|  |
| --- |
| 上海仰邦科技股份有限公司 |
| YQ系列控制器接口说明 |
| SDK编程指南 |

|  |
| --- |
| 2015-4-13 |

Copyright

All rights reserved. No part of this publication may be reproduced in any form by print, photo print, microfilm or any other means without written permission by ONBON.

© 2010-2014 Onbon

版本历史:

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 日期 | 作者 | 描述 |
| 1.0 | 2016-5-23 | 牛志敏 | 本手册适用于YQ1，YQ1-75，YQ2，YQ3，YQ4，YQ5，YQ2E控制器 |
| 1.1 | 2017-1-6 | 牛志敏 | 局域网模式添加节目锁定功能（[7.3.28](#_Net_LockProgram) [7.3.29](#_Net_UnLockProgram)） |

目录

[1. YQ系列控制器简介 5](#_Toc451777884)

[2. SDK简介 6](#_Toc451777885)

[2.1 局域网模式组网 6](#_Toc451777886)

[2.2 服务器模式组网 7](#_Toc451777887)

[2.3 云模式组网 7](#_Toc451777888)

[2.4 支持的功能 8](#_Toc451777889)

[2.5 文件兼容性 8](#_Toc451777890)

[3. 错误码及说明 9](#_Toc451777891)

[3.1 软件错误码 9](#_Toc451777892)

[3.2 设备错误码 9](#_Toc451777893)

[4. 函数概要说明 11](#_Toc451777894)

[5. 控制器说明 11](#_Toc451777895)

[6. 节目API 12](#_Toc451777896)

[6.1 API调用次序 12](#_Toc451777897)

[6.2 函数说明 12](#_Toc451777898)

[6.2.1 YQ\_CreatePlaylist 12](#_Toc451777899)

[6.2.2 YQ\_DestroyPlaylist 12](#_Toc451777900)

[6.2.3 YQ\_CreateProgram 13](#_Toc451777901)

[6.2.4 YQ\_CreateTimeArea 13](#_Toc451777902)

[6.2.5 YQ\_CreateCNTimeArea 14](#_Toc451777903)

[6.2.6 YQ\_CreateTimeCounterArea 15](#_Toc451777904)

[6.2.7 YQ\_CreateLunarArea 16](#_Toc451777905)

[6.2.8 YQ\_CreateClockArea 17](#_Toc451777906)

[6.2.9 YQ\_CreateClockArea 19](#_Toc451777907)

[6.2.10 YQ\_CreateVideoArea 20](#_Toc451777908)

[6.2.11 YQ\_CreatePicArea 20](#_Toc451777909)

[6.2.12 YQ\_VideoAreaAddUnit 20](#_Toc451777910)

[6.2.13 YQ\_PicAreaAddImageUnit 21](#_Toc451777911)

[6.2.14 YQ\_PicAreaAddRtfUnit 21](#_Toc451777912)

[6.2.15 YQ\_ProgramAddArea 21](#_Toc451777913)

[6.2.16 YQ\_PlaylistAddProgram 22](#_Toc451777914)

[6.2.17 YQ\_CreateDynamicArea 22](#_Toc451777915)

[6.2.18 YQ\_DestroyDynamic 23](#_Toc451777916)

[6.2.19 YQ\_DynamicAreaAddStrPage 23](#_Toc451777917)

[6.2.20 YQ\_DynamicAreaAddPicPage 23](#_Toc451777918)

[7. 局域网模式API 24](#_Toc451777919)

[7.1 API调用次序 24](#_Toc451777920)

[7.2 数据结构 25](#_Toc451777921)

[7.2.1 控制器单元定义 25](#_Toc451777922)

[7.3 函数说明 26](#_Toc451777923)

[7.3.1 Net\_SearchCards 26](#_Toc451777924)

[7.3.2 Net\_SetStaticip 26](#_Toc451777925)

[7.3.3 Net\_SetAutoip 26](#_Toc451777926)

[7.3.4 Net\_GetIp 26](#_Toc451777927)

[7.3.5 Net\_SetServerMode 27](#_Toc451777928)

[7.3.6 Net\_SetClientMode 27](#_Toc451777929)

[7.3.7 Net\_GetModeinfo 27](#_Toc451777930)

[7.3.8 Net\_GetNetinfo 27](#_Toc451777931)

[7.3.9 Net\_GetNetinfo 28](#_Toc451777932)

[7.3.10 Net\_GetFirmwareinfo 28](#_Toc451777933)

[7.3.11 Net\_GetBrightness 28](#_Toc451777934)

[7.3.12 Net\_GetVolume 29](#_Toc451777935)

[7.3.13 Net\_GetOnoff 29](#_Toc451777936)

[7.3.14 Net\_SetScreenSize 29](#_Toc451777937)

[7.3.15 Net\_SendProgram 29](#_Toc451777938)

[7.3.16 Net\_GetSendProcess 30](#_Toc451777939)

[7.3.17 Net\_CancelSend 30](#_Toc451777940)

[7.3.18 Net\_StopPlay 30](#_Toc451777941)

[7.3.19 Net\_SetVolume 30](#_Toc451777942)

[7.3.20 Net\_AdjustBrightness 31](#_Toc451777943)

[7.3.21 Net\_SystemTimeCorrect 31](#_Toc451777944)

[7.3.22 Net\_OpenScreen 31](#_Toc451777945)

[7.3.23 Net\_CloseScreen 32](#_Toc451777946)

[7.3.24 Net\_SwitchOnTime 32](#_Toc451777947)

[7.3.25 Net\_Update 32](#_Toc451777948)

[7.3.26 Net\_UpdateDynamicArea 32](#_Toc451777949)

[7.3.27 Net\_RemoveDynamicArea 33](#_Toc451777950)

7.3.28 Net\_LockProgrom............................................................................................33

7.3.29 Net\_UnLockProgram.......................................................................................33

[8. 局域网模式API 3](#_Toc451777919)3

[8.1 API调用次序 3](#_Toc451777920)3

[8.2 数据结构 3](#_Toc451777921)4

[8.2.1 控制器单元定义 3](#_Toc451777922)4

[8.3 函数说明 3](#_Toc451777923)4

[8.3.1 Server\_Start 3](#_Toc451777924)4

[8.3.2 Server\_SearchCards 3](#_Toc451777925)5

[8.3.3 Serve\_Closer 3](#_Toc451777926)5

[8.3.4 Serve\_GetModeinfo 3](#_Toc451777927)5

[8.3.5 Serve\_GetNetinfo 3](#_Toc451777928)5

[8.3.6 Serve\_GetScreenSize 3](#_Toc451777929)6

[8.3.7 Serve\_GetFirmwareinfo 3](#_Toc451777930)6

[8.3.8 Serve\_GetBrightness 3](#_Toc451777931)6

[8.3.9 Serve\_GetVolume 3](#_Toc451777932)7

[8.3.10 Serve\_GetOnoff 3](#_Toc451777933)7

[8.3.11 Serve\_SetScreenSize 3](#_Toc451777934)7

[8.3.12 Serve\_SendProgram 3](#_Toc451777935)7

[8.3.13 Serve\_GetSendProcess 3](#_Toc451777936)8

[8.3.14 Serve\_GetDownProcess 3](#_Toc451777937)8

[8.3.15 Serve\_CancelSend 3](#_Toc451777938)8

[8.3.16 Serve\_StopPlay 3](#_Toc451777939)9

[8.3.17 Serve\_SetVolume 3](#_Toc451777940)9

[8.3.18 Serve\_AdjustBrightness 3](#_Toc451777941)9

[8.3.19 Serve\_SystemTimeCorrect 4](#_Toc451777942)0

[8.3.20 Serve\_OpenScreen 4](#_Toc451777943)0

8[.3.21 Serve\_CloseScreen 4](#_Toc451777944)0

[8.3.22 Serve\_SwitchOnTime 4](#_Toc451777945)0

[8.3.23 Serve\_Update 4](#_Toc451777946)1

[8.3.24 Serve\_UpdateDynamicArea 4](#_Toc451777947)1

[8.3.25 Serve\_RemoveDynamicArea 4](#_Toc451777948)1

[8.3.26 Serve\_SaveDynamicArea 4](#_Toc451777949)1

[8.3.27 Serve\_DelDynamicArea 4](#_Toc451777950)2

YQ系列控制器接口说明

# YQ系列控制器简介

# SDK简介

通过网络连接控制器，本SDK支持3种模式：局域网模式、服务器模式、云模式。

## 局域网模式组网

所谓局域网模式就是指：使用网线直接连接控制器，或者和控制器在一个局域网（也可称为以太网）内。这种模式连接简单，知道控制器IP就可以连接。下图演示局域网模式组网结构：



使用网线直连控制器



通过局域网连接控制器

## 服务器模式组网

所谓服务器模式就是指：PC软件拥有一个公网IP，控制器都去主动连接PC软件，公网IP可以使用静态IP，或者通过在外部路由器上做端口映射获得。下图演示固定服务器模式组网结构：



## 云模式组网

由于固定IP模式只能限定一个局域网，但是实际应用时，控制器有可能接入另外一个局域网，与使用者不在同一个局域网下；也有可能接入3G或者4G网络；甚至有可能接入公网等等。这些情况下就必须使用云模式连接控制器，这时通过云服务，软件可以找到另一端的控制器，并传输数据。下图演示云模式适用的组网结构：



云模式组网

## 支持的功能

* 3种模式连接控制器
* 设置屏参
* 发送节目
* 获取控制器的各方面参数
* 设置网络参数和工作模式
* 调节音量
* 调节显示屏亮度
* 校时
* 开关屏幕
* 升级固件
* 更新动态区

## 文件兼容性

所有上传到控制器的文件，都必须符合ONBON控制器文件规范，否则可能引起未知的错误。

# 错误码及说明

错误码是函数原型过程中，产生的返回值。使用者可根据错误码做相应的处理。错误码分为2类：软件错误码，由SDK本身产生；设备错误码，由设备产生，并通过SDK反馈给客户。

## 软件错误码

|  |  |  |
| --- | --- | --- |
| 错误名称 | 代码 | 说明 |
| ERR\_Unknow | 0x0100 | 未知错误 |
| ERR\_TIMEOUT | 0x0101 | 通讯超时,可能网络故障 |
| ERR\_DiskSpace | 0x0102 | 控制器工作目录空间不足 |
| ERR\_Hand | 0x0103 | 未建立通讯连接 |
| ERR\_SCREENSIZE | 0x0104 | 屏幕宽或高不合理 |
| ERR\_PATH | 0x0105 | 错误的文件路径 |
| ERR\_CMD | 0x0106 | 错误的命令 |
| ERR\_PROPERTY | 0x0107 | 错误的属性类型 |
| ERR\_UPLOAD | 0x0108 | 上传失败 |
| ERR\_TASKEXIST | 0x0109 | 任务队列非空 |
| ERR\_PARAM | 0x010a | 参数不合理 |
| ERR\_CANCEL | 0x010b | 发送被取消 |
| ERR\_DOWNLOAD | 0x010c | 控制器下载失败 |
| ERR\_PROGRAM\_SIZE | 0x010d | 节目大小超出控制器最大容量 |
| ERR\_SENDHAND | 0x010e | 无效的发送句柄，发送已结束或为开始 |
| ERR\_Unknow | 0x0100 | 未知错误 |

## 设备错误码

|  |  |  |
| --- | --- | --- |
| 错误名称 | 代码 | 说明 |
| ERR\_OUTOFGROUP | 0x01 | 命令组错误 |
| ERR\_NOCMD | 0x02 | 此命令不存在 |
| ERR\_BUSY | 0x03 | 控制器忙 |
| ERR\_MEMORYVOLUME | 0x04 | 存储器容量越界 |
| ERR\_CHECKSUM | 0x05 | 数据包 CRC 校验错误 |
| ERR\_FILENOTEXIST | 0x06 | 此文件不存在 |
| ERR\_FLASH | 0x07 | Flash访问错误 |
| ERR\_FILE\_DOWNLOAD | 0x08 | 文件下载错误 |
| ERR\_FILE\_NAME | 0x09 | 文件名错误 |
| ERR\_FILE\_TYPE | 0x0a | 文件类型错误 |
| ERR\_FILE\_CRC16 | 0x0b | 文件校验错误 |
| ERR\_FONT\_NOT\_EXIST | 0x0c | 字库文件不存在 |
| ERR\_FIRMWARE\_TYPE | 0x0d | Firmware 与控制器类型不匹配 |
| ERR\_DATE\_TIME\_FORMAT | 0x0e | 日期时间格式错误 |
| ERR\_FILE\_EXIST | 0x0f | 此文件已存在 |
| ERR\_FILE\_BLOCK\_NUM | 0x10 | 文件 Block 号错误 |
| ERR\_CONTROLLER\_TYPE | 0x11 | 控制器类型不匹配 |
| ERR\_SCREEN\_PARA | 0x12 | 控制器参数越界或错误 |
| ERR\_CONTROLLER\_ID | 0x13 | 读取控制器 ID 错误 |
| ERR\_USER\_SECRET | 0x14 | 通讯时用户密码错误 |
| ERR\_OLD\_SECRET | 0x15 | 修改密码时输入的老密码错误 |
| ERR\_PHY1\_NO\_SECRET | 0x16 | 通讯时以加密 2 方式运行，提示物理层 PHY1 没有设定过密码 |
| ERR\_PHY\_USE\_SECTET | 0x17 | 通讯时以固定加密 1 方式运行，提示物理层 PHY1 没有设定过密码 |
| ERR\_FILE\_READ | 0x18 | 读取文件失败 |
| ERR\_XML\_TOP | 0x19 | 提取 XML 文件顶层元素失败 |
| ERR\_DIR\_NULL | 0x1a | 路径错误,空路径 |
| ERR\_DIR\_MK | 0x1b | 创建路径失败 |
| ERR\_DIR\_NOT\_EXIST | 0x1c | 此路径不存在 |
| ERR\_NOHEAD | 0x1d | 报文无头部 ，包头错误 |
| ERR\_DISK\_NAME | 0x1e | 驱动器名称错误 |
| ERR\_DISK\_NOT\_EXIST | 0x1f | 驱动器不存在(被拔除或者卸载了) |
| ERR\_OPEN\_FILE | 0x20 | 打开文件失败 |
| ERR\_FILE\_SEEK | 0x21 | 文件偏移错误 |
| ERR\_CMD\_UNFINISHED | 0x22 | 命令未能正常完成 ,但是原因不详 |
| ERR\_CMD\_DBG | 0x23 | 命令尚未设计完成 ，还处于调试阶段 |
| ERR\_CMD\_NOT\_SUPPORT | 0x24 | 命令不被支持 |
| ERR\_PERMISSIONS | 0x25 | 权限不够 |
| ERR\_UNLAWFUL\_OPERATION | 0x26 | 非法操作 |
| ERR\_NO\_RTC\_CHIP | 0x27 | 找不到 RTC 芯片 |
| ERR\_APP\_NUM | 0x28 | 已经安装有 5 个 app ，必须先卸载1个 |
| ERR\_APP\_NOT\_EXIST | 0x29 | 不存在这个程序 |
| ERR\_APP\_EXIST | 0x2a | 已经安装这个程序 |
| ERR\_CMD\_BUSY | 0x2b | 控制器忙 (这个命令暂时阻塞) |
| ERR\_NO\_LIST\_PLAYING | 0x2c | 播放列表未加载 |
| ERR\_NO\_PROGRAM\_PLAYING | 0x2d | 当前无节目播放 |
| ERR\_APP\_PATH | 0x2e | 路径出错 |
| ERR\_MEMRY\_USEOUT | 0x2f | 控制器内存耗尽 |
| ERR\_FTP\_IP | 0x30 | ftp 的 ip 不可达 |
| ERR\_USER\_EXIST | 0x31 | 已经有此用户 |
| ERR\_NOT\_LOGIN | 0x32 | 未登录 |
| ERR\_USER\_NOT\_EXIST | 0x33 | 无此用户 |
| ERR\_USER\_PASSWORD | 0x34 | 密码错误 |
| ERR\_USER\_ALREADY\_LOGIN | 0x35 | 已经有用户登录 |
| ERR\_PASSWORD\_INCONSISTENT | 0x36 | 密码不一致 |
| ERR\_USER\_NUM\_MAX | 0x37 | 用户已满 |
| ERR\_USER\_NAME\_NULL | 0x38 | 用户名为空 |
| ERR\_LOG\_MODE | 0x39 | 非法日志模式 |
| ERR\_READ\_DIR | 0x3a | 读目录失败 |
| ERR\_IP\_FORMAT | 0x3b | ip 格式错误 |
| ERR\_NULL\_TARGET | 0x3c | FTP 下载 目标名称为空 |
| ERR\_NULL\_PATH | 0x3d | FTP 下载 目标路径为空 |
| ERR\_NULL\_USER | 0x3e | FTP 下载 用户为空 |
| ERR\_NULL\_IP | 0x3f | FTP 下载 用户为空 |
| ERR\_FONT\_FORMAT | 0x40 | 字库格式错误 |
| ERR\_PROPERTY\_NOT\_ EXIST | 0x41 | 访问的属性不存在 |
| ERR\_PROPERTY\_READONLY | 0x42 | 只读属性，不能修改 |

# 函数概要说明

无论何种模式，在调用API之前需要先初始化SDK库，也就是调用Sdk\_Init，在程序退出时需要释放SDK库，也就是调用Sdk\_Release。

SDK库由4部分组成：

* 节目API，函数名以“YQ\_”开头，节目API与模式无关，只负责生成节目
* 局域网模式API，函数名以“Net\_”开头
* 服务器模式API，函数名以“Server\_”开头
* 云模式API，函数名以“Cloud\_”开头

只有局域网模式API能更改控制器的工作模式和网络地址

# 控制器说明

该版本的SDK支持全彩系列的控制器，具体支持型号如下定义：

typedef enum ONBON\_YQ\_CARD\_TYPE:USHORT

{

ONBON\_BX\_YQ1\_75=0x0158,

ONBON\_BX\_YQ1=0x0258,

ONBON\_BX\_YQ2=0x0358,

ONBON\_BX\_YQ3=0x0458,

ONBON\_BX\_YQ4=0x0558,

ONBON\_BX\_YQ2E=0x0658,

ONBON\_BX\_YQ5=0x758,

}onbon\_yq\_card\_type;

# 节目API

## API调用次序



## 函数说明

### YQ\_CreatePlaylist

**返回值：**

指向播放列表的句柄

**参数：** 无

**说明：**

创建播放列表

**函数：**

DWORD WINAPI YQ\_CreatePlaylist();

### YQ\_DestroyPlaylist

**返回值：** 无

**参数：**

DWORD hPlaylist：指向播放列表的句柄

**说明：**

销毁播放列表

**函数：**

void WINAPI YQ\_DestroyPlaylist(DWORD hPlaylist);

### YQ\_CreateProgram

**返回值：**

指向节目的句柄

**参数：**

short W：节目宽度，节目的宽高应与控制器的屏幕大小一致

short H：节目高度

**说明：**

创建节目

**函数：**

DWORD WINAPI YQ\_CreateProgram(short w,short h);

**示例：**

short w;

short h;

USHORT type;

int err=Net\_GetScreeninfo(card\_ip,&type,&w,&h);

if(err==0)

{

DWORD dwProgram=YQ\_CreateProgram(w,h);

}

### YQ\_CreateTimeArea

**返回值：**

指向节目区域的句柄

**参数：**

short X：分区x坐标

short Y：分区y坐标

short W：分区宽度

short H：分区高度

BYTE Transparency：分区不透明度 0-100

UCHAR timediff\_flag：0-正时差，1-负时差，

UCHAR timediff\_hour：时差的小时部分

UCHAR timediff\_min：时差的分钟部分

LPCWSTR font：时间区文本字体

int fontsize：字体大小

BOOL bold：是否加粗

BOOL italic：是否斜体

BOOL underline：是否下划线

UCHAR align：对齐方式，0-左对齐，1-居中对齐，2-右对齐

BOOL multiline：是否多行

BOOL day\_enable：是否使能显示日期

UINT daycolor：日期的颜色

UCHAR dayformat：日期格式0-18

BOOL week\_enable：是否显示星期

UINT weekcolor：星期颜色

UCHAR weekformat：星期格式0-8

BOOL time\_enable：是否显示时间

UINT timecolor：时间颜色

UCHAR timeformat：时间格式0-18

BOOL text\_enable：是否添加自定义文本

UINT textcolor：自定义文本颜色

LPCWSTR statictext：自定义文本内容

**说明：**

创建时间区域

**函数：**

DWORD WINAPI YQ\_CreateTimeArea(

short x,short y,short w,short h,UCHAR transparency,

UCHAR timediff\_flag,UCHAR timediff\_hour,UCHAR timediff\_min,

LPCWSTR font,UCHAR fontsize,BOOL bold,BOOL italic,BOOL underline,

UCHAR align,BOOL multiline,BOOL day\_enable,UINT daycolor,UCHAR dayformat,

BOOL week\_enable,UINT weekcolor,UCHAR weekformat,

BOOL time\_enable,UINT timecolor,UCHAR timeformat,

BOOL text\_enable,UINT textcolor,LPCWSTR statictext);

**示例：**

下面代码创建一个英文风格的包含年月日，星期，时间以及夏威夷标识的时间区

DWORD dwArea=YQ\_CreateCNTimeArea(0,0,128,96,0,0,0,0,

L"Arial",12,false,false,false,1,true,true,RGB(0,255,0),8,true,RGB(255,255,0),1,

true,RGB(0,255,0),2,true,RGB(255,0,0),L"Hawaii");

YQ\_ProgramAddArea(dwProgram,dwArea);

### YQ\_CreateCNTimeArea

**返回值：**

指向节目区域的句柄

**参数：**

short X：分区x坐标

short Y：分区y坐标

short W：分区宽度

short H：分区高度

BYTE Transparency：分区不透明度 0-100

UCHAR timediff\_flag：0-正时差，1-负时差，

UCHAR timediff\_hour：时差的小时部分

UCHAR timediff\_min：时差的分钟部分

LPCWSTR font：时间区文本字体

nt fontsize：字体大小

BOOL bold：是否加粗

BOOL italic：是否斜体

BOOL underline：是否下划线

UCHAR align：对齐方式，0-左对齐，1-居中对齐，2-右对齐

BOOL multiline：是否多行

BOOL day\_enable：是否使能显示日期

UINT daycolor：日期的颜色

BOOL week\_enable：是否显示星期

UINT weekcolor：星期颜色

BOOL time\_enable：是否显示时间

UINT timecolor：时间颜色

BOOL text\_enable：是否添加自定义文本

UINT textcolor：自定义文本颜色

LPCWSTR statictext：自定义文本内容

**说明：**

创建简单时间区域

**函数：**

DWORD WINAPI YQ\_CreateCNTimeArea(

short x,short y,short w,short h,UCHAR transparency,

UCHAR timediff\_flag,UCHAR timediff\_hour,UCHAR timediff\_min,

LPCWSTR font,UCHAR fontsize,BOOL bold,BOOL italic,BOOL underline,

UCHAR align,BOOL multiline,BOOL day\_enable,UINT daycolor,

BOOL week\_enable,UINT weekcolor,BOOL time\_enable,UINT timecolor,

BOOL text\_enable,UINT textcolor,LPCWSTR statictext);

**示例：**

下面代码创建一个中国风格的只有日期的北京标识的时间区

DWORD dwArea=YQ\_CreateCNTimeArea(0,0,128,96,0,0,0,0,

L"宋体",12,false,false,false,1,true,true,RGB(0,255,0),false,0,false,0,

true,RGB(255,0,0),L"北京时间");

YQ\_ProgramAddArea(dwProgram,dwArea);

### YQ\_CreateTimeCounterArea

**返回值：** 指向节目区域的句柄

**参数：**

short X：分区x坐标

short Y：分区y坐标

short W：分区宽度

short H：分区高度

BYTE Transparency：分区不透明度 0-100

LPCWSTR font：计时区文本字体

int fontsize：字体大小

BOOL bold：是否加粗

BOOL italic：是否斜体

BOOL underline：是否下划线

UCHAR align：对齐方式，0-左对齐，1-居中对齐，2-右对齐

BOOL multiline：是否多行

LPCWSTR target\_date：目标日期，格式2012-06-20

LPCWSTR target\_time：目标时间，格式02-23-55

BOOL bTimeFlag：0-倒计时，1-正计时

UINT counter\_color：计时器文本对应的颜色

BOOL day\_enable：是否显示天

LPCWSTR daytext：天数对应单位的字符串，默认天，英文默认day

BOOL hour\_enable：是否显示小时

LPCWSTR hourtext：小时对应单位的字符串，默认小时，英文默认hour

BOOL min\_enable：是否显示分钟

LPCWSTR minutetext：分钟单位对应的字符串，默认分，英文默认minute

BOOL sec\_enable：是否显示秒

LPCWSTR secondtext：秒对应单位的字符串，默认秒，英文默认s

BOOL add\_enable：是否计时累计

BOOL unit\_enable：是否显示单位（年，月，日）

BOOL text\_enable：是否添加自定义文本

UINT textcolor：自定义文本颜色

LPCWSTR statictext：自定义文本内容

**说明：**

创建时间区域

**函数：**

DWORD WINAPI YQ\_CreateTimeCounterArea(

short x,short y,short w,short h,UCHAR transparency,LPCWSTR font,

UCHAR fontsize,BOOL bold,BOOL italic,BOOL underline,UCHAR align,BOOL multiline,

LPCWSTR target\_date,LPCWSTR target\_time,BOOL bTimeFlag,

UINT counter\_color,BOOL day\_enable,LPCWSTR daytext,

BOOL hour\_enable,LPCWSTR hourtext,BOOL min\_enable,

LPCWSTR minutetext,BOOL sec\_enable,LPCWSTR secondtext,

BOOL add\_enable,BOOL unit\_enable,

BOOL text\_enable,UINT textcolor,LPCWSTR statictext);

**示例：**

创建一个英文风格的计时区

DWORD dwArea=YQ\_CreateTimeCounterArea(x,y,w,h,transparency,

L"Arial",10,true,false,false,1,true,L"2016-10-01",

L"01:01:01",0,RGB(192,192,192),

true,L"day",true,L"hour",true,L"min",true,L"s",true,true,false,0,L"");

创建一个中文风格的计时区

DWORD dwArea=YQ\_CreateTimeCounterArea(x,y,w,h,transparency,

L"宋体",10,true,false,false,1,true,L"2016-10-01",

L"01:01:01",0,RGB(192,192,192),

true,L"天",true,L"小时",true,L"分",true,L"秒",true,true,false,0,L"");

### YQ\_CreateLunarArea

**返回值：**

指向节目区域的句柄

**参数：**

short X：分区x坐标

short Y：分区y坐标

short W：分区宽度

short H：分区高度

BYTE Transparency：分区不透明度 0-100

LPCWSTR font：文本字体

int fontsize：字体大小

BOOL bold：是否加粗

BOOL italic：是否斜体

BOOL underline：是否下划线

UCHAR align：对齐方式，0-左对齐，1-居中对齐，2-右对齐

BOOL multiline：是否多行

BOOL year\_enable：是否使能显示年

UINT yearcolor：年的颜色

BOOL day\_enable：是否使能显示日期

UINT daycolor：日期的颜色

BOOL solarterms\_enable：是否使能显示节气

UINT solartermscolor：节气的颜色

BOOL text\_enable：是否添加自定义文本

UINT textcolor：自定义文本颜色

LPCWSTR statictext：自定义文本内容

**说明：**

创建农历区域

**函数：**

DWORD WINAPI YQ\_CreateLunarArea(

short x,short y,short w,short h,UCHAR transparency,

LPCWSTR font,UCHAR fontsize,BOOL bold,BOOL italic,BOOL underline,

UCHAR align,BOOL multiline,BOOL year\_enable,UINT yearcolor,

BOOL day\_enable,UINT daycolor,BOOL solarterms\_enable,UINT solartermscolor,

BOOL text\_enable,UINT textcolor,LPCWSTR statictext);

**示例：**

创建一个包含天干地支节气的分区

DWORD dwArea=YQ\_CreateLunarArea(x,y,w,h,transparency,

L"宋体",12,true,false,false,1,true,true,RGB(255,0,0),

true,RGB(255,0,0),true,RGB(255,0,0),false,0,L"");

### YQ\_CreateClockArea

**返回值：**

指向节目区域的句柄

**参数：**

short X：分区x坐标

short Y：分区y坐标

short W：分区宽度

short H：分区高度

BYTE Transparency：分区不透明度 0-100

BOOL text\_enable：是否添加自定义文本

UINT textcolor：自定义文本颜色

LPCWSTR statictext：自定义文本内容

LPCWSTR text\_font：文本字体

int text\_fontsize：字体大小

BOOL text\_bold：是否加粗

BOOL text\_italic：是否斜体

BOOL text\_underline：是否下划线

short text\_x：默认0

short text\_y：默认0

short text\_w：默认0

short text\_h：默认0

BOOL ymd\_enable：是否显示年月日

UINT ymdcolor：年月日颜色

UCHAR ymdformat：年月日格式0-18

LPCWSTR ymd\_font：年月日字体

UCHAR ymd\_fontsize：字体大小

BOOL ymd\_bold：是否加粗

BOOL ymd\_italic：是否斜体

BOOL ymd\_underline：是否下划线

short ymd\_x：默认0

short ymd\_y：默认0

short ymd\_w：默认0

short ymd\_h：默认0

BOOL week\_enable：是否显示星期

UINT weekcolor：星期颜色

UCHAR weekformat：星期格式0-8

LPCWSTR week\_font：星期字体

UCHAR week\_fontsize：字体大小

BOOL week\_bold：是否加粗

BOOL week\_italic：是否斜体

BOOL week\_underline：是否下划线

short week\_x：默认0

short week\_y：默认0

short week\_w：默认0

short week\_h：默认0

UINT hourhand\_color：时针颜色

UINT minhand\_color：分针颜色

UINT secondhand\_color：秒针颜色

UCHAR timediff\_flag：0-正时差，1-负时差，

UCHAR timediff\_hour：时差的小时部分

UCHAR timediff\_min：时差的分钟部分

int rightangle\_shape：时点形状，0-线形;1-点形;2-方形;3-数字;4-罗马

int rightangle\_width：时点宽度，默认2

int rightangle\_color：时点颜色

int integral\_shape：369点形状，0-线形;1-点形;2-方形;3-数字;4-罗马

int integral\_width,369点宽度

int integral\_color：369点颜色

int minute\_shape：分点形状，0-线形;1-点形;2-方形

int minute\_width：分店宽度

int minute\_color：分点颜色

LPCWSTR szClockFile：表盘刻度图片，默认风格是自定义绘制，所以默认为空

**说明：**

创建表盘区域

**函数：**

DWORD WINAPI YQ\_CreateClockArea(

short x,short y,short w,short h,UCHAR transparency,

BOOL text\_enable,UINT textcolor,LPCWSTR statictext,

LPCWSTR text\_font,UCHAR text\_fontsize,BOOL text\_bold,BOOL text\_italic,

BOOL text\_underline,short text\_x,short text\_y,short text\_w,short text\_h,

BOOL ymd\_enable,UINT ymdcolor,UCHAR ymdformat,

LPCWSTR ymd\_font,UCHAR ymd\_fontsize,BOOL ymd\_bold,BOOL ymd\_italic,

BOOL ymd\_underline,short ymd\_x,short ymd\_y,short ymd\_w,short ymd\_h,

BOOL week\_enable,UINT weekcolor,UCHAR weekformat,

LPCWSTR week\_font,UCHAR week\_fontsize,BOOL week\_bold,

BOOL week\_italic,BOOL week\_underline,short week\_x,short week\_y,

short week\_w,short week\_h,UINT hourhand\_color,UINT minhand\_color,

UINT secondhand\_color,UCHAR timediff\_flag,UCHAR timediff\_hour,

UCHAR timediff\_min,int rightangle\_shape,int rightangle\_width,

int rightangle\_color,int integral\_shape,int integral\_width,int integral\_color,

int minute\_shape,int minute\_width,int minute\_color,LPCWSTR szClockFile);

**示例：**

创建一个有一小时时差的点风格的表盘

DWORD dwArea=YQ\_CreateClockArea(x,y,w,h,transparency,

false,0,L"",L"宋体",12,false,false,false,0,0,0,0,

true,RGB(255,255,255),0,L"宋体",12,false,false,false,0,0,0,0,

true,RGB(255,255,255),0,L"宋体",12,false,false,false,0,0,0,0,

RGB(0,255,0),RGB(255,0,0),RGB(0,0,255),

0,1,0,1,2,RGB(0,255,0),1,2,RGB(0,255,0),1,1,RGB(0,255,0),NULL);

### YQ\_CreateClockArea

**返回值：**

指向节目区域的句柄

**参数：**

short X：分区x坐标

short Y：分区y坐标

short W：分区宽度

short H：分区高度

BYTE Transparency：分区不透明度 0-100

BOOL text\_enable：是否添加自定义文本

UINT textcolor：自定义文本颜色

LPCWSTR statictext：自定义文本内容

LPCWSTR text\_font：文本字体

int text\_fontsize：字体大小

BOOL text\_bold：是否加粗

BOOL text\_italic：是否斜体

BOOL text\_underline：是否下划线

BOOL ymd\_enable：是否显示年月日

UINT ymdcolor：年月日颜色

BOOL week\_enable：是否显示星期

UINT weekcolor：星期颜色

UINT hourhand\_color：时针颜色

UINT minhand\_color：分针颜色

UINT secondhand\_color：秒针颜色

UCHAR timediff\_flag：0-正时差，1-负时差，

UCHAR timediff\_hour：时差的小时部分

UCHAR timediff\_min：时差的分钟部分

int rightangle\_color：时点颜色

int integral\_color：369点颜色

int minute\_color：分点颜色

UINT dwClockFormat：表盘风格，0-线形;1-点形;2-方形;3-数字;4-罗马

**说明：**

创建特定风格的表盘区域

**函数：**

DWORD WINAPI YQ\_CreateClockStyleArea(

short x,short y,short w,short h,UCHAR transparency,

BOOL text\_enable,UINT textcolor,LPCWSTR statictext,

LPCWSTR text\_font,UCHAR text\_fontsize,BOOL text\_bold,BOOL text\_italic,

BOOL text\_underline,BOOL ymd\_enable,UINT ymdcolor,BOOL week\_enable,

UINT weekcolor,UINT hourhand\_color,UINT minhand\_color,

UINT secondhand\_color,UCHAR timediff\_flag,UCHAR timediff\_hour,

UCHAR timediff\_min,int rightangle\_color,int integral\_color

,int minute\_color,UINT dwClockFormat);

**示例：**

创建一个有一小时时差的点风格的表盘，与上述例子效果一样

DWORD dwArea=YQ\_CreateClockStyleArea(x,y,w,h,transparency,false,0,L"",

L"宋体",12,false,false,false,true,RGB(255,255,255),true,

RGB(255,255,255),RGB(0,255,0),RGB(255,0,0),RGB(0,0,255),0,1,0,RGB(0,255,0),

RGB(0,255,0),RGB(0,255,0),1);

### YQ\_CreateVideoArea

**返回值：**

指向节目区域的句柄

**参数：**

short X：分区x坐标

short Y：分区y坐标

short W：分区宽度

short H：分区高度

BYTE Transparency：分区不透明度 0-100

**说明：**

创建视频区域

**函数：**

DWORD WINAPI YQ\_CreateVideoArea(

short x,short y,short w,short h,UCHAR transparency);

### YQ\_CreatePicArea

**返回值：** 指向节目区域的句柄

**参数：**

short X：分区x坐标

short Y：分区y坐标

short W：分区宽度

short H：分区高度

BYTE Transparency：分区不透明度 0-100

UCHAR window\_type：窗口类型,

1：图片分区；2：文字分区；3：字幕分区；4：GIF 动态图片分区

BOOL bBgTransparent：背景是否透明

**说明：**

创建图文区域

**函数：**

DWORD WINAPI YQ\_CreatePicArea(

short x,short y,short w,short h,UCHAR transparency,UCHAR window\_type,

BOOL bBgTransparent);

### YQ\_VideoAreaAddUnit

**返回值：** 无

**参数：**

DWORD hArea：区域句柄

LPCWSTR szVideoFile：视频文件全路径

UCHAR scale\_mode：缩放模式，0-按原始比例进行缩放，1-按窗口比例进行缩放

UCHAR volume：音量

**说明：**

向视频区域的里面添加视频项

**函数：**

void WINAPI YQ\_VideoAreaAddUnit(

DWORD hArea,LPCWSTR szVideoFile,UCHAR scale\_mode,UCHAR volume);

**示例：**

### YQ\_PicAreaAddImageUnit

**返回值：** 无

**参数：**

DWORD hArea：区域句柄

LPCWSTR szRtfFile：图片文件全路径，目前可以是bmp,jpg,png,gif

USHORT StayTime：显示参数，停留时间

BYTE DisplayEffects：显示参数，特技编号

BYTE DisplaySpeed：显示参数，特技运行速度

**说明：**

向图文区域的里面添加图片项

**函数：**

void WINAPI YQ\_PicAreaAddImageUnit(

DWORD hArea,LPCWSTR szImgFile,UCHAR display\_effects,

UCHAR display\_speed,USHORT stay\_time);

**示例：**

YQ\_PicAreaAddImageUnit(dwArea,L"d:\\pic\\123.bmp",1,1,1);

### YQ\_PicAreaAddRtfUnit

**返回值：** 无

**参数：**

DWORD hArea：区域句柄

LPCWSTR szRtfFile：rtf文件全路径

USHORT StayTime：显示参数，停留时间

BYTE DisplayEffects：显示参数，特技编号

BYTE DisplaySpeed：显示参数，特技运行速度

**说明：**

向图文区域的里面添加文本项

**函数：**

void WINAPI YQ\_PicAreaAddRtfUnit(

DWORD hArea,LPCWSTR szRtfFile,UCHAR display\_effects,

UCHAR display\_speed,USHORT stay\_time);

### YQ\_ProgramAddArea

**返回值：** 无

**参数：**

DWORD hArea：区域句柄

DWORD hProgram：节目句柄

**说明：**

向节目添加区域

**函数：**

void WINAPI YQ\_ProgramAddArea(DWORD hProgram,DWORD hArea);

### YQ\_PlaylistAddProgram

**返回值：** 无

**参数：**

DWORD hPlaylist：播放列表句柄

DWORD hProgram：节目句柄

unsigned char play\_mode：播放模式，0 – 定长播放，1 – 定次播放

unsigned int play\_time：播放时长或者播放次数

LPCWSTR aging\_start\_time：播放时效的开始日期，格式2012-06-20

LPCWSTR aging\_stop\_time：播放时效结束日期

LPCWSTR period\_ontime：播放时段开始时间,格式08:30:00

LPCWSTR period\_offtime：播放时段结束时间

unsigned char play\_week：bit0 ～ bit6 依次表示星期一至星期天，

0—表示该天节目不能播放，1—表示该天节目可以播放

**说明：**

向播放列表添加节目

**函数：**

void WINAPI YQ\_PlaylistAddProgram(

DWORD hPlaylist,DWORD hProgram,unsigned char play\_mode,

unsigned int play\_time,LPCWSTR aging\_start\_time,

LPCWSTR aging\_stop\_time,LPCWSTR period\_ontime,

LPCWSTR period\_offtime,unsigned char play\_week);

**示例：**

YQ\_PlaylistAddProgram(dwPl,dwProgram,1,1,L"",L"",L"",L"",127)

### YQ\_CreateDynamicArea

**返回值：** 指向动态区的句柄

**参数：**

BYTE AreaID：

short X：动态区x坐标

short Y：动态区y坐标

short W：动态区宽度

short H：动态区高度

BYTE Transparency：动态区不透明度 0-100

BYTE ProgrmRelation：1 关联节目绑定播放；0 关联节目播放完后播放

USHORT RelatedProgram：所关联的节目编号，如果没有对应关联的节目应传0xffff

BYTE RunTime：0-绑定节目一起播放；1-与绑定节目轮播

BYTE RunMode：动态区运行模式

0— 动态区数据循环显示。

1— 动态区数据顺序显示，显示完最后一页后就不再显示

2— 动态区数据显示完成后静止显示最后一页数据。

3— 动态区数据循环显示，超过设定时间后数据仍未更新删除动态区信息

4--动态区数据循环显示，超过设定时间后数据仍未更新时播放 LOGO 图片

short Timeout：动态区数据更新超时时间，单位为秒

BYTE DataType：数据类型，0-图片；1-文本

**说明：**

创建动态区

**函数：**

DWORD WINAPI YQ\_CreateDynamicArea(

BYTE AreaID,short X,short Y,short W,short H,BYTE Transparency,BYTE ProgrmRelation, USHORT RelatedProgram,BYTE RunTime,BYTE RunMode,

short Timeout,BYTE DataType);

### YQ\_DestroyDynamic

**返回值：** 无

**参数：**

DWORD hArea：区域句柄

**说明：**

销毁创建的动态区

**函数：**

void WINAPI YQ\_DestroyDynamic(DWORD hArea);

### YQ\_DynamicAreaAddStrPage

**返回值：** 无

**参数：**

DWORD hArea：区域句柄

USHORT StayTime：显示参数，停留时间

BYTE DisplayEffects：显示参数，特技编号

BYTE DisplaySpeed：显示参数，特技运行速度

DWORD BgColor：文本背景色

BYTE LinesSizes：文本行间距

BOOL bold：是否加粗

BOOL italic：是否斜体

BOOL underline：是否下划线

BOOL strikeout：是否中划线

BOOL antialiasing：是否反锯齿

LPCWSTR str：文本内容

UINT txtcolor：文本颜色

LPCWSTR font：文本字体

int fontsize：字体大小

**说明：**

动态区添加文本分页

**函数：**

void WINAPI YQ\_DynamicAreaAddStrPage(

DWORD hArea,USHORT StayTime,BYTE DisplayEffects,BYTE DisplaySpeed,DWORD BgColor,BYTE LinesSizes,BOOL bold,BOOL italic

,BOOL underline,BOOL strikeout,BOOL antialiasing,LPCWSTR str,UINT txtcolor,

LPCWSTR font,int fontsize);

### YQ\_DynamicAreaAddPicPage

**返回值：** 无

**参数：**

DWORD hArea：区域句柄

USHORT StayTime：显示参数，停留时间

BYTE DisplayEffects：显示参数，特技编号

BYTE DisplaySpeed：显示参数，特技运行速度

char\* Suffix：图片格式后缀名,限”jpg” ”bmp” ”png”三种;必须为小写

DWORD imgdatalen：图片数据长度

BYTE\* imgdata：图片数据内容

**说明：**

动态区添加图片分页

**函数：**

void WINAPI YQ\_DynamicAreaAddPicPage(

DWORD hArea,USHORT StayTime,BYTE DisplayEffects,BYTE DisplaySpeed,char\*，Suffix,DWORD imgdatalen,BYTE\* imgdata);

# 局域网模式API

## API调用次序

如下表所示：



## 数据结构

### 控制器单元定义

typedef struct \_CARD\_UNIT

{

unsigned char PID[16]; //控制器唯一ID

char barcode[16]; //控制器条形码

char ip[16]; //控制器地址

}card\_unit;

该结构体主要用于搜索命令返回控制器信息

## 函数说明

### Net\_SearchCards

**函数：**

int WINAPI Net\_SearchCards(card\_unit\* pCardList,int ntimeout)

**返回值：**

搜索到的控制器个数

**参数：**

card\_unit\* pCardList：存放控制器列表的缓冲区

int ntimeout：查找时的超时时间

**说明：**

在局域网内搜索所有控制器。

### Net\_SetStaticip

**返回值：**

成功返回0；失败返回错误号

**参数：**

unsigned char\* pid：控制器唯一ID

char\* ip：控制器地址

char\* mask：子网掩码

char\* gateway：网关地址

**说明：**

设置控制器的网络地址。

**函数：**

int WINAPI Net\_SetStaticip(unsigned char\* pid,char\* ip,char\* mask,char\* gateway)

### Net\_SetAutoip

**返回值：**

成功返回通讯句柄，失败返回0

**参数：**

unsigned char\* pid：控制器唯一ID

**说明：**

设置控制器通过DHCP获取网络地址

**函数：**

int WINAPI Net\_SetAutoip(unsigned char\* pid)

### Net\_GetIp

**返回值：**

成功返回通讯句柄，失败返回0

**参数：**

unsigned char\* pid：控制器唯一ID

char\* ip：控制器地址

**说明：**

查询指定控制器的网络地址

**函数：**

int WINAPI Net\_GetIp(unsigned char\* pid,char\* ip)

### Net\_SetServerMode

**返回值：**

成功返回0；失败返回错误号

**参数：**

char\* ip：控制器地址

char\* serverip：控制器要连接的服务器的ip地址

unsigned short port：服务器所用的端口

**说明：**

将工作模式切换为连接服务器模式

警告: 执行本函数将会重启控制器

**函数：**

int WINAPI Net\_SetServerMode(char\* ip,char\* serverip,unsigned short port);

### Net\_SetClientMode

**返回值：**

成功返回0；失败返回错误号

**参数：**

char\* ip：控制器地址

**说明：**

将工作模式切换为连接服务器模式

警告: 执行本函数将会重启控制器

**函数：**

int WINAPI Net\_SetClientMode(char\* ip);

### Net\_GetModeinfo

**返回值：**

成功返回0；失败返回错误号

**参数：**

har\* ip：控制器地址

unsigned char\* mode：控制器工作模式

har\* ServerIPAddress：控制器要连接的服务器的ip地址

unsigned short port：服务器所用的端口

**说明：**

查询控制器工作模式，如果是服务器模式返回服务器地址

**函数：**

int WINAPI Net\_GetModeinfo(

char\* ip,unsigned char\* mode,char\* ServerIPAddress,unsigned short\* ServerPort);

### Net\_GetNetinfo

**返回值：**

成功返回0；失败返回错误号

**参数：**

char\* ip：控制器地址

USHORT\* ControllerType：传出参数，返回控制器型号

short\* ScreenWidth：传出参数，返回led屏幕宽度

short\* ScreenHeight：传出参数，返回led屏幕高度

char\* FirmwareVersion：传出参数，返回控制器固件版本

**说明：**

获取控制器参数信息和屏幕参数信息

**函数：**

int WINAPI Net\_GetNetinfo(

char\* ip,unsigned char\* ControlerID,char\* netip,char\* mask,

char\* gateway,BYTE\* pConnectMode);

### Net\_GetScreeninfo

**返回值：**

成功返回0；失败返回错误号

**参数：**

char\* ip：控制器地址

USHORT\* ControllerType：传出参数，返回控制器型号

short\* ScreenWidth：传出参数，返回led屏幕宽度

short\* ScreenHeight：传出参数，返回led屏幕高度

**说明：**

获取控制器参数信息和屏幕参数信息

**函数：**

int WINAPI Net\_GetScreeninfo(

char\* ip,unsigned short \*pControllerType,short\* w,short\* h);

### Net\_GetFirmwareinfo

**返回值：**

成功返回0；失败返回错误号

**参数：**

char\* ip：控制器地址

USHORT\* ControllerType：传出参数，返回控制器型号

char\* FirmwareTime：传出参数，返回固件创建时间

char\* FirmwareVersion：传出参数，返回控制器固件版本

**说明：**

获取控制器固件信息

**函数：**

int WINAPI Net\_GetFirmwareinfo(

char\* ip,char\* FirmwareTime,char\* FirmwarelVersion);

### Net\_GetBrightness

**返回值：**

成功返回0；失败返回错误号

**参数：**

char\* ip：控制器地址

BYTE\* pMode：传出参数，返回控制器亮度所用的模式,

0-手工调亮,1-定时调亮，2-传感器默认调亮，3-传感器调亮

BYTE\* pValue：传出参数，返回亮度值，如果不是手工调亮，

返回的是一个48字节长的亮度列表，每个字节对应半个小时的亮度值

**说明：**

获取控制器当前亮度信息

**函数：**

int WINAPI Net\_GetBrightness(char\* ip,BYTE\* pMode,BYTE\* pValue);

### Net\_GetVolume

**返回值：**

成功返回0；失败返回错误号

**参数：**

char\* ip：控制器地址

BYTE\* pVolume：传出参数，返回控制器音量

**说明：**

获取控制器参数信息和屏幕参数信息

**函数：**

int WINAPI Net\_GetVolume(char\* ip,BYTE\* pVolume);

### Net\_GetOnoff

**返回值：**

成功返回0；失败返回错误号

**参数：**

char\* ip：控制器地址

BYTE\* onoff：传出参数，返回控制器当前屏幕开关状态，

0:手动关闭 1：手动打开 2：自动关闭 3：自动打开

**说明：**

获取控制器屏幕开关信息

**函数：**

int WINAPI Net\_GetOnoff(char\* ip,BYTE\* onoff);

### Net\_SetScreenSize

**返回值：**

成功返回0；失败返回错误号

**参数：**

char\* ip：控制器地址

short ScreenWidth：led屏幕宽度

short ScreenHeight：led屏幕高度

**说明：**

设置控制器的屏幕参数信息

**函数：**

int WINAPI Net\_SetScreenSize(char\* ip,short ScreenWidth,short ScreenHeight);

### Net\_SendProgram

**返回值：**

成功返回发送需要的句柄；失败返回0

**参数：**

char\* ip：通讯地址

unsigned char media：控制器存储节目的介质

DWORD hPlaylist：播放列表对应的句柄

LPCWSTR szLocalTempDir：发送节目时需要在本地生成节目文件，

该参数为临时文件的存放目录

int \*pErr：如果有错误，返回的错误号

**说明：**

发送节目到控制器

**函数：**

DWORD WINAPI Net\_SendPrograms(

char\* ip,unsigned char media,DWORD hPlaylist,LPCWSTR szLocalTempDir,

int \*pErr);

### Net\_GetSendProcess

**返回值：**

成功返回0；失败返回错误号

**参数：**

DWORD dwSendHand：发送句柄

int\* total\_percent：传出参数，当前发送的总进度

int\* cur\_percent：传出参数，当前文件发送的进度

**说明：**

发送指定节目时查看发送进度

**函数：**

int WINAPI Net\_GetSendProcess(

DWORD dwSendHand,int\* total\_percent,int\* cur\_percent);

### Net\_CancelSend

**返回值：**

成功返回0；失败返回错误号

**参数：**

DWORD dwSendHand：发送句柄

**说明：**

无论发送是否完成，结束发送节目

**函数：**

int WINAPI Net\_CancelSend(DWORD dwSendHand);

### Net\_StopPlay

**返回值：**

成功返回0；失败返回错误号

**参数：**

char\* ip：控制器地址

**说明：**

停止播放节目

**函数：**

int WINAPI Net\_StopPlay(char\* ip);

### Net\_SetVolume

**返回值：**

成功返回0；失败返回错误号

**参数：**

char\* ip：控制器地址

BYTE bVolume：音量值，0-100

**说明：**

设置音量

**函数：**

int WINAPI Net\_SetVolume(char\* ip,BYTE bVolume);

### Net\_AdjustBrightness

**返回值：**

成功返回0；失败返回错误号

**参数：**

char\* ip：控制器地址

BYTE mode：调整模式，0x00–手动调亮；0x01–定时调亮；

0x02–默认调亮（需外接传感器），0x03–自动调亮（需外接传感器）

short\* bright\_table：亮度表

亮度表主要针非手工调亮，每30分钟一个值，所以一天48个值

bright\_table[0]=00:00 – 00:29 间的亮度值

bright\_table[1]=00:30 – 00:59 间的亮度值

......

bright\_table[47]=23:30 – 23:59 间的亮度值

手动调亮可以将所有值设为相同，每个值范围0-255

**说明：**

设置亮度

**函数：**

int WINAPI Net\_AdjustBrightness(char\* ip,BYTE mode,short\* bright\_table);

### Net\_SystemTimeCorrect

**返回值：**

成功返回0；失败返回错误号

**参数：**

char\* ip：控制器地址

**说明：**

校正控制器时间

**函数：**

int WINAPI Net\_TimeCorrect(char\* ip);

### Net\_OpenScreen

**返回值：**

成功返回0；失败返回错误号

**参数：**

char\* ip：控制器地址

**说明：**

开屏幕

**函数：**

int WINAPI Net\_OpenScreen(char\* ip);

### Net\_CloseScreen

**返回值：**

成功返回0；失败返回错误号

**参数：**

char\* ip：控制器地址

**说明：**

关屏幕

**函数：**

int WINAPI Net\_CloseScreen(char\* ip);

### Net\_SwitchOnTime

**返回值：**

成功返回0；失败返回错误号

**参数：**

char\* ip：控制器地址

char\* OnTm1：第一个时间段的开机时间，格式12:56:09

char\* OffTm1：第一个时间段的关机时间，格式12:56:09

LPCSTR SwitchScreenFilePath：定时开关机文件路径

**说明：**

设置定时关机

**函数：**

int WINAPI Net\_SwitchOnTime(

char\* ip,char\* OnTm1,char\* OffTm1,char\* OnTm2,char\* OffTm2,

char\* OnTm3,char\* OffTm3,char\* OnTm4,char\* OffTm4);

### Net\_Update

**返回值：**

成功返回0；失败返回错误号

**参数：**

char\* ip：控制器地址

LPCWSTR strMd5File：固件文件之一，固件的描述文件

LPCWSTR strUpdateFile：固件文件之一，固件文件

**说明：**

升级控制器的固件

**函数：**

int WINAPI Net\_Update(char\* ip,LPCWSTR strMd5File,LPCWSTR strUpdateFile);

### Net\_UpdateDynamicArea

**返回值：**

成功返回0；失败返回错误号

**参数：**

char\* ip：控制器地址

DWORD hArea：区域句柄

**说明：**

更新动态区

**函数：**

int WINAPI Net\_UpdateDynamicArea(char\* ip,DWORD hArea);

### Net\_RemoveDynamicArea

**返回值：**

成功返回0；失败返回错误号

**参数：**

char\* ip：控制器地址

int AreaId：动态区域编号

**说明：**

删除控制器上的动态区域

**函数：**

int WINAPI Net\_RemoveDynamicArea(char\* ip,int AreaId);

LockProgram

### Net\_LockProgram

**返回值：**

成功返回0；失败返回错误号

**参数：**

char\* ip：控制器地址

int program\_id：编号

BYTE SaveLock：0:掉电不保存锁定命令 1:掉电保存锁定命令

**说明：**

锁定指定的节目

**函数：**

int WINAPI Net\_LockProgram(char\* ip,int program\_id,BYTE SaveLock);

### Net\_UnLockProgram

**返回值：**

成功返回0；失败返回错误号

**参数：**

char\* ip：控制器地址

**说明：**

解除锁定的节目

**函数：**

int WINAPI Net\_UnLockProgram(char\* ip);

# 服务器模式API

## API调用次序

如下表所示：



## 数据结构

### 控制器单元定义

typedef struct \_CARD\_SERVER

{

unsigned char PID[16]; //控制器唯一ID

char barcode[16]; //控制器条形码

}server\_unit;

该结构体主要用于搜索命令返回控制器信息

## 函数说明

### Server\_Start

**函数：**

int WINAPI Server\_Start(USHORT port)

**返回值：**

成功返回0；失败返回错误号

**参数：**

USHORT port：服务器端口

**说明：**

启动服务器。

### Server\_SearchCards

**函数：**

int WINAPI Server\_SearchCards(server\_unit\* pList)

**返回值：**

搜索到的控制器个数

**参数：**

server\_unit\* pList：存放控制器列表的缓冲区

**说明：**

在局域网内搜索所有控制器。

### Server\_Close

**返回值：**

成功返回0；失败返回错误号

**参数：**

无

**说明：**

关闭服务器

**函数：**

int WINAPI Server\_Close()

### Server\_GetModeinfo

**返回值：**

成功返回0；失败返回错误号

**参数：**

unsigned char\* PID：控制器唯一ID

char\* ServerIPAddress：控制器要连接的服务器的ip地址

unsigned short port：服务器所用的端口

**说明：**

查询控制器工作模式，如果是服务器模式返回服务器地址

**函数：**

int WINAPI Server\_GetModeinfo(

unsigned char\* PID,char\* ServerIPAddress,unsigned short\* ServerPort);

### Server\_GetNetinfo

**返回值：**

成功返回0；失败返回错误号

**参数：**

unsigned char\* PID：控制器唯一ID

USHORT\* ControllerType：传出参数，返回控制器型号

short\* ScreenWidth：传出参数，返回led屏幕宽度

short\* ScreenHeight：传出参数，返回led屏幕高度

char\* FirmwareVersion：传出参数，返回控制器固件版本

**说明：**

获取控制器参数信息和屏幕参数信息

**函数：**

int WINAPI Server\_GetNetinfo(

unsigned char\* PID,unsigned char\* ControlerID,char\* netip,char\* mask,

char\* gateway,BYTE\* pConnectMode);

### Server\_GetScreeninfo

**返回值：**

成功返回0；失败返回错误号

**参数：**

unsigned char\* PID：控制器唯一ID

USHORT\* ControllerType：传出参数，返回控制器型号

short\* ScreenWidth：传出参数，返回led屏幕宽度

short\* ScreenHeight：传出参数，返回led屏幕高度

**说明：**

获取控制器参数信息和屏幕参数信息

**函数：**

int WINAPI Server\_GetScreeninfo(

unsigned char\* PID,unsigned short \*pControllerType,short\* w,short\* h);

### Server\_GetFirmwareinfo

**返回值：**

成功返回0；失败返回错误号

**参数：**

unsigned char\* PID：控制器唯一ID

char\* FirmwareTime：传出参数，返回固件创建时间

char\* FirmwareVersion：传出参数，返回控制器固件版本

**说明：**

获取控制器固件信息

**函数：**

int WINAPI Server\_GetFirmwareinfo(

unsigned char\* PID,char\* FirmwareTime,char\* FirmwarelVersion);

### Server\_GetBrightness

**返回值：**

成功返回0；失败返回错误号

**参数：**

unsigned char\*PID：控制器唯一ID

BYTE\* pMode：传出参数，返回控制器亮度所用的模式,

0-手工调亮,1-定时调亮，2-传感器默认调亮，3-传感器调亮

BYTE\* pValue：传出参数，返回亮度值，如果不是手工调亮，

返回的是一个48字节长的亮度列表，每个字节对应半个小时的亮度值

**说明：**

获取控制器当前亮度信息

**函数：**

int WINAPI Server\_GetBrightness(unsigned char\* PID,BYTE\* pMode,BYTE\* pValue);

### Server\_GetVolume

**返回值：**

成功返回0；失败返回错误号

**参数：**

unsigned char\* PID：控制器唯一ID

BYTE\* pVolume：传出参数，返回控制器音量

**说明：**

获取控制器参数信息和屏幕参数信息

**函数：**

int WINAPI Server\_GetVolume(unsigned char\* PID,BYTE\* pVolume);

### Server\_GetOnoff

**返回值：**

成功返回0；失败返回错误号

**参数：**

unsigned char\* PID：控制器唯一ID

BYTE\* onoff：传出参数，返回控制器当前屏幕开关状态，

0:手动关闭 1：手动打开 2：自动关闭 3：自动打开

**说明：**

获取控制器屏幕开关信息

**函数：**

int WINAPI Server\_GetOnoff(unsigned char\* PID,BYTE\* onoff);

### Server\_SetScreenSize

**返回值：**

成功返回0；失败返回错误号

**参数：**

unsigned char\* PID：控制器唯一ID

short ScreenWidth：led屏幕宽度

short ScreenHeight：led屏幕高度

**说明：**

设置控制器的屏幕参数信息

**函数：**

int WINAPI Server\_SetScreenSize(unsigned char\* PID,short ScreenWidth,short ScreenHeight);

### Server\_SendProgram

**返回值：**

成功返回发送需要的句柄；失败返回0

**参数：**

unsigned char\* PID：控制器唯一ID

unsigned char media：控制器存储节目的介质

DWORD hPlaylist：播放列表对应的句柄

LPCWSTR szLocalTempDir：发送节目时需要在本地生成节目文件，

该参数为临时文件的存放目录

char\* ftpIp：FTP服务器IP

short ftpPort：FTP服务器端口

char\* ftpUser：FTP服务器用户名

char\* ftppPwd：FTP服务器密码

int \*pErr：如果有错误，返回的错误号

**说明：**

发送节目到控制器

**函数：**

DWORD WINAPI Server\_SendPrograms(

unsigned char\* PID,unsigned char media,DWORD hPlaylist,LPCWSTR szLocalTempDir,char\* ftpIp,short ftpPort,char\* ftpUser,char\* ftppPwd

int \*pErr);

### Server\_GetSendProcess

**返回值：**

成功返回0；失败返回错误号

**参数：**

DWORD dwSendHand：发送句柄

int\* total\_percent：传出参数，当前发送的总进度

int\* cur\_percent：传出参数，当前文件发送的进度

**说明：**

发送指定节目时查看发送进度

**函数：**

int WINAPI Server\_GetSendProcess(

DWORD dwSendHand,int\* total\_percent,int\* cur\_percent);

### Server\_GetDownProcess

**返回值：**

成功返回0；失败返回错误号

**参数：**

DWORD dwSendHand：发送句柄

int\* down\_percent：传出参数，当前文件下载的总进度

**说明：**

发送指定节目时查看发送进度

**函数：**

int WINAPI Server\_GetSendProcess(

DWORD dwSendHand,int\* down\_percent);

### Server\_CancelSend

**返回值：**

成功返回0；失败返回错误号

**参数：**

DWORD dwSendHand：发送句柄

**说明：**

无论发送是否完成，结束发送节目

**函数：**

int WINAPI Server\_CancelSend(DWORD dwSendHand);

### Server\_StopPlay

**返回值：**

成功返回0；失败返回错误号

**参数：**

unsigned char\* PID：控制器唯一ID

**说明：**

停止播放节目

**函数：**

int WINAPI Server\_StopPlay(unsigned char\* PID);

### Server\_SetVolume

**返回值：**

成功返回0；失败返回错误号

**参数：**

unsigned char\* PID：控制器唯一ID

BYTE bVolume：音量值，0-100

**说明：**

设置音量

**函数：**

int WINAPI Server\_SetVolume(unsigned char\* PID,BYTE bVolume);

### Server\_AdjustBrightness

**返回值：**

成功返回0；失败返回错误号

**参数：**

unsigned char\* PID：控制器唯一ID

BYTE mode：调整模式，0x00–手动调亮；0x01–定时调亮；

0x02–默认调亮（需外接传感器），0x03–自动调亮（需外接传感器）

short\* bright\_table：亮度表

亮度表主要针非手工调亮，每30分钟一个值，所以一天48个值

bright\_table[0]=00:00 – 00:29 间的亮度值

bright\_table[1]=00:30 – 00:59 间的亮度值

......

bright\_table[47]=23:30 – 23:59 间的亮度值

手动调亮可以将所有值设为相同，每个值范围0-255

**说明：**

设置亮度

**函数：**

int WINAPI Server\_AdjustBrightness(unsigned char\* PID,BYTE mode,short\* bright\_table);

### Server\_SystemTimeCorrect

**返回值：**

成功返回0；失败返回错误号

**参数：**

unsigned char\* PID：控制器唯一ID

**说明：**

校正控制器时间

**函数：**

int WINAPI Server\_TimeCorrect(unsigned char\* PID);

### Server\_OpenScreen

**返回值：**

成功返回0；失败返回错误号

**参数：**

unsigned char\* PID：控制器唯一ID

**说明：**

开屏幕

**函数：**

int WINAPI Server\_OpenScreen(unsigned char\* PID);

### Server\_CloseScreen

**返回值：**

成功返回0；失败返回错误号

**参数：**

unsigned char\* PID：控制器唯一ID

**说明：**

关屏幕

**函数：**

int WINAPI Server\_CloseScreen(unsigned char\* PID);

### Server\_SwitchOnTime

**返回值：**

成功返回0；失败返回错误号

**参数：**

unsigned char\* PID：控制器唯一ID

char\* OnTm1：第一个时间段的开机时间，格式12:56:09

char\* OffTm1：第一个时间段的关机时间，格式12:56:09

....

char\* ftpIp：FTP服务器IP

short ftpPort：FTP服务器端口

char\* ftpUser：FTP服务器用户名

char\* ftppPwd：FTP服务器密码

**说明：**

设置定时关机

**函数：**

int WINAPI Server\_SwitchOnTime(

unsigned char\* PID,char\* OnTm1,char\* OffTm1,char\* OnTm2,char\* OffTm2,

char\* OnTm3,char\* OffTm3,char\* OnTm4,char\* OffTm4,char\* ftpIp,

short ftpPort,char\* ftpUser,char\* ftppPwd);

### Server\_Update

**返回值：**

成功返回0；失败返回错误号

**参数：**

unsigned char\* PID：控制器唯一ID

LPCWSTR strMd5File：固件文件之一，固件的描述文件

LPCWSTR strUpdateFile：固件文件之一，固件文件

char\* ftpIp：FTP服务器IP

short ftpPort：FTP服务器端口

char\* ftpUser：FTP服务器用户名

char\* ftppPwd：FTP服务器密码

**说明：**

升级控制器的固件

**函数：**

int WINAPI Server\_Update(unsigned char\* PID,LPCWSTR strMd5File,LPCWSTR strUpdateFile,char\* ftpIp,short ftpPort,char\* ftpUser,char\* ftppPwd);

### Server\_UpdateDynamicArea

**返回值：**

成功返回0；失败返回错误号

**参数：**

unsigned char\* PID：控制器唯一ID

DWORD hArea：区域句柄

**说明：**

更新动态区

**函数：**

int WINAPI Server\_UpdateDynamicArea(unsigned char\* PID,DWORD hArea);

### Server\_RemoveDynamicArea

**返回值：**

成功返回0；失败返回错误号

**参数：**

unsigned char\* PID：控制器唯一ID

int AreaId：动态区域编号

**说明：**

删除控制器上的动态区域

**函数：**

int WINAPI Server\_RemoveDynamicArea(unsigned char\* PID,int AreaId);

### Server\_SaveDynamicArea

**返回值：**

成功返回0；失败返回错误号

**参数：**

unsigned char\* PID：控制器唯一ID

int AreaId：动态区域编号

**说明：**

保存控制器上的动态区域

**函数：**

int WINAPI Server\_RemoveDynamicArea(unsigned char\* PID,int AreaId);

### Server\_DelDynamicArea

**返回值：**

成功返回0；失败返回错误号

**参数：**

unsigned char\* PID：控制器唯一ID

**说明：**

删除控制器上保存的动态区域

**函数：**

int WINAPI Server\_RemoveDynamicArea(unsigned char\* PID;