P2PAPI 开发说明 1.0.0.9

修改历史

日期	修改说明	修改人	版本
2012/08/16	The first draft	Iven,Ji	1.0.0.1
2012/09/21	1、增加参数设置与获取 2、增加用户合法性校验	Iven,Ji	1.0.0.2
2012/9/25	1、修改设置问题 2、增加报警通知 3、云台控制不使用单步 4、增加设备名称设置	Iven,Ji	1.0.0.3
2012/11/09	修正 android 和 iphone 在停止时的 bug	Iven,Ji	1.0.0.4
2012/12/26	1、修改 iphone 上启动线程时的 bug 2、增加获取 TF 卡录像列表消息 3、增加 TF 卡录像回放开始和停止消息 4、增加 P2PAPI_InitialWithServer 接口，可以传入服务器的地址 5、增加远程回放时的音视频数据回调设置接口 P2PAPI_SetPlaybackAVDataCallBack	Iven,Ji	1.0.0.5
2013/01/03	修正文档错误	Iven,Ji	1.0.0.5
2013/01/24	1、解决数据超过 64KB 时的传输问题 2、修正文档错误	Iven,Ji	1.0.0.6
2013/02/20	1、将设置回调改成针对某一个实例设置回调 2、支持分页获取 TF 卡录像文件列表（需要设备本身支持） 3、修改 STRU_RECORD_FILE_LIST 结构，增加 nRecordCount,nPageCount,nPageIndex,nPageSize 参数	Iven,Ji	1.0.0.7
2013/03/05	1、修改在 ios 下编译不过的问题 2、增加 ios Demo	Iven,Ji	1.0.0.8
2013/03/06	解决接收报警的 bug	Iven,Ji	1.0.0.9

目录

接口返回值

```
#define ERROR_P2PAPI_OK                     0
#define ERROR_P2PAPI_NOT_INITIALIZED        -1      //not initialized
#define ERROR_P2PAPI_INVALID_HANDLE         -2      //invalid handle
#define ERROR_P2PAPI_HANDLE_EXHAUST         -3      //handle is exhaust
#define ERROR_P2PAPI_INVALID_PARAM          -4      //invalid param
#define ERROR_P2PAPI_WRONG_CALL_ORDER       -5      //wrong call order
#define ERROR_P2PAPI_P2P_NOT_CONNECTED      -6
```

接口说明

long P2PAPI_Initial()

描述：初始化 P2PAPI 内部资源

参数：

无

返回值：

ERROR_P2PAPI_OK

ERROR_P2PAPI_NOT_INITIALIZED

long P2PAPI_InitialWithServer(char *svr)

描述：初始化 P2PAPI 内部资源

参数：

svr: 【IN】P2P 服务器的地址（具体地址请联系厂家索要）

返回值：

ERROR_P2PAPI_OK

ERROR_P2PAPI_NOT_INITIALIZED

NOTE：这个接口与 P2PAPI_Initial 只能调用一个

long P2PAPI_DeInitial()

描述：释放 P2PAPI 资源
参数：
　　无
返回值：
　　ERROR_P2PAPI_OK
　　ERROR_P2PAPI_NOT_INITIALIZED

long P2PAPI_GetAPIVersion()

描述：获取 P2PAPI 版本
参数：
　　无
返回值：
　　0x01020304 → Version 1.2.3.4

long P2PAPI_CreateInstance(long *nHandle)

描述：创建 P2PAPI 实例
参数：
　　nHandle：【OUT】P2P 实例句柄
返回值：
　　ERROR_P2PAPI_OK
　　ERROR_P2PAPI_NOT_INITIALIZED
　　ERROR_P2PAPI_HANDLE_EXHAUST

NOTE: 目前最大实例个数为 128

long P2PAPI_DestroyInstance(long nHandle)

描述：销毁 P2P 实例
参数：
　　nHandle：【IN】P2P 实例句柄
返回值：
　　ERROR_P2PAPI_OK
　　ERROR_P2PAPI_NOT_INITIALIZED
　　ERROR_P2PAPI_INVALID_HANDLE

long P2PAPI_DestroyAllInstance()

描述：销毁所有 P2P 实例

参数：

无

返回值：

ERROR_P2PAPI_OK

ERROR_P2PAPI_NOT_INITIALIZED

long P2PAPI_Connect(long nHandle, char *uid, char *user, char *pwd)

描述：开始 P2P 连接

参数：

nHandle：【IN】P2P 实例句柄

uid:【IN】P2P ID

user:【IN】登录用户名

pwd:【IN】登录密码

返回值：

ERROR_P2PAPI_OK

ERROR_P2PAPI_NOT_INITIALIZED

ERROR_P2PAPI_INVALID_HANDLE

ERROR_P2PAPI_INVALID_PARAM

long P2PAPI_Close(long nHandle)

描述：关闭 P2P 连接

参数：

nHandle：【IN】P2P 实例句柄

返回值：

ERROR_P2PAPI_OK

ERROR_P2PAPI_NOT_INITIALIZED

ERROR_P2PAPI_INVALID_HANDLE

ERROR_P2PAPI_INVALID_PARAM

long P2PAPI_CloseAll()

描述：关闭所有 P2P 连接

参数：

无

返回值：

ERROR_P2PAPI_OK

ERROR_P2PAPI_NOT_INITIALIZED

long P2PAPI_StartVideo(long nHandle)

描述：请求视频

参数：

nHandle：【IN】P2P 实例句柄

返回值：

ERROR_P2PAPI_OK

ERROR_P2PAPI_NOT_INITIALIZED

ERROR_P2PAPI_INVALID_HANDLE

ERROR_P2PAPI_WRONG_CALL_ORDER

ERROR_P2PAPI_P2P_NOT_CONNECTED

long P2PAPI_StopVideo(long nHandle)

描述：停止视频

参数：

nHandle：【IN】P2P 实例句柄

返回值：

ERROR_P2PAPI_OK

ERROR_P2PAPI_NOT_INITIALIZED

ERROR_P2PAPI_INVALID_HANDLE

ERROR_P2PAPI_WRONG_CALL_ORDER

long P2PAPI_StartAudio(long nHandle)

描述：请求音频

参数：

nHandle：【IN】P2P 实例句柄

返回值：

ERROR_P2PAPI_OK

ERROR_P2PAPI_NOT_INITIALIZED

ERROR_P2PAPI_INVALID_HANDLE

ERROR_P2PAPI_WRONG_CALL_ORDER
ERROR_P2PAPI_P2P_NOT_CONNECTED

long P2PAPI_StopAudio(long nHandle)

描述：停止音频
参数：
nHandle：【IN】P2P 实例句柄
返回值：
ERROR_P2PAPI_OK
ERROR_P2PAPI_NOT_INITIALIZED
ERROR_P2PAPI_INVALID_HANDLE
ERROR_P2PAPI_WRONG_CALL_ORDER

long P2PAPI_StartTalk(long nHandle)

描述：开始发言
参数：
nHandle：【IN】P2P 实例句柄
返回值：
ERROR_P2PAPI_OK
ERROR_P2PAPI_NOT_INITIALIZED
ERROR_P2PAPI_INVALID_HANDLE
ERROR_P2PAPI_WRONG_CALL_ORDER
ERROR_P2PAPI_P2P_NOT_CONNECTED

long P2PAPI_StopTalk(long nHandle)

描述：停止发言
参数：
nHandle：【IN】P2P 实例句柄
返回值：
ERROR_P2PAPI_OK
ERROR_P2PAPI_NOT_INITIALIZED
ERROR_P2PAPI_INVALID_HANDLE
ERROR_P2PAPI_WRONG_CALL_ORDER

long P2PAPI_TalkData(long nHandle,char *pData, int len)

描述：发送发言数据

参数：

nHandle：【IN】P2P 实例句柄

pData:【IN】ADPCM 数据指针

len：【IN】ADPCM 数据长度，注意，当前必须为 256

返回值：

ERROR_P2PAPI_OK

ERROR_P2PAPI_NOT_INITIALIZED

ERROR_P2PAPI_INVALID_HANDLE

ERROR_P2PAPI_WRONG_CALL_ORDER

long P2PAPI_SendMessage(long nHandle, char *msg, int len)

描述：发送消息

参数：

nHandle：【IN】P2P 实例句柄

返回值：

ERROR_P2PAPI_OK

ERROR_P2PAPI_NOT_INITIALIZED

ERROR_P2PAPI_INVALID_HANDLE

ERROR_P2PAPI_WRONG_CALL_ORDER

ERROR_P2PAPI_P2P_NOT_CONNECTED

ERROR_P2PAPI_INVALID_PARAM

long P2PAPI_SetAVDataCallBack(AVDataCallback AVCallback, void *pParam)

描述：设置音视频数据回调

参数：

AVCallback：【IN】

回调函数定义如下：

typedef void (*AVDataCallback)(long nHandle, int bVideo, char *pData, int len, void *pParam)

参数：

nHandle: P2P 实例句柄

bVideo: 0:音频 1:视频

pData: 数据指针

Len: 数据长度

pParam: 用户参数

NOTE: 音视频数据 = 音视频头 + 音视频数据

音视频头的定义为：

```
typedef struct tag_AV_HEAD
{
    unsigned int    startcode;    //  0xa815aa55
    unsigned char   type;         //  0->264 idr frame 1->264 p frame
    unsigned char   streamid;
    unsigned short  militime;     //  diff time
    unsigned int    sectime;      //  diff time
    unsigned int    frameno;      //  frameno
    unsigned int    len;          //  data len
    unsigned char   version;      // version
    unsigned char   sessid;       //ssid
    unsigned char   other[2];
    unsigned char   other1[8];
}AV_HEAD,*PAV_HEAD;
```

pParam：【IN】用户参数，在回调函数中将作为输入参数传回给用户

返回值：

ERROR_P2PAPI_OK

long P2PAPI_SetPlaybackAVDataCallBack(AVDataCallback AVCallback, void *pParam)

描述：设置远程回放时的音视频数据回调

参数：

请参考 P2PAPI_SetAVDataCallBack

返回值：

ERROR_P2PAPI_OK

long P2PAPI_SetMessageCallback(MessageCallback MsgCallback, void *pParam)

描述：设置消息回调
参数：
MsgCallback：【IN】
消息回调函数的定义如下：
typedef void (*MessageCallback)(long nHandle, int type, char *msg, int len, void *pParam)
参数：
nHandle: P2P 实例句柄
type: 消息的类型 （请参考**消息类型定义**）
msg: 消息内容
len: 消息长度
pParam: 用户参数
pParam：【IN】用户参数，会消息回调函数中将作为输入参数传回给用户
返回值：
ERROR_P2PAPI_OK

消息类型定义

MSG_TYPE_P2P_STATUS

描述：P2P 连接状态
发送：不支持
通知：支持
消息内容：int
消息长度：sizeof(int)
状态值定义：

```
#define P2P_STATUS_CONNECT_TIME_OUT      0x0      //连接超时
#define P2P_STATUS_INVALID_ID            0x1      //无效的 ID
#define P2P_STATUS_CONNECT_SUCCESS       0x2      //连接成功
```

```
#define P2P_STATUS_DISCONNECTED          0x3     //连接断开
#define P2P_STATUS_CONNECT_FAIlED        0x4     //连接失败
#define P2P_STATUS_CONNECTING            0x5     //正在连接
#define P2P_STATUS_DEVICE_NOT_ON_LINE    0x6     //设备不在线
#define P2P_STATUS_INVALID_USER_PWD      0x7     //用户名或密码错误
```

MSG_TYPE_P2P_MODE

描述：P2P 连接模式
发送：不支持
通知：支持
消息内容：int
消息长度：sizeof(int)
连接模式定义：

```
#define P2P_MODE_P2P_RELAY          0x0    //通过转发服务器连接
#define P2P_MODE_P2P_CONNECTED      0x1    //P2P 连接
```

MSG_TYPE_GET_CAMERA_PARAMS

描述：获取视频参数
发送：支持
消息内容：NULL
消息长度：0
通知：支持
消息内容：

```
typedef struct tag_STRU_CAMERA_PARAMS
{
    int resolution;
    int brightness;
    int contrast;
    int hue;
    int saturation;
    int flip;
}STRU_CAMERA_PARAMS,*PSTRU_CAMERA_PARAMS;
```

消息长度：sizeof(STRU_CAMERA_PARAMS)
消息内容说明：

resolution: 分辨率 0: 640*480 1: 320*240
brightness: 亮度 1~255
contrast: 对比度 1~255
hue: 色度，不支持
saturation: 饱和度，不支持

MSG_TYPE_DECODER_CONTROL

描述：云台控制
发送：支持
消息内容：NULL
消息长度：0
通知：支持
消息内容：int
消息长度：sizeof(int)
消息内容说明：

```
#define CMD_PTZ_UP                  0 //云台向上
#define CMD_PTZ_UP_STOP             1 //云台向上停止
#define CMD_PTZ_DOWN                2 //云台向下
#define CMD_PTZ_DOWN_STOP           3 //云台向下停止
#define CMD_PTZ_LEFT                4 //云台向左
#define CMD_PTZ_LEFT_STOP           5 //云台向左停止
#define CMD_PTZ_RIGHT               6 //云台向右
#define CMD_PTZ_RIGHT_STOP          7 //云台向右停止

#define CMD_PTZ_CENTER              25 //云台居中
#define CMD_PTZ_UP_DOWN             26 //云台上下巡航
#define CMD_PTZ_UP_DOWN_STOP        27 //云台上下巡航停止
#define CMD_PTZ_LEFT_RIGHT          28 //云台左右巡航
#define CMD_PTZ_LEFT_RIGHT_STOP     29 //云台左右巡航停止

#define CMD_PTZ_PREFAB_BIT_SET0     30 //设置预置位 1
#define CMD_PTZ_PREFAB_BIT_SET1     32 //设置预置位 2
#define CMD_PTZ_PREFAB_BIT_SET2     34 //设置预置位 3
#define CMD_PTZ_PREFAB_BIT_SET3     36 //设置预置位 4
#define CMD_PTZ_PREFAB_BIT_SET4     38 //设置预置位 5
#define CMD_PTZ_PREFAB_BIT_SET5     40 //设置预置位 6
```

```
#define CMD_PTZ_PREFAB_BIT_SET6          42 //设置预置位 7
#define CMD_PTZ_PREFAB_BIT_SET7          44 //设置预置位 8
#define CMD_PTZ_PREFAB_BIT_SET8          46 //设置预置位 9
#define CMD_PTZ_PREFAB_BIT_SET9          48 //设置预置位 10
#define CMD_PTZ_PREFAB_BIT_SETA          50 //设置预置位 11
#define CMD_PTZ_PREFAB_BIT_SETB          52 //设置预置位 12
#define CMD_PTZ_PREFAB_BIT_SETC          54 //设置预置位 13
#define CMD_PTZ_PREFAB_BIT_SETD          56 //设置预置位 14
#define CMD_PTZ_PREFAB_BIT_SETE          58 //设置预置位 15
#define CMD_PTZ_PREFAB_BIT_SETF          60 //设置预置位 16

#define CMD_PTZ_PREFAB_BIT_RUN0          31 //调用预置位 1
#define CMD_PTZ_PREFAB_BIT_RUN1          33 //调用预置位 2
#define CMD_PTZ_PREFAB_BIT_RUN2          35 //调用预置位 3
#define CMD_PTZ_PREFAB_BIT_RUN3          37 //调用预置位 4
#define CMD_PTZ_PREFAB_BIT_RUN4          39 //调用预置位 5
#define CMD_PTZ_PREFAB_BIT_RUN5          41 //调用预置位 6
#define CMD_PTZ_PREFAB_BIT_RUN6          43 //调用预置位 7
#define CMD_PTZ_PREFAB_BIT_RUN7          45 //调用预置位 8
#define CMD_PTZ_PREFAB_BIT_RUN8          47 //调用预置位 9
#define CMD_PTZ_PREFAB_BIT_RUN9          49 //调用预置位 10
#define CMD_PTZ_PREFAB_BIT_RUNA          51 //调用预置位 11
#define CMD_PTZ_PREFAB_BIT_RUNB          53 //调用预置位 12
#define CMD_PTZ_PREFAB_BIT_RUNC          55 //调用预置位 13
#define CMD_PTZ_PREFAB_BIT_RUND          57 //调用预置位 14
#define CMD_PTZ_PREFAB_BIT_RUNE          59 //调用预置位 15
#define CMD_PTZ_PREFAB_BIT_RUNF          61 //调用预置位 16
```

MSG_TYPE_GET_PARAMS

描述：获取参数（包含网络参数，无线网络参数，用户信息， FTP 参数，邮件参数，DDNS 参数，时间参数）

发送：支持

消息内容：NULL

消息长度：0

通知：不支持

NOTE: 请求的结果将通过 MSG_TYPE_WIFI_PARAMS,MSG_TYPE_USER_INFO,MSG_TYPE_FTP_PARAMS,MSG_TYPE_MAIL_PARAMS,MSG_TYPE_DDNS_PARAMS,MSG_TYPE_DATETIME_PARAMS 消息返回

MSG_TYPE_SNAPSHOT

描述：抓取一张 JPEG 图片

发送：支持

消息内容：NULL

消息长度：0

通知：支持

消息内容：一张 JPEG 图片

消息长度：实际的 JPEG 图片长度

MSG_TYPE_CAMERA_CONTROL

描述：视频参数调节

发送：支持

消息内容：

```
typedef struct tag_STRU_CAMERA_CONTROL
{
    int param;
    int value;
}STRU_CAMERA_CONTROL,*PSTRU_CAMERA_CONTROL;
```

消息长度：0

消息内容说明：

param	value
0：分辨率	0:VGA 1:QVGA
1：亮度	1~255
2：对比度	1~255
3：模式	0：50hz
	1：60hz
5：旋转	0：原始
	1：垂直翻转
	2：水平镜像
	3：垂直翻转 + 水平镜像

通知：支持

消息内容：int

消息长度：sizeof(int)

消息内容说明：0: 失败 1: 成功

MSG_TYPE_SET_NETWORK

描述：设置网络参数

发送：支持

消息内容：

```
typedef struct tag_STRU_NETWORK_PARAMS
{
    char ipaddr[64];
    char netmask[64];
    char gateway[64];
    char dns1[64];
    char dns2[64];
    int dhcp;
    int port;
    int rtspport;
}STRU_NETWORK_PARAMS,*PSTRU_NETWORK_PARAMS;
```

消息长度：sizeof(STRU_NETWORK_PARAMS)

消息内容说明：

ipaddr: ip 地址

netmask: 子网掩码

gateway: 网关

dns1: dns 服务器 1，默认 8.8.8.8 暂不支持修改

dns2: dns 服务器 2

dhcp: 是否开启 dhcp 1: 开启 0: 不开启

port: http 端口

rtspport: rtsp 端口，不支持

通知：支持

消息内容：int

消息长度：sizeof(int)

消息内容说明：0: 失败 1: 成功

MSG_TYPE_REBOOT_DEVICE

描述：重启摄像机
发送：支持
　　消息内容：NULL
　　消息长度：0
通知：不支持

MSG_TYPE_RESTORE_FACTORY

描述：恢复出厂设置
发送：支持
　　消息内容：NULL
　　消息长度：0
通知：不支持

MSG_TYPE_SET_USER

描述：设置用户账户
发送：支持
　　消息内容：

```
typedef struct tag_STRU_USER_INFO
{
    char user1[64];
    char pwd1[64];
    char user2[64];
    char pwd2[64];
    char user3[64];
    char pwd3[64];
}STRU_USER_INFO,*PSTRU_USER_INFO;
```

　　消息长度：sizeof(STRU_USER_INFO)
　　消息内容说明：
　　user1: 参观者用户名
　　pwd1: 参观者密码
　　user2: 操作者用户名
　　pwd2: 操作者密码

user3: 管理员用户名

pwd3: 管理员密码

通知：支持

消息内容：int

消息长度：sizeof(int)

消息内容说明：0: 失败 1: 成功

NOTE: 1、三个用户名不能相同，参观者和操作者的用户名和密码可以为空（即没有这两个账户），管理员的账户不能为空。2、设置后，需要重启设备

MSG_TYPE_SET_WIFI

描述：配置 WIFI 参数

发送：支持

消息内容：

```
typedef struct tag_STRU_WIFI_PARAMS
{
    int enable;
    char ssid[128];
    int channel;
    int mode;
    int authtype;
    int encrypt;
    int keyformat;
    int defkey;
    char key1[128];
    char key2[128];
    char key3[128];
    char key4[128];
    int key1_bits;
    int key2_bits;
    int key3_bits;
    int key4_bits;
    char wpa_psk[128];
}STRU_WIFI_PARAMS,*PSTRU_WIFI_PARAMS;
```

消息长度：sizeof(STRU_WIFI_PARAMS)

消息内容说明：

enable: 是否启用 wifi，1: 启用 0: 不启用

ssid[128]: ssid

channel: 通道号

mode: wifi 模式，填 0

authtype: wifi 认证类型 0: 无 1: WEP 2: WPA-PSK(AES) 2: WPA-PSK(TKIP) 3: WPA2-PSK(AES) 4: WPA2-PSK(TKIP)

encryp: 安全模式 0: 开放系统 1: 共享密钥

keyforma: 密钥格式 0: 16 进制数 1: ASCII 码

defke: 缺省使用密钥 0~3

key1[128]: 密钥 1

key2[128]: 密钥 2

key3[128]: 密钥 3

key4[128]: 密钥 4

key1_bits: 密钥 1 的长度 0: 64bits 1: 128bits

key2_bits: 密钥 2 的长度 0: 64bits 1: 128bits

key3_bits: 密钥 3 的长度 0: 64bits 1: 128bits

key4_bits: 密钥 4 的长度 0: 64bits 1: 128bits

wpa_psk[128]: wpa 密码

通知：支持

消息内容：int

消息长度：sizeof(int)

消息内容说明：0: 失败 1: 成功

NOTE: 设置后需要重启设备

MSG_TYPE_SET_DATETIME

描述：设置时间

发送：支持

消息内容：

```
typedef struct tag_STRU_DATETIME_PARAMS
{
    int now;
    int tz;
    int ntp_enable;
    char ntp_svr[64];
}STRU_DATETIME_PARAMS,*PSTRU_DATETIME_PARAMS;
```

消息长度：sizeof(STRU_DATETIME_PARAMS)

消息内容说明：

now: 当前时间（从 1970-1-1 0:0:0 到指定的时间所逝去的秒数），如果不设置当前时间值，则填 0

tz: 时区，可包含以下取值：

39600：(GMT -11:00) 中途岛，萨摩亚群岛

36000：(GMT -10:00) 夏威夷

32400：(GMT -09:00) 阿拉斯加

28800：(GMT -08:00) 太平洋时间(美国和加拿大)

25200：(GMT -07:00) 山地时间(美国和加拿大)

21600：(GMT -06:00) 中部时间(美国和加拿大)，墨西哥城

18000：(GMT -05:00) 东部时间(美国和加拿大)，利马，波哥大

14400：(GMT -04:00) 大西洋时间(加拿大)，圣地亚哥，拉巴斯

12600：(GMT -03:30) 纽芬兰

10800>(GMT -03:00) 巴西利亚，布宜诺斯艾丽斯，乔治敦

7200>(GMT -02:00) 中大西洋

3600>(GMT -01:00) 佛得角群岛

0>(GMT) 格林威治平时；伦敦，里斯本，卡萨布兰卡

-3600：(GMT +01:00) 布鲁赛尔，巴黎，柏林，罗马，马德里，斯多哥尔摩，贝尔格莱德，布拉格

-7200：GMT +02:00) 雅典，耶路撒冷，开罗，赫尔辛基

-10800：(GMT +03:00) 内罗毕，利雅得，莫斯科

-12600：(GMT +03:30) 德黑兰

-14400：(GMT +04:00) 巴库，第比利斯，阿布扎比，马斯科特

-16200：(GMT +04:30) 科布尔

-18000：(GMT +05:00) 伊斯兰堡，卡拉奇，塔森干

-19800：(GMT +05:30) 加尔各答，孟买，马德拉斯，新德里

-21600：(GMT +06:00) 阿拉木图，新西伯利亚，阿斯塔南，达尔

-25200：(GMT +07:00) 曼谷，河内，雅加达

-28800：(GMT +08:00) 北京，新加坡，台北

-32400：(GMT +09:00) 首尔，雅库茨克，东京

-34200：(GMT +09:30) 达尔文

-36000：(GMT +10:00) 关岛，，墨尔本，悉尼，莫尔兹比港，符拉迪沃斯托克

-39600：(GMT +11:00) 马加丹，所罗门群岛，新喀里多尼亚

-43200：(GMT +12:00) 奥克兰，惠灵顿，斐济

ntp_enable: 是否启用 ntp 服务 1：启用 0：不启用

ntp_svr: ntp 服务器地址

通知：支持

消息内容：int

消息长度：sizeof(int)
消息内容说明：0: 失败 1: 成功

MSG_TYPE_GET_STATUS

描述：获取当前设备的状态
发送：支持
消息内容：NULL
消息长度：0
通知：支持
消息内容：

```
typedef struct tag_STRU_CAMERA_STATUS
{
    char sysver[32];
    char devname[96];
    char devid[32];
    int alarmstatus;
    int sdcardstatus;
    int sdcardtotalsize;
    int sdcardremainsize;
    char mac[32];
    char wifimac[32];
        int dns_status;
        int upnp_status;
}STRU_CAMERA_STATUS,*PSTRU_CAMERA_STATUS;
```

消息长度：sizeof(STRU_CAMERA_STATUS)
消息内容说明：
sysver: 系统固件的版本号
devname: 设备名称，使用 utf-8 编码
devid: 设备 ID
alaramstatus: 报警状态
sdcardstatus: SD 卡插入状态 0: 未插入 1: 插入
sdcardtotalsize: SD 总容量
sdcardremainsize: SD 剩余容量
dns_status: dns 状态
upnp_status: upnp 状态

MSG_TYPE_GET_PTZ_PARAMS

描述：获取 PTZ 配置信息

发送：支持

消息内容：NULL

消息长度：0

通知：支持

消息内容：

```
typedef struct tag_STRU_PTZ_PARAMS
{
    int led_mode;
    int ptz_center_onstart;
    int ptz_run_times;
    int ptz_patrol_rate;
    int ptz_patrol_up_rate;
    int ptz_patrol_down_rate;
    int ptz_patrol_left_rate;
    int ptz_patrol_right_rate;
    int disable_preset;
    int ptz_preset;
}STRU_PTZ_PARAMS,*PSTRU_PTZ_PARAMS;
```

消息长度：sizeof(STRU_PTZ_PARAMS)

消息内容说明：

led_mode: led 指示灯状态 0：关闭 1：打开 （某些版本可用）

ptz_center_onstart: 启动时云台自动居中 0：不居中 1：居中

ptz_run_times: 云台在上下巡航，左右巡航时的巡航圈数 0：表示无限制（但是系统会限制最长时间为 1 小时）

ptz_patrol_rate: 云台上下巡航，左右巡航的速度，取值 1~10

ptz_patrol_up_rate: 云台向上转动速度，取值 1~10

ptz_patrol_down_rate: 云台向下转动速度，取值 1~10

ptz_patrol_left_rate: 云台向左转动速度，取值 1~10

ptz_patrol_right_rate: 云台向右转动速度，取值 1~10

disable_preset: 禁用预置位 0：不禁用 1： 禁用

ptz_preset: 启动时，对准预置位 取值 0~16，0 表示不对准预置位 1~16 分别对应预置位 1 至预置位 16

MSG_TYPE_SET_DDNS

不支持

MSG_TYPE_SET_MAIL

描述：设置邮件参数
发送：支持
消息内容：

```
typedef struct tag_STRU_MAIL_PARAMS
{
    char svr[64];
    char user[64];
    char pwd[64];
    char sender[64];
    char receiver1[64];
    char receiver2[64];
    char receiver3[64];
    char receiver4[64];
    int port;
    int ssl;
}STRU_MAIL_PARAMS,*PSTRU_MAIL_PARAMS;
```

消息长度：sizeof(STRU_MAIL_PARAMS)
消息内容说明：
svr: SMTP 服务器地址
user: 邮件用户名
pwd: 邮件密码
sender: 发送者邮箱地址
receiver1: 接收者 1 的邮箱地址
receiver2: 接收者 2 的邮箱地址
receiver3: 接收者 3 的邮箱地址
receiver4: 接收者 4 的邮箱地址
port: SMTP 服务器的端口
ssl: 邮件加密方式 0: 不加密 1: SSL 2: TLS
通知：支持
消息内容：int
消息长度：sizeof(int)
消息内容说明：0: 失败 1: 成功

MSG_TYPE_SET_FTP

描述：设置 FTP 参数
发送：支持
消息内容：
typedef struct tag_STRU_FTP_PARAMS
{
char svr_ftp[64];
char user[64];
char pwd[64];
char dir[128];
int port;
int mode;
int upload_interval;
}STRU_FTP_PARAMS,*PSTRU_FTP_PARAMS;
消息长度：sizeof(STRU_FTP_PARAMS)
消息内容说明：
svr_ftp: ftp 服务器的地址
user: ftp 登录用户名
pwd: ftp 登录密码
dir: ftp 上传目录，填 / (即根目录)
port: ftp 服务器端口（默认 21）
mode: 上传模式 0: PORT 1: PASV
upload_interval: ftp 定时上传的时间间隔（单位秒） 0 表示不定时上传
通知：支持
消息内容：int
消息长度：sizeof(int)
消息内容说明：0: 失败 1: 成功

MSG_TYPE_SET_ALARM

描述：报警设置
发送：支持
消息内容：
typedef struct tag_STRU_ALARM_PARAMS

```
{
    int motion_armed;
    int motion_sensitivity;
    int input_armed;
    int ioin_level;
    int iolinkage;
    int ioout_level;
    int alarmpresetsit;
    int mail;
    int snapshot;
    int record;
    int upload_interval;
    int schedule_enable;
    int schedule_sun_0;
    int schedule_sun_1;
    int schedule_sun_2;
    int schedule_mon_0;
    int schedule_mon_1;
    int schedule_mon_2;
    int schedule_tue_0;
    int schedule_tue_1;
    int schedule_tue_2;
    int schedule_wed_0;
    int schedule_wed_1;
    int schedule_wed_2;
    int schedule_thu_0;
    int schedule_thu_1;
    int schedule_thu_2;
    int schedule_fri_0;
    int schedule_fri_1;
    int schedule_fri_2;
    int schedule_sat_0;
    int schedule_sat_1;
    int schedule_sat_2;
}STRU_ALARM_PARAMS,*PSTRU_ALARM_PARAMS
```

消息长度：sizeof(STRU_ALARM_PARAMS)

消息内容说明：

motion_armed: 移动侦测开关 0: 关闭 1: 打开

motion_sensitivity: 移动侦测灵敏度 取值 1~10 （取值越小越灵敏）

input_armed: 输入报警开关 0: 关闭 1: 打开

ioin_level: 输入报警触发电平 0: 低电平 1: 高电平

iolinkage: 报警 IO 输出联动开关 0: 关闭 1: 打开

ioout_level: 报警输出电平 0: 低电平 1: 高电平

alarmpresetsit: 报警调用预置位 取值 0~16 0 表示不调用预置位

mail: 报警发送邮件 0: 关闭 1: 打开

snapshot: 报警抓图 0: 关闭 1: 打开 （目前不支持）

record: 报警录像 0: 关闭 1: 打开

upload_interval: 报警 ftp 上传时间间隔(单位秒) 0 表示不上传

schedule_enable: 报警布防开关 0: 关闭 1: 打开 （注意：关闭布防，将不会触发任何报警）

schedule_sun_0: 每天按 24 小时，每小时按 15 分钟划分为 96 个布防时段，使用 3 个整形表示，每个整形 32 位分别表示 1 个时段，如果该位为 0：该时段不布防；1：该时段布防

schedule_sun_1: 同上

schedule_sun_2: 同上

schedule_mon_0: 同上

schedule_mon_1: 同上

schedule_mon_2: 同上

schedule_tue_0: 同上

schedule_tue_1: 同上

schedule_tue_2: 同上

schedule_wed_0: 同上

schedule_wed_1: 同上

schedule_wed_2: 同上

schedule_thu_0: 同上

schedule_thu_1: 同上

schedule_thu_2: 同上

schedule_fri_0: 同上

schedule_fri_1: 同上

schedule_fri_2: 同上

schedule_sat_0: 同上

schedule_sat_1: 同上

schedule_sat_2: 同上

通知：支持

消息内容：int

消息长度：sizeof(int)

消息内容说明：0: 失败 1: 成功

MSG_TYPE_SET_PTZ

描述：设置 PTZ 参数

发送：支持

消息内容：

```
typedef struct tag_STRU_PTZ_PARAMS
{
    int led_mode;
    int ptz_center_onstart;
    int ptz_run_times;
    int ptz_patrol_rate;
    int ptz_patrol_up_rate;
    int ptz_patrol_down_rate;
    int ptz_patrol_left_rate;
    int ptz_patrol_right_rate;
    int disable_preset;
    int ptz_preset;
}STRU_PTZ_PARAMS,*PSTRU_PTZ_PARAMS;
```

消息长度：sizeof(STRU_PTZ_PARAMS)

消息内容定义：请参考 MSG_TYPE_GET_PTZ_PARAMS 的说明

通知：支持

消息内容：int

消息长度：sizeof(int)

消息内容说明：0: 失败 1: 成功

MSG_TYPE_WIFI_SCAN

描述：扫描 WIFI

发送：支持

消息内容：NULL

消息长度：0

通知：支持

消息内容：

```
typedef struct tag_STRU_WIFI_SEARCH_RESULT_LIST
{
    int nResultCount;
    STRU_WIFI_SEARCH_RESULT wifi[50];
}STRU_WIFI_SEARCH_RESULT_LIST,*PSTRU_WIFI_SEARCH_RESULT_LIST;
```

消息长度：sizeof(STRU_WIFI_SEARCH_RESULT_LIST)

消息内容说明：

nResultCount: WIFI 扫描结果个数

```
typedef struct tag_STRU_WIFI_SEARCH_RESULT
{
    char ssid[64]; //网络的 ssid
    char mac[64]; //wifi 的 mac 地址
    int security; //加密方式  0: 不加密  1: WEP 2: WPA-PSK(AES) 3: WPA-PSK(TKIP) 4:
                  WPA2-PSK(AES) 5: WPA2-PSK(TKIP)
    char dbm0[32];   //WIFI 信号强度 以 100 为最强
    char   dbm1[32]; //WIFI 基准值 ，暂时忽略改参数
    int mode; //wifi 模式
    int channel; //wifi 通道号
}STRU_WIFI_SEARCH_RESULT,*PSTRU_WIFI_SEARCH_RESULT;
```

MSG_TYPE_GET_ALARM_LOG

不支持

MSG_TYPE_GET_RECORD

不支持

MSG_TYPE_GET_RECORD_FILE

描述：获取 TF 卡录像文件

发送：支持

消息内容：STRU_GET_RECORD_FILE_PARAM

消息长度：sizeof(STRU_GET_RECORD_FILE_PARAM)

通知：支持

消息内容：STRU_SDCARD_RECORD_FILE

消息内容长度：sizeof(STRU_SDCARD_RECORD_FILE)

NOTE:消息结构定义请参考头文件

MSG_TYPE_SET_PPPOE

不支持

MSG_TYPE_SET_UPNP

不支持

MSG_TYPE_DEL_RECORD_FILE

不支持

MSG_TYPE_SET_MEDIA

不支持

MSG_TYPE_SET_RECORD_SCH

不支持

MSG_TYPE_CLEAR_ALARM_LOG

不支持

MSG_TYPE_WIFI_PARAMS

描述：WIFI 参数
发送：不支持
通知：支持

消息内容：

```
typedef struct tag_STRU_WIFI_PARAMS
{
    int enable;
    char ssid[128];
    int channel;
    int mode;
    int authtype;
    int encrypt;
    int keyformat;
    int defkey;
    char key1[128];
    char key2[128];
    char key3[128];
    char key4[128];
    int key1_bits;
    int key2_bits;
    int key3_bits;
    int key4_bits;
    char wpa_psk[128];
}STRU_WIFI_PARAMS,*PSTRU_WIFI_PARAMS;
```

消息长度：sizeof(STRU_WIFI_PARAMS)
消息内容说明：请参考 MSG_TYPE_SET_WIFI 的说明

MSG_TYPE_MAIL_PARAMS

描述：邮件参数
发送：不支持
通知：支持

消息内容：

```
typedef struct tag_STRU_MAIL_PARAMS
```

```
{
    char svr[64];
    char user[64];
    char pwd[64];
    char sender[64];
    char receiver1[64];
    char receiver2[64];
    char receiver3[64];
    char receiver4[64];
    int port;
    int ssl;
}STRU_MAIL_PARAMS,*PSTRU_MAIL_PARAMS;
```

消息长度：sizeof(STRU_MAIL_PARAMS)

消息内容定义：请参考 MSG_TYPE_SET_MAIL 的说明

MSG_TYPE_FTP_PARAMS

描述：FTP 参数

发送：不支持

通知：支持

消息内容：

```
typedef struct tag_STRU_FTP_PARAMS
{
    char svr_ftp[64];
    char user[64];
    char pwd[64];
    char dir[128];
    int port;
    int mode;
    int upload_interval;
}STRU_FTP_PARAMS,*PSTRU_FTP_PARAMS;
```

消息长度：sizeof(STRU_FTP_PARAMS)

消息内容定义：请参考 MSG_TYPE_SET_FTP 的说明

MSG_TYPE_NETWORK_PARAMS

描述：网络参数
发送：不支持
通知：支持

消息内容：

```
typedef struct tag_STRU_NETWORK_PARAMS
{
    char ipaddr[64];
    char netmask[64];
    char gateway[64];
    char dns1[64];
    char dns2[64];
    int dhcp;
    int port;
    int rtspport;
}STRU_NETWORK_PARAMS,*PSTRU_NETWORK_PARAMS;
```

消息长度：sizeof(STRU_NETWORK_PARAMS)
消息内容说明：请参考 MSG_TYPE_SET_NETWORK 的说明

MSG_TYPE_USER_INFO

描述：用户信息
发送：不支持
通知：支持

消息内容：

```
typedef struct tag_STRU_USER_INFO
{
    char user1[64];
    char pwd1[64];
    char user2[64];
    char pwd2[64];
    char user3[64];
    char pwd3[64];
}STRU_USER_INFO,*PSTRU_USER_INFO;
```

消息长度：sizeof(STRU_USER_INFO)
消息内容说明：请参考 MSG_TYPE_SET_USER 的的说明

MSG_TYPE_DDNS_PARAMS

不支持

MSG_TYPE_DATETIME_PARAMS

描述：时间参数
发送：不支持
通知：支持

消息内容：

```
typedef struct tag_STRU_DATETIME_PARAMS
{
    int now;
    int tz;
    int ntp_enable;
    char ntp_svr[64];
}STRU_DATETIME_PARAMS,*PSTRU_DATETIME_PARAMS;
```

消息长度：sizeof(STRU_DATETIME_PARAMS)
消息内容说明：请参考 MSG_TYPE_SET_DATETIME 的说明

MSG_TYPE_ALARM_PARAMS

描述：报警参数
发送：不支持
通知：支持

消息内容：

```
typedef struct tag_STRU_ALARM_PARAMS
{
    int motion_armed;
    int motion_sensitivity;
    int input_armed;
    int ioin_level;
```

```
    int iolinkage;
    int ioout_level;
    int alarmpresetsit;
    int mail;
    int snapshot;
    int record;
    int upload_interval;
    int schedule_enable;
    int schedule_sun_0;
    int schedule_sun_1;
    int schedule_sun_2;
    int schedule_mon_0;
    int schedule_mon_1;
    int schedule_mon_2;
    int schedule_tue_0;
    int schedule_tue_1;
    int schedule_tue_2;
    int schedule_wed_0;
    int schedule_wed_1;
    int schedule_wed_2;
    int schedule_thu_0;
    int schedule_thu_1;
    int schedule_thu_2;
    int schedule_fri_0;
    int schedule_fri_1;
    int schedule_fri_2;
    int schedule_sat_0;
    int schedule_sat_1;
    int schedule_sat_2;
}STRU_ALARM_PARAMS,*PSTRU_ALARM_PARAMS
```

消息长度：sizeof(STRU_ALARM_PARAMS)

消息内容定义：请参考 MSG_TYPE_SET_ALARM 的说明

MSG_TYPE_ALARM_NOTIFY

描述：报警状态通知

发送：不支持

通知：支持

消息内容：int

消息内容长度：sizeof(int)

报警状态说明： 1: 移动侦测报警 2: GPIO 报警

MSG_TYPE_SET_DEVNAME

描述：设置摄像机名称

发送：支持

消息内容： utf8 字符串，长度不超过 80

通知：支持

消息内容：int

消息内容长度：sizeof(int)

消息内容说明：0: 失败 1: 成功

MSG_TYPE_PLAY_BACK_START

描述：开始回放 TF 卡的某个录像文件

发送：支持

消息内容： STRU_PLAY_BACK_PARAM

消息内容长度：sizeof(STRU_PLAY_BACK_PARAM)

通知：不支持

MSG_TYPE_PLAY_BACK_STOP

描述：停止 TF 卡录像回放

发送：支持

消息内容： NULL

通知：不支持