JetFileII API Ver2.10

Renewal record

Ver.	Modified information	Person	Date
Ver1.0	Version 1	oliny	2011-09-01
Ver1.1	czReadPlayLogII 为 czReadDB Changing czReadPlayLogII to czReadDB	Lei	2011-11-07
Ver1.2	Adding Read/Write SysFile,FontFile,TextFile PictureFile,StringFile,ArrayPictureFile	oliny	2011-11-21
Ver1.3	clearing up the operation interface	Lei	2013-4-3
Ver1.4	adding scheduling process	Lei	2013-8-16
Ver1.5	Check the index	Lei	2013-9-2
Ver1.6	Add vpu	Lei	2015-8-10
Ver1.7	Add czEasyReadFile/czEasyWriteFile/cold	Lei	2016-4-5
Ver1.8	Add Waring Function	Lei	2016-5-18
Ver1.9	Add Switch offline/online command	Lei	2016-5-23
Ver2.0	Modify the czShowMsg	Lei	2016-9-2
Ver2.01	Add czReadBrightInfo	Lei	2016-9-23
Ver2.02	Add nmg file converter	Lei	2016-10-09
Ver2.03	EasyAPI Support Serial	Fm	2017-1-13
Ver2.04	Add czReadStats and EasyAPI add czShowFilesSpe	Fm	2017-10-16
Ver2.05	Clearing up a partial description of czGetPowerState	Fm	2017-12-12
Ver2.06	Clearing up a partial description of the interfrace	Fm	2018-6-6
Ver2.07	Add Car park interface	Fm	2018-9-6
Ver2.08	Add czReadCurScreenshot Add czPlaySoundFile Add czAdjustVolume	Fm	2018-9-24
Ver2.09	Add czSetEnableMulitZone Add czSetPageCount Add czDivideSpeZone Add czGetMulitZoneSetting	Fm	2018-12-3
Ver2.10	Add description of error code Add pix check operations Add czShowFilesII	Fm	2019-6-5

1.Introduction

JetFile II API provides a strong, simple and extensionable interface to the development of screen communication program. The protocol for communicating with the screen is JetFileII, it is capable, reliable and flexible. JetFileII API is improved on the basis of JetFileII, and it integrates the functions such as sending and receiving data, subpackaging, burstification, which means the users don't need to care about the detailed and complicated communication process, and can develop the powerful communication software.

JetFileII API function is packaged inside DLL, so in order to schedule JetFileII function, DLL must be added into related project. DLL has nothing to do with language, so it can be scheduled by any programming language which supports dynamic linkbases, such as VC,VB,C#,Delphi,C++Builder,PB. The development platform is WINDOWS.

Composing Note: the prefix of 0x means hexadecimal number, such as <0x30>, the symble of "" or ' ' means ASCII character. Others means decimal number.

basic data type definition
INT8U: Unsigned 8 bit quantity
INT8S: Signed 8 bit quantity
INT16U: Unsigned 16 bit quantity
INT16S: Signed 16 bit quantity
INT32U: Unsigned 32 bit quantity
INT32S: Signed 32 bit quantity

LED screen path Note:

Most LED screens currently only support 8.3 file format, and some support long file name. Please contact Support to confirm whether your screen supports long file name. In order to make the operation simple, there are Disc C,D,E and F. Disc C is for saving systematic files, the display files should not be saved here. Disc D,E and F are for saving the display files. Disc E is RAM disc, Disc F is extensionable Disc.

The normal formats of data discs of the screen are as below:
D,E,F
|- T :text file folder, For saving text files
|- S :string file folder, For saving String files
|- P :Picture file folder, For saving Picture files
|- A :Array picture file folder,For saving Array picture files
|- F :For saving animation and video format files
|- Q :For saving QST files
|- TEMP:For saving temporary files

2. Call API Process

2.1.Initialization interface

Before scheduling the basic operational interface, user needs to initialize API with the function below.

Function Prototype

INT32U czInitAPI(_czinterface* czif);

Description

The Function is for initializing the API. If there is multithreading program, the initializing function needs to be scheduled for each threading program which needs to use the API.

Parameter

czinterface: **[in]**

Note for control structure

```
tpedef struct czinterface
{
   conn_type type;
   INT32U inited;
   INT8U   ip[32];
   INT8U    com[16];
   INT32U port;
   INT32U baudrate;
   void    *signPointer;
   void    *taskPointer;
   INT16U SrcAddr;
   INT16U DstAddr;
   INT32U Retry;
   INT32U Timeoutms;
   INT32U Stoped;
   INT16U totalSteps;
   INT16U curSteps;
   INT16U packetSize;
   INT8U   Rev[26];
   void (*pre_comm) (struct czinterface*);
   void (*post_comm) (struct czinterface*);
   void (*update_status) (struct czinterface*,INT32U code,INT32U precent);
   INT32U (*read) (void*,INT8U *buf, INT32U size, INT32U timeoutms);
   INT32U (*write)(void*,INT8U *buf, INT32U size);

    void    (*log)  (struct czinterface*,const char *file, int line, int level,const char
                    *fmt, ...);
    void     (*curShowFilter) (struct czinterface*,const char *srcfile, const char
                               *dstfile,int frames,int delays);
}_czinterface;
```

conn_type type :

Type of the communication and definition can be seen as below:

```
typedef  enum
  {
     COMM_UDP,
     COMM_TCP,
     COMM_COM
}conn_type;
```

inited:
API Internal usage
ip:
IP address, in dotted fomat, eg.169.254.10.49
com:
Serial port communication by 'com' string, eg.COM1
port:
Network communication on the port number, it is in 9520 for communication
baudrate:
Serial communication with Baudrate, eg. 115200
signPointer:
Signpointer, being used by caller
taskPointer:
Taskpointer, being used by caller
SrcAddr:
Source address, it has to be matched with the JetFileII
DstAddr
GGUU address of the LED display, Unit Addr.is in high, Group Addr is on the is in low
Retry:
Communication retry times.
Timeoutms:
when it is in communication, and timeout and the unit is millisecond
Stoped:
when it is out of communication,
totalSteps:
to edit the schedule about the aggregate scheduling, it is used by the caller and read/written in the multi-files
curSteps:
the current step is matched to the aggregate scheduling
packetSize:
the unit is Bytes and the value is 64-1280,
Rev[26]:
reserve
void (*pre_comm) (struct czinterface*):
the callback function for the pre communication , API will change into callback function before being connected, and it is in the state of its system structure

void (*post_comm) (struct czinterface*):
the callback function of the post communication
void (*update_status) (struct czinterface*,INT32U code,INT32U precent):

the updated status of the callback function ,when the display is beinoperated,the updated status will be called back automatically.
Code shows o,shows no error,precent means shcedule,value range from 0 to 100.

INT32U (*read) (void*,INT8U *buf, INT32U size, INT32U timeoutms):
If there is configuration in the callback function, it means it will not use the Ethernet or serial port offered internally, and it needs to use the callback function to read the information. If there is no configuration, it means it is used with the DLL network or COM.

INT32U (*write)(void*,INT8U *buf, INT32U size):

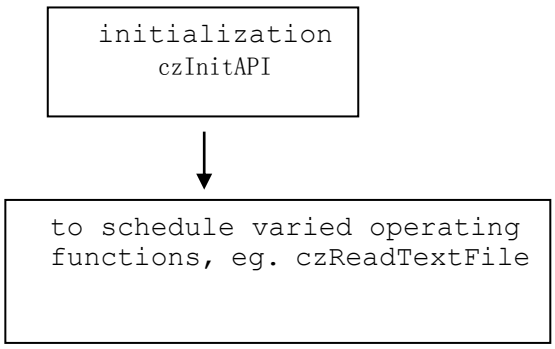
void (*log) (struct czinterface*,const char *file, int line, int level,const char *fmt, ...):
when the log is being called,and the information output also comes from here

void (*curShowFilter) (struct czinterface*,const char *srcfile, const char *dstfile,int frames,int delays):
When reading the playing information, it will use the callback function sometimes

Return Value
If the function is successfully operated, return to 0. Otherwise, return to Error code.

2.2 Call API Process

The chart of API scheduleing can be shown as below:



For Easy API, the function can be called directly and no need to operate the initialization

2.3 Error Code Description

If API is successfully operated, return to 0. Otherwise, return to Error code.Error code comparison table as described in Appendix Ⅰ.

Examples

```
Char *errDesc;
INT32U errCode=0x5100;
errDesc = czErrorDesc(errCode);
```

3. BACIS OPERATION INTERFACE DEFINITION

3.1 OPERATION OF INFORMATION READBACK

3.1.1(czReadSysFile)system file readback

Function Prototype

INT32U INT32U czReadSysFile(INT8U* systemFile,INT8U *PCPath)

description

To read the system file of the display ,eg. CONFIG.SYS,SEQUENT.SYS

parameter

systemFile: **[in]**

The system file name of the display

PCPath：**[in]**

To read the storage location of the file

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
INT8U SystemFile [64],path[256];
SystemFile = “CONFIG.SYS”;
path = “C:\\CONFIG.SYS”;
if(czReadSysFile(SystemFile,path)== 0)
{
      //OK
}
```

Correlation function

3.1.2 (czReadFontFile) Font file readback

Function Prototype

INT32U czReadFontFile (INT8U* fontName,INT8U* PCPath)

Description

To read the specific path file of the display

Parameter

fontName: **[in]**
The name of the font on the display
PCPath：**[in]**
To read the storage location of the font file

Return Value
If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
INT8U Fontname[64],path[256];
name = “Normal11.fnt”;
path = “C:\\Normal11.fnt”;
if(czReadFontFile (Fontname,path)== 0)
{
      //OK
}
```

Correlation function

3.1.3 (czReadTextFile) Text file readback

Function Prototype
INT32U czReadTextFile (INT8U Drive,INT8U* textName,INT8U* PCPath)

Description
To read the Text file of the specific disk on the display
Parameter

Drive: **[in]**
To read files in all the disks of the display

textName: **[in]**
To specify the name of the Text file on the display
PCPath：**[in]**
To read the storage location of the file

Return Value
If the function is successfully operated, return to 0. Otherwise, return to Error code.
Examples

```
INT8U Letter,TextName[64],path[256];
Letter = ‘D’;
TextName = “temp.pmg”;
path = “C:\\temp.pmg”;
if(czReadTextFile(Letter,TextName,path)== 0)
{
      //OK
}
```

Correlation function

3.1.4 (czReadStringFile) String file readback

Function Prototype

INT32U czReadStringFile (INT8U Drive,INT8U* stringName,INT8U* PCPath);

Decription

To read the String file in the specific disk of the display

Parameter

Drive: **[in]**

To read files in all the disks of the display

stringName: **[in]**

To specify the name of theString file on the display

PCPath：**[in]**

To read the storage location of the file

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
INT8U Drive,StringName[64],path[256];
Drive = 'D';
TextName = "1";
path = "C:\\11.txt";
if(czReadStringFile (Drive, StringName,path)== 0)
{
      //OK
}
```

Correlation function

3.1.5 (czReadPictureFile) Picture file readback(BMP file)

Function Prototype Function Prototype

INT32U czReadPictureFile (INT8U Drive,INT8U* pictureName,INT8U* PCPath);

Description

To read the picture file in the specific disk of the display

Parameter

Drive: **[in]**

To read files in all the disks of the display

pictureName: **[in]**

To specify the name of the Text file on the display

PCPath：**[in]**

To read the storage location of the file

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
INT8U Letter,PictureName[64],path[256];
Letter = 'D';
PictureName = "12.bmp";
path = "C:\\21.bmp";
if(czReadPictureFile (Letter,PictureName,path)== 0)
{
}
```

Correlation function

3.1.6 (czReadArrPicFile) ArrayPicture file readback

Function Prototype

INT32U czReadArrPicFile (INT8U Drive,INT8U* arrPicName,INT8U* PCPath)

Description

To read the pmg file of the specific disk on the display

Parameter

Drive: **[in]**

To read files in all the disks of the display

arrPicName: **[in]**

To specify the name of the Text file on the display

PCPath：**[in]**

To read the storage location of the file

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
INT8U Letter,ArrPicName[64],path[256];
Letter = 'D';
ArrPicName = "temp.pmg";
path = "C:\\ temp.pmg";
if(czReadArrPicFile(Letter,ArrPicName,path)== 0)
{
}
```

Correlation functionc

3.1.7 (czReadSpecPathFile) Specific path file readback

Function Prototype

INT32U czReadSpecPathFile (INT8U* SpecialFile,INT8U* PCPath);

Description

To read the specific path of the file on the display

Parameter

SpecialFile: **[in]**

The specific file and name on the display

PCPath：**[in]**

To read the storage location of the file

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
INT8U name[64],path[256];
name = “D:\\A\\temp.pmg”;
path = “D:\\temp.pmg”;
if(czReadSpecPathFile (name,path)== 0)
{
}
```

Correlation function

3.1.8 (czReadSystemSet) System configuration readback

Function Prototype Function Prototype

INT32U czReadSystemSet (SYSTEM_SET* systemSet);

Description

To read the parameter of the display

Parameter

systemSet: **[out]**

To read the parameter system of the display

```
typedef struct
{
   INT16U cpuv;
  INT16U tcpv;
  INT16U filesv;
  INT16U fpga;
  INT16U height;
  INT16U width;
  INT16U protocol;
  INT8U   gg;
  INT8U   uu;
```

```
}SYSTEM_SET;
```

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
SYSTEM_SET systemSet;
if(czReadSystemSet(&systemSet)== 0)
{
}
```

Correlation function

3.1.9 (czReadCurrentState) Current system status readback

Function Prototype

INT32U czReadCurrentState (CURRENT_STATE* State void *ExtData, INT32U DataSize);

Description

To read the current state of the system, eg. Brightness, playing mode

Parameter

currentState: **[out]**

To read the current state of the display

The size of structure system

```
typedef struct
{
  INT8U sysState;
  INT8U inTemp;
  INT8U outTemp;
  INT8U AutoPower;
  INT8U Humidity;
  INT8U Samples;
  INT8U BrightLevel;
}CURRENT_STATE;
```

sysState:

the task of the system

0	scheduling mode
1	emergency playing status
2	black out mode
3	remote control mode
4	test mode
5	wireless connection running mode
6	end-to-end running mode
7	backup mode

8 power off mode
9 high temperature protection mode

inTemp:
the inner temperature of the display is （℃）, otherwise, it is oxFF
outTemp:
the temperature outside the display is （℃）, otherwise, it is oxFF

AutoPower:
auto power off, 1 is to permit, 0 is to prohibit
Humidity:
humidity
Samples:
the samples from the light sensor
BrightLevel:
brightness level is divided into 0-100 , 100 is the brightest level

ExtData: **[out]**
Return Value additional returned value
DataSize
data size of the additional returned value

Return Value
If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
CURRENT_STATE currentState;
if(czReadCurrentState(&currentState,NULL,0) != 0)
{
}
```

Correlation function

3.1.10 (czGetSNMAC) System SN,MAC information readback

Function Prototype

INT32U czGetSNMAC (INT8U* SN,INT32U SNLen,INT8U* MAC,INT32U MACLen)

Description
To read the display information of SN,MAC

Parameter
SN: [out]
Serial number on the display
SNLen: **[in]**
Serial length
MAC:[out]
The Mac address of the display

MACLen: **[in]**
Mac length

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
INT8U SN[12],MAC[7];

if(czGetSNMAC(SN,12,MAC,7) == 0)
{
}
```

Correlation function

3.1.11 (czReadDefDisplayStyle) Default display style readback

Function Prototype

INT32U czReadDefDisplayStyle(DEFAULT_SET* defaultSet)

Description

To read the default display style of the system in order to control the image and colour of the showing effect

Parameter

defaultSet : **[out]**
When the information of the default display style is retrieved, the date can be shown as below:

```
Struct
{
   UWORD ID;                    //0x55aa
   UBYTE PlayListLoc;    // play list position
   UBYTE TimePre0En;     // prepose 0
   UBYTE Ddrive;             // default disk
   UBYTE Dback_color;    // default back color
   UBYTE Dfont_color;    // default font color
   UBYTE Dhor_just;       // horizontal alignment
   UBYTE Dver_just;       // vertical alignment
   UBYTE Dline_space; // line space
   UBYTE Dfont;           //     font
   UBYTE Din_mode;     //       mode in
   UBYTE Dout_mode;    //       mode out
   UBYTE Dspeed;        //       speed
   UBYTE Dstay_time;    //    stay time
  UBYTE Dwrap;          //        word wrap
    INT32U Lstay_time;
    INT8U TimeFormat;
    INT8U rev[31];
}DEFAULT_SET
```

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
DEFAULT_SET defaultSet;
if(czReadDefDisplayStyle(&defaultSet) == 0)
{
}
```

Correlation function

3.1.12 (czReadSpecPathFileEx) Specific path file extension readback

Function Prototype

INT32U czReadSpecPathFileEx (INT8U* SpecialFile,INT8U* PCPath)

Description

To read the known path and the file name, if it is the extension of czReadSpecPathFile, it is meant to supply a gap, that is because czReadSpecPathFile can not read the file capacity is over 64M

Parameter

SpecialFile: **[in]**

The specific file and name on the display

PCPath：**[in]**

To read the storage location of the file

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Program examples

```
INT8U name[64],path[256];
name = “D:\\A\\temp.pmg”;
path = “D:\\temp.pmg”;
if(czReadSpecPathFileEx(name,path)== 0)
{
}
```

Correlation function

3.1.13 (czReadSystemInfo) System Information Read

Function Prototype

INT32U czReadSystemInfo (INT8U* Info,INT32U BufLen,INT32U* size)

Description

Read the system information, specific content of information depends on Firmware.

Parameters

Info: **[out]**

The Firmware content which has been read back

BufLen: **[in]**

The length of buffer zone

size：**[out]**

Content-length

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
INT8U Buf[2048];
INT32U size;
if(czReadSystemInfo(Buf, 2048,&size) == 0)
{
}
```

Correlation function

3.1.14 (czReadDB) Database readback

Function Prototype

INT32U czReadDB(INT8U cmdType,INT8U* sql,INT8U* Field,INT32U FiledLen，INT8U* Record,INT32U RecordLen)

Description

Read the data in the data base of the system, read the play log.

Parameters

cmdType: **[in]**

The beginning cabinet number (logic, start with 0)

1=SELECT statement.

sql：**[in]**

SQL statement

Field: **[out]**

The returned field name set, it will be divided into 3 fields when it is being returned. The first and second bytes of this field means the quantity of records, the following bytes mean the quantity of fields, other bytes means field name set, the name will be enclosed by “<>”.

FiledLen：**[in]**

Size of the Field buffer zone

Record: **[out]**

Record, each field will be enclosed by ”<>”, no more than 1200Bytes.

RecordLen：**[in]**

Size of the Record buffer zone

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
INT8U cmd;
INT8U sql[256],Field[1024],Record[4096];
cmd = 1;
sql = “select * from PlayLog limit 20 offset 1”;
 if(czReadDB(cmd,sql,Field,1024,Record,4096) == 0)
```

```
{
    printf("record number = %d, filed number =%d",Field[0]+Field[1]<<8,
    Field[3]+Field[4]<<8;
}
```

Correlation function

3.1.15 (czReadBrightInfo) Brightness information readback

Function Prototype

INT32U czReadBrightInfo (INT8U* BrightType,INT8U* Percent,INT8U* ADValue)

Description

Read the brightness information of system;

Parameters

BrightType: **[out]**

Current brightness type. 0=automatically, 1=manually, 2=scheduling

Percent：**[out]**

Brightness percentage, effective value [1-100], 100 means the highest.

ADValue: **[out]**

Current AD value, it will be effective when the brightness type is automatically.

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
INT8U BrightType,Percent,ADValue;
if(czReadBrightInfo(&BrightType,&Percent,&ADValue) == 0)
{
}
```

Correlation function

3.1.16 (czReadPlayLog) Play log readback

Function Prototype

INT32U czReadPlayLog (INT8U* PCPath);

Description

Read the play log back to the file, this command is not supported on 5800 main board, please read 5800's log by czReadDB. About file protocol, please check the JetFileII protocol document for reference.

Parameters

PCPath: **[in]**

The path and the file name of the saved files in the PC.

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
if(czReadPlayLog (“C:\\plalog.log”) == 0)
{
}
```

Correlation function

3.1.17 (czInitProperty) Init. Sign Proterty

Function Prototype

INT32U czInitProperty ();

Description

Get Property from sign, and init. In memory.

Parameters

none.

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
if(czInitProperty () == 0)
{
}
```

Correlation function

czGetProperty

3.1.18 (czGetProperty) Get property from memory

Function Prototype

INT32U czGetProperty(void *buf, INT32U bufsize, INT32U type);

Description

Get Sign’s property from memory, please call czInitProperty first.

Parameters

buf: **[in/out]**

Get Property buffer。

bufsize:**[in]**

size of buffer。

type: **[in]**

Get type, define as follow：

```
#define PROPERTY_IS_F_DRIVE_EXIST      1      //F Drive if exist
#define PROPERTY_PLAYLIST_SAVE_TO      2      //playlist Save location
#define PROPERTY_GET_DEFAULT           3      //get default settings
```

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
INT32U f_drive_exist;
czInitProperty();
if(CZ_ERROR_OK != czGetProperty(&f_drive_exist, 4,
PROPERTY_IS_F_DRIVE_EXIST))
{
  //Error
}
```

Correlation function

czInitProperty

3.1.19 (czReadWarning) Get Waring xml file

Function Prototype

INT32U czReadWarning (INT8U* pcpath)

Description

Get Sign's Waring xml file, need application to process.

Parameters

pcpath: **[in/out]**

xml file to save in pc location.

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
if(CZ_ERROR_OK != czReadWarning ((INT8U*)"f:\\WARING.XML")
{
  //Error
}
```

Correlation function

czReadWaringReport, czReadPixelErrorReport

3.1.20 (czReadWaringReport) Get Waring Report

Function Prototype

INT32U czReadWaringReport(INT32U (*cb)(ERROR_INFO_STRUCT *errorInfo));

Description

Get Sign's Waring, each Waring will call "cb" function.

Parameters

cb: **[in/out]**

Waring call back

```
// ERROR_INFO_STRUCT
typedef struct
{
  error_type type;                                    //Error
```

```
  INT8U x;
  INT8U y;
  INT8U problem[64];                    //error string
  INT8U description[256];               //error description
  INT8U time[64];                       //report time
}ERROR_INFO_STRUCT;
```

X:X position，start from 0，left to right.
Y:Y position，start from 0，up to bottom.
Top left corner of the VMS is the Original Point (0,0), to rightward, X value will increase; to downward, Y value will increase. For pixel detection, one pixel is one unit. For signal, one module is considered as one unit.
Result: Display result
Pixel failure will display like:
R,G,B Error,(R,G,B refer to color: red, green, blue，it only shows the error data, when specific color fail.)
For example, if only red LED fails, then it will display: R Error
If Red and Green fail, it will show：R,G Error
RecordDT: report time，format will be like：YYYY-MM-DD HH:mm

```
//Error type
typedef enum
{
  ERROR_MAINBOARD,
  ERROR_TILE,
  ERROR_PIXEL,
  ERROR_SIGNAL
}error_type;
```

Callback result examples:
If the VMS has pixel failures at the locations like following table shows,

X	Y	Error	DateTime	
14	68	R Error	2016-06-24 18:05	First example
9	68	G Error	2016-06-24 18:05	Second example
0	65	B Error	2016-06-24 18:05	Third example

And has some signal failure like following table shows:

Signal Error

Port	X	Y	Error	DateTime	
0	0	9	R,G,B Error	2016-06-24 18:05	Fourth example
0	1	9	R,G,B Error	2016-06-24 18:05	Fifth example

Callback function will read back for 5 times. Every PixelInfo Data will display like following tables:
The Callback parameter value of First example, which highlighted in red color at above

table (Pixel failure)

PI	0x010fcf88 {Type=ERROR_PIXEL X=0x0000000e Y=0x0044 ...}
Type	ERROR_PIXEL
X	0x0000000e
Y	0x0044
Result	0x010fcf92 "R Error"
RecordDT	0x010fcff6 "2016-06-24 18:05"

The Callback parameter value of Second example, which highlighted in red color at above table (Pixel failure)

PI	0x010fcf88 {Type=ERROR_PIXEL X=0x00000009 Y=0x0044 ...}
Type	ERROR_PIXEL
X	0x00000009
Y	0x0044
Result	0x010fcf92 "G Error"
RecordDT	0x010fcff6 "2016-06-24 18:05"

The Callback parameter value of Third example, which highlighted in red color at above table (Pixel failure)

PI	0x010fcf88 {Type=ERROR_PIXEL X=0x00000000 Y=0x0041 ...}
Type	ERROR_PIXEL
X	0x00000000
Y	0x0041
Result	0x010fcf92 "B Error"
RecordDT	0x010fcff6 "2016-06-24 18:05"

The Callback parameter value of Fourth example, which highlighted in red color at above table (Signal failure)

PI	0x010fc1bc {Type=ERROR_SIGNAL X=0x00000000 Y=0x0009 ...}
Type	ERROR_SIGNAL
X	0x00000000
Y	0x0009
Result	0x010fc1c6 "R,G,B Error"
RecordDT	0x010fc22a "2016-06-24 18:05"

The Callback parameter value of Fifth example, which highlighted in red color at above table (Signal failure)

PI	0x010fc1bc {Type=ERROR_SIGNAL X=0x00000001 Y=0x0009 ...}
Type	ERROR_SIGNAL
X	0x00000001
Y	0x0009
Result	0x010fc1c6 "R,G,B Error"
RecordDT	0x010fc22a "2016-06-24 18:05"

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
INT32U cb(ERROR_INFO_STRUCT *errorInfo)
{
      //process err
 }

if(CZ_ERROR_OK != czReadWaringReport(cb))
{
  //Error
}
```

Correlation function

czReadWaring, czReadPixelErrorReport

3.1.21 (czReadPixelErrorReport) Get sign pixcel error report

Function Prototype

INT32U czReadPixelErrorReport(INT32U (*cb)(PixelInfo *PI))

Description

Get Sign's Pixcel error report

Parameters

cb: **[in/out]**

pixcel error report call back

```
typedef struct
{
  error_type Type;              // ERROR_PIXEL，ERROR_SIGNAL
  INT32U X;
  INT16U Y;
  INT8U   Result[100];
  INT8U   RecordDT[20];
}PixelInfo;

//Error type
typedef enum
{
  ERROR_MAINBOARD,
  ERROR_TILE,
  ERROR_PIXEL,
  ERROR_SIGNAL
}error_type;
```

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
INT32U cb(PixelInfo *errorInfo)
{
      //process err
 }

if(CZ_ERROR_OK != czReadPixelErrorReport(cb))
{
  //Error
}
```

Correlation function

czReadWaringReport, czReadWaring

3.1.22 (czReadBrightInfoExt) Brightness information readback-ext

Function Prototype

INT32U czReadBrightInfoExt(czBrightInfoExt *BrightInfo);

Description

Read the brightness information of system;

Parameters

BrightInfo: **[out]**

```
typedef struct
{
      INT8U   BrightType;
      INT8U   BrightPrecent1;
      INT16U BrightSensorAD1;
      INT8U   Rev;
      INT8U   BrightPrecent2;
      INT16U BrightSensorAD2;
}czBrightInfoExt;
```

BrightType:Current brightness type. 0=automatically, 1=manually, 2=scheduling

BrightPrecent1/ BrightPrecent2：

Brightness percentage, effective value [1-100], 100 means the highest.

BrightSensorAD1/ BrightSensorAD2:

Current AD value, it will be effective when the brightness type is automatically.

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
czBrightInfoExt bri;
if(czReadBrightInfoExt(&bri) == 0)
{
}
```

Correlation function

czReadBrightInfo

3.1.23 (czReadStats) Main board status Read

Function Prototype

INT32U czReadStats (INT8U statsType,INT8U* buf,INT32U bufSize)

Description

Read the Main board status, specific content of information depends on Firmware.

Parameters

statsType: **[in]**

Read the main board status type

0:all status 1:temp inside 2:temp outside
3:humidity inside 4:humidity outside 5:gradient /tilting angle
6:wind direction 7:wind speed 8:brightness
9:door open 10:power status 11:firmware Ver
12:pixel error count

buf: **[out]**

The Firmware content which has been read back

BufSize: **[in]**

The length of buffer zone

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
INT8U doorStat = 0;
if(czReadStats(9, &doorStat,1) == 0)
{
}
```

Correlation function

3.2 Information Writer Operation

3.2.1 (czWriteSystemFile) Writing system file to LED screen

Function Prototype

INT32U czWriteSystemFile (INT8U* FileName,INT8U* PCPath)

Description

Write the files into the specified path. It should be careful when using this function to write in CONFIG.SYS file, this function should be used when that engineer is familiar with CONFIG.SYS file composition completely.

Parameters

FileName：[in]

File name of system.”CONFIG.SYS”, “SEQUENT.SYS”

PCPath: [in]

The storage location of the file saved in the PC

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//code list2.8
INT8U FileName[128],path[256];
FileName = “CONFIG.SYS”;
Path = “c:\\CONFIG.SYS”;
if(czWriteSystemFile(FileName,Path) == 0)
{
}
```

Correlation function

3.2.2 (czWriteFontFile) Writing font file to LED screen

Function Prototype

INT32U czWriteFontFile(INT8U* FontName,INT8U* PCPath);

Description

Write font file or font list into LED screen. It is not suggested to be invoked, please use sigma3000 if changing font is needed.

Parameter

FontName：[in]

The name of the font file which will be written into LED screen.

PCPath: [in]

The storage location of the font file saved in the PC

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//code list 2.8
INT8U FileName[128],path[256];
FileName = “Normal11.fnt”;
Path = “c:\\Normal11.fnt”;
if(czWriteFontFile(FileName,Path) == 0)
{
}
```

Correlation function

3.2.3 (czWriteTextFile) Writing Text File to LED screen

Function Prototype

INT32U czWriteTextFile(INT8U Drive,INT8U* TextName,INT8U* PCPath);

Description

Write Text file into the specified drive of the LED screen.

Parameters

Drive：[in]

Specified Drive

TextName：[in]

The name of the text file which will be written in.

PCPath: [in]

The storage location of the file saved in the PC.

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//code list 2.8
INT8U Letter,FileName[128],path[256];
Letter = 'D';
FileName = "Temp.NMG";
Path = "C:\\Temp.NMG";
if(czWriteTextFile(Letter,FileName,Path) == 0)
{
}
```

Correlation function

3.2.4 (czWriteStringFile) WritingString file to LED screen

Function Prototype

INT32U czWriteStringFile(INT8U Drive,INT8U* StringName,INT8U* PCPath);

Description

Write String file into the specific drive of the LED screen

Parameters

Drive：[in]

Specific Drive

StringName：[in]

The name of the String file which will be written in.

PCPath: [in]

The storage location of the file saved in the PC.

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//code list 2.8
INT8U Letter,FileName[128],path[256];
Letter = 'D';
FileName = "1";
Path = "C:\\11.txt";
if(czWriteStringFile(Letter,FileName,Path) == 0)
{
}
```

Correlation function

3.2.5 (czWritePictureFile) Writing Picture file to LED screen

Function Prototype

INT32U czWritePictureFile(INT8U Drive,INT8U* PictureName,INT8U* PCPath);

Description

Write Picture file into the specific drive of the LED screen.

Parameters

Drive：[in]

Specified Drive

PictureName：[in]

The name of the Picture file which will be written in.

PCPath: [in]

The storage location of the file saved in the PC.

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//code list 2.8
INT8U Letter,FileName[128],path[256];
Letter = 'D';
FileName = "12.bmp";
Path = "C:\\12.bmp";
if(czWritePictureFile(Letter,FileName,Path) == 0)
{
}
```

Correlation function

3.2.6 (czWriteArrPicFile) Writing ArrayPicture file (PMG file) to LED screen

Function Prototype

INT32U czWriteArrPicFile(INT8U Drive,INT8U* ArrPicName,INT8U* PCPath);

Description

Write ArrayPicture file into the specific drive of the LED screen.

Parameters

Drive：[in]

Specific Drive

ArrPicName：[in]

The name of the text file which will be written in.

PCPath: [in]

The storage location of the file saved in the PC.

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//code list 2.8
INT8U Letter,FileName[128],path[256];
Letter = 'D';
FileName = "Temp.PMG";
Path = "C:\\Temp.PMG";
if(czWriteArrPicFile(Letter,FileName,Path) == 0)
```

```
{
}
```

Correlation function

3.2.7 (czWriteSpecFile) Writing flies into the specified path

Function Prototype

INT32U czWriteSpecFile(INT8U* SpecialFile,INT8U* PCPath)

Description

Write flies into the specified path. The file must be smaller than 64MB, please use the expanding function if the file is larger than 64MB.

Parameters

SpecialFile：[in]

The specified path of the special file which will be written in.

PCPath: [in]

The storage location of the file saved in the PC.

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//code list 2.8
INT8U FileName[128],path[256];
FileName = “D:\\T\\Temp.NMG”;
Path = “D:\\ Temp.NMG”;
if(czWriteSpecFile(FileName,Path) == 0)
{
}
```

Correlation function

3.2.8 (czWriteUrgentMsg) Writing in Urgent Message

Function Prototype

INT32U czWriteUrgentMsg(INT8U StayTime, INT8U SoundSwitch,INT8U* TextMsg,INT32 MsgLen)

Description

Write in Urgent Message

Parameters

StayTime：[in]

Stay time, 0 means play permanently, the unit is second.

SoundSwitch：[in]

Sound switch, 1 = on, 0 = off.

TextMsg: [in]

Urgent message data(Text File data)，not more than 1024 bytes.

MsgLen: [in]

The length of the message

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error

code.

Examples

```
//core list 2.9
INT8U StayTime,SoundSwitch,Msg[1024];
StayTime = 0;
SoundSwitch = 1;
Msg = “Welcome” if(czWriteUrgentMsg(StayTime,SoundSwitch,Msg,sizeof(Msg)) ==
0)
{
}
```

Correlation function

3.2.9 (czWriteBrightCtrlBlock) Writing in Brightness Control Block

Function Prototype

INT32U czWriteBrightCtrlBlock(BRIGHT_CTRL* brightCtrl)

Description

Write in Brightness Control Block. No influence for the screen without brightness control.

Parameters

brightCtrl：[in]

Brightness control block structural body information.

```
typedef struct
{
  INT16U x;     //X position
  INT16U Y;     //Y position
  INT8U    Red;
  INT8U    Green;
  INT8U    Blue;
  INT16U Width;
  INT16U Height;
}BRIGHT_CTRL;
```

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//code list 2.10
BRIGHT_CTRL brightCtrl;
brightCtrl.x = 0;
brightCtrl.y = 0;
brightCtrl.Red = 0xFF;
brightCtrl.Green = 0xFF;
brightCtrl.Blue    = 0xFF;
brightCtrl.Width = 64;
brightCtrl.Height = 32;
if(czWriteBrightCtrlBlock(&brightCtrl) == 0)
{
}
```

Correlation function

3.2.10 (czWriteDefDisplayStyle) Writing Default Display Style

Function Prototype

INT32U czWriteDefDisplayStyle(DEFAULT_SET* defaultSet)

Description

Read the absolute address information of main board.

Parameters

defaultSet：[in]

Default structural body information

```
Struct
{
   UWORD ID;                    //55aa
   UWORD Rev;                   //reserve
   UBYTE Ddrive;               //default drive
   UBYTE Dback_color;      //default background color
   UBYTE Dfont_color;     //default foreground color
   UBYTE Dhor_just;          //horizontal adjust
   UBYTE Dver_just;          //vertical adjust
   UBYTE Dline_space;       //line space
   UBYTE Dfont;               //font
   UBYTE Din_mode;          //in mode
   UBYTE Dout_mode;         //out mode
   UBYTE Dspeed;             //speed
   UBYTE Dstay_time;        //stay time
  UBYTE Dwrap;               //word wrap
    INT32U Lstay_time;
   INT8U TimeFormat;
    INT8U rev[31];
}DEFAULT_SET
```

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Example

```
//code list 2.12
DEFAULT_SET defaultSet;
if(czReadDefDisplayStyle(&defaultSet) == 0)
{
    defaultSet.Ddrive = “D”;
    if(czWriteDefDisplayStyle(defaultSet) == 0)
    {
    }
}
```

Correlation function

3.2.11 (czWriteSpecFileEX) Writing File to Specific Path Extension

Function Prototype

INT32U czWriteSpecFileEX(INT8U* SpecialFile,INT8U* PCPath)

Description

Write the file into the specified path, expanding function can used into writing the file >64MB, < 2GB.

Parameters

SpecialFile：[in]

The specified path of the special file which will be written in.

PCPath: [in]

The storage location of the file saved in the PC.

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//code list 2.13
INT8U FileName[128],path[256];
FileName = “D:\\T\\Temp.NMG”;
Path = “D:\\ Temp.NMG”;
if(czWriteSpecFileEX(FileName,Path) == 0)
{
}
```

Correlation function

3.2.12 (czWriteCRCForFile) Writing CRC file to LED Screen

Function Prototype

INT32U czWriteCRCForFile(INT8U* FileName,INT32U FileSize,INT16U FileCRC)

Description

Write a file’s CRC and save it into the LED screen, software can read D:\FILELST.SYS file, this file records each file and verification in the screen, if it is the same as the verification, means that the file has been saved in the screen (no need to send).

Note: please judge that if the function list will support to open the file CRC or not, or the command will be supported by the firmware.

Parameter

FileName：[in]

Filename

FileSize：[in]

File size

FileCRC: [in]

File CRC

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//code list 2.14
INT8U File[128] = “D:\\T\\TEMP.NMG”;
INT32U FileSize = GetFileSize(File);
INT16U FileCRC    = GetFileCRC(File);
if(czWriteCRCForFile(File,FileSize,FileCRC) == 0)
{
}
```

Correlation function

3.3 Test Function

3.3.1 (czConnectTest) Connection Test

Function Prototype

INT32U czConnectTest(INT16U* FirmwareVer, INT16U* FPGAVer, INT32U* IPAddress)

Description

Test if the screen has been connected or not, if it is connected, read back some parameters of the screen

Parameters

FirmwareVer：[out]
Return to firmware version.
FPGAVer：[out]
Return to FPGA version.
IPAddress: [out]
Return to IP address of the screen.

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//code list
INT16U FirmwareVer,FPGAVer;
INT32U ip;
if(czConnectTest(&FirmwareVer,&FPGAVer,&ip)==0)
{
}
```

Correlation function

3.3.2 (czPatternTest) Display Pattern Test

Function Prototype

INT32U czPatternTest(INT32U PatternMode)

Description

Let display get into the special test pattern to test if there is any problem of the LED screen.

Parameters

PatternMode：**[in]**
Test type

2=Auto
3=full bright
4=Red
5=Green
6=Blue
7=Horizontal
8=Vertical

Return Value
If the function calls successfully, back to 0, or back to error code
Examples

```
//code list
INT8U TestType=2;
if(czPatternTest(TestType)==0)
{
}
```

Correlation function

3.3 3 (czStopTest) Stop Test

Function Prototype
INT32U czStopTest()
Description
Stop sign body testing
Parameter
Return Value
If the function calls successfully, back to 0, or back to error code
Examples

```
// code list
if(czStopTest ()==0)
{
}
```

Correlation function

3.3.4 (czGrayTest) Gray level test

Function Prototype
INT32U czCrayTest(GRAY_TEST_PARAM* grayParam)
Description
Gray test for the sign body
Parameter
grayParam: **[in]**
gray testing parameter
typedef struct
{
INT8U Type; 0=fixed, 1=gradual change
INT8U Red; 0=no Red moment, 1=with Red moment.
INT8U Green; 0=no green moment, 1=with green moment

INT8U Blue; 0=no blue moment, 1=with blue moment。
INT16U Level; 1-256 gray level
} GRAY_TEST_PARAM;

Return Value

If the function calls successfully, back to 0, or back to error code

Examples

```
// code list
GRAY_TEST_PARAM grayParam;
grayParam.Type = 1;
grayParam.Red   = 1;
grayParam.Green = 1;
grayParam.Blue = 1;
grayParam.Level = 256;
if(czCrayTest(&grayParam)==0)
{
}
```

Correlation function

3.3.5 (czColorTest) Color TestFunction Prototype

Function Prototype

INT32U czColorTest(INT8U red, INT8U green, INT8U blue)

Description

Make the sign body goes into specified color testing status.

Parameter

red: **[in]**
Red moment [0-255]
green: **[in]**
green moment [0-255]
blue: **[in]**
blue moment [0-255]

Return Value

If the function calls successfully, back to 0, or back to error code

Examples

```
// code list
INT8U r,g,b;
r=255;g=255;b=255;
if(czColorTest(r,g,b)==0)
{
}
```

Correlation function

3.3.6 (czAreaTest) Specific area test

Function Prototype

INT32U czAreaTest(AREA_STRUCT areaStruct)

Description

Make the sign body goes into the color test status

Parameter

areaStruct: **[in]**
area and color specified
typedef struct
{
INT16U Beginx;
INT16U Beginy;
INT16U Endx;
INT16U Endy;
INT16U Intensity;
INT8U Pattern;
INT8U Red;
INT8U Green;
INT8U Blue;
INT16U modWidth;
INT16U modHeight;
}AREA_STRUCT; // meaning of each parameter please refer to the JetFileII's protocol.

Return Value

If the function calls successfully, back to 0, or back to error code

Examples

```
// code list
AREA_STRUCT areaStruct;
Init struct …
if(czAreaTest(areaStruct)==0)
{
}
```

Correlation function

3.4 System Operation

3.4.1 BlackScreen

Function Prototype

INT32U czBlackScreen()

Description

Make the sign body goes into the black screen status

Parameter

Return Value

If the function calls successfully, back to 0, or back to error code

Examples

```
// code list
if(czBlackScreen()==0)
{
}
```

Correlation function

3.4.2 (czEndBlackSystem) Ending blackScreen

Function Prototype

INT32U czEndBlackScreen()

Description

end the black screen status

Parameter

Return Value

If the function calls successfully, back to 0, or back to error code

Examples

```
// code list
if(czEndBlackScreen()==0)
{
}
```

Correlation function

3.4.3 System Reset (czResetSystem)

Function Prototype

INT32U czResetSystem()

Description

Reset the screen

Parameter

Return Value

If the function calls successfully, back to 0, or back to error code

Examples

```
// code list
if(czResetSystem()==0)
{
}
```

Correlation function

czResetSystemCool

3.4.4 Power on/power off of the LED screen(czPowerOnOff)

Function Prototype

INT32U czPowerOnOff(INT8U OnOFF)

Description

Turn on/off the screen

Parameter

OnOFF： **[in]**

0=off,1=on

Return Value

If the function calls successfully, back to 0, or back to error code

Examples

```
// code list
INT8U bDisplay;
bDisplay = 1;
if(czPowerOnOff(bDisplay)==0)
{
}
```

Correlation function

3.4.5 Power state readback(cz Get Power State)

Function Prototype

INT32U czGetPowerState(INT8U* MomState,INT8U* DriverState)

Description

By reading the on/off status of the screen, we can check the on/off status of mainboard and drive board.

Parameter

MomState： **[out]**

On/off status of main board 0 = on 1= off(when testing a BlackScreen,Need Firmware support) 2=off(Turn off the screen)

DriverState: **[out]**

On/off status of drive board 1 = off, 0 = on

Return Value

If the function calls successfully, back to 0, or back to error code

Examples

```
// code list
INT8U bFalg1,bFlag2;
if(czGetPowerState(&bFalg1,&bFlag2)==0)
{
}
```

Correlation function

3.4.6 (czChangeBaudRate) Changing the Baudrate

Function Prototype

INT32U czChangeBaudRate(INT8U BaudRate)

Description

Change the display parameter dynamically. But this change would not be saved when the power is off. And only some main board support this function.

Parameter

BaudRate： **[in]**

Change the display baud rate dynamically. 0 = 115200 1 = 57600 2 = 38400 3 = 19200 4 = 9600 5 = 4800 6 = 2400 7 = 1200 8 = 600

Return Value

If the function calls successfully, back to 0, or back to error code

Examples

```
// code list
INT8U BaudRate = 0;
if(czChangeBaudRate(BaudRate)==0)
{
}
```

Correlation function

3.4.7 (czBrightAdjust) Brightness Adjustment

Function Prototype

INT32U czBrightAdjust(INT8U bright)

Description

Adjusting the brightness

Parameter

bright： **[in]**

brightness. Virtual value is [0-100] 0 = automatically 1-100 brightness level，1 is black，100 is brightest

Return Value

If the function calls successfully, back to 0, or back to error code

Examples

```
// code list
INT8U bright = 0;
if(czBrightAdjust(BaudRate)==0)
{
}
```

Correlation function

3.4.8 (cz Set Beacon) Setting amber indication light

Function Prototype

INT32U czSetBeacon(BEACON_ST* beacon)

Description

Control the status of beacon

Parameter

beacon: **[in]**

definition of the parameter of beacon, see the following list.

```
typedef struct
{
  INT8U R;
  INT8U Y;
  INT8U G;
  INT8U Mode;
  INT8U Rev[4];
```

}BEACON_ST;

PS：R,Y,G control Red,Yellow,Green on/off，0xff means off,0x01 means on, Mode control the status of yellow beacon，0x01=all beacons flesh ,0x02=left side and the right side beacons flesh,0x03=up and down beacons flesh,0x04=all on ,0x05=all off

Return Value

If the function calls successfully, back to 0, or back to error code

Examples

```
// code list
BEACON_ST   beacon;
beacon.R = 1;
beacon.G = 1;
beacon.Y = 1;
beacon.Mode = 0x03;

if(czSetBeacon(indiStatus)==0)
{
}
```

Correlation function

3.4.9 Indication light status read-back(cz Get Beacon)

Function Prototype

INT32U czGetBeacon(BEACON_ST* beacon)

Description

Get the status of beacon

Parameter

beacon: **[out]**

read the parameter of beacon. Definition could be seen in czSetBeacon function.

Return Value

If the function calls successfully, back to 0, or back to error code

Examples

```
// code list
BEACON_ST beacon;
if(czGetBeacon(&beacon)==0)
{
}
```

Correlation function

3.4.10 System Reset (czResetSystemCool)

Function Prototype

INT32U czResetSystemCool()

Description

Reset the screen(Cold)

Parameter

Return Value

If the function calls successfully, back to 0, or back to error code

Examples

```
// code list
if(czResetSystemCold()==0)
{
}
```

Correlation function
czResetSystem

3.4.11 Online/offline switch(czSwitchOnlineOffline)

Function Prototype
INT32U czSwitchOnlineOffline(INT8U OnlineOffline)
Description
Turn on/off the screen
Parameter
OnlineOffline： **[in]**
0=online,1=offline
Return Value
If the function calls successfully, back to 0, or back to error code
Examples

```
// code list
INT8U Online=0;
if(czSwitchOnlineOffline (Online)==0)
{
}
```

Correlation function

3.5 Time

3.5.1 (cz ReadLEDTime) Time readback

Function Prototype
INT32U czReadLEDTime(INT16U *Y,INT8U *M,INT8U *D,INT8U *H,INT8U *MM,INT8U *S,INT8U *W,INT8U *TZ)
Description
Read LED time
Parameter
Y: **[out]**
year
M:**[out]**
month
D:**[out]**
date
H:**[out]**
hour
MM:**[out]**
minute
S:**[out]**
second

W:**[out]**
week,0=sunday,1=monday…6=saturday
TZ:**[out]**
Time zone meaning as follows:

+0x00 time zone HH：MIN （-12）
+0x01 time zone HH：MIN （-11）
+0x02 time zone HH：MIN （-10）
+0x03 time zone HH：MIN （-9）
+0x04 time zone HH：MIN （-8）
+0x05 time zone HH：MIN （-7）
+0x06 time zone HH：MIN （-6）
+0x07 time zone HH：MIN （-5）
+0x08 time zone HH：MIN （-4）
+0x09 time zone HH：MIN （-3）
+0x0a time zone HH：MIN （-2）
+0x0b time zone HH：MIN （-1）
+0x0c time zone HH：MIN （+0）
+0x0d time zone HH：MIN （+1）
+0x0e time zone HH：MIN （+2）
+0x0f time zone HH：MIN （+3）
+0x10 time zone HH：MIN （+4）
+0x11 time zone HH：MIN （+5）
+0x12 time zone HH：MIN （+6）
+0x13 time zone HH：MIN （+7）
+0x14 time zone HH：MIN （+8）
+0x15 time zone HH：MIN （+9）
+0x16 time zone HH：MIN （+10）
+0x17 time zone HH：MIN （+11）
+0x18 time zone HH：MIN （+12）
+0x19 time zone HH：MIN （+13）
+0x1a time zone HH：MIN （-3:30）
+0x1b time zone HH：MIN （+5:30）
+0x1c time zone HH：MIN （+5:45）
+0x1d time zone HH：MIN （+6:30）
+0x1e time zone HH：MIN （+9:30）
+0x1f time zone HH：MIN （+3:30）
+0x20 time zone 的 HH：MIN （+4:30）
+0x21 time zone HH：MIN （-4:30）

Return Value

If the function calls successfully, back to 0, or back to error code

Examples

```
// code list
 INT16U Y;
 INT8U M,D,H,MM,S,W,TZ;
```

```
if(czReadLEDTime(&Y,&M,&D,&H,&MM,&S,&W,&TZ)==0)
{
}
```

Correlation function

3.5.2 (cz Ajust LED Time Ex) Time adjustment

Function Prototype

INT32U czAjustLEDTimeEx(INT16U Y,INT8U M, INT8U D,INT8U H,INT8U MM, INT8U S,INT8U W,INT8U TZ);

Description

Adjust the time

Parameter

The meaning of the parameter is the same as czReadLEDTime, please refer to czReadLEDTime function

Return Value

If the function calls successfully, back to 0, or back to error code

Examples

```
// code list
 INT16U Y=2013;
INT8U M=4,D=11,H=1,MM=1,S=0,W=4,TZ=1;
if(czAjustLEDTimeEx (Y,M,D,H,MM,S,W,TZ)==0)
{
}
```

Correlation function

3.5.3 (czSendTempHumi) Temperature and humidity notification

Function Prototype

INT32U czSendTempHumi(INT8U Humidity, INT8S Temperature)

Description

Send the temp/humi to LED main board, that is, temp/humi sensor send the parameters to main board automatically.

Parameter

Humidity: **[in]**

Temp,[0-100]。

Humidity: **[in]**

Humidity,[-128~+127]

Return Value

If the function calls successfully, back to 0, or back to error code

Examples

```
// code list
INT8U Humidity;
INT8S Temperature;
```

```
Humidity = 50;
Temperature = 30;
if(czSendTempHumi (Humidity, Temperature)==0)
{
}
```

Correlation function

3.5.4 (czWriteSpeedLimit) Writing speed limit

Function Prototype

INT32U czWriteSpeedLimit(INT16U limitSpeed,INT16U limitOffset)

Description

Write the RADAR limit values and limit off set values. And only some main board support this function.

Parameter

limitSpeed: **[in]**

limit speed values.

limitOffset: **[in]**

limit off set values

Return Value

If the function calls successfully, back to 0, or back to error code

Examples

```
// code list
INT16U limitSpeed,INT16U limitOffset;
limitSpeed = 100;
limitOffset = 0;
if(czWriteSpeedLimit(limitSpeed,limitOffset)==0)
{
}
```

Correlation function

3.6 Play Control

3.6.1 (czReplayList) Replaying file list

Function Prototype

INT32U czReplayList()

Description

Replay the play list

Parameter

Return Value

If the function calls successfully, back to 0, or back to error code

Examples

```
// code list
if(czReplayList ()==0)
{
}
```

Correlation function

3.6.2(czReplayCurrFile) Replaying the current file

Function Prototype
INT32U czReplayCurrFile ()
Description
Replay the current file
Parameter

Return Value
If the function calls successfully, back to 0, or back to error code
Examples

```
// code list
if(czReplayCurrFile ()==0)
{
}
```

Correlation function

3.6.3 (czPlayPause) Play pause

Function Prototype
INT32U czPlayPause ()
Description
Pause the play
Parameter

Return Value
If the function calls successfully, back to 0, or back to error code
Examples

```
// code list if(czPlayPause ()==0)
{
}
```

Correlation function

3.6.4 (czPlayContinue) Play continue

Function Prototype
INT32U czPlayContinue()
Description
Continue the play
Parameter

Return Value
If the function calls successfully, back to 0, or back to error code
Examples

```
// code list
if(czPlayContinue()==0)
{
}
```

Correlation function

3.6.5 (czPlayNext) Playing next file

Function Prototype

INT32U czPlayNext()

Description

Play the next file in the playlist

Parameter

Return Value

If the function calls successfully, back to 0, or back to error code

Examples

```
// code list
if(czPlayNext()==0)
{
}
```

Correlation function

3.6.6(czPlayPriority) Play priority

Function Prototype

INT32U czPlayPriority (INT8U Drive, INT8U Type,INT8U* FileName);

Description

Play one of the file as priority, and the start the playlist

Parameter

Drive: **[in]**

partition，”D” ”E” ”F”

Type: **[in]**

File type,’T’=Text File, P=Picture File, A=Array Picture file,F=movie

FileName: **[in]**

File name

Return Value

If the function calls successfully, back to 0, or back to error code

Examples

```
// code list
INT8U Letter, INT8U Type
INT8U *FileName;
Letter = ‘D’;
Type    =   ‘T’;
FileName = “Temp.NMG”;
if(czPlayPriority(Letter,Type,FileName)==0)
```

```
{
}
```

Correlation function

3.6.7 (czGetPlayingFileName) Current display file readback

Function Prototype

INT32U czGetPlayingFileName(INT8U* FileName,INT32U NameLen);

Description

To read running file name. It can read the information by the file reading order

Parameter

FileName: **[out]**

NameLen: **[in]**

FileName :Max length

Return velue

If the function calls successfully, back to 0, or back to error code

Examples

```
//code list
INT8U FileName[256];
if(czGetPlayingFileName (FileName,256)==0)
{
}
```

Correlation function

3.6.8 (czGetNextPlayFileName) Getting next play file name

Function Prototype

INT32U czGetNextPlayFileName(INT8U* FileName, INT32U NameLen);

Descrption

(czGetNextPlayFileName). And only some old main board support this function.

Parameter

FileName: **[out]**

NameLen: **[in]**

FileName: Max length

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
// code listing
INT8U FileName[256]
if(czGetNextPlayFileName(FileName,256)==0)
{
}
```

Correlation function

3.6.9 (czPlayPrevious) Play previous file

Function Prototype
INT32U czPlayPrevious()
Dscription
(czPlayPrevious)
Prameter

Return Value
If the function is successfully operated, return to 0. Otherwise, return to Error code.
Examples

```
// code listing
if(czPlayPrevious()==0)
{
}
```

Correlation function

3.6.10 (czPlayForword) Play Forward

Function Prototype
INT32U czPlayForword()
Dscription
(czPlayForword)
Prameter

Return Value
If the function is successfully operated, return to 0. Otherwise, return to Error code.
Examples

```
// code listing
if(czPlayForword()==0)
{
}
```

Correlation function

3.6.11 (czPlayBack) Play back

Function Prototype
INT32U czPlayBack()
Dscription
(czPlayBack)
Parameter

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//code listing
if(czPlayBack()==0)
{
}
```

Correlation function

3.6.12 (czPlayNextFrame) Play next frame

Function Prototype INT32U czPlayNextFrame()

Description

Play Next Frame. And only some old main board support this function.

Parameter

Return Value

If the function is successfully operated, return to 0. Otherwise, return to

Examples

```
// code listing
if(czPlayNextFrame()==0)
{
}
```

Correlation function

3.6.13(czSoundCtrl) Sound control

Function Prototype

INT32U czSoundCtrl(INT8U SoundSwitch,INT8U mode,INT8U time);

Description

sound ctrl,supported by a mainboard with buzzer.

Prameter

switch: **[in]**

Buzzer mode, set up to turn it on, 0= it rings when receiving file. 1= it rings when switching files.

time: **[in]**

Buzzer ring time, from '0' – '9' (Second), '0' = not ring '1' = ringing for 1 second。

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
// code listing
INT8U switch,INT8U mode,INT8U time;
switch = 1;
mode       = 0;
time       = 1;
if(czSoundCtrl(switch,mode,time)==0)
{
}
```

Correlation function

3.6.14 (czBeginTiming) Beginning counting down or down counting the time

Function Prototype

INT32U czBeginTiming(INT8U day, INT8U hour,INT8U min,INT8U sec)

Description

(czBeginTiming, adjustable). And only some old main board support this function.

Prameter

day: **[in]**

hour: **[in]**

min: **[in]**

sec: **[in]**

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//code listing
INT8U day=0,hour=0,min=5,sec=0;
if(czBeginTiming(day,hour,min,sec)==0)
{
}
```

Correlation function

3.6.15 (czStopTiming) Stopping counting down or down counting the time

Function Prototype

INT32U czStopTiming()

Dscription

(czStopTiming). And only some old main board support this function.

Parameter

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
// code listing
if(czStopTiming()==0)
{
}
```

Correlation function

3.6.16 (czPauseTiming) Pauzing counting the time

Function Prototype

INT32U czPauseTiming()

Description

(czPauseTiming). And only some old main board support this function.

Parameter

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//code listing
if(czPauseTiming()==0)
{
}
```

Correlation function

3.6.17 (czContinueTiming) Continuing time counting

Function Prototype

INT32U czContinueTiming()

Description

(czContinueTiming). And only some old main board support this function.

Parameter

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
// code listing
if(czContinueTiming()==0)
{
}
```

Correlation function

6.18 (czReadCurScreenshot) Screen Readback

Function Prototype

INT32U czReadCurScreenshot(INT8U *pcpath)

Description

Get the screen shot of the real-time picture.

Parameter

pcPath: **[in]**

Save screenshots to a local path

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
// code listing
INT8U pcpath[256];
if(czReadCurScreenshot(pcpath)==0)
{
}
```

Correlation function

6.19 (czPlaySoundFile) Play an audio file

Function Prototype

INT32U czPlaySoundFile(SoundST soundST,INT8U* soundPath)

Description

Play the specified audio file.

Parameter

soundST: **[in]**

play audio file parameter, see the following Struct list.

```
typedef struct
{
   INT8U BuzzerSwitch;
   INT8U PlayTimes;
   INT8U Rev[6];
}SoundST;
```

PS：BuzzerSwitch:Switch(0:Stop, 1:Play)，playTimes:Play Times(1:Once，2:twice，0xFF:loop)

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
// code listing
SoundST soundST;
Memset(&soundST,0,sizeof(soundST));
soundST.BuzzerSwitch = 1;
soundST.PlayTimes = 1;
if(czPlaySoundFile(soundST,(INT8U*)”c://Test.mp3”)==0)
{
}
```

Correlation function

czAdjustVolume

6.20 (czAdjustVolume)Volume Adjustment

Function Prototype

INT32U czAdjustVolume(INT8U soundVolume)

Description

Adjust the play volume for the audio system.

Parameter

soundVolume: **[in]**

Sound Volume.

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
// code listing
if(czAdjustVolume(80)==0)
{
}
```

Correlation function

czPlaySoundFile

3.7 File Control

3.7.1 (czFormatDrive) Formatting zone

Function Prototype

INT32U czFormatDrive(INT8U Drive);

Description

”C”, ”D”, ”E”, ”F”。 (czFormatDrive) ”C”, ”D”, ”E”, ”F”

Parameter

Drive: **[in]**

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//code listing
INT8U Letter = “D”;
if(czFormatLetter(Letter)==0)
{
}
```

Correlation function

3.7.2 (czCreateDir) Creating directory

Function Prototype

INT32U czCreateDir(INT8U* Dir)

Description

(czCreate Dir)

Parameter

Dir: **[in]**

To creat a file path. Eg, to create folder”TEST” under disk C: “C:\TEST\”
Note: can not create a multistage folder, ending with “\”+ NULL.

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//code listing
INT8U DirPath = “C:\TEST\”;
if(czCreateDir(DirPath)==0)
{
```

```
}
```

Correlation function

3.7.3 (czRename) Renaming

Function Prototype

INT32U czRename(INT8U* SourceName, INT8U* DestName)

Description

Rename the specified file.

Parameter

SourceName: **[in]**

DestName: **[in]**

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
// code listing
INT8U *SourceName, *DestName;
SourceName = “D:\\T\\Temp.NMG”;
DestName = “D:\\T\\1.NMG”;
if(czRename(SourceName,DestName)==0)
{
}
```

Correlation function

3.7.4 (czMove) Moving files

Function Prototype

INT32U czMove(INT8U* SourceName, INT8U* DestName)

Description

Move the file only in same disk.

Parameter

SourceName: **[in]**

DestName: **[in]**

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
// code listing
INT8U SourceName[256], INT8U[256];
SourceName = “D:\\T\\Temp.NMG”;
DestName = “F:\\T\\1.NMG”;
if(czMove(SourceName,DestName)==0)
{
}
```

Correlation function

3.7.5 (czDelete) Deleting files

Function Prototype INT32U czDelete(INT8U* FileName)
Description
Delete the specified file.
Parameter
FileName: **[in]**
The Path for deleting file, and the file name.

Return Value If the function is successfully operated, return to 0. Otherwise, return to Error code.
Examples

```
// code listing
INT8U *FileName;
FileName = “D:\\T\\Temp.NMG”;
if(czDelete(FileName)==0)
{
}
```

Correlation function

3.7.6 (czDelTextFiles) Deleting a text file in a specific zone

Function Prototype
INT32U czDelTextFiles(INT8U Drive);
Description
Delete Text file in specified drive
Parameter
Drive: **[in]**
”D”, ”E”, ”F”
Return Value
If the function is successfully operated, return to 0. Otherwise, return to Error code.
Examples

```
//code listing
INT8U Letter;
Letter = “D”;
if(czDelTextFiles(Letter)==0)
{
}
```

Correlation function

3.7.7 (czDelStringFInLetter) Deleting a string file in a specific zone

Function Prototype
INT32U czDelStringFile(INT8U Drive);
Description
Delete string file in specified drive.
Parameter

Drive: **[in]**
”D”, ”E”, ”F”

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
// code listing
INT8U Letter;
Letter = “D”;
if(czDelStringFile(Letter)==0)
{
}
```

Correlation function

3.7.8 (czDelPictureFiles) Deleting a picture file in a specific zone

Function Prototype

INT32U czDelPictureFiles(INT8U Drive);

Description

Delete picture file in specified drive.

Parameter

Drive: **[in]**
”D”, ”E”, ”F”

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
// code listing
INT8U Letter;
Letter = “D”;
if(czDelPictureFiles (Letter)==0)
{
}
```

Correlation function

3.7.9 (czDelArrpicFiles) Deleting an array picture file in a specific zone

Function Prototype

INT32U czDelArrpicFiles(INT8U Drive);

Description

Delete ArrayPicture file in specified drive.

Parameter

Drive: **[in]**
”D”, ”E”, ”F”

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
// code listing
INT8U Letter;
Letter = “D”;
if(czDelArrpicFiles (Letter)==0)
{
}
```

Correlation function

3.7.10 (czGetDirFile) Getting file information in folder/directory

Function Prototype

INT32U czGetDirFile(INT8U* path,INT8U* Num,DIRECTORY_ENTRY_STRUCT* dirEntry,INT32U size);

Description

Get all the directory from the folder.

Parameter

path: **[in]**

This parameter indicates the path to get the file. Ex, “C:\TEST\”: the file in TEST folder under disk C.

Num: **[out]**

Numbers of files returned to the directory.

dirEntry: **[out]**

The returned file directory entry list.

size: **[in]**

- The max read back directory number.

```
typedef struct
{
   UBYTE badir_name[11];          //file name
   UBYTE bdir_attr;               //file attributes
   UBYTE bdir_rev;                // reserve
   UBYTE bcrt_time_tecth;         // created time 1, second.
   UWORD wcrt_time;                    // created time 2,min, hour.
   UWORD wcrt_date;                    // created date
   UWORD wlast_acc_time;          // the last access time
   UWORD wfst_clus_hi;            //
   UWORD wwrt_time;                    // the last modification time
   UWORD wwrt_date;                    // the last modification date.
   UWORD wfst_clus_lo;
   ULONG dwfile_size;             // file size
}DIRECTORY_ENTRY_STRUCT;
```

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//code listing
INT32U Num;
DIRECTORY_ENTRY_STRUCT entry[10];
if(czGetDirFile(“D:\\T\\”,&Num, entry,100)==0)
{
}
```

Correlation function

3.7.11 (czGetDirFileEX) Getting directory file extension

Function Prototype

INT32U czGetDirFileEx(INT8U* path,INT16U StarNo,INT16U RNum,INT32U* Num,
DIRECTORY_ENTRY_STRUCT* dirEntry, INT32U size);

Description

Get the specified directory from the folder, mainly used to read files from large size folder.

Parameter

path: **[in]**

This parameter indicates the path to get the file. Ex, “C:\TEST\”: the file in TEST folder under disk C.

StartNo: **[in]**

start the directory

RNum: **[in]**

read the numbers of directory

Num: **[out]**

Return to numbers of read back files directory.

dirEntry: **[out]**

The returned file directory entry list

size:**[in]**

The max read back directory number.

```
typedef struct
{
   UBYTE badir_name[11];          //file name
   UBYTE bdir_attr;               //file attributes
   UBYTE bdir_rev;                //reserve
  UBYTE bcrt_time_tecth;          //
        UWORD wcrt_time;                  //
   UWORD wcrt_date;                  // created date
   UWORD wlast_acc_time;          // the last access time
   UWORD wfst_clus_hi;            //
   UWORD wwrt_time;                  // the last modification time
   UWORD wwrt_date;                  // the last modification date
   UWORD wfst_clus_lo;
   ULONG dwfile_size;             // file size
}DIRECTORY_ENTRY_STRUCT;
```

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code

3.7.12 (czGetDriveInfo) Getting drive information

Function Prototype

INT32U czGetDriveInfo(INT8U Drive, INT32U *totalSize, INT32U *remainSize, INT8U *driveName);

Description
To get the information of thespecific disk
Parameter
Drive: **[in]**
”D”, ”E”, ”F”
totalSize: **[out]**
total size of the return disk ,unit Bytes。
remainSize: **[out]**
the remaing capacity of the return disk,unit Bytes。
driveName: **[out]**

Return Value
If the function is successfully operated, return to 0. Otherwise, return to Error code.
Examples

```
//code listing
INT32U totalSize,remainSize;
INT8U   driveName[64];
memset(driveName,0,sizeof(driveName));
if(czGetDriveInfo(‘D’,& totalSize, &remainSize, driveName )==0)
{
}
```

Correlation function

3.7.13 (czIsFileExist) Checking the existence of a specific file

Function Prototype
INT32U czIsFileExist(INT8U* FileName)
Description
To check the specific file is existing or not
Parameter
FileName: **[in]**
The path of the specific file and the file name

Return Value
If the function is successfully operated, return to 0. Otherwise, return to Error code.
Examples

```
//code listing
if(czIsFileExist((INT8U *)“D:\\T\\Temp.NMG”)==0)
{
}
```

Correlation function

3.7.14 (czClearAllPlayFile) Clearing all play files

Function Prototype
INT32U czClearAllPlayFile(INT8U Operation,INT8U* Process)
Description
To check the specific file is existing or not

Parameter
Operation: **[in]**

0 is to clear ,1 is to check theschedue
Process: **[out]**
Clearing schedule

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//code listing
INT8U Operation ,Process;
Operation = 0;
if(czClearAllPlayFile(Operation,&Process)==0)
{
}
```

Correlation function

3.7.15 (czGetDirLongFileEx) Getting directory file

Function Prototype

INT32U czGetDirLongFileEx(INT8U* path, INT8U* pcpath)

Description

Get the specified directory from the folder, mainly used to read files from large size folder.

Parameter

path: **[in]**
This parameter indicates the path to get the file. Ex, “C:\TEST\”: the file in TEST folder under disk C.

pcpath: **[in]**
save to pc file

file format is:
DIRECTORY_ENTRY_STRUCTExHead+ DIRECTORY_ENTRY_STRUCTEx*N

```
typedef struct
{
    UWORD Flag;             //”DL”flag   0x444C
    UWORD HeadLen;          //filehead len
    ULONG Count;            // DIRECTORY_ENTRY_STRUCTEx numbers
    ULONG   FileSize;       //file size
    ULONG   Recv;           //rev
} DIRECTORY_ENTRY_STRUCTExHead;

typedef struct   //DirectoryEntryStructure
{
    UBYTE badir_name[255];  //file name
    UBYTE bdir_attr;        //attrib
    UBYTE bdir_rev;         //rev
    UBYTE bcrt_time_tecth;  //create time
    UWORD wcrt_time;        //create time
    UWORD wcrt_date;        //create date
    UWORD wlast_acc_time; //access time
    UWORD wfst_clus_hi;     //
```

```
        UWORD wwrt_time;        //modify time
        UWORD wwrt_date;        //modify date
        UWORD wfst_clus_lo;
        ULONG dwfile_size;      //file size
    }DIRECTORY_ENTRY_STRUCTEx;
```

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//code listing
if(czGetDirLongFileEx ((INT8U*)"D:\\T\\",(INT8U*)"F:\\DIR.TXT")==0)
{
}
```

Correlation function

3.7.16 (czLstLongFolderCB) Getting directory Call back

Function Prototype

```
INT32U czLstLongFolderCB( INT8U* signpath,
                          INT32S (*cb)(DIRECTORY_ENTRY_STRUCTEx *dirEntry, void
*cbObj), void *cbObj);
```

Description

Get the specified directory from the folder,call back.

Parameter

path: **[in]**

This parameter indicates the path to get the file. Ex, "C:\TEST\": the file in TEST folder under disk C.

cb: **[in/out]**

call back function

cbObj: **[in/out]**

user define argument.

```
typedef struct    //DirectoryEntryStructure
 {
        UBYTE badir_name[255];      //file name
        UBYTE bdir_attr;            //attrib
        UBYTE bdir_rev;             //rev
        UBYTE bcrt_time_tecth;      //create time
        UWORD wcrt_time;            //create time
        UWORD wcrt_date;            //create date
        UWORD wlast_acc_time;       //access time
        UWORD wfst_clus_hi;         //
        UWORD wwrt_time;            //modify time
        UWORD wwrt_date;            //modify date
        UWORD wfst_clus_lo;
        ULONG dwfile_size;          //file size
    }DIRECTORY_ENTRY_STRUCTEx;
```

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//code listing
INT32S cb(DIRECTORY_ENTRY_STRUCTEx *dirEntry, void *cbObj)
{

}

if(czLstLongFolderCB ((INT8U*)“D:\\T\\”, cb, NULL)==0)
{

}
```

Correlation function

3.8 On LineTicker

3.8.1 (czBeginUnlimited) Beginning OnLine Ticker

Function Prototype

INT32U czTickerStart(INT8U dir,INT8U speed, INT8U dataFormat);

Description

Turn on play mode, under this mode, all of the content delivered to the screen in the form of pictures. And only some old main board support this function.

Parameter

dir: **[in]**

Direction 0x00 means move left, 0x01 means move right.

speed: **[in]**

Fastest speed 0x00 … Slowest speed 0x06 (7 levels)

DataFormat: **[in]**

Represents date format will be delivered later, please refer to JetFileII Protocol.

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
// Code list
INT8U direction,speed, DataFormat;
direction = 0x00;
speed = 0x00;
DataFormat = 1;
if(czTickerStart (direction,speed, DataFormat)==0)
{
}
```

Correlation function

3.8.2 (czTickerStop) Terminating ticker display

Function Prototype
INT32U czTickerStop()
Description
czTickerStop. And only some old main board support this function.
Parameter

Return Value
If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//Code list
 if(czTickerStop()==0)
{
}
```

Correlation function

3.8.3 (czGetBufferStatus) Getting Buffer Status

Function Prototype
INT32U czGetBufferStatus(INT16U* statusCode);
Description
Query the receive buffer status of czTicker. And only some old main board support this function.

Parameter
StatusCode: **[out]**

Return status code.
Definition as follow:

Status code table

BufPost	Buffer zone 1	Buffer zone 2	Free buffer zone	Status code
0	OLD_BUF	CUR_BUF	DISPLAY_BUF	0x8301
1	CUR_BUF	DISPLAY_BUF	OLD_BUF	0x8302
2	DISPLAY_BUF	OLD_BUF	CUR_BUF	0x8303

Abstract meaning of status code

①,②, ③represent buffer zones respectively：1_BUF, 2_BUF, 3_BUF

0x8301: ①,②buffer is showing, ③buffer does't showing. ①is moving in，②is moving out;

0x8302: ②,③buffer is showing, ①buffer does't showing. ②is moving in，③is

moving out;

0x8303: ③,①buffer is showing，②buffer does't showing。③is moving in ，①is moving out.

If system is not under status of unlimited czTicker, then return 0x8305

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

Program example

```
//Code list
INT16U StatusCode;
if(czCheckBufStatus(&StatusCode)==0)
{
}
```

Correlation function

3.8.4 (czUploadBuffer) Upload Butter

Date download instruction

Function Prototype

INT32U czUploadBuffer(INT8U* Data, INT32U dataLen)

Description

Download display date to screen buffer. And only some old main board support this function.

Parameter

Data: **[in]**

Date of 1 frame.

dataLen**[in]**

Size of date

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
// Code list
INT8U Data[12*1024];
//Set Data..
if(czUpdateBuf(Data, 12*1024)==0)
{
}
```

Correlation function

3.9 (OFF Line Ticker) Stopping off-line ticker

3.9.1 (czOffLineTickerStart) Starting off-line ticker

Function Prototype

INT32U czOffLineTickerStart (INT8U mode, INT8U dir,INT8U speed, INT8U sec);

Description

Turn on Ticker, all of the files in the playlist will be displayed

Parameter

mode: **[in]**
Mode,0x00 = keep moving,0x01=take a pause for each shift of screen
dir: **[in]**
Direction 0x00 means moving left, 0x01 means moving right.
speed: **[in]**
Fastest speed 0x00 … Slowest speed 0x06 (7 levels)
sec: **[in]**
When under mode 1, represents duration of time, in second.

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//Code list
INT8U direction,speed, sec;
direction = 0x00;
speed = 0x00;
sec = 1;
if(czOffLineTickerStart (1,direction,speed, sec)==0)
{
}
```

Correlation function

3.9.2 (czOffLineTickerStop) Stopping off-line ticker

Function Prototype

INT32U czOffLineTickerStop()

Description

Parameter

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//Code list
if(czOffLineTickerStop ()==0)
{
}
```

Correlation function

3.10 Log In

3.10.1 (czLogin) Log in

Function Prototype

INT32U czLogin (INT8U* UserName, INT8U* Password)

Description

Log on to the main control system

Parameter

UserName: **[in]**

Password: **[in]**

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//Code list
INT8U UserName[16],Password[8];
memset(UserName,0,16);
memset(Password,0,8);
memcpy(UserName,"admin",strlen("admin"));
memcpy(Password,"1234",strlen("1234"));
if(czLogin (UserName,Passsword)==0)
{
}
```

Correlation function

3.10.2 (czLogout) Log out

Function Prototype

INT32U czLogout()

Description

Logout

Parameter

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//Code list
if(czLogout()==0)
{
}
```

Correlation function

3.10.3 (czChangePSW) Changing the password

Function Prototype

INT32U czChangePSW(INT8U* UserName, INT8U* Password, INT8U* NewPassword)

Description

Log in to the system.
Parameter
UserName: **[in]**

Password: **[in]**

NewPassword: **[in]**

Return Value
If the function is successfully operated, return to 0. Otherwise, return to Error code.
Examples

```
//Code list
INT8U UserName[15],Password[7];
UserName = “admin”;
Password = “1234”;
NewPassword = “abcd”;
if(czChangePSW(UserName,Passsword,NewPassword)==0)
{
}
```

Correlation function
Related function

3.11 VPU3400 Operation

3.11.1 (czVPUSelChannel) choosing video input channel

Function Prototype
INT32U czVPUSelChannel(INT8U ch)

Description

Used for changing video source
Parameter
ch: **[in]**
Channel selected
0: YPbPr
1: S_Video
2: CVBS1
3: CVBS2
4: CVBS3
5: VGA
6: SDI
7: HDMI
8: Test

Return Value
If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//Code list
INT8U channel = 1;
if(czVPUSelChannel (channel)==0)
{
}
```

Correlation function

3.11.2 (czVPUSetMode) Setting display mode

Function Prototype

INT32U czVPUSetMode(INT8U mode, INT8U alpha)

Description

Setting overlay mode of DVI and video window.

Parameter

Mode: **[in]**

0: DVI only
1: video only
2: DVI on top
3: Video on top
4: Mixed
5: Title generator

alpha: **[in]**

Setting transparency of mixed mode [0-100]

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//Code list
INT8U Mode,Transparency;
Mode = 4; Transparency = 50;
if(czVPUSetMode (Mode,Transparency)==0)
{
}
```

Correlation function

3.11.3 (czVPUSetVideoRatio)Setting video ratio

Function Prototype

INT32U czVPUSetVideoRatio(INT8U ratio);

Description

Setting ratio of the video window

Parameter

Type: **[in]**

0: fill in the window
1: 16:10
2: 16:9
3: 5:4
4: 4:3

5: 3:2

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//Code list
INT8U Type;
Type = 2;
if(czVPUSetVideoRatio(Type)==0)
{
}
```

Correlation function

3.11.4 (czVPUSetDVIWin) Setting DVI window

Function Prototype

INT32U czVPUSetDVIWin(INT16U Type, INT16U value)

Description

Setting position and size of DVI window. Call this function 4 times for a complete setting.

Parameter

Type: **[in]**

0: Setting X coordinate of window
1: Setting Y coordinate of window
2: Setting width of window
3: Setting height of window

value: **[in]**

Value have have been set

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//Code list
INT16U Type,value;
Type = 2; value = 0;
if(czVPUSetDVIWin(Type,value)==0)
{
}
```

Correlation function

3.11.5 (czVPUSetVideoWin) Setting video window

Function Prototype

INT32U czVPUSetVideoWin(INT16U Type, INT16U value)

Description

Setting position and size of video window. Call this function 4 times for a complete setting.

Parameter

Type: **[in]**
0: Setting X coordinate of window
1: Setting Y coordinate of window
2: Setting width of window
3: Setting height of window
value: **[in]**
Setting value

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//Code list
INT16U Type,value;
Type = 2; value = 0;
if(czVPUSetVideoWin(Type,value)==0)
{
}
```

Correlation function

3.11.6 (czVPUSetVideoArg) Setting video parameter

Function Prototype

INT32U czVPUSetVideoArg(INT16U type, INT16U value);

Description

Setting brightness, contrast, saturation parameter of video.

Parameter

Type: **[in]**
0: setting brightness. effective value -50--50
1: setting contrast. Effective value 10--100
2: setting tone.effective value -90--90
3: setting saturation. Effective value 0--100
4： sharpness. Effective value 0--15
5： setting left and right deviation. Effective value 0--512
6： setting above and below deviation. Effective value 0--512
7： Restore to the default value.
value: **[in]**
Value have been set

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//Code list
INT16U Type,value;
Type = 2; value = 0;
if(czVPUSetVideoArg (Type,value)==0)
```

```
{
}
```

Correlation function

3.11.7 (czVPUGetSignalStatus) Getting signal status

Function Prototype

INT32U czVPUGetSignalStatus(INT16U *videoW, INT16U *vedioH, INT16U *dviW, INT16U *dviH);

Description

Get status of DVI input signal and video input signal.

Parameter

videoW: **[out]**

Read signal width of video.

vedioH: **[out]**

Read signal height of video.

dviW: **[out]**

Read signal width of DVI.

dviH: **[out]**

Read signal height of DVI.

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//Code list
INT16U VideoWidth, VideoHeight, DVIWidth,DVIHeight;
if(czVPUGetSignalStatus (&VideoWidth, &VideoHeight,
&DVIWidth,&DVIHeight)==0)
{
}
```

Correlation function

3.11.8 (czVPUType) Setting VPU system type

Function Prototype

INT32U czVPUType(INT8U type)

Description

Setting type of system, selected to be worked as master or slaver.

Parameter

Type: **[in]**

0： Set as master

1： Set as slaver

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//Code list
INT8U Type = 1;
if(czVPUType(Type)==0)
{
}
```

Correlation function

3.11.9 (czVPUSlaverStartLine) Setting VPU slaver start line

Function Prototype

INT32U czVPUSlaverStartLine(INT16U StartLine);

Description

Setting startline of slaver

Parameter

StartLine: **[in]**

Positon of Slaver's start line

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//Code list
INT16U StartLine = 512;
if(czVPUSlaverStartLine(StartLine)==0)
{
}
```

Correlation function

3.11.10 (czVPUSetColorTemp) Setting color temperature

Function Prototype

INT32U czVPUSetColorTemp(INT8U type,INT8U R, INT8U G, INT8U B);

Description

Setting value of colortemp

Parameter

Type: **[in]**

Type of colortemp 0：6500 1：9300 2:user-defined.

R: **[in]**

Value of red when user defining colortemp

G: **[in]**

Value of green when user defining colortemp

B: **[in]**

Value of blue when user defining colortemp

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//Code list
INT8U Type=0, Red=255, Green=220, Blue=220;
if(czVPUSetColorTemp(Type, Red, Green, Blue)==0)
{
}
```

Correlation function

3.11.11(czVPUSetBright) Setting screen brightness

Function Prototype

INT32U czVPUSetBright(INT8U type,INT8U manualValue, INT8U autoMin, INT8U autoMax);

Description

Setting brightness of screen

Parameter

Type: **[in]**

0： adjusting brightness manually

1： adjusting brightness automatically

manualValue: **[in]**

setting value of brightness manually

autoMin: **[in]**

setting minimum value of brightness automatically

autoMax: **[in]**

setting maximum value of brightness automatically

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//Code list
INT8U Type, Bright, Max=100, Min=10;
Type = 0;
Bright = 50;
if(czVPUSetBright(Type, Bright, Max, Min)==0)
{
}
```

Correlation function

3.11.12 (czVPUGetBright) Getting brightness status

Function Prototype

INT32U czVPUGetBright(INT8U *type,INT8U *manualValue, INT8U *autoMin, INT8U *autoMax);

Description

Get information of brightness

Parameter

Type: **[out]**

0: adjusting brightness manually 1:adjusting brightness automatically

Bright: **[out]**

Brightness value set manually

Max: **[out]**

Minimum value of brightness adjusted automatically

Min: **[out]**

Maximum value of brightness adjusted automatically

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//code listing
INT8U Type, Bright, Max, Min;
if(czVPUGetBright(&Type,&Bright,&Max,&Min)==0)
{
}
```

Correlation function

3.11.13 (czVPUSetGamma) Setting Gamma value of the screen

Function Prototype

INT32U czVPUSetGamma(INT8U index);

Description

To set correction value of display Gamma

Parameter

GammaValue: **[in]**

Gamma value. 0-3 are valid values.

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//code listing
INT8U GammaValue = 0;
if(czSetGamma (GammaValue)==0)
```

```
{
}
```

Correlation function

3.11.14 (czVPUSetLDUNums) Setting number of LDU

Function Prototype

INT32U czVPUSetLDUNums(INT8U LDUNums);

Description

To set LDU number.

Parameter

LDUNum: **[in]**

LDU numer. 1-8 are valid values.

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//code listing
INT8U LDUNum = 2;
if(czVPUSetLDUNums(LDUNum)==0)
{
}
```

Correlation function

3.11.15 (czVPUSetLDUPos)Setting LDU coordinates

Function Prototype

INT32U czVPUSetLDUPos(INT8U LDUID, INT16U x, INT16U y);

Description

To set LDU coordinates.

Parameter

LDUNo: **[in]**

Please set LDU sequence number. 1-8 are valid values.

x: **[in]**

x coordinate

y: **[in]**

y coordinate

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//code listing
INT8U LDUNo = 2;
INT16U x ＝ 512, y = 0;
if(czVPUSetLDUPos(LDUNo,x,y)==0)
{
}
```

Correlation function

3.11.16 (czVPUGetInfo) Getting version information of VPU

Function Prototype

INT32U czVPUGetInfo(VPUVerInfo *info);

Description

To obtain VPU version information.

Parameter

info: **[out]**

The obtained LDU version information structure is as followed.

```
typedef struct
{
    INT16U CPUVer;
    INT16U FPGA1Ver;
    INT16U FPGA2Ver;
    INT16U FPGA3Ver;
    INT8U  SN[12];
}LDU_VERINFO;
```

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//code listing
LDU_VERINFO lduInfo
if(czVPUGetInfo(&lduInfo)==0)
{
}
```

Correlation function

3.11.17 (czVPUSetPixcelMode) Setting pixel mode

Function Prototype

INT32U czVPUSetPixcelMode(INT8U mode);

Description

To set operating mode.

Parameter

mode: **[in]**

0:Real pixel

1:Virtual pixel

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//code listing
INT8U PixModel = 0;
if(czVPUSetPixcelMode(PixModel)==0)
{
}
```

Correlation function

3.12 Display Control

This order is suitable for displays with multiple tiles. Each tile is equipped with control boards. This order is used to read tile state.

3.12.1 Control card status readback

Function Prototype

INT32U czImposaGetTileStat(INT8U tileAddr, INT32U *rtCode);

Description

Request of reading state of specific tiles

Parameter

tileAddr: **[in]**

Tile address. 1-494 are valid values.

rtCode: **[Out**

Return code,which is needed to check the read state.

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//code listiing
INT32U retCode = 0;
if(czImposaGetTileStat(1, & retCode)==0)
{
}
```

Correlation function

czImposaGetResult

3.12.2 Operation information of the control card readback

Function PrototypeFunction Prototype

INT32U czImposaGetResult(INT32U rtCode, INT8U *result, INT32U size, INT16U *resultCode);

Description

To read the operating information of last request. rtCode is the read-back result of czImposaGetTileStat, which is saved in “result”.

Parameter

rtCode: **[in]**

czImposaGetTileStat

The obtained value of czImposaGetTileStat function.

result : **[Out]**

To save results. If it’s successfully read, the result is as followed.

Address offset	Data name	Data size (Byte)	Description
0x00000	Tile width	1	
0x00001	Tile height	1	
0x00002	Tile address	1	Address range 1-0xef. At most 239 tiles can be connected.

0x00003	CPU version	2	(BCD Code) 0x1000 means Ver1.0
0x00005	FPGA version	2	(BCD Code) 0x1000 means Ver1.0
0x00007	Current tile brightness	1	
0x00008	Current maximum display brightness	1	
0x00009	Tile temperature	1	0x7f means positive temperature, ranging from 0-127. 0x80 --- 0x99 means negative temperature, ranging from -25-0. 0xff means temperature sensor failure.
0x0000a	Fan's Start-up Temperature	1	0---127
0x0000b	Fan's Start-up state	1	0x00 means Power on. 0x80 means Power off.
0x0000c	Fan's current 0	2	0 --- 1023 The original value is returned by sampling device.
0x0000e	Fan's current 1	2	0 --- 1023 The original value is returned by sampling device.
0x00010	Power voltage 0	2	0 --- 1023 The original value is returned by sampling device.
0x00012	Power voltage 1	2	0 --- 1023 The original value is returned by sampling device.
0x00014	Tile state	2	DO Showing 1 means a FPGA load failure. D1 Showing 1 means an Ethernet connection failure for the tile.
0x00016	Ethernet state	2	
0x00018	Communication receiving times	2	Each time the control board receives a data package, this variable is incremented by 1.
0x0001a	r Data and verification	2	
0x0001c	Driving board data and verification	2	
0x0001e	FPGA data and verification	2	

0x00020	Configuration data and verification	2	
0x00022	X coordinate	2	Obtained by reading FPGA
0x00024	y coordinate	2	Obtained by reading FPGA
0x00026	If the data is valid	1	To recognize if tile state is valid, which is controlled bu HUB instead od control board.
0x00027	Channel of the tile	1	0-15 (Controlled by HUB)
0x00028	Reserved	4	
0x0002c	Reserved	1	
0x0002b	Reserved	1	
0x0002E	Tile's working hour at high temperature	4	Real working hour = the value x 5 (minute)
0x0002D	Frame frequency	1	
0x0002F	Cluster's checking state	1	
0x00030	SN serial number	8	SN serial number
0x0038	Red index	1	
0x0039	Type of driving board	1	
0x003a	Group address	1	
0x003b	Reserved	1	

size: **[in]**
To tell result size.
resultCode:[Out]
Codes of tile returning

0x0C00 : Requested order is in the buffer.
0x0C 01 : Task is successfully completed.
0x0C 02 : Sequence number error
0x0C 03 : Under processing
0x0C 04 : Failure of endpoints' bus
0x0C 05 : Retry times is greater than pre-set value.
0x0C 06 : General error
0x0C 07 : Order ID error
0x0C 08 : Broadcast cannot successfully write data.

Real data can only be returned when the operating result is 0x01, which means task is successfully finished. Any other results occurred, data cannot be returned.

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//read tile address 1 information
INT32U retCode = 0;
if(czImposaGetTileStat(1, & retCode)==0)
{
    INT8U result[1024];
```

```
INT16U resultCode;

while(czImposaGetResult(retCode, result,1024,&resultCode) ==0)
{
      if (resultCode == 0x0C01) //read ok
      {
            //result[0] =tile width
            //result[1] =tile height
            //…
            Break;
      }
       else if(resultCode == 0x0C00 || resultCode == 0x0C03)
      {
           //sleep(200) //wait some time
      }
      else
       {
            Break;
       }
}

}
```

Correlation function

czImposaGetTileStat

3.13 Car Park Display Control

This order is suitable for Car Park displays with multiple zones control.

3.13.1 Divide Zone

Function Prototype

INT32U czDivideZone(INT8U pageID, INT8U flag, ZoneSetHead zoneSetHead, ZoneSet* zoneSetArrs);

Description

Set multiple zones in Car park Display.

Parameter

pageID: **[in]**

page ID.

flag: **[in]**

0: modify zone or add zone

1: reset all zone

zoneSetHead: **[in]**

zone divide head, see the following Struct list.

```
typedef struct
{
   INT8U BGColor_R;          //the frame background red color
   INT8U BGColor_G;          //the frame background green color
```

```
    INT8U BGColor_B;        //the frame background blue color
    INT8U ZoneNum;          //zone number
    INT16U PageStayTime;    //page stay time: unit: 10ms，0 for unlimited display
    INT8U PortraitMode;     //display mode(0: horizontal(default)，1:vertical)
    INT8U Rev;              //reserve
}ZoneSetHead;
```

PS: BGColor_R, BGColor_G, BGColor_B:indicate the frame background color，ZoneNum:zone number，PageStayTime:page stay time: unit: 10ms，0 for unlimited display，PortraitMode:display mode(0: horizontal(default)，1:vertical)

zoneSetArrs: **[in]**
Zone Arrs to Set, see the following Struct list.

```
typedef struct
{
    INT8U ZoneID;           //zone ID
    INT8U Rev[3];           //reserve
    INT16U XPos;            //zone X position
    INT16U YPos;            //zone Y position
    INT16U ZoneWidth;       //zone width
    INT16U ZoneHeight;      //zone height
}ZoneSet;
```

PS: ZoneID: zone ID，XPos:zone X position，YPos:zone Y position，ZoneWidth:zone width，ZoneHeight:zone height

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code(refer to 3.13.10).

Examples

```
//code listiing
ZoneSetHead zoneSetHead;
ZoneSet zoneSetEntry[3];
memset(&zoneSetHead, 0, sizeof(ZoneSetHead));
memset(zoneSetEntry, 0, sizeof(ZoneSe)*3);
zoneSetHead.ZoneNum = 1;
zoneSetEntry[0].ZoneID = 1;
zoneSetEntry[0].XPos = 0;
zoneSetEntry[0].YPos = 0;
zoneSetEntry[0].ZoneWidth = 0x40;
zoneSetEntry[0].ZoneHeight = 0x10;

if(czDivideZone(1, 1, zoneSetHead, zoneSetEntry)==0)
{
}
```

Correlation function

czGetZone

3.13.2 Get Zone

Function Prototype

INT32U czGetZone(ZoneSetHead *zoneSetHead, ZoneSet *zoneSetArrs, INT8U maxEntryCount);

Description

Get multiple zones information in Car park Display.

Parameter

zoneSetHead: **[out]**
Return the zoneSet Head.
zoneSetArrs: **[out]**
Return Array Of Multiple Zones.
MaxEntryCount: **[in]**
Read back the maximum number of zone.

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code(refer to 3.13.10).

Examples

```
//code listiing
ZoneSetHead zoneSetHead;
ZoneSet zoneSetEntry[3];
memset(&zoneSetHead, 0, sizeof(ZoneSetHead));
memset(zoneSetEntry, 0, sizeof(ZoneSe)*3);
if(czGetZone(&zoneSetHead, zoneSetEntry,3)==0)
{
}
```

Correlation function

czDivideZone

3.13.3 Display Content with property

Function Prototype

INT32U czSetDisplay(INT8U pageID, INT8U zoneNum, INT8U setMode, ZoneDisplaySet parameter, INT8U* content);

Description

Display Content with property.

Parameter

pageID: **[in]**
page ID.
zoneNum: **[in]**
number of zone
setMode: **[in]**
1: save as template
parameter: **[in]**
content property, see the following Struct list.
typedef struct

```
{
  INT16U ZoneSize;        //zone pixel count(that is ZoneWidth*ZoneHeight)
  INT8U ZoneID;           //zone ID
  INT8U ZoneType;         //zone type,refer to PS
  INT8U CodeType;         //char code（0:ASCII， 1:Unicode）
  INT8U InMode;           //in mode（0:jump out<default>，refer to PS）
  INT8U OutMode;          //out mode（0:jump out<default>，refer to PS）
  INT8U Align;            //Horizontal and vertical alignment,refer to PS
  INT8U FGColor_R;        //foreground red color of the content,default 0xFF
  INT8U FGColor_G;        //foreground green color of the content,default 0x00
  INT8U FGColor_B;        //foreground blue color of the content,default 0x00
  INT8U BGColor_R;        //backround red color of the content,default 0x00
  INT8U BGColor_G;        //backround green color of the content,default 0x00
  INT8U BGColor_B;        //backround blue color of the content,default 0x00
  INT8U AutoLine;         //word wrap（0:NO<default>，1:yes）
  INT8U AutoWidth;        //fixed width font,refer to PS
  INT16U Speed;           //mode speed（n pixels per 10ms，default 1）
  INT8U StayTime;         //stay time，refer to PS
  INT8U Times;            //play tiems（0：display an infinite loop，default 0）
  INT8U FontStyle;        //font style,refer to PS
  INT8U LineSpace;        //line Space(default 1)
  INT8U CloumnSpace;      //column Space(default 1)
  INT8U Rev;              //reserve
  INT16U BlinkOnTime;     //Blink On Time(unit:10ms,default 50）
  INT16U BlinkOffTime;    //Blink Off Time(unit:10ms,default 50）
  INT16U ContentSize;     //content data size
  INT8U Rev2[2];          //reserve
}ZoneDisplaySet;
```

PS：ZoneSize:zone pixel count(that is ZoneWidth*ZoneHeight), ZoneID:zone ID，ZoneType:zone type（0：Text，1：picture，2: special char，because to special control characters in content used in conjunction,so ususally 0），CodeType:char code（0:ASCII，1:Unicode），InMode:in mode（0:jump out<default>，1:left move，2: right move，3:up move，4:down move，5:left scroll，6:right scroll，7:up scroll，8:down scroll），OutMode:out mode（0:jump out<default>，1:left move，2: right move，3:up move，4:down move，5:left scroll，6:right scroll，7:up scroll，8:down scroll。Ps:if you need a continuous scrolling,set retention time required to 0 ），Align:Horizontal and vertical alignment,default center（[7:4]Horizontal<0:center，1:align up，2:align down> ，[3:0]vertical<0:center，1:align left，2:align rigth>），FGColor_R,FGColor_G,FGColor_B:foreground color of the content(default red)， BGColor_R,BGColor_G,BGColor_B:background color of the content(default black),AutoLine:word wrap（0:NO<default>，1:yes），AutoWidth:fixed width font（0：font width same 1：font width different<default>），Speed:mode speed（n pixels per 10ms，default 1）,StayTIme:stay time（unit second，0：display an infinite loop，default 1），Times:play tiems（0：display an infinite loop，default 0），FontStyle:font style（refer to APIDemo，default 0x32<that is Normal14，8 pixel width，14 pixel height>），LineSpace:line

Space(default 1),CloumnSpace:column Space(default 1)，Rev:reserve data，BlinkOnTime:Blink On Time(unit:10ms,default 50），BlinkOffTime:Blink Off Time(unit:10ms,default 50)，ContentSize:content data size, Rev2:reserve data

content: **[in]**
displays the contents of the data.

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code(refer to 3.13.10).

Examples

```
//code listiing
ZoneDisplaySet zoneDisplayEntry;
    memset(&zoneDisplayEntry, 0, sizeof(ZoneDisplaySet));
    zoneDisplayEntry.ZoneSize = zoneSetEntry[0].ZoneWidth *
zoneSetEntry[0].ZoneHeight;
    zoneDisplayEntry.ZoneID = zoneSetEntry[0].ZoneID;
    zoneDisplayEntry.ZoneType = 0;
    zoneDisplayEntry.CodeType = 0;
    zoneDisplayEntry.InMode = 3;
    zoneDisplayEntry.OutMode = 4;
    zoneDisplayEntry.Align = 0;
    zoneDisplayEntry.FGColor_R = 0xFF;
    zoneDisplayEntry.FGColor_G = 0xFF;
    zoneDisplayEntry.FGColor_B = 0xFF;
    zoneDisplayEntry.BGColor_R = 0;
    zoneDisplayEntry.BGColor_G = 0;
    zoneDisplayEntry.BGColor_B = 0;
    zoneDisplayEntry.AutoLine = 0;
    zoneDisplayEntry.AutoWidth = 1;
    zoneDisplayEntry.Speed = 5;
    zoneDisplayEntry.StayTime = 2;
    zoneDisplayEntry.Times = 0;
    zoneDisplayEntry.FontStyle = CZ_FONT_EN_14x8;
    zoneDisplayEntry.LineSpace = 1;
    zoneDisplayEntry.CloumnSpace = 1;
    zoneDisplayEntry.BlinkOnTime = 0x0A;
    zoneDisplayEntry.BlinkOffTime = 0x0A;
    content = "Welcome";
    zoneDisplayEntry.ContentSize = content.length();

  if(czSetDisplay(1, 1, 0, zoneDisplayEntry, (INT8U*)content.c_str()))==0)
  {
  }
```

Correlation function

czSetContentDisplay

3.13.4 Display Content without property

Function Prototype

INT32U czSetContentDisplay(INT8U pageID, INT8U zoneNum, ZoneDisplayContentSet parameter, INT8U* content);

Description

Display content without property.

Parameter

pageID: **[in]**
pageID
zoneNum: **[in]**
number of zone

parameter: **[in]**
content data, see the following Struct list.

```
typedef struct
{
    INT8U ZoneSize;          //zone pixel count（that is ZoneWidth*ZoneHeight）
    INT8U ZoneID;            //zone ID
    INT16U CodeType;         //char code（0:ASCII， 1:Unicode）
    INT16U ZoneType;         //zone type,refer to PS
    INT16U Rev;              //reserve
    INT16U ContentSize;      //content data size
}ZoneDisplayContentSet;
```

PS：ZoneSize:zone pixel count（that is ZoneWidth*ZoneHeight）, ZoneID:zone ID, CodeType:char code（0:ASCII， 1:Unicode）,ZoneType:zone type（0：Text，1：picture，2: special char，because to special control characters in content used in conjunction,so ususally 0），Rev:reserve data，ContentSize:content data size

content: **[in]**
displays the contents of the data.

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code(refer to 3.13.10).

Examples

```
//code listiing
ZoneSetHead zoneSetHead;
ZoneSet zoneSetEntry[3];
memset(&zoneSetHead, 0, sizeof(ZoneSetHead));
memset(zoneSetEntry, 0, sizeof(ZoneSe)*3);
zoneSetHead.ZoneNum = 1;
zoneSetEntry[0].ZoneID = 1;
zoneSetEntry[0].XPos = 0;
```

```
    zoneSetEntry[0].YPos = 0;
    zoneSetEntry[0].ZoneWidth = 0x40;
    zoneSetEntry[0].ZoneHeight = 0x10;

     ZoneDisplayContentSet zoneDisplayContentEntry;
     memset(&zoneDisplayContentEntry, 0, sizeof(ZoneDisplayContentSet));
     zoneDisplayContentEntry.ZoneSize = zoneSetEntry[0].ZoneWidth *
zoneSetEntry[0].ZoneHeight;
     zoneDisplayContentEntry.ZoneID = zoneSetEntry[0].ZoneID;
     content = "Hello";
     zoneDisplayContentEntry.ContentSize = content.length();

    if(czSetContentDisplay(1, 1, zoneDisplayContentEntry,
    (INT8U*)content.c_str()))==0)
    {
    }
```

Correlation function

czSetDisplay

3.13.5 Enable/Disable MulitZone

Function Prototype

INT32U czSetEnableMulitZone(INT8U enabled);

Description

Enable or Disable the Mulit Zone.

Parameter

enabled: **[out]**

enabled flag.

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
    //code listiing
    if(czSetEnableMulitZone(1)==0)
    {
    }
```

Correlation function

czGetMulitZoneSetting

3.13.6 Set the display of pages

Function Prototype

INT32U czSetPageCount(INT8U pageCount);

Description

Set Mulit Zone the display of pages.

Parameter

pageCount: **[out]**

set the total number of pages.

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code(refer to 3.13.10).

Examples

```
//code listiing
if(czSetPageCount(1)==0)
{
}
```

Correlation function

czGetMulitZoneSetting

3.13.7 Divide Special Zone

Function Prototype

INT32U czDivideSpeZone(INT8U pageID, SpeZoneSet speZoneSet, SpeZoneXY* speZoneXYArrs);

Description

Divide Special Zone.

Parameter

pageID: **[in]**

the page ID to be set.

speZoneSet: **[in]**

Special zone divide head, see the following Struct list.

```
typedef struct
{
  INT8U SpeZoneID;        //the special zone ID
  INT8U PicID;            //the picture ID
  INT8U FreshPattern;     //flashing pattern
  INT8U Rev;              //reserve
  INT16U FlashHZ;         //flashing frequency
  INT16U ZoneNum;         //total number of zone
  INT16U ZoneWidth;       //zone width
  INT16U ZoneHeight;      //zone height
}SpeZoneSet;
```

PS: SpeZoneID:the special zone ID, PicID:the picture ID, FreshPattern:flashing pattern, FlashHZ: flashing frequency, ZoneNum: total number of zone, ZoneWidth: Zone width, ZoneHeight: zone height.

speZoneXYArrs: **[in]**

Specail Zone XY Arrs to Set, see the following Struct list.

```
typedef struct
{
  INT16U ZoneX;           // Zone X position
  INT16U ZoneY;           // Zone Y position
```

}SpeZoneXY;
PS: ZoneX: Zone X position, ZoneY: Zone Y position.

Return Value
If the function is successfully operated, return to 0. Otherwise, return to Error code(refer to 3.13.10).

Examples

```
//code listiing
SpeZoneSet speZoneSet;
SpeZoneXY speZoneXYArrs[3];
memset(&speZoneSet, 0, sizeof(SpeZoneSet));
memset(speZoneXYArrs, 0, sizeof(SpeZoneXY)*3);
if(czDivideSpeZone(1, speZoneSet, SpeZoneXY)==0)
{
}
```

Correlation function
czDivideZone

3.13.8 Get the Enabled and Page Count

Function Prototype
INT32U czGetMulitZoneSetting(INT8U* enabled, INT8U* pageCount);

Description

Get the Enabled of Mulit Zone and page count.

Parameter
enabled: **[out]**
Return the MultiZone enabled flag.
pageCount: **[out]**
Return the total number of pages.

Return Value
If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//code listiing
INT8U enabled,pageCount;
if(czGetMulitZoneSetting(&enabled, &pageCount)==0)
{
}
```

Correlation function
czSetEnableMulitZone, czSetPageCount

3.13.9 Display special char

For special objects，using special codes to represent. You can add it in the content.

code	Special object description
Time Code	
0x0800	HH:MM (24H)

0x0801	HH:MM:SS (24H)
0x0802	HH:MM (12H)
0x0803	HH:MM:SS (12H)
0x0804	HH:MM X(A/P)M (12H)
0x0805	HH:MM:SS X(A/P)M (12H)
0x0806	HH (24H)
0x0807	HH (12H)
0x0808	MM
0x0809	SS
0x080A	X(A/P)M
0x080B	GTM+X:00 (Time zone)
Date code	
0x0810	month/day/year (number)
0x0811	day/montrh/year (number)
0x0812	month.day.year (number)
0x0813	year-montrh-day (number)
0x0814	year (number)
0x0815	month (number)
0x0816	montrh (char)
0x0817	montrh (ASA)
0x0818	day (number)
0x0819	week (number)
0x081A	week (char)
0x081B	week (ASA)
Sensor code	
0x0820	temperature(celsius)
0x0821	temperature (fahrenheit)
0x0822	humidity
0x0823	Radar speed(KPH)
0x0824	Radar speed(MPH)
0x0825	Limit speed(KPH)
Special code	
0x0830	Font color，followed by 3 bytes RGB，exapmple: 0x080x30 0xFF0x000x00 ，For red 0x080x30 0x00xFF0x00，For green 0x080x20 0x000x000xFF，For blue
0x0831	Switch font，followed by 1 bytes font index，example 0x080x31 0x31，number 0x31 font 0x080x31 0x32，number 0x32 font
0x0832	Blink control code，Must be doubled，for loading flash content。

	Aim at CZ9270，content use the character, all flashing at the time of arrival.
0x0833	Picture insert code， followed by 1 byte indicates that the picture number，example 0x080x33 0x01，number 1 picture 0x080x33 0x02，number 2 picture The picture library path： F：\P\x.bmp Default picture： Picture size: 16X16 Number 1 picture：Up Green arrow Number 2 picture：Down Green arrow Number 3 picture：left green arrow Number 4 picture：right green arrow Number 5 picture：left up green arrow Number 6 picture：right up green arrow Number 7 picture：left down green arrow Number 8 picture：right down green arrow Number 9 picture：red fork Number 10 picture：user defined ….

3.13.10 MultiZone ErrorCode special description

Error Code	Description
0x36E1	The PageID is invalid!(Maybe the PageID exceed the actual PageCount , or the PageID is 0).
0x36E2	The PageID is valid,but The page has not been configured!
0x36E3	The PageCount must be less then 13 !
0x36E4	The ZoneID is invalid!(Maybe the ZoneID exceed the actual ZoneCount, Or the ZoneID is 0).
0x36E5	The ZoneID is valid,but the zone has not been configured!
0x36E6	The PageCount is wrong! and must be less then 21!

3.14 PlayList Control

This order is suitable for PlayList control.

3.14.1 (czPLInit) Initialize the playslit

Function Prototype
INT32U czWriteSystemFile (INT8U* workPath,INT8U* playListName)
Description
Open the playlist and load it, and if it doesn't exist,create an empty one,This function is called at the very beginning.
Parameters
workPath：[in]
Playlist working directory,plasylists and play files are placed here.
playListName: [in]
play list name

Return Value
If the function is successfully operated, return to 0. Otherwise, return to Error code.
Examples

```
//code list
INT8U workPath[128],playListName[256];
workPath = ".\\";
playListName = "DEMO.LST";
if(czPLInit(workPath,playListName) == 0)
{
}
```

Correlation function
czLoadSYSFromXML, czPLSpeSendToLED, czReadSpePlayListIndex, czPlaySpePlaylist

3.14.2 (czLoadSYSFromXML) Load PlayList File

Function Prototype
INT32U czLoadSYSFromXML (char* fileName)
Description
Load a playlist file in XML format.
Parameters
FileName：[in]
PlayList File Name

Return Value
If the function is successfully operated, return to 0. Otherwise, return to Error code.
Examples

```
//code list
INT8U playListName[256];
playListName = "DEMO.LST";
if(czLoadSYSFromXML(playListName) == 0)
{
}
```

Correlation function

czPLInit, czPLSpeSendToLED, czReadSpePlayListIndex, czPlaySpePlaylist

3.14.3 (czPLSpeSendToLED) Send a predefined playlist

Function Prototype
INT32U czPLSpeSendToLED(INT8U playListIndex,INT8U isSendChangedFiles)
Description
Send a predefined playlist to the display, and if the file content is updated and needs to be sent, send the file first.

Parameters
playListIndex：[in]
Pre-playlist index range：[1--255]
isSendChangedFiles: [in]
whether to send the updated content file,0: do not send, 1:send

Return Value
If the function is successfully operated, return to 0. Otherwise, return to Error code.
Examples

```
//code list
INT8U workPath[128],playListName[256];
workPath = ".\\";
playListName = "DEMO.LST";
czPLInit(workpath, playListName);
if(czPLSpeSendToLED(1,1) == 0)
{
}
```

Correlation function
czLoadSYSFromXML, czPLInit, czReadSpePlayListIndex, czPlaySpePlaylist

3.14.4 (czReadSpePlayListIndex) Get the index of the predefined playlist currently playing

Function Prototype
INT32U czReadSpePlayListIndex(INT8U* playListIndex)
Description
Write the files into the specified path. It should be careful when using this function to write in CONFIG.SYS file, this function should be used when that engineer is familiar with CONFIG.SYS file composition completely.

Parameters
playListIndex：[in]
Playlist index

Return Value
If the function is successfully operated, return to 0. Otherwise, return to Error code.
Examples

```
//code list
INT8U curPLIndex;
if(czReadSpePlayListIndex(&curPLIndex) == 0)
{
}
```

Correlation function

czLoadSYSFromXML, czPLSpeSendToLED, czPLInit, czPlaySpePlaylist

3.14.5(czPlaySpePlaylist) Specify to play a predefined playlist

Function Prototype

INT32U czPlaySpePlaylist(INT8U playListIndex)

Description

Specify to play a predefined playlist.

Parameters

playListIndex：[in]

playlist index

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//code list
if(czPlaySpePlaylist(1) == 0)
{
}
```

Correlation function

czLoadSYSFromXML, czPLSpeSendToLED, czReadSpePlayListIndex, czPLInit

3.15 Pix Check Operation

This order is mainly user for Pixel Check of operate.

3.15.1 (czBeginPixCheck) Beginning to pix check

Function Prototype

INT32U czBeginPixCheck ()

Description

Beginnint to pixel check,This function is called at the very beginning.

Parameters

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//code list
if(czBeginPixCheck() == 0)
```

```
{
}
```

Correlation function

czPixProgress, czReadPixResult, czReadSignalResult, czPxlChk

3.15.2 (czPixProgress) Query progress of pixel check

Function Prototype

INT32U czPixProgress ()

Description

Query progress of pixel check.

Parameters

Progress：**[out]**

Progress value: Virtual value is [0-100], 100 is finsh.

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//code list
INT8U Progress;
if(czPixProgress(&Progress) == 0)
{
}
```

Correlation function

czBeginPixCheck, czReadPixResult, czReadSignalResult, czPxlChk

3.15.3 (czReadPixResult) Get the pix result

Function Prototype

INT32U czReadPixResult (INT8U* PCPath)

Description

Get the pixel result of pixel check.

Parameters

PCPath：[in]

Save to pc path.

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
//code list
if(czReadPixResult(“D:\\pxl.csv”) == 0)
{
}
```

Correlation function

czBeginPixCheck, czczPixProgress, czReadSignalResult, czPxlChk

3.15.4 (czReadSignalResult) Get the signal result

Function Prototype
INT32U czReadSignalResult (INT8U* PCPath)
Description
Get the signal result of pixel check.
Parameters
PCPath：[in]
Save to pc path.

Return Value
If the function is successfully operated, return to 0. Otherwise, return to Error code.
Examples

```
//code list
if(czReadSignalResult("D:\\signal.csv") == 0)
{
}
```

Correlation function
czBeginPixCheck, czPixProgress, czReadPixResult, czPxlChk

3.15.5 (czPxlCheck) Block pix check

Function Prototype
INT32U czPxlCheck (INT8U* PCPxlPath, INT8U* PCSignalPath)
Description
Block pix check,you can initialize the interface update_status callback is in progress.
Please take the example in API Demo for reference.
Parameters
PCPxlPath：[in]
Save pix result to pc path.
PCSignalPath: [in]
Save signal result to pc path.

Return Value
If the function is successfully operated, return to 0. Otherwise, return to Error code.
Examples

```
//code list
if(czPxlCheck("D:\\pxl.csv","D:\\signal.csv") == 0)
{
}
```

Correlation function
czBeginPixCheck, czPixProgress, czReadPixResult, czReadSignalResult

4. Easy API Interface Definition

Easy API is used to encapsulate basic operation interfaces, which can be applied to general applications. Currently Easy API supports only internet operation.

4.1 Showing text information in the LED screen

Function Prototype

INT32U czShowMsg(char* msg, INT32U msg_size, INT32U font,INT32U color,INT32U mode,
INT32U stay_time_sec, char* sign_ip,INT32U sign_port,INT32U is_store_ram=0,
INT32U is_send_playlist=1);

Description

To send one message to display on the screen

Parameter

msg: **[in]**
Messages to be displayed.
msg_size：**[in]**
Size of displaying messages.
font :**[in]**
Font. Please take the example in API Demo for reference.
color :**[in]**
Color. Please refer to color define. If the least byte is defined as '/', it means user-define color and is aligned into "RGB/" color.
mode :**[in]**
Mode. Please refer to mode define.
stay_time_sec :**[in]**
Stay time,0-9999 second.
sign_ip :**[in]**
Display's IP address.
sign_port :**[in]**
Display's port
is_store_ram :**[in]**
To check if setting is saved in RAM. 1= means it's saved. 0 means it's saved in D or F disk.
is_send_playlist :**[in]**
1= send playlist after update play contents, 0 =just update play contents.

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
if(czShowMsg("show demo msg", 13, CZ_FONT_EN_14x8, CZ_COLOR_AMBER ,
                    CZ_MODE_STATIC, 0, TEST_SIGN_IP, 9520,1,1)== 0)
{
    //OK
}
```

Correlation function

czShowPic

4.2 Showing picture information in the LED screen

Function Prototype

INT32U czShowPic(char* bitmap, INT32U bmp_size, INT32U mode, INT32U stay_time_sec,
char* sign_ip,INT32U sign_port,INT32U is_store_ram=0);

Description

To send one bmp picture to display on the screen.

Parameter

bitmap:**[in]**
bmp picture.
bmp_size:**[in]**
Size of the information
mode:**[in]**
Mode. Please refer to mode define.
stay_time_sec:**[in]**
Stay time,0-9999 second.
sign_ip:**[in]**
Display's IP address.
sign_port:**[in]**
Display's ports.
is_store_ram:**[in]**
To check if setting is saved in RAM. 1 means it's saved. 0 means it's saved in Disk D or F.

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
FILE * fp = fopen(".\\demo.bmp", "rb");
if(fp)
{
    long size = get_file_size(fp);
    char *buf = (char*)malloc(size);
    if(buf)
    {
        fread(buf, 1, size, fp);
        czShowPic(buf, size, CZ_MODE_RAND,3, TEST_SIGN_IP, 9520);
    }
    free(buf);
    fclose(fp);
}
```

Correlation function

czShowMsg

4.3.Showing files in the LED display (czShowFiles)

Function Prototype

INT32U czShowFiles(char* files[], INT32U numfiles, INT32U mode, INT32U stay_time_sec,
char* sign_ip,INT32U sign_port,INT32U is_store_ram=0);

Description

To control the display to recurrently run a group of files, whose format can be bmp, pmg, qst or flw. The files will be saved in Disk F if the delivery is successfully operated. Otherwise, they will be saved in Disk D.

Parameter

files :**[in]**
Files' path on the local computer,which is with a two-dimension array structure.
numfiles :**[in]**
Number of files, that is to say, the length of one dimension.
mode :**[in]**

Mode. Please refer to mode define. This function is only valid for bmp files. For other file formats control, please refer to the Chapter of File Format in JetFileII.
stay_time_sec :**[in]**
Stay time, 0-9999 second. This function is only valid for bmp files.
sign_ip :**[in]**
Display's IP address.
sign_port :**[in]**
Display ports.
is_store_ram :**[in]**
To check if setting is saved in RAM. 1 means it's saved. 0 means it's saved in Disk D or F.

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
char *file[3] =
{
    {".\\1.bmp"},
    {".\\2.bmp"},
    {".\\3.bmp"}
};
INT32U RT = czShowFiles((char**)file, 3, CZ_MODE_MOVEUP, 2,TEST_SIGN_IP, 9520);
```

Correlation function

czShowMsg

4.4.Read file From the LED display (czEasyReadFile)

Function Prototype

INT32U czEasyReadFile(char* pc_file_path, char* sign_file_path,char* sign_ip,INT32U sign_port)

Description

Read file from the LED display

Parameter

pc_file_path :**[in]**
Files' path on the local computer
sign_file_path :**[in]**
Files' path on the sign.
sign_ip :**[in]**
Display's IP address.
sign_port :**[in]**
Display ports.

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
INT32U RT = czEasyReadFile ((char*)"E:\\1", (char*)"C:\\TEST.TXT", (char*)"169.254.10.49", 9520);
```

Correlation function

czEasyWriteFile, czReadSpecPathFile

4.5.Write file To LED display (czEasyWriteFile)

Function Prototype

INT32U czEasyWriteFile(char* pc_file_path, char* sign_file_path,char* sign_ip,INT32U sign_port)

Description

Write file To the LED display

Parameter

pc_file_path :**[in]**
Files' path on the local computer
sign_file_path :**[in]**
Files' path on the sign.
sign_ip :**[in]**
Display's IP address.
sign_port :**[in]**
Display ports.

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
INT32U RT = czEasyWriteFile ((char*)"E:\\1", (char*)"C:\\TEST.TXT", (char*)"169.254.10.49", 9520);
```

Correlation function

czEasyReadFile, czWriteSpecFile

4.6. make nmg file

Function Prototype

INT32U czShowMsgToNmg(char* msg, INT32U msg_size, char* nmg_path_name,INT32U font,
INT32U color, INT32U mode, INT32U stay_time_sec)
INT32U czShowBmpToNmg(char* bmp_path_name, char* nmg_path_name, INT32U mode, INT32U stay_time_sec)

Description

Convert message/bmp to nmg file format.

Parameter

msg :**[in]**
message
msg_size :**[in]**
message size
nmg_path_name:**[in]**
nmg file to save
font :**[in]**
font
color :**[in]**
color
mode :**[in]**
display mode
stay_time_sec:**[in]**
stye time, seconds.

bmp_path_name:**[in]**
bmp file path and name

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
INT32U RT = czShowMsgToNmg("Demo show",9,nmgfile[2],CZ_FONT_EN_14x8, CZ_COLOR_AMBER , CZ_MODE_STATIC,6);
```

Correlation function

czShowBmpToNmg, czShowMsgToNmg

4.7 Showing text information in the LED screen(czShowMsgEx support Serial)

Function Prototype

INT32U czShowMsgEx(char* msg, INT32U msg_size, INT32U font,INT32U color,INT32U mode,
INT32U stay_time_sec, char* sign_ip,INT32U sign_port,INT32U is_store_ram=0,
INT32U is_send_playlist=1,
conn_type type=COMM_UDP,char* comStr="COM3",INT32U

baudrate=9600,INT16U dstAddr=0x0101);

Description

To send one message to display on the screen

Parameter

msg: **[in]**

Messages to be displayed.

msg_size：**[in]**

Size of displaying messages.

font :**[in]**

Font. Please take the example in API Demo for reference.

color :**[in]**

Color. Please refer to color define. If the least byte is defined as '/', it means user-define color and is aligned into "RGB/" color.

mode :**[in]**

Mode. Please refer to mode define.

stay_time_sec :**[in]**

Stay time,0-9999 second.

sign_ip :**[in]**

Display's IP address.

sign_port :**[in]**

Display's port

is_store_ram :**[in]**

To check if setting is saved in RAM. 1= means it's saved. 0 means it's saved in D or F disk.

is_send_playlist :**[in]**

1= send playlist after update play contents, 0 =just update play contents.

type :**[in]**

Type of the communication and definition can be seen as below:

```
typedef enum
  {
    COMM_UDP,
    COMM_TCP,
   COMM_COM
 }conn_type;
```

comStr :**[in]**

Serial port communication by 'com' string, eg.COM1

baudrate :**[in]**

Serial communication with Baudrate, eg. 115200

dstAddr :**[in]**

GGUU address of the LED display, Unit Addr.is in high, Group Addr is on the is in low

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
if(czShowMsgsEx((char**)file,  3,  CZ_MODE_MOVEUP,  2,TEST_SIGN_IP,
                0,1,COMM_COM,TEST_SIGN_COM,TEST_SIGN_COM_BAUDRATE,TEST_
                N_COM_GGUU)== 0)
{
    //OK
}
```

Correlation function

czShowPicEx

4.8 Showing picture information in the LED screen(czShowPicEx support Serial)

Function Prototype

INT32U czShowPicEx(char* bitmap, INT32U bmp_size, INT32U mode, INT32U stay_time_sec,
char* sign_ip,INT32U sign_port,INT32U is_store_ram=0,
conn_type type=COMM_UDP,char* comStr="COM3",INT32U baudrate=9600,INT16U dstAddr=0x0101);

Description

To send one bmp picture to display on the screen.

Parameter

bitmap:**[in]**
bmp picture.
bmp_size:**[in]**
Size of the information
mode:**[in]**
Mode. Please refer to mode define.
stay_time_sec:**[in]**
Stay time,0-9999 second.
sign_ip:**[in]**
Display's IP address.
sign_port:**[in]**
Display's ports.
is_store_ram:**[in]**
To check if setting is saved in RAM. 1 means it's saved. 0 means it's saved in Disk D or F.
type :**[in]**
Type of the communication and definition can be seen as below:

```
typedef enum
  {
    COMM_UDP,
    COMM_TCP,
   COMM_COM
}conn_type;
```

comStr :**[in]**
Serial port communication by 'com' string, eg.COM1
baudrate :**[in]**
Serial communication with Baudrate, eg. 115200

dstAddr :**[in]**
GGUU address of the LED display, Unit Addr.is in high, Group Addr is on the is in low

Return Value
If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
FILE * fp = fopen(".\\demo.bmp", "rb");
if(fp)
{
    long size = get_file_size(fp);
    char *buf = (char*)malloc(size);
    if(buf)
    {
        fread(buf, 1, size, fp);
            czShowPicEx(buf, size, CZ_MODE_RAND,3, TEST_SIGN_IP, 9520,0
           ,COMM_COM,TEST_SIGN_COM,TEST_SIGN_COM_BAUDRATE,TEST_SIGN_
COM_GGUU);
    }
    free(buf);
    fclose(fp);
}
```

Correlation function
czShowMsgEx

4.9.Showing files in the LED display (czShowFilesEx support Serial)

Function Prototype
INT32U czShowFilesEx(char* files[], INT32U numfiles, INT32U mode, INT32U stay_time_sec,
char* sign_ip,INT32U sign_port,INT32U is_store_ram=0,
conn_type type=COMM_UDP,char* comStr="COM3",INT32U baudrate=9600,INT16U dstAddr=0x0101);

Description

To control the display to recurrently run a group of files, whose format can be bmp, pmg, qst or flw. The files will be saved in Disk F if the delivery is successfully operated. Otherwise, they will be saved in Disk D.

Parameter
files :**[in]**
Files' path on the local computer,which is with a two-dimension array structure.
numfiles :**[in]**
Number of files, that is to say, the length of one dimension.
mode :**[in]**

Mode. Please refer to mode define. This function is only valid for bmp files. For other file formats control, please refer to the Chapter of File Format in JetFileII.

stay_time_sec :**[in]**
Stay time, 0-9999 second. This function is only valid for bmp files.
sign_ip :**[in]**
Display's IP address.
sign_port :**[in]**
Display ports.
is_store_ram :**[in]**
To check if setting is saved in RAM. 1 means it's saved. 0 means it's saved in Disk D or F.
type :**[in]**
Type of the communication and definition can be seen as below:

```
typedef enum
  {
    COMM_UDP,
    COMM_TCP,
   COMM_COM
  }conn_type;
```

comStr :**[in]**
Serial port communication by 'com' string, eg.COM1
baudrate :**[in]**
Serial communication with Baudrate, eg. 115200
dstAddr :**[in]**
GGUU address of the LED display, Unit Addr.is in high, Group Addr is on the is in low

Return Value
If the function is successfully operated, return to 0. Otherwise, return to Error code.
Examples

```
char *file[3] =
{
    {".\\1.bmp"},
    {".\\2.bmp"},
    {".\\3.bmp"}
};
    INT32U RT = czShowFilesEx((char**)nmgfile, 3, CZ_MODE_MOVEDOWN, 2,TEST_SIGN_IP, 9520, 0,
                                COMM_COM,TEST_SIGN_COM,TEST_SIGN_COM_BAUDRATE,TEST_SIGN_COM_GGUU);
```

Correlation function
czShowMsg

4.10.Read file From the LED display (czEasyReadFileEx support Serial)

Function Prototype

```
INT32U czEasyReadFileEx(char* pc_file_path, char* sign_file_path,char* sign_ip,INT32U sign_port,
                        conn_type type=COMM_UDP,char* comStr="COM3",INT32U baudrate=9600,INT16U dstAddr=0x0101);
```

Description

Read file from the LED display

Parameter

pc_file_path :**[in]**
Files' path on the local computer
sign_file_path :**[in]**
Files' path on the sign.
sign_ip :**[in]**
Display's IP address.
sign_port :**[in]**
Display ports.
type :**[in]**
Type of the communication and definition can be seen as below:

```
typedef enum
  {
    COMM_UDP,
    COMM_TCP,
   COMM_COM
 }conn_type;
```

comStr :**[in]**
Serial port communication by 'com' string, eg.COM1
baudrate :**[in]**
Serial communication with Baudrate, eg. 115200
dstAddr :**[in]**
GGUU address of the LED display, Unit Addr.is in high, Group Addr is on the is in low

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
INT32U RT = czEasyReadFileEx ((char*)"E:\\1", (char*)"C:\\TEST.TXT", (char*)"169.254.10.49",
9520,
COMM_COM,TEST_SIGN_COM,TEST_SIGN_COM_BAUDRATE,TEST_SIGN_COM_GGUU);
```

Correlation function

czEasyWriteFile, czReadSpecPathFile

4.11.Write file To LED display (czEasyWriteFileEx support Serial)

Function Prototype

```
INT32U czEasyWriteFileEx(char* pc_file_path, char* sign_file_path,char* sign_ip,INT32U
sign_port,
                         conn_type type=COMM_UDP,char* comStr="COM3",INT32U
                         baudrate=9600,INT16U dstAddr=0x0101);
```

Description

Write file To the LED display

Parameter

pc_file_path :**[in]**
Files' path on the local computer
sign_file_path :**[in]**
Files' path on the sign.
sign_ip :**[in]**
Display's IP address.
sign_port :**[in]**
Display ports.
type :**[in]**
Type of the communication and definition can be seen as below:

```
typedef enum
  {
    COMM_UDP,
    COMM_TCP,
   COMM_COM
  }conn_type;
```

comStr :**[in]**
Serial port communication by 'com' string, eg.COM1
baudrate :**[in]**
Serial communication with Baudrate, eg. 115200
dstAddr :**[in]**
GGUU address of the LED display, Unit Addr.is in high, Group Addr is on the is in low

Return Value
If the function is successfully operated, return to 0. Otherwise, return to Error code.
Examples

```
INT32U RT = czEasyWriteFileEx ((char*)"E:\\1", (char*)"C:\\TEST.TXT", char*)"169.254.10.49",
9520,
COMM_COM,TEST_SIGN_COM,TEST_SIGN_COM_BAUDRATE,TEST_SIGN_COM_GGUU);
```

Correlation function
czEasyReadFile, czWriteSpecFile

4.12.Showing files in the LED display (czShowFilesSpe support Serial)

Function Prototype
INT32U czShowFilesSpe(char* files[], INT32U numfiles, INT32U mode, INT32U stay_time_sec, char* sign_ip,INT32U sign_port,INT32U is_store_ram=0,conn_type type=COMM_UDP,char* comStr="COM3",INT32U baudrate=9600,INT16U dstAddr=0x0101,INT8U sendType=0);
Description

To control the display to recurrently run a group of files, whose format can be bmp, pmg, qst or flw. The files will be saved in Disk F if the delivery is successfully operated. Otherwise, they will be saved in Disk D.Send files and playlist separately,Multiple screens are used to synchronize updates.

Parameter
files :**[in]**

Files' path on the local computer,which is with a two-dimension array structure.

numfiles :**[in]**

Number of files, that is to say, the length of one dimension.

mode :**[in]**

Mode. Please refer to mode define. This function is only valid for bmp files. For other file formats control, please refer to the Chapter of File Format in JetFileII.

stay_time_sec :**[in]**

Stay time, 0-9999 second. This function is only valid for bmp files.

sign_ip :**[in]**

Display's IP address.

sign_port :**[in]**

Display ports.

is_store_ram :**[in]**

To check if setting is saved in RAM. 1 means it's saved. 0 means it's saved in Disk D or F.

type :**[in]**

Type of the communication and definition can be seen as below:

```
typedef enum
  {
    COMM_UDP,
    COMM_TCP,
   COMM_COM
 }conn_type;
```

comStr :**[in]**

Serial port communication by 'com' string, eg.COM1

baudrate :**[in]**

Serial communication with Baudrate, eg. 115200

dstAddr :**[in]**

GGUU address of the LED display, Unit Addr.is in high, Group Addr is on the is in low

sendType :**[in]**

Send type

0：Send files and Playlist

1：Only Send files

2：Only Send Playlist

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
char *file[3] =
{
    {".\\1.bmp"},
    {".\\2.bmp"},
    {".\\3.bmp"}
};
//only send files
INT32U RT = czShowFilesSpe((char**)nmgfile, 3, CZ_MODE_MOVEDOWN, 2,IPArr[i], 9520
```

```
0,
                    COMM_COM,TEST_SIGN_COM,TEST_SIGN_COM_BAUDRATE,TEST_SIG
N_COM_GGUU,1);

      //do other things

      //play files
    INT32U    RT    =    czShowFilesSpe((char**)nmgfile,    3,    CZ_MODE_MOVEDOWN
2,TEST_SIGN_IP, 9520, 0,
                    COMM_COM,TEST_SIGN_COM,TEST_SIGN_COM_BAUDRATE,TEST_SIG
N_COM_GGUU,2);
```

Correlation function

czShowMsg

4.13.Showing files in the LED display, Each time a bmp file to a different (czShowFilesII support Serial)

Function Prototype

INT32U czShowFilesSpeII(char* files[], INT32U numfiles, INT32U stay_time_sec[], INT32U mode, char* sign_ip,INT32U sign_port,INT32U is_store_ram=0,conn_type type=COMM_UDP,char* comStr="COM3",INT32U baudrate=9600,INT16U dstAddr=0x0101,INT8U sendType=0);

Description

To control the display to recurrently run a group of files, whose format can be bmp, pmg, qst or flw. The files will be saved in Disk F if the delivery is successfully operated. Otherwise, they will be saved in Disk D.Send files and playlist separately,Multiple screens are used to synchronize updates.

Parameter

files :**[in]**
Files' path on the local computer,which is with a two-dimension array structure.

numfiles :**[in]**
Number of files, that is to say, the length of one dimension.

stay_time_sec :**[in]**
Each bmp stay time, 0-9999 second. This function is only valid for bmp files.

mode :**[in]**
Mode. Please refer to mode define. This function is only valid for bmp files. For other file formats control, please refer to the Chapter of File Format in JetFileII.

sign_ip :**[in]**
Display's IP address.

sign_port :**[in]**
Display ports.

is_store_ram :**[in]**
To check if setting is saved in RAM. 1 means it's saved. 0 means it's saved in Disk D or F.

type :**[in]**

Type of the communication and definition can be seen as below:

```
typedef enum
  {
    COMM_UDP,
    COMM_TCP,
   COMM_COM
 }conn_type;
```

comStr :**[in]**

Serial port communication by 'com' string, eg.COM1

baudrate :**[in]**

Serial communication with Baudrate, eg. 115200

dstAddr :**[in]**

GGUU address of the LED display, Unit Addr.is in high, Group Addr is on the is in low

sendType :**[in]**

Send type

0：Send files and Playlist

1：Only Send files

2：Only Send Playlist

Return Value

If the function is successfully operated, return to 0. Otherwise, return to Error code.

Examples

```
char *file[3] =
{
    {".\\1.bmp"},
    {".\\2.bmp"},
    {".\\3.bmp"}
};
//only send files
INT8U stayTimes[3] = {1,5,8}
INT32U RT = czShowFilesII((char**)nmgfile, 3, (INT8U*)stayTimes,
CZ_MODE_MOVEDOWN, IPArr[i], 9520, 0,COMM_COM,TEST_SIGN_COM,
TEST_SIGN_COM_BAUDRATE,TEST_SIGN_COM_GGUU,1);

 //do other things

 //play files
RT = czShowFilesII((char**)nmgfile, 3, (INT8U*)stayTimes,
CZ_MODE_MOVEDOWN, IPArr[i], 9520, 0,COMM_COM,TEST_SIGN_COM,
TEST_SIGN_COM_BAUDRATE,TEST_SIGN_COM_GGUU,2);
```

Correlation function

czShowMsg

5. Examples

Please refer to the API Demo project.

6. Appendix Ⅰ

Error code table

Error code	Description
DLL common error	
0x00	OK
0x01	SNMP Read/Write Faild
0x02	Arguments Error
0x03	DLL NOT Init Error
0x04	DLL Call Timeout
0x0100	DLL Call Error
0x0101	DLL Memory Allocation Error
0x0102	File Format Error
0x0103	Can't Modify Error，NTCIP be used
0x0104	Task Not Exist Error
0x0105	DLL NO THIS FUNCTION Error
JetFileII Error	
0x4B4F	Packet Process OK
0x5245	Packet Process Error
0x9000	Process OK
0x9001	Head error of com synchronization!
0x9002	Sum check error!
0x9003	Address error!
0x9004	Major category invalid!
0x9005	Minor category invalid!
0x9006	Incorrect data packet length!
0x9008	File does not exist!
0x9009	End of file!
0x9010	Failure in opening the file!
0x9011	Not support this command
0x9012	File write-in failure!
0x9013	The packet size is wrong!
0x9014	The Packet ID is wrong!
0x9015	Delete the File failed!
0x9030	Please login first
0x9031	Password incorrect!
0x9032	User name incorrect!
0x9033	Original password incorrect!
0x9035	The display is being used by others!
0x9036	User not exist or permission denied!

0x1101	Oversize when reading!
0x1102	The Address Invalid!
0x1F01	Update firmware program Failed!
0x1F02	Updating...
0x1F03	No update operation!
0x2101	Data size exceeds 320K when writing a file!
0x2102	Not enough space!
0x2103	Not enough space in driver C !
0x2104	Not enough space in driver D !
0x2105	Not enough space in driver E !
0x2106	Not enough space in driver F !
0x2107	Not enough space in driver G !
0x2000	Memory allocation Fail!
0x2901	Data size exceeds 1024 bytes when writing an emergency message!
0x3A01	Not Support the Command!
0x5201	Time set unsuccessful!
0x6601	Invaid license!
0x6701	No current display file!
0x6702	Failure in opening current display file!
0x6703	Current display file oversized and can not be read. Please use extended reading command!
0x7201	Format failure!
0x7301	Failure in creating a folder!
0x7401	Failure in renaming a file!
0x7402	Incorrect path in renaming!
0x7501	Failure in moving a file!
0x7601	Failure in deleting a file!
0x7B01	Failure in opening a file!
0x7D01	Failure in reading disk information
0x7E01	The file not exist!
MultiZone Error	
0x36E1	The PageID is invalid!
0x36E2	The Page has not been configured!
0x36E3	The PageCount must be less then 13 !
0x36E4	The ZoneID is invalid!
0x36E5	The Zone has not been configured!
0x36E6	The PageCount is wrong!, and must be less then 21 !
PC SofteWare Operate Error	
0x5000	Unknow Error!
0x5001	Open Com Error!
0x5002	TimeOut to Connenct for TCP!
0x5003	Failure in Operation!
0x5004	Inconsistent Synchronization code!
0x5005	Inconsistent Source Address!
0x5006	Inconsistent Destination Address!
0x5007	Inconsistent Pack ID!
0x5008	Inconsistent Major category!

0x5009	Inconsistent Minor category!
0x5010	Inconsistent Sum check!
0x5011	Operation Abort!
0x5012	Play List is not exists!
0x5013	Unexpected file format!
0x5100	Communication Error!
0x5101	Failed open PC file!
0x5102	PC File is empty!
0x8301	Buffer 1 and Buffer 2 displaying. Buffer 3 ready to receive data!
0x8302	Buffer 2 and Buffer 3 displaying. Buffer 1 ready to receive data!
0x8303	Buffer 3 and Buffer 2 displaying. Buffer 1 ready to receive data!
0x8305	System not in the unlimited connection display status!
0x8306	Receiving Buffer full!
0x8307	Incorrect data format!
CtrlBoard Operate Error	
0x0C00	CtrlBoard Task In Buf!
0x0C01	CtrlBoard Task Success!
0x0C02	CtrlBoard Task Queue Error!
0x0C03	CtrlBoard Task Processing!
0x0C04	CtrlBoard Bus Error!
0x0C05	CtrlBoard Retry Exceed!
0x0C06	CtrlBoard Normal Error!
0x0C07	CtrlBoard Task ID Error!
0x0C08	CtrlBoard Brocast Error!
0x0C09	CtrlBoard LDU Ctrl Error!
0x0C11	CtrlBoard LDU Ctrl Addr Error!
0x0C12	CtrlBoard TWI Error!
0x0C13	CtrlBoard PixelPro Error!
0xFFFF	CtrlBoard Task Queue Full Error!