



上海诣阔电子科技有限公司

**EQ
动
态
库
说
明
书**

创建日期：2014-09-08

修改日期：2019-03-05



开发须知本动态库接口适用于：**EQ 火凤凰系列和蓝精灵系列控制器！动态库本身不能多线程！**如果项目确实需要多线程开发要基于信号量或者互斥锁。**注释：添加节目开发模式或者实时发送开发模式，都必须遵循多线程开发在发送结束或者断开连接要基于信号量或者互斥锁释放后才能执行下一个添加节目或者实时发送，未释放信号量或者互斥锁执行下一个添加节目或者实时发送会引起内存溢出或则节目出错。**

火凤凰系列：EQ2013、EQ2023、EQ2033

蓝精灵系列：EQ2012、EQ2011、EQ2008-1/2E、EQ2008-M

一、参数表

1、§ 窗体设置

```
typedef struct    {
    int      iX;           //窗口的起点 X
    int      iY;           //窗口的起点 Y
    int      iWidth;       //窗体的宽度
    int      iHeight;      //窗体的高度
    int      iFrameMode;   //边框的样式
                        //蓝精灵系列：边框宽度，取值范围：0-3
                        //火凤凰系列：边框样式索引，取值范围：1-99，0 无边框
                        // 边框索引值参数“WaterFrame ”文件夹，本文件夹与 DLL 库必
                        // 须在同一级目录
    COLORREF FrameColor;   //边框颜色
}User_PartInfo;
```

2、§ 字体设置

```
typedef struct    {
    char*    strFontName;  //字体的名称
    Int      iFontSize;    //字体的大小
    BOOL     bFontBold;    //字体是否加粗
    BOOL     bFontItalic;  //字体是否是斜体
    BOOL     bFontUnderline; //字体是否带下划线
    COLORREF colorFont;    //字体的颜色
    Int      iAlignStyle;  //对齐方式
                        // 0 一左对齐  // 1 一居中
                        // 2 一右对齐
```

```

        Int            iValignerStyle;        //上下对齐方式
                                                //    0-顶对齐
                                                //    1-上下居中
                                                //    2-底对齐

        Int            iRowSpace;            //行间距
    }User_FontSet;

```

2

3 、 § 动画方式设置

```

typedef struct _User_MoveSet
{
    Int            iActionType;                //节目变换方式
    Int            iActionSpeed;              //节目的播放速度
                                                //蓝精灵系列：本参数为固定值 1
                                                //火凤凰系列：用来控制动画速度，取值范围：1(最快)-20 (最慢)

    BOOL          bClear;                    //是否需要清除背景
    Int            iHoldTime;                //在屏幕上停留的时间，单位 0.1 秒
    Int            iClearSpeed;              //清除显示屏的速度
    Int            iClearActionType;          //节目清除的变换方式
    Int            iFrameTime;               //每帧时间，单位毫秒
                                                //火凤凰系列：参数无效
                                                //蓝精灵系列：用来控制动画速度，取值范围：20(最快)-200(最慢)
}User_MoveSet;

```

动画方式索引

动画方式：iActionType/iClearActionType

火凤凰系列	蓝精灵系列
0 — 随机	0 — 随机
1 — 立即显示，无动画方式/快速清除	1 — 立即显示，无动画方式/快速清除
2 — 左移	2 — 左覆盖
3 — 连续左移	3 — 右覆盖
4 — 右移	4 — 上覆盖
5 — 上移	5 — 下覆盖
6 — 连续上移	6 — 左上角覆盖(直线)
7 — 下移	7 — 右上角覆盖(直线)
8 — 飘雪	8 — 左下角覆盖(直线)
9 — 冒泡	9 — 右下角覆盖(直线)

10 — 分散拉伸
11 — 画卷打开
12 — 画卷闭合
13 — 向左拉伸
14 — 向右拉伸
15 — 向上拉伸
16 — 向下拉伸
17 — 向左镭射
18 — 向右镭射
19 — 向上镭射
20 — 向下镭射
21 — 水平百叶
22 — 垂直百叶
23 — 左覆盖
24 — 右覆盖
25 — 下覆盖
26 — 上覆盖
27 — 左上角覆盖(斜线)
28 — 右上角覆盖(斜线)
29 — 左下角覆盖(斜线)
30 — 右下角覆盖(斜线)
31 — 左上角覆盖(直线)
32 — 右上角覆盖(直线)
33 — 左下角覆盖(直线)
34 — 右下角覆盖(直线)
35 — 左右对开
36 — 上下对开
37 — 左右闭合
38 — 上下闭合
39 — 中间向四周(矩形)
40 — 四周向中间(矩形)
41 — 中间向四周(十字)
42 — 四周向中间(十字)
43 — 中间向四周(菱形)
44 — 四周向中间(菱形)
45 — 闪烁
46 — 中间移出
47 — 左右移入
48 — 左右交叉移动
49 — 左右交叉覆盖
50 — 上下交叉覆盖

10 — 左上角覆盖(斜线)
11 — 右上角覆盖(斜线)
12 — 左下角覆盖(斜线)
13 — 右下角覆盖(斜线)
14 — 水平百叶
15 — 垂直百叶
16 — 左右对开
17 — 上下对开
18 — 左右闭合
19 — 上下闭合
20 — 左移
21 — 右移
22 — 上移
23 — 下移
24 — 马赛克(小)
25 — 马赛克(中)
26 — 马赛克(大)
27 — 渐变
28 — 中间向四周(矩形)
29 — 四周向中间(矩形)
30 — 中间向四周(十字)
31 — 四周向中间(十字)
32 — 中间向四周(菱形)
33 — 四周向中间(菱形)
34 — 画卷打开
35 — 画卷闭合
36 — 中间移出
37 — 左右移入
38 — 向右镭射
39 — 向左镭射
40 — 向下镭射
41 — 向上镭射
42 — 飘雪
43 — 闪烁

4、§ 计时区参数

```
typedef struct _User_Timer
{
    User_PartInfo    PartInfo;           //分区信息
    COLORREF         BkColor;           //背景颜色
    User_FontSet     FontInfo;          //字体设置
    DWORD            ReachTimeYear;      //到达年
    DWORD            ReachTimeMonth;     //到达月
    DWORD            ReachTimeDay;       //到达日
    DWORD            ReachTimeHour;      //到达时
    DWORD            ReachTimeMinute;    //到达分
    DWORD            ReachTimeSecond;    //到达秒
    BOOL             bDay;               //是否显示天 0 一不显示 1一显示
    BOOL             bHour;              //是否显示小时
    BOOL             bMin;               //是否显示分钟
    BOOL             bSec;               //是否显示秒
    BOOL             bMulOrSingleLine;   //单行还是多行
    Char             *chTitle;          //添加显示文字
}User_Timer;
```

5、§ 温度区参数

```
typedef struct _User_Temperature
{
    User_PartInfo    PartInfo;           //分区信息
    COLORREF         BkColor;           //背景颜色
    User_FontSet     FontInfo;          //字体设置
    Char             *chTitle;          //标题
    Int              DisplayType;        //显示格式：一度 1—℃
}User_Temperature;
```

6、§ 日期时间区参数

```
typedef struct _User_DateTime
{
    User_PartInfo    PartInfo;           //分区信息
    COLORREF         BkColor;           //背景颜色
    User_FontSet     FontInfo;          //字体设置
    int              iDisplayType;       //显示风格
    char             *chTitle;          //添加显示文字
    BOOL             bYearDisType;       //年份位数一；一位
    BOOL             bMulOrSingleLine;   //单行还是多行
    BOOL             bYear;              //是否显示年
}
```



```
        BOOL                bMonth;                //是否显示月
        BOOL                bDay;                  //是否显示日
        BOOL                bWeek;                //是否显示星期
        BOOL                bHour;                //是否显示时
        BOOL                bMin;                 //是否显示分
        BOOL                bSec;                 //是否显示秒
    }User_DateTime;
```

7、§ 图文区参数

```
typedef struct _User_Bmp
{
    User_PartInfo    PartInfo;                //分区信息
}User_Bmp;
```

8、§ 单行文本区参数

```
typedef struct _User_SingleText
{
    char                *chContent                //显示内容
    User_PartInfo    PartInfo;                //分区信息
    COLORREF        BkColor;                //背景颜色
    User_FontSet    FontInfo;                //字体设置
    User_MoveSet    MoveSet;                //动作方式设置
}User_SingleText;
```

9、§ 文本区参数

```
typedef struct _User_Text
{
    char                *chContent;                //显示内容
    User_PartInfo    PartInfo;                //分区信息
    COLORREF        BkColor;                //背景颜色
    User_FontSet    FontInfo;                //字体设置
    User_MoveSet    MoveSet;                //动作方式设置
}User_Text;
```



二、函数说明

1、§ 添加节目

int User_AddProgram(int CardNum,BOOL bWaitToEnd,int iPlayTime);

函数功能：添加节目索引

参数说明： **CardNum** — 控制卡地址，基数为 1，即第一块控制卡地址为 1
bWaitToEnd — TRUE 等待节目播放完成再播放下个节目
FALSE 节目播放时间为 **iPlayTime**
iPlayTime — 节目播放时间,单位为秒

返回值： 节目索引号

2、§ 添加图文区

Int User_AddBmpZone(int CardNum,User_Bmp *pBmp,int iProgramIndex);

函数功能：添加图文区

参数说明： **CardNum** — 控制卡地址，基数为 1，即第一块控制卡地址为 1
pBmp — 图文区参数表指针，参考【参数表】中 7
iProgramIndex — 节目索引号

返回值： -1 一添加图文区失败，非-1 一分区编号

3 、§ 向图片区添加图片

BOOL User_AddBmp(int CardNum,int iBmpPartNum,HBITMAP hBitmap,User_MoveSet* pMoveSet,int iProgramIndex);

函数功能：向图片区添加图片

参数说明： **CardNum** — 控制卡地址，基数为 1，即第一块控制卡地址为 1
iBmpPartNum — 图文区编号
hBitmap — 图片句柄
pMoveSet — 节目变换动作方式参数表指针，参考【参数表】中 3
iProgramIndex — 节目索引号

返回值： FALSE 一添加图片失败，TRUE 一添加图片成功

BOOL User_AddBmpFile(int CardNum,int iBmpPartNum, char *strFileName,User_MoveSet* pMoveSet,int iProgramIndex);

函数功能：向图片区添加图片

参数说明： **CardNum** — 控制卡地址，基数为 1，即第一块控制卡地址为 1
iBmpPartNum — 图文区编号

strFileName — 图片名称
pMoveSet — 节目变换动作方式参数表指针，参考【参数表】中 3
iProgramIndex — 节目索引号

返回值： FALSE — 添加图片失败，TRUE — 添加图片成功

4、§ 添加时间区

int User_AddTime(int CardNum,User_DateTime *pDateTime,int iProgramIndex);

函数功能：添加时间区

参数说明： CardNum — 控制卡地址，基数为 1，即第一块控制卡地址为 1
 pDateTime — 时间参数表指针，参考【参数表】中 6
 iProgramIndex — 节目索引号

返回值： -1 — 添加时间区失败，非-1 — 分区编号

5、§ 添加计时区

int User_AddTimeCount(int CardNum,User_Timer *pTimeCount,int iProgramIndex);

函数功能：添加倒计时区

参数说明： CardNum — 控制卡地址，基数为 1，即第一块控制卡地址为 1
 pTimeCount — 倒计时参数表指针，参考【参数表】中 4
 iProgramIndex — 节目索引号

返回值： -1 — 添加计时区失败，非-1 — 分区编号

6、§ 添加温度区

int User_AddTemperature(int CardNum,User_Temperature *pTemperature,int iProgramIndex);

函数功能：添加温度区

参数说明： CardNum — 控制卡地址，基数为 1，即第一块控制卡地址为 1
 pTemperature — 温度参数表指针，参考【参数表】中 5
 iProgramIndex — 节目索引号

返回值： -1 — 添加温度区失败，非-1 — 分区编号

7、§ 添加单行文本区

int User_AddSingleText(int CardNum,User_SingleText *pSingleText,int iProgramIndex);

函数功能：添加单行文本区

参数说明： CardNum — 控制卡地址，基数为 1，即第一块控制卡地址为 1
 pSingleText — 单行文本参数表指针，参考【参数表】中 8
 iProgramIndex — 节目索引号

返回值： -1 — 添加单行文本区失败，非-1 — 分区编号

8、§ 添加文本

`int User_AddText(int CardNum,User_Text *pText,int iProgramIndex);`

函数功能：添加文本区

参数说明： `CardNum` — 控制卡地址，基数为 1，即第一块控制卡地址为 1
 `pText` — 文本参数表指针，参考【参数表】中 9
 `iProgramIndex` — 节目索引号

返回值： -1 —添加文本区失败，非-1 —分区编号

9、§ 添加 RTF 文件区

`int User_AddRTF(int CardNum,User_RTF *pRTFt,int iProgramIndex);`

函数功能：添加文本区

参数说明： `CardNum` — 控制卡地址，基数为 1，即第一块控制卡地址为 1
 `pRTF` — RTF 文件参数表指针，参考【参数表】中 10
 `iProgramIndex` — 节目索引号

返回值： -1 —添加文本区失败，非-1 —分区编号

10、§ 删除一个节目

`BOOL User_DelProgram(int CardNum,int iProgramIndex);`

函数功能：删除节目

参数说明： `CardNum` — 控制卡地址，基数为 1，即第一块控制卡地址为 1
 `iProgramIndex` — 节目索引号

返回值： 0 —删除失败，1 —删除成功

11、§ 删除所有节目

`BOOL User_DelAllProgram(int CardNum);`

函数功能：删除所有节目

参数说明： `CardNum` — 控制卡地址，基数为 1，即第一块控制卡地址为 1

返回值： `FALSE` —删除失败，`TRUE` —删除成功

12、§ 向控制器发送数据

`BOOL User_SendToScreen (int CardNum);`

函数功能：向控制器发送数据

参数说明： `CardNum` — 控制卡地址，基数为 1，即第一块控制卡地址为 1

返回值： `FALSE` —发送失败，`TRUE` —发送成功

13、§ 校正板卡的时间

BOOL User_AdjustTime(int CardNum);

函数功能：校正板卡时间

参数说明： **CardNum** — 控制卡地址，基数为 1，即第一块控制卡地址为 1

返回值： **FALSE** 一板卡校正时间失败，**TRUE** 一板卡校正时间成功

14、§ 打开显示屏

BOOL User_OpenScreen (int CardNum);

函数功能：打开显示屏

参数说明： **CardNum** — 控制卡地址，基数为 1，即第一块控制卡地址为 1

返回值： **FALSE** 一关闭显示屏失败，**TRUE** 一关闭显示屏成功

15、§ 关闭显示屏

BOOL User_CloseScreen (int CardNum);

函数功能：关闭显示屏

参数说明： **CardNum** — 控制卡地址，基数为 1，即第一块控制卡地址为 1

返回值： **FALSE** 一关闭显示屏失败，**TRUE** 一关闭显示屏成功

16、§ 实时发送数据

(1)、建立连接

BOOL User_RealtimeConnect(int CardNum);

(2)、发送数据

BOOL User_RealtimeSendData(int CardNum,int x,int y,int iWidth,int iHeight,HBITMAP hBitmap);

(3、) 断开连接

BOOL User_RealtimeDisConnect(int CardNum);

实时发送数据是专门用在更新频率比较高，要求速度比较快的场合，可以实现区域更新！

实时数据发送可以达到同步控制卡的效果，而且没有更新次数的限制！

使用步骤：

首先，调用一次建立连接函数（1）；

接着，开始调用一次发送数据函数（2），实时向显示屏发送数据，显示屏一直保持原来的数据，直到有新数据到来；

最后，一定要调用一次断开连接函数（3）。

注：每发送一条数据都是（建立连接 发送一次 断开连接），不能建立连接，发送多次，断开连接（或不断开连接） 这样会在几天后出现显示问题

函数功能：实时发送数据（2）

参数说明： **CardNum** — 控制卡地址，基数为 1，即第一块控制卡地址为 1

x,y — 更新区域的左上角坐标；

iWidth, iHeight — 更新区域的宽度和高度；

hBitmap — 更新区域要显示的图片句柄。

注意：实时发送数据是通过网络更新数据！

17、§ 实时发送图片数据

```
BOOL User_RealtimeSendBmpData(int CardNum,int x,int y,int iWidth,int iHeight,char
*strFileName);
```

函数功能：实时发送图片数据

参数说明： CardNum — 控制卡地址，基数为 1，即第一块控制卡地址为 1
x,y — 更新区域的左上角坐标；
iWidth, iHeight — 更新区域的宽度和高度；
strFileName — 需要发送的图片名称。

返回值： FALSE — 发送失败，TRUE — 发送成功

18、§ 实时发送文本

```
BOOL User_RealtimeSendText(int CardNum,int x,int y,int iWidth,int iHeight,char
*strText,User_FontSet *pFontInfo);
```

函数功能：实时发送图片数据

参数说明： CardNum — 控制卡地址，基数为 1，即第一块控制卡地址为 1
x,y — 更新区域的左上角坐标；
iWidth, iHeight — 更新区域的宽度和高度；
strText — 需要发送的字符串。
pFontInfo — 字体结构体类型

返回值： FALSE — 发送失败，TRUE — 发送成功

19、§ 亮度调节

```
BOOL User_SetScreenLight(int CardNum,int iLightDegree);
```

函数功能：调节显示屏亮度

参数说明： CardNum — 控制卡地址，基数为 1，即第一块控制卡地址为 1
iLightDegree — 亮度值，范围是 0~15，取值越大，显示屏亮度越高

返回值： FALSE — 亮度调节失败，TRUE — 亮度调节成功

20、§ 向控制器发送文件数据

```
BOOL User_SendFileToScreen(int CardNum,char pSendPath[MAX_PATH],char
pIndexPath[MAX_PATH]);
```

函数功能：向控制器发送节目文件

参数说明： CardNum — 控制卡地址，基数为 1，即第一块控制卡地址为 1
pSendPath — 节目文件路径 默认生成 send.sys
pIndexPath — 节目索引文件路径 默认生成 index.sys

返回值： FALSE — 发送失败，TRUE — 发送成功

21、§ 清屏

BOOL User_RealtimeScreenClear(int CardNum);

函数功能：清空控制卡内的节目

参数说明： CardNum — 控制卡地址，基数为 1，即第一块控制卡地址为 1

返回值： FALSE — 发送失败，TRUE — 发送成功

22、§ 重载配置文件（EQ2008_Dll_Set.ini）

Void User_ReloadIniFile(char *pEQ2008_Dll_Set_Path);

函数功能：对于那些没有相对路径的调试模式或编译环境没法通过 dll 去调用配置文件，需要手动加载配置文件

参数说明： pEQ2008_Dll_Set_Path — 配置文件路径

三、参数配置文件

EQ 一卡通动态库自动读取当前目录下的“EQ2008_Dll_Set.ini”文件中的所有参数。

参数说明：

[地址： 0]

CardType=21

CardAddress=0

CommunicationMode=0

ScreemHeight=32

ScreemWidth=64

SerialBaud=57600

SerialNum=1

NetPort=5005

IpAddress0=192

IpAddress1=168

IpAddress2=1

IpAddress3=236

ColorStyle = 1

1、控制卡地址"[地址： n]"和"CardAddress" 范围为： 0~1023;

2、控制卡类型"CardType"的取值为：

EQ2008-I/II=5、EQ2011=7、EQ2012=8、EQ2008-M=9

EQ2013=21、EQ2023=22、EQ2033=23

3、控制卡通讯模式“CommunicationMode”的取值为：

串口通讯=0、网路通讯=1

4、显示屏的宽度和高度分别为“ScreemWidth”和“ScreemHeight”，取值为：

ScreemWidth=8 的倍数

5、串口波特率和串口号分别为“SerialBaud”和“SerialNum”，取值为：

SerialBaud=(9600, 19200, 57600, 115200);

(注：当 CardType=EQ2013/EQ2023/EQ2033 时,波特率只能为 9600 或 57600)

SerialNum =(1 为 COM1 口, 2 为 COM2 口);

6、网络端口号 “NetPort” 必须为 5005;

7、参数 “IpAddress” 为 IP 地址：默认值为 192.168.1.236

8、ColorStyle:显示屏颜色类型:0--单色屏, 1--双色屏。(仅 EQ2013、EQ2023、EQ2033 中有此参数)

注意:

*地址的个数可以根据实际显示屏的个数添加;

*不要修改本文件的文件名及后缀;

*本文件必须和应用程序放在同一个目录下。

四、节目编辑流程

第一步：编辑节目

→ 首先利用 User_AddProgram 添加节目，得到节目编号，然后通过下列函数添加显示区域;

→ 利用 User_AddTime 添加时间区

→ 利用 User_AddTimeCount 添加倒计时区

→ 利用 User_AddTemperature 添加温度区

→ 利用 User_AddSingleText 添加单行文本区

→ 利用 User_AddText 添加文本区

→ 利用 User_AddRTF 添加 RTF 文件区

→ 利用 User_AddBmpZone 添加图片区，然后利用 User_AddBmp 或者 User_AddBmpFile 向图片区添加图片

可以利用 User_DelProgram 删除特定节目，或者利用 User_DelAllProgram 删除所有节目。

注意:

各个区位置不能重叠，每个节目最多能添加十个图文区，五个时间区，一个温度区，三个倒计时区;

各个分区的宽度应该为 8 的倍数，X 方向坐标为 8 的倍数

第二步：利用 User_SendToScreen 发送节目数据到控制卡，或者利用 User_SendFileToScreen 直接发送节目文件和节目索引到控制卡。



实时发送数据流程：

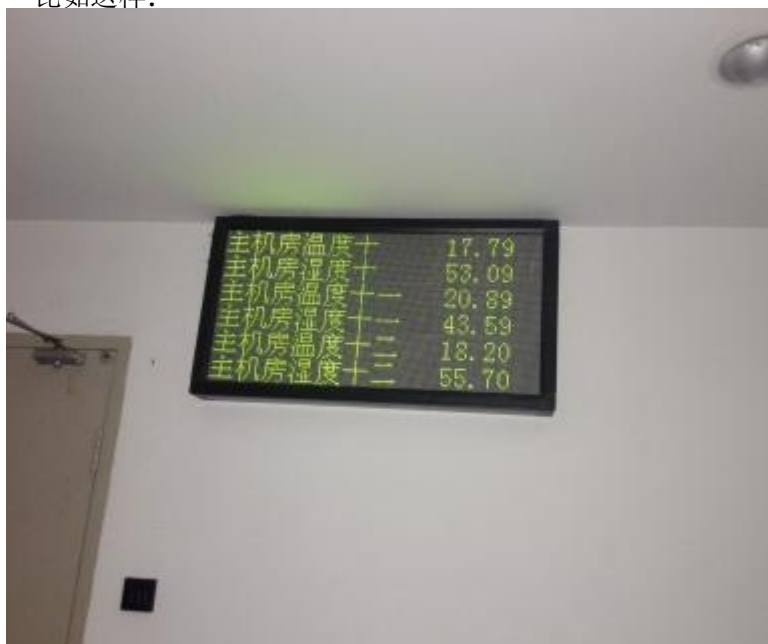
第一步：通过 User_RealtimeConnect 建立连接；

第二步：通过 User_RealtimeSendData 实时发送数据，或者直接通过 User_RealtimeSendBmpData 实时发送图片到控制卡，或者通过 User_RealtimeSendText 实时发送文本到控制卡；

第三步：不再需要发送数据之后，通过 User_RealtimeDisconnect 断开连接。

注：用EQ一卡通调试过硬件发送节目，开发程序代码逻辑初始化流程调用一次清屏接口User_RealtimeScreenClear;建议每发送一条更新数据都是遵循流程：建立连接 -> 发送一次数据 ->断开连接，不建议 建立一次连接，发送多次数据，断开连接（或不断开连接） 这样会在几天后出现显示问题

比如这样：



出问题：

