Research and realization of the data analysis of fixture library file

Zhijian Yin[1,a],Linhan Dai[1,a]

[1]College of Communication and Electronics,

Jiangxi Science & Technology Normal University,

Nanchang, Jiangxi, China.

[a]yinzhijian@jxstnu.edu.cn; [a]495358933@qq.com

Keywords: Stage lighting; Fixture library file; Software; Data analysis;

Abstract. With the continuous development of lighting industry and the lighting console built-in lamp library file update, so we need have an analysis of the software of different lighting manufacturers, different lamps and lanterns of the library file data. In this paper, there is a detailed analysis of the different lamp library files and data structure, and formulates a set of analytical library data structure, then puts forward the date with protocol and stored; at the same time, it will give the reason of this stored data. In this way can achieve a complete analysis of the data of all lights, and collect or extract of the data which control the station.

Introduction

Stage lighting console access to the network decoder output DMX signal control lamps and lanterns to transform a different model,logo, color, pattern, shining lights are relying on the fixture base data in the file. And every lighting manufacturers develop a fixture will use "Personality Bulider" or other software to make the corresponding library files. Therefore, analysis research for stage lighting console on the lamp library file is a crucial step in using software. The lamp library file parsing must be taken into account: 1 Comprehensive data analysis; 2 Data parsing speed; 3 The way data is stored, and the size of memory occupied. Only if all of the above conditions, the way the light library is to meet fixture the requirements of the resolution.

Fixture library file. Fixture library file for different console with different versions, due to the lighting industry in the fixture of the provisions of the data library files are stored in a certain way in accordance with the rules of storage. And thefixture library document is specially designed accordingtothe function whichthe manufacturer produces. Therefore, each lamp has its own lamp library files, and the content, size is not the same. General fixture file size is 10K to 40K, but also has a very powerful lighting function, the corresponding lamp library files to 100K. Lamp library file (D4 version of the library file) as shown in Fig.1:

```
<?xml version="1.0" encoding="utf-8"?>
<Fixture Name="EVA 75S" ShortName="EVA 75S" Company="Ablelite">
  <Copyright Notice="© Avolites Ltd. 2012" />
  <History Date="04-10-12" Author="Peter Budd" Comment="Created" />
  <History Date="08-10-12" Author="Peter Budd" Comment="Created from Ablelite EVA 60B" />
  <Manual Filename="" Summary="" />
```

Fig.1:Fixture library file

The beginning of the fixture library file describes the fixture library version of the file, select the name of the lighting, the corresponding manufacturers, as well as other information. Among the useful information are “Fixture name” and “Company”, the two information is the symbol of thislamp. The following is the lighting of the property information as shown in Fig.2:

```
<Attribute ID="Shutter_Func" Name="Shutter Func" Description="" Group="I">
  <Locate Locate="1" PowerOn="1" />
  <Function ID="1" Name="Normal" Display="'Normal'" Dmx="0~15" Update="Shutter" Intensity="0~0" />
  <Function ID="2" Name="Pulse FW" Display="'Pulse FW'" Dmx="16~31" Update="Shutter" Intensity="0~0" />
  <Function ID="3" Name="Pulse Rev" Display="'Pulse Rev'" Dmx="32~47" Update="Shutter" Intensity="0~0" />
  <Function ID="4" Name="Random" Display="'Random'" Dmx="48~63" Update="Shutter" Intensity="0~0" />
  <Function ID="5" Name="Reserved" Display="'Reserved %.f',64~255" Dmx="64~255" Update="Shutter" Intensity="0~0" />
```

Fig.2:Fixture library file

The attributes of the lighting fixture include the ID, the name of the lamp, the lamp group, and each attribute contains a number of corresponding small functions. Functions include the function of the ID, the function name, the output of the DMX range, etc.. Some attributes include not only the general function, but also the properties of the control, as shown in Fig.3:

```
<Attribute ID="Colour1" Name="Colour" Description="" Group="C">
  <Locate Locate="1:0" PowerOn="1:0" />
  <Condition Name="Spin CW" ID="1" If="'Colour1Func'==3" />
  <Condition Name="Spin CCW" ID="2" If="'Colour1Func'==4" />
  <Condition Name="Index" ID="3" If="'Colour1Func'==5" />
  <Condition Name="Fixed" ID="4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21" If="'Colour1Func'==1||'Colour1Func'==2" />
  <Condition Name="Shake" ID="23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40" If="'Colour1Func'==6" />
  <Else ID="22" />
  <Function ID="1" Name="Scroll CW" Display="'Scroll CW     %.f%%',0~100" Dmx="0~255" Colour="0,0,0" />
  <Function ID="2" Name="Scroll CCW" Display="'Scroll CCW    %.f%%',0~100" Dmx="0~255" Colour="0,0,0" />
  <Function ID="3" Name="Index" Display="'Index %.1f%%',0.0~100.0" Dmx="0~255" Colour="0,0,0" />
  <Function ID="4" Name="Open" Display="'Open'" Dmx="0~13" Colour="0,0,0" />
  <Function ID="5" Name="Colour 1" Display="'Colour 1'" Dmx="14~27" Colour="0,0,0" />
  <Function ID="6" Name="Colour 2" Display="'Colour 2'" Dmx="28~41" Colour="0,0,0" />
  <Function ID="7" Name="Colour 3" Display="'Colour 3'" Dmx="42~55" Colour="0,0,0" />
```

Fig.3:Fixture library file

Control property contains the name of the control, it controls the function ID, and meet the conditions corresponding to the selected control. Control property belonging to another regular fixture library attribute controls only control its property has the option to change its properties will change accordingly. According to a control specific functions corresponding to the ID to FuctionID.Immediately lighting pattern is shown in Fig.4:

```
<Mode Name="Basic, 16 DMX" Channels="16">
  <Import PearlRef="UK_AbEV75SB" DiamondRef="" WysiwygRef="" />
  <Physical>
    <Bulb Type="" Lumens="" ColourTemp="" />
    <Lens Name="" Degrees="" />
    <Weight Kg="" />
    <Size Height="" Width="" Depth="" />
    <Focus Type="Fixed" PanMax="0" TiltMax="0" PanMaxSpeed="0" TiltMaxSpeed="0" />
  </Physical>
  <Include>
    <Attribute ID="Shutter_Func" ChannelOffset="4" Wheel="5" />
    <Attribute ID="Shutter" ChannelOffset="5" Wheel="4" />
    <Attribute ID="Dimmer" ChannelOffset="6" Wheel="1" />
    <Attribute ID="Pan" ChannelOffset="1" Wheel="2" />
    <Attribute ID="Tilt" ChannelOffset="2" Wheel="3" />
    <Attribute ID="Colour1Func" ChannelOffset="7" Wheel="7" />
    <Attribute ID="Colour1" ChannelOffset="8" Wheel="8" />
    <Attribute ID="GoboIFunc" ChannelOffset="9" Wheel="13" />
    <Attribute ID="GoboI" ChannelOffset="10" Wheel="14" />
    <Attribute ID="GoboRotFunc" ChannelOffset="11" Wheel="43" />
    <Attribute ID="GoboRot" ChannelOffset="12" Wheel="44" />
    <Attribute ID="Focus" ChannelOffset="15" Wheel="20" />
    <Attribute ID="Prism" ChannelOffset="13" Wheel="23" />
    <Attribute ID="Prism_Rot" ChannelOffset="14" Wheel="24" />
    <Attribute ID="PT_Speed" ChannelOffset="3" Wheel="25" />
    <Attribute ID="Control" ChannelOffset="16" Wheel="28" />
  </Include>
```

Fig.4:Fixture library file

Each lamp has several different modes, parsing library files need to read all of the modes, different modes defines the number of channels configured lamps, and under that mode all properties. Specific to the name of the property, in the channel where the offset, in the display of the wheel in the position. The library must use this as an index to further read the corresponding properties.

Analysis of the data structure of the fixture library. Specifies the data structure of the lamp bank to analyze and observe a variety of lamp library files, including the data structure that contains the applicable to all lamp library files. Fixture library part of the data structure and finally determined as shown in Fig.5:

```
typedef struct
{
  Dmx_ModeTypedef DmxMode;
  LightAttribute Attribute[ATTRIBUTE_MAX_NUM];
  MacroTypedef Macro[MACRO_MAX_NUM];
  PalettesTypedef Palettes[PALETTES_MAX_NUM];
}LightDataStruct;
```

```
typedef struct CompanyNameStruct
{
  char CompanyName[STD_CHAR_LEN];
  struct CompanyNameStruct *Next ;
}CompanyNameStructTypedef;
```

```
typedef struct  ConditionId
{
  int ID[CONDT_ID_NUM];
  struct  ConditionId *Next ;
}ConditionIdList;
```

Fig.5:Data structure

Data structure including the lamp manufacturer's information, lamps and lanterns information.Lighting manufacturers information also contains the name of the manufacturer, lighting. Lighting information contains the mode information, lighting property information, modal properties, lighting material information. All of the above defined data structure can contain information about all the fixtures, all the lamps apply.

Theanalytical processes and fixture library data storage. Due to taking into account the efficiency of the lamp library file retrieval, before the lamp library file parsing. Need to prepare the host computer software will be all the lights in advance to retrieve it, from which to extract the key words. Therefore, the host computer software will be used to generate the fixturelibrary files of all the company's corresponding lamps and lanterns retrieval. As shown in Fig.6:

```
Tourspot 250                          AC Lighting_TourSpot 250.d4                    TourSpot575
 AC Lighting_TourSpot 575 16 DMX channels mode.d4  Tourspot 575                       AC Lighting_TourSpot 575.d4
            Tourwash 250                           AC Lighting_TourWash 250.d4                Tourwash 575
             AC Lighting_TourWash 575.D4
```

Fig.6:Document Retrieval

Then, check the lighting manufacturers to begin the appropriate library file parsed lamp fixtures selected as the flowchart shown in Fig.7:

Fig.7:Program flow chart

Parsing library files, it should always record the position of the fixture library when reading the data of the current position of the appropriate number of bytes read length, and then a bunch of these strings a keyword collection, extract stage lighting console information required data. Due to the large library files, the mounting structure can not be the way to the largest library files open up a maximum length of memory space. Therefore, the only way in accordance with the agreement depending on the size of the light library to open up some memory space. And read the file, it also must follow the protocol at the specified location fixture library stored data will formulate relevant data length read out. Finally, it must be written in the data analysis process file system, configured to avoid data loss lamps, and the need to release this memory. Fixture Library storage protocol analysis data shown in Fig.8:

Management array of bytes occupied by each attribute is fixed 15, but also know the number, so every 15 representatives of a Attribute data management, according to the order of the properties prior to sequentially read the same data array.

1Byte	4Byte
(Corresponding dmxmode Id)	Att read basic information about location
1Byte	4Byte
There are several Condition	Condition reading position
Without Condition is before one is 0, condition equal to a position or location	
1Byte	4Byte
There are several reading	Function Function position

Fig.8: Data storage protocol

System testing and results analysis. Through the Windows operating system to write PC software written for library files parsing, parsing out the part of the data shown in Fig.9:

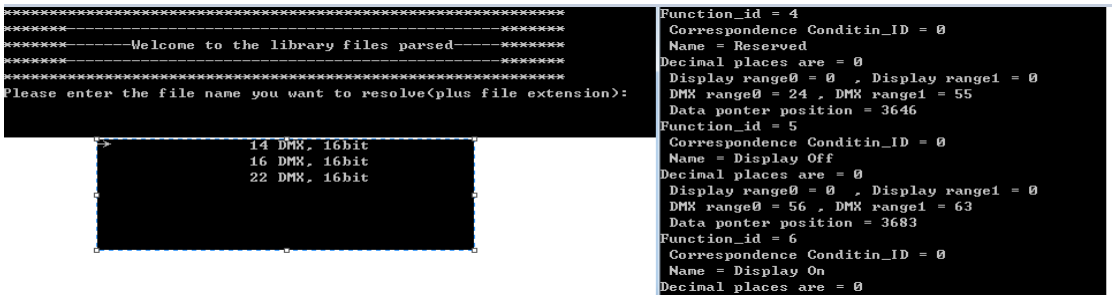

Fig.9:Test result

After entering the analysis software library lamp fixtures to enter the name of the library to be resolved, the software will find the appropriate library files in the lamp library file. This database and retrieve all fixture s mode, the user needs to select the required resolution mode, the software according to the corresponding mode data parsed. Any data includes data required for control stations, as well as data stored location according to the agreement.Debug information display, in the analysis of the lamp library file, the lamp library file each byte of the judgment and analysis, sorting out the light control console useful information data. After collecting a set of data, the data is stored in the memory, and the corresponding data storage address is recorded. In a large array of management, an array of storage data is responsible for the management of an array of data stored in the data space of the address. In this way, at any time can be extracted according to the address of the desired data.

Conclusion

By collecting information extraction library files, you can use the above method is applicable to any library documents. It has the following advantages: 1 Can quickly extract light library data, and comprehensive data; 2 Even if the manufacturer update the fixture library or add a new library files also can be resolved; 3 Saves memory, even 100K lamp library. file parsing out the memory occupied by about 10K or so; 4 Data to facilitate extraction agreement to develop and debug; the same time, the light library of analytical methods apply equally to resolve other types of file systems. In the actual product development process involves reading the file can be on the same reference.

References

[1] Xiao Guangbing.ARM embedded development example. Electronics Industry Press.2013.04

[2] Wang Lei, Zhu Xing. The art of software framework design. People's Posts and Telecommunications Publishing House,.2011.03

[3] Lewin A.R.W.Edwards. Embedded engineers must know will be People's Posts and Telecommunications Press .2011.03.4

[4] Ling Ming embedded system-level C language programming. Beijing University of Aeronautics and Astronautics Press .2011.01.1

[5] Dai Yan. Zero-based learning algorithm. Machinery Industry Press .2010.01

[6] KennethA. Reek. Pointers On C. People Post Press .2008