

C++语言基础

迂者 - 贺利坚

<http://blog.csdn.net/sxhelijian/>

<http://edu.csdn.net>



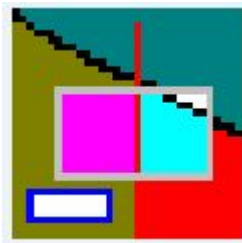


本节主题:

案例: bmp文件格式剖析

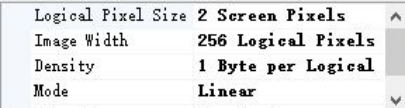
bmp文件格式的操作

- ❏ BMP，全称Bitmap，Windows操作系统中的标准图像文件格式
- ❏ 典型的BMP图像文件由四部分组成
 - ❏ 位图头文件头数据结构
 - ❏ 位图信息头
 - ❏ 彩色表/调色板
 - ❏ 位图数据
- ❏ 以bmp为例，体会二进制文件的存储，以及二进制数据的操纵



一幅32×32的16色bmp图像

□ ×



| Size: 630 Dec/276 Hex

[illegible]

 Bookmarks
 Structures
 Histogram

Ⓛ ×

Binary	N/A
Octal	N/A

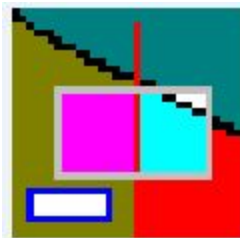
4 X

Temporary False

Size	630
------	-----

File name.

位图文件头



位图文件头(bitmap-file header)

位图信息数据结构 { 位图信息头(bitmap-information header)
彩色表(color table)

位图信息(bitmapinfo)

位图文件头格式

```
typedef struct tagBITMAPFILEHEADER
```

```
{
```

```
    UINT bfType; /*说明文件的类型*/
```

```
    DWORD bfSize; /*说明文件的大小
```

```
    UINT bfReserved1; /*保留，设置为0*/
```

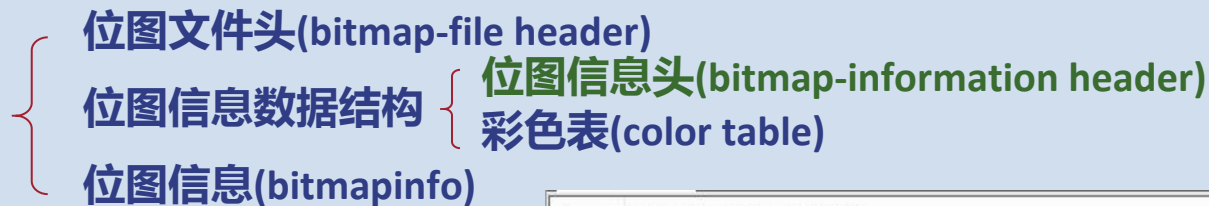
```
    UINT bfReserved2; /*保留，设置为0*/
```

```
    DWORD bfOffBits; /*到图像数据的偏移量*/
```

```
}BITMAPFILEHEADER;
```

A...	Hexadecimal (1 Byte)															
0000	42	4D	76	02	00	00	00	00	00	00	00	76	00	00	00	28 00
0010	00	00	20	00	00	00	20	00	00	00	01	00	04	00	00	00
0020	00	00	00	02	00	00	C4	0E	00	00	C4	0E	00	00	00	00
0030	00	00	00	00	00	00	00	00	00	00	00	00	00	80	00	80
0040	00	00	00	80	80	00	80	00	00	00	80	00	80	00	80	80
0050	00	00	80	80	80	00	C0	C0	C0	00	00	00	FF	00	00	FF
0060	00	00	00	FF	FF	00	FF	00	00	00	FF	00	FF	00	FF	FF
0070	00	00	FF	FF	FF	00	33	33	33	33	33	33	33	33	33	39 99
0080	99	99	99	99	99	99	33	33	33	33	33	33	33	33	33	39 99
0090	99	99	99	99	99	99	33	CC	CC	CC	CC	CC	CC	CC	33	39 99
00A0	99	99	99	99	99	99	33	CF	FF	FF	FF	FF	FF	FC	33	39 99
00B0	99	99	99	99	99	99	33	CF	FF	FF	FF	FF	FF	FC	33	39 99
00C0	99	99	99	99	99	99	33	CF	FF	FF	FF	FF	FF	FC	33	39 99
00D0	99	99	99	99	99	99	33	CC	CC	CC	CC	CC	CC	CC	33	39 99
00E0	99	99	99	99	99	99	33	33	33	33	33	33	33	33	33	39 99
00F0	99	99	99	99	99	99	33	33	33	33	88	88	88	88	88	88 88
0100	88	88	88	88	99	99	33	33	33	8D	DD	DD	DD	DD	DD	D9 EE
0110	EE	EE	EE	E8	99	99	33	33	33	8D	DD	DD	DD	DD	DD	D9 EE
0120	EE	EE	EE	E8	99	99	33	33	33	8D	DD	DD	DD	DD	DD	D9 EE
0130	EE	EE	EE	E8	99	99	33	33	33	8D	DD	DD	DD	DD	DD	D9 EE
0140	EE	EE	EE	E8	99	99	33	33	33	8D	DD	DD	DD	DD	DD	D9 EE

位图信息头



```
typedef struct tagBITMAPINFOHEADER {
```

```
    DWORD biSize; /*BITMAPINFOHEADER结构所需要的字节数*/
```

```
    LONG biWidth; /*图像的宽度，以像素为单位*/
```

```
    LONG biHeight; /*图像的高度，以像素为单位*/
```

```
    WORD biPlanes; /*为目标设备说明位面数，其值设置为1*/
```

```
    WORD biBitCount; /*位数/像素*/
```

```
    DWORD biCompression; /*图像数据压缩的类型：不压缩，或4/8位RLE */
```

```
    DWORD biSizeImage; /*图像的大小，以字节为单位。*/
```

```
    LONG biXPelsPerMeter; /*水平分辨率，用像素/米表示*/
```

```
    LONG biYPelsPerMeter; /*垂直分辨率，用像素/米表示*/
```

```
    DWORD biClrUsed; /*位图使用的彩色表中的颜色索引数：2/16/256/224*/
```

```
    DWORD biClrImportant; /*对图像显示有重要影响的颜色索引的数目，如果是0，表示都重要*/
```

```
} BITMAPINFOHEADER;
```

A...	Hexadecimal (1 Byte)														
0000	42	4D	76	02	00	00	00	00	00	00	76	00	00	00	28 00
0010	00	00	20	00	00	00	20	00	00	00	01	00	04	00	00 00
0020	00	00	00	02	00	00	C4	0E	00	00	C4	0E	00	00	00 00
0030	00	00	00	00	00	00	00	00	00	00	00	80	00	00	80
0040	00	00	00	80	80	00	80	00	00	00	80	00	80	00	80
0050	00	00	80	80	80	00	C0	C0	C0	00	00	00	FF	00	FF
0060	00	00	00	FF	FF	00	FF	00	00	00	FF	00	FF	00	FF
0070	00	00	FF	FF	FF	00	33	33	33	33	33	33	33	33	39 99
0080	99	99	99	99	99	99	33	33	33	33	33	33	33	33	39 99
0090	99	99	99	99	99	99	33	CC	CC	CC	CC	CC	CC	33	39 99
00A0	99	99	99	99	99	99	33	CF	FF	FF	FF	FF	FC	33	39 99
00B0	99	99	99	99	99	99	33	CF	FF	FF	FF	FF	FC	33	39 99
00C0	99	99	99	99	99	99	33	CF	FF	FF	FF	FF	FC	33	39 99
00D0	99	99	99	99	99	99	33	CC	CC	CC	CC	CC	CC	33	39 99
00E0	99	99	99	99	99	99	33	33	33	33	33	33	33	33	39 99
00F0	99	99	99	99	99	99	33	33	33	88	88	88	88	88	88
0100	88	88	88	88	99	99	33	33	33	8D	DD	DD	DD	DD	D9 EE
0110	EE	EE	EE	E8	99	99	33	33	33	8D	DD	DD	DD	DD	D9 EE
0120	EE	EE	EE	E8	99	99	33	33	33	8D	DD	DD	DD	DD	D9 EE
0130	EE	EE	EE	E8	99	99	33	33	33	8D	DD	DD	DD	DD	D9 EE
0140	EE	EE	EE	E8	99	99	33	33	33	8D	DD	DD	DD	DD	D9 EE

彩色表

位图信息数据结构

位图文件头(bitmap-file header)

位图信息数据结构

位图信息头(bitmap-information header)

彩色表(color table)

位图信息(bitmapinfo)

```
typedef struct tagRGBQUAD { /* rgbq */
    BYTE rgbBlue; /*指定蓝色强度*/
    BYTE rgbGreen; /*指定绿色强度*/
    BYTE rgbRed; /*指定红色强度*/
    BYTE rgbReserved; /*保留，设置为0 */
} RGBQUAD;

typedef struct tagBITMAPINFO {
    BITMAPINFOHEADER bmiHeader; // 位图信息头
    RGBQUAD bmiColors[16]; // 颜色表
} BITMAPINFO;
```

A...	Hexadecimal	(1 Byte)
0000	42	4D 76 02 00 00 00 00 00 00 00 76 00 00 00 28 00
0010	00	00 20 00 00 00 00 20 00 00 00 01 00 04 00 00 00
0020	00	00 00 02 00 00 00 C4 0E 00 00 C4 0E 00 00 00 00
0030	00	00 00 00 00 00 00 00 00 00 00 00 80 00 00 00 80
0040	00	00 00 80 80 00 80 00 00 00 80 00 80 00 80 80 80
0050	00	00 00 80 80 00 C0 C0 00 00 00 FF 00 00 FF
0060	00	00 00 FF FF 00 FF 00 00 00 FF 00 FF 00 FF FF
0070	00	00 FF FF FF 00 33 33 33 33 33 33 33 33 33 39 99
0080	99	99 99 99 99 99 99 33 33 33 33 33 33 33 33 39 99
0090	99	99 99 99 99 99 99 33 CC CC CC CC CC CC 33 39 99
00A0	99	99 99 99 99 99 99 33 CF FF FF FF FF FC 33 39 99
00B0	99	99 99 99 99 99 99 33 CF FF FF FF FF FC 33 39 99
00C0	99	99 99 99 99 99 99 33 CF FF FF FF FF FC 33 39 99
00D0	99	99 99 99 99 99 99 33 CC CC CC CC CC CC 33 39 99
00E0	99	99 99 99 99 99 99 33 33 33 33 33 33 33 33 39 99
00F0	99	99 99 99 99 99 99 33 33 33 88 88 88 88 88 88 88
0100	88	88 88 88 88 99 99 33 33 33 8D DD DD DD DD D9 EE
0110	EE	EE EE EE E8 99 99 33 33 33 8D DD DD DD DD D9 EE
0120	EE	EE EE EE E8 99 99 33 33 33 8D DD DD DD DD D9 EE
0130	EE	EE EE EE E8 99 99 33 33 33 8D DD DD DD DD D9 EE
0140	EE	EE EE EE E8 99 99 33 33 33 8D DD DD DD DD D9 EE

图像的位图信息

位图文件头(bitmap-file header)

位图信息数据结构

位图信息头(bitmap-information header)

彩色表(color table)

位图信息(bitmapinfo)

各个像素的颜色的序号(彩色表的下标)

例：3 —— 颜色值为bmiColors[3]，
BGR值分别为0x00, 0x80, 0x80

再例：9 —— 颜色值为bmiColors[9]，
BGR值分别为0x00, 0x00, 0xFF

.....

从左下到右上的像素

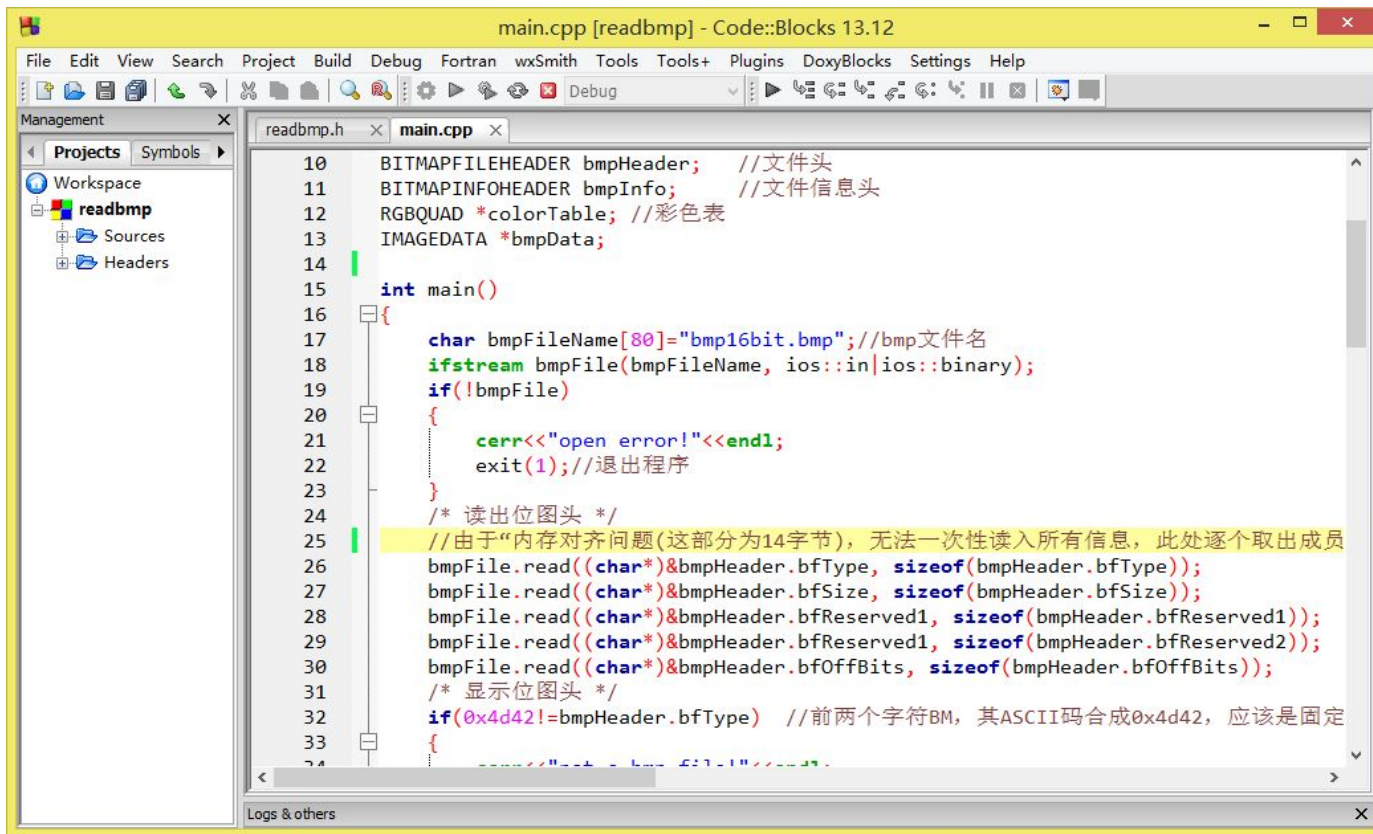
与设备操控有关

长度：biSizeImage



A...	Hexadecimal (1 Byte)
0000	42 4D 76 02 00 00 00 00 00 00 76 00 00 00 28 00
0010	00 00 20 00 00 00 20 00 00 00 01 00 04 00 00 00
0020	00 00 00 02 00 00 C4 0E 00 00 C4 0E 00 00 00 00
0030	00 00 00 00 00 00 00 00 00 00 00 00 80 00 00 80
0040	00 00 00 80 80 00 80 00 00 00 80 00 80 00 80 80
0050	00 00 80 80 80 00 C0 C0 C0 00 00 00 FF 00 00 FF
0060	00 00 00 FF FF 00 FF 00 00 00 FF 00 FF 00 FF FF
0070	00 00 FF FF FF 00 33 33 33 33 33 33 33 33 39 99
0080	99 99 99 99 99 99 33 33 33 33 33 33 33 33 39 99
0090	99 99 99 99 99 99 33 CC CC CC CC CC 33 39 99
00A0	99 99 99 99 99 99 33 00 FF FF FF FF FC 33 39 99
00B0	99 99 99 99 99 99 33 CF FF FF FF FF FC 33 39 99
00C0	99 99 99 99 99 99 33 CF FF FF FF FF FC 33 39 99
00D0	99 99 99 99 99 99 33 CC CC CC CC CC CC 33 39 99
00E0	99 99 99 99 99 99 33 33 33 33 33 33 33 33 39 99
00F0	99 99 99 99 99 99 33 33 33 88 88 88 88 88 88 88
0100	88 88 88 88 99 99 33 33 33 8D DD DD DD DD D9 EE
0110	EE EE EE E8 99 99 33 33 33 8D DD DD DD DD D9 EE
0120	EE EE EE E8 99 99 33 33 33 8D DD DD DD DD D9 EE
0130	EE EE EE E8 99 99 33 33 33 8D DD DD DD DD D9 EE
0140	EE EE EE E8 99 99 33 33 33 8D DD DD DD DD D9 EE

用程序读出BMP文件信息



```
main.cpp [readbmp] - Code::Blocks 13.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Management
Projects Symbols
Workspace
readbmp
Sources
Headers

10  BITMAPFILEHEADER bmpHeader; //文件头
11  BITMAPINFOHEADER bmpInfo; //文件信息头
12  RGBQUAD *colorTable; //彩色表
13  IMAGEDATA *bmpData;
14
15  int main()
16  {
17      char bmpFileName[80]="bmp16bit.bmp";//bmp文件名
18      ifstream bmpFile(bmpFileName, ios::in|ios::binary);
19      if(!bmpFile)
20      {
21          cerr<<"open error!"<<endl;
22          exit(1);//退出程序
23      }
24      /* 读出位图头 */
25      //由于“内存对齐问题(这部分为14字节),无法一次性读入所有信息,此处逐个取出成员
26      bmpFile.read((char*)&bmpHeader.bfType, sizeof(bmpHeader.bfType));
27      bmpFile.read((char*)&bmpHeader.bfSize, sizeof(bmpHeader.bfSize));
28      bmpFile.read((char*)&bmpHeader.bfReserved1, sizeof(bmpHeader.bfReserved1));
29      bmpFile.read((char*)&bmpHeader.bfReserved2, sizeof(bmpHeader.bfReserved2));
30      bmpFile.read((char*)&bmpHeader.bfOffBits, sizeof(bmpHeader.bfOffBits));
31      /* 显示位图头 */
32      if(0x4d42!=bmpHeader.bfType) //前两个字符BM,其ASCII码合成0x4d42,应该是固定
33      {
34          //...
35      }
36  }
```

THANKS

本课程由 迂者-贺利坚 提供

CSDN网站：www.csdn.net
企业服务：<http://ems.csdn.net/>
人才服务：<http://job.csdn.net/>
CTO俱乐部：<http://cto.csdn.net/>
高校俱乐部：<http://student.csdn.net/>
程序员杂志：<http://programmer.csdn.net/>

CODE平台：<https://code.csdn.net/>
项目外包：<http://www.csto.com/>
CSDN博客：<http://blog.csdn.net/>
CSDN论坛：<http://bbs.csdn.net/>
CSDN下载：<http://download.csdn.net/>

