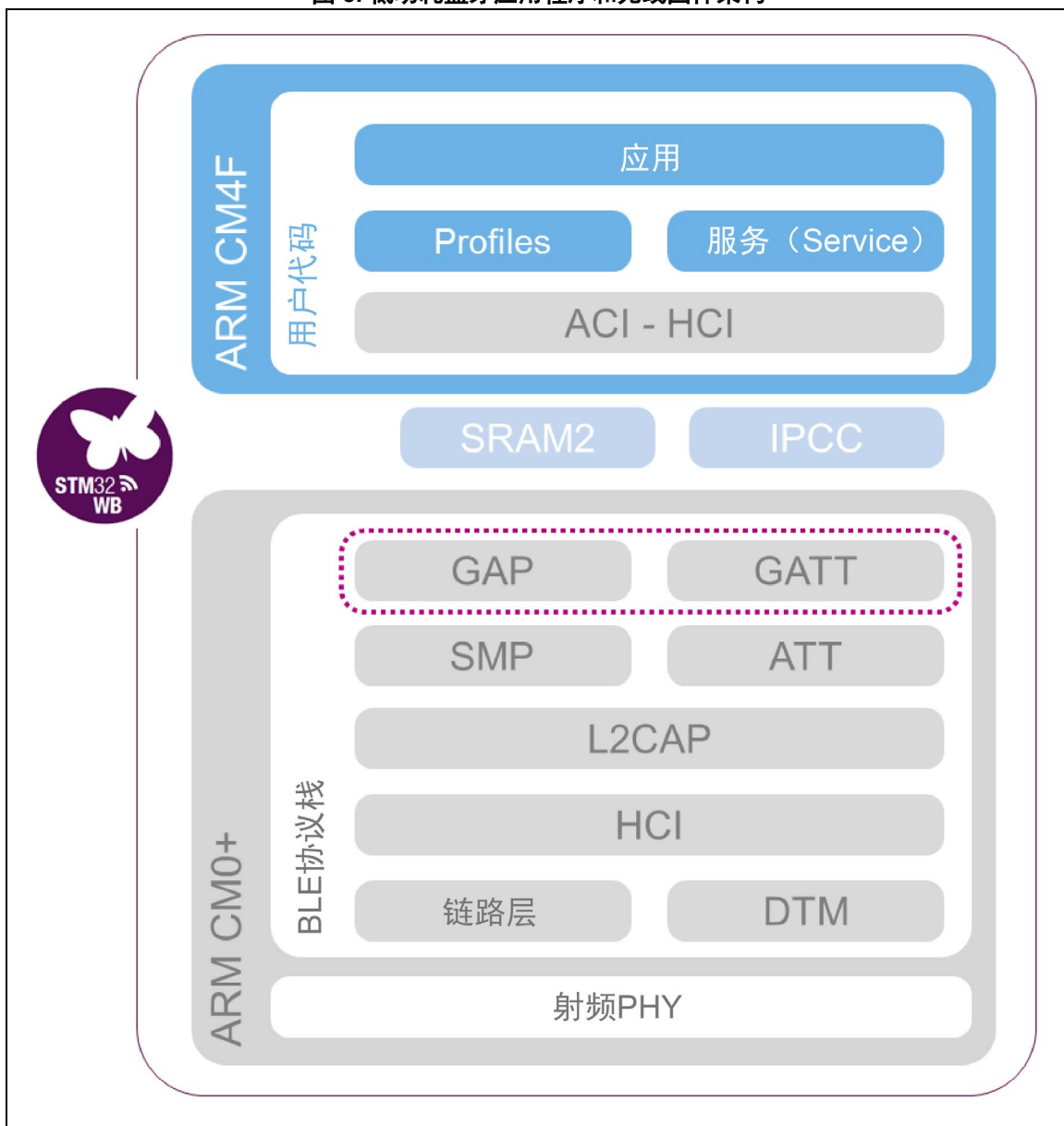
FUS 储存在 Flash 存储器的顶部。包含用于无线固件解密和验证的安全固件。它还为密钥提供安全存储。有关其功能的详细说明，请参阅[2]。

## 无线固件

它存储在 FUS 的下面。该固件驱动 MCU 的 RF 部分。支持以下几种协议：低功耗蓝牙全堆栈、仅 BLE HCI 层、Thread MTD、Thread FTD、并发模式 (BLE + Thread FTD) 和 MAC 802.15.4（底层访问 RF MAC）。所有协议栈均可从 [www.st.com](http://www.st.com) 获得。

对于 BLE，协议栈实施实时性方面。它包含用于管理所有实时链路层和无线电 PHY 交互所需的 LE 控制器和 LE 主机（图 5）。

图 5. 低功耗蓝牙应用程序和无线固件架构



### 用户应用程序和数据

这是主用户应用程序。它是设备的应用部分，实施依赖于无线协议的低功耗蓝牙配置文件和服务。得益于 SRAM2 和 IPCC 硬件模块，可通过实施的信箱系统实现用户应用程序和无线部分的通信（图 5）。它存储在用户 Flash 存储器中的 OTA 应用程序之上。

为方便起见，可以在两个应用程序之间，为在固件更新期间未擦除的应用程序数据保留一个扇区。

**注意：** 如要更新，必须提供可用于新无线映像文件下载的空闲 Flash 存储器区域。

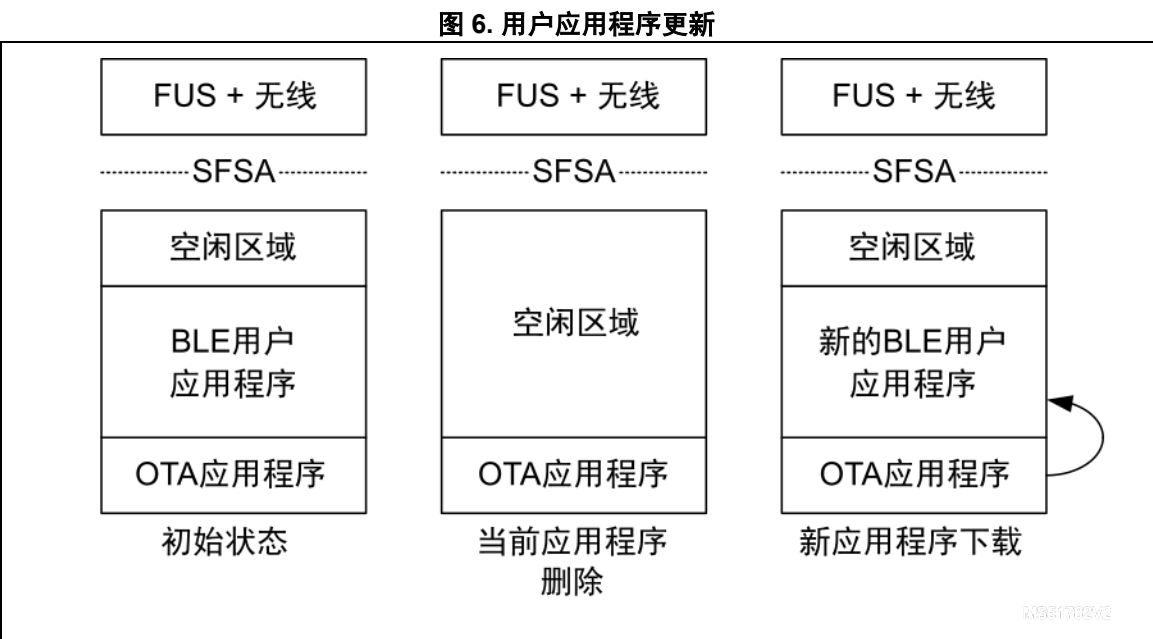
OTA 应用程序

它存储在 Flash 存储器的前六个扇区中（[0x08000000：0x08006000]）。系统服务后直接运行这段程序（CPU1 启动地址位于 Flash 存储器开始处）。如未检测到客户端发出的 OTA 请求，且检测到 Flash 中存在完整的用户应用程序，则直接跳转并执行用户应用程序。否则，停留在 OTA 应用程序运行状态并等待客户端的 OTA 请求。

**注意：**不得删除 OTA 应用程序；如果在该过程中出现问题，则设备无法恢复。建议针对 OTA 应用程序设置写保护。

3.1.4 用户应用程序更新

图 6 显示了发送用户应用程序更新请求后由 OTA 应用程序执行的操作序列。



首先，删除接收新固件的扇区。对 Flash 存储器进行重新编程时，此过程必不可少。建议对包含需要保持不变的应用程序数据区域设置写保护。

第二步包括下载新映像。完成固件传输后，OTA 将跳到新应用程序。

3.1.5 无线固件更新

无线固件加密提供且已签名。需要安全 FUS 固件，以对协议栈进行解密并将其安装在 Flash 存储器的安全部分。

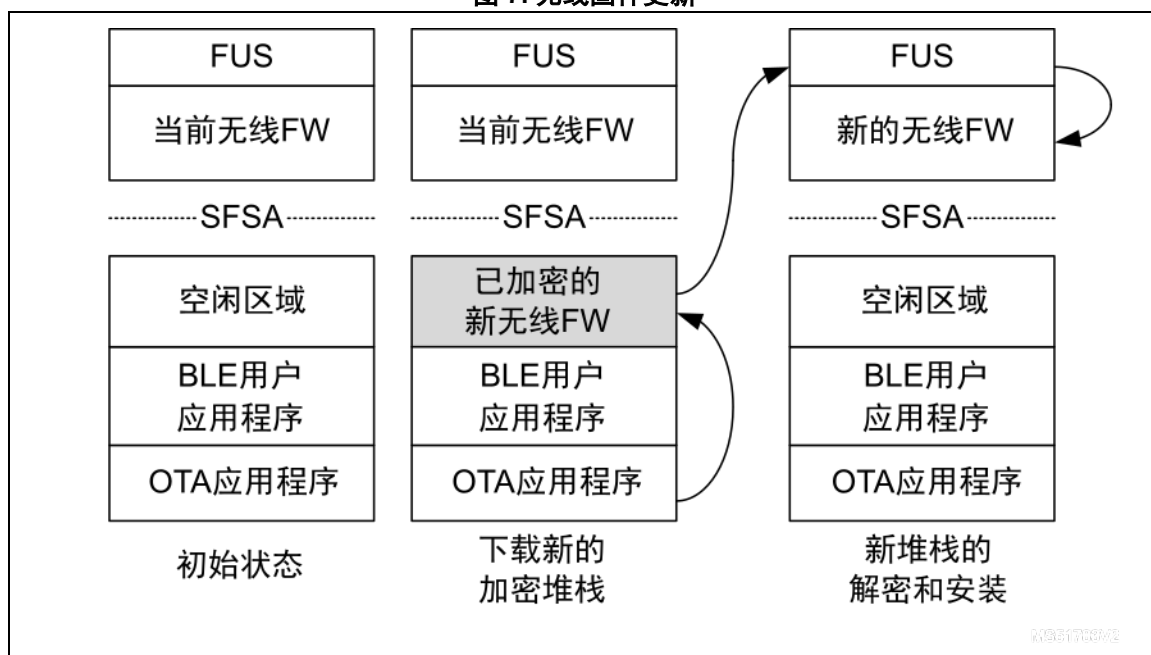
无线安装的第一步与用户应用程序的相同：需要删除的用户 Flash 存储器扇区以存储加密映像（图 7 中的“空闲区域”）。

在第二步中，OTA 应用程序下载加密的协议栈（包括新固件，图 7 中以灰色显示）并请求 FUS 更新服务。

第三步由在 CPU2 上运行的 FUS 执行。它验证、解密并安装协议栈。在安装结束时自动设置与安全边界（SFSA、SBRSA 和 SNBRSA）和复位向量（SBRV）相关的选项字节。

仅在通过所有验证和完整性校验后才能替换当前的无线固件，以确保设备的可连接性。

图 7. 无线固件更新



在升级无线协议栈过程中，低功耗蓝牙用户应用程序有可能会被删掉或被新的无线协议栈覆盖，以释放更多的 Flash 空间来容纳更大的协议栈，用户应用程序需要在协议栈升级过程结束后重新烧录。

## 3.2 用于固件更新的低功耗蓝牙服务的特性

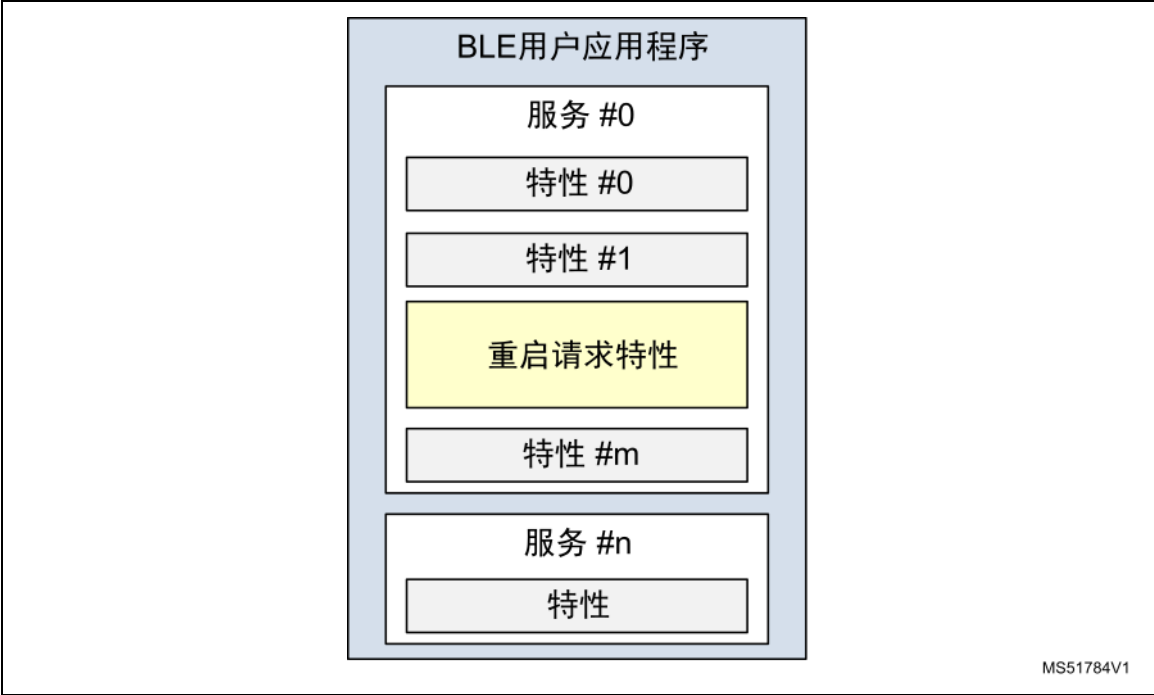
### 3.2.1 BLE 应用程序 – 重启请求特性

#### 低功耗蓝牙服务和特性

本应用笔记中考虑的用户应用程序为低功耗蓝牙应用程序。它支持特定的 GATT 服务：为 SIG 配置文件（如信标器、心率）定义的标准服务或用户定义的服务。这些服务具有一组特性值，及特定的访问权限。

开发人员可以扩展某个服务中的特性数量，甚至添加一些标准化的特性。为了从用户应用程序调用 OTA 应用程序，将“重启特性”（图 8 中的黄色框）添加到用户应用程序支持的服务。当客户端通过低功耗蓝牙协议设置该特性相关标志时，用户应用程序将跳到 OTA 应用程序。

图 8.低功耗蓝牙用户应用程序中添加的 OTA 重启特性



重启请求

重启请求具有三个要由客户端通知的不同字段：

- 1. 启动模式，用户应用程序重启或跳到 OTA 应用程序
- 2. 将在其中下载新应用程序的第一个扇区编号
- 3. 需要删除的扇区数量（新固件大小，粒度为 4kbytes）；

包含该特性的服务取决于用户应用。

表 1.重启请求特性

服务				
BLE 用户应用程序之一	UUID	-		
	特性			
基址	功能	OTA 应用程序请求设备重启		
	大小	3		
	模式	Write		
	UUID	0000FE11-8e22-4541-9d4c-21edae82ed19		
	位域	0	自举模式：	0x00 应用程序 0x01 OTA 应用程序
		1	扇区索引	-
		3	要擦除的扇区数	0x00 – 0xFF

3.2.2 OTA 应用 - 服务和特性

OTA 应用程序是标准低功耗蓝牙应用程序。它提供了用于实现固件更新的特定服务以及相关特性。

表 2. OTA 服务和特性声明

服务				
OTA FW 更新	UUID	0000FE20-cc7a-482a-984a-7f2ed5b3e58f		
特性				
基址	功能	要存储文件的地址		
	大小	4		
	模式	Write		
	UUID	000FE22-8e22-4541-9d4c-21edae82ed19		
	位域	0	动作	0x00： 停止所有上传 0x01： 开始无线文件上传 0x02： 开始用户应用程序文件上传 0x07： 文件上传完成 0x08： 取消上传
		1-3	地址	0x007000
文件上传确认重启	功能	在文件上传后确认应用程序的重启		
	大小	1		
	模式	指示		
	UUID	0000FE23-8e22-4541-9d4c-21edae82ed19		
	位域	0	0x01	重启
OTA 原始数据	功能	要传输文件的数据（以 20 字节划分）		
	大小	20		
	模式	写入，无响应		
	UUID	0000FE24-8e22-4541-9d4c-21edae82ed19		
	位域	0-19	原始数据	

此服务在广告序列后由客户端应用程序识别。

3.2.3 广播

在设备和客户端之间建立连接前，必须广播重启请求的可用性。

将该信息添加到制造商特定广播字段（0xFF 类型）。下表介绍了封装结构。在组 B 功能（位 13）中指定重启请求能力。

表 3. AD 结构 - 主

AD 字段名称	AD 类型	AD 长度	记录大小
TX_POWER_LEVEL	0x0A	2	3
COMPLETE_NAME	0x09	8	9
MANUF_SPECIFIC	0xFF	13	14
FLAGS	0x01	2	3

表 4. AD 结构 - 制造商特定字段

字节	名称	值	备注
0	长度	8	-
1	类型	0xFF	制造商特定
2	版本	0x01	-
3	ID	0x00 - 0xFF	用于 OTA 加载程序的 0x86
4	组 A 功能	位字段	保留
5			
6			
7	组 B 功能		用于启用 OTA 的设备的 0x20
8-13	公共设备地址	字节	可选

表 5. AD 结构 – 组 B 功能

位	位域
15	保留
14	Thread 支持
13	OTA 重启请求
12-0	保留

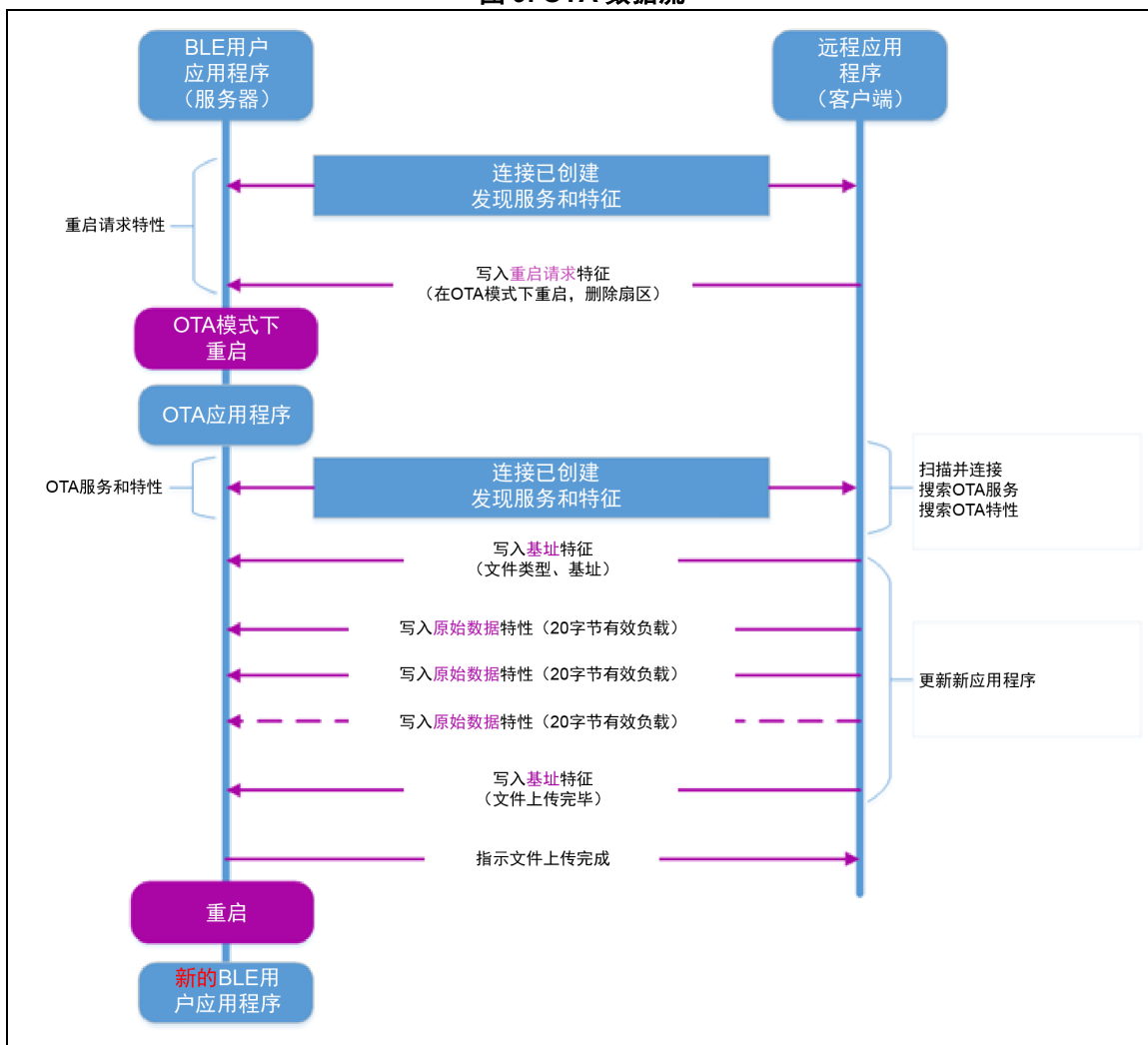
### 3.3 流程说明

[图 9](#) 详细说明了在更新过程中设备和客户端之间的消息和数据交换。

用户应用程序建立首次连接。向客户端广播重启请求特性（步骤 1）。客户端请求设备重启时，将在 OTA 应用程序和客户端之间建立新的连接（步骤 2）。



图 9. OTA 数据流



### 3.4 无线和 FUS 更新流程

无线协议栈和 FUS 是经过加密的。下载后，将调用一个 FUS 服务来解密新固件并在正确的位置安装新固件。

FUS 可以将新的无线协议栈从其下载区域移出，以优化内存空间。

但是，如果新的 FUS 映像大于现有的 FUS 映像，升级过程中可能需要将无线协议栈向 Flash 存储器的低地址区域移动（其实地址由 FUS 决定）。如果用户程序已经占用了无线协议栈原来起始扇区的邻居扇区，则在该操作期间用户代码可能会被擦除。更多详细信息，请参见[2]。

## 3.5 更新程序客户端

OTA 请求来自具有专用应用程序的低功耗蓝牙客户端。STMicroelectronics 提供具有此功能的两种客户端：

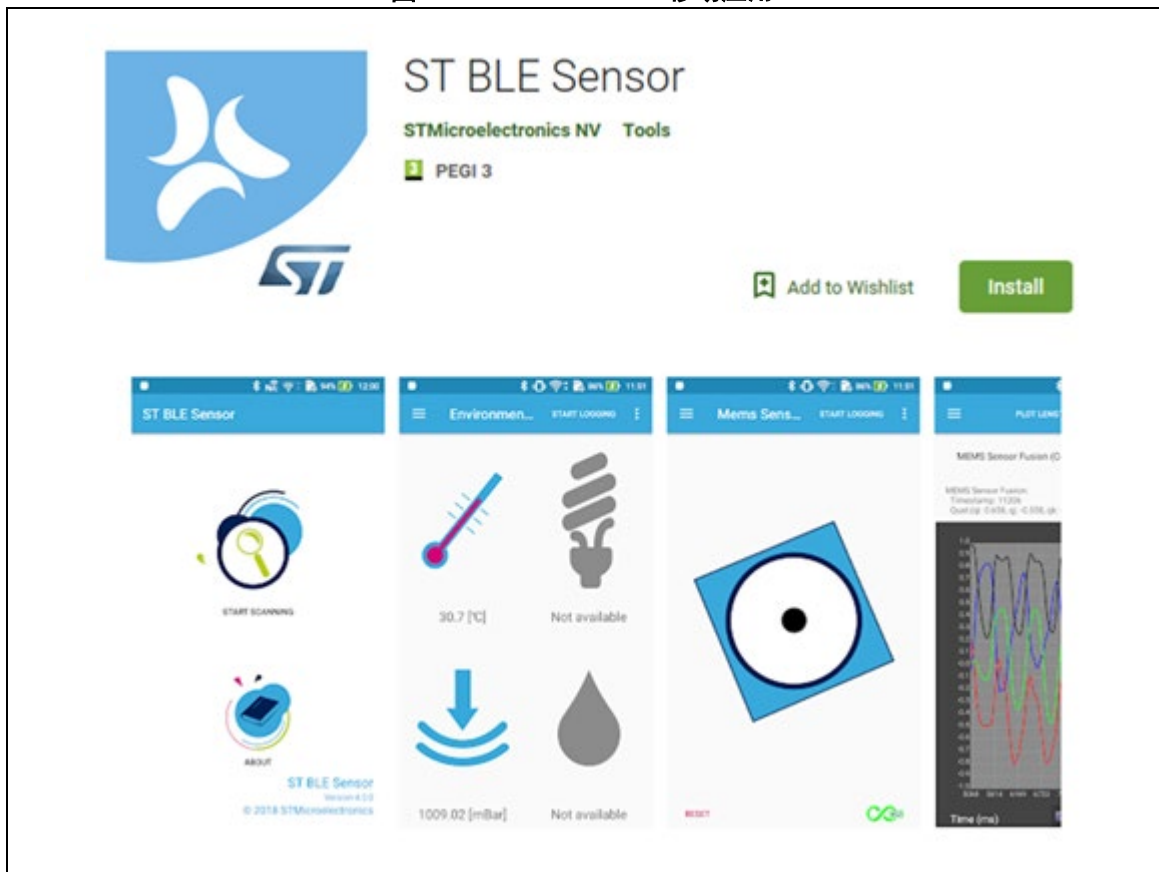
1. ST BLE Sensor，这是支持两种更新类型（应用程序和无线固件）的移动应用程序
2. STM32CubeMonitor RF，这是允许用户使用具有低功耗蓝牙透传模式实施的蓝牙模块更新其应用程序的 PC 工具。

### 3.5.1 ST BLE Sensor

ST BLE Sensor 应用程序用于与低功耗蓝牙协议兼容的 ST 开发板和固件结合使用。可访问板载的所有传感器数据，并将之上传到各类云服务器上。它还支持固件 OTA 更新过程。

该应用程序在 App Store®和 Google Play™上提供（图 10）。此外，可从 <https://github.com> 上的专门网页免费获得 SDK 和应用程序源代码。

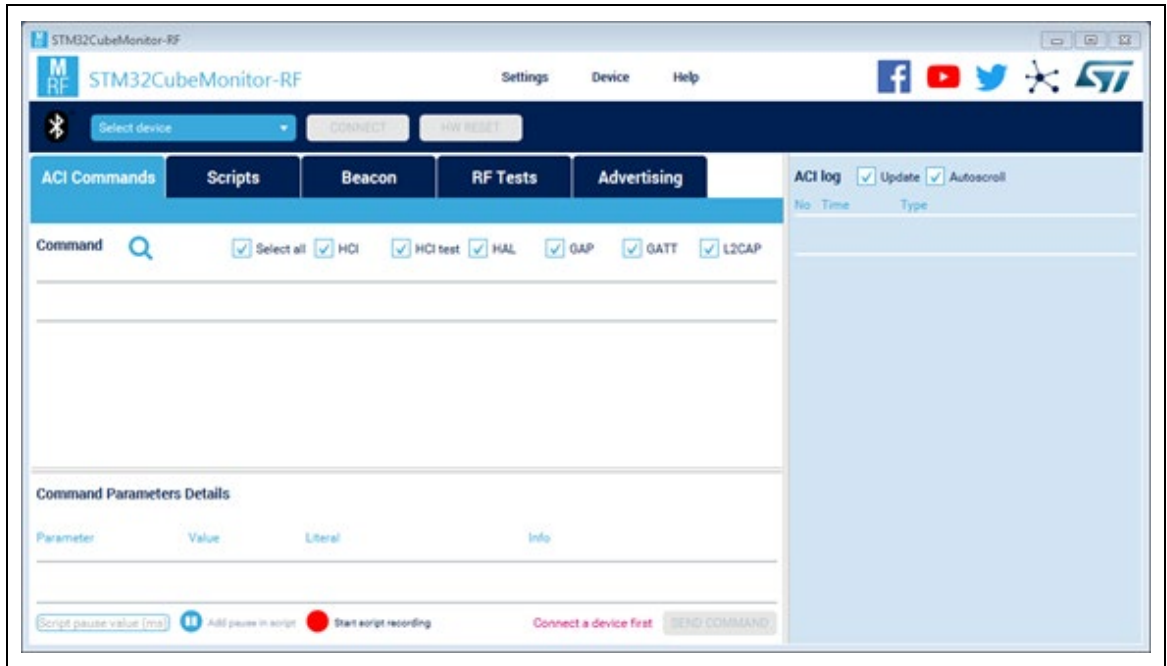
图 10. ST BLE Sensor 移动应用



### 3.5.2 STM32CubeMonitor-RF

STM32CubeMonitor-RF (图 11) 是一款可帮助开发人员测试其基于 STM32 微控制器所设计产品的软件工具。

图 11. STM32CubeMonitor-RF

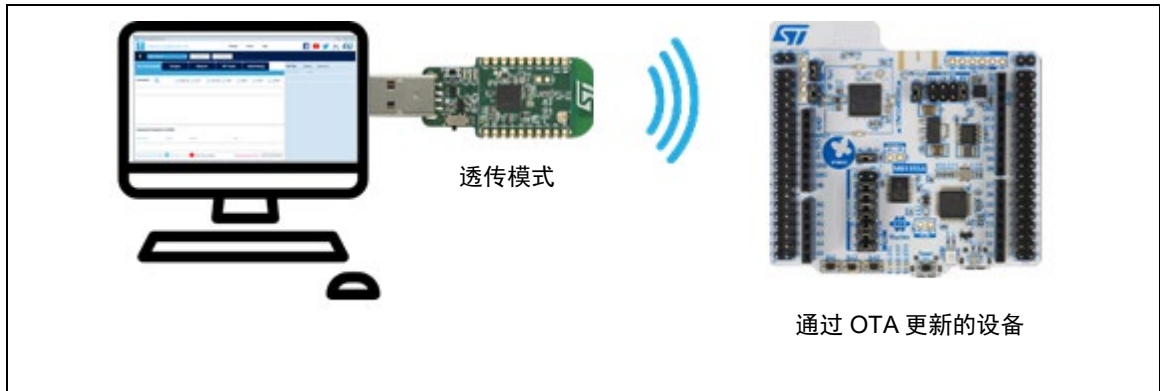


此工具可执行以下操作：

- 发送和接收测试包，以检查射频板的效率和计算包错误率（PER）
- 向低功耗蓝牙控制器发送命令，以进行标准化测试
- 发送和接收 BLE 命令，以进行快速的应用程序原型开发
- 通过 BLE 命令配置各种信标发送设备
- 将 OTA 数据从一台设备传输到另一台设备，以在无有线连接的情况下对远程设备进行配置或编程
- 向 OpenThread 设备发送命令，以进行应用程序原型开发。

STM32CubeMonitor-RF 通常通过 UART（物理链路）连接到目标设备 STM32WB。但是，OTA 操作需要使用加载了特殊固件（称为“透传模式”）的 STM32WB55 USB 蓝牙模块（图 12）。PC 工具发送/接收的数据和命令将直接传输至要更新的设备的 HCI 接口。STM32Cube\_FW\_WB 包中提供“透传模式”应用程序的固件。

图 12. STM32CubeMonitor RF 使用处于透传模式的 USB 蓝牙模块



第 4.3 节中详细介绍了使用 STM32CubeMonitor-RF 进行的 OTA 过程。

## 4 OTA 分步过程

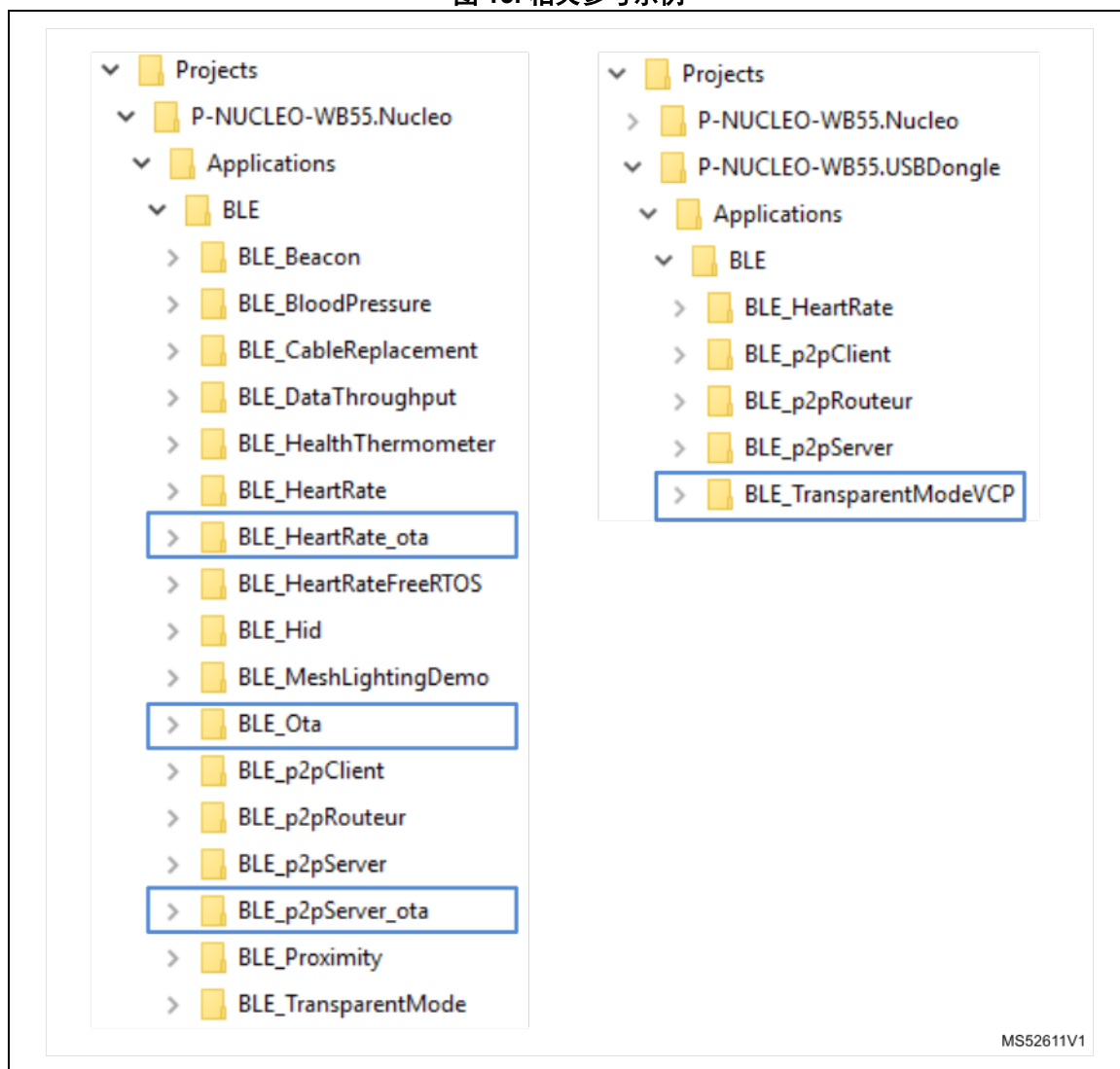
本节介绍如何在 Nucleo-WB55RG 板上更新 STM32WB 设备的固件。介绍了两个用例：

1. 使用 ST BLE Sensor 移动应用程序更新用户应用
2. 使用 STM32CubeMonitor-RF 工具和 USB 蓝牙模块更新用户应用。

分步过程中会使用到 3 个参考示例和透传模式固件（在图 13 中重点介绍）。它们在 [www.st.com](http://www.st.com) 上的 STM32Cube\_FW\_WB 包中提供。

目标设备为 Nucleo 板，移植到所有 STM32WB 设备和板上很容易。

图 13. 相关参考示例

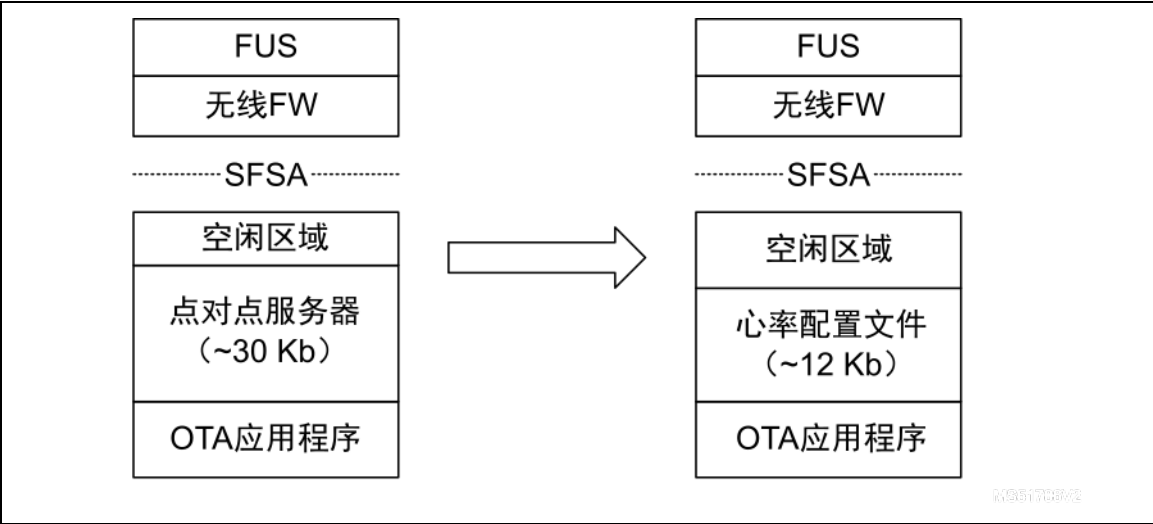


4.1 项目设置

4.1.1 用户应用程序

此处用到的两个序支持不同设备功能的用户应用程序。STM32WB 设备首先烧录并运行点对点服务端应用程序，OTA 升级过程结束后，新升级的程序支持典型的心率配置文件（图 14）。

图 14. 用户应用程序更新 - 用例



二者都是低功耗蓝牙应用程序，提供具有表 6 中定义的特性的服务。

将重启请求（如第 3.2.1 节中定义）添加到其主要服务中。即使此处所示的用例将此功能用于点对点服务器，它也被添加到心率配置文件中，以进行进一步更新。

表 6. 示例应用程序的服务和特性

应用	服务 (UUID)	特性
点对点服务器	P2P_SERVICE 0000FE40-cc7a-482a-984a-7fed5b3e58f	P2P_WRITE_CHAR
		P2P_NOTIFY
		REBOOT REQUEST
心率 SIG 配置文件	DEVICE_INFORMATION_SERVICE (0x180A)	MANUFACTURER_NAME
		MODEL_NUMBER
		SERIAL_NUMBER
		HARDWARE_REVISION.
	HEART_RATE_SERVICE (0x180D)	HEART_RATE_MEASUREMENT
		SENSOR_LOCATION
		CONTROL_POINT
		REBOOT REQUEST

表 6. 示例应用程序的服务和特性（续）

应用	服务 (UUID)	特性
OTA 应用程序	OTAS_SVC 0000fe20-cc7a-482a-984a-7f2ed5b3e58f	OTA_BASE_ADR_CHAR
		OTA_CONF_CHAR:
		OTA_RAW_DATA_CHAR

### 添加 REBOOT\_REQUEST 特性

对于两个应用程序，应该修改文件 ble\_conf.h，以包括新特性。将 BLE\_CFG\_OTA\_REBOOT\_CHAR 设置为 1 以进行添加。

```
/******
```

```
* 空中功能 (OTA) - STM 专利
```

```
*****/
```

```
#define BLE_CFG_OTA_REBOOT_CHAR 1 /*< 重启 OTA 模式特性 */
```

### 更改链接器文件

实现 OTA 功能需要重新规划 Flash 存储器空间，将用户应用程序放置在 OTA 应用程序后高位地址空间，因此需要修改项目链接文件，使最终生成的二进制可执行文件中的复位矢量上移 7 个扇区的地址。

#### IAR™ ICF file

在 SRAM1 中添加一个 4 字节的偏移量，“重启请求”特性使用该 32bit 字表示系统重启后的 OTA 请求。

```
/*-CPU 向量表 - 无 OTA 支持*/
```

```
define symbol __ICFEDIT_intvec_start__ = 0x08000000;
```

```
/*-存储器区域-*/
```

```
define symbol __ICFEDIT_region_ROM_start__ = 0x08000000;
```

```
define symbol __ICFEDIT_region_ROM_end__ = 0x0807FFFF;
```

```
define symbol __ICFEDIT_region_RAM1_start__ = 0x20000004;
```

```
define symbol __ICFEDIT_region_RAM1_end__ = 0x2002FFFF;
```

```
/*-CPU 向量表 - 具有 OTA 应用程序*/
```

```
define symbol __ICFEDIT_intvec_start__ = 0x08007000;
```

```
/*-存储器区域-*/
```

```
define symbol __ICFEDIT_region_ROM_start__ = 0x08007000;
```

```
define symbol __ICFEDIT_region_ROM_end__ = 0x0807FFFF;
```

```
define symbol __ICFEDIT_region_RAM1_start__ = 0x20000004;
```

```
define symbol __ICFEDIT_region_RAM1_end__ = 0x2002FFFF;
```

#### Keil® scatter file

```
/*-CPU 向量表 - 无 OTA 支持*/
```

```
LR_IROM1 0x08000000 0x00080000{ ; load region size_region
```

```
ER_IROM1 0x08000000 0x00080000 { ; load address = execution address
```

```

        *.o (RESET, +First)
        *(InRoot$$Sections)
        .ANY (+RO)
    }
    RW_IRAM1 0x20000004 0x2FFFC { ; RW data
        .ANY (+RW +ZI)
    }

/*-CPU 向量表 - 具有 OTA 应用程序*/
LR_IROM1 0x08007000 0x260 { ; 载入区域 size_region
    ER_IROM1_LOW 0x08007000 0x260 { ; 载入地址 = 执行地址
        *.o(RESET, +First)
        *.o (TAG_OTA_START)
    }

    RW_IRAM1 0x20000004 0x0002FFFC { ; RW data
        .ANY (+RW +ZI)
    }

```

### 删除 VTOR 表配置

当希望应用程序被 OTA 下载时，SCB→VTOR 不能被修改，因为它已经被 BLE\_Ota 应用程序设置为正确的值。

在 SstemInit() 中注释掉这行代码：

```
// SCB->VTOR = VECT_TAB_OFFSET;
```

### 编译和编程

编译和下载点对点应用程序到 Nucleo 板中。

编译心率应用程序。生成的二进制文件将由 OTA 客户端下载。如果使用 EWARM IDE，生成的二进制文件的目录路径为 EWARM/with\_ota/Exe directory BLE\_HeartRate\_ota.bin。

将此二进制文件复制到运行 ST BLE Sensor 应用程序的智能手机。

## 4.1.2 OTA 应用程序

OTA 应用程序的设置很简单。编译和下载二进制文件到 Flash 存储器起始地址（0x08000000）。

为了检查是否已经写入完整的用户应用程序固件。在固件的末尾写入一个 magic 关键字。得益于 CheckFwAppValidity() 函数，Ble\_Ota 应用程序在启动时检查该 magic 关键字是否存在于应用程序的末尾。

如果在上传过程中复位（可能导致仅写入部分映像），magic 关键字将不存在，已损坏的应用程序将被 Ble\_Ota 启动忽略。

由于应用程序长度信息不存在，magic 关键字的地址保存在地址 0x08007140 处。



为了让计算对用户透明，固件中定义了两个常量：

- 一个常量存放 magic 关键字的地址，并与 TAG\_OTA\_START 部分关联
- 另一个常量存放 magic 关键字，并与 TAG\_OTA\_END 相关联。

4.2 使用 ST BLE Sensor 更新固件

4.2.1 设备连接

手机端运行 ST BLE Sensor APP，并发起点对点的连接，并验证设备端是否工作正常？

设备蓝牙 MAC 地址

Nucleo 板的 MAC 地址使用的是芯片的序列号。它可以在地址 0x 0x1FFF7580 和 0x1FFF7584 处获取。

表 7. 示例（设备蓝牙 MAC 地址 = 80:E1:25:00:50:D6）

寄存器地址	值	位域
0x1FFF7580	0xXX0050D6	MAC 的 LSB 部分@
0x1FFF7584	0xXX80E125	MAC 的 MSB 部分@

## 连接

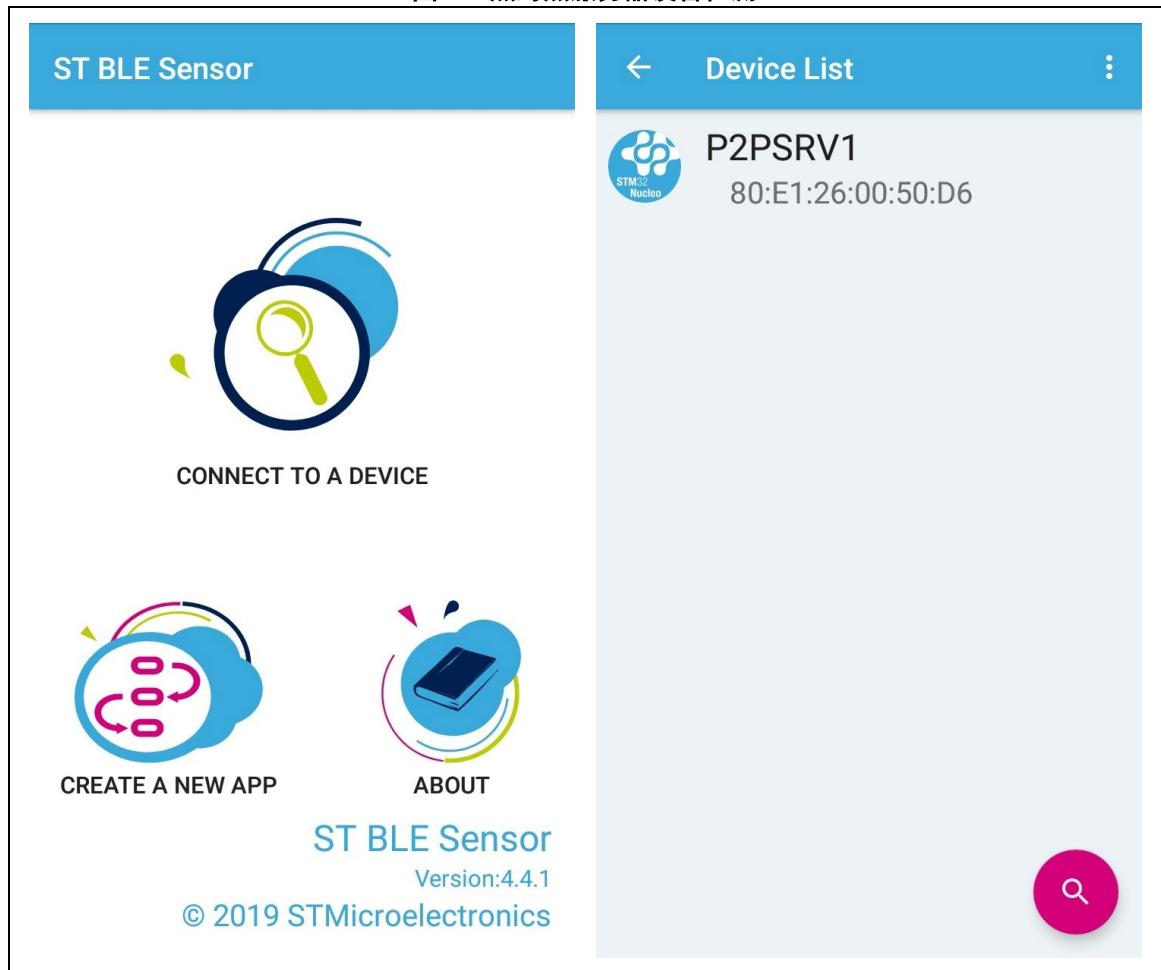
请按照下列步骤操作。

1. 按下 Nucleo 板的复位（SW4）按钮来启动设备广播（外设角色）。

**注意：** Nucleo 板的绿色 LED（LED2）在智能手机扫描过程中闪烁。如果不是这种情况，则因为广播在 1 分钟后停止。按下 Nucleo 板“复位”按钮（SW4）以重新启动广告。

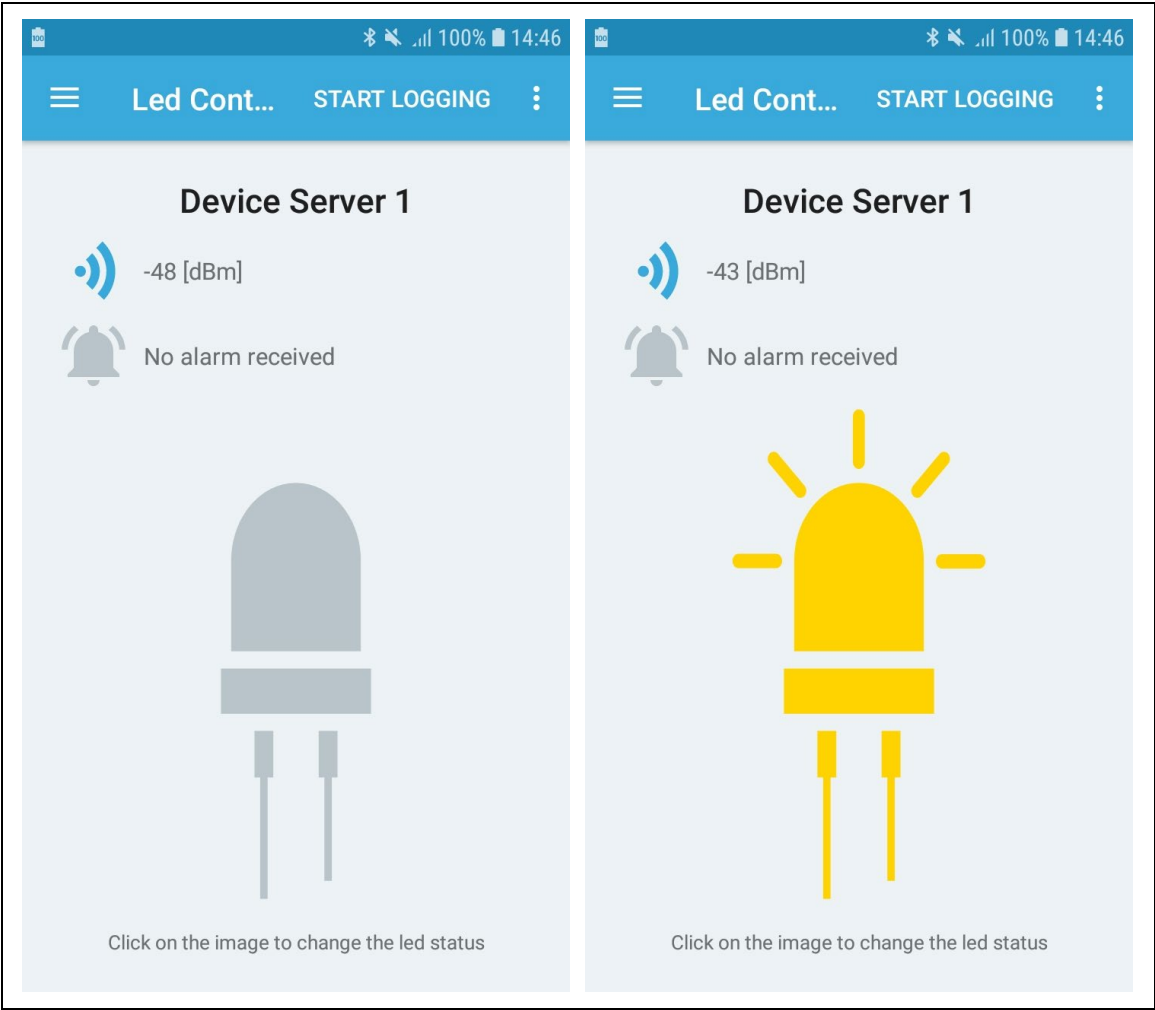
2. 智能手机充当中央设备的角色，它必须进行扫描以检测设备。单击“连接到设备”。将检测到名为“P2PSRV1”的设备，如 [图 15](#) 中所示。检查 BD 地址是否与您的板相对应。

图 15. 点对点服务器设备检测



3. 单击“P2PSRV1”以创建与 Nucleo 板的连接。
4. 出现应用程序主菜单时，智能手机和 Nucleo 板已连接。
5. 可通过单击 LED 图片（[图 16](#)）来打开和关闭 Nucleo 板的 LED1（蓝色）。

图 16. 点对点应用程序/LED 开关

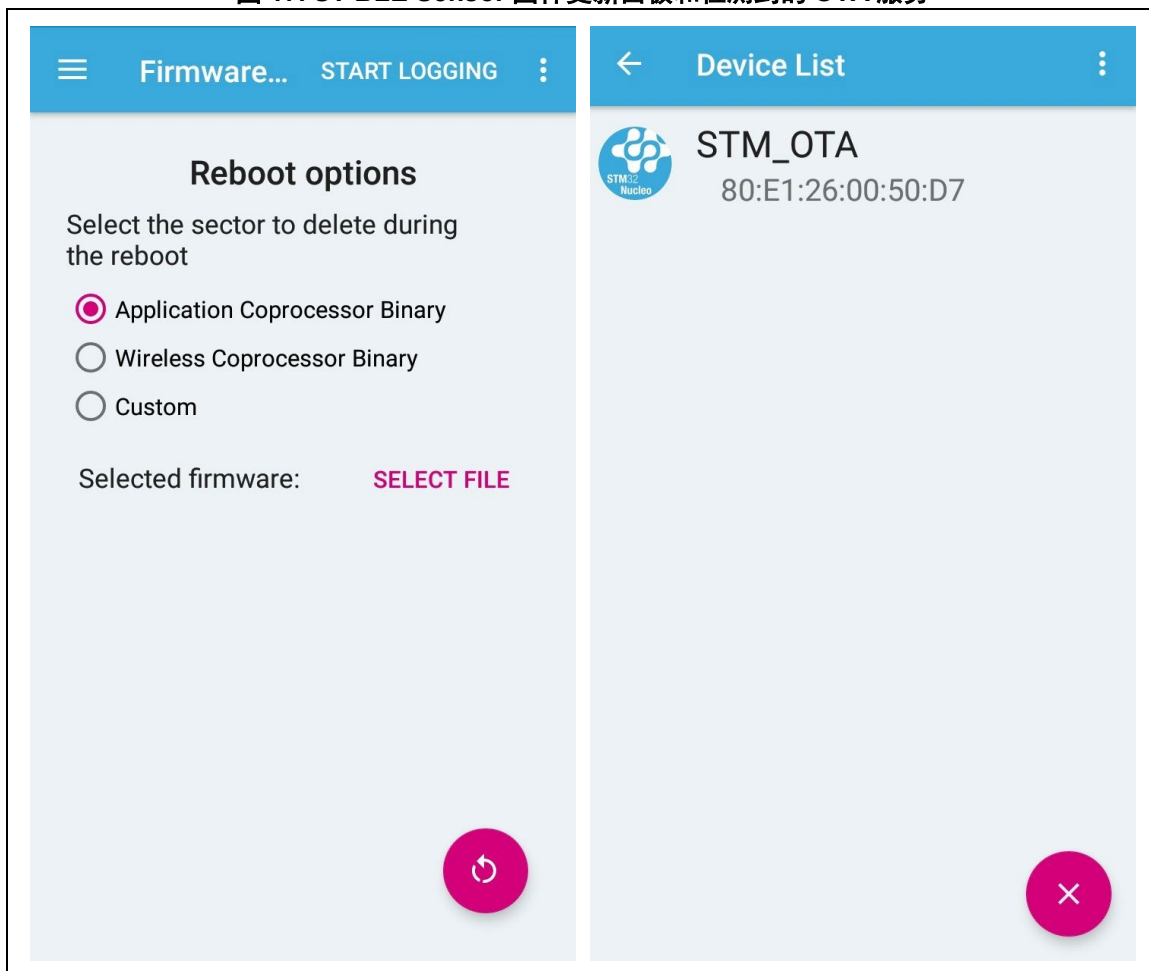


### 4.2.2 用户应用程序更新

检查点对点功能后，滑到左侧以获取固件更新面板（图 17 的左侧）。选择“应用程序”，然后单击屏幕底部的重启符号。您无需在此步骤选择文件名称。

重新扫描以查找新服务（图 17 的右侧）。请注意，OTA 应用程序的蓝牙 MAC 地址是在原来点对点应用程序的 MAC 地址首字节加 1 后得到的。这样做的目的是为了确保客户端启动新的发现服务进程。

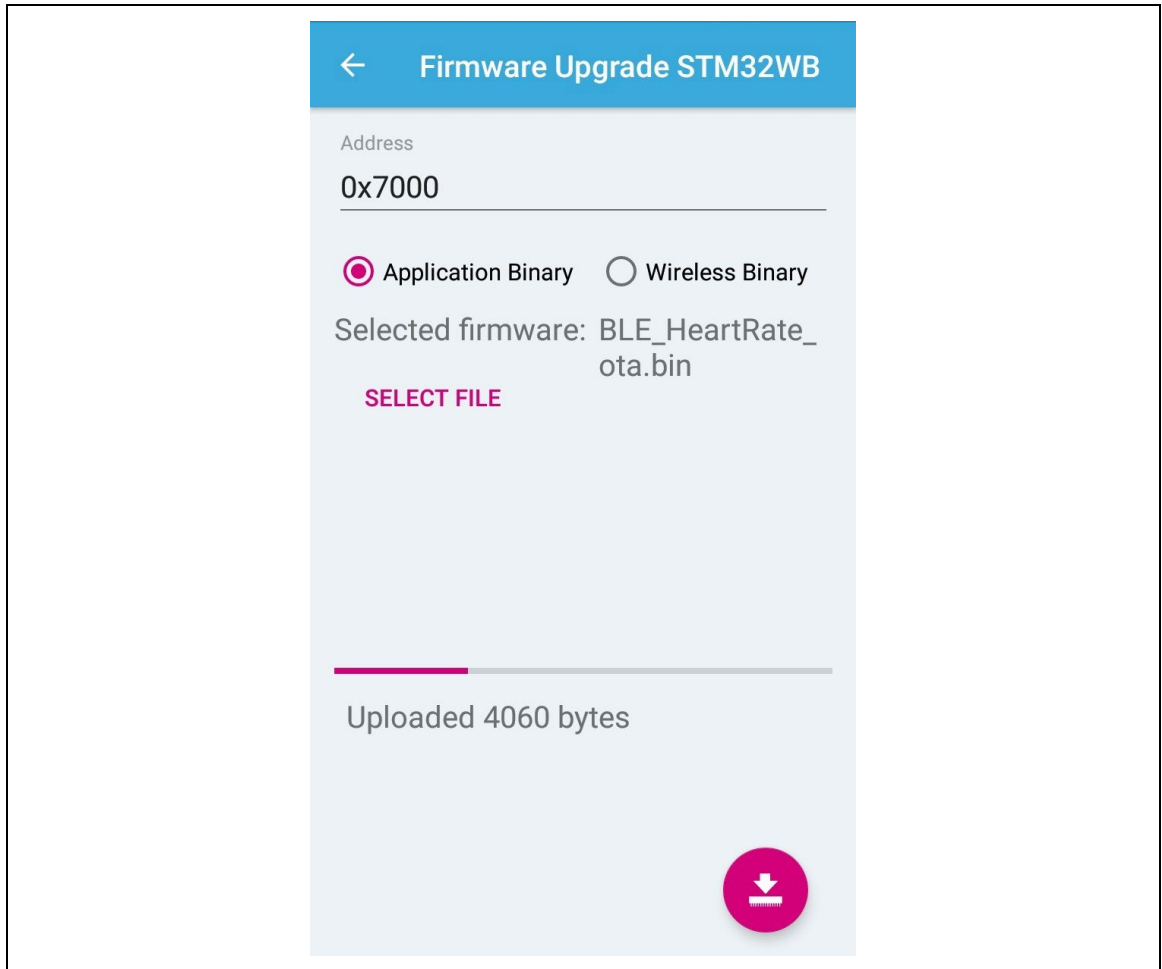
图 17. ST BLE Sensor 固件更新面板和检测到的 OTA 服务



与 STM\_OTA 应用程序（[图 18](#)）连接后，输入新应用程序的下载地址（在本例中为 0x7000）。

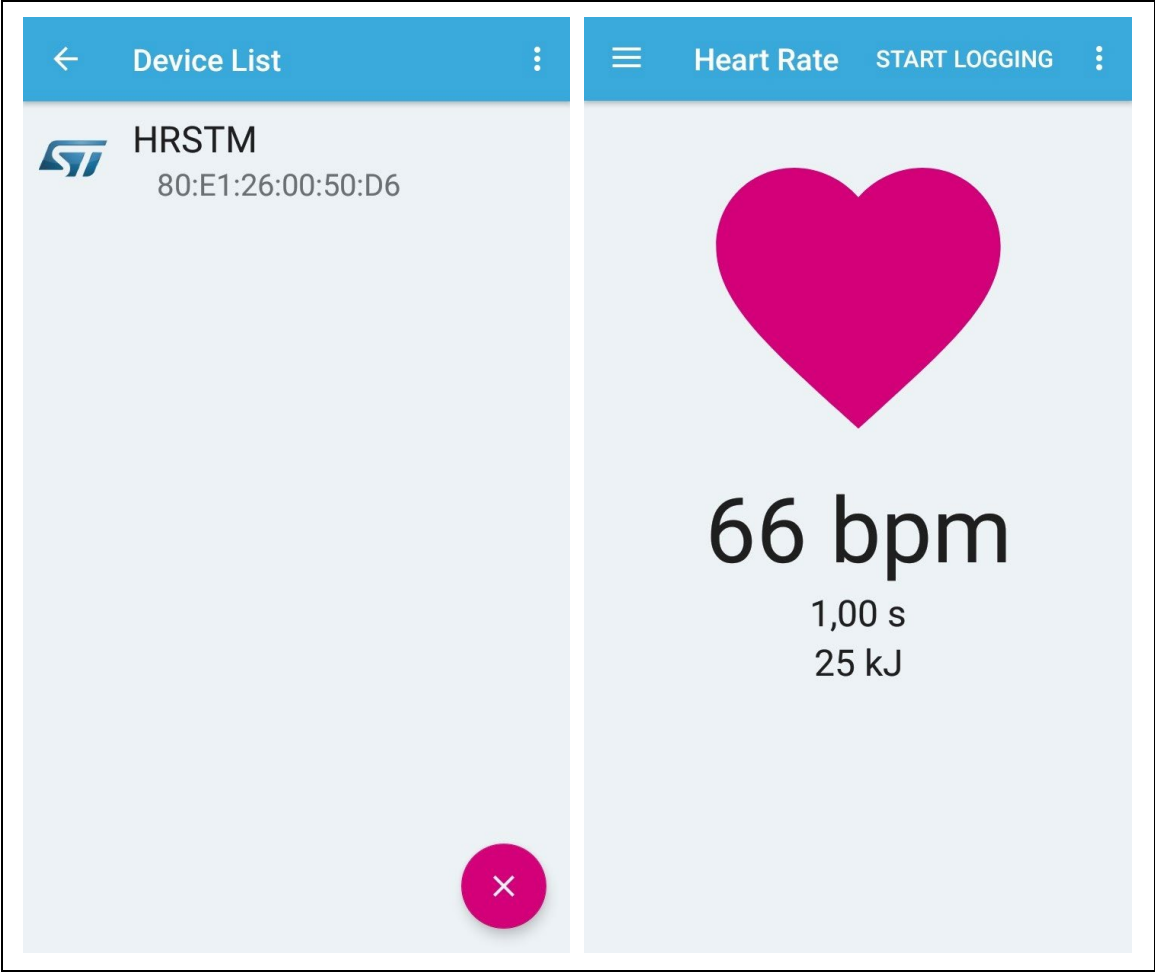
点击“选择文件”按钮，并选中之前传输到智能手机上的心率应用程序二进制文件。选择下载按钮；屏幕上显示上传进度，且最后将显示一个弹出窗口及所用时间（请参见[图 18](#)右侧）。

图 18. 新应用程序下载



重新连接到设备（相同的初始蓝牙地址）并验证功能（[图 19](#)）。

图 19. 更新后的心率配置文件

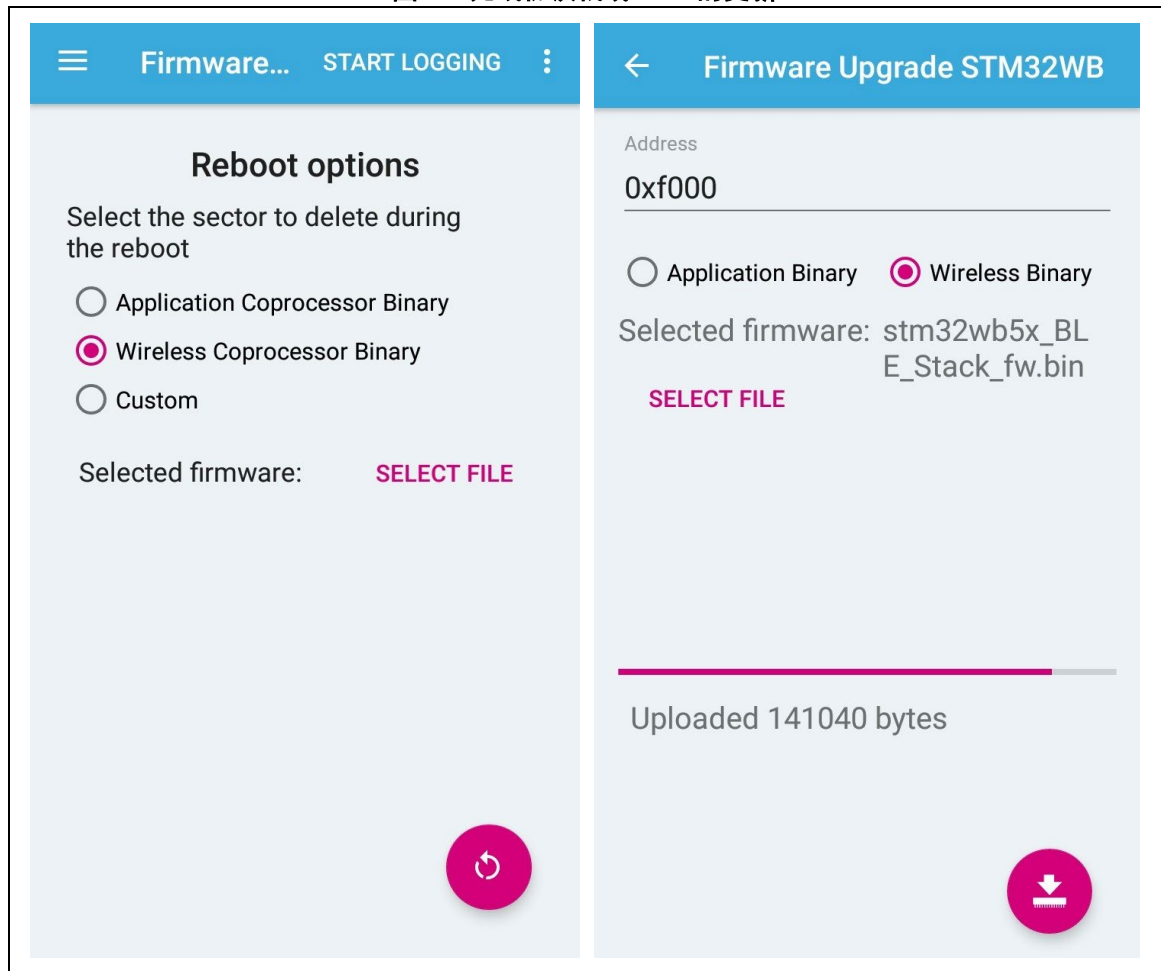


### 4.2.3 无线协议栈或 FUS 的更新

对于 CPU2 固件升级，选择“无线协处理器二进制文件”选项（图 20 左侧）并选择正确的二进制文件。无线协议栈和 FUS 固件是在 <https://github.com> 或 [www.st.com](http://www.st.com) 上的 STM32Cube\_FW\_WB 封装中提供的。

下载之后（图 20 的右侧），新固件由当前的 FUS 解密并安装。

图 20. 无线协议栈或 FUS 的更新



## 4.3 使用 STM32CubeMonitor-RF 进行固件更新

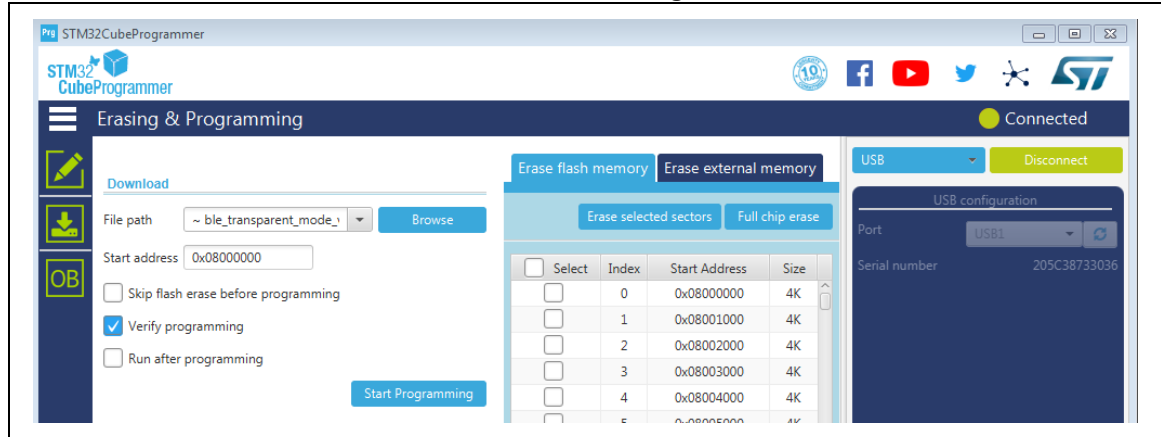
### 4.3.1 透传模式

STM32CubeMonitor-RF 通过蓝牙设备将 HCI 命令和数据传输到目标 Nucleo 板，以模拟真正的 OTA 应用程序。为此，随 Nucleo STM32WB 包提供的 USB 蓝牙模块在“透传模式”下进行配置。

1. 打开并编译 ~\Projects\NUCLEO-WB55.USB Dongle\Applications\BLE\BLE\_TransparentModeVCP 项目。
2. 编程后，将 BOOT0 开关移回 0。

可以使用 STM32CubeProgrammer（版本 2.0 及更高版本）工具在 USB-DFU 模式下轻松编程 USB 蓝牙模块（[图 21](#)）。要访问 DFU 模式，将 BOOT0 开关移回 1（[图 21](#) 右侧）。

图 21. 在 USB DFU 模式下使用 STM32CubeProgrammer 进行 USB 蓝牙模块编程

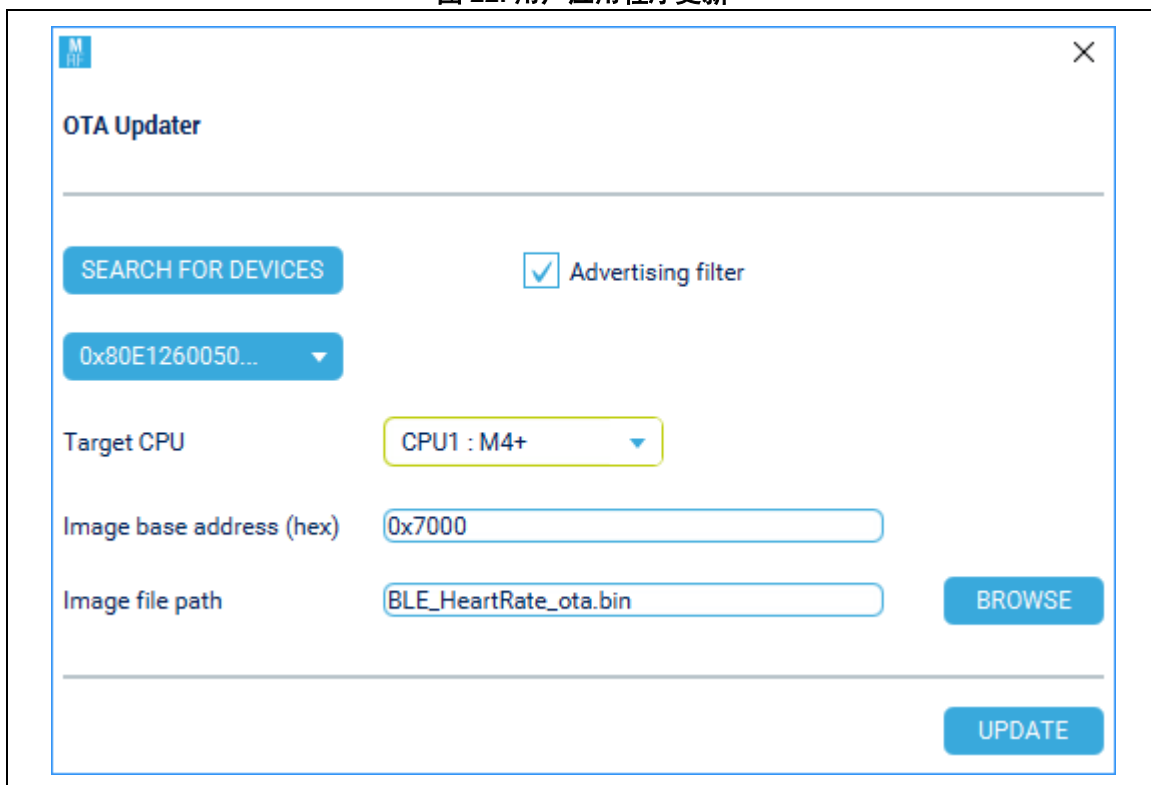


### 4.3.2 用户应用程序更新

1. 打开 STM32CubeMonitor-RF 并选择由 USB 蓝牙模块对应的 COM 端口。连接设备。
2. 确保使用 OTA 应用程序和点对点服务器配置文件正确配置 Nucleo 板（[第 4.1 节](#)）
3. 在上层菜单，单击“Device”和“OTA Updater”
4. 选择广播过滤框并单击“SEARCH FOR DEVICES”
5. 在扫描结束时，单击“选择设备”并选择您的设备（使用正确的蓝牙地址启用 P2PSRV1-OT）
6. 为用户应用程序选择 CPU1（[图 22](#)）或为无线或 FUS 更新选择 CPU2（[图 23](#)）。
7. 浏览并找到正确的二进制文件，然后选择“Update”



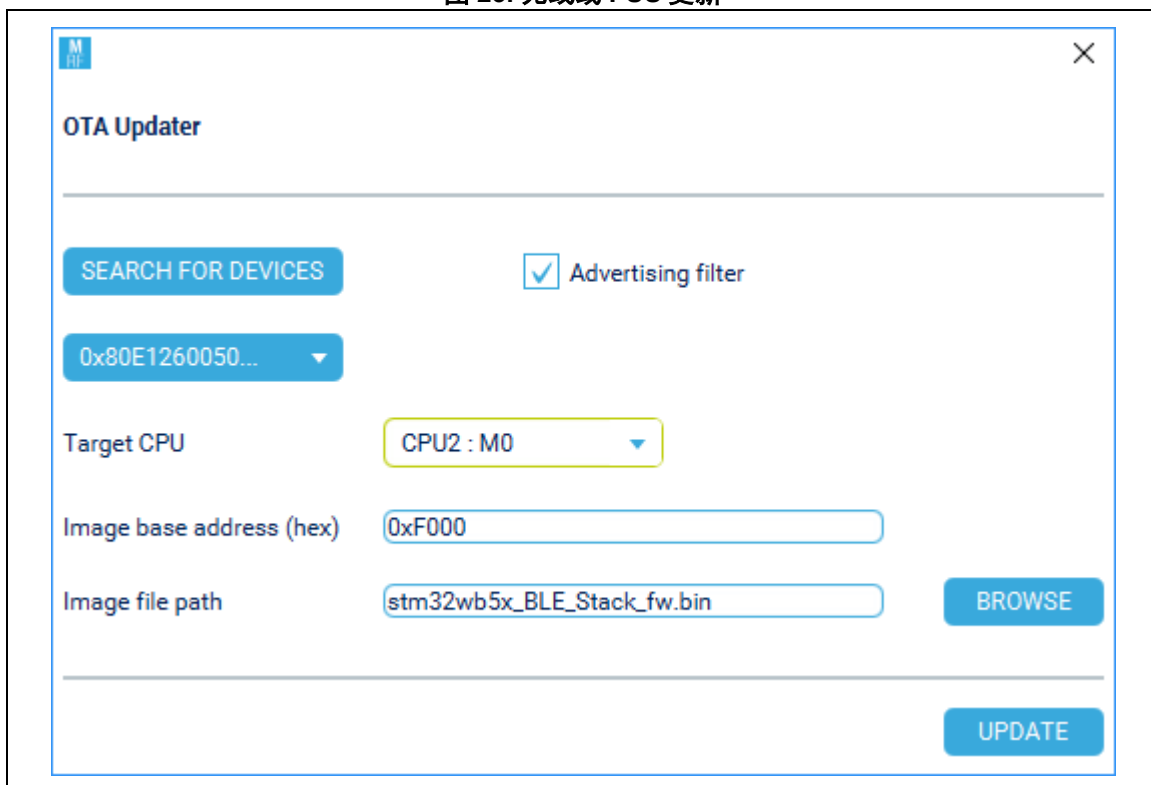
图 22. 用户应用程序更新



The screenshot shows the 'OTA Updater' window with the following settings:

- SEARCH FOR DEVICES** button
- ☒ Advertising filter
- Device ID: 0x80E1260050...
- Target CPU: CPU1 : M4+
- Image base address (hex): 0x7000
- Image file path: BLE\_HeartRate\_ota.bin
- BROWSE** button
- UPDATE** button

图 23. 无线或 FUS 更新



The screenshot shows the 'OTA Updater' window with the following settings:

- SEARCH FOR DEVICES** button
- ☒ Advertising filter
- Device ID: 0x80E1260050...
- Target CPU: CPU2 : M0
- Image base address (hex): 0xF000
- Image file path: stm32wb5x\_BLE\_Stack\_fw.bin
- BROWSE** button
- UPDATE** button

## 5 结论

该文档说明如何使用低功耗蓝牙协议更新固件。

除了用户低功耗蓝牙应用程序，OTA 还依赖于 Flash 存储器中嵌入的专用应用程序。ST 提供的两个客户端（移动应用程序和 STM32CubeMonitor-RF 开发工具）之一发送低功耗蓝牙请求时，从当前用户应用程序调用 OTA 服务。

用户应用程序、无线固件和 FUS 可以独立更新。

## 6 版本历史

表 8. 文档版本历史

日期	版本	变更
2019 年 2 月 21 日	1	初始版本。
2019 年 6 月 07 日	2	更新 <a href="#">简介</a> ， <a href="#">用户应用程序</a> ， <a href="#">无线固件</a> ，以及 <a href="#">FUS 更新</a> ，第 4.2.2 节： <a href="#">用户应用程序更新</a> ，第 4.3 节： <a href="#">使用 STM32CubeMonitor-RF 进行固件更新</a> ，第 4.3.2 节： <a href="#">用户应用程序更新</a> ，以及第 5 节： <a href="#">结论</a> 。 更新了表 2： <a href="#">OTA 服务和特性声明</a> ，表 4： <a href="#">AD 结构 - 制造商特定字段</a> ，表 6： <a href="#">示例应用程序的服务和特性</a> ，以及表 7： <a href="#">示例（设备蓝牙 MAC 地址 = 80:E1:25:00:50:D6）</a> 。 更新了图 1： <a href="#">STM32WB 双核 FW 架构</a> ，图 13： <a href="#">适用项目</a> ，图 15： <a href="#">点对点服务器设备检测</a> ，图 17： <a href="#">ST BLE Sensor 固件更新面板和检测到的 OTA 服务</a> ，图 18： <a href="#">新应用程序下载</a> ，以及图 19： <a href="#">更新后的心率配置文件</a> 。 增加了第 3.4 节： <a href="#">无线和 FUS 更新流程</a> ，以及第 4.2.3 节： <a href="#">无线协议栈或 FUS 的更新</a> 。
2021 年 2 月 18 日	3	在整个文档中用 FUS 替换了 RSS。更新了第 1 节： <a href="#">词汇表</a> ，第 3.1.1 节： <a href="#">准则</a> ，以及第 4.1.2 节： <a href="#">OTA 应用程序</a> 。 更新了图 1： <a href="#">STM32WB 双核 FW 架构</a> ，图 4： <a href="#">简化的应用程序存储器映射</a> ，图 6： <a href="#">用户应用程序更新</a> ，图 7： <a href="#">无线固件更新</a> ，以及图 14： <a href="#">用户应用程序更新 - 用例</a> 。 对整个文档进行少量文字修订。
2022 年 5 月 25 日	4	更新了 3.1.5 节： <a href="#">无线固件更新</a> 。 对整个文档进行少量文字修订。
2022 年 8 月 8 日	5	更新了表 1： <a href="#">重启请求特性</a> 。 对整个文档进行少量文字修订。

表 9. 中文文档版本历史

日期	版本	变更
2023 年 3 月 26 日	1	中文初始版本。

**重要通知 - 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。ST 产品的销售依照订单确认时的相关 ST 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](http://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