



Shanghai equal Software Co. Ltd.

Asynchronous full-color secondary development interface description

Create Time: 2014-01-28

ModifyTime: 2016-05-16

Modify department: R & D department

**本动态库接口适用于：
EQ 异步全彩控制卡 所有系列**



一、Parameter table

1、Font settings

typedef struct // Font settings

```
{
    char* strFontName; // Font name
    int iFontSize; // Font size
    BOOL bFontBold; // Whether the font is bold
    BOOL bFontItalic; // Whether the font is oblique
    BOOL bFontUnderline; // Whether the font is underlined
    COLORREF colorFont; // Font color
    int iAlignStyle; // Align 0- left 1- Center 2- right align
                // Only a single line of text on middle and level on the left
    int iValignerStyle; // Top and bottom alignment 0 -top alignment 1-up and down Center
                2-bottom alignment
    int iRowSpace; // Line spacing is not significant for single line text area
}
```

}User_FontSet;

2、Form setting

typedef struct // Form setting

```
{
    int iX; // Window starting point X
    int iY; // Window starting point Y
    int iWidth; // Form Width
    int iHeight; // Form height
    int iFrameMode; // Border style video area without borders
    COLORREF FrameColor; // Border color
}
```

}User_PartInfo;

3、Action stunt

typedef struct User_MoveSet

```
{
    int iActionType; // Program transformation method
    int iActionSpeed; // The broadcast speed of the program 0~50 (0 is the slowest of the
                50 is the fastest)
    BOOL bClear; // need to clear the background ?
    int iHoldTime; // The time to stay on the screen (unit is seconds)
    int iClearSpeed; // Clear display speed 0~50 (0 is the slowest of the 50 is the fastest)
    int iClearActionType; // Transformation mode of program removal
    int iFrameTime; // Per frame time
}
```

}User_MoveSet;

Animation style index

Animation mode: iActionType/iClearActionType	
Parameter Value	Explanation
0	Random
1	Appear immediately
2	Left cover
3	Right cover



4	On the cover
5	Under cover
6	Covering the upper left corner(slash)
7	Covering the upper right corner(slash)
8	Covering the lower left corner (slash)
9	Covering the lower right corner (slash)
10	Covering the upper left corner(straight line)
11	Covering the upper left corner(straight line)
12	Covering the lower left corner (straight line)
13	Covering the lower right corner (straight line)
14	Horizontal blinds
15	Vertical blinds
16	per pale
17	per fess
18	Left and right closure
19	Upper and lower closure
20	Left shift
21	Right shift
22	Move up
23	Move down
24	Mosaic (small)
25	Mosaic (Medium)
26	Mosaic (large)
27	Gradient

4、Picture frame

```
typedef struct User_Bmp
```

```
{  
    User_PartInfo PartInfo; // Partition information  
}User_Bmp;
```

5、Single-line text box

```
typedef struct User_SingleText
```

```
{  
    char* chContent; // Display content  
    User_PartInfo PartInfo; // Partition information  
    COLORREF BkColor; // background color  
    User_FontSet FontInfo; // Font settings  
    User_MoveSet MoveSet; // Action mode settings  
}User_SingleText;
```

6、Text Box

```
typedef struct User_Text
```

```
{  
    char* chContent; // Display content  
    User_PartInfo PartInfo; // Partition information  
    COLORREF BkColor; // background color
```



```
User_FontSet FontInfo; // Font settings
User_MoveSet MoveSet; // Action mode settings
}User_Text;
7、视屏文件
typedef struct User_Video
{
    User_PartInfo PartInfo; // Partition information
    char strFileName[MAX_PATH]; // Video files full path
    BOOL bMute; // Mute
}User_Video;
```

二、Function Description

1. Adding programs

int User_EQ6003_AddProgram(int CardNum, BOOL bWaitToEnd, int iPlayTime);

Function Features: Add program index

Parameter Description: CardNum — Control card address, the base num is 1, the first piece of control card address is 1

bWaitToEnd — TRUE wait for the program has finished playing and then play the next program to be broadcast FALSE is play iPlayTime

iPlayTime — Broadcast time in seconds

Return Value: Program index number

2. Erasing all programs

BOOL User_EQ6003_DelAllProgram(int CardNum);

Function Features: Erasing all programs

Parameter Description: CardNum — Control card address, the base num is 1, the first piece of control card address is 1

Return Value: FALSE — Delete Failed, TRUE — Delete Succeeded

3. Add Photo Area

Int User_EQ6003_AddBmpZone(int CardNum, User_Bmp* pBmp, int iProgramIndex);

Function Features: Add Photo Area

Parameter Description: CardNum — Control card address, the base num is 1, the first piece of control card address is 1

pBmp — Graphic zone parameter table pointer, iProgramIndex — Program index number

Return Value: -1 — Failed to add image area, 非-1 — Partition number

4. Add image to image area

4.1 Add pictures handle

BOOL User_EQ6003_AddBmpHandle(int CardNum, int iBmpPartNum, HBITMAP hBitmap, User_MoveSet* pMoveSet, int iProgramIndex);

Function Features: 向图片区添加图片

Parameter Description: CardNum — Control card address, the base num is 1, the first piece of control card address is 1

iBmpPartNum — Graphic zone number

hBitmap — Pictures handle

pMoveSet — Program converting operation mode parameter table pointer,



iProgramIndex — Program index number

Return Value: FALSE—Add image failed, TRUE—Add image Success

4.2 Add a picture and an absolute path

BOOL User_EQ6003_AddBmpFileName(int CardNum,int iBmpPartNum,char
*strFileName,User_MoveSet*pMoveSet,intiProgramIndex);

Function Features: Add image to image area

Parameter Description: CardNum — Control card address, the base num is 1, the first piece of
control card address is 1

iBmpPartNum — Graphic zone number strFileName — Picture Name pMoveSet — Program
converting operation mode parameter table pointer, iProgramIndex — Program index number

Return Value: FALSE—Add image failed, TRUE—Add image succeeded

5. Adding text

Int User_EQ6003_AddText(intCardNum,User_Text*pText,intiProgramIndex);

Function Features: Adding text area

Parameter Description: CardNum — Control card address, the base num is 1, the first piece of
control card address is 1 pText — Text parameter table pointer iProgramIndex — Program
index number

Return Value: -1—Failed to add text area, 非-1—Partition number

6. Add single line text

Int User_EQ6003_AddSingleText(intCardNum,User_SingleText*pSingleText,intiProgramIndex);

Function Features: Add a single line of text area

Parameter Description: CardNum — Control card address, the base num is 1, the first piece of
control card address is 1 pSingleText — Single-line text pointer parameter table, iProgramIndex
— Program index number

Return Value: -1—Failed to add single text area, 非-1—Partition number

7. Add Video

IntUser_EQ6003_AddVideo(intCardNum,User_Video*pVideo,intiProgramIndex);

Function Features: Add Video

Parameter Description: CardNum — Control card address, the base num is 1, the first piece of
control card address is 1 pVideo— Video area pointer table, iProgramIndex — Program index
number

Return Value: -1—Failed to add video area, 非-1—Partition number

8. Send program

BOOL User_EQ6003_SendToScreen(intCardNum);

Function Features: Send data to the controller

Parameter Description: CardNum — Control card address, the base num is 1, the first piece of
control card address is 1

Return Value: FALSE — Failed to send, TRUE — Send successfully

8.1 Single display content which can continue be sent from the point of interruption

BOOL User_EQ6003_SendToScreen_Continued(int CardNum);

Function Features: When the program failed which can continue be sent from the point of
interruption

Parameter Description: CardNum — Control card address, the base num is 1, the first piece of
control card address is 1



Return Value: FALSE — The content which can continue be sent from the point of interruption is failed, TRUE — The content which can continue be sent from the point of interruption is succeeded

8.2 Concurrent operation sends the content (multiple thread calls)

BOOL User_EQ6003_SendToMultScreen(int SrcCardNum,int DstCardNum);

Function Features: The transmission source screen contents to the target screen (can be multi-threaded) to program a control card, sent to any of the specified control card (including its own, if it's your own program just edited not made any) this function is possible in multi-thread concurrent calls.

Parameter Description: SrcCardNum — Source control card address, DstCardNum — Target control card address, the base is 1, only the first address of a control card 1

Return Value: FALSE — Failed to send, TRUE — Send successfully

8.3 Concurrent operation sends the content which can continue be sent from the point of interruption

BOOL User_EQ6003_SendToMultScreen_Continued(int SrcCardNum,int DstCardNum);

Function Features: Sends the content to the target source panel screen failed which can be sent from the point of interruption

Parameter Description: SrcCardNum — Source control card address, DstCardNum — Target control card address, the base is 1, only the first address of a control card 1

Return Value: FALSE — Failed to send, TRUE — Send successfully

9 实时发送数据

9.1 实时建立连接

BOOL User_EQ6003_RealtimeConnect(int CardNum,int x,int y,int iWidth,int iHeight);

//CardNum — Control card address x Position on the start screen x

//y Starting y position on the screen width

//iHeight height

9.2 Send real-time data

BOOL User_EQ6003_RealtimeSendData(int CardNum,int iWidth,int iHeight,HBITMAP hBitmap);

9.3 Real-time disconnect

BOOL User_EQ6003_RealtimeDisConnect(int CardNum); Send real-time data is specifically used in the update frequency is relatively high, requiring faster occasion, you can achieve regional update!

Real-time data transmission can achieve synchronous control card effect, but did not limit the number of updates!

Steps: First, call the function to establish a connection ,and next start calling function to send data to send real-time data display, the display of the original data remains until new data arrives; Finally, be sure to call the disconnect function.

Function Features: Send real-time data

Parameter Description: CardNum — Control card address, the base 1, namely a control card address is 1. , iWidth 、 iHeight : Update area

10 Turn off the screen

BOOL User_EQ6003_CloseScreen(int CardNum);

Function Features: Turn off led screen

Parameter Description: CardNum — Control card address, the base 1

Return Value: FALSE — turn off the screen failed, TRUE — turn off the screen succeeded

11 Turn on the screen



BOOLUser_EQ6003_OpenScreen(intCardNum);

Function Features: Turn on led screen

Parameter Description: CardNum — Control card address, the base 1

Return Value: FALSE — Turn on led screen failed, TRUE — Turn on led screen succeeded

12 Correction time

BOOLUser_EQ6003_AdjustTime(intCardNum);

Function Features: Correction the controller system time

Parameter Description: CardNum — Control card address, the base 1

Return Value: FALSE — Correction the controller system time failed, TRUE — Correction the controller system time succeeded

13 Adjust the brightness

BOOLUser_EQ6003_LightSet(intCardNum,intiLight);

Parameter Description: CardNum — Control card address, the base 1 iLightDegreen —

Brightness value, range value is 1~16, The higher the value, the higher the display brightness

Return Value: FALSE —Adjust the brightness failed, TRUE —Adjust the brightness succeeded

三、Parameter in the configuration file

EQ6003 or EQ6005 Series Player DLL automatically read in the current directory all the parameters in the “EQ6003_Dll_Set.ini” configuration file.

Parameter Description:

[Screen:0]

CardType=6006

ScreenWidth=256

ScreenHeight=160

NetPort=5050

IpAddress0=192

IpAddress1=168

IpAddress2=1

IpAddress3=230

#EQ6003 or EQ6005serials dynamic library configuration file parameters:

#

1、Control card address "[Screen:n]" range value: 0~1023;

#

2、Control card type "CardType":EQ6003=6003,EQ6005=6005 , EQ 云=6006

#

3、The screen's width and height “ScreenWidth” and “ScreenHeight”

#

4、The network port number “NetPort” default value is 5050;

#

5、Parameter “IpAddressn” is IP Address: default value is 192.168.1.230

#

matters needing attention:

* Address for the m control card corresponding to the configuration file in the following parameters [Screen:m-1];

* The number of addresses can be added according to the number of the actual display screen.;



* Do not modify the file name and suffix of this file.;

* This file must be placed in the same directory as the application..

四、Program editing process

1. First use User_EQ6003_DelAllProgram to clear all the programs
2. Using User_EQ6003_AddPProgram to add the program, and then add the display area through the following functions;
3. Add text area by calling User_EQ6003_AddText
4. Adding a single text area by calling User_EQ6003_AddSingleText
5. Adding the video area by calling User_EQ6003_AddVideo
6. Adding the Graphic area by calling User_EQ6003_AddBmpZone
7. Sending program by calling User_EQ6003_SendToScreen

Attention: Each location area cannot overlap, each program up to add ten graphics and text area, five time zones, a temperature zone, three countdown district; the width of each partition should be to a multiple of 8. X coordinate direction is a multiple of 8

Real time data sending process:

The first Steps: Establish connection through User_EQ6003_RealttimeConnect;

The second Steps: Through the User_EQ6003_RealttimeSendData real-time data transmission to the control card;

The third Steps: No longer need to send data, through the User_EQ6003_RealttimeDisConnect disconnected.

The content which can continue be sent from the point of interruption have the process:

- 1、Add program content
- 2、call User_EQ6003_SendToScreen(intCardNum) or User_EQ6003_SendToMultScreen(int SrcCardNum,intDstCardNum)发送节目
- 3、If the second step returns failed, call User_EQ6003_SendToScreen_Continued(intCardNum) or User_EQ6003_SendToMultScreen_Continued(intSrcCardNum,intDstCardNum)续传

Attention: The new controller support the content which can continue be sent from the point of interruption