



# Linux 音效调优 开发指南

版本号: 1.0  
发布日期: 2025.01.09

## 版本历史

版本号	日期	制/修订人	内容描述
1.0	2025.01.09	AWA1052	初始化版本



# 目 录

<b>1 前言</b>	<b>1</b>
1.1 编写目的	1
1.2 适用范围	1
1.3 相关人员	1
1.4 名词解析	1
<b>2 音效配置</b>	<b>2</b>
2.1 EQ 算法	2
2.2 Prescale	4
2.3 DRC 算法	4
2.4 调试控件介绍	6
2.5 ROUTE 算法	8



# 1 前言

## 1.1 编写目的

本文档目的是让开发者了解 H135/H136 的基本音效配置和使用方法。

## 1.2 适用范围

本开发指南适用于 H135/H136 平台。

## 1.3 相关人员

音频系统开发人员。

## 1.4 名词解析

- DRC: Dynamic Range Control 动态范围控制
- EQ: Equalizer 均衡器
- ROUTE: 路由, 用于指定声道的映射关系

## 2 音效配置

默认的音频配置文件路径位于/etc/cedarSE 目录下，其中整体的算法配置文件为audio\_device\_desc.xml，算法配置目录结构如下所示：

```
.
├── audio_device_desc.xml
├── audio_hw_drc_param.xml
├── audio_sw_eq_param.xml
├── audio_sw_route_param.xml
├── audio_sw_rtc_agc_param.xml
└── audio_sw_volume_param.xml
```

H135/H136 默认以名为 “Speaker” 的设备作为音频算法的输出设备。

```
<play_device type="Speaker" setup_gap_time="100">
  <sw_effect sw="ROUTE"/>
  <sw_effect sw="VOLUME"/>
  <sw_effect sw="EQ"/>
  <sw_effect sw="RTC_AGC"/>
  <hw_effect hw="HW_DRC"/>
</play_device>
```

如上表明 Speaker 分别经过了 ROUTE、VOLUME、EQ、RTC\_AGC和HW\_DRC的处理。每个算法开头处会有 enable 的属性，来控制是否开启该算法。

```
<param_tree>
  <param path="Speaker" param_id="0" enable="1"/>
</param_tree>
```

### 2.1 EQ 算法

EQ 算法配置文件为audio\_sw\_eq\_param.xml，配置文件结构如下所示：

```
<param_tree>
  <param path="Speaker" param_id="0" enable="1"/>
  <param path="Speaker standard" param_id="1" enable="0"/>
  <param path="Speaker music" param_id="2" enable="0"/>
  <param path="Speaker movie" param_id="3" enable="0"/>
  <param path="Speaker game" param_id="4" enable="0"/>
  <param path="Speaker sport" param_id="5" enable="0"/>
  <param path="Speaker custom" param_id="6" enable="0"/>
</param_tree>
<param_unit_pool>
  <param_unit param_id="0">
    <!-- set "1" then use different param per channel -->
    <param name="diff_param" value="0"/>
  </param_unit>
</param_unit_pool>
```

```

<param name="channels" value="2"/>
<param name="bin_num" value="5"/>
<param name="samplerate" value="48000"/>
<!-- filtertype fc gain quailty enable -->
<param name="BQ1" value="0,300,0,0,1"/>
<param name="BQ2" value="1,500,0,2,1"/>
<param name="BQ3" value="1,1000,0,2,1"/>
<param name="BQ4" value="1,2000,0,1,1"/>
<param name="BQ5" value="2,4000,0,0,1"/>
</param_unit>
....

```

其中 Speaker 为当前的配置，enable 设置为 1，其余的配置如 “Speaker standard” 等分别表示的是不同的 EQ 模式，包括 standard,music,movie, game,sport,custom 等。当音频模式切换时，音频中间件会拷贝对应的 EQ 配置到 Speaker 中，从而让其生效。

如上图所示，param\_id= “0” 的即对应的是 Speaker 的配置，其中 channels 表示的是通道数，bin\_num 表示的是滤波器的个数，samplerate 表示的是采样率，BQ1-BQ5 表示的是 5 个滤波器的配置，其中：

- filtertype 表示的是滤波器类型，0 表示低架，1 表示频点，2 表示高架
- fc 表示的是中心频率，如上分别为 300Hz,500Hz,1000Hz,2000Hz,4000Hz 共 5 组频点。
- gain 表示的是增益，应用可以通过修改 gain 值来调整 EQ 效果，范围可以设置-20 到 20。
- quailty 表示的是质量因子
- enable 表示的是是否开启该滤波器

以应用 lv Projector 为例，可以使用如下接口获取和设置 EQ 的参数：

```

// 需要引用头文件
#include "audio_hw.h"

int audio_set_eq_bass_gain(int8_t gain) {
    int ret = -1;
    if (adev == NULL) {
        ret = adev_open(&adev);
        if (ret < 0) {
            printf("adev_open failed\n");
            return ret;
        }
    }
    int channels = 0;
    int sample_rate = 0;
    // custom eq is fix to 5 bins.
    int bin_num = 5;
    eq_core_param *params = (eq_core_param *)malloc(sizeof(eq_core_param) * 5);
    // 获取当前的EQ配置参数
    ret = adev_get_eq_params(adev, &channels, &sample_rate, &params, bin_num);
    if (ret < 0) {
        printf("%s get eq failed\n", __func__);
        return ret;
    }
    // 设置低频段的增益，如有需要，可以修改其他频段的增益

    params[0].gain = gain * 2;
    params[1].gain = gain * 2;

```

```
ret = adev_set_eq_params(adev, channels, sample_rate, params, bin_num);
if (ret < 0) {
    printf("%s set eq failed\n", __func__);
    return ret;
}
free(params);
return ret;
}
```

上述调用方式需要引用 libaudioroute 库作为依赖，并且使用 audio\_hw.h 头文件。

## 2.2 Prescale

Prescale 前置增益可以利用 EQ 算法实现，即修改 EQ 的 gain 参数来实现，如需要增加 10db 增益，可以设置 EQ 的 gain 为 10 即可，gain 的范围是-100 DB 到 100 DB。

```
<param_unit param_id="0">
  <!-- set "1" then use diffrent param per channel -->
  <param name="diff_param" value="0"/>
  <param name="channels" value="2"/>
  <param name="bin_num" value="5"/>
  <param name="samplerate" value="48000"/>
  <!-- filtertype fc gain quailty enable -->
  <param name="BQ1" value="0,300,10,0,1"/>
  <param name="BQ2" value="1,500,10,2,1"/>
  <param name="BQ3" value="1,1000,10,2,1"/>
  <param name="BQ4" value="1,2000,10,1,1"/>
  <param name="BQ5" value="2,4000,10,0,1"/>
</param_unit>
```

## 2.3 DRC 算法

DRC(动态范围控制器) 是一种可以根据输入信号幅度范围对输出信号幅度进行动态控制的模块或设备。动态范围控制器主要用于音频信号的柔和处理和幅度限制，以取得良好听音效果，并防止声音过大引起削波失真。DRC 由 SoC 内部 codec 硬件实现，具体的描述请参考对应芯片平台的 User Manual

DRC 的配置文件如下所示，其中分为 DRC 参数部分和 DRC 寄存器部分。

```
<?xml version="1.0" encoding="UTF-8"?>
<audio_hw_drc_param version="2.10.0-later">
  <param_tree>
    <param path="Speaker" sub_id="0" param_id="0" enable="1"/>
  </param_tree>
  <param_unit_record>
    <param_unit param_id="0">
      <param name="channels" value="1,2"/>
      <param name="ET_EN" value="1"/>
      <param name="LT_EN" value="1"/>
      <param name="Detect_Noise" value="1"/>
    </param_unit>
  </param_unit_record>
</audio_hw_drc_param>
```

```
<param name="Max_Gain_Limit" value="1"/>
<param name="Min_Gain_Limit" value="1"/>
<param name="Max_Gain" value="20"/>
<param name="Min_Gain" value="-40"/>
<param name="Ke" value="1"/>
<param name="Kn" value="1"/>
<param name="Kc" value="1"/>
<param name="Kl" value="0.5"/>
<param name="ET" value="-54.66667"/>
<param name="OPE" value="-54.66667"/>
<param name="CT" value="-33.33333"/>
<param name="OPC" value="-33.33333"/>
<param name="LT" value="-4.666667"/>
<param name="OPL" value="-4.666667"/>
<param name="X" value="-100"/>
<param name="Y" value="-2.333333"/>
<param name="Peak Filter" value="1"/>
<param name="Left Attack Time" value="5"/>
<param name="Left Release Time" value="100"/>
<param name="Right Attack Time" value="5"/>
<param name="Right Release Time" value="100"/>
<param name="Left Average Time" value="10"/>
<param name="Right Average Time" value="10"/>
<param name="Smooth Attack Time" value="10"/>
<param name="Smooth Release Time" value="200"/>
<param name="Expander Smooth Time" value="5"/>
<param name="Optimum" value="17"/>
</param_unit>
</param_unit_record>
<param_unit_pool>
<param_unit param_id="0">
<param addr="0x00F0" mask="0xA0000000" value="0xA0000000"/>
<param addr="0x0108" mask="0xFFFFFFFF" value="0x000000FB"/>
<param addr="0x010C" mask="0xFFFFFFFF" value="0x00000002"/>
<param addr="0x0110" mask="0xFFFFFFFF" value="0x00005600"/>
<param addr="0x0114" mask="0xFFFFFFFF" value="0x00000002"/>
<param addr="0x0118" mask="0xFFFFFFFF" value="0x00005600"/>
<param addr="0x011C" mask="0xFFFFFFFF" value="0x000000FF"/>
<param addr="0x0120" mask="0xFFFFFFFF" value="0x0000E1F8"/>
<param addr="0x0124" mask="0xFFFFFFFF" value="0x000000FF"/>
<param addr="0x0128" mask="0xFFFFFFFF" value="0x0000E1F8"/>
<param addr="0x012C" mask="0xFFFFFFFF" value="0x00000001"/>
<param addr="0x0130" mask="0xFFFFFFFF" value="0x00002BB0"/>
<param addr="0x0134" mask="0xFFFFFFFF" value="0x00000001"/>
<param addr="0x0138" mask="0xFFFFFFFF" value="0x00002BB0"/>
<param addr="0x013C" mask="0xFFFFFFFF" value="0x00000589"/>
<param addr="0x0140" mask="0xFFFFFFFF" value="0x00005B1D"/>
<param addr="0x0144" mask="0xFFFFFFFF" value="0x00000100"/>
<param addr="0x0148" mask="0xFFFFFFFF" value="0x00000000"/>
<param addr="0x014C" mask="0xFFFFFFFF" value="0x0000FA76"/>
<param addr="0x0150" mask="0xFFFFFFFF" value="0x0000A4E3"/>
<param addr="0x0154" mask="0xFFFFFFFF" value="0x000000C6"/>
<param addr="0x0158" mask="0xFFFFFFFF" value="0x00006E09"/>
<param addr="0x015C" mask="0xFFFFFFFF" value="0x00000080"/>
<param addr="0x0160" mask="0xFFFFFFFF" value="0x00000000"/>
<param addr="0x0164" mask="0xFFFFFFFF" value="0x0000FF39"/>
<param addr="0x0168" mask="0xFFFFFFFF" value="0x000091F7"/>
<param addr="0x016C" mask="0xFFFFFFFF" value="0x00000914"/>
<param addr="0x0170" mask="0xFFFFFFFF" value="0x000076BE"/>
<param addr="0x0174" mask="0xFFFFFFFF" value="0x00000100"/>
```

```

<param addr="0x0178" mask="0xFFFFFFFF" value="0x00000000"/>
<param addr="0x017C" mask="0xFFFFFFFF" value="0x0000F6EB"/>
<param addr="0x0180" mask="0xFFFFFFFF" value="0x00008942"/>
<param addr="0x0184" mask="0xFFFFFFFF" value="0x00000100"/>
<param addr="0x0188" mask="0xFFFFFFFF" value="0x00000000"/>
<param addr="0x018C" mask="0xFFFFFFFF" value="0x00000001"/>
<param addr="0x0190" mask="0xFFFFFFFF" value="0x00002BB0"/>
<param addr="0x0194" mask="0xFFFFFFFF" value="0x00000000"/>
<param addr="0x0198" mask="0xFFFFFFFF" value="0x00000F04"/>
<param addr="0x019C" mask="0xFFFFFFFF" value="0x00000352"/>
<param addr="0x01A0" mask="0xFFFFFFFF" value="0x000069E0"/>
<param addr="0x01A4" mask="0xFFFFFFFF" value="0x0000F95B"/>
<param addr="0x01A8" mask="0xFFFFFFFF" value="0x00002C3F"/>
<param addr="0x01AC" mask="0xFFFFFFFF" value="0x00000002"/>
<param addr="0x01B0" mask="0xFFFFFFFF" value="0x00005600"/>
</param_unit>
</param_unit_pool>
</audio_hw_drc_param>
    
```

param addr 中的值为寄存器地址，mask 为掩码，value 为对应的值，该值是通过工具 TigerAudio 生成的，不可随意修改。

TigerAudio 关于调节 DRC 的界面如下所示：

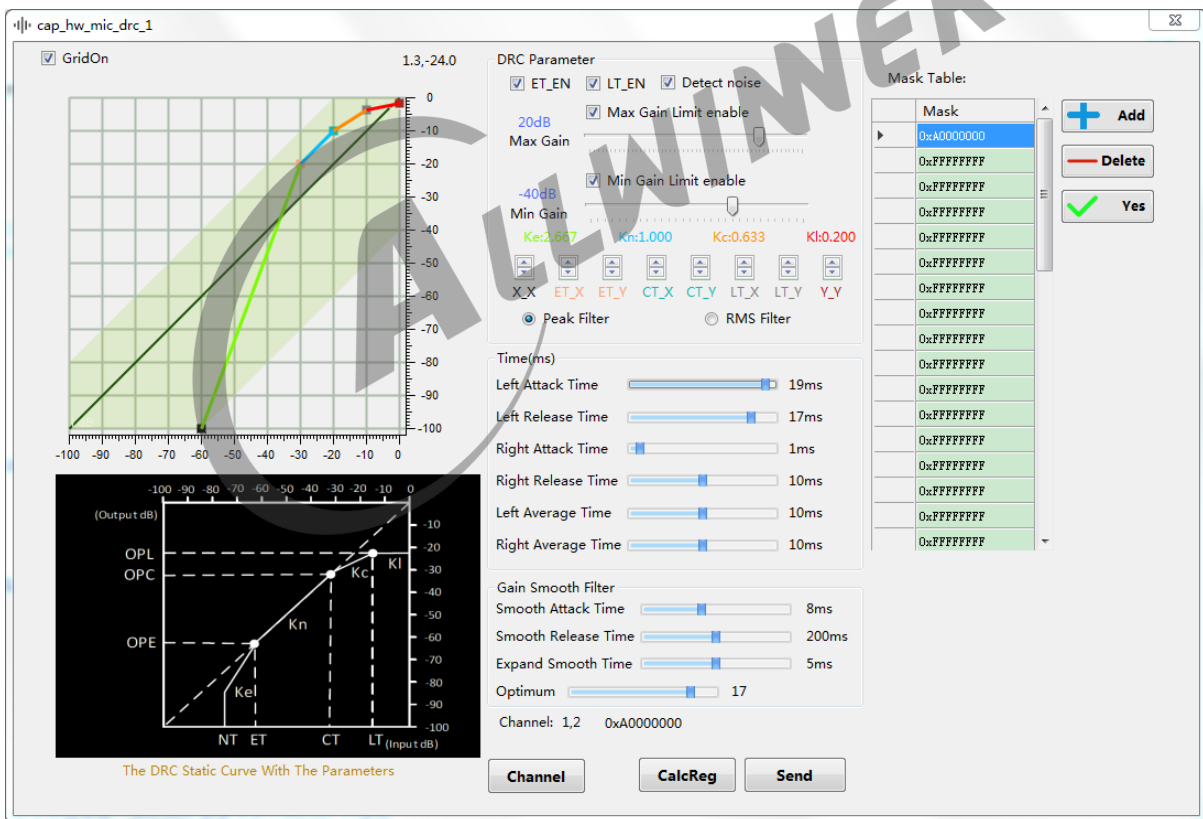


图 2-1: TigerAudio

## 2.4 调试控件介绍

操作控件	操作内容
ET_EN	设置是否使能 ET 这个点
LT_EN	设置是否使能 LT 这个点
Detect noise	是否使能噪声检测（只有在 ET 使能时才有效），默认阈值为 -120dB
Max Gain Limit enable	设置是否使能限制增益的最大值
Min Gain Limit enable	设置是否使能限制增益的最小值，使能时会覆盖 “Detect noise” 的作用
Max Gain	设置增益的最大值
Min Gain	设置增益的最小值
X_X	设置横坐标上的点
ET_X	设置 ET 对应点的横坐标
ET_Y	设置 ET 对应点的纵坐标
CT_X	设置 CT 对应点的横坐标
CT_Y	设置 CT 对应点的纵坐标
LT_X	设置 LT 对应点的横坐标
LT_Y	设置 LT 对应点的纵坐标
Y_Y	设置纵坐标上的点
Peak Filter	设置峰值滤波器作为能量滤波器，选择后可设置 AT_L RT_L AT_R RT_R
RMS Filter	设置均方根滤波器作为能量滤波器，选择后可设置 Aver_L Aver_R
AT_L	设置左声道 Attack Time
RT_L	设置左声道 Release Time
AT_R	设置右声道 Attack Time
RT_R	设置右声道 Release Time
Aver_L	设置左声道 Average Time
Aver_R	设置右声道 Average Time
AT	设置增益平滑滤波器的 Attack Time
RT	设置增益平滑滤波器的 Release Time
Expander Smooth	设置增益平滑滤波器的 Expander Smooth Time

DRC 的调试需要先通过工具生成对应的 conf 文件，再将 conf 的数值更新到对应的 xml 文件中，如下所示：

```
#This is the parameter of DRC
#AllWinner all rights reserved
#23 - 07 - 17 14 - 19 - 32
# DRC Control Parameters
# ET_EN = 1, LT_EN = 1
# Detect_Noise = 1
# Max_Gain_Limit = 1, Min_Gain_Limit = 1
# Max_Gain = 20, Min_Gain = - 40
# Ke = 1.355263, Kn = 1.306122, Kc = 1.017241, Kl = 0
# ET = - 52.33333, OPE = - 65.66666
# CT = - 36, OPC = - 44.33333
# LT = - 16.66666, OPL = - 24.66667
# X = - 77.66666, Y = - 24.66667
# Peak Filter = 1
# Peak Filter (ms)
```

```
# Left Attack Time = 8
# Left Release Time = 10
# Right Attack Time = 1
# Right Release Time = 10
# RMS Filter (ms)
# Left Average Time = 10
# Right Average Time = 10

# DRC Registers
[0x00F8] = 0xA0000000
[0x0208] = 0x00FB
[0x020C] = 0x0001
[0x0210] = 0x7665
[0x0214] = 0x000B
[0x0218] = 0x77F0
[0x021C] = 0x00FE
[0x0220] = 0xD450
[0x0224] = 0x00FE
.....
```

如需更详细的调试方法，请参考《TigerAudio 使用指南》。

## 2.5 ROUTE 算法

route 算法用于设置声道映射，如当前只有单喇叭的情况，可以将左右声道进行混音输出。

ROUTE 配置的 xml 文件是 audio\_sw\_route\_param.xml，配置文件如下所示：

```
<?xml version="1.0" encoding="UTF-8"?>
<audio_sw_route_param version="2.10.0-later">
  <param_tree>
    <param path="Speaker" param_id="0" enable="1"/>
  </param_tree>
  <param_unit_pool>
    <param_unit param_id="0">
      <!-- mixer_mode: linear, adaptive -->
      <param name="mixer_mode" value="linear"/>
      <param name="frame_max" value="4096"/>
      <param name="sch_cnt" value="2"/>
      <param name="dch_cnt" value="2"/>
      <!-- value: "-1" or "" mean silence -->
      <param name="dch_map0" value="0,1"/>
      <param name="dch_map1" value="0,1"/>
    </param_unit>
  </param_unit_pool>
</audio_sw_route_param>
```

如上 dch\_map0 即为对应的 0 通道，映射到原来的的 0, 1 通道，即可以实现左右声道的混音输出。




## 著作权声明

版权所有 ©2025 珠海全志科技股份有限公司。保留一切权利。

本档及内容受著作权法保护，其著作权由珠海全志科技股份有限公司（“全志”）拥有并保留一切权利。

本档是全志的原创作品和版权财产，未经全志书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本档内容的部分或全部，且不得以任何形式传播。

## 商标声明

、、**全志科技**、（不完全列举）均为珠海全志科技股份有限公司的商标或者注册商标。在本档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

## 免责声明

您购买的产品、服务或特性应受您与珠海全志科技股份有限公司（“全志”）之间签署的商业合同和条款的约束。本档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，全志概不负责。

本档作为使用指导仅供参考。由于产品版本升级或其他原因，本档内容有可能修改，如有变更，恕不另行通知。全志尽全力在本档中提供准确的信息，但并不确保内容完全没有错误，因使用本档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，全志概不负责。本档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本档未以明示或暗示或其他方式授予全志的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。全志不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。全志不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。