全志科技
Allwinner Technology

音视频解码库项目

Parser 库 API 文档

文档履历

版本号	日期	制/修订人	内容描述
V0.1	2014-02-14		初稿
V0.2	2014-09-01		修订 1、修改文档题目为《Parser 库概要设计报告》 2、删除 V1.0 中 2.3. 模块配置介绍 3、删除 V1.0 中 3. 模块体系结构设计 4、修改 V1.0 中 4. 接口和流程设计 为 V2.0 中的 3. 接口函数 5、新增 3.1 AwParserRegister 6、新增对 CdxParserPrepare 中参数 flags 的说明，包括 MUTIL_AUDIO 等 7、删除 V1.0 中 4.1.8. CdxParserForceStop，在 CdxParserControl 的控制命令中增加 CDX_PSR_CMD_SET_FORCESTOP，CDX_PSR_CMD_CLR_FORCESTOP,另外对 forcestop 要求与 V1.0 不同 8、新增 4.1 struct CdxParserS 4.4 enum CdxParserTypeE 4.5 enum CdxParserCommandE 4.6 enum CdxParserStatusE 4.10 struct ParserCacheStateS 4.11struct ParserUriKeyInfoS 修改 struct CdxMediaInfoS 9、新增 5. 代码样例
V0.3	2015-05-28		1. 修改 CdxParserPrepare 2. 修改 struct CdxParserCreatorS 3. 增加 3.11 CdxParserInit
V1.0	2015-05-28		Release 版
V1.1	2015-09-01		修改 4.8. struct CdxProgramS 结构体的 duration 属性
	2015-09-08		添加 4.5.CdxParserCommandE CDX_PSR_CMD_CLR_INFO 命令
V1.2	2016-11-04		添加 4.4 CdxParserTypeE 结构体成员 CDX_PARSER_SSTR, CDX_PARSER_CAF, CDX_PARSER_G729, CDX_PARSER_DSD, CDX_PARSER_AIFF, CDX_PARSER_ID3, CDX_PARSER_ENV, CDX_PARSER_AWRAWSTREAM, CDX_PARSER_AWSPECIALSTREAM; 添加 4.5 CdxParserCommandE 结构体成员中包含的命令：CDX_PSR_CMD_SET_SECURE_BUFFER_COUNT, CDX_PSR_CMD_SET_SECURE_BUFFERS, CDX_PSR_CMD_GET_STREAM_EXTRADATA, CDX_PSR_CMD_GET_REDIRECT_URL, CDX_PSR_CMD_GET_URL, CDX_PSR_CMD_SET_TIMESHIFT_LAST_SEQNUM; 添加 4.7 CdxMediaInfoS 结构体成员变量：albumsz,albumCharEncode 等;

			添加 4.8 CdxProgramS 结构体成员变量：firstPts； 添加 4.9 CdxPacketS 结构体成员指针变量：info。

目 录

音视频解码库项目.....1
Parser 库 API 文档.....1
1. 概述.....- 1 -
1.1. 编写目的.....- 1 -
1.2. 适用范围.....- 1 -
1.3. 相关人员.....- 1 -
2. 模块介绍.....- 2 -
2.1. 功能介绍.....- 2 -
2.2. 相关术语介绍.....- 2 -
2.3. 源码结构介绍.....- 2 -
3. 接口函数.....- 3 -
3.1. AwParserRegister.....- 3 -
3.2. CdxParserPrepare.....- 3 -
3.3. CdxParserGetMediaInfo.....- 4 -
3.4. CdxParserPrefetch.....- 5 -
3.5. CdxParserRead.....- 5 -
3.6. CdxParserSeekTo.....- 5 -
3.7. CdxParserControl.....- 6 -
3.8. CdxParserAttribute.....- 6 -
3.9. CdxParserGetStatus.....- 6 -
3.10. CdxParserClose.....- 6 -
3.11. CdxParserInit.....- 6 -
4. 数据结构设计.....- 7 -
4.1. struct CdxParserS.....- 7 -
4.2. struct CdxParserOpsS.....- 7 -
4.3. struct CdxParserCreatorS.....- 8 -
4.4. enum CdxParserTypeE.....- 9 -
4.5. enum CdxParserCommandE.....- 10 -
4.6. enum CdxParserStatusE.....- 11 -
4.7. struct CdxMediaInfoS.....- 11 -
4.8. struct CdxProgramS.....- 12 -
4.9. struct CdxPacketS.....- 13 -
4.10. struct ParserCacheStateS.....- 14 -
4.11. struct ParserUriKeyInfoS.....- 14 -
5. 代码样例.....- 15 -
6. Declaration.....- 18 -

1. 概述

1.1. 编写目的

设计播放器对 Parser 库的访问接口，指导具体 Parser 的开发、使用和后续维护。

1.2. 适用范围

A80/A83/H3/H8 等各芯片平台的 Android 系统 SDK 和 Linux SDK。

1.3. 相关人员

Parser 库的使用者，Parser 的开发和维护者。

2. 模块介绍

2.1. 功能介绍

各封装格式的具体 Parser 按照接口特性描述，实现本文档所规定的接口，完成对特定封装格式文件的解封装。本项目已实现的 Parser 类型包括：ASF、TS、AVI、FLV、MKV、MOV、DASH、RTSP、HLS、BD、PMP、OGG、MPG、MMS、MMSHTTP、M3U9、PLAYLIST、MP3、APE、FLAC、AMR、ATRAC、AWTS、REMUX 等。同时提供注册接口，可将新封装格式的 Parser 加入该库，以资扩展。

2.2. 相关术语介绍

Stream：流媒体文件，它提供播放的对象，同时也是 Parser 解封装的对象。

Parser：媒体文件解析器，完成媒体文件的解封装。

2.3. 源码结构介绍

```
PARSER/
├── Android.mk
├── config.mk   //全局的配置文件，子目录以下的 Android.mk 都需要包含该文件
├── include
│   └── CdxParser.h   //本模块对外的头文件
├── base
│   └── CdxParser.c   //实例化一个 Parser
├── asf
│   └── ……  //ASF Parser 实现代码
├── flv
│   └── ……  //FLV Parser 实现代码
├── mms
│   └── ……  //MMS Parser 实现代码
├── mov
│   └── ……  //MOV Parser 实现代码
├── remux
│   └── ……  //RTSP Parser 实现代码
├── ts
│   └── ……  //TS Parser 实现代码

    ……  //省略 其它 Parser
```

3. 接口函数

注册函数	
AwParserRegister	注册一个 Parser（将该 Parser 加入 Parser 库）
同步控制函数	
CdxParserPrepare	创建一个 Parser 对象
CdxParserGetMediaInfo	获取媒体信息
CdxParserPrefetch	探测下一笔待读数据的信息（包含类型、大小、pts 等）
CdxParserRead	从 Parser 对象读取一笔数据
CdxParserSeekTo	Seek 到给定的时间点
CdxParserControl	对 Parser 发控制命令
CdxParserAttribute	获取 Parser 的属性信息
CdxParserGetStatus	获取 Parser 当前的操作状态（OK、IO_ERR、EOS 等）
CdxParserClose	关闭 Parser，释放资源

当有新封装格式的 Parser 需要加入该库时，应在 AwParserInit 中调用注册函数 AwParserRegister，以扩展 Parser 库所支持的封装格式。

需要对一个媒体文件解封装，应先调用 CdxParserPrepare，该函数中会打开 stream，对 stream 做 probe 以确定 parser 类型，并打开该 parser 返回 parser 的句柄；拿到 prepare 返回的 parser 句柄，可以调用 CdxParserGetMediaInfo 获取对应的媒体信息，一般用于初始化解码器等操作；从 parser 读取一笔数据之前，需要先调用 CdxParserPrefetch，探测下一笔待读数据的数据类型，长度等信息，依据该信息从相应解码器申请对应长度的空间，然后调用 CdxParserRead 将数据读到申请到的空间。

注：对本文档涉及的结构体 CdxParserT、CdxParserCreatorT、CdxParserTypeT、CdxMediaInfoT、CdxPacketT 有下表定义：

typedef struct CdxParserS CdxParserT;
typedef struct CdxParserCreatorS CdxParserCreatorT;
typedef enum CdxParserTypeE CdxParserTypeT;
typedef struct CdxMediaInfoS CdxMediaInfoT;
typedef struct CdxPacketS CdxPacketT;

3.1. AwParserRegister

函数原型	int AwParserRegister(CdxParserCreatorT *creator, CdxParserTypeT type, struct ParserUriKeyInfoS *keyInfo);
功能	注册一个 Parser（将该 Parser 加入 Parser 库）
参数	creator：Parser 的构造器 type：Parser 的类型 keyInfo：Parser 的标志性信息（如文件后缀，属性等）
返回值	成功：返回 0 失败：返回错误码
调用说明	当有新封装格式的 Parser 需要加入该库时，应调用该函数。

3.2. CdxParserPrepare

函数原型	int CdxParserPrepare(CdxDataSourceT *source, cdx_uint32 flags, pthread_mutex_t *mutex, cdx_bool *exit, CdxParserT **parser, CdxStreamT **stream, ContorlTask *parserTasks, ContorlTask *streamTasks)
功能	打开一个 Parser 对象
参数	source ： 包含流媒体文件对应的 URI 信息和其他头部信息 flags ： 表明对 Parser 的属性要求，由 MUTIL_AUDIO/DISABLE_AUDIO 等组成 MUTIL_AUDIO：节目中的所有音轨均会解析。 MUTIL_SUBTITLE：节目中的所有字幕均会解析 DISABLE_AUDIO：不解析音频 DISABLE_VIDEO：不解析视频 DISABLE_SUBTITLE：不解析字幕 NO_NEED_DURATION：不需要 parser 获取 duration，例如 hls parser 不需要 ts parser 去获取分片的时长 MIRACST：指定该媒体文件的来源是 miracast BD_BASE：指定该媒体文件是一个 bd 主流文件，在 bd 和 ts parser 间使用 BD_DEPENDENT：指定该媒体文件是一个 bd 从流文件，在 bd 和 ts parser 间使用 BD_TXET：指定该媒体文件是一个 bd 文本字幕文件，在 bd 和 ts parser 间使用 SEGMENT_MP4:指定该媒体文件是 ISO 多个分片形式的 mov 文件，目前用于 dash、smooth streaming 与 mov parser 之间 NO_NEED_CLOSE_STREAM：指定该 parser 不需要 close stream，由上一层 parser 来关闭 stream，目前用于 mms playlist 和 asf parser 之间 mutex：同步用的锁 exit ： 退出标识 parser：创建的 parser 句柄 Stream：创建的 stream 句柄 parserTasks：需要在 CdxParserCreate 和 CdxParserInit 之间执行的 parser 的 control 命令 streamTasks：需要在 CdxStreamCreate 和 CdxStreamConnect 之间执行的 stream 的 control 命令
返回值	成功：0 失败：-1
调用说明	1、阻塞的函数，把*exit 置 TRUE 即可退出 2、flags 是对 Parser 的属性要求，但 parser 具体实现了哪些属性应调用 CdxParserAttribute 获知

3.3. CdxParserGetMediaInfo

函数原型	cdx_int32 CdxParserGetMediaInfo(CdxParserT *parser, CdxMediaInfoT *mediaInfo)

功能	获取媒体信息，包括音频、视频、字幕等信息
参数	parser ： 句柄 mediaInfo ： 媒体信息，由调用者申请空间
返回值	成功：返回 0 失败：返回错误码
调用说明	关于 ID3 信息： mediaInfo 里携带的 ID3 信息（有些 parser 会解析出此信息，如 mp3），Parser 解析出裸数据即可，字符集之间的转换放到中间件层进行处理，如：parser 解析出来的 id3 信息可能是 UTF-16 格式的字符，而显示要求的是 UTF-8，在这种场景下，中间件层如 awmetadataretriever 会将 UTF-16 转换成 UTF-8，parser 不需要作此处理。

3.4. CdxParserPrefetch

函数原型	cdx_int32 CdxParserPrefetch(CdxParserT *parser, CdxPacketT *pkt)
功能	探测下一笔待读数据的信息（包含类型、大小、pts 等）
参数	Parser ： 句柄 Pkt ： 数据包信息，由调用者申请空间
返回值	成功：0 失败：返回错误码
调用说明	1.成功时会向 pkt 填写该笔数据的信息，但并不会把数据拷贝到 pkt 中 2.阻塞函数，可调用 forceStop 终止执行，其中 forceStop 有如下定义 #define CdxParserForceStop(parser) \ (CdxParserControl(parser, CDX_PSR_CMD_SET_FORCESTOP, NULL)) 其中 CDX_PSR_CMD_SET_FORCESTOP 详见于 enum CdxParserCommandE

3.5. CdxParserRead

函数原型	cdx_int32 CdxParserRead(CdxParserT *parser, CdxPacketT *pkt)
功能	从 Parser 读取一笔数据
参数	Parser ： 句柄 Pkt ： 数据包信息
返回值	成功：返回 0 失败：返回错误码
调用说明	阻塞函数，可调用 forceStop 终止执行，其中 forceStop 参见 CdxParserPrefetch 的调用说明

3.6. CdxParserSeekTo

函数原型	cdx_int32 CdxParserSeekTo(CdxParserT *parser, cdx_int64 timeUs)
功能	Seek 到指定的时间点
参数	Parser ： 句柄 timeUs ： 以微秒为单位的播放进度时间
返回值	成功：返回 0 失败：返回错误码
调用说明	1、阻塞函数，可调用 forceStop 终止执行，其中 forceStop 参见 CdxParserPrefetch 的调用说明

	2、调用该函数后，之前 prefetch 的结果会无效，需要重新 prefetch

3.7. CdxParserControl

函数原型	cdx_int32 CdxParserControl(CdxParserT *parser, cdx_int32 cmd, void *param)
功能	向 parser 发控制命令
参数	Parser ： 句柄 Cmd ： 发送命令 Param ： 命令参数
返回值	成功：返回 0 失败：返回错误码
调用说明	所支持的命令由 enum CdxParserCommandE 定义，Param 的具体含义因 Cmd 而不同

3.8. CdxParserAttribute

函数原型	cdx_uint32 CdxParserAttribute(CdxParserT *parser)
功能	获取 parser 的属性信息
参数	Parser ： 句柄
返回值	返回 32 位整型
调用说明	具体位的含义由 MUTIL_AUDIO 等定义，参见 CdxParserPrepare 的 flags 参数

3.9. CdxParserGetStatus

函数原型	cdx_int32 CdxParserGetStatus(CdxParserT *parser)
功能	获取 parser 的状态，查询是否可操作，是否 IO 错误，是否 EOS 等
参数	Parser：句柄
返回值	返回 32 位整型，含义由 enum CdxParserStatusE 定义
调用说明	NA

3.10. CdxParserClose

函数原型	cdx_int32 CdxParserClose(CdxParserT *parser)
功能	关闭 parser，释放资源
参数	Parser ： 句柄
返回值	成功：返回 0 失败：返回错误码
调用说明	NA

3.11. CdxParserInit

函数原型	cdx_int32 CdxParserInit(CdxParserT *parser)
功能	初始化 parser
参数	Parser ： 句柄
返回值	成功：返回 0 失败：-1
调用说明	NA

4. 数据结构设计

4.1. struct CdxParserS

名称	struct CdxParserS		
功能描述	Parser 的结构体		
属性	类型	初始值	描述
type	enum CdxParserTypeE	CDX_PARSER_UNKNOW	Parser 的类型
ops	struct CdxParserOpsS *	NULL	Parser 的操作函数集合

4.2. struct CdxParserOpsS

名称	struct CdxParserOpsS	
功能描述	parser 的接口定义，是其操作函数集合	
方法		
control	原型	cdx_int32 (*control)(CdxParserT *parser , cdx_int32 cmd, void * param);
	参数	parser ： 句柄 cmd ： 命令，由 enum CdxParserCommandE 定义 param ： 命令参数
	返回值	成功：返回 0 失败：返回错误码
	功能	向 parser 发控制命令
prefetch	原型	cdx_int32 (*prefetch)(CdxParserT *parser, CdxPacketT *pkt);
	参数	parser ： 句柄 pkt ： 媒体数据结构体
	返回值	成功：返回 0 失败：返回错误码
	功能	探测下一笔待读数据的信息（包含类型、大小、pts 等）
read	原型	cdx_int32 (*read)(CdxParserT *parser, CdxPacketT *pkt);
	参数	parser ： 句柄 pkt ： 媒体数据结构体
	返回值	成功：返回 0 失败：返回错误码
	功能	从 parser 读取一笔数据
getMediaInfo	原型	cdx_int32 (*getMediaInfo)(CdxParserT *parser, CdxMediaInfoT *mediaInfo);
	参数	parser：句柄 mediaInfo：媒体信息，调用者申请空间
	返回值	成功：返回 0

		失败：返回错误码
	功能	获取媒体信息，包括音频、视频、字幕等信息
seekTo	原型	cdx_int32 (*seekTo)(CdxParserT *parser, cdx_int64 timeUs);
	参数	parser：句柄 timeUs：播放进度时间,单位微秒
	返回值	成功：返回 0 失败：返回错误码
	功能	Seek 到指定的时间点
attribute	原型	cdx_uint32 (*attribute)(CdxParserT *parser);
	参数	parser：句柄
	返回值	返回代表该 parser 属性的 flags，参见 CdxParserPrepare 的 flags 参数
	功能	查询 parser 的属性信息
getStatus	原型	cdx_int32 (*getStatus)(CdxParserT *parser);
	参数	parser：句柄
	返回值	返回 parser 当前的操作状态，为 32 位整型，由 enum CdxParserStatusE 定义
	功能	查询 parser 当前的操作状态
close	原型	cdx_int32 (*close)(CdxParserT *parser);
	参数	parser：句柄
	返回值	成功：返回 0 失败：返回错误码
	功能	关闭句柄，释放资源
init	原型	cdx_int32 (*init)(CdxParserT *parser)
	参数	parser：句柄
	返回值	成功：返回 0 失败：-1
	功能	初始化 parser

4.3. struct CdxParserCreatorS

名称	struct CdxStreamCreatorS	
功能描述	负责 parser 构造的结构体	
方法		
probe	原型	cdx_uint32 (*probe)(CdxStreamProbeDataT *data);/*return score(0-100)*/
	参数	data：待探测的数据
	返回值	成功：返回 100 失败：0
	功能	探测 stream 对应的 parser 类型
create	原型	CdxParserT *(*create)(CdxStreamT *stream, cdx_uint32 flags);
	参数	stream ： 已打开的 stream flags：对 parser 的属性要求，由 MUTIL_AUDIO 等定义，参

		见 CdxParserPrepare 的 flags 参数
	返回值	成功：返回 parser 句柄 失败：返回 NULL
	功能	创建一个特定 parser。

4.4. enum CdxParserTypeE

名称	enum CdxParserTypeE
功能	parser 的类型
取值	描述
CDX_PARSER_UNKNOW	parser 类型未知
CDX_PARSER_MOV	mov parser
CDX_PARSER_MKV	mkv parser
CDX_PARSER_ASF	asf parser
CDX_PARSER_TS	ts parser
CDX_PARSER_AVI	avi parser
CDX_PARSER_FLV	flv parser
CDX_PARSER_PMP	pmp parser
CDX_PARSER_HLS	hls parser
CDX_PARSER_DASH	dash parser
CDX_PARSER_MMS	mms parser
CDX_PARSER_BD	bd parser
CDX_PARSER_OGG	ogg parser
CDX_PARSER_M3U9	m3u9 parser
CDX_PARSER_PLAYLIST	playlist parser
CDX_PARSER_APE	ape parser
CDX_PARSER_FLAC	flac parser
CDX_PARSER_AMR	amr parser
CDX_PARSER_ATRAC	atrac parser
CDX_PARSER_MP3	mp3 parser
CDX_PARSER_AAC	aac parser
CDX_PARSER_WAV	wav parser
CDX_PARSER_REMUX	remux parser
CDX_PARSER_WVM	wvm parser
CDX_PARSER_MPG	mpg parser
CDX_PARSER_MMSHTTP	mmshttp parser
CDX_PARSER_AWTS	awts parser
CDX_PARSER_SSTR	sstr parser
CDX_PARSER_CAF	caf parser
CDX_PARSER_G729	g729 parser
CDX_PARSER_DSD	dsd parser
CDX_PARSER_AIFF	aiff parser
CDX_PARSER_ID3	id3 parser
CDX_PARSER_ENV	env parser

CDX_PARSER_AWRAWSTREAM	AwRawStream parser
CDX_PARSER_AWSPECIALSTREAM	AwSpecialStream parser

4.5. enum CdxParserCommandE

名称	enum CdxParserCommandE
功能	parser 的控制命令
取值	描述
CDX_PSR_CMD_SWITCH_AUDIO	切音轨。注：这是 parser 层的处理，不同于无缝切换
CDX_PSR_CMD_SWITCH_SUBTITLE	切字幕。注：同上
CDX_PSR_CMD_DISABLE_AUDIO	不解析音频
CDX_PSR_CMD_DISABLE_VIDEO	不解析视频
CDX_PSR_CMD_DISABLE_SUBTITLE	不解析字幕
CDX_PSR_CMD_SET_DURATION	设置 parser 的时长，此时时长不是由该 parser 自己获取，而是由上层设定。例如，hls 分片是一个 ts 文件，该 ts 文件的时长可以由 hls 进行设定
CDX_PSR_CMD_REPLACE_STREAM	替换 parser 的解析对象（即替换流媒体文件 stream）。 例如，hls parser 的下一级是 ts parser，当某 ts 分片解析完成后，hls parser 可调该命令用新的 ts 分片替换 ts parser 的原有分片，从而不关闭 ts parser，继续解析新的分片。
CDX_PSR_CMD_STREAM_SEEK	控制 parser seek 到解析对象（stream）的特定位置
CDX_PSR_CMD_GET_CACHESTATE	获取 cache 状态，其含义由 struct ParserCacheStateS 定义
CDX_PSR_CMD_SET_FORCESTOP	设置 parser 强制退出（forcestop）。该控制命令的目的是使 parser 从一个阻塞操作中迅速退出，即终止阻塞函数的执行，通常用于跳播、退出播放等。其实现要求是如果没有调用 CDX_PSR_CMD_CLR_FORCESTOP 控制命令，则 forcestop 一直有效，此时若调用 CdxParserPrefetch、CdxParserSeekTo 等阻塞函数，则该函数可能未能有效执行，因为该函数会被 forcestop 所终止。
CDX_PSR_CMD_CLR_FORCESTOP	清除 parser 的强制退出标记
CDX_PSR_CMD_SET_CALLBACK	为 parser 设置回调
CDX_PSR_CMD_UPDATE_PARSER	更新 parser。例如，当 hls 有两级时，master hls parser 切换带宽后需要调用该命令以更新 media hls parser
CDX_PSR_CMD_SET_HDCP	为 parser 设置 HDCP
CDX_PSR_CMD_CLR_INFO	清除信息。目前只有 HLS 用，用于清除 TS 子 parser 的节目信息。

CDX_PSR_CMD_SET_LASTSEGMENT_FLAG	设置 HLS 最后一个分片的标记。
CDX_PSR_CMD_SET_SECURE_BUFFER_COUNT	设置 WVM parser 使用的安全 buffer 个数。
CDX_PSR_CMD_SET_SECURE_BUFFERS	设置 WVM parser 使用的安全 buffer 指针。
CDX_PSR_CMD_GET_STREAM_EXTRADATA	获取 stream 层头域信息。
CDX_PSR_CMD_GET_REDIRECT_URL	获取重定向的 url。
CDX_PSR_CMD_GET_URL	获取 url。
CDX_PSR_CMD_SET_TIMESHIFT_LAST_SEQNUM	获取时移最后一片的分片号。

4.6. enum CdxParserStatusE

名称	enum CdxParserStatusE
功能	描述 parser 的操作状态
取值	描述
PSR_INVALID	表示 parser 对文件的初始解析尚未完成，即处于准备阶段，此时对 prefetch、read、getMediaInfo、seekto 等函数的调用是无效的
PSR_OK	表示 parser 已准备完成，且当前操作没有错误发生
PSR_OPEN_FAIL	打开失败
PSR_IO_ERR	发生 IO 错误
PSR_USER_CANCEL	操作被用户放弃
PSR_INVALID_OPERATION	无效操作，如无效的入参、parser 当前无法进行该操作等
PSR_UNKNOWN_ERR	未知错误
PSR_EOS	已解析完媒体数据

4.7. struct CdxMediaInfoS

名称	struct CdxMediaInfoS		
功能描述	描述媒体信息，视频、音频、字幕等信息		
属性	类型	初始值	描述
fileSize	cdx_int64	-1	媒体文件的大小
bitrate	cdx_uint32	0	媒体文件的比特率
bSeekable	cdx_bool	CDX_FALSE	是否支持 seek
programNum	cdx_int32	0	文件包含的节目数
programIndex	cdx_int32	0	当前所选节目的索引
program[PROGRAM_LIMIT]	struct CdxProgramS	N/A	所选节目的信息，目前 #define PROGRAM_LIMIT 1
privData	cdx_void *	NULL	其它信息，可留作扩展，目前没有使用，
album[64]	cdx_uint8	元素全 0	专辑名称
albumsz	cdx_int32	0	专辑字符串长度
albumCharEncode	cdx_int32	0	专辑字符串编码类型
artist[64]	cdx_uint8	元素全 0	艺术家（通常是歌曲演唱者）
author[64]	cdx_uint8	元素全 0	作者

authorsz	cdx_int32	0	作者字符串长度
authorCharEncode	cdx_int32	0	作者字符编码类型
composer[64]	cdx_uint8	元素全 0	作曲者
date[64]	cdx_uint8	元素全 0	出版日期
genre[64]	cdx_uint8	元素全 0	歌曲风格类型
genresz	cdx_int32	0	歌曲风格类型字符串长度
genreCharEncode	cdx_int32	0	歌曲风格类型字符编码类型
title[64]	cdx_uint8	元素全 0	标题
titlesz	cdx_int32	0	标题字符串长度
titleCharEncode	cdx_int32	0	标题字符串编码类型
year[64]	cdx_uint8	元素全 0	出版年份
yearsz	cdx_int32	0	出版年份字符长度
yearCharEncode	cdx_int32	0	出版年份字符编码类型
writer[64]	cdx_uint8	元素全 0	词作者
albumArtist[64]	cdx_uint8	元素全 0	专辑艺术家
compilation[64]	cdx_uint8	元素全 0	编辑
location[64]	cdx_uint8	元素全 0	拍摄视频者所在地理位置信息
rotate[4]	cdx_uint8	元素全 0	拍摄旋转角度
discNumber	cdx_int32	0	光盘号
pAlbumArtBuf	cdx_uint8 *	NULL	专辑图像封面数据缓冲
nAlbumArtBufSize	cdx_int32	0	专辑图像封面数据缓冲长度

4.8. struct CdxProgramS

名称	struct CdxProgramS		
功能描述	描述节目信息		
属性	类型	初始值	描述
id	cdx_int32	0	节目 ID
flags	cdx_uint32	0	节目的属性信息
duration	cdx_int32	0	节目时长，直播时该值应为-1
audioNum	cdx_int32	0	节目中 Audio 的数目
audioIndex	cdx_int32	0	当前所选 Audio 的索引
videoNum	cdx_int32	0	节目 Video 的数目
videoIndex	cdx_int32	0	当前所选 Video 的索引
subtitleNum	cdx_int32	0	节目 Subtitle 的数目
subtitleIndex	cdx_int32	0	当前所选 subtitle 的索引
firstPts	cdx_int64	0	当前节目的第一个 pts
audio[AUDIO_STREAM_LIMIT]	AudioStreamInfo	N/A	Audio 的相关信息，如编码格式、声道数、采样率等，详见于《音频解码库概要设计报告》。其中#define AUDIO_STREAM_LIMIT 32
video[VIDEO_STREAM_L	VideoStreamInf	N/A	Video 的相关信息，如编码格式、视

IMIT]	o		频宽、高等，详见《音频解码库概要设计报告》。其中#define AUDIO_STREAM_LIMIT 32
subtitle[SUBTITLE_STREAM_LIMIT]	SubtitleStreamInfo	N/A	Subtitle 的相关信息，如编码格式、语言等，详见《音频解码库概要设计报告》。其中#define AUDIO_STREAM_LIMIT 32

4.9. struct CdxPacketS

名称	struct CdxPacketS		
功能描述	描述媒体数据包信息		
属性	类型	初始值	描述
buf	cdx_void *	NULL	从解码器要来的空间
rinfBuf	cdx_void *	NULL	从解码器要来的第二段空间，当 bufLen 小于 length 的时候需要用到第二段空间
bufLen	cdx_int32	0	buf 的长度
ringBufLen	cdx_int32	0	rinfBuf 的长度
pts	cdx_int64	0	时间戳
duration	cdx_int64	0	该笔数据的时长，该笔数据是字幕时会用到
type	cdx_int32	0	该笔数据的类型，由 CdxMediaTypeE 定义，可取值为：CDX_MEDIA_UNKNOWN，CDX_MEDIA_VIDEO，CDX_MEDIA_AUDIO ， CDX_MEDIA_SUBTITLE ，CDX_MEDIA_DATA
length	cdx_int32	0	该数据包总的数据长度
flags	cdx_uint32	0	描述该笔数据的属性，由 MINOR_STREAM 等组成。 MINOR_STREAM：该笔数据是从流 Video FIRST_PART：该笔数据包含一个可解单元的第一部分 LAST_PART：该笔数据包含一个可解单元的最后一部分 注：FIRST_PART、LAST_PART 属性同时具备表明该笔数据由整数个可解单元组成。 KEY_FRAME：该笔数据是一个关键帧
streamIndex	cdx_int32	0	Stream 序号
pcr	cdx_int64	-1	该笔数据所属媒体分片的时基，目前只对 hls 有效。当 hls parse 出某分片的第一笔数据时，置该值为之前所有分片的 duration 之和；当跳播后 hls parse 出第一笔数据时，置该值为跳播的时间点；其余情况，置该值为-1。
info	void *	NULL	该笔数据的媒体信息，数据类型为：VideoInfo/AudioInfo/SubtitleInfo。

4.10. struct ParserCacheStateS

名称	struct ParserCacheStateS		
功能描述	描述 parser 的 cache 状态		
属性	类型	初始值	描述
nCacheCapacity	cdx_int32	0	Cache 空间的大小
nCacheSize	cdx_int32	0	已 cache 到的数据量
nBandwidthKbps	cdx_int32	0	当前带宽，单位 Kbps
nPercentage	cdx_int32	0	当前缓冲到的百分比，取值 0~100

4.11. struct ParserUriKeyInfoS

名称	struct ParserUriKeyInfoS		
功能描述	描述 Parser 的标志性信息，这有助于对 stream 进行 probe，判明封装格式		
属性	类型	初始值	描述
comment	const char *	NULL	注释，表明 parser 类型的字符串，如 ts parser 对应的 comment = “ts”
scheme[10]	const char *	元素全 NULL	该 parser 常用的流媒体协议的列表
suffix[10]	const char *	元素全 NULL	属于该 parser 的常用文件后缀的列表
attr[10]	const char *	元素全 NULL	该 parser 的流媒体文件 uri 中的常见属性的列表,例如 hls 流媒体文件的 uri 中,常见的属性有“ext=m3u8”“type=ipad”

5. 代码样例

1）flv parser 的实现示例

```
static cdx_int32 __CdxFlvParserControl(CdxParserT *parser, cdx_int32 cmd, void *param)
{
//实现各控制命令
}
static cdx_int32 __CdxFlvParserPrefetch(CdxParserT *parser, CdxPacketT *pkt)
{
//探测下一笔待读数据的信息（包含类型、大小、pts 等）
}
static cdx_int32 __CdxFlvParserRead(CdxParserT *parser, CdxPacketT *pkt)
{
//读取一笔数据
}
static cdx_int32 __CdxFlvParserGetMediaInfo(CdxParserT *parser, CdxMediaInfoT *pMediaInfo)
{
//获取媒体信息
}
static cdx_int32 __CdxFlvParserSeekTo(CdxParserT *parser, cdx_int64 timeUs)
{
//Seek 到指定的时间点
}
static cdx_uint32 __CdxFlvParserAttribute(CdxParserT *parser)
{
//
}
static cdx_int32 __CdxFlvParserGetStatus(CdxParserT *parser)
{
//
}
static cdx_int32 __CdxFlvParserClose(CdxParserT *parser)
{
//
}
static int __CdxFlvParserInit(CdxParserT *parser)
{
//
}
static struct CdxParserOpsS flvParserOps =
{
    .control        = __CdxFlvParserControl,
```

```
    .prefetch       = __CdxFlvParserPrefetch,
    .read           = __CdxFlvParserRead,
    .getMediaInfo = __CdxFlvParserGetMediaInfo,
    .seekTo         = __CdxFlvParserSeekTo,
    .attribute      = __CdxFlvParserAttribute,
    .getStatus      = __CdxFlvParserGetStatus,
    .close          = __CdxFlvParserClose,
    .init           = __CdxFlvParserInit
};
static cdx_bool __CdxFlvParserProbe(CdxStreamProbeDataT *probeData)
{
//
}
static CdxParserT *__CdxFlvParserCreate(CdxStreamT *stream, cdx_uint32 flags)
{
//
}
CdxParserCreatorT flvParserCtor =
{
    .create = __CdxFlvParserCreate,
    .probe = __CdxFlvParserProbe
};
```

2）注册

```
extern CdxParserCreatorT flvParserCtor;
static struct ParserUriKeyInfoS flvKeyInfo =
{
    "flv",
    {"rtmp"},    /*scheme*/
    {".flv"}, /*suffix*/
    {NULL}      /*attribute*/
};

static cdx_void AwParserInit(cdx_void) __attribute__((constructor));
cdx_void AwParserInit(cdx_void)
{
    CDX_LOGI("aw parser init...");
    CdxListInit(&parserList.list);
    parserList.size = 0;
    ......
    AwParserRegister(&flvParserCtor, CDX_PARSER_FLV, &flvKeyInfo);//注册 flv parser
    ......
    return ;
}
```

3.使用该 parser 库

```
{
    CdxDataSourceT dataSource;
    memset(dataSource, 0x00, sizeof(CdxDataSourceT));
    dataSource.uri = "file://xxxxxxx";
    int flags = 0;
    cdx_bool forceExit = CDX_FALSE;
    pthread_mutex_t *mutex;
    cdx_bool exit = 0;
    CdxParserT *parser;
    CdxStreamT *stream;
    CdxParserT *parser = CdxParserPrepare(dataSource, flags, &mutex, &exit,
        &parser, &stream, NULL, NULL);
```

CdxParserPrepare 成功后，则可以该句柄调用 parser 的操作函数 CdxParserGetMediaInfo、CdxParserPrefetch 、 CdxParserRead 、 CdxParserSeekTo 、 CdxParserControl 、CdxParserAttribute、CdxParserGetStatus 等，完成控制和解封装。

退出时，应调用 CdxParserClose，关闭 parser，释放相应资源。

```
}
```

6. Declaration

The information furnished by Allwinner is believed to be accurate and reliable. Allwinner reserves the right to make changes in circuit design and/or specifications at any time without notice. Allwinner does not assume any responsibility and liability for its use. Nor for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Allwinner. This datasheet neither states nor implies warranty of any kind, including fitness for any particular application.