



HiDPU

## API 参考

文档版本      00B04  
发布日期      2018-07-06

**版权所有 © 深圳市海思半导体有限公司 2018。保留一切权利。**

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明



**HISILICON**、海思和其他海思商标均为深圳市海思半导体有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受海思公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，海思公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

## 深圳市海思半导体有限公司

地址：深圳市龙岗区坂田华为基地华为电气生产中心 邮编：518129

网址：<http://www.hisilicon.com>

客户服务电话：+86-755-28788858

客户服务传真：+86-755-28357515

客户服务邮箱：[support@hisilicon.com](mailto:support@hisilicon.com)



# 前 言

## 概述

本文档为使用海思媒体处理芯片的 DPU(Depth Process Unit)开发的程序员而写，目的是供您在开发过程中查阅 DPU 支持的各种参考信息，包括 API、头文件、错误码等。



### 说明

未有特殊说明，Hi3559CV100 与 Hi3559AV100 内容一致。

未有特殊说明，Hi3556AV100 与 Hi3519AV100 内容一致。

## 产品版本

与本文档相对应的产品版本如下。

产品名称	产品版本
Hi3559A	V100
Hi3559C	V100
Hi3519A	V100
Hi3556A	V100

## 读者对象




本文档（本指南）主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师

## 符号约定

在本文中可能出现下列标志，它们所代表的含义如下。



符号	说明
 危险	表示有高度潜在危险，如果不能避免，会导致人员死亡或严重伤害。
 警告	表示有中度或低度潜在危险，如果不能避免，可能导致人员轻微或中等伤害。
 注意	表示有潜在风险，如果忽视这些文本，可能导致设备损坏、数据丢失、设备性能降低或不可预知的结果。
 窍门	表示能帮助您解决某个问题或节省您的时间。
 说明	表示是正文的附加信息，是对正文的强调和补充。

## 修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

日期	版本	修改描述
2018-07-06	00B04	HI_MPI_DPU_RECT_SendFrame 和 HI_MPI_DPU_RECT_GetFrame【参数】涉及修改 1.6.2 和 2.6.2 小节涉及修改
2018-04-10	00B03	添加 Hi3519AV100 的相关内容 2.4 小节，DPU_MATCH_GRP_ATTR_S 涉及修改
2018-03-15	00B02	第二次临时版本发布 1.6.2 和 2.6.2 小节涉及修改
2018-01-05	00B01	第一次临时版本发布



# 目 录

<b>1 DPU_RECT</b>	<b>1</b>
1.1 概述	1
1.2 功能描述	1
1.2.1 重要概念	1
1.2.2 调用流程	1
1.3 API 参考	2
1.4 数据类型和数据结构	20
1.5 错误码	24
1.6 Proc 调试信息	25
1.6.1 概述	25
1.6.2 Proc 信息说明	26
<b>2 DPU_MATCH</b>	<b>31</b>
2.1 概述	31
2.2 功能描述	31
2.2.1 重要概念	31
2.2.2 调用流程	31
2.3 API 参考	32
2.4 数据类型和数据结构	47
2.5 错误码	55
2.6 Proc 调试信息	56
2.6.1 概述	56
2.6.2 Proc 信息说明	56



## 插图目录

图 1-1 DPU_RECT_MEM_INFO_S 数据内存示意.....	1
图 1-2 DPU_RECT 无绑定目标场景调用流程.....	2
图 1-3 DPU_RECT 绑定目标场景调用流程.....	2
图 2-1 DPU_MATCH_MEM_INFO_S 数据内存示意.....	31
图 2-2 DPU_MATCH 调用流程.....	32



## 表格目录

表 1-1 DPU_RECT 模块 API 错误码 .....	24
表 2-1 DPU_MATCH 模块 API 错误码 .....	55



# 1 DPU\_RECT

## 1.1 概述

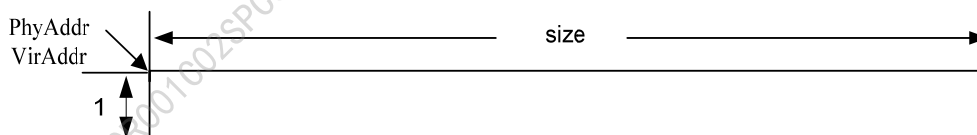
DPU(Depth Process Unit)对输入的左图像和右图像经过校正和匹配计算得出深度图。DPU 分为校正(DPU\_RECT)和匹配(DPU\_MATCH)两个模块，同时使用，也可单独使用。DPU\_RECT 对输入的左图像和右图像进行校正。

## 1.2 功能描述

### 1.2.1 重要概念

- 组(GROUP)  
DPU\_RECT 对用户提供的组 (GROUP) 的概念。最大可用数为 8 个，各组分时复用 DPU\_RECT 硬件。
- 通道(CHANNEL)  
DPU\_RECT 组的通道。每组包含两个通道，分为左右两通道。
- 管道(PIPE)  
DPU\_RECT 组的管道。每组包含两个管道，分为左右两管道。
- DPU\_RECT\_MEM\_INFO\_S 内存排布

图1-1 DPU\_RECT\_MEM\_INFO\_S 数据内存示意



### 1.2.2 调用流程

典型场景调用流程按是否绑定目标场景分为如图 1-2 和图 1-3 两种。若是绑定目标场景，一般绑定 DPU\_MATCH 模块。





图1-2 DPU\_RECT 无绑定目标场景调用流程

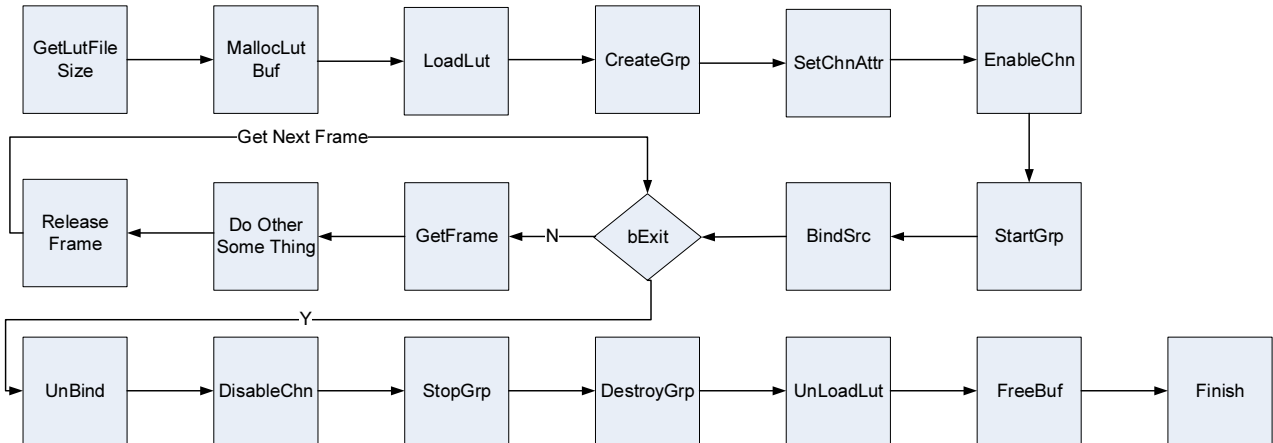
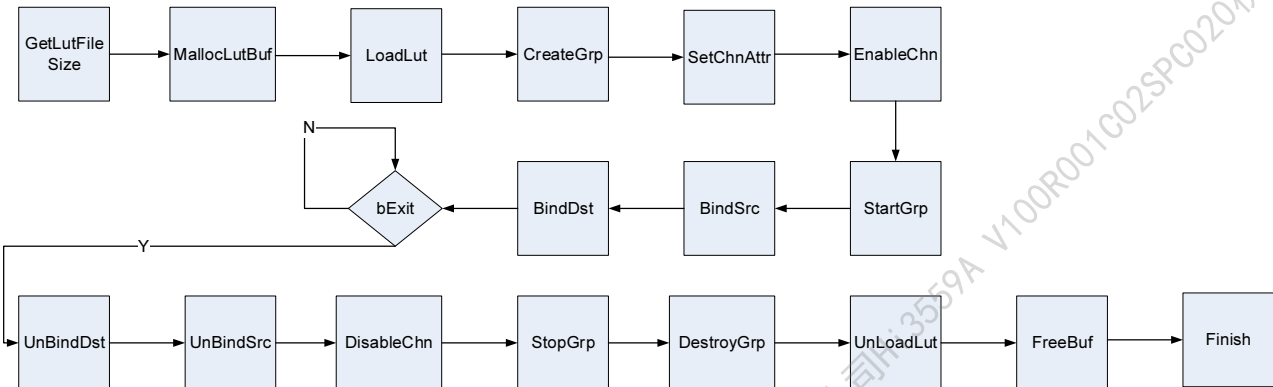


图1-3 DPU\_RECT 绑定目标场景调用流程



## 1.3 API 参考

该功能模块为用户提供以下 MPI:

- [HI\\_MPI\\_DPU\\_RECT\\_GetLutSize](#): 获取查找表所需内存字节。
- [HI\\_MPI\\_DPU\\_RECT\\_LoadLut](#): 加载查找表。
- [HI\\_MPI\\_DPU\\_RECT\\_UnloadLut](#): 卸载查找表。
- [HI\\_MPI\\_DPU\\_RECT\\_CreateGrp](#): 创建组。
- [HI\\_MPI\\_DPU\\_RECT\\_DestroyGrp](#): 销毁组。
- [HI\\_MPI\\_DPU\\_RECT\\_SetGrpAttr](#): 设置组属性。
- [HI\\_MPI\\_DPU\\_RECT\\_GetGrpAttr](#): 获取组属性。
- [HI\\_MPI\\_DPU\\_RECT\\_StartGrp](#): 启用组。
- [HI\\_MPI\\_DPU\\_RECT\\_StopGrp](#): 禁用组。
- [HI\\_MPI\\_DPU\\_RECT\\_SetChnAttr](#): 设置通道属性。



- [HI\\_MPI\\_DPU\\_RECT\\_GetChnAttr](#): 获取通道属性。
- [HI\\_MPI\\_DPU\\_RECT\\_EnableChn](#): 启用通道。
- [HI\\_MPI\\_DPU\\_RECT\\_DisableChn](#): 禁用通道。
- [HI\\_MPI\\_DPU\\_RECT\\_SendFrame](#): 用户发送数据。
- [HI\\_MPI\\_DPU\\_RECT\\_GetFrame](#): 用户从通道获取一帧处理完成的图像。
- [HI\\_MPI\\_DPU\\_RECT\\_ReleaseFrame](#): 用户释放一帧通道图像。

## HI\_MPI\_DPU\_RECT\_GetLutSize

### 【描述】

获取查找表所需内存字节。

### 【语法】

```
HI_S32 HI_MPI_DPU_RECT_GetLutSize(const HI_CHAR *pchFileName, HI_U32
*pu32Size);
```

### 【参数】

参数名称	描述	输入/输出
pchFileName	查找表文件名。 不能为空。	输入
pu32Size	查找表所需内存字节。 不能为空。	输出

### 【返回值】

返回值	描述
0	成功。
非 0	失败，参见 <a href="#">错误码</a> 。

### 【需求】

- 头文件: hi\_comm\_dpu\_rect.h、mpi\_dpu\_rect.h
- 库文件: libdpu\_rect.a

### 【注意】

查找表文件必须是利用海思提供的转化工具转化后的文件。

### 【举例】

无。



【相关主题】

[HI\\_MPI\\_DPU\\_RECT\\_LoadLut](#)

## HI\_MPI\_DPU\_RECT\_LoadLut

【描述】

加载查找表。

【语法】

```
HI_S32 HI_MPI_DPU_RECT_LoadLut(const HI_CHAR *pchFileName, const  
DPU_RECT_MEM_INFO_S *pstLutMem, DPU_RECT_LUT_ID *pDpuRectLutId);
```

【参数】

参数名称	描述	输入/输出
pchFileName	查找表文件名。 不能为空。	输入
pstLutMem	查找表所需内存字节。 不能为空。	输入
pDpuRectLutId	查找表 ID。 不能为空。	输出

参数名称	支持类型	地址对齐	分辨率
pstLutMem	-	pstLutMem->u64PhyAddr 16 字节对齐。	-

【返回值】

返回值	描述
0	成功。
非 0	失败，参见 <a href="#">错误码</a> 。

【需求】

- 头文件：hi\_comm\_dpu\_rect.h、mpi\_dpu\_rect.h
- 库文件：libdpu\_rect.a

【注意】



pstLutMem-> u64PhyAddr/ pstLutMem->u64VirAddr 必须是申请好的内存, pstLutMem-> u32Size 值从 [HI\\_MPI\\_DPU\\_RECT\\_GetLutSize](#) 获取。

【举例】

无。

【相关主题】

- [HI\\_MPI\\_DPU\\_RECT\\_GetLutSize](#)
- [HI\\_MPI\\_DPU\\_RECT\\_UnloadLut](#)

## HI\_MPI\_DPU\_RECT\_UnloadLut

【描述】

卸载查找表。

【语法】

```
HI_S32 HI_MPI_DPU_RECT_UnloadLut(DPU_RECT_LUT_ID DpuRectLutId);
```

【参数】

参数名称	描述	输入/输出
DpuRectLutId	查找表 ID。	输入

【返回值】

返回值	描述
0	成功。
非 0	失败, 参见 <a href="#">错误码</a> 。

【需求】

- 头文件: hi\_comm\_dpu\_rect.h、mpi\_dpu\_rect.h
- 库文件: libdpu\_rect.a

【注意】

DpuRectLutId 由 [HI\\_MPI\\_DPU\\_RECT\\_LoadLut](#) 获取到。

【举例】

无。

【相关主题】

[HI\\_MPI\\_DPU\\_RECT\\_LoadLut](#)



## HI\_MPI\_DPU\_RECT\_CreateGrp

### 【描述】

创建组。

### 【语法】

```
HI_S32 HI_MPI_DPU_RECT_CreateGrp(DPU_RECT_GRP DpuRectGrp, const  
DPU_RECT_GRP_ATTR_S *pstGrpAttr);
```

### 【参数】

参数名称	描述	输入/输出
DpuRectGrp	组号。 取值范围：[0, 8)。	输入
pstGrpAttr	组属性。 不能为空。	输入

### 【返回值】

返回值	描述
0	成功。
非 0	失败，参见 <a href="#">错误码</a> 。

### 【需求】

- 头文件：hi\_comm\_dpu\_rect.h、mpi\_dpu\_rect.h
- 库文件：libdpu\_rect.a

### 【注意】

不支持重复创建。

### 【举例】

无。

### 【相关主题】

- [HI\\_MPI\\_DPU\\_RECT\\_DestroyGrp](#)
- [HI\\_MPI\\_DPU\\_RECT\\_SetGrpAttr](#)
- [HI\\_MPI\\_DPU\\_RECT\\_GetGrpAttr](#)
- [HI\\_MPI\\_DPU\\_RECT\\_StartGrp](#)
- [HI\\_MPI\\_DPU\\_RECT\\_StopGrp](#)



## HI\_MPI\_DPU\_RECT\_DestroyGrp

### 【描述】

销毁组。

### 【语法】

```
HI_S32 HI_MPI_DPU_RECT_DestroyGrp(DPU_RECT_GRP DpuRectGrp);
```

### 【参数】

参数名称	描述	输入/输出
DpuRectGrp	组号。 取值范围：[0, 8)。	输入

### 【返回值】

返回值	描述
0	成功。
非 0	失败，参见 <a href="#">错误码</a> 。

### 【需求】

- 头文件：hi\_comm\_dpu\_rect.h、mpi\_dpu\_rect.h
- 库文件：libdpu\_rect.a

### 【注意】

- 组必须已创建。
- 调用此接口之前，必须先调用 [HI\\_MPI\\_DPU\\_RECT\\_StopGrp](#) 禁用此组。
- 调用此接口时，会一直等待此组当前任务处理结束才会真正销毁。

### 【举例】

无。

### 【相关主题】

- [HI\\_MPI\\_DPU\\_RECT\\_CreateGrp](#)
- [HI\\_MPI\\_DPU\\_RECT\\_SetGrpAttr](#)
- [HI\\_MPI\\_DPU\\_RECT\\_GetGrpAttr](#)
- [HI\\_MPI\\_DPU\\_RECT\\_StartGrp](#)
- [HI\\_MPI\\_DPU\\_RECT\\_StopGrp](#)



## HI\_MPI\_DPU\_RECT\_SetGrpAttr

### 【描述】

设置组属性。

### 【语法】

```
HI_S32 HI_MPI_DPU_RECT_SetGrpAttr(DPU_RECT_GRP DpuRectGrp, const  
DPU_RECT_GRP_ATTR_S *pstGrpAttr);
```

### 【参数】

参数名称	描述	输入/输出
DpuRectGrp	组号。 取值范围：[0, 8)。	输入
pstGrpAttr	组属性。 不能为空。	输入

### 【返回值】

返回值	描述
0	成功。
非 0	失败，参见 <a href="#">错误码</a> 。

### 【需求】

- 头文件：hi\_comm\_dpu\_rect.h、mpi\_dpu\_rect.h
- 库文件：libdpu\_rect.a

### 【注意】

- 组必须已创建。
- 组属性必须合法，其中部分静态属性不可动态设置，具体请参见[DPU\\_RECT\\_GRP\\_ATTR\\_S](#)。

### 【举例】

无。

### 【相关主题】

- [HI\\_MPI\\_DPU\\_RECT\\_CreateGrp](#)
- [HI\\_MPI\\_DPU\\_RECT\\_DestroyGrp](#)
- [HI\\_MPI\\_DPU\\_RECT\\_GetGrpAttr](#)
- [HI\\_MPI\\_DPU\\_RECT\\_StartGrp](#)



- [HI\\_MPI\\_DPU\\_RECT\\_StopGrp](#)

## HI\_MPI\_DPU\_RECT\_GetGrpAttr

### 【描述】

获取组属性。

### 【语法】

```
HI_S32 HI_MPI_DPU_RECT_GetGrpAttr(DPU_RECT_GRP DpuRectGrp,  
DPU_RECT_GRP_ATTR_S *pstGrpAttr);
```

### 【参数】

参数名称	描述	输入/输出
DpuRectGrp	组号。 取值范围：[0, 8)。	输入
pstGrpAttr	组属性。 不能为空。	输出

### 【返回值】

返回值	描述
0	成功。
非 0	失败，参见 <a href="#">错误码</a> 。

### 【需求】

- 头文件：hi\_comm\_dpu\_rect.h、mpi\_dpu\_rect.h
- 库文件：libdpu\_rect.a

### 【注意】

组必须已创建。

### 【举例】

无。

### 【相关主题】

- [HI\\_MPI\\_DPU\\_RECT\\_CreateGrp](#)
- [HI\\_MPI\\_DPU\\_RECT\\_DestroyGrp](#)
- [HI\\_MPI\\_DPU\\_RECT\\_SetGrpAttr](#)
- [HI\\_MPI\\_DPU\\_RECT\\_StartGrp](#)
- [HI\\_MPI\\_DPU\\_RECT\\_StopGrp](#)





## HI\_MPI\_DPU\_RECT\_StartGrp

### 【描述】

启用组。

### 【语法】

```
HI_S32 HI_MPI_DPU_RECT_StartGrp(DPU_RECT_GRP DpuRectGrp);
```

### 【参数】

参数名称	描述	输入/输出
DpuRectGrp	组号。 取值范围：[0, 8)。	输入

### 【返回值】

返回值	描述
0	成功。
非 0	失败，参见 <a href="#">错误码</a> 。

### 【需求】

- 头文件：hi\_comm\_dpu\_rect.h、mpi\_dpu\_rect.h
- 库文件：libdpu\_rect.a

### 【注意】

- 组必须已创建。
- 重复调用该函数设置同一个组返回成功。

### 【举例】

无。

### 【相关主题】

- [HI\\_MPI\\_DPU\\_RECT\\_CreateGrp](#)
- [HI\\_MPI\\_DPU\\_RECT\\_DestroyGrp](#)
- [HI\\_MPI\\_DPU\\_RECT\\_SetGrpAttr](#)
- [HI\\_MPI\\_DPU\\_RECT\\_GetGrpAttr](#)
- [HI\\_MPI\\_DPU\\_RECT\\_StopGrp](#)

## HI\_MPI\_DPU\_RECT\_StopGrp

### 【描述】



禁用组。

#### 【语法】

```
HI_S32 HI_MPI_DPU_RECT_StopGrp(DPU_RECT_GRP DpuRectGrp);
```

#### 【参数】

参数名称	描述	输入/输出
DpuRectGrp	组号。 取值范围：[0, 8)。	输入

#### 【返回值】

返回值	描述
0	成功。
非 0	失败，参见 <a href="#">错误码</a> 。

#### 【需求】

- 头文件：hi\_comm\_dpu\_rect.h、mpi\_dpu\_rect.h
- 库文件：libdpu\_rect.a

#### 【注意】

- 组必须已创建。
- 重复禁用同一组返回成功。

#### 【举例】

无。

#### 【相关主题】

- [HI\\_MPI\\_DPU\\_RECT\\_CreateGrp](#)
- [HI\\_MPI\\_DPU\\_RECT\\_DestroyGrp](#)
- [HI\\_MPI\\_DPU\\_RECT\\_SetGrpAttr](#)
- [HI\\_MPI\\_DPU\\_RECT\\_GetGrpAttr](#)
- [HI\\_MPI\\_DPU\\_RECT\\_StartGrp](#)

## HI\_MPI\_DPU\_RECT\_SetChnAttr

#### 【描述】

设置通道属性。

#### 【语法】



```
HI_S32 HI_MPI_DPU_RECT_SetChnAttr(DPU_RECT_GRP DpuRectGrp, DPU_RECT_CHN  
DpuRectChn, const DPU_RECT_CHN_ATTR_S *pstChnAttr);
```

#### 【参数】

参数名称	描述	输入/输出
DpuRectGrp	组号。 取值范围：[0, 8)。	输入
DpuRectChn	通道号。 取值范围：[0,2)。	输入
pstChnAttr	通道属性。 不能为空。	输入

#### 【返回值】

返回值	描述
0	成功。
非 0	失败，参见 <a href="#">错误码</a> 。

#### 【需求】

- 头文件：hi\_comm\_dpu\_rect.h、mpi\_dpu\_rect.h
- 库文件：libdpu\_rect.a

#### 【注意】

组必须已创建。

#### 【举例】

无。

#### 【相关主题】

- [HI\\_MPI\\_DPU\\_RECT\\_GetChnAttr](#)
- [HI\\_MPI\\_DPU\\_RECT\\_EnableChn](#)
- [HI\\_MPI\\_DPU\\_RECT\\_DisableChn](#)

## HI\_MPI\_DPU\_RECT\_GetChnAttr

#### 【描述】

获取通道属性。

#### 【语法】



```
HI_S32 HI_MPI_DPU_RECT_GetChnAttr(DPU_RECT_GRP DpuRectGrp, DPU_RECT_CHN  
DpuRectChn, DPU_RECT_CHN_ATTR_S *pstChnAttr);
```

#### 【参数】

参数名称	描述	输入/输出
DpuRectGrp	组号。 取值范围：[0, 8)。	输入
DpuRectChn	通道号。 取值范围：[0,2)。	输入
pstChnAttr	通道属性。 不能为空。	输出

#### 【返回值】

返回值	描述
0	成功。
非 0	失败，参见 <a href="#">错误码</a> 。

#### 【需求】

- 头文件：hi\_comm\_dpu\_rect.h、mpi\_dpu\_rect.h
- 库文件：libdpu\_rect.a

#### 【注意】

组必须已创建。

#### 【举例】

无。

#### 【相关主题】

- [HI\\_MPI\\_DPU\\_RECT\\_SetChnAttr](#)
- [HI\\_MPI\\_DPU\\_RECT\\_EnableChn](#)
- [HI\\_MPI\\_DPU\\_RECT\\_DisableChn](#)

## HI\_MPI\_DPU\_RECT\_EnableChn

#### 【描述】

启用通道。

#### 【语法】



```
HI_S32 HI_MPI_DPU_RECT_EnableChn(DPU_RECT_GRP DpuRectGrp, DPU_RECT_CHN  
DpuRectChn);
```

#### 【参数】

参数名称	描述	输入/输出
DpuRectGrp	组号。 取值范围：[0, 8)。	输入
DpuRectChn	通道号。 取值范围：[0,2)。	输入

#### 【返回值】

返回值	描述
0	成功。
非 0	失败，参见 <a href="#">错误码</a> 。

#### 【需求】

- 头文件：hi\_comm\_dpu\_rect.h、mpi\_dpu\_rect.h
- 库文件：libdpu\_rect.a

#### 【注意】

- 组必须已创建。
- 重复使能返回成功。

#### 【举例】

无。

#### 【相关主题】

- [HI\\_MPI\\_DPU\\_RECT\\_SetChnAttr](#)
- [HI\\_MPI\\_DPU\\_RECT\\_GetChnAttr](#)
- [HI\\_MPI\\_DPU\\_RECT\\_DisableChn](#)

## HI\_MPI\_DPU\_RECT\_DisableChn

#### 【描述】

禁用通道。

#### 【语法】

```
HI_S32 HI_MPI_DPU_RECT_DisableChn(DPU_RECT_GRP DpuRectGrp, DPU_RECT_CHN  
DpuRectChn);
```



#### 【参数】

参数名称	描述	输入/输出
DpuRectGrp	组号。 取值范围：[0, 8)。	输入
DpuRectChn	通道号。 取值范围：[0,2)。	输入

#### 【返回值】

返回值	描述
0	成功。
非 0	失败，参见 <a href="#">错误码</a> 。

#### 【需求】

- 头文件：hi\_comm\_dpu\_rect.h、mpi\_dpu\_rect.h
- 库文件：libdpu\_rect.a

#### 【注意】

- 组必须已创建。
- 重复禁用返回成功。

#### 【举例】

无。

#### 【相关主题】

- [HI\\_MPI\\_DPU\\_RECT\\_SetChnAttr](#)
- [HI\\_MPI\\_DPU\\_RECT\\_GetChnAttr](#)
- [HI\\_MPI\\_DPU\\_RECT\\_EnableChn](#)

## HI\_MPI\_DPU\_RECT\_SendFrame

#### 【描述】

用户发送数据。

#### 【语法】

```
HI_S32 HI_MPI_DPU_RECT_SendFrame(DPU_RECT_GRP DpuRectGrp, const  
VIDEO_FRAME_INFO_S *pstLeftFrame, const VIDEO_FRAME_INFO_S *pstRightFrame,  
HI_S32 s32MilliSec);
```



## 【参数】

参数名称	描述	输入/输出
DpuRectGrp	组号。 取值范围：[0, 8)。	输入
pstLeftFrame	左图图像。 不能为空。	输入
pstRightFrame	右图图像。 当组属性设置 DPU_RECT_MODE_DOUBLE 时，不能为空。	输入
s32MilliSec	超时参数 s32MilliSec 设为-1 时，为阻塞接口；0 时为非阻塞接口；大于 0 时为超时等待时间，超时的单位为毫秒（ms）。	输入

参数名称	支持类型	地址对齐	分辨率
pstLeftFrame	PIXEL_FORMAT_YVU_SEMIPLANAR_420/PIXEL_FORMAT_YVU_SEMIPLANAR_422/PIXEL_FORMAT_YUV_400	16 byte	128x64~2048x2048
pstRightFrame	PIXEL_FORMAT_YVU_SEMIPLANAR_420/PIXEL_FORMAT_YVU_SEMIPLANAR_422/PIXEL_FORMAT_YUV_400	16 byte	128x64~2048x2048

## 【返回值】

返回值	描述
0	成功。
非 0	失败，参见 <a href="#">错误码</a> 。

## 【需求】

- 头文件：hi\_comm\_dpu\_rect.h、mpi\_dpu\_rect.h
- 库文件：libdpu\_rect.a

## 【注意】

- 组必须已创建。
- 用户使用此接口时，可以自行进行帧率控制。



- pstLeftFrame/ pstRightFrame 图像地址必须是 VB 申请的，stride 要求 16 字节对齐。

## 【举例】

无。

## 【相关主题】

- [HI\\_MPI\\_DPU\\_RECT\\_GetFrame](#)
- [HI\\_MPI\\_DPU\\_RECT\\_ReleaseFrame](#)

## HI\_MPI\_DPU\_RECT\_GetFrame

## 【描述】

用户从通道获取一帧处理完成的图像。

## 【语法】

```
HI_S32 HI_MPI_DPU_RECT_GetFrame(DPU_RECT_GRP DpuRectGrp,
VIDEO_FRAME_INFO_S *pstSrcLeftFrame, VIDEO_FRAME_INFO_S
*pstSrcRightFrame, VIDEO_FRAME_INFO_S *pstDstLeftFrame, VIDEO_FRAME_INFO_S
*pstDstRightFrame, HI_S32 s32MilliSec);
```

## 【参数】

参数名称	描述	输入/输出
DpuRectGrp	组号。 取值范围：[0, 8)。	输入
pstSrcLeftFrame	左图原始图像。 当组属性的 bNeedSrcFrame 设置为 HI_TRUE 时，不能为空。	输出
pstSrcRightFrame	右图原始图像。 当组属性设置 DPU_RECT_MODE_DOUBLE 且组属性的 bNeedSrcFrame 设置为 HI_TRUE 时，不能为空。	输出
pstDstLeftFrame	校正后左图图像。 不能为空。	输出
pstDstRightFrame	校正后右图图像。 当组属性设置 DPU_RECT_MODE_DOUBLE 时，不能为空。	输出
s32MilliSec	超时参数 s32MilliSec 设为-1 时，为阻塞接口；0 时为非阻塞接口；大于 0 时为超时等待时间，超时时间的单位为毫秒（ms）。	输入





参数名称	支持类型	地址对齐	分辨率
pstSrcLeftFrame	PIXEL_FORMAT_YVU_SEMIPLANAR_420/PIXEL_FORMAT_YVU_SEMIPLANAR_422/PIXEL_FORMAT_YUV_400	16 byte	128x64~2048x2048
pstSrcRightFrame	PIXEL_FORMAT_YVU_SEMIPLANAR_420/PIXEL_FORMAT_YVU_SEMIPLANAR_422/PIXEL_FORMAT_YUV_400	16 byte	128x64~2048x2048
pstDstLeftFrame	PIXEL_FORMAT_YUV_400	16 byte	128x64~1920x1080
pstDstRightFrame	PIXEL_FORMAT_YUV_400	16 byte	128x64~1920x1080

**【返回值】**

返回值	描述
0	成功。
非 0	失败，参见 <a href="#">错误码</a> 。

**【需求】**

- 头文件：hi\_comm\_dpu\_rect.h、mpi\_dpu\_rect.h
- 库文件：libdpu\_rect.a

**【注意】**

- 组必须已创建。
- 只有设置组属性的队列深度不为 0，才能获取到图像。
- 若是用户发送数据模式，pstSrcLeftFrame 和 pstSrcRightFrame 数据信息由用户保证，若是绑定模式，由系统保证。
- pstDstLeftFrame/ pstDstRightFrame 数据信息由系统保证。

**【举例】**

无。

**【相关主题】**

- [HI\\_MPI\\_DPU\\_RECT\\_SendFrame](#)
- [HI\\_MPI\\_DPU\\_RECT\\_ReleaseFrame](#)

## HI\_MPI\_DPU\_RECT\_ReleaseFrame

**【描述】**



用户释放一帧通道图像。

#### 【语法】

```
HI_S32 HI_MPI_DPU_RECT_ReleaseFrame(DPU_RECT_GRP DpuRectGrp, const  
VIDEO_FRAME_INFO_S *pstSrcLeftFrame, const VIDEO_FRAME_INFO_S  
*pstSrcRightFrame, const VIDEO_FRAME_INFO_S *pstDstLeftFrame, const  
VIDEO_FRAME_INFO_S *pstDstRightFrame);
```

#### 【参数】

参数名称	描述	输入/输出
DpuRectGrp	组号。 取值范围：[0, 8)。	输入
pstSrcLeftFrame	左图原始图像。 当设置组属性的 bNeedSrcFrame 为 HI_TRUE 时，不能为空。	输入
pstSrcRightFrame	右图原始图像。 当组属性设置 DPU_RECT_MODE_DOUBLE 且组属性的 bNeedSrcFrame 设置为 HI_TRUE 时，不能为空。	输入
pstDstLeftFrame	校正后左图图像。 不能为空。	输入
pstDstRightFrame	校正后右图图像。 当组属性设置 DPU_RECT_MODE_DOUBLE 时，不能为空。	输入

#### 【返回值】

返回值	描述
0	成功。
非 0	失败，参见 <a href="#">错误码</a> 。

#### 【需求】

- 头文件：hi\_comm\_dpu\_rect.h、mpi\_dpu\_rect.h
- 库文件：libdpu\_rect.a

#### 【注意】

- 组必须已创建。



- pstSrcLeftFrame/ pstSrcRightFrame/ pstDstLeftFrame/ pstDstRightFrame 由 [HI\\_MPI\\_DPU\\_RECT\\_GetFrame](#) 获取，与其配对使用。

【举例】

无。

【相关主题】

- [HI\\_MPI\\_DPU\\_RECT\\_SendFrame](#)
- [HI\\_MPI\\_DPU\\_RECT\\_GetFrame](#)

## 1.4 数据类型和数据结构

DPU\_RECT 相关数据类型、数据结构定义如下：

- [DPU\\_RECT\\_LUT\\_ID](#)：定义查找表 ID。
- [DPU\\_RECT\\_GRP](#)：定义组号。
- [DPU\\_RECT\\_CHN](#)：定义通道号。
- [DPU\\_RECT\\_MODE\\_E](#)：定义校正模式。
- [DPU\\_RECT\\_GRP\\_ATTR\\_S](#)：定义组属性。
- [DPU\\_RECT\\_CHN\\_ATTR\\_S](#)：定义通道属性。
- [DPU\\_RECT\\_MEM\\_INFO\\_S](#)：定义一维内存信息。

### DPU\_RECT\_LUT\_ID

【说明】

定义查找表 ID。

【定义】

```
typedef HI_S32 DPU_RECT_LUT_ID;
```

【成员】

无。

【注意事项】

无。

【相关数据类型及接口】

无。

### DPU\_RECT\_GRP

【说明】

定义组号。



**【定义】**

```
typedef HI_S32 DPU_RECT_GRP;
```

**【成员】**

无。

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

## DPU\_RECT\_CHN

**【说明】**

定义通道号。

**【定义】**

```
typedef HI_S32 DPU_RECT_CHN;
```

**【成员】**

无。

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

## DPU\_RECT\_MODE\_E

**【说明】**

定义校正模式。

**【定义】**

```
typedef enum hiDPU_RECT_MODE_E
{
    DPU_RECT_MODE_SINGLE = 0x0, /* only channel 0 work */
    DPU_RECT_MODE_DOUBLE = 0x1, /* two channel work */
    DPU_RECT_MODE_BUTT
}DPU_RECT_MODE_E;
```

**【成员】**



成员名称	描述
DPU_RECT_MODE_SINGLE	单通道校正。
DPU_RECT_MODE_DOUBLE	双通道校正。

【注意事项】

无。

【相关数据类型及接口】

无。

## DPU\_RECT\_GRP\_ATTR\_S

【说明】

定义组属性。

【定义】

```
typedef struct hiDPU_RECT_GRP_ATTR_S
{
    DPU_RECT_MODE_E enRectMode; /* Rectification mode, it can not be
    changed dynamic */
    SIZE_S stLeftImageSize; /* Left image size. */
    SIZE_S stRightImageSize; /* Right image size. */
    DPU_RECT_LUT_ID LeftLutId; /* Left image rectification lut */
    DPU_RECT_LUT_ID RightLutId; /* Right image rectification lut */
    HI_U32 u32Depth; /* The depth of user image queue for getting
    Rectification output image, it can not be changed dynamic. Range:[0,8]
    */
    HI_BOOL bNeedSrcFrame; /* The flag of getting source videoframe.*/
    FRAME_RATE_CTRL_S stFrameRate; /* Grp frame rate contrl. */
}DPU_RECT_GRP_ATTR_S;
```

【成员】

成员名称	描述
enRectMode	校正模式，静态属性，不可动态更改。
stLeftImageSize	左图分辨率。取值范围：宽[128, 2048]; 高[64, 2048]。
stRightImageSize	右图分辨率。取值范围：宽[128, 2048]; 高[64, 2048]。
LeftLutId	左图查找表 ID。有效范围: [0, 16)。
RightLutId	右图查找表 ID。有效范围: [0, 16)。



成员名称	描述
u32Depth	通道图像的队列长度，静态属性，不可动态更改。取值范围: [0, 8]。
bNeedSrcFrame	是否需要获取原始图像帧标志。
stFrameRate	组帧率。

**【注意事项】**

分辨率宽、高要求偶数对齐。

**【相关数据类型及接口】**

无。

## DPU\_RECT\_CHN\_ATTR\_S

**【说明】**

定义通道属性。

**【定义】**

```
typedef struct hiDPU_RECT_CHN_ATTR_S
{
    SIZE_S stImageSize; /* Rectify output image size */
}DPU_RECT_CHN_ATTR_S;
```

**【成员】**

成员名称	描述
stImageSize	目标分辨率。取值范围: 宽[128, 1920]; 高[64, 1080]。

**【注意事项】**

- 分辨率宽、高要求偶数对齐。
- 目标分辨率与查找表分辨率一致。

**【相关数据类型及接口】**

无。

## DPU\_RECT\_MEM\_INFO\_S

**【说明】**

定义一维内存信息。

**【定义】**



```
typedef struct hiDPU_RECT_MEM_INFO_S
{
    HI_U64  u64PhyAddr;
    HI_U64  u64VirAddr;
    HI_U32  u32Size;
}DPU_RECT_MEM_INFO_S;
```

**【成员】**

成员名称	描述
u64PhyAddr	内存块物理地址。
u64VirAddr	内存块虚拟地址。
u32Size	内存块字节数，见图 1-1。

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

## 1.5 错误码

本模块 API 错误码如表 1-1 所示。

表1-1 DPU\_RECT 模块 API 错误码

错误代码	宏定义	描述
0xA0358001	HI_ERR_DPU_RECT_INVALID_DEVID	设备 ID 超出合法范围
0xA0358002	HI_ERR_DPU_RECT_INVALID_CHNID	通道组号错误或无效区域句柄
0xA0358003	HI_ERR_DPU_RECT_ILLEGAL_PARAM	参数超出合法范围
0xA0358004	HI_ERR_DPU_RECT_EXIST	重复创建已存在的设备、通道或资源
0xA0358005	HI_ERR_DPU_RECT_UNEXIST	试图使用或者销毁不存在的设备、通道或者资源
0xA0358006	HI_ERR_DPU_RECT_NULL_PTR	函数参数中有空指针
0xA0358007	HI_ERR_DPU_RECT_NOT_CONFIG	模块没有配置



错误代码	宏定义	描述
0xA0358008	HI_ERR_DPU_RECT_NOT_SUPPORT	不支持的参数或者功能
0xA0358009	HI_ERR_DPU_RECT_NOT_PERM	该操作不允许，如试图修改静态配置参数
0xA035800C	HI_ERR_DPU_RECT_NOMEM	分配内存失败，如系统内存不足
0xA035800D	HI_ERR_DPU_RECT_NOBUF	分配缓存失败，如申请的图像缓冲区太大
0xA035800E	HI_ERR_DPU_RECT_BUF_EMPTY	缓冲区中无图像
0xA035800F	HI_ERR_DPU_RECT_BUF_FULL	缓冲区中图像满
0xA0358010	HI_ERR_DPU_RECT_NOTREADY	系统没有初始化或没有加载相应模块
0xA0358011	HI_ERR_DPU_RECT_BADADDR	地址非法
0xA0358012	HI_ERR_DPU_RECT_BUSY	系统忙
0xA0358040	HI_ERR_DPU_RECT_SYS_TIMEOUT	系统超时
0xA0358041	HI_ERR_DPU_RECT_OPEN_FILE	打开文件失败
0xA0358042	HI_ERR_DPU_RECT_READ_FILE	读文件失败
0xA0358043	HI_ERR_DPU_RECT_WRITE_FILE	写文件失败

## 1.6 Proc 调试信息

### 1.6.1 概述

调试信息采用了 Linux 下的 proc 文件系统，可实时反映当前系统的运行状态，所记录的信息可供问题定位及分析时使用。

#### 【文件目录】

/proc/umap

#### 【信息查看方法】

- 在控制台上可以使用 cat 命令查看信息，cat /proc/umap/rect；也可以使用其他常用的文件操作命令，例如 cp /proc/umap/rect ./，将文件拷贝到当前目录。
- 在应用程序中可以将上述文件当作普通只读文件进行读操作，例如 fopen、fread 等。





## 说明

参数在描述时有以下 2 种情况需要注意：

- 取值为{0, 1}的参数，如未列出具体取值和含义的对应关系，则参数为 1 时表示肯定，为 0 时表示否定。
- 取值为{aaa, bbb, ccc}的参数，未列出具体取值和含义的对应关系，但可直接根据取值 aaa、bbb 或 ccc 判断参数含义

## 1.6.2 Proc 信息说明

### 【调试信息】

```
#cat /proc/umap/rect
```

```
[DPU_RECT] Version: [Hi3559AV100_MPP_V1.0.0.0 B010 Release], Build  
Time[Jun 25 2018, 19:57:26]
```

```
-----DPU_RECT HardWare STATUS-----
```

```
Busy      STCnt  
N         0
```

```
-----DPU_RECT GRP ATTR-----
```

```
GrpID  Mode  bStart  GrpState  bNeedSrcFrame  Depth  SrcFrmRate  
DstFrmRate  LeftWidth  LeftHeight  RightWidth  RightHeight  
0       DOUBLE      Y      CREATE      Y      0      -1      -1  
1920    1080    1920    1080
```

```
-----DPU_RECT CHN ATTR-----
```

```
GrpID  ChnID  Enable  Width  Height  
0       0      Y      1920   1080  
0       1      Y      1920   1080
```

```
-----DPU_RECT LUT ATTR-----
```

```
GrpID  ChnID  LutID  PhyAddr  Width  Height  
0       0      0      8a89d000  1920   1080  
0       1      1      8b086000  1920   1080
```

```
-----DPU_RECT PIPE QUEUE-----
```

```
GrpID  PipeID  BusyNum  FreeNum  
0       0      0        8  
0       1      0        8
```

```
-----DPU_RECT OUT QUEUE-----
```

```
GrpID  BusyNum  FreeNum
```

```
-----DPU_RECT WORK QUEUE-----
```

```
BusyNum  FreeNum  
0        64
```



-----DPU\_RECT\_GRP\_STATUS-----

GrpID	FrameRate	StartCnt	StartFailCnt	SendPicCnt	FrmLost
0	30	6815768	0	6815768	0

GrpID	PipeID	InFrmLost
0	0	0
0	1	0

-----DPU\_RECT\_RUN-TIME\_INFO-----

CntPerSec	MaxCntPerSec	TotalIntCntLastSec	TotalIntCnt	CostTm	MCostTm
30	31	6815746	6815768	24	84

CostTmPerSec	MCostTmPerSec	TotalIntCostTm	CostTmPerFrm	HwCostTmPerFrm
1279	260861470	6353	6335	227198

### 【调试信息分析】

记录当前 DPU\_RECT 属性配置以及状态信息。

### 【参数说明】

参数		描述
DPU_RECT HardWare STATUS DPU_RECT 硬件工 作状态信息	Busy	DPU_RECT 硬件是否忙。 Y:硬件忙; N:硬件空闲。
	STCnt	DPU_RECT 系统超时次数。
DPU_RECT_GRP ATTR DPU_RECT 组属性 信息。	GrpID	GRP ID 号。 有效范围: [0, 8)。
	Mode	校正处理模式。 有效范围见枚举 DPU_RECT_MODE_E。
	bStart	组启动状态。 Y:启动; N:未启动。
	GrpState	组工作状态。 CREATE:创建; DESTROYING:销毁。
	bNeedSrcFrame	是否需要获取原始图像标志。 Y: 需要获取原始图像; N: 不需要获取原始图像。



参数		描述
	Depth	输出队列深度。 取值范围: [0, 8]。
	SrcFrmRate	Grp 源帧率
	DstFrmRate	Grp 目标帧率
	LeftWidth	输入左图分辨率的宽。 取值范围: [128, 2048]。
	LeftHeight	输入左图分辨率的高。 取值范围: [64, 2048]。
	RightWidth	输入右图分辨率的宽。 取值范围: [128, 2048]。
	RightHeight	输入右图分辨率的高。 取值范围: [64, 2048]。
DPU_RECT_CHN_ATTR DPU_RECT 通道属性信息	GrpID	GRP ID 号。 有效范围: [0, 8)。
	ChnID	CHN ID 号。 有效范围: [0, 2)。
	Enable	通道使能标志。 Y:使能; N:未使能。
	Width	通道输出的目标图像分辨率的宽。 取值范围: [128, 1920]。
	Height	通道输出的目标图像分辨率的高。 取值范围: [64, 1080]。
DPU_RECT_LUT_ATTR DPU_RECT 查找表属性	GrpID	GRP ID 号。 有效范围: [0, 8)。
	ChnID	CHN ID 号。 有效范围: [0, 2)。
	LutID	查找表 ID 号。 有效范围: [0, 16)。
	PhyAddr	查找表物理地址。



参数		描述
	Width	查找表分辨率的宽。与通道输出的目标图像分辨率的宽一致。取值范围: [128, 1920]。
	Height	查找表分辨率的高。与通道输出的目标图像分辨率的高一致。取值范围: [64, 1080]。
DPU_RECT_PIPE_QUEUE DPU_RECT 输入管道队列信息	GrpID	GRP ID 号。 有效范围: [0, 8)。
	PipeID	PIPE ID 号。 有效范围: [0, 2)。
	BusyNum	已用节点数。
	FreeNum	可用节点数。
DPU_RECT_OUT_QUEUE DPU_RECT 输出通道队列信息	GrpID	GRP ID 号。 有效范围: [0, 8)。
	BusyNum	已用节点数。
	FreeNum	可用节点数。
DPU_RECT_WORK_QUEUE DPU_RECT 工作队列信息	BusyNum	已用节点数。
	FreeNum	可用节点数。
DPU_RECT_GRP_STATUS DPU_RECT 组状态信息	GrpID	GRP ID 号。 有效范围: [0, 8)。
	FrameRate	输出帧率。
	StartCnt	任务启动次数。
	StartFailCnt	任务启动失败次数。
	SendPicCnt	发图像成功次数。
	FrmLost	因队列满丢失的输出图像数
	CurTaskCostTm	当前完成任务的耗时。
	MaxTaskCostTm	历史上耗时最长任务的执行时间。
	GrpID	GRP ID 号。 有效范围: [0, 8)。



参数		描述
	PipeID	PIPE ID 号。 有效范围: [0, 2)。
	InFrmLost	因队列满丢失的输入图像数。
DPU_RECT RUN-TIME INFO DPU_RECT 运行时相关信息	CntPerSec	最近一次的 1 秒内中断执行次数。
	MaxCntPerSec	历史上的 1 秒内最大的中断执行次数。
	TotalIntCntLastSec	上一秒上报中断总次数。
	TotalIntCnt	DPU_RECT 产生中断的总次数。
	CostTm	最近一次执行中断的执行耗时。 单位: us
	MCostTm	执行一次中断的最大耗时。 单位: us
	CostTmPerSec	最近一秒执行中断的执行耗时。 单位: us
	MCostTmPerSec	历史上一秒执行中断的最大执行耗时。 单位: us
	TotalIntCostTm	中断处理总时间。 单位: us
	CostTmPerFrm	单帧图像处理的总时间。 单位: us
	HwCostTmPerFrm	单帧图像处理的硬件时间。 单位: us
	RunTm	DPU_RECT 运行总时间。 单位: s



# 2 DPU\_MATCH

## 2.1 概述

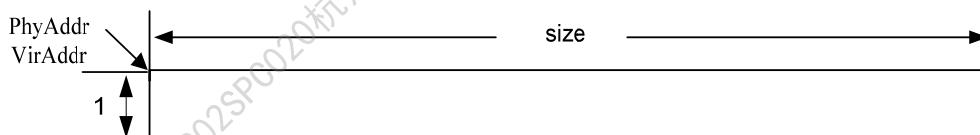
匹配 (DPU\_MATCH) 对输入的左图像和右图像进行匹配。

## 2.2 功能描述

### 2.2.1 重要概念

- 组(GROUP)  
DPU\_MATCH 对用户提供的组 (GROUP) 的概念。最大可用数为 8 个，各组分时复用 DPU\_MATCH 硬件。
- 通道(CHANNEL )  
DPU\_MATCH 组的通道。每组包含单个通道。
- 管道(PIPE)  
DPU\_MATCH 组的管道。每组包含两个管道，分为左右两管道
- [DPU\\_MATCH\\_MEM\\_INFO\\_S](#) 内存排布

图2-1 DPU\_MATCH\_MEM\_INFO\_S 数据内存示意

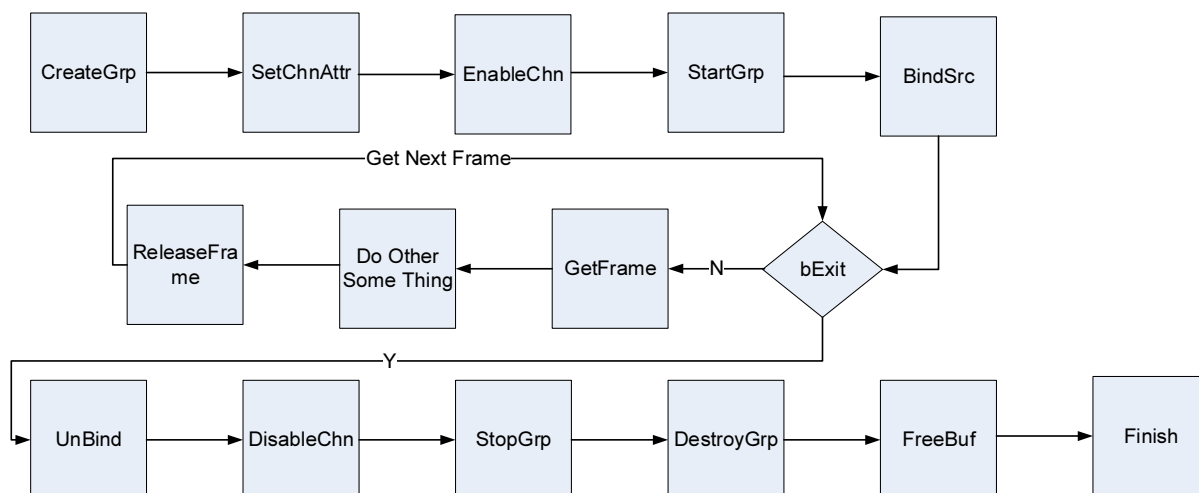


### 2.2.2 调用流程

目前无目标模块可以绑定 DPU\_MATCH,接收 DPU\_MATCH 进行处理，必须手动的调用 [HI\\_MPI\\_DPU\\_MATCH\\_GetFrame](#) 获取结果，再进行处理。如图 2-2 所示。



图2-2 DPU\_MATCH 调用流程



## 2.3 API 参考

该功能模块为用户提供以下 MPI：

- [HI\\_MPI\\_DPU\\_MATCH\\_GetAssistBufSize](#)：获取辅助内存字节数。
- [HI\\_MPI\\_DPU\\_MATCH\\_CreateGrp](#)：创建组。
- [HI\\_MPI\\_DPU\\_MATCH\\_DestroyGrp](#)：销毁组。
- [HI\\_MPI\\_DPU\\_MATCH\\_SetGrpAttr](#)：设置组属性。
- [HI\\_MPI\\_DPU\\_MATCH\\_GetGrpAttr](#)：获取组属性。
- [HI\\_MPI\\_DPU\\_MATCH\\_StartGrp](#)：启用组。
- [HI\\_MPI\\_DPU\\_MATCH\\_StopGrp](#)：禁用组。
- [HI\\_MPI\\_DPU\\_MATCH\\_SetChnAttr](#)：设置通道属性。
- [HI\\_MPI\\_DPU\\_MATCH\\_GetChnAttr](#)：获取通道属性。
- [HI\\_MPI\\_DPU\\_MATCH\\_EnableChn](#)：启用通道。
- [HI\\_MPI\\_DPU\\_MATCH\\_DisableChn](#)：禁用通道。
- [HI\\_MPI\\_DPU\\_MATCH\\_SendFrame](#)：用户发送数据。
- [HI\\_MPI\\_DPU\\_MATCH\\_GetFrame](#)：用户从通道获取一帧处理完成的图像。
- [HI\\_MPI\\_DPU\\_MATCH\\_ReleaseFrame](#)：用户释放一帧通道图像。

### HI\_MPI\_DPU\_MATCH\_GetAssistBufSize

#### 【描述】

获取辅助内存字节数。

#### 【语法】

```
HI_S32 HI_MPI_DPU_MATCH_GetAssistBufSize(HI_U16 u16DispNum, HI_U32
```



```
u32DstHeight, HI_U32 *pu32Size);
```

#### 【参数】

参数名称	描述	输入/输出
u16DispNum	视差值数目。 取值范围: [16, 256]。	输入
u32DstHeight	输出的图像高。	输入
pu32Size	辅助内存字节数。 不能为空。	输出

#### 【返回值】

返回值	描述
0	成功。
非 0	失败, 参见 <a href="#">错误码</a> 。

#### 【需求】

- 头文件: hi\_comm\_dpu\_match.h、mpi\_dpu\_match.h
- 库文件: libdpu\_match.a

#### 【注意】

无。

#### 【举例】

无。

#### 【相关主题】

[HI\\_MPI\\_DPU\\_MATCH\\_CreateGrp](#)

## HI\_MPI\_DPU\_MATCH\_CreateGrp

#### 【描述】

创建组。

#### 【语法】

```
HI_S32 HI_MPI_DPU_MATCH_CreateGrp(DPU_MATCH_GRP DpuMatchGrp, const  
DPU_MATCH_GRP_ATTR_S *pstGrpAttr);
```

#### 【参数】





参数名称	描述	输入/输出
DpuMatchGrp	组号。 取值范围：[0, 8)。	输入
pstGrpAttr	组属性。 不能为空。	输入

#### 【返回值】

返回值	描述
0	成功。
非 0	失败，参见 <a href="#">错误码</a> 。

#### 【需求】

- 头文件：hi\_comm\_dpu\_match.h、mpi\_dpu\_match.h
- 库文件：libdpu\_match.a

#### 【注意】

不支持重复创建。

#### 【举例】

无。

#### 【相关主题】

- [HI\\_MPI\\_DPU\\_MATCH\\_DestroyGrp](#)
- [HI\\_MPI\\_DPU\\_MATCH\\_SetGrpAttr](#)
- [HI\\_MPI\\_DPU\\_MATCH\\_GetGrpAttr](#)
- [HI\\_MPI\\_DPU\\_MATCH\\_StartGrp](#)
- [HI\\_MPI\\_DPU\\_MATCH\\_StopGrp](#)

## HI\_MPI\_DPU\_MATCH\_DestroyGrp

#### 【描述】

销毁组。

#### 【语法】

```
HI_S32 HI_MPI_DPU_MATCH_DestroyGrp(DPU_MATCH_GRP DpuMatchGrp);
```

#### 【参数】



参数名称	描述	输入/输出
DpuMatchGrp	组号。 取值范围：[0, 8)。	输入

#### 【返回值】

返回值	描述
0	成功。
非 0	失败，参见 <a href="#">错误码</a> 。

#### 【需求】

- 头文件：hi\_comm\_dpu\_match.h、mpi\_dpu\_match.h
- 库文件：libdpu\_match.a

#### 【注意】

- 组必须已创建。
- 调用此接口之前，必须先调用 [HI\\_MPI\\_DPU\\_MATCH\\_StopGrp](#) 禁用此组。
- 调用此接口时，会一直等待此组当前任务处理结束才会真正销毁。

#### 【举例】

无。

#### 【相关主题】

- [HI\\_MPI\\_DPU\\_MATCH\\_CreateGrp](#)
- [HI\\_MPI\\_DPU\\_MATCH\\_SetGrpAttr](#)
- [HI\\_MPI\\_DPU\\_MATCH\\_GetGrpAttr](#)
- [HI\\_MPI\\_DPU\\_MATCH\\_StartGrp](#)
- [HI\\_MPI\\_DPU\\_MATCH\\_StopGrp](#)

## HI\_MPI\_DPU\_MATCH\_SetGrpAttr

#### 【描述】

设置组属性。

#### 【语法】

```
HI_S32 HI_MPI_DPU_MATCH_SetGrpAttr(DPU_MATCH_GRP DpuMatchGrp, const  
DPU_MATCH_GRP_ATTR_S *pstGrpAttr);
```

#### 【参数】



参数名称	描述	输入/输出
DpuMatchGrp	组号。 取值范围：[0, 8)。	输入
pstGrpAttr	组属性。 不能为空。	输入

#### 【返回值】

返回值	描述
0	成功。
非 0	失败，参见 <a href="#">错误码</a> 。

#### 【需求】

- 头文件：hi\_comm\_dpu\_match.h、mpi\_dpu\_match.h
- 库文件：libdpu\_match.a

#### 【注意】

- 组必须已创建。
- 组属性必须合法，其中部分静态属性不可动态设置，具体请参见[DPU\\_MATCH\\_GRP\\_ATTR\\_S](#)。

#### 【举例】

无。

#### 【相关主题】

- [HI\\_MPI\\_DPU\\_MATCH\\_CreateGrp](#)
- [HI\\_MPI\\_DPU\\_MATCH\\_DestroyGrp](#)
- [HI\\_MPI\\_DPU\\_MATCH\\_GetGrpAttr](#)
- [HI\\_MPI\\_DPU\\_MATCH\\_StartGrp](#)
- [HI\\_MPI\\_DPU\\_MATCH\\_StopGrp](#)

## HI\_MPI\_DPU\_MATCH\_GetGrpAttr

#### 【描述】

获取组属性。

#### 【语法】

```
HI_S32 HI_MPI_DPU_MATCH_GetGrpAttr(DPU\_MATCH\_GRP DpuMatchGrp,  
DPU\_MATCH\_GRP\_ATTR\_S *pstGrpAttr);
```



#### 【参数】

参数名称	描述	输入/输出
DpuMatchGrp	组号。 取值范围：[0, 8)。	输入
pstGrpAttr	组属性。 不能为空。	输出

#### 【返回值】

返回值	描述
0	成功。
非 0	失败，参见 <a href="#">错误码</a> 。

#### 【需求】

- 头文件：hi\_comm\_dpu\_match.h、mpi\_dpu\_match.h
- 库文件：libdpu\_match.a

#### 【注意】

组必须已创建。

#### 【举例】

无。

#### 【相关主题】

- [HI\\_MPI\\_DPU\\_MATCH\\_CreateGrp](#)
- [HI\\_MPI\\_DPU\\_MATCH\\_DestroyGrp](#)
- [HI\\_MPI\\_DPU\\_MATCH\\_SetGrpAttr](#)
- [HI\\_MPI\\_DPU\\_MATCH\\_StartGrp](#)
- [HI\\_MPI\\_DPU\\_MATCH\\_StopGrp](#)

## HI\_MPI\_DPU\_MATCH\_StartGrp

#### 【描述】

启用组。

#### 【语法】

```
HI_S32 HI_MPI_DPU_MATCH_StartGrp(DPU_MATCH_GRP DpuMatchGrp);
```

#### 【参数】



参数名称	描述	输入/输出
DpuMatchGrp	组号。 取值范围：[0, 8)。	输入

#### 【返回值】

返回值	描述
0	成功。
非 0	失败，参见 <a href="#">错误码</a> 。

#### 【需求】

- 头文件：hi\_comm\_dpu\_match.h、mpi\_dpu\_match.h
- 库文件：libdpu\_match.a

#### 【注意】

- 组必须已创建。
- 重复调用该函数设置同一个组返回成功。

#### 【举例】

无。

#### 【相关主题】

- [HI\\_MPI\\_DPU\\_MATCH\\_CreateGrp](#)
- [HI\\_MPI\\_DPU\\_MATCH\\_DestroyGrp](#)
- [HI\\_MPI\\_DPU\\_MATCH\\_SetGrpAttr](#)
- [HI\\_MPI\\_DPU\\_MATCH\\_GetGrpAttr](#)
- [HI\\_MPI\\_DPU\\_MATCH\\_StopGrp](#)

## HI\_MPI\_DPU\_MATCH\_StopGrp

#### 【描述】

禁用组。

#### 【语法】

```
HI_S32 HI_MPI_DPU_MATCH_StopGrp(DPU_MATCH_GRP DpuMatchGrp);
```

#### 【参数】



参数名称	描述	输入/输出
DpuMatchGrp	组号。 取值范围：[0, 8)。	输入

#### 【返回值】

返回值	描述
0	成功。
非 0	失败，参见 <a href="#">错误码</a> 。

#### 【需求】

- 头文件：hi\_comm\_dpu\_match.h、mpi\_dpu\_match.h
- 库文件：libdpu\_match.a

#### 【注意】

- 组必须已创建。
- 重复禁用同一组返回成功。

#### 【举例】

无。

#### 【相关主题】

- [HI\\_MPI\\_DPU\\_MATCH\\_CreateGrp](#)
- [HI\\_MPI\\_DPU\\_MATCH\\_DestroyGrp](#)
- [HI\\_MPI\\_DPU\\_MATCH\\_SetGrpAttr](#)
- [HI\\_MPI\\_DPU\\_MATCH\\_GetGrpAttr](#)
- [HI\\_MPI\\_DPU\\_MATCH\\_StartGrp](#)

## HI\_MPI\_DPU\_MATCH\_SetChnAttr

#### 【描述】

设置通道属性。

#### 【语法】

```
HI_S32 HI_MPI_DPU_MATCH_SetChnAttr(DPU_MATCH_GRP DpuMatchGrp,  
DPU_MATCH_CHN DpuMatchChn, const DPU_MATCH_CHN_ATTR_S *pstChnAttr);
```

#### 【参数】



参数名称	描述	输入/输出
DpuMatchGrp	组号。 取值范围：[0, 8)。	输入
DpuMatchChn	通道号。 取值范围：[0,1)。	输入
pstChnAttr	通道属性。 不能为空。	输入

#### 【返回值】

返回值	描述
0	成功。
非 0	失败，参见 <a href="#">错误码</a> 。

#### 【需求】

- 头文件：hi\_comm\_dpu\_match.h、mpi\_dpu\_match.h
- 库文件：libdpu\_match.a

#### 【注意】

组必须已创建。

#### 【举例】

无。

#### 【相关主题】

- [HI\\_MPI\\_DPU\\_MATCH\\_GetChnAttr](#)
- [HI\\_MPI\\_DPU\\_MATCH\\_EnableChn](#)
- [HI\\_MPI\\_DPU\\_MATCH\\_DisableChn](#)

## HI\_MPI\_DPU\_MATCH\_GetChnAttr

#### 【描述】

获取通道属性。

#### 【语法】

```
HI_S32 HI_MPI_DPU_MATCH_GetChnAttr(DPU_MATCH_GRP DpuMatchGrp,  
DPU_MATCH_CHN DpuMatchChn, DPU_MATCH_CHN_ATTR_S *pstChnAttr);
```

#### 【参数】



参数名称	描述	输入/输出
DpuMatchGrp	组号。 取值范围：[0, 8)。	输入
DpuMatchChn	通道号。 取值范围：[0,1)。	输入
pstChnAttr	通道属性。 不能为空。	输出

#### 【返回值】

返回值	描述
0	成功。
非 0	失败，参见 <a href="#">错误码</a> 。

#### 【需求】

- 头文件：hi\_comm\_dpu\_match.h、mpi\_dpu\_match.h
- 库文件：libdpu\_match.a

#### 【注意】

组必须已创建。

#### 【举例】

无。

#### 【相关主题】

- [HI\\_MPI\\_DPU\\_MATCH\\_SetChnAttr](#)
- [HI\\_MPI\\_DPU\\_MATCH\\_EnableChn](#)
- [HI\\_MPI\\_DPU\\_MATCH\\_DisableChn](#)

## HI\_MPI\_DPU\_MATCH\_EnableChn

#### 【描述】

启用通道。

#### 【语法】

```
HI_S32 HI_MPI_DPU_MATCH_EnableChn(DPU_MATCH_GRP DpuMatchGrp,  
DPU_MATCH_CHN DpuMatchChn);
```

#### 【参数】





参数名称	描述	输入/输出
DpuMatchGrp	组号。 取值范围：[0, 8)。	输入
DpuMatchChn	通道号。 取值范围：[0,1)。	输入

#### 【返回值】

返回值	描述
0	成功。
非 0	失败，参见 <a href="#">错误码</a> 。

#### 【需求】

- 头文件：hi\_comm\_dpu\_match.h、mpi\_dpu\_match.h
- 库文件：libdpu\_match.a

#### 【注意】

- 组必须已创建。
- 重复使能返回成功。

#### 【举例】

无。

#### 【相关主题】

- [HI\\_MPI\\_DPU\\_MATCH\\_SetChnAttr](#)
- [HI\\_MPI\\_DPU\\_MATCH\\_GetChnAttr](#)
- [HI\\_MPI\\_DPU\\_MATCH\\_DisableChn](#)

## HI\_MPI\_DPU\_MATCH\_DisableChn

#### 【描述】

禁用通道。

#### 【语法】

```
HI_S32 HI_MPI_DPU_MATCH_DisableChn(DPU_MATCH_GRP DpuMatchGrp,  
DPU_MATCH_CHN DpuMatchChn);
```

#### 【参数】



参数名称	描述	输入/输出
DpuMatchGrp	组号。 取值范围：[0, 8)。	输入
DpuMatchChn	通道号。 取值范围：[0,1)。	输入

#### 【返回值】

返回值	描述
0	成功。
非 0	失败，参见 <a href="#">错误码</a> 。

#### 【需求】

- 头文件：hi\_comm\_dpu\_match.h、mpi\_dpu\_match.h
- 库文件：libdpu\_match.a

#### 【注意】

- 组必须已创建。
- 重复禁用返回成功。

#### 【举例】

无。

#### 【相关主题】

- [HI\\_MPI\\_DPU\\_MATCH\\_SetChnAttr](#)
- [HI\\_MPI\\_DPU\\_MATCH\\_GetChnAttr](#)
- [HI\\_MPI\\_DPU\\_MATCH\\_EnableChn](#)

## HI\_MPI\_DPU\_MATCH\_SendFrame

#### 【描述】

用户发送数据。

#### 【语法】

```
HI_S32 HI_MPI_DPU_MATCH_SendFrame(DPU_MATCH_GRP DpuMatchGrp, const  
VIDEO_FRAME_INFO_S *pstLeftFrame, const VIDEO_FRAME_INFO_S *pstRightFrame,  
HI_S32 s32MilliSec);
```

#### 【参数】



参数名称	描述	输入/输出
DpuMatchGrp	组号。 取值范围：[0, 8)。	输入
pstLeftFrame	左图图像。 不能为空。	输入
pstRightFrame	右图图像。 不能为空。	输入
s32MilliSec	超时参数 s32MilliSec 设为-1 时，为阻塞接口；0 时为非阻塞接口；大于 0 时为超时等待时间，超时的单位为毫秒（ms）。	输入

参数名称	支持类型	地址对齐	分辨率
pstLeftFrame	PIXEL_FORMAT_YUV_400	16 byte	128x64~1920x1080
pstRightFrame	PIXEL_FORMAT_YUV_400	16 byte	128x64~1920x1080

#### 【返回值】

返回值	描述
0	成功。
非 0	失败，参见 <a href="#">错误码</a> 。

#### 【需求】

- 头文件：hi\_comm\_dpu\_match.h、mpi\_dpu\_match.h
- 库文件：libdpu\_match.a

#### 【注意】

- 组必须已创建。
- 用户使用此接口时，可以自行进行帧率控制。
- pstLeftFrame/ pstRightFrame 图像地址必须是 VB 申请的，Stride 要求 16 字节对齐。

#### 【举例】

无。

#### 【相关主题】



- [HI\\_MPI\\_DPU\\_MATCH\\_GetFrame](#)
- [HI\\_MPI\\_DPU\\_MATCH\\_ReleaseFrame](#)

## HI\_MPI\_DPU\_MATCH\_GetFrame

### 【描述】

用户从通道获取一帧处理完成的图像。

### 【语法】

```
HI_S32 HI_MPI_DPU_MATCH_GetFrame(DPU_MATCH_GRP DpuMatchGrp,
VIDEO_FRAME_INFO_S *pstSrcLeftFrame, VIDEO_FRAME_INFO_S *pstSrcRightFrame,
VIDEO_FRAME_INFO_S *pstDstFrame, HI_S32 s32MilliSec);
```

### 【参数】

参数名称	描述	输入/输出
DpuMatchGrp	组号。 取值范围：[0, 8)。	输入
pstSrcLeftFrame	左图原始图像。 当组属性的 bNeedSrcFrame 设置为 HI_TRUE 时，不能为空。	输出
pstSrcRightFrame	右图原始图像。 当组属性的 bNeedSrcFrame 设置为 HI_TRUE 时，不能为空。	输出
pstDstFrame	匹配后图像。 不能为空。	输出
s32MilliSec	超时参数 s32MilliSec 设为-1 时，为阻塞接口；0 时为非阻塞接口；大于 0 时为超时等待时间，超时时间的单位为毫秒（ms）。	输入

参数名称	支持类型	地址对齐	分辨率
pstSrcLeftFrame	PIXEL_FORMAT_YUV_400	16 byte	128x64~1920x1080
pstSrcRightFrame	PIXEL_FORMAT_YUV_400	16 byte	128x64~1920x1080
pstDstFrame	PIXEL_FORMAT_S16	16 byte	128x64~1920x1080

### 【返回值】



返回值	描述
0	成功。
非 0	失败，参见 <a href="#">错误码</a> 。

#### 【需求】

- 头文件：hi\_comm\_dpu\_match.h、mpi\_dpu\_match.h
- 库文件：libdpu\_match.a

#### 【注意】

- 组必须已创建。
- 当组属性的队列深度不为 0 时，才能获取到图像。
- 若是用户发送数据模式，pstSrcLeftFrame 和 pstSrcRightFrame 数据信息由用户保证，若是绑定模式，由系统保证。
- pstDstFrame 数据信息由系统保证。
- 输出视差图以右图为准图像。
- 输出结果数据元素格式为 S10Q6(1bit 符号位+9bit 整数部分+6bit 小数部分)。
- 输出图像分辨率与输入右图相同。

#### 【举例】

无。

#### 【相关主题】

- [HI\\_MPI\\_DPU\\_MATCH\\_SendFrame](#)
- [HI\\_MPI\\_DPU\\_MATCH\\_ReleaseFrame](#)

## HI\_MPI\_DPU\_MATCH\_ReleaseFrame

#### 【描述】

用户释放一帧通道图像。

#### 【语法】

```
HI_S32 HI_MPI_DPU_MATCH_ReleaseFrame(DPU_MATCH_GRP DpuMatchGrp, const  
VIDEO_FRAME_INFO_S *pstSrcLeftFrame, const VIDEO_FRAME_INFO_S  
*pstSrcRightFrame, const VIDEO_FRAME_INFO_S *pstDstFrame);
```

#### 【参数】

参数名称	描述	输入/输出
DpuRectGrp	组号。 取值范围：[0, 8)。	输入



参数名称	描述	输入/输出
pstSrcLeftFrame	左图原始图像。 当组属性的 bNeedSrcFrame 设置为 HI_TRUE 时，不能为空。	输入
pstSrcRightFrame	右图原始图像。 当组属性的 bNeedSrcFrame 设置为 HI_TRUE 时，不能为空。	输入
pstDstFrame	匹配后图像。 不能为空。	输入

#### 【返回值】

返回值	描述
0	成功。
非 0	失败，参见 <a href="#">错误码</a> 。

#### 【需求】

- 头文件：hi\_comm\_dpu\_match.h、mpi\_dpu\_match.h
- 库文件：libdpu\_match.a

#### 【注意】

- 组必须已创建。
- pstSrcLeftFrame/pstSrcRightFrame/ pstDstFrame 由 [HI\\_MPI\\_DPU\\_MATCH\\_GetFrame](#) 获取，与其配对使用。

#### 【举例】

无。

#### 【相关主题】

- [HI\\_MPI\\_DPU\\_MATCH\\_SendFrame](#)
- [HI\\_MPI\\_DPU\\_MATCH\\_GetFrame](#)

## 2.4 数据类型和数据结构

DPU\_MATCH 相关数据类型、数据结构定义如下：

- [DPU\\_MATCH\\_GRP](#)：定义组号。
- [DPU\\_MATCH\\_CHN](#)：定义通道号。



- [DPU\\_MATCH\\_MASK\\_MODE\\_E](#): 定义聚合模板模式。
- [DPU\\_MATCH\\_DENS\\_ACCU\\_MODE\\_E](#): 定义稠密精度模式。
- [DPU\\_MATCH\\_SPEED\\_ACCU\\_MODE\\_E](#): 定义速度精度模式。
- [DPU\\_MATCH\\_DISP\\_SUBPIXEL\\_E](#): 定义是否计算亚像素枚举值。
- [DPU\\_MATCH\\_GRP\\_ATTR\\_S](#): 定义组属性。
- [DPU\\_MATCH\\_CHN\\_ATTR\\_S](#): 定义通道属性。
- [DPU\\_MATCH\\_MEM\\_INFO\\_S](#): 定义一维内存信息。

## DPU\_MATCH\_GRP

### 【说明】

定义组号。

### 【定义】

```
typedef HI_S32 DPU_MATCH_GRP;
```

### 【成员】

无。

### 【注意事项】

无。

### 【相关数据类型及接口】

无。

## DPU\_MATCH\_CHN

### 【说明】

定义通道号。

### 【定义】

```
typedef HI_S32 DPU_MATCH_CHN;
```

### 【成员】

无。

### 【注意事项】

无。

### 【相关数据类型及接口】

无。

## DPU\_MATCH\_MASK\_MODE\_E

### 【说明】



定义聚合模板模式。

#### 【定义】

```
typedef enum hiDPU_MATCH_MASK_MODE_E
{
    DPU_MATCH_MASK_MODE_DEFAULT = 0x0,
    DPU_MATCH_MASK_MODE_1X1     = 0x1,
    DPU_MATCH_MASK_MODE_3X3     = 0x2,
    DPU_MATCH_MASK_MODE_5X5     = 0x3,
    DPU_MATCH_MASK_MODE_7X7     = 0x4,
    DPU_MATCH_MASK_MODE_9X9     = 0x5,
    DPU_MATCH_MASK_MODE_BUTT
}DPU_MATCH_MASK_MODE_E;
```

#### 【成员】

成员名称	描述
DPU_MATCH_MASK_MODE_DEFAULT	默认模板。
DPU_MATCH_MASK_MODE_1X1	1x1 模板。
DPU_MATCH_MASK_MODE_3X3	3x3 模板。
DPU_MATCH_MASK_MODE_5X5	5x5 模板。
DPU_MATCH_MASK_MODE_7X7	7x7 模板。
DPU_MATCH_MASK_MODE_9X9	9x9 模板。

#### 【注意事项】

模板大小影响每一像素匹配时的关联区域大小,建议优先选择默认模板或 9x9 模板。

#### 【相关数据类型及接口】

无。

## DPU\_MATCH\_DENS\_ACCU\_MODE\_E

#### 【说明】

定义稠密精度模式。

#### 【定义】

```
typedef enum hiDPU_MATCH_DENS_ACCU_MODE_E
{
    DPU_MATCH_DENS_ACCU_MODE_D0_A9 = 0x0,
    DPU_MATCH_DENS_ACCU_MODE_D1_A8 = 0x1,
    DPU_MATCH_DENS_ACCU_MODE_D2_A7 = 0x2,
```





```

DPU_MATCH_DENS_ACCU_MODE_D3_A6 = 0x3,
DPU_MATCH_DENS_ACCU_MODE_D4_A5 = 0x4,
DPU_MATCH_DENS_ACCU_MODE_D5_A4 = 0x5,
DPU_MATCH_DENS_ACCU_MODE_D6_A3 = 0x6,
DPU_MATCH_DENS_ACCU_MODE_D7_A2 = 0x7,
DPU_MATCH_DENS_ACCU_MODE_D8_A1 = 0x8,
DPU_MATCH_DENS_ACCU_MODE_D9_A0 = 0x9,
DPU_MATCH_DENS_ACCU_MODE_BUTT
}DPU_MATCH_DENS_ACCU_MODE_E;

```

#### 【成员】

成员名称	描述
DPU_MATCH_DENS_ACCU_MODE_D0_A9	稠密度为 0，精度为 9。
DPU_MATCH_DENS_ACCU_MODE_D1_A8	稠密度为 1，精度为 8。
DPU_MATCH_DENS_ACCU_MODE_D2_A7	稠密度为 2，精度为 7。
DPU_MATCH_DENS_ACCU_MODE_D3_A6	稠密度为 3，精度为 6。
DPU_MATCH_DENS_ACCU_MODE_D4_A5	稠密度为 4，精度为 5。
DPU_MATCH_DENS_ACCU_MODE_D5_A4	稠密度为 5，精度为 4。
DPU_MATCH_DENS_ACCU_MODE_D6_A3	稠密度为 6，精度为 3。
DPU_MATCH_DENS_ACCU_MODE_D7_A2	稠密度为 7，精度为 2。
DPU_MATCH_DENS_ACCU_MODE_D8_A1	稠密度为 8，精度为 1。
DPU_MATCH_DENS_ACCU_MODE_D9_A0	稠密度为 9，精度为 0。

#### 【注意事项】

- 稠密度：DPU 输出包含无效点的稀疏视差图，无效点越多稠密度越低。  
例：DPU\_MATCH\_DENS\_ACCU\_MODE\_D0\_A9 表示稠密度最低，精度最高。建议采用 DPU\_MATCH\_DENS\_ACCU\_MODE\_D9\_A0。
- 不同场景下各模式表现会有所不同，可能出现相邻模式效果相似或颠倒的情况。

#### 【相关数据类型及接口】

无。

## DPU\_MATCH\_SPEED\_ACCU\_MODE\_E

#### 【说明】

定义速度精度模式。

#### 【定义】



```
typedef enum hiDPU_MATCH_SPEED_ACCU_MODE_E
{
    DPU_MATCH_SPEED_ACCU_MODE_SPEED = 0x0,
    DPU_MATCH_SPEED_ACCU_MODE_ACCU = 0x1,
    DPU_MATCH_SPEED_ACCU_BUTT
}DPU_MATCH_SPEED_ACCU_MODE_E;
```

#### 【成员】

成员名称	描述
DPU_MATCH_SPEED_ACCU_MODE_SPEED	速度模式。
DPU_MATCH_SPEED_ACCU_MODE_ACCU	精度模式。

#### 【注意事项】

速度模式下，DPU 处理速度提升约一倍，精度有所下降。

#### 【相关数据类型及接口】

无。

## DPU\_MATCH\_DISP\_SUBPIXEL\_E

#### 【说明】

定义是否计算亚像素枚举值。

#### 【定义】

```
typedef enum hiDPU_MATCH_DISP_SUBPIXEL_E
{
    DPU_MATCH_DISP_SUBPIXEL_DISABLE = 0x0,
    DPU_MATCH_DISP_SUBPIXEL_ENABLE = 0x1,
    DPU_MATCH_DISP_SUBPIXEL_BUTT
}DPU_MATCH_DISP_SUBPIXEL_E;
```

#### 【成员】

成员名称	描述
DPU_MATCH_DISP_SUBPIXEL_DISABLE	不计算亚像素。
DPU_MATCH_DISP_SUBPIXEL_ENABLE	计算亚像素。

#### 【注意事项】

计算亚像素时，输出视差包含 6bit 小数位，不计算时，输出视差仅包含整数位。



## 【相关数据类型及接口】

无。

## DPU\_MATCH\_GRP\_ATTR\_S

### 【说明】

定义组属性。

### 【定义】

```
typedef struct hiDPU_MATCH_GRP_ATTR_S
{
    SIZE_S stLeftImageSize; /* Left image size. */
    SIZE_S stRightImageSize; /* Right image size. */
    DPU_MATCH_MASK_MODE_E enMatchMaskMode; /* Aggregation mask mode. */
    DPU_MATCH_DENS_ACCU_MODE_E enDensAccuMode; /* Adjust density-accuracy
trade-off. */
    DPU_MATCH_SPEED_ACCU_MODE_E enSpeedAccuMode; /* Adjust speed-accuracy
trade-off. */
    DPU_MATCH_DISP_SUBPIXEL_E enDispSubpixelEn; /* Calculate subpixel
disparity or not. */
    HI_U16 u16DispNum; /* The number of disparity, when the value is less
than or equal to 128, it must be the multiple of 16, and when the value is
greater than 128, it must be the multiple of 32. Range:[16,256] */
    HI_U16 u16DispStartPos; /* Minimum disparity, it must be the multiple
of 2. Range:[0,126] */
    HI_U32 u32Depth; /* The depth of user image queue for getting Match
output image, it can not be changed dynamic. Range:[0,8] */
    HI_BOOL bNeedSrcFrame; /* The flag of getting source videoframe. It
will effect when bind dpu rect. */
    DPU_MATCH_MEM_INFO_S stAssistBuf; /* Assistance buffer. */
    FRAME_RATE_CTRL_S stFrameRate; /* Grp frame rate contrl. */
} DPU_MATCH_GRP_ATTR_S;
```

### 【成员】

成员名称	描述
stLeftImageSize	左图分辨率。取值范围: 宽[128, 1920]; 高[64, 1080]。
stRightImageSize	右图分辨率。取值范围: 宽[128, 1920]; 高[64, 1080]。
enMaskMode	聚合模式。
enDensAccuMode	稠密精度模式。
enSpeedAccuMode	速度精度模式。



成员名称	描述
enDispSubpixel	是否计算亚像素。
u16DispNum	视差值数目，取值范围请见【芯片差异】。
u16DispStartPos	最小视差起始地址，取值范围: [0, 126]。
u32Depth	通道图像的队列长度，静态属性，不可动态更改。取值范围: [0, 8]。
stAssistBuf	辅助内存。
stFrameRate	组帧率。

#### 【芯片差异】

芯片类型	差异
Hi3559AV100	u16DispNum 取值范围: [16, 256] u16DispNum 当取值小于等于 128 时，必须为 16 的倍数，当大于 128 时，必须为 32 的倍数。
Hi3519AV100	u16DispNum 取值范围: [16, 224]，必须为 16 的倍数。

#### 【注意事项】

- 右图宽度需大于等于视差起始位置+视差数组（即组属性 u16DispStartPos 和 u16DispNum）。
- 左图宽度需大于等于右图宽度。
- 左图右图高度相等。
- 分辨率宽、高要求偶数对齐。
- u16DispStartPos 必须为偶数。

#### 【相关数据类型及接口】

无。

## DPU\_MATCH\_CHN\_ATTR\_S

#### 【说明】

定义通道属性。

#### 【定义】

```
typedef struct hiDPU_MATCH_CHN_ATTR_S
{
    SIZE_S stImageSize; /* output image size */
} DPU_MATCH_CHN_ATTR_S;
```



【成员】

成员名称	描述
stImageSize	目标分辨率。取值范围: 宽[128, 1920]; 高[64, 1080]。

【注意事项】

分辨率宽、高要求偶数对齐。

【相关数据类型及接口】

无。

## DPU\_MATCH\_MEM\_INFO\_S

【说明】

定义一维内存信息。

【定义】

```
typedef struct hiDPU_MATCH_MEM_INFO_S
{
    HI_U64  u64PhyAddr;
    HI_U64  u64VirAddr;
    HI_U32  u32Size;
}DPU_MATCH_MEM_INFO_S;
```

【成员】

成员名称	描述
u64PhyAddr	内存块物理地址。
u64VirAddr	内存块虚拟地址。
u32Size	内存块字节数,见图 2-1。

【注意事项】

无。

【相关数据类型及接口】

无。



## 2.5 错误码

本模块 API 错误码如表 2-1 所示。

表2-1 DPU\_MATCH 模块 API 错误码

错误代码	宏定义	描述
0xA0368001	HI_ERR_DPU_MATCH_INVALID_DEVID	设备 ID 超出合法范围
0xA0368002	HI_ERR_DPU_MATCH_INVALID_CHNID	通道组号错误或无效区域句柄
0xA0368003	HI_ERR_DPU_MATCH_ILLEGAL_PARAM	参数超出合法范围
0xA0368004	HI_ERR_DPU_MATCH_EXIST	重复创建已存在的设备、通道或资源
0xA0368005	HI_ERR_DPU_MATCH_UNEXIST	试图使用或者销毁不存在的设备、通道或者资源
0xA0368006	HI_ERR_DPU_MATCH_NULL_PTR	函数参数中有空指针
0xA0368007	HI_ERR_DPU_MATCH_NOT_CONFIG	模块没有配置
0xA0368008	HI_ERR_DPU_MATCH_NOT_SUPPORT	不支持的参数或者功能
0xA0368009	HI_ERR_DPU_MATCH_NOT_PERM	该操作不允许，如试图修改静态配置参数
0xA036800C	HI_ERR_DPU_MATCH_NOMEM	分配内存失败，如系统内存不足
0xA036800D	HI_ERR_DPU_MATCH_NOBUF	分配缓存失败，如申请的图像缓冲区太大
0xA036800E	HI_ERR_DPU_MATCH_BUF_EMPTY	缓冲区中无图像
0xA036800F	HI_ERR_DPU_MATCH_BUF_FULL	缓冲区中图像满
0xA0368010	HI_ERR_DPU_MATCH_NOTREADY	系统没有初始化或没有加载相应模块
0xA0368011	HI_ERR_DPU_MATCH_BADADDR	地址非法
0xA0368012	HI_ERR_DPU_MATCH_BUSY	系统忙
0xA0368040	HI_ERR_DPU_MATCH_SYS_TIMEOUT	系统超时
0xA0368041	HI_ERR_DPU_MATCH_OPEN_FILE	打开文件失败



错误代码	宏定义	描述
0xA0368042	HI_ERR_DPU_MATCH_READ_FILE	读文件失败
0xA0368043	HI_ERR_DPU_MATCH_WRITE_FILE	写文件失败

## 2.6 Proc 调试信息

### 2.6.1 概述

调试信息采用了 Linux 下的 proc 文件系统，可实时反映当前系统的运行状态，所记录的信息可供问题定位及分析时使用。

#### 【文件目录】

/proc/umap

#### 【信息查看方法】

- 在控制台上可以使用 cat 命令查看信息，cat /proc/umap/match；也可以使用其他常用的文件操作命令，例如 cp /proc/umap/match ./，将文件拷贝到当前目录。
- 在应用程序中可以将上述文件当作普通只读文件进行读操作，例如 fopen、fread 等。



#### 说明

参数在描述时有以下 2 种情况需要注意：

- 取值为 {0, 1} 的参数，如未列出具体取值和含义的对应关系，则参数为 1 时表示肯定，为 0 时表示否定。
- 取值为 {aaa, bbb, ccc} 的参数，未列出具体取值和含义的对应关系，但可直接根据取值 aaa、bbb 或 ccc 判断参数含义

### 2.6.2 Proc 信息说明

#### 【调试信息】

```
#cat /proc/umap/match
```

```
[DPU_MATCH] Version: [Hi3559AV100_MPP_V1.0.0.0 B010 Release], Build
Time[Jun 25 2018, 17:49:57]
```

```
-----DPU MATCH HardWare STATUS-----
```

```
Busy      STCnt
Y          0
```

```
-----DPU MATCH GRP ATTR1-----
```

```
GrpID bStart  GrpState bNeedSrcFrame Depth SrcFrmRate DstFrmRate
LeftWidth LeftHeight RightWidth RightHeight
```



```
0      Y      CREATE      Y      8      30      26      1920
1080      1920      1080

-----DPU MATCH GRP ATTR-----
MaskMode  DensAccuMode  SpeedAccuMode  SubpixelEn  DispNum  DispStartPos
AssistBufAddr  AssistBufSize
9X9      D9_A0      ACCURACY      1      64      0      8b86f000
655968

-----DPU MATCH CHN ATTR-----
GrpID  ChnID  Enable  Width  Height
0      0      Y      1920  1080

-----DPU MATCH PIPE QUEUE-----
GrpID  BusyNum  FreeNum
0      0      8

-----DPU MATCH OUT QUEUE-----
GrpID  BusyNum  FreeNum
0      0      8

-----DPU MATCH WORK QUEUE-----
BusyNum  FreeNum
0      63

-----DPU MATCH GRP STATUS-----
GrpID  FrameRate      StartCnt  StartFailCnt  FrmLost  InFrmLost
0      26      5915130      0      0      0

-----DPU MATCH RUN-TIME INFO-----
CntPerSec  MaxCntPerSec  TotalIntCntLastSec  TotalIntCnt  CostTm  MCostTm
CostTmPerSec  MCostTmPerSec  TotalIntCostTm  CostTmPerFrm  HwCostTmPerFrm
RunTm
26      27      5915125      5915129      45      77      1109
1296      261232432      37769      37742      227505
```

### 【调试信息分析】

记录当前 DPU\_RECT 属性配置以及状态信息。

### 【参数说明】

参数		描述
DPU MATCH HarWare STATUS	Busy	DPU MATCH 硬件是否忙。 Y: 硬件忙;





参数		描述
DPU MATCH 硬件 工作状态信息		N: 硬件空闲。
	STCnt	DPU MATCH 系统超时次数。
DPU MATCH GRP ATTR1 DPU MATCH 组属 性信息。	GrpID	GRP ID 号。 有效范围: [0, 8)。
	bStart	组启动状态。 Y: 启动; N: 未启动。
	GrpState	组工作状态。 CREATE:创建; DESTROYING:销毁。
	bNeedSrcFrame	是否需要获取原始图像标志。 Y: 需要获取原始图像; N: 不需要获取原始图像。
	Depth	输出队列深度。 取值范围: [0, 8]。
	SrcFrmRate	Grp 源帧率。
	DstFrmRate	Grp 目标帧率。
	LeftWidth	输入左图分辨率的宽。 取值范围: [128, 1920]。
	LeftHeight	输入左图分辨率的高。 取值范围: [64, 1080]。
	RightWidth	输入右图分辨率的宽。 取值范围: [128, 1920]。
	RightHeight	输入右图分辨率的高。 取值范围: [64, 1080]。
DPU MATCH GRP ATTR2 DPU MATCH 组属 性信息。	MaskMode	聚合模板模式。取值参考对应的枚举定义。
	DensAccuMode	稠密度精度模式。取值参考对应的枚举定义。
	SpeedAccuMode	速度精度模式。取值参考对应的枚举定义。
	SubpixelEn	亚像素使能。取值参考对应的枚举定义。



参数		描述
	DispNum	视差数目。取值参考 GRP 属性结构体描述。
	DispStartPos	起始视差位置。取值参考 GRP 属性结构体描述。
	AssistBufAddr	辅助内存地址。
	AssistBufSize	辅助内存大小。
DPU MATCH CHN ATTR DPU MATCH 通道属性信息	GrpID	GRP ID 号。 有效范围: [0, 8)。
	ChnID	CHN ID 号。 有效范围: [0, 1)。
	Enable	通道使能标志。 Y: 使能; N: 未使能。
	Width	通道输出的目标图像分辨率的宽。 取值范围: [128, 1920]。
	Height	通道输出的目标图像分辨率的高。 取值范围: [64, 1080]。
DPU MATCH PIPE QUEUE DPU MATCH 输入管道队列信息	GrpID	GRP ID 号。 有效范围: [0, 8)。
	BusyNum	已用节点数。
	FreeNum	可用节点数。
DPU MATCH OUT QUEUE DPU MATCH 输出通道队列信息	GrpID	GRP ID 号。 有效范围: [0, 8)。
	BusyNum	已用节点数。
	FreeNum	可用节点数。
DPU MATCH WORK QUEUE DPU MATCH 工作队列信息	BusyNum	已用节点数。
	FreeNum	可用节点数。
DPU MATCH GRP STATUS DPU MATCH 组状	GrpID	GRP ID 号。 有效范围: [0, 8)。
	FrameRate	输出帧率。



参数		描述
态信息	StartCnt	任务启动次数。
	StartFailCnt	任务启动失败次数。
	SendPicCnt	发图像成功次数。
	FrmLost	因队列满丢失的输出图像数
	CurTaskCostTm	当前完成任务的耗时。
	MaxTaskCostTm	历史上耗时最长任务的执行时间。
	GrpID	GRP ID 号。 有效范围：[0, 8)。
	InFrmLost	因队列满丢失的输入图像数。
DPU MATCH RUN-TIME INFO DPU MATCH 运行时相关信息	CntPerSec	最近一次的 1 秒内中断执行次数。
	MaxCntPerSec	历史上的 1 秒内最大的中断执行次数。
	TotalIntCntLastSec	上一秒上报中断总次数。
	TotalIntCnt	DPU MATCH 产生中断的总次数。
	CostTm	最近一次执行中断的执行耗时。 单位：us
	MCostTm	执行一次中断的最大耗时。 单位：us
	CostTmPerSec	最近一秒执行中断的执行耗时。 单位：us
	MCostTmPerSec	历史上一秒执行中断的最大执行耗时。 单位：us
	TotalIntCostTm	中断处理总时间。 单位：us
	CostTmPerFrm	单帧图像处理的总时间。 单位：us
	HwCostTmPerFrm	单帧图像处理的硬件时间。 单位：us
	RunTm	DPU MATCH 运行总时间。 单位：s