

MECodecAPI Reference Manual

2015 年 9 月 1 日
MEDIAEDGE 株式会社

目次

目次	1
はじめに	3
概要	4
サポート環境	5
関数	6
MEC_OpenDevice	7
MECDevice_Release	8
MECDevice_GetFirmVersion	9
MECDevice_GetCoreTemperature	10
MECDevice_GetHDMIOutput	11
MECDevice_SetHDMIOutput	12
MECDevice_GetSPDIFOutput	13
MECDevice_SetSPDIFOutput	14
MECDevice_CreateDecoder	15
MECDecoder_Release	16
MECDecoder_Start	17
MECDecoder_Stop	18
MECDecoder_Pause	19
MECDecoder_SendStream	20
MECDecoder_GetState	21
MECDecoder_GetElapsedTime	22
MECDecoder_GetParam	23
MECDecoder_SetParam	24
MECDecoder_GetEventCallback	25
MECDecoder_SetEventCallback	26
構造体	27
MEC_HDMI_OUTPUT	28
MEC_SPDIF_OUTPUT	29
MEC_RESOLUTION	30
MEC_FRACTION	31
MEC_BUFFER	32
MEC_VIDEO_PARAM	33
MEC_AUDIO_PARAM	34
MEC_EXT_TS_PARAM	35
MEC_EXT_ES_PARAM	36
MEC_STREAM_PARAM	37

MEC_PACKET	38
MEC_EVENT_CALLBACK	39
列举值.....	40
MEC_ERROR.....	41
MEC_VIDEO_FORMAT	42
MEC_CODEC_ID.....	44
MEC_STREAM_TYPE.....	45
MEC_STATE.....	46
MEC_EVENT.....	47

はじめに

概要

本 SDK は、ME-DEC5000HDMI を使用したアプリケーションの開発を目的としたプログラムです。SDK で利用できる機能は以下の通りです。

- 対応ビデオコーデックは H265/HEVC、H264/AVC。
- 対応オーディオコーデックは AAC。
- トランスポートストリーム形式での入力。
- エレメンタリーストリーム形式での入力。(注: MP4 から抽出したストリームのみ対応)

サポート環境

SDK がサポートする環境は以下の通りです。

OS	Windows 7 (64bit)、Windows 8.1 (64bit)
開発環境	Visual Studio 2013

MECodecAPI に関しては Visual Studio 2013 Express でもお使いいただけますが、付属のサンプルプログラムをビルドするには、Professional 以上のバージョンが必要です。

また、本 SDK で提供されるライブラリは全て 64bit でビルドしたものです。32bit 版 DLL は含んでおりません。開発の際は、必ずプロジェクト構成のプラットフォームを x64 にしてください。

開発したアプリケーションを実行する際は、以下のファイルを同じフォルダに置いてください。

- MECodecAPI.dll
- PCIePXsCOM.dll
- PCIePXsEx.dll

また、上記のファイルに加えて Visual Studio 2013 のランタイムライブラリが必要です。

関数

MEC_OpenDevice

デバイスに接続します。

■ 定義

```
MEC_ERROR MEC_EXPORT MEC_OpenDevice(  
    ULONG deviceID,  
    MEC_DEVICE *pDevice  
);
```

■ 引数

- [in] deviceID
0 から始まるデバイス ID。
- [out] pDevice
デバイスオブジェクトへのポインター。

MECDevice_Release

デバイスオブジェクトを解放し、デバイスとの接続を閉じます。

■ 定義

```
MEC_ERROR MEC_EXPORT MECDevice_Release(  
    MEC_DEVICE device  
);
```

■ 引数

- [in]device
デバイスオブジェクト。

MECDevice_GetFirmVersion

ファームのバージョン情報を取得します。

■ 定義

```
MEC_ERROR MEC_EXPORT MECDevice_GetFirmVersion(  
    MEC_DEVICE device,  
    ULONG *pValue  
);
```

■ 引数

- [in] device
デバイスオブジェクト。
- [out] pValue
バージョン情報へのポインタ。

MECDevice_GetCoreTemperature

デバイスのコア温度を取得します。

■ 定義

```
MEC_ERROR MEC_EXPORT MECDevice_GetCoreTemperature(  
    MEC_DEVICE device,  
    INT *pValue  
);
```

■ 引数

- [in] device
デバイスオブジェクト。
- [out] pValue
温度値へのポインタ。

MECDevice_GetHDMIOutput

HDMI 出力の設定を取得します。

■ 定義

```
MEC_ERROR MEC_EXPORT MECDevice_GetHDMIOutput(  
    MEC_DEVICE device,  
    MEC_HDMI_OUTPUT *pValue  
);
```

■ 引数

- [in] device
デバイスオブジェクト。
- [out] pValue
HDMI 出力の設定値へのポインタ。

MECDevice_SetHDMIOutput

HDMI 出力を設定します。

■ 定義

```
MEC_ERROR MEC_EXPORT MECDevice_SetHDMIOutput(  
    MEC_DEVICE device,  
    const MEC_HDMI_OUTPUT *pValue  
);
```

■ 引数

- [in] device
デバイスオブジェクト。
- [in] pValue
HDMI 出力の設定値へのポインタ。

MECDevice_GetSPDIFOutput

SPDIF 出力の設定を取得します。

■ 定義

```
MEC_ERROR MEC_EXPORT MECDevice_GetSPDIFOutput(  
    MEC_DEVICE device,  
    MEC_SPDIF_OUTPUT *pValue  
);
```

■ 引数

- [in] device
デバイスオブジェクト。
- [out] pValue
SPDIF 出力の設定値へのポインタ。

MECDevice_SetSPDIFOutput

SPDIF 出力を設定します。

■ 定義

```
MEC_ERROR MEC_EXPORT MECDevice_SetSPDIFOutput(  
    MEC_DEVICE device,  
    const MEC_SPDIF_OUTPUT *pValue  
);
```

■ 引数

- [in] device
デバイスオブジェクト。
- [in] pValue
SPDIF 出力の設定値へのポインタ。

MECDevice_CreateDecoder

デコーダーオブジェクトを作成します。

■ 定義

```
MEC_ERROR MEC_EXPORT MECDevice_CreateDecoder(  
    MEC_DEVICE device,  
    MEC_DECODER *pDecoder  
);
```

■ 引数

- [in] device
デバイスオブジェクト。
- [out] pDecoder
デコーダーオブジェクトへのポインタ。

MECDecoder_Release

デコーダーオブジェクトを破棄します。

■ 定義

```
MEC_ERROR MEC_EXPORT MECDecoder_Release(  
    MEC_DECODER decoder  
);
```

■ 引数

- [in] decoder
デコーダーオブジェクト。

MECDecoder_Start

デコードを開始します。

■ 定義

```
MEC_ERROR MEC_EXPORT MECDecoder_Start(  
    MEC_DECODER decoder  
);
```

■ 引数

- [in] decoder
デコーダーオブジェクト。

■ 補足

状態が MEC_STATE_READY もしくは MEC_STATE_PAUSED でないと再生は開始されません。状態が MEC_STATE_STOPPED の場合は、一度、MECDecoder_SetParam 関数でデコードパラメーターを設定してから実行してください。

MECDecoder_Stop

デコードを停止します。

■ 定義

```
MEC_ERROR MEC_EXPORT MECDecoder_Stop(  
    MEC_DECODER decoder  
);
```

■ 引数

- [in] decoder
デコーダーオブジェクト。

■ 補足

デコードを停止すると、MECDecoder_GetElapseTime で取得できる時間は 0 に戻ります。

MECDecoder_Pause

デコードを一時停止します。

■ 定義

```
MEC_ERROR MEC_EXPORT MECDecoder_Pause(  
    MEC_DECODER decoder  
);
```

■ 引数

- [in] decoder
デコーダーオブジェクト。

■ 補足

デコードを再開するには MECDecoder_Start 関数を実行します。

MECDecoder_SendStream

デコーダーオブジェクトにデコードさせるデータを入力します。

■ 定義

```
MEC_ERROR MEC_EXPORT MECDecoder_SendStream(  
    MEC_DECODER decoder,  
    const MEC_PACKET *pPacket  
);
```

■ 引数

- [in] decoder
デコーダーオブジェクト。
- [in] pPacket
デコードさせるデータ。

■ 補足

トランスポートストリーム形式でデコードさせる場合は、pPacket の PID と Time の値は無視されます。トランスポートストリーム形式の場合、一度に入力するデータのサイズが小さすぎると、処理効率が低下してしまい、再生が遅延する可能性があります。Ver1.0 時点で、もっとも効率が良い入力サイズは 32768 バイトです。エレメンタリーストリーム形式に関しては、特に上記のような制限はありません。引数の pPacket に NULL を指定すると、デコーダーにストリームの終端(EOS)を入力します。デコーダーは EOS が入力された時点までのデータを出力した後、状態を一旦リセットします。この関数は、デコーダーへの入力バッファが満杯になると、バッファに空きができるまで処理をブロックします。ブロックされた処理を中断し、この関数から抜けたい場合は、別スレッドより MECDecoder_Stop 関数を実行してください。

MECDecoder_GetState

デコーダーオブジェクトの状態を取得します。

■ 定義

```
MEC_ERROR MEC_EXPORT MECDecoder_GetState(  
    MEC_DECODER decoder,  
    MEC_STATE *pState  
);
```

■ 引数

- [in] decoder
デコーダーオブジェクト。
- [out] pState
状態を示す値へのポインタ。

MECDecoder_GetElapseTime

デコードを開始してからの経過時間を取得します。

■ 定義

```
MEC_ERROR MEC_EXPORT MECDecoder_GetElapseTime(  
    MEC_DECODER decoder,  
    LONGLONG *pMSec  
);
```

■ 引数

- [in] decoder
デコーダーオブジェクト。
- [out] pMSec
経過時間を示す値へのポインタ。単位はミリ秒です。

■ 補足

MECDecoder_Stop 関数を実行する、もしくは MECDecoder_SendStream 関数で EOS を入力する (pPacket を NULL にする) と 0 にリセットされます。

MECDecoder_GetParam

現在のデコードパラメーターを取得します。

■ 定義

```
MEC_ERROR MEC_EXPORT MECDecoder_GetParam(  
    MEC_DECODER decoder,  
    MEC_STREAM_PARAM *pParam  
);
```

■ 引数

- [in] decoder
デコーダーオブジェクト。
- [out] pParam
デコードパラメーターを示す値へのポインタ。

MECDecoder_SetParam

デコードパラメーターを設定します。

■ 定義

```
MEC_ERROR MEC_EXPORT MECDecoder_SetParam(  
    MEC_DECODER decoder,  
    const MEC_STREAM_PARAM *pParam  
);
```

■ 引数

- [in] decoder
デコーダーオブジェクト。
- [in] pParam
デコードパラメーターを示す値へのポインタ。

■ 補足

状態が MEC_STATE_STOPPED の時に実行すると、パラメーターは即時にデバイスに反映され、MEC_STATE_READY に状態遷移します。

それ以外の状態(MEC_STATE_UNKNOWN は除く)の時に実行すると、渡されたパラメーターは予約として扱われ、状態が MEC_STATE_EOS になったときに反映されます。予約されたパラメーターが反映されると、MEC_EVENT_UPDATE_PARAM というイベントをコールバックします。

すでにパラメーターが予約されている場合、この関数はエラーを返します(予約の多重登録や上書きはできません)。一度、MECDecoder_Stop 関数を実行して、状態を MEC_STATE_STOPPED にするか、予約されたパラメーターがデバイスに反映されるのを待ってから実行してください。

MECDecoder_GetEventCallback

現在のイベントに関するコールバックの設定を取得します。

■ 定義

```
MEC_ERROR MEC_EXPORT MECDecoder_GetEventCallback(  
    MEC_DECODER decoder,  
    MEC_EVENT_CALLBACK *pCallback  
);
```

■ 引数

- [in] decoder
デコーダーオブジェクト。
- [out] pCallback
コールバックの設定を示す値へのポインタ。

MECDecoder_SetEventCallback

イベントに関するコールバックを設定します。

■ 定義

```
MEC_ERROR MEC_EXPORT MECDecoder_SetEventCallback(  
    MEC_DECODER decoder,  
    const MEC_EVENT_CALLBACK *pCallback  
);
```

■ 引数

- [in] decoder
デコーダーオブジェクト。
- [in] pCallback
コールバックの設定を示す値へのポインタ。

構造体

MEC_HDMI_OUTPUT

デバイスの HDMI 出力の設定です。

■ 定義

```
typedef struct {  
    struct {  
        ULONG Format;  
        ULONG RefreshRate  
    } Video;  
    struct {  
        BOOL Valid;  
    } Audio;  
} MEC_HDMI_OUTPUT;
```

■ 要素

- Video.Format
HDMI 出力のビデオ形式。
- Audio.Valid;
HDMI 出力の音声の有無。

■ 補足

音声に関して、HDMI 出力と SPDIF 出力は排他仕様です。

MEC_SPDIF_OUTPUT

デバイスの SPDIF 出力の設定です。

■ 定義

```
typedef struct {  
    struct {  
        BOOL Valid;  
    } Audio;  
} MEC_SPDIF_OUTPUT;
```

■ 要素

- Audio.Valid
SPDIF 出力のオーディオの有無。

■ 補足

音声に関して、HDMI 出力と SPDIF 出力は排他仕様です。

MEC_RESOLUTION

解像度を示す値です。

■ 定義

```
typedef struct {  
    LONG Width;  
    LONG Height;  
} MEC_RESOLUTION;
```

■ 要素

- Width
幅。単位はピクセルです。
- Height
高さ。単位はピクセルです。

MEC_FRACTION

分数を示す値です。

■ 定義

```
typedef struct {  
    LONG Num;  
    LONG Den;  
} MEC_FRACTION;
```

■ 要素

- Num
分子。
- Den
分母。

MEC_BUFFER

バイナリーデータを示す値です。

■ 定義

```
typedef struct {  
    BYTE *Ptr;  
    ULONG Size;  
} MEC_BUFFER;
```

■ 要素

- Ptr
バイナリーデータへのポインタ。
- Size
バイナリーデータのサイズ。

MEC_VIDEO_PARAM

ビデオに関する設定値です。

■ 定義

```
typedef struct {  
    ULONG PID;  
    ULONG Codec;  
    MEC_RESOLUTION Resolution;  
    MEC_FRACTION FrameRate;  
    MEC_FRACTION TimeBase;  
} MEC_VIDEO_PARAM;
```

■ 要素

- PID
ビデオを識別するための PID。
- Codec
ビデオのコーデック。MEC_CODEC_ID を指定します。
- Resolution
解像度。
- FrameRate
フレームレート。
- TimeBase
単位時間。

MEC_AUDIO_PARAM

オーディオに関する設定値です。

■ 定義

```
typedef struct {
    ULONG PID;
    ULONG Codec;
    ULONG BitsPerSample;
    ULONG Channels;
    ULONG SamplingRate;
    MEC_FRACTION TimeBase;
} MEC_AUDIO_PARAM;
```

■ 要素

- PID
オーディオを識別するための PID。
- Codec
オーディオのコーデック。MEC_CODEC_ID を指定します。
- BitsPerSample
1 サンプルあたりのビット数。
- Channels
チャンネル数。
- SamplingRate
サンプリングレート。
- TimeBase
単位時間。

MEC_EXT_TS_PARAM

トランスポートストリームをデコードする際の設定値です。

■ 定義

```
typedef struct {  
    ULONG PacketSize;  
} MEC_EXT_TS_PARAM;
```

■ 要素

- PacketSize
TS パケットのサイズ。

MEC_EXT_ES_PARAM

エレメンタリーストリームをデコードする際の設定値です。

■ 定義

```
typedef struct {  
    MEC_BUFFER ExtraData;  
} MEC_EXT_ES_PARAM;
```

■ 要素

- ExtraData
SPS や PPS といったパラメーターを含んだバイナリーデータ。

MEC_STREAM_PARAM

デコードに関する設定値です。

■ 定義

```
typedef struct {
    ULONG StreamType;
    MEC_VIDEO_PARAM Video;
    MEC_AUDIO_PARAM Audio;
    union {
        MEC_EXT_TS_PARAM TS;
        MEC_EXT_ES_PARAM ES;
    };
} MEC_STREAM_PARAM;
```

■ 要素

- StreamType
ストリーム形式。
- Video
ビデオに関する設定値。
- Audio
オーディオに関する設定値。
- TS
トランスポートストリームに関する設定値。
- ES
エレメンタリーストリームに関する設定値。

MEC_PACKET

デコーダーに入力するデータ。

■ 定義

```
typedef struct {  
    ULONG PID;  
    LONGLONG Time;  
    MEC_BUFFER Data;  
} MEC_PACKET;
```

■ 要素

- PID
データが属するストリームを識別するための PID。
- Time
データの開始時間。ここで指定する時間の単位は MEC_VIDEO_PARAM や MEC_AUDIO_PARAM の TimeBase で指定した値です。
- Data
デコードさせるデータ。

MEC_EVENT_CALLBACK

イベントコールバックに関する設定値です。

■ 定義

```
typedef struct {  
    void (*Function)(ULONG, void*);  
    void *UserData;  
} MEC_EVENT_CALLBACK;
```

■ 要素

- Function
コールバック時に呼び出される関数。
- UserData
コールバック時に渡されるユーザー変数。

列举值

MEC_ERROR

エラーコードです。

■ 要素

- MEC_ERROR_SUCCEEDED
成功。
- MEC_ERROR_FAILED
失敗。
- MEC_ERROR_DEVICE
デバイスエラー。
- MEC_ERROR_DRIVER
ドライバーエラー。
- MEC_ERROR_INVALID_VALUE
無効な引数によるエラー。
- MEC_ERROR_TIMEOUT
タイムアウトによるエラー。
- MEC_ERROR_STATE
状態遷移によるエラー。
- MEC_ERROR_UNEXCEPT
不明なエラー。

MEC_VIDEO_FORMAT

HDMI 出力のビデオ形式です。

■ 要素

- MEC_VIDEO_FORMAT_UNKNOWN
不明な形式。
- MEC_VIDEO_FORMAT_AUTO
自動選択。
- MEC_VIDEO_FORMAT_RGB
RGB(4K)形式。
- MEC_VIDEO_FORMAT_YUV420
YUV420(8bit)形式。
- MEC_VIDEO_FORMAT_YUV420_10
YUV420(10bit)形式
- MEC_VIDEO_FORMAT_YUV422
YUV422 形式。現在、未サポート。
- MEC\VIDEO_FORMAT_RGB_FHD
RGB(2K)形式

MEC_REFRESH_RATE

HDMI 出力のリフレッシュレートです。

■ 要素

- MEC_REFRESH_RATE_UNKNOWN
不明なレート。
- MEC_REFRESH_RATE_AUTO
自動選択。
- MEC_REFRESH_RATE_23_98
23.98Hz。現在、未サポート。
- MEC_REFRESH_RATE_24
24Hz。現在、未サポート。
- MEC_REFRESH_RATE_25
25Hz。現在、未サポート。
- MEC_REFRESH_RATE_29_97
29.97Hz。現在、未サポート。
- MEC_REFRESH_RATE_30
30Hz。現在、未サポート。
- MEC_REFRESH_RATE_50
50Hz。
- MEC_REFRESH_RATE_59_94
59.94Hz。
- MEC_REFRESH_RATE_60
60Hz。

MEC_CODEC_ID

コーデックを示す ID です。

■ 要素

- MEC_CODEC_ID_UNKNOWN
不明なコーデック。
- MEC_CODEC_ID_NONE
ビデオ、もしくは音声なし。
- MEC_CODEC_ID_H264
H.264/AVC。
- MEC_CODEC_ID_H265
H.265/HEVC。
- MEC_CODEC_ID_AAC
AAC。

MEC_STREAM_TYPE

ストリーム形式です。

■ 要素

- MEC_STREAM_TYPE_UNKNOWN
不明な形式。
- MEC_STREAM_TYPE_TS
トランスポートストリーム形式。
- MEC_STREAM_TYPE_ES
エレメンタリーストリーム形式。

MEC_STATE

デコーダーの状態です。

■ 要素

- MEC_STATE_UNKNOWN
不明な状態。
- MEC_STATE_STOPPED
停止中。
- MEC_STATE_EOS
映像終端に到達(停止)。
- MEC_STATE_READY
再生準備完了。
- MEC_STATE_PAUSED
一時停止中。
- MEC_STATE_STARTED
再生中。

MEC_EVENT

デコーダーから通知されるイベント値です。

■ 要素

- MEC_EVENT_UNKNOWN
不明なイベント。
- MEC_EVENT_START
デコードを開始しました。
- MEC_EVENT_PAUSE
デコードを一時停止しました。
- MEC_EVENT_STOP
デコードを停止しました。
- MEC_EVENT_END_OF_STREAM
ストリームの終端に到達しました。
- MEC_EVENT_UPDATE_PARAM
予約していたデコードパラメーターがデバイスに反映されました
- MEC_EVENT_DECODE_ERROR
デバイス内でエラーが発生しました。