UNF 接口

差异说明

文档版本　05

发布日期　2015-06-15

商标声明

 、HISILICON、海思和其他海思商标均为深圳市海思半导体有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受海思公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，海思公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

深圳市海思半导体有限公司

地址：　深圳市龙岗区坂田华为基地华为电总部　邮编：518129

网址：　http://www.hisilicon.com

客户服务邮箱：　support@hisilicon.com

前 言

概述

本文档是一个 UNF 接口差异的说明文档，用来指导客户了解 UNF 版本之间的差异点，使客户能够快速使用新的软件接口。

产品版本

与本文档相对应的产品版本如下。

产品名称	产品版本
Hi3716C	V2XX
Hi3719C	V1XX
Hi3719M	V1XX
Hi3718C	V1XX
Hi3718M	V1XX
Hi3716M	V4XX
Hi3798C	V1XX
Hi3796C	V1XX
Hi3798M	V1XX
Hi3796M	V1XX
Hi3798C	V2XX

读者对象

本文档（本指南）主要适用于以下工程师：

- 技术支持工程师

- 软件开发工程师

作者信息

模块名称	作者信息
概述	H00183514
Common	G00277009
AVPLAY	L00211254
CIPHER	Z00213260
Frontend	W00136565/W00203631
HDMI	Y00229039
VO	F00228503
PMOC	Z00149549
PQ	P00203646
PVR	Z00111416
SOUND	Z00183919
VI	L00214567
Closed Caption	H00217063
Teletext	D66707
Hiplayer	L00194915
HiGo	Z00141204
AVPLAY	L00211254
SPI	H00279113
Frontend	W00136565/W00203631
HiGO	Z00141204
HiFB	Z00141204
MCE	L00211254
PDM	L00211254
VO	F00228503
SOUND	Z00183919
SPI	H00279113
PQ	P00203646

模块名称	作者信息
Demux	L64565
Demux	L64565
Frontend	W00136565/W00203631
SOUND	Z00183919

修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

修订日期	版本	修订说明
2014-05-31	00B01	第一次临时版本发布。
2014-07-14	00B02	新增 4、5 两章。
2014-08-13	00B03	新增第 6 章。
2014-10-30	01	新增支持 Hi3796MV100 芯片。
2015-02-15	02	新增 7、8、9、10 章节。
2015-03-30	03	新增第 11 章节。
2015-05-07	04	新增支持 Hi3798CV200、Hi3716MV420/410 芯片。
2015-06-15	05	删除第 12 章节的 demux 小节。

目 录

1 概述

本文档主要描述不同版本的 UNF 接口的差异。涉及到以下三个方面：

- 功能、规格的变更；
- 接口函数、数据结构的变更；
- 调用流程的变更。

2 UNF3.2.0 与 UNF3.1 之间的差异

2.1 Common

2.1.1 概述

相对于 UNF 3.1 版本，UNF 3.2.0 在总体功能上有以下变更：

- 新增获取芯片的能力

2.1.1.1 获取芯片的能力

为支持获取芯片的能力。引起的变更如下：

数据结构

- 新增 HI_CHIP_CAP_E

接口

- 新增 HI_SYS_GetChipCapability

2.1.2 数据结构

2.1.2.1 新增

HI_CHIP_CAP_E

【结构体定义】

```
typedef enum hiCHIP_CAP_E
{
    HI_CHIP_CAP_DOLBY,
    HI_CHIP_CAP_DTS,
    HI_CHIP_CAP_ADVCA,
    HI_CHIP_CAP_MACROVISION
} HI_CHIP_CAP_E;
```

【新增原因】

新特性，枚举可获取的芯片能力。

【注意事项】

无

【范例】

无

2.1.2.2 修改

HI_CHIP_VERSION_E

【结构体定义】

修改前：

```
typedef enum hiCHIP_VERSION_E
{
    HI_CHIP_VERSION_V100 = 0x100,
    HI_CHIP_VERSION_V101 = 0x101,
    HI_CHIP_VERSION_V200 = 0x200,
    HI_CHIP_VERSION_V300 = 0x300,
    HI_CHIP_VERSION_BUTT
}HI_CHIP_VERSION_E;
```

修改后：

```
typedef enum hiCHIP_VERSION_E
{
    HI_CHIP_VERSION_V100 = 0x100,
    HI_CHIP_VERSION_V101 = 0x101,
    HI_CHIP_VERSION_V200 = 0x200,
    HI_CHIP_VERSION_V300 = 0x300,
    HI_CHIP_VERSION_V400 = 0x400,
    HI_CHIP_VERSION_BUTT
}HI_CHIP_VERSION_E;
```

【修改说明】

新增芯片版本号。

【注意事项】

无

【范例】

无

FLASH_OPT_S

【结构体定义】

修改前：

```
typedef struct tagFLASH_OPT_S
{
    int (*raw_read)(int fd, unsigned long long *startaddr, unsigned char *buffer, unsigned long length, unsigned long long openaddr, unsigned long long limit_leng, int read_oob, int skip_badblock);
    int (*raw_write)(int fd, unsigned long long *startaddr, unsigned char *buffer, unsigned long length, unsigned long long openaddr, unsigned long long limit_leng, int write_oob);
    int (*raw_erase)(int fd, unsigned long long startaddr, unsigned long long length, unsigned long long openaddr, unsigned long long limit_leng);
} FLASH_OPT_S;
```

修改后：

```
typedef struct tagFLASH_OPT_S
{
    int (*raw_read)(int fd, unsigned long long *startaddr, unsigned char *buffer, unsigned long length, unsigned long long openaddr, unsigned long long limit_leng, int read_oob, int skip_badblock);
    int (*raw_write)(int fd, unsigned long long *startaddr, unsigned char *buffer, unsigned long length, unsigned long long openaddr, unsigned long long limit_leng, int write_oob);
    long long (*raw_erase)(int fd, unsigned long long startaddr, unsigned long long length, unsigned long long openaddr, unsigned long long limit_leng);
} FLASH_OPT_S;
```

【修改说明】

raw_erase 函数指针返回的数据大于 int 所能表示的范围。

【注意事项】

无

【范例】

无

2.1.3 函数接口

2.1.3.1 新增

HI_SYS_GetChipCapability

【接口定义】

```
HI_S32 HI_SYS_GetChipCapability(HI_CHIP_CAP_E enChipCap, HI_BOOL
*pbSupport);
```

【新增原因】

新特性，增加获取芯片支持的能力。

【注意事项】

无。

【范例】

无。

2.2 AVPLAY

2.2.1 概述

UNF 3.2.0 AVPLAY 模块相对于 UNF 3.1 版本，在总体功能上有以下变更：

- 新增视频帧错误率上报事件。
- 新增 TVP 属性设置。
- 新增精准 seek 功能。
- 新增支持 full-band 功能。
- CC 数据增加顶底场优先标志。

2.2.1.1 视频帧错误率上报

视频帧解码出现错误的情况下，上报错误帧的错误率，范围 1-100，引起的变更如下：

数据结构

修改 HI_UNF_AVPLAY_EVENT_E

函数接口

无。

2.2.1.2 设置 TVP 属性

安全视频通路特性，从视频送流、解码到显示的整个通路上都使用安全内存。只有支持 Trusted Video Path 特性的时候，设置 TVP 属性 `bEnable` 为 TRUE，引起的变更如下：

数据结构

- 新增 HI_UNF_AVPLAY_TVP_ATTR_S
- 修改 HI_UNF_AVPLAY_ATTR_ID_E

函数接口

无。

2.2.1.3 精准 seek

该特性允许在 reset 解码器的时候指定特定的 pts，让音视频解码器 seek 到这个 pts 附近的码流继续解码。引起的变更如下：

数据结构

修改 HI_UNF_AVPLAY_RESET_OPT_S

函数接口

无。

2.2.1.4 Full-band

该特性可以实现快速换台的功能：同时启动多路 Avplayer，其中一路处于正常播放状态，其余的处于只接收 TS 码流但是不解码不输出音视频的预播放状态；换台的时候，将正在处于播放状态的 Avplayer 切换到预播放状态，如果切换的节目处于预播放状态的 Avplayer 中，则将其切换到正常播放状态，由于切换前该 Avplayer 已经接收并解复用出 ES 流，切换成播放状态后可以快速的将 ES 流解码输出达到快速换台的目的。引起的变更如下：

数据结构

- 新增 HI_UNF_AVPLAY_PRESTART_OPT_S
- 新增 HI_UNF_AVPLAY_PRESTOP_OPT_S
- 修改 HI_UNF_AVPLAY_STATUS_E

函数接口

- 新增 HI_UNF_AVPLAY_PreStart
- 新增 HI_UNF_AVPLAY_PreStop

2.2.1.5 CC 数据顶底场优先标志

为保证 CC 数据能够正确的解码,增加顶底场优先标志。引起的变更如下：

数据结构

修改 HI_UNF_VIDEO_USERDATA_S

函数接口

无。

2.2.2 数据结构

2.2.2.1 新增

HI_UNF_AVPLAY_PRESTART_OPT_S

【结构体定义】

```
typedef struct hiAVPLAY_PRESTART_OPT_S
{
    HI_U32      u32Reserved;
} HI_UNF_AVPLAY_PRESTART_OPT_S;
```

【新增原因】

支持 full-band 功能。

【注意事项】

该结构体作为预留后续扩展。

【范例】

无。

HI_UNF_AVPLAY_PRESTOP_OPT_S

【结构体定义】

```
typedef struct hiAVPLAY_PRESTOP_OPT_S
{
    HI_U32      u32Reserved;
} HI_UNF_AVPLAY_PRESTOP_OPT_S;
```

【新增原因】

支持 full-band 功能。

【注意事项】

该结构体作为预留后续扩展。

【范例】

无。

HI_UNF_AVPLAY_TVP_ATTR_S

【结构体定义】

```
typedef struct hiUNF_AVPLAY_TVP_ATTR_S
{
    HI_BOOL             bEnable;
}HI_UNF_AVPLAY_TVP_ATTR_S;
```

【新增原因】

支持 TVP 功能。

【注意事项】

只有支持 Trusted Video Path 特性的时候，设置该属性 bEnable 为 TRUE。

【范例】

无。

2.2.2.2 修改

HI_UNF_AVPLAY_EVENT_E

【结构体定义】

修改前：

```
typedef enum hiUNF_AVPLAY_EVENT_E
{
    HI_UNF_AVPLAY_EVENT_EOS,
    HI_UNF_AVPLAY_EVENT_STOP,
    HI_UNF_AVPLAY_EVENT_RNG_BUF_STATE,
    HI_UNF_AVPLAY_EVENT_NORM_SWITCH,
    HI_UNF_AVPLAY_EVENT_FRAMEPACKING_CHANGE,
    HI_UNF_AVPLAY_EVENT_NEW_VID_FRAME,
    HI_UNF_AVPLAY_EVENT_NEW_AUD_FRAME,
    HI_UNF_AVPLAY_EVENT_NEW_USER_DATA,
    HI_UNF_AVPLAY_EVENT_GET_AUD_ES,
    HI_UNF_AVPLAY_EVENT_IFRAME_ERR,
    HI_UNF_AVPLAY_EVENT_SYNC_PTS_JUMP,
    HI_UNF_AVPLAY_EVENT_SYNC_STAT_CHANGE,
    HI_UNF_AVPLAY_EVENT_VID_BUF_STATE,
    HI_UNF_AVPLAY_EVENT_AUD_BUF_STATE,
    HI_UNF_AVPLAY_EVENT_VID_UNSUPPORT,
    HI_UNF_AVPLAY_EVENT_BUTT
} HI_UNF_AVPLAY_EVENT_E;
```

修改后：

```
typedef enum hiUNF_AVPLAY_EVENT_E
{
    HI_UNF_AVPLAY_EVENT_EOS,
    HI_UNF_AVPLAY_EVENT_STOP,
    HI_UNF_AVPLAY_EVENT_RNG_BUF_STATE,
    HI_UNF_AVPLAY_EVENT_NORM_SWITCH,
    HI_UNF_AVPLAY_EVENT_FRAMEPACKING_CHANGE,
    HI_UNF_AVPLAY_EVENT_NEW_VID_FRAME,
    HI_UNF_AVPLAY_EVENT_NEW_AUD_FRAME,
    HI_UNF_AVPLAY_EVENT_NEW_USER_DATA,
    HI_UNF_AVPLAY_EVENT_GET_AUD_ES,
    HI_UNF_AVPLAY_EVENT_IFRAME_ERR,
    HI_UNF_AVPLAY_EVENT_SYNC_PTS_JUMP,
    HI_UNF_AVPLAY_EVENT_SYNC_STAT_CHANGE,
    HI_UNF_AVPLAY_EVENT_VID_BUF_STATE,
    HI_UNF_AVPLAY_EVENT_AUD_BUF_STATE,
    HI_UNF_AVPLAY_EVENT_VID_UNSUPPORT,
    HI_UNF_AVPLAY_EVENT_VID_ERR_RATIO,
    HI_UNF_AVPLAY_EVENT_BUTT
} HI_UNF_AVPLAY_EVENT_E;
```

【修改原因】

支持视频帧错误率上报。

【注意事项】

当视频解码没有错误的情况下，不会上 ERR_RATIO 该事件，上报该事件时，错误率范围为 1～100。

【范例】

无。

HI_UNF_AVPLAY_ATTR_ID_E

【结构体定义】

修改前：

```
typedef enum hiUNF_AVPLAY_ATTR_ID_E
{
    HI_UNF_AVPLAY_ATTR_ID_STREAM_MODE = 0,
    HI_UNF_AVPLAY_ATTR_ID_ADEC,
    HI_UNF_AVPLAY_ATTR_ID_VDEC,
    HI_UNF_AVPLAY_ATTR_ID_AUD_PID,
    HI_UNF_AVPLAY_ATTR_ID_VID_PID,
    HI_UNF_AVPLAY_ATTR_ID_PCR_PID,
```

```
    HI_UNF_AVPLAY_ATTR_ID_SYNC,
    HI_UNF_AVPLAY_ATTR_ID_AFD,
    HI_UNF_AVPLAY_ATTR_ID_OVERFLOW,
    HI_UNF_AVPLAY_ATTR_ID_MULTIAUD,
    HI_UNF_AVPLAY_ATTR_ID_FRMRATE_PARAM,
    HI_UNF_AVPLAY_ATTR_ID_FRMPACK_TYPE,
    HI_UNF_AVPLAY_ATTR_ID_LOW_DELAY,
    HI_UNF_AVPLAY_ATTR_ID_BUTT
} HI_UNF_AVPLAY_ATTR_ID_E;
```

修改后：

```
typedef enum hiUNF_AVPLAY_ATTR_ID_E
{
    HI_UNF_AVPLAY_ATTR_ID_STREAM_MODE = 0,
    HI_UNF_AVPLAY_ATTR_ID_ADEC,
    HI_UNF_AVPLAY_ATTR_ID_VDEC,
    HI_UNF_AVPLAY_ATTR_ID_AUD_PID,
    HI_UNF_AVPLAY_ATTR_ID_VID_PID,
    HI_UNF_AVPLAY_ATTR_ID_PCR_PID,
    HI_UNF_AVPLAY_ATTR_ID_SYNC,
    HI_UNF_AVPLAY_ATTR_ID_AFD,
    HI_UNF_AVPLAY_ATTR_ID_OVERFLOW,
    HI_UNF_AVPLAY_ATTR_ID_MULTIAUD,
    HI_UNF_AVPLAY_ATTR_ID_FRMRATE_PARAM,
    HI_UNF_AVPLAY_ATTR_ID_FRMPACK_TYPE,
    HI_UNF_AVPLAY_ATTR_ID_LOW_DELAY,
    HI_UNF_AVPLAY_ATTR_ID_TVP,
    HI_UNF_AVPLAY_ATTR_ID_BUTT
} HI_UNF_AVPLAY_ATTR_ID_E;
```

【修改原因】

设置/获取安全视频通路属性。

【注意事项】

只有支持 Trusted Video Path 特性的时候，设置 HI_UNF_AVPLAY_TVP_ATTR_S 属性 bEnable 为 TRUE。

【范例】

无。

HI_UNF_AVPLAY_STATUS_E

【结构体定义】

修改前：

```
typedef enum hiUNF_AVPLAY_STATUS_E
{
    HI_UNF_AVPLAY_STATUS_STOP = 0,
    HI_UNF_AVPLAY_STATUS_PLAY,
    HI_UNF_AVPLAY_STATUS_TPLAY,
    HI_UNF_AVPLAY_STATUS_PAUSE,
    HI_UNF_AVPLAY_STATUS_EOS,
    HI_UNF_AVPLAY_STATUS_SEEK ,

    HI_UNF_AVPLAY_STATUS_BUTT
} HI_UNF_AVPLAY_STATUS_E;
```

修改后：

```
typedef enum hiUNF_AVPLAY_STATUS_E
{
    HI_UNF_AVPLAY_STATUS_STOP = 0,
    HI_UNF_AVPLAY_STATUS_PREPLAY,
    HI_UNF_AVPLAY_STATUS_PLAY,
    HI_UNF_AVPLAY_STATUS_TPLAY,
    HI_UNF_AVPLAY_STATUS_PAUSE,
    HI_UNF_AVPLAY_STATUS_EOS,
    HI_UNF_AVPLAY_STATUS_SEEK ,

    HI_UNF_AVPLAY_STATUS_BUTT
} HI_UNF_AVPLAY_STATUS_E;
```

【修改原因】

支持 full-band 功能。

【注意事项】

调用 HI_UNF_AVPLAY_PreStart 后 Avplayer 处于 PREPLAY 状态，接收 TS 流并解复用出音视频 ES，但不解码输出。调用 HI_UNF_AVPLAY_Start 将处于 PREPLAY 状态的 Avplayer 切换至播放状态，可实现快速换台。

【范例】

无。

HI_UNF_AVPLAY_RESET_OPT_S

【结构体定义】

修改前：

```
typedef struct hiAVPLAY_RESET_OPT_S
{
    HI_U32      u32Reserved;
```

```
} HI_UNF_AVPLAY_RESET_OPT_S;
```

修改后：

```
typedef struct hiAVPLAY_RESET_OPT_S
{
    HI_U32      u32SeekPtsMs;
} HI_UNF_AVPLAY_RESET_OPT_S;
```

【修改原因】

支持精准 seek 功能。

【注意事项】

无。

【范例】

无。

HI_UNF_VIDEO_USERDATA_S

【结构体定义】

修改前：

```
typedef struct hiUNF_VIDEO_USERDATA_S
{
    HI_UNF_VIDEO_BROADCAST_PROFILE_E   enBroadcastProfile;
    HI_UNF_VIDEO_USER_DATA_POSITION_E  enPositionInStream;
    HI_U32                             u32Pts;
    HI_U32                             u32SeqCnt;
    HI_U32                             u32SeqFrameCnt;
    HI_U8                              *pu8Buffer;
    HI_U32                             u32Length;
    HI_BOOL                            bBufferOverflow;
}HI_UNF_VIDEO_USERDATA_S;
```

修改后：

```
typedef struct hiUNF_VIDEO_USERDATA_S
{
    HI_UNF_VIDEO_BROADCAST_PROFILE_E   enBroadcastProfile;
    HI_UNF_VIDEO_USER_DATA_POSITION_E  enPositionInStream;
    HI_U32                             u32Pts;
    HI_U32                             u32SeqCnt;
    HI_U32                             u32SeqFrameCnt;
    HI_U8                              *pu8Buffer;
    HI_U32                             u32Length;
    HI_BOOL                            bBufferOverflow;
```

```
    HI_BOOL                         bTopFieldFirst;
}HI_UNF_VIDEO_USERDATA_S;
```

【修改原因】

增加顶底场优先标志,确保 CC 数据能够正确解码。

【注意事项】

无。

【范例】

无。

2.2.3 函数接口

2.2.3.1 新增

HI_UNF_AVPLAY_PreStart

【接口定义】

```
HI_S32 HI_UNF_AVPLAY_PreStart(HI_HANDLE hAvplay,
HI_UNF_AVPLAY_MEDIA_CHAN_E enChn, const HI_UNF_AVPLAY_PRESTART_OPT_S
*pstPreStartOpt);
```

【新增原因】

用于启动 Avplayer 到预播放状态，接收 TS 流并解复用出音视频 ES，但不解码输出。调用 HI_UNF_AVPLAY_Start 将处于预播放状态的 Avplayer 切换至播放状态，可实现快速换台。

【注意事项】

无。

【范例】

sample/ fullband/ fbc_xswitch.c

HI_UNF_AVPLAY_PreStop

【接口定义】

```
HI_S32 HI_UNF_AVPLAY_PreStop(HI_HANDLE hAvplay,
HI_UNF_AVPLAY_MEDIA_CHAN_E enChn, const HI_UNF_AVPLAY_PRESTOP_OPT_S
*pstPreStopOpt);
```

【新增原因】

用于将 Avplayer 到预停止状态。

【注意事项】

该接口保留后续扩展。现阶段请调用 HI_UNF_AVPLAY_Stop 停止播放。

【范例】

sample/ fullband/ fbc_xswitch.c

2.3 CIPHER

2.3.1 概述

相对于 UNF 3.1 版本，UNF 3.2.0 在总体功能上有以下变更：

新增 AES 算法 CBC CTS 模式解密的功能

2.3.1.1 AES 算法 CBC CTS 模式解密的功能

由于新增 AES 算法 CBC CTS 模式解密的功能，引起的变更如下：

数据结构

修改 HI_UNF_CIPHER_WORK_MODE_E

2.3.2 数据结构

2.3.2.1 修改

HI_UNF_CIPHER_WORK_MODE_E

【结构体定义】

修改前：

```
typedef enum hiHI_UNF_CIPHER_WORK_MODE_E
{
    HI_UNF_CIPHER_WORK_MODE_ECB     = 0x0,
    HI_UNF_CIPHER_WORK_MODE_CBC     = 0x1,
    HI_UNF_CIPHER_WORK_MODE_CFB     = 0x2,
    HI_UNF_CIPHER_WORK_MODE_OFB     = 0x3,
    HI_UNF_CIPHER_WORK_MODE_CTR     = 0x4,
    HI_UNF_CIPHER_WORK_MODE_BUTT    = 0x5
}HI_UNF_CIPHER_WORK_MODE_E;
```

修改后：

```
typedef enum hiHI_UNF_CIPHER_WORK_MODE_E
{
    HI_UNF_CIPHER_WORK_MODE_ECB,
    HI_UNF_CIPHER_WORK_MODE_CBC,
    HI_UNF_CIPHER_WORK_MODE_CFB,
    HI_UNF_CIPHER_WORK_MODE_OFB,
    HI_UNF_CIPHER_WORK_MODE_CTR,
```

```
    HI_UNF_CIPHER_WORK_MODE_CBC_CTS,
    HI_UNF_CIPHER_WORK_MODE_BUTT  = 0xffffffff
}HI_UNF_CIPHER_WORK_MODE_E;
```

【修改说明】

增加 AES 算法 CBC CTS 模式的解密，可支持 Widevine DRM 的解密功能。

【注意事项】

无。

【范例】

无。

2.4 Frontend

2.4.1 概述

UNF3.2.0 相对于 UNF3.1 版本，在总体功能上有以下变更：

- 寻星仪采样长度新增 32/64/128/256 pts 项；
- 新增 DVB-T/T2 TUNER(CXD2861/Si2147)驱动支持；
- 新增 DVB-T/T2 DEMODE(CXD2837/Hi3137)驱动支持；

2.4.1.1 寻星仪采样长度新增 32/64/128/256 pts 项

数据结构

修改 HI_UNF_TUNER_SAMPLE_DATALEN_E 枚举体。

函数接口

无。

2.4.1.2 新增 DVB-T TUNER(CXD2861/Si2147)驱动支持

数据结构

修改 HI_UNF_TUNER_DEV_TYPE_E 枚举体。

函数接口

无。

2.4.1.3 新增 DVB-T DEMODE(CXD2837/Hi3137)驱动支持

该特性实现以下功能：

- 新增 CXD2837/Hi3137 DEMODE 驱动支持
- 多载波调试模式自动识别
- DVB-T2 信号时，选择接收 base 或者 lite 通道数据
- TS 同步头长度可配置为是否自动识别
- 晶振频率/复位管脚/TS 流输出模式等地面解调芯片初始化属性配置
- Hi3137 对天线供电开关打开/关闭
- 搜台时，配置仅搜索 DVB-T 信号
- 搜台时，配置仅搜索 DVB-T2 信号
- 搜台时，配置既搜索 DVB-T，也搜索 DVB-T2 信号
- 地面信号导频模式自动识别
- DVB-T2 信号时，信号是否混合自动识别
- 载波模式正常还是扩展自动识别
- 星座模式自动识别
- DVB-T 2 信号时，FEC 正常帧还是短帧自动识别
- 锁频频点参数配置，包括频率/带宽/ DVB-T 2 通道选择/ DVB-T 高低优先级码流选择
- 地面信号信息识别，包括频率/带宽/调制模式/FEC 码率/保护间隔/快速傅里叶变换大小/TS 码流优先级/PLP 模式/导频模式/载波模式/星座模式/FEC 帧长
- 向上层软件上报地面 TUNER 搜台状态，包括空闲/搜台中/完成/退出/失败
- 搜台成功后，保存频点信息，包括频率/带宽/DVB 模式/PLP 索引号/PLP ID/COMMON PLP ID/COMBINATION FLAG/TS 码流优先级
- 地面 TUNER 搜台信息上报，包括状态/进度/搜索到频点
- 地面 TUNER 信号单频点搜索前参数配置，包括频率/带宽/搜台模式/ DVB-T2 通道选择
- 地面 TUNER 搜台初始化配置
- 地面 TUNER 搜台所有频点信息存储
- 地面 TUNER 搜台总共发现频点数
- 识别物理层管道信息，包括索引号/管道 ID/管道组 ID/管道类型
- 识别物理层管道组合信息，包括管道 ID/共享管道 ID/组合标志/ DVB-T 2 通道选择
- 地面 TUNER 单频点搜台信息，包括节目数/DVB 模式/ DVB-T 2 通道选择/ DVB-T 2 信号模式/各物理层管道描述

引起的变更如下：

数据结构

- 修改 HI_UNF_DEMOD_DEV_TYPE_E 枚举体
- 修改 HI_UNF_TUNER_FE_FFT_E 枚举体
- 新增 HI_UNF_TUNER_TER_MODE_E 枚举体
- 新增 HI_UNF_TUNER_TS_SYNC_HEAD_E 枚举体
- 新增 HI_UNF_TUNER_TER_ATTR_S 结构体

- 新增 HI_UNF_TUNER_TER_ANTENNA_POWER_E 枚举体
- 新增 HI_UNF_TUNER_TER_SCAN_MODE_E 枚举体
- 新增 HI_UNF_TUNER_TER_PILOT_PATTERN_E 枚举体
- 新增 HI_UNF_TUNER_TER_CHANNEL_MODE_E 枚举体
- 新增 HI_UNF_TUNER_TER_CARRIER_MODE_E 枚举体
- 新增 HI_UNF_TUNER_CONSTELLATION_MODE_E 枚举体
- 新增 HI_UNF_TUNER_TER_FEC_FRAME_MODE_E 枚举体
- 新增 HI_UNF_TER_CONNECT_PARA_S 结构体
- 修改 HI_UNF_TUNER_TER_SIGNALINFO_S 结构体
- 新增 HI_UNF_TUNER_TER_SCAN_STATUS_E 枚举体
- 新增 HI_UNF_TUNER_TER_CHANNEL_ATTR_S 结构体
- 新增 HI_UNF_TUNER_TER_SCAN_NOTIFY_U 结构体
- 新增 HI_UNF_TUNER_TER_SCAN_ATTR_S 结构体
- 新增 HI_UNF_TUNER_TER_SCAN_PARA_S 结构体
- 新增 HI_UNF_TUNER_TER_PLP_ATTR_S 结构体
- 新增 HI_UNF_TUNER_TER_ACC_S 结构体
- 新增 HI_UNF_TUNER_TER_TPINFO_S 结构体

函数接口

- 新增 HI_UNF_TUNER_SetTerAttr 函数
- 新增 HI_UNF_TUNER_SetAntennaPower 函数
- 新增 HI_UNF_TUNER_SetCommonPLPID 函数
- 新增 HI_UNF_TUNER_SetCommonPLPCombination 函数
- 新增 HI_UNF_TUNER_TerScanStart 函数
- 新增 HI_UNF_TUNER_TerScanStop
- 新增 HI_UNF_TUNER_SetPLPMode 函数
- 新增 HI_UNF_TUNER_GetPLPId 函数
- 新增 HI_UNF_TUNER_GetPLPGrpId 函数

2.4.2 数据结构

2.4.2.1 新增

HI_UNF_TUNER_TER_MODE_E

【结构体定义】

```
typedef enum hiUNF_TUNER_TER_MODE_E
{
    HI_UNF_TUNER_TER_MODE_BASE = 0,
    HI_UNF_TUNER_TER_MODE_LITE,
```

```
    HI_UNF_TUNER_TER_MODE_BUTT
} HI_UNF_TUNER_TER_MODE_E;
```

【新增原因】

新特性，DVB-T/T2 信号接收。

【注意事项】

无。

【范例】

请参考 sample/tuner/tuner_demo.c。

HI_UNF_TUNER_TS_SYNC_HEAD_E

【结构体定义】

```
typedef enum hiUNF_TUNER_TS_SYNC_HEAD_E
{
    HI_UNF_TUNER_TS_SYNC_HEAD_AUTO,
    HI_UNF_TUNER_TS_SYNC_HEAD_8BIT,
    HI_UNF_TUNER_TS_SYNC_HEAD_BUTT
} HI_UNF_TUNER_TS_SYNC_HEAD_E;
```

【新增原因】

新特性，DVB-T/T2 信号接收。

【注意事项】

无。

【范例】

请参考 sample/tuner/tuner_demo.c。

HI_UNF_TUNER_TER_ATTR_S

【结构体定义】

```
typedef struct  hiUNF_TUNER_TER_ATTR_S
{
    HI_U32                      u32DemodClk;
    HI_U32                      u32ResetGpioNo;
    HI_U16                      u16TunerMaxLPF;
    HI_U16                      u16TunerI2CClk;
    HI_UNF_TUNER_RFAGC_MODE_E     enRFAGC;
    HI_UNF_TUNER_IQSPECTRUM_MODE_E enIQSpectrum;
    HI_UNF_TUNER_TSCLK_POLAR_E    enTSClkPolar;
    HI_UNF_TUNER_TS_FORMAT_E      enTSFormat;
    HI_UNF_TUNER_TS_SERIAL_PIN_E   enTSSerialPIN;
```

```
    HI_UNF_TUNER_TS_SYNC_HEAD_E    enTSSyncHead;
} HI_UNF_TUNER_TER_ATTR_S;
```

【新增原因】

新特性，DVB-T/T2 信号接收。

【注意事项】

无。

【范例】

请参考 sample/tuner/tuner_demo.c。

HI_UNF_TUNER_TER_ANTENNA_POWER_E

【结构体定义】

```
typedef enum hiUNF_TUNER_TER_ANTENNA_POWER_E
{
    HI_UNF_TUNER_TER_ANTENNA_POWER_OFF,
    HI_UNF_TUNER_TER_ANTENNA_POWER_ON,
    HI_UNF_TUNER_TER_ANTENNA_POWER_BUTT
} HI_UNF_TUNER_TER_ANTENNA_POWER_E;
```

【新增原因】

新特性，DVB-T/T2 信号接收。

【注意事项】

无。

【范例】

请参考 sample/tuner/tuner_demo.c。

HI_UNF_TUNER_TER_SCAN_MODE_E

【结构体定义】

```
typedef enum hiUNF_TUNER_TER_SCAN_MODE_E
{
    HI_UNF_TUNER_TER_SCAN_DVB_T2 = 0,
    HI_UNF_TUNER_TER_SCAN_DVB_T,
    HI_UNF_TUNER_TER_SCAN_DVB_T_T2_ALL,
    HI_UNF_TUNER_TER_SCAN_DVB_T_T2_BUTT
} HI_UNF_TUNER_TER_SCAN_MODE_E;
```

【新增原因】

新特性，DVB-T/T2 信号接收。

【注意事项】

无。

【范例】

请参考 sample/tuner/tuner_demo.c。

HI_UNF_TUNER_TER_PILOT_PATTERN_E

【结构体定义】

```
typedef enum hiUNF_TUNER_TER_PILOT_PATTERN_E
{
    HI_UNF_TUNER_T2_PILOT_PATTERN_PP1=0,
    HI_UNF_TUNER_T2_PILOT_PATTERN_PP2,
    HI_UNF_TUNER_T2_PILOT_PATTERN_PP3,
    HI_UNF_TUNER_T2_PILOT_PATTERN_PP4,
    HI_UNF_TUNER_T2_PILOT_PATTERN_PP5,
    HI_UNF_TUNER_T2_PILOT_PATTERN_PP6,
    HI_UNF_TUNER_T2_PILOT_PATTERN_PP7,
    HI_UNF_TUNER_T2_PILOT_PATTERN_PP8,
    HI_UNF_TUNER_T2_PILOT_PATTERN_BUTT
} HI_UNF_TUNER_TER_PILOT_PATTERN_E;
```

【新增原因】

新特性，DVB-T/T2 信号接收。

【注意事项】

无。

【范例】

请参考 sample/tuner/tuner_demo.c。

HI_UNF_TUNER_TER_CHANNEL_MODE_E

【结构体定义】

```
typedef enum hiUNF_TUNER_TER_CHANNEL_MODE_E
{
    HI_UNF_TUNER_TER_PURE_CHANNEL = 0,
    HI_UNF_TUNER_TER_MIXED_CHANNEL,
    HI_UNF_TUNER_TER_CHANNEL_MODE_BUTT
} HI_UNF_TUNER_TER_CHANNEL_MODE_E;
```

【新增原因】

新特性，DVB-T/T2 信号接收。

【注意事项】

无。

【范例】

请参考 sample/tuner/tuner_demo.c。

HI_UNF_TUNER_TER_CARRIER_MODE_E

【结构体定义】

```
typedef enum hiUNF_TUNER_TER_CARRIER_MODE_E
{
    HI_UNF_TUNER_TER_EXTEND_CARRIER = 0,
    HI_UNF_TUNER_TER_NORMAL_CARRIER,
    HI_UNF_TUNER_TER_CARRIER_MODE_BUTT
} HI_UNF_TUNER_TER_CARRIER_MODE_E;
```

【新增原因】

新特性，DVB-T/T2 信号接收。

【注意事项】

无。

【范例】

请参考 sample/tuner/tuner_demo.c。

HI_UNF_TUNER_CONSTELLATION_MODE_E

【结构体定义】

```
typedef enum hiUNF_TUNER_CONSTELLATION_MODE_E
{
    HI_UNF_TUNER_CONSTELLATION_STANDARD = 0,
    HI_UNF_TUNER_CONSTELLATION_ROTATION,
    HI_UNF_TUNER_CONSTELLATION_MODE_BUTT
} HI_UNF_TUNER_CONSTELLATION_MODE_E;
```

【新增原因】

新特性，DVB-T/T2 信号接收。

【注意事项】

无。

【范例】

请参考 sample/tuner/tuner_demo.c。

HI_UNF_TUNER_TER_FEC_FRAME_MODE_E

【结构体定义】

```
typedef enum hiUNF_TUNER_TER_FEC_FRAME_MODE_E
{
    HI_UNF_TUNER_TER_FEC_FRAME_NORMAL = 0,
    HI_UNF_TUNER_TER_FEC_FRAME_SHORT,
    HI_UNF_TUNER_TER_FEC_FRAME_MODE_BUTT
} HI_UNF_TUNER_TER_FEC_FRAME_MODE_E;
```

【新增原因】

新特性，DVB-T/T2 信号接收。

【注意事项】

无。

【范例】

请参考 sample/tuner/tuner_demo.c。

HI_UNF_TER_CONNECT_PARA_S

【结构体定义】

```
typedef struct  hiUNF_TER_CONNECT_PARA_S
{
    HI_U32                  u32Freq;
    HI_U32                  u32BandWidth;
    HI_UNF_MODULATION_TYPE_E    enModType;
    HI_BOOL                 bReverse;
    HI_UNF_TUNER_TER_MODE_E     enChannelMode;
    HI_UNF_TUNER_TS_PRIORITY_E  enDVBTPrio;
} HI_UNF_TER_CONNECT_PARA_S;
```

【新增原因】

新特性，DVB-T/T2 信号接收。

【注意事项】

无。

【范例】

请参考 sample/tuner/tuner_demo.c。

HI_UNF_TUNER_TER_SCAN_STATUS_E

【结构体定义】

```
typedef enum hiUNF_TUNER_TER_SCAN_STATUS_E
```

```
{
    HI_UNF_TUNER_TER_SCAN_STATUS_IDLE,
    HI_UNF_TUNER_TER_SCAN_STATUS_SCANNING,
    HI_UNF_TUNER_TER_SCAN_STATUS_FINISH,
    HI_UNF_TUNER_TER_SCAN_STATUS_QUIT,
    HI_UNF_TUNER_TER_SCAN_STATUS_FAIL,
    HI_UNF_TUNER_TER_SCAN_STATUS_BUTT
} HI_UNF_TUNER_TER_SCAN_STATUS_E;
```

【新增原因】

新特性，DVB-T/T2 信号接收。

【注意事项】

无。

【范例】

请参考 sample/tuner/tuner_demo.c。

HI_UNF_TUNER_TER_CHANNEL_ATTR_S

【结构体定义】

```
typedef struct hiUNF_TUNER_TER_CHANNEL_ATTR_S
{
    HI_U32 u32Frequency;
    HI_U32 u32BandWidth;
    HI_U8  u8DVBTMode;
    HI_U8  u8PlpIndex;
    HI_U8  u8PlpId;
    HI_U8  u8CommId;
    HI_U8  u8Combination;
    HI_UNF_TUNER_TER_MODE_E     enChannelMode;
    HI_UNF_TUNER_TS_PRIORITY_E  enTSPri;
} HI_UNF_TUNER_TER_CHANNEL_ATTR_S;
```

【新增原因】

新特性，DVB-T/T2 信号接收。

【注意事项】

无。

【范例】

请参考 sample/tuner/tuner_demo.c。

HI_UNF_TUNER_TER_SCAN_NOTIFY_U

【结构体定义】

```
typedef union hiUNF_TUNER_TER_SCAN_NOTIFY_U
{
    HI_UNF_TUNER_TER_SCAN_STATUS_E* penStatus;
    HI_U16*                         pu16ProgressPercent;
    HI_UNF_TUNER_TER_CHANNEL_ATTR_S *pstResult;
} HI_UNF_TUNER_TER_SCAN_NOTIFY_U;
```

【新增原因】

新特性，DVB-T/T2 信号接收。

【注意事项】

无。

【范例】

请参考 sample/tuner/tuner_demo.c。

HI_UNF_TUNER_TER_SCAN_ATTR_S

【结构体定义】

```
typedef struct hiUNF_TUNER_TER_SCAN_ATTR_S
{
    HI_U32 u32Frequency;
    HI_U32 u32BandWidth;
    HI_UNF_TUNER_TER_SCAN_MODE_E enScanMode;
    HI_BOOL  bScanLite;
    HI_VOID (*pfnEVTNotify)(HI_U32 u32TunerId,
HI_UNF_TUNER_TER_SCAN_STATUS_E enEVT, HI_UNF_TUNER_TER_SCAN_NOTIFY_U *
punNotify);
}HI_UNF_TUNER_TER_SCAN_ATTR_S;
```

【新增原因】

新特性，DVB-T/T2 信号接收。

【注意事项】

无。

【范例】

请参考 sample/tuner/tuner_demo.c。

HI_UNF_TUNER_TER_SCAN_PARA_S

【结构体定义】

```
typedef struct hiUNF_TUNER_TER_SCAN_PARA_S
{
    HI_UNF_TUNER_TER_SCAN_ATTR_S stTer;
    HI_UNF_TUNER_TER_CHANNEL_ATTR_S enChanArray[TER_MAX_TP];
    HI_U32 u32ChanNum;
}HI_UNF_TUNER_TER_SCAN_PARA_S;
```

【新增原因】

新特性，DVB-T/T2 信号接收。

【注意事项】

无。

【范例】

请参考 sample/tuner/tuner_demo.c。

HI_UNF_TUNER_TER_PLP_ATTR_S

【结构体定义】

```
typedef struct hiUNF_TUNER_TER_PLP_ATTR_S
{
    HI_U8                       u8PlpIndex;
    HI_U8                       u8PlpId;
    HI_U8                       u8PlpGrpId;
    HI_UNF_TUNER_T2_PLP_TYPE_E  enPlpType;
} HI_UNF_TUNER_TER_PLP_ATTR_S;
```

【新增原因】

新特性，DVB-T/T2 信号接收。

【注意事项】

无。

【范例】

请参考 sample/tuner/tuner_demo.c。

HI_UNF_TUNER_TER_ACC_S

【结构体定义】

```
typedef struct hiUNF_TUNER_TER_ACC_S
{
    HI_U8                       u8PlpId;
    HI_U8                       u8CommPlpId;
    HI_U8                       u8Combination;
    HI_UNF_TUNER_TER_MODE_E     enChannelAttr;
```

```
} HI_UNF_TUNER_TER_ACC_S;
```

【新增原因】

新特性，DVB-T/T2 信号接收。

【注意事项】

无。

【范例】

请参考 sample/tuner/tuner_demo.c。

HI_UNF_TUNER_TER_TPINFO_S

【结构体定义】

```
typedef struct hiUNF_TUNER_TER_TPINFO_S
{
    HI_U8                           u8ProgNum;
    HI_U8                           u8DVBTMode;
    HI_U8                           u8DVBTHier;
    HI_UNF_TUNER_TER_MODE_E         enChannelAttr;
    HI_UNF_TUNER_TER_CHANNEL_MODE_E enChannelMode;
    HI_UNF_TUNER_TER_PLP_ATTR_S     enPlpAttr[16];
} HI_UNF_TUNER_TER_TPINFO_S;
```

【新增原因】

新特性，DVB-T/T2 信号接收。

【注意事项】

无。

【范例】

请参考 sample/tuner/tuner_demo.c。

2.4.2.2 修改

HI_UNF_TUNER_SAMPLE_DATALEN_E

【结构体定义】

修改前：

```
typedef enum hiUNF_TUNER_SAMPLE_DATALEN_E
{
    HI_UNF_TUNER_SAMPLE_DATALEN_512,
    HI_UNF_TUNER_SAMPLE_DATALEN_1024,
    HI_UNF_TUNER_SAMPLE_DATALEN_2048,
    HI_UNF_TUNER_SAMPLE_DATALEN_BUTT
```

```
} HI_UNF_TUNER_SAMPLE_DATALEN_E;
```

修改后：

```
typedef enum hiUNF_TUNER_SAMPLE_DATALEN_E
{
    HI_UNF_TUNER_SAMPLE_DATALEN_32,
    HI_UNF_TUNER_SAMPLE_DATALEN_64,
    HI_UNF_TUNER_SAMPLE_DATALEN_128,
    HI_UNF_TUNER_SAMPLE_DATALEN_256,
    HI_UNF_TUNER_SAMPLE_DATALEN_512,
    HI_UNF_TUNER_SAMPLE_DATALEN_1024,
    HI_UNF_TUNER_SAMPLE_DATALEN_2048,
    HI_UNF_TUNER_SAMPLE_DATALEN_BUTT
} HI_UNF_TUNER_SAMPLE_DATALEN_E;
```

【修改原因】

寻星仪每次采样长度新增 32/64/128/256 pts 项。

【注意事项】

无。

【范例】

请参考 sample/tuner/tuner_demo.c

HI_UNF_TUNER_DEV_TYPE_E

【结构体定义】

修改前：

```
typedef enum    hiUNF_TUNER_DEV_TYPE_E
{
    HI_UNF_TUNER_DEV_TYPE_XG_3BL,
    HI_UNF_TUNER_DEV_TYPE_CD1616,
    HI_UNF_TUNER_DEV_TYPE_ALPS_TDAE,
    HI_UNF_TUNER_DEV_TYPE_TDCC,
    HI_UNF_TUNER_DEV_TYPE_TDA18250,
    HI_UNF_TUNER_DEV_TYPE_CD1616_DOUBLE,
    HI_UNF_TUNER_DEV_TYPE_MT2081,
    HI_UNF_TUNER_DEV_TYPE_TMX7070X,
    HI_UNF_TUNER_DEV_TYPE_R820C,
     HI_UNF_TUNER_DEV_TYPE_MXL203,
    HI_UNF_TUNER_DEV_TYPE_AV2011,
    HI_UNF_TUNER_DEV_TYPE_SHARP7903,
    HI_UNF_TUNER_DEV_TYPE_MXL101,
    HI_UNF_TUNER_DEV_TYPE_MXL603,
```

```
    HI_UNF_TUNER_DEV_TYPE_IT9170,
    HI_UNF_TUNER_DEV_TYPE_IT9133,
    HI_UNF_TUNER_DEV_TYPE_TDA6651,
    HI_UNF_TUNER_DEV_TYPE_TDA18250B,
    HI_UNF_TUNER_DEV_TYPE_M88TS2022,
    HI_UNF_TUNER_DEV_TYPE_RDA5815,
    HI_UNF_TUNER_DEV_TYPE_MXL254,
    HI_UNF_TUNER_DEV_TYPE_BUTT,
}HI_UNF_TUNER_DEV_TYPE_E ;
```

修改后：

```
typedef enum    hiUNF_TUNER_DEV_TYPE_E
{
    HI_UNF_TUNER_DEV_TYPE_XG_3BL,
    HI_UNF_TUNER_DEV_TYPE_CD1616,
    HI_UNF_TUNER_DEV_TYPE_ALPS_TDAE,
    HI_UNF_TUNER_DEV_TYPE_TDCC,
    HI_UNF_TUNER_DEV_TYPE_TDA18250,
    HI_UNF_TUNER_DEV_TYPE_CD1616_DOUBLE,
    HI_UNF_TUNER_DEV_TYPE_MT2081,
    HI_UNF_TUNER_DEV_TYPE_TMX7070X,
    HI_UNF_TUNER_DEV_TYPE_R820C,
    HI_UNF_TUNER_DEV_TYPE_MXL203,
    HI_UNF_TUNER_DEV_TYPE_AV2011,
    HI_UNF_TUNER_DEV_TYPE_SHARP7903,
    HI_UNF_TUNER_DEV_TYPE_MXL101,
    HI_UNF_TUNER_DEV_TYPE_MXL603,
    HI_UNF_TUNER_DEV_TYPE_IT9170,
    HI_UNF_TUNER_DEV_TYPE_IT9133,
    HI_UNF_TUNER_DEV_TYPE_TDA6651,
    HI_UNF_TUNER_DEV_TYPE_TDA18250B,
    HI_UNF_TUNER_DEV_TYPE_M88TS2022,
    HI_UNF_TUNER_DEV_TYPE_RDA5815,
    HI_UNF_TUNER_DEV_TYPE_MXL254,
    HI_UNF_TUNER_DEV_TYPE_CXD2861,
    HI_UNF_TUNER_DEV_TYPE_SI2147,
    HI_UNF_TUNER_DEV_TYPE_BUTT
} HI_UNF_TUNER_DEV_TYPE_E ;
```

【修改原因】

新增 CXD2861/Si2147 TUNER 驱动支持。

【注意事项】

CXD2861 搭配 CXD2837 接收 DVB-T/T2 信号，Si2147 搭配 Hi3137 接收 DVB-T/T2 信号。

【范例】

请参考 sample/tuner/tuner_demo.c

HI_UNF_DEMOD_DEV_TYPE_E

【结构体定义】

修改前：

```
typedef enum    hiUNF_DEMOD_DEV_TYPE_E
{
    HI_UNF_DEMOD_DEV_TYPE_NONE,
    HI_UNF_DEMOD_DEV_TYPE_3130I= 0x100,
    HI_UNF_DEMOD_DEV_TYPE_3130E,
    HI_UNF_DEMOD_DEV_TYPE_J83B,
    HI_UNF_DEMOD_DEV_TYPE_AVL6211,
    HI_UNF_DEMOD_DEV_TYPE_MXL101,
    HI_UNF_DEMOD_DEV_TYPE_MN88472,
    HI_UNF_DEMOD_DEV_TYPE_IT9170,
    HI_UNF_DEMOD_DEV_TYPE_IT9133,
    HI_UNF_DEMOD_DEV_TYPE_3136,
    HI_UNF_DEMOD_DEV_TYPE_3136I,
    HI_UNF_DEMOD_DEV_TYPE_MXL254,
    HI_UNF_DEMOD_DEV_TYPE_BUTT,
}HI_UNF_DEMOD_DEV_TYPE_E ;
```

修改后：

```
typedef enum    hiUNF_DEMOD_DEV_TYPE_E
{
    HI_UNF_DEMOD_DEV_TYPE_NONE,
    HI_UNF_DEMOD_DEV_TYPE_3130I = 0x100,
    HI_UNF_DEMOD_DEV_TYPE_3130E,
    HI_UNF_DEMOD_DEV_TYPE_J83B,
    HI_UNF_DEMOD_DEV_TYPE_AVL6211,
    HI_UNF_DEMOD_DEV_TYPE_MXL101,
    HI_UNF_DEMOD_DEV_TYPE_MN88472,
    HI_UNF_DEMOD_DEV_TYPE_IT9170,
    HI_UNF_DEMOD_DEV_TYPE_IT9133,
    HI_UNF_DEMOD_DEV_TYPE_3136,
    HI_UNF_DEMOD_DEV_TYPE_3136I,
    HI_UNF_DEMOD_DEV_TYPE_MXL254,
    HI_UNF_DEMOD_DEV_TYPE_CXD2837,
    HI_UNF_DEMOD_DEV_TYPE_3137,
```

```
    HI_UNF_DEMOD_DEV_TYPE_BUTT
} HI_UNF_DEMOD_DEV_TYPE_E;
```

【修改原因】

新增 CXD2837/Hi3137 DEMODE 驱动支持。

【注意事项】

CXD2861 搭配 CXD2837 接收 DVB-T/T2 信号，Si2147 搭配 Hi3137 接收 DVB-T/T2 信号。

【范例】

请参考 sample/tuner/tuner_demo.c

HI_UNF_TUNER_FE_FFT_E

【结构体定义】

修改前：

```
typedef enum hiUNF_TUNER_FE_FFT_E
{
    HI_UNF_TUNER_FE_FFT_DEFAULT = 0,
    HI_UNF_TUNER_FE_FFT_1K ,
    HI_UNF_TUNER_FE_FFT_2K ,
    HI_UNF_TUNER_FE_FFT_4K ,
    HI_UNF_TUNER_FE_FFT_8K ,
    HI_UNF_TUNER_FE_FFT_16K ,
    HI_UNF_TUNER_FE_FFT_32K ,
    HI_UNF_TUNER_FE_FFT_BUTT ,
}HI_UNF_TUNER_FE_FFT_E;
```

修改后：

```
typedef enum hiUNF_TUNER_FE_FFT_E
{
    HI_UNF_TUNER_FE_FFT_DEFAULT = 0,
    HI_UNF_TUNER_FE_FFT_1K ,
    HI_UNF_TUNER_FE_FFT_2K ,
    HI_UNF_TUNER_FE_FFT_4K ,
    HI_UNF_TUNER_FE_FFT_8K ,
    HI_UNF_TUNER_FE_FFT_16K ,
    HI_UNF_TUNER_FE_FFT_32K ,
    HI_UNF_TUNER_FE_FFT_64K ,
    HI_UNF_TUNER_FE_FFT_BUTT
} HI_UNF_TUNER_FE_FFT_E;
```

【修改原因】

FFT 大小新增 64K 项。

【注意事项】

无。

【范例】

请参考 sample/tuner/tuner_demo.c

HI_UNF_TUNER_TER_SIGNALINFO_S

【结构体定义】

修改前：

```
typedef struct  hiUNF_TUNER_TER_SIGNALINFO_S
{
    HI_U32                      u32Freq;
    HI_U32                      u32BandWidth;
    HI_UNF_MODULATION_TYPE_E      enModType;
    HI_UNF_TUNER_FE_FECRATE_E     enFECRate;
    HI_UNF_TUNER_FE_GUARD_INTV_E enGuardIntv;
    HI_UNF_TUNER_FE_FFT_E enFFTMode;
    HI_UNF_TUNER_FE_HIERARCHY_E enHierMod;
    HI_UNF_TUNER_TS_PRIORITY_E enTsPriority;
} HI_UNF_TUNER_TER_SIGNALINFO_S;
```

修改后：

```
typedef struct  hiUNF_TUNER_TER_SIGNALINFO_S
{
    HI_U32                          u32Freq;
    HI_U32                          u32BandWidth;
    HI_UNF_MODULATION_TYPE_E          enModType;
    HI_UNF_TUNER_FE_FECRATE_E         enFECRate;
    HI_UNF_TUNER_FE_FECRATE_E         enLowPriFECRate;
    HI_UNF_TUNER_FE_GUARD_INTV_E       enGuardIntv;
    HI_UNF_TUNER_FE_FFT_E             enFFTMode;
    HI_UNF_TUNER_FE_HIERARCHY_E        enHierMod;
    HI_UNF_TUNER_TS_PRIORITY_E         enTsPriority;
    HI_UNF_TUNER_T2_PLP_TYPE_E         enPLPType;
    HI_UNF_TUNER_TER_PILOT_PATTERN_E    enPilotPattern;
    HI_UNF_TUNER_TER_CARRIER_MODE_E     enCarrierMode;
    HI_UNF_TUNER_CONSTELLATION_MODE_E   enConstellationMode;
    HI_UNF_TUNER_TER_FEC_FRAME_MODE_E   enFECFrameMode;
} HI_UNF_TUNER_TER_SIGNALINFO_S;
```

【修改原因】

新增低优先级码流 FEC 速率/物理层管道类型/导频模式/载波模式/星座模式/帧长模式。

【注意事项】

无。

【范例】

请参考 sample/tuner/tuner_demo.c

2.4.3 函数接口

2.4.3.1 新增

HI_UNF_TUNER_SetTerAttr

【接口定义】

```
HI_S32 HI_UNF_TUNER_SetTerAttr(HI_U32  u32tunerId , const
HI_UNF_TUNER_TER_ATTR_S *pstTerTunerAttr);
```

【新增原因】

新特性。

【注意事项】

地面 TUNER 初始化调用。

【范例】

无。

HI_UNF_TUNER_SetPLPMode

【接口定义】

```
HI_S32 HI_UNF_TUNER_SetPLPMode(HI_U32 u32TunerId, HI_U8 u8Mode);
```

【新增原因】

新特性。

【注意事项】

地面 TUNER 接收 DVB-T2 信号时调用。

【范例】

无。

HI_UNF_TUNER_SetCommonPLPID

【接口定义】

```
HI_S32 HI_UNF_TUNER_SetCommonPLPID(HI_U32 u32TunerId, HI_U8 u8PLPID);
```

【新增原因】

新特性。

【注意事项】

地面 TUNER 接收 DVB-T2 信号时调用。

【范例】

无。

HI_UNF_TUNER_SetCommonPLPCombination

【接口定义】

```
HI_S32 HI_UNF_TUNER_SetCommonPLPCombination(HI_U32 u32TunerId, HI_U8
u8PLPID);
```

【新增原因】

新特性。

【注意事项】

地面 TUNER 接收 DVB-T2 信号时调用。

【范例】

无。

HI_UNF_TUNER_GetPLPId

【接口定义】

```
HI_S32 HI_UNF_TUNER_GetPLPId(HI_U32 u32TunerId, HI_U8 *pu8PLPId);
```

【新增原因】

新特性。

【注意事项】

地面 TUNER 接收 DVB-T2 信号时调用。

【范例】

无。

HI_UNF_TUNER_GetPLPGrpId

【接口定义】

```
HI_S32 HI_UNF_TUNER_GetPLPGrpId(HI_U32 u32TunerId, HI_U8 *pu8PLPGrpId);
```

【新增原因】

新特性。

【注意事项】

地面 TUNER 接收 DVB-T2 信号时调用。

【范例】

无。

HI_UNF_TUNER_SetAntennaPower

【接口定义】

```
HI_S32 HI_UNF_TUNER_SetAntennaPower(HI_U32 u32TunerId,
HI_UNF_TUNER_TER_ANTENNA_POWER_E enPower);
```

【新增原因】

新特性。

【注意事项】

地面 TUNER 接收信号调用。

【范例】

无。

HI_UNF_TUNER_TerScanStart

【接口定义】

```
HI_S32 HI_UNF_TUNER_SetTerAttr(HI_U32  u32tunerId , const
HI_UNF_TUNER_TER_ATTR_S *pstTerTunerAttr);
```

【新增原因】

新特性。

【注意事项】

地面 TUNER 接收信号调用。

【范例】

无。

HI_UNF_TUNER_TerScanStop

【接口定义】

```
HI_S32 HI_UNF_TUNER_TerScanStop( HI_U32 u32TunerId);
```

【新增原因】

新特性。

【注意事项】

地面 TUNER 接收信号调用。

【范例】

无。

2.5 HDMI

2.5.1 概述

UNF 3.2.0 相对于 UNF 3.1 版本，在总体功能上有以下变更：

- 支持获取 HDCP 握手状态
- 支持注册 CEC 回调函数

2.5.1.1 获取 HDCP 握手状态

用于获取握手状态和 Bksv 来给 Miracast+HDCP 2.2 使用。引起的变更如下：

数据结构

修改 HI_UNF_HDMI_STATUS_S

2.5.1.2 注册 CEC 回调函数

支持通过回调函数处理 CEC 消息，简化客户的开发流程。引起的变更如下：

数据结构

新增 HI_UNF_HDMI_CECCALLBACK

函数接口

新增 HI_UNF_HDMI_RegCECCallBackFunc

新增 HI_UNF_HDMI_UnRegCECCallBackFunc

2.5.2 数据结构

2.5.2.1 新增

HI_UNF_HDMI_CECCALLBACK

【结构体定义】

```
typedef HI_VOID (*HI_UNF_HDMI_CECCALLBACK)(HI_UNF_HDMI_ID_E enHdmi,
HI_UNF_HDMI_CEC_CMD_S *pstCECCmd, HI_VOID *pData);
```

【修改原因】

CEC 回调函数需要定义统一的注册事件类型

【注意事项】

无。

【范例】

无。

2.5.2.2 修改

HI_UNF_HDMI_STATUS_S

【结构体定义】

修改前：

```
typedef struct hiUNF_HDMI_STATUS_S
{
    HI_BOOL             bConnected;
    HI_BOOL             bSinkPowerOn;
}HI_UNF_HDMI_STATUS_S;
```

修改后：

```
typedef struct hiUNF_HDMI_STATUS_S
{
    HI_BOOL             bConnected;
    HI_BOOL             bSinkPowerOn;
    HI_BOOL             bAuthed;
    HI_U8               u8Bksv[5];
}HI_UNF_HDMI_STATUS_S;
```

【修改原因】

Miracast+HDCP 2.2 的功能，需要获取 BKSV。

【注意事项】

使用 u8Bksv 之前需要先检测 bAuthed 是否授权完成，HDCP 授权完成时 Bksv 才有效。

【范例】

无。

2.5.3 函数接口

2.5.3.1 新增

HI_UNF_HDMI_RegCECCallBackFunc

【接口定义】

```
HI_S32 HI_UNF_HDMI_RegCECCallBackFunc(HI_UNF_HDMI_ID_E enHdmi,
HI_UNF_HDMI_CECCALLBACK pCECCallback);
```

【新增原因】

CEC 事件注册函数。

【注意事项】

CEC 事件目前还不支持多进程，同时注册多次以最后一次为主。且 HI_UNF_HDMI_GetCECCommand 和注册 CEC 事件函数之间只能使用一个。同时使用则会丢失消息。

【范例】

无。

HI_UNF_HDMI_UnRegCECCallBackFunc

【接口定义】

```
HI_S32 HI_UNF_HDMI_UnRegCECCallBackFunc(HI_UNF_HDMI_ID_E enHdmi,
HI_UNF_HDMI_CECCALLBACK pCECCallback);
```

【新增原因】

CEC 事件注销函数

【注意事项】

无。

【范例】

无。

2.6 VO

2.6.1 概述

UNF 3.2.0 相对于 UNF 3.1 版本，在总体功能上有以下变更：

- 支持创建采用物理坐标系的窗口
- 支持获取窗口冻结状态
- 支持获取窗口快速输出状态
- 支持获取窗口场播放模式
- 支持获取窗口 3D 输出景深

2.6.1.1 支持创建采用物理坐标系的窗口

创建窗口时，指定窗口坐标系采用物理坐标系。引起的变更如下：

数据结构

无。

函数接口

新增 HI_UNF_VO_CreateWindowExt

2.6.1.2 支持获取窗口冻结状态

与 HI_UNF_VO_FreezeWindow 接口对应，用户可通过该接口获取窗口冻结状态。引起的变更如下：

数据结构

无。

函数接口

新增 HI_UNF_VO_GetWindowFreezeStatus

2.6.1.3 支持获取窗口快速输出状态

与 HI_UNF_VO_SetQuickOutputEnable 对应，用户可通过该接口获取窗口快速输出状态。引起的变更如下：

数据结构

无。

函数接口

新增 HI_UNF_VO_GetQuickOutputStatus

2.6.1.4 支持获取窗口场播放模式

与 HI_UNF_VO_SetWindowFieldMode 对应，用户可通过该接口获取窗口场播放模式。引起的变更如下：

数据结构

无。

函数接口

新增 HI_UNF_VO_GetWindowFieldMode

2.6.1.5 支持获取窗口 3D 输出景深

与 HI_UNF_VO_SetStereoDetpth 对应，用户可通过该接口获取窗口 3D 输出景深。引起的变更如下：

数据结构

无。

函数接口

新增 HI_UNF_VO_GetStereoDetpth

2.6.2 函数接口

2.6.2.1 新增

HI_UNF_VO_CreateWindowExt

【接口定义】

```
HI_S32 HI_UNF_VO_CreateWindowExt(const HI_UNF_WINDOW_ATTR_S* pWinAttr,
                                 HI_HANDLE *phWindow,
                                 HI_BOOL bVirtScreen);
```

【新增原因】

为支持窗口采用物理坐标系

【注意事项】

窗口如果采用该接口创建，则在窗口生命周期内均采用物理坐标系

【范例】

无。

HI_UNF_VO_GetWindowFreezeStatus

【接口定义】

```
HI_S32 HI_UNF_VO_GetWindowFreezeStatus(HI_HANDLE hWindow, HI_BOOL
*pbEnable, HI_UNF_WINDOW_FREEZE_MODE_E *penWinFreezeMode);
```

【新增原因】

为支持获取窗口冻结状态

【注意事项】

无。

【范例】

无。

HI_UNF_VO_GetQuickOutputStatus

【接口定义】

```
HI_S32 HI_UNF_VO_GetQuickOutputStatus(HI_HANDLE hWindow, HI_BOOL
*pbQuickOutputEnable);
```

【新增原因】

为支持获取窗口快速输出状态

【注意事项】

无。

【范例】

无。

HI_UNF_VO_GetWindowFieldMode

【接口定义】

```
HI_S32 HI_UNF_VO_GetWindowFieldMode(HI_HANDLE hWindow, HI_BOOL *pbEnable);
```

【新增原因】

为支持获取窗口场播放模式

【注意事项】

无。

【范例】

无。

HI_UNF_VO_GetStereoDetpth

【接口定义】

```
HI_S32 HI_UNF_VO_GetStereoDetpth(HI_HANDLE hWindow, HI_S32 *ps32Depth);
```

【新增原因】

为支持获取窗口 3D 输出景深

【注意事项】

无。

【范例】

无。

2.7 PMOC

2.7.1 概述

相对于 UNF 3.1 版本，UNF 3.2.0 在总体功能上有以下变更：

新增获取待机属性的接口

2.7.1.1 获取待机属性的接口

为了在设置某种待机属性的时候不覆盖其他的待机属性，添加了获取当前待机属性的接口。

接口

新增 HI_UNF_PMOC_GetWakeUpAttr

2.7.2 函数接口

2.7.2.1 新增

HI_UNF_PMOC_GetWakeUpAttr

【接口定义】

```
HI_S32 HI_UNF_PMOC_GetWakeUpAttr(HI_UNF_PMOC_WKUP_S_PTR pstAttr);
```

【新增原因】

支持获取当前已经设置到驱动里的待机属性。

【注意事项】

无。

【范例】

无。

2.8 PQ

2.8.1 概述

UNF 3.2.0 相对于 UNF 3.1 版本，在总体功能上有以下变更：

- 新增设置入网指标测试的默认参数功能
- 新增保存 PQ 参数功能

2.8.1.1 设置入网指标测试的默认参数功能

由于入网指标测试需要，对 PQ 相关的参数设置一组默认值，引起的变更如下：

接口

新增 HI_UNF_PQ_SetDefaultParam

2.8.1.2 保存 PQ 参数功能

因为对 PQ 参数需要进行掉电保存，引起的变更如下：

接口

新增 HI_UNF_PQ_UpdatePqParam

2.8.2 函数接口

2.8.2.1 新增

HI_UNF_PQ_SetDefaultParam

【接口定义】

```
HI_S32 HI_UNF_PQ_SetDefaultParam(HI_VOID);
```

【新增原因】

新特性，用于入网指标测试时配置 PQ 相关的默认参数。

【注意事项】

- 烧写了 PQ 参数分区才可以正常使用该接口
- 量产发布的单板中不建议调用该接口

【范例】

无。

HI_UNF_PQ_UpdatePqParam

【接口定义】

```
HI_S32 HI_UNF_PQ_UpdatePqParam(HI_VOID);
```

【新增原因】

新特性，更新 PQ 配置区信息，把 PQ 参数保存到 Flash。

【注意事项】

无。

【范例】

无。

2.9 PVR

2.9.1 概述

相对于 UNF 3.1 版本，UNF 3.2.0 在总体功能上有以下变更：

新增当前播放帧在文件中的偏移地址

2.9.1.1 当前播放帧在文件中的偏移地址

高安需求，在 HI_UNF_PVR_DATA_ATTR_S 结构体中增加当前播放帧在文件中偏移地址的成员。引起的变更如下：

数据结构

修改 HI_UNF_PVR_DATA_ATTR_S

2.9.2 数据结构

2.9.2.1 修改

HI_UNF_PVR_DATA_ATTR_S

【接口定义】

修改前：

```
typedef struct
{
    HI_U32      u32ChnID;
    HI_CHAR     CurFileName[PVR_MAX_FILENAME_LEN];
    HI_CHAR     IdxFileName[PVR_MAX_FILENAME_LEN+5];
    HI_U64      u64FileStartPos;
    HI_U64      u64FileEndPos;
    HI_U64      u64GlobalOffset;
} HI_UNF_PVR_DATA_ATTR_S;
```

修改后：

```
typedef struct
{
    HI_U32      u32ChnID;
    HI_CHAR     CurFileName[PVR_MAX_FILENAME_LEN];
    HI_CHAR     IdxFileName[PVR_MAX_FILENAME_LEN+5];
    HI_U64      u64FileStartPos;
    HI_U64      u64FileEndPos;
    HI_U64      u64GlobalOffset;
    HI_U64      u64FileReadOffset;
} HI_UNF_PVR_DATA_ATTR_S;
```

【修改说明】

在数据文件属性结构体中，增加当前播放帧在文件中的偏移地址。

【注意事项】

无。

【范例】

无。

2.10 SOUND

2.10.1 概述

UNF3.2.0 相对于 UNF3.1，在总体功能上有以下变更：

- 新增 SPDIF SCMS 模式设置与获取接口
- 新增 AllTrack 静音设置与获取接口
- 新增 Track 声道模式设置与获取接口
- 新增 Track DelayMs 获取接口
- 新增 Cast 音量设置与获取接口
- 新增 Cast 静音设置与获取接口

2.10.1.1 新增 SPDIF SCMS 模式设置与获取接口

数据结构

- 新增 HI_UNF_SND_SPDIF_SCMSMODE_E

函数接口

- 新增 HI_UNF_SND_SetSpdifSCMSMode
- 新增 HI_UNF_SND_GetSpdifSCMSMode

2.10.1.2 新增 AllTrack 静音设置与获取接口

数据结构

无

函数接口

- 新增 HI_UNF_SND_SetAllTrackMute
- 新增 HI_UNF_SND_GetAllTrackMute

2.10.1.3 新增 Track 声道模式设置与获取接口

数据结构

无

函数接口

- 新增 HI_UNF_SND_SetTrackChannelMode
- 新增 HI_UNF_SND_GetTrackChannelMode

2.10.1.4 新增 Track DelayMs 获取接口

数据结构

无。

函数接口

新增 HI_UNF_SND_GetTrackDelayMs

2.10.1.5 新增 Cast 音量设置与获取接口

数据结构

无。

函数接口

- 新增 HI_UNF_SND_SetCastAbsWeight
- 新增 HI_UNF_SND_GetCastAbsWeight

2.10.1.6 新增 Cast 静音设置与获取接口

数据结构

无。

函数接口

- 新增 HI_UNF_SND_SetCastMute
- 新增 HI_UNF_SND_GetCastMute

2.10.2 数据结构

2.10.2.1 新增

HI_UNF_SND_SPDIF_SCMSMODE_E

【枚举定义】

```
typedef enum hiHI_UNF_SND_SPDIF_SCMSMODE_E
{
    HI_UNF_SND_SPDIF_SCMSMODE_COPYALLOW,
    HI_UNF_SND_SPDIF_SCMSMODE_COPYONCE,
    HI_UNF_SND_SPDIF_SCMSMODE_COPYNOMORE,
    HI_UNF_SND_SPDIF_SCMSMODE_COPYPROHIBITED,
    HI_UNF_SND_SPDIF_SCMSMODE_BUTT
} HI_UNF_SND_OUTPUTPORT_E;
```

【新增原因】

指定 SPDIF SCMS 拷贝保护信息模式。

【注意事项】

无。

【范例】

无。

2.10.3 函数接口

2.10.3.1 新增

HI_UNF_SND_SetSpdifSCMSMode

【接口定义】

```
HI_S32 HI_UNF_SND_SetSpdifSCMSMode(HI_UNF_SND_E enSound,
HI_UNF_SND_OUTPUTPORT_E enOutPort, HI_UNF_SND_SPDIF_SCMSMODE_E
enSpdifSCMSMode);
```

【新增原因】

设置 SPDIF SCMS 拷贝保护信息模式。

【注意事项】

无。

【范例】

无。

HI_UNF_SND_GetSpdifSCMSMode

【接口定义】

```
HI_S32 HI_UNF_SND_GetSpdifSCMSMode(HI_UNF_SND_E enSound,
HI_UNF_SND_OUTPUTPORT_E enOutPort, HI_UNF_SND_SPDIF_SCMSMODE_E
*enSpdifSCMSMode);
```

【新增原因】

获取 SPDIF SCMS 拷贝保护信息模式状态。

【注意事项】

无。

【范例】

无。

HI_UNF_SND_SetAllTrackMute

【接口定义】

```
HI_S32  HI_UNF_SND_SetAllTrackMute(HI_UNF_SND_E enSound, HI_BOOL bMute);
```

【新增原因】

设置所有用户创建的 Track 静音功能。

【注意事项】

该接口对 Virtual Track 无效。

【范例】

无。

HI_UNF_SND_GetAllTrackMute

【接口定义】

```
HI_S32  HI_UNF_SND_GetAllTrackMute(HI_UNF_SND_E enSound, HI_BOOL
*pbMute);
```

【新增原因】

获取所有用户创建的 Track 静音功能状态。

【注意事项】

该接口对 Virtual Track 无效。

【范例】

无。

HI_UNF_SND_SetTrackChannelMode

【接口定义】

```
HI_S32 HI_UNF_SND_SetTrackChannelMode(HI_HANDLE hTrack,
HI_UNF_TRACK_MODE_E enMode);
```

【新增原因】

设置单路 Track 声道模式。

【注意事项】

该接口对 Virtual Track 无效。

【范例】

无。

HI_UNF_SND_GetTrackChannelMode

【接口定义】

```
HI_S32 HI_UNF_SND_GetTrackChannelMode(HI_HANDLE hTrack,
HI_UNF_TRACK_MODE_E *penMode);
```

【新增原因】

获取单路 Track 声道模式状态。

【注意事项】

该接口对 Virtual Track 无效。

【范例】

无。

HI_UNF_SND_GetTrackDelayMs

【接口定义】

```
HI_S32 HI_UNF_SND_GetTrackDelayMs(const HI_HANDLE hTrack, HI_U32
*pDelayMs);
```

【新增原因】

获取 Track delay 时间。

【注意事项】

该接口对 Virtual Track 无效。

【范例】

无。

HI_UNF_SND_SetCastAbsWeight

【接口定义】

```
HI_S32 HI_UNF_SND_SetCastAbsWeight(HI_HANDLE hCast, const
HI_UNF_SND_ABSGAIN_ATTR_S *pstAbsWeightGain);
```

【新增原因】

设置 Cast 获取到的数据音量。

【注意事项】

无。

【范例】

无

HI_UNF_SND_GetCastAbsWeight

【接口定义】

```
HI_S32  HI_UNF_SND_GetCastAbsWeight(HI_HANDLE hCast,
HI_UNF_SND_ABSGAIN_ATTR_S *pstAbsWeightGain);
```

【新增原因】

获取 Cast 获取到的数据音量大小。

【注意事项】

无。

【范例】

无。

HI_UNF_SND_SetCastMute

【接口定义】

```
HI_S32   HI_UNF_SND_SetCastMute(HI_HANDLE hCast, HI_BOOL bMute);
```

【新增原因】

设置 Cast 获取到的数据静音功能。

【注意事项】

无。

【范例】

无。

HI_UNF_SND_GetCastMute

【接口定义】

```
HI_S32   HI_UNF_SND_GetCastMute(HI_HANDLE hCast, HI_BOOL *pbMute);
```

【新增原因】

获取 Cast 获取到的数据静音状态。

【注意事项】

无。

【范例】

无。

2.11 VI

2.11.1 概述

UNF 3.2.0 相对于 UNF 3.1 版本，在总体功能上有以下变更：

支持 BT1120 数字接口 640x480P 分辨率输入

2.11.1.1 BT1120 数字接口 640x480P 分辨率输入

BT1120 输入时序 640x480P 分辨率 VI 可以正确接收。引起的变更如下：

数据结构

修改 HI_UNF_VI_INPUT_MODE_E

2.11.2 数据结构

2.11.2.1 修改

HI_UNF_VI_INPUT_MODE_E

【结构体定义】

```
typedef enum hiUNF_VI_INPUT_MODE_E
{
    HI_UNF_VI_MODE_BT656_576I = 0,
    HI_UNF_VI_MODE_BT656_480I,
    HI_UNF_VI_MODE_BT601_576I,
    HI_UNF_VI_MODE_BT601_480I,
    HI_UNF_VI_MODE_BT1120_480P,
    HI_UNF_VI_MODE_BT1120_576P,
    HI_UNF_VI_MODE_BT1120_720P_50,
    HI_UNF_VI_MODE_BT1120_720P_60,
    HI_UNF_VI_MODE_BT1120_1080I_50,
    HI_UNF_VI_MODE_BT1120_1080I_60,
    HI_UNF_VI_MODE_BT1120_1080P_25,
    HI_UNF_VI_MODE_BT1120_1080P_30,
    HI_UNF_VI_MODE_BT1120_1080P_50,
    HI_UNF_VI_MODE_BT1120_1080P_60,
    HI_UNF_VI_MODE_DIGITAL_CAMERA_48,
    HI_UNF_VI_MODE_DIGITAL_CAMERA_57,
```

```
    HI_UNF_VI_MODE_DIGITAL_CAMERA_720P_30,
    HI_UNF_VI_MODE_DIGITAL_CAMERA_1080P_30,
    HI_UNF_VI_MODE_BUTT
} HI_UNF_VI_INPUT_MODE_E;
```

修改后：

```
typedef enum hiUNF_VI_INPUT_MODE_E
{
    HI_UNF_VI_MODE_BT656_576I = 0,
    HI_UNF_VI_MODE_BT656_480I,
    HI_UNF_VI_MODE_BT601_576I,
    HI_UNF_VI_MODE_BT601_480I,
    HI_UNF_VI_MODE_BT1120_640X480P,
    HI_UNF_VI_MODE_BT1120_480P,
    HI_UNF_VI_MODE_BT1120_576P,
    HI_UNF_VI_MODE_BT1120_720P_50,
    HI_UNF_VI_MODE_BT1120_720P_60,
    HI_UNF_VI_MODE_BT1120_1080I_50,
    HI_UNF_VI_MODE_BT1120_1080I_60,
    HI_UNF_VI_MODE_BT1120_1080P_25,
    HI_UNF_VI_MODE_BT1120_1080P_30,
    HI_UNF_VI_MODE_BT1120_1080P_50,
    HI_UNF_VI_MODE_BT1120_1080P_60,
    HI_UNF_VI_MODE_DIGITAL_CAMERA_48,
    HI_UNF_VI_MODE_DIGITAL_CAMERA_57,
    HI_UNF_VI_MODE_DIGITAL_CAMERA_720P_30,
    HI_UNF_VI_MODE_DIGITAL_CAMERA_1080P_30,
    HI_UNF_VI_MODE_BUTT
} HI_UNF_VI_INPUT_MODE_E;
```

【修改说明】

增加 HI_UNF_VI_MODE_BT1120_640X480P，支持 BT1120 数字接口 640x480P 分辨率输入。

【注意事项】

无。

【范例】

无。

2.12 Closed Caption

2.12.1 概述

UNF 3.2.0 相对于 UNF 3.1 版本，在总体功能上有以下变更：

- 新增解析 ATSC 708CC 标准定义的 64 种颜色功能
- 新增获取 ARIB CC 字幕信息功能
- 新增解析字体边缘效果和字体边缘色
- 新增支持 SCTE20 标准

2.12.1.1 解析 ATSC 708CC 标准定义的 64 种颜色功能

UNF3.1 版本只支持解析 8 种颜色，UNF 3.2.0 版本扩展为支持解析 64 种颜色。引起的变更如下：

数据结构

- 修改 2.12.2.2 HI_UNF_CC_608_CONFIGPARAM_S
- 修改 HI_UNF_CC_708_CONFIGPARAM_S
- 修改 HI_UNF_CC_COLOR_E

2.12.1.2 获取 ARIB CC 字幕信息功能

用于获取 ARIB CC 字幕字符编码、字幕滚动模式、时间控制模式、播放模式、显示方式等字幕信息。引起的变更如下：

数据结构

- 新增 HI_UNF_CC_ARIB_ROLLUP_E
- 新增 HI_UNF_CC_ARIB_TCS_E
- 新增 HI_UNF_CC_ARIB_DF_E
- 新增 HI_UNF_CC_ARIB_DMF_E
- 新增 HI_UNF_CC_ARIB_TMD_E
- 新增 HI_UNF_CC_ARIB_INFONODE_S
- 新增 HI_UNF_CC_ARIB_INFO_S

接口

新增 HI_UNF_CC_GetARIBCCInfo

2.12.1.3 解析字体边缘效果和字体边缘色

将从码流中解析出来的字体边缘效果和字体边缘色信息通过回调返回给用户。引起的变更如下：

数据结构

修改 HI_UNF_CC_TEXT_S

2.12.1.4 支持 SCTE20 标准

为了区分 scte20 标准的字幕，增加了顶底场标记。引起的变更如下：

数据结构

修改 HI_UNF_CC_USERDATA_S

2.12.2 数据结构

2.12.2.1 新增

HI_UNF_CC_ARIB_ROLLUP_E

【结构体定义】

```
typedef enum hiUNF_CC_ARIB_ROLLUP_E
{
    HI_UNF_CC_ARIB_NON_ROLLUP,
    HI_UNF_CC_ARIB_ROLLUP,
    HI_UNF_CC_ARIB_ROLLUP_BUTT
}HI_UNF_CC_ARIB_ROLLUP_E;
```

【新增原因】

为了在获取 arib cc 字幕信息时，返回有意义的结果定义了 ARIB CC 的 roll-up 类型。

【注意事项】

无。

【范例】

无。

HI_UNF_CC_ARIB_TCS_E

【结构体定义】

```
typedef enum hiUNF_CC_ARIB_TCS_E
{
    HI_UNF_CC_ARIB_TCS_8BIT,
    HI_UNF_CC_ARIB_TCS_BUTT
```

```
}HI_UNF_CC_ARIB_TCS_E;
```

【新增原因】

为了在获取 arib cc 字幕信息时，返回有意义的结果定义了 ARIB CC 的字符编码方式。

【注意事项】

无。

【范例】

无。

HI_UNF_CC_ARIB_DF_E

【结构体定义】

```
typedef enum hiUNF_CC_ARIB_DF_E
{
    HI_UNF_CC_ARIB_DF_HORIZONTAL_SD,
    HI_UNF_CC_ARIB_DF_VERTICAL_SD,
    HI_UNF_CC_ARIB_DF_HORIZONTAL_HD,
    HI_UNF_CC_ARIB_DF_VERTICAL_HD,
    HI_UNF_CC_ARIB_DF_HORIZONTAL_WESTERN,
    HI_UNF_CC_ARIB_DF_HORIZONTAL_1920X1080,
    HI_UNF_CC_ARIB_DF_VERTICAL_1920X1080,
    HI_UNF_CC_ARIB_DF_HORIZONTAL_960X540,
    HI_UNF_CC_ARIB_DF_VERTICAL_960X540,
    HI_UNF_CC_ARIB_DF_HORIZONTAL_1280X720,
    HI_UNF_CC_ARIB_DF_VERTICAL_1280X720,
    HI_UNF_CC_ARIB_DF_HORIZONTAL_720X480,
    HI_UNF_CC_ARIB_DF_VERTICAL_720X480,
    HI_UNF_CC_ARIB_DF_BUTT
}HI_UNF_CC_ARIB_DF_E;
```

【新增原因】

为了在获取 arib cc 字幕信息时返回有意义的结果，所以定义了 ARIB CC 的显示方式。

【注意事项】

无。

【范例】

无。

HI_UNF_CC_ARIB_DMF_E

【结构体定义】

```
typedef enum hiUNF_CC_ARIB_DMF_E
```

```
{
    HI_UNF_CC_ARIB_DMF_AUTO_AND_AUTO=0x0,
    HI_UNF_CC_ARIB_DMF_AUTO_AND_NOT,
    HI_UNF_CC_ARIB_DMF_AUTO_AND_SELECT,
    HI_UNF_CC_ARIB_DMF_NON_AND_AUTO=0x4,
    HI_UNF_CC_ARIB_DMF_NON_AND_NON,
    HI_UNF_CC_ARIB_DMF_SELECT_AND_AUTO=0x8,
    HI_UNF_CC_ARIB_DMF_SELECT_AND_NON,
    HI_UNF_CC_ARIB_DMF_SELECT_AND_SELECT,
    HI_UNF_CC_ARIB_DMF_SPECIAL_AND_AUTO=0xc,
    HI_UNF_CC_ARIB_DMF_SPECIAL_AND_NON,
    HI_UNF_CC_ARIB_DMF_SPECIAL_AND_SELECT,
    HI_UNF_CC_ARIB_DMF_BUTT
}HI_UNF_CC_ARIB_DMF_E;
```

【新增原因】

为了在获取 arib cc 字幕信息时返回有意义的结果，所以定义了 ARIB CC 的显示模式。

【注意事项】

无。

【范例】

无。

HI_UNF_CC_ARIB_TMD_E

【结构体定义】

```
typedef enum hiUNF_CC_ARIB_TMD_E
{
    HI_UNF_CC_ARIB_TMD_FREE,
    HI_UNF_CC_ARIB_TMD_REAL_TIME,
    HI_UNF_CC_ARIB_TMD_OFFSET_TIME,
    HI_UNF_CC_ARIB_TMD_BUTT
}HI_UNF_CC_ARIB_TMD_E;
```

【新增原因】

为了在获取 arib cc 字幕信息时，返回有意义的结果定义了 ARIB CC 的时间控制模式。

【注意事项】

无。

【范例】

无。

HI_UNF_CC_ARIB_INFONODE_S

【结构体定义】

```
typedef struct hiUNF_CC_ARIB_INFONODE_S
{
    HI_U8 u8LanguageTag;
    HI_UNF_CC_ARIB_DMF_E    enCCAribDMF;
    HI_CHAR    acISO639LanguageCode[4];
    HI_UNF_CC_ARIB_DF_E enCCAribDF;
    HI_UNF_CC_ARIB_TCS_E enCCAribTCS;
    HI_UNF_CC_ARIB_ROLLUP_E enCCAribRollup;
}HI_UNF_CC_ARIB_INFONODE_S;
```

【新增原因】

为了实现获取 arib cc 字幕信息的功能，定义了 ARIB CC 单种语言的字幕信息结构体。

【注意事项】

无。

【范例】

无。

HI_UNF_CC_ARIB_INFO_S

【结构体定义】

```
typedef struct hiUNF_CC_ARIB_INFO_S
{
    HI_UNF_CC_ARIB_TMD_E enCCAribTMD;
    HI_U32 u32NumLanguage;
    HI_UNF_CC_ARIB_INFONODE_S stCCAribInfonode[ARIBCC_MAX_LANGUAGE];
}HI_UNF_CC_ARIB_INFO_S;
```

【新增原因】

为了实现获取 arib cc 字幕信息的功能，定义了 ARIB CC 信息结构体。

【注意事项】

无。

【范例】

无。

2.12.2.2 修改

HI_UNF_CC_608_CONFIGPARAM_S

【结构体定义】

修改前：

```
typedef struct hiUNF_CC_608_CONFIGPARAM_S
{
    HI_UNF_CC_608_DATATYPE_E    enCC608DataType;
    HI_UNF_CC_COLOR_E           enCC608TextColor;
    HI_UNF_CC_OPACITY_E         enCC608TextOpac;
    HI_UNF_CC_COLOR_E           enCC608BgColor;
    HI_UNF_CC_OPACITY_E         enCC608BgOpac;
    HI_UNF_CC_FONTSTYLE_E       enCC608FontStyle;
    HI_UNF_CC_DF_E              enCC608DispFormat;
    HI_BOOL                     bLeadingTailingSpace;
}HI_UNF_CC_608_CONFIGPARAM_S;
```

修改后：

```
typedef struct hiUNF_CC_608_CONFIGPARAM_S
{
    HI_UNF_CC_608_DATATYPE_E    enCC608DataType;
    HI_U32          u32CC608TextColor;
    HI_UNF_CC_OPACITY_E         enCC608TextOpac;
    HI_U32          u32CC608BgColor;
    HI_UNF_CC_OPACITY_E         enCC608BgOpac;
    HI_UNF_CC_FONTSTYLE_E       enCC608FontStyle;
    HI_UNF_CC_DF_E              enCC608DispFormat;
    HI_BOOL                     bLeadingTailingSpace;

}HI_UNF_CC_608_CONFIGPARAM_S;
```

【修改说明】

为了支持 64 色甚至更多颜色，将颜色值从枚举改为了无符号整型。

【注意事项】

无。

【范例】

无。

HI_UNF_CC_708_CONFIGPARAM_S

【结构体定义】

修改前：

```
typedef struct hiUNF_CC_708_CONFIGPARAM_S
{
    HI_UNF_CC_708_SERVICE_NUM_E    enCC708ServiceNum;
    HI_UNF_CC_FONTNAME_E       enCC708FontName;
    HI_UNF_CC_FONTSTYLE_E      enCC708FontStyle;
    HI_UNF_CC_FONTSIZE_E       enCC708FontSize;
    HI_UNF_CC_COLOR_E          enCC708TextColor;
    HI_UNF_CC_OPACITY_E        enCC708TextOpac;
    HI_UNF_CC_COLOR_E          enCC708BgColor;
    HI_UNF_CC_OPACITY_E        enCC708BgOpac;
    HI_UNF_CC_COLOR_E          enCC708WinColor;
    HI_UNF_CC_OPACITY_E        enCC708WinOpac;
    HI_UNF_CC_EdgeType_E       enCC708TextEdgeType;
    HI_UNF_CC_COLOR_E          enCC708TextEdgeColor;
    HI_UNF_CC_DF_E             enCC708DispFormat;
} HI_UNF_CC_708_CONFIGPARAM_S;
```

修改后：

```
typedef struct hiUNF_CC_708_CONFIGPARAM_S
{
    HI_UNF_CC_708_SERVICE_NUM_E    enCC708ServiceNum;
    HI_UNF_CC_FONTNAME_E       enCC708FontName;
    HI_UNF_CC_FONTSTYLE_E      enCC708FontStyle;
    HI_UNF_CC_FONTSIZE_E       enCC708FontSize;
    HI_U32          u32CC708TextColor;
    HI_UNF_CC_OPACITY_E        enCC708TextOpac;
    HI_U32          u32CC708BgColor;
    HI_UNF_CC_OPACITY_E        enCC708BgOpac;
    HI_U32          u32CC708WinColor;
    HI_UNF_CC_OPACITY_E        enCC708WinOpac;
    HI_UNF_CC_EdgeType_E       enCC708TextEdgeType;
    HI_U32          u32CC708TextEdgeColor;
    HI_UNF_CC_DF_E             enCC708DispFormat;
} HI_UNF_CC_708_CONFIGPARAM_S;
```

【修改说明】

为了支持 64 色甚至更多颜色，将颜色值从枚举改为了无符号整型。

【注意事项】

无。

【范例】

无。

HI_UNF_CC_COLOR_E

【结构体定义】

修改前：

```
typedef enum hiUNF_CC_COLOR_E
{
    HI_UNF_CC_COLOR_DEFAULT,
    HI_UNF_CC_COLOR_BLACK,
    HI_UNF_CC_COLOR_WHITE,
    HI_UNF_CC_COLOR_RED,
    HI_UNF_CC_COLOR_GREEN,
    HI_UNF_CC_COLOR_BLUE,
    HI_UNF_CC_COLOR_YELLOW,
    HI_UNF_CC_COLOR_MAGENTA,
    HI_UNF_CC_COLOR_CYAN,
    HI_UNF_CC_COLOR_BUTT
}HI_UNF_CC_COLOR_E;
```

修改后：

```
typedef enum hiUNF_CC_COLOR_E
{
    HI_UNF_CC_COLOR_DEFAULT=0x00000000,
    HI_UNF_CC_COLOR_BLACK=0xff000000,
    HI_UNF_CC_COLOR_WHITE=0xffffffff,
    HI_UNF_CC_COLOR_RED=0xffff0000,
    HI_UNF_CC_COLOR_GREEN=0xff00ff00,
    HI_UNF_CC_COLOR_BLUE=0xff0000ff,
    HI_UNF_CC_COLOR_YELLOW=0xffffff00,
    HI_UNF_CC_COLOR_MAGENTA=0xffff00ff,
    HI_UNF_CC_COLOR_CYAN=0xff00ffff,
}HI_UNF_CC_COLOR_E;
```

【修改说明】

将对应的 ARGB 值赋给基本的 8 色枚举，方便用户使用。

【注意事项】

无。

【范例】

无。

HI_UNF_CC_TEXT_S

【结构体定义】

修改前：

```
typedef struct hiUNF_CC_TEXT_S
{
    HI_U16              *pu16Text;
    HI_U8               u8TextLen;
    HI_UNF_CC_COLOR_S     stFgColor;
    HI_UNF_CC_COLOR_S     stBgColor;
    HI_U8               u8Justify;
    HI_U8               u8WordWrap;
    HI_UNF_CC_FONTSTYLE_E  enFontStyle;
    HI_UNF_CC_FONTSIZE_E   enFontSize;
} HI_UNF_CC_TEXT_S;
```

修改后：

```
typedef struct hiUNF_CC_TEXT_S
{
    HI_U16              *pu16Text;
    HI_U8               u8TextLen;
    HI_UNF_CC_COLOR_S     stFgColor;
    HI_UNF_CC_COLOR_S     stBgColor;
    HI_UNF_CC_COLOR_S     stEdgeColor;
    HI_U8               u8Justify;
    HI_U8               u8WordWrap;
    HI_UNF_CC_FONTSTYLE_E  enFontStyle;
    HI_UNF_CC_FONTSIZE_E   enFontSize;
    HI_UNF_CC_EdgeType_E    enEdgetype;
} HI_UNF_CC_TEXT_S;
```

【修改说明】

增加字体边缘类型和字体边缘色的解析。

【注意事项】

无。

【范例】

无。

HI_UNF_CC_USERDATA_S

【结构体定义】

修改前：

```
typedef struct hiUNF_CC_USERDATA_S
{
    HI_U8      *pu8userdata;
```

```
    HI_U32      u32dataLen;
} HI_UNF_CC_USERDATA_S;
```

修改后：

```
typedef struct hiUNF_CC_USERDATA_S
{
    HI_U8       *pu8userdata;
    HI_U32      u32dataLen;
    HI_BOOL     bTopFieldFirst;
} HI_UNF_CC_USERDATA_S;
```

【修改说明】

为支持 SCTE20 标准，增加顶底场标记。

【注意事项】

无。

【范例】

无。

2.12.3 函数接口

2.12.3.1 新增

HI_UNF_CC_GetARIBCCInfo

【接口定义】

```
HI_S32 HI_UNF_CC_GetARIBCCInfo(HI_HANDLE hCC,HI_UNF_CC_ARIB_INFO_S
*pstCCAribInfo);
```

【新增原因】

获取 arib cc 字幕信息。

【注意事项】

模块初始化，创建与启动实例，注入 arib cc 数据之后，调用该接口返回的结果才有意义。

【范例】

无。

2.13 Teletext

2.13.1 概述

UNF 3.2.0 相对于 UNF 3.1 版本，在总体功能上有以下变更：

设置 ISO639LanguageCode 功能

2.13.1.1 设置 ISO639LanguageCode 功能

某些地区的码流，本身没有按照 Teletext 规范发送字符集信息，造成解码器无法正常解码，因此在调用 HI_UNF_TTX_SwitchContent 时将 ISO639LanguageCode 一起设置给解码器，解码器可以根据此信息辅助判断当前码流的字符集。引起的变更如下：

数据结构

修改 HI_UNF_TTX_CONTENT_PARA_S

2.13.2 数据结构

2.13.2.1 修改

HI_UNF_TTX_CONTENT_PARA_S

【结构体定义】

修改前：

```
typedef struct hiUNF_TTX_CONTENT_PARA_S
{
    HI_UNF_TTX_TYPE_E      enType;
    HI_UNF_TTX_PAGE_ADDR_S stInitPgAddr;
} HI_UNF_TTX_CONTENT_PARA_S;
```

修改后：

```
typedef struct hiUNF_TTX_CONTENT_PARA_S
{
    HI_UNF_TTX_TYPE_E      enType;
    HI_U32                 u32ISO639LanCode;
    HI_UNF_TTX_PAGE_ADDR_S stInitPgAddr;
} HI_UNF_TTX_CONTENT_PARA_S;
```

【修改说明】

将 ISO639LanguageCode 设置给解码器，解码器可以根据此信息辅助判断当前码流的字符集，ISO639LanguageCode 可以从 PMT 表里解析 Teletext 描述符时一并解出。

【注意事项】

无。

【范例】

无。

2.14 HiPlayer

2.14.1 概述

UNF 3.2.0 相对于 UNF 3.1 版本，在总体功能上有以下变更：

- 支持 1 倍速倒退和 0.5 倍速慢放功能
- 支持设置 HTTP 协议 Headers
- 支持指定 HLS 直播启动时播放第一个 Segment

2.14.1.1 支持 1 倍速倒退和 0.5 倍速慢放功能

部分 APP 有低倍速慢放和倒退需求。引起的变更如下：

数据结构

修改 HI_SVR_PLAYER_PLAY_SPEED_E

2.14.1.2 支持设置 HTTP 协议 Headers

某些服务器对 HTTP 报文的 Headers 有特殊要求，此时 APP 可通过新增的 Invoke 接口进行设置，HiPlayer 会将这些 HTTP Headers 包含在向服务器发送的 HTTP 报文中。引起的变更如下：

数据结构

- 修改 HI_FORMAT_INVOKE_ID_E
- 修改 HI_SVR_FORMAT_PARAMETER_S

2.14.1.3 支持指定 HLS 直播启动时播放第一个 Segment

通过该接口指定播放的第一个 Segment 可调整直播的实时性。引起的变更如下：

数据结构

- 修改 HI_FORMAT_INVOKE_ID_E
- 修改 HI_SVR_FORMAT_PARAMETER_S

2.14.2 数据结构

2.14.2.1 修改

HI_SVR_FORMAT_PARAMETER_S

【结构体定义】

修改前：

```
typedef struct hiSVR_FORMAT_PARAMETER_S
{
    HI_FORMAT_BUFFER_CONFIG_S stBufConfig;
```

```
    HI_S64          s64BufMaxSize;
    HI_U32          u32Cookie;
    HI_FORMAT_HLS_START_MODE_E eHlsStartMode;
    HI_HANDLE       hDRMClient;
    HI_S32          s32DstCodeType;
    HI_VOID         *pArgCharsetMgr;
    HI_U32          u32Userdata;
    HI_U8           *pUserAgent;
    HI_U32          u32LogLevel;
    HI_U8           *pReferer;
    HI_U32          u32NotSupportByteRange;
} HI_SVR_FORMAT_PARAMETER_S;;
```

修改后：

```
typedef struct hiSVR_FORMAT_PARAMETER_S
{
    HI_FORMAT_BUFFER_CONFIG_S stBufConfig;
    HI_S64          s64BufMaxSize;
    HI_CHAR         *pCookie;
    HI_CHAR         *pHeaders;
    HI_FORMAT_HLS_START_MODE_E eHlsStartMode;
    HI_HANDLE       hDRMClient;
    HI_S32          s32DstCodeType;
    HI_VOID         *pArgCharsetMgr;
    HI_U32          u32Userdata;
    HI_U8           *pUserAgent;
    HI_U32          u32LogLevel;
    HI_U8           *pReferer;
    HI_U32          u32NotSupportByteRange;
    HI_S32          s32HlsLiveStartNum;
} HI_SVR_FORMAT_PARAMETER_S;
```

【修改说明】

新增成员 pHeaders，用于存放 APP 设置给 HiPlayer 的 HTTP Headers。

新增成员 s32HlsLiveStartNum，用于设置 HLS 直播启动时播放的第一个 Segment。

修改成员 u32Cookie 的类型为字符串指针类型。

【注意事项】

无。

【范例】

无。

HI_SVR_PLAYER_PLAY_SPEED_E

【结构体定义】

修改前：

```
typedef enum hiSVR_PLAYER_PLAY_SPEED_E
{
    HI_SVR_PLAYER_PLAY_SPEED_2X_FAST_FORWARD   = 2 *
HI_SVR_PLAYER_PLAY_SPEED_NORMAL,
    HI_SVR_PLAYER_PLAY_SPEED_4X_FAST_FORWARD   = 4 *
HI_SVR_PLAYER_PLAY_SPEED_NORMAL,
    HI_SVR_PLAYER_PLAY_SPEED_8X_FAST_FORWARD   = 8 *
HI_SVR_PLAYER_PLAY_SPEED_NORMAL,
    HI_SVR_PLAYER_PLAY_SPEED_16X_FAST_FORWARD  = 16 *
HI_SVR_PLAYER_PLAY_SPEED_NORMAL,
    HI_SVR_PLAYER_PLAY_SPEED_32X_FAST_FORWARD  = 32 *
HI_SVR_PLAYER_PLAY_SPEED_NORMAL,
    HI_SVR_PLAYER_PLAY_SPEED_64X_FAST_FORWARD  = 64 *
HI_SVR_PLAYER_PLAY_SPEED_NORMAL,
    HI_SVR_PLAYER_PLAY_SPEED_2X_FAST_BACKWARD  = -2 *
HI_SVR_PLAYER_PLAY_SPEED_NORMAL,
    HI_SVR_PLAYER_PLAY_SPEED_4X_FAST_BACKWARD  = -4 *
HI_SVR_PLAYER_PLAY_SPEED_NORMAL,
    HI_SVR_PLAYER_PLAY_SPEED_8X_FAST_BACKWARD  = -8 *
HI_SVR_PLAYER_PLAY_SPEED_NORMAL,
    HI_SVR_PLAYER_PLAY_SPEED_16X_FAST_BACKWARD = -16 *
HI_SVR_PLAYER_PLAY_SPEED_NORMAL,
    HI_SVR_PLAYER_PLAY_SPEED_32X_FAST_BACKWARD = -32 *
HI_SVR_PLAYER_PLAY_SPEED_NORMAL,
    HI_SVR_PLAYER_PLAY_SPEED_64X_FAST_BACKWARD = -64 *
HI_SVR_PLAYER_PLAY_SPEED_NORMAL,
    HI_SVR_PLAYER_PLAY_SPEED_BUTT
} HI_SVR_PLAYER_PLAY_SPEED_E;
```

修改后：

```
typedef enum hiSVR_PLAYER_PLAY_SPEED_E
{
    HI_SVR_PLAYER_PLAY_SPEED_1X2_SLOW_FORWARD  =
HI_SVR_PLAYER_PLAY_SPEED_NORMAL/2,
    HI_SVR_PLAYER_PLAY_SPEED_2X_FAST_FORWARD   = 2 *
HI_SVR_PLAYER_PLAY_SPEED_NORMAL,
    HI_SVR_PLAYER_PLAY_SPEED_4X_FAST_FORWARD   = 4 *
HI_SVR_PLAYER_PLAY_SPEED_NORMAL,
    HI_SVR_PLAYER_PLAY_SPEED_8X_FAST_FORWARD   = 8 *
```

```
HI_SVR_PLAYER_PLAY_SPEED_NORMAL,
    HI_SVR_PLAYER_PLAY_SPEED_16X_FAST_FORWARD  = 16 *
HI_SVR_PLAYER_PLAY_SPEED_NORMAL,
    HI_SVR_PLAYER_PLAY_SPEED_32X_FAST_FORWARD  = 32 *
HI_SVR_PLAYER_PLAY_SPEED_NORMAL,
    HI_SVR_PLAYER_PLAY_SPEED_64X_FAST_FORWARD  = 64 *
HI_SVR_PLAYER_PLAY_SPEED_NORMAL,
    HI_SVR_PLAYER_PLAY_SPEED_1X_BACKWARD       = -1 *
HI_SVR_PLAYER_PLAY_SPEED_NORMAL,
    HI_SVR_PLAYER_PLAY_SPEED_2X_FAST_BACKWARD  = -2 *
HI_SVR_PLAYER_PLAY_SPEED_NORMAL,
    HI_SVR_PLAYER_PLAY_SPEED_4X_FAST_BACKWARD  = -4 *
HI_SVR_PLAYER_PLAY_SPEED_NORMAL,
    HI_SVR_PLAYER_PLAY_SPEED_8X_FAST_BACKWARD  = -8 *
HI_SVR_PLAYER_PLAY_SPEED_NORMAL,
    HI_SVR_PLAYER_PLAY_SPEED_16X_FAST_BACKWARD = -16 *
HI_SVR_PLAYER_PLAY_SPEED_NORMAL,
    HI_SVR_PLAYER_PLAY_SPEED_32X_FAST_BACKWARD = -32 *
HI_SVR_PLAYER_PLAY_SPEED_NORMAL,
    HI_SVR_PLAYER_PLAY_SPEED_64X_FAST_BACKWARD = -64 *
HI_SVR_PLAYER_PLAY_SPEED_NORMAL,
    HI_SVR_PLAYER_PLAY_SPEED_BUTT
} HI_SVR_PLAYER_PLAY_SPEED_E;
```

【修改说明】

新增成员 HI_SVR_PLAYER_PLAY_SPEED_1X2_SLOW_FORWARD，表示 0.5 倍速慢放。

新增成员 HI_SVR_PLAYER_PLAY_SPEED_1X_BACKWARD，表示 1 倍速倒退。

【注意事项】

不支持 64 倍速的快进和快退。

【范例】

无。

HI_FORMAT_INVOKE_ID_E

【结构体定义】

修改前：

```
typedef enum hiFORMAT_INVOKE_ID_E
{
    HI_FORMAT_INVOKE_PRE_CLOSE_FILE = 0x0,
    HI_FORMAT_INVOKE_USER_OPE,
    HI_FORMAT_INVOKE_SET_SUB_LANTYPE,
```

```
    HI_FORMAT_INVOKE_SET_SOURCE_CODETYPE,
    HI_FORMAT_INVOKE_SET_DEST_CODETYPE,
    HI_FORMAT_INVOKE_SET_CHARSETMGR_FUNC,
    HI_FORMAT_INVOKE_GET_METADATA,
    HI_FORMAT_INVOKE_GET_THUMBNAIL,
    HI_FORMAT_INVOKE_CHECK_VIDEOSUPPORT,
    HI_FORMAT_INVOKE_SET_COOKIE,
    HI_FORMAT_INVOKE_GET_BANDWIDTH,
    HI_FORMAT_INVOKE_GET_HLS_STREAM_NUM,
    HI_FORMAT_INVOKE_GET_HLS_STREAM_INFO,
    HI_FORMAT_INVOKE_GET_HLS_SEGMENT_INFO,
    HI_FORMAT_INVOKE_GET_PLAYLIST_STREAM_DETAIL_INFO,
    HI_FORMAT_INVOKE_SET_HLS_STREAM,
    HI_FORMAT_INVOKE_SET_HLS_START_MODE,
    HI_FORMAT_INVOKE_SET_BUFFER_CONFIG,
    HI_FORMAT_INVOKE_GET_BUFFER_CONFIG,
    HI_FORMAT_INVOKE_GET_BUFFER_STATUS,
    HI_FORMAT_INVOKE_GET_BUFFER_LAST_PTS,
    HI_FORMAT_INVOKE_SET_BUFFER_MAX_SIZE,
    HI_FORMAT_INVOKE_GET_BUFFER_MAX_SIZE,
    HI_FORMAT_INVOKE_GET_MSG_EVENT,
    HI_FORMAT_INVOKE_GET_FIRST_IFRAME,
    HI_FORMAT_INVOKE_SET_DRM_HDL,
    HI_FORMAT_INVOKE_SET_LOCALTIME,
    HI_FORMAT_INVOKE_SET_USERAGENT,
    HI_FORMAT_INVOKE_SET_REFERER,
    HI_FORMAT_INVOKE_SET_NOT_SUPPORT_BYTERANGE,
    HI_FORMAT_INVOKE_SET_LOG_LEVEL,
    HI_FORMAT_INVOKE_RTMP_RECEIVEAUDIO,
    HI_FORMAT_INVOKE_RTMP_RECEIVEVIDEO,
    HI_FORMAT_INVOKE_SET_BUFFER_UNDERRUN,
    HI_FORMAT_INVOKE_PROTOCOL_USER=100,
    HI_FORMAT_INVOKE_SET_DOLBYRANGEINFO,
    HI_FORMAT_INVOKE_GET_DOLBYINFO,
    HI_FORMAT_INVOKE_BUTT
} HI_FORMAT_INVOKE_ID_E;
```

修改后：

```
typedef enum hiFORMAT_INVOKE_ID_E
{
    HI_FORMAT_INVOKE_PRE_CLOSE_FILE = 0x0,
    HI_FORMAT_INVOKE_USER_OPE,
    HI_FORMAT_INVOKE_SET_SUB_LANTYPE,
    HI_FORMAT_INVOKE_SET_SOURCE_CODETYPE,
```

```
    HI_FORMAT_INVOKE_SET_DEST_CODETYPE,
    HI_FORMAT_INVOKE_SET_CHARSETMGR_FUNC,
    HI_FORMAT_INVOKE_GET_METADATA,
    HI_FORMAT_INVOKE_GET_THUMBNAIL,
    HI_FORMAT_INVOKE_CHECK_VIDEOSUPPORT,
    HI_FORMAT_INVOKE_SET_COOKIE,
    HI_FORMAT_INVOKE_GET_BANDWIDTH,
    HI_FORMAT_INVOKE_GET_HLS_STREAM_NUM,
    HI_FORMAT_INVOKE_GET_HLS_STREAM_INFO,
    HI_FORMAT_INVOKE_GET_HLS_SEGMENT_INFO,
    HI_FORMAT_INVOKE_GET_PLAYLIST_STREAM_DETAIL_INFO,
    HI_FORMAT_INVOKE_SET_HLS_STREAM,
    HI_FORMAT_INVOKE_SET_HLS_START_MODE,
    HI_FORMAT_INVOKE_SET_BUFFER_CONFIG,
    HI_FORMAT_INVOKE_GET_BUFFER_CONFIG,
    HI_FORMAT_INVOKE_GET_BUFFER_STATUS,
    HI_FORMAT_INVOKE_GET_BUFFER_LAST_PTS,
    HI_FORMAT_INVOKE_SET_BUFFER_MAX_SIZE,
    HI_FORMAT_INVOKE_GET_BUFFER_MAX_SIZE,
    HI_FORMAT_INVOKE_GET_MSG_EVENT,
    HI_FORMAT_INVOKE_GET_FIRST_IFRAME,
    HI_FORMAT_INVOKE_SET_DRM_HDL,
    HI_FORMAT_INVOKE_SET_LOCALTIME,
    HI_FORMAT_INVOKE_SET_HEADERS,
    HI_FORMAT_INVOKE_SET_USERAGENT,
    HI_FORMAT_INVOKE_SET_REFERER,
    HI_FORMAT_INVOKE_SET_NOT_SUPPORT_BYTERANGE,
    HI_FORMAT_INVOKE_SET_LOG_LEVEL,
    HI_FORMAT_INVOKE_RTMP_RECEIVEAUDIO,
    HI_FORMAT_INVOKE_RTMP_RECEIVEVIDEO,
    HI_FORMAT_INVOKE_SET_BUFFER_UNDERRUN,
    HI_FORMAT_INVOKE_SET_HLS_LIVE_START_NUM,
    HI_FORMAT_INVOKE_PROTOCOL_USER=100,
    HI_FORMAT_INVOKE_SET_DOLBYRANGEINFO,
    HI_FORMAT_INVOKE_GET_DOLBYINFO,
    HI_FORMAT_INVOKE_BUTT
} HI_FORMAT_INVOKE_ID_E;
```

【修改说明】

新增成员 HI_FORMAT_INVOKE_SET_HEADERS，部分服务器对 HTTP Headers 有特殊要求，APP 可通过该接口将 Headers 设置给 HiPlayer，HiPlayer 将这些 Headers 附加在发送给服务器的报文中。

新增成员 HI_FORMAT_INVOKE_SET_HLS_LIVE_START_NUM，用于指定 HLS 直播启动时播放的第一个 Segment。

【注意事项】

无。

【范例】

无。

2.15 HiGO

2.15.1 概述

UNF 3.2.0 相对于 UNF 3.1 版本，在总体功能上有以下变更：

- 新增圆角矩形填充

2.15.1.1 圆角矩形填充

新增圆角矩形填充的功能。引起的变更如下：

接口

新增 HI_GO_FillRoundRect

2.15.2 函数接口

2.15.2.1 新增

HI_GO_FillRoundRect

【接口定义】

```
HI_S32 HI_GO_FillRoundRect(HI_HANDLE Surface, const HI_RECT* pRect,
HI_COLOR Color, HI_S32 s32Radius);
```

【新增原因】

UI 元素需要圆角矩形，使用圆角图片会降低性能，新增该接口。

【注意事项】

无。

【范例】

无。

3 UNF3.2.1 与 UNF3.2.0 之间的差异

3.1 AVPLAY

3.1.1 概述

UNF 3.2.1 相对于 UNF 3.2.0 版本，AVPLAY 模块在总体功能上有以下变更：

支持播放以 Tile 排布封装的 3D 码流

3.1.1.1 支持播放以 Tile 排布封装的 3D 码流

该特性实现以下功能：

直接播放以 Tile 排布封装的 3D 码流，引起的变更如下：

数据结构

修改 HI_UNF_VIDEO_FRAME_PACKING_TYPE_E

3.1.2 数据结构

3.1.2.1 新增

HI_UNF_VIDEO_FRAME_PACKING_TYPE_E

【结构体定义】

修改前：

```
typedef enum hiUNF_VIDEO_FRAME_PACKING_TYPE_E
{
    HI_UNF_FRAME_PACKING_TYPE_NONE,
    HI_UNF_FRAME_PACKING_TYPE_SIDE_BY_SIDE,
    HI_UNF_FRAME_PACKING_TYPE_TOP_AND_BOTTOM,
    HI_UNF_FRAME_PACKING_TYPE_TIME_INTERLACED,
    HI_UNF_FRAME_PACKING_TYPE_BUTT
}HI_UNF_VIDEO_FRAME_PACKING_TYPE_E;
```

修改后：

```
typedef enum hiUNF_VIDEO_FRAME_PACKING_TYPE_E
{
    HI_UNF_FRAME_PACKING_TYPE_NONE,
    HI_UNF_FRAME_PACKING_TYPE_SIDE_BY_SIDE,
    HI_UNF_FRAME_PACKING_TYPE_TOP_AND_BOTTOM,
    HI_UNF_FRAME_PACKING_TYPE_3D_TILE,
    HI_UNF_FRAME_PACKING_TYPE_TIME_INTERLACED,
    HI_UNF_FRAME_PACKING_TYPE_BUTT
}HI_UNF_VIDEO_FRAME_PACKING_TYPE_E;
```

【修改原因】

增加 Tile 格式枚举

【注意事项】

播放 Tile 格式封装的码流时，仅支持在 720P 制式下全屏输出

【范例】

```
HI_UNF_VIDEO_FRAME_PACKING_TYPE_E ePackType;
ePackType = HI_UNF_FRAME_PACKING_TYPE_3D_TILE;
HI_UNF_AVPLAY_SetAttr(hAvplay,HI_UNF_AVPLAY_ATTR_ID_FRMPACK_TYPE,&ePackTy
pe);
```

3.2 SPI

3.2.1 概述

UNF 3.2.1 相对于 UNF 3.2.0 版本，SPI 模块在总体功能有如下变更：

- 新增打开指定的 SPI 设备功能
- 新增关闭指定的 SPI 设备功能
- 新增设定 SPI 的工作模式和相关属性功能
- 新增读取当前 SPI 的工作模式和相关属性功能
- 新增发送数据功能
- 新增接收数据功能

3.2.1.1 打开 SPI 设备

数据结构

新增 HI_UNF_SPI_DEV_E

函数接口

新增 HI_UNF_SPI_Open

3.2.1.2 关闭 SPI 设备

数据结构

新增 HI_UNF_SPI_DEV_E

函数接口

新增 HI_UNF_SPI_Close

3.2.1.3 设定 SPI 的工作模式和相关属性

数据结构

- 新增 HI_UNF_SPI_SPH_E
- 新增 HI_UNF_SPI_SPH_E
- 新增 HI_UNF_SPI_FRF_E
- 新增 HI_UNF_SPI_BIGEND_E
- 新增 HI_UNF_SPI_ATTR_NM_S
- 新增 HI_UNF_SPI_ATTR_EXT_U
- 新增 HI_UNF_SPI_ATTR_S

函数接口

新增 HI_UNF_SPI_SetAttr

3.2.1.4 读取 SPI 的工作模式和相关属性

数据结构

新增 HI_UNF_SPI_ATTR_S

函数接口

新增 HI_UNF_SPI_SetAttr

3.2.1.5 发送数据

数据结构

无。

函数接口

新增 HI_UNF_SPI_Read

3.2.1.6 数据接收

数据结构

无。

函数接口

新增 HI_UNF_SPI_Write

3.2.2 数据结构

3.2.2.1 新增

HI_UNF_SPI_DEV_E

【枚举定义】

```
typedef enum hiUNF_SPI_DEV_E
{
    HI_UNF_SPI_DEV_0 =0 ,
    HI_UNF_SPI_DEV_1 = 1,
    HI_UNF_SPI_DEV_BUTT
}HI_UNF_SPI_DEV_E;
```

【新增原因】

一些芯片有两个 SPI 主控制器。

【注意事项】

无

【范例】

无。

HI_UNF_SPI_SPO_E

【枚举定义】

```
typedef enum hiUNF_SPI_SPO_E
{
    HI_UNF_SPI_SPO_0 = 0,
    HI_UNF_SPI_SPO_1 = 1
}HI_UNF_SPI_SPO_E;
```

【新增原因】

设定和获取 SPI 时钟输出的极性值。

【注意事项】

无。

【范例】

无。

HI_UNF_SPI_SPH_E

【枚举定义】

```
typedef enum hiUNF_SPI_SPH_E
{
    HI_UNF_SPI_SPH_0 = 0,
    HI_UNF_SPI_SPH_1 = 1
}HI_UNF_SPI_SPH_E;
```

【新增原因】

设定和获取 SPI 时钟输出的相位值。

【注意事项】

无。

【范例】

无。

HI_UNF_SPI_FRF_E

【枚举定义】

```
typedef enum hiUNF_SPI_FRF_E
{
    HI_UNF_SPI_FRF_MOTO = 0,
    HI_UNF_SPI_FRF_TI   = 1,
    HI_UNF_SPI_FRF_NM   = 2,
    HI_UNF_SPI_FRF_BUTT = 3
}HI_UNF_SPI_FRF_E;
```

【新增原因】

设定和获取 SPI 通信的帧格式类型。

【注意事项】

无。

【范例】

无。

HI_UNF_SPI_BIGEND_E

【枚举定义】

```
typedef enum hiUNF_SPI_BIGEND_E
{
    HI_UNF_SPI_BIGEND_LITTLE,
    HI_UNF_SPI_BIGEND_BIG,
}HI_UNF_SPI_BIGEND_E;
```

【新增原因】

设定大小端。

【注意事项】

无。

【范例】

无。

HI_UNF_SPI_ATTR_MOTO_S

【结构体定义】

```
typedef struct hiUNF_SPI_ATTR_MOTO_S
{
    HI_UNF_SPI_SPO_E enSpo;
    HI_UNF_SPI_SPH_E enSph;
}HI_UNF_SPI_ATTR_MOTO_S;
```

【新增原因】

设定 Motorola SPI 协议时钟的极值和相位。

【注意事项】

无。

【范例】

无。

HI_UNF_SPI_ATTR_NM_S

【结构体定义】

```
typedef struct hiUNF_SPI_ATTR_NM_S
{
    HI_BOOL bWaitEn;
    HI_U32 u32Waitval;
}HI_UNF_SPI_ATTR_NM_S;
```

【新增原因】

Microwire SPI 协议等待时间设定。

【注意事项】

无。

【范例】

无。

HI_UNF_SPI_ATTR_EXT_U

【联合体定义】

```
typedef union
{
    HI_UNF_SPI_ATTR_MOTO_S stMoto;
    HI_UNF_SPI_ATTR_NM_S stNm;
}HI_UNF_SPI_ATTR_EXT_U;
```

【新增原因】

设定 Microwire SPI 和 Motorola SPI 协议的专属属性。

【注意事项】

无。

【范例】

无。

HI_UNF_SPI_ATTR_S

【结构体定义】

```
typedef struct hiUNF_SPI_ATTR_S
{
    HI_UNF_SPI_DEV_E        enDev;
    HI_U32  u32Baud;
    HI_UNF_SPI_FRF_E        enFrf;
    HI_U32      u32Dss;
    HI_UNF_SPI_BIGEND_E enBigend;
    HI_UNF_SPI_ATTR_EXT_U unExtAttr;
}HI_UNF_SPI_ATTR_S;
```

【新增原因】

设定和获取 SPI 的工作方式属性。

【注意事项】

无。

【范例】

无。

3.2.3 函数接口

3.2.3.1 新增

HI_UNF_SPI_Open

【接口定义】

```
HI_S32 HI_UNF_SPI_Open(HI_UNF_SPI_DEV_E enDev);
```

【新增原因】

按客户需求定义。

【注意事项】

无

【范例】

无。

HI_UNF_SPI_Close

【接口定义】

```
HI_S32 HI_UNF_SPI_Open(HI_UNF_SPI_DEV_E enDev);
```

【新增原因】

按客户需求定义。

【注意事项】

无。

【范例】

无。

HI_UNF_SPI_SetAttr

【接口定义】

```
HI_S32 HI_UNF_SPI_SetAttr(HI_UNF_SPI_DEV_E enDev, HI_UNF_SPI_ATTR_S
*stAttr);
```

【新增原因】

该功能将设定 SPI 的工作方式。

【注意事项】

无。

【范例】

无。

HI_UNF_SPI_GetAttr

【接口定义】

```
HI_S32 HI_UNF_SPI_SetAttr(HI_UNF_SPI_DEV_E enDev, HI_UNF_SPI_ATTR_S
*stAttr);
```

【新增原因】

获取 SPI 当前工作模式相关属性。

【注意事项】

无。

【范例】

无。

HI_UNF_SPI_Read

【接口定义】

```
HI_S32 HI_UNF_SPI_Read(HI_UNF_SPI_DEV_E enDev, HI_U8 *pu8Read, HI_U32
u32ReadCnt);
```

【新增原因】

SPI 数据发送。

【注意事项】

无。

【范例】

无。

HI_UNF_SPI_Write

【接口定义】

```
HI_S32 HI_UNF_SPI_Write(HI_UNF_SPI_DEV_E enDev, HI_U8 *pu8Send, HI_U32
u32SendCnt);
```

【新增原因】

SPI 数据接收。

【注意事项】

无。

【范例】

无。

3.3 Frontend

3.3.1 概述

UNF3.2.1 相对于 UNF3.2.0 版本，在总体功能上有以下变更：

- 调整 TS 输出中，valid 和 sync 信号的配置顺序；
- 新增 hi3137 待机及唤醒功能；
- 新增 rafael836/mxl608/mxl214 驱动支持；

3.3.1.1 调整 TS 输出中，valid 和 sync 信号的配置顺序

该特性实现以下功能：

除 clk 信号外，其他信号从哪个管脚输出均可以配置。引起的变更如下：

数据结构

互换 HI_UNF_TUNER_OUTPUT_TS_E 结构体中，HI_UNF_TUNER_OUTPUT_TSVLD 和 HI_UNF_TUNER_OUTPUT_TSSYNC 的位置，保证和芯片 datasheet 一致。

3.3.1.2 新增 hi3137 待机及唤醒功能

该特性实现以下功能：

- 让正常工作的 hi3137 进入低功耗待机状态；
- 让低功耗待机状态的 hi3137 恢复正常工作。

引起的变更如下：

数据结构

新增 HI_UNF_TUNER_DEMOD_STATUS_E 结构体。

3.3.1.3 新增 rafael836/mxl608/mxl214 驱动支持

该特性实现以下功能：

- 新增-T tuner rafael836/mxl608 驱动，适配 hi3137；
- 新增-C tuner mxl214 驱动

引起的变更如下：

数据结构

- HI_UNF_TUNER_DEV_TYPE_E 结构体中，新增：
  - HI_UNF_TUNER_DEV_TYPE_RAFAEL836
  - HI_UNF_TUNER_DEV_TYPE_MXL608
  - HI_UNF_TUNER_DEV_TYPE_MXL214 成员
- HI_UNF_DEMOD_DEV_TYPE_E 结构体中，新增：
  HI_UNF_DEMOD_DEV_TYPE_MXL214 成员

3.3.2 数据结构

3.3.2.1 新增

HI_UNF_TUNER_DEMOD_STATUS_E

【结构体定义】

```
typedef enum hiUNF_TUNER_DEMOD_STATUS_E
{
    HI_UNF_TUNER_DEMODE_WAKE_UP = 0,
    HI_UNF_TUNER_DEMODE_STANDBY,
    HI_UNF_TUNER_DEMOD_STATUS_BUTT
}HI_UNF_TUNER_DEMOD_STATUS_E;
```

【新增原因】

新特性，方便配置 hi3137 进入待机及唤醒状态。

【注意事项】

无。

【范例】

请参考 sample/tuner/tuner_demo.c。

3.3.2.2 修改

HI_UNF_TUNER_DEV_TYPE_E

【结构体定义】

修改前：

```
typedef enum    hiUNF_TUNER_DEV_TYPE_E
{
    HI_UNF_TUNER_DEV_TYPE_XG_3BL,
    HI_UNF_TUNER_DEV_TYPE_CD1616,
    HI_UNF_TUNER_DEV_TYPE_ALPS_TDAE,
    HI_UNF_TUNER_DEV_TYPE_TDCC,
    HI_UNF_TUNER_DEV_TYPE_TDA18250,
```

```
    HI_UNF_TUNER_DEV_TYPE_CD1616_DOUBLE,
    HI_UNF_TUNER_DEV_TYPE_MT2081,
    HI_UNF_TUNER_DEV_TYPE_TMX7070X,
    HI_UNF_TUNER_DEV_TYPE_R820C,
    HI_UNF_TUNER_DEV_TYPE_MXL203,
    HI_UNF_TUNER_DEV_TYPE_AV2011,
    HI_UNF_TUNER_DEV_TYPE_SHARP7903,
    HI_UNF_TUNER_DEV_TYPE_MXL101,
    HI_UNF_TUNER_DEV_TYPE_MXL603,
    HI_UNF_TUNER_DEV_TYPE_IT9170,
    HI_UNF_TUNER_DEV_TYPE_IT9133,
    HI_UNF_TUNER_DEV_TYPE_TDA6651,
    HI_UNF_TUNER_DEV_TYPE_TDA18250B,
    HI_UNF_TUNER_DEV_TYPE_M88TS2022,
    HI_UNF_TUNER_DEV_TYPE_RDA5815,
    HI_UNF_TUNER_DEV_TYPE_MXL254,
    HI_UNF_TUNER_DEV_TYPE_CXD2861,
    HI_UNF_TUNER_DEV_TYPE_SI2147,
    HI_UNF_TUNER_DEV_TYPE_BUTT
} HI_UNF_TUNER_DEV_TYPE_E ;
```

修改后：

```
typedef enum    hiUNF_TUNER_DEV_TYPE_E
{
    HI_UNF_TUNER_DEV_TYPE_XG_3BL,
    HI_UNF_TUNER_DEV_TYPE_CD1616,
    HI_UNF_TUNER_DEV_TYPE_ALPS_TDAE,
    HI_UNF_TUNER_DEV_TYPE_TDCC,
    HI_UNF_TUNER_DEV_TYPE_TDA18250,
    HI_UNF_TUNER_DEV_TYPE_CD1616_DOUBLE,
    HI_UNF_TUNER_DEV_TYPE_MT2081,
    HI_UNF_TUNER_DEV_TYPE_TMX7070X,
    HI_UNF_TUNER_DEV_TYPE_R820C,
    HI_UNF_TUNER_DEV_TYPE_MXL203,
    HI_UNF_TUNER_DEV_TYPE_AV2011,
    HI_UNF_TUNER_DEV_TYPE_SHARP7903,
    HI_UNF_TUNER_DEV_TYPE_MXL101,
    HI_UNF_TUNER_DEV_TYPE_MXL603,
    HI_UNF_TUNER_DEV_TYPE_IT9170,
    HI_UNF_TUNER_DEV_TYPE_IT9133,
    HI_UNF_TUNER_DEV_TYPE_TDA6651,
    HI_UNF_TUNER_DEV_TYPE_TDA18250B,
    HI_UNF_TUNER_DEV_TYPE_M88TS2022,
    HI_UNF_TUNER_DEV_TYPE_RDA5815,
```

```
    HI_UNF_TUNER_DEV_TYPE_MXL254,
    HI_UNF_TUNER_DEV_TYPE_CXD2861,
    HI_UNF_TUNER_DEV_TYPE_SI2147,
    HI_UNF_TUNER_DEV_TYPE_RAFAEL836,
    HI_UNF_TUNER_DEV_TYPE_MXL608,
    HI_UNF_TUNER_DEV_TYPE_MXL214,
    HI_UNF_TUNER_DEV_TYPE_BUTT
} HI_UNF_TUNER_DEV_TYPE_E ;
```

【修改原因】

新增成员。

【注意事项】

Rafael836/mxl608 适配 hi3137，mxl214 拥有 tuner 和 demode 二合一功能

【范例】

无

HI_UNF_DEMOD_DEV_TYPE_E

【结构体定义】

修改前：

```
typedef enum    hiUNF_DEMOD_DEV_TYPE_E
{
    HI_UNF_DEMOD_DEV_TYPE_NONE,
    HI_UNF_DEMOD_DEV_TYPE_3130I = 0x100,
    HI_UNF_DEMOD_DEV_TYPE_3130E,
    HI_UNF_DEMOD_DEV_TYPE_J83B,
    HI_UNF_DEMOD_DEV_TYPE_AVL6211,
    HI_UNF_DEMOD_DEV_TYPE_MXL101,
    HI_UNF_DEMOD_DEV_TYPE_MN88472,
    HI_UNF_DEMOD_DEV_TYPE_IT9170,
    HI_UNF_DEMOD_DEV_TYPE_IT9133,
    HI_UNF_DEMOD_DEV_TYPE_3136,
    HI_UNF_DEMOD_DEV_TYPE_3136I,
    HI_UNF_DEMOD_DEV_TYPE_MXL254,
    HI_UNF_DEMOD_DEV_TYPE_CXD2837,
    HI_UNF_DEMOD_DEV_TYPE_3137,
    HI_UNF_DEMOD_DEV_TYPE_BUTT
} HI_UNF_DEMOD_DEV_TYPE_E;
```

修改后：

```
typedef enum    hiUNF_DEMOD_DEV_TYPE_E
{
```

```
    HI_UNF_DEMOD_DEV_TYPE_NONE,
    HI_UNF_DEMOD_DEV_TYPE_3130I = 0x100,
    HI_UNF_DEMOD_DEV_TYPE_3130E,
    HI_UNF_DEMOD_DEV_TYPE_J83B,
    HI_UNF_DEMOD_DEV_TYPE_AVL6211,
    HI_UNF_DEMOD_DEV_TYPE_MXL101,
    HI_UNF_DEMOD_DEV_TYPE_MN88472,
    HI_UNF_DEMOD_DEV_TYPE_IT9170,
    HI_UNF_DEMOD_DEV_TYPE_IT9133,
    HI_UNF_DEMOD_DEV_TYPE_3136,
    HI_UNF_DEMOD_DEV_TYPE_3136I,
    HI_UNF_DEMOD_DEV_TYPE_MXL254,
    HI_UNF_DEMOD_DEV_TYPE_CXD2837,
    HI_UNF_DEMOD_DEV_TYPE_3137,
    HI_UNF_DEMOD_DEV_TYPE_MXL214,
    HI_UNF_DEMOD_DEV_TYPE_BUTT
} HI_UNF_DEMOD_DEV_TYPE_E;
```

【修改原因】

新增成员。

【注意事项】

mxl214 拥有 tuner 和 demode 二合一功能。

【范例】

无

4 UNF3.2.2 与 UNF3.2.1 之间的差异

4.1 HiGO

4.1.1 概述

UNF3.2.2 相对于 UNF3.2.1 版本，HiGO 在总体功能上有以下变更：

新增文本处理错误码

4.1.1.1 文本处理错误码

增加文本处理错误码，方便问题定位修改：

数据结构

修改 HIGO_TEXTOUT_ERR_S

接口

无。

4.1.2 数据结构

4.1.2.1 新增

无

4.1.2.2 修改

HIGO_TEXTOUT_ERR_S

【结构体定义】

修改前：

```
typedef enum
{
```

```
    ERR_TEXTOUT_INVRECT = 0,
    ERR_TEXTOUT_UNSUPPORT_CHARSET,
    ERR_TEXTOUT_ISUSING,
    ERR_TEXTOUT_BUTT
} HIGO_TEXTOUT_ERR_S;
```

修改后：

```
typedef enum
{
    ERR_TEXTOUT_INVRECT = 0,
    ERR_TEXTOUT_UNSUPPORT_CHARSET,
    ERR_TEXTOUT_ISUSING,
    ERR_TEXTOUT_NOIMPLEMENT,
    ERR_TEXTOUT_SHAPE,
    ERR_TEXTOUT_MAX_CHAR,
    ERR_TEXTOUT_CHAR_SET,
    ERR_TEXTOUT_BIDI,
    ERR_TEXTOUT_ERRCODE_MAX = 0x1F,
    ERR_TEXTOUT_INTERNAL,
    ERR_TEXTOUT_BUTT
} HIGO_TEXTOUT_ERR_S;
```

【修改说明】

无。

【注意事项】

无。

【范例】

无。

4.2 HiFB

4.2.1 概述

UNF3.2.2 相对于 UNF3.2.1 版本，HiFB 在总体功能上有以下变更：

- 新增支持 Fence 同步送显接口

4.2.1.1 Fence 同步送显

为了解决图形裂屏问题及提高 UI 帧率（流畅性），新增 Fence 同步送显功能，对如下数据结构以及接口进行了变更：

数据结构

新增 HIFB_HWC_LAYERINFO_S

接口

新增 FBIO_HWC_REFRESH

4.2.2 数据结构

4.2.2.1 新增

HIFB_HWC_LAYERINFO_S

【结构体定义】

```
typedef struct
{
    HIFB_POINT_S stPos;
    HIFB_RECT    stInRect;
    HI_U32       u32LayerAddr;
    HI_U32       u32Stride;
    HI_U32       u32Alpha;
    HI_BOOL      bPreMul;
    HIFB_COLOR_FMT_E eFmt;
    HI_S32 s32AcquireFenceFd;
    HI_S32 s32ReleaseFenceFd;
}HIFB_HWC_LAYERINFO_S;
```

【新增原因】

为了解决图形裂屏问题及提高 UI 帧率（流畅性）。【注意事项】

无。

【范例】

无。

4.2.3 函数接口

4.2.3.1 新增

FBIO_HWC_REFRESH

【接口定义】

```
#define FBIO_HWC_REFRESH _IOR(IOC_TYPE_HIFB, 146, HIFB_HWC_LAYERINFO_S*)
```

【新增原因】

为了彻底解决图形裂屏问题及提高 UI 帧率（流畅性）。【注意事项】

无。

【范例】

无。

4.3 MCE

4.3.1 概述

UNF 3.2.2 相对于 UNF 3.2.1 版本，MCE 在总体功能上有以下变更：

- 查询播放数据是否有效功能。

4.3.1.1 查询播放数据是否有效功能

该功能支持用户查询 fastplay 分区的播放数据是否有效，以便用户判断是否需要更新 fastplay 分区。引起的变更如下：

数据结构

修改 HI_UNF_MCE_PLAY_PARAM_S

函数接口

无

4.3.2 数据结构

4.3.2.1 修改

HI_UNF_MCE_PLAY_PARAM_S

【结构体定义】

修改前：

```
typedef struct hiUNF_MCE_PLAY_PARAM_S
{
    HI_UNF_MCE_PLAY_TYPE_E          enPlayType;
    HI_BOOL                         bPlayEnable;
    union
    {
        HI_UNF_MCE_DVB_PARAM_S     stDvbParam;
        HI_UNF_MCE_TSFILE_PARAM_S  stTsParam;
        HI_UNF_MCE_ANI_PARAM_S     stAniParam;
    }unParam;
}HI_UNF_MCE_PLAY_PARAM_S;
```

修改后：

```
typedef struct hiUNF_MCE_PLAY_PARAM_S
{
    HI_UNF_MCE_PLAY_TYPE_E          enPlayType;
    HI_BOOL                         bPlayEnable;
    union
    {
        HI_UNF_MCE_DVB_PARAM_S     stDvbParam;
        HI_UNF_MCE_TSFILE_PARAM_S  stTsParam;
        HI_UNF_MCE_ANI_PARAM_S     stAniParam;
    }unParam;
    HI_BOOL                         bContentValid;
}HI_UNF_MCE_PLAY_PARAM_S;
```

【修改原因】

标志相应的 fastplay 分区的播放数据是否有效，以便用户判断是否需要更新 fastplay 分区。

【注意事项】

无。

【范例】

无。

4.4 PDM

4.4.1 概述

UNF 3.2.2 相对于 UNF 3.2.1 版本，PDM 模块在总体功能上有以下变更：

- 查询更新瞬播备份分区的参数和数据

4.4.1.1 查询更新瞬播备份分区的参数和数据

该功能引起的变更如下：

数据结构

无。

函数接口

- 新增 HI_UNF_PDM_GetPlayBakParam
- 新增 HI_UNF_PDM_UpdatePlayBakParam
- 新增 HI_UNF_PDM_GetPlayBakContent
- 新增 HI_UNF_PDM_UpdatePlayBakContent

4.4.2 函数接口

4.4.2.1 新增

HI_UNF_PDM_GetPlayBakParam

【接口定义】

```
HI_S32  HI_UNF_PDM_GetPlayBakParam(HI_UNF_MCE_PLAY_PARAM_S *pstPlayParam);
```

【新增原因】

获取瞬播备份分区参数的接口。

【注意事项】

无。

【范例】

无。

HI_UNF_PDM_UpdatePlayBakParam

【接口定义】

```
HI_S32  HI_UNF_PDM_UpdatePlayBakParam(HI_UNF_MCE_PLAY_PARAM_S
*pstPlayParam);
```

【新增原因】

更新瞬播备份分区参数的接口。

【注意事项】

无。

【范例】

无。

HI_UNF_PDM_GetPlayBakContent

【接口定义】

```
HI_S32  HI_UNF_PDM_GetPlayBakContent(HI_U8 *pu8Content, HI_U32 u32Size);
```

【新增原因】

获取瞬播备份分区数据的接口。

【注意事项】

无。

【范例】

无。

HI_UNF_PDM_UpdatePlayBakContent

【接口定义】

```
HI_S32  HI_UNF_PDM_UpdatePlayBakContent(HI_U8 *pu8Content, HI_U32
u32Size);
```

【新增原因】

更新瞬播备份分区数据的接口。

【注意事项】

无。

【范例】

无。

4.5 VO

4.5.1 概述

UNF 3.2.2 相对于 UNF 3.2.1 版本，VO 模块在总体功能上有以下变更：

- 支持 4K 相关制式

4.5.1.1 支持 4K 相关制式

该特性实现如下功能：可以通过 HI_UNF_DISP_SetFormat 接口设置 4K 制式。引起的变更如下：

数据结构

修改 HI_UNF_ENC_FMT_E

4.5.2 数据结构

4.5.2.1 修改

HI_UNF_ENC_FMT_E

【结构体定义】

修改前：

```
typedef enum hiUNF_ENC_FMT_E
{
    …
    HI_UNF_ENC_FMT_VESA_2560X1440_60_RB,
    HI_UNF_ENC_FMT_VESA_2560X1600_60_RB,
    HI_UNF_ENC_FMT_BUTT
}HI_UNF_ENC_FMT_E;
```

修改后：

```
typedef enum hiUNF_ENC_FMT_E
{
    …
    HI_UNF_ENC_FMT_VESA_2560X1440_60_RB,
    HI_UNF_ENC_FMT_VESA_2560X1600_60_RB,
    HI_UNF_ENC_FMT_3840X2160_24 = 0x100,
    HI_UNF_ENC_FMT_3840X2160_25,
    HI_UNF_ENC_FMT_3840X2160_30,
    HI_UNF_ENC_FMT_4096X2160_24,

    HI_UNF_ENC_FMT_BUTT
}HI_UNF_ENC_FMT_E;
```

【修改原因】

制式枚举中新增 4K 相关制式。

【注意事项】

无。

【范例】

无。

4.6 SOUND

4.6.1 概述

UNF3.2.2 相对于 UNF3.2.1，SOUND 在总体功能上有以下变更：

- 新增 SPDIF Category Code 设置与获取接口
- 新增基于 Track 的智能音量设置与获取接口
- 修改 SPDIF SCMS 结构体

4.6.1.1 新增 SPDIF Category Code 设置与获取接口

数据结构

- 新增 HI_UNF_SND_SPDIF_CATEGORYCODE_E

函数接口

- 新增 HI_UNF_SND_SetSpdifCategoryCode
- 新增 HI_UNF_SND_GetSpdifCategoryCode

4.6.1.2 新增基于 Track 的智能音量设置与获取接口

数据结构

无。

函数接口

- 新增 HI_UNF_SND_SetTrackSmartVolume
- 新增 HI_UNF_SND_GetTrackSmartVolume

4.6.1.3 修改 SPDIF SCMS 结构体

数据结构

- 修改 HI_UNF_SND_SPDIF_SCMSMODE_E

函数接口

无。

4.6.2 数据结构

4.6.2.1 新增

HI_UNF_SND_SPDIF_CATEGORYCODE_E

【枚举定义】

```
typedef enum hiHI_UNF_SND_SPDIF_CATEGORYCODE_E
{
    HI_UNF_SND_SPDIF_CATEGORY_GENERAL = 0x00,
    HI_UNF_SND_SPDIF_CATEGORY_BROADCAST_JP = 0x10,
    HI_UNF_SND_SPDIF_CATEGORY_BROADCAST_USA,
    HI_UNF_SND_SPDIF_CATEGORY_BROADCAST_EU,
    HI_UNF_SND_SPDIF_CATEGORY_PCM_CODEC = 0x20,
    HI_UNF_SND_SPDIF_CATEGORY_DIGITAL_SNDSAMPLER,
    HI_UNF_SND_SPDIF_CATEGORY_DIGITAL_MIXER,
    HI_UNF_SND_SPDIF_CATEGORY_DIGITAL_SNDPROCESSOR,
    HI_UNF_SND_SPDIF_CATEGORY_SRC,
    HI_UNF_SND_SPDIF_CATEGORY_MD = 0x30,
    HI_UNF_SND_SPDIF_CATEGORY_DVD,
    HI_UNF_SND_SPDIF_CATEGORY_SYNTHESISER = 0x40,
    HI_UNF_SND_SPDIF_CATEGORY_MIC,
    HI_UNF_SND_SPDIF_CATEGORY_DAT = 0x50,
    HI_UNF_SND_SPDIF_CATEGORY_DCC,
    HI_UNF_SND_SPDIF_CATEGORY_VCR,
    HI_UNF_SND_SPDIF_CATEGORY_BUTT
} HI_UNF_SND_SPDIF_CATEGORYCODE_E;
```

【新增原因】

增加SPDIF Category Code分类码。

【注意事项】

无。

【范例】

无。

4.6.2.2 删除

无。

4.6.2.3 修改

HI_UNF_SND_SPDIF_SCMSMODE_E

【结构体定义】

修改前：

```
typedef enum hiHI_UNF_SND_SPDIF_SCMSMODE_E
{
    HI_UNF_SND_SPDIF_SCMSMODE_COPYALLOW,
    HI_UNF_SND_SPDIF_SCMSMODE_COPYONCE,
```

```
    HI_UNF_SND_SPDIF_SCMSMODE_COPYPROHIBITED,
    HI_UNF_SND_SPDIF_SCMSMODE_BUTT
} HI_UNF_SND_SPDIF_SCMSMODE_E;
```

修改后：

```
typedef enum hiHI_UNF_SND_SPDIF_SCMSMODE_E
{
    HI_UNF_SND_SPDIF_SCMSMODE_COPYALLOW,
    HI_UNF_SND_SPDIF_SCMSMODE_COPYONCE,
    HI_UNF_SND_SPDIF_SCMSMODE_COPYNOMORE,
    HI_UNF_SND_SPDIF_SCMSMODE_COPYPROHIBITED,
    HI_UNF_SND_SPDIF_SCMSMODE_BUTT
} HI_UNF_SND_SPDIF_SCMSMODE_E;
```

【修改说明】

增加不可复制模式。

【注意事项】

无。

【范例】

无。

4.6.3 函数接口

4.6.3.1 新增

HI_UNF_SND_SetSpdifCategoryCode

【接口定义】

```
HI_S32 HI_UNF_SND_SetSpdifCategoryCode(HI_UNF_SND_E enSound,
HI_UNF_SND_OUTPUTPORT_E enOutPort, HI_UNF_SND_SPDIF_CATEGORYCODE_E
enSpdifCategoryCode);
```

【新增原因】

设置 SPDIF Category Code 分类码。

【注意事项】

无。

【范例】

无。

HI_UNF_SND_GetSpdifCategoryCode

【接口定义】

```
HI_S32 HI_UNF_SND_GetSpdifCategoryCode(HI_UNF_SND_E enSound,
HI_UNF_SND_OUTPUTPORT_E enOutPort, HI_UNF_SND_SPDIF_CATEGORYCODE_E
*penSpdifCategoryCode);
```

【新增原因】

获取 SPDIF Category Code 分类码。

【注意事项】

无。

【范例】

无。

HI_UNF_SND_SetTrackSmartVolume

【接口定义】

```
HI_S32 HI_UNF_SND_SetTrackSmartVolume(HI_HANDLE hTrack, HI_BOOL
bEnable);
```

【新增原因】

设置智能音量功能。

【注意事项】

无。

【范例】

无。

HI_UNF_SND_GetTrackSmartVolume

【接口定义】

```
HI_S32  HI_UNF_SND_GetAllTrackMute(HI_UNF_SND_E enSound, HI_BOOL
*pbMute);
```

【新增原因】

获取智能音量设置状态。

【注意事项】

无。

【范例】

无。

4.7 SPI

4.7.1 概述

UNF 3.2.2 相对于 UNF 3.2.1 版本，SPI 模块在总体功能有如下变更：

- 新增 SPI 片选（CS）信号控制选择功能

4.7.1.1 SPI 片选（CS）信号控制选择

数据结构

- 新增 HI_UNF_SPI_DEV_E
- 修改 HI_UNF_SPI_ATTR_S

函数接口

新增 HI_UNF_SPI_ReadExt

4.7.2 数据结构

4.7.2.1 新增

HI_UNF_SPI_CFGCS_E

【枚举定义】

```
typedef enum hiUNF_SPI_CFGCS_E
{
    HI_UNF_SPI_LOGIC_CS =0 ,
    HI_UNF_SPI_GPIO_CS = 1
}HI_UNF_SPI_CFGCS_E;
```

【新增原因】

由于一些 SPI 器件通信时，要求 CS 信号在发送个别命令与接收数据期间始终保持低电平，逻辑 CS 无法满足上述需求，需要配置成 GPIO 模式自行控制，满足通信需求。

【注意事项】

无。

【范例】

无。

4.7.2.2 修改

HI_UNF_SPI_ATTR_S

【结构体定义】

修改前：

```
typedef struct hiUNF_SPI_ATTR_S
{
    HI_UNF_SPI_DEV_E        enDev;
    HI_U32                          u32Baud;
    HI_UNF_SPI_FRF_E        enFrf;
    HI_U32      u32Dss;
    HI_UNF_SPI_BIGEND_E enBigend;
    HI_UNF_SPI_ATTR_EXT_U unExtAttr;
}HI_UNF_SPI_ATTR_S;
```

修改后：

```
typedef struct hiUNF_SPI_ATTR_S
{
    HI_UNF_SPI_DEV_E        enDev;
    HI_UNF_SPI_CFGCS_E csCfg;
    HI_U32                          u32Baud;
    HI_UNF_SPI_FRF_E        enFrf;
    HI_U32      u32Dss;
    HI_UNF_SPI_BIGEND_E enBigend;
    HI_UNF_SPI_ATTR_EXT_U un ExtAttr;
}HI_UNF_SPI_ATTR_S;
```

【修改原因】

SPI 属性中新增片选（CS）信号控制选择成员。

【注意事项】

无。

【范例】

无。

4.7.3 函数接口

4.7.3.1 新增

HI_UNF_SPI_ReadExt

【接口定义】

```
HI_S32 HI_UNF_SPI_ReadExt(HI_UNF_SPI_DEV_E enDev, HI_U8 *pu8Send, HI_U32
u32SendCnt, HI_U8 *pu8Read, HI_U32 u32ReadCnt);
```

【新增原因】

按客户需求定义。

【注意事项】

如果配置为 GPIO 控制 CS，调用 HI_UNF_SPI_Write 发送命令，再调用 HI_UNF_SPI_Read 接收数据时，CS 在空闲时仍会处于高电平态，因为任何 SPI 器件都需要这样的通信模式。如果有器件要求个别命令发送后，CS 持续保持低电平接收数据，调用 HI_UNF_SPI_ReadExt 接口即可满足需求。

【范例】

无。

4.8 PQ

4.8.1 概述

UNF3.2.2 相对于 UNF3.2.1 版本，PQ 在总体功能上有以下变更：

- 新增单个 PQ 参数设置和获取功能
- 新增 SR 演示功能
- 新增颜色增强功能
- 新增动态对比度增强功能
- 新增 PQ 算法模块开关功能
- 新增演示模式开关功能

4.8.1.1 新增单个 PQ 参数设置和获取功能

为了更加灵活使用 PQ 参数，新增了如下接口：

数据结构

无。

接口

- 新增 HI_UNF_PQ_GetBrightness
- 新增 HI_UNF_PQ_SetBrightness
- 新增 HI_UNF_PQ_GetContrast
- 新增 HI_UNF_PQ_SetContrast
- 新增 HI_UNF_PQ_GetSaturation
- 新增 HI_UNF_PQ_SetSaturation
- 新增 HI_UNF_PQ_GetHue
- 新增 HI_UNF_PQ_SetHue
- 新增 HI_UNF_PQ_GetSharpness
- 新增 HI_UNF_PQ_SetSharpness

4.8.1.2 新增 SR 演示功能

为了使用非 4K 源的 4K 输出，新增如下数据结构及接口：

数据结构

- 新增 HI_UNF_PQ_SR_DEMO_E

接口

- 新增 HI_UNF_PQ_GetSRMode
- 新增 HI_UNF_PQ_SetSRMode

4.8.1.3 新增颜色增强功能

为了增强颜色的效果，适配 UI 菜单，新增如下数据结构及接口：

数据结构

- 新增 HI_UNF_PQ_COLOR_ENHANCE_E
- 新增 HI_UNF_PQ_FLESHTONE_E
- 新增 HI_UNF_PQ_SIX_BASE_S
- 新增 HI_UNF_PQ_COLOR_SPEC_MODE_E
- 新增 HI_UNF_PQ_COLOR_ENHANCE_S

接口

- 新增 HI_UNF_PQ_GetColorGain
- 新增 HI_UNF_PQ_SetColorGain
- 新增 HI_UNF_PQ_GetColorEnhanceParam
- 新增 HI_UNF_PQ_SetColorEnhanceParam

4.8.1.4 新增动态对比度增强功能

为了自适应增强图像对比度的效果，新增如下接口：

数据结构

无。

接口

- 新增 HI_UNF_PQ_GetDynamicContrast
- 新增 HI_UNF_PQ_SetDynamicContrast

4.8.1.5 新增 PQ 算法模块开关功能

增加对 PQ 算法开关的控制，新增如下接口：

数据结构

- 新增 HI_UNF_PQ_MODULE_E

接口

- 新增 HI_UNF_PQ_SetPQModule
- 新增 HI_UNF_PQ_GetPQModule

4.8.1.6 新增演示模式开关功能

为了对比显示算法开与关对图像质量效果的影响，增加如下接口：

数据结构

- 新增 HI_UNF_PQ_DEMO_E

接口

- 新增 HI_UNF_PQ_SetDemo

4.8.2 数据结构

4.8.2.1 新增

HI_UNF_PQ_DEMO_E

【枚举定义】

```
typedef enum hiUNF_PQ_DEMO_E
{
    HI_UNF_PQ_DEMO_SHARPNESS = 0,
    HI_UNF_PQ_DEMO_DCI,
    HI_UNF_PQ_DEMO_COLOR,
    HI_UNF_PQ_DEMO_SR,
    HI_UNF_PQ_DEMO_ALL,

    HI_UNF_PQ_DEMO_BUTT
}  HI_UNF_PQ_DEMO_E;
```

【新增原因】

演示模式子项，用于演示模式的选择控制。

【注意事项】

无。

【范例】

无。

HI_UNF_PQ_MODULE_E

【枚举定义】

```
typedef enum hiUNF_PQ_MODULE_E
```

```
{
    HI_UNF_PQ_MODULE_SHARPNESS = 0,
    HI_UNF_PQ_MODULE_DCI,
    HI_UNF_PQ_MODULE_COLOR,
    HI_UNF_PQ_MODULE_SR,
    HI_UNF_PQ_MODULE_ALL,

    HI_UNF_PQ_MODULE_BUTT
}  HI_UNF_PQ_MODULE_E;
```

【新增原因】

PQ 算法子项，用于算法的开关控制。

【注意事项】

无。

【范例】

无。

HI_UNF_PQ_SR_DEMO_E

【枚举定义】

```
typedef enum hiUNF_PQ_SR_DEMO_E
{
    HI_UNF_PQ_SR_DISABLE  = 0,
    HI_UNF_PQ_SR_ENABLE_R,
    HI_UNF_PQ_SR_ENABLE_L,
    HI_UNF_PQ_SR_ENABLE_A,

    HI_UNF_PQ_SR_DEMO_BUTT
}HI_UNF_PQ_SR_DEMO_E;
```

【新增原因】

SR 算法作用区域子项，用于 SR 算法作用区域控制。

【注意事项】

无。

【范例】

无。

HI_UNF_PQ_SIX_BASE_S

【枚举定义】

```
typedef struct  hiUNF_PQ_SIX_BASE_S
```

```
{
    HI_U32  u32Red;
    HI_U32  u32Green;
    HI_U32  u32Blue;

    HI_U32  u32Cyan;
    HI_U32  u32Magenta;
    HI_U32  u32Yellow;
}HI_UNF_PQ_SIX_BASE_S;
```

【新增原因】

为了进行自定义颜色段调节。

【注意事项】

无。

【范例】

无。

HI_UNF_PQ_FLESHTONE_E

【枚举定义】

```
typedef enum hiUNF_PQ_FLESHTONE_E
{
    HI_UNF_PQ_FLESHTONE_GAIN_OFF = 0,
    HI_UNF_PQ_FLESHTONE_GAIN_LOW,
    HI_UNF_PQ_FLESHTONE_GAIN_MID,
    HI_UNF_PQ_FLESHTONE_GAIN_HIGH,

    HI_UNF_PQ_FLESHTONE_GAIN_BUTT
}  HI_UNF_PQ_FLESHTONE_E;
```

【新增原因】

为了肤色增强等级调节。

【注意事项】

无。

【范例】

无。

HI_UNF_PQ_COLOR_ENHANCE_E

【枚举定义】

```
typedef enum hiUNF_PQ_COLOR_ENHANCE_E
```

```
{
    HI_UNF_PQ_COLOR_ENHANCE_FLESHTONE = 0,
    HI_UNF_PQ_COLOR_ENHANCE_SIX_BASE,
    HI_UNF_PQ_COLOR_ENHANCE_SPEC_COLOR_MODE,
    HI_UNF_PQ_COLOR_ENHANCE_BUTT
} HI_UNF_PQ_COLOR_ENHANCE_E;
```

【新增原因】

为了颜色增强功能的类型选择。

【注意事项】

无。

【范例】

无。

HI_UNF_PQ_COLOR_SPEC_MODE_E

【枚举定义】

```
typedef enum hiUNF_PQ_COLOR_SPEC_MODE_E
{
    HI_UNF_PQ_COLOR_MODE_RECOMMEND = 0,
    HI_UNF_PQ_COLOR_MODE_BLUE,
    HI_UNF_PQ_COLOR_MODE_GREEN,
    HI_UNF_PQ_COLOR_MODE_BG,
    HI_UNF_PQ_COLOR_MODE_BUTT
} HI_UNF_PQ_COLOR_SPEC_MODE_E;
```

【新增原因】

为了适配颜色增强类型的选择。

【注意事项】

无。

【范例】

无。

HI_UNF_PQ_COLOR_ENHANCE_S

【枚举定义】

```
typedef struct  hiUNF_PQ_COLOR_ENHANCE_S
{
    HI_UNF_PQ_COLOR_ENHANCE_E    enColorEnhanceType;
    union
    {
```

```
        HI_UNF_PQ_FLESHTONE_E    enFleshtone;
        HI_UNF_PQ_SIX_BASE_S     stSixBase;
        HI_UNF_PQ_COLOR_SPEC_MODE_E   enColorMode;

    } unColorGain;
} HI_UNF_PQ_COLOR_ENHANCE_S;
```

【新增原因】

颜色增强功能的参数选择。

【注意事项】

无。

【范例】

无。

4.8.3 函数接口

4.8.3.1 新增

HI_UNF_PQ_GetBrightness

【接口定义】

```
extern HI_S32 HI_UNF_PQ_GetBrightness(HI_UNF_DISP_E enChan, HI_U32
*pu32Brightness);
```

【新增原因】

Android UI 界面调用接口，获取亮度。

【注意事项】

无。

【范例】

无。

HI_UNF_PQ_SetBrightness

【接口定义】

```
extern HI_S32 HI_UNF_PQ_SetBrightness(HI_UNF_DISP_E enChan, HI_U32
u32Brightness);
```

【新增原因】

Android UI 界面调用接口，设置亮度。

【注意事项】

亮度值，有效范围：0~100。

【范例】

无。

HI_UNF_PQ_GetContrast

【接口定义】

```
extern HI_S32 HI_UNF_PQ_GetContrast(HI_UNF_DISP_E enChan, HI_U32
*pu32Contrast);
```

【新增原因】

Android UI 界面调用接口，获取对比度。

【注意事项】

无

【范例】

无。

HI_UNF_PQ_SetContrast

【接口定义】

```
extern HI_S32 HI_UNF_PQ_SetContrast(HI_UNF_DISP_E enChan, HI_U32
u32Contrast);
```

【新增原因】

Android UI 界面调用接口，设置对比度。

【注意事项】

对比度，有效范围：0~100。

【范例】

无。

HI_UNF_PQ_GetSaturation

【接口定义】

```
extern HI_S32 HI_UNF_PQ_GetSaturation(HI_UNF_DISP_E enChan, HI_U32
*pu32Saturation);
```

【新增原因】

Android UI 界面调用接口，获取饱和度。

【注意事项】

无。

【范例】

无。

HI_UNF_PQ_SetSaturation

【接口定义】

```
extern HI_S32 HI_UNF_PQ_SetSaturation(HI_UNF_DISP_E enChan, HI_U32
u32Saturation);
```

【新增原因】

Android UI 界面调用接口，设置饱和度。

【注意事项】

饱和度，有效范围：0~100。

【范例】

无。

HI_UNF_PQ_GetHue

【接口定义】

```
extern HI_S32 HI_UNF_PQ_GetHue(HI_UNF_DISP_E enChan, HI_U32 *pu32Hue);
```

【新增原因】

Android UI 界面调用接口，获取色调。

【注意事项】

无。

【范例】

无。

HI_UNF_PQ_SetHue

【接口定义】

```
extern HI_S32 HI_UNF_PQ_SetHue(HI_UNF_DISP_E enChan, HI_U32 u32Hue);
```

【新增原因】

Android UI 界面调用接口，设置色调。

【注意事项】

色调，有效范围：0~100。

【范例】

无。

HI_UNF_PQ_GetSRMode

【接口定义】

```
extern HI_S32 HI_UNF_PQ_GetSRMode(HI_UNF_DISP_E enChan,
HI_UNF_PQ_SR_DEMO_E *penType);
```

【新增原因】

Android UI 界面调用接口，获取 SR 演示类型。

【注意事项】

无。

【范例】

无。

HI_UNF_PQ_SetSRMode

【接口定义】

```
extern HI_S32 HI_UNF_PQ_SetSRMode(HI_UNF_DISP_E enChan,
HI_UNF_PQ_SR_DEMO_E enType);
```

【新增原因】

Android UI 界面调用接口，设置 SR 演示类型。

【注意事项】

无。

【范例】

无。

HI_UNF_PQ_GetSharpness

【接口定义】

```
extern HI_S32 HI_UNF_PQ_GetSharpness(HI_UNF_DISP_E enChan, HI_U32
*pu32Sharpness);
```

【新增原因】

Android UI 界面调用接口，获取清晰度。

【注意事项】

无。

【范例】

无。

HI_UNF_PQ_SetSharpness

【接口定义】

```
extern HI_S32 HI_UNF_PQ_SetSharpness(HI_UNF_DISP_E enChan, HI_U32
u32Sharpness);
```

【新增原因】

Android UI 界面调用接口，设置清晰度。

【注意事项】

清晰度，有效范围：0~100。

【范例】

无。

HI_UNF_PQ_GetColorGain

【接口定义】

```
extern HI_S32 HI_UNF_PQ_GetColorGain(HI_UNF_DISP_E enChan, HI_U32
*pu32ColorGainLevel);
```

【新增原因】

获取颜色增强。

【注意事项】

无

【范例】

无。

HI_UNF_PQ_SetColorGain

【接口定义】

```
extern HI_S32 HI_UNF_PQ_SetColorGain(HI_UNF_DISP_E enChan, HI_U32
u32ColorGainLevel);
```

【新增原因】

设置颜色增强。

【注意事项】

颜色增强值范围：0-100。

【范例】

无。

HI_UNF_PQ_GetColorEnhanceParam

【接口定义】

```
extern HI_S32 HI_UNF_PQ_GetColorEnhanceParam(HI_UNF_PQ_COLOR_ENHANCE_S
*pstColorEnhanceParam);
```

【新增原因】

Android UI 界面调用接口，获取颜色增强的类型和强度。

【注意事项】

无。

【范例】

无。

HI_UNF_PQ_SetColorEnhanceParam

【接口定义】

```
extern HI_S32 HI_UNF_PQ_SetColorEnhanceParam(HI_UNF_PQ_COLOR_ENHANCE_S
stColorEnhanceParam);
```

【新增原因】

Android UI 界面调用接口，设置颜色增强的类型和强度。

【注意事项】

无。

【范例】

无。

HI_UNF_PQ_GetDynamicContrast

【接口定义】

```
extern HI_S32 HI_UNF_PQ_GetDynamicContrast(HI_U32 *pu32DCIlevel);
```

【新增原因】

Android UI 界面调用接口，获取 DCI（动态对比度增强）的强度范围。

【注意事项】

无。

【范例】

无。

HI_UNF_PQ_SetDynamicContrast

【接口定义】

```
extern HI_S32 HI_UNF_PQ_SetDynamicContrast(HI_U32 u32DCIlevel);
```

【新增原因】

Android UI 界面调用接口，设置 DCI（动态对比度增强）的强度范围。

【注意事项】

动态对比度等级，有效范围：0~100。

【范例】

无。

HI_UNF_PQ_SetPQModule

【接口定义】

```
extern HI_S32 HI_UNF_PQ_SetPQModule( HI_UNF_PQ_MODULE_E enFlags, HI_U32
u32OnOff);
```

【新增原因】

Android UI 界面调用接口，设置 PQ 算法开关。

【注意事项】

无。

【范例】

无。

HI_UNF_PQ_GetPQModule

【接口定义】

```
extern HI_S32 HI_UNF_PQ_GetPQModule( HI_UNF_PQ_MODULE_E enFlags, HI_U32
*pu32OnOff);
```

【新增原因】

Android UI 界面调用接口，获取 PQ 算法开关。

【注意事项】

无。

【范例】

无。

HI_UNF_PQ_SetDemo

【接口定义】

```
extern HI_S32 HI_UNF_PQ_SetDemo( HI_UNF_PQ_DEMO_E enFlags, HI_U32
u32OnOff);
```

【新增原因】

Android UI 界面调用接口，设置卖场模式开关。

【注意事项】

无。

【范例】

无。

5 UNF3.2.3 与 UNF3.2.2 之间的差异

5.1 Demux

5.1.1 概述

UNF3.2.3 相对于 UNF.3.2.2 版本，Demux 在总体功能上有以下变更：

- 新增支持 TAG DEAL 特性。

5.1.1.1 支持 TAG DEAL 特性

因为新增支持 TAG DEAL 的原因，对如下数据结构以及接口进行了变更：

数据结构

- 新增 HI_UNF_DMX_TAG_SYNC_MODE_E
- 新增 HI_UNF_DMX_TAG_ATTR_S

接口

- 新增 HI_UNF_DMX_GetDmxTagAttr
- 新增 HI_UNF_DMX_SetDmxTagAttr

5.1.2 数据结构

5.1.2.1 新增

HI_UNF_DMX_TAG_SYNC_MODE_E

【定义】

```
typedef enum hiUNF_DMX_TAG_SYNC_MODE_E
{
    HI_UNF_DMX_TAG_HEAD_SYNC = 0x0,
    HI_UNF_DMX_NORMAL_HEAD_SYNC = 0x1,
```

```
}HI_UNF_DMX_TAG_SYNC_MODE_E;
```

【新增原因】

新特性，枚举支持的 Tag 同步模式。

【注意事项】

无。

【范例】

无。

HI_UNF_DMX_TAG_ATTR_S

【定义】

```
typedef struct hiUNF_DMX_TAG_ATTR_S
{
    HI_U8    au8Tag[MAX_TAG_LENGTH];
    HI_U32   u32TagLen;
    HI_BOOL bEnabled;
    HI_UNF_DMX_TAG_SYNC_MODE_E enSyncMod;
}HI_UNF_DMX_TAG_ATTR_S;
```

【新增原因】

新特性，定义 Tag 可配置的属性参数。

【注意事项】

无。

【范例】

无。

5.1.3 函数接口

5.1.3.1 新增

HI_UNF_DMX_GetDmxTagAttr

【接口定义】

```
HI_S32 HI_UNF_DMX_GetDmxTagAttr(HI_U32 u32DmxId, HI_UNF_DMX_TAG_ATTR_S
*pstAttr);
```

【新增原因】

新特性，增加获取 Tag 当前配置属性的功能。

【注意事项】

无。

【范例】

无。

HI_UNF_DMX_SetDmxTagAttr

【接口定义】

```
HI_S32 HI_UNF_DMX_SetDmxTagAttr(HI_U32 u32DmxId, HI_UNF_DMX_TAG_ATTR_S
*pstAttr);
```

【新增原因】

新特性，增加设置 Tag 配置属性的功能。

【注意事项】

无。

【范例】

无。

6 UNF3.2.4 与 UNF3.2.3 之间的差异

6.1 Demux

6.1.1 概述

UNF3.2.4 相对于 UNF.3.2.3 版本，Demux 在总体功能上有以下变更：

- 新增支持 TSO 输出反压 Ram 输出速率特性。

6.1.1.1 支持 TSO 输出反压 Ram 输出速率特性

判断 TSO 端口的数据是否来自 Ram 端口，如果是，则对该 Ram 端口开启反压以实现对 TSO 端口速率的控制，对如下数据结构以及接口进行了变更：

数据结构

- 新增 HI_UNF_DMX_TSI_ATTACH_TSO_S
- 修改 HI_UNF_DMX_INVOKE_TYPE_E

6.1.2 数据结构

6.1.2.1 新增

HI_UNF_DMX_TSI_ATTACH_TSO_S

【定义】

```
typedef struct hiUNF_DMX_TSI_ATTACH_TSO_S
{
    HI_UNF_DMX_PORT_E       enTSI;
    HI_UNF_DMX_TSO_PORT_E   enTSO;
}HI_UNF_DMX_TSI_ATTACH_TSO_S;
```

【新增原因】

新特性，支持 TSO 输出反压 Ram 输出速率，用以标记相应的 TSO 和 TSI 端口。

【注意事项】

无。

【范例】

无。

6.1.2.2 修改

HI_UNF_DMX_INVOKE_TYPE_E

【定义】

```
typedef enum hiUNF_DMX_INVOKE_TYPE
{
    HI_UNF_DMX_INVOKE_TYPE_CHAN_CC_REPEAT_SET
    HI_UNF_DMX_INVOKE_TYPE_PUSI_SET,
    HI_UNF_DMX_INVOKE_TYPE_TEI_SET,
    HI_UNF_DMX_INVOKE_TYPE_TSI_ATTACH_TSO,
    HI_UNF_DMX_INVOKE_TYPE_BUTT
} HI_UNF_DMX_INVOKE_TYPE_E;
```

【修改原因】

新增 HI_UNF_DMX_Invoke 接口支持命令 HI_UNF_DMX_INVOKE_TYPE_TSI_ATTACH_TSO，用于绑定 TSI 的输入为某一个 TSO。

【注意事项】

无。

【范例】

无。

6.2 Frontend

6.2.1 概述

UNF3.2.4 相对于 UNF3.2.3 版本，Frontend 在总体功能上有以下变更：

- 新增 DVB-T or DVB-T2 信号的自动检测功能；
- 新增 unicable 用户频段自动盲扫功能。

6.2.1.1 新增 DVB-T or DVB-T2 信号的自动检测功能

数据结构

修改 HI_UNF_TUNER_SAMPLE_DATALEN_E。

函数接口

无。

6.2.1.2 新增 unicable 用户频段自动盲扫功能

数据结构

新增 HI_UNF_TUNER_SCR_UB_S

函数接口

- 新增 HI_UNF_TUNER_GetAgc
- 新增 HI_UNF_UNICABLE_ScanAndInstall_UB
- 新增 HI_UNF_UNICABLE_GetUBInfo
- 新增 HI_UNF_UNICABLE_SetCurUB

6.2.2 数据结构

6.2.2.1 新增

HI_UNF_TUNER_SCR_UB_S

【结构体定义】

```
typedef struct hiUNF_TUNER_SCR_UB_S
{
    HI_U32          u32SCRNo;
    HI_S32          s32CenterFreq;
}HI_UNF_TUNER_SCR_UB_S;
```

【新增原因】

新特性，用于记录用户频段的频段号和中心频率。

【注意事项】

无。

【范例】

请参考 source/msp/api/frontend/unf_unicable.c。

6.2.2.2 修改

HI_UNF_TUNER_SIG_TYPE_E

【结构体定义】

修改前：

```
typedef enum    hiTUNER_SIG_TYPE_E
{
    HI_UNF_TUNER_SIG_TYPE_CAB = 0,
    HI_UNF_TUNER_SIG_TYPE_SAT,
    HI_UNF_TUNER_SIG_TYPE_DVB_T,
    HI_UNF_TUNER_SIG_TYPE_DVB_T2,
    HI_UNF_TUNER_SIG_TYPE_ISDB_T,
    HI_UNF_TUNER_SIG_TYPE_ATSC_T,
    HI_UNF_TUNER_SIG_TYPE_DTMB,
    HI_UNF_TUNER_SIG_TYPE_J83B,
    HI_UNF_TUNER_SIG_TYPE_BUTT
} HI_UNF_TUNER_SIG_TYPE_E;
```

修改后：

```
typedef enum    hiTUNER_SIG_TYPE_E
{
    HI_UNF_TUNER_SIG_TYPE_CAB = 1,
    HI_UNF_TUNER_SIG_TYPE_SAT = 2,
    HI_UNF_TUNER_SIG_TYPE_DVB_T = 4,
    HI_UNF_TUNER_SIG_TYPE_DVB_T2 = 8,
    HI_UNF_TUNER_SIG_TYPE_ISDB_T = 16,
    HI_UNF_TUNER_SIG_TYPE_ATSC_T = 32,
    HI_UNF_TUNER_SIG_TYPE_DTMB = 64,
    HI_UNF_TUNER_SIG_TYPE_J83B = 128,
    HI_UNF_TUNER_SIG_TYPE_BUTT
} HI_UNF_TUNER_SIG_TYPE_E;
```

【修改原因】

将各种信号类型的枚举值修改为 2 的指次方数，方便位操作。

【注意事项】

无。

【范例】

请参考 source/msp/api/frontend/unf_tuner.c

6.2.3 函数接口

6.2.3.1 新增

HI_UNF_TUNER_GetAgc

【接口定义】

```
HI_S32 HI_UNF_TUNER_GetAgc(HI_U32 u32TunerId,HI_S32 s32CenterFreq, HI_S32 *ps32Agc);
```

【新增原因】

unicable 用户频段盲扫需要获取 agc 值。

【注意事项】

无。

【范例】

无。

HI_UNF_UNICABLE_ScanAndInstall_UB

【接口定义】

```
HI_S32 HI_UNF_UNICABLE_ScanAndInstall_UB(HI_U32 u32TunerId);
```

【新增原因】

unicable 用户频段自动盲扫接口。

【注意事项】

无。

【范例】

无。

HI_UNF_UNICABLE_GetUBInfo

【接口定义】

```
HI_S32 HI_UNF_UNICABLE_GetUBInfo(HI_U32 u32TunerId, HI_UNF_TUNER_SCR_UB_S *pUBInfo);
```

【新增原因】

获取 unicable 用户频段盲扫结果。

【注意事项】

无。

【范例】

无。

HI_UNF_UNICABLE_SetCurUB

【接口定义】

```
HI_S32 HI_UNF_UNICABLE_SetCurUB(HI_U32 u32TunerId);
```

【新增原因】

占用指定 unicable 用户频段。

【注意事项】

无。

【范例】

无。

6.3 SOUND

6.3.1 概述

UNF3.2.4 相对于 UNF3.2.3，在总体功能上有以下变更：

- 修改 Codec StreamInfo 结构体

6.3.1.1 修改 Codec StreamInfo 结构体

数据结构

修改 HI_UNF_ACODEC_STREAMINFO_S

函数接口

无。

6.3.2 数据结构

6.3.2.1 新增

无。

6.3.2.2 删除

无。

6.3.2.3 修改

HI_UNF_ACODEC_STREAMINFO_S

【结构体定义】

修改前：

```
typedef struct hiUNF_ACODEC_STREAMINFO_S
{
    HI_U32   enACodecType;
    HI_U32              enSampleRate;
    HI_UNF_BIT_DEPTH_E   enBitDepth;
}HI_UNF_ACODEC_STREAMINFO_S;
```

修改后：

```
typedef struct hiUNF_ACODEC_STREAMINFO_S
{
    HI_U32   enACodecType;
    HI_U32              enSampleRate;
    HI_UNF_BIT_DEPTH_E   enBitDepth;
    HI_U32              u32Channel;
}HI_UNF_ACODEC_STREAMINFO_S;
```

【修改说明】

增加声道数。

【注意事项】

无。

【范例】

无。

6.3.3 函数接口

无。

7 UNF3.2.5 与 UNF3.2.4 之间的差异

7.1 AVPLAY

7.1.1 概述

UNF 3.2.5 相对于 UNF 3.2.4 版本，AVPLAY 模块在总体功能上有以下变更：

增加音频事件回调函数类型

7.1.1.1 增加音频事件回调函数类型

该特性实现以下功能：

AVPLAY 新增三种上报事件：音频信息变化事件，音频不支持事件，音频帧出错事件，引起的变更如下：

数据结构

修改 HI_UNF_AVPLAY_EVENT_E

7.1.2 数据结构

7.1.2.1 修改

HI_UNF_AVPLAY_EVENT_E

【结构体定义】

修改前：

```
typedef enum hiUNF_AVPLAY_EVENT_E
{
    HI_UNF_AVPLAY_EVENT_EOS,
    HI_UNF_AVPLAY_EVENT_STOP,
    HI_UNF_AVPLAY_EVENT_RNG_BUF_STATE,
```

```
    HI_UNF_AVPLAY_EVENT_NORM_SWITCH,
    HI_UNF_AVPLAY_EVENT_FRAMEPACKING_CHANGE,
    HI_UNF_AVPLAY_EVENT_NEW_VID_FRAME,
    HI_UNF_AVPLAY_EVENT_NEW_AUD_FRAME,
    HI_UNF_AVPLAY_EVENT_NEW_USER_DATA,
    HI_UNF_AVPLAY_EVENT_GET_AUD_ES,
    HI_UNF_AVPLAY_EVENT_IFRAME_ERR,
    HI_UNF_AVPLAY_EVENT_SYNC_PTS_JUMP,
    HI_UNF_AVPLAY_EVENT_SYNC_STAT_CHANGE,
    HI_UNF_AVPLAY_EVENT_VID_BUF_STATE,
    HI_UNF_AVPLAY_EVENT_AUD_BUF_STATE,
    HI_UNF_AVPLAY_EVENT_VID_UNSUPPORT,
    HI_UNF_AVPLAY_EVENT_VID_ERR_RATIO,
    HI_UNF_AVPLAY_EVENT_BUTT
} HI_UNF_AVPLAY_EVENT_E;
```

修改后：

```
typedef enum hiUNF_AVPLAY_EVENT_E
{
    HI_UNF_AVPLAY_EVENT_EOS,
    HI_UNF_AVPLAY_EVENT_STOP,
    HI_UNF_AVPLAY_EVENT_RNG_BUF_STATE,
    HI_UNF_AVPLAY_EVENT_NORM_SWITCH,
    HI_UNF_AVPLAY_EVENT_FRAMEPACKING_CHANGE,
    HI_UNF_AVPLAY_EVENT_NEW_VID_FRAME,
    HI_UNF_AVPLAY_EVENT_NEW_AUD_FRAME,
    HI_UNF_AVPLAY_EVENT_NEW_USER_DATA,
    HI_UNF_AVPLAY_EVENT_GET_AUD_ES,
    HI_UNF_AVPLAY_EVENT_IFRAME_ERR,
    HI_UNF_AVPLAY_EVENT_SYNC_PTS_JUMP,
    HI_UNF_AVPLAY_EVENT_SYNC_STAT_CHANGE,
    HI_UNF_AVPLAY_EVENT_VID_BUF_STATE,
    HI_UNF_AVPLAY_EVENT_AUD_BUF_STATE,
    HI_UNF_AVPLAY_EVENT_VID_UNSUPPORT,
```

```
    HI_UNF_AVPLAY_EVENT_VID_ERR_RATIO,
    HI_UNF_AVPLAY_EVENT_AUD_INFO_CHANGE,
    HI_UNF_AVPLAY_EVENT_AUD_UNSUPPORT,
    HI_UNF_AVPLAY_EVENT_AUD_FRAME_ERR,
    HI_UNF_AVPLAY_EVENT_BUTT
} HI_UNF_AVPLAY_EVENT_E;
```

【修改原因】

新增音频信息变化，不支持音频，音频帧错误三种事件回调函数类型

【注意事项】

无

【范例】

无

7.2 SOUND

7.2.1 概述

UNF3.2.5 相对于 UNF3.2.4，SOUND 模块在总体功能上有以下变更：

修改 PCM DELAY 结构体

7.2.1.1 修改 PCM DELAY 结构体

数据结构

修改 HI_UNF_I2S_PCMDELAY_E

7.2.2 数据结构

7.2.2.1 修改

HI_UNF_I2S_PCMDELAY_E

【结构体定义】

修改前：

```
typedef enum hiHI_UNF_I2S_PCMDELAY_E
{
    HI_UNF_I2S_PCM_0_DELAY = 0,
    HI_UNF_I2S_PCM_1_DELAY = 1,
    HI_UNF_I2S_PCM_8_DELAY = 8,
```

```
    HI_UNF_I2S_PCM_16_DELAY = 16,
    HI_UNF_I2S_PCM_32_DELAY = 32,
    HI_UNF_I2S_PCM_DELAY_BUTT
} HI_UNF_I2S_PCMDELAY_E;
```

修改后：

```
typedef enum hiHI_UNF_I2S_PCMDELAY_E
{
    HI_UNF_I2S_PCM_0_DELAY = 0,
    HI_UNF_I2S_PCM_1_DELAY = 1,
    HI_UNF_I2S_PCM_8_DELAY = 8,
    HI_UNF_I2S_PCM_16_DELAY = 16,
    HI_UNF_I2S_PCM_17_DELAY = 17,
    HI_UNF_I2S_PCM_24_DELAY = 24,
    HI_UNF_I2S_PCM_32_DELAY = 32,
    HI_UNF_I2S_PCM_DELAY_BUTT
} HI_UNF_I2S_PCMDELAY_E;
```

【修改说明】

增加 I2S PCM Delay 选项。

【注意事项】

无

【范例】

无

7.3 VENC

7.3.1 概述

UNF 3.2.5 相对于 UNF 3.2.4 版本，VENC 模块在总体功能上有以下变更：

新增码率控制波动阈值

7.3.1.1 码率控制波动阈值

视频编码器在码率控制的过程中，遇到异常情况，会有一定的码率波动，编码器可以通过内部丢帧的方式，有效控制编码器输出码率的波动范围。通过设置该值，可以配置编码器码率控制允许的波动阈值，范围 0-100 或者 0xFFFFFFFF，如该值设置成 20，则表示码率控制算法允许输出码率存在一定的波动，且波动范围是设置的目标码率的±20%；设置成 0，则表示码率控制算法需要严格控制，基本不允许输出码率波

动；设置成 100，则表示码率控制相当宽松，且波动范围是设置的目标码率的±100%；设置成 0xFFFFFFFF，则表示编码器码率控制时不会做丢帧处理，靠内部算法收敛来控制码率，引起的变更如下：

数据结构

修改 HI_UNF_VENC_CHN_ATTR_S

7.3.2 数据结构

7.3.2.1 修改

HI_UNF_VENC_CHN_ATTR_S

【结构体定义】

修改前：

```
typedef struct hiUNF_VENC_CHN_ATTR_S
{
    HI_UNF_VCODEC_TYPE_E        enVencType;
    HI_UNF_VCODEC_CAP_LEVEL_E   enCapLevel;
    HI_UNF_H264_PROFILE_E       enVencProfile;
    HI_U32                      u32Width;
    HI_U32                      u32Height;
    HI_U32                      u32StrmBufSize;
    HI_U32                      u32RotationAngle;
    HI_BOOL                     bSlcSplitEn;
    HI_U32                      u32TargetBitRate;
    HI_U32                      u32TargetFrmRate;
    HI_U32                      u32InputFrmRate;
    HI_U32                      u32MaxQp;
    HI_U32                      u32MinQp;
    HI_BOOL                     bQuickEncode;
    HI_U8                       u8Priority;
    HI_U32                     u32Qlevel;
}HI_UNF_VENC_CHN_ATTR_S;
```

修改后：

```
typedef struct hiUNF_VENC_CHN_ATTR_S
{
    HI_UNF_VCODEC_TYPE_E        enVencType;
    HI_UNF_VCODEC_CAP_LEVEL_E   enCapLevel;
    HI_UNF_H264_PROFILE_E       enVencProfile;
    HI_U32                      u32Width;
    HI_U32                      u32Height;
    HI_U32                      u32StrmBufSize;
```

```
    HI_U32                          u32RotationAngle;
    HI_BOOL                         bSlcSplitEn;
    HI_U32                          u32TargetBitRate;
    HI_U32                          u32TargetFrmRate;
    HI_U32                          u32InputFrmRate;
    HI_U32                          u32MaxQp;
    HI_U32                          u32MinQp;
    HI_BOOL                         bQuickEncode;
    HI_U8                           u8Priority;
    HI_U32                          u32Qlevel;
    HI_U32                          u32DriftRateThr;
}HI_UNF_VENC_CHN_ATTR_S;
```

【修改原因】

支持用户根据实际场景设置是否允许码率控制丢帧及相应的码率波动范围。

【注意事项】

当设置为 0xFFFFFFFF 时，表示编码器码率控制时不作丢帧处理。

【范例】

无

7.4 CIPHER

7.4.1 概述

UNF 3.2.5 相对于 UNF 3.2.4 版本，CIPHER 模块在总体功能上有以下变更：

新增 Irdeto MSR2.2 相关规格的支持。

新增 Irdeto MSR2.2 相关规格的支持

该特性引起的变更如下：

数据结构

修改 HI_UNF_CIPHER_CA_TYPE_E

修改 HI_UNF_CIPHER_HASH_TYPE_E

7.4.2 数据结构

7.4.2.1 修改

HI_UNF_CIPHER_CA_TYPE_E

【结构体定义】

修改前：

```
typedef enum hiUNF_CIPHER_CA_TYPE_E
{
    HI_UNF_CIPHER_CA_TYPE_R2R = 0x0,
    HI_UNF_CIPHER_CA_TYPE_SP,
    HI_UNF_CIPHER_CA_TYPE_CSA2,
    HI_UNF_CIPHER_CA_TYPE_CSA3,
    HI_UNF_CIPHER_CA_TYPE_MISC,
    HI_UNF_CIPHER_CA_TYPE_GDRM,
    HI_UNF_CIPHER_CA_TYPE_BLPK,
    HI_UNF_CIPHER_CA_TYPE_LPK,
}HI_UNF_CIPHER_CA_TYPE_E;
```

修改后：

```
typedef enum hiUNF_CIPHER_CA_TYPE_E
{
    HI_UNF_CIPHER_CA_TYPE_R2R = 0x0,
    HI_UNF_CIPHER_CA_TYPE_SP,
    HI_UNF_CIPHER_CA_TYPE_CSA2,
    HI_UNF_CIPHER_CA_TYPE_CSA3,
    HI_UNF_CIPHER_CA_TYPE_MISC,
    HI_UNF_CIPHER_CA_TYPE_GDRM,
    HI_UNF_CIPHER_CA_TYPE_BLPK,
    HI_UNF_CIPHER_CA_TYPE_LPK,
    HI_UNF_CIPHER_CA_TYPE_IRDETO_HCA,
}HI_UNF_CIPHER_CA_TYPE_E;
```

【修改原因】

新增 Irdeto MSR2.2 CBC-MAC 计算功能。

【注意事项】

无

【范例】

请参考：sample/advca/ sample_ca_irdeto_msr2.2.c 中的第 52 个命令：

“52: AES CBC-MAC HIGH-LEVEL CODE AUTHENTICATION”。

HI_UNF_CIPHER_HASH_TYPE_E

【结构体定义】

修改前：

```
typedef enum hiHI_UNF_CIPHER_HASH_TYPE_E
{
    HI_UNF_CIPHER_HASH_TYPE_SHA1,
    HI_UNF_CIPHER_HASH_TYPE_SHA256,
    HI_UNF_CIPHER_HASH_TYPE_HMAC_SHA1,
    HI_UNF_CIPHER_HASH_TYPE_HMAC_SHA256,
    HI_UNF_CIPHER_HASH_TYPE_BUTT,
}HI_UNF_CIPHER_HASH_TYPE_E;
```

修改后：

```
typedef enum hiHI_UNF_CIPHER_HASH_TYPE_E
{
    HI_UNF_CIPHER_HASH_TYPE_SHA1,
    HI_UNF_CIPHER_HASH_TYPE_SHA256,
    HI_UNF_CIPHER_HASH_TYPE_HMAC_SHA1,
     HI_UNF_CIPHER_HASH_TYPE_HMAC_SHA256,
     HI_UNF_CIPHER_HASH_TYPE_IRDETO_CBCMAC,
    HI_UNF_CIPHER_HASH_TYPE_BUTT,
}HI_UNF_CIPHER_HASH_TYPE_E;
```

【修改原因】

新增 Irdeto MSR2.2 CBC-MAC 计算功能。

【注意事项】

无

【范例】

请参考如下流程：

```
stCipherHashAttr.eShaType = HI_UNF_CIPHER_HASH_TYPE_IRDETO_CBCMAC;
…
HI_UNF_CIPHER_HashInit(…);
HI_UNF_CIPHER_HashUpdate(…);
HI_UNF_CIPHER_HashFinal(…);
```

7.5 KEYLED

7.5.1 概述

UNF 3.2.5 相对于 UNF 3.2.4 版本，KEYLED 模块在总体功能上有以下变更：

新增设置面板(仅支持 FD650)锁频 LED。

新增设置锁频 LED

可以设置面板锁频 LED 的开关，但仅支持 FD650 面板设置。

函数接口

新增 HI_UNF_LED_SetLockPin

7.5.2 函数接口

7.5.2.1 新增

HI_UNF_LED_SetLockPin

【接口定义】

```
HI_S32 HI_UNF_LED_SetLockPin(HI_BOOL setLock);
```

【新增原因】

用于设置面板锁频 LED 的开关。

【注意事项】

目前只有 FD650 面板支持该设置。

【范例】

无

7.6 PDM

7.6.1 概述

UNF 3.2.5 相对于 UNF 3.2.4 版本，PDM 模块在总体功能上有以下变更：

新增 HDMI 基本配置参数结构体

7.6.1.1 增加 HDMI 基本配置参数结构体

该功能引起的变更如下：

数据结构

新增 HI_UNF_PDM_HDMI_PARAM_S

修改 HI_UNF_PDM_BASEPARAM_TYPE_E

7.6.2 数据结构

7.6.2.1 新增

HI_UNF_PDM_HDMI_PARAM_S

【结构体定义】

```
typedef struct hiUNF_PDM_HDMI_PARAM_S
{
    HI_U8    *pu8EDID;
    HI_U32   *pu32EDIDLen;
}HI_UNF_PDM_HDMI_PARAM_S;
```

【新增原因】

保存 HDMI 基本配置参数

【注意事项】

无

【范例】

无

7.6.2.2 修改

HI_UNF_PDM_BASEPARAM_TYPE_E

【枚举定义】

修改前：

```
typedef enum hiUNF_PDM_BASEPARAM_TYPE_E
{
    HI_UNF_PDM_BASEPARAM_DISP0 = 0,
    HI_UNF_PDM_BASEPARAM_DISP1,
    HI_UNF_PDM_BASEPARAM_SOUND0 = 10,
    HI_UNF_PDM_BASEPARAM_SOUND1,
    HI_UNF_PDM_BASEPARAM_SOUND2,
    HI_UNF_PDM_BASEPARAM_BUTT = 0xFFFF,
```

```
}HI_UNF_PDM_BASEPARAM_TYPE_E;
```

修改后：

```
typedef enum hiUNF_PDM_BASEPARAM_TYPE_E
{
    HI_UNF_PDM_BASEPARAM_DISP0 = 0,
    HI_UNF_PDM_BASEPARAM_DISP1,
    HI_UNF_PDM_BASEPARAM_SOUND0 = 10,
    HI_UNF_PDM_BASEPARAM_SOUND1,
    HI_UNF_PDM_BASEPARAM_SOUND2,
    HI_UNF_PDM_BASEPARAM_HDMI = 20,
    HI_UNF_PDM_BASEPARAM_BUTT = 0xFFFF,
}HI_UNF_PDM_BASEPARAM_TYPE_E;
```

【修改原因】

基本参数中增加 HDMI 类型

【注意事项】

无

【范例】

无

8 UNF3.2.6 与 UNF3.2.5 之间的差异

8.1 VO

8.1.1 概述

UNF 3.2.6 相对于 UNF 3.2.5 版本，VO 模块在总体功能上有以下变更：

增加刷新率为 23.976、29.97、59.94 的显示制式

8.1.1.1 增加显示制式

增加刷新率为 23.976、29.97、59.94 的显示制式，引起的变更如下：

数据结构

修改 HI_UNF_ENC_FMT_E

8.1.2 数据结构

8.1.2.1 修改

HI_UNF_ENC_FMT_E

【结构体定义】

修改前：

```
typedef enum hiUNF_ENC_FMT_E
{
    HI_UNF_ENC_FMT_1080P_60 = 0,     /**<1080p 60 Hz*/
    HI_UNF_ENC_FMT_1080P_50,         /**<1080p 50 Hz*/
    HI_UNF_ENC_FMT_1080P_30,         /**<1080p 30 Hz*/
    HI_UNF_ENC_FMT_1080P_25,         /**<1080p 25 Hz*/
    HI_UNF_ENC_FMT_1080P_24,         /**<1080p 24 Hz*/

    HI_UNF_ENC_FMT_1080i_60,         /**<1080i 60 Hz*/
```

```
    HI_UNF_ENC_FMT_1080i_50,        /**<1080i 50 Hz*/

    HI_UNF_ENC_FMT_720P_60,         /**<720p 60 Hz*/
    HI_UNF_ENC_FMT_720P_50,         /**<720p 50 Hz */

    HI_UNF_ENC_FMT_576P_50,         /**<576p 50 Hz*/
    HI_UNF_ENC_FMT_480P_60,         /**<480p 60 Hz*/

    HI_UNF_ENC_FMT_PAL,            /* B D G H I PAL */
    HI_UNF_ENC_FMT_PAL_N,           /* (N)PAL       */
    HI_UNF_ENC_FMT_PAL_Nc,          /* (Nc)PAL      */

    HI_UNF_ENC_FMT_NTSC,           /* (M)NTSC      */
    HI_UNF_ENC_FMT_NTSC_J,          /* NTSC-J       */
    HI_UNF_ENC_FMT_NTSC_PAL_M,      /* (M)PAL       */

    HI_UNF_ENC_FMT_SECAM_SIN,       /**< SECAM_SIN*/
    HI_UNF_ENC_FMT_SECAM_COS,       /**< SECAM_COS*/

    HI_UNF_ENC_FMT_1080P_24_FRAME_PACKING,
    HI_UNF_ENC_FMT_720P_60_FRAME_PACKING,
    HI_UNF_ENC_FMT_720P_50_FRAME_PACKING,

    HI_UNF_ENC_FMT_861D_640X480_60,
    HI_UNF_ENC_FMT_VESA_800X600_60,
    HI_UNF_ENC_FMT_VESA_1024X768_60,
    HI_UNF_ENC_FMT_VESA_1280X720_60,
    HI_UNF_ENC_FMT_VESA_1280X800_60,
    HI_UNF_ENC_FMT_VESA_1280X1024_60,
    HI_UNF_ENC_FMT_VESA_1360X768_60,
    HI_UNF_ENC_FMT_VESA_1366X768_60,
    HI_UNF_ENC_FMT_VESA_1400X1050_60,
    HI_UNF_ENC_FMT_VESA_1440X900_60,
    HI_UNF_ENC_FMT_VESA_1440X900_60_RB,
    HI_UNF_ENC_FMT_VESA_1600X900_60_RB,
    HI_UNF_ENC_FMT_VESA_1600X1200_60,
    HI_UNF_ENC_FMT_VESA_1680X1050_60,
    HI_UNF_ENC_FMT_VESA_1680X1050_60_RB,
    HI_UNF_ENC_FMT_VESA_1920X1080_60,
    HI_UNF_ENC_FMT_VESA_1920X1200_60,
    HI_UNF_ENC_FMT_VESA_1920X1440_60,
    HI_UNF_ENC_FMT_VESA_2048X1152_60,
    HI_UNF_ENC_FMT_VESA_2560X1440_60_RB,
    HI_UNF_ENC_FMT_VESA_2560X1600_60_RB,
```

```
    HI_UNF_ENC_FMT_3840X2160_24 = 0x100,
    HI_UNF_ENC_FMT_3840X2160_25,
    HI_UNF_ENC_FMT_3840X2160_30,
    HI_UNF_ENC_FMT_4096X2160_24,
    HI_UNF_ENC_FMT_BUTT
}HI_UNF_ENC_FMT_E;
```

修改后：

```
typedef enum hiUNF_ENC_FMT_E
{
    HI_UNF_ENC_FMT_1080P_60 = 0,     /**<1080p 60 Hz*/
    HI_UNF_ENC_FMT_1080P_50,         /**<1080p 50 Hz*/
    HI_UNF_ENC_FMT_1080P_30,         /**<1080p 30 Hz*/
    HI_UNF_ENC_FMT_1080P_25,         /**<1080p 25 Hz*/
    HI_UNF_ENC_FMT_1080P_24,         /**<1080p 24 Hz*/

    HI_UNF_ENC_FMT_1080i_60,         /**<1080i 60 Hz*/
    HI_UNF_ENC_FMT_1080i_50,         /**<1080i 50 Hz*/

    HI_UNF_ENC_FMT_720P_60,          /**<720p 60 Hz*/
    HI_UNF_ENC_FMT_720P_50,          /**<720p 50 Hz */

    HI_UNF_ENC_FMT_576P_50,          /**<576p 50 Hz*/
    HI_UNF_ENC_FMT_480P_60,          /**<480p 60 Hz*/

    HI_UNF_ENC_FMT_PAL,             /* B D G H I PAL */
    HI_UNF_ENC_FMT_PAL_N,            /* (N)PAL       */
    HI_UNF_ENC_FMT_PAL_Nc,           /* (Nc)PAL      */

    HI_UNF_ENC_FMT_NTSC,            /* (M)NTSC      */
    HI_UNF_ENC_FMT_NTSC_J,           /* NTSC-J       */
    HI_UNF_ENC_FMT_NTSC_PAL_M,       /* (M)PAL       */

    HI_UNF_ENC_FMT_SECAM_SIN,        /**< SECAM_SIN*/
    HI_UNF_ENC_FMT_SECAM_COS,        /**< SECAM_COS*/

    HI_UNF_ENC_FMT_1080P_24_FRAME_PACKING,
    HI_UNF_ENC_FMT_720P_60_FRAME_PACKING,
    HI_UNF_ENC_FMT_720P_50_FRAME_PACKING,

    HI_UNF_ENC_FMT_861D_640X480_60,
    HI_UNF_ENC_FMT_VESA_800X600_60,
    HI_UNF_ENC_FMT_VESA_1024X768_60,
```

```
    HI_UNF_ENC_FMT_VESA_1280X720_60,
    HI_UNF_ENC_FMT_VESA_1280X800_60,
    HI_UNF_ENC_FMT_VESA_1280X1024_60,
    HI_UNF_ENC_FMT_VESA_1360X768_60,
    HI_UNF_ENC_FMT_VESA_1366X768_60,
    HI_UNF_ENC_FMT_VESA_1400X1050_60,
    HI_UNF_ENC_FMT_VESA_1440X900_60,
    HI_UNF_ENC_FMT_VESA_1440X900_60_RB,
    HI_UNF_ENC_FMT_VESA_1600X900_60_RB,
    HI_UNF_ENC_FMT_VESA_1600X1200_60,
    HI_UNF_ENC_FMT_VESA_1680X1050_60,
    HI_UNF_ENC_FMT_VESA_1680X1050_60_RB,
    HI_UNF_ENC_FMT_VESA_1920X1080_60,
    HI_UNF_ENC_FMT_VESA_1920X1200_60,
    HI_UNF_ENC_FMT_VESA_1920X1440_60,
    HI_UNF_ENC_FMT_VESA_2048X1152_60,
    HI_UNF_ENC_FMT_VESA_2560X1440_60_RB,
    HI_UNF_ENC_FMT_VESA_2560X1600_60_RB,

    HI_UNF_ENC_FMT_3840X2160_24 = 0x100,
    HI_UNF_ENC_FMT_3840X2160_25,
    HI_UNF_ENC_FMT_3840X2160_30,
    HI_UNF_ENC_FMT_4096X2160_24,

    HI_UNF_ENC_FMT_3840X2160_23_976 = 0x200,
    HI_UNF_ENC_FMT_3840X2160_29_97,
    HI_UNF_ENC_FMT_720P_59_94,
    HI_UNF_ENC_FMT_1080P_59_94,
    HI_UNF_ENC_FMT_1080P_29_97,
    HI_UNF_ENC_FMT_1080P_23_976,
    HI_UNF_ENC_FMT_1080i_59_94,
    HI_UNF_ENC_FMT_BUTT
}HI_UNF_ENC_FMT_E;
```

【修改原因】

新增刷新率为 23.976、29.97、59.94 的显示制式

【注意事项】

无

【范例】

无

8.2 Common

8.2.1 概述

UNF3.2.6 相对于 UNF3.2.5 版本，Common 模块在总体功能上有以下变更：

扩展获取芯片支持功能接口

8.2.1.1 获取芯片支持功能

因为扩展获取芯片支持功能接口，增加 DTS、ADVCA、Macrovision 支持判断及芯片唯一 ID 获取功能，对如下数据结构进行了变更：

数据结构

修改 HI_hiSYS_CHIP_ATTR_S

8.2.2 数据结构

8.2.2.1 修改

HI_hiSYS_CHIP_ATTR_S

【结构体定义】

修改前：

```
typedef struct hiSYS_CHIP_ATTR_S
{
    HI_BOOL bDolbySupport;
}HI_SYS_CHIP_ATTR_S;
```

修改后：

```
typedef struct hiSYS_CHIP_ATTR_S
{
    HI_BOOL bDolbySupport;
    HI_BOOL bDTSSupport;
    HI_BOOL bADVCASupport;
    HI_BOOL bMacrovisionSupport;
    HI_U64  u64ChipID;
}HI_SYS_CHIP_ATTR_S;
```

【修改说明】

扩展获取芯片支持功能接口，扩展相关结构。

【注意事项】

无

【范例】

无

8.3 HiPlayer

8.3.1 概述

UNF 3.2.6 相对于 UNF 3.2.5 版本，HiPlayer 模块在总体功能上有以下变更：

- 支持 DTS_EXPRESS 音频格式
- Mpegts 网络播放支持 position seek 模式
- Invoke 接口扩展
- 增加文件信息更新事件上报接口
- 支持私有数据消息上报
- 支持 WINDOW 操作句柄获取
- 增加设置 VDEC 输出 buffer 对齐参数

8.3.1.1 支持 DTS_EXPRESS 音频格式

该特性实现以下功能：

支持 DTS EXPRESS 音频格式。

数据结构

修改 HI_FORMAT_AUDIO_TYPE_E

8.3.1.2 Mpegts 网络播放支持 position seek 模式

该特性实现以下功能：

当播放网络 mpegts 格式时，支持 position seek 模式。

数据结构

新增 HI_FORMAT_SEEK_MODE_E

8.3.1.3 Invoke 接口扩展

该特性实现以下功能：

- 获取视频 rating 信息
- 获取音频 rating 信息
- 获取字幕 rating 信息
- 设置网速统计的采样时间间隔
- 获取在播放当前 URL 时 demux 是否支持 seek
- 设置音频 track 的命令
- 设置 seek 模式

数据结构

修改 HI_FORMAT_INVOKE_ID_E

8.3.1.4 文件信息更新事件上报

该特性实现以下功能：

增加文件信息更新事件上报。

数据结构

- 修改 HI_FORMAT_MSG_TYPE_E
- 修改 HI_SVR_PLAYER_EVENT_E

8.3.1.5 支持私有数据消息上报

该特性实现以下功能：

支持私有数据上报。

数据结构

- 修改 HI_FORMAT_MSG_S
- 修改 HI_SVR_PLAYER_EVENT_E

8.3.1.6 支持 WINDOW 操作句柄获取

该特性实现以下功能：

支持 WINDOW 操作句柄获取

数据结构

修改 HI_SVR_PLAYER_PARAM_S

8.3.1.7 增加设置 VDEC 输出 buffer 对齐参数

该特性实现以下功能：

设置外 buffer 物理对齐参数

数据结构

修改 HI_SVR_PICTURE_S

8.3.2 数据结构

8.3.2.1 新增

HI_FORMAT_SEEK_MODE_E

【枚举定义】

```
typedef enum hiFORMAT_SEEK_MODE_E
{
    HI_FORMAT_SEEK_MODE_PTS = 0x0,
    HI_FORMAT_SEEK_MODE_POS,
    HI_FORMAT_SEEK_MODE_BUTT
} HI_FORMAT_SEEK_MODE_E;
```

【新增原因】

当网络播放无索引文件时需要按 POS 做 seek。

【注意事项】

POS seek 是模糊 seek，seek 不准确。只有当网络播放无索引文件时才按 POS 做 seek。默认使用 PTS seek。

【范例】

```
HI_FORMAT_SEEK_MODE_E eMode = HI_FORMAT_SEEK_MODE_POS;
HI_SVR_PLAYER_Invoke(hPlayer,  HI_FORMAT_INVOKE_SET_SEEK_MODE, &eMode);
```

HI_FORMAT_STREAM_INFO_S

【结构体定义】

```
typedef struct hiFORMAT_STREAM_INFO_S
{
    HI_S32      s32ProgID;
    HI_S32      s32VidID;
    HI_S32      s32AudID;
    HI_S32      s32SubID;
    HI_S64      s64PlayTime;
} HI_FORMAT_STREAM_INFO_S;
```

【新增原因】

解析器获取播放器信息。

【注意事项】

Hiplayer 内部数据结构，目前此功能尚未正式使用。

【范例】

无

HI_SVR_PLAYER_PROC_SEEKINFO_S

【结构体定义】

```
typedef struct hiSVR_PLAYER_PROC_SEEKINFO_S
{
    HI_U32      u32DoReadSeek;
```

```
    HI_U32      u32ReadSeekDone;
    HI_U32      u32DoSeekFrameBinary;
    HI_U32      u32SeekFrameBinaryDone;
    HI_U32      u32DoSeekFrameGeneric;
    HI_U32      u32SeekFrameGenericDone;
    HI_U32      u32DoInitInput;
    HI_U32      u32DoAvioOpenH;
    HI_U32      u32AvioOpenHDone;
    HI_U32      u32DoAvProbeInputBuffer;
    HI_U32      u32AvProbeInputBufferDone;

    HI_U32      u32DoHiSvrFormatSeekPts;
    HI_U32      u32CmdSeek;
    HI_U32      u32DoAvformatOpenInput;
    HI_U32      u32AvformatOpenInputDone;
    HI_U32      u32DoSvrFormatFindStream;
    HI_U32      u32SvrFormatFindStreamDone;
    HI_U32      u32DoSvrFormatGetFileInfo;
    HI_U32      u32SvrFormatGetFileInfoDone;

} HI_SVR_PLAYER_PROC_SEEKINFO_S;
```

【新增原因】

配合 invoke 接口修改，获取 seek 时对应的 proc 信息。

【注意事项】

无

【范例】

无

HI_SVR_PLAYER_PROC_SWITCHPG_INFOTYPE_E

【枚举定义】

```
typedef enum hiSVR_PLAYER_PROC_SWITCHPG_INFOTYPE_E
{
    HI_SVR_PLAYER_PROC_DO_STOP=1,
    HI_SVR_PLAYER_PROC_HIMEDIAPLAYER_CONSTRUCT,
    HI_SVR_PLAYER_PROC_SETDATASOURCE,
    HI_SVR_PLAYER_PROC_UNF_AVPLAY_CREATE,
    HI_SVR_PLAYER_PROC_DO_PREPARE,
    HI_SVR_PLAYER_PROC_PREPARE_ASYNC_COMPLETE,
    HI_SVR_PLAYER_PROC_DO_START_ENTER,
    HI_SVR_PLAYER_PROC_PLAYER_STATE_PLAY,
```

```
    HI_SVR_PLAYER_PROC_MEDIA_INFO_FIRST_FRAME_TIME,
    HI_SVR_PLAYER_PROC_DO_RESET,
    HI_SVR_PLAYER_PROC_DO_DESTRUCTOR,

} HI_SVR_PLAYER_PROC_SWITCHPG_INFOTYPE_E;
```

【新增原因】

配合 invoke 接口修改，切台动作枚举。

【注意事项】

无

【范例】

无

HI_SVR_PLAYER_PROC_SWITCHPG_S

【结构体定义】

```
typedef struct hiSVR_PLAYER_PROC_SWITCHPG_S
{
    HI_SVR_PLAYER_PROC_SWITCHPG_INFOTYPE_E eType;
    HI_U32 u32DoStop;
    HI_U32 u32HiMediaPlayerConstruct;
    HI_U32 u32SetDataSource;
    HI_U32 u32DoCreateAVPlay;
    HI_U32 u32DoPrepare;
    HI_U32 u32prepareAsyncComplete;
    HI_U32 u32DoStartEnter;
    HI_U32 u32PlayedEvent;
    HI_U32 u32FirstFrameTime;
    HI_U32 u32DoReset;
    HI_U32 u32DoDestructor;

} HI_SVR_PLAYER_PROC_SWITCHPG_S;
```

【新增原因】

配合 invoke 接口修改，获取切台信息。

【注意事项】

无

【范例】

无

8.3.2.2 修改

HI_FORMAT_AUDIO_TYPE_E

【枚举定义】

修改前：

```
typedef enum hiFORMAT_AUDIO_TYPE_E
{
HI_FORMAT_AUDIO_MP2 = 0x000,
……
    HI_FORMAT_AUDIO_BINKAUDIO_RDFT,
    HI_FORMAT_AUDIO_BINKAUDIO_DCT,
    HI_FORMAT_AUDIO_DRA,

    HI_FORMAT_AUDIO_PCM = 0x100,
    HI_FORMAT_AUDIO_PCM_BLURAY = 0x121,
    HI_FORMAT_AUDIO_ADPCM = 0x130,
……
} HI_FORMAT_AUDIO_TYPE_E;
```

修改后：

```
typedef enum hiFORMAT_AUDIO_TYPE_E
{
HI_FORMAT_AUDIO_MP2 = 0x000,
……
    HI_FORMAT_AUDIO_BINKAUDIO_RDFT,
    HI_FORMAT_AUDIO_BINKAUDIO_DCT,
    HI_FORMAT_AUDIO_DRA,
    HI_FORMAT_AUDIO_DTS_EXPRESS,

    HI_FORMAT_AUDIO_PCM = 0x100,
    HI_FORMAT_AUDIO_PCM_BLURAY = 0x121,
    HI_FORMAT_AUDIO_ADPCM = 0x130,
……
} HI_FORMAT_AUDIO_TYPE_E;
```

【修改原因】

支持 DTS EXPRESS 音频格式。

【注意事项】

DTS EXPRESS 进行格式转换的时候会转换为 DTS 格式。

【范例】

```
    case HI_FORMAT_AUDIO_DTS_EXPRESS: \
```

```
      (dest) = FORMAT_DTS;
```

HI_FORMAT_INVOKE_ID_E

【枚举定义】

修改前：

```
typedef enum hiFORMAT_INVOKE_ID_E
{
    ......
    HI_FORMAT_INVOKE_SET_LOCALTIME,
    HI_FORMAT_INVOKE_SET_HEADERS,
    HI_FORMAT_INVOKE_SET_USERAGENT,
    HI_FORMAT_INVOKE_SET_REFERER,
    HI_FORMAT_INVOKE_SET_NOT_SUPPORT_BYTERANGE,
    HI_FORMAT_INVOKE_SET_LOG_LEVEL,
    HI_FORMAT_INVOKE_RTMP_RECEIVEAUDIO,
    HI_FORMAT_INVOKE_RTMP_RECEIVEVIDEO,
    HI_FORMAT_INVOKE_SET_BUFFER_UNDERRUN,
    HI_FORMAT_INVOKE_SET_HLS_LIVE_START_NUM,
    HI_FORMAT_INVOKE_SET_STREAM_INFO,
    HI_FORMAT_INVOKE_PROTOCOL_USER=100,
    HI_FORMAT_INVOKE_SET_DOLBYRANGEINFO,
    HI_FORMAT_INVOKE_GET_DOLBYINFO,
    HI_FORMAT_INVOKE_SET_DOLBYDRCMODE,
    HI_FORMAT_INVOKE_SET_EXTERNALFORMAT,
    HI_FORMAT_INVOKE_FIND_BESTSTREAM,
    HI_FORMAT_INVOKE_BUTT
} HI_FORMAT_INVOKE_ID_E;
```

修改后：

```
typedef enum hiFORMAT_INVOKE_ID_E
{
    ......
    HI_FORMAT_INVOKE_SET_LOCALTIME,
    HI_FORMAT_INVOKE_SET_HEADERS,
    HI_FORMAT_INVOKE_SET_USERAGENT,
    HI_FORMAT_INVOKE_SET_REFERER,
    HI_FORMAT_INVOKE_SET_NOT_SUPPORT_BYTERANGE,
    HI_FORMAT_INVOKE_SET_LOG_LEVEL,
    HI_FORMAT_INVOKE_RTMP_RECEIVEAUDIO,
    HI_FORMAT_INVOKE_RTMP_RECEIVEVIDEO,
    HI_FORMAT_INVOKE_SET_BUFFER_UNDERRUN,
    HI_FORMAT_INVOKE_SET_HLS_LIVE_START_NUM,
```

```
    HI_FORMAT_INVOKE_SET_STREAM_INFO,
    HI_FORMAT_INVOKE_GET_VIDEO_RATING,
    HI_FORMAT_INVOKE_GET_AUDIO_RATING,
    HI_FORMAT_INVOKE_GET_SUBTITLE_RATING,
    HI_FORMAT_INVOKE_SET_BAND_COLLECT_FREQ_MS,
    HI_FORMAT_INVOKE_GET_SEEKABLE,
    HI_FORMAT_INVOKE_PROTOCOL_USER=100,
    HI_FORMAT_INVOKE_SET_DOLBYRANGEINFO,
    HI_FORMAT_INVOKE_GET_DOLBYINFO,
    HI_FORMAT_INVOKE_SET_DOLBYDRCMODE,
    HI_FORMAT_INVOKE_SET_EXTERNALFORMAT,
    HI_FORMAT_INVOKE_FIND_BESTSTREAM,
    HI_FORMAT_INVOKE_SET_EXTERNAL_AUDIOTRACK,
    HI_FORMAT_INVOKE_SET_SEEK_MODE,
    HI_FORMAT_INVOKE_BUTT
} HI_FORMAT_INVOKE_ID_E;
```

【修改原因】

更具需求和功能要求，增加以下 invoke 接口：

- 获取视频 rating 信息
- 获取音频 rating 信息
- 获取字幕 rating 信息
- 设置网速统计的采样时间间隔
- 获取在播放当前 URL 时 demux 是否支持 seek
- 设置音频 track 的命令
- 设置 seek 模式

【注意事项】

以下接口目前未启用：

- HI_FORMAT_INVOKE_SET_STREAM_INFO
- HI_FORMAT_INVOKE_GET_VIDEO_RATING
- HI_FORMAT_INVOKE_GET_AUDIO_RATING
- HI_FORMAT_INVOKE_GET_SUBTITLE_RATING
- HI_FORMAT_INVOKE_FIND_BESTSTREAM

【范例】

无

HI_FORMAT_MSG_TYPE_E

【结构体定义】

修改前：

```
typedef enum hiFORMAT_MSG_TYPE_E
{
    HI_FORMAT_MSG_NONE = 0x0,
    HI_FORMAT_MSG_UNKNOW,
    HI_FORMAT_MSG_DOWNLOAD_FINISH,
    HI_FORMAT_MSG_TIME_OUT,
    HI_FORMAT_MSG_NETWORK,
    HI_FORMAT_MSG_NOT_SUPPORT,
    HI_FORMAT_MSG_DISCONTINUITY_SEEK,
    HI_FORMAT_MSG_DISCONTINUITY_AUD_FORMAT,
    HI_FORMAT_MSG_DISCONTINUITY_VID_FORMAT,
    HI_FORMAT_MSG_USER_PRIVATE = 100,
    HI_FORMAT_MSG_BUTT,
} HI_FORMAT_MSG_TYPE_E;
```

修改后：

```
typedef enum hiFORMAT_MSG_TYPE_E
{
    HI_FORMAT_MSG_NONE = 0x0,
    HI_FORMAT_MSG_UNKNOW,
    HI_FORMAT_MSG_DOWNLOAD_FINISH,
    HI_FORMAT_MSG_TIME_OUT,
    HI_FORMAT_MSG_NETWORK,
    HI_FORMAT_MSG_NOT_SUPPORT,
    HI_FORMAT_MSG_DISCONTINUITY_SEEK,
    HI_FORMAT_MSG_DISCONTINUITY_AUD_FORMAT,
    HI_FORMAT_MSG_DISCONTINUITY_VID_FORMAT,
    HI_FORMAT_MSG_UPDATE_FILE_INFO,
    HI_FORMAT_MSG_USER_PRIVATE = 100,
    HI_FORMAT_MSG_BUTT,
} HI_FORMAT_MSG_TYPE_E;
```

【修改原因】

文件信息更新事件上报。

【注意事项】

未启用。

【范例】

无

HI_FORMAT_MSG_S

【结构体定义】

修改前：

```
typedef struct hiFORMAT_MSG_S
{
    HI_FORMAT_MSG_TYPE_E      eMsgType;
    HI_FORMAT_BUFFER_STATUS_S stBuffer;
    HI_FORMAT_NET_STATUS_S    stNetStatus;
} HI_FORMAT_MSG_S;
```

修改后：

```
typedef struct hiFORMAT_MSG_S
{
    HI_FORMAT_MSG_TYPE_E      eMsgType;
    HI_FORMAT_BUFFER_STATUS_S stBuffer;
    HI_FORMAT_NET_STATUS_S    stNetStatus;
    HI_VOID  *pPriData;
} HI_FORMAT_MSG_S;
```

【修改原因】

支持私有数据上报。

【注意事项】

私有信息参数地址，当上报消息为 HI_FORMAT_MSG_USER_PRIVATE，该值有效。

【范例】

```
(HI_VOID)_SVR_PCTRL_NotifyEvnt(pCtrl, bLock,
HI_SVR_PLAYER_EVENT_USER_PRIVATE,sizeof(stMsg.pPriData),
(HI_U8*)(stMsg.pPriData));
```

HI_SVR_PLAYER_EVENT_E

【枚举定义】

修改前：

```
typedef enum hiSVR_PLAYER_EVENT_E
{
    HI_SVR_PLAYER_EVENT_STATE_CHANGED = 0x0,
    HI_SVR_PLAYER_EVENT_SOF,

    HI_SVR_PLAYER_EVENT_EOF,
    HI_SVR_PLAYER_EVENT_PROGRESS,
    HI_SVR_PLAYER_EVENT_STREAMID_CHANGED,
    HI_SVR_PLAYER_EVENT_SEEK_FINISHED,

    HI_SVR_PLAYER_EVENT_CODETYPE_CHANGED,
```

```
    HI_SVR_PLAYER_EVENT_DOWNLOAD_PROGRESS,
    HI_SVR_PLAYER_EVENT_BUFFER_STATE,
    HI_SVR_PLAYER_EVENT_FIRST_FRAME_TIME,

    HI_SVR_PLAYER_EVENT_ERROR,
    HI_SVR_PLAYER_EVENT_NETWORK_INFO,
    HI_SVR_PLAYER_EVENT_DOWNLOAD_FINISH,
    HI_SVR_PLAYER_EVENT_BUTT
} HI_SVR_PLAYER_EVENT_E;
```

修改后：

```
typedef enum hiSVR_PLAYER_EVENT_E
{
    HI_SVR_PLAYER_EVENT_STATE_CHANGED = 0x0,
    HI_SVR_PLAYER_EVENT_SOF,

    HI_SVR_PLAYER_EVENT_EOF,
    HI_SVR_PLAYER_EVENT_PROGRESS,
    HI_SVR_PLAYER_EVENT_STREAMID_CHANGED,
    HI_SVR_PLAYER_EVENT_SEEK_FINISHED,

    HI_SVR_PLAYER_EVENT_CODETYPE_CHANGED,
    HI_SVR_PLAYER_EVENT_DOWNLOAD_PROGRESS,
    HI_SVR_PLAYER_EVENT_BUFFER_STATE,
    HI_SVR_PLAYER_EVENT_FIRST_FRAME_TIME,

    HI_SVR_PLAYER_EVENT_ERROR,
    HI_SVR_PLAYER_EVENT_NETWORK_INFO,
    HI_SVR_PLAYER_EVENT_DOWNLOAD_FINISH,
    HI_SVR_PLAYER_EVENT_UPDATE_FILE_INFO,
    HI_SVR_PLAYER_EVENT_USER_PRIVATE = 100,
    HI_SVR_PLAYER_EVENT_BUTT
} HI_SVR_PLAYER_EVENT_E;
```

【修改原因】

支持私有数据上报。

【注意事项】

HI_SVR_PLAYER_EVENT_UPDATE_FILE_INFO：

文件信息更新事件上报,未启用。

HI_SVR_PLAYER_EVENT_USER_PRIVATE：

私有信息参数地址，当上报消息为 HI_FORMAT_MSG_USER_PRIVATE，该值有效。

【范例】

```
(HI_VOID)_SVR_PCTRL_NotifyEvnt(pCtrl, bLock,
HI_SVR_PLAYER_EVENT_USER_PRIVATE,sizeof(stMsg.pPriData),
(HI_U8*)(stMsg.pPriData));
```

HI_SVR_PLAYER_PARAM_S

【结构体定义】

修改前：

```
typedef struct hiSVR_PLAYER_PARAM_S
{
    HI_U32  u32DmxId;
    HI_U32  u32PortId;
    HI_U32  x;
    HI_U32  y;
    HI_U32  w;
    HI_U32  h;
    HI_U32  u32MixHeight;
    HI_HANDLE hAVPlayer;
    HI_U32   u32SndPort;
    HI_U32   u32Display;
    HI_U32   u32VDecErrCover;
    HI_HANDLE hDRMClient;
} HI_SVR_PLAYER_PARAM_S;
```

修改后：

```
typedef struct hiSVR_PLAYER_PARAM_S
{
    HI_U32  u32DmxId;
    HI_U32  u32PortId;
    HI_U32  x;
    HI_U32  y;
    HI_U32  w;
    HI_U32  h;
    HI_U32  u32MixHeight;
    HI_HANDLE hAVPlayer;
    HI_HANDLE hVSink;
    HI_HANDLE hASink;
    HI_U32   u32SndPort;
    HI_U32   u32Display;
    HI_U32   u32VDecErrCover;
    HI_HANDLE hDRMClient;
    HI_HANDLE hWindow;
```

```
} HI_SVR_PLAYER_PARAM_S;
```

【修改原因】

HiPlayer 适配层需要对 WINDOW 进行控制，通过 hWindow 获取 WINDOW 操作句柄。

【注意事项】

本数据结构中 hVSink 和 hASink 只在 Android 版本上使用，linux 版本不支持。

【范例】

无

HI_SVR_PICTURE_S

【结构体定义】

修改前：

```
typedef struct hiSVR_PICTURE_S
{
    HI_U32              u32Width;
    HI_U32              u32Height;
    HI_S64              s64Pts;
    HI_HANDLE           hBuffer;
    HI_VOID*            priv;
} HI_SVR_PICTURE_S;
```

修改后：

```
typedef struct hiSVR_PICTURE_S
{
    HI_U32              u32Width;
    HI_U32              u32Height;
    HI_U32              u32Stride;
    HI_S64              s64Pts;
    HI_HANDLE           hBuffer;
    HI_VOID*            priv;
} HI_SVR_PICTURE_S;
```

【修改原因】

用于 VDEC 输出时，解码输出 buffer 的 STRIDE 和 GALLOC 分配的 GPU 读入 buffer 的 STRIDE 对齐。

【注意事项】

只在 Android 版本上使用，linux 版本不支持。

【范例】

```
HI_U32 u32Stride = xx;
```

```
HI_MPI_AVPLAY_UseExternalBuffer(pstMember->hAVPlay, hBuffers, cnt,
u32BufSize, u32Stride);
```

9 UNF3.2.7 与 UNF3.2.6 之间的差异

9.1 AVPLAY

9.1.1 概述

UNF 3.2.7 相对于 UNF 3.2.6 版本，AVPLAY 模块在总体功能上有以下变更：

增加视频事件回调函数类型

9.1.1.1 增加视频事件回调函数类型

该特性实现以下功能：

AVPLAY 上报设置视频协议类型错误事件，引起的变更如下：

数据结构

修改 HI_UNF_AVPLAY_EVENT_E

9.1.2 数据结构

9.1.2.1 修改

HI_UNF_AVPLAY_EVENT_E

【结构体定义】

修改前：

```
typedef enum hiUNF_AVPLAY_EVENT_E
{
    HI_UNF_AVPLAY_EVENT_EOS,
    HI_UNF_AVPLAY_EVENT_STOP,
    HI_UNF_AVPLAY_EVENT_RNG_BUF_STATE,
    HI_UNF_AVPLAY_EVENT_NORM_SWITCH,
```

```
    HI_UNF_AVPLAY_EVENT_FRAMEPACKING_CHANGE,
    HI_UNF_AVPLAY_EVENT_NEW_VID_FRAME,
    HI_UNF_AVPLAY_EVENT_NEW_AUD_FRAME,
    HI_UNF_AVPLAY_EVENT_NEW_USER_DATA,
    HI_UNF_AVPLAY_EVENT_GET_AUD_ES,
    HI_UNF_AVPLAY_EVENT_IFRAME_ERR,
    HI_UNF_AVPLAY_EVENT_SYNC_PTS_JUMP,
    HI_UNF_AVPLAY_EVENT_SYNC_STAT_CHANGE,
    HI_UNF_AVPLAY_EVENT_VID_BUF_STATE,
    HI_UNF_AVPLAY_EVENT_AUD_BUF_STATE,
    HI_UNF_AVPLAY_EVENT_VID_UNSUPPORT,
    HI_UNF_AVPLAY_EVENT_VID_ERR_RATIO,
    HI_UNF_AVPLAY_EVENT_AUD_INFO_CHANGE,
    HI_UNF_AVPLAY_EVENT_AUD_UNSUPPORT,
    HI_UNF_AVPLAY_EVENT_AUD_FRAME_ERR,
    HI_UNF_AVPLAY_EVENT_BUTT
} HI_UNF_AVPLAY_EVENT_E;
```

修改后：

```
typedef enum hiUNF_AVPLAY_EVENT_E
{
    HI_UNF_AVPLAY_EVENT_EOS,
    HI_UNF_AVPLAY_EVENT_STOP,
    HI_UNF_AVPLAY_EVENT_RNG_BUF_STATE,
    HI_UNF_AVPLAY_EVENT_NORM_SWITCH,
    HI_UNF_AVPLAY_EVENT_FRAMEPACKING_CHANGE,
    HI_UNF_AVPLAY_EVENT_NEW_VID_FRAME,
    HI_UNF_AVPLAY_EVENT_NEW_AUD_FRAME,
    HI_UNF_AVPLAY_EVENT_NEW_USER_DATA,
    HI_UNF_AVPLAY_EVENT_GET_AUD_ES,
    HI_UNF_AVPLAY_EVENT_IFRAME_ERR,
    HI_UNF_AVPLAY_EVENT_SYNC_PTS_JUMP,
    HI_UNF_AVPLAY_EVENT_SYNC_STAT_CHANGE,
    HI_UNF_AVPLAY_EVENT_VID_BUF_STATE,
```

```
    HI_UNF_AVPLAY_EVENT_AUD_BUF_STATE,
    HI_UNF_AVPLAY_EVENT_VID_UNSUPPORT,
    HI_UNF_AVPLAY_EVENT_VID_ERR_RATIO,
    HI_UNF_AVPLAY_EVENT_AUD_INFO_CHANGE,
    HI_UNF_AVPLAY_EVENT_AUD_UNSUPPORT,
    HI_UNF_AVPLAY_EVENT_AUD_FRAME_ERR,
    HI_UNF_AVPLAY_EVENT_VID_ERR_TYPE,
    HI_UNF_AVPLAY_EVENT_BUTT
} HI_UNF_AVPLAY_EVENT_E;
```

【修改原因】

AVPLAY 新增上报事件：设置视频协议类型错误事件

【注意事项】

无

【范例】

无

9.2 Frontend

9.2.1 概述

UNF 3.2.7 相对于 UNF 3.2.6 版本，Frontend 模块在总体功能上有以下变更：

- 增加 TDA182I5A tuner 支持
- ACM/VCM 功能 UNF 接口命名优化
- Unicable 用户频段盲扫时，增加事件上报机制，上报盲扫进度，盲扫状态
- DVB-S 锁频时增加物理层相位扰码的初始值配置
- DVB-T 增加超时时间

9.2.1.1 增加 TDA182I5A tuner 支持

增加 TDA182I5A tuner 支持，引起的变更如下：

数据结构

修改 HI_UNF_TUNER_DEV_TYPE_E

9.2.1.2 ACM/VCM 功能 UNF 接口命名优化

ACM/VCM 功能 UNF 接口命名优化，引起的变更如下：

数据结构

- 修改 HI_UNF_TUNER_SAT_SIGNALINFO_S
- 新增 HI_UNF_TUNER_CODE_MODULATION_E

9.2.1.3 Unicable 用户频段盲扫时，增加事件上报机制

Unicable 用户频段盲扫时，增加事件上报机制，引起的变更如下：

数据结构

- 新增 HI_UNF_TUNER_UNICABLE_SCAN_STATUS_E
- 新增 HI_UNF_TUNER_UNICABLE_SCAN_USER_BAND_EVT_E
- 新增 HI_UNF_TUNER_UNICABLE_SCAN_USER_BAND_NOTIFY_S
- 新增 HI_UNF_TUNER_UNICABLE_SCAN_PARA_S

函数接口

- 删除 HI_UNF_TUNER_GetAgc
- 删除 HI_UNF_UNICABLE_SetCurUB
- 删除 HI_UNF_UNICABLE_ScanAndInstall_UB
- 删除 HI_UNF_UNICABLE_GetUBInfo
- 新增 HI_UNF_TUNER_UNICABLE_ExitScanUserBands
- 新增 HI_UNF_TUNER_GetSatIsiID
- 新增 HI_UNF_TUNER_GetSatTotalStream
- 新增 HI_UNF_TUNER_SetSatIsiID

9.2.1.4 DVB-S 锁频时增加物理层相位扰码的初始值配置

DVB-S 锁频时增加物理层相位扰码的初始值配置，引起的变更如下：

数据结构

修改 HI_UNF_SAT_CONNECT_PARA_S

9.2.1.5 DVB-T 增加超时时间

DVB-T 增加超时时间，引起的变更如下：

数据结构

修改 HI_UNF_TUNER_TER_SCAN_PARA_S

9.2.2 数据结构

9.2.2.1 新增

HI_UNF_TUNER_CODE_MODULATION_E

【结构体定义】

```
typedef enum hiUNF_TUNER_CODE_MODULATION_E
{
    HI_UNF_TUNER_CODE_MODULATION_VCM_ACM,
    HI_UNF_TUNER_CODE_MODULATION_CCM,
    HI_UNF_TUNER_CODE_MODULATION_MULTISTREAM,
    HI_UNF_TUNER_CODE_MODULATION_BUTT
} HI_UNF_TUNER_CODE_MODULATION_E;
```

【新增原因】

描述卫星信号的编码和调制信息参数，作为 HI_UNF_TUNER_SAT_SIGNALINFO_S 结构体成员。

【注意事项】

无

【范例】

无

HI_UNF_TUNER_UNICABLE_SCAN_STATUS_E

【结构体定义】

```
typedef enum hiUNF_TUNER_UNICABLE_SCAN_STATUS_E
{
    HI_UNF_TUNER_UNICABLE_SCAN_STATUS_IDLE,
    HI_UNF_TUNER_UNICABLE_SCAN_STATUS_SCANNING,
    HI_UNF_TUNER_UNICABLE_SCAN_STATUS_FINISH,
    HI_UNF_TUNER_UNICABLE_SCAN_STATUS_QUIT,
    HI_UNF_TUNER_UNICABLE_SCAN_STATUS_FAIL,
    HI_UNF_TUNER_UNICABLE_SCAN_STATUS_BUTT
} HI_UNF_TUNER_UNICABLE_SCAN_STATUS_E;
```

【新增原因】

描述扫描的状态，如空闲，扫描中，扫描完成等。

【注意事项】

无

【范例】

无

HI_UNF_TUNER_UNICABLE_SCAN_USER_BAND_EVT_E

【结构体定义】

```
typedef enum hiUNF_TUNER_UNICABLE_SCAN_USER_BAND_EVT_E
{
    HI_UNF_TUNER_UNICABLE_SCAN_EVT_STATUS,
    HI_UNF_TUNER_UNICABLE_SCAN_EVT_PROGRESS,
    HI_UNF_TUNER_UNICABLE_SCAN_EVT_BUTT
} HI_UNF_TUNER_UNICABLE_SCAN_USER_BAND_EVT_E;
```

【新增原因】

描述上报的事件，是状态变化还是扫描进度变化。

【注意事项】

无

【范例】

无

HI_UNF_TUNER_UNICABLE_SCAN_USER_BAND_NOTIFY_S

【结构体定义】

```
typedef union hiUNF_TUNER_UNICABLE_SCAN_USER_BAND_NOTIFY_S
{
    HI_UNF_TUNER_UNICABLE_SCAN_STATUS_E* penStatus;
    HI_U16*    pu16ProgressPercent;
} HI_UNF_TUNER_UNICABLE_SCAN_USER_BAND_NOTIFY_S;
```

【新增原因】

描述扫描状态和扫描进度百分比。

【注意事项】

无

【范例】

无

HI_UNF_TUNER_UNICABLE_SCAN_PARA_S

【结构体定义】

```
typedef struct hiUNF_TUNER_UNICABLE_SCAN_PARA_S
{
    HI_VOID (*pfnEVTNotify)(HI_U32 u32TunerId,
    HI_UNF_TUNER_UNICABLE_SCAN_USER_BAND_EVT_E enEVT,
    HI_UNF_TUNER_UNICABLE_SCAN_USER_BAND_NOTIFY_S *pNotify);
} HI_UNF_TUNER_UNICABLE_SCAN_PARA_S;
```

【新增原因】

描述上报事件的函数。

【注意事项】

无

【范例】

无

9.2.2.2 修改

HI_UNF_TUNER_DEV_TYPE_E

【结构体定义】

修改前：

```
typedef enum    hiUNF_TUNER_DEV_TYPE_E
{
    HI_UNF_TUNER_DEV_TYPE_XG_3BL,
    HI_UNF_TUNER_DEV_TYPE_CD1616,
    HI_UNF_TUNER_DEV_TYPE_ALPS_TDAE,
    HI_UNF_TUNER_DEV_TYPE_TDCC,
    HI_UNF_TUNER_DEV_TYPE_TDA18250,
    HI_UNF_TUNER_DEV_TYPE_CD1616_DOUBLE,
    HI_UNF_TUNER_DEV_TYPE_MT2081,
    HI_UNF_TUNER_DEV_TYPE_TMX7070X,
    HI_UNF_TUNER_DEV_TYPE_R820C,
    HI_UNF_TUNER_DEV_TYPE_MXL203,
    HI_UNF_TUNER_DEV_TYPE_AV2011,
    HI_UNF_TUNER_DEV_TYPE_SHARP7903,
```

```
    HI_UNF_TUNER_DEV_TYPE_MXL101,
    HI_UNF_TUNER_DEV_TYPE_MXL603,
    HI_UNF_TUNER_DEV_TYPE_IT9170,
    HI_UNF_TUNER_DEV_TYPE_IT9133,
    HI_UNF_TUNER_DEV_TYPE_TDA6651,
    HI_UNF_TUNER_DEV_TYPE_TDA18250B,
    HI_UNF_TUNER_DEV_TYPE_M88TS2022,
    HI_UNF_TUNER_DEV_TYPE_RDA5815,
    HI_UNF_TUNER_DEV_TYPE_MXL254,
    HI_UNF_TUNER_DEV_TYPE_CXD2861,
    HI_UNF_TUNER_DEV_TYPE_SI2147,
    HI_UNF_TUNER_DEV_TYPE_RAFAEL836,
    HI_UNF_TUNER_DEV_TYPE_MXL608,
    HI_UNF_TUNER_DEV_TYPE_MXL214,
    HI_UNF_TUNER_DEV_TYPE_TDA18280,
    HI_UNF_TUNER_DEV_TYPE_BUTT
} HI_UNF_TUNER_DEV_TYPE_E ;
```

修改后：

```
typedef enum    hiUNF_TUNER_DEV_TYPE_E
{
    HI_UNF_TUNER_DEV_TYPE_XG_3BL,
    HI_UNF_TUNER_DEV_TYPE_CD1616,
    HI_UNF_TUNER_DEV_TYPE_ALPS_TDAE,
    HI_UNF_TUNER_DEV_TYPE_TDCC,
    HI_UNF_TUNER_DEV_TYPE_TDA18250,
    HI_UNF_TUNER_DEV_TYPE_CD1616_DOUBLE,
    HI_UNF_TUNER_DEV_TYPE_MT2081,
    HI_UNF_TUNER_DEV_TYPE_TMX7070X,
    HI_UNF_TUNER_DEV_TYPE_R820C,
    HI_UNF_TUNER_DEV_TYPE_MXL203,
    HI_UNF_TUNER_DEV_TYPE_AV2011,
    HI_UNF_TUNER_DEV_TYPE_SHARP7903,
    HI_UNF_TUNER_DEV_TYPE_MXL101,
```

```
    HI_UNF_TUNER_DEV_TYPE_MXL603,
    HI_UNF_TUNER_DEV_TYPE_IT9170,
    HI_UNF_TUNER_DEV_TYPE_IT9133,
    HI_UNF_TUNER_DEV_TYPE_TDA6651,
    HI_UNF_TUNER_DEV_TYPE_TDA18250B,
    HI_UNF_TUNER_DEV_TYPE_M88TS2022,
    HI_UNF_TUNER_DEV_TYPE_RDA5815,
    HI_UNF_TUNER_DEV_TYPE_MXL254,
    HI_UNF_TUNER_DEV_TYPE_CXD2861,
    HI_UNF_TUNER_DEV_TYPE_SI2147,
    HI_UNF_TUNER_DEV_TYPE_RAFAEL836,
    HI_UNF_TUNER_DEV_TYPE_MXL608,
    HI_UNF_TUNER_DEV_TYPE_MXL214,
    HI_UNF_TUNER_DEV_TYPE_TDA18280,
    HI_UNF_TUNER_DEV_TYPE_TDA182I5A,
    HI_UNF_TUNER_DEV_TYPE_BUTT
} HI_UNF_TUNER_DEV_TYPE_E ;
```

【修改原因】

增加 TDA182I5A 支持。

【注意事项】

无

【范例】

无

HI_UNF_TUNER_SAT_SIGNALINFO_S

【结构体定义】

修改前：

```
typedef struct  hiUNF_TUNER_SAT_SIGNALINFO_S
{
    HI_U32                      u32Freq;
    HI_U32                      u32SymbolRate;
    HI_UNF_MODULATION_TYPE_E      enModType;
    HI_UNF_TUNER_FE_POLARIZATION_E enPolar;
```

```
    HI_UNF_TUNER_FE_FECTYPE_E      enSATType;
    HI_UNF_TUNER_FE_FECRATE_E      enFECRate;
} HI_UNF_TUNER_SAT_SIGNALINFO_S;
```

修改后：

```
typedef struct  hiUNF_TUNER_SAT_SIGNALINFO_S
{
    HI_U32                      u32Freq;
    HI_U32                      u32SymbolRate;
    HI_UNF_MODULATION_TYPE_E       enModType;
    HI_UNF_TUNER_FE_POLARIZATION_E enPolar;
    HI_UNF_TUNER_FE_FECTYPE_E      enSATType;
    HI_UNF_TUNER_FE_FECRATE_E      enFECRate;
    HI_UNF_TUNER_CODE_MODULATION_E enCodeModulation;
} HI_UNF_TUNER_SAT_SIGNALINFO_S;
```

【修改原因】

增加卫星信号的编码和调制信息参数。

【注意事项】

无

【范例】

无

HI_UNF_SAT_CONNECT_PARA_S

【结构体定义】

修改前：

```
typedef struct  hiUNF_SAT_CONNECT_PARA_S
{
    HI_U32    u32Freq;
    HI_U32    u32SymbolRate;
    HI_UNF_TUNER_FE_POLARIZATION_E enPolar;
  } HI_UNF_SAT_CONNECT_PARA_S;
```

修改后：

```
typedef struct  hiUNF_SAT_CONNECT_PARA_S
{
```

```
    HI_U32    u32Freq;
    HI_U32    u32SymbolRate;
    HI_UNF_TUNER_FE_POLARIZATION_E enPolar;
    HI_U32    u32ScrambleValue;
  } HI_UNF_SAT_CONNECT_PARA_S;
```

【修改原因】

锁频时增加物理层相位扰码的初始值配置。

【注意事项】

如果客户锁屏时不涉及扰码，必须将该值初始化为 0

【范例】

无

HI_UNF_TUNER_TER_SCAN_PARA_S

【结构体定义】

修改前：

```
typedef struct hiUNF_TUNER_TER_SCAN_PARA_S
{
    HI_UNF_TUNER_TER_SCAN_ATTR_S stTer;
    HI_UNF_TUNER_TER_CHANNEL_ATTR_S enChanArray[TER_MAX_TP];
    HI_U32 u32ChanNum;
}HI_UNF_TUNER_TER_SCAN_PARA_S;
```

修改后：

```
typedef struct hiUNF_TUNER_TER_SCAN_PARA_S
{
    HI_UNF_TUNER_TER_SCAN_ATTR_S stTer;
    HI_UNF_TUNER_TER_CHANNEL_ATTR_S enChanArray[TER_MAX_TP];
    HI_U32 u32ChanNum;
    HI_S32 s32TimeOut;
}HI_UNF_TUNER_TER_SCAN_PARA_S;
```

【修改原因】

增加超时时间设置。

【注意事项】

该值暂时未使用，后续扩展【范例】

无

9.2.3 函数接口

9.2.3.1 新增

HI_UNF_TUNER_UNICABLE_ScanUserBands

【接口定义】

```
HI_S32 HI_UNF_TUNER_UNICABLE_ScanUserBands(HI_U32 u32TunerId,
HI_UNF_TUNER_UNICABLE_SCAN_PARA_S stScanPara);
```

【新增原因】

扫描 950~2150 频率范围找出用户频段接口，替代原来的接口 HI_UNF_UNICABLE_ScanAndInstall_UB。

【注意事项】

无

【范例】

无

HI_UNF_TUNER_UNICABLE_ExitScanUserBands

【接口定义】

```
HI_S32 HI_UNF_TUNER_UNICABLE_ExitScanUserBands(HI_U32 u32TunerId);
```

【新增原因】

增加主动退出扫描用户频段接口。

【注意事项】

无

【范例】

无

HI_UNF_TUNER_UNICABLE_GetUserBandsInfo

【接口定义】

```
HI_S32 HI_UNF_TUNER_UNICABLE_GetUserBandsInfo(HI_U32 u32TunerId,
HI_UNF_TUNER_SCR_UB_S **ppUBInfo, HI_U32 *pu32Num);
```

【新增原因】

获取扫描到的所有用户频段信息，替代原来的接口 HI_UNF_UNICABLE_GetUBInfo。

【注意事项】

无

【范例】

无

HI_UNF_TUNER_GetSatTotalStream

【接口定义】

```
HI_S32 HI_UNF_TUNER_GetSatTotalStream(HI_U32 u32TunerId, HI_U8
*pu8TotalStream);
```

【新增原因】

当前端是 VCM 信号时，获取共有多少套流。

【注意事项】

无

【范例】

无

HI_UNF_TUNER_GetSatIsiID

【接口定义】

```
HI_S32 HI_UNF_TUNER_GetSatIsiID(HI_U32 u32TunerId, HI_U8 u8StreamIndex,
HI_U8 *pu8IsiID);
```

【新增原因】

当前端是 VCM 信号时，通过流序号获取得到流的 ID 号。

【注意事项】

无

【范例】

无

HI_UNF_TUNER_SetSatIsiID

【接口定义】

```
HI_S32 HI_UNF_TUNER_SetSatIsiID(HI_U32 u32TunerId, HI_U8 u8IsiID);
```

【新增原因】

当前端是 VCM 信号且存在多套流时，通过设置流 ID 来接收指定的 VCM 流。

【注意事项】

无

【范例】

无

9.2.3.2 删除

HI_UNF_TUNER_GetAgc

【接口定义】

```
HI_S32 HI_UNF_TUNER_GetAgc(HI_U32 u32TunerId,HI_S32 s32CenterFreq, HI_S32
*ps32Agc);
```

【删除原因】

用户不需要关心该参数。

【注意事项】

无

【范例】

无

HI_UNF_UNICABLE_SetCurUB

【接口定义】

```
HI_S32 HI_UNF_UNICABLE_SetCurUB(HI_U32 u32TunerId);
```

【删除原因】

该功能在 HI_UNF_TUNER_SetLNBConfig 已有。

【注意事项】

无

【范例】

无

HI_UNF_UNICABLE_ScanAndInstall_UB

【接口定义】

```
HI_S32 HI_UNF_UNICABLE_ScanAndInstall_UB(HI_U32 u32TunerId);
```

【删除原因】

由 HI_UNF_TUNER_UNICABLE_ScanUserBands 替代。

【注意事项】

无

【范例】

无

HI_UNF_UNICABLE_GetUBInfo

【接口定义】

```
HI_S32 HI_UNF_UNICABLE_GetUBInfo(HI_U32 u32TunerId, HI_UNF_TUNER_SCR_UB_S
*pUBInfo);
```

【删除原因】

由 HI_UNF_TUNER_UNICABLE_GetUserBandsInfo 替代。

【注意事项】

无

【范例】

无

10 UNF3.2.8 与 UNF3.2.7 之间的差异

10.1 DEMUX

10.1.1 概述

UNF3.2.8 相对于 UNF3.2.7 版本，DEMUX 模块在总体功能上有以下变更：

新增支持获取和设置解扰器属性

10.1.1.1 支持获取和设置解扰器属性

增加解扰器属性的获取和设置功能。引起的变更如下：

接口

- 新增 HI_UNF_DMX_GetDescramblerAttr
- 新增 HI_UNF_DMX_SetDescramblerAttr

10.1.2 函数接口

10.1.2.1 新增

HI_UNF_DMX_GetDescramblerAttr

【接口定义】

```
HI_S32 HI_UNF_DMX_GetDescramblerAttr(HI_HANDLE hKey,
HI_UNF_DMX_DESCRAMBLER_ATTR_S *pstAttr);
```

【新增原因】

支持动态获取解扰器的属性。

【注意事项】

无

【范例】

无

HI_UNF_DMX_SetDescramblerAttr

【接口定义】

```
HI_S32 HI_UNF_DMX_SetDescramblerAttr(HI_HANDLE hKey,
HI_UNF_DMX_DESCRAMBLER_ATTR_S *pstAttr)
```

【新增原因】

支持动态设置解扰器的属性。

【注意事项】

无

【范例】

无

10.2 ADVCA

10.2.1 概述

UNF 3.2.8 相对于 UNF 3.2.7 版本，ADVCA 模块在总体功能上有以下变更：

新增两个 Flash 启动类型的支持：SPI_NAND 和 SD。

10.2.1.1 新增两个 Flash 启动类型的支持

该特性引起的变更如下：

数据结构

修改 HI_UNF_ADVCA_FLASH_TYPE_E

10.2.2 数据结构

10.2.2.1 修改

HI_UNF_ADVCA_FLASH_TYPE_E

【结构体定义】

修改前：

```
typedef enum hiUNF_ADVCA_FLASH_TYPE_E
{
    HI_UNF_ADVCA_FLASH_TYPE_SPI = 0,
    HI_UNF_ADVCA_FLASH_TYPE_NAND,
```

```
    HI_UNF_ADVCA_FLASH_TYPE_NOR,
    HI_UNF_ADVCA_FLASH_TYPE_EMMC,
    HI_UNF_ADVCA_FLASH_TYPE_BUTT
}HI_UNF_ADVCA_FLASH_TYPE_E;
```

修改后：

```
typedef enum hiUNF_ADVCA_FLASH_TYPE_E
{
    HI_UNF_ADVCA_FLASH_TYPE_SPI = 0,
    HI_UNF_ADVCA_FLASH_TYPE_NAND,
    HI_UNF_ADVCA_FLASH_TYPE_NOR,
    HI_UNF_ADVCA_FLASH_TYPE_EMMC,
    HI_UNF_ADVCA_FLASH_TYPE_SPI_NAND,
    HI_UNF_ADVCA_FLASH_TYPE_SD,
    HI_UNF_ADVCA_FLASH_TYPE_BUTT
}HI_UNF_ADVCA_FLASH_TYPE_E;;
```

【新增原因】

新增 SPI_NAND 和 SD 两种 Flash 启动类型。

【注意事项】

无

【范例】

请参考：sample/advca/sample_ca_opensecboot.c 和 sample/advca/sample_ca_get_otp_fuse.c。

10.3 Frontend

10.3.1 概述

UNF 3.2.8 相对于 UNF 3.2.7 版本，Frontend 模块在总体功能上有以下变更：

马达转动步长由两档可配，增加至十档可配

10.3.1.1 马达转动由两档可配，增加至十档可配

马达转动由两档可配，增加至十档可配，引起的变更如下：

数据结构

修改 HI_UNF_TUNER_DISEQC_MOVE_TYPE_E

新增 HI_UNF_TUNER_DISEQC_RUN_S

函数接口

新增 HI_UNF_TUNER_DISEQC_RunStep

10.3.2 数据结构

10.3.2.1 新增

HI_UNF_TUNER_DISEQC_RUN_S

【结构体定义】

```
typedef struct hiUNF_TUNER_DISEQC_RUN_S
{
    HI_UNF_TUNER_DISEQC_LEVEL_E    enLevel;
    HI_UNF_TUNER_DISEQC_MOVE_DIR_E  enDir;
    HI_U32  u32RunningSteps;
} HI_UNF_TUNER_DISEQC_RUN_S;
```

【新增原因】

支持 HI_UNF_TUNER_DISEQC_RunStep 配置马达转到档位参数。

【注意事项】

无

【范例】

无

10.3.2.2 修改

HI_UNF_TUNER_DISEQC_MOVE_TYPE_E

【结构体定义】

修改前：

```
typedef enum hiUNF_TUNER_DISEQC_MOVE_TYPE_E
{
    HI_UNF_TUNER_DISEQC_MOVE_STEP_SLOW,
    HI_UNF_TUNER_DISEQC_MOVE_STEP_FAST,
    HI_UNF_TUNER_DISEQC_MOVE_CONTINUE,
    HI_UNF_TUNER_DISEQC_MOVE_TYPE_BUTT
} HI_UNF_TUNER_DISEQC_MOVE_TYPE_E;
```

修改后：

```
typedef enum hiUNF_TUNER_DISEQC_MOVE_TYPE_E
{
```

```
    HI_UNF_TUNER_DISEQC_MOVE_STEP_SLOW,
    HI_UNF_TUNER_DISEQC_MOVE_STEP_SLOW1,
    HI_UNF_TUNER_DISEQC_MOVE_STEP_SLOW2,
    HI_UNF_TUNER_DISEQC_MOVE_STEP_SLOW3,
    HI_UNF_TUNER_DISEQC_MOVE_STEP_SLOW4,
    HI_UNF_TUNER_DISEQC_MOVE_STEP_FAST,
    HI_UNF_TUNER_DISEQC_MOVE_STEP_FAST1,
    HI_UNF_TUNER_DISEQC_MOVE_STEP_FAST2,
    HI_UNF_TUNER_DISEQC_MOVE_STEP_FAST3,
    HI_UNF_TUNER_DISEQC_MOVE_STEP_FAST4,
    HI_UNF_TUNER_DISEQC_MOVE_CONTINUE,
    HI_UNF_TUNER_DISEQC_MOVE_TYPE_BUTT
} HI_UNF_TUNER_DISEQC_MOVE_TYPE_E;
```

【修改原因】

增加马达转动步长档数。

- HI_UNF_TUNER_DISEQC_MOVE_STEP_SLOW 每次转动 1 步
- HI_UNF_TUNER_DISEQC_MOVE_STEP_SLOW1 每次转动 2 步
- HI_UNF_TUNER_DISEQC_MOVE_STEP_SLOW2 每次转动 3 步
- HI_UNF_TUNER_DISEQC_MOVE_STEP_SLOW3 每次转动 4 步
- HI_UNF_TUNER_DISEQC_MOVE_STEP_SLOW4 每次转动 5 步
- HI_UNF_TUNER_DISEQC_MOVE_STEP_FAST 每次转动 6 步
- HI_UNF_TUNER_DISEQC_MOVE_STEP_FAST1 每次转动 7 步
- HI_UNF_TUNER_DISEQC_MOVE_STEP_FAST2 每次转动 8 步
- HI_UNF_TUNER_DISEQC_MOVE_STEP_FAST3 每次转动 9 步
- HI_UNF_TUNER_DISEQC_MOVE_STEP_FAST4 每次转动 10 步
- HI_UNF_TUNER_DISEQC_MOVE_CONTINUE 连续转动

【注意事项】

无

【范例】

无

10.3.3 函数接口

10.3.3.1 新增

HI_UNF_TUNER_DISEQC_RunStep

【接口定义】

```
HI_S32 HI_UNF_TUNER_DISEQC_RunStep(HI_U32 u32TunerId,
HI_UNF_TUNER_DISEQC_RUN_S* pstPara);
```

【新增原因】

设置马达转动档位。后续准备用此接口逐渐替换 HI_UNF_TUNER_DISEQC_Move，因为该函数更方便档数扩展。

【注意事项】

无

【范例】

无

11 UNF3.2.9 与 UNF3.2.8 之间的差异

11.1 DEMUX

11.1.1 概述

UNF3.2.9 相对于 UNF3.2.8 版本，DEMUX 模块在总体功能上有以下变更：

- 新增同步获取索引和录制数据功能
- 新增获取 Tag 口数目功能
- 扩展端口定义

11.1.1.1 支持同步获取索引和录制数据功能

数据结构

- 新增 DMX_MAX_IDX_ACQUIRED_EACH_TIME
- 新增 HI_UNF_DMX_REC_DATA_INDEX_S

接口

- 新增 HI_UNF_DMX_AcquireRecDataAndIndex
- 新增 HI_UNF_DMX_ReleaseRecDataAndIndex

11.1.1.2 支持获取 Tag 口数目功能

数据结构

修改 HI_UNF_DMX_CAPABILITY_S

11.1.1.3 扩展端口定义

修改 HI_UNF_DMX_PORT_E

11.1.2 数据结构

11.1.2.1 新增

DMX_MAX_IDX_ACQUIRED_EACH_TIME

【结构体定义】

```
#define DMX_MAX_IDX_ACQUIRED_EACH_TIME  256
```

【新增原因】

定义 HI_UNF_DMX_AcquireRecDataAndIndex 单次获取索引的最大个数

【注意事项】

无

【范例】

无

HI_UNF_DMX_REC_DATA_INDEX_S

【结构体定义】

```
typedef struct hiUNF_DMX_REC_DATA_INDEX_S
{
    HI_U32 u32IdxNum;
    HI_U32 u32RecDataCnt;
    HI_UNF_DMX_REC_INDEX_S stIndex[DMX_MAX_IDX_ACQUIRED_EACH_TIME];
    HI_UNF_DMX_REC_DATA_S  stRecData[2];
} HI_UNF_DMX_REC_DATA_INDEX_S;
```

【新增原因】

定义 HI_UNF_DMX_AcquireRecDataAndIndex 获取的索引以及录制数据信息

【注意事项】

无

【范例】

无

11.1.2.2 修改

HI_UNF_DMX_CAPABILITY_S

【结构体定义】

修改前：

```
typedef struct hiUNF_DMX_CAPABILITY_S
{
    HI_U32 u32IFPortNum;
    HI_U32 u32TSIPortNum;
    HI_U32 u32TSOPortNum;
    HI_U32 u32RamPortNum;
    HI_U32 u32DmxNum;
    HI_U32 u32ChannelNum;
    HI_U32 u32AVChannelNum;
    HI_U32 u32FilterNum;
    HI_U32 u32KeyNum;
    HI_U32 u32RecChnNum;
} HI_UNF_DMX_CAPABILITY_S;
```

修改后：

```
typedef struct hiUNF_DMX_CAPABILITY_S
{
    HI_U32 u32IFPortNum;
    HI_U32 u32TSIPortNum;
    HI_U32 u32TSOPortNum;
    HI_U32 u32RamPortNum;
    HI_U32 u32DmxNum;
    HI_U32 u32ChannelNum;
    HI_U32 u32AVChannelNum;
    HI_U32 u32FilterNum;
    HI_U32 u32KeyNum;
    HI_U32 u32RecChnNum;
    HI_U32 u32TagPortNum;
} HI_UNF_DMX_CAPABILITY_S;
```

【修改原因】

新增域 u32TagPortNum，通过 HI_UNF_DMX_GetCapability 获取 Tag 口的数目。

【注意事项】

无

【范例】

无

HI_UNF_DMX_PORT_E

【结构体定义】

修改前：

```
typedef enum hiUNF_DMX_PORT_E
{
    HI_UNF_DMX_PORT_IF_0 = 0x0,
    HI_UNF_DMX_PORT_IF_1,
    HI_UNF_DMX_PORT_IF_2,
    HI_UNF_DMX_PORT_IF_3,
    HI_UNF_DMX_PORT_IF_4,
    HI_UNF_DMX_PORT_IF_5,
    HI_UNF_DMX_PORT_IF_6,
    HI_UNF_DMX_PORT_IF_7,
    HI_UNF_DMX_PORT_TSI_0 = 0x20,
    HI_UNF_DMX_PORT_TSI_1,
    HI_UNF_DMX_PORT_TSI_2,
    HI_UNF_DMX_PORT_TSI_3,
    HI_UNF_DMX_PORT_TSI_4,
    HI_UNF_DMX_PORT_TSI_5,
    HI_UNF_DMX_PORT_TSI_6,
    HI_UNF_DMX_PORT_TSI_7,
    HI_UNF_DMX_PORT_RAM_0 = 0x80,
    HI_UNF_DMX_PORT_RAM_1,
    HI_UNF_DMX_PORT_RAM_2,
    HI_UNF_DMX_PORT_RAM_3,
    HI_UNF_DMX_PORT_RAM_4,
    HI_UNF_DMX_PORT_RAM_5,
    HI_UNF_DMX_PORT_RAM_6,
    HI_UNF_DMX_PORT_RAM_7,
    HI_UNF_DMX_PORT_BUTT
} HI_UNF_DMX_PORT_E;
```

修改后：

```
typedef enum hiUNF_DMX_PORT_E
{
    HI_UNF_DMX_PORT_IF_0 = 0x0,
    HI_UNF_DMX_PORT_IF_1,
```

```
    HI_UNF_DMX_PORT_IF_2,
    HI_UNF_DMX_PORT_IF_3,
    HI_UNF_DMX_PORT_IF_4,
    HI_UNF_DMX_PORT_IF_5,
    HI_UNF_DMX_PORT_IF_6,
    HI_UNF_DMX_PORT_IF_7,
    HI_UNF_DMX_PORT_IF_8,
    HI_UNF_DMX_PORT_IF_9,
    HI_UNF_DMX_PORT_IF_10,
    HI_UNF_DMX_PORT_IF_11,
    HI_UNF_DMX_PORT_IF_12,
    HI_UNF_DMX_PORT_IF_13,
    HI_UNF_DMX_PORT_IF_14,
    HI_UNF_DMX_PORT_IF_15,
    HI_UNF_DMX_PORT_TSI_0 = 0x20,
    HI_UNF_DMX_PORT_TSI_1,
    HI_UNF_DMX_PORT_TSI_2,
    HI_UNF_DMX_PORT_TSI_3,
    HI_UNF_DMX_PORT_TSI_4,
    HI_UNF_DMX_PORT_TSI_5,
    HI_UNF_DMX_PORT_TSI_6,
    HI_UNF_DMX_PORT_TSI_7,
    HI_UNF_DMX_PORT_TSI_8,
    HI_UNF_DMX_PORT_TSI_9,
    HI_UNF_DMX_PORT_TSI_10,
    HI_UNF_DMX_PORT_TSI_11,
    HI_UNF_DMX_PORT_TSI_12,
    HI_UNF_DMX_PORT_TSI_13,
    HI_UNF_DMX_PORT_TSI_14,
    HI_UNF_DMX_PORT_TSI_15,
    HI_UNF_DMX_PORT_RAM_0 = 0x80,
    HI_UNF_DMX_PORT_RAM_1,
    HI_UNF_DMX_PORT_RAM_2,
    HI_UNF_DMX_PORT_RAM_3,
    HI_UNF_DMX_PORT_RAM_4,
    HI_UNF_DMX_PORT_RAM_5,
    HI_UNF_DMX_PORT_RAM_6,
```

```
    HI_UNF_DMX_PORT_RAM_7,
    HI_UNF_DMX_PORT_RAM_8,
    HI_UNF_DMX_PORT_RAM_9,
    HI_UNF_DMX_PORT_RAM_10,
    HI_UNF_DMX_PORT_RAM_11,
    HI_UNF_DMX_PORT_RAM_12,
    HI_UNF_DMX_PORT_RAM_13,
    HI_UNF_DMX_PORT_RAM_14,
    HI_UNF_DMX_PORT_RAM_15,
    HI_UNF_DMX_PORT_BUTT
} HI_UNF_DMX_PORT_E;
```

【修改原因】

扩展 IF、TSI、RAM 端口个数。

【注意事项】

无

【范例】

无

11.1.3 函数接口

11.1.3.1 新增

HI_UNF_DMX_AcquireRecDataAndIndex

【接口定义】

```
HI_S32  HI_UNF_DMX_AcquireRecDataAndIndex(HI_HANDLE hRecChn,
HI_UNF_DMX_REC_DATA_INDEX_S* pstRecDataIdx);
```

【新增原因】

支持同步获取录制数据和索引数据。

【注意事项】

无

【范例】

无

HI_UNF_DMX_ReleaseRecDataAndIndex

【接口定义】

```
HI_S32  HI_UNF_DMX_ReleaseRecDataAndIndex(HI_HANDLE hRecChn,
HI_UNF_DMX_REC_DATA_INDEX_S* pstRecDataIdx);
```

【新增原因】

支持同步释放录制数据和索引数据。

【注意事项】

无

【范例】

无

11.2 SOUND

11.2.1 概述

UNF3.2.9 相对于 UNF3.2.8，SOUND 在总体功能上有以下变更：

新增卡拉 OK 功能

11.2.1.1 新增卡拉 OK 功能

实现卡拉 OK 低延时方案，修改 Track Type 结构体、新增设置获取延时接口。

数据结构

修改 HI_UNF_SND_TRACK_TYPE_E

函数接口

- 新增 HI_UNF_SND_SetLowLatency
- 新增 HI_UNF_SND_GetLowLatency

11.2.2 数据结构

11.2.2.1 修改

HI_UNF_SND_TRACK_TYPE_E

【结构体定义】

修改前：

```
typedef enum hiHI_UNF_SND_TRACK_TYPE_E
{
    HI_UNF_SND_TRACK_TYPE_MASTER = 0,
    HI_UNF_SND_TRACK_TYPE_SLAVE,
    HI_UNF_SND_TRACK_TYPE_VIRTUAL,
```

```
    HI_UNF_SND_TRACK_TYPE_BUTT
} HI_UNF_SND_TRACK_TYPE_E;
```

修改后：

```
typedef enum hiHI_UNF_SND_TRACK_TYPE_E
{
    HI_UNF_SND_TRACK_TYPE_MASTER = 0,
    HI_UNF_SND_TRACK_TYPE_SLAVE,
    HI_UNF_SND_TRACK_TYPE_VIRTUAL,
    HI_UNF_SND_TRACK_TYPE_LOWLATENCY,
    HI_UNF_SND_TRACK_TYPE_BUTT
} HI_UNF_SND_TRACK_TYPE_E;
```

【修改说明】

增加低延时 Track。

【注意事项】

低延时通道不支持 AVPLAY 绑定，此通道只支持采样率为 48K,双声道,16 位位宽的音频数据,且仅支持创建一个低延时通道。

【范例】

无

11.2.3 函数接口

11.2.3.1 新增

HI_UNF_SND_SetLowLatency

【接口定义】

```
HI_S32 HI_UNF_SND_SetLowLatency(HI_UNF_SND_E enSound,
HI_UNF_SND_OUTPUTPORT_E eOutPort, HI_U32 u32LatecnyMs);
```

【新增原因】

设置基于 Sound 的延时大小。

【注意事项】

eOutPort 为输出端口，可以根据需求进行设置。

u32LatecnyMs 建议范围 10 到 40。

【范例】

无。

HI_UNF_SND_GetLowLatency

【接口定义】

```
HI_S32 HI_UNF_SND_GetLowLatency(HI_UNF_SND_E enSound,
HI_UNF_SND_OUTPUTPORT_E eOutPort, HI_U32 *p32LatecnyMs);
```

【新增原因】

获取基于 Sound 的延时大小。

【注意事项】

`eOutPort` 为输出端口，可以根据需求进行设置。

【范例】

无

11.3 DISPLAY

11.3.1 概述

UNF 3.2.9 相对于 UNF 3.2.8 版本，DISPLAY 模块在总体功能上有以下变更：

新增同源模式下一个接口同时设置高、标清两通道的制式信息的接口

11.3.1.1 新增同时设置高、标清两通道制式信息

增加新接口同时设置高、标清两通道制式信息。用于优化切换制式。引起的变更如下：

数据结构

新增 HI_UNF_DISP_ISOGENY_ATTR_S

函数接口

新增 HI_UNF_DISP_SetIsogenyAttr

11.3.2 数据结构

11.3.2.1 新增

HI_UNF_DISP_ISOGENY_ATTR_S

【结构体定义】

```
typedef struct  hiUNF_DISP_ISOGENY_ATTR_S
{
    HI_UNF_DISP_E         enDisp;
```

```
    HI_UNF_ENC_FMT_E      enFormat;

}HI_UNF_DISP_ISOGENY_ATTR_S ;
```

【新增原因】

一个接口同时设置高、标清通道制式，可以达到缩短制式切换时间、优化制式切换显示效果的目的

【注意事项】

该结构体作为预留后续扩展。

【范例】

无

11.3.3 函数接口

11.3.3.1 新增

HI_UNF_DISP_SetIsogenyAttr

【接口定义】

```
HI_S32 HI_UNF_DISP_SetIsogenyAttr(const HI_UNF_DISP_ISOGENY_ATTR_S
*pstIsogeny, const HI_U32 u32ChannelNum);
```

【新增原因】

增加新接口同时设置高、标清两通道制式信息。用于优化切换制式：

- 单接口切换制式时间相对于高、标清分别设置制式缩短百 ms 级别；
- 减少切换制式时 hdmi 的闪烁情况。

【注意事项】

和原有切换制式接口 HI_UNF_DISP_SetFormat 不能同时使用。

【范例】

参考 HI_UNF_DISP_SetFormat 的使用方法。

11.4 VO

11.4.1 概述

UNF 3.2.9 相对于 UNF 3.2.8 版本，VO 模块在总体功能上有以下变更：

增加分辨率为 3840*2160 和 4096*2160 的显示制式

11.4.1.1 增加显示制式

增加分辨率为 3840*2160 和 4096*2160 的显示制式，引起的变更如下：

数据结构

修改 HI_UNF_ENC_FMT_E

11.4.2 数据结构

11.4.2.1 修改

HI_UNF_ENC_FMT_E

【结构体定义】

修改前：

```
typedef enum hiUNF_ENC_FMT_E
{
    HI_UNF_ENC_FMT_1080P_60 = 0,     /**<1080p 60 Hz*/
    HI_UNF_ENC_FMT_1080P_50,         /**<1080p 50 Hz*/
    HI_UNF_ENC_FMT_1080P_30,         /**<1080p 30 Hz*/
    HI_UNF_ENC_FMT_1080P_25,         /**<1080p 25 Hz*/
    HI_UNF_ENC_FMT_1080P_24,         /**<1080p 24 Hz*/

    HI_UNF_ENC_FMT_1080i_60,         /**<1080i 60 Hz*/
    HI_UNF_ENC_FMT_1080i_50,         /**<1080i 50 Hz*/

    HI_UNF_ENC_FMT_720P_60,          /**<720p 60 Hz*/
    HI_UNF_ENC_FMT_720P_50,          /**<720p 50 Hz */

    HI_UNF_ENC_FMT_576P_50,          /**<576p 50 Hz*/
    HI_UNF_ENC_FMT_480P_60,          /**<480p 60 Hz*/

    HI_UNF_ENC_FMT_PAL,             /* B D G H I PAL */
    HI_UNF_ENC_FMT_PAL_N,            /* (N)PAL       */
    HI_UNF_ENC_FMT_PAL_Nc,           /* (Nc)PAL      */

    HI_UNF_ENC_FMT_NTSC,            /* (M)NTSC      */
    HI_UNF_ENC_FMT_NTSC_J,           /* NTSC-J       */
    HI_UNF_ENC_FMT_NTSC_PAL_M,       /* (M)PAL       */

    HI_UNF_ENC_FMT_SECAM_SIN,        /**< SECAM_SIN*/
    HI_UNF_ENC_FMT_SECAM_COS,        /**< SECAM_COS*/

    HI_UNF_ENC_FMT_1080P_24_FRAME_PACKING,
    HI_UNF_ENC_FMT_720P_60_FRAME_PACKING,
    HI_UNF_ENC_FMT_720P_50_FRAME_PACKING,
```

```
    HI_UNF_ENC_FMT_861D_640X480_60,
    HI_UNF_ENC_FMT_VESA_800X600_60,
    HI_UNF_ENC_FMT_VESA_1024X768_60,
    HI_UNF_ENC_FMT_VESA_1280X720_60,
    HI_UNF_ENC_FMT_VESA_1280X800_60,
    HI_UNF_ENC_FMT_VESA_1280X1024_60,
    HI_UNF_ENC_FMT_VESA_1360X768_60,
    HI_UNF_ENC_FMT_VESA_1366X768_60,
    HI_UNF_ENC_FMT_VESA_1400X1050_60,
    HI_UNF_ENC_FMT_VESA_1440X900_60,
    HI_UNF_ENC_FMT_VESA_1440X900_60_RB,
    HI_UNF_ENC_FMT_VESA_1600X900_60_RB,
    HI_UNF_ENC_FMT_VESA_1600X1200_60,
    HI_UNF_ENC_FMT_VESA_1680X1050_60,
    HI_UNF_ENC_FMT_VESA_1680X1050_60_RB,
    HI_UNF_ENC_FMT_VESA_1920X1080_60,
    HI_UNF_ENC_FMT_VESA_1920X1200_60,
    HI_UNF_ENC_FMT_VESA_1920X1440_60,
    HI_UNF_ENC_FMT_VESA_2048X1152_60,
    HI_UNF_ENC_FMT_VESA_2560X1440_60_RB,
    HI_UNF_ENC_FMT_VESA_2560X1600_60_RB,

    HI_UNF_ENC_FMT_3840X2160_24 = 0x100,
    HI_UNF_ENC_FMT_3840X2160_25,
    HI_UNF_ENC_FMT_3840X2160_30,
    HI_UNF_ENC_FMT_4096X2160_24,

    HI_UNF_ENC_FMT_3840X2160_23_976,
    HI_UNF_ENC_FMT_3840X2160_29_97,
    HI_UNF_ENC_FMT_720P_59_94,
    HI_UNF_ENC_FMT_1080P_59_94,
    HI_UNF_ENC_FMT_1080P_29_97,
    HI_UNF_ENC_FMT_1080P_23_976,
    HI_UNF_ENC_FMT_1080i_59_94,
    HI_UNF_ENC_FMT_BUTT
}HI_UNF_ENC_FMT_E;
```

修改后：

```
typedef enum hiUNF_ENC_FMT_E
{
    HI_UNF_ENC_FMT_1080P_60 = 0,     /**<1080p 60 Hz*/
    HI_UNF_ENC_FMT_1080P_50,         /**<1080p 50 Hz*/
    HI_UNF_ENC_FMT_1080P_30,         /**<1080p 30 Hz*/
    HI_UNF_ENC_FMT_1080P_25,         /**<1080p 25 Hz*/
```

```
    HI_UNF_ENC_FMT_1080P_24,        /**<1080p 24 Hz*/

    HI_UNF_ENC_FMT_1080i_60,        /**<1080i 60 Hz*/
    HI_UNF_ENC_FMT_1080i_50,        /**<1080i 50 Hz*/

    HI_UNF_ENC_FMT_720P_60,         /**<720p 60 Hz*/
    HI_UNF_ENC_FMT_720P_50,         /**<720p 50 Hz */

    HI_UNF_ENC_FMT_576P_50,         /**<576p 50 Hz*/
    HI_UNF_ENC_FMT_480P_60,         /**<480p 60 Hz*/

    HI_UNF_ENC_FMT_PAL,            /* B D G H I PAL */
    HI_UNF_ENC_FMT_PAL_N,           /* (N)PAL       */
    HI_UNF_ENC_FMT_PAL_Nc,          /* (Nc)PAL      */

    HI_UNF_ENC_FMT_NTSC,           /* (M)NTSC      */
    HI_UNF_ENC_FMT_NTSC_J,          /* NTSC-J       */
    HI_UNF_ENC_FMT_NTSC_PAL_M,      /* (M)PAL       */

    HI_UNF_ENC_FMT_SECAM_SIN,       /**< SECAM_SIN*/
    HI_UNF_ENC_FMT_SECAM_COS,       /**< SECAM_COS*/

    HI_UNF_ENC_FMT_1080P_24_FRAME_PACKING,
    HI_UNF_ENC_FMT_720P_60_FRAME_PACKING,
    HI_UNF_ENC_FMT_720P_50_FRAME_PACKING,

    HI_UNF_ENC_FMT_861D_640X480_60,
    HI_UNF_ENC_FMT_VESA_800X600_60,
    HI_UNF_ENC_FMT_VESA_1024X768_60,
    HI_UNF_ENC_FMT_VESA_1280X720_60,
    HI_UNF_ENC_FMT_VESA_1280X800_60,
    HI_UNF_ENC_FMT_VESA_1280X1024_60,
    HI_UNF_ENC_FMT_VESA_1360X768_60,
    HI_UNF_ENC_FMT_VESA_1366X768_60,
    HI_UNF_ENC_FMT_VESA_1400X1050_60,
    HI_UNF_ENC_FMT_VESA_1440X900_60,
    HI_UNF_ENC_FMT_VESA_1440X900_60_RB,
    HI_UNF_ENC_FMT_VESA_1600X900_60_RB,
    HI_UNF_ENC_FMT_VESA_1600X1200_60,
    HI_UNF_ENC_FMT_VESA_1680X1050_60,
    HI_UNF_ENC_FMT_VESA_1680X1050_60_RB,
    HI_UNF_ENC_FMT_VESA_1920X1080_60,
    HI_UNF_ENC_FMT_VESA_1920X1200_60,
    HI_UNF_ENC_FMT_VESA_1920X1440_60,
```

```
    HI_UNF_ENC_FMT_VESA_2048X1152_60,
    HI_UNF_ENC_FMT_VESA_2560X1440_60_RB,
    HI_UNF_ENC_FMT_VESA_2560X1600_60_RB,

    HI_UNF_ENC_FMT_3840X2160_24 = 0x100,
    HI_UNF_ENC_FMT_3840X2160_25,
    HI_UNF_ENC_FMT_3840X2160_30,
    HI_UNF_ENC_FMT_3840X2160_50,
    HI_UNF_ENC_FMT_3840X2160_60,

    HI_UNF_ENC_FMT_4096X2160_24,
    HI_UNF_ENC_FMT_4096X2160_25,
    HI_UNF_ENC_FMT_4096X2160_30,
    HI_UNF_ENC_FMT_4096X2160_50,
    HI_UNF_ENC_FMT_4096X2160_60,

    HI_UNF_ENC_FMT_3840X2160_23_976,
    HI_UNF_ENC_FMT_3840X2160_29_97,
    HI_UNF_ENC_FMT_720P_59_94,
    HI_UNF_ENC_FMT_1080P_59_94,
    HI_UNF_ENC_FMT_1080P_29_97,
    HI_UNF_ENC_FMT_1080P_23_976,
    HI_UNF_ENC_FMT_1080i_59_94,
    HI_UNF_ENC_FMT_BUTT
}HI_UNF_ENC_FMT_E;
```

【修改原因】

增加分辨率为 3840*2160 和 4096*2160 的显示制式

【注意事项】

无

【范例】

无

11.5 HDMI

11.5.1 概述

UNF3.2.9 相对于 UNF3.2.8，HDMI 模块在总体功能上有以下变更：

- 修改支持 YCBCR420 像素格式
- 修改支持新的色域空间

- 修改 EDID 支持新的色域空间能力集

11.5.1.1 修改支持 YCBCR420 像素格式

数据结构

修改 HI_UNF_HDMI_VIDEO_MODE_E

11.5.1.2 修改支持新的色域空间

数据结构

修改 HI_UNF_HDMI_COLORSPACE_E

11.5.1.3 修改 EDID 支持新的色域空间能力集

数据结构

修改 HI_UNF_EDID_COLORIMETRY_S

11.5.2 数据结构

11.5.2.1 修改

HI_UNF_HDMI_VIDEO_MODE_E

【结构体定义】

修改前：

```
typedef enum hiUNF_HDMI_VIDEO_MODE
{
    HI_UNF_HDMI_VIDEO_MODE_RGB444,
    HI_UNF_HDMI_VIDEO_MODE_YCBCR422,
    HI_UNF_HDMI_VIDEO_MODE_YCBCR444,
    HI_UNF_HDMI_VIDEO_MODE_BUTT
}HI_UNF_HDMI_VIDEO_MODE_E;
```

修改后：

```
typedef enum hiUNF_HDMI_VIDEO_MODE
{
    HI_UNF_HDMI_VIDEO_MODE_RGB444,
    HI_UNF_HDMI_VIDEO_MODE_YCBCR422,
    HI_UNF_HDMI_VIDEO_MODE_YCBCR444,
    HI_UNF_HDMI_VIDEO_MODE_YCBCR420,
    HI_UNF_HDMI_VIDEO_MODE_BUTT
}HI_UNF_HDMI_VIDEO_MODE_E;
```

【修改说明】

增加 YCBCR420 像素格式。

【注意事项】

无

【范例】

无

HI_UNF_HDMI_COLORSPACE_E

【结构体定义】

修改前：

```
typedef enum hiUNF_HDMI_COLORSPACE_E
{
    HDMI_COLORIMETRY_NO_DATA,
    HDMI_COLORIMETRY_ITU601,
    HDMI_COLORIMETRY_ITU709,
    HDMI_COLORIMETRY_EXTENDED,
    HDMI_COLORIMETRY_XVYCC_601,
    HDMI_COLORIMETRY_XVYCC_709,
} HI_UNF_HDMI_COLORSPACE_E;
```

修改后：

```
typedef enum hiUNF_HDMI_COLORSPACE_E
{
    HDMI_COLORIMETRY_NO_DATA,
    HDMI_COLORIMETRY_ITU601,
    HDMI_COLORIMETRY_ITU709,
    HDMI_COLORIMETRY_EXTENDED,
    HDMI_COLORIMETRY_XVYCC_601,
    HDMI_COLORIMETRY_XVYCC_709,
    HDMI_COLORIMETRY_S_YCC_601,
    HDMI_COLORIMETRY_ADOBE_YCC_601,
    HDMI_COLORIMETRY_ADOBE_RGB,
    HDMI_COLORIMETRY_2020_CONST_LUMINOUS,
    HDMI_COLORIMETRY_2020_NON_CONST_LUMINOUS,
} HI_UNF_HDMI_COLORSPACE_E;
```

【修改说明】

增加 S_YCC_601、ADOBE_YCC_601、ADOBE_RGB、BT2020cYCC、BT2020RGB 或 BT2020YCC 色域空间。

【注意事项】

无

【范例】

无

HI_UNF_EDID_COLORIMETRY_S

【结构体定义】

修改前：

```
typedef struct hiUNF_EDID_COLORIMETRY_S
{
    HI_BOOL    bxvYCC601     ;
    HI_BOOL    bxvYCC709     ;
} HI_UNF_EDID_COLORIMETRY_S;
```

修改后：

```
typedef struct hiUNF_EDID_COLORIMETRY_S
{
    HI_BOOL    bxvYCC601     ;
    HI_BOOL    bxvYCC709     ;
    HI_BOOL    bsYCC601      ;
    HI_BOOL    bAdobleYCC601 ;
    HI_BOOL    bAdobleRGB    ;
    HI_BOOL    bBT2020cYCC   ;
    HI_BOOL    bBT2020YCC    ;
    HI_BOOL    bBT2020RGB    ;
} HI_UNF_EDID_COLORIMETRY_S;
```

【修改说明】

增加是否支持 S_YCC_601、ADOBE_YCC_601、ADOBE_RGB、BT2020cYCC、BT2020RGB 或 BT2020YCC 色域空间能力集。

【注意事项】

无

【范例】

无

11.6 Common

11.6.1 概述

UNF3.2.9 相对于 UNF3.2.8 版本，common 模块在总体功能上有以下变更：

新增 HI_SYS_CRC32 接口

11.6.1.1 新增 HI_SYS_CRC32 接口

为支持 64 位 ChipID 转换为 32 位 ChipID，新增 HI_SYS_CRC32 接口。

接口

新增 HI_SYS_CRC32

11.6.2 函数接口

11.6.2.1 新增

HI_SYS_CRC32

【接口定义】

```
HI_S32 HI_SYS_CRC32(HI_U8 *pu8Src, HI_U32 u32SrcLen, HI_U32 *pu32Dst);
```

【新增原因】

新特性，支持将 64 位芯片 ChipID 转换为 32 位芯片 ChipID。

【注意事项】

第一个参数为输入据的 buffer 指针，即 64 位 ChipID 数据的指针，第二个参数为 buffer 长度，第三个参数转换后 32 位数据的 buffer 指针。

【范例】

无

11.7 CIPHER

11.7.1 概述

UNF 3.2.9 相对于 UNF 3.2.8 版本，CIPHER 模块在总体功能上有以下变更：

新增使用 STB_ROOT_KEY 加解密数据。

11.7.1.1 使用 STB_ROOT_KEY 加解密数据

允许用户使用 STB_ROOT_KEY 加解密数据，引起的变更如下：

数据结构

修改 HI_UNF_CIPHER_CA_TYPE_E

函数接口

无

11.7.2 数据结构

11.7.2.1 修改

HI_UNF_CIPHER_CA_TYPE_E

【结构体定义】

修改前：

```
typedef enum hiUNF_CIPHER_CA_TYPE_E
{
    HI_UNF_CIPHER_CA_TYPE_R2R   = 0x0,
    HI_UNF_CIPHER_CA_TYPE_SP,
    HI_UNF_CIPHER_CA_TYPE_CSA2,
    HI_UNF_CIPHER_CA_TYPE_CSA3,
    HI_UNF_CIPHER_CA_TYPE_MISC,
    HI_UNF_CIPHER_CA_TYPE_GDRM,
    HI_UNF_CIPHER_CA_TYPE_BLPK,
    HI_UNF_CIPHER_CA_TYPE_LPK,
    HI_UNF_CIPHER_CA_TYPE_IRDETO_HCA,
}HI_UNF_CIPHER_CA_TYPE_E;
```

修改后：

```
typedef enum hiUNF_CIPHER_CA_TYPE_E
{
    HI_UNF_CIPHER_CA_TYPE_R2R   = 0x0,
    HI_UNF_CIPHER_CA_TYPE_SP,
    HI_UNF_CIPHER_CA_TYPE_CSA2,
    HI_UNF_CIPHER_CA_TYPE_CSA3,
    HI_UNF_CIPHER_CA_TYPE_MISC,
    HI_UNF_CIPHER_CA_TYPE_GDRM,
    HI_UNF_CIPHER_CA_TYPE_BLPK,
    HI_UNF_CIPHER_CA_TYPE_LPK,
    HI_UNF_CIPHER_CA_TYPE_IRDETO_HCA,
    HI_UNF_CIPHER_CA_TYPE_STBROOTKEY,
    HI_UNF_CIPHER_CA_TYPE_BUTT
}HI_UNF_CIPHER_CA_TYPE_E;
```

【修改原因】

允许用户使用 STB_ROOT_KEY 加解密数据。

【注意事项】

无

【范例】

sample\cipher\sample_anticopy.c。

11.8 CC

11.8.1 概述

UNF 3.2.9 相对于 UNF 3.2.8 版本，CC 模块在整体功能上有以下变更：

修改使奇偶场数据能同时送到 VO 进行输出。

11.8.1.1 修改使奇偶场数据能同时送到 VO 进行输出

为保证将一帧的奇场和偶场数据同时送到 VO 进行输出，引起的变更如下：

接口

修改 HI_UNF_CC_VBI_CB_FN

11.8.2 函数接口

11.8.2.1 修改

HI_UNF_CC_VBI_CB_FN

【接口定义】

修改前：

```
typedef HI_S32 (*HI_UNF_CC_VBI_CB_FN)(HI_U32 u32UserData,
HI_UNF_CC_VBI_DADA_S *pstVBIDataField);
```

修改后：

```
typedef HI_S32 (*HI_UNF_CC_VBI_CB_FN)(HI_U32 u32UserData,
HI_UNF_CC_VBI_DADA_S *pstVBIOddDataField1,HI_UNF_CC_VBI_DADA_S
*pstVBIEvenDataField2);
```

【修改说明】

在 VBI 回调处理函数中，将奇场数据和偶场数据同时传递，以保证这两场数据为同一帧的数据。

【注意事项】

无

【范例】

无

11.9 PMOC

11.9.1 概述

UNF 3.2.9 相对于 UNF 3.2.8 版本，PMOC 模块在总体功能上有以下变更：

新增智能待机可单独控制不下电的多个模块

11.9.1.1 智能待机可单独控制不下电的多个模块的功能

添加了多个模块的枚举，在智能待机的时候可以单独控制这些模块是否不下电。

数据结构

修改 HI_UNF_PMOC_HOLD_MOD_E

11.9.2 数据结构

11.9.2.1 修改

HI_UNF_PMOC_HOLD_MOD_E

【结构体定义】

修改前：

```
typedef enum hiUNF_PMOC_HOLD_MOD_E
{
    HI_UNF_PMOC_HOLD_ETH = 0x0001,
    HI_UNF_PMOC_HOLD_WIFI = 0x0002,
    HI_UNF_PMOC_HOLD_USB = 0x0004,
    HI_UNF_PMOC_HOLD_TUNER = 0x0008,
    HI_UNF_PMOC_HOLD_DEMUX = 0x0010,
    HI_UNF_PMOC_HOLD_SDIO = 0x0020,
    HI_UNF_PMOC_HOLD_BUTT
}HI_UNF_PMOC_HOLD_MOD_E;
```

修改后：

```
typedef enum hiUNF_PMOC_HOLD_MOD_E
{
    HI_UNF_PMOC_HOLD_ETH = 0x0001,
    HI_UNF_PMOC_HOLD_WIFI = 0x0002,
    HI_UNF_PMOC_HOLD_USB = 0x0004,
    HI_UNF_PMOC_HOLD_TUNER = 0x0008,
```

```
    HI_UNF_PMOC_HOLD_DEMUX = 0x0010,
    HI_UNF_PMOC_HOLD_SDIO = 0x0020,
    HI_UNF_PMOC_HOLD_SCI = 0x0040,
    HI_UNF_PMOC_HOLD_VENC = 0x0080,
    HI_UNF_PMOC_HOLD_PNG = 0x0100,
    HI_UNF_PMOC_HOLD_JPGE = 0x0200,
    HI_UNF_PMOC_HOLD_JPEG = 0x0400,
    HI_UNF_PMOC_HOLD_WDG = 0x0800,
    HI_UNF_PMOC_HOLD_HDMI = 0x1000,
    HI_UNF_PMOC_HOLD_VO = 0x2000,
    HI_UNF_PMOC_HOLD_DISP = 0x4000,
    HI_UNF_PMOC_HOLD_AO = 0x8000,
    HI_UNF_PMOC_HOLD_AI = 0x10000,
    HI_UNF_PMOC_HOLD_ADSP = 0x20000,
    HI_UNF_PMOC_HOLD_CIPHER = 0x40000,
    HI_UNF_PMOC_HOLD_VDEC = 0x80000,
    HI_UNF_PMOC_HOLD_VPSS = 0x100000,
    HI_UNF_PMOC_HOLD_OTP = 0x200000,
    HI_UNF_PMOC_HOLD_TDE = 0x400000,
    HI_UNF_PMOC_HOLD_I2C = 0x800000,
    HI_UNF_PMOC_HOLD_GPIO = 0x1000000,
    HI_UNF_PMOC_HOLD_BUTT = 0x80000000,
}HI_UNF_PMOC_HOLD_MOD_E;
```

【修改原因】

为了支持智能待机时单独关闭 DISPLAY 等模块；

【注意事项】

这个枚举表示的是哪个模块在智能待机的时候不下电，如果只想让某个模块下电，那么需要将其他的所有模块的枚举逻辑或在一起，作为参数传入进去；

【范例】

无

11.10 HiPlayer

11.10.1 概述

UNF 3.2.9 相对于 UNF 3.2.8 版本，HiPlayer 模块在总体功能上有以下变更：

- Invoke 接口扩展
- 支持 proc 获取 seek 信息和节目切换信息
- 增加同步 FENCE 接口

11.10.1.1 Invoke 接口扩展

该特性实现以下功能：

- 获取视频缓冲最后一帧 pts
- 获取音频缓冲最后一帧 pts
- 设置 proc 记录 stream
- proc 获取 seek 信息
- proc 获取切台信息

数据结构

修改 HI_FORMAT_INVOKE_ID_E

11.10.1.2 支持 proc 获取 seek 信息和换台时间信息

该特性实现以下功能：

支持 proc 获取 seek 信息和换台时间信息。

数据结构

- 新增 HI_SVR_PLAYER_PROC_SEEKINFO_S
- 新增 HI_SVR_PLAYER_PROC_SWITCHPG_INFOTYPE_E
- 新增 HI_SVR_PLAYER_PROC_SWITCHPG_S

11.10.1.3 增加同步 FENCE 接口

该特性实现以下功能：

用于同步 FENCE。

数据结构

修改 HI_SVR_PLAYER_PROC_SEEKINFO_S

函数接口

新增 HI_SVR_VSINK_CheckFence

11.10.2 数据结构

11.10.2.1 新增

HI_SVR_PLAYER_PROC_SEEKINFO_S

【结构体定义】

```
typedef struct hiSVR_PLAYER_PROC_SEEKINFO_S
{
```

```
    HI_U32      u32DoReadSeek;
    HI_U32      u32ReadSeekDone;
    HI_U32      u32DoSeekFrameBinary;
    HI_U32      u32SeekFrameBinaryDone;
    HI_U32      u32DoSeekFrameGeneric;
    HI_U32      u32SeekFrameGenericDone;
    HI_U32      u32DoInitInput;
    HI_U32      u32DoAvioOpenH;
    HI_U32      u32AvioOpenHDone;
    HI_U32      u32DoAvProbeInputBuffer;
    HI_U32      u32AvProbeInputBufferDone;

    HI_U32      u32DoHiSvrFormatSeekPts;
    HI_U32      u32CmdSeek;
    HI_U32      u32DoAvformatOpenInput;
    HI_U32      u32AvformatOpenInputDone;
    HI_U32      u32DoSvrFormatFindStream;
    HI_U32      u32SvrFormatFindStreamDone;
    HI_U32      u32DoSvrFormatGetFileInfo;
    HI_U32      u32SvrFormatGetFileInfoDone;

} HI_SVR_PLAYER_PROC_SEEKINFO_S;
```

【新增原因】

获取 seek 时对应的 proc 信息。

【注意事项】

无

【范例】

无

HI_SVR_PLAYER_PROC_SWITCHPG_INFOTYPE_E

【枚举定义】

```
typedef enum hiSVR_PLAYER_PROC_SWITCHPG_INFOTYPE_E
{
    HI_SVR_PLAYER_PROC_DO_STOP=1,
    HI_SVR_PLAYER_PROC_HIMEDIAPLAYER_CONSTRUCT,
    HI_SVR_PLAYER_PROC_SETDATASOURCE,
    HI_SVR_PLAYER_PROC_UNF_AVPLAY_CREATE,
    HI_SVR_PLAYER_PROC_DO_PREPARE,
    HI_SVR_PLAYER_PROC_PREPARE_ASYNC_COMPLETE,
    HI_SVR_PLAYER_PROC_DO_START_ENTER,
```

```
    HI_SVR_PLAYER_PROC_PLAYER_STATE_PLAY,
    HI_SVR_PLAYER_PROC_MEDIA_INFO_FIRST_FRAME_TIME,
    HI_SVR_PLAYER_PROC_DO_RESET,
    HI_SVR_PLAYER_PROC_DO_DESTRUCTOR,
HI_SVR_PLAYER_ATTR_BUTT
} HI_SVR_PLAYER_PROC_SWITCHPG_INFOTYPE_E;
```

【新增原因】

切台动作枚举。

【注意事项】

无

【范例】

无

HI_SVR_PLAYER_PROC_SWITCHPG_S

【结构体定义】

```
typedef struct hiSVR_PLAYER_PROC_SWITCHPG_S
{
    HI_SVR_PLAYER_PROC_SWITCHPG_INFOTYPE_E eType;
    HI_U32 u32DoStop;
    HI_U32 u32HiMediaPlayerConstruct;
    HI_U32 u32SetDataSource;
    HI_U32 u32DoCreateAVPlay;
    HI_U32 u32DoPrepare;
    HI_U32 u32prepareAsyncComplete;
    HI_U32 u32DoStartEnter;
    HI_U32 u32PlayedEvent;
    HI_U32 u32FirstFrameTime;
    HI_U32 u32DoReset;
    HI_U32 u32DoDestructor;

} HI_SVR_PLAYER_PROC_SWITCHPG_S;
```

【新增原因】

获取切台信息。

【注意事项】

无

【范例】

无

11.10.2.2 修改

HI_FORMAT_INVOKE_ID_E

【枚举定义】

修改前：

```
typedef enum hiFORMAT_INVOKE_ID_E
{
    ......
    HI_FORMAT_INVOKE_SET_EXTERNALFORMAT,
    HI_FORMAT_INVOKE_FIND_BESTSTREAM,
    HI_FORMAT_INVOKE_SET_EXTERNAL_AUDIOTRACK,
    HI_FORMAT_INVOKE_SET_SEEK_MODE,
    HI_FORMAT_INVOKE_BUTT
} HI_FORMAT_INVOKE_ID_E;
```

修改后：

```
typedef enum hiFORMAT_INVOKE_ID_E
{
    ......
    HI_FORMAT_INVOKE_SET_EXTERNALFORMAT,
    HI_FORMAT_INVOKE_FIND_BESTSTREAM,
    HI_FORMAT_INVOKE_SET_EXTERNAL_AUDIOTRACK,
    HI_FORMAT_INVOKE_SET_SEEK_MODE,
    HI_FORMAT_INVOKE_GET_LAST_VIDBUF_PTS,
    HI_FORMAT_INVOKE_GET_LAST_AUDBUF_PTS,
    HI_FORMAT_INVOKE_SET_RECORD_STREAM,
    HI_FORMAT_INVOKE_GET_TRACE_SEEK,
    HI_FORMAT_INVOKE_GET_SWITCH_PG_TIME,
    HI_FORMAT_INVOKE_BUTT
} HI_FORMAT_INVOKE_ID_E;
```

【修改原因】

根据需求和功能要求，增加以下 invoke 接口：

- 获取视频缓冲最后一帧 pts
- 获取音频缓冲最后一帧 pts
- 设置 proc 记录 stream
- proc 获取 seek 信息
- proc 获取切台信息

【注意事项】

无

【范例】

无

HI_SVR_PLAYER_PROC_SEEKINFO_S

【枚举定义】

修改前：

```
typedef enum hiSVR_VSINK_CMD_E
{
    HI_SVR_VSINK_SET_DIMENSIONS,
    HI_SVR_VSINK_SET_FORMAT,
    HI_SVR_VSINK_SET_PICNB,
    HI_SVR_VSINK_WRITE_PIC,
    HI_SVR_VSINK_SET_CROP,
    HI_SVR_VSINK_GET_MINBUFNB,
} HI_SVR_VSINK_CMD_E;
```

修改后：

```
typedef enum hiSVR_VSINK_CMD_E
{
    HI_SVR_VSINK_SET_DIMENSIONS,
    HI_SVR_VSINK_SET_FORMAT,
    HI_SVR_VSINK_SET_PICNB,
    HI_SVR_VSINK_WRITE_PIC,
    HI_SVR_VSINK_SET_CROP,
    HI_SVR_VSINK_GET_MINBUFNB,
    HI_SVR_VSINK_CHECK_FENCE,
} HI_SVR_VSINK_CMD_E;
```

【修改原因】

增加同步 FENCE 接口对应的枚举变量。

【注意事项】

只适用于 Android 版本，linux 版本不支持。

【范例】

无

11.10.3 函数接口

11.10.3.1 新增

HI_SVR_VSINK_CheckFence

【函数定义】

```
static inline HI_S32 HI_SVR_VSINK_CheckFence(HI_SVR_VSINK_S* vsink,
       HI_SVR_PICTURE_S* pic)
```

【新增原因】

检测 FENCE 同步接口。

【注意事项】

只用于 Android 版本，linux 版本不支持。

【范例】

无

11.11 PVR

11.11.1 概述

UNF 3.2.9 相对于 UNF 3.2.8 版本，PVR 模块在总体功能上有以下变更：

新增慢退播放模式的枚举定义。

11.11.1.1 新增慢退播放模式定义

数据结构

修改 HI_UNF_PVR_PLAY_STATE_E

函数接口

无

11.11.2 数据结构

11.11.2.1 修改

HI_UNF_PVR_PLAY_STATE_E

【结构体定义】

修改前：

```
typedef enum hiUNF_PVR_PALY_STATE_E
{
    HI_UNF_PVR_PLAY_STATE_INVALID,
    HI_UNF_PVR_PLAY_STATE_INIT,
    HI_UNF_PVR_PLAY_STATE_PLAY,
    HI_UNF_PVR_PLAY_STATE_PAUSE,
    HI_UNF_PVR_PLAY_STATE_FF,
    HI_UNF_PVR_PLAY_STATE_FB,
    HI_UNF_PVR_PLAY_STATE_SF,
    HI_UNF_PVR_PLAY_STATE_STEPF,
    HI_UNF_PVR_PLAY_STATE_STEPB,
    HI_UNF_PVR_PLAY_STATE_STOP,
    HI_UNF_PVR_PLAY_STATE_BUTT
} HI_UNF_PVR_PLAY_STATE_E;
```

修改后：

```
typedef enum hiUNF_PVR_PALY_STATE_E
{
    HI_UNF_PVR_PLAY_STATE_INVALID,
    HI_UNF_PVR_PLAY_STATE_INIT,
    HI_UNF_PVR_PLAY_STATE_PLAY,
    HI_UNF_PVR_PLAY_STATE_PAUSE,
    HI_UNF_PVR_PLAY_STATE_FF,
    HI_UNF_PVR_PLAY_STATE_FB,
    HI_UNF_PVR_PLAY_STATE_SF,
    HI_UNF_PVR_PLAY_STATE_SB,
    HI_UNF_PVR_PLAY_STATE_STEPF,
    HI_UNF_PVR_PLAY_STATE_STEPB,
    HI_UNF_PVR_PLAY_STATE_STOP,
    HI_UNF_PVR_PLAY_STATE_BUTT
} HI_UNF_PVR_PLAY_STATE_E;
```

【修改原因】

补齐 PVR 各种播放状态。

【注意事项】

本次修改只是针对 PVR 播放状态的补充，功能暂时不支持。

【范例】

无

11.12 Frontend

11.12.1 概述

UNF 3.2.9 相对于 UNF 3.2.8 版本，Frontend 模块在总体功能上有以下变更：

- 增加 AV2018 和 SI2144 两款 Tuner 支持。
- 增加设置物理层扰码初始相位接口。

11.12.1.1 增加 AV2018 和 SI2144 两款 Tuner 支持

增加 AV2018 和 SI2144 两款 Tuner 支持，引起的变更如下：

数据结构

修改 HI_UNF_TUNER_DEV_TYPE_E

11.12.1.2 增加设置物理层扰码初始相位接口

设置物理层扰码初始相位接口，引起的变更如下：

函数接口

新增 HI_UNF_TUNER_SetScramble

11.12.2 数据结构

11.12.2.1 修改

HI_UNF_TUNER_DEV_TYPE_E

【结构体定义】

修改前：

```
typedef enum    hiUNF_TUNER_DEV_TYPE_E
{
    HI_UNF_TUNER_DEV_TYPE_XG_3BL,
    HI_UNF_TUNER_DEV_TYPE_CD1616,
    HI_UNF_TUNER_DEV_TYPE_ALPS_TDAE,
    HI_UNF_TUNER_DEV_TYPE_TDCC,
    HI_UNF_TUNER_DEV_TYPE_TDA18250,
    HI_UNF_TUNER_DEV_TYPE_CD1616_DOUBLE,
    HI_UNF_TUNER_DEV_TYPE_MT2081,
    HI_UNF_TUNER_DEV_TYPE_TMX7070X,
    HI_UNF_TUNER_DEV_TYPE_R820C,
```

```
    HI_UNF_TUNER_DEV_TYPE_MXL203,
    HI_UNF_TUNER_DEV_TYPE_AV2011,
    HI_UNF_TUNER_DEV_TYPE_SHARP7903,
    HI_UNF_TUNER_DEV_TYPE_MXL101,
    HI_UNF_TUNER_DEV_TYPE_MXL603,
    HI_UNF_TUNER_DEV_TYPE_IT9170,
    HI_UNF_TUNER_DEV_TYPE_IT9133,
    HI_UNF_TUNER_DEV_TYPE_TDA6651,
    HI_UNF_TUNER_DEV_TYPE_TDA18250B,
    HI_UNF_TUNER_DEV_TYPE_M88TS2022,
    HI_UNF_TUNER_DEV_TYPE_RDA5815,
    HI_UNF_TUNER_DEV_TYPE_MXL254,
    HI_UNF_TUNER_DEV_TYPE_CXD2861,
    HI_UNF_TUNER_DEV_TYPE_SI2147,
    HI_UNF_TUNER_DEV_TYPE_RAFAEL836,
    HI_UNF_TUNER_DEV_TYPE_MXL608,
    HI_UNF_TUNER_DEV_TYPE_MXL214,
    HI_UNF_TUNER_DEV_TYPE_TDA18280,
    HI_UNF_TUNER_DEV_TYPE_TDA182I5A,
    HI_UNF_TUNER_DEV_TYPE_BUTT
} HI_UNF_TUNER_DEV_TYPE_E ;
```

修改后：

```
typedef enum    hiUNF_TUNER_DEV_TYPE_E
{
    HI_UNF_TUNER_DEV_TYPE_XG_3BL,
    HI_UNF_TUNER_DEV_TYPE_CD1616,
    HI_UNF_TUNER_DEV_TYPE_ALPS_TDAE,
    HI_UNF_TUNER_DEV_TYPE_TDCC,
    HI_UNF_TUNER_DEV_TYPE_TDA18250,
    HI_UNF_TUNER_DEV_TYPE_CD1616_DOUBLE,
    HI_UNF_TUNER_DEV_TYPE_MT2081,
    HI_UNF_TUNER_DEV_TYPE_TMX7070X,
    HI_UNF_TUNER_DEV_TYPE_R820C,
```

```
    HI_UNF_TUNER_DEV_TYPE_MXL203,
    HI_UNF_TUNER_DEV_TYPE_AV2011,
    HI_UNF_TUNER_DEV_TYPE_SHARP7903,
    HI_UNF_TUNER_DEV_TYPE_MXL101,
    HI_UNF_TUNER_DEV_TYPE_MXL603,
    HI_UNF_TUNER_DEV_TYPE_IT9170,
    HI_UNF_TUNER_DEV_TYPE_IT9133,
    HI_UNF_TUNER_DEV_TYPE_TDA6651,
    HI_UNF_TUNER_DEV_TYPE_TDA18250B,
    HI_UNF_TUNER_DEV_TYPE_M88TS2022,
    HI_UNF_TUNER_DEV_TYPE_RDA5815,
    HI_UNF_TUNER_DEV_TYPE_MXL254,
    HI_UNF_TUNER_DEV_TYPE_CXD2861,
    HI_UNF_TUNER_DEV_TYPE_SI2147,
    HI_UNF_TUNER_DEV_TYPE_RAFAEL836,
    HI_UNF_TUNER_DEV_TYPE_MXL608,
    HI_UNF_TUNER_DEV_TYPE_MXL214,
    HI_UNF_TUNER_DEV_TYPE_TDA18280,
    HI_UNF_TUNER_DEV_TYPE_TDA182I5A,
    HI_UNF_TUNER_DEV_TYPE_SI2144,
    HI_UNF_TUNER_DEV_TYPE_AV2018,
    HI_UNF_TUNER_DEV_TYPE_BUTT
} HI_UNF_TUNER_DEV_TYPE_E ;
```

【修改原因】

新增两款 tuner(Si2144/Av2018)驱动支持。

【注意事项】

无

【范例】

无

11.12.3 函数接口

11.12.3.1 新增

HI_UNF_TUNER_SetScramble

【接口定义】

```
HI_S32 HI_UNF_TUNER_SetScramble(HI_U32 u32TunerId,HI_U32 u32N);
```

【新增原因】

设置物理层扰码初始相位值。

【注意事项】

无

【范例】

无

12 UNF3.2.10 与 UNF3.2.9 的差异说明

12.1 ADVCA

12.1.1 概述

UNF 3.2.10 相对于 UNF 3.2.9 版本，ADVCA 模块在总体功能上有以下变更：

- 新增 keyladder 类型。
- 新增 PANACCESS CA 芯片类型。
- 新增设置和获取安全启动使能标记功能。
- 新增设置和获取安全启动 Flash 类型功能。
- 新增设置和获取多种锁标记位的功能。

12.1.1.1 新增 keyladder 类型

新增了 BLPK，LPK 和 IRDETO HCA 三种 key ladder 类型，引起的变更如下：

数据结构

修改 HI_UNF_ADVCA_CA_TYPE_E

12.1.1.2 新增 PANACCESS CA 芯片类型

新增了 PANACCESS CA 的芯片类型，引起的变更如下：

数据结构

修改 HI_UNF_ADVCA_VENDORID_E

12.1.1.3 新增设置和获取安全启动使能标记功能

该特性允许使能安全启动或者查询当前安全启动是否开启。引起的变更如下：

数据结构

修改 HI_UNF_ADVCA_OTP_FUSE_E

12.1.1.4 新增设置和获取安全启动 Flash 类型功能。

该特性可以实现设置和获取安全启动 Flash 类型，引起的变更如下：

数据结构

- 新增 HI_UNF_ADVCA_OTP_BOOT_FLASH_TYPE_ATTR_S
- 修改 HI_UNF_ADVCA_OTP_FUSE_E

12.1.1.5 新增设置和获取多种锁标记位的功能。

新增了设置和获取 RSA KEY LOCK、STBSN LOCK、MSID LOCK、VERSIONID LOCK、OEM ROOTKEY LOCK、R2R ROOTKEY LOCK、JTAG KEY LOCK 和 TZ AREA LOCK 八种锁标记的功能。引起的变更如下：

数据结构

修改 HI_UNF_ADVCA_OTP_FUSE_E

12.1.2 数据结构

12.1.2.1 新增

HI_UNF_ADVCA_OTP_BOOT_FLASH_TYPE_ATTR_S

【结构体定义】

```
typedef struct hiUNF_ADVCA_OTP_BOOT_FLASH_TYPE_ATTR_S
{
    HI_BOOL bBootSelCtrl;    /**<0--the boot flash type is defined by chipset pin, 1--the boot flash type is defined by OTP value*/
    HI_UNF_ADVCA_FLASH_TYPE_E enFlashType; /**<Boot flash type, only valid when bBootSelCtrl is 1*/
}HI_UNF_ADVCA_OTP_BOOT_FLASH_TYPE_ATTR_S;
```

【新增原因】

支持设置和获取安全启动 Flash 类型功能。

【注意事项】

无。

【范例】

无。

12.1.2.2 修改

HI_UNF_ADVCA_CA_TYPE_E

【结构体定义】

修改前：

```
typedef enum hiUNF_ADVCA_CA_TYPE_E
{
    HI_UNF_ADVCA_CA_TYPE_R2R      = 0x0,
    HI_UNF_ADVCA_CA_TYPE_SP       = 0x1,
    HI_UNF_ADVCA_CA_TYPE_CSA2     = 0x1,
    HI_UNF_ADVCA_CA_TYPE_CSA3     = 0x1,
    HI_UNF_ADVCA_CA_TYPE_MISC     = 0x2,
    HI_UNF_ADVCA_CA_TYPE_GDRM     = 0x3,
}HI_UNF_ADVCA_CA_TYPE_E;
```

修改后：

```
typedef enum hiUNF_ADVCA_CA_TYPE_E
{
    HI_UNF_ADVCA_CA_TYPE_R2R   = 0x0,
    HI_UNF_ADVCA_CA_TYPE_SP,
    HI_UNF_ADVCA_CA_TYPE_CSA2,
    HI_UNF_ADVCA_CA_TYPE_CSA3,
    HI_UNF_ADVCA_CA_TYPE_MISC,
    HI_UNF_ADVCA_CA_TYPE_GDRM,
    HI_UNF_ADVCA_CA_TYPE_BLPK,
    HI_UNF_ADVCA_CA_TYPE_LPK,
    HI_UNF_ADVCA_CA_TYPE_IRDETO_HCA,
}HI_UNF_ADVCA_CA_TYPE_E;
```

【修改原因】

新增三种 keyladder 类型。

【注意事项】

无。

【范例】

无。

HI_UNF_ADVCA_VENDORID_E

【结构体定义】

修改前：

```
typedef enum hiUNF_ADVCA_VENDORID_E
{
    HI_UNF_ADVCA_NULL       = 0x00,
    HI_UNF_ADVCA_NAGRA      = 0x01,
    HI_UNF_ADVCA_IRDETO     = 0x02,
    HI_UNF_ADVCA_CONAX      = 0x03,
    HI_UNF_ADVCA_SUMA       = 0x05,
    HI_UNF_ADVCA_NOVEL      = 0x06,
    HI_UNF_ADVCA_VERIMATRIX = 0x07,
    HI_UNF_ADVCA_CTI        = 0x08,
    HI_UNF_ADVCA_COMMONDCA  = 0x0b,
    HI_UNF_ADVCA_DCAS       = 0x0c,
    HI_UNF_ADVCA_VENDORIDE_BUTT
}HI_UNF_ADVCA_VENDORID_E;
```

修改后：

```
typedef enum hiUNF_ADVCA_VENDORID_E
{
    HI_UNF_ADVCA_NULL       = 0x00,
    HI_UNF_ADVCA_NAGRA      = 0x01,
    HI_UNF_ADVCA_IRDETO     = 0x02,
    HI_UNF_ADVCA_CONAX      = 0x03,
    HI_UNF_ADVCA_SUMA       = 0x05,
    HI_UNF_ADVCA_NOVEL      = 0x06,
    HI_UNF_ADVCA_VERIMATRIX = 0x07,
    HI_UNF_ADVCA_CTI        = 0x08,
    HI_UNF_ADVCA_COMMONCA   = 0x0b,
    HI_UNF_ADVCA_DCAS       = 0x0c,
    HI_UNF_ADVCA_PANACCESS  = 0x0e,
    HI_UNF_ADVCA_VENDORIDE_BUTT
}HI_UNF_ADVCA_VENDORID_E;
```

【修改原因】

新增支持 PANACCESSCA 的芯片类型。

【注意事项】

无。

【范例】

无。

HI_UNF_ADVCA_OTP_FUSE_E

【结构体定义】

修改前：

```
typedef enum hiUNF_ADVCA_OTP_FUSE_E
{
    HI_UNF_ADVCA_OTP_NULL = 0,
    HI_UNF_ADVCA_OTP_SECURE_BOOT_ACTIVATION,
    HI_UNF_ADVCA_OTP_BOOT_DECRYPTION_ACTIVATION,
    HI_UNF_ADVCA_OTP_SELF_BOOT_DEACTIVATION,
    HI_UNF_ADVCA_OTP_DDR_WAKEUP_DEACTIVATION,
    HI_UNF_ADVCA_OTP_CSA2_KL_LEVEL_SEL,
    HI_UNF_ADVCA_OTP_R2R_KL_LEVEL_SEL,
    HI_UNF_ADVCA_OTP_SP_KL_LEVEL_SEL,
    HI_UNF_ADVCA_OTP_CSA3_KL_LEVEL_SEL,
    HI_UNF_ADVCA_OTP_LP_DEACTIVATION,
    HI_UNF_ADVCA_OTP_CSA2_CW_HARDONLY_ACTIVATION,
    HI_UNF_ADVCA_OTP_SP_CW_HARDONLY_ACTIVATION,
    HI_UNF_ADVCA_OTP_CSA3_CW_HARDONLY_ACTIVATION,
    HI_UNF_ADVCA_OTP_CSA2_KL_DEACTIVATION,
    HI_UNF_ADVCA_OTP_SP_KL_DEACTIVATION,
    HI_UNF_ADVCA_OTP_CSA3_KL_DEACTIVATION,
    HI_UNF_ADVCA_OTP_MISC_KL_DEACTIVATION,
    HI_UNF_ADVCA_OTP_GOOGLE_KL_DEACTIVATION,
    HI_UNF_ADVCA_OTP_DCAS_KL_DEACTIVATION,
    HI_UNF_ADVCA_OTP_DDR_SCRAMBLE_ACTIVATION,
    HI_UNF_ADVCA_OTP_GLOBAL_LOCK_ACTIVATION,
    HI_UNF_ADVCA_OTP_RUNTIME_CHECK_ACTIVATION,
    HI_UNF_ADVCA_OTP_DDR_WAKEUP_CHECK_ACTIVATION,
    HI_UNF_ADVCA_OTP_VERSION_ID_CHECK_ACTIVATION,
    HI_UNF_ADVCA_OTP_BOOT_MSID_CHECK_ACTIVATION,
    HI_UNF_ADVCA_OTP_JTAG_MODE,
    HI_UNF_ADVCA_OTP_JTAG_READ_DEACTIVATION,
    HI_UNF_ADVCA_OTP_R2R_ROOTKEY,
    HI_UNF_ADVCA_OTP_SP_ROOTKEY,
    HI_UNF_ADVCA_OTP_CSA3_ROOTKEY,
    HI_UNF_ADVCA_OTP_MISC_ROOTKEY,
    HI_UNF_ADVCA_OTP_OEM_ROOTKEY,
    HI_UNF_ADVCA_OTP_ESCK_ROOTKEY,
    HI_UNF_ADVCA_OTP_JTAG_KEY,
    HI_UNF_ADVCA_OTP_CHIP_ID,
    HI_UNF_ADVCA_OTP_MARKET_SEGMENT_ID,
    HI_UNF_ADVCA_OTP_VERSION_ID,
    HI_UNF_ADVCA_OTP_MISC_KL_LEVEL_SEL,
    HI_UNF_ADVCA_OTP_VMX_BL_FUSE, /
    HI_UNF_ADVCA_OTP_IRDETO_ITCSA3_ACTIVATION,
```

```
    HI_UNF_ADVCA_OTP_BOOTINFO_DEACTIVATION,
    HI_UNF_ADVCA_OTP_ITCSA3_IMLB,
    HI_UNF_ADVCA_OTP_FUSE_BUTT
}HI_UNF_ADVCA_OTP_FUSE_E;
```

修改后：

```
typedef enum hiUNF_ADVCA_OTP_FUSE_E
{
    HI_UNF_ADVCA_OTP_NULL = 0,
    HI_UNF_ADVCA_OTP_SECURE_BOOT_ACTIVATION,
    HI_UNF_ADVCA_OTP_BOOT_DECRYPTION_ACTIVATION,
    HI_UNF_ADVCA_OTP_SELF_BOOT_DEACTIVATION,
    HI_UNF_ADVCA_OTP_DDR_WAKEUP_DEACTIVATION,
    HI_UNF_ADVCA_OTP_CSA2_KL_LEVEL_SEL,
    HI_UNF_ADVCA_OTP_R2R_KL_LEVEL_SEL,
    HI_UNF_ADVCA_OTP_SP_KL_LEVEL_SEL,
    HI_UNF_ADVCA_OTP_CSA3_KL_LEVEL_SEL,
    HI_UNF_ADVCA_OTP_LP_DEACTIVATION,
    HI_UNF_ADVCA_OTP_CSA2_CW_HARDONLY_ACTIVATION,
    HI_UNF_ADVCA_OTP_SP_CW_HARDONLY_ACTIVATION,
    HI_UNF_ADVCA_OTP_CSA3_CW_HARDONLY_ACTIVATION,
    HI_UNF_ADVCA_OTP_CSA2_KL_DEACTIVATION,
    HI_UNF_ADVCA_OTP_SP_KL_DEACTIVATION,
    HI_UNF_ADVCA_OTP_CSA3_KL_DEACTIVATION,
    HI_UNF_ADVCA_OTP_MISC_KL_DEACTIVATION,
    HI_UNF_ADVCA_OTP_GOOGLE_KL_DEACTIVATION,
    HI_UNF_ADVCA_OTP_DCAS_KL_DEACTIVATION,
    HI_UNF_ADVCA_OTP_DDR_SCRAMBLE_ACTIVATION,
    HI_UNF_ADVCA_OTP_GLOBAL_LOCK_ACTIVATION,
    HI_UNF_ADVCA_OTP_RUNTIME_CHECK_ACTIVATION,
    HI_UNF_ADVCA_OTP_DDR_WAKEUP_CHECK_ACTIVATION,
    HI_UNF_ADVCA_OTP_VERSION_ID_CHECK_ACTIVATION,
    HI_UNF_ADVCA_OTP_BOOT_MSID_CHECK_ACTIVATION,
    HI_UNF_ADVCA_OTP_JTAG_MODE,
    HI_UNF_ADVCA_OTP_JTAG_READ_DEACTIVATION,
    HI_UNF_ADVCA_OTP_CSA2_ROOTKEY,
    HI_UNF_ADVCA_OTP_R2R_ROOTKEY,
    HI_UNF_ADVCA_OTP_SP_ROOTKEY,
    HI_UNF_ADVCA_OTP_CSA3_ROOTKEY,
    HI_UNF_ADVCA_OTP_MISC_ROOTKEY,
    HI_UNF_ADVCA_OTP_OEM_ROOTKEY,
    HI_UNF_ADVCA_OTP_ESCK_ROOTKEY,
    HI_UNF_ADVCA_OTP_JTAG_KEY,
    HI_UNF_ADVCA_OTP_CHIP_ID,
```

```
    HI_UNF_ADVCA_OTP_MARKET_SEGMENT_ID,
    HI_UNF_ADVCA_OTP_VERSION_ID,
    HI_UNF_ADVCA_OTP_MISC_KL_LEVEL_SEL,
    HI_UNF_ADVCA_OTP_VMX_BL_FUSE,
    HI_UNF_ADVCA_OTP_IRDETO_ITCSA3_ACTIVATION,
    HI_UNF_ADVCA_OTP_BOOTINFO_DEACTIVATION,
    HI_UNF_ADVCA_OTP_ITCSA3_IMLB,
    HI_UNF_ADVCA_OTP_SECURE_BOOT_ACTIVATION_ONLY,
    HI_UNF_ADVCA_OTP_BOOT_FLASH_TYPE,
    HI_UNF_ADVCA_OTP_RSA_KEY_LOCK_FLAG,
    HI_UNF_ADVCA_OTP_STBSN_LOCK_FLAG,
    HI_UNF_ADVCA_OTP_MSID_LOCK_FLAG,
    HI_UNF_ADVCA_OTP_VERSIONID_LOCK_FLAG,
    HI_UNF_ADVCA_OTP_OEM_ROOTKEY_LOCK_FLAG,
    HI_UNF_ADVCA_OTP_R2R_ROOTKEY_LOCK_FLAG,
    HI_UNF_ADVCA_OTP_JTAG_KEY_LOCK_FLAG,
    HI_UNF_ADVCA_OTP_TZ_AREA_LOCK_FLAG,
    HI_UNF_ADVCA_OTP_FUSE_BUTT
}HI_UNF_ADVCA_OTP_FUSE_E;
```

【修改原因】

新增设置和获取安全启动使能标记功能。
新增设置和获取安全启动 Flash 类型功能。
新增设置和获取多种锁标记位的功能。

【注意事项】

无。

【范例】

无。

12.2 Common

12.2.1 概述

UNF3.2.10 相对于 UNF3.2.9 版本，Common 模块在总体功能上有以下变更：

- 新增获取芯片封装类型功能
- 新增 Hi3716MV410、Hi3716MV420 芯片子版本号支持

12.2.1.1 新增获取芯片封装类型功能

因为新增获取芯片封装类型功能原因，对如下数据结构以及接口进行了变更：

数据结构

- 新增 HI_CHIP_PACKAGE_TYPE_E

接口

- 新增 HI_SYS_GetChipPackageType

12.2.1.2 新增 Hi3716MV410、Hi3716MV420 芯片子版本号支持

因为新增 Hi3716MV410、Hi3716MV420 芯片子版本号支持原因对如下数据结构进行了变更：

数据结构

- 修改 HI_CHIP_VERSION_E

12.2.2 数据结构

12.2.2.1 新增

HI_CHIP_PACKAGE_TYPE_E

【结构体定义】

```
typedef enum
{
    HI_CHIP_PACKAGE_TYPE_BGA_15_15 = 0,
    HI_CHIP_PACKAGE_TYPE_BGA_16_16,
    HI_CHIP_PACKAGE_TYPE_BGA_19_19,
    HI_CHIP_PACKAGE_TYPE_BGA_23_23,
    HI_CHIP_PACKAGE_TYPE_BGA_31_31,
    HI_CHIP_PACKAGE_TYPE_QFP_216,
    HI_CHIP_PACKAGE_TYPE_BUTT
} HI_CHIP_PACKAGE_TYPE_E;
```

【新增原因】

新特性，芯片封装类型枚举。

【注意事项】

无

【范例】

无

12.2.2.2 修改

HI_CHIP_VERSION_E

【结构体定义】

修改前：

```
typedef enum hiCHIP_VERSION_E
{
    HI_CHIP_VERSION_V100 = 0x100,
    HI_CHIP_VERSION_V101 = 0x101,
    HI_CHIP_VERSION_V200 = 0x200,
    HI_CHIP_VERSION_V300 = 0x300,
    HI_CHIP_VERSION_V400 = 0x400,
    HI_CHIP_VERSION_BUTT
}HI_CHIP_VERSION_E;
```

修改后：

```
typedef enum hiCHIP_VERSION_E
{
    HI_CHIP_VERSION_V100 = 0x100,
    HI_CHIP_VERSION_V101 = 0x101,
    HI_CHIP_VERSION_V200 = 0x200,
    HI_CHIP_VERSION_V300 = 0x300,
    HI_CHIP_VERSION_V400 = 0x400,
    HI_CHIP_VERSION_V410 = 0x410,
    HI_CHIP_VERSION_V420 = 0x420,
    HI_CHIP_VERSION_BUTT
}HI_CHIP_VERSION_E;
```

【修改说明】

新增 Hi3716MV410、 Hi3716MV420 芯片子版本号。

【注意事项】

无

【范例】

无

12.2.3 函数接口

12.2.3.1 新增

HI_SYS_GetChipPackageType

【接口定义】

```
HI_S32 HI_SYS_GetChipPackageType(HI_CHIP_PACKAGE_TYPE_E *penPackageType);
```

【新增原因】

新特性，新增获取芯片封装类型功能。

【注意事项】

无

【范例】

无

12.3 VENC

12.3.1 概述

UNF 3.2.10 相对于 UNF 3.2.9 版本，VENC 模块在总体功能上有以下变更：

- 新增 slice 大小配置。

12.3.1.1 slice 大小配置

视频编码当设置分 slice 编码模式时，可以通过新增属性 u32SplitSize 来配置 slice 的大小，设置范围需要大于等于 512，单位是字节，引起的变更如下：

数据结构

修改 HI_UNF_VENC_CHN_ATTR_S

12.3.2 数据结构

12.3.2.1 修改

HI_UNF_VENC_CHN_ATTR_S

【结构体定义】

修改前：

```
typedef struct hiUNF_VENC_CHN_ATTR_S
{
```

```
    HI_UNF_VCODEC_TYPE_E        enVencType;
    HI_UNF_VCODEC_CAP_LEVEL_E   enCapLevel;
    HI_UNF_H264_PROFILE_E       enVencProfile;
    HI_U32                      u32Width;
    HI_U32                      u32Height;
    HI_U32                      u32StrmBufSize;
    HI_U32                      u32RotationAngle;
    HI_BOOL                     bSlcSplitEn;
    HI_U32                      u32TargetBitRate;
    HI_U32                      u32TargetFrmRate;
    HI_U32                      u32InputFrmRate;
    HI_U32                      u32Gop;
    HI_U32                      u32MaxQp;
    HI_U32                      u32MinQp;
    HI_BOOL                     bQuickEncode;
    HI_U8                       u8Priority;
    HI_U32                      u32Qlevel;
    HI_U32                      u32DriftRateThr;
}HI_UNF_VENC_CHN_ATTR_S;
```

修改后：

```
typedef struct hiUNF_VENC_CHN_ATTR_S
{
    HI_UNF_VCODEC_TYPE_E        enVencType;
    HI_UNF_VCODEC_CAP_LEVEL_E   enCapLevel;
    HI_UNF_H264_PROFILE_E       enVencProfile;
    HI_U32                      u32Width;
    HI_U32                      u32Height;
    HI_U32                      u32StrmBufSize;
    HI_U32                      u32RotationAngle;
    HI_BOOL                     bSlcSplitEn;
    HI_U32                      u32SplitSize;
    HI_U32                      u32TargetBitRate;
    HI_U32                      u32TargetFrmRate;
    HI_U32                      u32InputFrmRate;
    HI_U32                      u32Gop;
    HI_U32                      u32MaxQp;
    HI_U32                      u32MinQp;
    HI_BOOL                     bQuickEncode;
    HI_U8                       u8Priority;
    HI_U32                      u32Qlevel;
    HI_U32                      u32DriftRateThr;
}HI_UNF_VENC_CHN_ATTR_S;
```

【修改原因】

支持用于根据实际需要配置 slice 分块大小。

【注意事项】

- H.264 协议：slice 大小需要有一定的余量，由于 VEDU 的硬件条件限制，余量的大小最恶劣情况下，要能容忍在此 slicesize 的基础上增加 2304 字节（但是此时最恶劣情况，通常实际测试结果输出的 slice 都仅仅比设置值大 100 个 bytes 以内）。
- 当分 slice 编码模式不使能的情况下，该配置不生效。

【范例】

无。

12.4 PVR

12.4.1 概述

UNF 3.2.10 相对于 UNF 3.2.9 版本，PVR 模块在总体功能上有以下变更：

- 新增添加录制 PID
- 新增删除录制 PID
- 新增删除所有录制 PID

12.4.1.1 新增添加录制 PID

创建 PID 通道并建立与录制通道的绑定关系，引起的变更如下：

函数接口

新增 HI_UNF_PVR_RecAddPID。

12.4.1.2 删除录制 PID

销毁 PID 通道并解除与录制通道的绑定关系，引起的变更如下：

函数接口

新增 HI_UNF_PVR_RecDelPID。

12.4.1.3 删除所有录制 PID

销毁所有与录制通道有绑定关系的 PID 通道，引起的变更如下：

函数接口

新增 HI_UNF_PVR_RecDelAllPID。

12.4.2 函数接口

12.4.2.1 新增

HI_UNF_PVR_RecAddPID

【接口定义】

```
HI_S32 HI_UNF_PVR_RecAddPID(HI_U32 u32ChnID, HI_U32 u32Pid);
```

【新增原因】

支持一个 Demux 录制多个节目。如果需要在现有录制通道的基础上增加一套节目的录制，使用该接口将新增的录制节目的 PID 加入到现有的录制通道上。

【注意事项】

不支持添加当前录制通道不同频点的 PID。

【范例】

sample/common/hi_adp_pvr.c

HI_UNF_PVR_RecDelPID

【接口定义】

```
HI_S32 HI_UNF_PVR_RecDelPID(HI_U32 u32ChnID, HI_U32 u32Pid);
```

【新增原因】

用于将录制通道上绑定的 PID 通道删除，结束对该节目的录制。

【注意事项】

无。

【范例】

sample/common/hi_adp_pvr.c

HI_UNF_PVR_RecDelAllPID

【接口定义】

```
HI_S32 HI_UNF_PVR_RecDelAllPID(HI_U32 u32ChnID);
```

【新增原因】

用于将录制通道上绑定的所有 PID 通道删除，结束对该录制通道上所有节目的录制。

【注意事项】

无。

【范例】

sample/common/hi_adp_pvr.c

12.5 PQ

12.5.1 概述

相对于 UNF3.2.9 版本，UNF3.2.10 在总体功能上有以下变更：

- 新增降噪功能
- 新增卖场显示模式功能
- 新增单独设置视频/图形的亮度、色调、对比度、饱和度功能
- 新增设置 TNR、DEI、DBM 的卖场模式功能

12.5.1.1 新增降噪功能

PQ 新集成了 TNR 算法，引起的变更如下：

数据结构

- 修改 HI_UNF_PQ_MODULE_E
- 修改 HI_UNF_PQ_DEMO_E

接口

- 新增 HI_UNF_PQ_GetNR
- 新增 HI_UNF_PQ_SetNR

12.5.1.2 新增卖场显示模式功能

为了呈现不同的卖场模式，滚动/固定方式，左边/右边算法增强，引起的变更如下：

数据结构

- 新增 HI_UNF_PQ_DEMO_MODE_E

接口

- 新增 HI_UNF_PQ_SetDemoMode
- 新增 HI_UNF_PQ_GetDemoMode

12.5.1.3 新增单独设置视频/图形的亮度、色调、对比度、饱和度功能

为了设置视频层的图像效果而不影响图形界面的效果，故新增单独设置视频/图形的亮度、色调、对比度、饱和度功能，引起的变更如下：

数据结构

- 新增 HI_UNF_PQ_IMAGE_TYPE_E
- 新增 HI_UNF_PQ_IMAGE_PARAM_S

接口

- 新增 HI_UNF_PQ_SetBasicImageParam
- 新增 HI_UNF_PQ_GetBasicImageParam

12.5.1.4 新增设置 TNR、DEI、DBM 的卖场模式功能

为了对比显示 TNR、DEI、DBM 的显示效果，引起的变更如下：

数据结构

- 修改 HI_UNF_PQ_DEMO_E

12.5.2 数据结构

12.5.2.1 新增

HI_UNF_PQ_DEMO_MODE_E

【枚举定义】

```
typedef enum hiUNF_PQ_DEMO_MODE_E
{
    HI_UNF_PQ_DEMO_MODE_FIXED_R,
    HI_UNF_PQ_DEMO_MODE_FIXED_L,
    HI_UNF_PQ_DEMO_MODE_SCROLL_R,
    HI_UNF_PQ_DEMO_MODE_SCROLL_L,

    HI_UNF_PQ_DEMO_MODE_BUTT
}  HI_UNF_PQ_DEMO_MODE_E;
```

【新增原因】

卖场模式的各种显示方式。

【注意事项】

无。

【范例】

sample/pq/sample_pq_v2.c

HI_UNF_PQ_IMAGE_TYPE_E

【枚举定义】

```
typedef enum hiUNF_PQ_IMAGE_TYPE_E
{
    HI_UNF_PQ_IMAGE_GRAPH = 0,
    HI_UNF_PQ_IMAGE_VIDEO,
```

```
    HI_UNF_PQ_IMAGE_BUTT
} HI_UNF_PQ_IMAGE_TYPE_E;
```

【新增原因】

设置图像参数，选择图形层还是视频层开关控制。

【注意事项】

无。

【范例】

sample/pq/sample_pq_v2.c

HI_UNF_PQ_IMAGE_PARAM_S

【枚举定义】

```
typedef struct hiUNF_PQ_IMAGE_PARAM_S
{
    HI_U32      u32Brightness;
    HI_U32      u32Contrast;
    HI_U32      u32Hue;
    HI_U32      u32Saturation;
} HI_UNF_PQ_IMAGE_PARAM_S;
```

【新增原因】

设置图像的亮度、色调、对比度、饱和度。

【注意事项】

亮度、色调、对比度、饱和度的范围为 0 至 100。

【范例】

sample/pq/sample_pq_v2.c

12.5.2.2 修改

HI_UNF_PQ_DEMO_E

【结构体定义】

修改前：

```
typedef enum hiUNF_PQ_DEMO_E
{
    HI_UNF_PQ_DEMO_SHARPNESS = 0,
    HI_UNF_PQ_DEMO_DCI,
```

```
    HI_UNF_PQ_DEMO_COLOR,
    HI_UNF_PQ_DEMO_SR,
    HI_UNF_PQ_DEMO_ALL,

    HI_UNF_PQ_DEMO_BUTT
}  HI_UNF_PQ_DEMO_E;
```

修改后：

```
typedef enum hiUNF_PQ_DEMO_E
{
    HI_UNF_PQ_DEMO_SHARPNESS = 0,
    HI_UNF_PQ_DEMO_DCI,
    HI_UNF_PQ_DEMO_COLOR,
    HI_UNF_PQ_DEMO_SR,
    HI_UNF_PQ_DEMO_TNR,
    HI_UNF_PQ_DEMO_DEI,
    HI_UNF_PQ_DEMO_DBM,
    HI_UNF_PQ_DEMO_ALL,

    HI_UNF_PQ_DEMO_BUTT
}  HI_UNF_PQ_DEMO_E;
```

【修改原因】

增加设置 TNR、DEI、DBM 的卖场模式功能，观看 TNR、DEI、DBM 算法开关的对比效果。

【注意事项】

无。

【范例】

sample/pq/sample_pq_v2.c

HI_UNF_PQ_MODULE_E

【结构体定义】

修改前：

```
typedef enum hiUNF_PQ_MODULE_E
{
    HI_UNF_PQ_MODULE_SHARPNESS = 0,
```

```
    HI_UNF_PQ_MODULE_DCI,
    HI_UNF_PQ_MODULE_COLOR,
    HI_UNF_PQ_MODULE_SR,
   HI_UNF_PQ_MODULE_ALL,

    HI_UNF_PQ_MODULE_BUTT
}  HI_UNF_PQ_MODULE_E;
```

修改后：

```
typedef enum hiUNF_PQ_MODULE_E
{
   HI_UNF_PQ_MODULE_SHARPNESS = 0,
   HI_UNF_PQ_MODULE_DCI,
   HI_UNF_PQ_MODULE_COLOR,
   HI_UNF_PQ_MODULE_SR,
   HI_UNF_PQ_MODULE_TNR,
   HI_UNF_PQ_MODULE_ALL,

   HI_UNF_PQ_MODULE_BUTT
}  HI_UNF_PQ_MODULE_E;
```

【修改原因】

增加对 TNR 算法的开关控制。

【注意事项】

无。

【范例】

sample/pq/sample_pq_v2.c

12.5.3 函数接口

12.5.3.1 新增

HI_UNF_PQ_GetNR

【接口定义】

```
HI_S32 HI_UNF_PQ_GetNR(HI_UNF_DISP_E enChan, HI_U32* pu32NRLevel);
```

【新增原因】

获取降噪强度

【注意事项】

无

【范例】

sample/pq/sample_pq_v2.c

HI_UNF_PQ_SetNR

【接口定义】

```
HI_S32 HI_UNF_PQ_SetNR(HI_UNF_DISP_E enChan, HI_U32 u32NRLevel);
```

【新增原因】

设置降噪强度。

【注意事项】

降噪强度有效范围: 0~100。

【范例】

sample/pq/sample_pq_v2.c

HI_UNF_PQ_SetDemoMode

【接口定义】

```
extern HI_S32 HI_UNF_PQ_SetDemoMode( HI_UNF_DISP_E enChan,
HI_UNF_PQ_DEMO_MODE_E enMode);
```

【新增原因】

设置卖场模式的显示方式。

【注意事项】

无

【范例】

sample/pq/sample_pq_v2.c

HI_UNF_PQ_GetDemoMode

【接口定义】

```
extern HI_S32 HI_UNF_PQ_GetDemoMode( HI_UNF_DISP_E enChan,
HI_UNF_PQ_DEMO_MODE_E* penMode);
```

【新增原因】

获取卖场显示模式。

【注意事项】

无

【范例】

sample/pq/sample_pq_v2.c

HI_UNF_PQ_SetBasicImageParam

【接口定义】

```
xtern HI_S32 HI_UNF_PQ_SetBasicImageParam(HI_UNF_PQ_IMAGE_TYPE_E enType,
HI_UNF_DISP_E enChan, HI_UNF_PQ_IMAGE_PARAM_S stParam);
```

【新增原因】

单独设置图形或者视频的亮度、色调、对比度、饱和度。

【注意事项】

亮度、色调、对比度、饱和度的范围为 0 至 100。

【范例】

sample/pq/sample_pq_v2.c

HI_UNF_PQ_GetBasicImageParam

【接口定义】

```
extern HI_S32 HI_UNF_PQ_GetBasicImageParam(HI_UNF_PQ_IMAGE_TYPE_E enType,
HI_UNF_DISP_E enChan, HI_UNF_PQ_IMAGE_PARAM_S* pstParam);
```

【新增原因】

获取图形或者视频的亮度、色调、对比度、饱和度。

【注意事项】

无

【范例】

sample/pq/sample_pq_v2.c

12.6 Subtitle

12.6.1 概述

UNF3.2.10 相对于 UNF3.2.9 版本，Subtitle 模块在总体功能上有以下变更：

- 新增设置可接受的最大同步时间接口

12.6.1.1 新增设置可接受的最大同步时间接口

增加设置可接受的最大同步时间区间,用于处理字幕数据显示 PTS 同视频 PTS 相差过大情况,时间单位为 MS,引起变更如下

函数接口

新增 HI_UNF_SO_SetMaxInterval

12.6.2 函数接口

12.6.2.1 新增

HI_UNF_SO_SetMaxInterval

【接口定义】

```
HI_S32 HI_UNF_SO_SetMaxInterval(HI_HANDLE handle, HI_U32 u32IntervalMs )
```

【新增原因】

测试发现有部分特殊码流，字幕解析出来的显示 PTS 和视频的 PTS 相差太大，导致这种格式的字幕可能无法输出，增加接口设置最大偏差时间，若接收数据时间偏差超限，则立即输出

【注意事项】

无。

【范例】

无

12.7 Teletext

12.7.1 概述

UNF3.2.10 相对于 UNF3.2.9 版本，Teletext 模块在总体功能上有以下变更：

- 新增设置可接受的最大同步时间接口

12.7.1.1 新增设置可接受的最大同步时间接口

增加设置可接受的最大同步时间区间,用于处理图文数据显示 PTS 同视频 PTS 相差过大情况,时间单位为 MS,引起变更如下

数据结构

无

函数接口

新增 HI_UNF_TTX_SetMaxInterval

12.7.2 函数接口

12.7.2.1 新增

HI_UNF_TTX_SetMaxInterval

【接口定义】

```
HI_S32 HI_UNF_TTX_SetMaxInterval(HI_HANDLE hTTX, HI_U32 u32IntervalMs )
```

【新增原因】

Teletext 模块内部有一个默认的 10S 偏差值，即待显示的 PTS 和视频的 PTS 相差大于 10S 则立即输出，处理相对僵化，存在能同步成功的情况下数据立即输出了，与期望不符，因此新增接口供上层灵活设置同步偏差。

【注意事项】

无。

【范例】

无

12.8 Frontend

12.8.1 概述

UNF 3.2.10 相对于 UNF 3.2.9 版本，Frontend 模块在总体功能上有以下变更：

- 增加 MXL251，HIFDVBC100 和 HIFJ83B100 器件的支持
- 增加获取全频段频谱数据接口

12.8.1.1 增加 MXL251，HIFDVBC100 和 HIFJ83B100 器件支持

增加 MXL251，HIFDVBC100 和 HIFJ83B100 器件支持，引起的变更如下：

数据结构

- 修改 HI_UNF_TUNER_DEV_TYPE_E
- 修改 HI_UNF_DEMOD_DEV_TYPE_E

12.8.1.2 增加获取全频段频谱数据接口

增加获取全频段频谱数据接口，引起的变更如下：

函数接口

新增 HI_UNF_TUNER_GetTunerPowerSpectrumData

12.8.2 数据结构

12.8.2.1 修改

HI_UNF_TUNER_DEV_TYPE_E

【结构体定义】

修改前：

```
typedef enum    hiUNF_TUNER_DEV_TYPE_E
{
    HI_UNF_TUNER_DEV_TYPE_XG_3BL,
    HI_UNF_TUNER_DEV_TYPE_CD1616,
    HI_UNF_TUNER_DEV_TYPE_ALPS_TDAE,
    HI_UNF_TUNER_DEV_TYPE_TDCC,
    HI_UNF_TUNER_DEV_TYPE_TDA18250,
    HI_UNF_TUNER_DEV_TYPE_CD1616_DOUBLE,
    HI_UNF_TUNER_DEV_TYPE_MT2081,
    HI_UNF_TUNER_DEV_TYPE_TMX7070X,
    HI_UNF_TUNER_DEV_TYPE_R820C,
    HI_UNF_TUNER_DEV_TYPE_MXL203,
    HI_UNF_TUNER_DEV_TYPE_AV2011,
    HI_UNF_TUNER_DEV_TYPE_SHARP7903,
    HI_UNF_TUNER_DEV_TYPE_MXL101,
    HI_UNF_TUNER_DEV_TYPE_MXL603,
    HI_UNF_TUNER_DEV_TYPE_IT9170,
    HI_UNF_TUNER_DEV_TYPE_IT9133,
    HI_UNF_TUNER_DEV_TYPE_TDA6651,
    HI_UNF_TUNER_DEV_TYPE_TDA18250B,
    HI_UNF_TUNER_DEV_TYPE_M88TS2022,
    HI_UNF_TUNER_DEV_TYPE_RDA5815,
    HI_UNF_TUNER_DEV_TYPE_MXL254,
    HI_UNF_TUNER_DEV_TYPE_CXD2861,
    HI_UNF_TUNER_DEV_TYPE_SI2147,
    HI_UNF_TUNER_DEV_TYPE_RAFAEL836,
    HI_UNF_TUNER_DEV_TYPE_MXL608,
```

```
    HI_UNF_TUNER_DEV_TYPE_MXL214,
    HI_UNF_TUNER_DEV_TYPE_TDA18280,
    HI_UNF_TUNER_DEV_TYPE_TDA182I5A,
    HI_UNF_TUNER_DEV_TYPE_SI2144,
    HI_UNF_TUNER_DEV_TYPE_AV2018,
    HI_UNF_TUNER_DEV_TYPE_BUTT
} HI_UNF_TUNER_DEV_TYPE_E ;
```

修改后：

```
typedef enum    hiUNF_TUNER_DEV_TYPE_E
{
    HI_UNF_TUNER_DEV_TYPE_XG_3BL,
    HI_UNF_TUNER_DEV_TYPE_CD1616,
    HI_UNF_TUNER_DEV_TYPE_ALPS_TDAE,
    HI_UNF_TUNER_DEV_TYPE_TDCC,
    HI_UNF_TUNER_DEV_TYPE_TDA18250,
    HI_UNF_TUNER_DEV_TYPE_CD1616_DOUBLE,
    HI_UNF_TUNER_DEV_TYPE_MT2081,
    HI_UNF_TUNER_DEV_TYPE_TMX7070X,
    HI_UNF_TUNER_DEV_TYPE_R820C,
    HI_UNF_TUNER_DEV_TYPE_MXL203,
    HI_UNF_TUNER_DEV_TYPE_AV2011,
    HI_UNF_TUNER_DEV_TYPE_SHARP7903,
    HI_UNF_TUNER_DEV_TYPE_MXL101,
    HI_UNF_TUNER_DEV_TYPE_MXL603,
    HI_UNF_TUNER_DEV_TYPE_IT9170,
    HI_UNF_TUNER_DEV_TYPE_IT9133,
    HI_UNF_TUNER_DEV_TYPE_TDA6651,
    HI_UNF_TUNER_DEV_TYPE_TDA18250B,
    HI_UNF_TUNER_DEV_TYPE_M88TS2022,
    HI_UNF_TUNER_DEV_TYPE_RDA5815,
    HI_UNF_TUNER_DEV_TYPE_MXL254,
    HI_UNF_TUNER_DEV_TYPE_CXD2861,
    HI_UNF_TUNER_DEV_TYPE_SI2147,
```

```
    HI_UNF_TUNER_DEV_TYPE_RAFAEL836,
    HI_UNF_TUNER_DEV_TYPE_MXL608,
    HI_UNF_TUNER_DEV_TYPE_MXL214,
    HI_UNF_TUNER_DEV_TYPE_TDA18280,
    HI_UNF_TUNER_DEV_TYPE_TDA182I5A,
    HI_UNF_TUNER_DEV_TYPE_SI2144,
    HI_UNF_TUNER_DEV_TYPE_AV2018,
    HI_UNF_TUNER_DEV_TYPE_MXL251,
    HI_UNF_TUNER_DEV_TYPE_BUTT
} HI_UNF_TUNER_DEV_TYPE_E ;
```

【修改原因】

新增 Tuner 器件支持

【注意事项】

MXL251 为 FullBand Tuner 器件，Demod 和 Tuner 一体

【范例】

无

HI_UNF_DEMOD_DEV_TYPE_E

【结构体定义】

修改前：

```
typedef enum    hiUNF_DEMOD_DEV_TYPE_E
{
    HI_UNF_DEMOD_DEV_TYPE_NONE,
    HI_UNF_DEMOD_DEV_TYPE_3130I = 0x100,
    HI_UNF_DEMOD_DEV_TYPE_3130E,
    HI_UNF_DEMOD_DEV_TYPE_J83B,
    HI_UNF_DEMOD_DEV_TYPE_AVL6211,
    HI_UNF_DEMOD_DEV_TYPE_MXL101,
    HI_UNF_DEMOD_DEV_TYPE_MN88472,
    HI_UNF_DEMOD_DEV_TYPE_IT9170,
    HI_UNF_DEMOD_DEV_TYPE_IT9133,
    HI_UNF_DEMOD_DEV_TYPE_3136,
```

```
    HI_UNF_DEMOD_DEV_TYPE_3136I,
    HI_UNF_DEMOD_DEV_TYPE_MXL254,
    HI_UNF_DEMOD_DEV_TYPE_CXD2837,
    HI_UNF_DEMOD_DEV_TYPE_3137,
    HI_UNF_DEMOD_DEV_TYPE_MXL214,
    HI_UNF_DEMOD_DEV_TYPE_TDA18280,
    HI_UNF_DEMOD_DEV_TYPE_HIFDVBC100,
    HI_UNF_DEMOD_DEV_TYPE_HIFJ83B100,
    HI_UNF_DEMOD_DEV_TYPE_BUTT
} HI_UNF_DEMOD_DEV_TYPE_E;
```

修改后：

```
typedef enum    hiUNF_DEMOD_DEV_TYPE_E
{
    HI_UNF_DEMOD_DEV_TYPE_NONE,
    HI_UNF_DEMOD_DEV_TYPE_3130I = 0x100,
    HI_UNF_DEMOD_DEV_TYPE_3130E,
    HI_UNF_DEMOD_DEV_TYPE_J83B,
    HI_UNF_DEMOD_DEV_TYPE_AVL6211,
    HI_UNF_DEMOD_DEV_TYPE_MXL101,
    HI_UNF_DEMOD_DEV_TYPE_MN88472,
    HI_UNF_DEMOD_DEV_TYPE_IT9170,
    HI_UNF_DEMOD_DEV_TYPE_IT9133,
    HI_UNF_DEMOD_DEV_TYPE_3136,
    HI_UNF_DEMOD_DEV_TYPE_3136I,
    HI_UNF_DEMOD_DEV_TYPE_MXL254,
    HI_UNF_DEMOD_DEV_TYPE_CXD2837,
    HI_UNF_DEMOD_DEV_TYPE_3137,
    HI_UNF_DEMOD_DEV_TYPE_MXL214,
    HI_UNF_DEMOD_DEV_TYPE_TDA18280,
    HI_UNF_DEMOD_DEV_TYPE_HIFDVBC100,
    HI_UNF_DEMOD_DEV_TYPE_HIFJ83B100,
    HI_UNF_DEMOD_DEV_TYPE_MXL251,
    HI_UNF_DEMOD_DEV_TYPE_BUTT
```

```
} HI_UNF_DEMOD_DEV_TYPE_E;
```

【修改原因】

新增 Demod 器件支持

【注意事项】

MXL251 为 FullBand Tuner 器件，Demod 和 Tuner 一体，HIFDVBC100 和 HIFJ83B100 为芯片内置 Demod

【范例】

无

12.8.3 函数接口

12.8.3.1 新增

HI_UNF_TUNER_GetTunerPowerSpectrumData

【接口定义】

```
HI_S32 HI_UNF_TUNER_GetTunerPowerSpectrumData(HI_U32 u32TunerId, HI_U32
u32freqStartInHz,HI_U32 u32freqStepInHz,HI_U32 u32numOfFreqSteps,HI_S16
*ps16powerData);
```

【新增原因】

增加获取全频段频谱数据功能

【注意事项】

无

【范例】

无