

# VDS 库

视频解码管理系统。完成

1. 视频解码系统的对用户接口。
2. 视频解码器和文件解析的调度协调。
3. 视频输入为文件句柄。
4. 系统输出到显示 **buffer** 模块。

## VDS 库函数

- 1、函数 `vds_init()` ;
- 2、函数 `vds_open()` ;
- 3、函数 `vds_ioctl(vds_t *vds, void *pram, u32 cmd);`
- 4、函数 `vds_close(vds_t *vds);`

### 1、函数 vds\_init()

函数名	vds_init
功能描述	初始化
输入参数	无
返回值	VDS_ERR_OK
先决条件	无
被调用函数	无

### 2、函数 vds\_open()

函数名	vds_open
函数原型	vds_t * vds_open()
功能描述	开辟空间，赋值零
输入参数	无
返回值	vds_t 结构指针
先决条件	无
被调用函数	

vds\_t 结构定义域 vds.h

```
typedef struct __vds
{
void *fb_mng;           //视频帧 BUF
void *afb_mng;          //音频帧 BUF
fb_pipe_t *vpipe;
fb_pipe_t *apipe;
void *mng;
u8 type;                //vds 类型 :0  AVI 解码 1 显示 2 jpg 解码
}vds_t;
```

### 3、函数 vds\_ioctl()

函数名	vds_ioctl
函数原型	u32 vds_ioctl(vds_t *vds, void* parm, u32 cmd)
功能描述	开辟空间，赋值零
输入参数 1	vds_t *vds: 传递 vds_open 返回的变量
输入参数 2	void* parm: 传递视频图片文件句柄或视频图片信息
输入参数 3	u32 cmd: 控制指令，参阅 cmd 值说明
返回值	错误状态
先决条件	无
被调用函数	无

cmd 值说明:

// avi 解码		
0x9000	VDS_DEC_OPEN	设置 IMB, 打开视频解码模块
0x9001	VDS_DEC_START	开始解码
0x9002	VDS_DEC_PAUSE	暂停视频解码
0x9003	VDS_DEC_STOP	停止解码模块
0x9004	VDS_DEC_FF	快进
0x9005	VDS_DEC_FR	快退
0x9006	VDS_DEC_SET_AVI	设置视频文件, 设置 imc 模块
0x9007	VDS_DEC_GET_AVI_INF	传递 avi 解码信息
0x9008	VDS_DEC_CUR_FRAME	获取当前解码帧数
0x9009		
0x900A	VDS_DEC_CONTINUE	继续视频解码
0x900B		
0x900C	VDS_DEC_KFBG	启动硬件解码一帧
0x900D		
0x900E	VDS_DEC_PREVIEW	预览视频
0x900F	VDS_DEC_CLOSE	关闭视频解码模块
图片解码		
0xa000	VDS_JPG_OPEN	设置显示模块, 打开图片解码模块
0xa001	VDS_JPG_DEC_PIC	解码图片 (文件读取方式)
0xa002	VDS_JPG_CLOSE	关闭图片解码模块
0xa003	VDS_JPG_DEC_BUF_PIC	解码通过参数传递的图片

返回值说明:

0xb000	VDS_ERR_OK	
0xb001	VDS_ERR_NULL_PTR	
0xb002	VDS_ERR_SET_FILE_FAILED	
0xb003	VDS_ERR_UNKNOW_CMD	
0xb004	VDS_ERR_UNPKG_UNINIT	
0xb005	VDS_ERR_FILE_NOT_READY	
0xb006	VDS_ERR_NOT_OPEN	
0xb007	VDS_ERR_FF_FR	
0xb008	VDS_ERR_NO_MEM	
0xb009	VDS_ERR_HEAD	
0xb00A	VDS_ERR_TYPE	
0xb00B	VDS_ERR_IMC	
0xb00C	VDS_ERR_JPG_START	
0xb00D	VDS_ERR_STATUS	
0xb00E	VDS_ERR_TIMEOUT	

#### 4、函数 vds\_close ()

函数名	vds_close
函数原型	u32 vds_close(vds_t *vds)
功能描述	释放指针 vds
输入参数	vds_t *vds
返回值	0
先决条件	无
被调用函数	无