

---

**BlueNRG、BlueNRG-MS OTA 自举程序**

---

**前言**

本应用笔记介绍使用意法半导体的 BlueNRG 和 BlueNRG-MS 开发套件时，运行于低功耗蓝牙（BLE）之上的 BlueNRG 和 BlueNRG-MS 无线（OTA）自举程序。首先介绍与 OTA 自举过程相关的一些概念，然后帮助用户了解运行一些自举会话所需的所有步骤。

**注意：**应用笔记内容适用于 BlueNRG 和 BlueNRG-MS 设备。任何对 BlueNRG 设备的参考文档也同时适用于 BlueNRG-MS 设备。必要时，会着重标明具体的区别。

## 目录

<b>1</b>	<b>OTA 自举的概念 .....</b>	<b>5</b>
<b>2</b>	<b>OTA 自举程序服务描述 .....</b>	<b>6</b>
2.1	自举事务 .....	6
2.2	BLE OTA 服务自举程序架构 .....	7
2.2.1	复位管理器 .....	8
2.2.2	具有 BLE OTA 服务的 BLE 应用 .....	9
2.2.3	新的 BLE 应用 .....	9
2.3	OTA 自举程序服务管理器（独立 OTA 自举程序应用） .....	10
2.3.1	复位管理器 .....	11
2.3.2	OTA 服务管理器 .....	11
2.3.3	新的 BLE 应用 .....	11
<b>3</b>	<b>硬件和软件资源 .....</b>	<b>13</b>
<b>4</b>	<b>运行 OTA 自举程序 GUI 和 BlueNRG 软件包演示应用 .....</b>	<b>14</b>
4.1	使用 OTA 自举程序服务构建 BLE 应用 .....	15
4.2	构建 BLE 应用以使用 BLE OTA 服务管理器 .....	19
<b>5</b>	<b>缩略语列表 .....</b>	<b>25</b>
<b>6</b>	<b>相关材料和参考 .....</b>	<b>26</b>
<b>7</b>	<b>版本历史 .....</b>	<b>27</b>

## 表格列表

表 1: BlueNRG 与 BlueNRG-MS 平台的命名规则 .....	5
表 2: 使用 BLE OTA 自举程序演示低地址存储应用程序时候的 LED 显示状态.....	9
表 3: 使用 BLE OTA 自举程序演示高地址存储应用程序时候的 LED 显示状态.....	9
表 4: BlueNRG OTA 服务管理时候的 LED 显示状态 .....	11
表 5: 使用 BLE OTA 服务管理器演示应用程序时候的 LED 显示状态 .....	12
表 6: BlueNRG OTA 自举程序硬件和软件资源 .....	13
表 7: 具有 OTA 自举程序功能的 Sensor Demo 的 IAR 工作区 .....	14
表 8: 具有 OTA 自举程序功能的 BLE_Chat 的 IAR 工作区 .....	15
表 9: 本文档中使用的缩略语 .....	25
表 10: 相关文档 .....	26
表 11: 文档版本历史 .....	27
表 12: 中文文档版本历史 .....	27

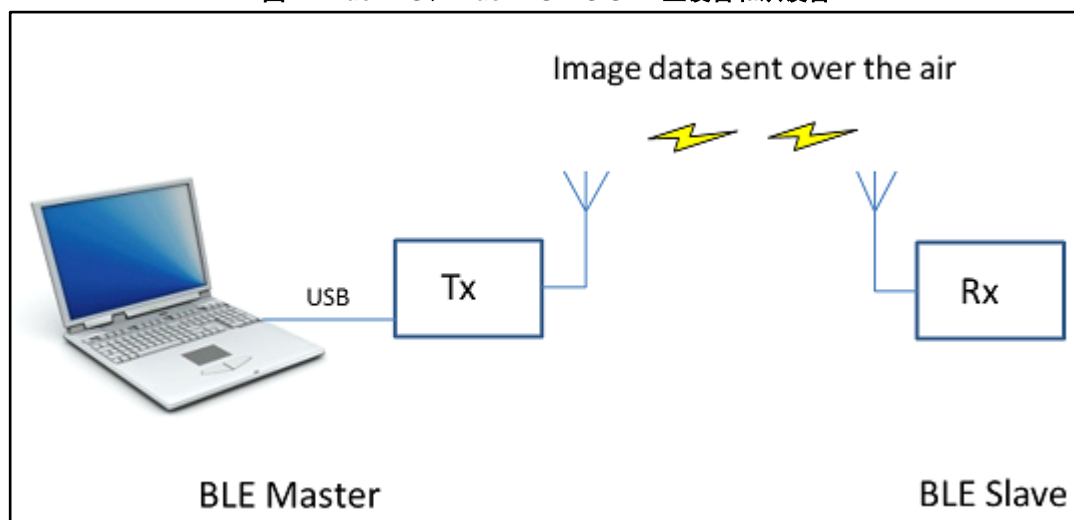
## 图列表

图 1: BlueNRG、BlueNRG-MS OTA 主设备和从设备 .....	5
图 2: BLE OTA 服务自举程序 .....	7
图 3: BLE OTA 服务自举程序 Flash 布局 .....	8
图 4: BLE OTA 服务管理器 Flash 布局 .....	10
图 5: BlueNRG GUI: OTA 自举程序窗口 .....	16
图 6: BlueNRG GUI: OTA 自举程序搜索设备并连接 .....	17
图 7: BlueNRG GUI: OTA 自举程序升级 .....	18
图 8: BlueNRG GUI: OTA 自举程序升级完成 .....	19
图 9: BlueNRG GUI: OTA 自举程序窗口 .....	21
图 10: BlueNRG GUI: OTA 自举程序搜索设备并连接 .....	22
图 11: BlueNRG GUI: OTA 自举程序升级 .....	23
图 12: BlueNRG GUI: OTA 自举程序升级完成 .....	24

## 1 OTA 自举的概念

OTA 自举程序本质上是设备无线接收来自主设备的固件映像，并将其写入 Flash 存储器。就低功耗蓝牙而言，OTA 自举程序是一种提供其自己的特性，并能与 BLE 栈上运行的任何给定应用使用的其他服务共存的服务。本文实现方式是将 BlueNRG 或者 BlueNRG-MS 平台套件当做主设备通过 USB 连接到 PC。BlueNRG 或 BlueNRG-MS 平台通过所连接 PC 上的 BlueNRG GUI 驱动。这种选择允许用户使用 PC 上的大量可用资源，特别是对用于固件映像生成的编译器和在发送映像进行自举之前保存映像所需的存储器空间而言。

图 1: BlueNRG、BlueNRG-MS OTA 主设备和从设备



**注意：**关于 BlueNRG 和 BlueNRG-MS 套件平台的更多信息，请参考 ST BlueNRG 网页上的相关用户手册（参见 [第6节：“相关材料 and 参考”](#)）。

**表 1: “BlueNRG 与 BlueNRG-MS 平台的命名规则”** 描述了 BlueNRG 与 BlueNRG-MS 平台 LED 和按钮使用的命名规则。

表 1: BlueNRG 与 BlueNRG-MS 平台的命名规则

资源名称	BlueNRG、BlueNRG-MS开发平台	BlueNRG、BlueNRG-MS USB Dongles
LED_D1	DL1（黄色LED）	D2（红色LED）
LED_D2	DL2（橙色LED）	D3（橙色LED）
BUTTON_1	Push_Button	SW1
BUTTON_2	Joystick SEL	SW2

## 2 OTA 自举程序服务描述

OTA 自举程序服务通过 BlueNRG DK 软件中提供的文件 btl.[ch] 寻址，该软件支持 BlueNRG 和 BlueNRG-MS 设备。

下面将提供 OTA 自举程序服务和相关特性的简要描述：

- 自举程序 OTA 服务：自举程序服务
  - aci\_gatt\_add\_serv(UUID\_TYPE\_128, UUID, PRIMARY\_SERVICE, 10, &btlServHandle)
- Btl 映像的特征值：包含与空闲内存范围相关的信息，这是包含 OTA 自举程序服务的当前应用建议的范围。
  - aci\_gatt\_add\_char(btlServHandle, UUID\_TYPE\_128, UUID, 8, CHAR\_PROP\_READ, ATTR\_PERMISSION\_NONE, GATT\_INTIMATE\_APPL\_WHEN\_READ\_N\_WAIT, 16, 0, &btlImageCharHandle)
- Btl 新映像的特征值：包含主设备想要无线发送的映像的基址和大小。
  - aci\_gatt\_add\_char(btlServHandle, UUID\_TYPE\_128, UUID, 9, CHAR\_PROP\_READ|CHAR\_PROP\_WRITE|CHAR\_PROP\_WRITE\_WITHOUT\_RESP, ATTR\_PERMISSION\_NONE, GATT\_INTIMATE\_APPL\_WHEN\_READ\_N\_WAIT | GATT\_SERVER\_ATTR\_WRITE, 16, 0, &btlNewImageCharHandle)
- Btl 新映像内容的特性：包含主设备发送（通过特性写入指令）的 16 字节固件映像数据块，以及一些控制信息，例如块序号（2 字节）和用于完整性检查的校验和（1 字节）。
  - aci\_gatt\_add\_char(btlServHandle, UUID\_TYPE\_128, UUID, 20, CHAR\_PROP\_READ|CHAR\_PROP\_WRITE|CHAR\_PROP\_WRITE\_WITHOUT\_RESP, ATTR\_PERMISSION\_NONE, GATT\_INTIMATE\_APPL\_WHEN\_READ\_N\_WAIT | GATT\_SERVER\_ATTR\_WRITE, 16, 0, &btlNewImageTUContentCharHandle)
- Btl 预期映像序号特性，从设备通过该特性通知主设备其预期的下一个数据块。
  - aci\_gatt\_add\_char(btlServHandle, UUID\_TYPE\_128, UUID, 4, CHAR\_PROP\_NOTIFY|CHAR\_PROP\_READ, ATTR\_PERMISSION\_NONE, GATT\_INTIMATE\_APPL\_WHEN\_READ\_N\_WAIT, 16, 0, &btlExpectedImageTUSeqNumberCharHandle)

**注意：**在 btl.c 文件中定义 OTA 自举程序服务和外设 128 位 UUID 特性。

### 2.1 自举事务

本节介绍 OTA 自举的步骤：

1. 在安装了运行 OTA 自举程序服务的主设备和从设备后，需执行搜索程序以便连接两个设备。通过收听无线传输范围内的设备的广播进行搜索（主动扫描），并选择扫描响应中包含 OTA BTL 服务 UUID 128 位的设备。
2. 将从广播消息中读取所选设备的名称，主设备将使用该名称强化从设备识别程序。
3. 连接后，主设备将发送“GATT DISC CHARAC BY UUID”指令，以便读取所有 OTA 自举程序特性。
4. 主设备将通过“GATT READ CHARACTERISTIC VALUE”指令读取 Btl 映像特性，以获悉目标从设备 Flash 存储器上的空闲空间。
5. 根据上一步中获得的信息，主设备将选择合适的映像进行无线发送。候选映像（位于主设备上某位置的\*.bin 文件）必须处于目标的空闲 Flash 存储范围内，并且必须是 512 字节对齐（与 STM32L 中断向量限制要求的一致）。完成选择后，主设备发送“GATT

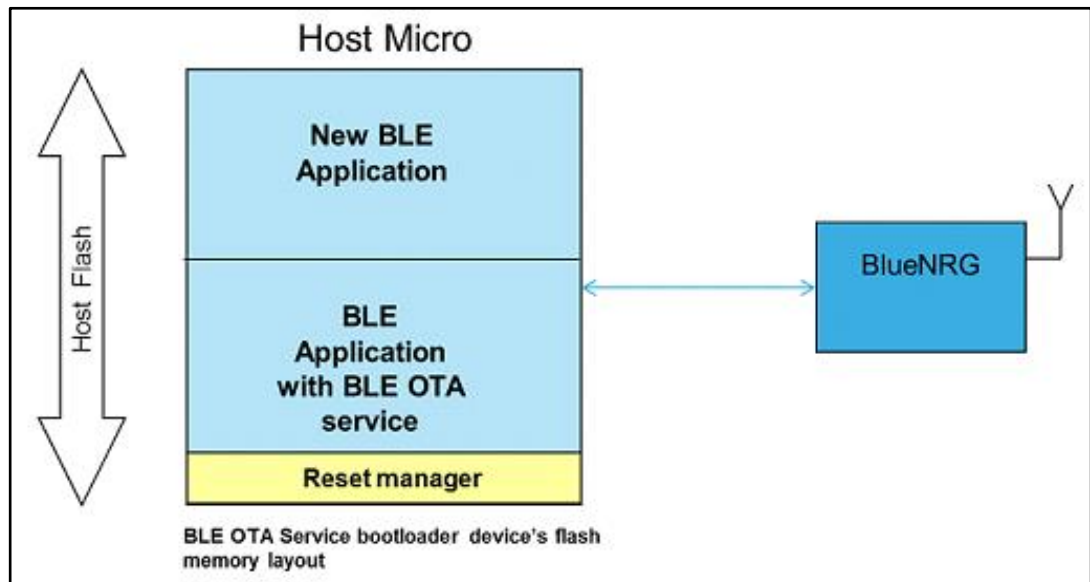
- WRITE CHARACTERISTIC VALUE”指令，以将映像基址和大小写入 Btl 新映像特性，并通过“GATT READ CHARACTERISTIC VALUE”读回以进行确认。
6. 主设备写入 Btl 预期映像序号特性描述符，以启用映像块序号和错误的从设备通知。一旦从设备接收到该指令，它还将擦除主设备写入 Btl 新映像特性中的 Flash 存储范围内的页面，并发回通知。
  7. 映像传输开始。主设备将通过一系列“GATT WRITE WITHOUT RESPONSE”指令以 16 字节数据块发送映像（每个 16 字节数据块一个）。目标从设备每次收到新的 WRITE 指令时，将新数据块写入 Btl 新映像内容特性。每个 16 字节数据块附带 2 字节序号和 1 字节校验和字段，用于在目标位置检查顺序和消息完整性。每次 16 字节数据块占满从设备的内部缓冲区时，数据块被下载到 Flash 存储器。一旦从设备完成当前 16 字节数据块的管理，它将向主设备发送通知消息，其中包含下一个预期数据块的数据块号。它可能通知 Flash 写入错误和 Flash 验证错误，假如问题出在目标设备 Flash 上，此时自举会话应停止。
  8. 如果成功下载到目标设备 Flash 上的字节数与最初提供的映像大小信息相符，自举程序将新映像的基址和结束地址写入 EEPROM 存储器的最后两个位置（地址 0x08080FF8），并向包含新应用固件的位置转移控制。

## 2.2 BLE OTA 服务自举程序架构

BLE OTA 服务自举程序架构包括以下组成部分：

- Reset Manager
  - 应在复位后启动，并向最近更新并有效的 BLE 应用程序转移控制。
  - “BLE application with BLE OTA service”。这是运行的 BLE 应用，允许无线接收新的应用映像数据包并将其保存在特定 Flash 扇区。
- New BLE Application
  - 这是通过 RF 链接下载的应用映像。
  - 为了充分利用 BLE OTA 服务方法，可以为新的 BLE 应用构建 BLE 服务，使之具备 OTA 服务自举程序功能。

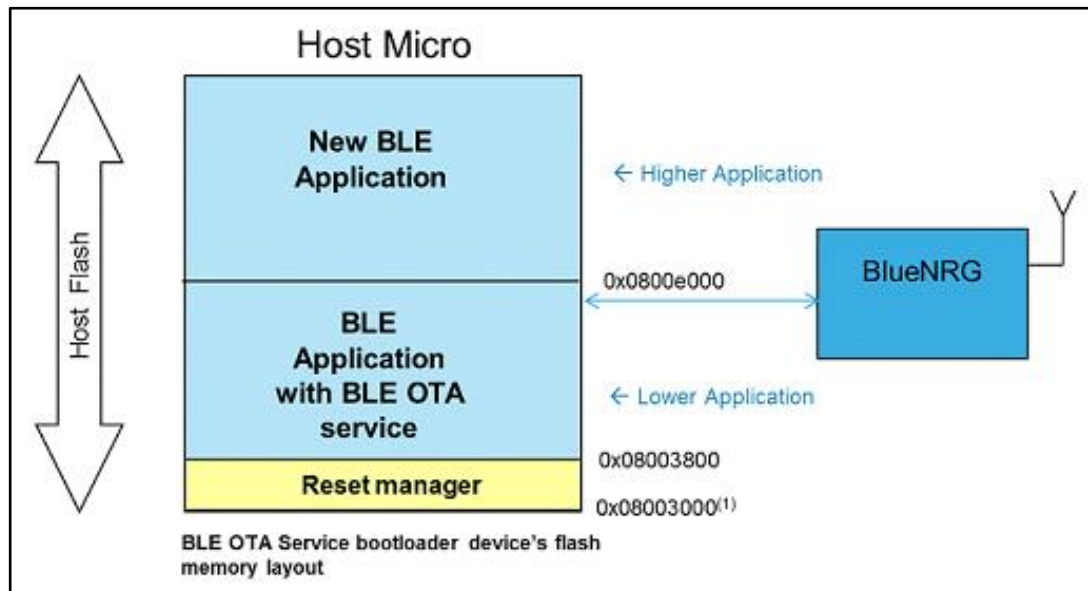
图 2: BLE OTA 服务自举程序



### 2.2.1 复位管理器

STM32L1 的 Flash 基址从 0x8000000 开始，意味着在设备复位时，微控制器将开始执行该范围内的应用映像。所有 BlueNRG 和 BlueNRG-MS 套件平台都有预编程的 DFU 应用，可以将二进制映像下载到 STM32L 的内部 Flash。当 OTA 自举程序从大于“Flash 基址 + DFU 大小 (0x3000)”的基址开始写入新映像时，需要一种可以在每次复位时向新应用转移控制的方法。为此，首先需要为计划包含 OTA 自举程序服务的每个设备加载复位管理器（从“Flash 基址 + DFU 大小”开始）。在设备复位时，复位管理器将负责跳转至 OTA 自举程序成功加载的最后一个映像的位置，如设备 EEPROM 存储范围内的较高 8 字节所示（从地址 0x08080FF8 开始）。图 3: “BLE OTA 服务自举程序 Flash 布局”总结了以 STM32L 为主机微控制器时，具有板载 OTA 自举程序服务的设备的典型 Flash 内存布局。在最初设置设备时，我们将使用 BlueNRG GUI 将 Flash 中的应用上传到 STM32L 的地址 0x8003800（复位管理器上端），然后以相同方式将复位管理器本身上传到地址 0x8003000。为了正确完成复位管理器的初步设置，必须采取措施擦除相关 EEPROM 位置。在 BlueNRG 和 BlueNRG-MS 套件平台上，可在按住复位按钮的同时按下 BUTTON\_2（LED\_D1 将点亮 5 秒）。这会擦除 EEPROM 位置，即下一次按下复位按钮后，复位管理器读取的内容“全部为 0”。因此，在默认情况下，复位管理器假设必须跳转至 0x8003800，这正是最初设置的运行应用的参考基址。

图 3: BLE OTA 服务自举程序 Flash 布局



(1) 假设 DFU.hex 应用位于 Flash 基址 0x08000000。

**注意：**BlueNRG DK 软件包的 Project 文件夹中提供了 OTA\_ResetManager IAR 项目及相关头文件和源文件。此外，Project/OTA\_prebuilt\_images 文件夹中提供了预建映像。此 IAR 项目对 BlueNRG 和 BlueNRG-MS 设备有效。



## 2.2.2 具有 BLE OTA 服务的 BLE 应用

为了添加 btl.c 文件中提供的 BLE OTA 服务，请执行以下步骤：

1. 在 EWARM 工作区，添加 ST\_OTA\_BTL 预处理器和链接器选项；
2. 在 EWARM 工作区，使用参考性的 stm32l1xx\_flash.icf 链接器文件，以与 [图 3: "BLE OTA 服务自举程序 Flash 布局"](#)中描述的 Flash 布局保持一致（在 BlueNRG 软件包的 Projects\Project 文件夹中提供）；
3. 在 EWARM 工作区，添加对 btl.c 文件的参考，并添加 btl.h 文件的路径；
4. 在 BLE 用户应用中，包括头文件 btl.h；
5. 在 BLE 用户应用中，调用 btl.c 函数 Add\_Btl\_Service()，以便添加 OTA BLE 服务和相关特性；
6. 在 BLE 用户应用中，为扫描响应添加 OTA 服务 UUID：  
hci\_le\_set\_scan\_resp\_data(18,BTLServiceUUID4Scan)
7. 在 BLE 用户应用、HCI 事件处理器（HCI\_Event\_CB）、  
EVT\_BLUE\_GATT\_ATTRIBUTE\_MODIFIED 事件中，添加对 btl.c 文件中定义的  
OTA\_Write\_Request\_CB() API 的调用；
8. 在 BLE 用户应用、HCI 事件处理器（HCI\_Event\_CB）中，添加  
EVT\_BLUE\_GATT\_READ\_PERMIT\_REQ 事件用例和对 aci\_gatt\_allow\_read() API 的  
调用（以便允许 BLE 栈发送响应）。

**注意：**为便于识别具有 OTA 自举程序服务的 IAR BlueNRG 演示应用，根据 [表 2: "使用 BLE OTA 自举程序演示低地址存储应用程序时候的 LED 显示状态"](#)和 [表 3: "使用 BLE OTA 自举程序演示高地址存储应用程序时候的 LED 显示状态"](#)中的方案使用平台 LED。

表 2: 使用 BLE OTA 自举程序演示低地址存储应用程序时候的 LED 显示状态

BlueNRG、BlueNRG-MS开发平台（具有OTA自举程序服务的低层应用）	BlueNRG、BlueNRG-MS USB Dongles（具有OTA自举程序服务的低层应用）
DL1（黄色）一直闪烁 <sup>(1)</sup>	D2（红色）一直闪烁 <sup>(1)</sup>
DL2（橙色）在OTA升级期间发光	D3（橙色）在OTA升级期间发光

**Notes:**

<sup>(1)</sup>当连接了设备时，LED 闪烁频率更高。

表 3: 使用 BLE OTA 自举程序演示高地址存储应用程序时候的 LED 显示状态

BlueNRG、BlueNRG-MS开发平台（具有OTA自举程序服务的高层应用）	BlueNRG、BlueNRG-MS USB Dongles（具有OTA自举程序服务的高层应用）
DL2（橙色）一直闪烁 <sup>(1)</sup>	D3（橙色）一直闪烁 <sup>(1)</sup>
DL1（黄色）在OTA升级期间发光	D2（红色）在OTA升级期间发光

**Notes:**

<sup>(1)</sup>当连接了设备时，LED 闪烁频率更高。

## 2.2.3 新的 BLE 应用

为了支持 BLE OTA 自举程序服务，可以构建新的 BLE 应用，以便提供 OTA 自举功能（超出其使用其他 BLE 服务和相关特性的特定 BLE 应用场景）。用户可以按照 [第2.2.2 节: "具有 BLE:OTA 服务的 BLE 应用"](#)中描述的相同步骤操作。



如果构建的 BLE 应用将位于较高 Flash 区（参见图 3: "BLE OTA 服务自举程序 Flash 布局" 中的高层应用），必须使用 HIGHER\_APP 预处理器和链接器选项。具有 OTA 自举程序服务支持的 BlueNRG 演示应用是为将 0x0800e000 用作高层应用的参考基址而量身定制的。在 btl.h 文件中定义相关的向量表起始地址偏移 0xe000，如下：

```
VECTOR_TABLE_BASE_DFU_OFFSET +  
VECTOR_TABLE_BASE_HIGHER_APP_OFFSET
```

其中：

VECTOR\_TABLE\_BASE\_DFU\_OFFSET = 0x3000（DFU 大小）。

VECTOR\_TABLE\_BASE\_HIGHER\_APP\_OFFSET = 0xB000（用户定义的位置）。

用户可通过修改 btl.h 文件中的链接器文件和起始地址

VECTOR\_TABLE\_BASE\_HIGHER\_APP\_OFFSET 进行自定义。

**注意：**如果没有使用 HIGHER\_APP 预处理器和链接器选项，在默认情况下，构建的 BLE 应用将位于默认起始地址 0x08003800。

在 btl.h 文件中定义相关的向量表起始地址偏移 0x3800，如下：

```
VECTOR_TABLE_BASE_DFU_OFFSET +  
VECTOR_TABLE_BASE_RESET_MANAGER_OFFSET
```

其中：

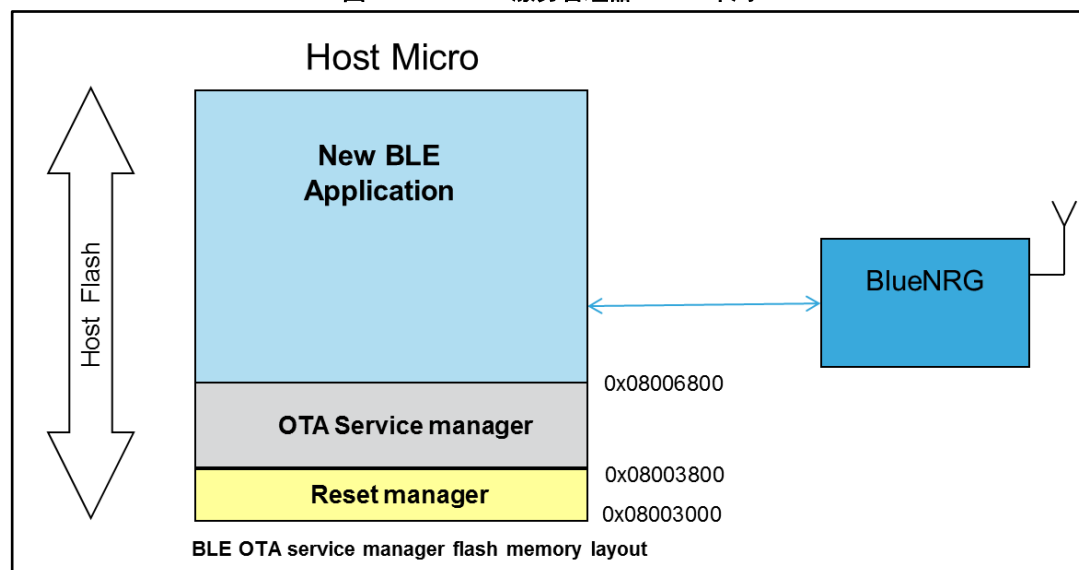
VECTOR\_TABLE\_BASE\_DFU\_OFFSET = 0x3000（DFU 大小）。

VECTOR\_TABLE\_BASE\_RESET\_MANAGER\_OFFSET = 0x800（复位管理器大小）。

## 2.3 OTA 自举程序服务管理器（独立 OTA 自举程序应用）

图 2: "BLE OTA 服务自举程序"中描述的 BLE OTA 服务架构提供了一种更简单的方法，需使用基础 OTA 服务管理器应用，其中仅包括 BLE OTA 服务和特性。

图 4: BLE OTA 服务管理器 Flash 布局



(1) 假设 DFU.hex 应用位于 Flash 基址 0x8000000。

BLE OTA 服务管理器架构包括以下组成部分：

- 复位管理器
  - 应在复位后启动，并向最近更新并有效的 BLE 应用转移控制。
- OTA 服务管理器
  - 仅包含 BLE OTA 服务和特性，向任何 BLE 应用提供 OTA 自举程序功能。
- 新的 BLE 应用
  - 这是通过 OTA 服务管理器下载的应用映像。
  - 它不要求包括 OTA 自举程序服务。
  - 为了使用此功能，BLE 应用必须激活 OTA 服务管理器（btl.c 文件中提供了特定的 API Switch\_To\_OTA\_Service\_Manager\_Application() 函数）。
  - 它始终位于固定的 Flash 地址：0x8006800。

### 2.3.1 复位管理器

请参见 [第2.2.1 节：“复位管理器”](#)。

### 2.3.2 OTA 服务管理器

OTA 服务管理器相当于独立的 OTA 自举程序，为任何需要使用自举程序功能的 BLE 应用提供 BLE OTA 服务，因此这些应用无需包括任何 OTA 功能。

项目文件夹中的 BlueNRG 软件包提供了 OTA 服务管理器应用 IAR 项目 + 头文件和源文件，如下所示：

1. OTA\_ServiceManager\EWARM\OTA\_ServiceManager.eww，用于 BlueNRG 设备。
2. OTA\_ServiceManager\EWARM\_BlueNRG-MS\OTA\_ServiceManager.eww，用于 BlueNRG-MS 设备。

OTA 服务管理器应用使用如下平台 LED：

表 4: BlueNRG OTA 服务管理时候的 LED 显示状态

BlueNRG、BlueNRG-MS开发平台	BlueNRG、BlueNRG-MS USB Dongles
DL1（黄色）一直闪烁 <sup>(1)</sup>	D2（红色）一直闪烁 <sup>(1)</sup>
DL2（橙色）在OTA升级期间发光	D3（橙色）在OTA升级期间发光

**Notes:**

<sup>(1)</sup>当连接了设备时，LED 闪烁频率更高。

### 2.3.3 新的 BLE 应用

对于需要使用 OTA 服务管理器功能的 BLE 应用，只需执行以下基本步骤：

1. 在 EWARM 工作区，添加 ST\_OTA\_BASIC\_APPLICATION 预处理器和链接器选项；
2. 在 EWARM 工作区，使用参考性的 stm32l1xx\_flash.icf 链接器文件，以与 [图 4: “BLE OTA 服务管理器 Flash 布局”](#)中描述的 Flash 布局保持一致（在 BlueNRG 软件包的演示应用文件夹中提供）；
3. 在 EWARM 工作区，添加对 btl.c 文件的参考，并添加 btl.h 文件的路径；
4. 在 BLE 用户应用中，包括头文件 btl.h；

5. 添加一些用户代码，以允许 BLE 应用启动 OTA 服务管理器应用。可以使用简单的按钮控制对 Switch\_To\_OTA\_Service\_Manager\_Application (btl.c 文件中提供了该 API) 的调用。

**注意：**在使用 OTA 服务管理器时，BLE 应用无需包含任何 OTA 自举程序服务。另外，应用始终可以位于固定的基址 0x08006200。在 btl.h 文件中定义相关的向量表起始地址偏移 0x6800，如下：

```
VECTOR_TABLE_BASE_DFU_OFFSET +  
VECTOR_TABLE_BASE_RESET_MANAGER_OFFSET +  
VECTOR_TABLE_BASE_ADDRESS_OTA_BASIC_APP
```

其中：

VECTOR\_TABLE\_BASE\_DFU\_OFFSET = 0x3000 (DFU 大小)。

VECTOR\_TABLE\_BASE\_RESET\_MANAGER\_OFFSET = 0x800 (复位管理器大小)。

VECTOR\_TABLE\_BASE\_ADDRESS\_OTA\_BASIC\_APP = 0x3000 (OTA 服务管理器大小)。

**注意：**为便于识别使用 OTA 服务管理器的 BlueNRG 演示应用，根据表 5: "使用 BLE OTA 服务管理器演示应用程序时候的 LED 显示状态"中的方案使用平台 LED。

表 5: 使用 BLE OTA 服务管理器演示应用程序时候的 LED 显示状态

BlueNRG、BlueNRG-MS开发平台	BlueNRG、BlueNRG-MS USB Dongles
DL2 (橙色) 一直闪烁 <sup>(1)</sup>	D3 (橙色) 一直闪烁 <sup>(1)</sup>
DL1 (黄色) 在OTA升级期间发光	D2 (红色) 在OTA升级期间发光

**Notes:**

<sup>(1)</sup>当连接了设备时，LED 闪烁频率更高。

3 硬件和软件资源

在说明如何设置 OTA 自举程序演示之前，我们将首先提供所需硬件和软件资源的清单：

表 6: BlueNRG OTA 自举程序硬件和软件资源

状态	说明
强制	2个BlueNRG或BlueNRG-MS套件平台和相关USB线（仅开发平台需要）。
	包含OTA自举程序服务的复位管理器映像和预建应用映像
	BlueNRG图形用户界面v1.4.0或更高版本
	具有以下资源的PC： Windows® XP SP3或Windows® Vista或Windows® 7 至少28 MB RAM 2个USB端口 40 MB硬盘空间 Adobe Reader 6.0或更高版本
可选 但 推荐	IAR embedded workbench 6.60。用于访问代码，以便学习提供的应用样例。
	STLink Flash编程工具
	STLink实用工具：用于执行对使用STLink Flash编程工具测试和验证自举会话有用的Flash管理操作。

4 运行 OTA 自举程序 GUI 和 BlueNRG 软件包演示应用

寻址 BlueNRG 软件包演示应用 Sensor Demo 和 BLE Chat 的 IAR 项目具备特定的预配置工作区，允许使用以下两种方法：

- 1. 构建 BLE 应用以包含 OTA 自举程序服务
- 2. 使用 BLE OTA 服务管理器构建 BLE 应用

对于 Sensor Demo 和 BLE Chat 演示应用，分别可使用 EWARM 和 EWARM\_BlueNRG-MS 文件夹下的两个 IAR 项目进行 BlueNRG 和 BlueNRG-MS 设备的寻址。

表 7: "具有 OTA 自举程序功能的 Sensor Demo 的 IAR 工作区" 对具有 OTA 自举程序功能的 Sensor Demo BLE 应用可用 IAR 工作区进行了简单描述：

表 7: 具有 OTA 自举程序功能的 Sensor Demo 的 IAR 工作区

Sensor Demo应用的IAR工作区	说明
LowPower_LowerApp_OTA	包含采用“低层应用”内存布局的OTA自举程序服务的低功耗配置（参考图 3: "BLE OTA服务自举程序Flash布局"和第2.2.2节"具有BLE:OTA服务的BLE应用"及第2.2.3节"新的BLE应用"）
LowPower_HigherApp_OTA	包含采用“高层应用”内存布局的OTA自举程序服务的低功耗配置（参考图 3: "BLE OTA服务自举程序Flash布局"和第2.2.2节"具有BLE:OTA服务的BLE应用"及第2.2.3节"新的BLE应用"）
USB_LowerApp_OTA	包含采用“低层应用”内存布局的OTA自举程序服务的USB配置（参考图 3: "BLE OTA服务自举程序Flash布局"和第2.2.2节"具有BLE:OTA服务的BLE应用"及第2.2.3节"新的BLE应用"）
USB_HigherApp_OTA	包含采用“高层应用”内存布局的OTA自举程序服务的USB配置（参考图 3: "BLE OTA服务自举程序Flash布局"和第2.2.2节"具有BLE:OTA服务的BLE应用"及第2.2.3节"新的BLE应用"）
OTA_Basic	使用OTA服务管理器的低功耗配置（参考图 4: "BLE OTA服务管理器Flash布局"和第2.3.2节"OTA服务管理器"及第2.3.3节"新的BLE应用"）

下面是对具有 OTA 功能的 BLE Chat 应用可用 IAR 工作区的简单描述：



表 8: 具有 OTA 自举程序功能的 BLE\_Chat 的 IAR 工作区

BLE_Chat应用的IAR工作区	说明
Server_LowerApp_OTA	包含采用“低层应用”内存布局的OTA自举程序服务的服务器配置（参考图 3: “BLE OTA服务自举程序Flash布局”和 第2.2.2节“具有BLE:OTA服务的BLE应用”及 第2.2.3节“新的BLE应用”）
Server_HigherApp_OTA	包含采用“高层应用”内存布局的OTA自举程序服务的服务器配置（参考图 3: “BLE OTA服务自举程序Flash布局”和 第2.2.2节“具有BLE:OTA服务的BLE应用”及 第2.2.3节“新的BLE应用”）
Server_OTA_Basic	服务配置使用OTA服务管理器（参考图 4: “BLE OTA服务管理器Flash布局”以及 第3.2节OTA服务管理器和 第3.3节新的BLE应用）

**注意：**仅 BlueNRG 和 BlueNRG-MS 开发平台支持 Sensor Demo 应用。

### 4.1 使用 OTA 自举程序服务构建 BLE 应用

下面是对使用 Sensor Demo 应用（具有 OTA 服务支持）和两个开发平台运行 BlueNRG GUI OTA 自举程序的分步描述：

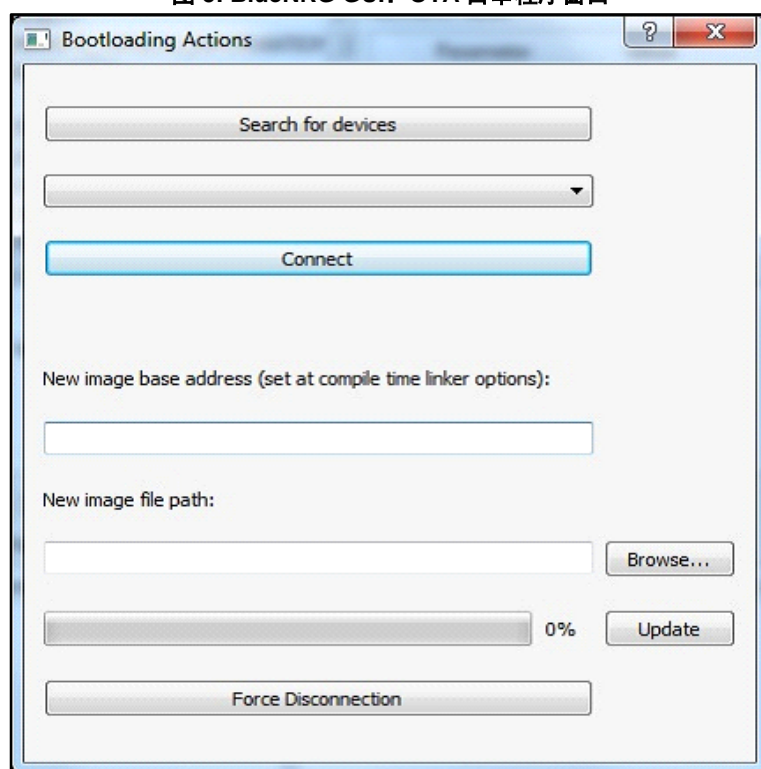
- 1 将两个开发平台插入 PC 上的 USB 连接器。
- 2 使用 BlueNRG GUI 对将用作 OTA 自举程序从设备的两个平台套件之一进行编程。这需要将板插入运行 GUI 的 PC 上的 USB 连接器，然后在按下 BUTTON\_1 的同时按下平台复位按钮，以便启动 DFU 自举程序（此时 LED\_D2 应闪烁）。
  - a. 从“Tools”->“Flash Motherboard FW”中选择复位管理器 hex 预建映像（STM32L\Project\OTA\_prebuilt\_images\DFU\_ResetManager.hex），以将复位管理器上传到板上（hex 格式考虑了合适的应用基址，就该应用而言，为 0x8003000）。
  - b. 必须采用相同操作（从 DFU 自举程序激活开始），将初始 OTA 自举程序应用上传到板上（对于 BlueNRG 设备，为 STM32L\Project\OTA\_prebuilt\_images\BlueNRG\_DFU\_OTA\_Lower\_SensorDemo.hex；对于 BlueNRG-MS 设备，为 BlueNRG-MS\_DFU\_OTA\_Lower\_SensorDemo.hex）；（hex 格式考虑了合适的应用基址，就该应用而言，为 0x8003800）。
  - c. 加载复位管理器后，需在按下 BUTTON\_2 的同时执行复位，以擦除 OTA 自举程序 EEPROM 中的原有数据，并引导最初启动时的合适复位管理器行为（LED\_D1 发光 5 秒表示过程运行成功）。
  - d. （可选）获取“BlueNRG app for smartphones (IoT or Android)”，以发现、连接并使用刚才在步骤 2.b 中加载到平台上的 Sensor Demo 应用。只需确认普通 BLE 应用的服务与下一步要运行的 OTA 自举程序服务共存。
  - e. （可选）如果执行了步骤 2.d 中的操作，则从开发套件平台上断开启了“BlueNRG app for smart phones (IoT or Android)”的设备。Sensor Demo 应用将被断开并开始广播。这将使该应用成为通过 BlueNRG GUI 进行 BLE OTA 升



级的潜在候选应用。

- 3 使用 BlueNRG GUI，在第二个平台上对要使用的必要应用进行编程（在 BlueNRG DK 软件包的 Firmware\STM32L1\_prebuilt\_images 文件夹中可以找到 BlueNRG\_VCOM\_x\_x.hex 文件）。
- 4 在 PC 上打开 BlueNRG GUI，选择与步骤 3 中配置的 BlueNRG 平台相关的 COM 端口（在下拉列表中选择“Port”并单击“Open”）。
- 5 选择“Tools”->“OTA bootloader”以打开包含 OTA 自举程序操作的对话框，并单击“Search for devices”。

图 5: BlueNRG GUI: OTA 自举程序窗口

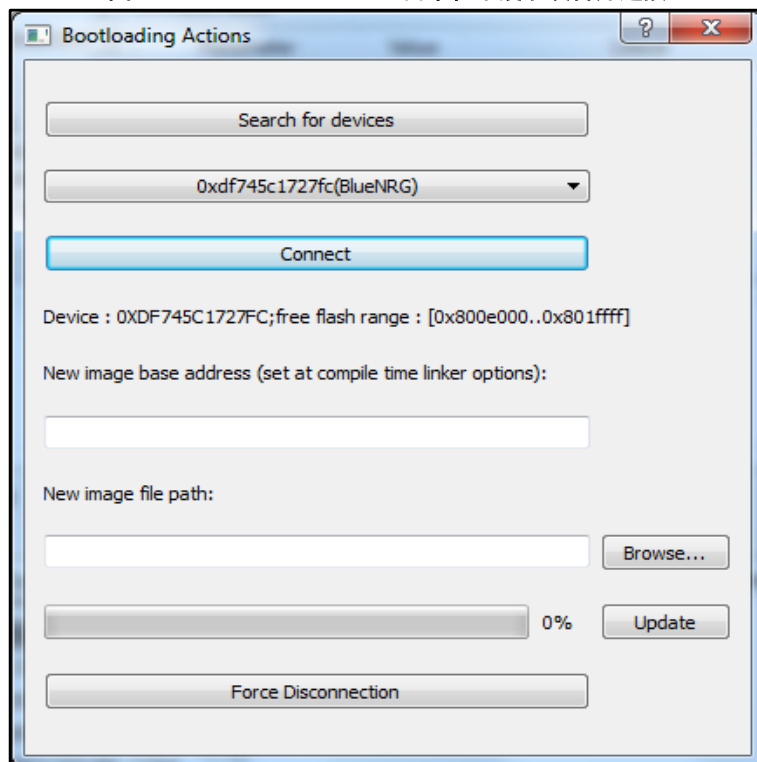


- 6 在单击“Search for devices”后，GUI 将启动搜索过程，并返回无线传输范围内运行 OTA 自举程序服务的设备的地址和应用名称相关信息。一旦前面的过程结束，则可以通过“Search for devices”按钮下方的组合框箭头打开设备列表，此时用户可以自由选择要连接的设备，然后使用“Connect”按钮执行自举程序过程（显示应用地址和名称）。如果设备地址和运行的应用不足以识别正确设备，建议在运行 OTA 自举程序服务的从设备上使用连接激活指示灯。例如，我们在演示平台上使用一个 LED，使之在连接建立后以更高的频率闪烁，这可以用作获得与被寻址设备相关的确认的视觉反馈。如果发现连接了错误的设备，只需单击“Force Disconnection”按钮，并在组合框内重新选择设备。在选择设备并单击“Connect”按钮进行连接后，可以读取其广播的相关空闲内存范围，它限制我们在从设备上提供以预期范围内的基址和大小编译的映像文件。
  - a. 下一步，我们需要提供“Base address”，并单击“browse”按钮以选择文件的位置，该文件包含我们计划无线发送至从设备的映像。因此，用户可以打开 SensorDemo 应用的 BlueNRG IAR 工作区 LowPower\_HigherApp\_OTA，并



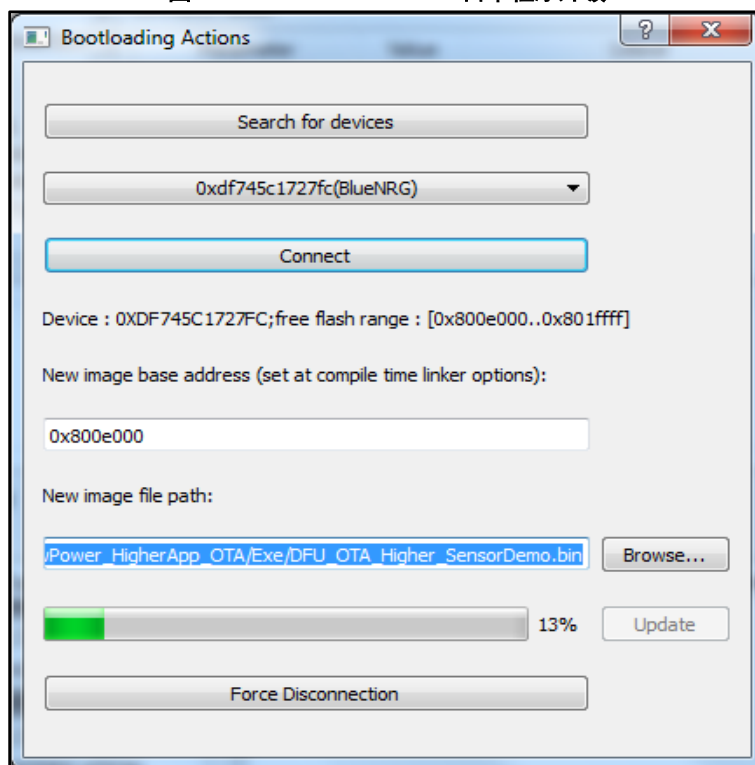
构建相关的 DFU\_OTA\_Higher\_SensorDemo.bin 映像（位于 SensorDemo\EWARM\LowPower\_HigherApp\_OTA\Exe 文件夹），该映像包含使用基址 0x800e000 编译的 Sensor Demo 版本。除了内存偏移，与 LowPower\_LowerApp\_OTA 的唯一区别在于闪烁的 LED 的颜色（LowerApp 为 LED\_D1 闪烁，Higher\_App 为 LED\_D2 闪烁）。或者，用户可以打开 SensorDemo 应用的 BlueNRG-MS IAR 工作区 LowPower\_HigherApp\_OTA，并构建相关的 DFU\_OTA\_Higher\_SensorDemo.bin 映像（位于 SensorDemo\EWARM\_BlueNRG-MS\LowPower\_HigherApp\_OTA\Exe 文件夹）。

图 6: BlueNRG GUI: OTA 自举程序搜索设备并连接



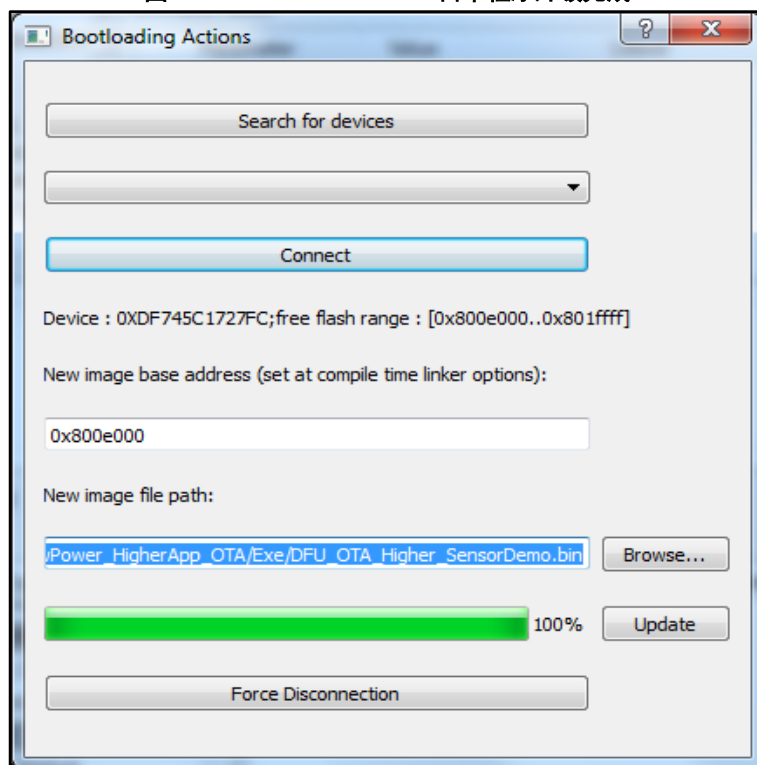
- 7 在浏览到使用正确基址（0x800e000）编译的应用映像（SensorDemo\EWARM\LowPower\_HigherApp\_OTA\Exe\DFU\_OTA\_Higher\_SensorDemo.bin 或 SensorDemo\EWARM\_BlueNRG-MS\LowPower\_HigherApp\_OTA\Exe\DFU\_OTA\_Higher\_SensorDemo.bin）后，可以单击“Update”按钮启动 OTA 自举过程。进度条将指示完成这一过程需要的时间。

图 7: BlueNRG GUI: OTA 自举程序升级



- 8 当进度条达到 100%时，过程完成，如下图所示。

图 8: BlueNRG GUI: OTA 自举程序升级完成



如果我们观察从板，会发现闪烁的 LED 的颜色变化，这表明新的应用正在运行。可以使用工具（例如，STM32 STLink 实用工具），对照选定二进制文件中的映像检查升级的设备上下载的 Flash 存储区。从主设备断开后，可以按照步骤 2 中的可选条目，对照“BlueNRG app for smartphones (iOS or Android)”重新检查 Sensor Demo 新版本的功能。

**注意：**可以对 BlueNRG BLE\_Chat 演示应用使用类似程序，以便验证 BLE OTA 自举程序服务功能（对于 BlueNRG 设备，请参考 BLE\_Chat\EWARM\BLE\_Chat.eww、Server\_LowerApp\_OTA 和 Server\_HigherApp\_OTA 工作区；对于 BlueNRG-MS 设备，请参考 BLE\_Chat\EWARM\_BlueNRG-MS\BLE\_Chat.eww、Server\_LowerApp\_OTA 和 Server\_HigherApp\_OTA 工作区）。

## 4.2 构建 BLE 应用以使用 BLE OTA 服务管理器

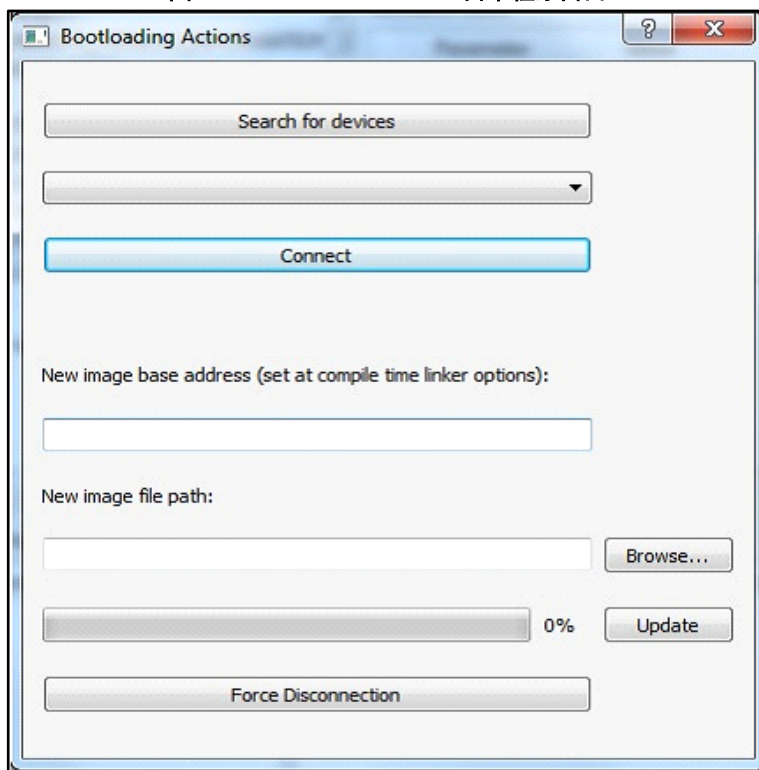
下面是对使用 BlueNRG OTA 服务管理器和 Sensor Demo 的 OTA\_Basic 工作区运行 BlueNRG GUI OTA 自举程序的分步描述：

- 1 将两个开发平台插入 PC 上的 USB 连接器。
- 2 使用 BlueNRG GUI 对将用作 OTA 自举程序从设备的两个平台套件之一进行编程。这需要将板插入运行 GUI 的 PC 上的 USB 连接器，然后在按下 BUTTON\_1 的同时按下平台复位按钮，以便启动 DFU 自举程序（此时 LED\_D2 应闪烁）。

- a. 从“Tools”->“Flash Motherboard FW”中选择复位管理器 hex 预建映像（STM32L\Project\OTA\_prebuilt\_images\DFU\_ResetManager.hex），以将复位管理器上传到板上（hex 格式考虑了合适的应用基址，就该应用而言，为 0x8003000）。
  - b. 加载复位管理器后，需在按下 BUTTON\_2 的同时执行复位，以擦除 OTA 自举程序 EEPROM 中的原有数据，并引导最初启动时的合适复位管理器行为（LED\_D1 发光 5 秒表示过程运行成功）。
  - c. 必须采用相同操作（从 DFU 自举程序激活开始）上传通过 IAR 工作区构建的 OTA 服务管理器应用：对于 BlueNRG 设备，为 OTA\_ServiceManager\EWARM\OTA\_ServiceManager.eww；对于 BlueNRG-MS 设备，为 OTA\_ServiceManager\EWARM\_BlueNRG-MS\OTA\_ServiceManager.eww（预建映像 BlueNRG\_OTA\_ServiceManager\_App.hex 和 BlueNRG-MS\_OTA\_ServiceManager\_App.hex 同样位于 OTA\_prebuilt\_images 文件夹）。
  - d. 必须采用类似操作（从 DFU 自举程序激活开始）上传 SensorDemo BLE 应用（通过 SensorDemo 的 OTA\_Basic 工作区上传），并在构建时使之与 OTA 服务管理器共存（hex 格式考虑了合适的应用基址，就该应用而言，为 0x8006800）。
  - e. （可选）获取“BlueNRG app for smartphones (IoT or Android)”，以发现、连接并使用刚才在步骤 2.d 中加载到平台上的 Sensor Demo 应用。只需确认普通 BLE 应用的服务与下一步要运行的 OTA 自举程序服务共存。
  - f. 为了激活 OTA 服务管理器，用户只需单击按钮 BUTTON\_1：这将激活使软件跳转至 OTA 服务管理器的特定 API（在 OTA 服务管理器上单击相同按钮可跳转至当前有效 BLE 应用）。
- 3 使用 BlueNRG GUI，在第二个平台上对要使用的必要应用进行编程（在 BlueNRG DK 软件包的 Firmware\STM32L1\_prebuilt\_images 文件夹中可以找到 BlueNRG\_VCOM\_x\_x.hex 文件）。
  - 4 在 PC 上打开 BlueNRG GUI，选择与步骤 3 中配置的 BlueNRG 平台相关的 COM 端口（在下拉列表中选择“Port”并单击“Open”）。

- 5 选择“Tools”->“OTA bootloader”以打开包含 OTA 自举程序操作的对话框，并单击“Search for devices”。

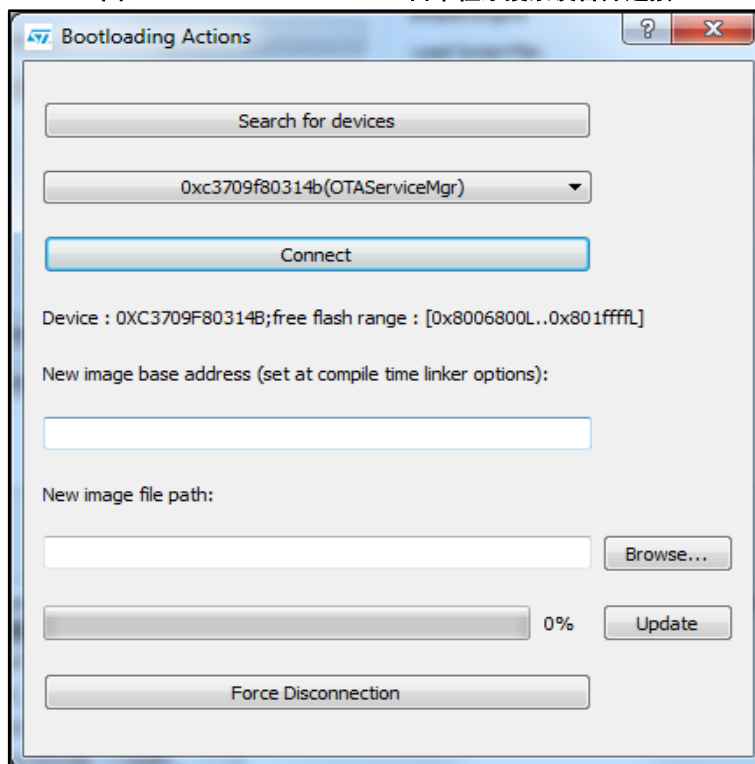
图 9: BlueNRG GUI: OTA 自举程序窗口



- 6 在单击“Search for devices”后，GUI 将启动搜索过程，并返回无线传输范围内运行 OTA 自举程序服务的设备的地址和应用名称相关信息。一旦前面的过程结束，则可以通过“Search for devices”按钮下方的组合框箭头打开设备列表，用户可以从中找到运行 OTA 服务管理器的设备，然后单击“Connect”按钮。如果发现连接了错误的设备，只需单击“Force Disconnection”按钮，并在组合框内重新选择设备。在选择设备并单击“Connect”按钮进行连接后，可以读取其广播的相关空闲内存范围，它限制我们在 BLE 设备上提供以预期范围内的基址和大小编译的映像文件。
- a. 下一步，我们需要提供“基址：0x8006800”，并单击“browse”按钮以选择文件的位置，该文件包含我们计划无线发送至 BLE 设备的映像。因此，用户可以打开 SensorDemo 的 BlueNRG IAR 工作区 OTA\_Basic，并构建相关的 SensorDemo\_OTA\_Basic.bin 映像（位于 SensorDemo\EWARM\OTA\_Basic\Exe 文件夹），该映像包含使用基址 0x8006800 编译的 Sensor Demo 版本。或者，用户可以打开 BLE\_Chat 的 BlueNRG IAR 工作区 Server\_OTA\_Basic，并构建相关的 Server\_OTA\_Basic.bin 映像（位于 BLE\_Chat\EWARM\Server\_OTA\_Basic\Exe 文件夹），该映像包含使用基址 0x8006800 编译的 BLE\_Chat 服务器应用版本。在使用 BlueNRG-MS 设备时，必须执行类似步骤：用户可以打开 SensorDemo 的工作区 OTA\_Basic，并构建相关的 SensorDemo\_OTA\_Basic.bin 映像（位于 SensorDemo\EWARM\_BlueNRG-MS\OTA\_Basic\Exe 文件夹），该映像

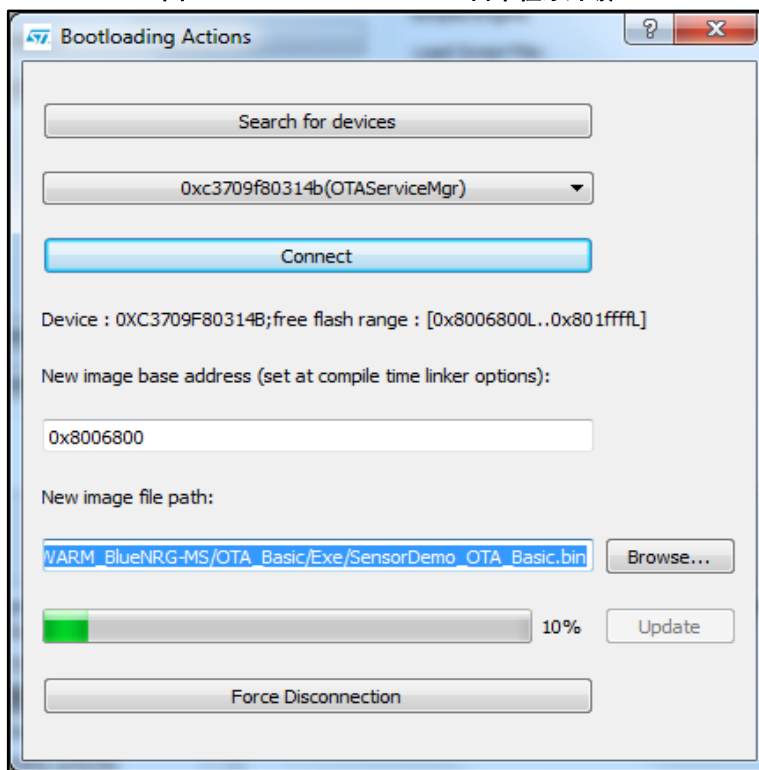
包含使用基址 0x8006800 编译的 Sensor Demo 版本。或者，用户可以打开 BLE\_Chat 的 IAR 工作区 Server\_OTA\_Basic，并构建相关的 Server\_OTA\_Basic.bin 映像（位于 BLE\_Chat\EWARM\_BlueNRG-MS\Server\_OTA\_Basic\Exe 文件夹），该映像包含使用基址 0x8006800 编译的 BLE\_Chat 服务器应用版本。

图 10: BlueNRG GUI: OTA 自举程序搜索设备并连接



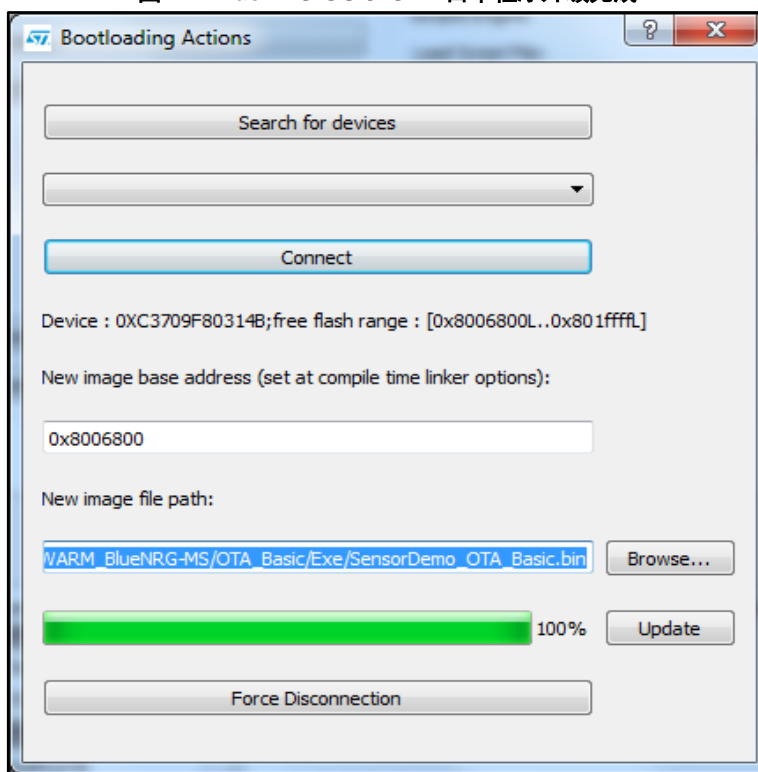
- 7 在浏览到使用正确基址（0x8006800）编译的应用映像后，可以单击“Update”按钮启动 OTA 自举过程。进度条将指示完成这一过程需要的时间：

图 11: BlueNRG GUI: OTA 自举程序升级



- 8 当进度条达到 100%时，过程完成，如下图所示。图 12: "BlueNRG GUI: OTA 自举程序升级完成"

图 12: BlueNRG GUI: OTA 自举程序升级完成



如果我们观察 BLE 平台，会发现闪烁的 LED 的颜色变化，这表明新的应用正在运行。



## 5 缩略语列表

表 9: 本文档中使用的缩略语

缩略语	意义
BLE	低功耗蓝牙
BTL	自举程序
IFR	信息寄存器
OTA	无线
USB	通用串行总线

## 6 相关材料和参考

表 10: 相关文档

术语	说明
BlueNRG, BlueNRG-MS DK	BlueNRG 开发套件软件包
UM1686	BlueNRG 开发套件用户手册

## 7 版本历史

表 11: 文档版本历史

日期	版本	变更
2014 年 5 月 30 日	1	初始版本。
2015 年 4 月 22 日	2	将本文修改为同时适合 BlueNRG 和 BlueNRG-MS 两种器件。

表 12: 中文文档版本历史

日期	版本	变更
2017 年 6 月 20 日	1	中文初始版本。

**重要通知 - 请仔细阅读**

意法半导体公司及其子公司（“ST”）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 ST 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 ST 产品的选择和使用，ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 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 ST 产品如有不同于此处提供的信息的规定，将导致 ST 针对该产品授予的任何保证失效。

ST 和 ST 徽标是 ST 的商标。所有其他产品或服务名称均为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。本文档的中文版本为英文版本的翻译件，仅供参考之用；若中文版本与英文版本有任何冲突或不一致，则以英文版本为准。

© 2017 STMicroelectronics - 保留所有权利