

Magewell Pro Capture Series Card

SDK Guide for Linux

Copyright 2011–2016 [Nanjing Magewell Electronics Co., Ltd.](#) All rights reserved.

Revised 17/02/2016

1 Introduction

This section mainly introduces how to develop the application software base on the Pro Capture Series card in Linux platform.

1.1 Program Interface

In general, developers usually use two main kinds of development programming interface:

- V4L2 and ALSA API
 - Using standard interface to realize the video and audio data capture;
 - Supporting the standard capture control. Including the sampling size, color space, and other parameter settings;
- I/O Control API
 - Realizing the I/O Control extended function which basing on V4L2 driver;
 - The capture features will be more flexibly and abundantly;

Both of methods can be used either separating or combining, developer can choose by the actually requirement.

1.2 I/O Control Features

The software, which is basing on the I/O Control interface to develop, can achieve the following features:

- Low latency image data capture;

Branch of image data transmission, the latency time required to input 64 lines image signals.

- Image data output with multi-streaming;

For the same capture devices can create multiple instances, each instance has different capture size and color space, each capture device keeps 400MB–700MB exclusive transmission bandwidth. User can feel flexible while using.

- Support 10 bits color format of the image capture:

Support AYUV、V408、Y410、V410、RGB10、BGR10 color format for capturing image.

- Scheduling the device Clock with high-precision:

Accuracy of 100 nanoseconds on-board clock provides the time information for all the captured video frame, and supports the clock synchronization processing in multi capture devices on the Internet.

- Hardware synthesis of OSD image data:

Support RGBA format text or image through hardware real-time synthesis to the capture images, and it can implement the multiple capture image synthesis and superposition, form the PIP and PAP effects.

- Obtaining HDMI InfoFrame data package;
- Support for obtaining HDMI original InfoFrame data package, including AVI、Audio、SPD、MS、VS、ACP and etc. Developer can obtain the various extensions information from these data.
- Audio capture with 8 channels:

Support for capturing audio data through IEC60958 / IEC61937 format , including the original audio format (5.1 channel, 7.1 channel、DTS、THX、SRS, etc.) and all kinds of compressed audio formats(AAC,MP3,etc.)

- Firmware on board upgrade:

Support for upgrading onboard firmware through API. The deep integration and the remote upgrade features will be easy to be put into effect.

2 Sample Introduction

This section mainly introduces the various of sample code in the SDK. There are two types of the sample code;

2.1 I/O Control Samples

- Lib source file saved in lib folder
- Demo source file saved in src folder;
- In sdk folder, using “make” to compile all the examples, the executable file will be generated in bin folder;
- Including kinds of example programs, covering various contents of I/O Control development;
 - mwcap-upgrade : Demonstrating to read MWF firmware update file, then write these data in capture device for finish firmware updating operation;
 - mwcap-info : Demonstrating get Pro Capture device information, such as input signal information, device hardware information and HDMI info frame;
 - mwcap-video-capture : Demonstrating the capture devices information and video capture data, then saving these data as BMP file;
 - mwcap-audio-capture : Demonstrating the capture devices information and audio capture data, then saving these data as WAV file;
 - mwcap-video-set : Demonstrating the input port information from the video channel of the capture device, and also can set the choice for the port;
 - mwcap-audio-set : Demonstrating the input port information from the audio channel of the capture device, and also can set the choice for the port;
 - mwcap-lowlatency : Demonstrating for the input progressive signal, using low latency mode to do the video block capture;

3 I/O Control Developing Interface

This section mainly introduces the I/O Control programming interface. The mainly development as per the below steps:

3.1 Developing Steps

I/O Control development is basing the steps as below:

- Using open() to open /dev/video* video equipment (* is the number which start form 0);
- Using ioctl() function to call the difference interface of I/O Control;
- Using close() function to close video equipment;

3.2 I/O Control Instructions

I/O Control control instructions is defined in the src\public\mw-linux.h,it including following instructions:

3.2.1 Get Device Stream Information Instructions

After installing the device driver, Pro Capture Series devices will generate several video* devices(* is the number which start form 0) in the /dev folder. Each of the video* device corresponds to a capture channel, each channel can be open several times at the same time, the device stream is defined to open some capture channel for one time. The following control instructions are using for getting the detailed information of the device stream of the current device.

NAME	ANNOTATION
MWCAP_IOCTL_GET_STREAMS_COUNT	Obtaining the device streaming count of the current device
MWCAP_IOCTL_GET_STREAMS_INFO	Obtaining the device streaming detail of the current device. Reference to 3.3.2 MWCAP_STREAMS_INFO
MWCAP_IOCTL_GET_CTRL_STREAM_ID	Obtaining the device streaming ID of current controlled device
MWCAP_IOCTL_SET_CTRL_STREAM_ID	Setting the device streaming ID of current controlled device
MWCAP_IOCTL_GET_SELF_STREAM_ID	Obtaining the device streaming ID of current device

3.2.2 Get Device Information Instructions;

NAME	ANNOTATION
MWCAP_IOCTL_GET_CHANNEL_INFO	Obtaining the current capture channel information. Reference to 3.3.3 MWCAP_CHANNEL_INFO
MWCAP_IOCTL_GET_FAMILY_INFO	Obtaining the device type information of the current capture channel. Reference to 3.3.4 MWCAP_PRO_CAPTURE_INFO
MWCAP_IOCTL_GET_VIDEO_CAPS	Obtaining the support features of the current video channel. Reference to 3.3.5 MWCAP_VIDEO_CAPS
MWCAP_IOCTL_GET_AUDIO_CAPS	Obtaining the support features of the current audio channel. Reference to 3.3.6 MWCAP_AUDIO_CAPS

3.2.3 Firmware Update Instructions;

NAME	ANNOTATION
MWCAP_IOCTL_GET_FIRMWARE_STORAGE	Writing the firmware storage information of the current channel

MWCAP_IOCTL_SET_FIRMWARE_ERASE	el. Reference to 3.3.7 MWCAP_FIRMWARE_STORAGE Erasing the firmware storage area of the current channel. Reference to 3.3.8 MWCAP_FIRMWARE_ERASE
MWCAP_IOCTL_SET_FIRMWARE_DATA	Writing data to the firmware storage area of the current channel. Reference to 3.3.9 MWCAP_FIRMWARE_DATA
MWCAP_IOCTL_GET_FIRMWARE_DATA	Reading data from the firmware storage area of the current channel. Reference to 3.3.9 MWCAP_FIRMWARE_DATA

3.2.4 Firmware Clock Instructions;

NAME	ANNOTATION
MWCAP_IOCTL_GET_TIME	Obtaining the hardware clock of the current channel
MWCAP_IOCTL_SET_TIME	Setting the hardware clock of the current channel
MWCAP_IOCTL_TIME_REGULATION	Regulate the hardware clock of the current channel
MWCAP_IOCTL_TIMER_REGISTRATION	Register clock scheduling object
MWCAP_IOCTL_TIMER_DEREGISTRATION	Deregister clock scheduling object
MWCAP_IOCTL_TIMER_EXPIRE_TIME	Setting expire time of the clock scheduling object. Reference to 3.3.10 MWCAP_TIMER_EXPIRE_TIME
MWCAP_IOCTL_TIMER_WAIT	Waiting for the clock scheduling object failure. Reference to 3.3.11 MWCAP_TIMER_WAIT
MWCAP_IOCTL_TIMER_CANCEL	Cancel the latest waiting time of the clock scheduling object

3.2.5 Asynchronous Notification Instructions;

NAME	ANNOTATION
MWCAP_IOCTL_NOTIFY_REGISTRATION	Register the notification object of asynchronous event. Reference to 3.3.12 MWCAP_NOTIFY_REGISTRATION_S
MWCAP_IOCTL_NOTIFY_DEREGISTRATION	Deregister the notification object of asynchronous event
MWCAP_IOCTL_NOTIFY_STATUS	Obtaining the event notification status. Reference to 3.3.13 MWCAP_NOTIFY_STATUS
MWCAP_IOCTL_NOTIFY_WAIT	Waiting the asynchronous event notification, Reference to 3.3.14 MWCAP_NOTIFY_WAIT
MWCAP_IOCTL_NOTIFY_CANCEL	Cancel the latest waiting asynchronous event notification
MWCAP_IOCTL_NOTIFY_ENABLE	Setting the type of the asynchronous event to be available , Reference to 3.3.15 MWCAP_NOTIFY_ENABLE

3.2.6 Audio And Video Capture Instructions;

名称	注释
MWCAP_IOCTL_GET_VIDEO_BUFFER_INFO	Obtaining video buffer information. Reference to 3.3.16 MWCAP_VIDEO_BUFFER_INFO
MWCAP_IOCTL_GET_VIDEO_FRAME_INFO	Obtaining video frame information. Reference to 3.3.17 MWCAP_VIDEO_FRAME_INFO
MWCAP_IOCTL_VIDEO_CAPTURE_OPEN	Open video capture channel

MWCAP_IOCTL_VIDEO_CAPTURE_FRAME	Capturing one video frame data.Reference to 3.3.18
MWCAP_IOCTL_VIDEO_CAPTURE_STATUS	Obtaining the video capture status.Reference to 3.3.19 MWCAP_VIDEO_CAPTURE_STATUS
MWCAP_IOCTL_VIDEO_CAPTURE_CLOSE	Close video capture channel
MWCAP_IOCTL_AUDIO_CAPTURE_OPEN	Open audio capture channel
MWCAP_IOCTL_AUDIO_CAPTURE_FRAME	Capture one audio frame data.Reference to 3.3.20 MWCAP_AUDIO_CAPTURE_FRAME
MWCAP_IOCTL_AUDIO_CAPTURE_CLOSE	Close audio capture channel

3.2.7 Capture Channel Input Instructions;

名称	注释
MWCAP_IOCTL_VIDEO_INPUT_SOURCE_COUNT	Obtaining the input interface count of the video channel
MWCAP_IOCTL_VIDEO_INPUT_SOURCE_ARRAY	Obtaining the input interface of the video channel.Regerence to 3.3.21 MWCAP_INPUT_SOURCE_ARRAY
MWCAP_IOCTL_AUDIO_INPUT_SOURCE_COUNT	Obtaining the input interface count of the audio channel
MWCAP_IOCTL_AUDIO_INPUT_SOURCE_ARRAY	Obtaining the input interface of the audio channel.Regerence to3.3.21 MWCAP_INPUT_SOURCE_ARRAY
MWCAP_IOCTL_SET_INPUT_SOURCE_SCAN	Setting the video input whether to automatically scan
MWCAP_IOCTL_GET_INPUT_SOURCE_SCAN	Obtaining the video input whether to automatically scan
MWCAP_IOCTL_GET_INPUT_SOURCE_SCAN_STATE	Obtaining the scanning state of the input interface of the capture channel
MWCAP_IOCTL_SET_AV_INPUT_SOURCE_LINK	Setting the audio input whether to automatically link to the video input
MWCAP_IOCTL_GET_AV_INPUT_SOURCE_LINK	Obtaining the audio input whether to automatically link to the video input
MWCAP_IOCTL_SET_VIDEO_INPUT_SOURCE	Setting the current video input source in this channel.Reference to 3.4.16 MWCAP_VIDEO_INPUT_TYPE
MWCAP_IOCTL_GET_VIDEO_INPUT_SOURCE	Obtaining the current video input source in this channel.Reference to3.4.16 MWCAP_VIDEO_INPUT_TYPE
MWCAP_IOCTL_SET_AUDIO_INPUT_SOURCE	Setting the current audio input source in this channel.Reference to3.4.17 MWCAP_AUDIO_INPUT_TYPE
MWCAP_IOCTL_GET_AUDIO_INPUT_SOURCE	Obtaining the current audio input source in this channel.Reference to3.4.17 MWCAP_AUDIO_INPUT_TYPE

3.2.8 Signal Status Instructions;

NAME	ANNOTATION
MWCAP_IOCTL_GET_INPUT_SPECIFIC_STATUS	Obtaining the input signal status of this channel.Reference to 3.3.28 MWCAP_INPUT_SPECIFIC_STATUS

MWCAP_IOCTL_GET_VIDEO_SIGNAL_STATUS	Obtaining the video signal status.Reference to 3.3.29 MWCA P_VIDEO_SIGNAL_STATUS
MWCAP_IOCTL_GET_AUDIO_SIGNAL_STATUS	Obtaining the audio signal status.Reference to 3.3.30 MWCA P_AUDIO_SIGNAL_STATUS
MWCAP_IOCTL_GET_HDMI_INFOFRAME_VALID	Obtaining the valid HDMI information frame.Reference to 3.4.13 MWCAP_HDMI_INFOFRAME_ID
MWCAP_IOCTL_GET_HDMI_INFOFRAME_PACKET	Obtaining HDMI information frame data.Reference to 3.3.32 HDMI_INFOFRAME_PACKET

3.2.9 Video Process Instructions

NAME	ANNOTATION
MWCAP_IOCTL_GET_VIDEO_INPUT_ASPECT_RATIO	Obtaining the video input aspect ratio.Reference to 3.3.33 M WCAP_VIDEO_ASPECT_RATIO
MWCAP_IOCTL_SET_VIDEO_INPUT_ASPECT_RATIO	Setting the video input aspect ratio.Reference to 3.3.33 MWC AP_VIDEO_ASPECT_RATIO
MWCAP_IOCTL_GET_VIDEO_INPUT_COLOR_FORMAT	Obtaining the video input color format.Reference to 3.4.4 MW CAP_VIDEO_COLOR_FORMAT
MWCAP_IOCTL_SET_VIDEO_INPUT_COLOR_FORMAT	Setting the video input color format.Reference to 3.4.4 MWC AP_VIDEO_COLOR_FORMAT
MWCAP_IOCTL_GET_VIDEO_INPUT_QUANTIZATION_RANGE	Obtaining the video input quantization range.Reference to 3.4.5 MWCAP_VIDEO_QUANTIZATION_RANGE
MWCAP_IOCTL_SET_VIDEO_INPUT_QUANTIZATION_RANGE	Setting the video input quantization range.Reference to 3.4.5 MWCAP_VIDEO_QUANTIZATION_RANGE

3.2.10 OSD Instructions

NAME	ANNOTATION
MWCAP_IOCTL_VIDEO_CREATE_IMAGE	Create OSD image.Reference to 3.3.34 MWCAP_VIDEO_C REATE_IMAGE
MWCAP_IOCTL_VIDEO_OPEN_IMAGE	Open OSD image.Reference to 3.3.35 MWCAP_VIDEO_IM AGE_REF
MWCAP_IOCTL_VIDEO_CLOSE_IMAGE	Close OSD image.Reference to 3.3.35 MWCAP_VIDEO_IM AGE_REF
MWCAP_IOCTL_VIDEO_UPLOAD_IMAGE	Upload OSD image.Reference to 3.3.36 MWCAP_VIDEO_U PLOAD_IMAGE

3.2.11 Timing sequence instructions

NAME	ANNOTATION
MWCAP_IOCTL_GET_VIDEO_AUTO_HORIZONTAL_ALIGN	Obtaining the video auto adjust on horizontal direction
MWCAP_IOCTL_SET_VIDEO_AUTO_HORIZONTAL_ALIGN	Setting the video auto adjust on horizontal direction
MWCAP_IOCTL_GET_VIDEO_SAMPLING_PHASE	Obtaining the size of the video sampling phase
MWCAP_IOCTL_SET_VIDEO_SAMPLING_PHASE	Setting the size of the video sampling phase

MWCAP_IOCTL_GET_VIDEO_SAMPLING_PHASE_AUTO	Obtaining the video sampling phase whether to automatic adjustment
MWCAP_IOCTL_SET_VIDEO_SAMPLING_PHASE_AUTO	Setting the video sampling phase whether to automatic adjustment
MWCAP_IOCTL_SET_VIDEO_TIMING	Setting the video timing sequence information.Reference to 3.3.37 MWCAP_VIDEO_TIMING
MWCAP_IOCTL_GET_VIDEO_PREFERRED_TIMING_ARRAY	Obtaining the index of the video preferred timing sequence. Reference to 3.3.37 MWCAP_VIDEO_TIMING
MWCAP_IOCTL_SET_VIDEO_CUSTOM_TIMING	Setting the video custom timing sequence parameter.Reference to 3.3.38 MWCAP_VIDEO_CUSTOM_TIMING
MWCAP_IOCTL_GET_VIDEO_CUSTOM_TIMING_COUNT	Obtaining the video custom timing sequence count
MWCAP_IOCTL_GET_VIDEO_CUSTOM_TIMING_ARRAY	Obtaining the video custom timing sequence.Reference to 3.3.39 MWCAP_VIDEO_CUSTOM_TIMING_PAR
MWCAP_IOCTL_SET_VIDEO_CUSTOM_TIMING_ARRAY	Setting the video custom timing sequence.Reference to 3.3.39 MWCAP_VIDEO_CUSTOM_TIMING_PAR
MWCAP_IOCTL_GET_VIDEO_CUSTOM_RESOLUTION_COUNT	Obtaining the video custom resolution count
MWCAP_IOCTL_GET_VIDEO_CUSTOM_RESOLUTION_ARRAY	Obtaining the video custom resolution.Reference to 3.3.40 MWCAP_VIDEO_CUSTOM_RESOLUTION_PAR
MWCAP_IOCTL_SET_VIDEO_CUSTOM_RESOLUTION_ARRAY	Setting the video custom resolution.Reference to 3.3.40 MWCAP_VIDEO_CUSTOM_RESOLUTION_PAR

3.2.12 Other Instructions;

NAME	ANNOTATION
MWCAP_IOCTL_SET_LED_MODE	Setting LED mode.Reference to 3.4.15 MWCAP_LED_MODE
MWCAP_IOCTL_SET_POST_RECONFIG	Sending the reconfig setting command.Latency execute
MWCAP_IOCTL_GET_CORE_TEMPERATURE	Obtaining the core temperature
MWCAP_IOCTL_GET_EDID_DATA	Obtaining EDID data.Reference to 3.3.41 MWCAP_EDID_DATA
MWCAP_IOCTL_SET_EDID_DATA	Setting EDID data.Reference to 3.3.41 MWCAP_EDID_DATA

3.2.13 V4L2 Control Instructions

NAME	ANNOTATION
MWCAP_IOCTL_GET_VIDEO_CONNECTION_FORMAT	Obtaining the video connection format.Reference to 3.3.42 MWCAP_VIDEO_CONNECTION_FORMAT
MWCAP_IOCTL_GET_VIDEO_PROCESS_SETTINGS	Obtaining the video process configuration parameter.Reference to 3.3.43 MWCAP_VIDEO_PROCESS_SETTINGS
MWCAP_IOCTL_SET_VIDEO_PROCESS_SETTINGS	Setting the video process configuration parameter.Reference to 3.3.43 MWCAP_VIDEO_PROCESS_SETTINGS
MWCAP_IOCTL_GET_VIDEO_OSD_SETTINGS	Obtaining the video OSD configuration parameter.Reference to 3.3.44 MWCAP_VIDEO_OSD_SETTINGS

MWCAP_IOCTL_SET_VIDEO_OSD_SETTINGS	Setting the video OSD configuration parameter.Reference to 3.3.44 MWCAP_VIDEO_OSD_SETTINGS
MWCAP_IOCTL_GET_VIDEO_OSD_IMAGE	Obtaining the ODS image area of the video capture.Regerence to 3.3.45 MWCAP_VIDEO_OSD_IMAGE
MWCAP_IOCTL_SET_VIDEO_OSD_IMAGE	Setting the ODS image area of the video capture.Regerence to 3.3.45 MWCAP_VIDEO_OSD_IMAGE
MWCAP_IOCTL_GET_BRIGHTNESS	Obtaining brightness
MWCAP_IOCTL_SET_BRIGHTNESS	Setting brightness
MWCAP_IOCTL_GET_CONTRAST	Obtaining the video contrast
MWCAP_IOCTL_SET_CONTRAST	Setting the video contrast
MWCAP_IOCTL_GET_HUE	Obtaining the video hue
MWCAP_IOCTL_SET_HUE	Setting the video hue
MWCAP_IOCTL_GET_SATURATION	Obtaining the video saturation
MWCAP_IOCTL_SET_SATURATION	Setting the video saturation

3.3 I/O Control Structure Definition

3.3.1 MWCAP_STREAM_INFO

- Structure prototype

```
typedef struct MWCAPSTREAM_INFO {
    int stream_id;
    pid_t pid;
    char comm[TS_COMM_LEN];
    bool streaming;
} MWCAP_STREAM_INFO;
```

- Parameter Description

NAME	ANNOTATION
stream_id	Capture stream ID
pid	Process ID which occupy the capture stream
comm	Process Name which occupy the capture stream
streaming	Whether to be v4l2 device streaming

3.3.2 MWCAP_STREAMS_INFO

- 结构原型

```
typedef struct MWCAPSTREAMS_INFO {
    int count;
    MWCAP_STREAM_INFO *infos;
} MWCAP_STREAMS_INFO;
```

- 参数说明

NAME	ANNOTATION
count	The count of the device streaming
infos	The data of device streaming,reference to 3.3.1 MWCAP_STREAM_INFO

3.3.3 MWCAP_CHANNEL_INFO

- Structure prototype

```
typedef struct MWCAPCHANNEL_INFO {
    WORD wFamilyID;
    WORD wProductID;
    CHAR chHardwareVersion;
    BYTE byFirmwareID;
    DWORD dwFirmwareVersion;
    DWORD dwDriverVersion;
    CHAR szFamilyName[MW_FAMILY_NAME_LEN];
    CHAR szProductName[MW_PRODUCT_NAME_LEN];
    CHAR szFirmwareName[MW_FIRMWARE_NAME_LEN];
    CHAR szBoardSerialNo[MW_SERIAL_NO_LEN];
    BYTE byBoardIndex;
    BYTE byChannelIndex;
} MWCAP_CHANNEL_INFO;
```

- Parameter Description

NAME	ANNOTATION
wFamilyID	Product Series of the capture device in this channel,reference to 3.4.1 MW_FAMILY_ID
wProductID	Product ID of the capture device in this channel,reference to 3.4.2 MW_PRODUCT_ID
chHardwareVersion	Hardware version of the capture device in this channel
byFirmwareID	Firmware ID of the capture device in this channel
dwFirmwareVersion	Firmware version of the capture device in this channel
dwDriverVersion	Driver version of the capture device in this channel.
szFamilyName	Product Family Name of the capture device in this channel
szProductName	Product type name of the capture device in this channel
szFirmwareName	Firmware name of the capture device in this channel
szBoardSerialNo	Hardware Serial Number of the capture device in this channel
byBoardIndex	The switch index of the device, the index of the knob switch indicator is on the capture device, it starts from 0
byChannelIndex	Channel Index the index starts from 0 if one capture device has multiple channels

3.3.4 MWCAP_PRO_CAPTURE_INFO

- Structure prototype

```
typedef struct MWCAPPRO_CAPTURE_INFO {
    BYTE byPCIBusID;
    BYTE byPCIDevID;
    BYTE byLinkType;
    BYTE byLinkWidth;
    BYTE byBoardIndex;
    WORD wMaxPayloadSize;
    WORD wMaxReadRequestSize;
    DWORD cbTotalMemorySize;
    DWORD cbFreeMemorySize;
} MWCAP_PRO_CAPTURE_INFO;
```

- Parameter Description

NAME	ANNOTATION
byPCIBusID	PCI Bus ID of computer after the capture device connected
byPCIDevID	The PCI device ID of computer after the capture device connected
byLinkType	Data transmission standard on the interface negotiated consensus of the capture device and computer
byLinkWidth	The number of PCI-e slot Lane while the capture device connected to the computer
byBoardIndex	The board index on the capture device
wMaxPayloadSize	The max payload size on the capture device
wMaxReadRequestSize	The max read request size on the capture device
cbTotalMemorySize	The total memory size on the device
cbFreeMemorySize	The free memory size on the device

3.3.5 MWCAP_VIDEO_CAPS

- Structure prototype

```
typedef struct MWCAPVIDEO_CAPS {
    DWORD dwCaps;
    WORD wMaxInputWidth;
    WORD wMaxInputHeight;
    WORD wMaxOutputWidth;
    WORD wMaxOutputHeight;
} MWCAP_VIDEO_CAPS;
```

- Parameter Description

NAME	ANNOTATION
dwCaps	Video capture capability
wMaxInputWidth	The maximum width of the input video
wMaxInputHeight	The maximum height of the input video
wMaxOutputWidth	The maximum width of the output video
wMaxOutputHeight	The maximum height of the output video

3.3.6 MWCAP_AUDIO_CAPS

- Structure prototype

```
typedef struct MWCAPAUDIO_CAPS {
    DWORD dwCaps;
} MWCAP_AUDIO_CAPS;
```

- Parameter Description

NAME	ANNOTATION
dwCaps	Audio capture capability

3.3.7 MWCAP_FIRMWARE_STORAGE

- Structure prototype

```
typedef struct MWCAPFIRMWARE_STORAGE {
```

```

DWORD cbStorage;
DWORD cbEraseBlock;
DWORD cbProgramBlock;
DWORD cbHeaderOffset;
} MWCAP_FIRMWARE_STORAGE;

```

- Parameter Description

NAME	ANNOTATION
cbStorage	Length of the firmware storage area
cbEraseBlock	Erase the length of the firmware storage area
cbProgramBlock	Length of the program block storage area
cbHeaderOffset	The storage offset of firmware head area

3.3.8 MWCAP_FIRMWARE_ERASE

- Structure prototype

```

typedef struct MWCAPFIRMWARE_ERASE {
    DWORD cbOffset;
    DWORD cbErase;
} MWCAP_FIRMWARE_ERASE;

```

- Parameter Description

NAME	ANNOTATION
cbOffset	The offset address of the storage area need to be erased
cbErase	The length of the erase area

3.3.9 MWCAP_FIRMWARE_DATA

- Structure prototype

```

typedef struct MWCAPFIRMWARE_DATA {
    unsigned int offset;
    unsigned int size;
    unsigned char *data;
} MWCAP_FIRMWARE_DATA;

```

- Parameter Description

NAME	ANNOTATION
cbOffset	Start position of the input data
size	The length of the input data
data	Input data

3.3.10 MWCAP_TIMER_EXPIRE_TIME

- Structure prototype

```

typedef struct MWCAPTIMER_EXPIRE_TIME {
    MWCAP_PTR pvTimer;
    LONGLONG llExpireTime;
} MWCAP_TIMER_EXPIRE_TIME;

```

- Parameter Description

NAME	ANNOTATION

pvTimer	Clock scheduling object
llExpireTime	Expire time of the clock scheduling object

3.3.11 MWCAP_TIMER_WAIT

- Structure prototype

```
typedef struct MWCAPTIMER_WAIT {
    MWCAP_PTR pvTimer;
    LONGLONG llTimeout;
} MWCAP_TIMER_WAIT;
```

- Parameter Description

NAME	ANNOTATION
pvTimer	Clock scheduling object
llTimeout	Waiting time of the clock scheduling object

3.3.12 MWCAP_NOTIFY_REGISTRATION_S

- Structure prototype

```
typedef struct MWCAPNOTIFY_REGISTRATION_S {
    MWCAP_PTR pvNotify;
    ULONGLONG ullEnableBits;
} MWCAP_NOTIFY_REGISTRATION_S;
```

- Parameter Description

NAME	ANNOTATION
pvNotify	Asynchronous event notify the object
ullEnableBits	Incident response type value

3.3.13 MWCAP_NOTIFY_STATUS

- Structure prototype

```
typedef struct MWCAPNOTIFY_STATUS {
    MWCAP_PTR pvNotify;
    ULONGLONG ullStatusBits;
} MWCAP_NOTIFY_STATUS;
```

- Parameter Description

NAME	ANNOTATION
pvNotify	Asynchronous event notify the object
ullEnableBits	Incident response type value

3.3.14 MWCAP_NOTIFY_WAIT

- Structure prototype

```
typedef struct MWCAPNOTIFY_WAIT {
    MWCAP_PTR pvNotify;
    MWCAP_BOOL bGetStatus;
    LONGLONG llTimeout;
    ULONGLONG ullStatusBits;
} MWCAP_NOTIFY_WAIT;
```

- Parameter Description

NAME	ANNOTATION
pvNotify	Asynchronous event notify the object
bGetStatus	Whether to get time response type
lTimeout	Waiting for an asynchronous event timeout
ullEnableBits	Incident response type value

3.3.15 MWCAP_NOTIFY_ENABLE

- Structure prototype

```
typedef struct MWCAP_NOTIFY_ENABLE {
    MWCAP_PTR pvNotify;
    ULONGLONG ullEnableBits;
} MWCAP_NOTIFY_ENABLE;
```

- Parameter Description

NAME	ANNOTATION
pvNotify	Asynchronous event notify the object
ullEnableBits	Incident response type value

3.3.16 MWCAP_VIDEO_BUFFER_INFO

- Structure prototype

```
typedef struct MWCAP_VIDEO_BUFFER_INFO {
    DWORD cMaxFrames;
    BYTE iNewestBuffering;
    BYTE iBufferingFieldIndex;
    BYTE iNewestBuffered;
    BYTE iBufferedFieldIndex;
    BYTE iNewestBufferedFullFrame;
    DWORD cBufferedFullFrames;
} MWCAP_VIDEO_BUFFER_INFO;
```

- Parameter Description

NAME	ANNOTATION
cMaxFrames	The max frame of the onboard buffer
iNewestBuffering	The current buffering chip sequence, per frame video data contains multiple chips
iBufferingFieldIndex	The buffering field index
iNewestBuffered	The newest buffered chip sequence
iBufferedFieldIndex	The buffered field index
iNewestBufferedFullFrame	The newest buffered frame sequence
cBufferedFullFrames	The count of the buffered frame

3.3.17 MWCAP_VIDEO_FRAME_INFO

- Structure prototype

```
typedef struct MWCAP_VIDEO_FRAME_INFO {
    MWCAP_VIDEO_FRAME_STATE state;
    BOOLEAN bInterlaced;
```

```

BOOLEAN bSegmentedFrame;
BOOLEAN bTopFieldFirst;
BOOLEAN bTopFieldInverted;
int cx;
int cy;
int nAspectX;
int nAspectY;
LONGLONG allFieldStartTimes[2];
LONGLONG allFieldBufferedTimes[2];
MWCAP_SMPTE_TIMECODE aSMPTETimeCodes[2];
} MWCAP_VIDEO_FRAME_INFO;

```

- Parameter Description

NAME	ANNOTATION
state	Video frame state. Reference to 3.4.3 MWCAP_VIDEO_FRAME_STATE
bInterlaced	Whether to interlace
bSegmentedFrame	Whether to be the segmented frame
bTopFieldFirst	Whether to be the top field first
bTopFieldInverted	Whether the top field inverted
cx	Video frame width
cy	Video frame height
nAspectX	The width of the aspect ratio
nAspectY	The height of the aspect ratio
allFieldStartTimes	The top field capture time and the bottom field capture time
allFieldBufferedTimes	The top field buffered time and the bottom field buffered time
aSMPTETimeCodes	The top field timing codes and the bottom field timing codes

3.3.18 MWCAP_VIDEO_CAPTURE_FRAME

- Structure prototype

```

typedef struct MWCAP_VIDEO_CAPTURE_FRAME {
    DWORD dwFOURCC;
    WORD cx;
    WORD cy;
    int nAspectX;
    int nAspectY;
    MWCAP_VIDEO_COLOR_FORMAT colorFormat;
    MWCAP_VIDEO_QUANTIZATION_RANGE quantRange;
    MWCAP_VIDEO_SATURATION_RANGE satRange;
    SHORT sContrast;
    SHORT sBrightness;
    SHORT sSaturation;
    SHORT sHue;
    RECT rectSource;
    RECT rectTarget;
    MWCAP_VIDEO_DEINTERLACE_MODE deinterlaceMode;
    MWCAP_VIDEO_ASPECT_RATIO_CONVERT_MODE aspectRatioConvertMode;
    char iSrcFrame;
    MWCAP_PTR pOSDImage;
    RECT aOSDRects[MWCAP_VIDEO_MAX_NUM_OSD_RECTS];
}

```

```

int cOSDRects;
BOOLEAN bPhysicalAddress;
union {
    MWCAP_PTR pvFrame;
    LARGE_INTEGER liPhysicalAddress;
};
DWORD cbFrame;
DWORD cbStride;
BOOLEAN bBottomUp;
WORD cyPartialNotify;
DWORD dwProcessSwitchs;
MWCAP_PTR pvContext;
} MWCAP_VIDEO_CAPTURE_FRAME;

```

- Parameter Description

NAME	ANNOTATION
dwFOURCC	Color format.Reference MWFOURCC.h
cx	Capture width
cy	Capture height
nAspectX	The width of the aspect ratio
nAspectY	The height of the aspect ratio
colorFormat	Color format standard.Reference to 3.4.3 MWCAP_VIDEO_FRAME_STATE
quantRange	Quant range.Reference to 3.4.5 MWCAP_VIDEO_QUANTIZATION_RANGE
satRange	The saturation range.Reference to 3.4.11 MWCAP_VIDEO_SATURATION_RANGE
sContrast	Contrast
sBrightness	Brightness
sSaturation	Saturation
sHue	Hue
rectSource	The capture source area
rectTarget	The Capture target area
deinterlaceMode	deinterlace mode.Reference 3.4.6 MWCAP_VIDEO_DEINTERLACE_MODE
aspectRatioConvertMode	Aspect ratio convert mode , Reference to 3.4.7 MWCAP_VIDEO_ASPECT_RATIO_CONVERT_MODE
iSrcFrame	The capture frame index
pOSDImage	Composing the handle of OSD image by MWCAP_IOCTL_VIDEO_CREATE_IMAGE interface
aOSDRects	The target area for composing OSD image
cOSDRects	The count of the target area for composing OSD image
bPhysicalAddress	Whether to store the capture data to the designated physical address
pvFrame	The memory address which is used to store the capture frame data
liPhysicalAddress	The physical address which is used to store the capture frame data
cbFrame	The length for this frame

cbStride	The stride used in this frame
bBottomUp	Whether to bottom up
cyPartialNotify	The number of the partial notify
dwProcessSwitchs	Video process switchs,Reference to MWCAP_VIDEO_PROCESS_xx
pvContext	The context of the video capture frame

3.3.19 MWCAP_VIDEO_CAPTURE_STATUS

- Structure prototype

```
typedef struct MWCAPVIDEO_CAPTURE_STATUS {
    MWCAP_PTR pvContext;
    BOOLEAN bPhysicalAddress;
    union {
        MWCAP_PTR pvFrame;
        LARGE_INTEGER liPhysicalAddress;
    };
    BOOLEAN bFrameCompleted;
    WORD cyCompleted;
    WORD cyCompletedPrev;
} MWCAP_VIDEO_CAPTURE_STATUS;
```

- Parameter Description

NAME	ANNOTATION
pvContext	The context of the video capture
bPhysicalAddress	Whether to use the physical address to store the capture data
pvFrame	The memory address which is used to store the capture frame data
liPhysicalAddress	The physical address which is used to store the capture frame data
bFrameCompleted	If a frame capture is complete
cyCompleted	The count of the captured frame
cyCompletedPrev	The count of the previously captured frame

3.3.20 MWCAP_AUDIO_CAPTURE_FRAME

- Structure prototype

```
typedef struct MWCAPAUDIO_CAPTURE_FRAME {
    DWORD cFrameCount;
    DWORD iFrame;
    DWORD dwSyncCode;
    DWORD dwReserved;
    LONGLONG liTimestamp;
    DWORD adwSamples[MWCAP_AUDIO_SAMPLES_PER_FRAME * MWCAP_AUDIO_MAX_NUM_CHANNELS];
} MWCAP_AUDIO_CAPTURE_FRAME;
```

- Parameter Description

NAME	ANNOTATION
cFrameCount	The count of the buffer frame
iFrame	Current frame index

dwSyncCode	Synchronous code for audio frame data
dwReserved	Reserved field
llTimestamp	Time stamp of this audio frame
adwSamples	Audio sampling data, each sample has 32 bit width, high bit effective, the channel sequence is: left 0 , left 1 , left 2 , left 3 , right 0 , right 1 , right 2 , right 3

3.3.21 MWCAP_INPUT_SOURCE_ARRAY

- Structure prototype

```
typedef struct MWCAPINPUT_SOURCE_ARRAY {
    unsigned int count;
    unsigned int *data;
} MWCAP_INPUT_SOURCE_ARRAY;
```

- Parameter Description

NAME	ANNOTATION
count	The count of video input source type
data	The index of video input source type

3.3.22 MWCAP_HDMI_VIDEO_TIMING

- Structure prototype

```
typedef struct MWCAPHDMI_VIDEO_TIMING {
    BOOLEAN bInterlaced;
    DWORD dwFrameDuration;
    WORD wHSyncWidth;
    WORD wHFrontPorch;
    WORD wHBackPorch;
    WORD wHActive;
    WORD wHTotalWidth;
    WORD wField0VSyncWidth;
    WORD wField0VFrontPorch;
    WORD wField0VBackPorch;
    WORD wField0VActive;
    WORD wField0VTotalHeight;
    WORD wField1VSyncWidth;
    WORD wField1VFrontPorch;
    WORD wField1VBackPorch;
    WORD wField1VActive;
    WORD wField1VTotalHeight;
} MWCAP_HDMI_VIDEO_TIMING;
```

- Parameter Description

NAME	ANNOTATION
bInterlaced	Whether to interlace
dwFrameDuration	Frame duration
wHSyncWidth	Horizontal sync width
wHFrontPorch	Horizontal front porch width
wHBackPorch	Horizontal back porch width
wHActive	The effective width of the horizontal direction
wHTotalWidth	The total width of the horizontal direction

wField0VSyncWidth	Vertical sync width on 0 field
wField0VFrontPorch	Vertical front porch width on 0 field
wField0VBackPorch	Vertical back porch width on 0 field
wField0VActive	The effective width of the vertical direction on 0 field
wField0VTotalHeight	Vertical total height on 0 field
wField1VSyncWidth	Vertical sync width on 1 field
wField1VFrontPorch	Vertical front porch width on 1 field
wField1VBackPorch	Vertical back porch width on 1 field
wField1VActive	The effective width of the vertical direction on 1 field
wField1VTotalHeight	Vertical total height on 1 field

3.3.23 MWCAP_VIDEO_TIMING_SETTINGS

- Structure prototype

```
typedef struct MWCAPVIDEO_TIMING_SETTINGS {
    WORD wAspectX;
    WORD wAspectY;
    WORD x;
    WORD y;
    WORD cx;
    WORD cy;
    WORD cxTotal;
    BYTE byClampPos;
} MWCAP_VIDEO_TIMING_SETTINGS;
```

- Parameter Description

NAME	ANNOTATION
wAspectX	The width of the video aspect ratio
wAspectY	The height of the video aspect ratio
x	The start position of horizontal direction
y	The start position of vertical direction
cx	Width
cy	Height
cxTotal	The total width of horizontal direction
byClampPos	Clamp position

3.3.24 MWCAP_SDI_SPECIFIC_STATUS

- Structure prototype

```
typedef struct MWCAPSDI_SPECIFIC_STATUS {
    SDI_TYPE sdiType;
    SDI_SCANNING_FORMAT sdiScanningFormat;
    SDI_BIT_DEPTH sdiBitDepth;
    SDI_SAMPLING_STRUCT sdiSamplingStruct;
    BOOLEAN bST352DataValid;
    DWORD dwST352Data;
} MWCAP_SDI_SPECIFIC_STATUS;
```

- Parameter Description

NAME	ANNOTATION

sdiType	SDI signal type
sdiScanningFormat	SDI scan type
sdiBitDepth	SDI Bit Depth
sdiSamplingStruct	SDI Sampling Struct
bST352DataValid	Whether the ST352 data is valid
dwST352Data	ST352 data

3.3.25 MWCAP_HDMI_SPECIFIC_STATUS

- Structure prototype

```
typedef struct MWCAPIHDMI_SPECIFIC_STATUS {
    BOOLEAN bHDMIMode;
    BOOLEAN bHDCP;
    BYTE byBitDepth;
    HDMI_PIXEL_ENCODING pixelEncoding;
    BYTE byVIC;
    BOOLEAN bITContent;
    BOOLEAN b3DFormat;
    BYTE by3DStructure;
    BYTE bySideBySideHalfSubSampling;
    MWCAP_HDMI_VIDEO_TIMING videoTiming;
} MWCAP_HDMI_SPECIFIC_STATUS;
```

- Parameter Description

NAME	ANNOTATION
bHDMIMode	Whether to be HDMI mode
bHDCP	Whether to be HDCP encrypted signal
byBitDepth	Bit depth
pixelEncoding	Pixel format
byVIC	Video identification code in EDID , using for specify the standard resolution and timing sequence
bITContent	IT Content bit zone
b3DFormat	Whether to be 3D format
by3DStructure	3D Structure
bySideBySideHalfSubSampling	Side by side half sub sampling
videoTiming	Video timing sequence.Reference to 3.3.22 MWCAP_HDMI_VIDEO_TIMING

3.3.26 MWCAP_COMPONENT_SPECIFIC_STATUS

- Structure prototype

```
typedef struct MWCAPCOMPONENT_SPECIFIC_STATUS {
    MWCAP_VIDEO_SYNC_INFO syncInfo;
    BOOLEAN bTriLevelSync;
    MWCAP_VIDEO_TIMING videoTiming;
    MWCAP_VIDEO_TIMING_SETTINGS videoTimingSettings;
} MWCAP_COMPONENT_SPECIFIC_STATUS;
```

- Parameter Description

NAME	ANNOTATION
syncInfo	Synchronize video information

bTriLevelSync	Whether to be thrid level sync
videoTiming	The video timing sequence information
videoTimingSettings	Setting video timing.Reference to 3.3.23 MWCAP_VIDEO_TIMING_SETTINGS/td>

3.3.27 MWCAP_CVBS_YC_SPECIFIC_STATUS

- Structure prototype

```
typedef struct MWCAPCVBS_YC_SPECIFIC_STATUS {
    MWCAP_SD_VIDEO_STANDARD standard;
    BOOLEAN b50Hz;
} MWCAP_CVBS_YC_SPECIFIC_STATUS;
```

- Parameter Description

NAME	ANNOTATION
standard	Defind video standard.Reference to 3.4.8 MWCAP_SD_VIDEO_STANDARD
b50Hz	Whether the scan frequency is 50Hz

3.3.28 MWCAP_INPUT_SPECIFIC_STATUS

- Structure prototype

```
typedef struct MWCAPINPUT_SPECIFIC_STATUS {
    BOOLEAN bValid;
    DWORD dwVideoInputType;
    union {
        MWCAP_SDI_SPECIFIC_STATUS sdiStatus;
        MWCAP_HDMI_SPECIFIC_STATUS hdmiStatus;
        MWCAP_COMPONENT_SPECIFIC_STATUS vgaComponentStatus;
        MWCAP_CVBS_YC_SPECIFIC_STATUS cvbsYcStatus;
    };
} MWCAP_INPUT_SPECIFIC_STATUS;
```

- Parameter Description

NAME	ANNOTATION
bValid	Whether the input signal is valid
dwVideoInputType	The type of the input video signal
sdiStatus	The status of SDI signal.Reference to 3.3.24 MWCAP_SDI_SPECIFIC_STATUS
hdmiStatus	HDMI signal status.Reference to 3.3.25 MWCAP_HDMI_SPECIFIC_STATUS
vgaComponentStatus	VGA component signal status.Reference to 3.3.26 MWCAP_COMPONENT_SPECIFIC_STATUS
cvbsYcStatus	CVBS-YC signal status.Reference to 3.3.27 MWCAP_CVBS_YC_SPECIFIC_STATUS

3.3.29 MWCAP_VIDEO_SIGNAL_STATUS

- Structure prototype

```
typedef struct MWCAPVIDEO_SIGNAL_STATUS {
```

```

MWCAP_VIDEO_SIGNAL_STATE state;
int x;
int y;
int cx;
int cy;
int cxTotal;
int cyTotal;
BOOLEAN bInterlaced;
DWORD dwFrameDuration;
int nAspectX;
int nAspectY;
BOOLEAN bSegmentedFrame;
MWCAP_VIDEO_FRAME_TYPE frameType;
MWCAP_VIDEO_COLOR_FORMAT colorFormat;
MWCAP_VIDEO_QUANTIZATION_RANGE quantRange;
MWCAP_VIDEO_SATURATION_RANGE satRange;
} MWCAP_VIDEO_SIGNAL_STATUS;

```

- Parameter Description

名称	注释
state	Define the access state for this video channel.Reference to 3.4.9 MWCAP_VIDEO_SIGNAL_STATE
x	The start position of horizontal direction
y	The start position of vertical direction
cx	The width of the video
cy	The height of the video
cxTotal	Total width
cyTotal	Total height
bInterlaced	Whether have the interlaced signal
dwFrameDuration	The duration of the video frame
nAspectX	The width of the video aspect ratio
nAspectY	The height of the video aspect ratio
bSegmentedFrame	Whether to be the segmented frame
frameType	Video frame type.Reference to 3.4.10 MWCAP_VIDEO_FRAME_TYPE
colorFormat	Video color format.Reference to 3.4.4 MWCAP_VIDEO_COLOR_FORMAT
quantRange	Quant range.Reference to 3.4.5 MWCAP_VIDEO_QUANTIZATION_RANGE
satRange	Sat range.Reference to 3.4.11 MWCAP_VIDEO_SATURATION_RANGE

3.3.30 MWCAP_AUDIO_SIGNAL_STATUS

- Structure prototype

```

typedef struct MWCAPAUDIO_SIGNAL_STATUS {
    WORD ChannelValid;
    BOOLEAN bLPCM;
    BYTE cBitsPerSample;
    DWORD dwSampleRate;
    BOOLEAN bChannelStatusValid;
    IEC60958_CHANNEL_STATUS channelStatus;
}

```

```
} MWCAP_AUDIO_SIGNAL_STATUS;
```

- Parameter Description

NAME	ANNOTATION
ChannelValid	The mask value of the valid channel, the lowest bit means 1 and 2 sound channel is available, the second lowest bit means means 3 and 4 sound channel is available, the third lowest bit means means 5 and 6 sound channel is available, the fourth lowest bit means means 7 and 8 sound channel is available
bLPCM	Whether to be LPCM format
cBitsPerSample	Bit depth per audio capture
dwSampleRate	Sample rate
bChannelStatusValid	Whether the channel status are valid
channelStatus	Audio channel status

3.3.31 HDMI_INFOFRAME_HEADER

- Structure prototype

```
typedef struct HDMI_INFOFRAME_HEADER {
    BYTE byPacketType;
    BYTE byVersion;
    BYTE byLength : 5;
    BYTE byReservedZero : 3;
} HDMI_INFOFRAME_HEADER;
```

- Parameter Description

NAME	ANNOTATION
byPacketType	Return HDMI InfoFrame type
byVersion	Return HDMI InfoFrame version
byLength	Return HDMI InfoFrame size (Unit:byte)
byReservedZero	Reserved byte size

3.3.32 HDMI_INFOFRAME_PACKET

- Structure prototype

```
typedef struct HDMI_INFOFRAME_PACKET {
    HDMI_INFOFRAME_HEADER header;
    BYTE byChecksum;
    union {
        BYTE abyPayload[27];
        HDMI_AVI_INFOFRAME_PAYLOAD aviInfoFramePayload;
        HDMI_AUDIO_INFOFRAME_PAYLOAD audioInfoFramePayload;
        HDMI_SPD_INFOFRAME_PAYLOAD spdInfoFramePayload;
        HDMI_VS_INFOFRAME_PAYLOAD vsInfoFramePayload;
    };
} HDMI_INFOFRAME_PACKET;
```

- Parameter Description

NAME	ANNOTATION
header	The header information of HDMI InfoFrame.Reference to 3.3.31 HDMI_INFOFRAME_HEADER

byChecksum	The amount of HDMI InfoFrame and the check code
abyPayload	HDMI InfoFrame data
aviInfoFramePayload	HDMI AVI InfoFrame data
audioInfoFramePayload	HDMI AUDIO InfoFrame data
spdInfoFramePayload	HDMI SPD InfoFrame data
vsInfoFramePayload	HDMI VS InfoFrame data

3.3.33 MWCAP_VIDEO_ASPECT_RATIO

- Structure prototype

```
typedef struct MWCAPVIDEO_ASPECT_RATIO {
    int nAspectX;
    int nAspectY;
} MWCAP_VIDEO_ASPECT_RATIO;
```

- Parameter Description

NAME	ANNOTATION
nAspectX	The width of the video aspect ratio
nAspectY	The height of the video aspect ratio

3.3.34 MWCAP_VIDEO_CREATE_IMAGE

- Structure prototype

```
typedef struct MWCAPVIDEO_CREATE_IMAGE {
    WORD cx;
    WORD cy;
    MWCAP_PTR pvlImage;
} MWCAP_VIDEO_CREATE_IMAGE;
```

- Parameter Description

NAME	ANNOTATION
cx	Creat the width of OSD image
cy	Creat the height of OSD image
pvlImage	Return the handle of the created OSD image

3.3.35 MWCAP_VIDEO_IMAGE_REF

- Structure prototype

```
typedef struct MWCAPVIDEO_IMAGE_REF {
    MWCAP_PTR pvlImage;
    long nRefCount;
} MWCAP_VIDEO_IMAGE_REF;
```

- Parameter Description

NAME	ANNOTATION
pvlImage	The handle of OSD image
nRefCount	Return the reference count of OSD image

3.3.36 MWCAP_VIDEO_UPLOAD_IMAGE

- Structure prototype

```
typedef struct MWCAP_VIDEO_UPLOAD_IMAGE {
    MWCAP_PTR pvDestImage;
    MWCAP_VIDEO_COLOR_FORMAT cfDest;
    WORD xDest;
    WORD yDest;
    WORD cxDest;
    WORD cyDest;
    MWCAP_VIDEO_QUANTIZATION_RANGE quantRangeDest;
    MWCAP_VIDEO_SATURATION_RANGE satRangeDest;
    BOOLEAN bSrcPhysicalAddress;
    union {
        MWCAP_PTR pvSrcFrame;
        LARGE_INTEGER liSrcPhysicalAddress;
    };
    DWORD cbSrcFrame;
    DWORD cbSrcStride;
    WORD cxSrc;
    WORD cySrc;
    BOOLEAN bSrcBottomUp;
    BOOLEAN bSrcPixelAlpha;
    BOOLEAN bSrcPixelXBGR;
} MWCAP_VIDEO_UPLOAD_IMAGE;
```

- Parameter Description

NAME	ANNOTATION
pvDestImage	The handle of the target destimage
cfDest	The standard color format.Reference to 3.4.4 MWCAP_VIDEO_COLOR_FORMAT
xDest	The start position of the horizontal direction on target area
yDest	The start position of the vertical direction on target area
cxDest	The width of the target area
cyDest	The height of the target area
quantRangeDest	The quant range of the target image.Reference to 3.4.5 MWCAP_VIDEO_QUANTIZATION_RANGE
satRangeDest	The sat range of the target image.Reference to 3.4.11 MWCAP_VIDEO_SATURATION_RANGE
bSrcPhysicalAddress	Whether the original video use the physical address
pvSrcFrame	The memory address which is used by the original video
liSrcPhysicalAddress	The physical address of the original video data
cbSrcFrame	The total length of the original video data
cbSrcStride	The stride of the original video data
cxSrc	The width of the original video
cySrc	The height of the original video
bSrcBottomUp	Whether the original video is bottom up
bSrcPixelAlpha	Whether the original video pixel have Alpha
bSrcPixelXBGR	Whether the original video pixel have XBGR format

3.3.37 MWCAP_VIDEO_TIMING

- Structure prototype


```
typedef struct MWCAPVIDEO_TIMING {
    DWORD dwType;
    DWORD dwPixelClock;
    BOOLEAN bInterlaced;
    BYTE bySyncType;
    BOOLEAN bHSPolarity;
    BOOLEAN bVSPolarity;
    WORD wHActive;
    WORD wHFrontPorch;
    WORD wHSyncWidth;
    WORD wHBackPorch;
    WORD wVActive;
    WORD wVFrontPorch;
    WORD wVSyncWidth;
    WORD wVBackPorch;
} MWCAP_VIDEO_TIMING;;
```

- Parameter Description

NAME	ANNOTATION
dwType	The type of video channel timing sequence. Reference to 3.4.14 MWCAP_VIDEO_TIMING_TYPE
dwPixelClock	The pixel clock of the video channel timing
bInterlaced	Whether the video timing is interlaced
bySyncType	Sync type of the video timing sequence
bHSPolarity	Whether the horizontal directional synchronization polarity of the video channel timing is true
bVSPolarity	Whether the vertical directional synchronization polarity of the video channel timing is true
wHActive	The horizontal directional active time of the video channel timing
wHFrontPorch	The horizontal directional front porch of the video channel timing sequence
wHSyncWidth	The horizontal directional sync width of the video channel timing sequence
wHBackPorch	The horizontal directional back porch of the video channel timing sequence
wVActive	The vertical directional effect time of the video channel timing sequence
wVFrontPorch	The vertical directional front porch of the video channel timing sequence
wVSyncWidth	The vertical directional sync width of the video channel timing sequence
wVBackPorch	The vertical directional back porch of the video channel timing sequence

3.3.38 MWCAP_VIDEO_CUSTOM_TIMING

- Structure prototype

```
typedef struct MWCAPVIDEO_CUSTOM_TIMING {
    MWCAP_VIDEO_SYNC_INFO syncInfo;
    MWCAP_VIDEO_TIMING_SETTINGS videoTimingSettings;
} MWCAP_VIDEO_CUSTOM_TIMING;
```

- Parameter Description

NAME	ANNOTATION
syncInfo	Video information sync
videoTimingSettings	Video timing setting.Reference to 3.3.23 MWCAP_VIDEO_TIMING_SETTINGS

3.3.39 MWCAP_VIDEO_CUSTOM_TIMING_PAR

- Structure prototype

```
typedef struct MWCAPVIDEO_CUSTOM_TIMING_PAR {
    long count;
    MWCAP_VIDEO_CUSTOM_TIMING timings[];
} MWCAP_VIDEO_CUSTOM_TIMING_PAR;
```

- Parameter Description

NAME	ANNOTATION
count	Custom video sequence count
timings	Custom video timing interface

3.3.40 MWCAP_VIDEO_CUSTOM_RESOLUTION_PAR

- Structure prototype

```
typedef struct MWCAPVIDEO_CUSTOM_RESOLUTION_PAR {
    long count;
    MWCAP_SIZE resolutions[];
} MWCAP_VIDEO_CUSTOM_RESOLUTION_PAR;
```

- Parameter Description

NAME	ANNOTATION
count	Custom video sequence count
resolutions	Custom video resolutions

3.3.41 MWCAP_EDID_DATA

- Structure prototype

```
typedef struct MWCAPEDID_DATA {
    int size;
    unsigned char *data;
} MWCAP_EDID_DATA;
```

- Parameter Description

NAME	ANNOTATION
size	EDID data size
data	EDID data

3.3.42 MWCAP_VIDEO_CONNECTION_FORMAT

- Structure prototype

```
typedef struct MWCAPVIDEO_CONNECTION_FORMAT {
    BOOLEAN bConnected;
```

```

LONG cx;
LONG cy;
DWORD dwFrameDuration;
DWORD dwFOURCC;
int nAspectX;
int nAspectY;
MWCAP_VIDEO_COLOR_FORMAT colorFormat;
MWCAP_VIDEO_QUANTIZATION_RANGE quantRange;
MWCAP_VIDEO_SATURATION_RANGE satRange;
} MWCAP_VIDEO_CONNECTION_FORMAT;

```

- Parameter Description

NAME	ANNOTATION
bConnected	Whether to be connected
cx	Video width
cy	Video height
dwFrameDuration	Video frame duration
dwFOURCC	Color space
nAspectX	The width fo the aspect ratio
nAspectY	The height fo the aspect ratio
colorFormat	Video color format.Reference to 3.4.4 MWCAP_VIDEO_C OLOR_FORMAT
quantRange	Quant range.Reference to 3.4.5 MWCAP_VIDEO_QUAN TIZATION_RANGE
satRange	Sat range.Reference to 3.4.11 MWCAP_VIDEO_SATUR ATION_RANGE

3.3.43 MWCAP_VIDEO_PROCESS_SETTINGS

- Structure prototype

```

typedef struct MWCAPVIDEO_PROCESS_SETTINGS {
    DWORD dwProcessSwitchs;
    RECT rectSource;
    int nAspectX;
    int nAspectY;
    BOOLEAN bLowLatency;
    MWCAP_VIDEO_COLOR_FORMAT colorFormat;
    MWCAP_VIDEO_QUANTIZATION_RANGE quantRange;
    MWCAP_VIDEO_SATURATION_RANGE satRange;
    MWCAP_VIDEO_DEINTERLACE_MODE deinterlaceMode;
    MWCAP_VIDEO_ASPECT_RATIO_CONVERT_MODE aspectRatioConvertMode;
} MWCAP_VIDEO_PROCESS_SETTINGS;

```

- Parameter Description

NAME	ANNOTATION
dwProcessSwitchs	Video processing mask value
rectSource	To deal with the source area
nAspectX	The width fo the aspect ratio
nAspectY	The height fo the aspect ratio
bLowLatency	Whether to use the low latency
colorFormat	The standard video color format.Reference to 3.4.4 MWC AP_VIDEO_COLOR_FORMAT

quantRange	Quant range.Reference to 3.4.5 MWCAP_VIDEO_QUANTIZATION_RANGE
satRange	Sat range.Reference to 3.4.11 MWCAP_VIDEO_SATURATION_RANGE
deinterlaceMode	Deinterlace mode.Reference to 3.4.6 MWCAP_VIDEO_DEINTERLACE_MODE
aspectRatioConvertMode	Aspect ratio convert mode.Reference to 3.4.7 MWCAP_VIDEO_ASPECT_RATIO_CONVERT_MODE

3.3.44 MWCAP_VIDEO_OSD_SETTINGS

- Structure prototype

```
typedef struct MWCAPVIDEO_OSD_SETTINGS {
    BOOLEAN bEnable;
    char szPNGFilePath[MAXPATH];
} MWCAP_VIDEO_OSD_SETTINGS;
```

- Parameter Description

NAME	ANNOTATION
bEnable	Whether to use OSD image
szPNGFilePath	Upgrade PNG image address

3.3.45 MWCAP_VIDEO_OSD_IMAGE

- Structure prototype

```
typedef struct MWCAPVIDEO_OSD_IMAGE {
    MWCAP_PTR pvOSDImage;
    RECT aOSDRects[MWCAP_VIDEO_MAX_NUM_OSD_RECTS];
    int cOSDRects;
} MWCAP_VIDEO_OSD_IMAGE;
```

- Parameter Description

NAME	ANNOTATION
pvOSDImage	The handle of OSD image
aOSDRects	The target area which is used for composing image
cOSDRects	The number of the target area which is used for composing image

3.4 I/O Control Enum definition

3.4.1 MW_FAMILY_ID

- Enum prototype

```
typedef enum MWFAMILY_ID {
    MW_FAMILY_ID_PRO_CAPTURE = 0x00,
    MW_FAMILY_ID_VALUE_CAPTURE = 0x01,
    MW_FAMILY_ID_USB_CAPTURE = 0x02
} MW_FAMILY_ID;
```

- Constant Definition

NAME	ANNOTATION
------	------------

MW_FAMILY_ID_PRO_CAPTURE	This capture device is using PCI-e interfact to connect the computer
MW_FAMILY_ID_VALUE_CAPTURE	This capture device is using PCI-e interfact to connect the computer, it compare with the Pro Capture family, it has some simplification on the software and hardware,such as not supporting multi stream output
MW_FAMILY_ID_USB_CAPTURE	This capture device is using USB interfact to connect the computer

3.4.2 MWCAP_PRODUCT_ID

- Enum prototype

```
typedef enum MWCAPPRODUCT_ID {
    MWCAP_PRODUCT_ID_PRO_CAPTURE_AIO = 0x00000102,
    MWCAP_PRODUCT_ID_PRO_CAPTURE_DVI = 0x00000103,
    MWCAP_PRODUCT_ID_PRO_CAPTURE_HDMI = 0x00000104,
    MWCAP_PRODUCT_ID_PRO_CAPTURE_SDI = 0x00000105,
    MWCAP_PRODUCT_ID_PRO_CAPTURE_DUAL_SDI = 0x00000106,
    MWCAP_PRODUCT_ID_PRO_CAPTURE_DUAL_DVI = 0x00000107,
    MWCAP_PRODUCT_ID_PRO_CAPTURE_DUAL_HDMI = 0x00000108,
    MWCAP_PRODUCT_ID_PRO_CAPTURE_QUAD_SDI = 0x00000109,
    MWCAP_PRODUCT_ID_PRO_CAPTURE_QUAD_HDMI = 0x00000110,
    MWCAP_PRODUCT_ID_PRO_CAPTURE_MINI_HDMI = 0x00000111,
    MWCAP_PRODUCT_ID_PRO_CAPTURE_HDMI_4K = 0x00000112
} MWCAP_PRODUCT_ID;
```

- Constant Definition

NAME	ANNOTATION
MWCAP_PRODUCT_ID_PRO_CAPTURE_AIO	The capture card is provided with PCI-e input and output interface
MWCAP_PRODUCT_ID_PRO_CAPTURE_DVI	The capture card is provided with PCI-e host interface and DVI input interface
MWCAP_PRODUCT_ID_PRO_CAPTURE_HDMI	The capture card is provided with PCI-e host interface and HDMI input interface
MWCAP_PRODUCT_ID_PRO_CAPTURE_SDI	The capture card is provided with PCI-e host interface and SDI input interface
MWCAP_PRODUCT_ID_PRO_CAPTURE_DUAL_SDI	The capture card is provided with PCI-e host interface and dual SDI input interface
MWCAP_PRODUCT_ID_PRO_CAPTURE_DUAL_DVI	The capture card is provided with PCI-e host interface and dual DVI input interface
MWCAP_PRODUCT_ID_PRO_CAPTURE_DUAL_HDMI	The capture card is provided with PCI-e host interface and dual HDMI input interface
MWCAP_PRODUCT_ID_PRO_CAPTURE_QUAD_SDI	The capture card is provided with PCI-e host interface and quad SDI input interface
MWCAP_PRODUCT_ID_PRO_CAPTURE_QUAD_HDMI	The capture card is provided with PCI-e host interface and quad HDMI input interface
MWCAP_PRODUCT_ID_PRO_CAPTURE_MINI_HDMI	The capture card is provided with PCI-e host interface and MINI HDMI input interface
MWCAP_PRODUCT_ID_PRO_CAPTURE_HDMI_4K	The capture card is provided with PCI-e host interface and 4K HDMI input interface

3.4.3 MWCAP_VIDEO_FRAME_STATE

- Enum prototype

```
typedef enum MWCAPVIDEO_FRAME_STATE {  
    MWCAP_VIDEO_FRAME_STATE_INITIAL,  
    MWCAP_VIDEO_FRAME_STATE_F0_BUFFERING,  
    MWCAP_VIDEO_FRAME_STATE_F1_BUFFERING,  
    MWCAP_VIDEO_FRAME_STATE_BUFFERED  
} MWCAP_VIDEO_FRAME_STATE;
```

- Constant Definition

NAME	ANNOTATION
MWCAP_VIDEO_FRAME_STATE_INITIAL	The initial state
MWCAP_VIDEO_FRAME_STATE_F0_BUFFERING	Buffering No.0 field
MWCAP_VIDEO_FRAME_STATE_F1_BUFFERING	Buffering No.1 field
MWCAP_VIDEO_FRAME_STATE_BUFFERED	Video frame has already buffered

3.4.4 MWCAP_VIDEO_COLOR_FORMAT

- Enum prototype

```
typedef enum MWCAPVIDEO_COLOR_FORMAT {  
    MWCAP_VIDEO_COLOR_FORMAT_UNKNOWN = 0x00,  
    MWCAP_VIDEO_COLOR_FORMAT_RGB = 0x01,  
    MWCAP_VIDEO_COLOR_FORMAT_YUV601 = 0x02,  
    MWCAP_VIDEO_COLOR_FORMAT_YUV709 = 0x03,  
    MWCAP_VIDEO_COLOR_FORMAT_YUV2020 = 0x04,  
    MWCAP_VIDEO_COLOR_FORMAT_YUV2020C = 0x05  
} MWCAP_VIDEO_COLOR_FORMAT;
```

- Constant Definition

NAME	ANNOTATION
MWCAP_VIDEO_COLOR_FORMAT_UNKNOWN	Unknown color format
MWCAP_VIDEO_COLOR_FORMAT_RGB	RGB color format
MWCAP_VIDEO_COLOR_FORMAT_YUV601	YUV601 color format
MWCAP_VIDEO_COLOR_FORMAT_YUV709	YUV709 color format
MWCAP_VIDEO_COLOR_FORMAT_YUV2020	YUV2020 color format
MWCAP_VIDEO_COLOR_FORMAT_YUV2020C	YUV2020C color format

3.4.5 MWCAP_VIDEO_QUANTIZATION_RANGE

- Enum prototype

```
typedef enum MWCAPVIDEO_QUANTIZATION_RANGE {
    MWCAP_VIDEO_QUANTIZATION_UNKNOWN = 0x00,
    MWCAP_VIDEO_QUANTIZATION_FULL = 0x01,
    MWCAP_VIDEO_QUANTIZATION_LIMITED = 0x02
} MWCAP_VIDEO_QUANTIZATION_RANGE;
```

- Constant Definition

NAME	ANNOTATION
MWCAP_VIDEO_QUANTIZATION_UNKNOWN	Unknown video quantization range
MWCAP_VIDEO_QUANTIZATION_FULL	Full video quantization range,8 bit Black and white color range should from 0 to 255
MWCAP_VIDEO_QUANTIZATION_LIMITED	limited video quantization range,8 bit Black and white color range should from 16 to 235

3.4.6 MWCAP_VIDEO_DEINTERLACE_MODE

- Enum prototype

```
typedef enum MWCAPVIDEO_DEINTERLACE_MODE {
    MWCAP_VIDEO_DEINTERLACE_WEAVE = 0x00,
    MWCAP_VIDEO_DEINTERLACE_BLEND = 0x01,
    MWCAP_VIDEO_DEINTERLACE_TOP_FIELD = 0x02,
    MWCAP_VIDEO_DEINTERLACE_BOTTOM_FIELD = 0x03
} MWCAP_VIDEO_DEINTERLACE_MODE;
```

- Constant Definition

NAME	ANNOTATION
MWCAP_VIDEO_DEINTERLACE_WEAVE	Don't execut deinterlace
MWCAP_VIDEO_DEINTERLACE_BLEND	Using blend mode to deinterlace video, this is default mode
MWCAP_VIDEO_DEINTERLACE_TOP_FIELD	Using top field mode to deinterlace video
MWCAP_VIDEO_DEINTERLACE_BOTTOM_FIELD	Using bottom field mode to deinterlace video

3.4.7 MWCAP_VIDEO_ASPECT_RATIO_CONVERT_MODE

- Enum prototype

```
typedef enum MWCAPVIDEO_ASPECT_RATIO_CONVERT_MODE {
    MWCAP_VIDEO_ASPECT_RATIO_IGNORE = 0x00,
    MWCAP_VIDEO_ASPECT_RATIO_CROPPING = 0x01,
    MWCAP_VIDEO_ASPECT_RATIO_PADDING = 0x02
} MWCAP_VIDEO_ASPECT_RATIO_CONVERT_MODE;
```

- Constant Definition

NAME	ANNOTATION
MWCAP_VIDEO_ASPECT_RATIO_IGNORE	Ignore video aspect ratio

MWCAP_VIDEO_ASPECT_RATIO_CROPPING	Transform video aspect ratio by cropping
MWCAP_VIDEO_ASPECT_RATIO_PADDING	Transform video aspect ratio by padding

3.4.8 MWCAP_SD_VIDEO_STANDARD

- Enum prototype

```
typedef enum MWCAPSD_VIDEO_STANDARD {
    MWCAP_SD_VIDEO_NONE,
    MWCAP_SD_VIDEO_NTSC_M,
    MWCAP_SD_VIDEO_NTSC_433,
    MWCAP_SD_VIDEO_PAL_M,
    MWCAP_SD_VIDEO_PAL_60,
    MWCAP_SD_VIDEO_PAL_COMBN,
    MWCAP_SD_VIDEO_PAL_BGHID,
    MWCAP_SD_VIDEO_SECAM,
    MWCAP_SD_VIDEO_SECAM_60
} MWCAP_SD_VIDEO_STANDARD;
```

- Constant Definition

NAME	ANNOTATION
MWCAP_SD_VIDEO_NONE	None
MWCAP_SD_VIDEO_NTSC_M	Using NTSC_M standard
MWCAP_SD_VIDEO_NTSC_433	Using NTSC_433 standard
MWCAP_SD_VIDEO_PAL_M	Using PAL_M standard
MWCAP_SD_VIDEO_PAL_60	Using PAL_60 standard
MWCAP_SD_VIDEO_PAL_COMBN	Using PAL_COMBN standard
MWCAP_SD_VIDEO_PAL_BGHID	Using PAL_BGHID standard
MWCAP_SD_VIDEO_SECAM	Using SECAM standard
MWCAP_SD_VIDEO_SECAM_60	Using SECAM_60 standard

3.4.9 MWCAP_VIDEO_SIGNAL_STATE

- Enum prototype

```
typedef enum MWCAPVIDEO_SIGNAL_STATE {
    MWCAP_VIDEO_SIGNAL_NONE,
    MWCAP_VIDEO_SIGNAL_UNSUPPORTED,
    MWCAP_VIDEO_SIGNAL_LOCKING,
    MWCAP_VIDEO_SIGNAL_LOCKED
} MWCAP_VIDEO_SIGNAL_STATE;
```

- Constant Definition

NAME	ANNOTATION
MWCAP_VIDEO_SIGNAL_NONE	No video signal
MWCAP_VIDEO_SIGNAL_UNSUPPORTED	Invalid video signal, capture device scan the input signal but can not identify the signal
MWCAP_VIDEO_SIGNAL_LOCKING	Locking the video signal, video signal is available but not on lock

MWCAP_VIDEO_SIGNAL_LOCKED	Locked the video signal, capture device can capture the current input signal
---------------------------	--

3.4.10 MWCAP_VIDEO_FRAME_TYPE

- Enum prototype

```
typedef enum MWCAPVIDEO_FRAME_TYPE {
    MWCAP_VIDEO_FRAME_2D = 0x00,
    MWCAP_VIDEO_FRAME_3D_TOP_AND_BOTTOM_FULL = 0x01,
    MWCAP_VIDEO_FRAME_3D_TOP_AND_BOTTOM_HALF = 0x02,
    MWCAP_VIDEO_FRAME_3D_SIDE_BY_SIDE_FULL = 0x03,
    MWCAP_VIDEO_FRAME_3D_SIDE_BY_SIDE_HALF = 0x04
} MWCAP_VIDEO_FRAME_TYPE;
```

- Constant Definition

NAME	ANNOTATION
MWCAP_VIDEO_FRAME_2D	2D video frame
MWCAP_VIDEO_FRAME_3D_TOP_AND_BOTTOM_FULL	The panoramic view of 3D video frames on top field and bottom field
MWCAP_VIDEO_FRAME_3D_TOP_AND_BOTTOM_HALF	The half view of 3D video frames on top field and bottom field
MWCAP_VIDEO_FRAME_3D_SIDE_BY_SIDE_FULL	3D video frames of the side by side panoramic view
MWCAP_VIDEO_FRAME_3D_SIDE_BY_SIDE_HALF	3D video frames of the side by side half view

3.4.11 MWCAP_VIDEO_SATURATION_RANGE

- Enum prototype

```
typedef enum MWCAPVIDEO_SATURATION_RANGE {
    MWCAP_VIDEO_SATURATION_UNKNOWN = 0x00,
    MWCAP_VIDEO_SATURATION_FULL = 0x01,
    MWCAP_VIDEO_SATURATION_LIMITED = 0x02,
    MWCAP_VIDEO_SATURATION_EXTENDED_GAMUT = 0x03
} MWCAP_VIDEO_SATURATION_RANGE;
```

- Constant Definition

NAME	ANNOTATION
MWCAP_VIDEO_SATURATION_UNKNOWN	The range of the unknown video saturation
MWCAP_VIDEO_SATURATION_FULL	The range of the full video saturation, 8 bit Black and white color range should from 0 to 255
MWCAP_VIDEO_SATURATION_LIMITED	The range of the limited video saturation, 8 bit Black and white color range should from 16 to 235
MWCAP_VIDEO_SATURATION_EXTENDED_GAMUT	The range of the extended video saturation gamut, 8 bit Black and white color range should from 1 to 254

3.4.12 MWCAP_HDMI_INFOFRAME_ID

- Enum prototype

```
typedef enum MWCAPHDMI_INFOFRAME_ID {
    MWCAP_HDMI_INFOFRAME_ID_AVI,
    MWCAP_HDMI_INFOFRAME_ID_AUDIO,
    MWCAP_HDMI_INFOFRAME_ID_SPD,
    MWCAP_HDMI_INFOFRAME_ID_MS,
    MWCAP_HDMI_INFOFRAME_ID_VS,
    MWCAP_HDMI_INFOFRAME_ID_ACP,
    MWCAP_HDMI_INFOFRAME_ID_ISRC1,
    MWCAP_HDMI_INFOFRAME_ID_ISRC2,
    MWCAP_HDMI_INFOFRAME_ID_GAMUT,
    MWCAP_HDMI_INFOFRAME_COUNT
} MWCAP_HDMI_INFOFRAME_ID;
```

- Constant Definition

NAME	ANNOTATION
MWCAP_HDMI_INFOFRAME_ID_AVI	AVI InfoFrame
MWCAP_HDMI_INFOFRAME_ID_AUDIO	Audio InfoFrame
MWCAP_HDMI_INFOFRAME_ID_SPD	SPD InfoFrame
MWCAP_HDMI_INFOFRAME_ID_MS	MS InfoFrame
MWCAP_HDMI_INFOFRAME_ID_VS	VS InfoFrame
MWCAP_HDMI_INFOFRAME_ID_ACP	ACP InfoFrame
MWCAP_HDMI_INFOFRAME_ID_ISRC1	ISRC1 InfoFrame
MWCAP_HDMI_INFOFRAME_ID_ISRC2	ISRC2 InfoFrame
MWCAP_HDMI_INFOFRAME_ID_GAMUT	GAMUT InfoFrame
MWCAP_HDMI_INFOFRAME_COUNT	The amount of the InfoFrame types

3.4.13 MWCAP_HDMI_INFOFRAME_ID

- Enum prototype

```
typedef enum MWCAPHDMI_INFOFRAME_MASK {
    MWCAP_HDMI_INFOFRAME_MASK_AVI=(1<< MWCAP_HDMI_INFOFRAME_ID_AVI),
    MWCAP_HDMI_INFOFRAME_MASK_AUDIO=(1<< MWCAP_HDMI_INFOFRAME_ID_AUDIO),
    MWCAP_HDMI_INFOFRAME_MASK_SPD=(1<< MWCAP_HDMI_INFOFRAME_ID_SPD),
    MWCAP_HDMI_INFOFRAME_MASK_MS=(1<< MWCAP_HDMI_INFOFRAME_ID_MS),
    MWCAP_HDMI_INFOFRAME_MASK_VS=(1<< MWCAP_HDMI_INFOFRAME_ID_VS),
    MWCAP_HDMI_INFOFRAME_MASK_ACP=(1<< MWCAP_HDMI_INFOFRAME_ID_ACP),
    MWCAP_HDMI_INFOFRAME_MASK_ISRC1=(1<< MWCAP_HDMI_INFOFRAME_ID_ISRC1),
    MWCAP_HDMI_INFOFRAME_MASK_ISRC2=(1<< MWCAP_HDMI_INFOFRAME_ID_ISRC2),
    MWCAP_HDMI_INFOFRAME_MASK_GAMUT=(1<< WCAP_HDMI_INFOFRAME_ID_GAMUT)
} MWCAP_HDMI_INFOFRAME_MASK;
```

- Constant Definition

NAME	ANNOTATION
MWCAP_HDMI_INFOFRAME_INFOF	AVI InfoFrame

RAME_MASK_AVI	
MWCAP_HDMI_INFOFRAME_INFOFRAME_MASK_AUDIO	Audio InfoFrame
MWCAP_HDMI_INFOFRAME_INFOFRAME_MASK_SPD	SPD InfoFrame
MWCAP_HDMI_INFOFRAME_INFOFRAME_MASK_MS	MS InfoFrame
MWCAP_HDMI_INFOFRAME_INFOFRAME_MASK_VS	VS InfoFrame
MWCAP_HDMI_INFOFRAME_INFOFRAME_MASK_ACP	ACP InfoFrame
MWCAP_HDMI_INFOFRAME_INFOFRAME_MASK_ISRC1	ISRC1 InfoFrame
MWCAP_HDMI_INFOFRAME_INFOFRAME_MASK_ISRC2	ISRC2 InfoFrame
MWCAP_HDMI_INFOFRAME_INFOFRAME_MASK_GAMUT	GAMUT InfoFrame

3.4.14 MWCAP_VIDEO_TIMING_TYPE

- Enum prototype

```
typedef enum MWCAPVIDEO_TIMING_TYPE {
    MWCAP_VIDEO_TIMING_NONE = 0x00000000,
    MWCAP_VIDEO_TIMING_LEGACY = 0x00000001,
    MWCAP_VIDEO_TIMING_DMT = 0x00000002,
    MWCAP_VIDEO_TIMING_CEA = 0x00000004,
    MWCAP_VIDEO_TIMING_GTF = 0x00000008,
    MWCAP_VIDEO_TIMING_CVT = 0x00000010,
    MWCAP_VIDEO_TIMING_CVT_RB = 0x00000020,
    MWCAP_VIDEO_TIMING_FAILSAFE = 0x00002000
} MWCAP_VIDEO_TIMING_TYPE;
```

- Constant Definition

NAME	ANNOTATION
MWCAP_VIDEO_TIMING_NONE	Video channel uses no timing sequence
MWCAP_VIDEO_TIMING_LEGACY	Video channel uses LEGACY timing sequence
MWCAP_VIDEO_TIMING_DMT	Video channel uses DMT timing sequence
MWCAP_VIDEO_TIMING_CEA	Video channel uses CEA timing sequence
MWCAP_VIDEO_TIMING_GTF	Video channel uses GTF timing sequence
MWCAP_VIDEO_TIMING_CVT	Video channel uses CVT timing sequence
MWCAP_VIDEO_TIMING_CVT_RB	Video channel uses CVT_RB timing sequence
MWCAP_VIDEO_TIMING_FAILSAFE	Video channel uses FAILSAFE timing sequence

3.4.15 MWCAP_LED_MODE

- Enum prototype

```
typedef enum MWCAPLED_MODE {
    MWCAP_LED_AUTO = 0x00000000,
```

```

MWCAP_LED_OFF = 0x80000000,
MWCAP_LED_ON = 0x80000001,
MWCAP_LED_BLINK = 0x80000002,
MWCAP_LED_DBL_BLINK = 0x80000003,
MWCAP_LED_BREATH = 0x80000004
} MWCAP_LED_MODE;

```

- Constant Definition

NAME	ANNOTATION
MWCAP_LED_AUTO	Auto Mode
MWCAP_LED_OFF	Keeping LED on “OFF” status
MWCAP_LED_ON	Keeping LED on “ON” status
MWCAP_LED_BLINK	Keeping LED on “BLINK” status
MWCAP_LED_DBL_BLINK	LED blink twice per time
MWCAP_LED_BREATH	Breath Mode

3.4.16 MWCAP_VIDEO_INPUT_TYPE

- Enum prototype

```

typedef enum MWCAPVIDEO_INPUT_TYPE {
    MWCAP_VIDEO_INPUT_TYPE_NONE = 0x00,
    MWCAP_VIDEO_INPUT_TYPE_HDMI = 0x01,
    MWCAP_VIDEO_INPUT_TYPE_VGA = 0x02,
    MWCAP_VIDEO_INPUT_TYPE_SDI = 0x04,
    MWCAP_VIDEO_INPUT_TYPE_COMPONENT = 0x08,
    MWCAP_VIDEO_INPUT_TYPE_CVBS = 0x10,
    MWCAP_VIDEO_INPUT_TYPE_YC = 0x20
} MWCAP_VIDEO_INPUT_TYPE;

```

- Constant Definition

NAME	ANNOTATION
MWCAP_VIDEO_INPUT_TYPE_NONE	No signal input interface
MWCAP_VIDEO_INPUT_TYPE_HDMI	HDMI signal input interface
MWCAP_VIDEO_INPUT_TYPE_VGA	VGA signal input interface
MWCAP_VIDEO_INPUT_TYPE_SDI	SDI signal input interface
MWCAP_VIDEO_INPUT_TYPE_COMPONENT	COMPONENT signal input interface
MWCAP_VIDEO_INPUT_TYPE_CVBS	CVBS signal input interface
MWCAP_VIDEO_INPUT_TYPE_YC	YC signal input interface

3.4.17 MWCAP_AUDIO_INPUT_TYPE

- Enum prototype

```

typedef enum MWCAPAUDIO_INPUT_TYPE {
    MWCAP_AUDIO_INPUT_TYPE_NONE = 0x00,
    MWCAP_AUDIO_INPUT_TYPE_HDMI = 0x01,
    MWCAP_AUDIO_INPUT_TYPE_SDI = 0x02,
    MWCAP_AUDIO_INPUT_TYPE_LINE_IN = 0x04,

```

```
MWCAP_AUDIO_INPUT_TYPE_MIC_IN = 0x08  
} MWCAP_AUDIO_INPUT_TYPE;
```

- Constant Definition

NAME	ANNOTATION
MWCAP_AUDIO_INPUT_TYPE_NONE	No signal input interface
MWCAP_AUDIO_INPUT_TYPE_HDMI	HDMI signal input interface
MWCAP_AUDIO_INPUT_TYPE_SDI	SDI signal input interface
MWCAP_AUDIO_INPUT_TYPE_LINE_IN	Line_In signal input interface
MWCAP_AUDIO_INPUT_TYPE_MIC_IN	Mic_In signal input interface