SV32WB0xx 系列

音频播放功能说明

Nov 2018

关于本手册

本手册介绍了 SV32WB0xx 系列音频播放功能说明

发布说明：

版本	日期	编辑	描述
V1.0	2021/08/18	jon	初始版本

文档变更通知&证书下载：

本文档更新不会逐一通知，用户需要使用时请自行去南方硅谷官网上下载最新版资料；需要相关证书的用户请联系南方硅谷客服　请知悉！

目录

1. 前言

本文介绍 SV32WB0xx 的音频播放 api 说明，以及使用文件系统测试音频的说明。

SV32WB0xx 提供三种 protocol（数据来源）如下：

- FS
- HTTP
- MEMORY

SV32WB0xx 提供两种音频输出格式，如下：

- PDM
- I2S

SV32WB0xx 提供三种音频解码格式，如下：

- DEC_FOR_MP3
- DEC_FOR_WAV
- DEC_FOR_M4A

根据应用场景需要调用相对应的 API，设置不同的数据来源、音频输出格式、音频解码格式即可达到播放效果。

2. API 功能接口

使用 SDK 提供的接口做音频播放功能时，需要按照下面的顺序进行配置：

- 设置并初始化播放器的音频输出格式
- 设置播放器的状态回调
- 设置播放器的数据来源以及路径
- 设置解码器音频解码的格式
- 设置播放器初始音量
- 设置 PA 的状态
- 播放音频

示例代码：（FS 方式播放）

```
ssv_player_init(PDM);
ssv_player_status_register_cb(ssv_player_status);
ssv_player_set_url(FS, (uint8_t *)"/JayChou3.mp3",0);
ssv_player_set_decodec(DEC_FOR_MP3);
ssv_player_set_vol(50);
ssv_player_set_mute(ON_PA);
ssv_player_start();
```

2.1. ssv_player_init

功能：	设置并初始化播放器的音频输出格式,支持 PDM、I2S
函数定义：	int8_t ssv_player_init(ssv_player_output_interface_t interface)
参数：	nProtocol 0 : PDM 1 : I2S
返回值：	0 - Set player success. -1 - Set player fail.

2.2. ssv_player_status_register_cb

功能：	设置播放器的状态回调
函数定义：	int8_t ssv_player_status_register_cb(void

	(*ssv_player_status_cb)(ssv_player_status_t sts,int32_t val));
参数：	cbstatus - ssv_player_status_fun
返回值：	0 - Set success. -1 - Set fail.

2.3. ssv_player_set_url

功能：	设置播放器的数据来源以及路径
函数定义：	int8_t ssv_player_set_url(SSV_FILE_T type, uint8_t *url, uint32_t file_size)
参数：	Protocol-0 : HTTP, 1: MEMORY 2: FS filepath-mp3 file path filesize-file size
返回值：	0 - Set success. -1 - Set fail.

2.4. ssv_player_set_decodec

功能：	设置解码器音频解码的格式
函数定义：	int8_t ssv_player_set_decodec(ssv_player_dec_t dec)
参数：	mode-0:DEC_FOR_MP3 1:DEC_FOR_WAV 2:DEC_FOR_M4A
返回值：	0 - Set success. -1 - Set fail.

2.5. ssv_player_set_vol

功能：	设置播放器初始音量
函数定义：	int8_t ssv_player_set_vol(int8_t vol)
参数：	vol - 0~100
返回值：	0 - Set success. -1 - Set fail.

2.6. ssv_player_set_mute

功能：	设置 PA 的状态
函数定义：	int8_t ssv_player_set_mute(ssv_player_papin_t enable)
参数：	setpa－0: enabled 1: disability
返回值：	0 - Set success. -1 - Set fail.

2.7. ssv_player_start

功能：	播放音频
函数定义：	int8_t ssv_player_start()
参数：	parameter－void
返回值：	0 - Set success. -1 - Set fail.

3. flash 音频播放测试

3.1. pdm 接口读取文件系统的方式播放音频

以下在 SV32WB01 基础上实现，以 story2 工程为例，采用 pdm 接口读取文件系统的方式播放 mp3 音频。

配置一：该目录 projects\story2\mk\board.mk 的配置需要与实际硬件匹配，如果开发板芯片型号是 WB01 晶振 26M 就按下图配置。

```
################################################################
# Project header
################################################################
#BOARD                := SSW114BS_24M
#BOARD                 := SV32WB01_24M
#BOARD                := SV32WB01L_24M
#BOARD                := SV32WB01H_24M
#BOARD                := SV32WB01T_24M
#BOARD                := SV32WB05_24M
#BOARD                := SV32WB06_24M
#BOARD                := SV32WB07_24M

#BOARD                := SSW114BS_26M
BOARD               := SV32WB01_26M
#BOARD                := SV32WB01L_26M
```

配置二：SDK 根目录下的 Makefile 需要打开 IMAGE_FS，此处作用是将 mp3 文件编译到文件系统。

```
IMAGE_MAC         := $(IMAGE_DIR)/$(PROJECT)_mac.bin
IMAGE_PAD         := $(IMAGE_DIR)/$(PROJECT)_pad.bin
IMAGE_INFO        := $(IMAGE_DIR)/$(PROJECT)_info.bin
IMAGE_USER        := $(IMAGE_DIR)/$(PROJECT)_user.bin
IMAGE_ALL         := $(IMAGE_DIR)/$(PROJECT)_all.bin
IMAGE_FULL        := $(IMAGE_DIR)/$(PROJECT)_full.bin
#IMAGE_FS          :=
IMAGE_FS          := $(IMAGE_DIR)/$(PROJECT)_fs.bin
IMAGE_DUMMY       := $(IMAGE_DIR)/$(PROJECT)_dummy.bin
IMAGE_MAP                := $(IMAGE_DIR)/$(PROJECT)_xip1.map
IMAGE_MAP_XIP2    := $(IMAGE_DIR)/$(PROJECT)_xip2.map
```

配置三：将需要的播放的 mp3 文件放置 projects\story2\fs 指定目录，编译时会在此处获取 mp3 文件进行编译。

```
/projects/story2/fs$ ls
jaychou3.mp3
```

配置四：配置目录\projects\story2\mk\feature.mk 下的 feature.mk 文件。

4-1：DEMO_TYPE 选择 0，DEMO_TYPE 为 0 表示 PDM 相关配置，DEMO_TYPE 为 2 表示 I2S 相关配置。

4-2：屏蔽 SUPPORT_LITTLEVGL，目前 story2 还没有开发 VGL 相关的代码，为编译通过加#屏蔽，屏蔽掉不会影响到 stroy2 的功能。

4-3：SUPPORT_FFS 选择 2，0 是关闭文件系统。

```
# 0: none, 2: kws not ds demo
DEMO_TYPE                       ?= 0

#SUPPORT_LITTLEVGL               := 6.1.2
#SUPPORT_LITTLEVGL               := 7.1.0

SETTING_CACHE_WT                := 1
SUPPORT_SPI_LCM                 := 1
#########################################################################
# Filesystem feature
#########################################################################
SUPPORT_FFS                     := 2
# SD Card
SUPPORT_SDC                     := 1
```

配置五：配置相关的 PDM 输出引脚，该目录 projects\story2\src\board\inc\custom\ssv6020C\P32\custom_io_hal.h 下的 custom_io_hal.h，配置 PDM 相关引脚。

```
//ALT0 : AICE_TMSC   /ALT1 : ADC0            /ALT2 : ANTSW_BT_SW_ii       /ALT3 : UART0_RXD
#define M_CUSTOM_P00_MODE       M_CUSTOM_ALT3

//ALT0 : AICE_TCKC   /ALT1 : ADC1            /ALT2 : ANTSW_WIFI_TX_SW_ii /ALT3 : UART0_TXD
#define M_CUSTOM_P01_MODE       M_CUSTOM_ALT3

//ALT0 : SIO13       /ALT1 : NONE            /ALT2 : NONE                 /ALT3 : NONE
#define M_CUSTOM_P13_MODE       M_CUSTOM_ALT0

//ALT0 : GPIO14      /ALT1 : NONE            /ALT2 : NONE                 /ALT3 : PDMTX0_DOUT0
#define M_CUSTOM_P14_MODE       M_CUSTOM_ALT3
```

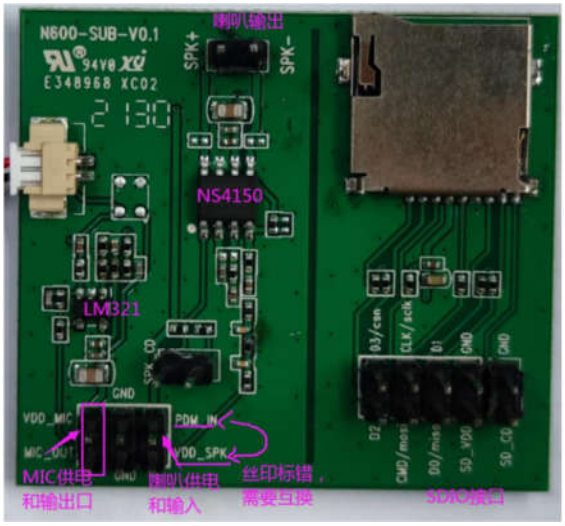
示例代码：（FS 方式播放）

```
ssv_player_init(PDM);//初始化 pdm 输出
ssv_player_status_register_cb(ssv_player_status);//播放器状态回调
ssv_player_set_url(FS, (uint8_t *)"/JayChou3.mp3",0);//FS 方式播放 JayChou3.mp3 文件
ssv_player_set_decodec(DEC_FOR_MP3);//配置 mp3 解码器
ssv_player_set_vol(50);//设置初始音量
ssv_player_set_mute(ON_PA);//打开 PA(在 custom_io_hal.h 上任意选一根引脚控制 PA)
```

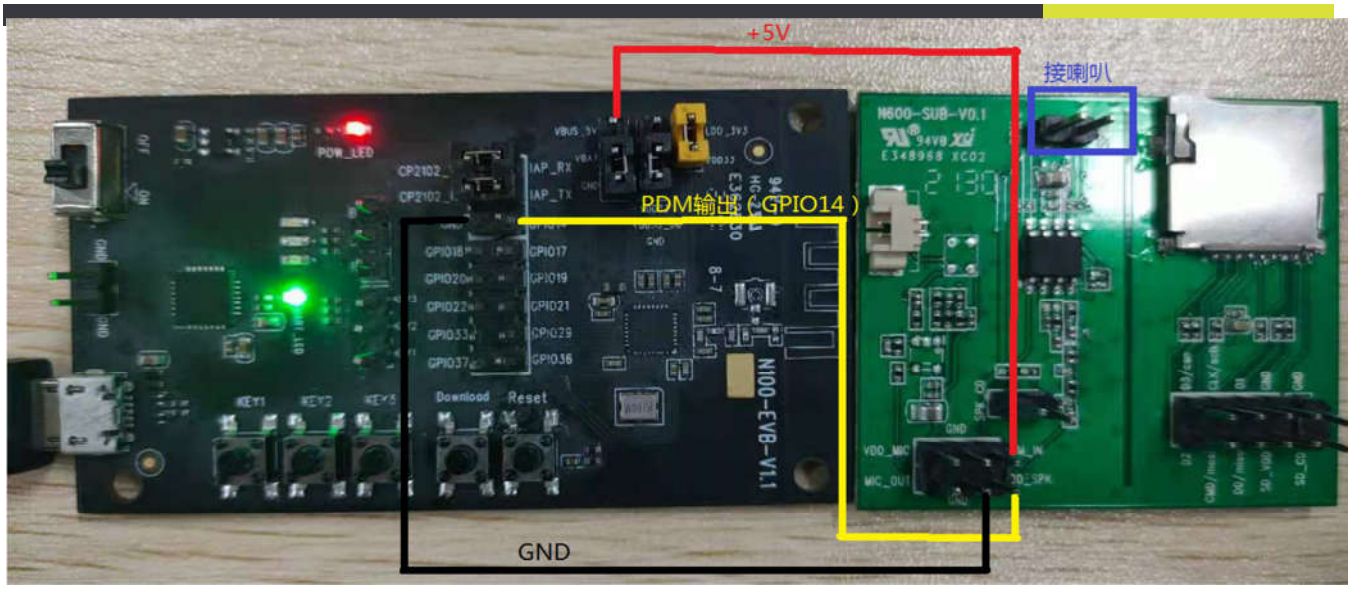
```
ssv_player_start();//开始播放
```

3.2. 文件系统播放硬件接线介绍

PA 硬件采用开发板调试。（附：该 PA 硬件默认情况下已使能，注意：这个 PA 型号是高电平使能低电平失能，这与 SDK 代码逻辑相反，这主要是由外界硬件型号所解决的，想控制 PA 开关拉低或拉高 SPK_CD 丝印处的引脚电平即可，由于 PA 硬件默认已使能，此处演示过程不接 PA 线）



硬件接线示意图，在 SDK 目录下编译分别运行 make clean/make setup p= story2/make 编译烧录 story2_full.bin，运行上述演示代码即可播放/JayChou3.mp3



3.3. PDM 接口读取 SD 卡的方式播放 mp3 音频

PDM 接口读取 SD 卡的方式播放 mp3 音频，在前面的基础上，加入如下配置：

配置一：配置目录 projects\story2\src\board\inc\custom\ssv6020C\P32\custom_io_hal.h 下的 custom_io_hal.h，配置 SPI 相关引脚，这儿选用的是 SPIMAS1

```
//ALT0 : GPIO18     /ALT1 : SD_DATA3        /ALT2 : NONE          /ALT3 : NONE        /ALT4 : SPISLV1_CSN     /ALT5 : SPIMAS1_CSN
#define M_CUSTOM_P18_MODE       M_CUSTOM_ALT5
//ALT0 : GPIO19     /ALT1 : SD_CMD          /ALT2 : NONE          /ALT3 : NONE        /ALT4 : SPISLV1_MOSI    /ALT5 : SPIMAS1_MOSI
#define M_CUSTOM_P19_MODE       M_CUSTOM_ALT5
//ALT0 : GPIO20     /ALT1 : SD_CLK          /ALT2 : NONE          /ALT3 : NONE        /ALT4 : SPISLV1_MISO    /ALT5 : SPIMAS1_MISO
#define M_CUSTOM_P20_MODE       M_CUSTOM_ALT5
//ALT0 : GPIO21     /ALT1 : SD_DATA0        /ALT2 : NONE          /ALT3 : NONE        /ALT4 : SPISLV1_SCLK    /ALT5 : SPIMAS1_SCLK
#define M_CUSTOM_P21_MODE       M_CUSTOM_ALT5
```

配置二：配置目录\projects\story2\mk\feature.mk 下的 feature.mk 文件，打开 SD 卡功能。

```
#########################################################################
# Filesystem feature
#########################################################################
SUPPORT_FFS                     := 2
# SD Card
SUPPORT_SDC                     := 1
```

配置三：配置目录\projects\story2\src\board\param\sdc_param.h 下的 sdc_param.h 文件

3-1：SDC_CSN_PIN 片选脚 custom_io_hal.h 上配置的是 GPIO18，那么这儿配置 18。

3-2：SDC_CD_PIN 插入检测脚，任选一根空闲的 GPIO 引脚即可，演示过程选择了 GPIO36 那么这儿配置 36。

3-3：custom_io_hal.h 上配置的 SPI 是第一组（SPIMAS1），那么这儿配置 DRV_SPIMST_1。

```
#if(CHIP_ID==6006)
#define SDC_CSN_PIN             11  //suggested for 6006
#define SDC_CD_PIN              9   //suggested for 6006
#elif(CHIP_ID==6020)
#define SDC_CSN_PIN             18  //suggested for 6020
#define SDC_CD_PIN              36  //suggested for 6020
//#define SDC_WP_PIN            6   //optional
#else
#error "not support"
#endif


#define SDC_DEBOUNCE_CNT        500
#define SDC_DEBOUNCE_TIME       1

#define SDC_SPI_MST_CHOICE      DRV_SPIMST_1    //DRV_SPIMST_2
#define SDC_SPI_MST_PHASE       SPI_MST_CPHA_0
#define SDC_SPI_MST_POLARITY    SPI_MST_CPOL_0
```

3.4. SD 卡播放硬件接线介绍

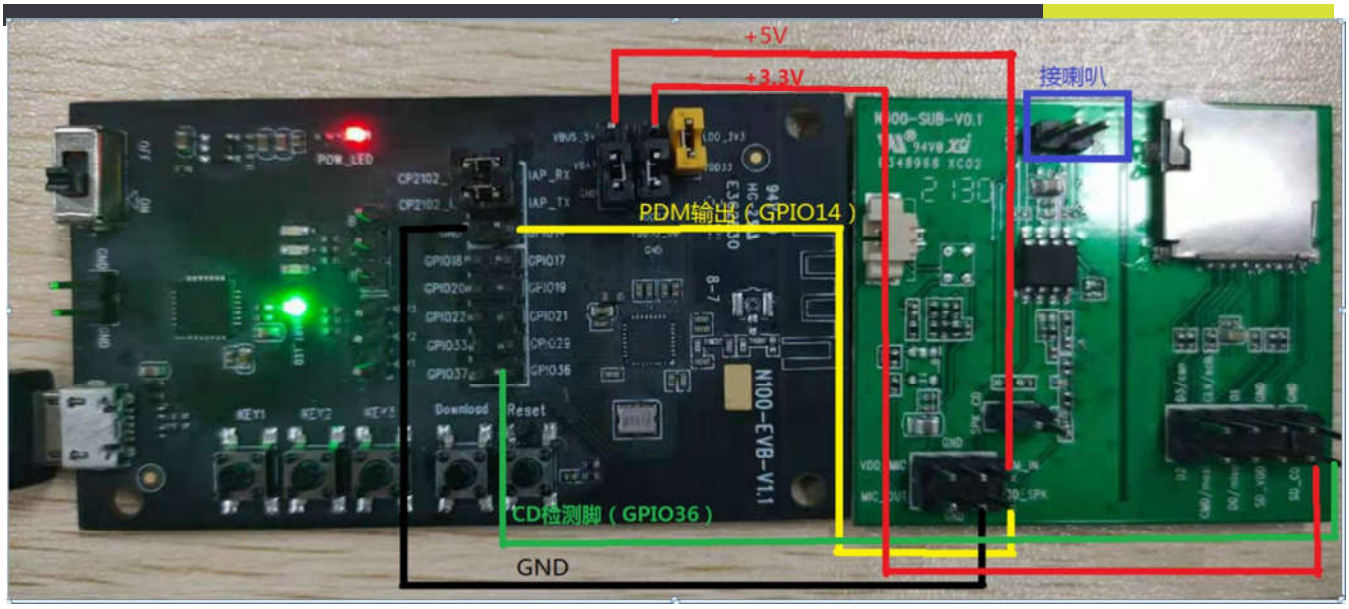
SV32WB0X 系列按图 1 进行 SD 卡接线。

芯片只支持 SPI 模式接 TF 卡，SPI 时钟 clk 最大支持 20MHz：

TF 卡座 pin	SD mode	SPI mode			SV32WB01X SPI	SV32WB05/7 SPI	SV32WB06X SPI	
					SPIMAS1	SPIMAS1	SPIMAS1	SPIMAS2
1	DAT2							
2	CD/DAT3	CS	<----	SPIM_CSN	GPIO18	GPIO18	GPIO18	GPIO36
3	CMD	DI	<----	SPIM_MOSI	GPIO19	GPIO19	GPIO19	GPIO33
4	VDD	VDD						
5	CLK	SCLK	<----	SPIM_SCLK	GPIO21	GPIO21	GPIO21	GPIO32
6	VSS	VSS						
7	DAT0	DO	<----	SPIM_MISO	GPIO20	GPIO20	GPIO20	GPIO35
8	DAT1							
9	SW							

（图 1）

SV32WB01 硬件接线示意图（图 2），其余 CSN（GPIO18）、MOSI（GPIO19）、MISO（GPIO20）、SCLK（GPIO21）四根引脚按（图 1）配置进行连接即可。



（图 2）

若接线没有问题，上电 log 显示 f mount ok 说明 SD 卡识别成功。

```
MAC[44:57:18:3f:49:00]
MAC[44:57:18:3f:49:01]
task ssvradio_init will be deleted
set hz 400000

Init: SDCARD_V2
init card = 2
max_tran_speed: 25000000, _transfer_sck: 20000000

SDHC Card
capacity: 7580 MBytes, sectors: 15523840
set hz 20000000
f mount ok
```

输入 lsa 指令，可显示所有 mp3 文件，/JayChou3.mp3 路径就是内置 flash 的 mp3 文件，带有 sdcard 字样的路径/sdcard/001.mp3 就是 SD 卡内部的 mp3 文件。

```
?>
?>lsa
[cmd_fsal_fsa] Line:845, fs all list
flash total: 1040384 used: 311296
/JayChou3.mp3, size:300024
sd card total: 7560 MB, used: 88 MB
/sdcard/001.mp3, size:5741465
/sdcard/木偶奇遇记.mp3, size:5857832
/sdcard/听妈妈讲那过去的事情.mp3, size:51612453
/sdcard/兔妈妈搬家.mp3, size:4690472
/sdcard/睡美人.mp3, size:5590129
/sdcard/生日快乐.mp3, size:3363743
/sdcard/果冻小人.mp3, size:5296095
/sdcard/歌声与微笑.mp3, size:3711605
/sdcard/世上只有妈妈好.mp3, size:7361434
```

通过下述示例代码，即可播放 SD 卡内部的 001.mp3 文件。

```
ssv_player_init(PDM);//初始化 pdm 输出
ssv_player_status_register_cb(ssv_player_status);//播放器状态回调
ssv_player_set_url(FS, (uint8_t *)"/sdcard/001.mp3",0);//FS 方式播放 001.mp3 文件
ssv_player_set_decodec(DEC_FOR_MP3);//配置 mp3 解码器
ssv_player_set_vol(50);//设置初始音量
ssv_player_set_mute(ON_PA);//打开 PA(在 custom_io_hal.h 上任意选一根引脚控制 PA)
ssv_player_start();//开始播放
```