

DMSDK V1.0 软件 开发参考

文档版本 00B06

发布日期 2019-02-21

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。

修订日期	版本	修订说明
2019-02-21	00B06	DM_Infrared_Stream_Register 添加用户数据参数 修改数据类型 INFRARED_NOTIFY_CB、INFRARED_DATA_CB 回调时传回用户数据 去掉部分无用函数
2019-01-22	00B05	新增接口 DM_Gray_To_Temperature、DM_Infrared_Stream_Start、DM_Infrared_Stream_Stop、DM_Infrared_Stream_Register 修改接口 DM_Stream_Start、DM_Stream_Stop、DM_Stream_Register 到 DM_Video_Stream_Start、DM_Video_Stream_Stop、DM_Video_Stream_Register 修改数据类型 NOTIFY_CB、VIDEO_CB 到 VIDEO_NOTIFY_CB、VIDEO_STREAM_CB 新增数据类型 INFRARED_NOTIFY_CB、INFRARED_DATA_CB
2018-12-19	00B04	新增加接口 DM_Sys_Init、DM_Sys_UnInit、DM_Stream_Start、DM_Stream_Stop、DM_Stream_Register 新增加数据类型 NOTIFY_CB、VIDEO_CB
2018-11-06	00B03	涉及修改数据类型 DM_ALARM_INFO 新增加数据类型 DM_ALARM_CELL、DM_OSD_UDF、DM_LOG、MAX_LOG_LEVEL、LOG_LEVEL_DEFAULT 新增加接口 DM_Log_Init、DM_Log_Exit、DM_Set_Osd_Udf、DM_Get_Osd_Udf
2018-11-01	00B02	涉及修改数据类型 TEMP_PONIT、TEMP_AREA DM_SYS_INFO、DM_OSD_CAM、DM_OSD_TIME、DM_TEMP_MCONFIG
2018-10-31	00B01	第 1 次临时版本发布

目 录

目 录.....	3
1、概述.....	4
1.1、目的.....	4
1.2、软件架构.....	4
2、系统接口.....	5
2.1、API 参考.....	5
DM_Sys_Init.....	5
DM_Sys_UnInit.....	5
DM_Open.....	6
DM_Close.....	6
DM_Version.....	7
DM_Log_Init.....	7
DM_Log_Exit.....	8
2.2、数据类型.....	8
LOG_LEVEL_DEFAULT.....	8
MAX_LOG_LEVEL.....	8
RESV_LEN.....	9
DM_VERSION.....	9
DM_LOG.....	10
3、功能接口.....	11
3.1、API 参考.....	11
DM_Vedio_Stream_Start.....	11
DM_Vedio_Stream_Stop.....	11
DM_Vedio_Stream_Register.....	12
DM_Infrared_Stream_Start.....	12
DM_Infrared_Stream_Stop.....	13
DM_Infrared_Stream_Register.....	13
DM_Gray_To_Temperature.....	14
3.2、数据类型.....	14
VIDEO_NOTIFY_CB.....	14
VIDEO_STREAM_CB.....	15
INFRARED_NOTIFY_CB.....	15
INFRARED_DATA_CB.....	15
MAX_TMEP_NUM.....	16
MAX_CAM_NAME_LEN.....	16
附录.....	17
错误码.....	17

1、概述

1.1、目的

DMSDK 是针对红外网络摄像机在 linux 下开发的一套 sdk，方便客户在 linux 下进行二次开发，发布的库文件基于 64 位编译，在 Ubuntu16.04（64 位）系统上经过验证，如果需要 32 位的库，可以联系技术支持获得。

1.2、软件架构

初始化程序时，先通过 DM_Open 调用，传入设备端的相关信息，调用成功后，返回一个句柄，此后的操作都基于这个句柄，软件退出时，调用 DM_Close 关闭即可。

2、系统接口

2.1、API 参考

DM_Sys_Init

【描述】

系统初始化，放在程序最开始调用，而且只能调用一次。

【语法】

```
int DM_Sys_Init();
```

【参数】

参数名称	描述	输入/输出

【返回值】

返回值	描述
0	成功
非 0	参见 错误码

DM_Sys_UnInit

【描述】

系统反初始化，放在程序最后调用，而且只能调用一次。

【语法】

```
int DM_Sys_UnInit();
```

【参数】

参数名称	描述	输入/输出

【返回值】

返回值	描述
0	成功
非 0	参见 错误码

DM_Open

【描述】

初始化打开一个连接通道，并返回句柄，之后的其它功能操作，都要基于这个句柄。

【语法】

```
int DM_Open(long *hdl, const char *ip, int port, char *user, char *pwd);
```

【参数】

参数名称	描述	输入/输出
hdl	返回的接口句柄	输出
ip	待连接设备的 ip 地址	输入
port	待连接设备的 ip 端口	输入
user	待连接设备的用户名	输入
pwd	待连接设备的用户密码	输入

【返回值】

返回值	描述
0	成功
非 0	参见 错误码

DM_Close

【描述】

关闭一个连接通道。

【语法】

```
int DM_Close(long hdl);
```

【参数】

参数名称	描述	输入/输出
hdl	接口句柄	输入

【返回值】

返回值	描述
0	成功
非 0	参见 错误码

DM_Version

【描述】

获取 sdk 版本信息。

【语法】

```
int DM_Version(DM\_VERSION *version);
```

【参数】

参数名称	描述	输入/输出
version	版本信息	输出

【返回值】

返回值	描述
0	成功
非 0	参见 错误码

DM_Log_Init

【描述】

初始化日志功能。

【语法】

```
int DM_Log_Init(DM\_LOG *log);
```

【参数】

参数名称	描述	输入/输出
log	日志配置	输入

【返回值】

返回值	描述
0	成功
非 0	参见 错误码

DM_Log_Exit

【描述】

退出日志功能。

【语法】

```
int DM_Log_Exit(void);
```

【参数】

参数名称	描述	输入/输出

【返回值】

返回值	描述
0	成功
非 0	参见 错误码

2.2、数据类型

LOG_LEVEL_DEFAULT

【说明】

定义默认日志等级。

【定义】

```
#define LOG_LEVEL_DEFAULT    3
```

【成员】

无

MAX_LOG_LEVEL

【说明】

定义最大日志等级。

【定义】

```
#define MAX_LOG_LEVEL    8
```

【成员】

无

RESV_LEN**【说明】**

定义保留字段长度。

【定义】

```
#define RESV_LEN    64
```

【成员】

无

DM_VERSION**【说明】**

定义 sdk 软件版本信息结构体。

【定义】

```
typedef struct
{
    char version[64];
    int resv[RESV_LEN];
}DM_VERSION;
```

【成员】

成员名称	描述
version	版本信息字符串
resv	保留字段

DM_LOG

【说明】

定义 sdk 软件版本信息结构体。

【定义】

```
typedef struct
{
    char path[64];        /* log file save path */
    int level[16];        /* log level, level[0] valid */
    int roll;             /* rollback, [1, MAX_ROLL_NUM] */
    int unused;
    int resv[RESV_LEN];
}DM_LOG;
```

【成员】

成员名称	描述
path	日志存储目录，若不存在，则默认为./log
level	日志等级，level[0]有效，level[1-15]保留范围[0, MAX LOG LEVEL]，0:关闭日志，建议设置为 LOG LEVEL DEFAULT ，值越大，输出越多
roll	日志回滚周期，以天为单位
unused	保留字段
resv	保留字段

3、功能接口

3.1、API 参考

DM_Vedio_Stream_Start

【描述】

准备开始获取视频码流。

【语法】

```
int DM_Vedio_Stream_Start(long hdl, int port);
```

【参数】

参数名称	描述	输入/输出
hdl	接口句柄	输入
port	rtsp 端口号, 默认 554	输入

【返回值】

返回值	描述
0	成功
非 0	参见 错误码

DM_Vedio_Stream_Stop

【描述】

停止获取视频码流。

【语法】

```
int DM_Vedio_Stream_Stop(long hdl);
```

【参数】

参数名称	描述	输入/输出
hdl	接口句柄	输入

【返回值】

返回值	描述
-----	----

0	成功
非 0	参见 错误码

DM_Vedio_Stream_Register

【描述】

注册视频回调函数。

【语法】

```
int DM_Vedio_Stream_Register(long hdl, VIDEO_NOTIFY_CB notify_cb,  
VIDEO_STREAM_CB video_cb);
```

【参数】

参数名称	描述	输入/输出
hdl	接口句柄	输入
notify_cb	事件回调函数指针	输入
video_cb	视频流回调函数指针	输入

【返回值】

返回值	描述
0	成功
非 0	参见 错误码

DM_Infrared_Stream_Start

【描述】

准备开始获取 Infrared 码流。

【语法】

```
int DM_Infrared_Stream_Start(long hdl, int port, int framerate);
```

【参数】

参数名称	描述	输入/输出
hdl	接口句柄	输入
port	Infrared 端口号, 默认 5000	输入

Framerate	帧率	输入
-----------	----	----

【返回值】

返回值	描述
0	成功
非 0	参见 错误码

DM_Infrared_Stream_Stop

【描述】

停止获取 Infrared 码流。

【语法】

```
int DM_Infrared_Stream_Stop(long hdl);
```

【参数】

参数名称	描述	输入/输出
hdl	接口句柄	输入

【返回值】

返回值	描述
0	成功
非 0	参见 错误码

DM_Infrared_Stream_Register

【描述】

注册 infrared 回调函数。

【语法】

```
int DM_Infrared_Stream_Register(long hdl, INFRARED_NOTIFY_CB notify_cb,  
INFRARED_DATA_CB video_cb);
```

【参数】

参数名称	描述	输入/输出
------	----	-------

hdl	接口句柄	输入
notify_cb	事件回调函数指针	输入
video_cb	视频流回调函数指针	输入

【返回值】

返回值	描述
0	成功
非 0	参见 错误码

DM_Gray_To_Temperature

【描述】

灰度转温度。

【语法】

```
int DM_Gray_To_Temperature(long hdl, int gray, float *pfTemp);
```

【参数】

参数名称	描述	输入/输出
hdl	接口句柄	输入
gray	灰度值	输入
pfTemp	温度	输出

【返回值】

返回值	描述
0	成功
非 0	参见 错误码

3.2、数据类型

VIDEO_NOTIFY_CB

【说明】

事件回调函数指针。

【定义】

```
typedef int (*VIDEO_NOTIFY_CB)(int, void *);
```

【成员】

无

VIDEO_STREAM_CB**【说明】**

视频流回调函数指针。

【定义】

```
typedef int (*VIDEO_STREAM_CB)(unsigned char*, int, unsigned int,  
unsigned short, void *);
```

【成员】

无

INFRARED_NOTIFY_CB**【说明】**

事件回调函数指针。

【定义】

```
typedef int (*INFRARED_NOTIFY_CB)(int, void *);
```

【成员】

无

INFRARED_DATA_CB**【说明】**

INFRARED 流回调函数指针。

【定义】

```
typedef int (*INFRARED_DATA_CB)(unsigned char*, int, int, int,  
void *);
```

【成员】

无

MAX_TMEP_NUM**【说明】**

定义实时测温的最大个数。

【定义】

```
#define MAX_TMEP_NUM    6
```

【成员】

无

MAX_CAM_NAME_LEN**【说明】**

定义通道名称最大长度。

【定义】

```
#define MAX_CAM_NAME_LEN    64
```

【成员】

无

附录

错误码

错误代码	宏定义描述	描述
1	ERR_UNKNOWN	
2	ERR_INVALID_HANDLE	
3	ERR_NO_MEMORY	
4	ERR_SOCKET_INVALID	
5	ERR_SOCKET_CANT_SET	
6	ERR_SOCKET_RESOLVE	
7	ERR_SOCKET_CONNECT	
8	ERR_SOCKET_TIME_OUT	
9	ERR_SOCKET_RECV	
10	ERR_SOCKET_SEND	
11	ERR_HEADER_RECV	
12	ERR_HEADER_NOT_FOUND	
13	ERR_HEADER_BIG_CLUE	
14	ERR_HEADER_NO_LENGTH	
15	ERR_CHUNK_TOO_BIG	
16	ERR_AUTH_HOST	
17	ERR_AUTH_PROXY	
18	ERR_BAD_VERB	
19	ERR_LONG_INPUT	
20	ERR_BAD_STATE	
21	ERR_CHUNK	
22	ERR_BAD_URL	
23	ERR_BAD_HEADER	
24	ERR_BUFFER_RSIZE	
25	ERR_BAD_AUTH	
26	ERR_AUTH_MISMATCH	
27	ERR_NO_DIGEST_TOKEN	
28	ERR_NO_DIGEST_ALG	
29	ERR_SOCKET_BIND	
30	ERR_TLS_NEGO	
64	ERR_NOT_IMPLEMENTED	