

ESP8266 云端升级指南



版本 1.7
版权 © 2016

关于本手册

本手册结构如下：

章	标题	内容
第 1 章	概述	简单介绍两种固件模式：FOTA 模式（可升级模式）和 Non-FOTA 模式（不可升级模式）。
第 2 章	使用指南	以 <i>ESP8266_NONOS_SDK\examples\IoT_Demo</i> 为例，介绍如何使用云端升级功能。
第 3 章	升级实现原理	介绍云端升级的原理，和软件实现示例。
附录 I	固件版本命名规则	介绍使用 <i>ESP8266_NONOS_SDK\examples\IoT_Demo</i> 时，固件命名的规则要求。

发布说明

日期	版本	发布说明
2016.04	V1.6	首次发布。
2016.08	V1.7	重大修订。

目录

1. 概述.....	1
2. 使用指南	2
2.1. 编译固件.....	2
2.1.1. 修改 IoT_Demo	2
2.1.2. 编译 IoT_Demo	4
2.2. 烧录固件.....	5
2.2.1. 获取 master_device_key.bin	5
2.2.2. 烧录说明.....	6
2.3. 运行说明.....	7
2.4. 乐鑫云升级操作说明	9
3. 升级实现原理	13
3.1. Flash 布局	13
3.2. 软件实现.....	13
3.2.1. ESP8266_NONOS_SDK 应用	14
3.2.2. ESP8266_RTOS_SDK 应用	14
3.2.3. 自定义服务器的要求.....	14
I. 附录-固件版本命名规则.....	15
I.I. 命名规则.....	15
I.II. 版本值规则	16



1.

概述

ESP8266 支持两种主程序固件模式，具体如下：

- Non-FOTA (Firmware Over the Air) 模式，不支持通过 Wi-Fi 从云端下载新版本 ESP8266 固件。
- FOTA 模式，可支持通过 Wi-Fi 从云端下载新版本 ESP8266 固件，进行升级。

表 1-1. 两种烧录方式对比

固件模式	主程序固件	编译区别
Non-FOTA	<i>eagle.flash.bin</i>	编译固件时，STEP 1 选择 Non-boot 模式。
	<i>eagle.irom0text.bin</i>	
FOTA	<i>boot.bin</i>	编译固件时，STEP 1 选择 boot 模式，STEP 2 选择 <i>user1.bin</i> 。
	<i>user1.bin</i>	

 说明：

更多关于编译烧录的信息，请参考文档 [《ESP8266 SDK 入门指南》](#)。

本文介绍使用乐鑫云 (<http://iot.espressif.cn/#/>) 进行云端升级的操作流程，和实现 FOTA 模式的原理。



2.

使用指南

本章以 *ESP8266_NONOS_SDK* 的 *IoT_Demo* 应用程序为示例，介绍如何使用升级功能。

2.1. 编译固件

2.1.1. 修改 *IoT_Demo*

1. ESP8266 SDK 下载链接：<http://www.espressif.com/zh-hans/support/download/sdks-demos>


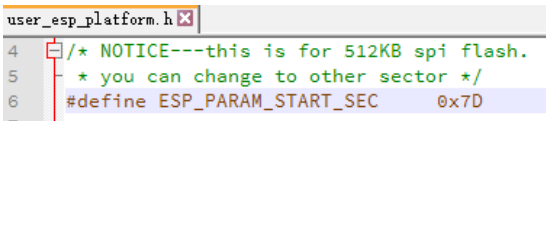
步骤	结果
<ul style="list-style-type: none"> • 以 <i>ESP8266_NONOS_SDK_V2.0.0_16_07_19</i> 为例，下载并解压缩。 • 将待编译的 <i>ESP8266_NONOS_SDK</i> \examples\IoT_Demo 文件夹复制到 <i>ESP8266_NONOS_SDK</i> 根目录下，如右图所示。 	<p>> VM > share > ESP8266_NONOS_SDK</p> <p>Name ^</p> <ul style="list-style-type: none"> bin documents driver_lib examples include IoT_Demo ld lib tools

2. *IoT_Demo* 提供 Smart Light、Smart Plug 和 Sensor 三种设备的简单示例，其中 Sensor 设备不支持升级功能。默认设备类型为 Smart Light。

步骤	结果
<ul style="list-style-type: none"> • 在 <i>ESP8266_NONOS_SDK</i>\IoT_Demo \include\user_config.h 中使能设备类型。 • 以 Smart Light 设备为例，如右图所示。 <p>⚠ 注意： 请每次只使能一种设备类型，进行调试。</p>	<pre> 8 9 #if ESP_PLATFORM 10 #define PLUG_DEVICE 0 11 #define LIGHT_DEVICE 1 12 #define SENSOR_DEVICE 0 </pre>

3. 根据实际使用的 ESP8266 硬件模块 Flash 大小，修改的用户参数区位置。



步骤	结果
<p>如右图  所示，以使用 2048 KB Flash，512+512 map 为例，修改 <code>ESP8266_NONOS_SDK\IoT_Demo\include\user_light.h</code> 中 <code>#define PRIV_PARAM_START_SEC</code> 的值。</p> <p> 提示： 如果使用的是 Smart Plug 设备类型，则修改 <code>user_plug.h</code> 中的同名定义。</p>	
<p>以使用 2048 KB Flash，512+512 map 为例，修改 <code>ESP8266_NONOS_SDK\IoT_Demo\include\user_esp_platform.h</code> 中 <code>#define ESP_PARAM_START_SEC</code> 的值。</p> <p> 提示： 如果使用的是 Smart Plug 设备类型，需同样修改此地址。</p>	

使用不同的 Flash map 对应头文件中的修改位置，如下表 2-1。

表 2-1. 修改 include 文件中的字段（单位：KB）

默认值 (512)	修改后的值				
	1024	2048 (512+512)	2048 (1024+1024)	4096 (512+512)	4096 (1024+1024)
0x3C	0x7C	0x7C	0xFC	0x7C	0xFC
0x3D	0x7D	0x7D	0xFD	0x7D	0xFD



2.1.2. 编译 IoT_Demo

编译 `ESP8266_NONOS_SDK\IoT_Demo`，步骤如下图 2-1，详细的编译指南可参考文档《[ESP8266 SDK 入门指南](#)》。

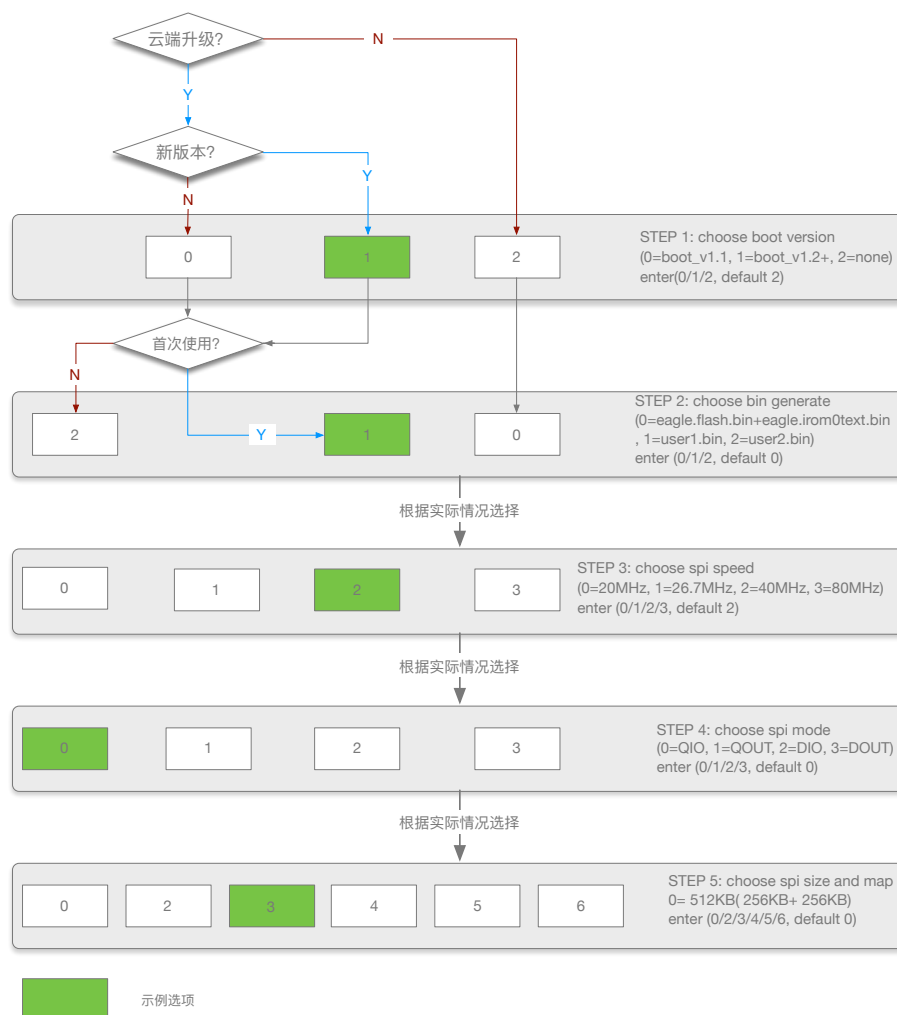


图 2-1. 编译说明

⚠ 注意:

- `user1.bin` 和 `user2.bin` 由同一份应用代码在编译过程中 STEP 2 选择不同的选项，分别编译生成：
 - 编译生成 `user1.bin` 后，先运行 `make clean` 清除上次编译生成的临时文件后，再重新编译生成 `user2.bin`。
- `user1.bin` 和 `user2.bin` 的区别在于它们在 Flash 中的存放位置不同。
- 默认烧录 `user1.bin` 到 Flash，并将 `user1.bin` 和 `user2.bin` 上传到云端，用于循环下载升级。
- 图 2-1 中，颜色标示部分为示例选项，您可以按照实际需求选择。
- 编译 STEP 5 的选项 5 和 6 仅 `sdk_v1.1.0 + boot 1.4 + flash download tool_v1.2` 及之后版本支持。



2.2. 烧录固件

2.2.1. 获取 `master_device_key.bin`

`master_device_key` 是开发者在乐鑫云服务器创建一个智能设备时，乐鑫云自动为该智能设备分配的 ID 值，具有唯一性。智能设备依此享受乐鑫云端服务。

📖 说明:

初次使用乐鑫云，请参考帮助文档 <http://iot.espressif.cn/#/help-zh-cn/>。

1. 注册用户，并登录乐鑫云 (<http://iot.espressif.cn/#/>)，创建一个智能灯设备。

步骤	结果
如右图  所示，登录乐鑫云，点击“设备开发”，点击“+ 创建”。	
创建一个智能灯设备，例如： <ul style="list-style-type: none"> -名字: light-001 -隐私设为“公开设备”，支持共享给他人 -产品选择“创建新的产品” -产品名字: ESP-light -产品类型: 灯光 配置完成后，点击下方的“创建”。 <p> 提示:</p> 开发者可自行任意定义“设备名字”和“产品名字”。	
<ul style="list-style-type: none"> • 创建完成后，将自动跳转到新设备页面。 • 在设备页面，可以看到该设备的 Master Device Key 值。 	

2. 从乐鑫云导出 *master_device_key.bin*。

步骤	结果
如右图  所示，在设备“light-001”的页面右下角，点击“下载 Key BIN”。	
如上点击后，将下载设备“light-001”的 <i>master_device_key.bin</i> 。 - bin 文件名称与设备“light-001”的 Master Device Key 值一致。	 cdecad163d892f30392c2ab065b30e4c17c... 8/5/2016 9:49 PM

2.2.2. 烧录说明

根据实际使用的 ESP8266 硬件模块 Flash 大小，对应烧录地址如下表。

表 2-2. FOTA 固件下载地址（单位：KB）

BIN 文件	各个 Flash 容量对应的下载地址					
	512	1024	2048		4096	
			512+512	1024+1024	512+512	1024+1024
<i>master_device_key.bin</i>	0x3E000	0x7E000	0x7E000	0xFE000	0x7E000	0xFE000
<i>blank.bin</i> （烧录位置一）	0x7B000	0xFB000	0x1FB000		0x3FB000	
<i>esp_init_data_default.bin</i>	0x7C000	0xFC000	0x1FC000		0x3FC000	
<i>blank.bin</i> （烧录位置二）	0x7E000	0xFE000	0x1FE000		0x3FE000	
<i>boot.bin</i>	0x00000					
<i>user1.bin</i>	0x01000					

表 2-3. FOTA 固件说明

BIN 文件	说明
<i>master_device_key.bin</i>	<ul style="list-style-type: none"> 用户从乐鑫云申请，依此享受乐鑫云端服务。 存放于用户参数区，储存地址由用户应用程序自定义。 表 2-2 中的烧录位置为 <i>IoT_Demo</i> 程序按照 2.1.1 设定的示例位置。
<i>blank.bin</i> （烧录位置一）	<ul style="list-style-type: none"> 初始化 <i>RF_CAL</i> 参数区。 烧录位置由应用程序中的 <i>user_rf_cal_sector_set</i> 设置决定。 表 2-2 中的烧录位置为 <i>IoT_Demo</i> 程序中设定的示例位置。 由乐鑫官方提供，位于 <i>ESP266_SDK\bin</i> 路径下。
<i>esp_init_data_default.bin</i>	<ul style="list-style-type: none"> 初始化其他射频参数区，至少烧录一次。 当 <i>RF_CAL</i> 参数区初始化烧录时，本区域也需烧录。 由乐鑫官方提供，位于 <i>ESP266_SDK\bin</i> 路径下。



BIN 文件	说明
<i>blank.bin</i> (烧录位置二)	<ul style="list-style-type: none">初始化系统参数区。由乐鑫官方提供，位于 <i>ESP266_SDK\bin</i> 路径下。
<i>boot.bin</i>	主程序，由乐鑫官方提供，位于 <i>ESP266_SDK\bin</i> 路径下。
<i>user1.bin</i>	主程序，编译应用程序生成，位于 <i>ESP266_SDK\bin\upgrade</i> 路径下。

2.3. 运行说明

1. 参考章节 2.1 修改 *IoT_Demo* 应用程序并编译。
2. 参考章节 2.2 烧录固件到 ESP8266 模块中。
3. 将 ESP8266 模块上电运行（默认波特率 74880），模块将执行智能灯应用。
4. 使用乐鑫手机 APP，配置 ESP8266 模块连接路由器。

说明：

获取乐鑫手机 APP <http://www.espressif.com/zh-hans/support/download/apks>。

乐鑫手机 APP 配置步骤如下：

- (1) 使用在乐鑫云注册的用户名和密码登录手机 APP。



图 2-2. 登录页面



(2) 登录后，点击页面左上角的菜单栏，选择“添加设备”。



图 2-3. 菜单栏

(3) 进入“添加设备”页面后，在右上角点击选择“SoftAP 配置”。



图 2-4.“添加设备”页面

(4) 乐鑫手机 APP 列出附近以“ESP”开头命名 SSID 的 AP（ESP8266 设备）。点击目标 ESP8266 设备，选择“使设备连接至云端”。

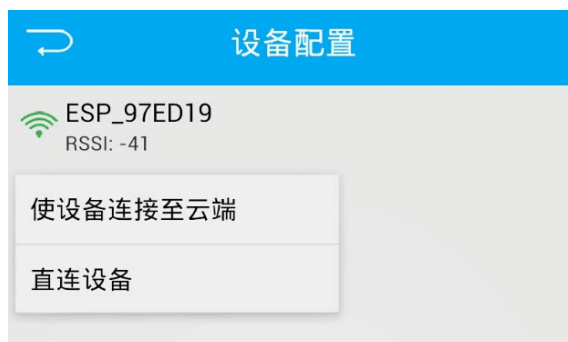


图 2-5.设备配置页面

(5) 输入路由器 SSID 和密码，ESP8266 设备将连接路由器。手机 APP 配置完成。



图 2-6. 设备连接路由器

5. ESP8266 设备通过路由器连接外网后，自动向乐鑫云激活认证。激活认证成功后，登录乐鑫云，在设备页面右侧，可看到设备状态更新为“**Activated**”。

步骤	结果
<ul style="list-style-type: none"> • 登录乐鑫云，点击“设备开发”。 • 点击进入设备“light-001”，页面右侧，设备状态更新为“Activated”，如右图  所示。 	<p>Activated 2016-08-08T12:15:03+08:00</p> <p>Last Active a few seconds ago</p> <p>Device Status developing</p>

6. 乐鑫云认证通过后，ESP8266 智能灯将可享受乐鑫云端服务。

2.4. 乐鑫云升级操作说明

说明：

- 乐鑫云帮助文档 <http://iot.espressif.cn/#/help-zh-cn/>。
- 乐鑫云 API 说明 <http://iot.espressif.cn/#/api-zh-cn/>。

1. 通过用户名和密码登录乐鑫云 (<http://iot.espressif.cn/#/>)，点击“产品管理”。



步骤	结果
<ul style="list-style-type: none"> • 登录乐鑫云，点击“产品管理”。 • 点击待升级的产品，例如，点击前文章节 2.2.1 创建的产品“ESP-light”，如右图  所示。 	
在产品页面右侧，看到“ROM 发布”。	

2. 上传新版本的固件 *user1.bin* 和 *user2.bin* 到云端服务器。

📖 说明：

- 若使用 *ESP8266_NONOS_SDK\examples\IoT_Demo*，要求固件版本号命名规则如本文附录，形如“v1.1.0t45772(a)”，否则将升级失败。
 - 云端固件版本号命名，请与 *IoT_Demo\include\user_iot_version.h* 代码中的定义保持一致，详细说明请见附录。
- 若自行实现应用程序，或者基于 *ESP8266_RTOS_SDK*，则无此版本命名规则要求。

步骤	结果
<p>如右图  所示，在 ROM 发布中，点击“+ 发布”，上传新版本固件。</p> <ul style="list-style-type: none"> - 版本：v1.1.0t45772(a) - 选择上传 <i>user1.bin</i> - 点击“+”符号 - 选择上传 <i>user2.bin</i> - 点击“保存” <p>⚠️ 注意：</p> <ul style="list-style-type: none"> • 固件名称必须为 <i>user1.bin</i> 或者 <i>user2.bin</i>。 • <i>user1.bin</i> 和 <i>user2.bin</i> 均必须上传服务器。 	
保存新版本固件后，点击“设置为当前版本”。	



步骤	结果
如右图  所示，在弹出的提示框中，点击“OK”，确定版本设置。	 <p>The page at iot.espressif.cn says:</p> <p>小心！确定要设置版本“v1.1.0t45772(a)”为当前使用版本吗？设备将收到更新消息，进行版本升级</p> <p>OK Cancel</p>

3. 新版本固件上传到云端服务器后，用户有两种方式可以进行升级：

(1) 方式一：通过乐鑫云设置，升级设备。

步骤	结果
<ul style="list-style-type: none"> 在乐鑫云点击进入待升级的设备页面，例如之前创建的“<i>light-001</i>”设备。 在设备页面右下角，看到“ROM 发布”。 在“ROM 发布”选择新固件版本，点击“升级”，如右图  所示。 	 <p>ROM 发布</p> <p>当前设备的 ROM 版本是 v1.0.5t45772(a)，可以升级到 <input type="text" value="v1.1.0t45772(a)"/> <input type="button" value="升级"/></p>
在弹出的提示框中，点击选择“好”，开始升级。	 <p>小心！确定要升级到发布版本“v1.1.0t45772(a)”吗？在这个过程中设备将进行远程下载和重启</p> <p>取消 好</p>
<ul style="list-style-type: none"> ESP8266 设备下载新版本固件成功后，在乐鑫云的设备页面右下角，“RPC 请求”中设置 action=sys_reboot，点击“请求”。 乐鑫云将通知 ESP8266 重启运行新版本固件。 	 <p>RPC 请求</p> <p>可以发送任意的 action 到设备，附带参数，首先选择请求密钥</p> <p><input type="text" value="device key 23be771b95fec2a84ab623faa47b62b9cab77170"/></p> <p>请求参数 /v1/device/rpc/?deliver_to_device=true&</p> <p><input type="text" value="action=sys_reboot"/></p> <p><input type="button" value="请求"/></p>
在乐鑫云刷新设备页面，可以看到“ROM 发布”中显示已经更新为新版本。	 <p>ROM 发布</p> <p>当前设备的 ROM 版本是 v1.1.0t45772(a)，可以升级到 <input type="text" value="v1.1.0t45772(a)"/> <input type="button" value="升级"/></p>



(2) 方式二：通过乐鑫手机 APP 设置，进行云端升级。

注意，使用乐鑫手机 APP 设置升级，则固件版本号不能超过 v1.4.0，否则将升级失败。

步骤	结果
如右图  所示，登录乐鑫手机 APP，看到目标设备，当前版本号为 v1.0.5。	
点击目标设备，在设备页面右上角，点击选择“云端升级设备”。	
等待升级完成后，手机 APP 主页面显示 ESP8266 智能灯当前运行的固件版本，已经更新为新版本：v1.1.0t45772(a)。	

 说明：

- 设备名称“ESP_97ED19”为应用程序（IoT_Demo）中设置的 ESP8266 softAP 名称。如未设置，则默认为“ESP_XXXXXX”，“XXXXXX”为设备实际 MAC 地址的后三个字节。
- 手机 APP 设置“云端升级设备”后，ESP8266 设备将从服务器下载新版本固件。
 - 下载完成后，ESP8266 设备发出消息通知“下载完成”。
 - 手机 APP 收到“下载完成”的消息后，将通知 ESP8266 设备重启运行新版本固件。
- 手机 APP 如果选择“本地升级设备”，则是由手机 APP 从服务器下载新版本固件到手机，再通过本地局域网将新版本固件推送给 ESP8266 设备进行升级。当用户处在代理服务器网络中，且不能直接通过 ESP 设备上网时，则采用此方法。



3. 升级实现原理

3.1. Flash 布局

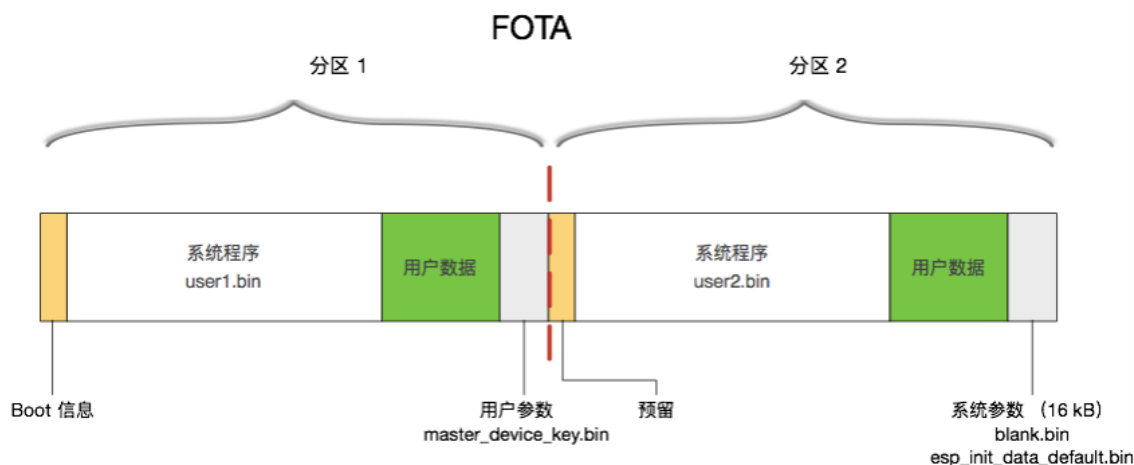


图 3-1. Flash 布局图

分区说明：

- **user1.bin** 和 **user2.bin** 是同一个应用程序，选择不同的编译步骤，分别生成的两个固件，存放在 SPI Flash 不同位置，均可以运行。
- 系统参数区存储了一个标志位，标识启动时应当运行 **user1.bin** 还是 **user2.bin**。
- 启动时先运行 **Boot**，**Boot** 读取系统参数区中的标志位，判断运行 **user1.bin** 还是 **user2.bin**，然后到 SPI Flash 的对应位置读取运行。

示例：

1. 初始状态：**boot.bin** + 版本 **v1.0.0** 的 **user1.bin** + 系统参数区标志位为使用 **user1**。
2. 上传新版本 **v1.1.0** 的固件 **user1.bin** 和 **user2.bin** 到云端服务器。
3. ESP8266 智能设备升级时，先读取系统标志位（当前正运行 **user1**），则从云端服务器下载 **v1.1.0** 的 **user2.bin** 到图 3-1 所示的分区 2 中。
4. 下载完成后，用户可设置重启 ESP8266 智能设备，更新固件。ESP8266 智能设备会修改系统参数区中的标志为使用 **user2**，重启（**system_upgrade_reboot**）运行 **v1.1.0** 的 **user2.bin**。
5. 再下次升级，则参考前述步骤 2 及之后流程。下载 **v1.2.0** 的 **user1.bin** 到图 3-1 所示的分区 1 空间，覆盖之前 **v1.0.0** 的 **user1.bin**。

3.2. 软件实现

⚠ 注意：

ESP8266 不限制开发者必须使用乐鑫云端服务，开发者可以使用自定义的其他云端服务器，调用 ESP8266 SDK 提供的软件接口，通过 HTTP 协议实现下载升级功能。



3.2.1. ESP8266_NONOS_SDK 应用

- 升级接口说明：参考文档《[ESP8266 Non-OS SDK API 参考](#)》。
- 升级应用示例：参考 `ESP8266_NONOS_SDK\examples\IoT_Demo\user\user_esp_platform.c`，由 `user_esp_platform_upgrade_begin` 系列函数实现。

3.2.2. ESP8266_RTOS_SDK 应用

- 升级接口说明：参考文档《[ESP8266 RTOS SDK API 参考](#)》。
- 升级应用示例：
 - 参考 `ESP8266_IOT_PLATFORM\upgrade`。
 - 应用示例下载链接：https://github.com/espressif/ESP8266_IOT_PLATFORM。

 说明：

`ESP8266_IOT_PLATFORM` 是一个基于 `ESP8266_RTOS_SDK` 的应用程序示例。

3.2.3. 自定义服务器的要求

如果开发者不使用乐鑫云，而使用自定义的云端服务器实现升级，自定义云端服务器需支持以下升级步骤：

1. ESP8266 智能设备发送 HEAD 指令云端服务器，询问待下载的固件长度，服务器回复的 HTTP 包头中要求带有固件的长度信息。
2. ESP8266 智能设备根据步骤 1 查询到的长度，在 Flash 待写入区域擦除该指定长度 (`spi_flash_erase_sector`)，等待下载新固件写入 Flash。
3. ESP8266 智能设备发送 GET 指令，从云端服务器下载新版本固件，写入 Flash。

因此，如果开发者使用自定义的服务器，则要求服务器支持步骤 1 中返回固件长度信息的功能，和步骤 3 下载固件的功能，否则无法进行云端升级。



I. 附录-固件版本命名规则

ESP8266_NONOS_SDK 的 *IoT_Demo* 示例中，对下载的固件版本命名做了限制，上传固件至乐鑫云时，版本命名请遵循以下规则，否则将导致云端升级失败。

使用 *ESP8266_RTOS_SDK* 或者开发者自行基于 *ESP8266_NONOS_SDK* 实现升级应用，则无以下限制。

I.1. 命名规则

- 版本命名模板：[v|b]Num1.Num2.Num3.tPTYPE([o||a|n])
- 版本命名形如：v1.0.2t45772(a)

表 I. 固件版本命名示例解析

v	1.0.2	t	45772	(a)
版本类型	版本值	类型标志	设备类型	是否支持升级
可变	可变	固定不变	可变	可变

- 版本类型：v 或者 b
 - v: version, 正式版本
 - b: beta, 测试版本
 - 对应 *IoT_Demo\include\user_iot_version.h* 中的 `#define VERSION_TYPE` 定义

⚠ 注意：

同一版本值的固件命名，请勿同时存在正式版本和测试版本。例如，v1.0.5 与 b1.0.5 请勿共存。

- 版本值：Num1.Num2.Num3
 - Num: 整数，取值范围 [0, 9]
 - 版本值命名示例：1.0.5
 - 对应 *IoT_Demo\include\user_iot_version.h* 中的如下定义
 - 乐鑫手机 APP 目前最大支持到 1.4.0，版本定义请勿超过 1.4.0。

```
#define IOT_VERSION_MAJOR      1U
#define IOT_VERSION_MINOR     0U
#define IOT_VERSION_REVISION  5U
```

- 类型标志：t
 - t: 类型标志符号，后面跟随设备类型 PTYPE 值
- 设备类型值：PTYPE
 - PTYPE: 设备在乐鑫云上的 ptype (product type) 值



- 在乐鑫云查询 ptype 值: <http://iot.espressif.cn/#/api-zh-cn/>
- 对应 `IoT_Demo\include\user_iot_version.h` 中的 `#define device_type` 定义

创建产品

POST /v1/products/

参数 **require userkey**

来源	Key	Value
POST		<pre>{ "products": [{ "name": "name", "description": "description", "serial": "serial", "is_private": 1, "ptype": 27388, "status": 1 }] }</pre>

所有参数可选, 如果提供 serial, 需要保证唯一, is_private: 1/0 (是否私有产品), status: 1/2 (开发状态/发布状态)

目前支持非常多的产品类型, 我们推荐为每一款产品选择一个合适的类型, 以下是具体的类型介绍:



- 升级支持
 - o: online, 支持在线升级
 - l: local, 支持本地升级
 - a: all, 支持在线升级, 也支持本地升级
 - n: not support, 不支持升级
 - 对应 `IoT_Demo\include\user_iot_version.h` 中的如下定义

```
#define ONLINE_UPGRADE    0
#define LOCAL_UPGRADE     0
#define ALL_UPGRADE       1
#define NONE_UPGRADE      0
```

I.II. 版本值规则

- ptype 相同时, 同一版本值, 至多存在 1 个可升级 (o, l, a) 版本。
 - 例如, 当 b1.0.3t45772(o) 已经存在, 则 v1.0.3t45772(o)、v1.0.3t45772(l)、v1.0.3t45772(a)、b1.0.3t45772(l)、b1.0.3t45772(a) 不允许存在。
- ptype 相同时, 同一版本值, 至多存在 1 个不可升级 (n) 版本。
 - 例如, 当 b1.0.3t45772(n) 已存在, 则 v1.0.3t45772(n) 不允许存在。
- ptype 相同时, 不可升级 (n) 版本存在时, 至多存在 1 个可升级 (o, l, a) 版本。



- 例如，b1.0.3t45772(n) 已存在，b1.0.3t45772(o)、b1.0.3t45772(l)、b1.0.3t45772(a)、v1.0.3t45772(o)、v1.0.3t45772(l)、v1.0.3t45772(a) 至多允许存在其中1个。
- 不同 ptype 可存在相同版本值。
 - 例如，当 b1.0.3t45772(n) 已存在，b1.0.3t12335(n) 也可以存在。



乐鑫 IOT 团队
www.espressif.com

免责声明和版权公告

本文中的信息，包括供参考的 URL 地址，如有变更，恕不另行通知。

文档“按现状”提供，不负任何担保责任，包括对适销性、适用于特定用途或非侵权性的任何担保，和任何提案、规格或样品在他处提到的任何担保。本文档不负任何责任，包括使用本文档内信息产生的侵犯任何专利权行为的责任。本文档在此未以禁止反言或其他方式授予任何知识产权使用许可，不管是明示许可还是暗示许可。

Wi-Fi 联盟成员标志归 Wi-Fi 联盟所有。蓝牙标志是 Bluetooth SIG 的注册商标。文中提到的所有商标名称、商标和注册商标均属其各自所有者的财产，特此声明。

版权归© 2016 乐鑫所有。保留所有权利。