

# Protel 原理图文件的结构分析

杨应成 唐大全 靳天成 (烟台海军航空工程学院 302 教研室 264001)

**摘要:**本文分析了 Protel 原理图文件的结构,讲述了在 Visual C++ 下读取和显示 Protel 原理图的方法和步骤,并给出了程序流程图。

**关键词:**Protel 原理图文件 Visual C++ 显示

## 一、前言

Protel 是澳大利亚 PROTEL TECHNOLOGY 公司研制的普及型电路辅助设计软件,其强大的功能及方便、易学、实用、快速的特点,使其进入中国以来得到了广泛的应用。现在,许多科技工作者利用 Protel 进行电路设计,并补充了一些新功能,使其功能进一步完善。

在进行电路故障诊断、排版设计时,会遇到电路图的编辑、显示和打印等问题,如果在自己的程序中加入电路图编辑功能,工作量和难度会很大,若能直接从 Protel 原理图文件中得到电路图,就会很方便。本文首先分析 Protel 原理图文件的结构,然后介绍在 Visual C++ 下读取和显示 Protel 原理图的方法和步骤。

## 二、原理图文件结构

Protel 原理图是利用原理图编辑工具 SCHEDIT 生成的以 .SOX 为扩展名的文件,它保存了电路中的所有信息,主要有:原理图的尺寸、各个部件的坐标、点阵信息、文字注释等。下面对原理图的各个部分进行分析。

### 1. 文件头的结构

文件头位于文件的开始(OFFSET:0000H),长度为 16 字节,它定义了原理图的尺寸及标志字符串,可用如下结构表示:

```
struct FILEHEADER{
    BYTE reserved;//保留
    BYTE schemsize;
    BYTE reserved;//保留
    BYTE resvered;//保留
    BYTE lablelength;//原理图标志字符串的长度
    char lable[11];//标志字符串“SCHEMATIC”
    }fileheader;
```

其中 fileheader.schemsize 是原理图图号大小,其值与图号的对应关系如下:

表 1 原理图尺寸

值	图号	图纸尺寸	值	图号	图纸尺寸
01H	A4	11.1×7.2 英寸	06H	A	10×8 英寸
02H	A3	15.5×11.1 英寸	07H	B	15×10 英寸
03H	A2	22.3×15.7 英寸	08H	C	20×15 英寸
04H	A1	31.5×22.3 英寸	09H	D	32×20 英寸
05H	A0	44.6×31.5 英寸	0AH	E	

### 2. 线段的结构(索引:10H)

原理图上的线段有信号线、总线、虚线,每一条线段的描述项长度是 16 字节,可用如下结构表示:

```
struct LINE{
    BYTE lineindex;//线段的索引:10H
    WORD startx;
    WORD starty;
    WORD endx;
    WORD endy;//线段的起点和终点坐标
    BYTE linetype;
    BYTE reserved[6];//保留
```

其中 file.linetype 表示线段的类型,其值的对应关系如下:

表 2 线段的类型

值	类型	线宽
01H	细信号线	一个像素
02H	粗信号线	四个像素
03H	总线	四个像素
04H	虚线	一个像素

### 3. 节点的结构(索引:20H,60H)

节点的索引为 20H 或 60H,每个节点的描述项长度为 16 字节,可用如下结构表示:

```
struct JUNCTION{
    BYTE junctionindex;//节点的索引:20H 或 60H
```

```
WORD junctionx;
WORD junctiony;//节点的坐标
BYTE reserved;//保留
BYTE size;//节点的大小
BYTE reserved[9];//保留
}junction;
```

其中当 INDEX = 20H 时, 节点直径固定为 4; 当 index = 60H 时, size = 02H 节点直径为 6, size = 03H 节点直径为 8。

#### 4. 文字注释的结构(索引:30H)

原理图上的注释和网络标号均以相同的结构存于文件中, 每一文字注释的描述项长度为 32 字节, 能容纳 22 个英文字符, 可用如下结构表示:

```
struct TEXT{
    BYTE textindex;//文字注释的索引:30H
    WORD textx;
    WORD texty;//文字注释的坐标
    BYTE textmode;
    BYTE texttype;//类型:00H 为文字注释, 01H 为网络标号
    BYTE reserved;//保留
    BYTE textlength;//注释字符串的长度
    char textcontent1[7];//注释字符串
    BYTE relay;//附加标志:31H
    char textcontent2[15];//注释字符串
}text;
```

其中 text.textmode 保存了两种字符串信息: 字号和方向, 其高四位(MSB)定义字符串的方向, MSB = 0 表示水平方向; MSB = 8 表示垂直方向。其低四位(LSB)定义字符串的字号, 其对应关系如下:

表 3 字符的大小

LSB	字符大小	LSB	字符大小
1	8 点阵	3	16 点阵
2	10 点阵	4	20 点阵

#### 5. 文字盒的结构(索引:50H)

文字盒用于在电路图上标注大块的文字, 它的描述项长度随盒内字符串的长短而变, 由两部分组成, 一是文字盒的定义, 二是文字, 文字盒的定义部分可用如下的结构表示:

```
struct DOCUMENT{
    BYTE documentboxindex;//文字盒的索引
```

```
WORD documentboxstartx;
WORD documentboxstarty;
WORD documentboxendx;
WORD documentboxendy;//文字盒的左上角和右下角坐标
BYTE textmaxlength;//文字盒内字符串的最大长度
BYTE reserved;//保留
BYTE border;//文字盒边框的显隐, 00H 表示显示, 01H 表示隐藏
BYTE paragraph;//文字盒内字符串的长度
BYTE reserved[3];//保留
}document;
```

其中 paragraph 的值决定了文字盒内文字的长度, 可由下面的公式推出:

$$\text{textlength} = 16 * \text{document.paragraph};$$

#### 6. 元器件的结构(索引:70H)

在原理图中, 元器件的结构最复杂, 它的描述项长度不定, 大致分为元器件头、元器件图形信息和元器件管脚。每一元器件都有元器件头, 其长度为 64 字节, 而元器件图形信息和元器件管脚随元器件的种类不同, 可能有一项、两项或没有。元器件头的描述项可用如下结构表示:

```
struct COMHEADER{
    BYTE comindex;//元器件的索引:70H
    BYTE dsnlenght;//元器件标号的字符串长度
    char designator[10];//标号的字符串内容
    WORD bitmapx;
    WORD bitmapy;//元件的点阵或矩形轮廓的左上角坐标
    char char A;//字符 A
    BYTE valuelenght;//元件值的字符串长度
    BYTE value[14];//元件值的字符串内容
    char charB;//字符 B
    BYTE invstate;//翻转角度 00:0°, 01:90°, 02:180°, 03:270°, 04:360°
    char reserved;//保留
    WORD designatorx;
    WORD designatory;//元件标号的坐标
    WORD valuex;
    WORD vauley;//元件值的坐标
    BYTE serialno;//元件序号, 表示此元件在片子中是第几套
    BYTE comset;//片子内元件的套数
```

```

BYTE bitmapwidth;//元件点阵或矩形框的宽度
BYTE bitmapheight;//元件点阵或矩形框的高度
BYTE comtype;//元件类型
char charC;//字符 C
BYTE pinnumber;//元件的管脚数
BYTE textsize;//元件标号的字符串字号
BYTE valuesize;//元件值的字符串字号
BYTE packagesize;//元件封装形式字符串的字号
BYTE packagelength;//元件封装形式字符串的长度
char package[10];//元件封装形式字符串的内容
    }comheader;

```

在元器件头的描述项之后是元件图形信息描述项,若一个芯片内有一至二个元件,其长度为16字节,三至四个元件,其长度为32字节,依次类推,可用如下结构表示:

```

struct COMBITMAP{
    BYTE comtype;//元件图形的种类:00H为矩形框,01H为点阵图形
    BYTE reserved;//保留
    BYTE comwidth;//元件的宽度
    BYTE comheight;//元件的高度
    }combitmap;

```

在元件图形信息描述项之后是元件管脚描述项,每一管脚的长度是16字节,可用如下结构表示:

```

struct COMPIN{
    BYTE pincomnum;//表示该管脚属于元件的第几个单元
    BYTE pintype;//管脚类型
    BYTE pinnumlen;//管脚号的长度
    char pinnum[4];//管脚号
    BYTE pinnamelen;//管脚名称长度
    BYTE pinname[6];//管脚名称
    BYTE pinpos;//管脚位置 00H:左;01H:底;02H:右;03H:顶
    BYTE pinoffset;//管脚距器件顶部的偏移量
    }compin;

```

若元件图形的种类是点阵,那么在管脚描述项之后,还有元件点阵描述项,其字节数可由图形信息的长度和宽度得到。若元件图形的种类是矩形框,则元件的所有信息在管脚描述项之后就结束了。

### 三、原理图文件的读取与显示

Visual C++ 是面向对象的程序设计语言,其 MFC 封装了几乎所有的 WINDOWS API 函数,提供了完善的文件读取、图形绘制能力,可以使我们以各种方式读取文件,绘制各种图形如矩形、点、线和点阵等。因此我们可利用第二部分介绍的 Protel 原理图文件结构,从文件中读取数据,并赋予相应结构的成员,然后根据所得的信息,将部件一一绘制出来,具体实现的步骤为:

1. 利用 Visual C++ 的应用向导生成单文档接口或多文档接口应用程序;
2. 在文档类中定义上述几种结构体类型变量;
3. 创建 CFile 类的对象,用其成员函数 Open() 以只读方式打开 Protel 原理图文件,然后用 Seek() 函数定位文件指针,用 Read() 函数读取该位置的数据,并根据索引值将结果存入文档中相应结构的成员变量中;
4. 在视图类中,用 GetDocument() 函数获取文档类的指针,用该指针读取文档类中的数据,根据索引值在视图中依次绘制出线段、节点、文字注释、文字盒和元器件。获取和显示原理图的过程如图 1 所示:

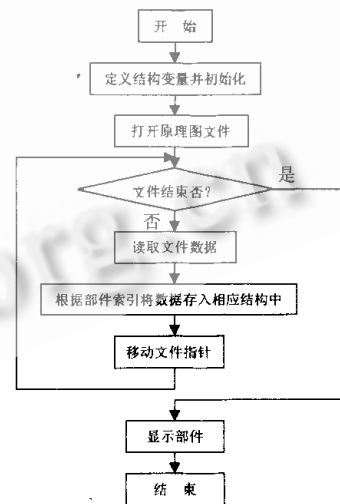


图 1 程序流程图

### 参考文献

- [1] 《多层自动布线印制板的设计与实现》胡万帆 学苑出版社 1995.10
- [2] Tango 和 Protel 原理图文件结构分析 计算机世界月刊 1995.9

(来稿时间:1998年10月)