

# ESP32-A1S ES8388 版本 替换 AC101 指导说明

版本 V1.0

版权 ©2021

## 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。

文中所得测试数据均为实验室测试所得，实际结果可能略有差异。

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。

文件制定/修订/废止履历表

版本	日期	制定/修订内容	制定	核准
V1.0	2021.6.03	首次制定		

目录

一、硬件区别.....4

二、软件区别.....4

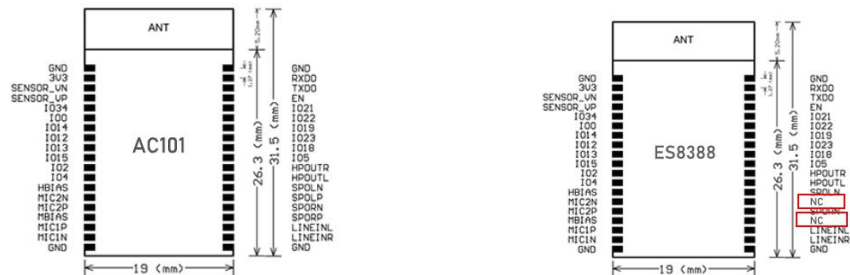
    1. 区别说明.....4

    2.适配 V2.3 模组的 SDK 配置流程..... 5

联系我们.....7

## 一、硬件区别

ESP32-A1S 的音频编解码芯片由 AC101 替换为 ES8388 之后,ESP32-A1S 模组的版本也从 ESP32-A1S-V1.3 更新到了 ESP32-A1S-V2.3。V2.3 版的模组接口依旧兼容 V1.3 版的接口。但是依旧有些差别:



V2.3(ES8388 版)中的红框部分原是 V1.3(AC101 版)的 SPORN 和 SPOLP, 取消掉之后不会影响使用, 因为 HPOUTR 和 HPOUTL 就是左右声道的 P 极, 它们的对应关系: **HPOUTR** 对应的是 **SPORN**; **HPOUTL** 对应 **SPOLP**。

因此 V2.3 版的模组音频输出为:

右声道输出是: HPOUTR (正) 和 SPORN (负);

左声道输出是: HPOUTL (正) 和 SPOLN (负)。

补充:

- 1、8388 的 Linein 和 MIC2 是共用的, 两者不能同时使用。
- 2、MIC1 和 MIC2 都可以当 linein 使用, 软件调整配置即可

## 二、软件区别

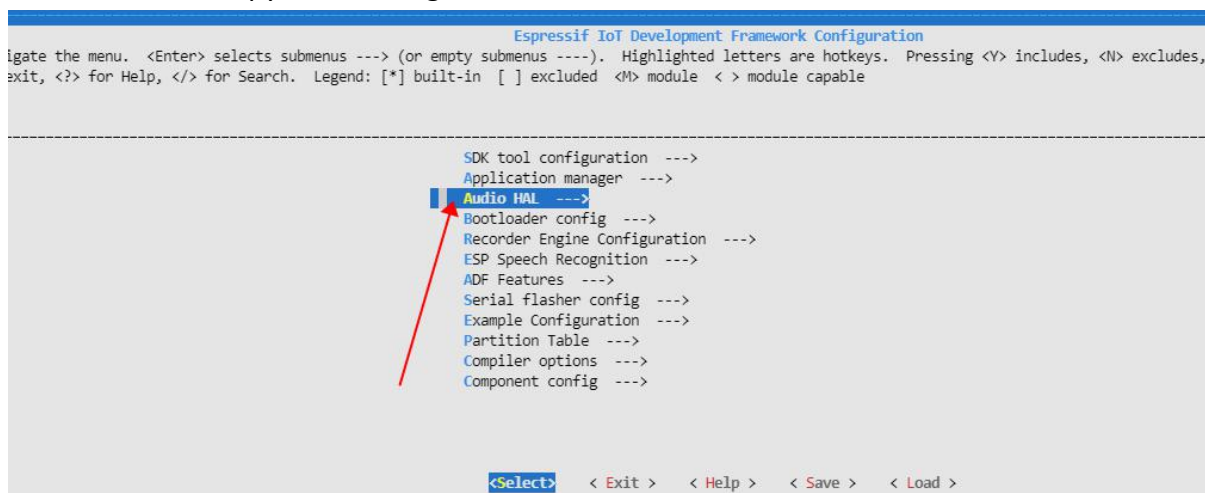
### 1. 区别说明

乐鑫所提供的 SDK 中的例程都是以板卡的基础上运行的。之前的 AC101 芯片, 乐鑫 SDK 并没有相应的驱动组件, 因此我们都是需要写 AC101 的 IIC 通信驱动, 无疑为开发者增加开发难度。

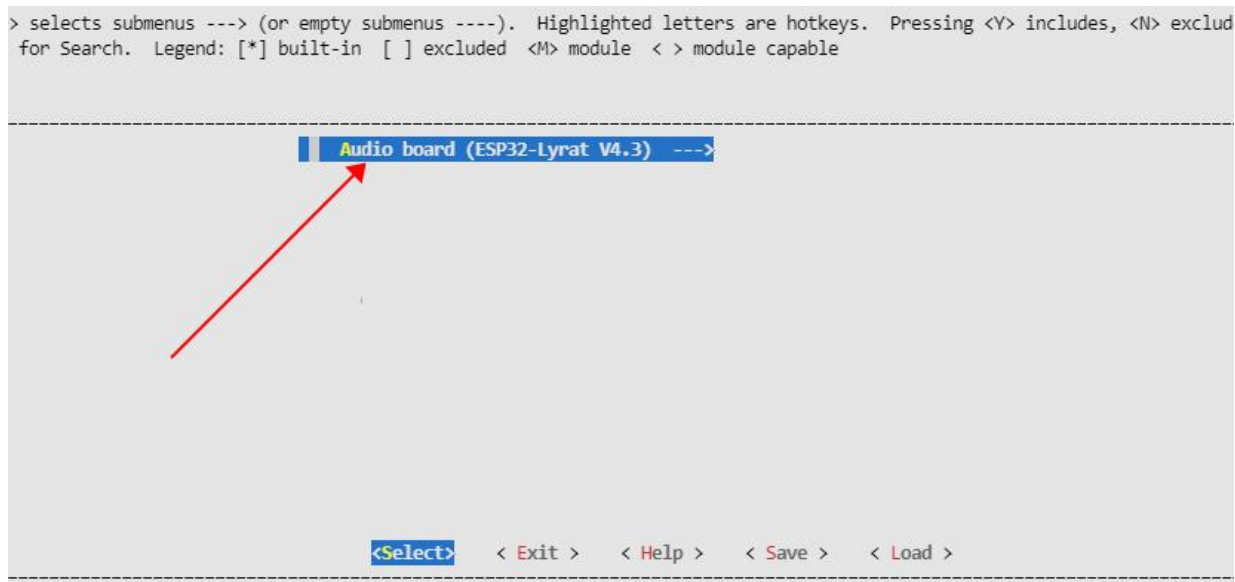
而 V2.3 版本更换成了 ES8388 之后, 可以直接使用乐鑫写好的驱动, 我们只需要适配好板卡之后, 开发者甚至可以通过 menuconfig 配置界面来选择 ESP32-A1S-Audio-Kit V2.2 开发板。选择好开发板之后, 就可以运行 SDK 中的例程了。

## 2. 适配 V2.3 模组的 SDK 配置流程

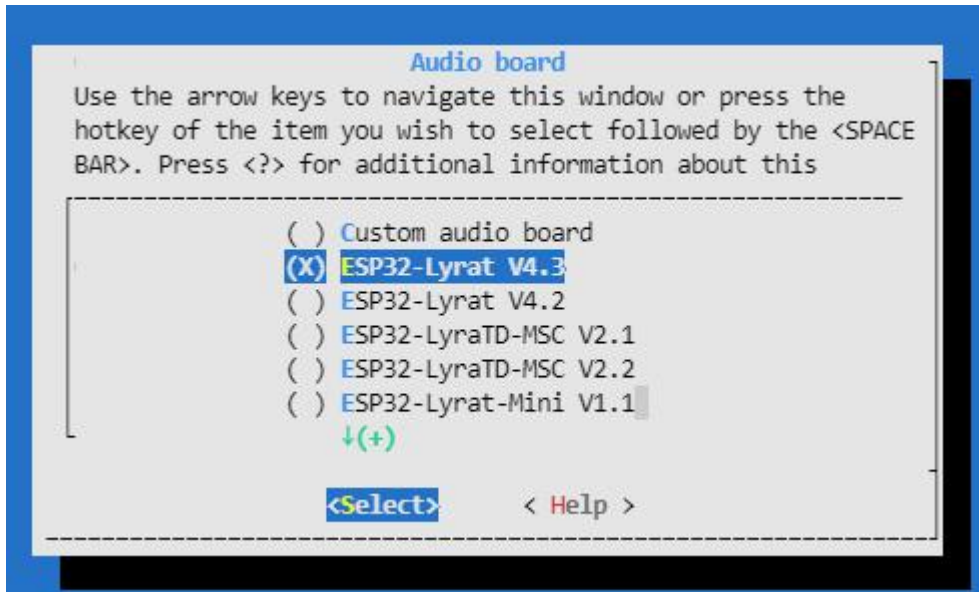
首先搭建完毕 esp-adf 开发环境, SDK 源码环境: <https://github.com/espressif/esp-adf> 之后, 运行: idf.py menuconfig



进入: Audio HAL ... 选项



选择: ESP32-Lyrat V4.3



修改 board\_pins\_config.c 代码，路径：

[esp-adf/components/audio\\_board/lyrat\\_v4\\_3/board\\_pins\\_config.c](#)

```
esp_err_t get_i2c_pins(i2c_port_t port, i2c_config_t *i2c_config)
{
    AUDIO_NULL_CHECK(TAG, i2c_config, return ESP_FAIL);
    if (port == I2C_NUM_0 || port == I2C_NUM_1) {
        i2c_config->sda_io_num = GPIO_NUM_33;
        i2c_config->scl_io_num = GPIO_NUM_32;
    } else {
        i2c_config->sda_io_num = -1;
        i2c_config->scl_io_num = -1;
        ESP_LOGE(TAG, "i2c port %d is not supported", port);
        return ESP_FAIL;
    }
    return ESP_OK;
}

esp_err_t get_i2s_pins(i2s_port_t port, i2s_pin_config_t *i2s_config)
{
    AUDIO_NULL_CHECK(TAG, i2s_config, return ESP_FAIL);
    if (port == I2S_NUM_0 || port == I2S_NUM_1) {
        i2s_config->bck_io_num = GPIO_NUM_27;
        i2s_config->ws_io_num = GPIO_NUM_25;
        i2s_config->data_out_num = GPIO_NUM_26;
        i2s_config->data_in_num = GPIO_NUM_35;
    } else {
        memset(i2s_config, -1, sizeof(i2s_pin_config_t));
        ESP_LOGE(TAG, "i2s port %d is not supported", port);
        return ESP_FAIL;
    }
    return ESP_OK;
}
```

修改I2C 端口

修改I2S端口

保存退出即可编译。

