



# AD24N

V1.0.0 SDK培训

作者 杰理AD应用研发组

2025 / 3 / 10

AD系列32位芯片 杰理开源社区 [服务](#)



此二维码365天内有效 (2025-10-08前)

 钉钉扫一扫群二维码，立即加入群聊

# 目录



1

## 芯片规格介绍

介绍AD24N芯片基本规格与指标

2

## SDK应用框架与新增功能介绍

介绍SDK框架以及一些新增功能

3

## SDK应用配置说明

介绍应用功能配置文件以及其作用

4

## 应用开发注意事项

介绍开发、调试注意事项



---

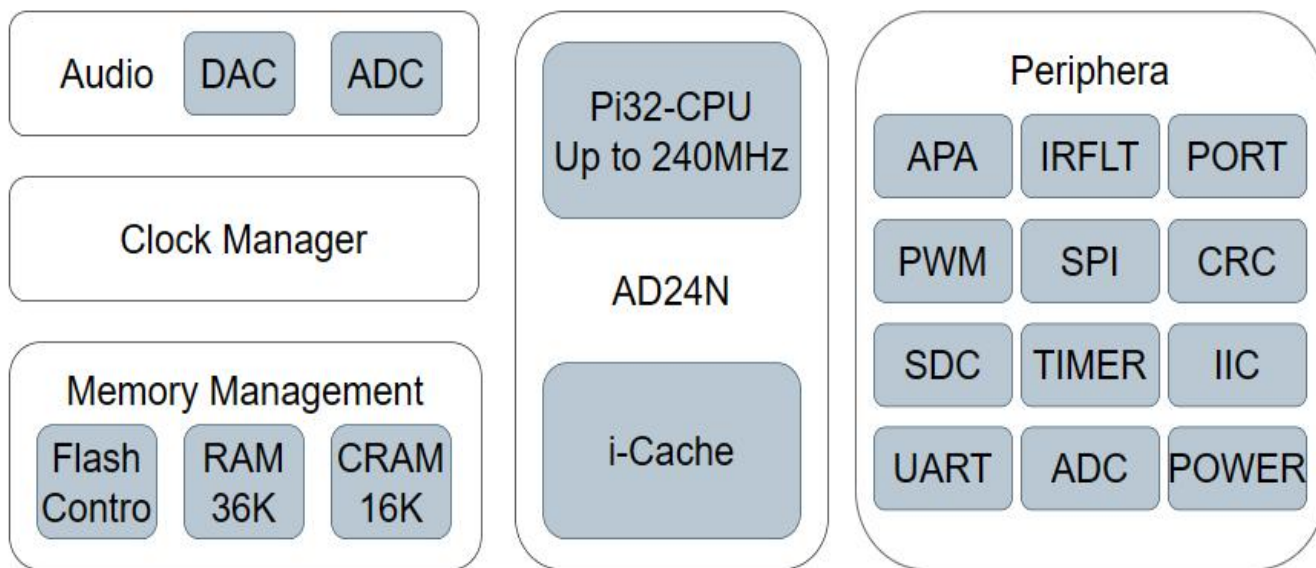
# 01

## 芯片规格介绍

介绍AD24N芯片基本规格与指标

---

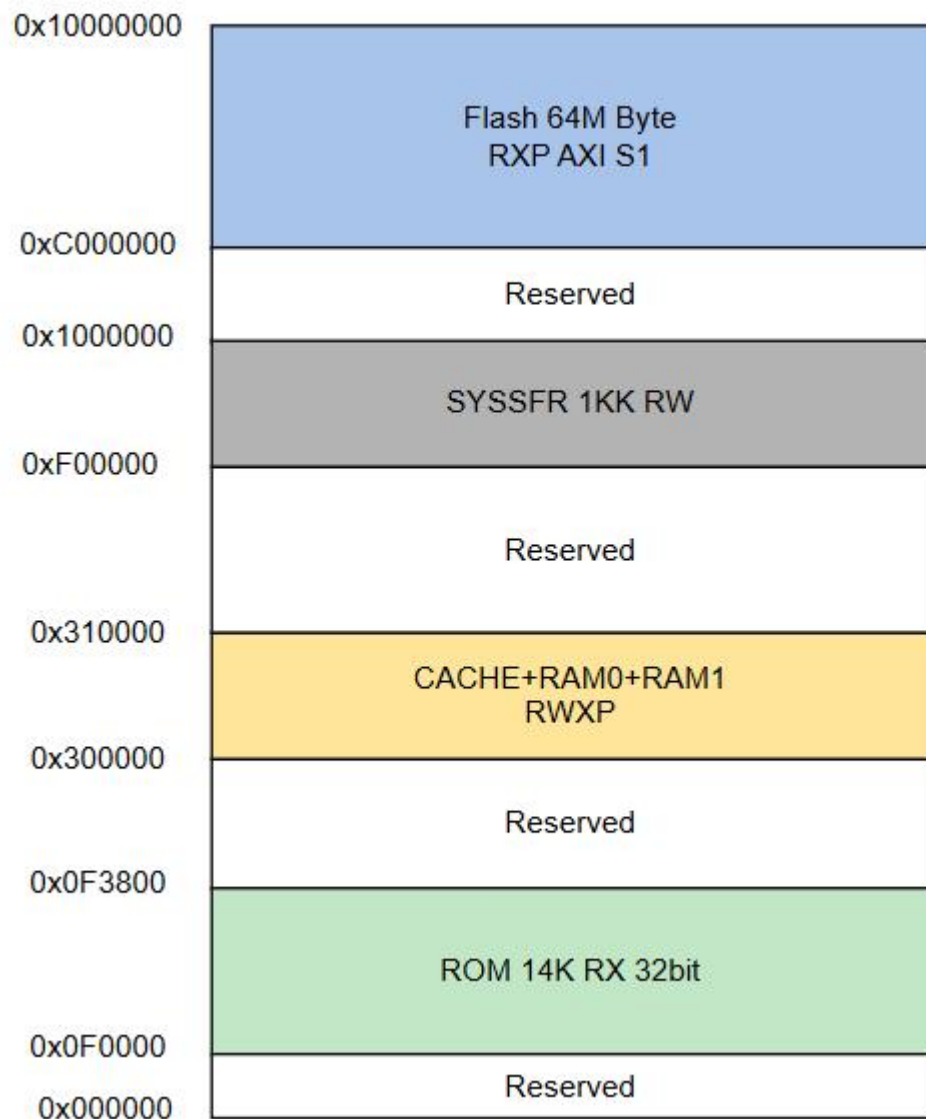
# 一、芯片规格介绍



## ◆ AD24N 芯片规格：

- ◆ 1、拥有单声道DAC，SNR最高可达96dB；
- ◆ 2、拥有单声道AUDIO\_ADC，单端SNR最高可达97dB，支持数字麦和APA口差分输入；
- ◆ 3、HSB max 240MHz，LSB max 120MHz；
- ◆ 4、32+4K字节SRAM，16K i-Cache；
- ◆ 5、支持常见外设功能；

# 一、芯片规格介绍



## ◆ AD24N Memory 映射:

- ◆ 1、拥有32+4K字节SRAM;
- ◆ 2、拥有16K字节的CACHE;
- ◆ 3、拥有64M字节的FLASH寻址空间;



# 一、芯片规格介绍

杰理科技AD系列芯片主要差异-芯片

芯片	AD14N	AD24N	AD15N	AD17N	AD18N	AD16N	AW30N
成本			低	最低		高	
CPU	32位	32位	32位	32位	32位	32位	32位
最高运行时钟	192MHz	240MHz	160MHz	160MHz	160MHz	160MHz	240MHz
cache最大flash寻址空间	32M字节	64M字节	64M字节	64M字节	64M字节	64M字节	64M字节
RAM	32K	36K	20K	14K	40K	40K	64K
Cache ram	16K	16K	8K	4K	16K	16K	16K
AUDIO_ADC	A-MIC/AUX	A-MIC/D-MIC/AU	无	无	无	A-MIC/AUX	A-MIC/AUX
AUDIO_ADC采样率	8K~24K	8K~48K	无	无	无	8K~48K	8K~48K
CLASS-D直驱喇叭	支持	支持	支持	支持	支持	无	支持
APA THD	-33dB	-71dB	-33dB	-70+dB	-70+db	无	-72+dB
模拟AUDIO DAC	单声道	单声道	无	无	单声道	立体声	单声道
模拟AUDIO DAC SNR	83db	93dB	无	无	81dB	>97db	>92dB
AUDIO_DAC/APA采样率	8K~32K	8K~96K/48K	8K~32K	32K~48K	AUDIO_DAC:8K~96K APA:32K~48K	8K~96K	AUDIO_DAC:8K~96K APA:32K~48K
gpio	25MAX	27MAX	33MAX	22MAX	47MAX	33MAX	25MAX
io映射	2 out 4 input	I/O Crossbar+ 8 input+8 out	4 out 4 input	I/O Crossbar+ 8 input+8 out	I/O Crossbar+ 8 input+8 out	I/O Crossbar+ 8 input+8 out	I/O Crossbar+ 4 input+4 out
MCPWM	4路	4路	4路	2组（每1组有2路）	2组（每1组有2路）	2组（每1组有2路）	3组（每1组有2路）
TIMER_PWM	1路(Timer2)	4路	1路(Timer2)	3路	4路	3路	4路
RTC功能	不支持		软件RTC	不支持	软件RTC (v1.8.0SDK不支持)	硬件RTC	软件RTC
RTC（32K晶振时钟源）	不支持	不支持	不支持	不支持	v1.8.0 SDK不支持	部分封装支持	部分封装支持
SDMMC	支持	支持	支持	不支持	不支持	支持	支持
USB	支持	支持	不支持	不支持	暂不支持	支持	支持
SRC（重采样）	硬件（多路）	硬件（多路）	硬件（多路）	硬件（多路）	软件（2路）	硬件（多路）	硬件（多路）

# 一、芯片规格介绍-AD24N与AD14芯片硬件规格对比

## ◆ AD24N

- ◆ 1、32bit pi32架构 CPU, **HSB max 240MHz**;
- ◆ 2、32K + **4K RAM** + 16K CACHE;
- ◆ 3、SFC SUPPORT **64M** FLASH;
- ◆ 4、27 I/O max, with **I/O Crossbar**;
- ◆ 5、GPIO上下拉都支持**10K/100K/1M**档位
- ◆ 6、16bit 单声道 APA+DAC, **支持两者同时输出声音**;
- ◆ 7、16bit单声道AUDIO\_ADC, **模拟麦支持单端/差分输入, 支持数字麦**;
- ◆ 8、**支持在APA口接入的喇叭做为输入采集外部音频信号**
- ◆ 9、**AUDIO OUT & AUDIO IN更高的指标性能**
- ◆ 10、**AUDIO 支持更多采样率 (但软件编解码及算法未支持到这么多)**

## ◆ AD14N

- ◆ 1、32bit pi32架构 CPU, HSB max 160MHz;
- ◆ 2、32K RAM + 16K CACHE
- ◆ 3、SFC SUPPORT 32M FLASH
- ◆ 4、25 I/O max, 2 out / 4 input channel
- ◆ 5、GPIO上拉支持10K, 下拉支持60K
- ◆ 6、16bit 单声道 APA + DAC;
- ◆ 7、16bit单声道AUDIO\_ADC, 模拟麦仅支持单端输入

# 一、芯片规格介绍-功耗指标介绍

## ◆ AD24N低功耗性能对比其他芯片

模式 / CPU	AD14N / AC104N	AD15N	AD17N	AD18N	AD24N
Soft off	1.7uA+	1.7uA+	1.4uA+	3.3uA+	2uA+
Power down	27uA+	27uA+	23uA+	33uA+	19uA+
Power off	/	/	/	/	约10uA+
支持最多唤醒口	8个	12个	8个	8个	8个

注：AD24N首版SDK暂不支持poweroff，后续版本会发布；



# 一、芯片规格介绍-音频指标介绍

## ◆ AUDIO\_ADC性能指标 (MIC\_PGA=27dB)

测试参数	value/UNIT
input swing	40 mVrms
SNR	71 dB
DR	70 dB
THD+N	-71 dB
THD	-76 dB
Noise Ratio	-72 dB

## ◆ AUDIO\_DAC性能指标

测试参数	value/UNIT
Output Swing	680 mVrms
SNR	93 dB
DR	92 dB
THD+N	-75 dB
THD	-75 dB
Noise floor	15 uVrms

## ◆ AUDIO\_APA性能指标

测试参数	value/UNIT (APA空载)
Output Swing	2.59 Vrms
DR	92 dB
THD+N	-71 dB
THD	-71 dB
Noise floor	36 uVrms



---

# 02

## SDK应用框架与新增功能介绍

介绍SDK应用框架以及一些新增功能

---

## 二、SDK应用框架介绍 - 启动与初始化

涉及函数:

涉及文件:

### 1、工程主循环入口

`int c_main(int cfg_addr);`

`app/bsp/start/sh58/main.c`

### 2、系统初始化

`void system_init(void);`

`app/bsp/start/sh58/init.c`

### 3、应用主循环入口

`void app(void);`

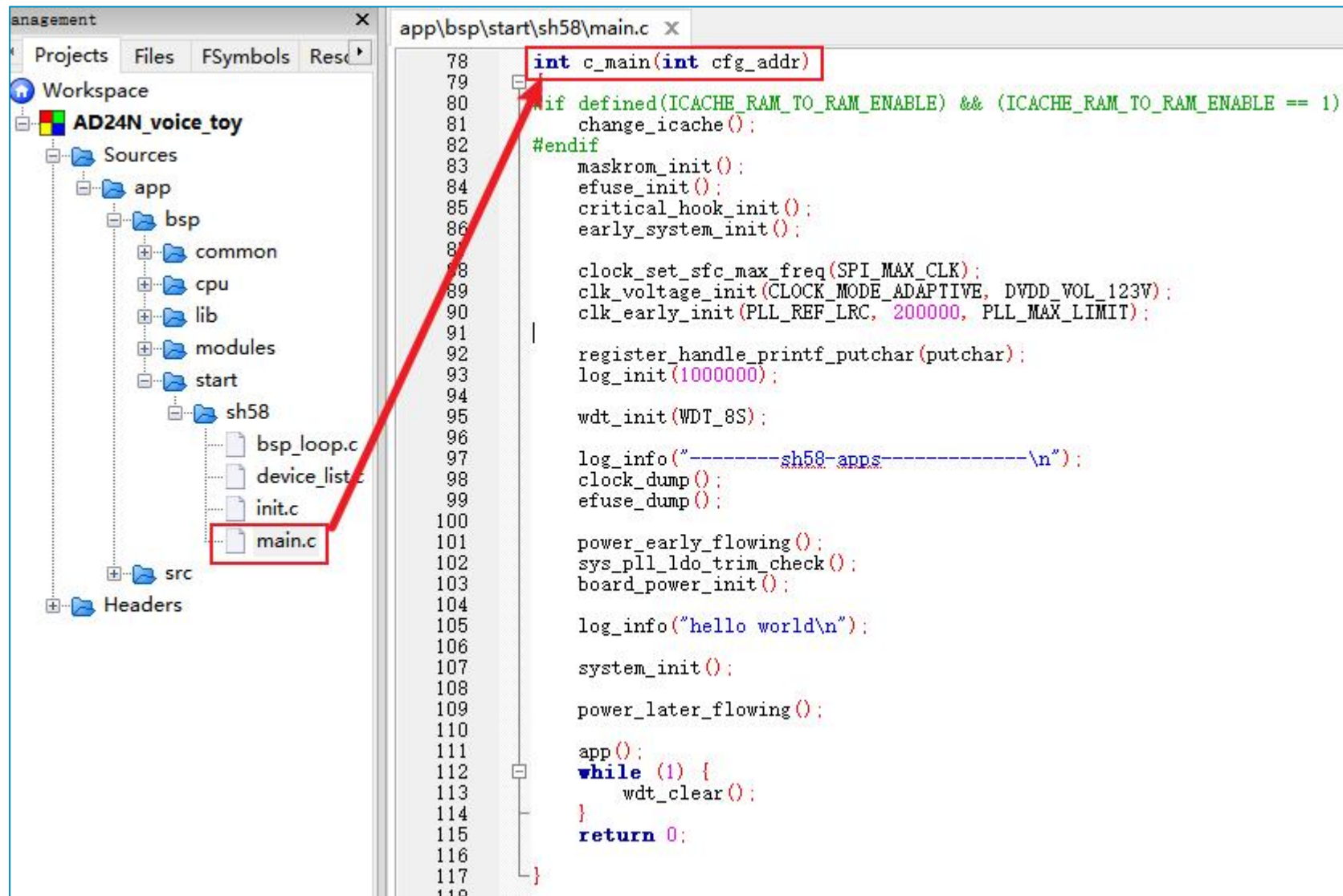
`app/src/voice_toy/toy_main.c`



## 二、SDK应用框架介绍 - 工程主循环入口

### 1、工程主循环入口

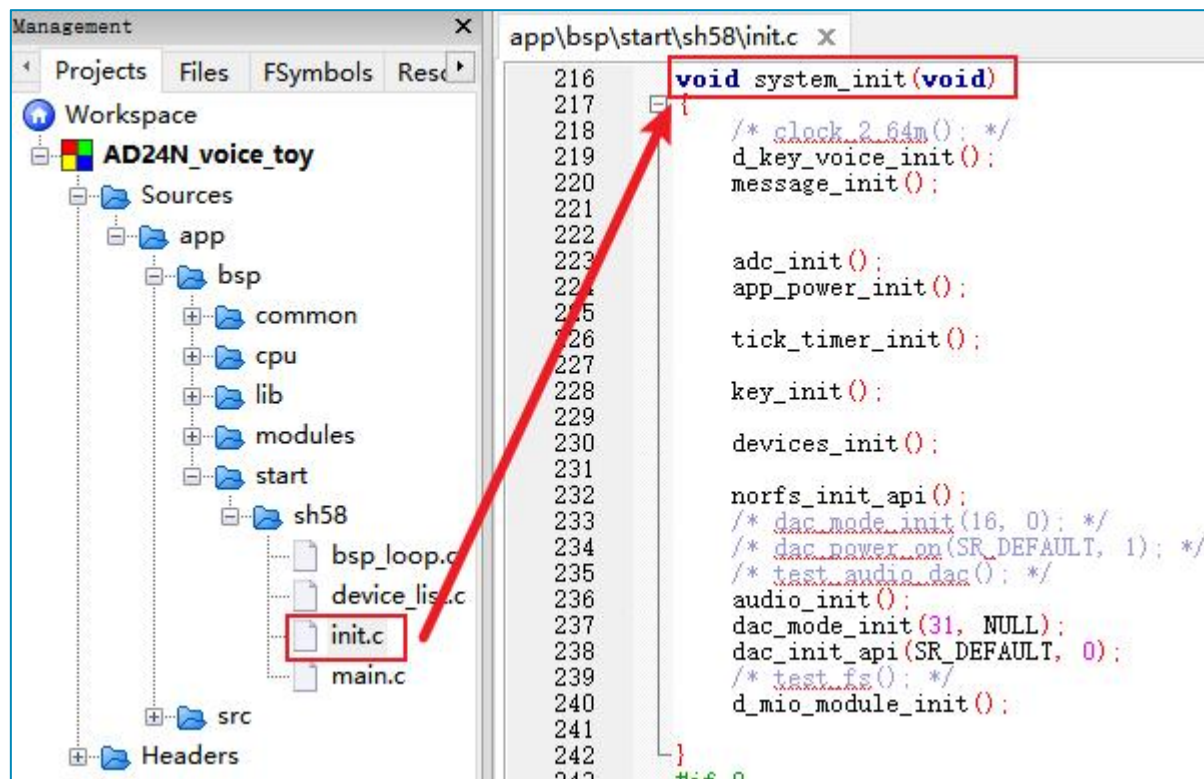
- ◆ `int c_main(int cfg_addr);`
- ◆ `app/bsp/start/sh58/main.c`



## 二、SDK应用框架介绍 - 系统初始化

### 2、系统初始化

- ◆ `void system_init(void);`
- ◆ `app/bsp/start/sh58/init.c`



The screenshot displays an IDE interface. On the left, the 'Management' pane shows a project tree for 'AD24N\_voice\_toy'. The tree structure is as follows:

- Workspace
  - AD24N\_voice\_toy
    - Sources
      - app
        - bsp
          - common
          - cpu
          - lib
          - modules
          - start
            - sh58
              - bsp\_loop.c
              - device\_list.c
              - init.c**
              - main.c
  - Headers

On the right, the code editor shows the file `app\bsp\start\sh58\init.c`. The code defines the `system_init` function:

```
216 void system_init(void)
217 {
218     /* clock 2.64m */
219     d_key_voice_init();
220     message_init();
221
222
223     adc_init();
224     app_power_init();
225
226     tick_timer_init();
227
228     key_init();
229
230     devices_init();
231
232     norfs_init_api();
233     /* dac_mode_init(16, 0); */
234     /* dac_power_on(SR_DEFAULT, 1); */
235     /* test_audio_dac(); */
236     audio_init();
237     dac_mode_init(31, NULL);
238     dac_init_api(SR_DEFAULT, 0);
239     /* test_fs(); */
240     d_mio_module_init();
241
242 }
```

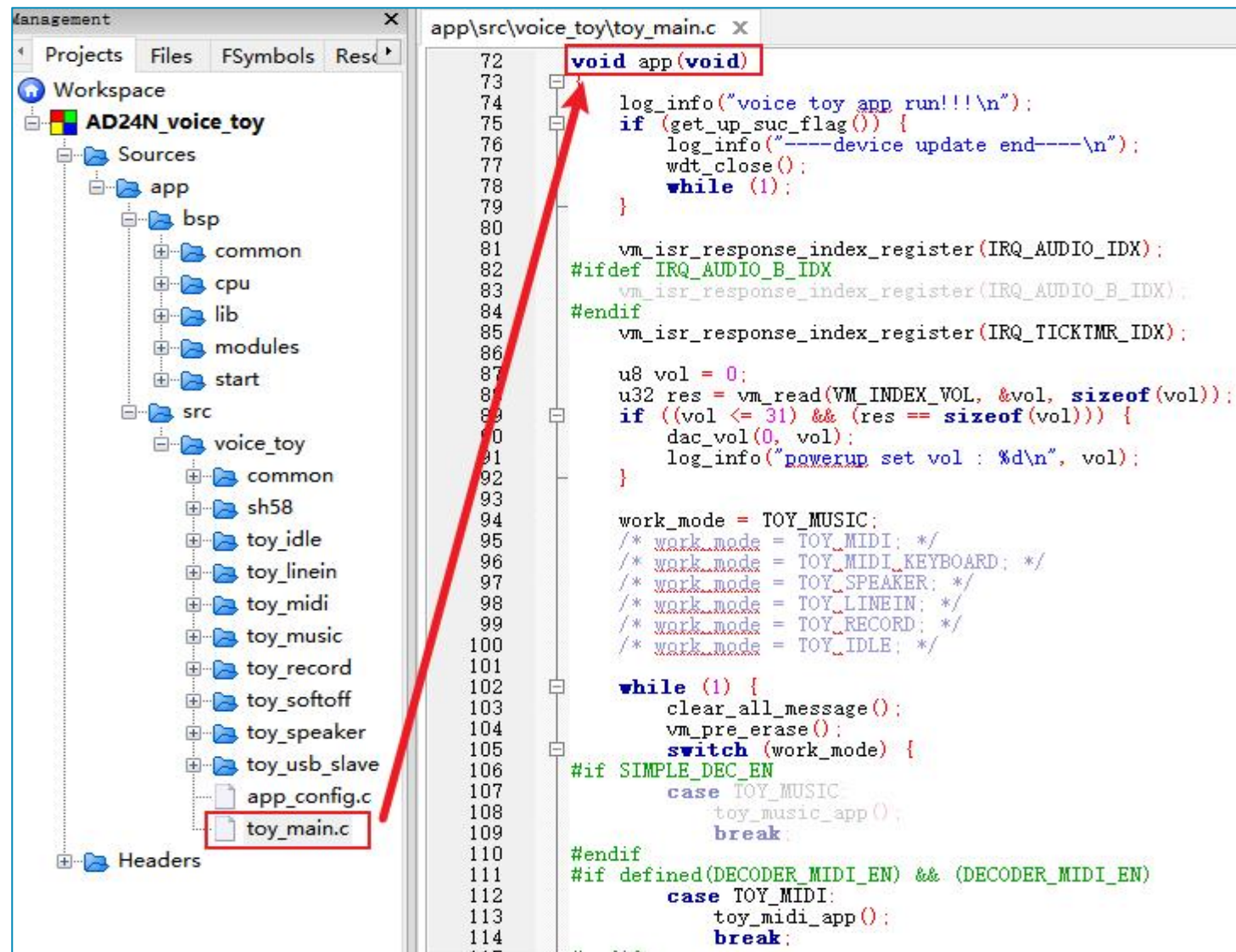
A red arrow originates from the `init.c` file in the project tree and points to the `void system_init(void)` function definition in the code editor.



## 二、SDK应用框架介绍 - 工程主循环入口

### 3、工程主循环入口

- ◆ `void app(void);`
- ◆ `app/src/voice_toy/toy_main.c`



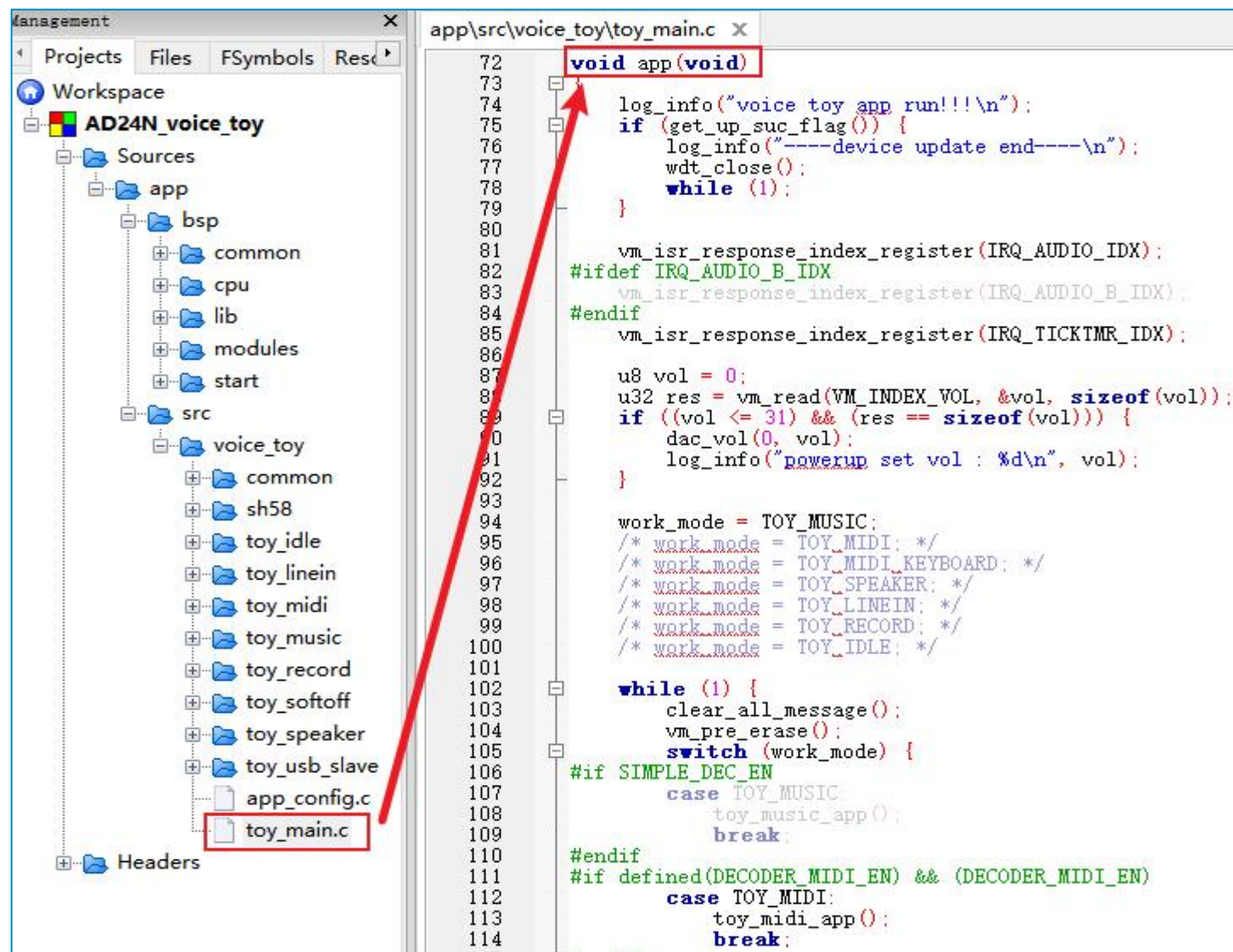
## 二、SDK应用框架介绍 - 应用循环的作用

涉及文件:

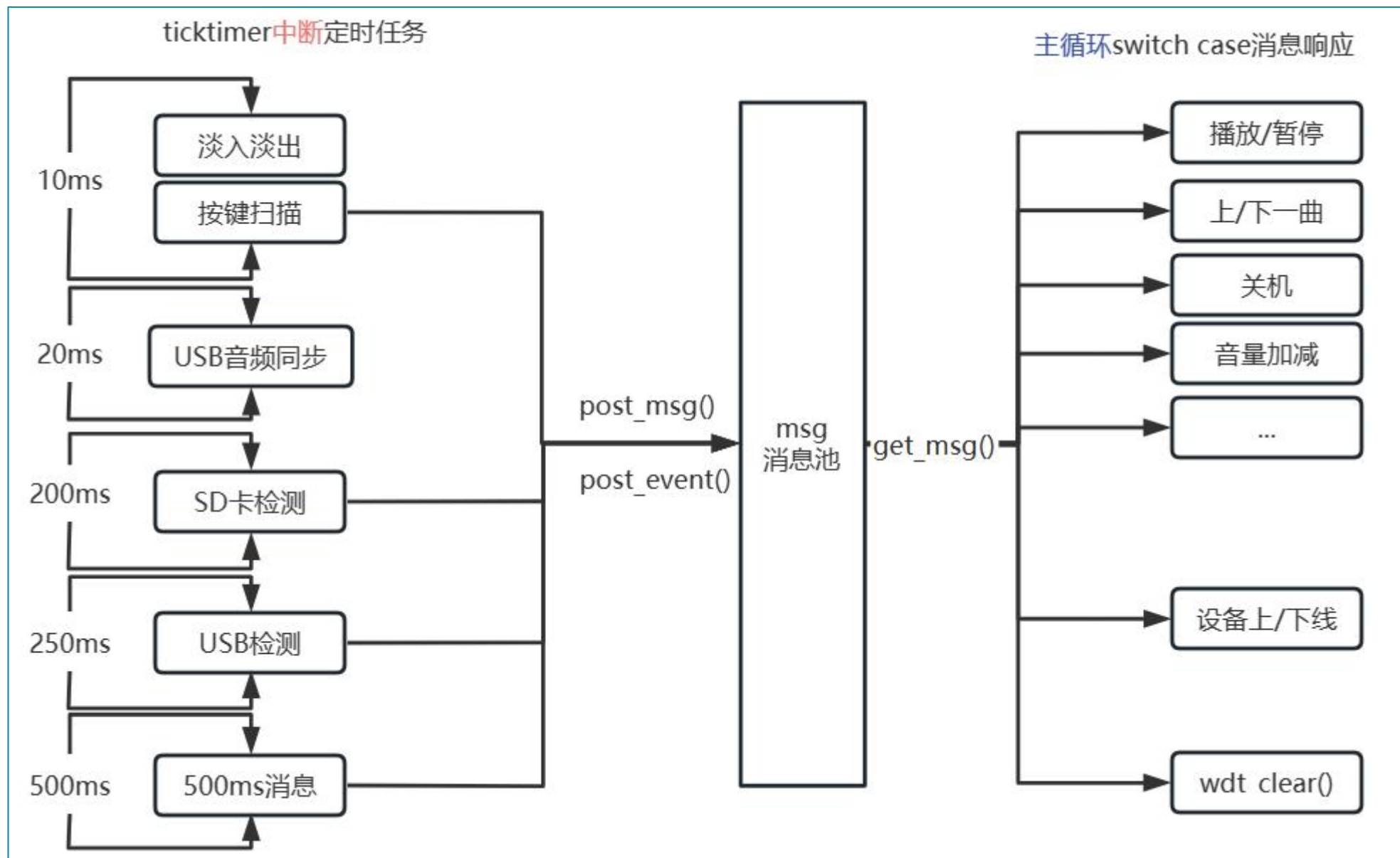
- ◆ toy\_main.c

涉及函数:

- ◆ void app(void);
- ◆ void toy\_music(void);
- ◆ void toy\_midi(void);
- ◆ void toy\_midi\_keyboard(void);
- ◆ void toy\_speaker(void);
- ◆ void toy\_linein(void);
- ◆ void toy\_record(void);
- ◆ void toy\_idle(void);
- ◆ void toy\_usb\_slave(void);



## 二、SDK应用框架介绍 - TickTimer系统节拍的作用



## 二、SDK应用框架介绍 - TickTimer系统节拍的作用

```
app\bsp\modules\timer\tick_timer_sys.c X
73
74 void tick_timer_loop()
75 {
76
77     g_tick_cnt ++;
78
79     sys_tick_timer(g_tick_cnt);
80     app_timer_loop();
81     modules_tick_timer(g_tick_cnt);
82
83     if (0 == (g_tick_cnt % 250)) { //500ms
84         g_tick_cnt = 0;
85     }
86 }
87
88
```

涉及文件:

- ◆ tick\_timer\_sys.c
- ◆ tick\_timer\_modules.c
- ◆ toy\_main.c

涉及函数:

- ◆ void tick\_timer\_loop(void);
- ◆ void tick\_timer\_ram\_loop(void);
- ◆ void app\_timer\_loop(void);

## 二、SDK新增功能介绍 - ANS降噪算法

### 主要函数:

- ◆ `void *link_ans_sound(void *p_sound_out, void *p_ans_obuf, u32 sr)`

### 主要参数:

- ◆ **ANS\_AggressFactor**: 噪声前级动态压制
- ◆ **ANS\_MinSuppress**: 噪声后级动态压制
- ◆ **ANS\_NoiseLevel**: 噪声初始等级

### 涉及文件:

- ◆ `ans_api.c`
- ◆ `NoiseSuppressLib.h`
- ◆ `libNoiseSuppress_pi32_OnChip.a`
- ◆ `lib_SW_FFT_pi32_OnChip.a`



## 二、SDK新增功能介绍 - ANS降噪算法

```
app\bsp\common\sound_effect_list\ans\ans_api.c X
117  /*-----*/
118  void *ans_api(void *obuf, void **ppsound, u32 sr)
119  {
120      const u32 ans_supprt_sr[2] = {8000, 16000};
121      if (sr != ans_supprt_sr[NS_IS_WIDEBAND]) {
122          log_error("ans not support curr sr %d\n", sr);
123          return NULL;
124      }
125      int tolbufsize = NoiseSuppress_QueryBufSize(NS_MODE, NS_IS_WIDEBAND);
126      log_info("tolbufsize %d %d\n", tolbufsize, sizeof(ans_runbuf));
127      ASSERT(ANS_RUN_BUFSIZE >= tolbufsize);
128      int maxtmpbufsize = NoiseSuppress_QueryTempBufSize(NS_MODE, NS_IS_WIDEBAND);
129      log_info("maxtmpbufsize %d %d\n", maxtmpbufsize, sizeof(ans_tmpbuf));
130      ASSERT(ANS_TMP_BUFSIZE >= maxtmpbufsize);
131      int ANS_AggressFactor = (int)(125 * 65536 / 100); /*范围: 1~2, 动态调整, 越大越强(1.25f)*/
132      int ANS_MinSuppress = (int)(10 * 65536 / 100); /*范围: 0~1, 静态定死最小调整, 越小越强(0.1f)*/
133      int ANS_NoiseLevel = (int)(1429 * 1024); /*范围: -100dB ~ -40dB (-75dB) (1429 = (10^(-75/20))*2^23)*/
134      NoiseSuppress_Init(ans_runbuf, ANS_AggressFactor, ANS_MinSuppress, NS_MODE, NS_IS_WIDEBAND, ANS_NoiseLevel);
135      return ans_phy(obuf, ppssound);
136  }
137  /*-----*/
138  /**@brief 录音文件seek函数
```

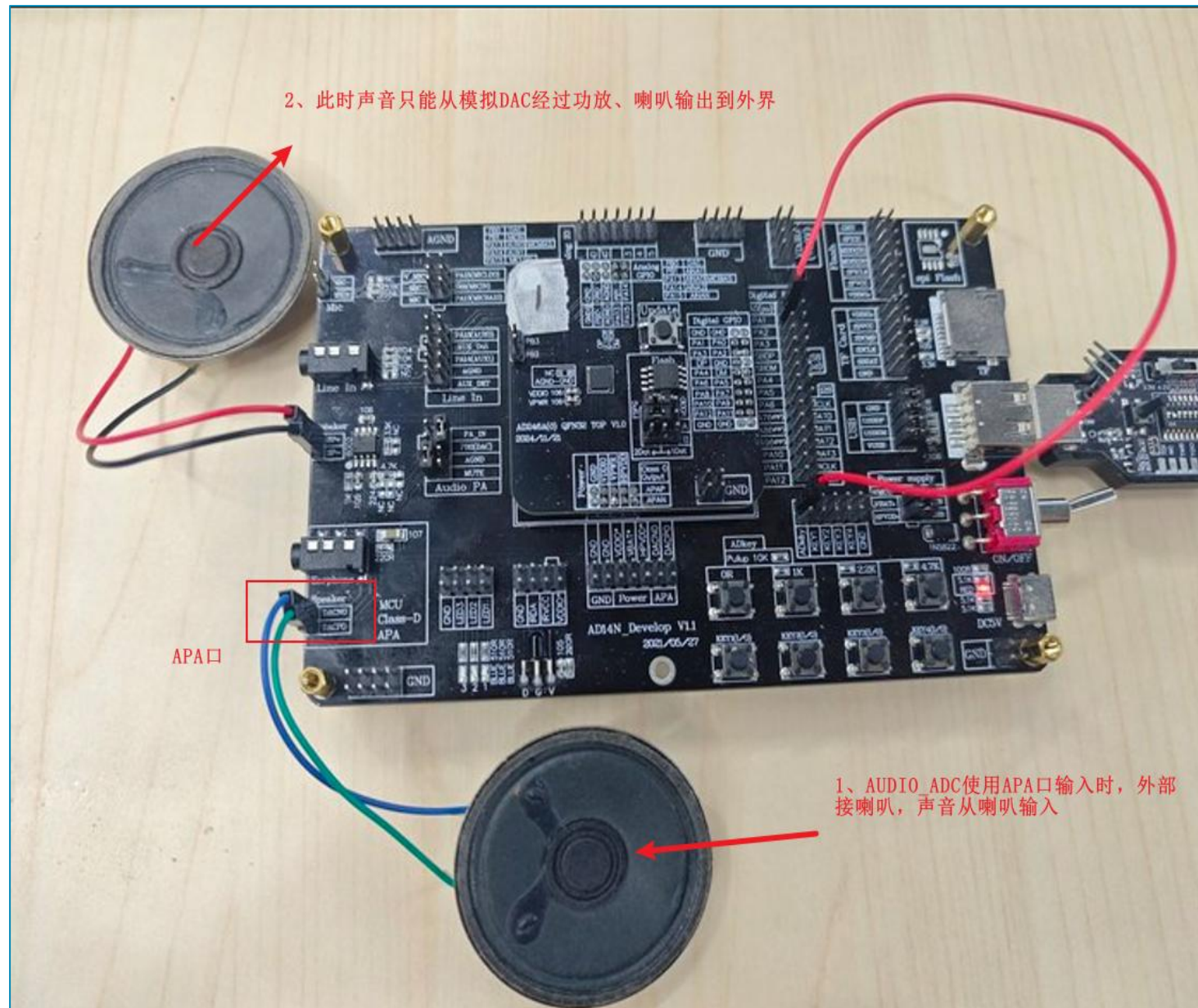
## 二、SDK新增功能介绍 - AUDIO\_ADC使用APA口输入

### ◆ 主要流程

1、关闭APA

2、初始化Audio ADC

3、使能Audio ADC



## 二、SDK新增功能介绍 - AUDIO\_ADC使用APA输入

### 1、关闭APA

- ◆ 调用apa\_close(), 关闭APA功能模块

### 2、初始化Audio ADC

- ◆ 调用函数audio\_adc\_init\_api(sr, ADC\_MIC\_APA, MIC\_INPUT\_ANA3\_APAP)初始化Audio ADC

### 3、使能Audio ADC

- ◆ 调用audio\_adc\_enable(1),使能audio\_adc。

#### ◆ 注:

- ◆ a、在第2步, 需要关闭APA功能, APA口设置成高阻态;
- ◆ b、使用此功能时, APA不能输出、声音只能从模拟DAC输出;



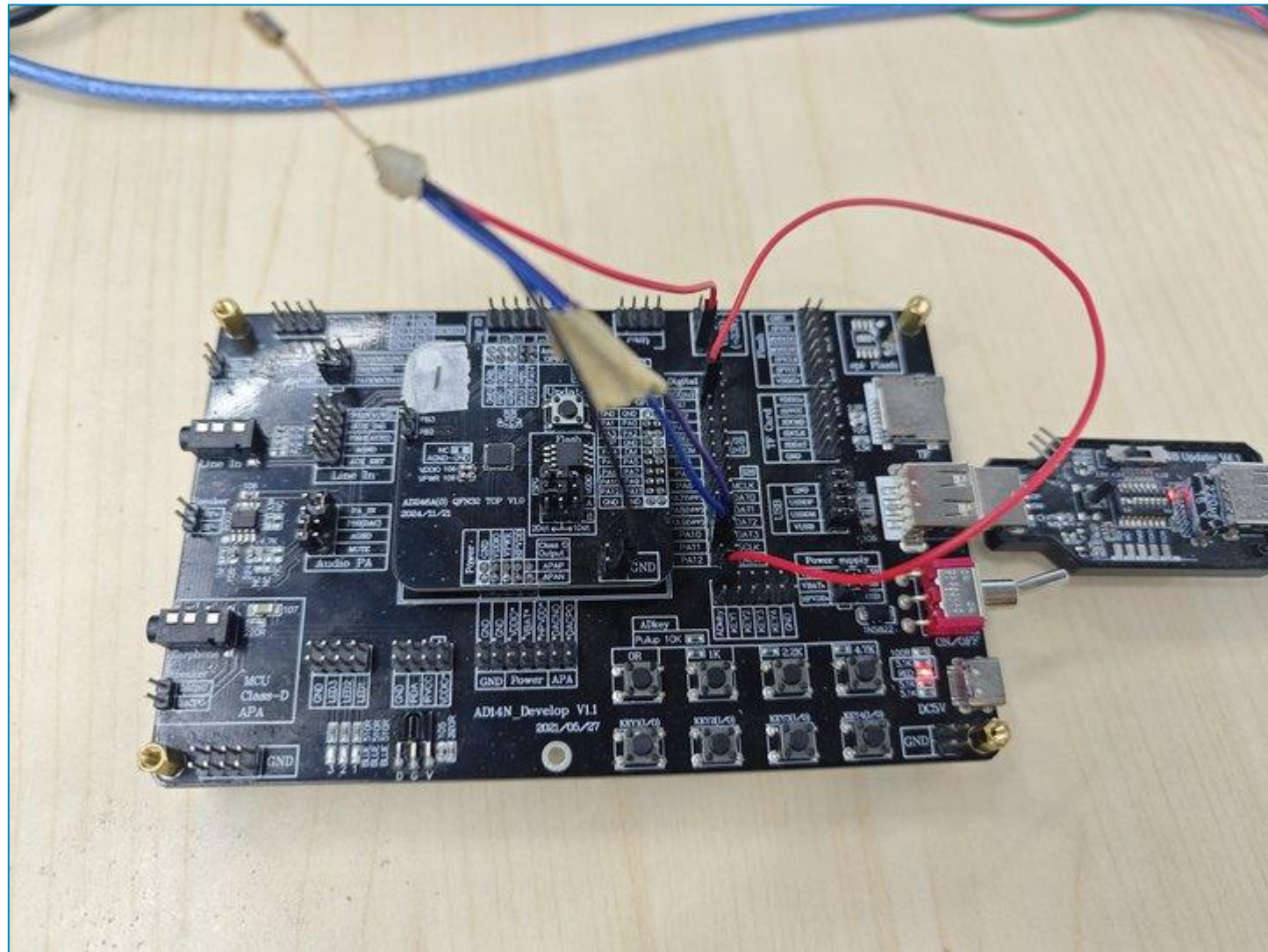
## 二、SDK新增功能介绍 - AUDIO\_ADC使用DMIC

### ◆ DMIC的使用方法

1、选择数字麦引脚和通道

2、初始化Audio ADC

3、使能Audio ADC



## 二、SDK新增功能介绍 - AUDIO\_ADC使用DMIC

### 1、选择数字麦引脚和通道

- ◆ 修改audio\_adc\_dmic\_hdl结构体, 选择需要数字麦使用的引脚和通道

### 2、初始化Audio ADC

- ◆ 调用函数audio\_adc\_init\_api(sr, DIGITAL\_MIC, 0)初始化Audio\_ADC

### 3、使能Audio ADC

- ◆ 调用audio\_adc\_enable(1),使能Audio\_ADC。声音从数字麦输入到Audio\_ADC







---

# 03

## SDK应用配置说明

介绍应用功能配置文件以及其作用

---

# 三、SDK应用配置说明 - 与过往SDK配置文件差异

## ◆ AD24N\_SDK配置文件

- ◆ 1、app\_modules.h
- ◆ 2、app\_config.h
- ◆ 3、app\_config.c
- ◆ 4、cpu\_config.c
- ◆ 5、**board\_cfg.h (新增)**

## ◆ 过往玩具系列SDK配置文件

- ◆ 1、app\_modules.h
- ◆ 2、app\_config.h
- ◆ 3、app\_config.c
- ◆ 4、cpu\_config.c (AW30N\_SDK开始沿用)

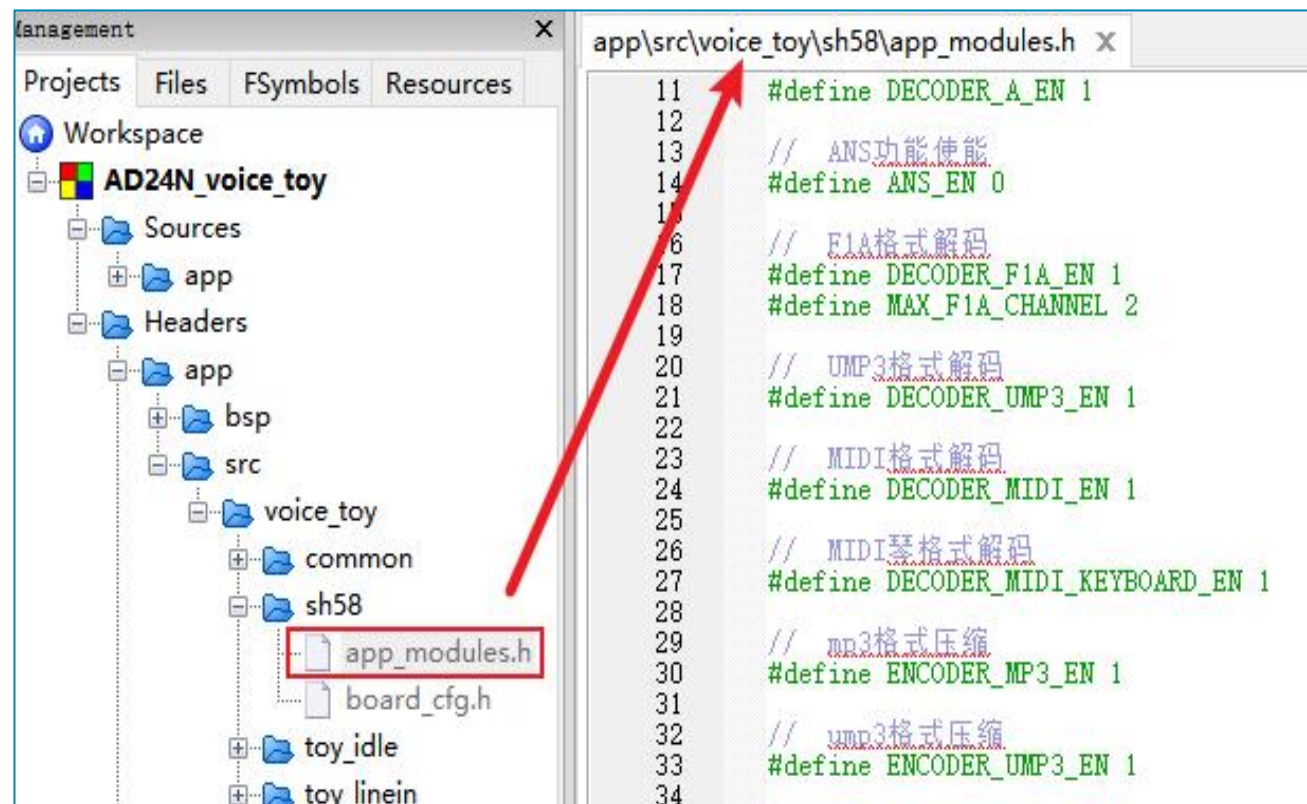
**注：** 新增的 **board\_cfg.h** 文件集合了工程绝大多数的IO配置；

# 三、SDK应用配置说明 - app\_modules.h

## ◆ app\_modules.h 部分配置说明

### ◆ (详情请看SDK文档)

- ◆ 1、各编解码算法、音效算法使能
- ◆ 2、app应用模式使能
- ◆ 3、芯片差异相关外设功能使能
- ◆ 4、isd\_config\_ini.c功能使能



# 三、SDK应用配置说明 - app\_config.h

## ◆ app\_config.h 部分配置说明

## ◆ (详情请看SDK文档)

- ◆ 1、按键音开关
- ◆ 2、各种按键功能开关，如AD、红外、矩阵按键等配置
- ◆ 3、红外驱动选择
- ◆ 4、系统记忆功能驱动选择
- ◆ 5、外挂flash功能配置
- ◆ 6、SD卡功能开关
- ◆ 7、USB模块配置
- ◆ 8、PMU功能配置

```
app\src\voice_toy\app_config.h x
11 //*****
12 //                                     KEY VOICE Configuration
13 //*****
14 #define KEY_VOICE_EN                0
15
16 //*****
17 //                                     KEY Configuration
18 //*****
19 #define KEY_IO_EN                    0    ///
```



# 三、SDK应用配置说明 - app\_config.c

## ◆ app\_config.c 部分配置说明

### ◆ (详情请看SDK文档)

- ◆ 1、中断优先级配置
- ◆ 2、系统flash\_dtr功能配置
- ◆ 3、malloc内部断言配置
- ◆ 4、midi与midi琴相关配置
- ◆ 5、sd挂起时间配置
- ◆ 6、异常配置
- ◆ 7、通用定时器功能配置
- ◆ 8、打印开关

```
app\src\voice_toy\app_config.c X
4 //系统使用到的
5 //*****
6 //                               ISR Configuration                               //
7 //*****
8 const int IRQ_IRTMR_IP = 5;
9 const int IRQ_AUDIO_IP = 4;
10 const int IRQ_AUDAC_IP = 4;
11 const int IRQ_AUAPA_IP = 4;
12 const int IRQ_AUADC_IP = 4;
13 const int IRQ_DECODER_IP = 1;
14 const int IRQ_WFILE_IP = 1;
15 const int IRQ_ADC_IP = 2;
16 const int IRQ_ENCODER_IP = 0;
17 const int IRQ_TICKTMR_IP = 3;
18 const int IRQ_USB_IP = 2;
19 const int IRQ_SD_IP = 3;
20 const int IRQ_PMU_TIMER1_IP = 0;
21 //系统还未使用到的
22 const int IRQ_UART0_IP = 3;
23 const int IRQ_UART1_IP = 3;
24 const int IRQ_ALINK0_IP = 3;
25
26 //*****
27 //                               vm_sfc Configuration                               //
28 //*****
29 const u8 config_spi_code_user_cache = 1; //sfc放code区
30
31 #if (CONFIG_FLASH_DTR_EN)
32 const u8 sfc0_dtr_mode_en = 1;
33 #else
34 const u8 sfc0_dtr_mode_en = 0;
35 #endif
36 const s8 sfc0_dtr_dummy_num = -1; //查看flash文档填入ini对应线宽的dummy
37 const u32 sfc0_dtr_clk_freq = 192000000; //查看flash文档*2
38 const u8 sfc0_continue_mode_en = 0;
39
```



# 三、SDK应用配置说明 - cpu\_config.c

## ◆ cpu\_config.c 部分配置说明

### ◆ (详情请看SDK文档)

- ◆ 1、audio模块高/低电压模式配置
- ◆ 2、audio模块工作依赖常量配置
- ◆ 3、audio\_adc配置

```
app\src\voice_toy\sh58\cpu_config.c X
1      #include "typedef.h"
2
3      /* *****
4      * 在不同供电场景下config_adda_voltage_mode配置不同的值:
5      * 0: 高压供电vbat, 工作电压在2.7v ~ 5.4v之间
6      * 1: 低压供电到ioydd&vbat, 工作电压在1.8v ~ 3.6v之间
7      * 注: 低压供电时, ioydd需要和vbat短接;
8      * 注: 此项配置只会影响到audio dac & audio adc的性能; 不会对APA性能产生影响
9      * *****
10     const unsigned char config_adda_low_voltage_mode = 0;
11
12
13     //*****
14     //                                AUDIO Configuration                                //
15     //*****
16     /*
17     +-----+-----+-----+-----+
18     | AD24N AUDIO & APA 工作依赖常量 |
19     +-----+-----+-----+-----+
20     |                               | Audio DAC | Audio ADC | Audio APA |
21     +-----+-----+-----+-----+
22     | au_const_apa_en                | x         | x         | ✓         |
23     +-----+-----+-----+-----+
24     | au_const_dac_digital_en        | ✓         | x         | ✓         |
25     +-----+-----+-----+-----+
26     | au_const_dac_analog_en         | ✓         | x         | x         |
27     +-----+-----+-----+-----+
28     | au_const_adda_common_en        | ✓         | ✓         | x         |
29     +-----+-----+-----+-----+
30     */
31     const u8 au_const_apa_en = 1;
32     const u8 au_const_dac_digital_en = 1;
33     const u8 au_const_dac_analog_en = 1;
34     const u8 au_const_adda_common_en = 1;
35
```

# 三、SDK应用配置说明 - board\_cfg.h

## ◆ board\_cfg.h 部分配置说明

## ◆ (详情请看SDK文档)

- ◆ 1、串口打印IO配置
- ◆ 2、按键模块IO配置
- ◆ 3、唤醒IO配置
- ◆ 4、外挂资源flashIO配置
- ◆ 5、SD模块IO配置

```
app\src\voice_toy\sh58\board_cfg.h X
4
5 //*****
6 //                                UART Configuration
7 //*****
8 #define UART_OUTPUT_CH_PORT      IO_PORTA_04
9
10
11 //*****
12 //                                KEY Configuration
13 //*****
14 //AD KEY
15 #define AD_KEY_CH_SEL            ADC_CH_PA1
16 //IR KEY
17 #define IR_KEY_IO_SEL           IO_PORTA_09
18 #define TCFG_ADKEY_IR_IO_REUSE  DISABLE//ADKEY 和 红外IO复用
19 //MATRIX KEY
20 ///X轴 io 要求是AD口, 详细AD口看adc_drv.h
21 #define X_ADC_CH_SEL             {ADC_CH_PA2, ADC_CH_PA3, ADC_CH_PA5}
22 ///Y轴 io 要求是普通IO口
23 #define Y_PORT_SEL              {IO_PORTA_10, IO_PORTA_11}
24 //TOUCH KEY
25 #define TOUCH_KEY_SEL           {IO_PORTA_09, IO_PORTA_10, IO_PORTA_11}
26
27
28 //*****
29 //                                POWER WAKEUP IO Configuration
30 //*****
31 #define POWER_WAKEUP_IO         IO_PORTA_01
32
33
34 //*****
35 //                                EXFLASH PORT Configuration
36 //*****
37 //port select for hardware spi
38 //support any io
39 #define EXFLASH_CS_PORT_SEL      IO_PORTA_05
40 #define EXFLASH_CLK_PORT_SEL    IO_PORTA_11
41 #define EXFLASH_D0_PORT_SEL     IO_PORTA_12
42 #define EXFLASH_D1_PORT_SEL     IO_PORTA_10
43
```

# 三、SDK应用配置说明 - AUDIO ADC

## ◆ 1、AUDIO\_ADC数字麦相关配置

```
app\bsp\cpu\sh58\audio_adc_cpu.c x
49     };
50
51
52     u8 const audio_adc_dmic_use_ch = 4; // adc0 mic sel 0~3:不使用
53     AUDIO_ADC_DIG_MIC_HDL audio_adc_dmic_hdl = {
54         .clk_io = IO_PORTA_10, //支持output channel映射
55         .data_io = IO_PORTA_11, //支持output channel映射
56         .data_ch = PORT_FUNC_ICH_PLNK_DAT0, //对应数字麦0
57         /* .data_ch = PORT_FUNC_ICH_PLNK_DAT1, //对应数字麦1 */
58         /* .enable = 0, */
59     };
60
```

# 三、SDK应用配置说明 - AUDIO DAC & APA

## ◆ 2、DAC&APA采样率相关配置

```
include_lib\cpu\sh58\audio_dac_cpu.h x
66     DAC0_CHPHSET  |\
67     DAC0_DITHER(3) \
68 )
69
70 #define DAC_CON1_DEFAULT \
71 ( \
72     DAC1_DFIFO_SYNC \
73 )
74
75 #define SR_DEFAULT 32000
76
77
78
```

## ◆ 3、APA调制模式配置

```
include_lib\cpu\sh58\audio_apa_cpu.h x
30 #define APA0_DIFF_SNR (1 << 3)
31 #define APA0_DIFF_THD (2 << 3)
32
33 #define APA1_IIR_PAR(n) ((n & 0xf) << 20)
34
35 /*****
36 #define APA_PWM_192M_EN 1
37 #define APA_DSM_CLK_MODE 0 // 0: 384kHz 352.9kHz low clk
38 // #define APA_PWM_MODE APA0_DIFF_THD
39 #define APA_PWM_MODE APA0_DIFF_SNR //开发板硬件默认为差分
40
41 #define APA_CON0_DEFAULT (APA_PWM_MODE)
42
43 typedef struct _APA_PHY_PARA {
44
```



---


# 04 应用开发注意事项

介绍开发、调试、生产注意事项

---



## 四、应用开发注意事项

- 
- ◆ 开发使用的样片，必须经过我司一拖二烧写器进行校准

- ◆ 烧写器需要对芯片内部LRC时钟做校准，该项在芯片出厂阶段不做校准；



# 谢谢观看

AD系列32位芯片 杰理开源社区 [服务](#)



此二维码365天内有效 (2025-10-08前)

 钉钉扫一扫群二维码，立即加入群聊