

报警联动摄像机 CGI 文档

一、新增 CGI

1、获取传感器状态

CGI: /get_sensorstatus.cgi?loginuse=admin&loginpas=888888&user=admin&pwd=888888&

返回值: cmd: 0x60b2 数据:

armsetstatus=1; (0:撤防; 1:布防)

alarmstatus=0; (0:没有报警 1:报警)

codestatus=0; (0:关闭对码 1:开启对码)

doorbell=0; (1: 开启 0: 关闭)

ptzspeed = 0; (0-10 速度越大云台转动越快)

2、获取传感器列表

CGI: GET

/get_sensorlist.cgi?loginuse=admin&loginpas=888888&user=admin&pwd=888888&

返回值: cmd: 0x60b6 数据: (说明: 若id1/id2/id3 全为 0 或者全为 255 表示此传感器无效; CGI中所传递的sensorid为通道号(下面的既为: 0); 传感器用来显示的ID为id1/id2/id3 转换成 16 进制然后拼接成字符串(下面既为: 01F0FF))

sensorid1[0]=1;//01

sensorid2[0]=16;//F0

sensorid3[0]=255;//FF

sensortype[0]=0x01; (

typedef enum {

SENSOR_TYPE_DOOR = 0x01, //门磁

SENSOR_TYPE_INFRARED = 0x02, //红外

SENSOR_TYPE_SMOKE = 0x03, //烟感

SENSOR_TYPE_SMELL = 0x04, //气感

SENSOR_TYPE_REMOTE = 0x07, //遥控

SENSOR_TYPE_CAMERA = 0x0A, //摄像头

SENSOR_TYPE_CURTAIN = 0x0B//幕帘

} SENSORTYPE;

)

sensorstatus[0]=0;

presetid[0]=0;//与传感器联动的预置位ID (1-16)

sensorname[0]="gg";//传感器名字

3、获取报警日志

CGI:

/get_alarmlog.cgi?logid=0&sensorid=1&loginuse=admin&loginpas=888888&user=admin&pwd=888888&

返回值: cmd: 0x60b5 数据:

ncount=100;//报警日志总数

armtype=0;//报警类型 (1:移动侦测 2:GPIO 3:传感器报警)

alarmdevsname[0]="";//报警设备名字

time[0]=1379146962;//报警时间

dvstype[0]=7;//报警设备类型
actiontype[0]=6;//报警动作类型

4、删除传感器

CGI:/del_sensor.cgi?sensorid=31&loginuse=admin&loginpas=888888&user=admin&pwd=888888&

参数: sensorid(既为 2 中所提到的索引号范围是: 0-31)

返回值: cmd: 0x60b3 数据: result=0;

5、编辑传感器

CGI: GET

/set_sensorname.cgi?sensorid=30&sensorname=ggg&loginuse=admin&loginpas=888888&user=admin&pwd=888888&

参数: sensorid: 同上 sensorname: 传感器名字

返回值: cmd: 0x60b4 数据: result=0;

6、设置预置位

CGI:

/set_sensor_preset.cgi?sensorid=30&presetid=2&loginuse=admin&loginpas=888888&user=admin&pwd=888888&

参数: presetid

: (0-16) 0:表示不绑定预制位 1-16:分别对应各个预制位 sensorid: 通道号(特殊情况当sensorid=255 为设置摄像机的看守位)

7、获取绑定的预置位

CGI:

/get_sensor_preset.cgi?cmd=0&sensorid=0&loginuse=admin&loginpas=888888&user=admin&pwd=888888&

参数: cmd (0 表示获取对应sensorid绑定的预置位, 1 表示获取摄像机的看守位(此时sensorid无效)), sensorid (cmd=0 时值为传感器的通道号, cmd=1 时无意义)

返回值:

cmd=0; //这个cmd即为发CGI时所传的cmd参数

presetid=0; //presetid为绑定的预置位或者看守位

sensorid=0; //sensorid (当cmd=0 时表示传感器的通道号 (0-31), cmd=1 时无意义)

8: 设置传感器状态

CGI: GET /set_sensorstatus.cgi?cmd=0& doorbell =0&

ptzspeed=0&loginuse=admin&loginpas=888888&user=admin&pwd=888888&

参数: cmd: 0: 布防 1:撤防 2:对码 3:取消对码 4: 设置门铃状态(doorbell = 1 开启, 0: 关闭) 5: 设置发生报警时云台转速 (ptzspeed = 0-10 越大速度越快)

说明: doorbell: cmd = 4 时才有效

ptzspeed: cmd=5 时才有效

返回值: result =0: 0 成功 -1 失败

cmd=0: 0: 布防 1:撤防 2:对码 3:取消对码 4: 设置门铃状态 5: 设置发生报警时云台转速

9:CGI: CGI: GET

/edit.whitelist.cgi?cmd=0&whitelistname=qqaazz&whitelistpass=888888&editwhitelistid=0&loginuse=admin&loginpas=888888&user=admin&pwd=888888&

cmd: 0 增加白名单成员 1: 删除白名单成员(editwhitelistid 只有在cmd=1 cmd = 4 cmd = 5时才有效, 取值为0-15表示要删除的各个白名单 0:是超级管理员权限不能删除)

2: 获取白名单列表 3: 删除所有白名单成员 4: 修改白名单密码 5: 转交管理员权限(editwhitelistid 表示需要转交白名单ID)

whitelistname: (cmd = 0 cmd = 4 时表示需要操作的白名单 cmd = 5 cmd = 3 cmd = 2 cmd = 1 时此参数无意义)

返回值定义:

cmd = 0; 0 增加白名单成员 1: 删除白名单成员 2: 获取白名单列表 3: 删除所有白名单成员 4: 修改白名单密码 (cmd = 0 cmd = 1 cmd = 3 cmd = 4 cmd = 5 则下面其他返回值无意义)

whitelistname[0] = "qqaazz";白名单成员的名字

10、新增获取报警抓图文件名的接口

GET

/get_alarmsnapshot_file.cgi?snapshotIndex=23&loginuse=admin&loginpas=&user=admin&pwd=&

参数: 无

返回值: //报警抓图最多: 50

result = 0; //0: 文件未获取完成 -1: 文件已获取完成

snapshotCount = 50; //文件总数

pagecount = 20; //本次返回的文件数量

snapshot_name[0] = "20131009085724_7_7_FF00EF_1.jpeg";

/*时间_传感器类型_报警类型_(传感器ID)_标示符*/

11:发生报警时返回的数据结构

```
typedef struct tag_STRU_SENSOR_ALARM_INFO{
```

int cmd; //1:推送报警信息 2:抓图 3:操作 : (说明: 抓图和操作不会附带传感器的信息) (1、3 通道返回的信息相同)

int SensorID[3]; //传感器ID (用于显示, CGI参数不使用这个值, 而用下面的传感器通道)

int LinkPreset; //联动的预制位

char SensorName[64]; //传感器名字

int Sensortype; //传感器类型/*typedef enum {

SENSOR_TYPE_DOOR = 0x01, //门磁

SENSOR_TYPE_INFRARED = 0x02, //红外

SENSOR_TYPE_SMOKE = 0x03, //烟感

SENSOR_TYPE_SMELL = 0x04, //气感

SENSOR_TYPE_REMOTE = 0x07, //遥控

SENSOR_TYPE_CAMERA = 0x0A, //摄像头

SENSOR_TYPE_CURTAIN = 0x0B //窗帘

} SENSORTYPE;*/

int SensorAction; //动作类别/*typedef enum {

SENSOR_ALARM_ACTION_NON = 0x00, //无动作

```

SENSOR_ALARM_ACTION_ALARM = 0x01, //告警
SENSOR_ALARM_ACTION_LOWBATT = 0x02, //低电
SENSOR_ALARM_ACTION_GARRISON = 0x05, //布防
SENSOR_ALARM_ACTION_CANCELGARRISON = 0x06, //撤防
SENSOR_ALARM_ACTION_SOS = 0x07, //紧急报警SOS
SENSOR_ALARM_ACTION_OPENCODE = 0x08, //对码
SENSOR_ALARM_ACTION_DOORBELL = 0x0A, //门铃
SENSOR_ALARM_ACTION_OPEN = 0x0B, //返现新的传感器（在开启对码状态并且cmd=1时表示搜索到新的传感器）
SENSOR_ALARM_ACTION_CLOSE = 0x0C, //关闭
SENSOR_ALARM_ACTION_CAMGROUP = 0x0D, //摄像机群组
SENSOR_ALARM_ACTION_CANCELALARM = 0x0E, //取消报警
SENSOR_ALARM_ACTION_ADDNEWSSENSOR = 0x0F, //已保存
SENSOR_ALARM_ACTION_DELETEALL = 0x10, //删除全部传感器
SENSOR_ALARM_ACTION_EXITCODE = 0x11 //退出对码
} SENSORALARMACTION;*/
    int channel; //传感器通道（为上面所有对传感器操作的CGI的参数sensorid）
}STRU_SENSOR_ALARM_INFO, *PSTRU_SENSOR_ALARM_INFO;

```

二、在原来CGI基础上增加一些参数及返回值

1、CGI: GET/checkuser.cgi&loginuse=admin&loginpas=888888&user=admin&pwd=888888&
 用户-1 : 用户名错误 但是密码正确
 密码-2 : 用户名正确但是密码错误
 用户和密码-3 : 用户名和密码都错误

2、GET /get_status.cgi?loginuse=admin&loginpas=&user=admin&pwd=&
 在此CGI的返回值上添加一下几个变量
 alarm_module_version=16; //RF模块版本号

alarm_module_id[0]=0; //

alarm_module_id[1]=3; //

alarm_module_id[2]=214; //传感器ID

alarm_module_rebootnum=0; //RF模块重启次数

3、新增: SD卡录像: SD的fileName新增一位 20130915095712_100X.h264 (X:0 表示为以前的报警 X: 1 表示为传感器报警)

4、报警上报接口再新加一个参数: 报警抓图的文件名
 /alarm.asp的参数, 在原来的基础上增加&stype=1&sid1=12&sid2=12&sid3=23&sname="好好"
 "&salarmttype=2 &salarmsnapshotfile="20131012085838_7_7_30_ffff70_1.jpeg"&
 stype: 传感器类型
 sid1/sid2/sid3: 传感器ID
 sname: 传感器名字
 salarmttype: 传感器动作类型
 salarmsnapshotfile: 报警抓图的文件名

5、报警抓图通过 6 通道返回, GET/

livestream.cgi?streamid=5&filename=%s&offset=0&loginuse=admin&loginpas=888888&user=admin&pwd=888888&

数据结构:

HEAD+ (64 位) 图片文件名 + 图片数据

头部结构:

```
typedef struct tag_AV_HEAD
{
    unsigned int    startcode;    // 0xA815AA55
    char            type;          // 4
    char            streamid;      //
    unsigned short  militime;      //
    unsigned int    sectime;       //
    unsigned int    frameno;       //
    unsigned int    len;           // 数据长度

    unsigned char   version;       //
    unsigned char   sessid;
    unsigned char   other[2];
    unsigned char   other1[8];
} AV_HEAD, *PAV_HEAD;
```