

INGENIC®

T31 IPU API Reference

Date: 2022-04

Viewer: Jason Xu



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

INGENIC®

T31 IPU API Reference

Copyright © Ingenic Semiconductor Co. Ltd 2021. All rights reserved.

Release history

Date	Revision	Change
2022-04	1.0	First release

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](http://www.ingenic.com)

CONTENT

1 IPU MODULE OVERVIEW	2
1.1 MODULE INTRODUCTION	2
1.2 ORGANIZATION CHART	3
1.3 KEY WORDS DESCRIPTION	3
2 SDK USAGE	4
2.1 IMP_OSD IMAGE INFORMATION OVERLAY MODULE	4
2.1.1 Function introduction	4
2.1.2 Module operation process	4
2.2 IPU MODULE API	5
2.2.1 IMP_OSD_CreateGroup	5
2.2.2 IMP_OSD_DestroyGroup	8
2.2.3 IMP_OSD_CreateRgn	9
2.2.4 IMP_OSD_DestroyRgn	10
2.2.5 IMP_OSD_RegisterRgn	11
2.2.6 IMP_OSD_UnRegisterRgn	11
2.2.7 IMP_OSD_SetRgnAttr	12
2.2.8 IMP_OSD_GetRgnAttr	13
2.2.9 IMP_OSD_SetRgnAttrWithTimestamp	14
2.2.10 IMP_OSD_UpdateRgnAttrData	15
2.2.11 IMP_OSD_SetGrpRgnAttr	16
2.2.12 IMP_OSD_GetGrpRgnAttr	16
2.2.13 IMP_OSD_ShowRgn	17
2.2.14 IMP_OSD_Start	18
2.2.15 IMP_OSD_Stop	19
2.3 DATA TYPE	20
2.3.1 IMPOsdColour	20
2.3.2 IMPOsdRgnType	21
2.3.3 IMPOSDRgnAttrData	22
2.3.4 IMPOSDRgnAttr	23
2.3.5 IMPOSDRgnTimestamp	23
2.3.6 IMPOSDGrpRgnAttr	24
2.4 SDK SAMPLEINTRODUCTION	25

1 IPU Module overview

1.1 Module introduction

IPU is an image processing unit (Image Processing Unit). It supports the data processing from the image collected by ISP module to the video display module. For example, add specific information to the video, such as straight line, rectangular box, rectangle occlusion, picture data, etc.

The operation of the image by the IPU module mainly includes OSD module and CSC module. OSD module mainly superimposes frame lines, rectangular occlusion, pictures and other data on the video frame; CSC can transform the input video frame into hardware supported image, such as: HSV, nv12, nv21, rgb32, and ARGB.

The data information superimposed on the video data is called OSD area (Region). The OSD device module is used in the system to manage these OSD areas, it supports region creation and destruction, configuration of the type of region through the region property setting interface (For example, configure the properties, position coordinates, foreground background, etc. displayed in the area), and other operations. After creating, registering and configuring properties in OSD area, IPU will be called to draw in the OSD area.

Users can use [IMP_OSD_CreateRgn](#) to create a region, then use [IMP_OSD_RegisterRgn](#) to register the region to the OSD group and [IMP_OSD_SetRgnAttr](#) to set the area attribute to add the superimposed information to the OSD module.

1.2 Organization chart

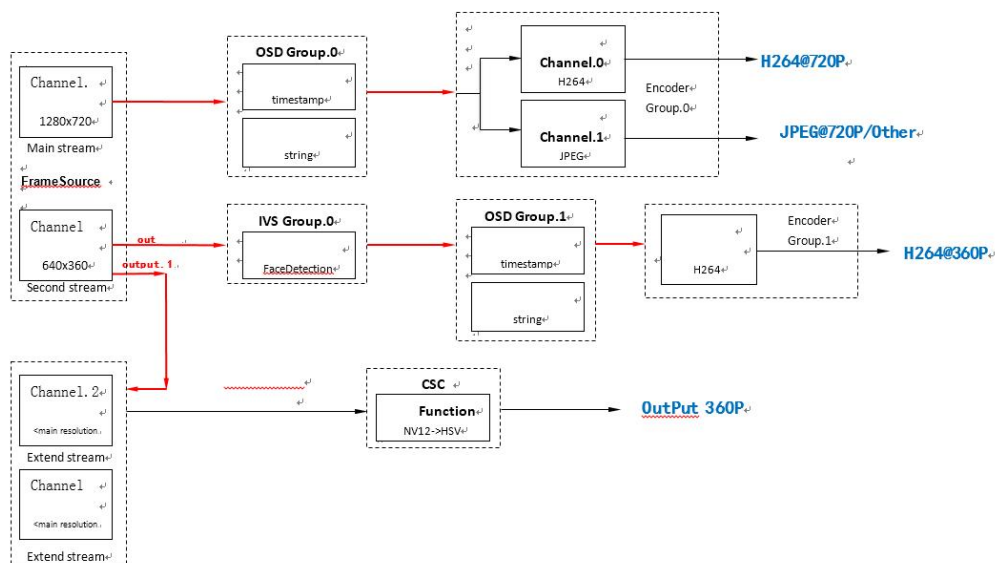


Figure 1.1 OSD, CSC context diagram

1.3 Key words description

- IPU: Image Processing Unit.
- OSD: On-Screen Display (OSD) adjusts the screen menu mode
- REGION: It is the module area.
- DEVICE MOUDLE: Device management module, it is the internal system through the device management of each module.
- CSC: The Color Space Conversion (CSC) is used to convert the input image format into any format supported by hardware
- REGION PROPERTIES: Users need to configure the area information to be displayed. These attributes include the information to be displayed, such as the address information of the foreground to be displayed, the position information of the foreground in the background, the format of the foreground pixels, etc.

2 SDK usage

2.1 IMP_OSD Image information overlay module

2.1.1 Function introduction

OSD (On-Screen Display) module can overlay pictures, bitmaps, lines, and rectangular boxes on the video stream. It is used to superimpose lines, pictures and other information in each frame of videos.

After compiling, run the following sample in the device to generate the OSD file configured in the overlay sample. A video file will be created under /tmp/, you can play the file with VLC or other multimedia software.

Application file	Function	Command	Results
sample-OSD.c	Video overlay OSD data	./sample-OSD	Generate video file stream-0.h265

2.1.2 Module operation process

2.1.2.1 Init process

- 1) Create OSD group: use [IMP_OSD_CreateGroup](#) to create an OSD group;
- 2) Create OSD area: use [IMP_OSD_CreateRgn](#) to create an area;
- 3) Register OSD area to OSD group: use [IMP_OSD_RegisterRgn](#) to register the created region to OSD group;
- 4) Set the OSD group area attribute and area attribute: use [IMP_OSD_SetRgnAttr](#) and [IMP_OSD_SetGrpRgnAttr](#) to set the attributes of the region and group respectively
- 5) Set OSD function display switch: use [IMP_OSD_ShowRgn](#) to turn on OSD display;
- 6) Bind the OSD group to the system: use [IMP_System_Bind](#) to bind the OSD module to other modules.

2.1.2.2 Exit process

- 1) Unbind: use [IMP_System_UnBind](#) to unbind the OSD group from the system.
- 2) Turn off OSD function display: use [IMP_OSD_ShowRgn](#) to turn off the OSD display.
- 3) Log off the OSD area from the OSD group with [IMP_OSD_UnRegisterRgn](#).

- 4) Destroy the OSD area with [IMP_OSD_DestroyRgn](#).
- 5) Destroy OSD group with [IMP_OSD_DestroyGroup](#).

Note: please refer to sample for other details

2.2 IPU Module API

API Name	Function
IMP_OSD_CreateGroup	Create an OSD group
IMP_OSD_DestroyGroup	Destroy an OSD group
IMP_OSD_CreateRgn	Create an OSD area
IMP_OSD_DestroyRgn	Destroy an OSD area
IMP_OSD_RegisterRgn	Registration area
IMP_OSD_UnRegisterRgn	Logout area
IMP_OSD_SetRgnAttr	Set region properties
IMP_OSD_GetRgnAttr	Get region properties
IMP_OSD_SetRgnAttrWithTimestamp	Set the effective time of regional property
IMP_OSD_UpdateRgnAttrData	Update area data properties
IMP_OSD_SetGrpRgnAttr	Set OSD group area properties
IMP_OSD_GetGrpRgnAttr	Get OSD group area properties
IMP_OSD_ShowRgn	Set up the OSD display switch
IMP_OSD_Start	Set to start the display of OSD group
IMP_OSD_Stop	Set to stop the display of OSD group

2.2.1 IMP_OSD_CreateGroup

【Function】

Create an OSD group.

【Grammar】

```
int IMP_OSD_CreateGroup(int grpNum);
```

【Formal parameter】

Parameter name	Describe	Input/output
grpNum	OSD group ID: Value range :[0, nr_max_osd_groups-1]	Input

【Return value】

Return value	Describe
0	Success
none 0	Fail

【Dependence】

Head file: imp_osd.h

Lib file: libimp.a / libimp.so

【Example】

```

1. int grpNum = 0;
2. IMPRgnHandle rHandlerLogo;
3. IMPOSDRgnAttr rAttrLogo;
4. IMPOSDGrpRgnAttr grAttrLogo;
5. IMPOSDRgnCreateStat stStatus;
6. IMPOSDRgnRegisterStat stRigStatus;
7. memset(&stRigStatus,0,sizeof(IMPOSDRgnRegisterStat));
8. memset(&stStatus,0,sizeof(IMPOSDRgnCreateStat));
9. memset(&rAttrLogo, 0, sizeof(IMPOSDRgnAttr));
10. memset(&grAttrLogo, 0, sizeof(IMPOSDGrpRgnAttr));
11.
12. if (IMP_OSD_CreateGroup(grpNum) < 0) {
13.     IMP_LOG_ERR(TAG, "IMP_OSD_CreateGroup(%d) error !\n", grpNum);
14.     return -1;
15. }
16.
17. rHandlerLogo = IMP_OSD_CreateRgn(NULL);
18. if (rHandlerLogo == INVHANDLE) {
19.     IMP_LOG_ERR(TAG, "IMP_OSD_CreateRgn Logo error !\n");
20.     return -1;
21. }
22. //query osd rgn create status
23. ret = IMP_OSD_RgnCreate_Query(rHandlerLogo,&stStatus);
24. if(ret < 0){
25.     IMP_LOG_ERR(TAG, "IMP_OSD_RgnCreate_Query error !\n");
26.     return NULL;
27. }
28. ret = IMP_OSD_RegisterRgn(rHandlerLogo, grpNum, NULL);
29. if (ret < 0) {

```



```

30.    MP_LOG_ERR(TAG, "IVS IMP_OSD_RegisterRgn failed\n");
31.    return -1;
32. }
33.
34. ret = IMP_OSD_RgnRegister_Query(rHandlerFont, grpNum, &stRigStatus);
35. if (ret < 0) {
36.    IMP_LOG_ERR(TAG, "IMP_OSD_RgnRegister_Query failed\n");
37.    return NULL;
38. }
39. int picw = 100;
40. int pich = 100;
41. rAttrLogo.type = OSD_REG_PIC;
42. rAttrLogo.rect.p0.x = SENSOR_WIDTH - 100;
43. rAttrLogo.rect.p0.y = SENSOR_HEIGHT - 100;
44. rAttrLogo.rect.p1.x = rAttrLogo.rect.p0.x+picw-1;    /*p0 is start , and
    p1 well be epual p0+width(or heighth)-1*/
45. rAttrLogo.rect.p1.y = rAttrLogo.rect.p0.y+pich-1;
46. rAttrLogo.fmt = PIX_FMT_BGRA;
47. rAttrLogo.data.picData.pData = logodata_100x100_bgra;
48. ret = IMP_OSD_SetRgnAttr(rHandlerLogo, &rAttrLogo);
49. if (ret < 0) {
50.    IMP_LOG_ERR(TAG, "IMP_OSD_SetRgnAttr Logo error !\n");
51.    return NULL;
52. }
53.
54. if (IMP_OSD_GetGrpRgnAttr(rHandlerLogo, grpNum, &grAttrLogo) < 0) {
55.    IMP_LOG_ERR(TAG, "IMP_OSD_GetGrpRgnAttr Logo error !\n");
56.    return NULL;
57. }
58. }
59. memset(&grAttrLogo, 0, sizeof(IMPOSDGrpRgnAttr));
60. grAttrLogo.show = 0;
61.
62. /* Set Logo global alpha to 0x7f, it is semi-transparent. */
63. grAttrLogo.gAlphaEn = 1;
64. grAttrLogo.fgAlhpa = 0x7f;
65. grAttrLogo.layer = 2;
66.
67. if (IMP_OSD_SetGrpRgnAttr(rHandlerLogo, grpNum, &grAttrLogo) < 0) {
68.    IMP_LOG_ERR(TAG, "IMP_OSD_SetGrpRgnAttr Logo error !\n");
69.    return NULL;

```

```

70. }
71.
72. /*other operation*/
73.

```

2.2.2 IMP_OSD_DestroyGroup

【Function】

Destroy OSD group.

【Grammar】

```
int IMP_OSD_DestroyGroup(int grpNum);
```

【Formal parameter】

Parameter name	Describe	Input/output
grpNum	OSD group ID: Value range :[0, nr_max_osd_groups-1]	Input

【Return value】

Return value	Describe
0	Success
none 0	Fail

【Dependence】

Head file: imp_osd.h

Lib file: libimp.a / libimp.so

【NB】

The grpNum group for this interface must have been created.

【Example】

```

1. int ret = 0;
2. int grpNum = 0;
3. ret = IMP_OSD_ShowRgn(rHandlerLogo, grpNum, 0);

```

```

4. if (ret < 0) {
5.     IMP_LOG_ERR(TAG, "IMP_OSD_ShowRgn close timeStamp error\n");
6. }
7.
8. ret = IMP_OSD_UnRegisterRgn(rHandlerLogo, grpNum);
9. if (ret < 0) {
10.    IMP_LOG_ERR(TAG, "IMP_OSD_UnRegisterRgn timeStamp error\n");
11. }
12.
13. IMP_OSD_DestroyRgn(rHandlerLogo);
14.
15. ret = IMP_OSD_DestroyGroup(grpNum);
16. if (ret < 0) {
17.    IMP_LOG_ERR(TAG, "IMP_OSD_DestroyGroup(0) error\n");
18.    return -1;
19. }
20. return ret;

```

2.2.3 IMP_OSD_CreateRgn

【Function】

Create an OSD area.

【Grammar】

IMPRgnHandle IMP_OSD_CreateRgn(IMPOSDRgnAttr *prAttr);

【Formal parameter】

Parameter name	Describe	Input/output
prAttr	Osd Area property structure pointer	Input

【Return value】

Return value	Describe
>=0	Success
<0	Fail

【Dependence】

Head file: `imp_osd.h`

Lib file: `libimp.a / libimp.so`

【NB】

The `grpNum` group for this interface must have been created.

【Example】

Please refer to [IMP_OSD_CreateGroup](#) Example.

2.2.4 IMP_OSD_DestroyRgn

【Function】

Destroy an OSD area.

【Grammar】

```
void IMP_OSD_DestroyRgn(IMPRgnHandle handle);
```

【Formal parameter】

Parameter name	Describe	Input/output
handle	Handle to the created region	Input

【Return value】

None.

【Dependence】

Head file: `imp_osd.h`

Lib file: `libimp.a / libimp.so`

【NB】

The area corresponding to handle has been created for this interface..

【Example】

Please refer to [IMP_OSD_DestroyGroup](#) Example.

2.2.5 IMP_OSD_RegisterRgn

【Function】

Registeran OSD area.

【Grammar】

```
Int IMP_OSD_RegisterRgn(IMPRgnHandle handle, int grpNum, IMPOSDGrpRgnAttr
*pgrAttr);
```

【Formal parameter】

Parameter name	Describe	Input/output
handle	Handle to the created region	Input
grpNum	OSD group ID: Value range :[0, nr_max_osd_groups-1]	Input
pgrAttr	OSD group displays property structure Pointers	Input

【Return value】

Return value	Describe
0	Success
none 0	Fail

【Dependence】

Head file: imp_osd.h

Lib file: libimp.a / libimp.so

【NB】

The area corresponding to handle has been created for this interface..

【Example】

Please refer to [IMP_OSD_CreateGroup](#) Example.

2.2.6 IMP_OSD_UnRegisterRgn

【Function】

Destroy the OSD area.

【Grammar】

```
int IMP_OSD_UnRegisterRgn(IMPRgnHandle handle, int grpNum);
```

【Formal parameter】

Parameter name	Describe	Input/output
handle	Handle to the created region	Input
grpNum	OSD group ID	Input

【Return value】

Return value	Describe
0	Success
none 0	Fail

【Dependence】

Head file: `imp_osd.h`

Lib file: `libimp.a / libimp.so`

【NB】

The area corresponding to handle has been created for this interface..

【Example】

Please refer to [IMP_OSD_DestroyGroup](#) Example.

2.2.7 IMP_OSD_SetRgnAttr

【Function】

Set region propertie.

【Grammar】

```
int IMP_OSD_SetRgnAttr(IMPRgnHandle handle, IMPOSDRgnAttr *prAttr);
```

【Formal parameter】

Parameter name	Describe	Input/output
handle	Handle to the created region	Input
prAttr	Osd Area property structure pointer	Input

【Return value】

Return value	Describe
0	Success
none 0	Fail

【Dependence】

Head file: imp_osd.h

Lib file: libimp.a / libimp.so

【Example】

Please refer to [IMP_OSD_CreateGroup](#) Example.

2.2.8 IMP_OSD_GetRgnAttr

【Function】

Get region properties.

【Grammar】

```
int IMP_OSD_GetRgnAttr(IMPRgnHandle handle, IMPOSDRgnAttr *prAttr);
```

【Formal parameter】

Parameter name	Describe	Input/output
handle	Handle to the created region	Input
prAttr	Osd Area property structure pointer	Output

【Return value】

Return value	Describe
0	Success
none 0	Fail

【Dependence】

Head file: imp_osd.h

Lib file: libimp.a / libimp.so

【Example】

Please refer to [IMP_OSD_CreateGroup](#) Example.

2.2.9 IMP_OSD_SetRgnAttrWithTimestamp

【Function】

Set the effective time of regional property.

【Grammar】

```
int IMP_OSD_SetRgnAttrWithTimestamp(IMPRgnHandle handle, IMPOSDRgnAttr *prAttr,
IMPOSDRgnTimestamp *prTs);
```

【Formal parameter】

Parameter name	Describe	Input/output
handle	Handle to the created region	Input
prAttr	Osd Area property structure pointer	Input
prTs	Effect of time	Input

【Return value】

【Return value】	Describe
0	Success
none 0	Fail

【Dependence】

Head file: imp_osd.h

Lib file: libimp.a / libimp.so

【Example】

2.2.10 IMP_OSD_UpdateRgnAttrData

【Function】

Update area data properties, only for the region types of OSD_REG_BITMAP and OSD_REG_PIC, OSD_REG_PIC_RMEM.

【Grammar】

```
int IMP_OSD_UpdateRgnAttrData(IMPRgnHandle handle, IMPOSDRgnAttrData *prAttrData);
```

【Formal parameter】

Parameter name	Describe	Input/output
handle	Handle to the created region	Input
prAttrData	OSD Area data attribute structure pointer	Input

【Return value】

Return value	Describe
0	Success
none 0	Fail

【Dependence】

Head file: imp_osd.h

Lib file: libimp.a / libimp.so

【Example】

2.2.11 IMP_OSD_SetGrpRgnAttr

【Function】

设置 Area attributes of the OSD group.

【Grammar】

```
int IMP_OSD_SetGrpRgnAttr(IMPRgnHandle handle, int grpNum, IMPOSDGrpRgnAttr
*pgrAttr);
```

【Formal parameter】

Parameter name	Describe	Input/output
handle	Handle to the created region	Input
grpNum	OSD group ID	Input
pgrAttr	Area attributes of the OSD group	Input

【Return value】

Return value	Describe
0	Success
none 0	Fail

【Dependence】

Head file: imp_osd.h

Lib file: libimp.a / libimp.so

【Example】

Please refer to [IMP_OSD_CreateGroup](#) Example.

2.2.12 IMP_OSD_GetGrpRgnAttr

【Function】

Get OSD group area properties.

【Grammar】

```
int IMP_OSD_GetGrpRgnAttr(IMPRgnHandle handle, int grpNum,
    IMPOSDGrpRgnAttr *pgrAttr);
```

【Formal parameter】

Parameter name	Describe	Input/output
handle	Handle to the created region	Input
grpNum	OSD group ID	Input
pgrAttr	Area attributes of the OSD group structure pointer	Output

【Return value】

Return value	Describe
0	Success
none 0	Fail

【Dependence】

Head file: `imp_osd.h`

Lib file: `libimp.a / libimp.so`

【NB】

This API calls require that the corresponding OSD group has been created and the region is created and registered.

【Example】

Please refer to [IMP_OSD_CreateGroup](#) Example.

2.2.13 IMP_OSD_ShowRgn

【Function】

Set up the OSD display switch.

【Grammar】

```
int IMP_OSD_ShowRgn(IMPRgnHandle handle, int grpNum, int showFlag);
```

【Formal parameter】

Parameter name	Describe	Input/output
handle	Handle to the created region	Input
grpNum	OSD group ID	Input
showFlag	OSD display switch sign, 0 is off and 1 is on	Input

【Return value】

Return value	Describe
0	Success
none 0	Fail

【Dependence】

Head file: imp_osd.h

Lib file: libimp.a / libimp.so

【Example】

Please refer to [IMP_OSD_DestroyGroup](#) Example.

2.2.14 IMP_OSD_Start

【Function】

Set up the display of the start OSD group.

【Grammar】

```
int IMP_OSD_Start(int grpNum);
```

【Formal parameter】

Parameter name	Describe	Input/output

grpNum	OSD group ID	Input
--------	--------------	-------

【Return value】

Return value	Describe
0	Success
none 0	Fail

【Dependence】

Head file: imp_osd.h

Lib file: libimp.a / libimp.so

【Example】

2.2.15 IMP_OSD_Stop

【Function】

Set to stop the display of OSD group.

【Grammar】

```
int IMP_OSD_Stop(int grpNum);
```

【Formal parameter】

Parameter name	Describe	Input/output
grpNum	OSD group ID	Input

【Return value】

Return value	Describe
0	Success
none 0	Fail

【Dependence】

Head file: imp_osd.h

Lib file: libimp.a / libimp.so

【NB】

When this API is called, the corresponding OSD group has been created.

【Example】

2.3 Data type

Name	Definition
IMPOsdColour	Define the OSD color type for non IPU painting
IMPOsdRgnType	Define OSD area type
IMPOSDRgnAttrData	Define the data of OSD area
IMPOSDRgnAttr	Define OSD area attributes
IMPOSDRgnTimestamp	Define the effective time stamp of OSD locale
IMPOSDGrpRgnAttr	Define OSD group area properties

2.3.1 IMPOsdColour

【Explain】

OSD color type, with the color format given as bgra.

【Definition】

```
typedef enum {
    OSD_BLACK    = 0xff000000,
    OSD_WHITE    = 0xffffffff,
    OSD_RED      = 0xffff0000,
    OSD_GREEN    = 0xff00ff00,
    OSD_BLUE    = 0xff0000ff,
}IMPOsdColour;
```

【Member】

Member name	Describe
-------------	----------

OSD_BLACK	The drawing area content color is black
OSD_WHITE	The drawing area content color is white
OSD_RED	The drawing area content color is red
OSD_GREEN	The drawing area content color is green
OSD_BLUE	The drawing area content color is blue

【NB】

None.

2.3.2 IMPOsdRgnType

【Explain】

The OSD region type enumeration.

【Definition】

```
typedef enum {
    OSD_REG_INV          = 0,
    OSD_REG_LINE         = 1,
    OSD_REG_RECT         = 2,
    OSD_REG_BITMAP       = 3,
    OSD_REG_COVER        = 4,
    OSD_REG_PIC          = 5,
    OSD_REG_PIC_RMEM     = 6,
} IMPOsdRgnType;
```

【Member】

Member name	Describe
OSD_REG_INV	Not Definition, reserved
OSD_REG_LINE	Line type
OSD_REG_RECT	Rectangular type
OSD_REG_BITMAP	Pot array image type
OSD_REG_COVER	Rectangular block type
OSD_REG_PIC	Picture type, suitable for logo or timestamp
OSD_REG_PIC_RMEM	Picture type, suitable for logo or

	timestamp, the memory type is rmem type
--	---

【NB】

OSD_REG_PIC and OSD_REG_PIC_RMEM are drawn by using IPU hardware. OSD_REG_PIC copies the contents of an OSD area to RMEM and then invokes IPU hardware to draw. OSD_REG_PIC_RMEM transfers the area to the IPU as the OSD node. Therefore, OSD_REG_PIC_RMEM performs better than OSD_REG_PIC.

2.3.3 IMOSDRgnAttrData

【Explain】

OSD zone attribute data.

【Definition】

```
typedef union {
    void          *bitmapData;
    lineRectData  lineRectData;
    coverData     coverData;
    picData       picData;
} IMOSDRgnAttrData;
```

【Member】

Member name	Describe
bitmapData	Dot array data
lineRectData	Line and box data
coverData	Block data
picData	Picture data

【NB】

None.

2.3.4 IMPOSDRgnAttr

【Explain】

OSD area attribute.

【Definition】

```
typedef struct {
    IMPOsdRgnType    type;
    IMPRect          rect;
    IMPPixelFormat    fmt;
    IMPOSDRgnAttrData data;
} IMPOSDRgnAttr;
```

【Member】

Member name	Describe
type	OSD area type
rect	Rectangular data
fmt	dot format
data	The OSD zone attribute data

【NB】

None.

2.3.5 IMPOSDRgnTimestamp

【Explain】

OSD area effective timestamp.

【Definition】

```
typedef struct {
    uint64_t ts;
    uint64_t minus;
    uint64_t plus;
} IMPOSDRgnTimestamp;
```

【Member】

Member name	Describe
ts	time stamp
minus	lower limit
plus	superior limit

【NB】

None.

2.3.6 IMPOSDGrpRgnAttr

【Explain】

Area attributes of the OSD group.

【Definition】

```
typedef struct {
    int          show;
    IMPPoint     offPos;
    float        scalex;
    float        scaley;
    int          gAlphaEn;
    int          fgAlpha;
    int          bgAlpha;
    int          layer;
} IMPOSDGrpRgnAttr;
```

【Member】

Member name	Describe
show	osd area display switch, 0 is off display and 1 on display
offPos	Displays the starting coordinates
scalex	Scale the x parameter
scaley	Scale y parameter
gAlphaEn	Alpha switch
fgAlpha	foreground Alpha
bgAlpha	background Alpha

layer	presentation layer
-------	--------------------

【NB】

None.

2.4 SDK Sampleintroduction

The Ingenic team is on it. No one specific. Applicati on file	Function	Command	Results
sample-OSD.c	Overlay OSD on video stream	./sample-OSD	Stream-0.h265 is generated in the board /tmp path by default. Users can put the file into VLC and other video playback software for playback