Control Card Development API Interface Manual (Java)

Revision record:

|  |  |  |
| --- | --- | --- |
| **date** | **version** | **Modify content** |
| 2019-4-10 | V1.0 | The first version |
| 2019-4-14 | V1.1.1 | Add clock, temperature and humidity, gif playitem interface |
| 2019-4-23 | V1.1.2 | Add picture playitem interface |
| 2019-4-28 | V1.1.3 | Send interface, add formatted text playitem interface (html) |
| 2019-5-20 | V1.2.0 | Add multi-window protocol to send program interface |
| 2019-5-24 | V1.2.1 | Multi-window protocol increases sending scalable pictures |
| 2019-6-6 | V1.2.2 | Add interface for setting communication key of control card |
| 2019-6-21 | V1.2.3 | Support communication to encryption control card (need to set key first) |
| 2019-7-29 | V1.2.4 | Multi-window protocol: add video transmission interface (8200); |
| 2019-8-22 | V1.2.5 | Multi-window protocol:  1, increase the timely switch screen interface;  2. Add and exit the multi-window program display interface.  Fix some known problems. |
| 2019-12-11 | V1.2.6 | Add text alignment attribute  The program file (lpb) can be sent directly without manually generating the program list file (lpp). Automatically generated when sending. )  3. Fix some known problems |
| 2020-3-26 | V1.3.0 | 1. Increase the number of the currently played program and take screenshots 2. Fix some problems already |
| 2020-4-28 | V1.3.1 | 1. Add a text playitem interface that supports any text size 2. Add LmServerAPI interface |
| 2020-5-27 | V1.3.2 | 1. Add the anycast program interface and exit the anycast interface 2. Add setting variable value interface |
| 2020-7-20 | V1.3.3 | 1. Add additional program file interface (FileUploadProtocol and LmServer) 2. Add interfaces (ControlUtil and LmServer) for obtaining the remaining space in the program area 3. Add interfaces for obtaining program file list (ControlUtil and LmServer) 4. Add and delete interfaces for designated program files (ControlUtil and LmServer) 5. Fix some problems already |
| 2020-8-3 | V1.3.4 | Add serial communication interface:  Send program file interface;  Multi-window protocol  control  2. Add formatted user disk interface |
| 2020-12-07 | V1.3.5 | 1、Add interface for sending timely information (ControlUtil)  sendInstantMsg  The LmServer adds the following interfaces  GetCardDevices: get all the devices under the login account  2、LmServer\_GetScreenshot: screenshot  GetPlayingProgram: get the program information (program serial number and content) being played  GetProgramContent: get the content of the specified program  GetMultiProgramContent: get the information of all programs (program serial number and content) |
| 2021-3-4 | V1.3.6 | 1.1.2：  AddTextItem add: sets the background color of the window, and the window is transparent  Add and send text with picture background window  4.2：  Add: the time of obtaining control card  Add: the time of setting control card  Modify: callback interface |
| 2021-4-15 | V1.3.7 | Update catalog: 2.2  Update content: Add program template related interface:  1. Enter the program template mode  2. Send the program  3. Delete the program  4. Enter or exit the program template method |
| 2021-5-27 | V1.3.8 | Update content:  1. Added the support for error retransmission of uploadFileRetryMode  2. Fix some problems |
| 2022-2-16 | V1.4.0 | Update content:  1. The multi-window interface added the timeout period for receiving returned packet data  2. The multi-window interface added the timeout period for obtaining and receiving returned packet data  3.send the program new segment upload  4. multi-window increase call extended font to send text |
| 2022-3-11 | V1.4.1 | Update content:  Increase program operation plan |
| 2022-8-27 | V1.4.2 | Update content:  1. Multi-window protocol network initialization  2. Add the text content of window 1 to obtain the program number  3. Obtain the latest multi-window protocol sendtext content  4. Optimize timeout Settings |
| 2022-09-19 | V1.4.3 | Update content:  1. Adjust the document to supplement the call example of the interface  2. multi-window protocol sendText method began to support the use of labels, window background color: <bkcolor=1>, text size: <size=1>, text color: <color=2>  3. Fix the file name error when creating the lpp program file  4. Add two interfaces, workload counting and timing, to the API for program generation  5. control (network) and control (serial port) increase, restart the App and restart the hardware interface  6.(2022-12-14) Control (network) Add interface: get the value of global variables;  7.LmServer added interface: sendSimpleText -- send formatted text to the specified window  8.increased control (network) : get the current screenshot interface to increase to monitor events, to be able to return the IP address of the control card, controlUtil. SetOnControlListenerNew |
| 2023-02-06 | V1.4.4 | Update content:  1. The LmServer interface: sendSimpleText() adds the parameter compliance judgment. getCardDevices() adds one second delay when receiving the returned packet data to solve the problem of receiving incomplete data;  2.multi-window protocol (network + serial port) :  Add an interface, create a global zone window and clear all global zones;  LmServer:  The content is modified. When sending heartbeat packets, the return packets are no longer received.  3. Multi-window Protocol (Network)  Add an interface and set the high and low levels of 5 I/O ports.  4. Control (network)  Add an interface and set the high and low levels of 5 I/O ports. |
| 2023-04-11 | V1.4.5 | Update content:  1. control, multi-window, LmSever interface added, can set the text Chinese text content as traditional Chinese, text content supports the use of color and size labels;  2. Add font labels to the content of the text program, such as <font=1>, the value is 0~7, which corresponds to the font library file in the system disk;  3. In the control interface, the time date is added to the return object of the api for obtaining program file information |
| 2023-04-27 | V1.4.6 | Update content:  1. Add content to the interface of multi-window network and serial network: If the protocol of the card is switched to the new protocol when executing the method, the protocol of the card will be switched back to the original old protocol after execution; |
| 2023-06-03 | V1.4.6 | Update content:  1. multi-window network interface added content: sendTextRGB() support RGB to change the text color |
| 2023-07-24 | V1.4.6 | Update content:  1. multi-window network interface added content: sendPicture02() support call control card pre-stored gif images; |
| 2024-01-29 | V1.4.7 | Update content:  1. In the "addTextItem" interface: all optional pixel text sizes are supported.  2.Add two parameters in the uploadFile interface, uploadFileAppend interface, uploadFileSegment Interface, and formatUserDisk interface to support WAN communication:  A. Parameter Device unique Mac address;  B.The parameter protocol type can be 0:auto, 1:old, 2:new |
| 2024-01-30 | V1.4.7 | Update content:  1. Add the getCardMacAndPort interface to the control (network) to obtain the MAC address and port of the control card |
| 2024-02-26 | V1.4.7 | Update content:  1. Add parameter isDeleteFileSign to the uploadFile interface and check whether the tag for deleting files is true: yes; false: no. |
| 2024-05-09 | V1.4.7 | Update content:  1.add the getSendTextData interface in the multi-window (serial port) to get the send data (send formatted text to the specified window). |

Part I: Overview

## Interface classification

|  |  |  |
| --- | --- | --- |
| **No** | **Interface** | **explain** |
| 1 | Generate program files and send them | Basic program delivery method, supporting various forms of program combination |
| 2 | Run plan | Create a run plan file with the rsf extension |
| 3 | Multi-window protocol(network) | Single program; Support multiple windows; quick |
| 4 | Multi-window protocol(serial port) | Single program; Support multiple windows; quick |
| 5 | control(network) | Read or set program information; Control program playing |
| 6 | control(serial port) | Read or set program information; Control program playing |
| 7 | LmServer | A set of remote sending interfaces, which need server support |
| 8 | Communication key management | Set the communication key for the control card |

## description of parameter

### 1, color and gray level

|  |  |  |
| --- | --- | --- |
| name | value | A constant name (ProtocolConstant) |
| Single red | 0x01 | COLOR\_TYPE\_MONO |
| Dan Hong 256 gray scale | 0x71 | COLOR\_TYPE\_MONO\_WITH\_GRAY |
| two-tone | 0x03 | COLOR\_TYPE\_RG\_COLOR |
| Two-color 256 gray scale | 0x73 | COLOR\_TYPE\_RG\_COLOR\_WITH\_GRAY |
| various colours | 0x07 | COLOR\_TYPE\_RGB\_COLOR |
| true color | 0x77 | COLOR\_TYPE\_FULL\_COLOR |

### 2,Coding of text and picture display effect

Example of calling: ShowEffect.Random.getEffect ();

|  |  |
| --- | --- |
| Effect code | Effect result |
| Random | randomize |
| Instant | Show immediately |
| Open\_left | Left open |
| Open\_right | Open right |
| Open\_horizontal | Hengzhongkai |
| Open\_vertical | Open vertically |
| Shutter\_vertical | Louver \_ vertical |
| Shift\_left | left shift |
| Shift\_right | right shift |
| Shift\_up | shift up |
| Shift\_down | move down |
| Scroll\_up | Scroll up |
| Scroll\_left | Scroll left |
| Scrollleft\_continuously | Scroll left continuously |
| Scroll\_right | Scroll right |
| Scroll\_right\_continuously | Scroll right continuously |
| Blink | glimmer |
| Shutter\_horizontal | Louver \_ horizontal |
| Open\_clockwise | Expand clockwise |
| Open\_anticlockwise | Expand counterclockwise |
| Windmill\_clockwise | windmill |
| Windmill\_anticlockwise | Windmill \_ counterclockwise |
| Rectangle\_out | Rectangular outward |
| Rectangle\_in | Rectangular inward |
| Corner\_out | The four corners are outward |
| Corner\_in | Four corners inward |
| Round\_out | Round outward |
| Round\_in | Circular inward |
| Open\_top\_left | Expand upper left corner |
| Open\_top\_right | Expand upper right corner |
| Open\_bottom\_left | Expand the lower left corner |
| Open\_bottom\_right | Expand lower right corner |
| Open\_slash | Oblique expansion |
| Open\_backslash | Anti-bevel expansion |
| Slide\_top\_left | Enter in upper left corner |
| Slide\_top\_right | Enter in upper right corner |
| Slide\_bottom\_left | Enter the lower left corner |
| Slide\_bottom\_right | Enter the lower right corner |
| Open\_cross\_in | Oblique entry |
| Open\_cross\_out | Anti-bevel entry |
| Zebra\_cross\_horizontal | Horizontal zebra crossing |
| Zebra\_cross\_vertical | Vertical zebra crossing |
| Mosaic\_large | Mosaic \_ big |
| Mosaic\_small | Mosaic \_ small |
| Laser\_line\_upward | Radioactivity \_ up |
| Laser\_line\_downward | Radioactivity \_ down |
| Scrape\_up | gather |
| Drop\_down | drop |
| Slide\_left\_right | Merge \_ level |
| Slide\_top\_bottom | Merge \_ vertical |
| Slewing\_out | back-out |
| Slewing\_in | screwing in |
| Chessboard\_horizontal | Chessboard \_ horizontal |
| Chessboard\_vertical | Chessboard \_ vertical |
| Scroll\_up\_continuously | Scroll up continuously |
| Scroll\_down\_continuously | Scroll down continuously |
| Expand\_from\_top | Gradually become larger \_ up |
| Expand\_from\_bottom | Gradually get bigger \_ down |
| Expand\_vertical | Gradually getting bigger \_ vertical |
| Blind\_horizontal | Flashing \_ level |
| Blind\_vertical | Flashing \_ vertical |
| Snow\_fall | drifting snow |
| Scroll\_down | Scroll down |
| Open\_left\_right | Expand left and right |
| Open\_up\_down | Expand up and down |
| Open\_2\_fan | Sector expansion |
| Slide\_zebra\_horizontal | Zebra bar \_ level |
| Slide\_zebra\_vertical | Zebra strip \_ vertical |

### 3,Image scaling method coding

|  |  |  |
| --- | --- | --- |
| **Coded value** | **Image processing mode** | **A constant value (ProtocolConstant)** |
| 0 | centre | SHOWMODE\_CENTER |
| 1 | Scale in proportion | SHOWMODE \_SCALE |
| 2 | stretching | SHOWMODE \_STRETCH |
| 3 | tile | SHOWMODE \_TILED |

### 4,Text size

**Note: the dot matrix size of the extended font supported by each control card is different,If the text is not displayed under a certain size, the dot matrix size is not supported**

|  |  |  |
| --- | --- | --- |
| **Coded value** | **Text size (points)** | **A constant value (ProtocolConstant)** |
| 0 | 8 | FONTSIZE\_8 |
| 1 | 12 | FONTSIZE\_12 |
| 2 | 16 | FONTSIZE\_16 |
| 3 | 24 | FONTSIZE\_24 |
| 4 | 32 | FONTSIZE\_32 |
| 5 | 40 | FONTSIZE\_40 |
| 6 | 48 | FONTSIZE\_48 |
| 7 | 56 | FONTSIZE\_56 |

### 5,Clock attributes

Class: com.lumen.ledcenter3.protocol.clockattrib.

#### 5.1,Attributes

Year: displays the year

Month: displays the month

Day: display the day

Hour: Displays the hour

Min: displays the minutes

Sec: display seconds

Week: display the week

TimeHand: display pointer

TimeDegree: displays time scales and sub-scales

Note: For the above attributes, passing 0 means not displaying, and passing 1 means displaying

TransEnable: transparent display (0: opaque; 1: transparent)

TweentyFour: time system (0: 12 hour system; 1: 24 hour system)

TwoDigitYear: year digits (0: 4 digits; 1: 2 bits)

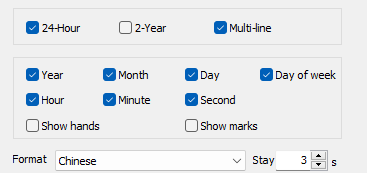
Branch: branch (0: single line; 1: multiple lines).

TimePattern: time format and value, as follows

|  |  |
| --- | --- |
| format | value |
| Friday, November 12, 2010 16:20:30 | 0 |
| Fri，12112010 16:20:30 | 1 |
| 2010-11-12 Fri. 16:20:30 | 2 |
| Friday，12 November 2010 16:20:30 | 3 |
| Fri，Nov 12,2010 16:20:30 | 4 |
| Friday，November 12 2010 16:20:30 | 5 |
| Fri，11122010 16:20:30 | 6 |
| 20101112，Fri.16:20:30 | 7 |

#### 5.2,Default options

Check to display as shown in the following figure



#### 5.3,Instructions for use

1,using an empty constructor is the default option

### 6,Time zone adjustment value

Target time zone minus local time zone.

For example, if the local time zone is +08:00 and the target time zone is +03:00, the adjusted value is -05:00 .

Part II: Interface Details

# 1,generate and send program files

## 1.1,API interface for generating programs

The following interfaces are all in the com.lumen.ledcenter3.protocol.programcreator class.

### 1.1.1 overview of API interface for program generation

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **API interface function name** | **Interface description** | **completeness** |
| 1 | ProgramCreator constructor | Create a program object | Achieved |
| 2 | addWindow | Add a play window to the program | Achieved |
| 3 | addTextItem | Add protocol text playitems to the specified window | Achieved |
| 4 | addPicture | Add picture playitems to the specified window | Achieved |
| 5 | addAnimator | Adds animation (gif) playitems to the specified window | Achieved |
| 6 | addClock | Add a clock playitem to the specified window | Achieved |
| 7 | addTemperature | Adds a temperature playitem to the specified window | Achieved |
| 8 | addHumidity | Add humidity playitems to the specified window | Achieved |
| 9 | createLpbFile | Create a program file and save the program to the file | Achieved |
| 10 | addExtraTextItem | Adds a text playitem to the specified window (supports fonts of any size) | Achieved |
| 11 | addFormTextItem | Add a text playitem (HTML text) to the specified window | Achieved |
| 12 | addTextItemWithPictureBackground | Adds a background image text player to the specified window | Achieved |
| 13 | addWorkCounter | Work count | Achieved |
| 14 | addTimeCounter | timekeeping | Achieved |
| 15 | setTextComplex | Set the text in the text program to traditional Chinese | Achieved |
| 16 | getTextComplex | Gets whether the text in the text program is in traditional Chinese | Achieved |

**The use steps of this kind of function are as follows:**

Step 1: Create program objects

Step 2: Increase the playback window

Step 3: Add play items to the designated play window

Step 4: Create an Lpb file and save the program to the file

### 1.1.2,Detailed description of API interface for program generation

#### ProgramCreator

|  |  |
| --- | --- |
| ProgramCreator (int screenWidth, int screenHeight, int colorType) | |
| function | Create a program object |
| parameter | ScreenWidth: the width of the display screen in points |
| ScreenHeight: the height of the display screen in points |
| ColorType: color and gray level. [color and gray level](#_1, color and gray level) |
| Return value | without |
| Other instructions | Unique constructor |

#### addWindow

|  |  |
| --- | --- |
| int addWindow (int startX, int startY, int windowWidth, int windowHeight) | |
| function | Add a play window to the program |
| parameter | StartX: the starting X coordinate of the playback window |
| StartY: the starting Y coordinate of the playback window |
| WindowWidth: width of playing window |
| WindowHeight: height of playing window |
| Return value | Playback window serial number |
| Other instructions | When adding a playitem, specify the playwindow serial number |

#### Add text item (modified in v1.3.6)

|  |  |
| --- | --- |
| ProgramCreator addTextItem (int winNo, String textContent, int textColor, int textSize, int speed, int stayTime, int showEffect, int hAlign, int vAlign, int lineTween,  boolean windowTransparent, int winColor) | |
| function | Add text items to the specified playback window |
| parameter | WinNo: play window number |
| TextContent: Text string, supporting the use of text labels, fonts, colors, sizes |
| TextSize: text size , [text size](#_4,Text size)  Use constant values:  24 pixels:ProtocolConstant.***FONTSIZE\_24***  16 pixels:ProtocolConstant.***FONTSIZE\_16*** |
| TextColor: text color,Such as 0x00ffffff |
| ShowEffect: text display effect , [Coding of text and picture display effect](#_2,Coding of text and picture display effect) |
| Speed: the speed of text display |
| StayTime: stay time |
| (the following are optional parameters) |
| HAlign: horizontal alignment: 0- left; 1- medium; 2- right |
| VAlign: vertical alignment: 0- up; 1- medium; 2- down |
| LineTween: line spacing (0-15) |
| WindowTransparent：Whether the background is transparent (default is opaque) |
| WinColor: background color value (default is black) |
| Return value | Current program object |
| Other instructions | Plain text program |

#### addTextItem (Expanded font library, new V1.4.0)

|  |  |
| --- | --- |
| ProgramCreator addTextItem (int winNo, String textContent, int textColor, int textSize, int speed, int stayTime, int showEffect, int hAlign, int vAlign, int lineTween,  Int winColor, int isExt) | |
| function | Add text items to the specified playback window |
| parameter | WinNo: play window number |
| TextContent: a text string |
| TextColor: text color |
| TextSize: text size ， [text size](#_4,Text size)  Use constant values:  24 pixels:ProtocolConstant.***FONTSIZE\_24***  16 pixels:ProtocolConstant.***FONTSIZE\_16*** |
| Speed: the speed of text display |
| StayTime: stay time |
| ShowEffect: text display effect， [Coding of text and picture display effect](#_2,Coding of text and picture display effect) |
| (the following are optional parameters) |
| HAlign: horizontal alignment: 0- left; 1- medium; 2- right |
| VAlign: vertical alignment: 0- up; 1- medium; 2- down |
| LineTween: line spacing (0-15) |
| BackPicturePath: background picture path (absolute path) |
|  |
| Return value | Current program object |
| Other instructions | Plain text program |

#### addPicture

|  |  |
| --- | --- |
| ProgramCreator addPicture(int winNo, String picFile, int showMode, int showEffect, int speed, int stayTime) | |
| function | Add picture items to the specified playback window |
| parameter | WinNo: play window number |
| PicFile: path and name of picture file |
| ShowMode: Processing mode when displaying ,[Image scaling method coding](#_3,Image scaling method coding) |
| ShowEffect: picture display effect ,[Coding of text and picture display effect](#_1, color and gray level) |
| Speed: the display speed of the effect. 0 is the fastest |
| StayTime: stay time, in seconds |
| Return value | Current program object |
| Other instructions |  |

#### addAnimator

|  |  |
| --- | --- |
| ProgramCreator addAnimator(int winNo, String aniFile, int showMode, int repeat) | |
| function | Add animation items to the specified playback window |
| parameter | WinNo: play window number |
| AniFile: path and name of animation file (.gif) |
| ShowMode: Processing mode when displaying , [image scaling encoding](#_3,Image scaling method coding) |
| Repeat: the number of times the animation plays in a loop |
| Return value | Current program object |
| Other instructions |  |

#### addClock

|  |  |
| --- | --- |
| ProgramCreator addClock(int winNo, String extraText, int textSize, int textColor, int stayTime, ClockAttrib clockAttrib, String timeZoneAjustValue) | |
| function | Add clock items to the specified playback window |
| parameter | WinNo: play window number |
| ExtraText: a text string |
| TextColor: text color |
| TextSize: text size ， [text size](#_4,Text size) |
| StayTime: stay time |
| ClockAttrib: clock attribute , [clock attribute](#_5,Clock attributes) |
|  | TimeZoneAjustValue: time zone adjustment value , [time zone adjustment value](#_6,Time zone adjustment value) |
| Return value | Current program object |
| Other instructions |  |

#### addTemperature

|  |  |
| --- | --- |
| ProgramCreator addTemperature(int winNo, String extraText, int textSize, int textColor, int stayTime, int tempAttrib) | |
| function | Add a temperature item to the specified playback window |
| parameter | WinNo: play window number |
| ExtraText: a text string |
| TextSize: text size , [text size](#_4,Text size) |
| TextColor: text color |
| StayTime: stay time |
| TempAttrib: temperature attribute  0: Celsius temperature  1: Fahrenheit temperature |
| Return value | Current program object |
| Other instructions |  |

#### addHumidity

|  |  |
| --- | --- |
| ProgramCreator addHumidity(int winNo, String extraText, int textSize, int textColor, int stayTime) | |
| function | Add humidity items to the designated playback window |
| parameter | WinNo: play window number |
| ExtraText: a text string |
| TextSize: text size , [text size](#_4,Text size) |
| TextColor: text color |
| StayTime: stay time |
| Return value | Current program object |
| Other instructions |  |

#### createLpbFile

|  |  |
| --- | --- |
| void createLpbFile (String path, String fileName) | |
| function | Create a program file and save the program to the file |
| parameter | Path: Lpb file storage directory |
| FileName: file name. The file name cannot exceed 8 bytes (2 Chinese characters or 8 numbers and letters) |
| Return value | without |
| Other instructions |  |

#### addExtraTextItem

|  |  |
| --- | --- |
| ProgramCreator addExtraTextItem (int winNo, String textContent, int textColor,  int textSize, String fontFamily, int speed, int stayTime, int showEffect, int hAlign, int vAlign, int lineSpace) | |
| function | Adds a text playitem to the specified window (supports fonts of any size) |
| parameter | WinNo: play window number  TextContent: text content  TextColor: text color value, such as 0xff0000  TextSize: text size (in pixels)  FontFamily: fonts, such as' Tahoma', must be existing fonts on the computer  Speed: the text display speed is 0-100, and the larger it is, the faster it is (related to the display effect showEffect)  StayTime: the text stay time  ShowEffect: text display effect, Coding of text and picture display effect  HAlign: horizontally aligned, 0 left 1 middle 2 right  VAlign: vertically aligned, 0 up 1 middle 2 down  LineSpace: Line spacing (in pixels) |
| Return value | Current program object |
| Other instructions |  |

#### addFormatTextItem

|  |  |
| --- | --- |
| ProgramCreator addTextItem (int winNo, String htmlContent, int speed, int stayTime, int showEffect) | |
| function | Add text items to the specified playback window |
| parameter | WinNo: play window number |
| Htmlcontent: strings in html format, such as:  "<p><span style=\"color:#FF0000; font-size:18px; Font-family: Chinese script; \">Welcome<span><p>" |
| ShowEffect: text display effect , text and picture display effect coding  Common effect: Scrollleft\_continuously scroll left continuously |
| Speed: the speed of text display |
| StayTime: stay time |
| Return value | Current program object |
| Other instructions | 1、Html does not support js and < img >  2、< META > or < meta > tags are not supported in Html |

#### addTextItemWithPictureBackground（Added in V1.3.6）

|  |  |
| --- | --- |
| ProgramCreator addTextItemWithPictureBackground(int winNo, String textContent, int textColor, int textSize, int speed, int stayTime, int showEffect, int hAlign, int vAlign, int lineTween, String backPicturePath) | |
| function | Adds a text item to the specified playwindow |
| parameter | winNo: indicates the player window number |
| textContent: text string |
| textSize: Text size  Use constant values:  24 pixels：ProtocolConstant.***FONTSIZE\_24***  16 pixels：ProtocolConstant.***FONTSIZE\_16*** |
| textColor: text color |
| showEffect: text display effect |
| speed: text display speed |
| stayTime: residence time |
| (The following parameters are optional) |
| hAlign: Horizontal alignment: 0-left; 1- medium; 2- Right |
| vAlign: Vertical alignment: 0- up; 1- medium; 2- Down |
| lineTween: Line spacing (0-15) |
| backPicturePath: Background image path (absolute path) |
|  |
| Return value | Current program object |
| Other instructions | Text-only program |

#### addWorkCounter （Added in V1.4.3）

Example of the method:



|  |  |
| --- | --- |
| ProgramCreator addWorkCounter (int winNo, int textColor, int textSize, int stayTime, int countType, int hoursOfDay,int selectedWeek, int valueAdjust, int startFrom, int baseDateYear,int baseDateMonth,int baseDateDay) | |
| function | Adds a work count to the specified playback window |
| parameter | winNo: indicates the player window number |
| textColor: text color |
| textSize: indicates the text size |
| stayTime: residence time |
| countType: indicates the count type. 0: number of days, 1: number of hours |
| hoursOfDay: How many hours are worked in a day |
| selectedWeek: The day of the week is selected;  For example, 0011 1110 indicates that Sunday and Saturday are not selected, and all other bits are selected. Sunday is the first bit, Monday is the second bit, 0 is not selected, and 1 is selected |
| valueAdjust: Adjusts the value |
| startFrom: What time do you start |
| baseDateYear: indicates the year of the base date |
| baseDatelonth: The month of the base date |
| baseDateDay: indicates the day of the base date |
| Return value | Current program object |
| Other instructions |  |

#### addTimeCounter （Added in V1.4.3）

Example of the method:



|  |  |
| --- | --- |
| ProgramCreator addTimeCounter (int winNo, int textColor, int textSize,int fontIndex, int stayTime,String timeCountContent, int baseTimeHour,int baseTimeMinute,int baseTimeSecond,int timeCountType,int baseDateYear,int baseDateMonth,int baseDateDay) | |
| function | Adds a timer to the specified playback window |
| parameter | winNo: indicates the player window number |
| textColor: text color |
| textSize: Text size |
| fontIndex: indicates the font index. The value ranges from 0 to 7 |
| stayTime: residence time |
| timeCountContent：Timing content; For example: "{! d}days” |
| baseTimeHour:The base time |
| baseTimeMinute：The minutes of the base time |
| baseTimeSecond：Second of the base time |
| timeCountType：Timing type; 0: active timer, 1: countdown timer |
| baseDateYear：The year of the base date |
| baseDateMonth：Month of the base date |
| baseDateDay：The day of the base date |
| Return value | Current program object |
| Other instructions |  |

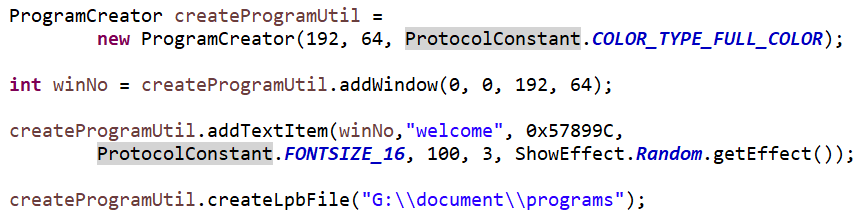
#### setTextComplex（Added in V1.4.5）

|  |  |  |
| --- | --- | --- |
| void setTextComplex (int textComplex) | | |
| function | Sets whether the text in the program is in traditional Chinese | |
| parameter | textComplex: indicates whether the text is traditional, 0: no, 1: traditional | |
| Return value |  |
| Other instructions |  | |

#### getTextComplex（Added in V1.4.5）

|  |  |  |
| --- | --- | --- |
| int getTextComplex () | | |
| function | Check whether the text in the text program is in traditional Chinese | |
| parameter |  | |
| Return value | textComplex: indicates whether the text is traditional, 0: no, 1: traditional |
| Other instructions |  | |

### 1.1.3,Demo calling example



## ~~1.2,API interface for generating program list~~

**Note: After the program file is generated, it can be sent directly by using the interface of "Send Program File", without having to create a program list.**

~~The following interfaces are all in the com.lumen.ledcenter3.protocol.playbillcreator class~~

### ~~1.2.1 overview of API interface for program list generation~~

|  |  |  |  |
| --- | --- | --- | --- |
| **~~serial number~~** | **~~API interface function name~~** | **~~Interface description~~** | **~~completeness~~** |
| ~~one~~ | ~~PlaybillCreator constructor~~ | ~~Create program guide object~~ | ~~Achieved~~ |
| ~~2~~ | ~~addLpbFile~~ | ~~Add program files to the program list~~ | ~~Achieved~~ |
| ~~three~~ | ~~createLppFile~~ | ~~Create a program guide file and save the program guide to the file~~ | ~~Achieved~~ |

**~~The use steps of this kind of function are as follows:~~**

~~Step 1: Create a program guide object~~

~~Step 2: Add program files to the program list~~

~~Step 3: Save the program to a file~~

### ~~1.2.2,detailed description of API interface for program list generation~~

#### ~~PlaybillCreator constructor~~

|  |  |
| --- | --- |
| ~~PlaybillCreator (int screenWidth, int screenHeight, int colorType)~~ | |
| ~~function~~ | ~~Create program guide object~~ |
| ~~parameter~~ | ~~ScreenWidth: the width of the display screen in points~~ |
| ~~ScreenHeight: the height of the display screen in points~~ |
| ~~Color: color and gray level. 1.4, color and gray level~~ |
| ~~Return value~~ | ~~without~~ |
| ~~Other instructions~~ |  |

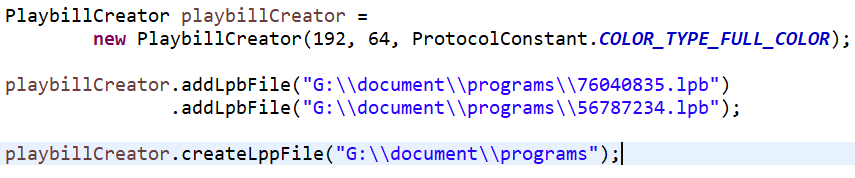
#### ~~addLpbFile~~

|  |  |
| --- | --- |
| ~~PlaybillCreator addLpbFile (String filepath)~~ | |
| ~~function~~ | ~~Add program files to the program list~~ |
| ~~parameter~~ | ~~Filepath: the program file path~~ |
| ~~Return value~~ | ~~Program object~~ |
| ~~Other instructions~~ |  |
| ~~Other instructions~~ |  |

#### ~~createLppFile~~

|  |  |
| --- | --- |
| ~~void createLppFile (String path)~~ | |
| ~~function~~ | ~~Save the program to a file~~ |
| ~~parameter~~ | ~~Path: save the directory of the program~~ |
| ~~Return value~~ | ~~without~~ |
| ~~Other instructions~~ |  |

### ~~1.2.3,Demo calling example~~



## 1.3,interface for sending program files

Network:

com.lumen.ledcenter3.protocol.FileUploadProtocol

com.lumen.ledcenter3.protocol.FileUploadProtocol.OnUploadListener

Serial port: (added in V1.3.4)

com.lumen.ledcenter3.protocol.FileUpload\_COM

com.lumen.ledcenter3.protocol.FileUpload\_COM.OnUploadListener

### 1.3.1,Overview of interface for sending program files

FileUploadProtocol

|  |  |  |
| --- | --- | --- |
| **serial number** | **API interface function name** | **Interface description** |
| 1 | setListener | Set send listener |
| 2 | uploadFile | Send program file |
| 3 | uploadFileAppend | Additional transmission program file |
| 4 | uploadFileSegment | upload files |

FileUploadProtocol.OnUploadListener

|  |  |  |
| --- | --- | --- |
| **serial number** | **API interface function name** | **Interface description** |
| 1 | onStatus | Listen for connection status |
| 2 | onProcess | Monitor the sending progress |

FileUpload\_COM

|  |  |  |
| --- | --- | --- |
| **serial number** | **API interface function name** | **Interface description** |
| 1 | setUploadListener | Set send listener |
| 2 | uploadFile | Send program file |

FileUpload\_COM.OnUploadListener

|  |  |  |
| --- | --- | --- |
| **serial number** | **API interface function name** | **Interface description** |
| 1 | onStatus | Listen for connection status |
|  |  |  |
|  |  |  |
| 2 | onProcess | Monitor the sending progress |

### 1.3.2,Detailed description of program transmission interface

#### Network:

#### setListener

|  |  |
| --- | --- |
| void setListener (OnUploadListener listener) | |
| function | Set up a sending monitor to monitor the sending progress and connection status of programs |
| parameter | Listener: Listener object |
| Return value | without |
| Other instructions |  |

#### uploadFile

|  |  |
| --- | --- |
| void uploadFile(String tempDir, List<String> lpdFilePaths, int screenWidth, int screenHeight, int colorType, String ip, int port, int socketindex) | |
| function | Send program file |
| parameter | TempDir: directory where temporary files are stored |
|  | LpdFilePaths: list of lpd file paths to send |
|  | ScreenWidth: screen width. It needs to be consistent with the program file |
|  | ScreenHeight: screen height. It needs to be consistent with the program file |
|  | ColorType: screen color and gray level. It needs to be consistent with the program file |
|  | Ip: control card communication IP |
|  | Port: control card communication port |
|  | Socketindex: network communication number |
|  | deviceMac: unique Mac address of a device |
|  | protocolModel: indicates that the protocol type is 0:auto, 1:old, and 2:new |
|  | isDeleteFileSign: Indicates whether to delete files. = true: yes, false: no |
|  |  |
| Return value | without |
| Other instructions | 1、Temporary folders and files will be generated during sending, and will be cleared next time (same sending number);  2、This interface is network communication, which takes a long time. Please call it in the child thread |

#### Upload file append (added in v1.3.3)

|  |  |
| --- | --- |
| void uploadFileAppend(String tempDir, List<String> lpdFilePaths, int screenWidth, int screenHeight, int colorType, String ip, int port, int socketindex) | |
| function | Additional transmission program file |
| parameter | TempDir: directory where temporary files are stored |
|  | LpdFilePaths: list of lpd file paths to send |
|  | ScreenWidth: screen width. It needs to be consistent with the program file |
|  | ScreenHeight: screen height. It needs to be consistent with the program file |
|  | ColorType: screen color and gray level. It needs to be consistent with the program file |
|  | Ip: control card communication IP |
|  | Port: control card communication port |
|  | Socketindex: network communication number |
|  | deviceMac (Optional) : unique Mac address of a device |
|  | protocolModel (Optional) : indicates a protocol type of 0:auto, 1:old, and 2:new |
| Return value | without |
| Other instructions | 3、Temporary folders and files will be generated during sending, and will be cleared next time (same sending number);  4、This interface is network communication, which takes a long time. Please call it in the child thread |

#### uploadFileSegment（Segmented upload - New in V1.4.0）

|  |  |
| --- | --- |
| void uploadFileSegment(int fileType, boolean isSystemFile, String filePath, int sendIndex) | |
| function | upload files |
| parameter | fileTypeParam: indicates the file type. 1: indicates a common file |
|  | isSystemFile: indicates whether it is a system disk file. true: indicates a system disk file; false: indicates a user disk file |
|  | filePath : indicates the file path |
|  | sendIndex: indicates the index of the current file |
|  | deviceMac (Optional) : unique Mac address of a device |
|  | protocolModel (Optional) : indicates a protocol type of 0:auto, 1:old, and 2:new |
| Return value | Without |
| Other instructions | 5. This interface is used for network communication, which takes a long time. Please call it in the subthread  Note: methods in com) lumen. Ledcenter3. Protocol. The FileUpload\_Net call in the class.  Example of parameters Protocol Type and Mac Address: |

#### onStatus

|  |  |
| --- | --- |
| void onStatus (int status, int socketindex) | |
| function | Receive the network connection status |
| parameter | Status: connection status, 0 is connection failure, 1 is success, and -1 is exception |
|  | Socketindex; Network communication number |
| Return value | without |
| Other instructions |  |

#### onProcess

|  |  |
| --- | --- |
| void onProcess (int process, int totalprocess, int socketindex) | |
| function | File upload progress |
| parameter | Process: current progress (uploaded data size) |
|  | Totalprocess: total progress (total data size) |
|  | Socketindex; Network communication number |
| Return value | without |
| Other instructions |  |

#### Serial port: (added in V1.3.4)

#### setUploadListener

|  |  |
| --- | --- |
| void setUploadListener (OnUploadListener listener) | |
| function | Set up a sending monitor to monitor the sending progress and connection status of programs |
| parameter | Listener: Listener object |
| Return value | without |
| Other instructions |  |

#### uploadFile

|  |  |
| --- | --- |
| void uploadFile(String tempDir, List<String> lpdFilePaths, int screenWidth, int screenHeight, int colorType, String portName, int baudrate, int sendIndex) | |
| function | Send program file |
| parameter | TempDir: directory where temporary files are stored |
|  | LpdFilePaths: list of lpd file paths to send |
|  | ScreenWidth: screen width. It needs to be consistent with the program file |
|  | ScreenHeight: screen height. It needs to be consistent with the program file |
|  | ColorType: screen color and gray level. It needs to be consistent with the program file |
|  | PortName: serial port number of control card |
|  | Baudrate: control the baud rate of serial port |
|  | SendIndex: communication number |
| Return value | without |
| Other instructions | 6、Temporary folders and files will be generated during sending, and will be cleared next time (same sending number);  7、This interface is network communication, which takes a long time. Please call it in the child thread |

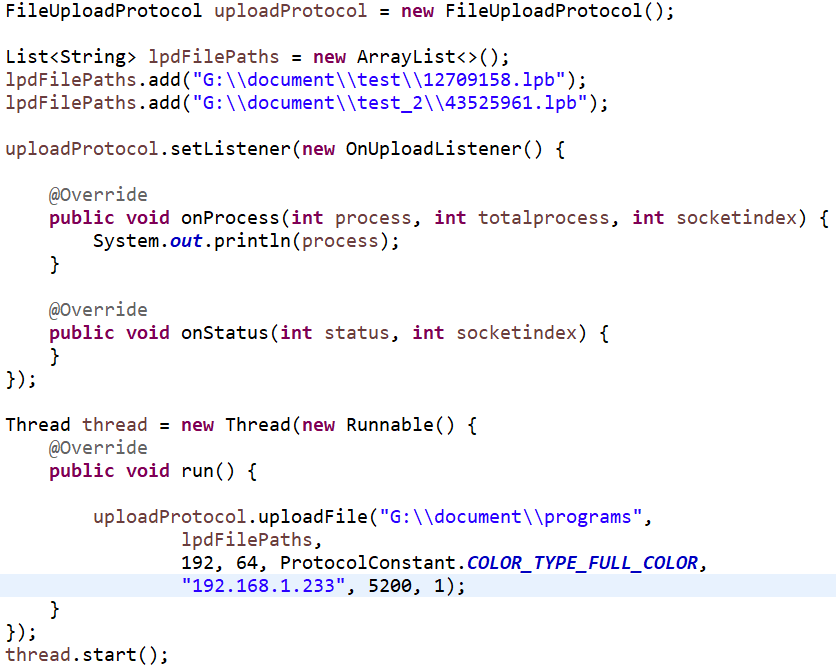
#### onStatus

|  |  |
| --- | --- |
| void onStatus (int status, int sendIndex) | |
| function | Receive serial port connection status |
| parameter | Status: connection status, 0 is connection failure, 1 is success, and -1 is exception |
|  | sendIndex; Communication number |
| Return value | without |
| Other instructions |  |

#### onProcess

|  |  |
| --- | --- |
| void onProcess (int process, int totalprocess, int sendIndex) | |
| function | File upload progress |
| parameter | Process: current progress (uploaded data size) |
|  | Totalprocess: total progress (total data size) |
|  | sendIndex; Communication number |
| Return value | without |
| Other instructions |  |

### 1.3.3,Demo calling example

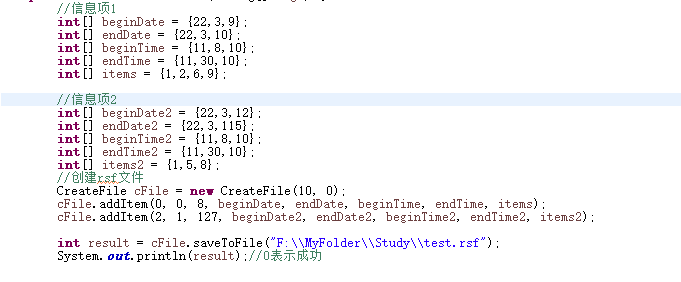


# **2.**Operation Plan (added in V1.4.1)

## 2.1 Overview of the Run Plan Interface

|  |  |  |
| --- | --- | --- |
| 1 | new CreateFile | Create a run plan object |
| 2 | void addItem | Add a run plan item |
| 3 | int saveToFile | Save the run plan to a file |

## Call example



## Operational plan details

#### new CreateFile(int proSum, int attrib);

|  |  |
| --- | --- |
| CreateFile cFile = **new** CreateFile(**int** proSum, **int** attrib); | |
| function | Create a run plan object |
| parameter | proSum: Total number of programs |
| attrib: attributes  0: No program is played in the default time range  1: All programs are played within the default time range |
| returned value | Without |
| Other instructions |  |

#### void addItem(…params)

|  |  |
| --- | --- |
| **void** addItem(**int** nGrade, **int** nWeekDateRelative, **int** nWeeks,  **int**[] pBeginDate, **int**[] pEndDate, **int**[] pBeginTime,  **int**[] pEndTime, **int**[] pItems) | |
| function | Add a run plan item |
| parameter | nGrade: Project level. The higher the level from 0 to 9, the higher the priority. |
| nWeekDateRelative: Relation of week to date  0: The plan shall be executed only when both the week and the date are met  1: Execute this plan when one of the weeks or dates is met |
| nWeeks: Indicates the mark of the week, which can be one or a combination of the following values  1: Sunday  2: Monday  4: Tuesday  8: Wednesday  16: Thursday  32: Friday  64: Saturday |
| pBeginDate: indicates the start date. Three integer values represent "year", "month" and "day" respectively. |
| pEndDate: indicates the end date. Three integer values represent "year", "month" and "day" respectively. |
| pBeginTime: indicates the start time. The three integer values represent "hours", "minutes" and "seconds" respectively. |
| pEndTime: indicates the end time. The three integer values represent "hours", "minutes" and "seconds" respectively. |
| pItems: The number of the program to be played. Each integer is the number of the program to be played, numbered from 1 and less than the number specified by "proSum" in the first parameter of new CreateFile(int proSum,int attrib) |
| returned value |  |
| Other instructions |  |

#### int saveToFile(String fileName)；

|  |  |
| --- | --- |
| **int** saveToFile(String fileName) | |
| function | Save the run plan to a file |
| parameter | fileName: indicates the file path and name |
| returned value | >=0: Success  <0: failed |
| Other instructions |  |

# 3,Multi-window protocol（Network）

## com.lumen.ledcenter3.protocol.ExtSendUtil

Program sending process:

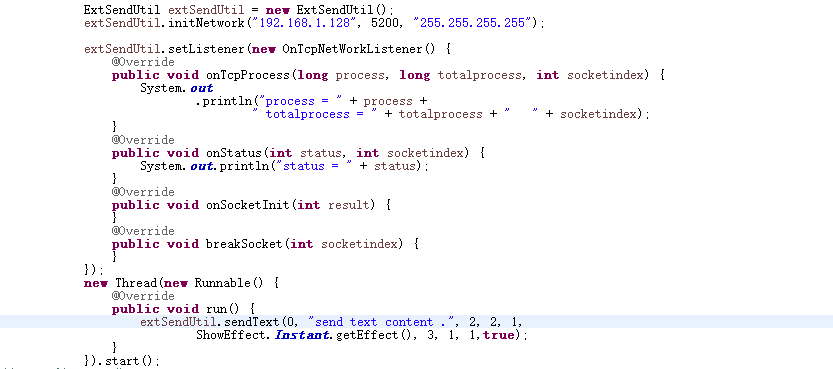
1. Invoke initNetwork to initialize network communication parameters
2. Call setListener to set listener and listen to communication status

**(Please call the following interface in the worker thread)**

1. Call splitScreen to split windows
2. Call sendXXX to send programs

com.lumen.ledcenter3.protocol.ExternalNetworkSendProtocol.OnTcpNetWorkListener

**Call example**



## 3.1,overview

ExtSendUtil.

|  |  |  |
| --- | --- | --- |
|  | **Name** | **Description** |
| 1 | initNetwork | Initialize network parameters |
| 2 | setListener | Set up network communication listener |
| 3 | splitScreen | Split window |
| 4 | sendText | Send text |
| 5 | sendPicture | Send pictures (including jpg.png.gif) |
| 6 | sendPictureWithScale | Send pictures, you can set zoom (gif is not supported) |
|  | sendVideo | Send video |
| 7 | saveSentDataToFlash | Save the transmitted program |
| 8 | clearSavedDataFromFlash | Don't save the transmitted program (it will be cleared after restarting) |
| 9 | getBrightness | Get brightness |
| 10 | setBrightness | Set brightness |
| 11 | setScreenOnAndOffTime | Set the switch screen time |
| 12 | getScreenOnAndOffTime | Get the switch screen time |
| 13 | exitSetScreenOnAndOffTime | There is no automatic switch screen |
| 14 | setScreenSwitch | Open and close the screen immediately |
| 15 | quitExternalScreen | Exit the multi-window program |
| 16 | selectPlayProgram | Selective broadcasting program |
| 17 | exitSelectPlay | Quit the selective broadcast program |
| 18 | setVariableValue | Set variable value |
| 19 | ReadyToUseProgramTemplate | Ready to enter the template mode to display the program |
| 20 | SendTemplateProgram | Send template program |
| 21 | deleteTemplateProgram | Deletes a program with a specified serial number |
| 22 | enterOrExitTemplateProgramMode | Enter or exit a program template |
| 23 | setReceiveTimeout | Set the timeout period for receiving returned packet data |
| 24 | getReceiveTimeout | Gets the timeout period for receiving returned packet data |
| 27 | createGlobalAreaWindow | Create a global area window |
| 28 | clearAllGlobalArea | Clear all global zones |
| 29 | setIO | Set the high and low levels of the five I/O ports |
| 30 | setTextComplex | Set the text in the text program to traditional Chinese |
| 31 | getTextComplex | Gets whether the text in the text program is in traditional Chinese |

OnTcpNetWorkListener

|  |  |  |
| --- | --- | --- |
| **No** | **Name** | **Description** |
| 1 | onSocketInit | Network initialization |
| 2 | onStatus | Communication status |
| 3 | onBackBytes | Data returned |
| 4 | onTcpProcess | Upload progress (when sending pictures or files) |
| 5 | breakSocket | Network disconnection |

## 3.2、Detail description

### initNetwork (Updated 1.4.2)

|  |  |
| --- | --- |
| initNetwork (String ip, int port, String idcode) | |
| Use | Initialize network parameters |
| Params | Ip : control card’s network ip address  Port: net port , default 5200  Idcode: default 255.255.255.255 |
| initNetwork (String ip, int port, String mac, int protocolMode) | |
| Use | Initializing network parameters (WAN/cross-route calls) |
| Params | Ip : control card’s network ip address  Port: net port , default 5200  mac: device mac address  protocolMode:0:auto, 1:old, 2:new |

### setListener

|  |  |
| --- | --- |
| setListener (OnTcpNetWorkListener listener) | |
| Use | Set up network communication listener |
| Params | Listener: Monitor the network communicate status and progress |

### splitScreen

|  |  |
| --- | --- |
| boolean splitScreen(int windowCount, int[]... winRects) | |
| Use | Divided into windows, with a maximum of 8. To save the program to flash memory, you must call. |
| Params | windowCount: The number of windows to be split, and within the limits of 1 to 8.  winRects：array of windows’s rects.  winRect is The coordinates of a window. Each window uses four integers to represent its coordinate values of "left, top, right and bottom" |
| Return | True: successfully，false otherwise. |
| example |  |
| notice | Time-consuming operation, please call in the worker thread, and get the call result in the set listener |

### sendText(Updated 1.4.3)

|  |  |
| --- | --- |
| boolean sendText( int nWndNo,  String content, int crColor,  int nFontSize, int nSpeed, int nEffect,  int nStayTime, int nAlignmentHori, int nAlignmentVert) | |
| Use | Send formatted text to specified window |
| Params | nWndNo: Sequence number of window, virtual value 0-31.  Content: text string.  Background color of the window:<bkcolor=1>, the value is the same as crColor  Text size: <size=1>, the value is the same as nFontSize  Text color: <color=2>, the value is the same as crColor  Text font: <font=1>, the value is 0~7, corresponding to the character library file)  crColor: value 1-7 for color: red, green, yellow, blue, purple, blue and white.  nFontSize: value 0-7 for size: 8, 12, 16, 24,32, 40, 48, 56  nSpeed: value 1 ~ 100. The smaller the value, the faster the speed.  nEffect: Display effect coding  nStayTime: The unit is seconds.  nAlignmentHori: Horizontal alignment (0: left alignment, 1: horizontal middle, 2: right alignment)  nAlignmentVert: Vertical alignment (0: top alignment, 1: vertical middle, 2: bottom alignment) |
| Return | True: successfully，false otherwise. |
| example |  |
| notice | Time-consuming operation, please call in the worker thread, and get the call result OnTcpNetWorkListener.onStatus in the set listener |

### sendText(Expanded library, new V1.4.0)

|  |  |
| --- | --- |
| boolean sendText( int nWndNo,  String content, int crColor,  int nFontSize, int nSpeed, int nEffect,  int nStayTime, int nAlignmentHori, int nAlignmentVert, boolean isExt) | |
| Use | Sends formatted text (ASCII) to the specified window |
| Params | nWndNo: Sequence number of window, virtual value 0-7.  Content: text string.  crColor: value 1-7 for color: red, green, yellow, blue, purple, skyblue and white.  nFontSize: value 0-7 for size: 8, 12, 16, 24,32, 40, 48, 56  nSpeed: value 1 ~ 100. The smaller the value, the faster the speed.  nEffect: Display effect coding  nStayTime: The unit is seconds.  nAlignmentHori: Horizontal alignment (0: left alignment, 1: horizontal middle, 2: right alignment)  nAlignmentVert: Vertical alignment (0: top alignment, 1: vertical middle, 2: bottom alignment)  isExt: ASCII(default true)  The corresponding lmf font file must be on the card |
| Return | True: successfully，false otherwise. |
| example |  |
| notice | Perform time-consuming operations in the worker thread and obtain the call result OnTcpNetWorkListener.onStatus in the set listener |

### sendTextRGB (Expanded font color, added V1.4.6)

|  |  |
| --- | --- |
| boolean sendTextRGB( int nWndNo,  String content, byte[] crColors, int nFontSize,  int nSpeed, int nEffect, int nStayTime,  int nAlignmentHori, int nAlignmentVert) | |
| Use | Sends plain text to the specified window |
| Params | nWndNo: Sequence number of window, virtual value 0-7.  Content: text string.  crColor: value RGB, crColors[0]= color R, crColors[1]= color G, crColors[2]= color B.  nFontSize: value 0-7 for size: 8, 12, 16, 24,32, 40, 48, 56  nSpeed: value 1 ~ 100. The smaller the value, the faster the speed.  nEffect: Display effect coding  nStayTime: The unit is seconds.  nAlignmentHori: Horizontal alignment (0: left alignment, 1: horizontal middle, 2: right alignment)  nAlignmentVert: Vertical alignment (0: top alignment, 1: vertical middle, 2: bottom alignment) |
| Return | True: successfully，false otherwise. |
| example |  |
| notice | Perform time-consuming operations in the worker thread and obtain the call result OnTcpNetWorkListener.onStatus in the set listener |

### sendPicture

|  |  |
| --- | --- |
| boolean sendPicture(int nWndNo,  int nSpeed, int nEffect,  int nStayTime, String picturePath) | |
| Use | Send pictures to the specified window, which can be gif, jpg, or png pictures |
| Params | nWndNo: Sequence number of window, virtual value 0-31.  nSpeed: value 1 ~ 100. The smaller the value, the faster the speed.  nEffect: display effect code  nStayTime: The unit is seconds.  picturePath: absolute path of picture file. Include gif, jpg, png. |
| Return | True: successfully，false otherwise. |
| example |  |
| notice | Time-consuming operation, please call in the worker thread, and get the call result OnTcpNetWorkListener.onStatus in the set listener |

### sendPictureWithScale

|  |  |
| --- | --- |
| boolean sendPicture(int nWndNo, int winWidth, int winHeight,  int nSpeed, int nEffect, int scaleType,  int nStayTime, String picturePath) | |
| Use | Send pictures to the specified window, which can be jpg or png pictures. And the scaleType can be selected |
| Params | nWndNo: Sequence number of window, virtual value 0-31.  winWidth: width of window  winHeight: height of window  nSpeed: value 1 ~ 100. The smaller the value, the faster the speed.  Ineffect: display effect code  ScaleType: image scaling method coding  nStayTime: The unit is seconds.  picturePath: absolute path of picture file. Include gif, jpg, png. |
| Return | True: successfully，false otherwise. |
| example |  |
| notice | Time-consuming operation, please call in the worker thread, and get the call result OnTcpNetWorkListener.onStatus in the set listener |

### sendVideo

|  |  |
| --- | --- |
| boolean sendVideo (int nWndNo ,String videoPath) | |
| Use | Send video to specified window. |
| Params | nWndNo: Sequence number of window, virtual value 0-31. picturePath: absolute path of video file. |
| Return | True: successfully，false otherwise. |
| example |  |
| notice | Time-consuming operation, please call in the worker thread, and get the call result OnTcpNetWorkListener.onStatus in the set listener |

### saveSentDataToFlash

|  |  |
| --- | --- |
| boolean saveSentDataToFlash() | |
| Use | Save split screen information and programs which sent by sendText and sendPicture. If not call this, the sent data will not be saved when control card restart. |
| Params |  |
| Return | True: successfully，false otherwise. |
| notice | Time-consuming operation, please call in the worker thread, and get the call result OnTcpNetWorkListener.onStatus in the set listener |

### clearSavedDataFromFlash

|  |  |
| --- | --- |
| boolean clearSavedDataFromFlash() | |
| Use | Clear Save split screen information and programs which sent by sendText and sendPicture. And the sent data will not be saved when control card restart. |
| Params |  |
| Return | True: successfully，false otherwise. |
| notice | Time-consuming operation, please call in the worker thread, and get the call result OnTcpNetWorkListener.onStatus in the set listener |

### getBrightness

|  |  |
| --- | --- |
| int[] getBrightness() | |
| Use | Get the current brightness value, with 24 values ranging from 00: 00 to 24: 00. The value range is 0-31, and -1 indicates photosensitive control. |
| Params |  |
| Return | The brightness value, length is 24, represent 24 time slot from 00:00. |
| notice | This method is time-consuming, you can call it in work thread to avoid blocking of main thread, and monitor the result int OnTcpNetWorkListener.onStatus |

### setBrightness

|  |  |
| --- | --- |
| boolean setBrightness(int[] brigghtnessValue) | |
| Use | Set brightness, with 24 values from 00: 00 to 24: 00. The value range is 0-31, and -1 indicates photosensitive control. |
| Params | brigghtnessValue: The brightness value, length is 24, represent 24 time slot from 00:00. When all 24 value is -1, then is in photosensitivity mode. |
| Return | True: successfully，false otherwise. |
| notice | Time-consuming operation, please call in the worker thread, and get the call result OnTcpNetWorkListener.onStatus in the set listener |

### setScreenOnAndOffTime

|  |  |
| --- | --- |
| boolean setScreenOnAndOffTime(int nHourOn, int nMinuteOn, int nHourOff, int nMinuteOff) | |
| Use | Set the timing switch screen |
| Params | nHourOn: 0~23  nMinuteOn: 0~59  nHourOff: 0~23  nMinuteOff: 0~59 |
| Return | True: successfully，false otherwise. |
| notice | Time-consuming operation, please call in the worker thread, and get the call result OnTcpNetWorkListener.onStatus in the set listener |

### getScreenOnAndOffTime

|  |  |
| --- | --- |
| int[] getScreenOnAndOffTime () | |
| Use |  |
| Params |  |
| return | The screen on and off time, length is 4. When value is 255, explain that the switch screen settings are not turned on. |
| notice | Time-consuming operation, please call in the worker thread, and get the call result OnTcpNetWorkListener.onStatus in the set listener |

### exitSetScreenOnAndOffTime

|  |  |
| --- | --- |
| boolean exitSetScreenOnAndOffTime () | |
| Use | Exit screen on and off settings. |
| Params |  |
| Return | True: successfully，false otherwise. |
| notice | Time-consuming operation, please call in the worker thread, and get the call result OnTcpNetWorkListener.onStatus in the set listener |

### setScreenSwitch

|  |  |
| --- | --- |
| boolean setScreenSwitch(boolean turnOn) | |
| Use | Open and close the screen immediately (the restart card fails) |
| Params | TurnOn: true means on, false means off |
| Return | True: successfully，false otherwise. |
| notice | Time-consuming operation, please call in the worker thread, and get the call result OnTcpNetWorkListener.onStatus in the set listener |

### quitExternalScreen

|  |  |
| --- | --- |
| boolean quitExternalScreen() | |
| Use | Exit the multi-window program mode and return to the regular program playing |
| Params |  |
| Return | True: successfully，false otherwise. |
| notice | Time-consuming operation, please call in the worker thread, and get the call result OnTcpNetWorkListener.onStatus in the set listener |

### Selectplayprogram (added in v1.3.2)

|  |  |
| --- | --- |
| Boolean selectPlayProgram (boolean save, byte[] programIndex) | |
| function | Selective broadcasting program |
| parameter | Save: save or not. After saving, the power-off restart is still valid  ProgramIndex: an array of program numbers to be broadcast, starting from 1 |
| Return value | True: successfully，false otherwise. |
| Other instructions | Time-consuming operation, please call in the worker thread, and get the call result OnTcpNetWorkListener.onStatus in the set listener |

### Exitselectplay (added in v1.3.2)

|  |  |
| --- | --- |
| boolean exitSelectPlay () | |
| function | Exit the selective broadcasting mode and return to normal broadcasting |
| parameter | without |
| Return value | True: successfully，false otherwise. |
| Other instructions | Time-consuming operation, please call in the worker thread, and get the call result OnTcpNetWorkListener.onStatus in the set listener |

### Setvariablevalue (added in v1.3.2)

|  |  |
| --- | --- |
| boolean setVariableValue (boolean saveToFlash, int variableNum, byte[]... datas) | |
| Use | Set variable value |
| Params | saveToFlash: Whether to save  variableNum: Number of variables  datas :variable value, two-dimensional byte array  For example, set two variable values: (V1.4.5 added this description)[  [0x01,0xaa,0xbb,0xcc],  [0x02,0xaa,0xbb,0xcc]  ]  0x01: Indicates the serial number of the variable, the first variable  0x02: Indicates the serial number of the variable, the second variable  0xaa,0xbb,0xcc: Byte values representing variable content, text content |
| Return | True: successfully，false otherwise. |
| notice | Time-consuming operation, please call in the worker thread, and get the call result OnTcpNetWorkListener.onStatus in the set listener |

### readyToUseProgramTemplate（V1.3.7 added）

|  |  |
| --- | --- |
| Void readyToUseProgramTemplate(int screenWidth, int screenHeight) | |
| Use | Ready to enter Program Template Mode |
| Params | screenWidth  screenHeight |
| Return |  |
| notice | Time-consuming operation, please call in the worker thread, and get the call result OnTcpNetWorkListener.onStatus in the set listener |

### sendTemplateProgram（V1.3.7 added）

|  |  |
| --- | --- |
| Void sendTemplateProgram(int programNo, List<WindowAttr> windows, boolean showImmediately) | |
| Use | Send template programs, support Text, Image, Temperature |
| Params | programNo: program number  windows : Window attributes and program parameters  WindowAttr：  Window：startX, startY, width, height  Program：  TextItem Text  ImageItem Image  TempHumiItem temperature or humidity  showImmediately: Whether to show the show immediately |
| Return |  |
| notice | Time-consuming operation, please call in the worker thread, and get the call result OnTcpNetWorkListener.onStatus in the set listener. And  OnTcpNetWorkListener.onTcpProcess to Monitor sending progress |

### deleteTemplateProgram（V1.3.7 added）

|  |  |
| --- | --- |
| Void deleteTemplateProgram(int... programNos) | |
| Use | Delete the program of the specified number |
| Params | programNos: Program number, support multiple |
| Return |  |
| notice | Time-consuming operation, please call in the worker thread, and get the call result OnTcpNetWorkListener.onStatus in the set listener |

### enterOrExitTemplateProgramMode（V1.3.7 added）

|  |  |
| --- | --- |
| Void enterOrExitTemplateProgramMode(boolean enter) | |
| Use | Enter or exit Program Template Mode |
| Params | Enter: true - enter，false - exit |
| Return |  |
| notice | Time-consuming operation, please call in the worker thread, and get the call result OnTcpNetWorkListener.onStatus in the set listener |

### setReceiveTimeout（Added in V1.4.0）

|  |  |
| --- | --- |
| setReceiveTimeout(int timeout); | |
| Use | Set the timeout period for receiving returned packet data |
| Params | timeout: Indicates the timeout period, expressed in milliseconds. The default value is 5000 milliseconds. |

### getReceiveTimeout（Added in V1.4.0）

|  |  |
| --- | --- |
| getReceiveTimeout (); | |
| Use | Gets the timeout period for receiving returned packet data |
| Params |  |

### createGlobalAreaWindow（Added in V1.4.4）

|  |  |
| --- | --- |
| boolean createGlobalAreaWindow(boolean saveToFlash, List<AreaWindowPojo> dataList) | |
| Use | Create a global area window |
| Params | saveToFlash: indicates whether to save the data to flash. 0: no, 1: yes.  dataList: AreaWindowPojo, the List length must be 1 or 2.  View AreaWindowPojo objects:  **int** windowType = 1; //Window type, default 1: is a text type  **int** showType ; //Display mode, Bit 0 to 1: Align left and right  0 justified left, 1 centered left and right, 2 justified right  Bit 2 to 3: Alignment up and down  0 is aligned up, 1 is centered up and down, and 2 is aligned down  **int** startX; // Window start X wheelbase away  **int** startY; // Window start Y wheelbase away  **int** width; // Window width  **int** height; // Window height  **int** textColor; // word color， value 1-7 for color: red, green, yellow, blue, purple, skyblue and white.  **int** textSize; // Text size，value 0-7 for size: 8, 12, 16, 24,32, 40, 48, 56 |
| Return | true: succeeds, false: fails |
| notice | Perform time-consuming operations in the worker thread |

### clearAllGlobalArea（Added in V1.4.4）

|  |  |
| --- | --- |
| boolean clearAllGlobalArea() | |
| Use | Clear all global zones |
| Params | without |
| Return | true: succeeds, false: fails |
| notice | Perform time-consuming operations in the worker thread |

### setIO（Added in V1.4.4）

|  |  |
| --- | --- |
| boolean setIO(**byte** io1,**byte** io2,**byte** io3,**byte** io4,**byte** io5) | |
| Use | Set the high and low levels of the five I/O ports |
| Params | io1: indicates the I/O port of route 1  io2: indicates the I/O port of route 2  io3: indicates the third I/O port  io4: indicates the I/O port of route 4  io5: indicates the I/O port of route 5  0x81: High, 0x80: low, 0x00 or 0x01: invalid |
| Return | true: succeeds, false: fails |
| notice | Perform time-consuming operations in the worker thread |

### setTextComplex（Added in V1.4.5）

|  |  |  |
| --- | --- | --- |
| void setTextComplex (int textComplex) | | |
| 作用 | Sets whether the text in the program is in traditional Chinese | |
| 参数 | textComplex: indicates whether the text is traditional, 0: no, 1: traditional | |
| 返回值 |  |
| 其它说明 |  | |

### getTextComplex（Added in V1.4.5）

|  |  |  |
| --- | --- | --- |
| int getTextComplex () | | |
| 作用 | Check whether the text in the text program is in traditional Chinese | |
| 参数 |  | |
| 返回值 | textComplex: indicates whether the text is traditional, 0: no, 1: traditional |
| 其它说明 |  | |

**OnTcpNetWorkListener**

### onSocketInit

|  |  |
| --- | --- |
| onSocketInit(int result); | |
| Use | Monitor socket connect status |
| Params | Result: 0 fail,1 success |

### onStatus

|  |  |
| --- | --- |
| onStatus(int status, int socketindex); | |
| Use | Monitor data send status |
| Params | Status: -1 means error occur , 1 means ok.  Socketindex: socket Index |

### onBackBytes

|  |  |
| --- | --- |
| onBackBytes(int[] backbytes, int socketindex); | |
| Use |  |
| Params | Backbytes: back data from control card  Socketindex：socket Index |

### onTcpProcess

|  |  |
| --- | --- |
| onTcpProcess(int process, int totalprocess, int socketindex); | |
| Use | Monitor data send progress when send picture or other files |
| Params | Process：  Totalprocess：  Socketindex：socket Index |

### breakSocket

|  |  |
| --- | --- |
| breakSocket(int socketindex); | |
| Use | Monitor socket closed. |
| Params | Socketindex: socket Index |

# Multi-window Protocol (Serial port) (added in V1.3.4)

## com.lumen.ledcenter3.protocol.ExtSendUtil\_COM

Program delivery process:

1. Call initCOM to initialize network communication parameters

2. Call setListener to set the listener and monitor the communication status

(Please make the following interface call in the worker thread)

3. Call splitScreen to split Windows

4. Call sendXXX to send the program

com.lumen.ledcenter3.protocol.ExtSendUtil\_COM.OnCOMListener

## Call example



**4.1. Overview**

## 3.1 Overview of Key Management Interface

CypherManager

|  |  |  |  |
| --- | --- | --- | --- |
| **serial number** | **API interface function name** | **Interface description** | **completeness** |
| one | setSecretKey | Set the control card communication key | Achieved |

## 3.2,Details of Key Management Interface

### setSecretKey

|  |  |
| --- | --- |
| void setSecretKey (String secretKey) | |
| function | Set the control card communication key |
| parameter | SecretKey: key |
| Return value | without |
| Other instructions | 1、Please call this method before communicating with the control card  2、Password length must be 8, which can be a mixture of numbers and letters |

# 4,Control (added in V1.3.0)

com.lumen.ledcenter3.protocol.ControlUtil

com.lumen.ledcenter3.protocol. ControlUtil.OnControlListener

## 4.1,Overview

ExtSendUtil\_COM

|  |  |  |
| --- | --- | --- |
|  | **Name** | **Description** |
| 1 | initCOM | Example Initialize serial port connection parameters |
| 2 | setListener | Set up the network communication monitor |
| 3 | splitScreen | split window |
| 4 | sendText | Send text |
| 5 | sendPicture | Send pictures (including jpg.png.gif) |
| 6 | sendPictureWithScale | Send pictures, zoom can be set (gif not supported) |
| 7 | saveSentDataToFlash | Save the sent program |
| 8 | clearSavedDataFromFlash | Do not save sent programs (will be cleared after restart) |
| 9 | getBrightness | Get brightness |
| 10 | setBrightness | set brightness |
| 11 | setScreenOnAndOffTime | Set the switch screen time |
| 12 | getScreenOnAndOffTime | Gets the switch screen time |
| 13 | exitSetScreenOnAndOffTime | No automatic switch screen |
| 14 | setScreenSwitch | Immediate switch panel |
| 15 | quitExternalScreen | Exit the multi-window program |
| 16 | selectPlayProgram | Selected program |
| 17 | exitSelectPlay | Exit the program |
| 18 | setVariableValue | Set variable value |
| 20 | createGlobalAreaWindow | Create a global area window |
| 21 | clearAllGlobalArea | Know all global areas |
| 22 | setTextComplex | Set the text in the text program to traditional Chinese |
| 23 | getTextComplex | Gets whether the text in the text program is in traditional Chinese |

OnCOMListener

|  |  |  |
| --- | --- | --- |
| **No** | **Name** | **Description** |
| 1 | onStatus | Communication status |
| 2 | onBackBytes | Returned data |
| 3 | onProcess | Upload progress (when sending pictures or files) |

## 4.2、Detail description

### initCOM

|  |  |
| --- | --- |
| initCOM (String portName, int baudrate) | |
| Use | Example Initialize network parameters |
| Params | portName  baudrate |

### setListener

|  |  |
| --- | --- |
| setListener (OnCOMListener listener) | |
| Use | Set up a communication monitor |
| Params | Listener: Serial communication monitor |

### splitScreen

|  |  |
| --- | --- |
| boolean splitScreen(int windowCount, int[]... winRects) | |
| Use | Divided into Windows, maximum 8. If you want to save the program to the flash memory (savesentdata atoflash), you must call. |
| Params | windowCount: The number of windows to be split, and within the limits of 1 to 8.  winRects：array of windows’s rects.  winRect is The coordinates of a window. Each window uses four integers to represent its coordinate values of "left, top, right and bottom" |
| Return | True: successfully，false otherwise. |
| example |  |
| notice | Perform time-consuming operations in the worker thread and obtain the call result in the set listener |

### sendText

|  |  |
| --- | --- |
| boolean sendText( int nWndNo,  String content, int crColor,  int nFontSize, int nSpeed, int nEffect,  int nStayTime, int nAlignmentHori, int nAlignmentVert) | |
| Use | Sends formatted text to the specified window |
| Params | nWndNo: Sequence number of window, virtual value 0-7.  Content: text string.  crColor: value 1-7 for color: red, green, yellow, blue, purple, skyblue and white.  nFontSize: value 0-7 for size: 8, 12, 16, 24,32, 40, 48, 56  nSpeed: value 1 ~ 100. The smaller the value, the faster the speed.  nEffect: Display effect coding  nStayTime: The unit is seconds.  nAlignmentHori: Horizontal alignment (0: left alignment, 1: horizontal middle, 2: right alignment)  nAlignmentVert: Vertical alignment (0: top alignment, 1: vertical middle, 2: bottom alignment) |
| Return | True: successfully，false otherwise. |
| example |  |
| notice | Perform time-consuming operations in the worker thread and obtain the call result in the set listener |

### sendPicture

|  |  |
| --- | --- |
| boolean sendPicture(int nWndNo,  int nSpeed, int nEffect,  int nStayTime, String picturePath) | |
| Use | Send an image to the specified window, which can be a gif, jpg, or png image |
| Params | nWndNo: Sequence number of window, virtual value 0-7.  nSpeed: value 1 ~ 100. The smaller the value, the faster the speed.  nEffect:Display effect coding  nStayTime: The unit is seconds.  picturePath: absolute path of picture file. Include gif, jpg, png. |
| Return | True: successfully，false otherwise. |
| example |  |
| notice | Perform time-consuming operations in the worker thread and obtain the call result in the set listener |

### sendPictureWithScale

|  |  |
| --- | --- |
| boolean sendPicture(int nWndNo, int winWidth, int winHeight,  int nSpeed, int nEffect, int scaleType,  int nStayTime, String picturePath) | |
| Use | Send the image to the specified window. It can be a jpg or png image. And you can choose the scaleType. |
| Params | nWndNo: Sequence number of window, virtual value 0-7.  winWidth: width of window  winHeight: height of window  nSpeed: value 1 ~ 100. The smaller the value, the faster the speed.  nEffect: Display effect coding  scaleType: Image scaling mode encoding  nStayTime: The unit is seconds.  picturePath: absolute path of picture file. Include ~~gif,~~ jpg, png. |
| Return | True: successfully，false otherwise. |
| example |  |
| notice | Perform time-consuming operations in the worker thread and obtain the call result in the set listener |

### saveSentDataToFlash

|  |  |
| --- | --- |
| boolean saveSentDataToFlash() | |
| Use | Save split screen information and programs which sent by sendText and sendPicture. If not call this, the sent data will not be saved when control card restart. |
| Params |  |
| Return | True: successfully，false otherwise. |
| notice | Perform time-consuming operations in the worker thread and obtain the call result in the set listener |

### clearSavedDataFromFlash

|  |  |
| --- | --- |
| boolean clearSavedDataFromFlash() | |
| Use | Clear Save split screen information and programs which sent by sendText and sendPicture. And the sent data will not be saved when control card restart. |
| Params |  |
| Return | True: successfully，false otherwise. |
| notice | Perform time-consuming operations in the worker thread and obtain the call result in the set listener |

### getBrightness

|  |  |
| --- | --- |
| int[] getBrightness() | |
| Use | Gets the current brightness value, 24 values from 00:00 to 24:00. Value range 0-31, -1 indicates light sensitive control. |
| Params |  |
| Return | The brightness value, length is 24, represent 24 time slot from 00:00. |
| notice | This method is time-consuming, you can call it in work thread to avoid blocking of main thread, and monitor the result int |

### setBrightness

|  |  |
| --- | --- |
| boolean setBrightness(int[] brigghtnessValue) | |
| Use | Set brightness, 24 values from 00:00 to 24:00. Value range 0-31, -1 indicates light sensitive control. |
| Params | brigghtnessValue: The brightness value, length is 24, represent 24 time slot from 00:00. When all 24 value is -1, then is in photosensitivity mode. |
| Return | True: successfully，false otherwise. |
| notice | Perform time-consuming operations in the worker thread and obtain the call result in the set listener |

### setScreenOnAndOffTime

|  |  |
| --- | --- |
| boolean setScreenOnAndOffTime(int nHourOn, int nMinuteOn, int nHourOff, int nMinuteOff) | |
| Use | Set the timing switch panel |
| Params | nHourOn: 0~23  nMinuteOn: 0~59  nHourOff: 0~23  nMinuteOff: 0~59 |
| Return | True: successfully，false otherwise. |
| notice | Perform time-consuming operations in the worker thread and obtain the call result in the set listener |

### getScreenOnAndOffTime

|  |  |
| --- | --- |
| int[] getScreenOnAndOffTime () | |
| Use |  |
| Params |  |
| return | The screen on and off time, length is 4. When value is 255, explain that the switch screen settings are not turned on. |
| notice | Perform time-consuming operations in the worker thread and obtain the call result in the set listener |

### exitSetScreenOnAndOffTime

|  |  |
| --- | --- |
| boolean exitSetScreenOnAndOffTime () | |
| Use | Exit screen on and off settings. |
| Params |  |
| Return | True: successfully，false otherwise. |
| notice | Perform time-consuming operations in the worker thread and obtain the call result in the set listener |

### setScreenSwitch

|  |  |
| --- | --- |
| boolean setScreenSwitch(boolean turnOn) | |
| Use | Switch the screen immediately (restart card fails) |
| Params | TurnOn: true indicates on, false indicates off |
| Return | True: successfully，false otherwise. |
| notice | Perform time-consuming operations in the worker thread and obtain the call result in the set listener |

### quitExternalScreen

|  |  |
| --- | --- |
| boolean quitExternalScreen() | |
| Use | Exit multi-window program mode and return to regular program playback |
| Params |  |
| Return | True: successfully，false otherwise. |
| notice | Perform time-consuming operations in the worker thread and obtain the call result in the set listener |

### selectPlayProgram

|  |  |
| --- | --- |
| boolean selectPlayProgram (boolean save, byte[] programIndex) | |
| Use | Selected program |
| Params | save: Whether to save. After saving, power off restart still works  programIndex: An array of program numbers to broadcast, starting with 1 |
| Return | True: successfully，false otherwise. |
| notice | Perform time-consuming operations in the worker thread and obtain the call result in the set listener |

### exitSelectPlay

|  |  |
| --- | --- |
| boolean exitSelectPlay () | |
| function | Exit the program mode and return to normal playback |
| Params | 无 |
| Return | True: successfully，false otherwise. |
| notice | Perform time-consuming operations in the worker thread and obtain the call result in the set listener |

### setVariableValue

|  |  |
| --- | --- |
| boolean setVariableValue (boolean saveToFlash, int variableNum, byte[]... datas) | |
| Use | Set variable value |
| Params | saveToFlash: Whether to save  variableNum: Number of variables  datas :variable value, two-dimensional byte array  For example, set two variable values: (V1.4.5 added this description)[  [0x01,0xaa,0xbb,0xcc],  [0x02,0xaa,0xbb,0xcc]  ]  0x01: Indicates the serial number of the variable, the first variable  0x02: Indicates the serial number of the variable, the second variable  0xaa,0xbb,0xcc: Byte values representing variable content, text content |
| Return | True: successfully，false otherwise. |
| notice | Perform time-consuming operations in the worker thread and obtain the call result in the set listener |

### createGlobalAreaWindow（Added in V1.4.4）

|  |  |
| --- | --- |
| boolean createGlobalAreaWindow(boolean saveToFlash, List<AreaWindowPojo> dataList) | |
| Use | Create a global area window |
| Params | saveToFlash: indicates whether to save the data to flash. 0: no, 1: yes.  dataList: AreaWindowPojo, the List length must be 1 or 2.  View AreaWindowPojo objects:  **int** windowType = 1; //Window type, default 1: is a text type  **int** showType ; //Display mode, Bit 0 to 1: Align left and right  0 justified left, 1 centered left and right, 2 justified right  Bit 2 to 3: Alignment up and down  0 is aligned up, 1 is centered up and down, and 2 is aligned down  **int** startX; // Window start X wheelbase departure  **int** startY; // Window start Y wheelbase departure  **int** width; // window width  **int** height; // window height  **int** textColor; // text color， value 1-7 for color: red, green, yellow, blue, purple, skyblue and white.  **int** textSize; // Text size，value 0-7 for size: 8, 12, 16, 24,32, 40, 48, 56 |
| Return | true: succeeds, false: fails |
| notice | Perform time-consuming operations in the worker thread |

### clearAllGlobalArea（Added in V1.4.4）

|  |  |
| --- | --- |
| boolean clearAllGlobalArea() | |
| Use | Clear all global zones |
| Params | without |
| Return | true: succeeds, false: fails |
| notice | Perform time-consuming operations in the worker thread |

### setTextComplex（Added in V1.4.5）

|  |  |  |
| --- | --- | --- |
| void setTextComplex (int textComplex) | | |
| function | Sets whether the text in the program is in traditional Chinese | |
| params | textComplex: indicates whether the text is traditional, 0: no, 1: traditional | |
| Return |  |
| notice |  | |

### getTextComplex（Added in V1.4.5）

|  |  |  |
| --- | --- | --- |
| int getTextComplex () | | |
| function | Check whether the text in the text program is in traditional Chinese | |
| params |  | |
| return | textComplex: indicates whether the text is traditional, 0: no, 1: traditional |
| notice |  | |

### **getSendTextData（Added in V1.4.7）**

|  |  |
| --- | --- |
| boolean getSendTextData(int nWndNo,  String content, int crColor,  int nFontSize, int nSpeed, int nEffect,  int nStayTime, int nAlignmentHori, int nAlignmentVert) | |
| Use | Get send data (send formatted text to specified window) |
| Params | nWndNo: Sequence number of window, virtual value 0-7.  Content: text string.  crColor: value 1-7 for color: red, green, yellow, blue, purple, skyblue and white.  nFontSize: value 0-7 for size: 8, 12, 16, 24,32, 40, 48, 56  nSpeed: value 1 ~ 100. The smaller the value, the faster the speed.  nEffect: Display effect coding  nStayTime: The unit is seconds.  nAlignmentHori: Horizontal alignment (0: left alignment, 1: horizontal middle, 2: right alignment)  nAlignmentVert: Vertical alignment (0: top alignment, 1: vertical middle, 2: bottom alignment) |
| Return | True: successfully，false otherwise. |
| example | 9dbb881e375053449530c1690a594e9 |
| notice | The initCOM interface method is not required |

**OnCOMListener**

### onStatus

|  |  |
| --- | --- |
| onStatus(int status, int portIndex); | |
| Use | Monitor data send status |
| Params | Status: -1 means error occur , 1 means ok.  portIndex: Port Number identification (Custom) |

### onBackBytes

|  |  |
| --- | --- |
| onBackBytes(int[] backbytes, int portIndex); | |
| Use |  |
| Params | Backbytes: back data from control card  portIndex: Port Number identification (Custom) |

### onProcess

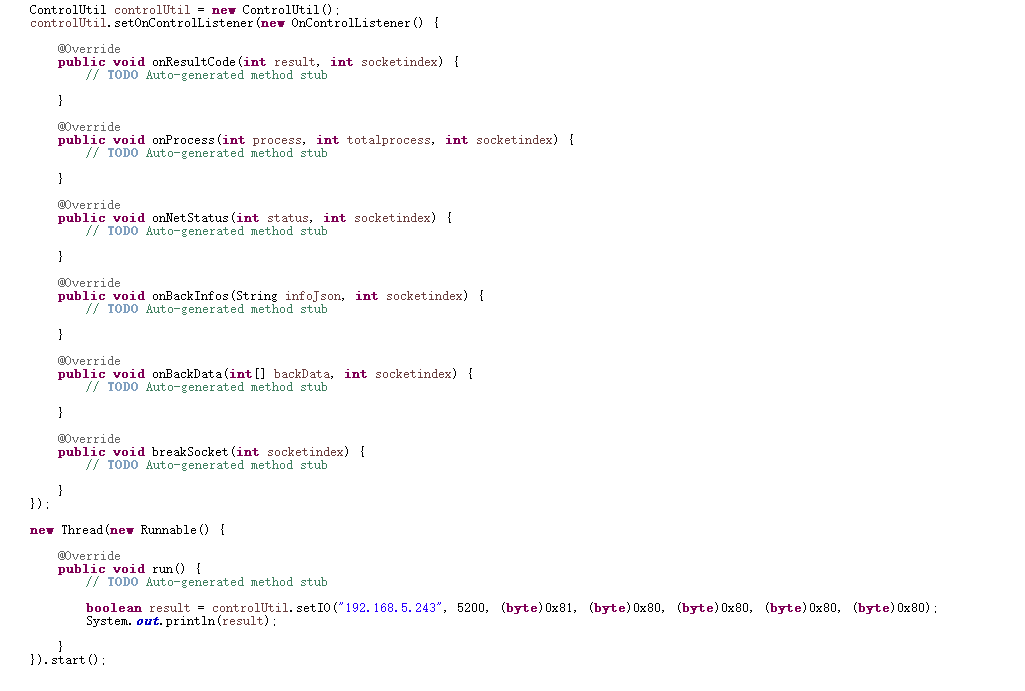
|  |  |
| --- | --- |
| onProcess(int process, int totalprocess, int portIndex); | |
| Use | Monitor data send progress when send picture or other files |
| Params | Process：  Totalprocess：  portIndex: Port Number identification (Custom) |

# 5、Control (network)（Added in V1.3.0）

com.lumen.ledcenter3.protocol.ControlUtil

com.lumen.ledcenter3.protocol. ControlUtil.OnControlListener

## Call example



## 5.1、overview

ControlUtil

|  |  |  |
| --- | --- | --- |
| **serial number** | **API interface function name** | **Interface description** |
| 1 | setOnControlListener | Set listener |
| 2 | getCurrentPlayProgramNo | Acquiring the program number and the total number of programs currently played |
| 3 | getCurrentProgramScreenshot | Get a screenshot of the currently playing program |
| 4 | getCurrentPlayProgramContent | Gets the text content of window 1 of the program number |
| 5 | getMultiWindowsData | Gets the latest multi-window protocol sendtext content |
| 6 | selectPlayProgram | Selected program |
| 7 | exitSelectPlay | Exit the program |
| 8 | getProgramFileInfos | Gets a list of sent program files |
| 9 | getUserDiscRemainingSpace | Gets the size of the remaining space in the program area |
| 10 | deleteLpbFile | Deletes the specified program file |
| 11 | formatUserDisk | Formatting user disk |
| 12 | sendInstantMsg | Send instant message |
| 13 | getCurrentCardTimeDate | Obtain the current time of the controller card |
| 14 | setCardTimeDateToNow | Set the controller card time to the current time |
| 15 | setCardTimeDate | Set the control card time manually |
| 16 | restartApp | Restart the App |
| 17 | restartHard | Restart hardware |
| 18 | getGlobalVariable | Gets the value of a global variable |
| 19 | setIO | Set the high and low levels of the five I/O ports |
| 20 | setTextComplex | Set the text in the text program to traditional Chinese |
| 21 | getTextComplex | Gets whether the text in the text program is in traditional Chinese |

OnControlListener

|  |  |  |
| --- | --- | --- |
| **serial number** | **API interface function name** | **Interface description** |
| 1 | onStatus | Listen for connection status |
| 2 | onProcess | Monitor the progress of sending (receiving) data |
| 3 | onBackInfos | Listen for the returned information (string) |
| 4 | breakSocket | The listening connection is broken |
| 5 | onBackData（V1.3.2） | Information returned by listening (array) |
| 6 | onResultCode | Task execution state |
| 7 | onNetStatus | Receiving network communication status (whether communication is successful) |

## 5.2,Interface Details

1、ControlUtil

### setOnControlListener

|  |  |
| --- | --- |
| void setListener (OnControlListener listener) | |
| function | Set listener |
| parameter | Listener: Listener object |
| Return value | without |
| Other instructions |  |

### Getcurrentplayprogramno (modified by v1.3.2)

|  |  |
| --- | --- |
| void getCurrentPlayProgramNo (String cardIp, int port) | |
| function | ~~Get the program number and total number of programs currently played (V1.3.0)~~  ~~Get the current broadcast type, total number of programs and the current broadcast program number (V1.3.2)~~ |
| parameter | CardIP: IP address of control card  Port: port |
| Return value | without |
| Other instructions | ~~Get the value in onBackInfos of OnControlListener (V1.3.0)~~  Get the value (V1.3.2) from onbackdata (int [] backdata, int socket index) of OnControlListener:  There are three values in backData, namely, playing type, total number of programs and currently playing program number.  The play types are as follows:  0 general programs;  1 selective broadcasting programs;  2 Play the program specified in the schedule;  3 selective broadcasting program (trigger the small board selective broadcasting) |

### getCurrentProgramScreenshot

|  |  |
| --- | --- |
| void getCurrentProgramScreenshot (String cardIp, int port，String screenshotFileDir) | |
| function | Get a screenshot of the current screen |
| parameter | CardIP: IP address of control card  Port: port  ScreenshotFileDir: screenshot saving directory |
| Return value | without |
| Other instructions | In the onProcess of OnControlListener, you can monitor the progress of obtaining the screenshot  1.4.3 Update (Added a listening event that returns the control card IP address) :  controlUtil.setOnControlListenerNew(**new** OnControlListenerNew() {  @Override  **public** **void** onProcessWithIp(String ip, **int** process, **int** totalprocess, **int** socketindex) {  // **TODO** Auto-generated method stub  System.***out***.println(ip);  System.***out***.println(process+"---"+totalprocess);  }    }); |

### getCurrentPlayProgramContent（Added in V1.4.2）

|  |  |
| --- | --- |
| void getCurrentPlayProgramContent (String cardIp, int port, int programNo) | |
| function | Gets the text content of window 1 of the program number |
| params | cardIP: IP address of the controller card  port: port  programNo: Program number, 0: current program, 1: first program, 2: second program |
| return | without |
| Other instructions | Obtain the value from onBackInfos in OnControlListener |
| void getCurrentPlayProgramContent (String cardIp, int port, int programNo, int receiveTimeout) | |
| function | Get the program content of the program number |
| params | cardIP: IP address of the controller card  port: port  programNo: Program number, 0: current program, 1: first program, 2: second program  receiveTimeout: indicates the receiving timeout period |
| return | without |
| Other instructions | Obtain the value from onBackInfos in OnControlListener |

### getMultiWindowsData（Added in V1.4.2）

|  |  |
| --- | --- |
| void getMultiWindowsData (String cardIP, int port) | |
| Function | Gets the latest multi-window protocol sendtext content |
| Params | cardIP: IP address of the controller card  port; port |
| Return | Without |
| Other instructions | Obtain the value from onBackInfos in OnControlListener  Return data for example:  Success: {"size":2, "color":1, "content": a b c d e,}  Failed: Additional information  size: Text size, value 0-7 for size: 8, 12, 16, 24,32, 40, 48, 56  value 1-7 for color: red, green, yellow, blue, purple, skyblue and white.  content: indicates the text content |
|  |  |

### Selectplayprogram (added in v1.3.2)

|  |  |
| --- | --- |
| void selectPlayProgram (String cardIp, int port，boolean save, byte[] programIndex) | |
| function | Selective broadcasting program |
| parameter | CardIP: IP address of control card  Port: port  Save: save or not. After saving, the power-off restart is still valid  ProgramIndex: an array of program numbers to be broadcast, starting from 1 |
| Return value | without |
| Other instructions | OnStatus in OnControlListener monitors whether the call is successful. |

### Exitselectplay (added in v1.3.2)

|  |  |
| --- | --- |
| void exitSelectPlay (String cardIp, int port) | |
| function | Exit the selective broadcasting mode and return to normal broadcasting |
| parameter | CardIP: IP address of control card  Port: port |
| Return value | without |
| Other instructions | OnStatus in OnControlListener monitors whether the call is successful. |

### Getprogramfileinfos (added in v1.3.3)

|  |  |
| --- | --- |
| List<LpbFileInfo> getProgramFileInfos (String cardIp, int port) | |
| function | Get the list of sent program files |
| parameter | CardIP: IP address of control card  Port: port |
| Return value | List of program files |
| Other instructions | LpbFileInfo includes program file name and file size |

### Getuserdiscremainspace (added in v1.3.3)

|  |  |
| --- | --- |
| long getUserDiscRemainingSpace (String cardIp, int port) | |
| function | Acquire that remaining space size of the program area |
| parameter | CardIP: IP address of control card  Port: port |
| Return value | Size of remaining space in program area |
| Other instructions |  |

### Deletelpbfile (added in v1.3.3)

|  |  |
| --- | --- |
| boolean deleteLpbFile (String cardIp, int port, String LpbFileNameWithSuffix) | |
| function | Delete the specified program file |
| parameter | CardIP: IP address of control card  Port: port  LpbFileNameWithSuffix: the program file name is xxx.lpb |
| Return value | True indicates that the deletion was successful, otherwise false is returned |
| Other instructions |  |

### Formatuserdisk (added in v1.3.4)

|  |  |
| --- | --- |
| True formatUserDisk(  String cardIp, int port, int protocolModel, String deviceMac) | |
| function | Format user disk (area) |
| parameter | cardIP: IP address of the controller card  port: port  deviceMac (Optional) : unique Mac address of a device  protocolModel (Optional) : indicates a protocol type of 0:auto, 1:old, and 2:new |
| Return value | True indicates successful formatting, otherwise false is returned |
| Other instructions |  |

### Sendinstantmsg (added in v1.3.5)

|  |  |
| --- | --- |
| True sendInstantMsg(String cardIP, int port, int startX, int startY, int nWidth, int nHeight,  int playCount, String content, int crColor,  int nFontSize, int nSpeed,  int nStayTime, boolean scroll) | |
| function | Send timely information |
| parameter | cardIP: IP address of the controller card  port：port  startX : X starting position  startY : Y starting position  nWidth : width  nHeight: height  playCount: times of play  Content:information content  crColor value 1-7 for color: red, green, yellow, blue, purple, skyblue and white.  nFontSize Font size value 0-7 for size: 8, 12, 16, 24,32, 40, 48, 56  nSpeed Playback speed ranges from 100 to 0  nStayTime Residence time  scroll Whether to scroll (scroll left, text centered) |
| Return value |  |
| Other instructions |  |

### Getcurrentcardtimedate (added in v1.3.6)

|  |  |
| --- | --- |
| void getCurrentCardTimeDate(String cardIp, int port) | |
| function | Get the current time of the control card in the format of {"year", "month", "day", "hour", "minute", "second"} |
| parameter | CardIP: IP address of control card  Port: port |
| Return value | without |
| Other instructions | OnBackInfos in OnControlListener listens for the returned results. |

### Setcardtimedatetonow (added in v1.3.6)

|  |  |
| --- | --- |
| void setCardTimeDateToNow(String cardIp, int port) | |
| function | Set the control card time as the current time |
| parameter | CardIP: IP address of control card  Port: port |
| Return value | without |
| Other instructions | OnResultCode in OnControlListener listens for the returned result. |

### Setcardtimedate (added in v1.3.6)

|  |  |
| --- | --- |
| void setCardTimeDate(String cardIp, int port, int year, int month, int day, int hour, int minute, int second, int week) | |
| function | Set the control card time and set it manually. |
| parameter | CardIP: IP address of control card  Port: port  Year:  Month:  Day:  Hour:  Minute:  Second:  Week: |
| Return value | without |
| Other instructions | OnResultCode in OnControlListener listens for the returned result. |

### restartApp（V1.4.3新增）

|  |  |
| --- | --- |
| boolean restartApp(String cardIp, int port) | |
| function | Restart the APP |
| params | cardIP: IP address of the controller card  port：port |
| return | true表示重启成功，否则返回false |
| Other instructions |  |

### restartHard（Added in V1.4.3）

|  |  |
| --- | --- |
| boolean restartHard(String cardIp, int port) | |
| function | Restart hardware |
| params | cardIP: IP address of the controller card  port: port |
| return | true: The restart succeeds. Otherwise, false is returned |
| Other instruction |  |

3、OnControlLisenter

### getGlobalVariable（Added in V1.4.3）

|  |  |
| --- | --- |
| void getGlobalVariable (String cardIp, int port, byte[] variableNum) | |
| function | Gets the value of a global variable |
| params | cardIp: IP address of the controller card  port; port  variableNum: The variable number of each variable, for example, get the values of variables with variable numbers 1 and 2: [0x01,0x02] |
| return | Without |
| Other instructions | Obtain the value from onBackInfos in OnControlListener  Return data for example:  Success: {"1":" Value of variable number 1","2":" value of variable number 2"}  Failed: Additional information |

### setIO（Added in V1.4.4）

|  |  |
| --- | --- |
| boolean setIO(String ip, **int** port,**byte** io1,**byte** io2,**byte** io3,**byte** io4,**byte** io5) | |
| function | Set the high and low levels of the five I/O ports |
| params | cardIp: IP address of the controller card  port; port  io1: indicates the I/O port of route 1  io2: indicates the I/O port of route 2  io3: indicates the third I/O port  io4: indicates the I/O port of route 4  io5: indicates the I/O port of route 5  (0x81: High, 0x80: low, 0x00 or 0x01: invalid) |
| return | true: succeeds, false: fails |
| Other instructions |  |

### getCardMacAndPort（Added in V1.4.7）

|  |  |
| --- | --- |
| void getCardMacAndPort(String ip) | |
| function | Gets the MAC address and port of the control card |
| params | cardIp: IP address of the controller card |
| return | Without |
| Other instructions | Obtain the value from onBackInfos in OnControlListener  Return data for example:  Success: {"MAC": value, "Port": value}  Failed: Additional information |

### setTextComplex（Added in V1.4.5）

|  |  |  |
| --- | --- | --- |
| void setTextComplex (int textComplex) | | |
| function | Set whether the text in thn program is in traditional Chinese | |
| params | textComplex: indicates whether the text is traditional, 0: no, 1: traditional | |
| return |  |
| Other instructions |  | |

### getTextComplex（Added in V1.4.5）

|  |  |  |
| --- | --- | --- |
| int getTextComplex () | | |
| function | Check whether the text in the text program is in traditional Chinese | |
| params |  | |
| return | textComplex: indicates whether the text is traditional, 0: no, 1: traditional |
| Other instruction |  | |

### Onbackdata (added in v1.3.2)

|  |  |
| --- | --- |
| void onBackInfos (int[] backdata, int socketindex) | |
| function | Get the data returned by network communication |
| parameter | Backdata: returned data information.  Socketindex: request flag, used to distinguish which request returns data |
| Return value | without |
| Other instructions |  |

### Onbackinfos (modified by v1.3.6)

|  |  |
| --- | --- |
| void onBackInfos (String infoJson, int socketindex) | |
| function | According to socketindex, different infoJson contents are returned  INDEX\_PROGRAM\_INFO: Get the number and total number of programs currently played  INDEX\_GET\_CARD\_TIME\_DATE: get the current card time |
| parameter | InfoJson: returned data information. Json format string  Socketindex: request flag, used to distinguish which request returns data |
| Return value | without |
| Other instructions |  |

### Onnet status (modified by v1.3.6)

|  |  |
| --- | --- |
| void onNetStatus (int status, int socketindex) | |
| function | Receive network communication status (whether communication is successful) |
| parameter | Status: status, 0 failed, 1 succeeded, -1 exception |
|  | Socketindex; Request flag, used to distinguish which request returns data |
| Return value | without |
| Other instructions |  |

### onProcess

|  |  |
| --- | --- |
| void onProcess (int process, int totalprocess, int socketindex) | |
| function | File upload (download) progress |
| parameter | Process: current progress (uploaded data size)  Totalprocess: total progress (total data size)  Socketindex; Request flag, used to distinguish which request returns data |
| Return value | without |
| Other instructions |  |

### breakSocket

|  |  |
| --- | --- |
| void breakSocket (int socketindex) | |
| function | Receive network disconnection status |
| parameter | Socketindex; Request flag, used to distinguish which request returns data |
| Return value | without |
| Other instructions |  |

### On result code (added in v1.3.6)

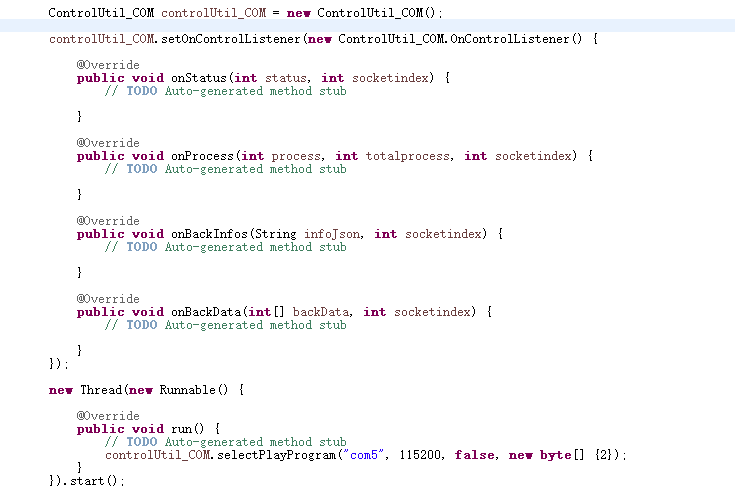
|  |  |
| --- | --- |
| void onResultCode(int resultCode, int socketindex) | |
| function | Task execution status |
| parameter | ResultCode: 1: success, 0: failure  Socketindex; Request flag, used to distinguish which request returns data |
| Return value | without |
| Other instructions |  |

# 6、Control (serial port)（Added in V1.3.4）

com.lumen.ledcenter3.protocol. ControlUtil\_COM

com.lumen.ledcenter3.protocol. ControlUtil\_COM.OnControlListener

## Call example



## 6.1、overview

ControlUtil\_COM

|  |  |  |
| --- | --- | --- |
| **serial number** | **API interface function name** | **interface specification** |
| 1 | setOnControlListener | Set monitor |
| 2 | getCurrentPlayProgramNo | Gets the number and total number of programs currently being played |
| 3 | getCurrentProgramScreenshot | Get a screenshot of the show currently playing |
| 4 | selectPlayProgram | Selected program |
| 5 | exitSelectPlay | Exit the program |
| 6 | getProgramFileInfos | Gets a list of sent program files |
| 7 | getUserDiscRemainingSpace | Gets the size of the remaining space in the program area |
| 8 | deleteLpbFile | Deletes the specified program file |
| 9 | formatUserDisk | Formatting user disk |
| 10 | restartApp | Restart the App |
| 11 | restartHard | Restart hardware |

ControlUtil\_COM.OnControlListener

|  |  |  |
| --- | --- | --- |
| **serial number** | **API interface function name** | **interface specification** |
| 1 | onStatus | Monitor connection status |
| 2 | onProcess | Monitor the progress of sending (receiving) data |
| 3 | onBackInfos | Listen for returned information (string) |
| 4 | onBackData（V1.3.2） | Listen for returned information (array) |

## 6.2、Interface Details

1、ControlUtil\_COM

### setOnControlListener

|  |  |
| --- | --- |
| void setListener (OnControlListener listener) | |
| function | Set monitor |
| params | listener: indicates the listener |
| return | Without |
| Other instructions |  |

### getCurrentPlayProgramNo

|  |  |
| --- | --- |
| String getCurrentPlayProgramNo (String portName, int baudrate) | |
| function | Gets the current play type, the total number of programs, and the number of programs currently being played |
| params | portName: Serial port number of the controller card  baudrate：Baud rate |
| return | Json string：{playingType: , programSum, programNumber, } |
| Other instructions | The value of onBackData (int[] backdata, int socketindex) in OnControlListener is obtained:  There are three values in backData, namely the playback type, the total number of programs, and the number of programs currently being played.  The playback type is divided into:  0 general programs;  1 Select broadcast programs;  2 Play programs specified in the schedule;  3 Program selection (trigger board selection) |

### getCurrentProgramScreenshot

|  |  |
| --- | --- |
| void getCurrentProgramScreenshot (String portName, int baudrate，String screenshotFileDir) | |
| function | Get a screenshot of the current screen |
| params | portName: Serial port number of the controller card  baudrate：Baud rate  screenshotFileDir：Screenshot save directory |
| return | Without |
| Other instructions | In the onProcess of OnControlListener, you can monitor the progress of obtaining the screenshot |

### selectPlayProgram

|  |  |
| --- | --- |
| void selectPlayProgram (String portName, int baudrate，boolean save, byte[] programIndex) | |
| function | Selected program |
| params | portName: serial port number of the controller card  baudrate: baudrate save: whether to save. After saving, power off restart still works  save: Whether to save. After saving, power off restart still works  programIndex: An array of program numbers to broadcast, starting with 1 |
| return | without |
| Other instructions | The onStatus in OnControlListener listens for whether the call was successful. |

### exitSelectPlay

|  |  |
| --- | --- |
| void exitSelectPlay (String portName, int baudrate) | |
| function | Exit the program mode and return to normal playback |
| params | portName: serial port number of the controller card  baudrate：Baud rate |
| return | Without |
| Other instructions | The onStatus inside OnControlListener listens for success. |

### getProgramFileInfos

|  |  |
| --- | --- |
| List<LpbFileInfo> getProgramFileInfos (String portName, int baudrate) | |
| function | Gets a list of sent program files |
| params | portName: serial port number of the controller card  baudrate：baud rate |
| return | Program file list |
| Other instructions | LpbFileInfo includes the program filename and file size |

### getUserDiscRemainingSpace

|  |  |
| --- | --- |
| long getUserDiscRemainingSpace (String portName, int baudrate) | |
| function | Gets the amount of space left in the program area |
| params | portName: control card serial port  baudrate：baud rate |
| return | The amount of space left in the program area |
| Other instructions |  |

### deleteLpbFile

|  |  |
| --- | --- |
| boolean deleteLpbFile (String portName, int baudrate, String LpbFileNameWithSuffix) | |
| function | Delete the specified program file |
| params | portName: control card serial port  baudrate：baud rate  LpbFileNameWithSuffix：program filename for example xxx.lpb |
| return | True indicates that the deletion was successful; otherwise, false is returned |
| Other instructions |  |

### formatUserDisk

|  |  |
| --- | --- |
| boolean formatUserDisk(String portName, int baudrate) | |
| function | Format the user disk (area) |
| params | portName: control card serial port  baudrate：baud rate |
| return | True indicates that the deletion was successful; otherwise, false is returned |
| Other instructions |  |

### restartApp（Added in V1.4.3）

|  |  |
| --- | --- |
| boolean restartApp(String portName, int baudrate) | |
| function | Restart app |
| params | portName: control card serial port  baudrate：baud rate |
| return | true: The restart succeeds. Otherwise, false is returned |
| Other instructions |  |

### restartHard（Added in V1.4.3）

|  |  |
| --- | --- |
| boolean restartHard(String portName, int baudrate) | |
| function | Restart hardware |
| params | portName: control card serial port  baudrate：baud rate |
| return | True:the restart succeeds，otherwise ,false is returned |
| Other instructions |  |

2、OnControlLisenter

### onBackData

|  |  |
| --- | --- |
| void onBackInfos (int[] backdata, int socketindex) | |
| function | Obtain the data returned by the serial port communication |
| params | backdata: Returned data information  socketindex：Request flag, used to distinguish which request returned the data |
| return | Without |
| Other instructions |  |

### onBackInfos

|  |  |
| --- | --- |
| void onBackInfos (String infoJson, int socketindex) | |
| function | Gets the number and total number of programs currently being played |
| params | infoJson: Returned data information. json format string  socketindex: Request flag, used to distinguish which request returned data |
| return | Without |
| Other instructions |  |

### onStatus

|  |  |
| --- | --- |
| void onStatus (int status, int socketindex) | |
| function | Receiving network communication status (whether it was successfully sent) |
| params | status: indicates the status. 0 is failed, 1 is successful, and -1 is abnormal |
| Socketindex; Request flag, used to distinguish which request returned the data |
| return | Without |
| Other instructions |  |

### onProcess

|  |  |
| --- | --- |
| void onProcess (int process, int totalprocess, int socketindex) | |
| function | File upload (download) progress |
| params | process: Current progress (size of uploaded data)  totalprocess: Total progress (total data size)  Socketindex; Request flag, used to distinguish which request returned the data |
| return | 无 |
| Other instructions |  |

# 7,LMServer (added in v1.3.1)

## 7.1 Overview of LmServerAPI Interface

|  |  |  |
| --- | --- | --- |
| **serial number** | **API interface function name** | **Interface description** |
| 1 | LmServer\_Login | log in |
| 2 | LmServer\_Logout | Exit login |
| 3 | LmServer\_LockCard | Lock the control card, called before communicating with the control card |
| 4 | LmServer\_UnLockCard | Unlock the control card and call it after the communication of the control card is completed |
| 5 | LmServer\_SendPacketData\_Net | Send communication data packet to control card (network client) |
| 6 | LmServer\_SendPacketData\_GPRS | Send communication data packet to control card (GPRS client) |
| 7 | LmServer\_SendProgramFiles\_Net | Send program package to control card (network client) |
| LmServer\_SendProgramFilesAppend\_Net | **Add and send program package to control card (network client)** |
| 8 | LmServer\_SendProgramFiles\_GPRS | Send program package to control card (GPRS client) |
| LmServer\_SendProgramFilesAppend\_GPRS | **Add and send program package to control card (GPRS client)** |
| 9 | LmServer\_SendHeartbeatPacket | Occurrence of heartbeat packet |
| 10 | LmServer\_RecvCommPacket | Receive data from the server |
| The following is the specific function interface, the call needs to follow the following steps of the use of this class function. Called in step 3 | | |
| 11 | LmServer\_SelectProgram | Selective broadcasting program |
| 12 | LmServer\_ExitSelectProgram | Quit the selective broadcast program |
| 13 | LmServer\_SetVariableValue | Set variable value |
| 14 | LmServer\_GetLpbFileInfos | Get the list of sent program files |
| 15 | LmServer\_GetFreeSpace | Acquire that remaining space size of the program area |
| 16 | LmServer\_DeleteLpb | Delete the specified program file |
| 17 | getCardDevices | Get all device information |
| 18 | LmServer\_GetScreenshot | Gets the current screenshot |
| 19 | getPlayingProgram | Get the number and content of the currently playing program (valid for text programs only) |
| 20 | getProgramContent | Get the playback content of the specified program number (valid for text programs only) |
| 21 | getMultiProgramContent | Get the number and content of all programs (valid for text programs only) |
| 22 | sendSimpleText | Sends formatted text to the specified window |
| 23 | setTextComplex | Set the text in the text program to traditional Chinese characters |
| 24 | getTextComplex | Gets whether the text in the text program is traditional Chinese |

## Example call

**The use steps of this kind of function are as follows:**

Step 1: User login

Step 2: Lock the control card

Step 3: Send communication data packet

(Step 4: Receive communication packets) Called as needed

Step 5: Unlock the control card

Step 6: Exit the login

Note: If the user needs to log in online for a long time, it is necessary to send heartbeat packets to the server regularly to maintain the long connection, and the longest interval between heartbeat packets is not more than 120 seconds. The interval time refers to the interval time between the application and LmServer without data exchange.

## 

## 7.2,Detailed description of LmServerAPI interface

#### LmServer\_Login

|  |  |
| --- | --- |
| int LmServer\_Login(String ip, int port, String userName, String password); | |
| function | User login |
| parameter | Ip: IP address of the server, such as "192.168.1.100" |
| Port: server user listening port |
| UserName: user name |
| Password: user password |
| Return value | -1: communication error  0: login failed  1: login succeeded.  2: User is not registered  3: The user has logged in  4: Wrong password  5: Unable to connect to the server |
| Other instructions |  |

#### LmServer\_Logout

|  |  |
| --- | --- |
| int LmServer\_Logout(); | |
| function | User logs out |
| parameter |  |
| Return value |  |
| Other instructions |  |

#### LmServer\_LockCard

|  |  |
| --- | --- |
| int LmServer\_LockCard( string deviceID ); | |
| function | User locking control card |
| parameter | DeviceID: Control card device ID |
| Return value | -1: communication error  0: failed  1: success |
| Other instructions | The user calls this function before communicating with the control card |

#### LmServer\_UnLockCard

|  |  |
| --- | --- |
| int LmServer\_UnLockCard( string deviceID ); | |
| function | User unlock control card |
| parameter | DeviceID: Control card device ID |
| Return value | -1: communication error  0: failed  1: success |
| Other instructions | The user calls this function after the communication with the control card is completed |

#### LmServer\_SendPacketData\_Net

|  |  |
| --- | --- |
| int LmServer\_SendPacketData\_Net(String deviceID, byte[] packetData) | |
| function | Send communication data packet to control card (network client) |
| parameter | StrDeviceID: Control card device ID of communication  PacketData: protocol data, note: "protocol data" is consistent according to the general protocol of our company, and is 0x68 to check code. |
| Return value | -1: communication error  0: send failed  1: sent, waiting for data to be returned  2: No right to communicate with this control card  3: The control card is not online  4: The control card is currently communicating with other users  5: Send successfully  6: The receiving buffer length is insufficient |
| Other instructions |  |

#### LmServer\_SendPacketData\_GPRS

|  |  |
| --- | --- |
| int LmServer\_SendPacketData\_GPRS(String deviceID, byte[] packetData) | |
| function | Send communication data packet to control card (GPRS client) |
| parameter | StrDeviceID: Control card device ID of communication  PacketData: protocol data, note: "protocol data" is consistent according to the general protocol of our company, and is 0x68 to check code. |
| Return value | -1: communication error  0: send failed  1: sent, waiting for data to be returned  2: No right to communicate with this control card  3: The control card is not online  4: The control card is currently communicating with other users  5: Send successfully  6: The receiving buffer length is insufficient |
| Other instructions |  |

#### LmServer\_RecvCommPacket

|  |  |
| --- | --- |
| Int[] LmServer\_RecvCommPacket(); | |
| function | Receive data from the server |
| parameter |  |
| Return value | The received data, if null, means there is no returned data |
| Other instructions |  |

#### LmServer\_SendProgramFiles\_Net

|  |  |
| --- | --- |
| int LmServer\_SendProgramFiles\_Net(  String deviceID,  String tempDir,  List<String> lpdFilePaths,  int screenWidth,  int screenHeight,  int colorType) | |
| function | Send program package to control card (network client) |
| parameter | StrDeviceID: Control card device ID of communication  TempDir: directory where temporary files are stored  LpdFilePaths: list of lpd file paths to send  ScreenWidth: screen width. It needs to be consistent with the program file  ScreenHeight: screen height. It needs to be consistent with the program file  ColorType: screen color and gray level. It needs to be consistent with the program file |
| Return value | -1: communication error  0: send failed  1: sent, waiting for data to be returned  2: No right to communicate with this control card  3: The control card is not online  4: The control card is currently communicating with other users  5: Send successfully  6: The receiving buffer length is insufficient |
| Other instructions |  |

#### Lmserver \_ sendprogramfilesappend \_ net (added in v1.3.3)

|  |  |
| --- | --- |
| int LmServer\_SendProgramFilesAppend\_Net(  String deviceID,  String tempDir,  List<String> lpdFilePaths,  int screenWidth,  int screenHeight,  int colorType) | |
| function | Send program package to control card (network client) |
| parameter | StrDeviceID: Control card device ID of communication  TempDir: directory where temporary files are stored  LpdFilePaths: list of lpd file paths to send  ScreenWidth: screen width. It needs to be consistent with the program file  ScreenHeight: screen height. It needs to be consistent with the program file  ColorType: screen color and gray level. It needs to be consistent with the program file |
| Return value | -1: communication error  0: send failed  1: sent, waiting for data to be returned  2: No right to communicate with this control card  3: The control card is not online  4: The control card is currently communicating with other users  5: Send successfully  6: The receiving buffer length is insufficient |
| Other instructions |  |

#### LmServer\_SendPacketFiles\_GPRS

|  |  |
| --- | --- |
| int LmServer\_SendProgramFiles\_GPRS(  String deviceID,  String tempDir,  List<String> lpdFilePaths,  int screenWidth,  int screenHeight,  int colorType) | |
| function | Send program package to control card (GPRS client) |
| parameter | StrDeviceID: Control card device ID of communication  TempDir: directory where temporary files are stored  LpdFilePaths: list of lpd file paths to send  ScreenWidth: screen width. It needs to be consistent with the program file  ScreenHeight: screen height. It needs to be consistent with the program file  ColorType: screen color and gray level. It needs to be consistent with the program file |
| Return value | -1: communication error  0: send failed  1: sent, waiting for data to be returned  2: No right to communicate with this control card  3: The control card is not online  4: The control card is currently communicating with other users  5: Send successfully  6: The receiving buffer length is insufficient |
| Other instructions |  |

#### Lmserver \_ sendprogramfilesappend \_ GPRS (added in v1.3.3)

|  |  |
| --- | --- |
| int LmServer\_SendProgramFilesAppend\_GPRS(  String deviceID,  String tempDir,  List<String> lpdFilePaths,  int screenWidth,  int screenHeight,  int colorType) | |
| function | Send program package to control card (GPRS client) |
| parameter | StrDeviceID: Control card device ID of communication  TempDir: directory where temporary files are stored  LpdFilePaths: list of lpd file paths to send  ScreenWidth: screen width. It needs to be consistent with the program file  ScreenHeight: screen height. It needs to be consistent with the program file  ColorType: screen color and gray level. It needs to be consistent with the program file |
| Return value | -1: communication error  0: send failed  1: sent, waiting for data to be returned  2: No right to communicate with this control card  3: The control card is not online  4: The control card is currently communicating with other users  5: Send successfully  6: The receiving buffer length is insufficient |
| Other instructions |  |

#### LmServer\_SendHeartbeatPacket

|  |  |
| --- | --- |
| int LmServer\_SendHeartbeatPacket (); | |
| function | Send heartbeat packet |
| parameter |  |
| Return value | 1: success  0: failed  -1: communication error |
| Other instructions | The maximum interval between heartbeat packet is no more than 120 seconds |

#### Lmserver \_ select program (added in v1.3.2)

|  |  |
| --- | --- |
| int LmServer\_SelectProgram(String deviceID, boolean save, byte[] programIndex) | |
| function | Selective broadcasting program |
| parameter | StrDeviceID: Control card device ID of communication  Save: save or not. After saving, the power-off restart is still valid  ProgramIndex: an array of program numbers to be broadcast, starting from 1 |
| Return value | -1: communication error  0: send failed  1: sent, waiting for data to be returned  2: No right to communicate with this control card  3: The control card is not online  4: The control card is currently communicating with other users  5: Send successfully  6: The receiving buffer length is insufficient |
| Other instructions |  |

#### Lmserver \_ exitselectprogram (added in v1.3.2)

|  |  |
| --- | --- |
| int LmServer\_ ExitSelectProgram (String deviceID) | |
| function | Quit the selective broadcast program |
| parameter | StrDeviceID: Control card device ID of communication |
| Return value | -1: communication error  0: send failed  1: sent, waiting for data to be returned  2: No right to communicate with this control card  3: The control card is not online  4: The control card is currently communicating with other users  5: Send successfully  6: The receiving buffer length is insufficient |
| Other instructions |  |

#### Lmserver \_ setvariablevalue (added in v1.3.2)

|  |  |
| --- | --- |
| int LmServer\_ SetVariableValue (String deviceID, boolean saveToFlash, int variableNum, byte[]... datas) | |
| function | Set variable value |
| parameter | StrDeviceID: Control card device ID of communication  Save saveToFlash?  Number of variableNum variables  Datas variable value |
| Return value | -1: communication error  0: send failed  1: sent, waiting for data to be returned  2: No right to communicate with this control card  3: The control card is not online  4: The control card is currently communicating with other users  5: Send successfully  6: The receiving buffer length is insufficient |
| Other instructions |  |

#### Lmserver \_ getlpbfileinfos (added in v1.3.3)

|  |  |
| --- | --- |
| List<LpbFileInfo> LmServer\_GetLpbFileInfos (String deviceID) | |
| function | Get the list of sent program files |
| parameter | StrDeviceID: Control card device ID of communication |
| Return value | List of program files |
| Other instructions | LpbFileInfo includes program file name and file size |

#### Lmserver \_ getfreespace (added in v1.3.3)

|  |  |
| --- | --- |
| long LmServer\_GetFreeSpace (String deviceID) | |
| function | Acquire that remaining space size of the program area |
| parameter | StrDeviceID: Control card device ID of communication |
| Return value | Size of remaining space in program area |
| Other instructions |  |

#### LmServer\_DeleteLpb(V1.3.3 (added in v1.3.3)

|  |  |
| --- | --- |
| Int LmServer\_DeleteLpb (String deviceID, String LpbFileNameWithSuffix) | |
| function | Delete the specified program file |
| parameter | StrDeviceID: Control card device ID of communication  LpbFileNameWithSuffix: the program file name is xxx.lpb |
| Return value | -1: communication error  0: send failed  1: sent, waiting for data to be returned  2: No right to communicate with this control card  3: The control card is not online  4: The control card is currently communicating with other users  5: Send successfully  6: The receiving buffer length is insufficient |
| Other instructions |  |

#### Getcard devices (added in v1.3.5)

|  |  |
| --- | --- |
| List<CardDevice> getCardDevices() | |
| function | Get all device information |
| parameter | without |
| Return value | List<CardDevice > device list |
| Other instructions |  |

#### Lmserver \_ getscreenshot (added in v1.3.5)

|  |  |
| --- | --- |
| LmServer\_GetScreenshot(String screenshotFilePath) | |
| function | Get the current screen shot |
| parameter | Save path and name of screenshot file (PNG picture) |
| Return value | 5: Success  Others: failed |
| Other instructions |  |

#### Getplaying program (added in v1.3.5)

|  |  |
| --- | --- |
| PlayingProgram getPlayingProgram() | |
| function | Get the number and content of the currently playing program (only valid for text programs) |
| parameter | without |
| Return value | PlayingProgram: the program being played |
| Other instructions |  |

#### Getprogramcontent (added in v1.3.5)

|  |  |
| --- | --- |
| String getProgramContent(int programNo) | |
| function | Get the broadcast content of the specified program number (only valid for text programs) |
| parameter | ProgramNo program number (starting from 1) |
| Return value | Text content of program |
| Other instructions |  |

#### Getmultiprogramcontent (added in v1.3.5)

|  |  |
| --- | --- |
| List<PlayingProgram> getMultiProgramContent() | |
| function | Get the numbers and contents of all programs (valid for text programs only) |
| parameter | without |
| Return value | List<PlayingProgram >: list of program information |
| Other instructions |  |

#### sendSimpleText (Added in V1.4.3)

|  |  |
| --- | --- |
| int sendSimpleText( int nWndNo,  String content, int crColor,  int nFontSize, int nSpeed, int nEffect,  int nStayTime, int nAlignmentHori, int nAlignmentVert) | |
| Use | Sends formatted text to the specified window |
| Params | nWndNo: Sequence number of window, virtual value 0-7.  Content: text string.  crColor: value 1-7 for color: red, green, yellow, blue, purple, skyblue and white.  nFontSize: value 0-7 for size: 8, 12, 16, 24,32, 40, 48, 56  nSpeed: value 1 ~ 100. The smaller the value, the faster the speed.  nEffect: Display effect coding  nStayTime: The unit is seconds.  nAlignmentHori: Horizontal alignment (0: left alignment, 1: horizontal middle, 2: right alignment)  nAlignmentVert: Vertical alignment (0: top alignment, 1: vertical middle, 2: bottom alignment) |
| Return | 5. Success  Other: Failure |

#### setTextComplex（Added in V1.4.5）

|  |  |  |
| --- | --- | --- |
| void setTextComplex (int textComplex) | | |
| function | Sets whether the text in the text program is traditional Chinese | |
| params | textComplex: Traditional or not, 0: No, 1: Traditional is used | |
| return |  |
| Other instructions |  | |

#### getTextComplex（Added in V1.4.5）

|  |  |  |
| --- | --- | --- |
| int getTextComplex () | | |
| function | Query whether the text in the text program is traditional Chinese | |
| params |  | |
| return | textComplex: Traditional or not, 0: No, 1: Traditional is used |
| Other instructions |  | |

# 8、Communication key management

## 8.1、An overview of the key management interface

CypherManager

|  |  |  |  |
| --- | --- | --- | --- |
| **Serial number** | **API interface function name** | **Interface description** | **completion** |
| 1 | setSecretKey | set control card message key | Implemented |

## 8.2、Key management interface details

### setSecretKey

|  |  |
| --- | --- |
| void setSecretKey (String secretKey) | |
| function | Set control card message key |
| params | secretKey: The key |
| return | Without |
| Other instructions | 1. Call this method before communicating with the control card  2. The password must be 8 characters long and can be a mixture of numbers and letters |