
RT-THREAD PLAYER

用户手册

RT-THREAD 文档中心

上海睿赛德电子科技有限公司版权 ©2018



WWW.RT-THREAD.ORG

Monday 15th April, 2019

目录

| | |
|------------------------------|----------|
| 目录 | i |
| 1 版本和修订 | 1 |
| 2 简介 | 2 |
| 2.1 功能特点 | 2 |
| 2.2 许可证 | 2 |
| 2.3 依赖性 | 2 |
| 3 软件架构 | 3 |
| 3.1 结构设计 | 3 |
| 3.2 目录结构 | 4 |
| 4 组件配置 | 5 |
| 4.1 功能裁剪 | 5 |
| 4.2 组件更新 | 6 |
| 5 player 组件 API | 7 |
| 5.1 player API | 7 |
| 5.1.1 player_init | 7 |
| 5.1.2 player_play | 7 |
| 5.1.3 player_play_position | 8 |
| 5.1.4 player_play_websession | 8 |
| 5.1.5 player_stop | 9 |
| 5.1.6 player_pause | 9 |
| 5.1.7 player_do_seek | 9 |
| 5.1.8 player_set_uri | 11 |
| 5.1.9 player_get_uri | 11 |

| | | |
|--------|--|----|
| 5.1.10 | player_get_state | 12 |
| 5.1.11 | player_get_state_str | 12 |
| 5.1.12 | player_fetch_seek | 12 |
| 5.1.13 | player_set_duration | 13 |
| 5.1.14 | player_set_position | 13 |
| 5.1.15 | player_get_duration | 13 |
| 5.1.16 | player_get_position | 14 |
| 5.2 | player app | 14 |
| 5.2.1 | player_app_init | 14 |
| 5.2.2 | player_app_register | 15 |
| 5.2.3 | player_app_connect | 15 |
| 5.2.4 | player_app_disconnect | 15 |
| 5.2.5 | player_app_get_activated | 16 |
| 5.2.6 | player_app_is_activated | 16 |
| 5.2.7 | player_app_set_volume | 16 |
| 5.2.8 | player_app_get_volume | 17 |
| 5.2.9 | player_app_frozen | 17 |
| 5.2.10 | player_app_unfrozen | 17 |
| 5.2.11 | player_app_is_frozen | 18 |
| 5.2.12 | player_app_stop | 18 |
| 5.2.13 | player_app_suspend | 18 |
| 5.2.14 | player_app_resume | 19 |
| 5.3 | list player API | 19 |
| 5.3.1 | list_player_init | 19 |
| 5.3.2 | list_player_current_item | 19 |
| 5.3.3 | list_player_current_index | 20 |
| 5.3.4 | list_player_current_state | 20 |
| 5.3.5 | list_player_current_items | 21 |
| 5.3.6 | list_player_current_position | 21 |
| 5.3.7 | list_player_is_exist | 21 |
| 5.3.8 | list_player_play | 22 |
| 5.3.9 | list_player_switch | 22 |
| 5.3.10 | list_player_num | 23 |

| | | |
|--------|---|----|
| 5.3.11 | list_player_play_index | 23 |
| 5.3.12 | list_player_item | 23 |
| 5.3.13 | list_player_play_item | 24 |
| 5.3.14 | list_player_stop | 24 |
| 5.3.15 | list_player_pause | 24 |
| 5.3.16 | list_player_suspend | 25 |
| 5.3.17 | list_player_resume | 25 |
| 5.3.18 | list_player_prev | 25 |
| 5.3.19 | list_player_next | 26 |
| 5.3.20 | list_player_detach | 26 |
| 5.3.21 | list_player_detach_items | 27 |
| 5.3.22 | list_player_empty | 27 |
| 5.3.23 | list_player_set_mode | 27 |
| 5.3.24 | list_player_mode_set | 28 |
| 5.3.25 | list_player_items_create | 29 |
| 5.3.26 | list_player_items_delete | 29 |
| 5.3.27 | list_player_items_empty | 29 |
| 5.3.28 | list_player_items_set_handler | 30 |
| 5.3.29 | list_player_set_table_handler | 30 |
| 5.3.30 | list_player_items_get_num | 31 |
| 5.3.31 | list_player_items_get_index | 31 |
| 5.3.32 | list_player_items_get_item | 31 |
| 5.3.33 | list_player_item_add | 32 |
| 5.3.34 | list_player_item_del | 32 |
| 5.3.35 | list_player_item_del_by_index | 33 |
| 5.3.36 | list_player_item_get | 33 |
| 5.3.37 | list_player_index_get | 33 |
| 5.3.38 | list_player_dump | 34 |

第 1 章

版本和修订

| Date | Version | Author | Note |
|------------|---------|-------------|----------------------|
| 2018-06-28 | v0.1 | Ernest Chan | 初始版本 |
| 2018-07-05 | v1.0 | Ernest Chan | 同步更新 |
| 2019-01-29 | v1.1 | liu2guang | 同步到 Player 组件 v1.2.2 |

第 2 章

简介

Player 组件是 RT-Thread 开发的适合 MCU 的轻型流媒体音频框架，资源占用小，响应快。

2.1 功能特点

RT-Thread 提供的 Player 组件具有以下优势：1. 本地播放：支持播放本地文件系统中的音频文件。2. 网络播放：支持播放网络数据流歌曲，支持 http 与 https。3. 支持多音频应用相互独立。4. 支持多种音频格式：支持 mp3、wav、m4a、acc、amr、pcm 音频格式。5. 支持歌单播放功能：支持添加、删除、播放、暂停、上一曲、下一曲、切换歌单等多种歌单播放管理功能。6. 消耗资源小：player 本身 RAM 消耗约 28K，不同编解码器消耗的 RAM 不同。7. 方便移植：基于 RT_Thread 设备框架驱动，可根据自带说明文档进行移植使用，代码具有非常强的通用性。8. 基于可靠性高的 RT_Thread 系统，player 组件已经过项目量产的检验。

2.2 许可证

Player 组件以库文件的方式输出给开发者使用，严禁一切个人、组织、团体以任何形式传播源代码。

2.3 依赖性

Player 组件依赖以下 RT-Thread 软件包：- RT-Thread 3.0+ - **LwIP**：网络播放功能。- **cJson**：网络播放功能。- **WebClient v1.0.0**：网络播放功能。- **DFS**：本地与网络数据流播放。- **Finsh**：调试部分。- **Codec Package**：各种音频格式的支持。

除了以上软件包之外，还需要 BSP 实现对应的 Audio 驱动。

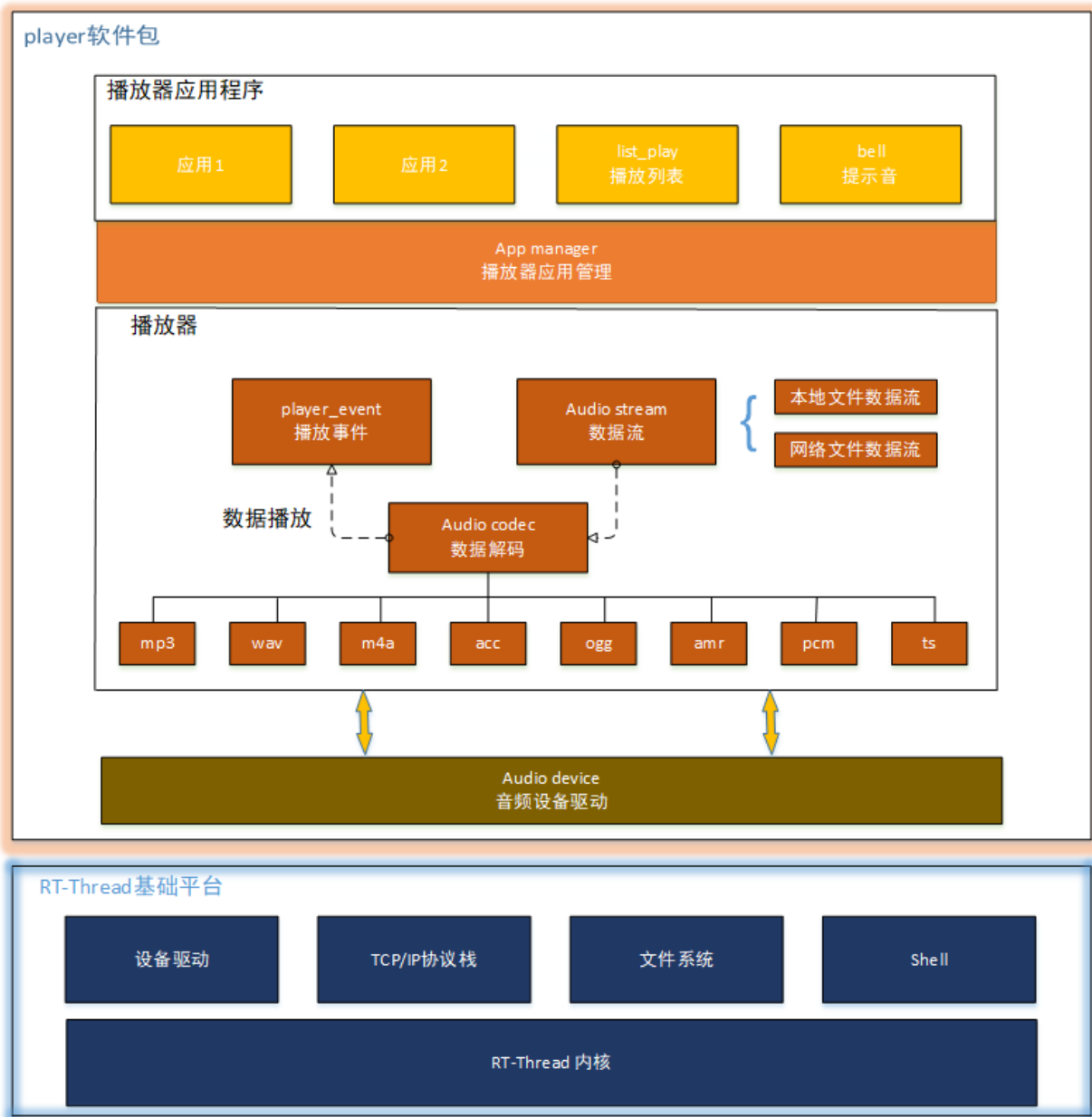
第 3 章

软件架构

本节所描述的 player 组件是基于 RT_Thread 基础平台播放器，该播放器主要是通过基础平台的设备管理框架调度、使用相关音频设备完成音频播放的功能。

3.1 结构设计

player 组件总体结构设计如下所示：



在总体结构图中，最底层的基础平台为 RT_Thread 操作系统，这是 player 组件的基础。在基础平台之上，player 组件主要分为四层结构：音频设备驱动、播放器、播放器应用管理、播放器应用。音频设备驱动是用于连接音频设备与播放器，对音频设备进行管理；播放器用于播放处理单首歌曲，协调完成数据流的处理、解码以及数据播放事件，保障播放器的正常工作；播放器应用管理提供给播放应用控制播放器的接口；播放器应用提供给用户完成一个或者多个相互独立的播放器应用功能。

3.2 目录结构

RT-Thread 提供的 Player 组件目录结构如下：

| 名称 | 说明 |
|----------------|--|
| build | 编译中间文件, 使用者无需关心 |
| docs | Player 组件相关文档 |
| example | Player 组件使用例程 |
| player | Player 组件源码 |
| .gitignore | git 忽略文件, 使用者无需关心 |
| .gitlab-ci.yml | gitlab ci 脚本, 使用者无需关心 |
| LICENSE | 许可声明 |
| README.md | 简单介绍 |
| SConscript | Player 组件 scons 脚本, 用于组织 player 源码编译方式 |

第 4 章

组件配置

Player 组件具有丰富的功能, 开发者可以针对这些功能做项目级别的裁剪与功能配置, 可以将不需要使用的功能关闭来减少 ROM 与 RAM 的占用。

4.1 功能裁剪

Player 组件需要使用 RT-Thread 的 ENV 中的包管理器进行配置, 具体配置选项如下:

```
Privated Packages of RealThread
[*] Player: The RT-Thread lightweight streaming audio playback package
    version (v1.2.2) ---> /* 版本号 */
    examples ---> /* 例程 */
        [*] Using player msh command /* player 层 msh 测试命令 */
        [*] Using listplayer msh command /* listplayer 层 msh 测试命令 */
    Audio codec configuration ---> /* 音频格式解码配置 */
        [*] Enable wav codec support /* WAV 格式支持 */
        [ ] Enable mp3 codec support /* MP3 格式支持 */
        [ ] Enable aac codec support /* AAC 格式支持 */
    Net related configuration ---> /* 网络相关配置 */
        [*] Enable net stream /* 开启网络数据流支持 */
        (720) net stream buffer size(KB) /* 网络数据流缓存 buffer 大小 */
        [*] Enable hls stream /* 使能 HLS 功能支持 */
    Application configuration ---> /* 应用支持配置 */
        [*] Enable listplayer app /* 开启 listplayer API 支持 */
        [*] Enable bellplayer app (built-in test) /* 开启 bellplayer API 支持(内测中) */
```

开发者可以根据需要选择启用解码器、网络播放、列表播放 app、提示音播放 app、命令例程。

4.2 组件更新

保存功能配置后, 使用 `pkgs --update` 命令手动更新 Player 组件到 BSP 中。

第 5 章

player 组件 API

Player 组件 API 主要包含内容分别为 `player API`、`player app API`、`list player API`。每个部分再进行相关 API 详细的介绍。另外需要注意：每层是不向上兼容的，意味着开发者使用了 `player API` 后就不能使用 `player app API` 和 `list player API` 层的 API 了，所以建议不要混合使用 API。

5.1 player API

5.1.1 player_init

```
int player_init(void);
```

功能描述：初始化播放器，作用是初始化音频设备，如果配置了网络音频播放，则还会初始化网络数据流接口。

| 参数 | 描述 |
|------|----|
| void | 无 |

| 返回值 | 返回初始化状态 |
|---------------|---------|
| PLAYER_OK | 正确执行 |
| PLAYER_FAILED | 失败 |

5.1.2 player_play

```
int player_play(void);
```

功能描述：从头开始播放之前设置好的歌曲。

| 参数 | 描述 |
|---------------|---------|
| void | 无 |
| | |
| 返回值 | 返回初始化状态 |
| PLAYER_OK | 正确执行 |
| PLAYER_FAILED | 失败 |

注意事项：1. 注意当播放音乐时，直接调用此函数将返回失败。2. 当播放暂停且 uri 没有更改时，恢复播放。3. 当播放暂停但 uri 已更改时切换播放。

5.1.3 player_play_position

```
int player_play_position(int position);
```

功能描述：从指定位置开始播放之前设置好的歌曲。

| 参数 | 描述 |
|---------------|--------------|
| position | 歌曲指定位置 (单位秒) |
| | |
| 返回值 | 返回初始化状态 |
| PLAYER_OK | 正确执行 |
| PLAYER_FAILED | 失败 |

注意：1. 注意当播放音乐时，直接调用此函数将返回失败。2. 当播放暂停且 uri 没有更改时，恢复播放。3. 当播放暂停但 uri 已更改时切换播放。

5.1.4 player_play_websession

```
int player_play_websession(struct webclient_session* session);
```

功能描述：播放网络音频，前提条件是连接上网络，否则直接返回失败。

| 参数 | 描述 |
|---------|------|
| session | 网络会话 |

| | |
|-----|---------|
| 返回值 | 返回初始化状态 |
|-----|---------|

| | |
|-----------|------|
| PLAYER_OK | 正确执行 |
|-----------|------|

| | |
|---------------|----|
| PLAYER_FAILED | 失败 |
|---------------|----|

5.1.5 player_stop

```
int player_stop(void);
```

功能描述：停止正在播放的歌曲，停止状态调用直接返回成功。

| | |
|----|----|
| 参数 | 描述 |
|----|----|

| | |
|------|---|
| void | 无 |
|------|---|

| | |
|-----|---------|
| 返回值 | 返回初始化状态 |
|-----|---------|

| | |
|-----------|------|
| PLAYER_OK | 正确执行 |
|-----------|------|

| | |
|---------------|----|
| PLAYER_FAILED | 失败 |
|---------------|----|

5.1.6 player_pause

```
int player_pause(void);
```

功能描述：暂停播放歌曲。

| | |
|----|----|
| 参数 | 描述 |
|----|----|

| | |
|------|---|
| void | 无 |
|------|---|

| | |
|-----|---------|
| 返回值 | 返回初始化状态 |
|-----|---------|

| | |
|-----------|------|
| PLAYER_OK | 正确执行 |
|-----------|------|

| | |
|---------------|----|
| PLAYER_FAILED | 失败 |
|---------------|----|

注意事项：1. player 层中并未设计单独的恢复播放 API，当暂停的情况下播放器的 URI 没有改变的情况下，调用 `int player_play(void)` 可以恢复播放。

5.1.7 player_do_seek

```
int player_do_seek(int position);
```

功能描述：切换播放进度。

| 参数 | 描述 |
|---------------|--------------|
| position | 歌曲指定位置 (单位秒) |
| | |
| 返回值 | 返回初始化状态 |
| PLAYER_OK | 正确执行 |
| PLAYER_FAILED | 失败 |

5.1.8 player_set_uri

```
int player_set_uri(const char *uri);
```

功能描述：设置 uri，及设置播放音频位置。

| 参数 | 描述 |
|---------------|-----------------------------|
| uri | 播放位置统一资源标识符，支持本地播放文件与网络播放文件 |
| | |
| 返回值 | 返回初始化状态 |
| PLAYER_OK | 正确执行 |
| PLAYER_FAILED | 失败 |

5.1.9 player_get_uri

```
const char* player_get_uri(void);
```

功能描述：获取当前正在播放的歌曲的 URI。

| 参数 | 描述 |
|----------|------------|
| void | 无 |
| | |
| 返回值 | 返回初始化状态 |
| RT_NULL | 没有歌曲 URI |
| !RT_NULL | 对应播放中的 URI |

5.1.10 player_get_state

```
int player_get_state(void)
```

功能描述：获取播放状态。

| 参数 | 描述 |
|---------------------|---------|
| void | 无 |
| 返回值 | 返回初始化状态 |
| PLAYER_STAT_STOPPEI | 停止状态 |
| PLAYER_STAT_PLAYING | 播放状态 |
| PLAYER_STAT_PAUSED | 暂停状态 |

5.1.11 player_get_state_str

```
const char* player_get_state_str(void);
```

功能描述：获取播放状态字符串。

| 参数 | 描述 |
|-----------|---------|
| void | 无 |
| 返回值 | 返回初始化状态 |
| “STOPPED” | 停止状态字符串 |
| “PLAYING” | 播放状态字符串 |
| “PAUSED” | 暂停状态字符串 |

5.1.12 player_fetch_seek

```
int player_fetch_seek(void);
```

功能描述：获取播放进度。

| 参数 | 描述 |
|------|----|
| void | 无 |

| | |
|-----|---------|
| 返回值 | 返回初始化状态 |
|-----|---------|

| | |
|-----|------|
| num | 播放进度 |
|-----|------|

注意事项：1. 不建议开发者使用, 内部使用接口。

5.1.13 player_set_duration

```
int player_set_duration(int duration);
```

功能描述：设置歌曲总播放时长。

| | |
|----|----|
| 参数 | 描述 |
|----|----|

| | |
|----------|--------------|
| duration | 歌曲总播放时长 (毫秒) |
|----------|--------------|

| | |
|-----|---------|
| 返回值 | 返回初始化状态 |
|-----|---------|

| | |
|----------|--------------|
| duration | 歌曲总播放时长 (毫秒) |
|----------|--------------|

注意事项：1. 不建议开发者使用, 内部使用接口。

5.1.14 player_set_position

```
int player_set_position(int position);
```

功能描述：设置播放位置。

| | |
|----|----|
| 参数 | 描述 |
|----|----|

| | |
|----------|-----------|
| position | 播放位置 (毫秒) |
|----------|-----------|

| | |
|-----|---------|
| 返回值 | 返回初始化状态 |
|-----|---------|

| | |
|----------|-----------|
| position | 播放位置 (毫秒) |
|----------|-----------|

注意事项：1. 不建议开发者使用, 内部使用接口。

5.1.15 player_get_duration

```
int player_get_duration(void);
```

功能描述：获取播放时间长度。

| 参数 | 描述 |
|----------|-----------|
| void | 无 |
| | |
| 返回值 | 返回初始化状态 |
| duration | 播放位置 (毫秒) |

5.1.16 player_get_position

```
int player_get_position(void);
```

功能描述：获取播放时间位置。

| 参数 | 描述 |
|----------|-----------|
| void | 无 |
| | |
| 返回值 | 返回初始化状态 |
| position | 播放位置 (毫秒) |

5.2 player app

5.2.1 player_app_init

```
int player_app_init(void);
```

功能描述：初始化播放器，设置默认配置。

| 参数 | 描述 |
|------|---------|
| void | 无 |
| | |
| 返回值 | 返回初始化状态 |
| 0 | 正确执行 |
| -1 | 失败 |

5.2.2 player_app_register

```
void player_app_register(struct player_app *app);
```

功能描述：注册音频播放器应用。

| 参数 | 描述 |
|------|------------|
| app | 将要注册的播放器应用 |
| 返回值 | 返回初始化状态 |
| void | 无 |

5.2.3 player_app_connect

```
int player_app_connect(struct player_app *app);
```

功能描述：连接音频播放器应用。

| 参数 | 描述 |
|-----|------------|
| app | 将要连接的播放器应用 |
| 返回值 | 返回初始化状态 |
| 0 | 正确执行 |
| -1 | 失败 |

5.2.4 player_app_disconnect

```
void player_app_disconnect(struct player_app *app);
```

功能描述：分离音频播放器应用。

| 参数 | 描述 |
|------|------------|
| app | 将要分离的播放器应用 |
| 返回值 | 返回初始化状态 |
| void | 无 |

5.2.5 player_app_get_activated

```
struct player_app *player_app_get_activated(void);
```

功能描述：获取当前音频播放器应用。

| 参数 | 描述 |
|-------------------|------------|
| void | 无 |
| | |
| 返回值 | 返回初始化状态 |
| struct player_app | 返回当前播放器的应用 |

5.2.6 player_app_is_activated

```
rt_bool_t player_app_is_activated(struct player_app *app);
```

功能描述：判断当前音频播放器应用连接情况，如果是正在连接的应用，则返回 RT_TRUE；否则返回 RT_FALSE。

| 参数 | 描述 |
|----------|---------|
| app | 将判断的应用 |
| | |
| 返回值 | 返回初始化状态 |
| RT_TRUE | 连接 |
| RT_FALSE | 未连接 |

5.2.7 player_app_set_volume

```
void player_app_set_volume(struct player_app *app, int volume);
```

功能描述：设置播放器应用的音量。

| 参数 | 描述 |
|--------|---------------|
| app | 将设置的应用 |
| volume | 音量，范围 0 ~ 100 |

| | |
|-----|---------|
| 返回值 | 返回初始化状态 |
|-----|---------|

| | |
|------|---|
| void | 无 |
|------|---|

5.2.8 player_app_get_volume

```
int player_app_get_volume(struct player_app *app);
```

功能描述：获取播放器应用的音量。

| | |
|----|----|
| 参数 | 描述 |
|----|----|

| | |
|-----|--------|
| app | 将获取的应用 |
|-----|--------|

| | |
|-----|---------|
| 返回值 | 返回初始化状态 |
|-----|---------|

| | |
|-----|---------------|
| int | 音量，范围 0 ~ 100 |
|-----|---------------|

5.2.9 player_app_frozen

```
void player_app_frozen(void);
```

功能描述：冻结操作，当 `frozen` 设置为 1 时，也就是使用该函数时，播放器应用处于无法操作停止、恢复等状态。

| | |
|----|----|
| 参数 | 描述 |
|----|----|

| | |
|------|---|
| void | 无 |
|------|---|

| | |
|-----|---------|
| 返回值 | 返回初始化状态 |
|-----|---------|

| | |
|------|---|
| void | 无 |
|------|---|

5.2.10 player_app_unfrozen

```
void player_app_unfrozen(void);
```

功能描述：释放操作，使用该函数时，将播放器处于可操作状态，值设置为 0。

| | |
|----|----|
| 参数 | 描述 |
|----|----|

| | |
|------|---|
| void | 无 |
|------|---|

| | |
|-----|---------|
| 返回值 | 返回初始化状态 |
|-----|---------|

| | |
|------|---|
| void | 无 |
|------|---|

5.2.11 player_app_is_frozen

```
rt_bool_t player_app_is_frozen(void);
```

| | |
|----|----|
| 参数 | 描述 |
|----|----|

| | |
|------|---|
| void | 无 |
|------|---|

| | |
|-----|---------|
| 返回值 | 返回初始化状态 |
|-----|---------|

| | |
|---------|----|
| RT_TRUE | 冻结 |
|---------|----|

| | |
|----------|-----|
| RT_FALSE | 未冻结 |
|----------|-----|

查询播放器是否能够操作。

5.2.12 player_app_stop

```
struct player_app *player_app_stop(void);
```

功能描述：停止播放器应用，在播放器应用处于可操作状态，断开播放器应用的连接，停止播放。

| | |
|----|----|
| 参数 | 描述 |
|----|----|

| | |
|------|---|
| void | 无 |
|------|---|

| | |
|-----|---------|
| 返回值 | 返回初始化状态 |
|-----|---------|

| | |
|---------------------|-------|
| struct player_app * | 播放器应用 |
|---------------------|-------|

5.2.13 player_app_suspend

```
struct player_app *player_app_suspend(void);
```

功能描述：挂起播放器应用。

| | |
|----|----|
| 参数 | 描述 |
|----|----|

| | |
|------|---|
| void | 无 |
|------|---|

| | |
|-----|---------|
| 返回值 | 返回初始化状态 |
|-----|---------|

| | |
|---------------------|-------|
| struct player_app * | 播放器应用 |
|---------------------|-------|

5.2.14 player_app_resume

```
void player_app_resume (struct player_app *app);
```

功能描述：恢复播放器应用。

| | |
|----|----|
| 参数 | 描述 |
|----|----|

| | |
|-----|-------------|
| app | 将恢复播放的播放器应用 |
|-----|-------------|

| | |
|-----|---------|
| 返回值 | 返回初始化状态 |
|-----|---------|

| | |
|------|---|
| void | 无 |
|------|---|

5.3 list player API

5.3.1 list_player_init

```
int list_player_init(void);
```

功能描述：初始化列表播放应用程序。

| | |
|----|----|
| 参数 | 描述 |
|----|----|

| | |
|------|---|
| void | 无 |
|------|---|

| | |
|-----|---------|
| 返回值 | 返回初始化状态 |
|-----|---------|

| | |
|-----|---------|
| int | 返回初始化结果 |
|-----|---------|

注意事项：1. 允许重复调用，但只有第一次初始化列表播放器。

5.3.2 list_player_current_item

```
music_item_t list_player_current_item(void);
```

功能描述：查询播放曲目, 返回播放歌曲信息。

| 参数 | 描述 |
|----------|-----------------------------|
| void | 无 |
| <hr/> | |
| 返回值 | 返回初始化状态 |
| RT_NULL | 列表播放器未初始化/列表播放器未关联歌单/没有播放歌曲 |
| !RT_NULL | 返回播放列表 |

5.3.3 list_player_current_index

```
int list_player_current_index(void);
```

功能描述：获取当前播放歌单的歌曲编号。

| 参数 | 描述 |
|----------|----------|
| void | 无 |
| <hr/> | |
| 返回值 | 返回初始化状态 |
| ≥ 0 | 歌曲编号 |
| -1 | 没有歌曲正在播放 |

5.3.4 list_player_current_state

```
int list_player_current_state(void);
```

功能描述：获取当前列表播放的状态。

| 参数 | 描述 |
|---------------------------|---------|
| void | 无 |
| <hr/> | |
| 返回值 | 返回初始化状态 |
| 0: PLAYER_STAT_STOPPEI | 暂停状态 |
| 1: PLAYER_STAT_PLAYING | 播放状态 |
| 2: PLAYER_STAT_PAUSED | 暂停状态 |

| | |
|-----|---------|
| 返回值 | 返回初始化状态 |
|-----|---------|

5.3.5 list_player_current_items

```
music_list_t list_player_current_items(void);
```

功能描述：获取当前列表播放关联的歌单。

| | |
|----|----|
| 参数 | 描述 |
|----|----|

| | |
|------|---|
| void | 无 |
|------|---|

| | |
|-----|---------|
| 返回值 | 返回初始化状态 |
|-----|---------|

| | |
|---------|----------------------|
| RT_NULL | 列表播放器未初始化/列表播放器未关联歌单 |
|---------|----------------------|

| | |
|----------|------|
| !RT_NULL | 歌单句柄 |
|----------|------|

5.3.6 list_player_current_position

```
int list_player_current_position(void);
```

功能描述：获取当前列表播放播放歌曲的播放进度。

| | |
|----|----|
| 参数 | 描述 |
|----|----|

| | |
|------|---|
| void | 无 |
|------|---|

| | |
|-----|---------|
| 返回值 | 返回初始化状态 |
|-----|---------|

| | |
|---------|----------------------|
| RT_NULL | 列表播放器未初始化/列表播放器未关联歌单 |
|---------|----------------------|

| | |
|----------|------|
| !RT_NULL | 歌单句柄 |
|----------|------|

5.3.7 list_player_is_exist

```
int list_player_is_exist(void);
```

功能描述：获取当前列表播放播放歌曲的播放进度。

| | |
|----|----|
| 参数 | 描述 |
|----|----|

| | |
|------|---|
| void | 无 |
|------|---|

| 返回值 | 返回初始化状态 |
|-----|---------|
| 0 | 不存在 |
| 1 | 存在 |

5.3.8 list_player_play

```
int list_player_play(music_list_t table);
```

功能描述：播放歌单，默认从第一首开始播放。

| 参数 | 描述 |
|-------|------------|
| table | 将要播放的歌单的句柄 |

| 返回值 | 返回初始化状态 |
|------------|----------------|
| -RT_ENOSYS | 列表播放器未初始化 |
| -RT_EINVAL | 将要播放的歌单的句柄为空指针 |
| -RT_ERROR | 播放失败 |
| RT_EOK | 播放成功 |

5.3.9 list_player_switch

```
int list_player_switch(music_list_t table, int index, int position, int state);
```

功能描述：列表播放器自定义切换功能，可以在任何状态下切换到任意的歌单中的任意一首，同时支持切换到歌曲的任意位置，也支持指定切换后的状态：停止/暂停/播放，该函数是高级功能，在 listplayer 层提供的 API 无法满足开发者的应用层时，可以使用这套函数接口来实现复杂的切换。

| 参数 | 描述 |
|----------|--------------|
| table | 切换后的歌单 |
| index | 切换后的歌单中歌曲的编号 |
| position | 切换后的播放位置 |
| state | 切换后的状态 |

| 返回值 | 返回初始化状态 |
|------------|----------------|
| -RT_ENOSYS | 列表播放器未初始化 |
| -RT_EINVAL | 将要播放的歌单的句柄为空指针 |

| | |
|-----|---------|
| 返回值 | 返回初始化状态 |
|-----|---------|

| | |
|--------|------|
| RT_EOK | 切换成功 |
|--------|------|

注意事项：1. 此函数将停止前一个播放并切换音乐列表。

5.3.10 list_player_num

```
int list_player_num(int num);
```

功能描述：播放当前关联歌单中指定歌曲编号的歌曲。

| | |
|----|----|
| 参数 | 描述 |
|----|----|

| | |
|-----|------|
| num | 歌曲编号 |
|-----|------|

| | |
|-----|---------|
| 返回值 | 返回初始化状态 |
|-----|---------|

| | |
|--------|------|
| RT_EOK | 播放成功 |
|--------|------|

注意事项:1. 在 v1.3.0 之后的版本会移除 `list_player_num()` API,建议用户使用 `list_player_play_index()` API。

5.3.11 list_player_play_index

```
int list_player_play_index(int index);
```

功能描述：播放当前关联歌单中指定歌曲编号的歌曲。

| | |
|----|----|
| 参数 | 描述 |
|----|----|

| | |
|-------|------|
| index | 歌曲编号 |
|-------|------|

| | |
|-----|---------|
| 返回值 | 返回初始化状态 |
|-----|---------|

| | |
|--------|------|
| RT_EOK | 播放成功 |
|--------|------|

5.3.12 list_player_item

```
int list_player_item(music_item_t item);
```

功能描述：播放当前关联歌单中指定歌曲句柄的歌曲。

| 参数 | 描述 |
|--------|------|
| item | 歌曲句柄 |
| 返回值 | |
| RT_EOK | 播放成功 |

注意 事项: 1. 在 v1.3.0 之后的版本会移除 `list_player_item()` API, 建议用户使用 `list_player_play_item()` API。

5.3.13 list_player_play_item

```
int list_player_play_item(music_item_t item);
```

功能描述: 播放当前关联歌单中指定歌曲句柄的歌曲。

| 参数 | 描述 |
|--------|------|
| item | 歌曲句柄 |
| 返回值 | |
| RT_EOK | 播放成功 |

5.3.14 list_player_stop

```
void list_player_stop(void);
```

功能描述: 停止播放

| 参数 | 描述 |
|------|----|
| void | 无 |
| 返回值 | |
| void | 无 |

5.3.15 list_player_pause

```
void list_player_pause(void);
```

功能描述：暂停播放

| 参数 | 描述 |
|------|---------|
| void | 无 |
| | |
| 返回值 | 返回初始化状态 |
| void | 无 |

5.3.16 list_player_suspend

```
void list_player_suspend(void);
```

功能描述：同上面暂停播放 `list_player_pause` 功能一样。

| 参数 | 描述 |
|------|---------|
| void | 无 |
| | |
| 返回值 | 返回初始化状态 |
| void | 无 |

5.3.17 list_player_resume

```
void list_player_resume(void);
```

功能描述：恢复播放。

| 参数 | 描述 |
|------|---------|
| void | 无 |
| | |
| 返回值 | 返回初始化状态 |
| void | 无 |

5.3.18 list_player_prev

```
void list_player_prev(void);
```

功能描述：播放上一曲。

| 参数 | 描述 |
|------|---------|
| void | 无 |
| | |
| 返回值 | 返回初始化状态 |
| void | 无 |

注意事项：1. 当正在播放第一首歌曲时，调用此函数将切换到最后一首歌曲。

5.3.19 list_player_next

```
void list_player_next(void);
```

功能描述：播放下一曲。

| 参数 | 描述 |
|------|---------|
| void | 无 |
| | |
| 返回值 | 返回初始化状态 |
| void | 无 |

注意事项：1. 当正在播放最后一首歌曲时，调用此函数将切换到第一首歌曲。

5.3.20 list_player_detach

```
music_list_t list_player_detach(void);
```

功能描述：脱离当前播放器歌单。

| 参数 | 描述 |
|----------|----------------------|
| void | 无 |
| | |
| 返回值 | 返回初始化状态 |
| RT_NULL | 列表播放器未初始化/列表播放器未关联歌单 |
| !RT_NULL | 脱离当前播放器歌单的句柄 |

5.3.21 list_player_detach_items

```
music_list_t list_player_detach_items(void);
```

功能描述：脱离当前播放器歌单。

| 参数 | 描述 |
|----------|----------------------|
| void | 无 |
| 返回值 | 返回初始化状态 |
| RT_NULL | 列表播放器未初始化/列表播放器未关联歌单 |
| !RT_NULL | 脱离当前播放器歌单的句柄 |

注意事项：1. 在 v1.3.0 之后的版本会移除 `list_player_detach_items()` API，建议用户使用 `list_player_detach()` API。2. 清空歌单后会停止当前播放的歌单。

5.3.22 list_player_empty

```
int list_player_empty(void);
```

功能描述：清空当前列表播放器关联歌单，首先是分离歌单然后再是删除歌单。应当注意，删除或者清空歌单，不再保留歌单信息，但是分离歌单会保留歌单信息。

| 参数 | 描述 |
|-----------|-----------|
| void | 无 |
| 返回值 | 返回初始化状态 |
| -RT_ERROR | 列表播放器未初始化 |
| RT_EOK | 清空歌单成功 |

注意事项：1. 清空歌单后会停止当前播放的歌单。

5.3.23 list_player_set_mode

```
int list_player_set_mode(int mode);
```

功能描述：设置播放模式有五种，各参数定义与解释分别如下：

| 参数 | 描述 |
|------|-----------|
| mode | 列表播放器播放模式 |

返回值 返回初始化状态

| | |
|------------|-----------|
| -RT_ENOSYS | 列表播放器未初始化 |
| -RT_EINVAL | 错误的模式参数 |
| RT_EOK | 设置成功 |

```
/* 播放模式 */
#define LISTER_NONE_MODE      (0x00) //模式未设置
#define LISTER_LIST_ONCE_MODE (0x01) //列表单次循环（顺序播放）
#define LISTER_SONG_ONCE_MODE (0x02) //歌曲单次循环（单曲播放）
#define LISTER_LIST_REPEAT_MODE (0x03) //列表周期循环（循环播放）
#define LISTER_SONG_REPEAT_MODE (0x04) //歌曲周期循环（单曲循环）
```

5.3.24 list_player_mode_set

```
int list_player_mode_set(int mode);
```

功能描述：设置播放模式有五种，各参数定义与解释分别如下：

| 参数 | 描述 |
|------|-----------|
| mode | 列表播放器播放模式 |

返回值 返回初始化状态

| | |
|------------|-----------|
| -RT_ENOSYS | 列表播放器未初始化 |
| -RT_EINVAL | 错误的模式参数 |
| RT_EOK | 设置成功 |

```
/* 播放模式 */
#define LISTER_NONE_MODE      (0x00) //模式未设置
#define LISTER_LIST_ONCE_MODE (0x01) //列表单次循环（顺序播放）
#define LISTER_SONG_ONCE_MODE (0x02) //歌曲单次循环（单曲播放）
#define LISTER_LIST_REPEAT_MODE (0x03) //列表周期循环（循环播放）
#define LISTER_SONG_REPEAT_MODE (0x04) //歌曲周期循环（单曲循环）
```

注意事项：1. 在 v1.3.0 之后的版本会移除 `list_player_mode_set()` API，建议用户使用

list_player_set_mode() API。

5.3.25 list_player_items_create

```
music_list_t list_player_items_create(void);
```

功能描述：创建歌单。

| 参数 | 描述 |
|----------|-------------|
| void | 无 |
| 返回值 | |
| RT_NULL | 内存不足，创建失败 |
| !RT_NULL | 创建成功，返回歌单句柄 |

5.3.26 list_player_items_delete

```
void list_player_items_delete(music_list_t table);
```

功能描述：删除将要被删除的歌单。

| 参数 | 描述 |
|-------|---------|
| table | 将要删除的歌单 |
| 返回值 | |
| void | 无 |

5.3.27 list_player_items_empty

```
int list_player_items_empty(music_list_t table);
```

功能描述：清指定空歌单。

| 参数 | 描述 |
|-------|---------|
| table | 将要清指的歌单 |

| | |
|------------|----------|
| 返回值 | 返回初始化状态 |
| -RT_EINVAL | 指定歌单未空指针 |
| -RT_EBUSY | 指定歌单正在播放 |
| RT_EOK | 清空成功 |

5.3.28 list_player_items_set_handler

```
void list_player_items_set_handler(music_list_t table, list_event_handler handler, void *arg);
```

功能描述：设置歌单播放事件回调函数。

| | |
|---------|----------|
| 参数 | 描述 |
| table | 将要设置的歌单 |
| handler | 回调函数 |
| arg | 回调函数传入参数 |

| | |
|------|---------|
| 返回值 | 返回初始化状态 |
| void | 无 |

5.3.29 list_player_set_table_handler

```
void list_player_set_table_handler(music_list_t table, list_event_handler handler, void *arg);
```

功能描述：设置歌单播放事件回调函数。

| | |
|---------|----------|
| 参数 | 描述 |
| table | 将要设置的歌单 |
| handler | 回调函数 |
| arg | 回调函数传入参数 |

| | |
|------|---------|
| 返回值 | 返回初始化状态 |
| void | 无 |

注意事项：1. 在 v1.3.0 之后的版本会移除 `list_player_set_table_handler()` API，建议用户使用

list_player_items_set_handler() API。

5.3.30 list_player_items_get_num

```
int list_player_items_get_num(music_list_t table);
```

功能描述：获取指定歌单的歌曲数量。

| 参数 | 描述 |
|------------|---------|
| table | 将要获取的歌单 |
| 返回值 | 返回初始化状态 |
| -RT_EINVAL | 歌单句柄为空 |
| >=0 | 歌曲数量 |

5.3.31 list_player_items_get_index

```
int list_player_items_get_index(music_list_t table);
```

功能描述：获取指定歌单的歌曲数量。

| 参数 | 描述 |
|------------|---------|
| table | 将要获取的歌单 |
| 返回值 | 返回初始化状态 |
| -RT_EINVAL | 歌单句柄为空 |
| >=0 | 歌曲数量 |

5.3.32 list_player_items_get_item

```
music_item_t list_player_items_get_item(music_list_t table);
```

功能描述：获取指定歌单最后一次播放的歌曲句柄。

| 参数 | 描述 |
|-------|---------|
| table | 将要获取的歌单 |

| | |
|----------|----------------------|
| 返回值 | 返回初始化状态 |
| RT_NULL | 指定歌单句柄未空指针，或者歌单从未播放过 |
| !RT_NULL | 最后一次播放的歌曲句柄 |

5.3.33 list_player_item_add

```
int list_player_item_add(music_list_t table, music_item_t item, int index);
```

功能描述：添加歌曲 API 支持动态与静态添加歌曲，也就是支持在播放开始、播放过程与播放结束后添加歌曲。另外添加歌曲位置参数 index 的值，表示意义如下：在歌单末尾值为 -1，且默认为末尾插入；开始处插入为 0；其他地方为相应位置，当然需要在歌单数据之间，否则报错。

| | |
|-------|---------|
| 参数 | 描述 |
| table | 将要添加的歌单 |
| item | 将添加的歌曲 |
| index | 将要插入的位置 |

| | |
|------------|-------------------------------------|
| 返回值 | 返回初始化状态 |
| -RT_EINVAL | 指定歌单句柄未空指针/歌曲未为空指针/歌曲中为设定 URI |
| -RT_ERROR | 添加的位置超出歌单的范围，(-1 ~ table->item_num) |
| -RT_ENOMEM | 内存分配失败 |
| RT_EOK | 添加成功 |

5.3.34 list_player_item_del

```
int list_player_item_del(music_list_t table, struct music_item *item);
```

功能描述：删除歌曲，可以通过歌曲信息删除歌曲，删除歌曲 API 支持动态与静态删除歌曲，也就是支持在播放开始、播放过程与播放结束后删除歌曲。

| | |
|-------|--------|
| 参数 | 描述 |
| table | 歌单 |
| item | 将删除的歌曲 |

| | |
|------------|--------------------|
| 返回值 | 返回初始化状态 |
| -RT_EINVAL | 指定歌单句柄未空指针/歌曲未为空指针 |

| | |
|-----------|---------|
| 返回值 | 返回初始化状态 |
| -RT_ERROR | 歌曲未发现 |
| RT_EOK | 删除成功 |

5.3.35 list_player_item_del_by_index

```
int list_player_item_del_by_index(music_list_t table, int index);
```

功能描述：通过歌曲列表删除第几条。

| | |
|------------|--------------------------------------|
| 参数 | 描述 |
| table | 歌单 |
| index | 将删除的歌曲的编号 |
| 返回值 | 返回初始化状态 |
| -RT_EINVAL | 指定歌单句柄未空指针/歌曲未为空指针 |
| -RT_ERROR | 删除的位置超出歌单的范围, (-1 ~ table->item_num) |
| RT_EOK | 删除成功 |

5.3.36 list_player_item_get

```
music_item_t list_player_item_get(music_list_t table, int index);
```

功能描述：返回指定歌单中指定编号的歌曲句柄。

| | |
|----------|---------|
| 参数 | 描述 |
| table | 歌单 |
| index | 指定歌曲的编号 |
| 返回值 | 返回初始化状态 |
| RT_NULL | 歌曲未找到 |
| !RT_NULL | 获取成功 |

5.3.37 list_player_index_get

```
int list_player_index_get(music_list_t table, music_item_t item);
```

功能描述：返回指定歌单中指定歌曲的编号。

| 参数 | 描述 |
|------------|----------------------------|
| table | 歌单 |
| item | 指定歌曲 |
| 返回值 | 返回初始化状态 |
| ≥ 0 | 歌曲编号 |
| -RT_EINVAL | 获取失败，歌单或者歌曲为空指针 |
| -RT_ERROR | 获取失败，歌单中歌曲 name 和 URL 为空指针 |

5.3.38 list_player_dump

```
int list_player_dump(music_list_t table);
```

功能描述：打印指定歌单的相关消息，用于调试。

| 参数 | 描述 |
|------------|----------------|
| table | 指定歌单 |
| 返回值 | 返回初始化状态 |
| -RT_EINVAL | dump 失败，歌单为空指针 |
| RT_EOK | dump 成功 |