

Set-Top Box NV-300

Спецификация С API, версия 1.0 (07.2014)

Цифровая телевизионная IP-приставка

Версия документа	Дата выпуска	Содержание изменений
Версия 1.0	07.2014	Первая публикация
Версия C API: 1.2 Версия STB API: 1.2		

Целевая аудитория

Спецификация C API предназначена для разработчиков native приложений медиаплеера Eltex NV-300. Квалификация специалистов предполагает знание языка C, библиотеки directFB или языка C++, framework Qt4 (предпочтительно) и практические навыки работы с командной строкой linux.

Условные обозначения

Обозначение	Описание
Полужирный шрифт	Полужирным шрифтом выделены примечания и предупреждения, название глав, заголовков, заголовков таблиц.
<i>Курсивом Calibri</i>	Курсивом Calibri указывается информация, требующая особого внимания.

Примечания и предупреждения



Примечания содержат важную информацию, советы или рекомендации по использованию и настройке устройства.



Предупреждения информируют пользователя о ситуациях, которые могут нанести вред устройству или человеку, привести к некорректной работе устройства или потере данных.

СОДЕРЖАНИЕ

1	ВВЕДЕНИЕ	6
1.1	Аннотация	6
1.2	Управление плеером	6
1.3	Управление экраном	6
1.4	Управление системными настройками	6
1.5	Функции обратного вызова	7
1.6	События	8
2	ГРУППЫ	9
2.1	Управление проигрывателем	9
2.1.1	Макросы	11
	ELT_AUTO_FPS -1.0	11
2.1.2	Типы	11
	ELT_ID	11
2.1.3	Перечисления	11
	ELT_AudioDualModeType_e	11
	ELT_Error_e	12
	ELT_MulticastState_e	12
	ELT_PlaybackSpeed_e	12
	ELT_PlayerDisplayMode_e	13
	ELT_PlayerStatus_e	13
	ELT_RTSP_TRANSPORT_e	14
	ELT_SubtitleEncoding_e	14
2.1.4	Функции	15
	ELT_ClosePlayer	15
	ELT_Continue	15
	ELT_CreatePlayer	15
	ELT_DeleteEventHandler	16
	ELT_AudioDualModeType_e ELT_GetAudioDualMode (ELT_ID id)	16
	ELT_GetAudioPid	16
	ELT_GetAudioTrackList	17
	ELT_GetDuration	17
	ELT_GetFps	18
	ELT_GetLastMulticastEvent	18
	ELT_GetPeersDownloadProgress	18
	ELT_GetPlayerAlphaLevel	19
	ELT_PlayerDisplayMode_e ELT_GetPlayerDisplayMode	19
	ELT_GetPlayerPos	19
	ELT_GetPlayersCount	20
	ELT_PlayerStatus_e ELT_GetPlayerState	20
	ELT_GetPlayerViewPort	20
	ELT_GetPosition	21
	ELT_PlaybackSpeed_e ELT_GetSpeed	21
	ELT_GetSubtitlePid	21
	ELT_GetSubtitleTrackList	22
	ELT_GetTrackName	22
	ELT_GetVideoPid	23
	ELT_GetVideoTrackList	23
	ELT_GetVolume	23
	ELT_GetZoom	24
	ELT_IsMute	24
	ELT_IsPlayerVisible	24
	ELT_IsSubtitlesVisible	25

ELT_MovePlayerToPoint	25
ELT_NextField	25
ELT_Pause	26
ELT_PlayFile	26
ELT_PlayHttp	27
ELT_PlayMagnet	27
ELT_PlayStream	28
ELT_RegisterEventHandler	28
ELT_SetAudioDualMode	28
ELT_SetAudioPid	29
ELT_SetMute	29
ELT_SetPlayerAlphaLevel	30
ELT_SetPlayerDisplayMode	30
ELT_SetPlayerViewPort	30
ELT_SetPlayerVisible	31
ELT_SetPosition	31
ELT_SetSpeed	31
ELT_SetSubtitlePid	32
ELT_SetSubtitlesVisible	32
ELT_SetVolume	33
ELT_SetZoom	33
ELT_Stop	34
2.2 Управление экраном	35
2.2.1 Макросы	35
ELT_MAIN_STB_ID	35
2.2.2 Функции	36
ELT_GetChromaKey	36
ELT_GetGraphicsAlphaLevel	36
ELT_IsChromaKeyEnable	36
ELT_IsGraphicsVisible	36
ELT_SetChromaKey	37
ELT_SetChromaKeyEnable	37
ELT_SetGraphicsAlphaLevel	37
ELT_SetGraphicsVisible	38
ELT_SetResolution	38
2.3 Управление системой	39
2.3.1 Перечисления	39
ELT_RemoteLayout_e	39
2.3.2 Функции	40
ELT_GetBootParamValue	40
ELT_IsInFWUpdate	40
ELT_ReloadGui	40
ELT_ResetBootParams	40
ELT_SetBootParamValue	41
ELT_SyncBootParams	41
ELT_UpgradeFirmware	41
ELT_UpgradeStatus	42
2.4 Функции обратного вызова	43
2.4.1 Типы	43
typedef void	43
2.4.2 Перечисления	43
ELT_Event_e	43
3 СТРУКТУРЫ ДАННЫХ	45

3.1 Структура <code>ELT_Color_t</code>	45
3.2 Структура <code>ELT_MulticastState_t</code>	45
3.3 Структура <code>ELT_Point_t</code>	46
3.4 Структура <code>ELT_Size_t</code>	46
3.5 Структура <code>ELT_Stream_t</code>	47
3.6 Структура <code>ELT_Viewports_t</code>	47
ТЕХНИЧЕСКАЯ ПОДДЕРЖКА	48

1 ВВЕДЕНИЕ

1.1 Аннотация

В данном руководстве приводится подробное описание C API для приставок производства Eltex. API представляет собой разделяемую библиотеку с интерфейсом на языке C. API состоит из трёх логических частей: плеер, экран и система. Также присутствует механизм функций обратного вызова, информирующих о наступлении различных событий.

1.2 Управление плеером

Для начала воспроизведения контента нужно создать экземпляр плеера, вызвав `id = ELT_CreatePlayer()`. Далее этот `id` можно передавать в разные функции `start/stop/pause`, `speed`, `volume`, `viewport` и т.д. Удалить экземпляр плеера можно вызовом `ELT_ClosePlayer()`. Узнать актуальную правильную информацию о воспроизводимом файле можно только после события `ELT_PLAYER_METADATA_UPDATED`. Тип воспроизводимого файла распознается автоматически. Если вызвать `playFile()` с заданной частотой кадров (далее FPS), то будет получено меньше метаданных о файле, так как детальный анализ файла будет пропущен. Когда плеер начинает воспроизведение, порт HDMI переконфигурируется под частоту кадров видеоконтента. Это выглядит как отключение экрана в течение пары секунд. Если плееру задать область вывода, то он будет размещён внутри этой области. Чтобы получать актуальную информацию о воспроизведении, нужно зарегистрировать соответствующие функции обратного вызова.

1.3 Управление экраном

Слой, в котором отображается интерфейс пользователя, имеет следующие настройки: альфа канал, прозрачный цвет. Используя эти настройки можно полностью скрыть слой с интерфейсом. Нижние слои с плеерами будут видны через области, окрашенные прозрачным цветом.

1.4 Управление системными настройками

Буквы русского, английского алфавита и цифры можно вводить, используя пульт приставки. Для переключения раскладки нужно вызвать соответствующую функцию API. В зависимости от разрабатываемого ПО (на основе Qt или на основе directfb) коды клавиш различаются.

Таблица 1 - Отображение кнопок пульта на коды directfb

Код Qt4	Код DirectFB	Unicode	Кнопка пульта
Qt::Key_Escape	DIKS_ESCAPE	0x0000001B	KEY_BACKSPACE
Qt::Key_Return	DIKS_RETURN	0x0000000D	KEY_RETURN
-	DIKS_ENTER	0x0000000D	KEY_RETURN
Qt::Key_NumberSign	DIKS_NUMBER_SIGN	0x00000023	KEY_NUMBER_SIGN(#)
Qt::Key_Backspace	DIKS_BACKSPACE	0x00000008	KEY_ASTERISK
Qt::Key_0	DIKS_0	0x00000030	KEY_0
Qt::Key_1	DIKS_1	0x00000031	KEY_1
Qt::Key_2	DIKS_2	0x00000032	KEY_2

Qt::Key_3	DIKS_3	0x00000033	KEY_3
Qt::Key_4	DIKS_4	0x00000034	KEY_4
Qt::Key_5	DIKS_5	0x00000035	KEY_5
Qt::Key_6	DIKS_6	0x00000036	KEY_6
Qt::Key_7	DIKS_7	0x00000037	KEY_7
Qt::Key_8	DIKS_8	0x00000038	KEY_8
Qt::Key_9	DIKS_9	0x00000039	KEY_9
Qt::Key_Left	DIKS_CURSOR_LEFT	0x0000F000	KEY_CURSOR_LEFT
Qt::Key_Right	DIKS_CURSOR_RIGHT	0x0000F001	KEY_CURSOR_RIGHT
Qt::Key_Up	DIKS_CURSOR_UP	0x0000F002	KEY_CURSOR_UP
Qt::Key_Down	DIKS_CURSOR_DOWN	0x0000F003	KEY_CURSOR_DOWN
-	DIKS_POWER	0x0000F00F	KEY_POWER
Qt::Key_Super_L	DIKS_SUPER	0x0000F220	KEY_MENU
Qt::Key_Help	DIKS_HELP	0x0000F013	KEY_HELP
Qt::Key_F4	DIKS_F4	0x0000F104	KEY_INFO
Qt::Key_Dead_Abovering	DIKS_DEAD_ABOVEERING	0x0000F401	KEY_SUBTITLE
Qt::Key_Dead_Caron	DIKS_DEAD_CARON	0x0000F404	KEY_ANGLE
Qt::Key_Dead_Cedilla	DIKS_DEAD_CEDILLA	0x0000F405	KEY_ZOOM
Qt::Key_Dead_Diaeresis	DIKS_DEAD_DIAERESIS	0x0000F407	KEY_KEYBOARD
Qt::Key_Dead_Abovedot	DIKS_DEAD_ABOVEDOT	0x0000F400	KEY_AUDIO
Qt::Key_F9	DIKS_F9	0x0000F109	KEY_RED
Qt::Key_F10	DIKS_F10	0x0000F10A	KEY_GREEN
Qt::Key_F11	DIKS_F11	0x0000F10B	KEY_YELLOW
Qt::Key_F12	DIKS_F12	0x0000F10C	KEY_BLUE
Qt::Key_Dead_Acute	DIKS_DEAD_ACUTE	0x0000F402	KEY_CHANNEL_UP
Qt::Key_Dead_Breve	DIKS_DEAD_BREVE	0x0000F403	KEY_CHANNEL_DOWN
Qt::Key_VolumeUp	DIKS_VOLUME_UP	0x0000F04C	KEY_VOLUME_UP
Qt::Key_VolumeDown	DIKS_VOLUME_DOWN	0x0000F04D	KEY_VOLUME_DOWN
Qt::Key_VolumeMute	DIKS_MUTE	0x0000F04E	KEY_MUTE
Qt::Key_Pause	DIKS_PLAYPAUSE	0x0000F050	KEY_PLAYPAUSE
Qt::Key_MediaPlay	DIKS_PLAY	0x0000F051	KEY_PLAY
Qt::Key_MediaStop	DIKS_STOP	0x0000F052	KEY_STOP
Qt::Key_MediaRecord	DIKS_RECORD	0x0000F056	KEY_RECORD
Qt::Key_Back	DIKS_BACK	0x0000F048	KEY_REWIND
Qt::Key_Forward	DIKS_FORWARD	0x0000F049	KEY_FASTFORWARD
Qt::Key_MediaPrevious	DIKS_PREVIOUS	0x0000F05B	KEY_PREVIOUS
Qt::Key_MediaNext	DIKS_NEXT	0x0000F05C	KEY_NEXT
Qt::Key_F1	DIKS_F1	0x0000F101	KEY_F1
Qt::Key_F2	DIKS_F2	0x0000F102	KEY_F2
Qt::Key_Dead_Doubleacute	DIKS_DEAD_DOUBLEACUTE	0x0000F408	KEY_CAPS_LOCK
Qt::Key_OpenUrl	DIKS_GOTO	0x0000F00D	KEY_SEARCH

1.5 Функции обратного вызова

Работа с API ориентирована на события. Все вызовы ставят команду в очередь и возвращают управление. О завершении работы этой команды можно узнать, зарегистрировав «callback» на соответствующее событие. API позволяет зарегистрировать функции обратного вызова для каждого экземпляра плеера отдельно, а также на системные события. Функции обратного вызова исполняются в отдельном потоке, в связи с этим нужно следить за синхронизацией. Само по себе API потокобезопасно. Из «callback» функций можно вызывать любые функции API.

1.6 События

- `ELT_PLAYER_STATUS_CHANGED` — состояние плеера изменилось;
- `ELT_PLAYER_METADATA_UPDATED` — новые метаданные о видео получены;
- `ELT_PLAYER_POSITION_CHANGED` — перемещение на новую позицию завершено;
- `ELT_PLAYER_AUDIO_TRACK_SWITCHED` — переключение аудио дорожки завершено;
- `ELT_PLAYER_VIDEO_TRACK_SWITCHED` — переключение видео дорожки завершено;
- `ELT_PLAYER_SUBTL_TRACK_SWITCHED` — переключение дорожки субтитров завершено;
- `ELT_PLAYER_VOLUME_CHANGED` — громкость изменилась;
- `ELT_PLAYER_ZOOM_CHANGED` — зум изменился;
- `ELT_PLAYER_MUTE_CHANGED` — звук отключился или включился.

Когда плеер начинает воспроизведение нового контента, срабатывают описанные выше события. Это означает, что плеер разобрал поток и прочитал из него эту информацию.

Плеер может находиться в следующих состояниях:

- `ERROR` — плеер не может воспроизвести запрашиваемый контент;
- `LOADING` — плеер начинает воспроизведение (например, наполняет кэш);
- `PAUSED` — плеер приостановлен;
- `STOPPED` — плеер остановлен;
- `MOVING` — плеер в процессе перехода на новую позицию;
- `END_OF_STREAM` — плеер дошёл до конца файла или конца потока;
- другие события, описывающие текущую скорость воспроизведения.

2 ГРУППЫ

2.1 Управление проигрывателем

Структуры данных

- struct ELT_Viewport_t
- struct ELT_Point_t
- struct ELT_Size_t
- struct ELT_MulticastState_t
- struct ELT_Stream_t

Макросы

- #define ELT_AUTO_FPS -1.0

Определения типов

- typedef int ELT_ID

Перечисления

- enum ELT_Error_e { ELT_SUCCESS = 0, ELT_INVARG = -128, ELT_INVID, ELT_EXIST, ELT_TOOMUCH, ELT_FATALERR }
- enum ELT_PlaybackSpeed_e { ELT_SPEED_FWD_1X = 0x1, ELT_SPEED_FWD_2X = 0x2, ELT_SPEED_FWD_4X = 0x4, ELT_SPEED_FWD_8X = 0x8, ELT_SPEED_FWD_16X = 0x10, ELT_SPEED_FWD_32X = 0x20, ELT_SPEED_BWD_1X = 0x101, ELT_SPEED_BWD_2X = 0x102, ELT_SPEED_BWD_4X = 0x104, ELT_SPEED_BWD_8X = 0x108, ELT_SPEED_BWD_16X = 0x110, ELT_SPEED_BWD_32X = 0x120 }
- enum ELT_PlayerStatus_e { ERROR = 0x80000000, ERROR_LOW_DR = 0x80000001, ERROR_NOT_CN = 0x80000002, ERROR_NOT_FN = 0x80000004, ERROR_CANT_OPEN = 0x80000008, ERROR_UNKNOWN_CONTENT = 0x80000010, ERROR_NO_DATA = 0x80000020, LOADING = 1, PAUSED, STOPPED, MOVING, END_OF_STREAM, PLAYING_FWD_1x = 0x10000000 | ELT_SPEED_FWD_1X, PLAYING_FWD_2X = 0x10000000 | ELT_SPEED_FWD_2X, PLAYING_FWD_4X = 0x10000000 | ELT_SPEED_FWD_4X, PLAYING_FWD_8X = 0x10000000 | ELT_SPEED_FWD_8X, PLAYING_FWD_16X = 0x10000000 | ELT_SPEED_FWD_16X, PLAYING_FWD_32X = 0x10000000 | ELT_SPEED_FWD_32X, PLAYING_BWD_1X = 0x10000000 | ELT_SPEED_BWD_1X, PLAYING_BWD_2X = 0x10000000 | ELT_SPEED_BWD_2X, PLAYING_BWD_4X = 0x10000000 | ELT_SPEED_BWD_4X, PLAYING_BWD_8X = 0x10000000 | ELT_SPEED_BWD_8X, PLAYING_BWD_16X = 0x10000000 | ELT_SPEED_BWD_16X, PLAYING_BWD_32X = 0x10000000 | ELT_SPEED_BWD_32X }
- enum ELT_PlayerDisplayMode_e { DISPLAY_NORMAL = 1, DISPLAY_CROP, DISPLAY_STRETCH }
- enum ELT_SubtitleEncoding_e { NONE = 0, LATIN1, UTF8 }
- enum ELT_MulticastState_e { LEAVE, JOIN }
- enum ELT_AudioDualModeType_e { ELT_STEREO = 0, ELT_MONO_LEFT, ELT_MONO_RIGHT, ELT_MONO_MIXED }
- enum ELT_RTSP_TRANSPORT_e { ELT_TRANSPORT_UDP = 1, ELT_TRANSPORT_RTP = 2 }

Функции

- `ELT_EXPORT ELT_ID ELT_CreatePlayer ()`
- `ELT_EXPORT int ELT_ClosePlayer (ELT_ID id)`
- `ELT_EXPORT int ELT_GetPlayersCount ()`
- `ELT_EXPORT int ELT_RegisterEventHandler (ELT_ID id, ELT_Event_e event, clbk_fn fn)`
- `ELT_EXPORT int ELT_DeleteEventHandler (ELT_ID id, ELT_Event_e event)`
- `ELT_EXPORT int ELT_PlayFile (ELT_ID id, const char *path, long long pos, long long duration, double fps, const char *subt, int vid_id, int audio_id, int subt_id)`
- `ELT_EXPORT int ELT_PlayStream (ELT_ID id, const char *path, int vid_id, int audio_id, int subt_id)`
- `ELT_EXPORT int ELT_PlayHttp (ELT_ID id, const char *path, const char *opt, int vid_id, int audio_id, int subt_id)`
- `ELT_EXPORT int ELT_PlayMagnet (ELT_ID id, const char *magnet, const char *name, long long size, long long pos, long long duration, double fps, int vid_id, int audio_id, int subt_id)`
- `ELT_EXPORT int ELT_Pause (ELT_ID id)`
- `ELT_EXPORT int ELT_Stop (ELT_ID id)`
- `ELT_EXPORT int ELT_Continue (ELT_ID id)`
- `ELT_EXPORT int ELT_NextField (ELT_ID id)`
- `ELT_EXPORT int ELT_SetPosition (ELT_ID id, long long pos)`
- `ELT_EXPORT long long ELT_GetPosition (ELT_ID id)`
- `ELT_EXPORT long long ELT_GetDuration (ELT_ID id)`
- `ELT_EXPORT int ELT_GetTrackName (ELT_ID id, char **name, unsigned int *len)`
- `ELT_EXPORT int ELT_SetSpeed (ELT_ID id, ELT_PlaybackSpeed_e speed)`
- `ELT_EXPORT ELT_PlaybackSpeed_e ELT_GetSpeed (ELT_ID id)`
- `ELT_EXPORT int ELT_GetPeersDownloadProgress (ELT_ID id)`
- `ELT_EXPORT int ELT_GetVideoPid (ELT_ID id)`
- `ELT_EXPORT int ELT_GetVideoTrackList (ELT_ID id, ELT_Stream_t **tracks, int *trackNum)`
- `ELT_EXPORT int ELT_SetAudioPid (ELT_ID id, int pid)`
- `ELT_EXPORT int ELT_GetAudioPid (ELT_ID id)`
- `ELT_EXPORT int ELT_GetAudioTrackList (ELT_ID id, ELT_Stream_t **ptr, int *num)`
- `ELT_EXPORT int ELT_SetSubtitlePid (ELT_ID id, int pid)`
- `ELT_EXPORT int ELT_GetSubtitlePid (ELT_ID id)`
- `ELT_EXPORT int ELT_GetSubtitleTrackList (ELT_ID id, ELT_Stream_t **ptr, int *num)`
- `ELT_EXPORT int ELT_SetSubtitlesVisible (ELT_ID id, int visibility)`
- `ELT_EXPORT int ELT_IsSubtitlesVisible (ELT_ID id)`
- `ELT_EXPORT int ELT_SetAudioDualMode (ELT_ID id, ELT_AudioDualModeType_e mode)`
- `ELT_EXPORT ELT_AudioDualModeType_e ELT_GetAudioDualMode (ELT_ID id)`
- `ELT_EXPORT int ELT_SetVolume (ELT_ID id, int vol)`
- `ELT_EXPORT int ELT_GetVolume (ELT_ID id)`
- `ELT_EXPORT int ELT_IsMute (ELT_ID id)`
- `ELT_EXPORT int ELT_SetMute (ELT_ID id, int mute)`
- `ELT_EXPORT double ELT_GetZoom (ELT_ID id)`
- `ELT_EXPORT int ELT_SetZoom (ELT_ID id, double zoom)`
- `ELT_EXPORT double ELT_GetFps (ELT_ID id)`
- `ELT_EXPORT int ELT_SetPlayerViewPort (ELT_ID id, ELT_Viewport_t *vp)`
- `ELT_EXPORT int ELT_GetPlayerViewPort (ELT_ID id, ELT_Viewport_t *vp)`
- `ELT_EXPORT int ELT_GetPlayerPos (ELT_ID id, ELT_Point_t *p)`

- `ELT_EXPORT int ELT_MovePlayerToPoint (ELT_ID id, ELT_Point_t *p)`
- `ELT_EXPORT int ELT_SetPlayerAlphaLevel (ELT_ID id, int alpha)`
- `ELT_EXPORT int ELT_GetPlayerAlphaLevel (ELT_ID id)`
- `ELT_EXPORT int ELT_SetPlayerVisible (ELT_ID id, int visibility)`
- `ELT_EXPORT int ELT_IsPlayerVisible (ELT_ID id)`
- `ELT_EXPORT ELT_PlayerStatus_e ELT_GetPlayerState (ELT_ID id)`
- `ELT_EXPORT ELT_PlayerDisplayMode_e ELT_GetPlayerDisplayMode (ELT_ID id)`
- `ELT_EXPORT int ELT_SetPlayerDisplayMode (ELT_ID id, ELT_PlayerDisplayMode_e mode)`
- `ELT_EXPORT int ELT_GetLastMulticastEvent (ELT_ID playerId, ELT_MulticastState_t *state)`

2.1.1 Макросы

ELT_AUTO_FPS -1.0

```
#define ELT_AUTO_FPS -1.0
```

Автоматическое определение частоты кадров в секунду.

2.1.2 Типы

ELT_ID

```
int ELT_ID
```

Идентификатор объекта.

2.1.3 Перечисления

ELT_AudioDualModeType_e

```
enum ELT_AudioDualModeType_e
```

Тип режима аудио.

ELT_Error_e

`enum ELT_Error_e`

Коды ошибок Eltex API.

Элементы перечислений:

- **ELT_SUCCESS** операция завершилась успешно;
- **ELT_INVARG** неправильный аргумент;
- **ELT_INVID** неправильный идентификатор объекта;
- **ELT_EXIST** объект уже существует или операция выполняется;
- **ELT_TOOMUCH** количество плееров достигло максимума, новый плеер создать невозможно;
- **ELT_FATALERR** проблемы с нижележащим ПО или аппаратной частью, дальнейшая работа API непредсказуема.

ELT_MulticastState_e

`enum ELT_MulticastState_e`

Состояние IGMP.

Элементы перечислений:

- **LEAVE** плеер вышел из состава мультикастовой группы;
- **JOIN** плеер вошёл в состав мультикастовой группы.

ELT_PlaybackSpeed_e

`enum ELT_PlaybackSpeed_e`

Битовые маски скорости.

Элементы перечислений:

- **ELT_SPEED_FWD_1X** 1x вперёд;
- **ELT_SPEED_FWD_2X** 2x вперёд;
- **ELT_SPEED_FWD_4X** 4x вперёд;
- **ELT_SPEED_FWD_8X** 8x вперёд;
- **ELT_SPEED_FWD_16X** 16x вперёд;
- **ELT_SPEED_FWD_32X** 32x вперёд;
- **ELT_SPEED_BWD_1X** 1x назад;
- **ELT_SPEED_BWD_2X** 2x назад;
- **ELT_SPEED_BWD_4X** 4x назад;
- **ELT_SPEED_BWD_8X** 8x назад;
- **ELT_SPEED_BWD_16X** 16x назад;
- **ELT_SPEED_BWD_32X** 32x назад.

ELT_PlayerDisplayMode_e

enum ELT_PlayerDisplayMode_e

Режим отображения видео.

Элементы перечислений:

- **DISPLAY_NORMAL** видео без изменения;
- **DISPLAY_CROP** видео обрезается сверху и снизу так, что 4:3 становится 16:9;
- **DISPLAY_STRETCH** видео растягивается так, что 4:3 вписывается в экран 16:9.

ELT_PlayerStatus_e

enum ELT_PlayerStatus_e

Статусы плеера.

Элементы перечислений:

- **ERROR** плеер не может воспроизвести этот контент;
- **ERROR_LOW_DR** низкая скорость скачивания р2р;
- **ERROR_NOT_CN** не могу подключиться (для сетевого проигрывания);
- **ERROR_NOT_FN** файл не найден (rtsp or udcp);
- **ERROR_CANT_OPEN** невозможно открыть файл;
- **ERROR_UNKNOWN_CONTENT** невозможно определить тип контента;
- **ERROR_NO_DATA** невозможно качать дальше (dcp/torrent);
- **LOADING** плеер подгружает данные;
- **PAUSED** плеер поставлен на паузу;
- **STOPPED** плеер остановлен;
- **MOVING** плеер перемещается на позицию в файле;
- **END_OF_STREAM** конец файла или конце потока (IGMP мультикаст не поддерживается);
- **PLAYING_FWD_1x** проигрывается в прямом направлении со скоростью 1x ;
- **PLAYING_FWD_2X** проигрывается в прямом направлении со скоростью 2x;
- **PLAYING_FWD_4X** проигрывается в прямом направлении со скоростью 4x;
- **PLAYING_FWD_8X** проигрывается в прямом направлении со скоростью 8x;
- **PLAYING_FWD_16X** проигрывается в прямом направлении со скоростью 16x;
- **PLAYING_FWD_32X** проигрывается в прямом направлении со скоростью 32x;
- **PLAYING_BWD_1X** проигрывается в обратном направлении со скоростью 1x;
- **PLAYING_BWD_2X** проигрывается в обратном направлении со скоростью 2x;
- **PLAYING_BWD_4X** проигрывается в обратном направлении со скоростью 4x;
- **PLAYING_BWD_8X** проигрывается в обратном направлении со скоростью 8x;
- **PLAYING_BWD_16X** проигрывается в обратном направлении со скоростью 16x;
- **PLAYING_BWD_32X** проигрывается в обратном направлении со скоростью 32x.

ELT_RTSP_TRANSPORT_e

enum **ELT_RTSP_TRANSPORT_e**

Транспорт для RTSP-проигрывания.

Элементы перечислений:

- **ELT_TRANSPORT_UDP** запрашивается mp2t юникастовый поток UDP (live555media может вещать такой поток, чтобы уменьшить нагрузку на приставку);
- **ELT_TRANSPORT_RTP** протокол RTP является транспортным. Приставка потребляет больше ресурсов, чем UDP (VLC вещает RTSP+RTP).

ELT_SubtitleEncoding_e

enum **ELT_SubtitleEncoding_e**

Кодировка субтитров.

Элементы перечислений:

- **NONE** нет субтитров;
- **LATIN1** Latin-1 (ISO 8859-1);
- **UTF8** UTF-8.

2.1.4 Функции

ELT_ClosePlayer

```
ELT_EXPORT int ELT_ClosePlayer (ELT_ID id)
```

Удаляет экземпляр плеера.

Параметры:

id	идентификатор плеера
----	----------------------

Возвращает:

Статус операции.

ELT_Continue

```
ELT_EXPORT int ELT_Continue (ELT_ID id)
```

Возобновляет воспроизведение после паузы.

Параметры:

id	идентификатор плеера
----	----------------------

Возвращает:

Статус операции.

ELT_CreatePlayer

```
ELT_EXPORT ELT_ID ELT_CreatePlayer ()
```

Создаёт новый экземпляр плеера.

Возвращает:

Идентификатор созданного плеера или отрицательное число в случае ошибки.

ELT_DeleteEventHandler

```
ELT_EXPORT int ELT_DeleteEventHandler (ELT_ID id, ELT_Event_e event)
```

Удаляет обработчик события.

Параметры:

id	идентификатор плеера или системный идентификатор ELT_MAIN_STB_ID
event	тип события

Возвращает:

Статус операции.

ELT_AudioDualModeType_e ELT_GetAudioDualMode (ELT_ID id)

```
ELT_EXPORT ELT_AudioDualModeType_e ELT_GetAudioDualMode (ELT_ID id)
```

Возвращает текущий режим аудио.

Параметры:

id	идентификатор плеера
----	----------------------

Возвращает:

Текущий режим аудио.

ELT_GetAudioPid

```
ELT_EXPORT int ELT_GetAudioPid (ELT_ID id)
```

Возвращает PID текущей аудиодорожки.

Параметры:

id	идентификатор плеера
----	----------------------

Возвращает:

Статус операции.

ELT_GetAudioTrackList

```
ELT_EXPORT int ELT_GetAudioTrackList (ELT_ID id, ELT_Stream_t ** ptr, int * num)
```

Возвращает список доступных аудиодорожек.



Функция выделяет память. Не забудьте её освободить.



Функцию можно вызывать только по событию ELT_PLAYER_METADATA_UPDATED.

Параметры:

in	id	идентификатор плеера
out	ptr	указатель на массив дорожек
out	num	количество дорожек

Возвращает:

Список аудиодорожек.

ELT_GetDuration

```
ELT_EXPORT long long ELT_GetDuration (ELT_ID id)
```

Возвращает длительность файла.



Функцию можно вызывать только по событию ELT_PLAYER_METADATA_UPDATED.

Параметры:

id	идентификатор плеера
----	----------------------

Возвращает:

Длительность в миллисекундах.

ELT_GetFps

```
ELT_EXPORT double ELT_GetFps (ELT_ID id)
```

Возвращает частоту кадров (FPS) для текущего контента.

Параметры:

id	идентификатор плеера
----	----------------------

Возвращает:

FPS.

ELT_GetLastMulticastEvent

```
ELT_EXPORT int ELT_GetLastMulticastEvent (ELT_ID playerInstance, ELT_MulticastState_t * state)
```

Возвращает последнее IGMP-событие для плеера. Вход и выход из мультикастовой группы.



Функцию можно вызывать только по событию ELT_PLAYER_MULTICAST_GROUP_SWITCHED.

Параметры:

	id	идентификатор плеера
out	state	последнее состояние IGMP

Возвращает:

Статус операции.

ELT_GetPeersDownloadProgress

```
ELT_EXPORT int ELT_GetPeersDownloadProgress (ELT_ID id)
```

Возвращает прогресс скачивания p2p (доступно для плеера в режиме LOADING и MOVING).



**Выдаёт корректные данные только для dcpp/torrent воспроизведения.
Функцию можно вызывать только по событию ELT_PEERS_DOWNLOAD_PROGRESS.**

Параметры:

id	идентификатор плеера
----	----------------------

Возвращает:

Прогресс загрузки (1..100%).

ELT_GetPlayerAlphaLevel

```
ELT_EXPORT int ELT_GetPlayerAlphaLevel (ELT_ID id)
```

Возвращает уровень прозрачности плеера.

Параметры:

id	идентификатор плеера
----	----------------------

Возвращает:

Уровень прозрачности (0-255), где 0 - полностью прозрачный.

ELT_PlayerDisplayMode_e ELT_GetPlayerDisplayMode

```
ELT_EXPORT ELT_PlayerDisplayMode_e ELT_GetPlayerDisplayMode (ELT_ID id)
```

Возвращает режим отображения видео:

NORMAL - нормальное;

CROP - обрезанное по краям;

STRETCH - растянутое.

Параметры:

id	идентификатор плеера
----	----------------------

Возвращает:

Режим отображения (NORMAL, CROP, STRETCH).

ELT_GetPlayerPos

```
ELT_EXPORT int ELT_GetPlayerPos (ELT_ID id, ELT_Point_t * p)
```

Возвращает координаты области плеера.

Параметры:

id	идентификатор плеера
p	указатель на ELT_Point_t

Возвращает:

Статус операции.

ELT_GetPlayersCount

```
ELT_EXPORT int ELT_GetPlayersCount ()
```

Возвращает количество созданных экземпляров плеера.

Возвращает:

Количество экземпляров плеера.

ELT_PlayerStatus_e ELT_GetPlayerState

```
ELT_EXPORT ELT_PlayerStatus_e ELT_GetPlayerState (ELT_ID id)
```

Возвращает состояние плеера.



Функцию можно вызывать только по событию **ELT_PLAYER_STATUS_CHANGED**.

Параметры:

id	идентификатор плеера
----	----------------------

Возвращает:

Состояние плеера.

ELT_GetPlayerViewPort

```
ELT_EXPORT int ELT_GetPlayerViewPort (ELT_ID id, ELT_ViewPort_t * vp)
```

Возвращает параметры прямоугольной области, в которую вписан плеер.

Параметры:

	id	идентификатор плеера
out	vp	прямоугольная область

Возвращает:

Статус операции.

ELT_GetPosition

```
ELT_EXPORT long long ELT_GetPosition (ELT_ID id)
```

Возвращает текущую позицию.



Функцию можно вызывать только по событию ELT_PLAYER_POSITION_CHANGED.

Параметры:

id	идентификатор плеера
----	----------------------

Возвращает:

Позиция в миллисекундах.

ELT_PlaybackSpeed_e ELT_GetSpeed

```
ELT_EXPORT ELT_PlaybackSpeed_e ELT_GetSpeed (ELT_ID id)
```

Возвращает текущую скорость воспроизведения.

Параметры:

id	идентификатор плеера
----	----------------------

Возвращает:

Одно значение скорости из возможных.

ELT_GetSubtitlePid

```
ELT_EXPORT int ELT_GetSubtitlePid (ELT_ID id)
```

Возвращает PID текущей дорожки субтитров.

Параметры:

id	идентификатор плеера
----	----------------------

Возвращает:

PID дорожки субтитров.

ELT_GetSubtitleTrackList

```
ELT_EXPORT int ELT_GetSubtitleTrackList (ELT_ID id, ELT_Stream_t ** ptr, int * num)
```

Возвращает список доступных дорожек субтитров.



Функция выделяет память. Не забудьте её освободить.



Функцию можно вызывать только по событию ELT_PLAYER_METADATA_UPDATED.

Параметры:

in	id	идентификатор плеера
out	ptr	указатель на массив дорожек
out	num	количество дорожек

Возвращает:

Список PID дорожек субтитров.

ELT_GetTrackName

```
ELT_EXPORT int ELT_GetTrackName (ELT_ID id, char ** name, unsigned int * len)
```

Возвращает название трека. При воспроизведении shoutcast радио это название текущего трека, в остальных случаях - путь к воспроизводимому файлу или ссылка.



Функция выделяет память. Не забудьте её освободить.



Функцию можно вызывать только по событию ELT_PLAYER_METADATA_UPDATED.

Параметры:

in	id	идентификатор плеера
out	name	указатель на символьный массив, содержащий название трека
out	len	длина названия трека

Возвращает:

Статус операции.

ELT_GetVideoPid

```
ELT_EXPORT int ELT_GetVideoPid (ELT_ID id)
```

Возвращает PID видео дорожки.

Параметры:

id	идентификатор плеера
----	----------------------

Возвращает:

PID видео дорожки.

ELT_GetVideoTrackList

```
ELT_EXPORT int ELT_GetVideoTrackList (ELT_ID id, ELT_Stream_t ** tracks, int * trackNum)
```

Возвращает список доступных видео дорожек.



Функция выделяет память. Не забудьте её освободить.



Функцию можно вызывать только по событию ELT_PLAYER_METADATA_UPDATED.

Параметры:

in	id	идентификатор плеера
out	tracks	указатель на массив дорожек
out	trackNum	количество дорожек

Возвращает:

Список видеодорожек.

ELT_GetVolume

```
ELT_EXPORT int ELT_GetVolume (ELT_ID id)
```

Возвращает текущий уровень громкости.

Параметры:

id	идентификатор плеера
----	----------------------

Возвращает:

Текущий уровень громкости.

ELT_GetZoom

```
ELT_EXPORT double ELT_GetZoom (ELT_ID id)
```

Возвращает коэффициент зумирования.

Параметры:

id	идентификатор плеера
----	----------------------

Возвращает:

Коэффициент зумирования.

ELT_IsMute

```
ELT_EXPORT int ELT_IsMute (ELT_ID id)
```

Возвращает состояние режима "без звука".

Параметры:

id	идентификатор плеера
----	----------------------

Возвращает:

Состояние:

1 - без звука;

0 - со звуком.

ELT_IsPlayerVisible

```
ELT_EXPORT int ELT_IsPlayerVisible (ELT_ID id)
```

Возвращает состояние плеера (скрыт/отображается).

Параметры:

id	идентификатор плеера
----	----------------------

Возвращает:

Состояние плеера:

1 - отображается;

0 - скрыт.

ELT_IsSubtitlesVisible

```
ELT_EXPORT int ELT_IsSubtitlesVisible (ELT_ID id)
```

Возвращает состояние субтитров (скрыты/отображаются).

Параметры:

in	id	идентификатор плеера
----	----	----------------------

Возвращает:

Состояние субтитров:
 1 - субтитры отображаются;
 0 - субтитры скрыты.

ELT_MovePlayerToPoint

```
ELT_EXPORT int ELT_MovePlayerToPoint (ELT_ID id, ELT_Point_t * p)
```

Передвигает плеер на позицию с заданными координатами.

Параметры:

id	идентификатор плеера
p	координаты ELT_Point_t

Возвращает:

Статус операции.

ELT_NextFrame

```
ELT_EXPORT int ELT_NextFrame (ELT_ID id)
```

Переводит плеер на следующий кадр.



Функция работает только в режиме паузы.

Параметры:

id	идентификатор плеера
----	----------------------

Возвращает:

Статус операции.

ELT_Pause

```
ELT_EXPORT int ELT_Pause (ELT_ID id)
```

Ставит плеер на паузу.



Когда воспроизведение будет приостановлено, произойдёт событие **ELT_PLAYER_STATUS_CHANGED**, статус плеера изменится на **PAUSED**.

Параметры:

id	идентификатор плеера
----	----------------------

Возвращает:

Статус операции.

ELT_PlayFile

```
ELT_EXPORT int ELT_PlayFile (ELT_ID id, const char * path, long long pos, long long duration, double fps, const char * subt, int vid_id, int audio_id, int subt_id)
```

Запускает файл на воспроизведение.

Параметры:

id	идентификатор плеера
path	путь к файлу "/tmp/video.avi"
pos	начальная позиция в файле
duration	продолжительность файла по времени
fps	количество кадров в секунду или ELT_AUTO_FPS для авто определения
subt	путь к внешним субтитрам (формат .srt), устанавливается в случае необходимости
vid_id	желаемый номер видеодорожки (по умолчанию -1)
audio_id	желаемый номер аудиодорожки (по умолчанию -1)
subt_id	желаемый номер дорожки субтитров (по умолчанию -1)



Для установки определенного FPS нужно установить "duration", иначе "duration" будет равен -1.

Когда плеер начнёт воспроизводить контент, произойдёт событие **ELT_PLAYER_STATUS_CHANGED**, статус плеера изменится на **PLAYING_FWD_1x**.

Когда плеер распознает тип файла, произойдёт событие **ELT_PLAYER_METADATA_UPDATED**, далее можно запрашивать информацию о дорожках соответствующими вызовами API.

Возвращает:

Статус операции.

ELT_PlayHttp

```
ELT_EXPORT int ELT_PlayHttp (ELT_ID id, const char * path, const char * opt, int vid_id, int audio_id, int subt_id)
```

Запускает на воспроизведение контент по HTTP-ссылке.



Когда плеер начнёт воспроизводить контент, произойдёт событие **ELT_PLAYER_STATUS_CHANGED**, статус плеера изменится на **PLAYING_FWD_1x**.
Когда плеер распознает тип файла, произойдёт событие **ELT_PLAYER_METADATA_UPDATED**, далее можно запрашивать информацию о дорожках соответствующими вызовами API.

Параметры:

id	идентификатор плеера
path	HTTP-ссылка "http://videourl.com/somemedia.avi"
opt	специальный HTTP-заголовок, который будет использован плеером (NULL, без особого заголовка)
vid_id	желаемый номер видеодорожки (по умолчанию -1)
audio_id	желаемый номер аудиодорожки (по умолчанию -1)
subt_id	желаемый номер дорожки субтитров (по умолчанию -1)

Возвращает:

Статус операции.

ELT_PlayMagnet

```
ELT_EXPORT int ELT_PlayMagnet (ELT_ID id, const char * magnet, const char * name, long long size, long long pos, long long duration, double fps, int vid_id, int audio_id, int subt_id)
```

Запускает на воспроизведение файл по magnet-ссылке.



Когда плеер начнёт воспроизводить контент, произойдёт событие **ELT_PLAYER_STATUS_CHANGED**, статус плеера изменится на **PLAYING_FWD_1x**.
Когда плеер распознает тип файла, произойдёт событие **ELT_PLAYER_METADATA_UPDATED**, далее можно запрашивать информацию о дорожках соответствующими вызовами API.

Параметры:

id	идентификатор плеера
magnet	magnet-ссылка
name	имя файла
size	размер
length	продолжительность
pos	start позиция начала воспроизведения
vid_id	желаемый номер видеодорожки (по умолчанию -1)
audio_id	желаемый номер аудиодорожки (по умолчанию -1)
subt_id	желаемый номер дорожки субтитров (по умолчанию -1)

Возвращает:

Статус операции.

ELT_PlayStream

```
ELT_EXPORT int ELT_PlayStream (ELT_ID id, const char * path, int vid_id, int audio_id, int subtit_id)
```

Запускает поток на воспроизведение.

Параметры:

id	идентификатор плеера
path	URL-потока "udp://226.0.0.1:1111" или "rtp://226.0.0.1:1111"
vid_id	желаемый номер видео дорожки (по умолчанию -1)
audio_id	желаемый номер аудио дорожки (по умолчанию -1)
subtit_id	желаемый номер дорожки субтитров (по умолчанию -1)



Когда плеер начнёт воспроизводить контент, произойдёт событие **ELT_PLAYER_STATUS_CHANGED**, статус плеера изменится на **PLAYING_FWD_1x**.
Когда плеер распознает тип файла, произойдёт событие **ELT_PLAYER_METADATA_UPDATED**, далее можно запрашивать информацию о дорожках соответствующими вызовами API.

Возвращает:

Статус операции.

ELT_RegisterEventHandler

```
ELT_EXPORT int ELT_RegisterEventHandler (ELT_ID id, ELT_Event_e event, clbk_fn fn)
```

Устанавливает обработчик события.

Параметры:

id	идентификатор плеера или системный идентификатор ELT_MAIN_STB_ID
event	тип события
fn	указатель на функцию

Возвращает:

Статус операции.

ELT_SetAudioDualMode

```
ELT_EXPORT int ELT_SetAudioDualMode (ELT_ID id, ELT_AudioDualModeType_e mode)
```

Устанавливает режим аудио.

Параметры:

id	идентификатор плеера
mode	режим аудио

Возвращает:

Статус операции.

ELT_SetAudioPid

```
ELT_EXPORT int ELT_SetAudioPid (ELT_ID id, int pid)
```

Выбирает аудиодорожку.



Когда дорожка переключилась, происходит событие **ELT_PLAYER_AUDIO_TRACK_SWITCHED**.

Параметры:

id	идентификатор плеера
pid	PID аудио дорожки

Возвращает:

Статус операции.

ELT_SetMute

```
ELT_EXPORT int ELT_SetMute (ELT_ID id, int mute)
```

Устанавливает режим "без звука".



Когда установится режим "без звука", произойдет событие **ELT_PLAYER_MUTE_CHANGED**.

Параметры:

id	идентификатор плеера
mute	назначаемое действие: 1 - без звука; 0 - со звуком.

Возвращает:

Статус операции.

ELT_SetPlayerAlphaLevel

```
ELT_EXPORT int ELT_SetPlayerAlphaLevel (ELT_ID id, int alpha)
```

Устанавливает уровень прозрачности для плеера.

Параметры:

id	идентификатор плеера
alpha	уровень прозрачности (0-255), где 0 - полностью прозрачный.

Возвращает:

Статус операции.

ELT_SetPlayerDisplayMode

```
ELT_EXPORT int ELT_SetPlayerDisplayMode (ELT_ID id, ELT_PlayerDisplayMode_e mode)
```

Устанавливает режим отображения видео.

Параметры:

id	идентификатор плеера
mode	режим отображения видео: NORMAL – нормальное; CROP – обрезанное по краям; STRETCH – растянутое.

Возвращает:

Статус операции.

ELT_SetPlayerViewPort

```
ELT_EXPORT int ELT_SetPlayerViewPort (ELT_ID id, ELT_Viewport_t * vp)
```

Вписывает плеер в прямоугольную область.

Параметры:

id	идентификатор плеера
vp	прямоугольная область

Возвращает:

Статус операции.

ELT_SetPlayerVisible

```
ELT_EXPORT int ELT_SetPlayerVisible (ELT_ID id, int visibility)
```

Скрывает плеер с экрана, не останавливая воспроизведение, или возвращает плеер обратно на экран.

Параметры:

id	идентификатор плеера
visibility	назначаемое действие: 1 - показать плеер; 0 - скрыть плеер.

Возвращает:

Статус операции.

ELT_SetPosition

```
ELT_EXPORT int ELT_SetPosition (ELT_ID id, long long pos)
```

Установить позицию в миллисекундах.

Параметры:

id	идентификатор плеера
pos	позиция в миллисекундах

Возвращает:

Статус операции.

ELT_SetSpeed

```
ELT_EXPORT int ELT_SetSpeed (ELT_ID id, ELT_PlaybackSpeed_e speed)
```

Устанавливает скорость воспроизведения.



Когда скорость увеличится, произойдёт событие **ELT_PLAYER_STATUS_CHANGED**, статус плеера изменится на **PLAYING_FWD_xx** или **PLAYING_BWD_xx**.

Параметры:

id	идентификатор плеера
speed	одно значение скорости из возможных

Возвращает:

Статус операции.

ELT_SetSubtitlePid

```
ELT_EXPORT int ELT_SetSubtitlePid (ELT_ID id, int pid)
```

Выбирает дорожку субтитров.



Когда дорожка переключилась, происходит событие `ELT_PLAYER_SUBTL_TRACK_SWITCHED`.

Параметры:

id	идентификатор плеера
pid	PID дорожки субтитров

Возвращает:

Статус операции.

ELT_SetSubtitlesVisible

```
ELT_EXPORT int ELT_SetSubtitlesVisible (ELT_ID id, int visibility)
```

Позволяет скрыть субтитры с экрана. При этом выбранная дорожка субтитров сохраняется.

Параметры:

id	идентификатор плеера
visibility	назначаемое действие: 1 - показать субтитры; 0 - скрыть субтитры.

Возвращает:

Статус операции.

ELT_SetVolume

```
ELT_EXPORT int ELT_SetVolume (ELT_ID id, int vol)
```

Устанавливает уровень громкости.



Когда установится новый уровень громкости, происходит событие ELT_PLAYER_VOLUME_CHANGED.

Параметры:

id	идентификатор плеера
vol	уровень громкости (0-72)



Когда громкость применилась, происходит событие ELT_PLAYER_VOLUME_CHANGED.

Возвращает:

Статус операции.

ELT_SetZoom

```
ELT_EXPORT int ELT_SetZoom (ELT_ID id, double zoom)
```

Устанавливает коэффициент зумирования.



Когда установится новый коэффициент зумирования, произойдет событие ELT_PLAYER_ZOOM_CHANGED.

Параметры:

id	идентификатор плеера
zoom	коэффициент зумирования, где значение "1" - без зумирования

Возвращает:

Статус операции.

ELT_Stop

`ELT_EXPORT int ELT_Stop (ELT_ID id)`

Останавливает воспроизведение.



Когда воспроизведение будет остановлено, произойдёт событие **ELT_PLAYER_STATUS_CHANGED**, статус плеера изменится на **STOPPED**.

Параметры:

id	идентификатор плеера
----	----------------------

Возвращает:

Статус операции.

2.2 Управление экраном

Структуры данных

- struct ELT_Color_t

Макросы

- #define ELT_MAIN_STB_ID -128

Функции

- ELT_EXPORT int ELT_GetChromaKey (ELT_Color_t *ptr)
- ELT_EXPORT int ELT_SetChromaKey (ELT_Color_t color)
- ELT_EXPORT int ELT_IsChromaKeyEnable ()
- ELT_EXPORT int ELT_SetChromaKeyEnable (int enable)
- ELT_EXPORT int ELT_SetGraphicsAlphaLevel (int alpha)
- ELT_EXPORT int ELT_GetGraphicsAlphaLevel ()
- ELT_EXPORT int ELT_SetGraphicsVisible (int visibility)
- ELT_EXPORT int ELT_IsGraphicsVisible ()
- ELT_EXPORT int ELT_SetResolution (ELT_Size_t *size)

2.2.1 Макросы

ELT_MAIN_STB_ID

```
#define ELT_MAIN_STB_ID -128
```

Идентификатор главного системного объекта API.

2.2.2 Функции

ELT_GetChromaKey

```
ELT_EXPORT int ELT_GetChromaKey (ELT_Color_t * ptr)
```

Возвращает текущий прозрачный цвет.

Параметры:

out	ptr	указатель на пустой ELT_Color_t
-----	-----	---------------------------------

Возвращает:

Статус операции.

ELT_GetGraphicsAlphaLevel

```
ELT_EXPORT int ELT_GetGraphicsAlphaLevel ()
```

Возвращает уровень прозрачности пользовательского интерфейса.

Возвращает:

Alpha уровень прозрачности (0-255).

ELT_IsChromaKeyEnable

```
ELT_EXPORT int ELT_IsChromaKeyEnable ()
```

Возвращает состояние режима прозрачного цвета.

Возвращает:

Состояние режима прозрачного цвета:

0 - выключен;

1 - включен.

ELT_IsGraphicsVisible

```
ELT_EXPORT int ELT_IsGraphicsVisible ()
```

Возвращает состояние видимости пользовательского интерфейса.

Возвращает:

Состояние:

0 - пользовательский интерфейс отображается;

1 - пользовательский интерфейс не отображается.

ELT_SetChromaKey

```
ELT_EXPORT int ELT_SetChromaKey (ELT_Color_t color)
```

Устанавливает прозрачный цвет.

Параметры:

color	цвет в структуре ELT_Color_t
-------	-------------------------------------

Возвращает:

Статус операции.

ELT_SetChromaKeyEnable

```
ELT_EXPORT int ELT_SetChromaKeyEnable (int enable)
```

Включает/выключает режим прозрачного цвета.

Параметры:

enable	назначаемое действие: 0 - выключить режим; 1 - включить режим
--------	---

Возвращает:

Статус операции.

ELT_SetGraphicsAlphaLevel

```
ELT_EXPORT int ELT_SetGraphicsAlphaLevel (int alpha)
```

Устанавливает уровень прозрачности пользовательского интерфейса.

Параметры:

alpha	уровень прозрачности (0-255), где 0 - полностью прозрачный
-------	--

Возвращает:

Статус операции.

ELT_SetGraphicsVisible

```
ELT_EXPORT int ELT_SetGraphicsVisible (int visibility)
```

Устанавливает видимость пользовательского интерфейса.

Параметры:

visibility	назначаемое действие: 0 - не отображать пользовательский интерфейс; 1 - отображать пользовательский интерфейс
------------	---

Возвращает:

Статус операции.

ELT_SetResolution

```
ELT_EXPORT int ELT_SetResolution (ELT_Size_t * size)
```

Устанавливает новое разрешение экрана. Это может быть полезно для приложений, требующих смену разрешения без перезапуска directfb.

Параметры:

size	размер, в пикселях
------	--------------------

Возвращает:

Статус операции.

2.3 Управление системой

Перечисления

- enum ELT_RemoteLayout_e { DIG = 0, ENG, RU }

Функции

- ELT_EXPORT void ELT_ReloadGui ()
- ELT_EXPORT int ELT_SyncBootParams ()
- ELT_EXPORT int ELT_ResetBootParams ()
- ELT_EXPORT int ELT_SetBootParamValue (const char *key, const char *value)
- ELT_EXPORT int ELT_GetBootParamValue (const char *key, char *value)
- ELT_EXPORT int ELT_UpgradeFirmware (const char *path)
- ELT_EXPORT int ELT_UpgradeStatus (int *progress, int *err)
- ELT_EXPORT int ELT_IsInFWUpdate ()

2.3.1 Перечисления

ELT_RemoteLayout_e

enum ELT_RemoteLayout_e

Раскладка пульта.

Элементы перечислений:

- **DIG** цифры;
- **ENG** английские символы;
- **RU** русские символы.

2.3.2 Функции

ELT_GetBootParamValue

```
ELT_EXPORT int ELT_GetBootParamValue (const char * key, char * value)
```

Возвращает загрузочный параметр по ключу. Приставка применяет эти параметры при загрузке. Поддерживаются следующие ключи : "dfbres" - разрешение пользовательского интерфейса.

Параметры:

	key	ключ
out	value	указатель на char[255], по которому будет записан результат

Возвращает:

Статус операции.

ELT_IsInFWUpdate

```
ELT_EXPORT int ELT_IsInFWUpdate ()
```

Возвращает состояние обновления.

Возвращает:

Состояние обновления:

1 - выполняется;

0 - не выполняется.

ELT_ReloadGui

```
ELT_EXPORT void ELT_ReloadGui ()
```

Рестартует графическое приложение.

ELT_ResetBootParams

```
ELT_EXPORT int ELT_ResetBootParams ()
```

Сброс к заводским настройкам.



Временная функция. Когда сброс к заводским настройкам завершится, произойдёт событие ELT_SYNC_BOOT_PARAMS_DONE.

Возвращает:

Статус операции.

ELT_SetBootParamValue

```
ELT_EXPORT int ELT_SetBootParamValue (const char * key, const char * value)
```

Управление загрузочными параметрами приставки в формате ключ-значение. Приставка применяет эти параметры при загрузке. Настройки сохраняются во внутренней флэш-памяти. Когда запись будет завершена, сработает событие ELT_SYNC_BOOT_PARAMS_DONE. Поддерживаются следующие ключи: "dfbres" - разрешение пользовательского интерфейса.

Параметры:

key	ключ
value	значение

Возвращает:

Статус операции.

ELT_SyncBootParams

```
ELT_EXPORT int ELT_SyncBootParams ()
```



Функция устарела, см. описание ELT_SetBootParamValue(). Синхронизирует загрузочные параметры с флеш-накопителя. Перезагружать приставку можно только по событию ELT_SYNC_BOOT_PARAMS_DONE.

Возвращает:

Статус операции.

ELT_UpgradeFirmware

```
ELT_EXPORT int ELT_UpgradeFirmware (const char * path)
```

Обновление с локального диска.

Параметры:

path	путь до файла обновления "/mnt/stb/local/disk/img"
------	--

Возвращает:

Статус операции.

ELT_UpgradeStatus

```
ELT_EXPORT int ELT_UpgradeStatus (int * progress, int * err)
```

Возвращает статус обновления.



Функцию можно вызывать только по событию ELT_UPDATE_PROGRESS.

Параметры:

out	progress	прогресс (0..100%)
out	err	код ошибки, чтобы узнать тип ошибки нужно наложить маску ELT_UpdateStatusMask_e, далее, чтобы узнать причину ошибки нужно сравнить с ELT_UpdateStatusPrimitives_e.

Возвращает:

Статус операции.

2.4 Функции обратного вызова

Определения типов

- `typedef void(* clbk_fn)(int)`

Перечисления

- `enum ELT_Event_e { ELT_PLAYER_STATUS_CHANGED = 1, ELT_PLAYER_METADATA_UPDATED, ELT_PLAYER_POSITION_CHANGED, ELT_PLAYER_AUDIO_TRACK_SWITCHED, ELT_PLAYER_VIDEO_TRACK_SWITCHED, ELT_PLAYER_SUBTL_TRACK_SWITCHED, ELT_PLAYER_VOLUME_CHANGED, ELT_PLAYER_ZOOM_CHANGED, ELT_PLAYER_MUTE_CHANGED, ELT_PEERS_DOWNLOAD_PROGRESS, ELT_PLAYER_MULTICAST_GROUP_SWITCHED, ELT_NOTIFY_INPUT_DEVICE = 128, ELT_HDMI_STATE, ELT_MOUNT_EVENT, ELT_SWAP_AREA, ELT_ETH_LINK_STATE, ELT_UPDATE_FOUND, ELT_UPDATE_PROGRESS, ELT_UPDATE_FINISHED, ELT_USB_DEVICE_EVENT, ELT_SYNC_BOOT_PARAMS_DONE, ELT_GRAPHICS_VIEWPORT_CHANGED }`

2.4.1 Типы

typedef void

```
typedef void(* clbk_fn)(int)
```

Тип функции обратного вызова. Принимает идентификатор объекта как аргумент.

2.4.2 Перечисления

ELT_Event_e

```
enum ELT_Event_e
```

События, для которых можно зарегистрировать функцию обратного вызова.

Элементы перечислений:

- ***ELT_PLAYER_STATUS_CHANGED*** состояние плеера;
- ***ELT_PLAYER_METADATA_UPDATED*** метаданные получены;
- ***ELT_PLAYER_POSITION_CHANGED*** перемещение на новую позицию завершено;
- ***ELT_PLAYER_AUDIO_TRACK_SWITCHED*** переключение на новую аудиодорожку завершено;
- ***ELT_PLAYER_VIDEO_TRACK_SWITCHED*** переключение на новую видеодорожку завершено;
- ***ELT_PLAYER_SUBTL_TRACK_SWITCHED*** переключение на новую дорожку субтитров завершено;
- ***ELT_PLAYER_VOLUME_CHANGED*** громкость изменилась;

- **ELT_PLAYER_ZOOM_CHANGED** зум изменился;
- **ELT_PLAYER_MUTE_CHANGED** режим "без звука" изменился;
- **ELT_PEERS_DOWNLOAD_PROGRESS** прогресс загрузки файла из p2p изменился;
- **ELT_PLAYER_MULTICAST_GROUP_SWITCHED** состояние IGMP-изменилось; (LEAVE, JOIN)
- **ELT_NOTIFY_INPUT_DEVICE** не реализовано;
- **ELT_HDMI_STATE** не реализовано;
- **ELT_MOUNT_EVENT** не реализовано;
- **ELT_SWAP_AREA** не реализовано;
- **ELT_ETH_LINK_STATE** не реализовано;
- **ELT_UPDATE_FOUND** не реализовано;
- **ELT_UPDATE_PROGRESS** прогресс обновления изменился;
- **ELT_UPDATE_FINISHED** процесс обновления завершился;
- **ELT_USB_DEVICE_EVENT** не реализовано;
- **ELT_SYNC_BOOT_PARAMS_DONE** параметры приставки записаны на флеш-память;
- **ELT_GRAPHICS_VIEWPORT_CHANGED** область вывода интерфейса изменилась.

3 СТРУКТУРЫ ДАННЫХ

3.1 Структура ELT_Color_t

```
#include <EltexStbApiTypes.h>
```

Поля данных

- unsigned char **r**
- unsigned char **g**
- unsigned char **b**
- unsigned char **a**

Описание

Цвет в формате RGBA.

Поля

- unsigned char ELT_Color_t::a – прозрачность;
- unsigned char ELT_Color_t::b – синий;
- unsigned char ELT_Color_t::g – зелёный;
- unsigned char ELT_Color_t::r – красный.

3.2 Структура ELT_MulticastState_t

```
#include <EltexStbApiTypes.h>
```

Поля данных

- char **ipport** [32]
- ELT_MulticastState_e state

Описание

Состояние воспроизведения контента по мультикаст.

Поля

- char ELT_MulticastState_t::ipport[32] – мультикастовая группа и порт "226.0.0.1:8000";
- ELT_MulticastState_e ELT_MulticastState_t::state – IGMP состояние плеера по отношению к группе.

3.3 Структура ELT_Point_t

```
#include <EltexStbApiTypes.h>
```

Поля данных

- int x
- int y

Описание

Точка с координатами X и Y.

Поля

- int ELT_Point_t::x – координата X;
- int ELT_Point_t::y – координата Y.

3.4 Структура ELT_Size_t

```
#include <EltexStbApiTypes.h>
```

Поля данных

- int width
- int height

Описание

Размер прямоугольника.

Поля

- int ELT_Size_t::height – высота, в пикселях;
- int ELT_Size_t::width – ширина, в пикселях.

3.5 Структура ELT_Stream_t

```
#include <EltexStbApiTypes.h>
```

Поля данных

- int **id**
- char **descr** [128]
- int supported
- ELT_SubtitleEncoding_e encoding

Подробное описание

Дорожка (аудио, видео, субтитры).

Поля

- char ELT_Stream_t::descr[128] – краткое текстовое описание (кодек, разрешение и т.д.);
- ELT_SubtitleEncoding_e ELT_Stream_t::encoding – кодировка дорожки субтитров;
- int ELT_Stream_t::id – PID дорожки;
- int ELT_Stream_t::supported – поддерживается ли данный тип дорожки(1 да, 0 нет).

3.6 Структура ELT_Viewport_t

```
#include <EltexStbApiTypes.h>
```

Поля данных

- int **x**
- int **y**
- int width
- int height

Подробное описание

Область вывода.

Поля

- int ELT_Viewport_t::height – высота области вывода в пикселях;
- int ELT_Viewport_t::width – ширина области вывода в пикселях;
- int ELT_Viewport_t::x – координата X в пикселях;
- int ELT_Viewport_t::y – координата Y в пикселях.

ТЕХНИЧЕСКАЯ ПОДДЕРЖКА

Для получения технической консультации по вопросам эксплуатации оборудования ООО «Предприятие «Элтекс» Вы можете обратиться в Сервисный центр компании:

Российская Федерация, 630020, г. Новосибирск, ул. Окружная, дом 29 В.

Телефон:

+7(383)274-10-01

+7(383) 274-47-87

+7(383) 272-83-31

E-mail: techsupp@eltex.nsk.ru

На официальном сайте компании Вы можете найти техническую документацию и программное обеспечение для продукции ООО «Предприятие «Элтекс» или проконсультироваться у инженеров Сервисного центра на техническом форуме:

<http://eltex.nsk.ru>

<http://eltex.nsk.ru/support/documentations>

<http://eltex.nsk.ru/forum>

