

ESP-WROOM-S2

硬件测试平台



版本 1.0
版权 © 2016

关于本手册

本文介绍了乐鑫自主研发的 ESP-WROOM-S2 硬件测试平台。

本手册结构如下：

章	标题	内容
第 1 章	硬件测试平台	介绍硬件测试平台组成部分。
第 2 章	硬件准备	介绍开发板连接方法。
第 3 章	烧录 ESP-WROOM-S2	介绍 ESP-WROOM-S2 串口烧录 HSPI Flash 方法。
第 4 章	Host MCU 配置与编程	介绍 Host 侧处理。
第 5 章	Host - ESP 交互	介绍 Host 与 ESP8266 基本交互流程以及脚本工具。

发布说明

日期	版本	发布说明
2016.08	V1.0	首次发布。

目录

1. 硬件测试平台.....	1
2. 硬件准备.....	3
3. 烧录 ESP-WROOM-S2.....	5
4. Host MCU 配置与烧录	7
5. Host - ESP 交互	8
5.1. 推荐交互步骤.....	8
5.2. 脚本工具.....	8



1.

硬件测试平台

硬件测试平台由以下 3 部分组成：ESP-WROOM-S2 + ESP8266 开发板（ESP-Launcher）底板 + STM32 开发板（型号为：ALIENTEK ELITE STM32F103）。

开发板详细信息：

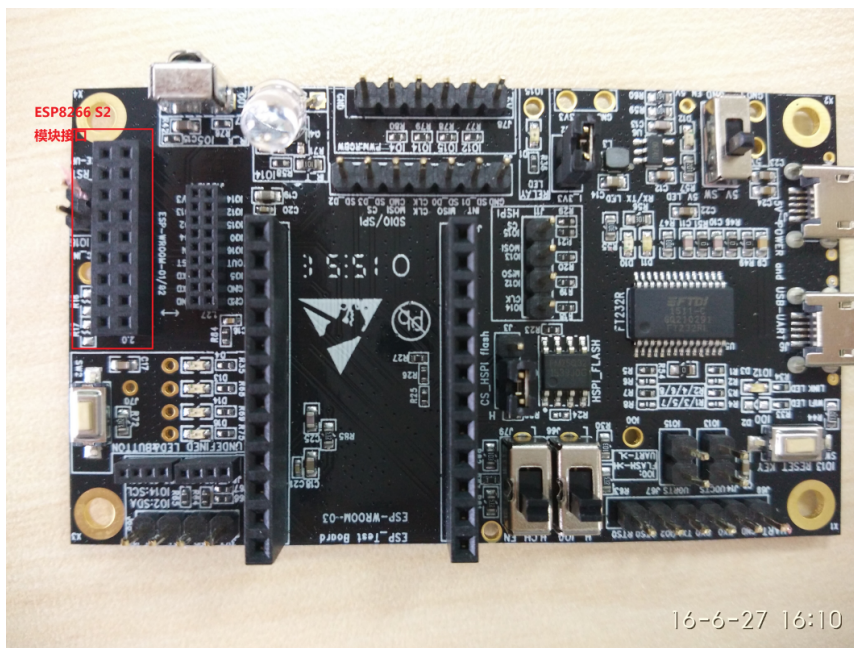


图 1-1. ESP8266 开发板底板

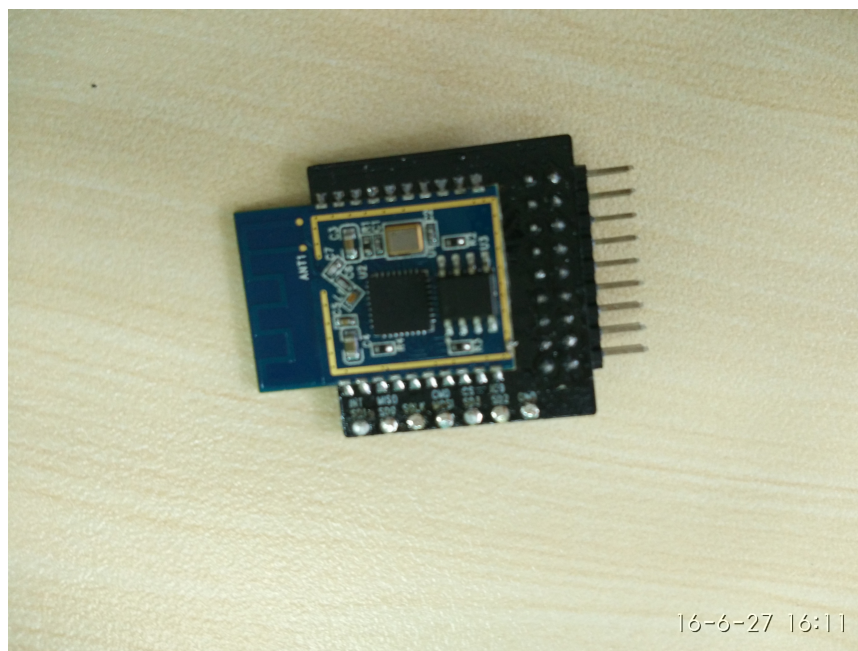


图 1-2. ESP-WROOM-S2

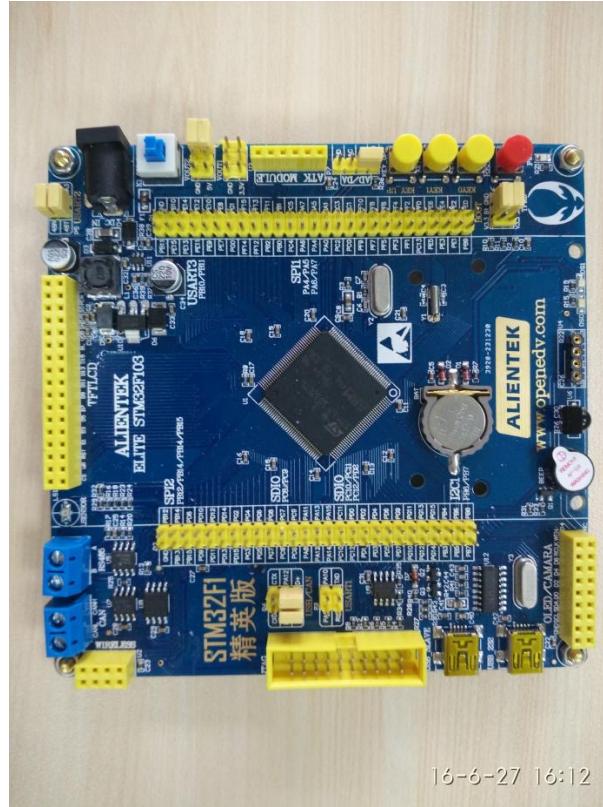


图 1-3. STM32 开发板正面 (型号为: ALIENTEK ELITE STM32F103 精英版)

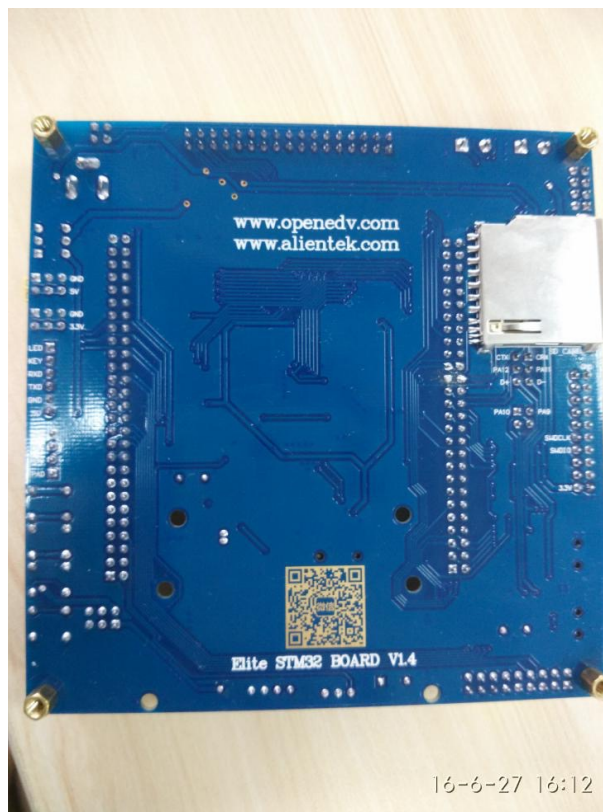


图 1-4. STM32 开发板背面 (型号为: ALIENTEK ELITE STM32F103 精英版)



2.

硬件准备

1. ESP8266 开发板底板移除 R29 电阻。
2. ESP-WROOM-S2 引出 SDIO 通讯线，和 STM32 开发板（或者其它具有 SDIO Host 功能的 MCU 开发板，以下简称 MCU）的 SDIO 进行连接。具体连接须参照 MCU 开发板的要求进行。其中 STM32F103ZET6 的 SDIO 管脚为：
 - PC8 ----> SD0 功能。
 - PC9 ----> SD1 功能。
 - PC10 ----> SD2 功能。
 - PC11 ----> SD3 功能。
 - PC12 ----> SCLK 功能。
 - PD2 ----> CMD 功能。

说明：

用长度相近的跳线连接 SDIO 接口管脚，避免接近高频运行的部件。由于 Host MCU 开发板的不同，可能需要加上上拉电阻或者终端电阻。

3. ESP-WROOM-S2 连接插入到 ESP8266 开发板底板的 ESP-WROOM-S2 接口处，见图 1-4 左上角。

注意：

注意管脚的对应，不要接反了。

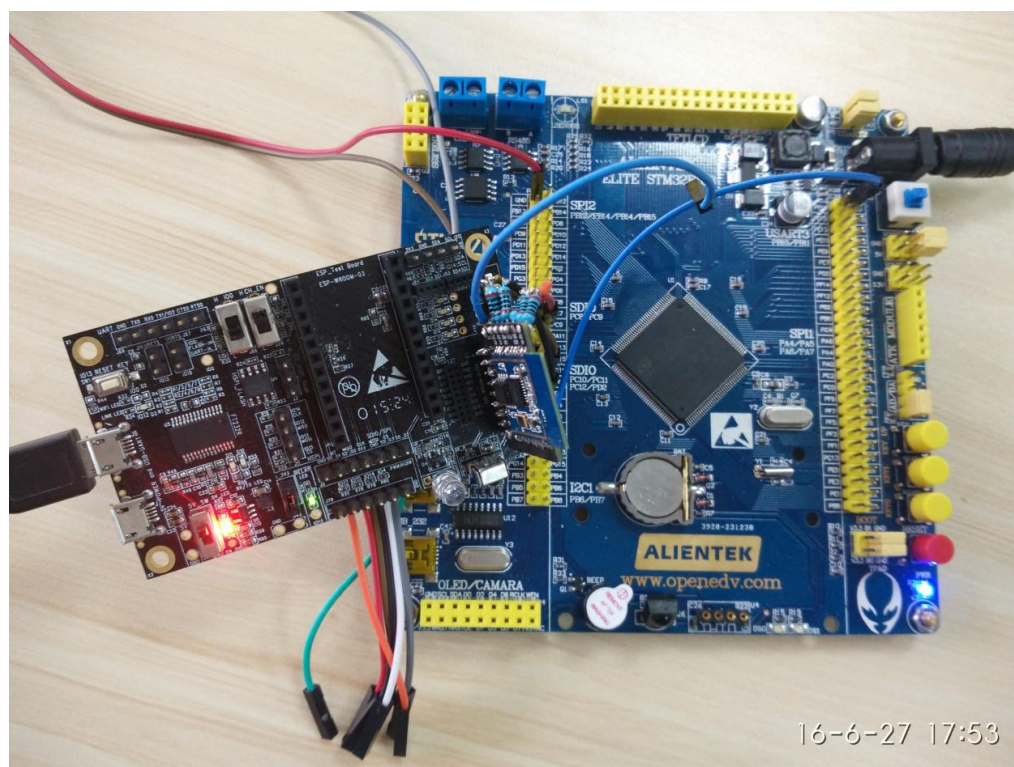


图 2-1. 连接后图片



3. 烧录 ESP-WROOM-S2

ESP-WROOM-S2 模组串口烧录 HSPI Flash 的方法：

1. IO0 拨到低电平，CH_EN 拨到高电平。
2. IO15 管脚使用杜邦线接地，然后上电，使 ESP-WROOM-S2 进入下载状态。
3. 断开 IO15 管脚和地之间的连接。
4. 打开 HSPI Flash 下载工具，进入 HSPI 烧录界面。SPI MODE 选择 DIO，打开 COM 口和 BIN 文件，点击 START 即可实现烧录。
5. 烧录完成后，重新上下电 MCU 开发板和 ESP-WROOM-S2，即可正常运行。

```
blocks:1 ->3ffe8310,228

SDIO Communication thread created

first reg:2,0,9,7,7,wr_busy 0

ready
STM32 recv 8 bytes

AT+GMR
AT version:1.2.0.0(Jun 23 2016 09:17:32)
SDK version:1.5.4.1(3ea431d4)
compile time:Jun 24 2016 19:12:17
OK
```

图 3-1. 启动后信息

⚠ 注意：

基于 SDIO 的 AT 指令，由于管脚冲突，不能使用 UART 流控功能。



烧录工具界面如下（具体以官方发布为准）：

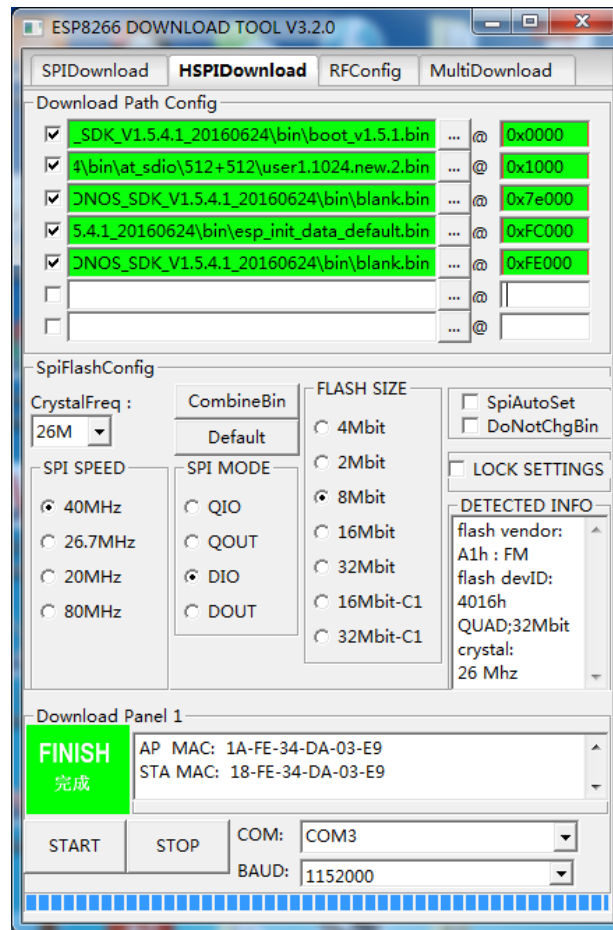


图 3-2. HSPID 烧录工具



4. Host MCU 配置与烧录

由于 ESP-WROOM-S2 需要 MCU 侧传递 *boot* 到 ESP-WROOM-S2，然后才能启动，所以 Host 侧需要软件实现功能，需将 *boot.bin* 同步烧录到 MCU 侧。具体代码实现可参考官方例程。

在例程中，会将 ESP-WROOM-S2 的 *boot.bin* 转化为数组，然后编译到 STM32 代码中，在开发板启动后，将 *boot.bin* 通过 SDIO 传递到 ESP-WROOM-S2，ESP-WROOM-S2 即可使用 *boot* 进行启动。然后通过串口向 STM32 发送 AT 命令，STM32 会将 AT 命令通过 SDIO 透明传输到 ESP8266。

注意 STM32 可以单独运行，ESP-WROOM-S2 作为从机，处理网络连接任务并为 STM32 分流。此外，通过配置 STM32，可以在云端烧录 ESP-WROOM-S2，不需要手动连接或断开跳线。

⚠ 注意：

本文专门介绍 *Host MCU* 与 *ESP-WROOM-S2* 接口连接的配置和烧录，没有涉及 *MCU* 开发板本身的配置与烧录。请参阅开发板用户手册获得更多信息。



5. Host - ESP 交互

5.1. 推荐交互步骤

为了顺利启动 Host MCU 并运行最初存储在 ESP-WROOM-S2 HSPI Flash 上的固件，Host MCU 必须按照特定的步骤与 ESP-WROOM-S2 进行交互。以下是推荐步骤：

1. 在 MCU 上运行的用户程序将 Host MCU 初始化为 SDIO 模式。
2. Host MCU 探测 Slave 设备，若探测到，则将 ESP8266 的 **boot.bin** 加载到 ESP8266 的 RAM。
3. Host MCU 加载 **boot.bin** 完成后，发送命令，控制 ESP8266 跳转到 **boot.bin** 运行。
4. ESP8266 收到 **boot** 跳转指令后，跳转到 **boot** 运行，并从 HSPI Flash 中读取程序运行。
5. 运行过程中，若 ESP8266 接收到 SDIO 传递过来的数据，则 ESP8266 会产生相应中断，并将数据张贴到应用层进行处理。
6. ESP8266 可以通过将数据加载到 SDIO Buffer 中，并通过 IRQ 管脚通知到 Host 的方式，将数据发送到 Host。
7. Host MCU 产生 IRQ 的中断后，读取 ESP8266 中断信息，若有数据则进行读取。

5.2. 脚本工具

Espressif 提供的脚本工具 **bin_to_hex_flash.py** 可以将 **boot.bin** 转化为数组，直接编译到 Host MCU 的源代码中。将 **boot_v1.5.1.bin** 和 **bin_to_hex_flash.py** 放在相同目录下即可运行该脚本工具。脚本工具代码如下：

```
def bin_to_hex(bin_data):  
    hex_list = []  
    for c in bin_data:  
        hex_list.append(ord(c))  
    return hex_list  
pass  
  
def modify_flash_map(flash_map, hex_list):
```



```
4) hex_list[3] = (hex_list[3] & 0x0F) + (int(flash_map, base=10) <<
    return hex_list

def hex_to_array(hex_data):
    data = "unsigned char bin_array[] = {\r\n"
    i = 0
    for _hex in hex_data:
        i += 1
        data += "0x%02x, " % _hex
        if i % 16 == 0:
            data += "\r\n    "
    data = data[:-2]
    data += "};\r\n\r\n"
    data += "unsigned int bin_array_len = %d;\r\n" % len(hex_data)
    return data
pass

def main():
    #f_name = raw_input("Please input file name:\r\n")
    f_name = "boot_v1.5.1.bin"
    flash_map = raw_input("Please select flash map:\r\n")
                    "0. 4M_256_256;\r\n"
                    "2. 8M_512_512;\r\n"
                    "3. 16M_512_512;\r\n"
                    "4. 32M_512_512;\r\n"
                    "5. 16M_1024_1024;\r\n"
                    "6. 32M_1024_1024;\r\n")
    with open(f_name, "rb") as f:
        bin_data = f.read()

    hex_data = bin_to_hex(bin_data)
    hex_data = modify_flash_map(flash_map, hex_data)
```



```
file_content = hex_to_array(hex_data)

with open(f_name + ".flash.bin", "wb+") as f:
    f.write("".join([chr(h) for h in hex_data]))

with open(f_name + ".flash.c", "wb+") as f:
    f.write(file_content)
pass

if __name__ == '__main__':
    main()
```



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。