

【自动化技术】

基于 DXP 的 PCB 无核变压器设计

周党培^a, 陈业仙^b

(五邑大学 a. 物理实验中心; b. 教务处, 广东 江门 529020)

摘要: 概述了 DXP 软件的运行机制和二次开发功能, 提出了一种利用 DXP 脚本开发 PCB 螺旋线圈绘制工具的方法, 实现了具有不同形状、线宽、线间距以及圈数的螺旋线圈的自动绘制功能, 并把脚本程序挂载到菜单栏。实验结果表明, 所提方法绘制的 PCB 螺旋线形状规则、线条平滑, 具有可编辑性, 为设计和制作 PCB 无核变压器提供了极大的方便。

关键词: PCB 无核变压器; 无线充电; DXP 软件; PCB 螺旋线圈

中图分类号: TN41

文献标识码: A

文章编号: 1006-0707(2011)09-0102-03

移动电子产品如手机、MP3、笔记本电脑等的普及, 使充电器几乎成了每个家庭中使用频率最高的电子设备。然而目前的充电器使用电源线与被充电设备相联, 携带极不方便, 加上规格众多, 互不通用, 造成极大的浪费。因此, 通用性更强、使用更方便的无线充电技术成为了近年来人们研究的热点^[1]。目前, 近距离小功率无线充电技术的研究已经取得了一些进展, 利用该技术制造的充电器由 2 部分组成: 电能发射装置, 用于把电网电能转换成电磁场, 把能量在小范围内无线发射出去; 电能接收装置, 安装在被充电的手机、MP3 或笔记本电脑内部, 用于接收电磁能并将其转换为电能对电池充电。接收装置要求体积小、效率高、通用性强以及具有抗干扰能力。通过研究人们发现, 利用电磁感应和电磁共振的方法可以实现近距离较高效率的电能传输, 在较高的工作频率下, 采用无核的 PCB 变压器作为发射和接收装置是可行的^[2]。PCB 无核变压器实质上是由 2 组或多组互相独立的平面螺旋线圈构成, 他们通过空间介质实现耦合, 与传统的通过铁氧体磁芯耦合的高频变压器相比, 具有易于设计、可分离、体积小、成本低、趋肤效应小、电流密度大、一致性好等优点^[3], 可以满足近距离小功率无线电能传输的要求。然而, 目前流行的电路板设计软件均没有直接绘制 PCB 螺旋线圈的工具, 手工绘制十分困难且效果不好, 因此本文提出了一种利用 DXP 软件的脚本功能开发 PCB 螺旋线绘制工具的方法, 为设计和制作 PCB 无核变压器提供了极大的方便。

1 DXP 的运行机制与脚本功能

DXP 是 Protel 软件的新版本, 是澳大利亚 Protel Technology 公司研制的普及型电路辅助设计软件, 他集强大的设计

能力、复杂工艺的可生产性、设计过程的管理于一体, 可完整地实现电子产品从电学概念设计到生成物理生产数据的全过程, 以及这中间的所有分析、仿真验证。DXP 还提供了一个基于脚本的二次开发接口, 以满足不同用户的特殊要求^[4]。

DXP 是一个 Client/Server 结构的应用程序, 主程序提供了用户接口, 所有应用程序如原理图和印刷电路板编辑器等, 都是作为主程序的服务器来使用的。服务器可以看作主程序的插件, DXP 的所有功能都是由这些服务器提供的, 其运行机制如图 1 所示。

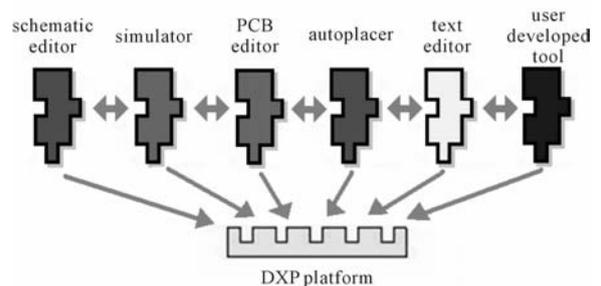


图 1 DXP 的运行机制

DXP 提供了一个精致的宏语言服务器, 可以创建在任何服务器上工作的宏脚本文件。宏文件可以运行任何服务器上的进程。DXP 的脚本系统支持 DelphiScript、Visual Basic 和 Jscript 语言, 无需外部开发工具即可编写和调试脚本, 采用脚本可自动执行简单的重复任务, 生成定制报告, 或者直接控制设计对象等。使用脚本与 DXP 进行接口有 2 种方法。第 1 种是进程加参数调用, 这种方法与使用标志菜单、工具和快捷键相同; 第 2 种方法是通过应用编程接口 (applica-

收稿日期: 2011-07-26

作者简介: 周党培(1971—), 男, 硕士, 实验师, 主要从事应用物理与电光源驱动研究。

tion programming interface),这种方法更加快捷、灵活,功能更强。

2 PCB螺旋线的绘制方法和程序

螺旋线的绘制可以从中心开始,按顺时针或逆时针方向螺旋式向外展开,他实质上是由许多段很短的直线段连接而成的,在绘制时需要确定几个参数,即线间距、线宽、步长(段数)、圈数、绕向等^[5]。其中段数指的是围成螺旋线1圈的多边形的边数,边数越大则螺旋线的形状越接近于圆形,实际使用中,要根据线圈的最大外径来选取。段数取定后,步长可以通过程序自动计算。

为了方便使用,通过 DelphiScript 编制简单的对话框提供用户输入数据的界面,如图2所示。

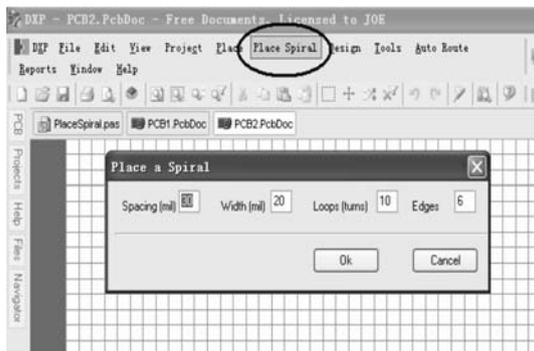


图2 程序的菜单项加载及数据输入对话框

图2中:Spacing为线间距,Width为线宽,单位均为英制(mil);Loops为圈数,Edges为段数。源程序如下:

Interface (* 界面定义 *)

Type

TPlaceSpiral = Class(TForm)

eSpacing:TEdit; eWidth:TEdit; eLoops:TEdit; eEdges:TEdit;

bOk:TButton; bCancel:TButton;

Label1:TLabel; Label2:TLabel; Label3:TLabel; Label4:TLabel;

procedure fDataCreate(Sender:TObject);

procedure bOkClick(Sender:TObject);

procedure bCancelClick(Sender:TObject);

End;

Var (* 全局变量定义 *)

PlaceSpiral:TPlaceSpiral;

Board:IPCB_Board;

WorkSpace:IWorkSpace;

x0,y0,stp,growth:Double;

p,p0,pi,xc0,xc1,yc0,yc1,spacing:Double;

loops,trackwidth:Integer;

Implementation

Procedure TPlaceSpiral.fDataCreate(Sender:TObject); (* 输入数据 *)

Begin

eSpacing.Text := 30; eWidth.Text := 20;

eLoops.Text := 10; eEdges.Text := 6;

End;

Procedure TPlaceSpiral.bOkClick(Sender:TObject); (* OK按钮处理 *)

Begin

ResetParameters;

pi := 3.141592654;

x0 := 3000; y0 := 3000;

spacing := eSpacing.Text; spacing := spacing/cos(pi/6);

loops := eLoops.Text; loops := loops * 2;

trackwidth := eWidth.Text;

stp := pi/eEdges.Text * 2;

growth := spacing/pi * stp/2;

Close;

End;

Procedure TPlaceSpiral.bCancelClick(Sender:TObject); (* Cancel按钮处理 *)

Begin

Close;

End;

Procedure PlaceAPCBTrack(X1,Y1,X2,Y2,Layer,Width);

(* Track 放置子程序 *)

Var

Track:IPCB_Track;

Begin

Track := PCBServer.PCBOBJECTFactory(eTrackObject,eNoDimension,eCreate_Default);

Track.X1 := MilsToCoord(X1); Track.Y1 := MilsToCoord(Y1);

Track.X2 := MilsToCoord(X2); Track.Y2 := MilsToCoord(Y2);

Track.Layer := Layer; Track.Width := MilsToCoord(Width); Board.AddPCBObject(Track);

End;

Procedure RunPlaceSpiral; (* 画螺旋线主程序 *)

Begin

WorkSpace := GetWorkSpace;

If WorkSpace = Nil Then Exit;

WorkSpace.DM_CreateNewDocument(PCB);

If PCBServer = Nil Then Exit;

Board := PCBServer.GetCurrentPCBBoard;

If Board = Nil then exit;

PlaceSpiral.ShowModal;

For p0 := 0 To loops * pi/stp + 1 Do

Begin

p := p0 * stp - stp;

xc1 := (cos(p) * spacing); yc1 := (sin(p) * spacing);

```
spacing: = spacing + growth;
If p0 < >0 Then
Begin
PlaceAPCBTrack(x0 + xc0, y0 + yc0, x0 + xc1, y0 + yc1, eTo-
pLayer, trackwidth);
End;
xc0: = xc1; yc0: = yc1;
End;
ResetParameters;
AddStringParameter( 'Action', All);
RunProcess( PCB:Zoom);
End;
End.
```

3 脚本的运行和实验结果

3.1 制作运行脚本的菜单

DXP 运行脚本的方法有 2 种。第 1 种通过 DXP 主菜单中的 Run Script 对话框调用,这种方法常用于临时使用脚本的情况,但对于要定期使用的脚本,最好的办法是把脚本与菜单和工具栏按钮结合。具体步骤如下。

- 1) 打开一个 PCB 文件,双击菜单栏空白处打开 Customizing PCB Editor 对话框;
 - 2) 点击 New 按钮,打开 New Command 对话框,如图 3 所示;
 - 3) 在 Process 栏点击 Browse 找到 ScriptingSystem; Run-Script 选项;
 - 4) 在 Parameters 参数栏填写脚本项目文件名及程序名,如 ProjectName = E:\PlaceSpiral\PlaceSpiral. Prjscr | ProcName = PlaceSpiral > RunPlaceSpiral;
 - 5) 在 Caption 栏填写命令的名称,如 Place Spiral;
 - 6) 点击 OK 返回后将会在 Custom 的 Command 列表中显示出上述名称,把他拖放到菜单栏即可,如图 3 所示。
- 以后在使用时只要在菜单栏直接点击 Place Spiral 即可打开对话框,输入参数自动生成对应的螺旋线圈。

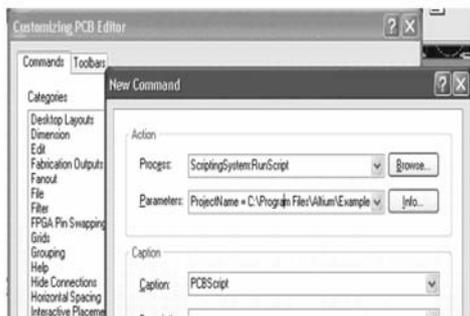


图 3 菜单加载设置

3.2 不同形状螺旋线圈的绘制

如图 4 所示,分别生成六边形、四边形和圆形的螺旋线圈。具体参数为:线间距 40mil,线宽 25mil,圈数 23,段数依次为 6,4 和 60。

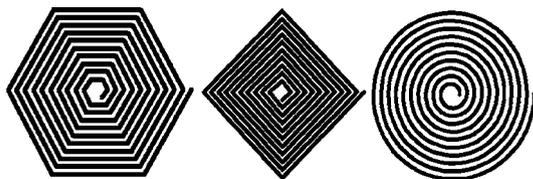


图 4 不同形状的螺旋线圈

由图 4 可见,生成的图形形状规则、线条平滑,基本上达到印刷电路板生产工艺的要求。通过对话框改变不同的参数设置,可以十分方便地生成不同形状、线宽、间距和圈数的螺旋线圈。实物模型如图 5 所示。

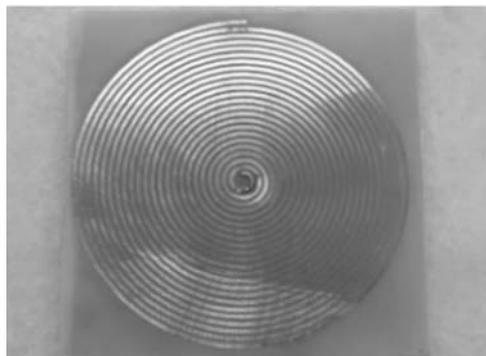


图 5 PCB 螺旋线圈实物模型

3.3 螺旋线圈的再编辑和变压器的实现

由于这些图形实质上是由许多直线段分段连接而成的,所以可以根据实际需要进行再编辑。可以把螺旋线圈分割成几组不同匝数的绕组,在两端放置焊盘或过孔,即可实现无核 PCB 变压器的设计。如图 6 所示的 PCB 变压器 3 个绕组分别为 4,3,3 匝。



图 6 平面 PCB 变压器