

# SPI-Flash相关说明

---

AD系列芯片按内部有没有封装FLASH分成两种，一种内部有FLASH芯片丝印后缀是xxxA2(256K Byte)或者xxxA4(512k Byte)，另外一种是没有FLASH(xxxA0)。外接的FLASH分成两种，一种是A0芯片必须外接一个FLASH用来存放代码和少量资源文件，另外一种是非存放程序的FLASH用来存储音频资源文件(A2或者A4芯片的外挂FLASH，或者是A0芯片外接两个FLASH一个存放程序一个专门存音频)。

## 一、存放程序的flash

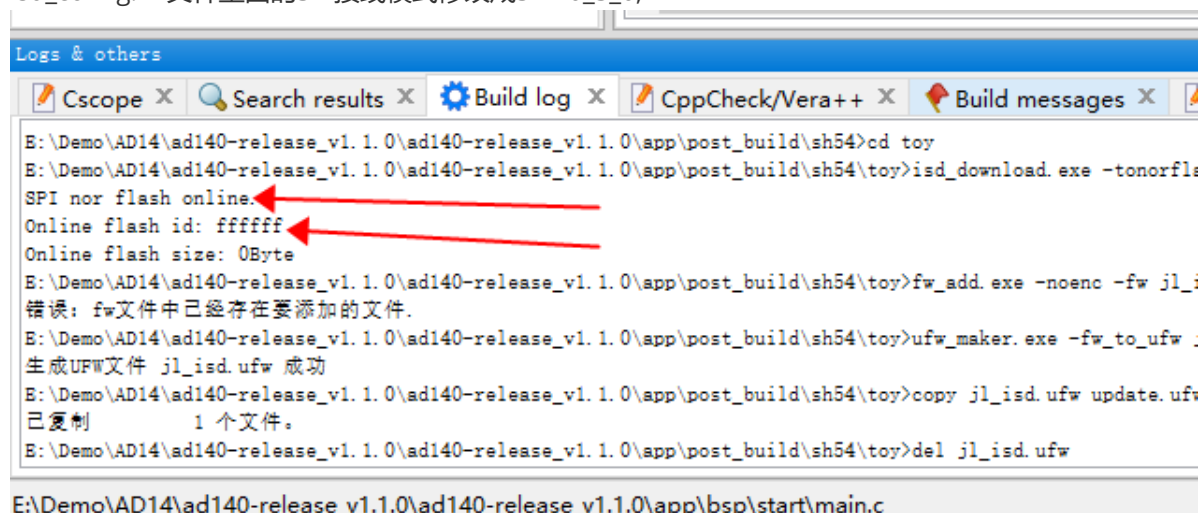
---

这里说的是A2或者A4芯片的内部flash或者A0芯片的SPI0。按图片中的路径找到控制下载的download\_bat.c文件。

```
C usb_host.c  SPI-Flash.md  C download_bat.c X  C port_init.c  B  ...
app > post_build > sh54 > toy > C download_bat.c
28 cd toy
29 isd_download.exe -tonorflash -dev sh54 -boot 0xb00 -div8 -wait
   midi_cfg dir_midi dir_a dir_song dir_eng dir_poetry dir_story
   dir_notice dir_bin_f1x
30 @rem isd_download.exe -tonorflash -dev sh54 -boot 0xb00 -div8
   -wait 300 -uboot uboot.boot -uboot_compress -app app.bin
   0x20000 -res midi_cfg dir_midi dir_a dir_song dir_eng dir_poetry
   dir_story dir_notice -wflash dir_song 0 [PA05_1B_NULL]
31
32 @REM
33 @rem -format vm
34 @rem -format all
35 @rem -reboot 500
36
37
38 @REM //烧写外置flash 命令说明：
39 @rem -wflash dir_song 0 [PA05_1B_NULL]
40 @rem // dir_song : 要烧写的文件名（文件需在download.bat文件夹
   下）
41 @rem // 0 : 文件烧录到外置flash的起始地址
42 @rem // [PA05_1B_NULL]: PA05: 外置flash片选引脚（注意：不能选
   USBDP/USBDM）
43 @rem // 1B : spi1 ,B端口
44 @rem // NULL: power_io & spi1_data_width,
   power_io连接到外置flash vcc引脚 可控制flash电源；
   spi1_data_width:0:单线；1: 双向
45 @rem // 例:NULL/PA00: power_io:无/pa0;
   spi1: 双向模式 （注: power_io不能选USBDP/USBDM）
46 @rem // NUL0/B010: power_io:无/pb1;
   spi1: 单线模式
47 @rem // NUL1/A081: power_io:无/pa8;
   spi1: 双向模式
48 @rem //注意: spi端口只能选B/C（B:PA11,PA12,PA10[CLK,DO,DI];
   C:PA4,PA5,PA6[CLK,DO,DI]）
49
```

## 常见问题1

A0芯片SPI0没有读到，会出现设备可以正常连上但是下载的时候提示flash id全是F。需要找到isd\_config.ini文件里面的SPI接线模式修改成SPI=0\_3\_0;



```
E:\Demo\AD14\ad140-release_v1.1.0\ad140-release_v1.1.0\app\post_build\sh54>cd toy
E:\Demo\AD14\ad140-release_v1.1.0\ad140-release_v1.1.0\app\post_build\sh54\toy>isd_download.exe -tonorfla
SPI nor flash online.
Online flash id: fffffff
Online flash size: 0Byte
E:\Demo\AD14\ad140-release_v1.1.0\ad140-release_v1.1.0\app\post_build\sh54\toy>fw_add.exe -noenc -fw jl_i
错误: fw文件中已经存在要添加的文件.
E:\Demo\AD14\ad140-release_v1.1.0\ad140-release_v1.1.0\app\post_build\sh54\toy>ufw_maker.exe -fw_to_ufw :
生成UFW文件 jl_isd.ufw 成功
E:\Demo\AD14\ad140-release_v1.1.0\ad140-release_v1.1.0\app\post_build\sh54\toy>copy jl_isd.ufw update.ufw
已复制          1 个文件。
E:\Demo\AD14\ad140-release_v1.1.0\ad140-release_v1.1.0\app\post_build\sh54\toy>del jl_isd.ufw

E:\Demo\AD14\ad140-release_v1.1.0\ad140-release_v1.1.0\app\bsp\start\main.c
```

## 常见问题2

下载计数到一半失败了，没有提示下载完成。可能是因为线路接触不好或者FLASH部分区域损坏。

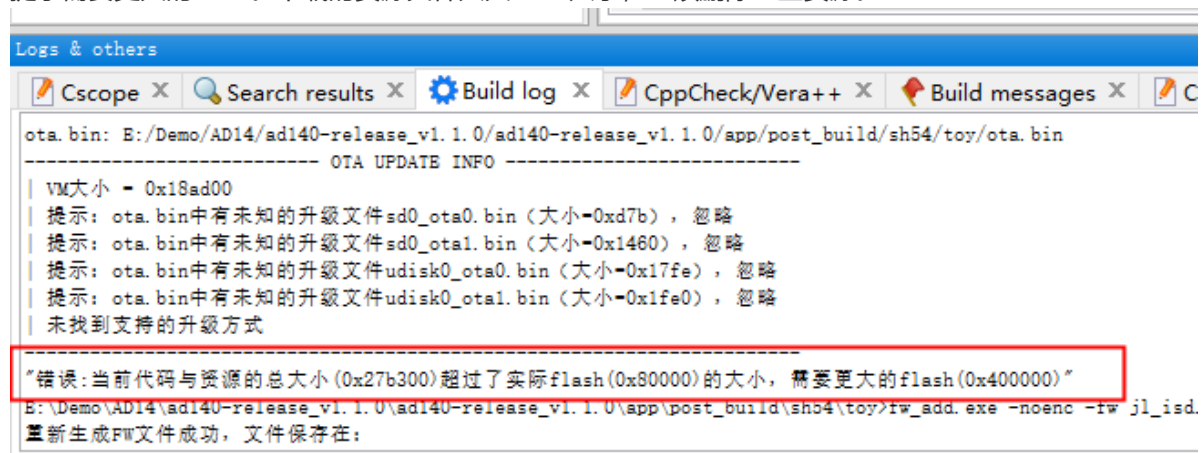


```
开始下载 .....
Write block:38 .
37 .
36 .
.
错误: 下载数据到设备 (sh54)失败, 错误信息:433
E:\Demo\AD14\ad140-release_v1.1.0\ad140-release_v1.1.0\app\post_build\sh54\toy>fw_add.exe -noenc -fw jl_isd.fw .
重新生成FW文件成功, 文件保存在:
jl_isd.fw
E:\Demo\AD14\ad140-release_v1.1.0\ad140-release_v1.1.0\app\post_build\sh54\toy>ufw_maker.exe -fw_to_ufw jl_isd.:
生成UFW文件 jl_isd.ufw 成功

E:\Demo\AD14\ad140-release_v1.1.0\ad140-release_v1.1.0\app\bsp\start\main.c
```

## 常见问题3

提示需要更大的flash。下载的资源文件太大flash太小，必须删除一些资源。



The screenshot shows a build log window with the title "Logs & others". It contains several tabs: "Cscope", "Search results", "Build log", "CppCheck/Vera++", "Build messages", and "C". The "Build log" tab is active, displaying the following content:

```
ota.bin: E:/Demo/AD14/ad140-release_v1.1.0/ad140-release_v1.1.0/app/post_build/sh54/toy/ota.bin
----- OTA UPDATE INFO -----
| VM大小 - 0x18ad00
| 提示: ota.bin中有未知的升级文件sd0_ota0.bin (大小-0xd7b), 忽略
| 提示: ota.bin中有未知的升级文件sd0_ota1.bin (大小-0x1460), 忽略
| 提示: ota.bin中有未知的升级文件udisk0_ota0.bin (大小-0x17fe), 忽略
| 提示: ota.bin中有未知的升级文件udisk0_ota1.bin (大小-0x1fe0), 忽略
| 未找到支持的升级方式
|-----|
| "错误:当前代码与资源的总大小(0x27b300)超过了实际flash(0x80000)的大小, 需要更大的flash(0x400000)"
| E:\Demo\AD14\ad140-release_v1.1.0\ad140-release_v1.1.0\app\post_build\sh54\toy>fw_add.exe -noenc -fw jl_isd.
| 重新生成FW文件成功, 文件保存在:
```

删除方法如图



The screenshot shows a code editor with a file named "download.bat.c" open. The editor displays the following code:

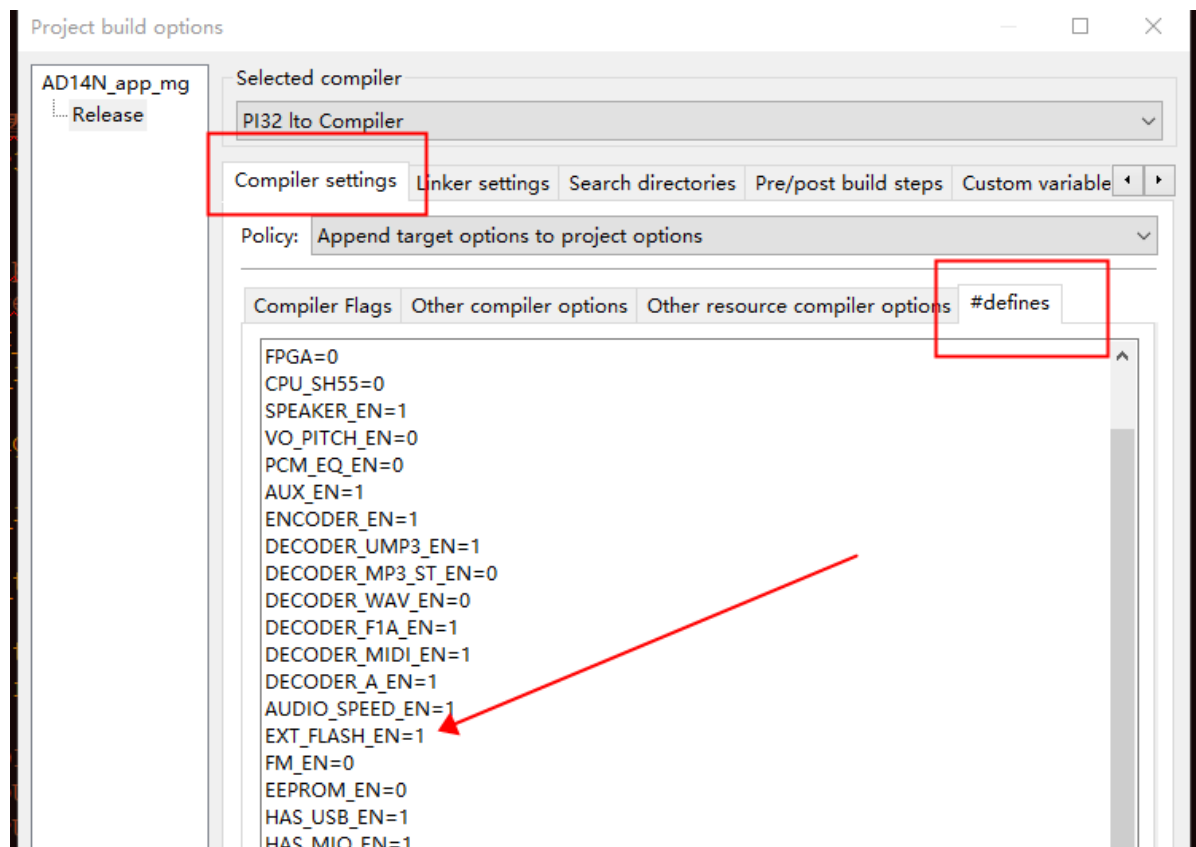
```
app > post_build > sh54 > toy > C download.bat.c
27
28 cd toy
29 isd_download.exe -tonorflash -dev sh54 -boot 0xb00 -div8 -wait
300 -uboot uboot.boot -uboot_compress -app app.bin 0x20000 -res
midicfg dir_midi dir_a dir_song dir_eng dir_poetry dir_story
dir_notice dir_bin flx
30 @rem isd_download.exe -tonorflash -dev sh54 -boot 0xb00 -div8
-wait 300 -uboot uboot.boot -uboot_compress -app app.bin
0x20000 -res midicfg dir_midi dir_a dir_song dir_eng dir_poetry
dir_story dir_notice -wflash dir_song 0 [PA05_1B_NULL]
31
32 @REM
```

A red box highlights the resource file paths in the script, and a red arrow points from the box to a red comment at the bottom: "res后面都是音频资源文件太多太大了需要把这里不用的删掉".

## 二、存放音频资源的flash

A2或者A4芯片的外挂FLASH或者A0芯片的SPI1，只用来存放音频资源。

### 1.需要在工程设置里面打开EXflash



## 2.在device\_list.c里面配置flash的设置选项

```

C download_bat.c  C app_music.c  C device_list.c X  SPI-Flash B ...
app > bsp > start > C device_list.c > ...
4  #include "device.h"
5  #include "norflash.h"
6  #include "spi1.h"
7  #include "sdmmc/sd_host_api.h"
8  #include "usb/otg.h"
9  #include "app_config.h"
10
11 #define LOG_TAG_CONST      MAIN
12 #define LOG_TAG            "[dev list]"
13 #include "debug.h"
14
15 // *INDENT-OFF*
16 //sh54: SPI1:  CLK      , DO      , DI
17 //SPI1: A组IO: IO_PORT_DP, IO_PORT_DM, IO_PORTA_03,
18 //SPI1: B组IO: IO_PORTA_11, IO_PORTA_12, IO_PORTA_10,
19 //SPI1: C组IO: IO_PORTA_04, IO_PORTA_05, IO_PORTA_06
20 //sh55: SPI1:  CLK      , DO      , DI
21 //SPI1: A组IO: IO_PORTB_00, IO_PORTB_01, IO_PORTB_02,
22 //SPI1: B组IO: IO_PORTA_14, IO_PORTA_15, IO_PORTA_13,
23 //SPI1: C组IO: IO_PORTA_06, IO_PORTA_07, IO_PORTA_08,
24 //SPI1: D组IO: IO_PORTB_08, IO_PORTB_09, IO_PORTB_07
25 #if (EXT_FLASH_EN)
26 const struct spi_platform_data spi1_p_data = {
27     .port = {
28         | | | SPI1_GROUPC_IO
29     },
30     .mode = SPI_MODE_UNIDIR_1BIT,
31     .clk = 10000000,
32     .role = SPI_ROLE_MASTER,
33 };
34 //norflash
35 NORFLASH_DEV_PLATFORM_DATA_BEGIN(norflash_data)
36     .spi_hw_num = 1,
37     .spi_cs_port = IO_PORTA_06,
38     .spi_read_width = SPI_MODE_UNIDIR_1BIT,
39     .spi_pdata = &spi1_p_data,
40     .start_addr = 0,
41     .size = 2 * 1024 * 1024,
42     NORFLASH_DEV_PLATFORM_DATA_END()
43 #endif

```

选择模式  
MO和SO共用一根线的  
单线模式

CS脚

容量

### 3.批处理里面下载外挂flash资源

```

26
27
28 cd toy
29 isd_download.exe -tonorflash -dev sh54 -boot 0xb00 -div8 -wait 300 -uboot uboot.boot -uboot_compress -app app.bin -res test_rain_u test_f1b
-wflash dir_story 0 [PA06_1C_A120]
30 @rem isd_download.exe -tonorflash -dev sh54 -boot 0xb00 -div8 -wait 300 -uboot uboot.boot -uboot_compress -app app.bin 0x20000 -res midi_cfg
dir_midi dir_a dir_song -wflash fwq_music_4m.img 0 [PA05_1B_NULL]
31
32 @REM fwq_music_4m.img
33 @rem -format vm
34 @rem -format all
35 @rem -reboot 500
36
37
38 @REM //烧写外置flash 命令说明:
39 @rem -wflash dir_song 0 [PA05_1B_NULL]
40 @rem // dir_song : 要烧写的文件名 (文件需在download.bat文件夹下)
41 @rem // 0 : 文件烧录到外置flash的起始地址
42 @rem // [PA05_1B_NULL]: PA05, 外置flash片选引脚 (注意: 不能选USBDP/USBDM)
43 @rem //
44 @rem // 1B : spi1_B端口
45 @rem // NULL: power_io & spi1_data_width. power_io连接到外置flash vcc引脚 可控制flash电源; spi1_data_width:0:单线; 1: 双向
46 @rem // 例: NULL/PA00: power_io:无/pa0; spi1: 双向模式 (注: power_io不能选USBDP/USBDM)
47 @rem // NULL/B010: power_io:无/pb1; spi1: 单线模式
48 @rem // NULL/A081: power_io:无/pa8; spi1: 双向模式
49 @rem //注意: spi端口只能选B/C (B: PA11, PA12, PA10[CLK, DO, DI]; C: PA4, PA5, PA6[CLK, DO, DI])

```

这个命令表示后面是设置外部FLASH

下载内容

## 4.程序里面设置播放外部FLASH的内容

```

129 }
130
131 static int app_music_init(void *param)
132 {
133     void *bp_info = NULL;
134     u32 file_index = 1;
135     dir_index = 0;
136     key_table_sel((void *)music_msg_filter);
137     decoder_init();
138     app_device_ioctl(APP_DEV_CMD_SET, APP_DEV_INNER_FLASH);
139     device = (void *)app_device_ioctl(APP_DEV_CMD_GET_HDL, 0);
140     fs_type = (void *)app_device_ioctl(APP_DEV_CMD_GET_FS_TYPE, 0);
141     music_device_info_switch(app_device_ioctl(APP_DEV_CMD_GET, 0);
142     music_decode_succ_cb_regist(music_decode_succ_func);
143     #if MUSIC_BREAK_POINT_EN
144     ///read break point

```

```

134 }
135
136 static int app_music_init(void *param)
137 {
138     void *bp_info = NULL;
139     u32 file_index = 1;
140     dir_index = 0;
141     key_table_sel((void *)music_msg_filter);
142     decoder_init();
143     // app_device_ioctl(APP_DEV_CMD_SET, APP_DEV_INNER_FLASH);
144     app_device_ioctl(APP_DEV_CMD_SET, APP_DEV_EXT_FLASH);
145     device = (void *)app_device_ioctl(APP_DEV_CMD_GET_HDL, 0);
146     fs_type = (void *)app_device_ioctl(APP_DEV_CMD_GET_FS_TYPE, 0);
147     music_device_info_switch(app_device_ioctl(APP_DEV_CMD_GET, 0);
148     music_decode_succ_cb_regist(music_decode_succ_func);
149     #if MUSIC_BREAK_POINT_EN
150     ///read break point

```