* HT\_VPE 使用说明

VPE stream 流向

1. insmod halut.ko
2. echo vpe 0 1 1 1 0 0 0 1 1 1 1 1 1 0 2 1 1 1 2 2 9 3 1 1 1 3 3 9 >/proc/hal/uttest

* 最后一个参数和> 之间至少需要一个空格。

每个channel 7个参数 分别是

* channelId MAX 16
* Iq 0代表IqOnOffconfig 全是FALSE；1~3 IqOnOffconfig 全是TRUE,分别代表iq三组参数的强中弱：

const HalVpeIqConfig\_t stIqConfig[HT\_DIVP\_MAX\_CASE] =

{

{255,255,255,255,14,15,16,16,16,{255,255,255,255,255,0},255},

{80,60,80,60,4,8,16,4,0,{0,20,80,120,160,160},0},

{0,0,0,0,0,0,0,0,0,{0,0,0,0,0,0},128}

};

* Roation 0~3 分别代表off，90，180，270度旋转。
* Crop 0~3分别代表 0代表bCropEn = FALSE，1~3分别代表下面三组参数

const HalSclCropWindowConfig\_t stCropWin[HT\_DIVP\_MAX\_CASE]=

{

{0,0,960,576},//WD1

{0,0,704,288},// 2CIF

{0,0,176,144} // QCIF

};

* inputsize 0~3分别代表下面4组值

HT\_size\_t stInPutSizeParam[HT\_DIVP\_MAX\_CASE]=

{

{1920,1088},

{1280,720},

{720,480},

{640,480},

};

* outputsize 分别代表下面4组值，4个output port的size相同。

HT\_size\_t stOutPutSizeParam[HT\_DIVP\_MAX\_CASE]=

{

{1920,1088},

{1280,720},

{720,480},

{640,480},

};

* 在调试环境中需要设置 inputsize == output size
* pixelformat 0 代表E\_HT\_PIXEL\_FORMAT\_YUV422\_YUYV格式，9代表 E\_HT\_PIXEL\_FORMAT\_YUV\_SEMIPLANAR\_420 格式。目前只支持这两种格式。

1. 主流程图

根据用户输入参数的个数和每个channel 参数个数，得倒用户一共创建多少个channel. chnlwork\_num。以下是channeInfo 结构体，每个channel创建一个。

typedef struct HT\_VPE\_ChannelInfo\_s

{

void \*pIspCtx; // HAL layer: ISP context pointer

void \*pIqCtx; // HAL layer: IQ context pointer

void \*pSclCtx; // HAL layer: SCL context pointer

HT\_BOOL bCreate;

HT\_U8 VpeCh;

HT\_S64 s64ReadPos;

HT\_S64 s64WritePos[HT\_VPE\_MAXPORT];

char \*pFileName;

/\*User Set\*/

HalVpeIqConfig\_t stIqConfig; // Channel Iq param

HalVpeIqOnOff\_t stIqOnOff;

HalVpeIspRotationType\_e eRoationConfig; // Channel roation parameter

HalVpeSclCropConfig\_t stCropConfig; // Channel crop param

HT\_size\_t stInPutSize; // input port size

HT\_size\_t stOutPutSize[HT\_VPE\_MAXPORT]; // output size

HT\_PxlFmt\_e ePixelFormat;

} HT\_VPE\_ChannelInfo\_t;

4.VPE\_Run 流程图

每个channel做一帧，一个channel做完切换到下一个channel。