INGENIC®
T31 Encoder API Reference

Date: 2022-06

北京君正集成电路股份有限公司
Ingenic Semiconductor Co., Ltd.

INGENIC®

T31 Encoder API Reference

Copyright © Ingenic Semiconductor Co. Ltd 2022. All rights reserved.

Release history

Date	Revision	Change
2022-04	1.0	First release
2022-04	1.1	Chapter 3 Add IMP_Encoder_SetFrameRelease

Disclaimer

This documentation is provided for use with Ingenic products. No license to Ingenic property rights is granted. Ingenic assumes no liability, provides no warranty either expressed or implied relating to the usage, or intellectual property right infringement except as provided for by Ingenic Terms and Conditions of Sale.

Ingenic products are not designed for and should not be used in any medical or life sustaining or supporting equipment.

All information in this document should be treated as preliminary. Ingenic may make changes to this document without notice. Anyone relying on this documentation should contact Ingenic for the current documentation and errata.

Ingenic Semiconductor Co. Ltd

Add: Junzheng R&D Center, Phase II, Zhongguancun Software Park, Dongbeiwangxi Road, Haidian District, Beijing, China

Tel: 86-10-56345000

Fax: 86-10-56345001

Http: //www.ingenic.com

Content

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

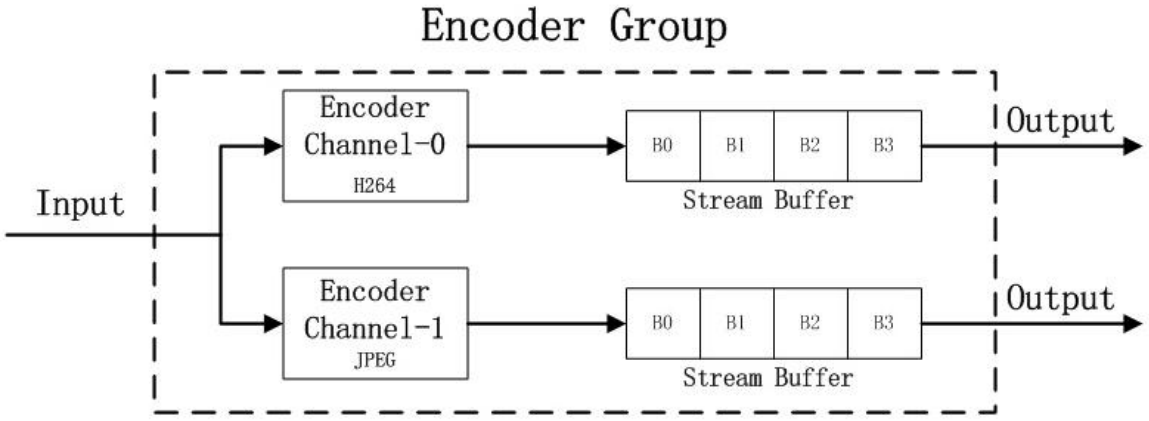
Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

1 Structure of Video Coding Module

1.1 Module Introduction

The video coding (JPEG, H264, H265) module mainly provides the Functions of creating and destroying video coding channel, starting and stopping receiving image, setting and acquiring the attributes of coding channel, acquiring and releasing code stream, etc. This module supports multi-channel real-time coding, each channel is independent, and different coding protocols and profiles can be set.

1.2 Structure block diagram



Pic 1-1 Coding module structure

As shown in the figure above, the coding module is composed of several Groups, and each Group is composed of a coding Channel. Each coding channel is attached to an output bitstream buffer, which is composed of multiple buffers. At present, 6 Groups are supported on T31, and each Group is composed of 2 Channels. Up to 6 Channels are used.

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

1.3 Key words description

QP

QP(Quantization Parameter)，QP value corresponds to the sequence number of quantization steps. The smaller the value is, the smaller the quantization step is, the higher the quantization accuracy is, the better the image quality is, and the larger the encoded size is..

GOP

GOP(Group Of Pictures)， Refers to the interval between two I frames. Video image sequence consists of one or more GOPs, which are independent.

MB

MB(Macroblock)， Coding macroblock, the basic unit of H.264 coding.

✧ MCU

MCU(Minimum code unit)， The basic unit of JPEG coding.

VPS

VPS(Video Parameter Set)， Video parameter set, h265 unique nal unit type.

SPS

SPS(Sequence Parameter Set)， Sequence parameter set contains the common information of all images in a GOP.

PPS

PPS(Picture Parameter Set)， The image parameter set contains the parameters used for image coding.

SEI

SEI(Supplemental Enhancement information)， Auxiliary enhancement information.

1.4 Rate Control

1.4.1 CBR

CBR (Constant Bit Rate) is a constant bit rate, that is, the coding rate is constant in the rate

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

statistics time. Taking H.264 coding as an example, users can set maxQp, minQp, bitrate and so on. maxQp and minQp are used to control the quality range of the image, and bitrate is used to calculate the average coding rate in the clamping time.

When the coding rate is greater than the constant rate, the image QP will gradually adjust to maxQp. When the coding rate is much less than the constant rate, the image QP will gradually adjust to minQp.

When the image QP reaches maxQp, QP is clamped to the maximum, the bitrate clamping effect is invalid, and the coding rate may exceed bitrate.

When the image QP reaches minQp, QP is clamped to the minimum value. At this time, the coding rate has reached the maximum value, and the image quality is the best.

1.4.2 VBR

VBR Variable bit rate.

1.4.3 CAPPED_Quality

Variable bit rate based on Max PSNR. Recommended.

1.4.4 FixQP

Fix Qp fix the value of Qp. In the code rate statistics time, the Qp values of all macroblocks in the coded image are the same, and the image Qp values set by the user are adopted.

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

2 Video Encoder SDK

2.1 Encoder Module API

API Name	Function
IMP_Encoder_CreateGroup	Create Encoder Group
IMP_Encoder_DestroyGroup	Destruction Encoder Grouop
IMP_Encoder_CreateChn	Create Encoder Group
IMP_Encoder_DestroyChn	Destruction Encoder Grouop
IMP_Encoder_GetChnAttr	Create Encoder Group
IMP_Encoder_RegisterChn	Destruction Encoder Grouop
IMP_Encoder_UnRegisterChn	Anti registration Encoder Channel to Group
IMP_Encoder_StartRecvPic	Open Encoder Channel Receive image
IMP_Encoder_StopRecvPic	Stop Encoder Channel Receive image
IMP_Encoder_Query	Select Encoder Channel status
IMP_Encoder_GetStream	Get Encoder Stream
IMP_Encoder_ReleaseStream	Select Encoder Channel status
IMP_Encoder_PollingStream	Get Encoder Stream
IMP_Encoder_GetFd	Get Encoder Channel Corresponding device file handle
IMP_Encoder_SetbufshareChn	Set JPEG channel to share 265/264 encoding channel memory
IMP_Encoder_SetChnResizeMode	Set whether encoding scaling requires requesting additional rmem memory
IMP_Encoder_SetMaxStreamCnt	Set the number of bitstream buffers
IMP_Encoder_GetMaxStreamCnt	Get the number of bitstream buffers
IMP_Encoder_RequestIDR	Request IDR frame
IMP_Encoder_FlushStream	Brush out the old code stream remaining in the encoder and start encoding with the IDR frame
IMP_Encoder_GetChnFrmRate	Get frame rate control properties
IMP_Encoder_SetChnFrmRate	Setting frame rate control properties dynamically
IMP_Encoder_GetChnAttrRcMode	Get the rate control mode attribute
IMP_Encoder_SetChnAttrRcMode	Set rate control mode properties
IMP_Encoder_SetFisheyeEnableStatus	Set the enabling state of the fish eye correction algorithm

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

	provided by Ingenic
IMP_Encoder_GetFisheyeEnableStatus	Get the enabling state of the fish eye correction algorithm provided by Ingenic
IMP_Encoder_GetChnEncType	Get the image-encoding protocol type
IMP_Encoder_SetPool	Bind chnnel to the memory pool, that is, the Encoder applies for MEM from the pool
IMP_Encoder_SetDefaultParam	Set the default parameters for Encoder
IMP_Encoder_SetChnBitRate	Setting the Encoder chnnel rate
IMP_Encoder_SetChnGopLength	Setting GOP length in Encoder chnnel
IMP_Encoder_GetChnGopAttr	Get GOP structure properties in Encoder chnnel
IMP_Encoder_SetChnGopAttr	Setting GOP structure properties in Encoder chnnel
IMP_Encoder_SetChnQp	Dynamic set code rate control attribute QP
IMP_Encoder_SetChnQpBounds	Set the encoding code rate control properties MinQP and MaxQP
IMP_Encoder_SetChnQpIPDelta	Dynamic set code rate control property IPDelta
IMP_Encoder_SetStreamBufSize	Set Encoder buffer size
IMP_Encoder_GetStreamBufSize	Get Encoder buffer size
IMP_Encoder_GetChnAveBitrate	Get the average code rate for the specified number of frames
IMP_Encoder_GetChnEvalInfo	Assessment parameters were obtained for each frame in the channel
IMP_AI_DisableAecRefFrame	Disable AI channel to obtain reference frame.

Tab2-1 AI API Function table

2.1.1 IMP_Encoder_CreateGroup

【Function】

Create Encoder Group.

【Grammar】

int IMP_Encoder_CreateGroup(int encGroup);

【Formal parameter】

Parameter name	Describe	Input/output
encGroup	Group number， Value range: [0, NR_MAX_ENC_GROUPS - 1]	Input

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

- One way group only supports one way resolution. New groups need to be started for different resolutions. One way Group can register up to two coding channels.
- If the specified Group already exists, a failure is returned.

【Example】

```
for (i = 0; i < FS_CHN_NUM; i++) {
 if (chn[i].enable) {
    ret = IMP_Encoder_CreateGroup(chn[i].index);
    if (ret < 0) {
       IMP_LOG_ERR(TAG, "IMP_Encoder_CreateGroup(%d) error !\n", chn[i].index);
       return -1;
    }
 }
}
```

2.1.2 IMP_Encoder_DestroyGroup

【Function】

Destruction Encoder Grouop.

【Grammar】

int IMP_Encoder_DestroyGroup(int encGroup);

【Formal parameter】

Parameter name	Describe	Input/output
encGroup	Group number，Value range: [0, NR_MAX_ENC_GROUPS - 1]	Input

【Return value】

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

return 0 success；no 0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

- When destroying a Group, you must ensure that the Group is empty, that is, no Channel is registered in the Group, or the Channel registered in the Group has been de-registered, otherwise failure will be returned.
- To destroy a nonexistent Group, a failure is returned.

【Example】

None.

2.1.3 IMP_Encoder_CreateChn

【Function】

Create Encoder Channel.

【Grammar】

int IMP_Encoder_CreateChn(int encChn,const IMPEncoderChnAttr *attr);

【Formal parameter】

Parameter name	Describe	Input/output
encChn	Encoder Channel number,Value range: [0, NR_MAX_ENC_CHN - 1]	Input
attr	Encoder Channel attribute point	Input

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

【NB】

- The Encoder Channel attribute consists of two parts: encoder attribute and rate control attribute.
- Firstly, encoder attributes need to select encoding protocols, and then assign values to the corresponding attributes of various protocols.

【Example】

None.

2.1.4 IMP_Encoder_DestroyChn

【Function】

Destruction Encoder Channel.

【Grammar】

int IMP_Encoder_DestroyChn(int encChn);

【Formal parameter】

Parameter name	Describe	Input/output
encChn	Encoder Channel Number,Value Range: [0, NR_MAX_ENC_CHN - 1]	Input

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

- If the nonexistent Channel is destroyed, a failure is returned.
- Before destroying, you must ensure that the Channel has been de-registered from the Group, otherwise, it will return failure.

【Example】

Please refer to IMP_Encoder_UnRegisterChn Example.

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

2.1.5 IMP_Encoder_GetChnAttr

【Function】

Get Encoder Channel attribute.

【Grammar】

Int IMP_Encoder_GetChnAttr(int encChn,
IMPEncoderChnAttr *const attr);

【Formal parameter】

Parameter name	Describe	Input/output
encChn	Encoder Channel Number,Value Range: [0, NR_MAX_ENC_CHN - 1]	Input
attr	Encoder Channel attribute point	Output

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

None.

【Example】

None.

2.1.6 IMP_Encoder_RegisterChn

【Function】

Register Encoder Channel to Group.

【Grammar】

int IMP_Encoder_RegisterChn(int encGroup, int encChn);

【Formal parameter】

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

Parameter name	Describe	Input/output
encGroup	Encoder Group Number,Value Range: [0, NR_MAX_ENC_GROUPS - 1]	Input
encChn	Encoder Channel Number,Value Range: [0, NR_MAX_ENC_CHN - 1]	Input

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

- If the Channel does not exist, failure will be returned.
- Register the Channel to a nonexistent Group, otherwise, failure will be returned.
- The same Encoder Channel can only be registered to one Group. If the Channel has been registered to a Group, failure will be returned.
- If a Group has been registered, the Group cannot be registered by other Channels unless the previous registration relationship is cancelled.

【Example】

Please refer to Example.

2.1.7 IMP_Encoder_UnRegisterChn

【Function】

Anti registration Encoder Channel to Group.

【Grammar】

int IMP_Encoder_UnRegisterChn(int encChn);

【Formal parameter】

Parameter name	Describe	Input/output

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

encChn	Encoder Channel Number,Value Range: [0, NR_MAX_ENC_CHN - 1]	Input

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

- If you log off a Channel that has not been created, a failure is returned.
- If the unregistered Channel is unregistered, a failure is returned.
- If the Encoder Channel does not stop receiving image Encoder , a failure is returned.
- After the Channel is logged off, the Encoder Channel will be reset and the Encoder stream buffers in the Encoder Channel will be cleared. If the user is still using the Encoder stream buffer that has not been released in time, the correctness of the buffer data will not be guaranteed, and the user can use imp_ Encoder_ Query interface to query the status of the Encoder stream buffer of the Encoder Channel, confirm that the code stream in the code Encoder buffer is retrieved, and then de-register the Channel.

【Example】

```
1. int ret = 0, i = 0, chnNum = 0;
2. IMPEncoderChnStat chn_stat;
3.
4. for (i = 0; i <  FS_CHN_NUM; i++) {
5.    if (chn[i].enable) {
6.       chnNum = chn[i].index;
7.       memset(&chn_stat, 0, sizeof(IMPEncoderChnStat));
8.       ret = IMP_Encoder_Query(chnNum, &chn_stat);
9.       if (ret < 0) {
10.              IMP_LOG_ERR(TAG, "IMP_Encoder_Query(%d) error: %d\n", chnNum, ret);
11.              return -1;
12.          }
13.                 if (chn_stat.registered) {
14.              ret = IMP_Encoder_UnRegisterChn(chnNum);
15.              if (ret < 0) {
16.                 IMP_LOG_ERR(TAG, "IMP_Encoder_UnRegisterChn(%d) error: %d\n", chnNum, ret);
17.                 return -1;
18.              }
```


Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

```
            ret = IMP_Encoder_DestroyChn(chnNum);
            if (ret < 0) {
                IMP_LOG_ERR(TAG, "IMP_Encoder_DestroyChn(%d) error: %d\n", chnNum, ret);
                return -1;
            }
        }
    }
}
```

2.1.8 IMP_Encoder_StartRecvPic

【Function】

Open Encoder Channel Receive image.

【Grammar】

int IMP_Encoder_StartRecvPic(int encChn);

【Formal parameter】

Parameter name	Describe	Input/output
encChn	Encoder Channel Number,Value Range: [0, NR_MAX_ENC_CHN - 1]	Input

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

- The Encoder can only be started after the Encoder Channel is turned on to receive the image.
- If the Channel is not created, a failure is returned.
- If Channel is not registered with Group, failure is returned.

【Example】

```
int val, i, chnNum, ret;
char stream_path[64];
IMPEncoderEncType encType;
```


Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

```
int stream_fd = -1, totalSaveCnt = 0;

val = (int)args;
chnNum = val & 0xffff;
encType = (val >> 16) & 0xffff;

    ret = IMP_Encoder_StartRecvPic(chnNum);
    if (ret < 0) {
      IMP_LOG_ERR(TAG, "IMP_Encoder_StartRecvPic(%d) failed\n", chnNum);
      return ((void *)-1);
    }

    for (i = 0; i < totalSaveCnt; i++) {
      ret = IMP_Encoder_PollingStream(chnNum, 1000);
      if (ret < 0) {
        IMP_LOG_ERR(TAG, "IMP_Encoder_PollingStream(%d) timeout\n", chnNum);
        continue;
      }

      IMPEncoderStream stream;
      /* Get H264 or H265 Stream */
      ret = IMP_Encoder_GetStream(chnNum, &stream, 1);
      if (ret < 0) {
        IMP_LOG_ERR(TAG, "IMP_Encoder_GetStream(%d) failed\n", chnNum);
        return ((void *)-1);
      }

      if (encType == IMP_ENC_TYPE_JPEG) {
        ret = save_stream_by_name(stream_path, i, &stream);
        if (ret < 0) {
          return ((void *)ret);
        }
      }
      else {
        ret = save_stream(stream_fd, &stream);
        if (ret < 0) {
          close(stream_fd);
          return ((void *)ret);
        }
      }
      IMP_Encoder_ReleaseStream(chnNum, &stream);
    }
```


Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

```
46.
47.     close(stream_fd);
48.
49.     ret = IMP_Encoder_StopRecvPic(chnNum);
50.     if (ret < 0) {
51.       IMP_LOG_ERR(TAG, "IMP_Encoder_StopRecvPic(%d) failed\n", chnNum);
52.       return ((void *)-1);
53.     }
```

2.1.9 IMP_Encoder_StopRecvPic

【Function】

Stop Encoder Channel Receive image.

【Grammar】

int IMP_Encoder_StopRecvPic(int encChn);

【Formal parameter】

Parameter name	Describe	Input/output
encChn	Encoder Channel Number,Value Range: [0, NR_MAX_ENC_CHN - 1]	Input

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

- This interface does not judge whether to stop receiving at present, that is, it allows repeated stop receiving without returning an error. Calling this interface only stops receiving the original data encoding, and the Encoder stream buffer will not be eliminated.
- If the Channel is not created, a failure is returned.
- If Channel is not registered with Group, failure is returned.

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

【Example】

Please refer to IMP_Encoder_StartRecvPic Example.

2.1.10 IMP_Encoder_Query

【Function】

Select Encoder Channel status

【Grammar】

int IMP_Encoder_Query(int encChn, IMPEncoderChnStat *stat);

【Formal parameter】

Parameter name	Describe	Input/output
encChn	Encoder Channel Number,Value Range: [0, NR_MAX_ENC_CHN - 1]	Input
stat	Encoder Channel status	Input

【Return value】

return 0 success； no 0 fail.

【Dependence】

Head file： imp_encoder.h

Lib file： libimp.a / libimp.so

【NB】

None.

【Example】

Please refer to IMP_Encoder_UnRegisterChn Example.

2.1.11 IMP_Encoder_GetStream

【Function】

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

Get Encoder Stream.

【Grammar】

int IMP_Encoder_GetStream(int encChn, IMPEncoderStream *stream, bool blockFlag);

【Formal parameter】

Parameter name	Describe	Input/output
encChn	Encoder Channel Number,Value Range: [0, NR_MAX_ENC_CHN - 1]	Input
stream	Bit stream structure pointer	Input
blockFlag	Whether to use blocking method to obtain, 0 is not blocked; 1 block	Input

【Return value】

return 0 success； no 0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

- If the user does not get the data of a frame stream for a long time, the stream buffer will be full. If the code stream buffer of a Encoder Channel is full, the received image will be discarded until the user obtains the Encoder stream and has enough Encoder stream buffer for coding. It is suggested that the user's interface call to acquire the Encoder stream and the interface call to release the Encoder stream appear in pairs, and release the Encoder stream as soon as possible, so as to prevent the Encoder stream buffer from being full due to the untimely acquisition and release of the Encoder stream in user mode and stop coding.
- For H264 and h265 type Encoder streams, the Encoder stream of one frame can be successfully obtained in one call, which may contain multiple packets.
- For JPEG type Encoder stream, the Encoder stream of one frame is successfully obtained in one call. This frame code stream only contains one package, and this frame contains the complete information of JPEG image file.

【Example】

Please refer to IMP_Encoder_StartRecvPic Example.

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

2.1.12 IMP_Encoder_ReleaseStream

【Function】

Release Encoder Stream cache.

【Grammar】

int IMP_Encoder_ReleaseStream(int encChn,IMPEncoderStream *stream);

【Formal parameter】

【Formal parameter】	Describe	Input/output
encChn	Encoder Channel Number,Value Range: [0, NR_MAX_ENC_CHN - 1]	Input
stream	Bit stream structure pointer	Output

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

- This interface should be compatible with IMP_ Encoder_ GetStream is used in pairs. Users must release the obtained stream buffer in time after acquiring the stream, otherwise the stream buffer may be full and the encoder encoding will be affected. And the user must release the obtained stream cache in the order of first acquire and first release; After the de registration of the Encoder Channel, all the unreleased stream packets are invalid and can no longer be used or released.
- If pstStream is null, a failure is returned.
- If the Channel is not created, a failure is returned.
- Releasing an invalid stream will return a failure.

【Example】

Please refer to IMP_Encoder_StartRecvPic Example.

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

2.1.13 IMP_Encoder_PollingStream

【Function】

Polling Stream cache.

【Grammar】

int IMP_Encoder_PollingStream(int encChn,uint32_t timeoutMsec);

【Formal parameter】

Parameter name	Describe	Input/Output
encChn	Encoder Channel Number,Value Range: [0, NR_MAX_ENC_CHN - 1]	Input
timeoutMsec	Timeout in milliseconds	Output

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

- Before obtaining the Encoder stream, you can use this API to poll. When the Encoder stream cache is not empty or the time-out is exceeded, the Function returns.

【Example】

Please refer to IMP_Encoder_StartRecvPic Example.

2.1.14 IMP_Encoder_GetFd

【Function】

Gets the device file handle corresponding to the encoded Channel.

【Grammar】

int IMP_Encoder_GetFd(int encChn);

【Formal parameter】

Parameter	Describe	Input/Outp

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

name		ut
encChn	Encoder Channel Number,Value Range: [0, NR_MAX_ENC_CHN - 1]	Input

【Return value】

return >0 success；< 0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

- In situations where IMP_Encoder_PollingStream is inappropriate, such as when Polling multiple encoded channels in the same place, you can use this file handle to call select, poll, and the like to block waiting for the encoding event to complete.
- Calling this API requires an existing channel.

【Example】

None.

2.1.15 IMP_Encoder_SetbufshareChn

【Function】

Set JPEG channel to share 265/264 encoding channel memory.

【Grammar】

int IMP_Encoder_SetbufshareChn(int encChn, int shareChn);

【Formal parameter】

Parameter name	Describe	Input/Output
encChn	Encoder Channel Number,Value Range: [0, NR_MAX_ENC_CHN - 1]	Input
shareChn	Shared 264/265 Encoder Channel number,Value Range: [0, NR_MAX_ENC_CHN - 1]	Input

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

- The shared memory Encoder Channel was created before using this API.
- This API needs to be called before the Channel is created.

【Example】

None.

2.1.16 IMP_Encoder_SetChnResizeMode

【Function】

Set whether additional rmem memory is required for encoding scaling.

【Grammar】

int IMP_Encoder_SetChnResizeMode(int encChn, int en);

【Formal parameter】

Parameter name	Describe	Input/Output
encChn	Encoder Channel Number,Value Range: [0, NR_MAX_ENC_CHN - 1]	Input
en	Enable: no need to apply for rmem, 1: no need to apply for rmem, 0: need to apply for rmem	Input

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_encoder.h

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

Lib file：libimp.a / libimp.so

【NB】

- This API is only called when the resolution of encoding scaling is less than the original resolution, and there is no need to call when the resolution of encoding scaling is greater than the original resolution.

【Example】

None.

2.1.17 IMP_Encoder_SetMaxStreamCnt

【Function】

Set the number of bitstream buffers.

【Grammar】

int IMP_Encoder_SetMaxStreamCnt(int encChn, int nrMaxStream);

【Formal parameter】

Parameter name	Describe	Input/Output
encChn	Encoder Channel Number,Value Range: [0, NR_MAX_ENC_CHN - 1]	Input
nrMaxStream	Stream Buffer Number,Value Range: [1, NR_MAX_ENC_CHN_STREAM]	Input

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

- Because the number of Buffer buffers is fixed at the time the Channel is created, the secondary API needs to be called before the Channel is created.
- If this API is not called to set the number of buffers in the bitstream cache before the Channel is created, the default number of buffers in the SDK will be used.

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

【Example】

None.

2.1.18 IMP_Encoder_GetMaxStreamCnt

【Function】

Get the number of bitstream buffers.

【Grammar】

int IMP_Encoder_GetMaxStreamCnt(int encChn, int *nrMaxStream);

【Formal parameter】

Parameter name	Describe	Input/Output
encChn	Encoder Channel Number,Value Range: [0, NR_MAX_ENC_CHN - 1]	Input
nrMaxStream	Stream Buffer Number,Value Range: [1, NR_MAX_ENC_CHN_STREAM]	Input

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

None.

【Example】

None.

2.1.19 IMP_Encoder_RequestIDR

【Function】

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

Request IDR frame.

【Grammar】

int IMP_Encoder_RequestIDR(int encChn);

【Formal parameter】

Parameter name	Describe	Input/Output
encChn	Encoder Channel Number,Value Range: [0, NR_MAX_ENC_CHN - 1]	Input

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

- After calling this API, IDR frame coding will be applied in the latest coding frame.
- This API is only applicable to H264 and h265 encoding channels.

【Example】

None.

2.1.20 IMP_Encoder_FlushStream

【Function】

The residual old code stream in the encoder is wiped out, and the coding starts with the IDR frame.

【Grammar】

int IMP_Encoder_FlushStream(int encChn);

【Formal parameter】

Parameter name	Describe	Input/Output
encChn	Encoder Channel Number,Value Range:	Input

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

	[0, NR_MAX_ENC_CHN - 1]	

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

- After calling this API, IDR frame coding will be applied in the latest coding frame.

【Example】

None.

2.1.21 IMP_Encoder_GetChnFrmRate

【Function】

Get frame rate control properties.

【Grammar】

int IMP_Encoder_GetChnFrmRate(int encChn, IMPEncoderFrmRate *pstFps);

【Formal parameter】

Parameter name	Describe	Input/Output
encChn	Encoder Channel Number,Value Range: [0, NR_MAX_ENC_CHN - 1]	Input
pstFps	Frame rate control attribute parameters	Output

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

【NB】

- Calling this API will get the frame rate control property of the channel. Calling this API requires the channel already exists.
- This API is only applicable to H264 and h265 encoding channels.

【Example】

None.

2.1.22 IMP_Encoder_SetChnFrmRate

【Function】

Setting frame rate control properties dynamically.

【Grammar】

int IMP_Encoder_SetChnFrmRate(int encChn, const IMPEncoderFrmRate *pstFps);

【Formal parameter】

Parameter name	Describe	Input/Output
encChn	Encoder Channel Number,Value Range: [0, NR_MAX_ENC_CHN - 1]	Input
pstFps	Frame rate control attribute parameters	Output

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

- Calling this API will reset the encoder frame rate attribute. The frame rate attribute takes effect in the next GOP with a maximum delay of 1 second. Calling this API requires that the channel already exists.
- If imp is called_ FrameSource_ Setchnfps() Function dynamically changes the system frame rate, so you need to call this Function to modify the encoder frame rate and complete the correct parameter configuration.
- This API is only applicable to H264 and h265 encoding channels.

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

【Example】

None.

2.1.23 IMP_Encoder_GetChnAttrRcMode

【Function】

Get the rate control mode attribute.

【Grammar】

int IMP_Encoder_GetChnAttrRcMode(
int encChn, IMPEncoderAttrRcMode *pstRcModeCfg);

【Formal parameter】

Parameter name	Describe	Input/Output
encChn	Encoder Channel Number,Value Range: [0, NR_MAX_ENC_CHN - 1]	Input
pstRcModeCfg	Rate control mode attribute parameters	Output

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

- Calling this API will get the bit rate control mode property of the channel. Calling this API requires that the channel already exists.
- This API is only applicable to H264 and h265 encoding channels.

【Example】

None.

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

2.1.24 IMP_Encoder_SetChnAttrRcMode

【Function】

Set rate control mode properties.

【Grammar】

int IMP_Encoder_SetChnAttrRcMode(
int encChn, IMPEncoderAttrRcMode *pstRcModeCfg);

【Formal parameter】

Parameter name	Describe	Input/Output
encChn	Encoder Channel Number,Value Range: [0, NR_MAX_ENC_CHN - 1]	Input
pstRcModeCfg	Rate control mode attribute parameters	Output

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

- Calling this API will set the bit rate control mode property of the channel, and the next IDR will take effect. Calling this API requires that the channel already exists.
- At present, the rate control mode supports ENC_RC_MODE_FIXQP, ENC_RC_MODE_CBR, ENC_RC_MODE_VBR 与 ENC_RC_MODE_SMART.
- This API is only applicable to H264 and h265 encoding channels.

【Example】

None.

2.1.25 IMP_Encoder_SetFisheyeEnableStatus

【Function】

Set the enabling state of the fish eye correction algorithm provided by Ingenic.

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

【Grammar】

int IMP_Encoder_SetFisheyeEnableStatus(int encChn, int enable);

【Formal parameter】

Parameter name	Describe	Input/Output
encChn	Encoder Channel Number,Value Range: [0, NR_MAX_ENC_CHN - 1]	Input
enable	0: not enabled (default), 1: enabled	Output

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

- Because the enabling state of fisheye correction algorithm is fixed when the channel is created, sub API calls before channel creation.
- If this API is not called to set the enable state of the fish eye correction algorithm provided by Ingenic before the channel is created, it is not enabled by default, that is, the fish eye correction algorithm provided by Ingenic cannot be used.
- This API is only applicable to H264 and h265 encoding channels.

【Example】

None.

2.1.26 IMP_Encoder_GetFisheyeEnableStatus

【Function】

Get the enabling state of the fish eye correction algorithm provided by Ingenic.

【Grammar】

int IMP_Encoder_GetFisheyeEnableStatus(int encChn,int *enable);

【Formal parameter】

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

Parameter name	Describe	Input/Output
encChn	Encoder Channel Number,Value Range: [0, NR_MAX_ENC_CHN - 1]	Input
enable	Returns the enabled state of the fish eye correction algorithm provided by Ingenic, 0: not enabled, 1: enabled	Output

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

- This API is only applicable to H264 and h265 encoding channels.

【Example】

None.

2.1.27 IMP_Encoder_GetChnEncType

【Function】

Get image coding protocol type.

【Grammar】

int IMP_Encoder_GetChnEncType(int encChn, IMPEncoderEncType *encType);

【Formal parameter】

Parameter name	Describe	Input/Output
encChn	Encoder Channel Number,Value Range: [0, NR_MAX_ENC_CHN - 1]	Input
encType	Return to get the image coding protocol type	Output

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

- If the channel is not created, the failure is returned.

【Example】

None.

2.1.28 IMP_Encoder_SetPool

【Function】

Bind chnnel to the memory pool, that is, the Encoder applies for MEM from the pool.

【Grammar】

int IMP_Encoder_SetPool(int chnNum, int poolID);

【Formal parameter】

Parameter name	Describe	Input/Output
chnNum	Channel number	Input
poolID	Memory pool number	Input

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

- To solve the problem of rmem fragmentation, the channel encoder is bound to the corresponding MemPool. If the encoder applies for MEM, it will apply in the MemPool. If it is not called, the encoder will apply in rmem. At this time, there is the possibility of fragmentation for rmem.

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

【Example】

None.

2.1.29 IMP_Encoder_SetDefaultParam

【Function】

Set the default parameters for Encoder.

【Grammar】

int IMP_Encoder_SetDefaultParam (IMPEncoderChnAttr *chnAttr,
IMPEncoderProfile profile, IMPEncoderRcMode rcMode,
uint16_t uWidth, uint16_t uHeight, uint32_t frmRateNum,
uint32_t frmRateDen, uint32_t uGopLength,
int uMaxSameSenceCnt,int iInitialQP,
uint32_t uTargetBitRate);

【Formal parameter】

Parameter name	Describe	Input/Output
chnAttr	Encoding chnnel attribute structure	Output
profile	Coding protocol type	Input
rcMode	Code rate control mode	Input
uWidth	Encoding image width	Input
uHeight	Coded image height	Input
frmRateNum	Coding frame rate numerator	Input
frmRateDen	Coded frame rate denominator	Input
uGopLength	GOPlength	Input
uMaxSameSenceCnt	Maximum number of scene switches	Input
iInitialQP	Initialization QP	Input
uTargetBitRate	Standard code stream entry	Input

【Return value】

return 0 success； no 0 fail.

【Dependence】

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

None.

【Example】

```
unsigned int uTargetBitRate = BITRATE_720P_Kbs * ratio;

ret = IMP_Encoder_SetDefaultParam(&channel_attr, chn[i].payloadType, S_RC_METHOD,
        imp_chn_attr_tmp->picWidth, imp_chn_attr_tmp->picHeight,
        imp_chn_attr_tmp->outFrmRateNum, imp_chn_attr_tmp->outFrmRateDen,
        imp_chn_attr_tmp->outFrmRateNum * 2 / imp_chn_attr_tmp->outFrmRateDen, 2,
        (S_RC_METHOD == IMP_ENC_RC_MODE_FIXQP) ? 35 : -1,
        uTargetBitRate);
if (ret < 0) {
            IMP_LOG_ERR(TAG, "IMP_Encoder_SetDefaultParam(%d) error !\n", chnNum);
            return -1;
        }
```

2.1.30 IMP_Encoder_SetChnBitRate

【Function】

Setting the Encoder chnnel rate.

【Grammar】

int IMP_Encoder_SetChnBitRate(int encChn,int iTargetBitRate, int iMaxBitRate);

【Formal parameter】

Parameter name	Describe	Input/Output
encChn	Encoder Channel Number,Value Range: [0, NR_MAX_ENC_CHN - 1]	Input
iTargetBitRate	Coding target bit rate	Input
iMaxBitRate	Maximum coding rate	Input

【Return value】

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

return 0 success；no 0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

None.

【Example】

None.

2.1.31 IMP_Encoder_SetChnGopLength

【Function】

Setting GOP length in Encoder chnnel.

【Grammar】

int IMP_Encoder_SetChnGopLength(int encChn, int iGopLength);

【Formal parameter】

Parameter name	Describe	Input/Output
encChn	Encoder Channel Number,Value Range: [0, NR_MAX_ENC_CHN - 1]	Input
iGopLength	Encoder GOP length	Input

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

None.

【Example】

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

None.

2.1.32 IMP_Encoder_GetChnGopAttr

【Function】

Get GOP structure properties in Encoder chnnel.

【Grammar】

int IMP_Encoder_GetChnGopAttr(int encChn,IMPEncoderGopAttr *pGopAttr);

【Formal parameter】

Parameter name	Describe	Input/Output
encChn	Encoder Channel Number,Value Range: [0, NR_MAX_ENC_CHN - 1]	Input
pGopAttr	Encoder GOP structure properties	Output

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

None.

【Example】

None.

2.1.33 IMP_Encoder_SetChnGopAttr

【Function】

Setting GOP structure properties in Encoder chnnel.

【Grammar】

int IMP_Encoder_SetChnGopAttr(int encChn,const IMPEncoderGopAttr *pGopAttr);

【Formal parameter】

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

Parameter name	Describe	Input/Output
encChn	Encoder Channel Number,Value Range: [0, NR_MAX_ENC_CHN - 1]	Input
pGopAttr	EnCoder GOP structure	Input

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

None.

【Example】

None.

2.1.34 IMP_Encoder_SetChnQp

【Function】

Dynamically set the bit rate control property QP.

【Grammar】

int IMP_Encoder_SetChnQp(int encChn, int iQP);

【Formal parameter】

Parameter name	Describe	Input/Output
encChn	Encoder Channel Number,Value Range: [0, NR_MAX_ENC_CHN - 1]	Input
iQP	QP value, value range: [iMinQP, iMaxQP]	Input

【Return value】

return 0 success；no 0 fail.

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

- Calling this API resets the encoding QP for the next frame and the set QP takes effect for the next frame. The attention API only works with H264 and H265 encoded channels.

【Example】

None.

2.1.35 IMP_Encoder_SetChnQpBounds

【Function】

Set the code rate control attributes MinQP and MaxQP.

【Grammar】

int IMP_Encoder_SetChnQpBounds(int encChn, int iMinQP, int iMaxQP);

【Formal parameter】

Parameter name	Describe	Input/Output
encChn	Encoder Channel Number,Value Range: [0, NR_MAX_ENC_CHN - 1]	Input
iMinQP	Bit rate control minimum QP	Input
iMaxQP	Bit rate control maximum QP	Input

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

- The default iMinQP and iMaxQP are 15 and 48 respectively. You can call the API to set iMinQP and iMaxQP.
- IMinQP and iMaxQP should not exceed [0, 51].

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

【Example】

None.

2.1.36 IMP_Encoder_SetChnQpIPDelta

【Function】

Dynamically sets the bit rate control property IPDelta.

【Grammar】

int IMP_Encoder_SetChnQpIPDelta(int encChn, int uIPDelta);

【Formal parameter】

Parameter name	Describe	Input/Output
encChn	Encoder Channel Number,Value Range: [0, NR_MAX_ENC_CHN - 1]	Input
IPDelta	The difference between I frame and the first subsequent P frame QP	Input

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

- Calling this API requires that the channel already exists, and this API only applies to H264 and H265 encoded channels.

【Example】

None.

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

2.1.37 IMP_Encoder_SetStreamBufSize

【Function】

Set Encoder buffer size.

【Grammar】

int IMP_Encoder_SetStreamBufSize(int encChn, uint32_t nrStreamSize);

【Formal parameter】

Parameter name	Describe	Input/Output
encChn	Encoder Channel Number,Value Range: [0, NR_MAX_ENC_CHN - 1]	Input
nrStreamSize	Encoder buffer size	Input

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

- This API must be used before an encoding channel is created.
- nrstreamsize size depends on memory size allocation, please do not exceed the maximum limit.

【Example】

None.

2.1.38 IMP_Encoder_GetStreamBufSize

【Function】

Get Encoder buffer size.

【Grammar】

int IMP_Encoder_GetStreamBufSize(int encChn, uint32_t *nrStreamSize);

【Formal parameter】

Parameter name	Describe	Input/Output

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

encChn	Encoder Channel Number,Value Range: [0, NR_MAX_ENC_CHN - 1]	Input
nrStreamSize	Encoder buffer size	Output

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

- This API must be used after the Channel is created.

【Example】

None.

2.1.39 IMP_Encoder_GetChnAveBitrare

【Function】

Gets the average bit rate for the specified number of frames.

【Grammar】

int IMP_Encoder_GetChnAveBitrate(int encChn,
IMPEncoderStream *stream, int frames, double *br);

【Formal parameter】

Parameter name	Describe	Input/Output
encChn	Encoder Channel Number,Value Range: [0, NR_MAX_ENC_CHN - 1]	Input
Stream	stream	Input
frame	Frames to be counted (integer multiple of GOP length)	Input
br	The average bit rate	Output

【Return value】

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

return >=0 && < 32 success；<0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

- This API must be used after IMP_Encoder_GetStream.

【Example】

None.

2.1.40 IMP_Encoder_GetChnEvalInfo

【Function】

Gets the evaluation parameters for each frame in the channel.

【Grammar】

int IMP_Encoder_GetChnEvalInfo(int encChn, void *info);

【Formal parameter】

Parameter name	Describe	Input/Output
encChn	Encoder Channel Number,Value Range: [0, NR_MAX_ENC_CHN - 1]	Input
info	Assessment information	Output

【Return value】

return >=0 && < 32 success；<0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

- This API must be used after IMP_Encoder_GetStream.

【Example】

None.

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

2.1.41 IMP_Encoder_SetFrameRelease

【Function】

Set numerator and denominator of early release frame.

【Grammar】

int IMP_Encoder_GetChnEvalInfo(int encChn, void *info);

【Formal parameter】

Parameter name	Describe	Input/Output
encChn	Encoder Channel Number	Input
num	Release molecule	Input
den	Release denominator	Input

【Return value】

return 0 success；None 0 fail.

【Dependence】

Head file：imp_encoder.h

Lib file：libimp.a / libimp.so

【NB】

- This API must be used after the channel is created.

【Example】

None.

2.2 VENC Date Type

Venc related data types are defined as follows：

Name	Definition
IMPEncoderPack	Encoder frame stream packet structure
IMPEncoderStreamInfo	264/265 encoding frame information structure
IMPEncoderJpegInfo	JPeg encoded frame information structure
IMPEncoderStream	Define the coding framdestream type structure
IMPEncoderEncAttr	Define the encoder attribute structure
IMPEncoderAttrCbr	Attribute structure in CBR mode of encoder
IMPEncoderAttrVbr	Attribute structure in VBR mode of encoder
IMPEncoderAttrCappedVbr	Attribute structure in encoder CappedVBR mode

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

IMPEncoderAttrFixQP	Attribute structure in FixQP mode of encoder
IMPEncoderAttrRcMode	Encoder encoding mode attribute structure

2.2.1 IMPEncoderPack

【Explain】

Encoder frame stream packet structure.

【Definition】

```
typedef struct {
        uint32_t        offset;
        uint32_t        length;
        int64_t timestamp;
        bool        frameEnd;
        IMPEncoderNalType      nalType;
/**< H.264 and H.265 Encoder Channel frame NALTpye */
        IMPEncoderSliceType sliceType;
} IMPEncoderPack;
```

【Member】

Member name	Describe
offset	Address offset of bitstream packet
length	Stream packet length
timestamp	Time stamp in us
frameEnd	End of frame flag
nalType	Encoder Channel Frame NAL Type
sliceType	Encoder chip Type

【NB】

None.

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

2.2.2 IMPEncoderStreamInfo

【Explain】

264/265 encoding frame information structure.

【Definition】

```
typedef struct {
    int32_t         iNumBytes;
    uint32_t        uNumIntra;
    uint32_t        uNumSkip;
    uint32_t        uNumCU8x8;
    uint32_t        uNumCU16x16;
    uint32_t        uNumCU32x32;
    uint32_t        uNumCU64x64;
    int16_t         iSliceQP;
    int16_t         iMinQP;
    int16_t         iMaxQP;
} IMPEncoderStreamInfo;
```

【Member】

Member name	Describe
iNumBytes	The number of bytes in a stream
uNumIntra	Number of encoding units in frame mode 8*8
uNumSkip	The transformation skips the 8*8 number of coding units
uNumCU8x8	8 x 8 Number of coding units
uNumCU16x16	16 x 16 Number of coding units
uNumCU32x32	32 x 32 Number of coding units
uNumCU64x64	Number of 64 x 64 encoding units
iSliceQP	Coding QP
iMinQP	Minimum QP

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

iMaxQP	The biggest QP

【NB】

None.

2.2.3 IMPEncoderJpegInfo

【Explain】

JPeg encoded frame information structure.

【Definition】

```
typedef struct {
    int32_t         iNumBytes;
    uint32_t        uNumIntra;
    uint32_t        uNumSkip;
    uint32_t        uNumCU8x8;
    uint32_t        uNumCU16x16;
    uint32_t        uNumCU32x32;
    uint32_t        uNumCU64x64;
    int16_t         iSliceQP;
    int16_t         iMinQP;
    int16_t         iMaxQP;
} IMPEncoderStreamInfo;
```

【Member】

Member name	Describe
iNumBytes	The number of bytes in a stream
iQPfactor	JPEG quantization parameter

【NB】

None.

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

2.2.4 IMPEncoderStream

【Explain】

Define the coding framdestream type structure.

【Definition】

```
typedef struct {
    uint32_t            phyAddr;
    uint32_t            virAddr;
    uint32_t            streamSize;
 /**< virAddr corresponds      to the size of the allocated address space*/
    IMPEncoderPack    *pack;
    uint32_t            packCount;
    uint32_t            seq;
} IMPEncoderStream;
```

【Member】

Member name	Describe
phyAddr	Physical address of frame stream
virAddr	Frame stream packet virtual address
streamSize	Viraddr corresponds to the size of the allocated address space
pack	Frame stream packet structure
packCount	The number of all packets in a frame stream
seq	Encoder frame code stream sequence number

【NB】

None.

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

2.2.5 IMPEncoderEncAttr

【Explain】

Define the encoder attribute structure.

【Definition】

```
typedef struct {
    IMPEncoderProfile       eProfile;
    uint8_t                 uLevel;
    uint8_t                 uTier;
    uint16_t                uWidth;
    uint16_t                uHeight;
    IMPEncoderPicFormat     ePicFormat;
    uint32_t                eEncOptions;
    uint32_t                eEncTools;
    IMPEncoderCropCfg       crop;
} IMPEncoderEncAttr;
```

【Member】

Member name	Describe
eProfile	Encoder protocol + coding framework profile
uLevel	Encoder level, only for h264、h265
uTier	Only for h265；0：main　1：high
uWidth	Encoder image width
uHeight	Encoder image height
ePicFormat	h264/5Z support 400,420；JPEG support 400,420,422
eEncOptions	Encoder options
eEncTools	Encoder tools
crop	Encoder clipping properties

【NB】

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

None.

2.2.6 IMPEncoderAttrCbr

【Explain】

Encoder in CBR mode attribute structure.

【Definition】

```
typedef struct {
    uint32_t            uTargetBitRate;
    int16_t             iInitialQP;
    int16_t             iMinQP;
    int16_t             iMaxQP;
    int16_t             iIPDelta;
    int16_t             iPBDelta;
    uint32_t            eRcOptions;
    uint32_t            uMaxPictureSize;
} IMPEncoderAttrCbr;
```

【Member】

Member name	Describe
uTargetBitRate	Target stream
iInitialQP	Initialization QP
iMinQP	Minimum QP
iMaxQP	Maximum QP
iIPDelta	QP difference between I frame and next P frame
iPBDelta	QP difference between P frame and B frame
eRcOptions	Encoder options
uMaxPictureSize	Maximum picture size

【NB】

None.

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

2.2.7 IMPEncoderAttrVbr

【Explain】

Encoder in VBR mode attribute structure.

【Definition】

```
typedef struct {
    uint32_t            uTargetBitRate;
    uint32_t            uMaxBitRate;
    int16_t             iInitialQP;
    int16_t             iMinQP;
    int16_t             iMaxQP;
    int16_t             iIPDelta;
    int16_t             iPBDelta;
    uint32_t            eRcOptions;
    uint32_t            uMaxPictureSize;
} IMPEncoderAttrVbr;
```

【Member】

Member name	Describe
uTargetBitRate	Target stream
uMaxBitRate	Initialization QP
iInitialQP	Minimum QP
iMinQP	Maximum QP
iMaxQP	QP difference between I frame and next P frame
iIPDelta	QP difference between P frame and B frame
iPBDelta	Encoder options
eRcOptions	Maximum picture size
uMaxPictureSize	Target stream

【NB】

None.

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

2.2.8 IMPEncoderAttrCappedVbr

【Explain】

Encoder in CappedVBR mode attribute structure.

【Definition】

```
typedef struct {
    uint32_t            uTargetBitRate;
    uint32_t            uMaxBitRate;
    int16_t             iInitialQP;
    int16_t             iMinQP;
    int16_t             iMaxQP;
    int16_t             iIPDelta;
    int16_t             iPBDelta;
    uint32_t            eRcOptions;
    uint32_t            uMaxPictureSize;
    uint16_t            uMaxPSNR;
} IMPEncoderAttrCappedVbr;
```

【Member】

Member name	Describe
uTargetBitRate	Target stream
uMaxBitRate	Initialization QP
iInitialQP	Minimum QP
iMinQP	Maximum QP
iMaxQP	QP difference between I frame and next P frame
iIPDelta	QP difference between P frame and B frame
iPBDelta	Encoder options
eRcOptions	Maximum picture size
uMaxPictureSize	Target stream

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

uMaxPSNR	When the value of uMaxPSNR is reached, the bit stream is not increased

【NB】

None.

2.2.9 IMPEncoderAttrFixQP

【Explain】

Encoder in FixQP mode attribute structure.

【Definition】

```
typedef struct {
    int16_t          iInitialQP;
} IMPEncoderAttrFixQP;
```

【Member】

Member name	Describe
iInitialQP	Initialization QP

【NB】

None.

2.2.10 IMPEncoderAttrRcMode

【Explain】

Encoder encoding mode attribute structure.

【Definition】

```
typedef struct {
    IMPEncoderRcMode                rcMode;
    union {
        IMPEncoderAttrFixQP         attrFixQp;
```


Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.

```
        IMPEncoderAttrCbr                attrCbr;
        IMPEncoderAttrVbr                attrVbr;
        IMPEncoderAttrCappedVbr          attrCappedVbr;
        IMPEncoderAttrCappedQuality attrCappedQuality;
    };
} IMPEncoderAttrRcMode;
```

【Member】

Member name	Describe
rcMode	Encoder mode
attrFixQp	FixQP mode
attrCbr	CBR mode
attrVbr	VBR mode
attrCappedVbr	CappedVBR mode
attrCappedQuality	CapedQuality mode

【NB】

None.

Copyright® 2021-2023 Ingenic Semiconductor Co., Ltd. All rights reserved.