

ESP8266 跳转测试方案指南

(v1.8+)



版本 3.0
版权 © 2018

关于本手册

本手册结构如下：

章	标题	内容
第 1 章	概述	介绍 ESP8266 二级 Bootloader V1.8+的跳转测试模式
第 2 章	使能配置步骤	介绍 ESP8266 二级 Bootloader V1.8+ BIN 文件的配置及 Flash 下载工具的配置。

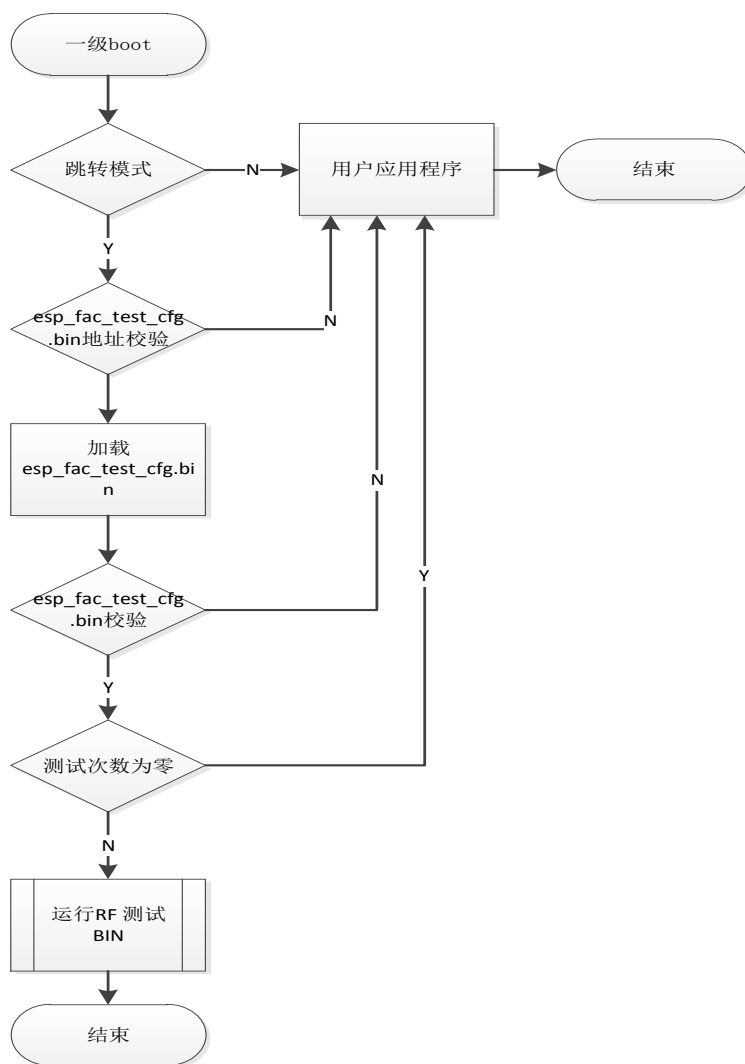
发布说明

日期	版本	发布说明
2016.09	V1.0	首次发布。
2017.05	V2.0	重大修改
2018.01	V3.0	由原方案的 GPIO 使能测试升级为修改 flash 标志位的方式



1. 概述

ESP8266 的二级 boot V1.8+ 支持跳转测试功能，即系统上电后，boot 通过判断 flash 中的 esp_init_data.bin 的第 119 个字节的值来决定是否进入跳转测试模式。当被配置成使能跳转测试时，系统加载 esp_fac_test_cfg.bin 读取产测 bin 的入口地址，然后运行产测 BIN 进行产测。产测 BIN 文件可以提前和应用固件一起离线烧写至 Flash 中，减少生产测试时在线 download 测试 BIN 的时间，有利于提高生产效率。方案流程如下：



说明：

适用版本，boot V1.8+（*SDK Non-OS SDK* 与 *RTOS SDK* 均适用，与 *SDK* 版本无无关）。跳转测试 *BIN* 文件需为乐鑫特定的测试文文件，并不能放置客户二次开发的 *BIN*。



2. 配置方法

2.1. 使能跳转测试

在跳转测试方案中，用于使能该功能的配置在 phy_init_data.bin 中。该文件共有 128 字节，其中和跳转测试有关的字节描述如下：

字节位置	功能	说明
Byte[119]	跳转测试使能位	0xff:不使能 0xf0:使能
Byte[120]	esp_fac_test_cfg.bin 扇区位置的高字节	配置 bin 的位置由 2 个字节决定，例如 byte[120]=0x01, byte[121]=0xfa, 这表示 esp_fac_test_cfg.bin 位于 flash 的 0x1fa000 的位置。
Byte[121]	esp_fac_test_cfg.bin 扇区位置的低字节	
Byte[122]	异或校验位	对 byte[119], byte[120], byte[121]进行异或校验，并写入 byte[122]

2.2 配置测试 bin 文件

和配置有关的 bin 文件由脚本生成，用户需要运行 gen_esp_fac_test_cfg.py 脚本生成 esp_fac_test_cfg.bin 文件。操作步骤如下：

1. 运行 gen_esp_fac_test_cfg.py

输入“cmd”，打开命令终端。输入 Python gen_esp_fac_test_cfg.py，界面如下：

```
D:\>python gen_esp_fac_test_cfg.py
Enter your factory test bin addr(eg. 0x101000):
```

2. 输入产测 bin 文件在 flash 中的位置，如 0x101000。
3. 输入产测次数的最大值，默认为 8 次。如下所示：

```
D:\>python gen_esp_fac_test_cfg.py
Enter your factory test bin addr(eg. 0x101000): 0x101000
Enter the times you want test bin run(default is 8 times, max is 10 times): 8
```

4. 回车，即可得到 esp_fac_test_cfg.bin 文件。



5. 生成 esp_init_data_default.bin

点击烧录工具的“RFConfig”按钮，将第 119 字节的值更改为 0xf0 后，点击“GenInitBin”即可生成，如下所示：

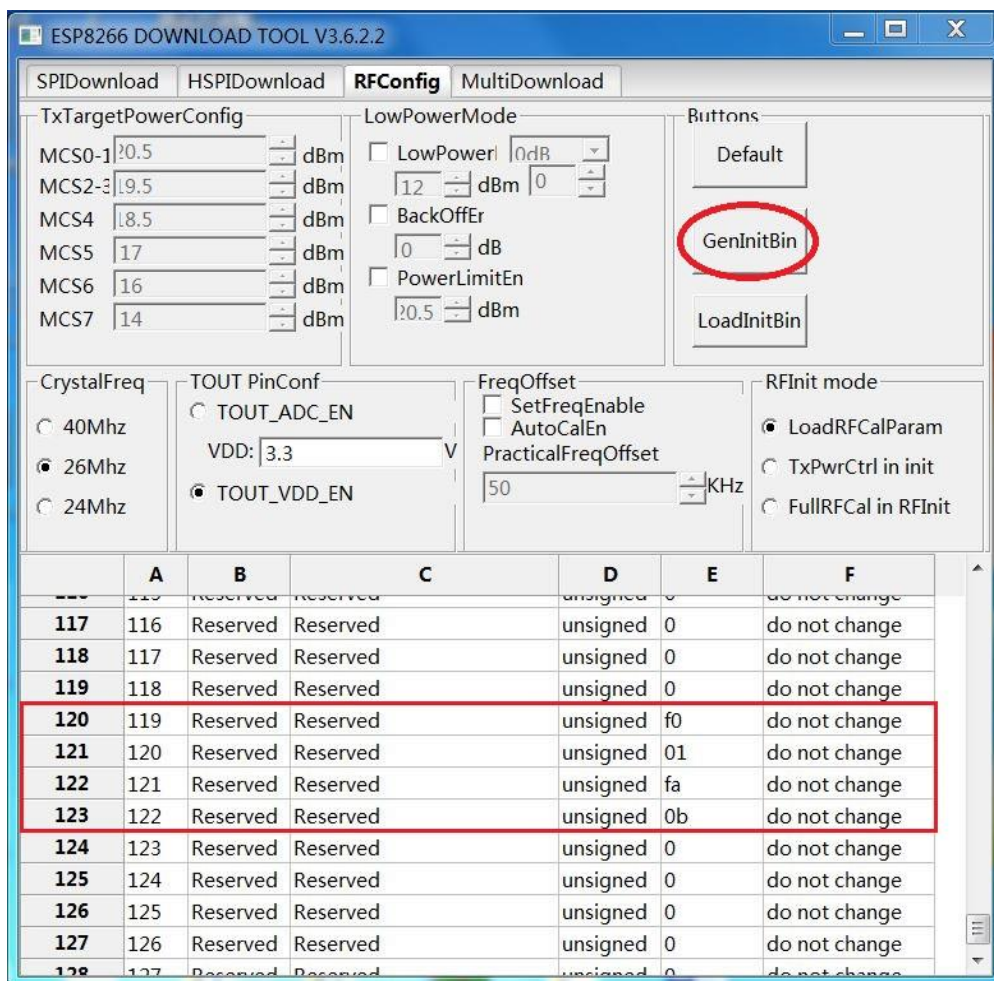


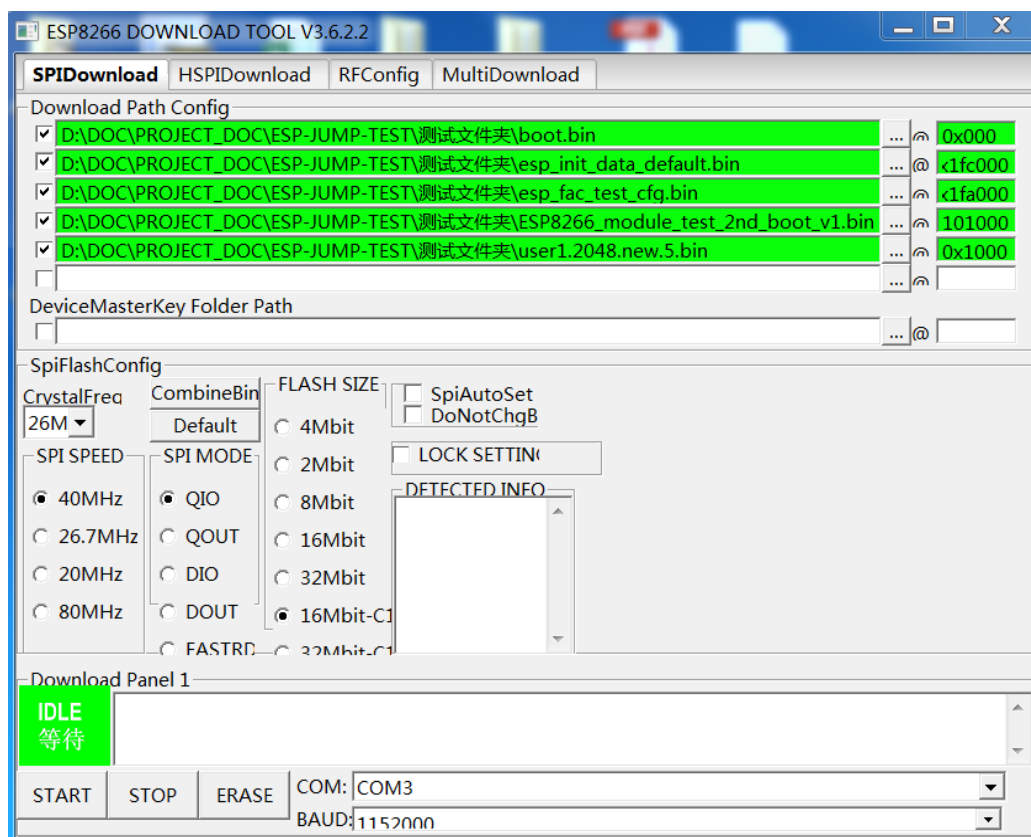
图 2-1 初始化数据配置说明

2.3 烧录

烧录示例如下图所示，以烧写 ESP8266_module_test_2nd_boot_v1.bin 测试 BIN 文件为例，该文件放置在 0x101000 位置，则如下示例 Flash map 进行烧写。烧录完成后，系统正常运行时，若使能了跳转测试功能，则会跳转至 0x101000 位置运行测试 BIN 文件，客户通过这个方式可以完成测试。若未使能跳转测试功能，系统会跳转到 0x1000 位置运行客户的应用固件。



2. 配置方法





3. 运行流程

如果使用模组测试固件，当系统跳转到模组产测 bin 运行时，为方便多个模组同时测试时出现干扰现象，添加了开启测试指令，通过串口向待测模组输入如下指令开始测试。对于 RF 测试固件，不需要输入开始测试指令即可进行测试，只需在测试结束后输入工厂及时间信息即可。详细说明如下：

- 上电后会判断是否是跳转模式。可以根据如下串口日志判断

esp_set_flash_jump_chk_ok_fail, err_code:1 (不是跳转模式)

esp_set_flash_jump_chk_ok_fail, err_code:2 (esp_init_data_default.bin 跳转校验失败)

esp_set_flash_jump_chk_ok_fail, err_code:3 (esp_fac_test_cfg.bin 校验失败)

esp_set_flash_jump_chk_ok [XX:XX:XX:XX:XX:XX] (跳转成功)

- 通过串口向待测模组输入如下指令开始测试：

命令格式：

esp_set_flash_jump_start <param1>

<param1>: 1, 代表使能 模组测试程序

例如指令：esp_set_flash_jump_start 1

(1) 返回：fact test start ok

表示：模组测试程序正常启动。

(2) 返回：fact test start fail

表示：模组测试程序异常。

- 产测成功后需要将工厂及时间等相关信息回写入模组的 flash。

命令格式：

esp_set_flash_test_pass_info <param1> <param1> <param2> <param3> ...
<param17>

<param1> ... <param17>:17byte 的回写 log 参数，有效位为低 byte。参数的格式如下：VV-YYYYMMDDHHMMSS 用于保存产测 pass 的日期和工厂信息。VV, 表示产测工厂代码，默认为 00。YYYYMMDDHHMMSS 表示产测通过时的当前时间“年月日时分秒”。例如：20180103153423 表示 2018 年 01 月 03 号 15 时 34 分 23 秒 在 11 工厂进行了产测并且



2. 配置方法

pass。对应的串口指令为 `esp_set_flash_test_pass_info 0x31 0x31 0x2d 0x32 0x30 0x31 0x38 0x30 0x31 0x30 0x33 0x31 0x35 0x33 0x34 0x32 0x33`

返回打印：

- (1) `esp_set_fac_info_pass` (回写成功)
- (2) `esp_set_fac_info_fail` (回写失败)



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。