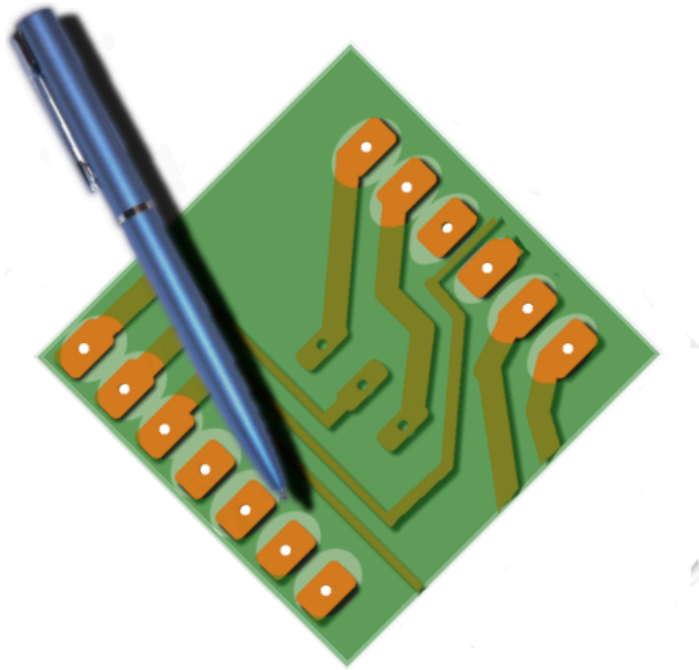


FidoCadJ 0.24.1

用户手册

Davide Bucci



January 4, 2013

本手册的发布遵守创意公共许可证（2.5 版本及以上）。以下网址可以查阅到许可证的完整内容：

<http://creativecommons.org/licenses/by-nc-nd/3.0/>

您可以自由地复制、分发、交流或展览、演示、印刷并使用本手册，但必须接受下列条件：

署名 您必须按照作者或许可人指定的方式对手册进行署名（而不是用其它方式表明作者或许可人授权给您或授权给您使用本手册）。

非商业 您不可以把本手册用于商业。

禁止衍生 您不可以修改、转换本手册或建立基于本手册的作品。

如果您得到著作权人（Davide Bucci）的同意，可以取消上述任何限制条件。

本手册中引用的所有商业名称、标志、商标都归属于注册它们的所有者。

摘要

本手册是 FidoCadJ 的官方用户手册。首先简单介绍 FidoCadJ 的起源和历史，接着描述它的基本使用方法。目标是让读者学会使用 FidoCadJ 绘制简单的电路原理图以及相应的印制电路板图。然后，详细描述 FidoCadJ 的格式（也可以说是 FidoCad 的格式）。最后，我们将给出在目前比较流行的操作系统（Linux、MacOSX 和 Windows）平台上下载和安装 FidoCadJ 的方法。

鸣谢

许多人从软件发布的第一版就开始使用了，并提出他们的意见帮助我改进此软件。在此，我要感谢 it.hobby.elettronica 新闻组的所有参与者，是他们进行了非常富有成效的讨论。

感谢 Stefano Martini 耐心地在 Linux 系统平台上对本软件做非常仔细的测试，他是一个非常细心的 alpha 版本/beta 版本的测试员。同时，我也要感谢 Olaf Marzocchi 和 Emanuele Baggetta 在 MacOSX 平台上做的测试工作。

感谢 F Bertolazzi 提出关于本软件使用方面的非常有用的建议，他也集成了与 CadSoft Eagle 兼容的库，该库用于把 FidoCadJ 的绘图输出给 Eagle。感谢 Celsius 测试 PCB 绘图及相关元件库的功能。感谢 Andrea D'Amore 提出关于苹果机上 FidoCadJ 外观方面的建议，主要应用始于 0.21.1 版本。感谢 Roby IZ1CYN 关于库的有效论述，见本手册的 [A.2](#) 章节（关于 Linux 系统上的安装和运行 FidoCadJ 的方法）。感谢 Quaqua LookAndFeel（来自 Werner Randelshofer），使得苹果机用户可以享用到一个与本地系统整合地很好的 FidoCadJ。同时，Werner 还有一个贡献是提出许多有效的关于 FidoCadJ 用户界面的建议，谢谢！

本手册已经由 Pasu 翻译成英文，感谢他付出的工作。我也要感谢 Miles (QH007)，对本手册进行仔细校对和提出一些有效的评论。

2010 年 4 月，FidoCadJ 整合进了有名的意大利网站 www.electroportal.it。它默默地运行在服务器端，自动转换论坛用户张贴的绘图。我要表达对版主 Zeno Martini、站长 Nicoli Martini 和该网站上其他所有用户的感激之情，是他们提出了许多实用的想法，例如 FidoCadJ 的批处理（这是目前跟踪到的一个很有潜力的发展方向，应该进行全面探讨）。最近，FidoReadPHP 类已经完成，用于在 PHP 服务器上读取和解析 FidoCadJ 绘图。感谢 Arniek 和 Sstrix，已经把它应用到流行的 www.grix.it 网站上，我要赞美他们！

FidoCadJ 许可证

版权 © 2007-2012 Davide Bucci davbucci@tiscali.it

本软件是自由软件：您可以在自由软件基金会公布的 GUN 通用公共许可证的条款下对它再发布和/或修改。

为了能够发挥软件实用价值而发布本软件，但没有任何保证，甚至没有商业或适用于特定用途的隐含保证。

您收到本软件的同时应该收到 GNU 公共通用许可证的一份拷贝。如果没有，请查阅：


<http://www.gnu.org/licenses/>

目录

1. 介绍	1
1.1. FidoCadJ 的理念	1
1.2. FidoCadJ 的历史	2
1.3. FidoCadJ 的未来	3
2. FidoCadJ 绘图	6
2.1. 绘图工具	6
2.2. 一张简单的原理图	9
2.3. 层	15
2.4. 网格	15
2.5. 一张简单的 PCB 图	16
2.6. 使用标尺	22
2.7. 箭头和线条样式	23
2.8. 导出	25
2.9. 命令行选项	25
2.10. 库管理	29
3. 绘图格式、宏元件和库	31
3.1. 绘图文件头	31
3.2. 坐标系统	31
3.3. 绘图元件	32
3.4. FidoCadJ 扩展	38
3.4.1. 层设置	39
3.4.2. 电气连接点设置	40
3.4.3. 线条宽度	40
3.5. 语法容错	40
3.6. 库文件格式	41
3.7. 标准库	42

4. 总结	43
A. 操作系统上的特殊性	44
A.1. MacOSX 系统	44
A.1.1. 下载并运行 FidoCadJ	44
A.2. Linux 系统	45
A.2.1. 通用的终端安装方法	45
A.2.2. 图形安装方法	46
A.3. Windows 系统	48
A.3.1. 下载并运行 FidoCadJ	48

插图

1.1. www.electroyou.it 上我的帖子的一部分。您只需要用鼠标点击一个图标就可以缩放查看原理图。点击另一个图标就可以立即得到绘图代码，并可以把代码复制/粘贴到 FidoCadJ 中进行修改。	4
1.2. www.grix.it 论坛上整合了 FidoCadJ 绘图帖子的示例，您只需要一次鼠标点击就可以取得绘图代码。	5
2.1. 运行在 MacOSX Tiger 系统上的 FidoCadJ 的典型界面。附录 A介绍苹果机操作系统的特性。	7
2.2. FidoCadJ 的 Metal LookAndFeel	7
2.3. 在加载库中快速查找。	9
2.4. FidoCadJ 绘图的文本元件参数设置对话框。	10
2.5. 参考原理图：NPN 三极管的一个电路镜像。	10
2.6. 首先，我们画一对三极管。	11
2.7. 选择并用  键翻转左边的晶体管。	12
2.8. 我们画得太靠近绘图区域上边界了：让我们选择所有已绘制的元件并把它们移动到中央。	12
2.9. 电路图基本完成了。	13
2.10. 最终的电路图。	14
2.11. 一个非常简单的由 NPN 晶体管构成的共射极放大器电路板。	17
2.12. 最主要的元件已布置到板上。	17
2.13. 用多边形画电源线。	18
2.14. 画其它的 PCB 导线。	19
2.15. PCB 基本完成。	19
2.16. 画上丝印层，工作完成。	20
2.17. 在 ISO-UNI A4 纸上（翻转）打印 PCB 图的效果。	21
2.18. 鼠标右键点击并拖拉，出现 FidoCadJ 标尺。	23

插图

2.19. 电路图 (Antoniou 通用阻抗转换器 (GIC)), 其中用到一些 FidoCadJ 的扩展。	24
2.20. 图 2.19 中的贝塞尔曲线的参数对话框。	24
2.21. 在 MacOSX 上程序的外观, 使用 Motif LookAndFeel。	29
3.1. 图 2.19 在 Windows 系统上, 用 FidoCad 打开的绘图。	39
A.1. Ubuntu 8.04 系统, 文件权限设置。	47
A.2. Ubuntu 8.04 系统, 设置执行程序使用的 Java 虚拟机。	48

表格

- 2.1. FidoCadJ 绘图命令汇总表。最左边的“快捷键”栏给出快速选择该命令的按键。在元件放置模式时，右键点击任何已绘制的元件可以打开相应的属性窗口。 8
- 2.2. FidoCadJ 输出文件格式列表。 26
- 3.1. 表示线条类基元件的端点箭头的 a 参数的含义。 33
- 3.2. 表示箭头样式的 b 参数的含义。 33
- 3.3. 文本样式字段位的含义。 34

1. 介绍

在本章中，我们将简单介绍 FidoCadJ，特别描述它的理念以及发展史。

1.1. FidoCadJ 的理念

FidoCad（后面不带 J）是一个矢量绘图软件，尤其适用于绘制电路原理图（以下简称为：原理图）和印制电路板图（以下简称为：PCB 图）。二十世纪九十年代后期，它在意大利 Usenet 社区特别流行。

FidoCad 可以从 Lorenzo Lutti 的网页上自由下载（有一个 Windows 系统上运行的意大利语版本的软件）：

<http://www.enetsystems.com/~lorenzo/fidocad.asp>

其生成的绘图文件是一个非常紧凑的文本描述。对于在文本信息中加入绘图信息很方便，例如那些使用文本格式传输信息的 Usenet 群组正需要该功能。

可惜的是，FidoCad 只能运行在 Windows 系统上。Linux 系统的用户需要借助 WInE 才能运行它。其它系统的用户，像我（我使用的是 MacOSX 系统）需要寻找另外的解决办法。因此，我决定对 Usenet 社区做一点小小的贡献，编写 FidoCadJ（这里，后面加了个 J）。这个软件是用纯 Java 编写的，是一个完全平台的软件。FidoCadJ 可以用于显示和修改 FidoCad 格式的绘图。

只要使用过 FidoCad 的用户，应该很快就能熟悉并使用 FidoCadJ，因为许多命令和操作非常类似于原软件。据我了解，除了一些细节外，当前（在本手册编写时）的 FidoCadJ 几乎完全与原 FidoCad 兼容。一直以来，我都是以追求尽可能完全兼容为目标。然而，近来为了满足新用户的需求，对原格式进行了一些扩展。

FidoCadJ 不同于 FidoCad 的其中一项特性就是它提供了导出功能。因为我是一位 L^AT_EX 用户，我决定包含一些矢量格式的导出，其中包括 Encapsulated PostScript (EPS) 格式。当然，FidoCadJ 可以导出非常有名的 PDF 格式。另一种用于电路原理图的文件格式是 CadSoft 的 Eagle 脚本，从 FidoCadJ 的 0.21 版本开始，可以导出该格式。这样，用 FidoCadJ 绘制的原理图可以导出给 Eagle 使用。附录 A 简单描述了在各个常用操作系统上安装 FidoCadJ 的方法。

1. 介绍

1.2. FidoCadJ 的历史

我对电子电路一直都很感兴趣。当我开始关注几个电子电路方面专业的意大利 Usenet 新闻组时，我注意到许多原理图是用 Windows 系统上的 FidoCad 画的。这样就避免了使用笨拙的 ASCII 绘图。由于我已经多年不使用 Windows 了，那对我来说几乎是不可能看见它们的。我想试着做些事情来解决这个问题。

开始，我学习 FidoCad 的文件格式并写了一个 Java 脚本 FidoReadJ，它能解析电路图并显示在网页中。接下来，我总是到处寻找旧帖子、网页，对其中的 FidoCad 文件做反向工程，做了很多。我下载过 FidoCad 原码，它是由 Lorenzo Lutti 用 C++ 编写的，写得非常漂亮和整洁。

大约是在 2007 年的 3 月做了这些事情。几个月后，脚本就上线了，并且一部分社区人士对它进行了测试，主要集中在 it.hobby.elettronica 和 it.hobby.fai-da-te 网站。

¹

由于我已经实现了一个 FidoCad 文件格式的解析器，自然很有兴趣去完成一个完整的编辑器。在 2008 年的 6、7 月，分步完成了大部分的工作。FidoCadJ 是一个完全重写的程序，并不是对 Windows 的 FidoCad 程序的改写或移植。

选择使用 Java 实现程序，是由于最近几年我转换使用了多种操作系统。我不再愿意花费时间和精力在不兼容的东西上了。如果只要求运行在 MacOSX 系统上，努力学习使用 Cocoa 框架，应该会获得更好的效果，但它会使 FidoCadJ 完全失去兼容性。我不是一个电脑迷，我愿意花时间在编程上面完全是因为我对电子电路的爱好。事实上，只要简单分析 FidoCadJ 源码就可以看出我不是一个纯 Java 编程和纯面向对象编程的程序员。相信实现该程序会有多种更实用和更优雅解决方案。

然而，软件并不在于选择哪种语言编写，而在于最终用户的使用体验。为此，我总是听取您的建议，以便未来更好地发展这个项目。概括来说，Java 并不是解决各类问题的完美选择或方案。然而，我确信它的负面影响大多来自写得不好的应用程序，那些不能很好地与用户界面集成的应用程序。

由于我编程技能有限，不能做到尽善尽美，我的目标是确保 FidoCadJ 不会沦为一个质量差的应用程序。为此，更欢迎提出任何 BUG 报告和程序应用方面的建议。

2009 年 11 月，我在 SourceForge 上开了一个项目致力于 FidoCadJ。那里，您可以下载到所有的可执行文件、手册和源码。您可以积极参与到 FidoCadJ 源码的开发工作中，这时，您需要使用 Subversion(SVN) 工具或 SourceForge 的 SVN browser。项目地址：

<http://fidocadj.svn.sourceforge.net/viewvc/fidocadj/>

¹FidoReadJ 仍然可以从这个地址获得：

<http://davbucci.chez-alice.fr/index.php?argument=elettronica/fidoreadj/fidoreadj.inc>

顺便说下，我认为新做一个程序解决方案要比在原程序中增加声明不同与 Windows 系统的部分更好。

其实，在 1993 年左右我就开始试着写一个 2D 矢量绘图系统。

如果您想来帮助我和 FidoCadJ，请不要担心：您不需要是一个专业的 Java 程序员。您可以用一种新的语言翻译用户界面或手册、仔细检查当前手册中的矛盾和错误；您还可以从事标准库的工作... 细想一下，我给予 FidoCadJ 的时间中，实际只有一半是用于编程的，其余时间都用于回答用户的问题、编写和改善文档等等。在 SourceForge 上，您可以加入论坛，写一个程序评论、提出改进建议或给一个 BUG 报告。项目首页：

<http://sourceforge.net/projects/fidocadj/>

1.3. FidoCadJ 的未来

2010 年 4 月，我偶然（几乎是意外）登录了一个非常有名的意大利网站 www.electroyou.it。有个用户 Giancarlo Boletti，提出直接在论坛上集成一个原理图捕捉工具，类似于 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 的公式。我在论坛上落了款并主动提出愿意进行合作。这样，由于他的发言引出了 FidoCadJ。经过几天与站长（他是一个很不错的站长）的交流，系统准备就绪了：在有代码形式绘图的帖子中，使用一对标签实现简单地复制和粘贴功能。论坛软件调用服务器上的 FidoCadJ 获得代码的绘图。然后把这个绘图显示在论坛的帖子中，并且仍然可以取得绘图的代码。这样，如果他只是浏览而不是修改的话，不需要安装任何插件，并且他还可以通过很少的鼠标点击动作取得绘图代码。

Figure 1.1 显示了 www.electroyou.it 上我的帖子的一部分。这个系统是如此强大和灵活，以至于用户对它表现出的热情令我感到了前所未有的惊讶！²

FidoCadJ 在 www.electroyou.it 上取得了成功，激发了其它平台上实现类似系统的一些需求，特别是 www.grix.it，另一个众所周知的意大利电子专业网站。该网站的服务器无法运行 Java 程序，于是就有了 FidoReadPHP 类。它可以运行在 PHP 服务器上，取得 FidoCadJ 绘图代码的图形。FidoReadPHP 项目是开放源码的，可以从 SourceForge 上获得：

<https://sourceforge.net/projects/fidoreadphp/>

尽管 PHP 的图形能力与 Java 相比是相当有限的，但从获得的图形效果来看还是可以的，类似于从 www.electroyou.it 上获得的图形³。Figure 1.2 显示了一个 FidoReadPHP 绘制的图形。

我认为 FidoCadJ 的未来，不是变得更加复杂，发展成一个完整的电子 CAD。发

²如果您能阅读意大利文，这里有一篇我写的文章：


<http://www.electroyou.it/darwinne/wiki/fidocadj>

³再次，如果您能阅读意大利文，这里有篇文章：

<http://www.grix.it/viewer.php?page=9335>

1. 介绍

[12] Re: Orologi e orologiai

di  DarwinNE il 25 mag 2011, 1:36

“ TardoFreak ha scritto:
Beh ... ecco ... timidamente potrei offrirti di sviluppare il firmware (a disposizione di tutti) per fare l'orologio di EY. 😊

Penso proprio che i ragazzi abbiano intenzione di lasciare fuori i microcontrollori, per fare un ottimo esercizio di stile. Perché non utilizzare un contatore analogico? Anni fa ne avevo realizzato uno per un tracciacurve, è un circuito un bel po' critico, ma abbastanza affascinante. Penso che parte della criticità venga dal fatto che ho simulato degli unijunction programmabili con una coppia di transistor NPN+PNP (Q2 e Q3, Q7 e Q8). Inoltre, i generatori di corrente che ho adottato sono probabilmente migliorabili.

Sorgente FidoCadJ | Ingrandisci | Cos'è Fidocad

DarwinNE
2 6 10
G.Master EP

Messaggi: 275
Iscritto il: 18 apr 2010, 9:32
Località: Grenoble - France

Figure 1.1.: www.electroyou.it上我的帖子的一部分。您只需要用鼠标点击一个图标就可以缩放查看原理图。点击另一个图标就可以立即得到绘图代码，并可以把代码复制/粘贴到 FidoCadJ 中进行修改。

1.3. FidoCadJ 的未来

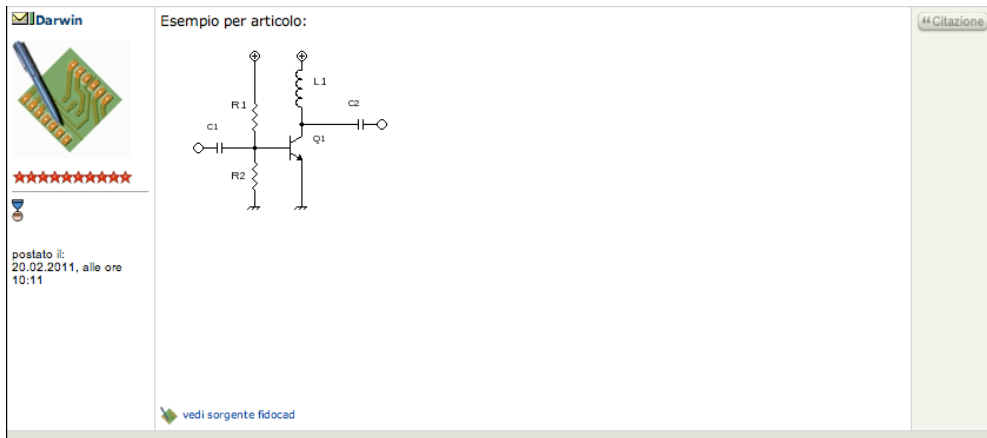


Figure 1.2.: www.grix.it论坛上整合了 FidoCadJ 绘图帖子的示例，您只需要一次鼠标点击就可以取得绘图代码。

展与讨论组和论坛的整合可能更加理想，经验表明了这种可能性，而且已经激起了用户的热情。

2. FidoCadJ 绘图

专业的苹果用户会注意到窗口的
布局很合理！

对那些原来用过矢量绘图软件的用户来说，FidoCadJ 应该是凭经验就能使用的软件。图 2.1 显示了本软件在 MacOSX 系统上运行的一个截屏。在其它系统上运行的外观（例如，图 2.2 显示了在 Sun/Oracle 系统上运行的外观，设置为 Metal LookAndFeel）可能会有些细微的差别，但操作理念是一致的。我们将会发现该绘图软件的特点是利用基本图形元件（以下简称为：基元件）组合成一张绘图。

2.1. 绘图工具

设计这种操作方式的灵感来自于
七十年代非常流行的老式真空管
收音机的开关方式。

我们可以在工具栏（位于窗口顶部）上找到常用的工具，用来创建和编辑绘图。表 2.1 简要汇总了这些工具的功能、命令和关联动作。你会发现，一旦按下一个工具栏按钮，它会保持在按下状态，直到按下另一个工具栏按钮。我们可以从工具栏上选择需要使用的基元件¹。靠工具栏的右边，有个下拉菜单显示了当前的工作层（见 2.3 章节，有更多的说明）。

工具栏可以部分定制。特别是，我们可以定制按钮的外观（显示图标或显示图标和文本；显示图标也有两种方式可供选择）。我们可以通过菜单“文件/选项”来改变这些设置²。任何设置的改变都要重新启动应用程序后才会生效，以后我们可能经常要改变一些设置，会证实这一点。图 2.1 和图 2.2 显示的工具栏（位于窗口标题栏的下方）是设置成文本和小图标的显示方式。工具栏的第二排靠左边，我们可以看到缩放比例设置和按钮“适合窗口”、“显示网格”和“对齐到网格”。第一个“适合窗口”按钮，可以自动调节到最合适的比例，以在窗口中最大化显示整张绘图。第二个“显示网格”按钮，可以切换网格可见或不可见。第三个“对齐到网格”按钮，可以设置在绘制元件时是否对齐到附近的网格。如果需要把元件对整齐，您可以按住 **Alt** 键的同时使用方向键来移动选中的元件。

窗口右边有一个树形列表，列出应用程序加载的元件库（库中的元件称为：宏图形元件，以下简称：为宏元件）。要在绘图中加入一个的宏元件，我们只需要从列表

¹更多关于绘图元件的介绍请见 3.3 章节。

²MacOSX 系统例外，在 FidoCadJ 菜单的“Preferences”里改变设置。

2.1. 绘图工具

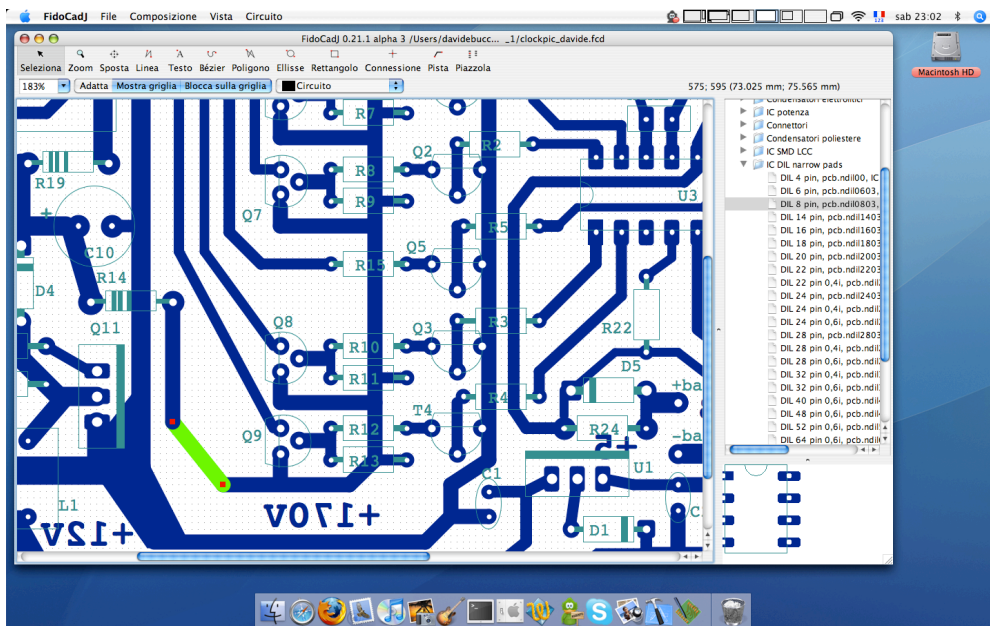


Figure 2.1.: 运行在 MacOSX Tiger系统上的 FidoCadJ 的典型界面。附录 A介绍苹果机操作系统的特性。

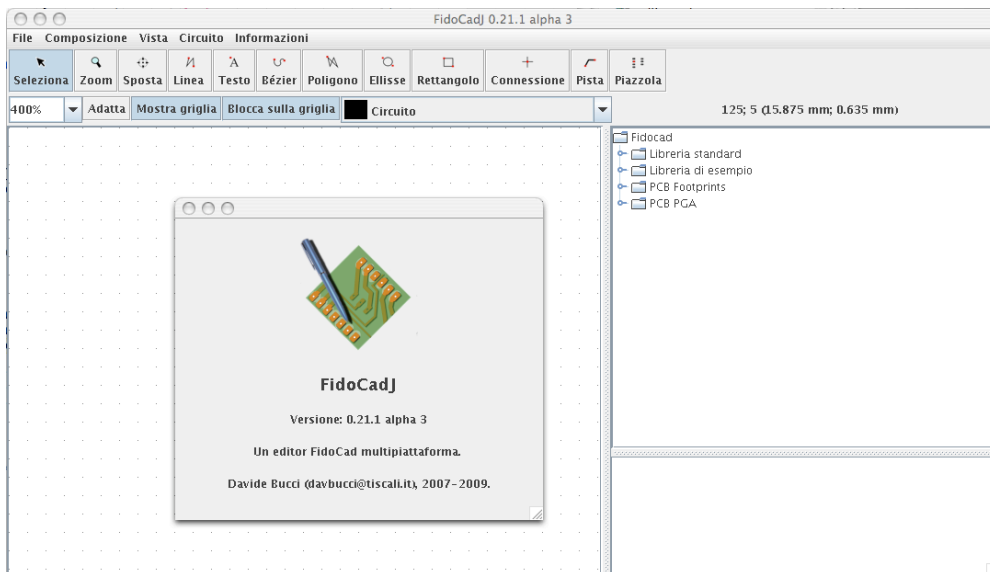


Figure 2.2.: FidoCadJ 的 Metal LookAndFeel

2. FidoCadJ 绘图














快捷键	命令	使用方法
A 或 空格	 选择	选择一个或多个元件。按住 Control 键 (MacOSX 系统中是 Command 键) 进行多选, 或从多选中去掉一个。也可以框选多个元件。按 R 键旋转选择的元件; 按 S 键翻转选择的元件。双击一个元件可以修改它的属性。
	 缩放	左键点击放大; 右键点击缩小。
	 浏览	在图纸上左键点击并移动鼠标可以移动绘图。
L	 直线	画一条或多条连续直线。双击或按 Esc 键完成绘制。
T	 A 文本	放置一个文本。
B	 贝塞尔曲线	画一条贝塞尔曲线。
P	 多边形	画一个多边形 (填充或不填充)。双击或按 Esc 键完成绘制。
K	 曲线	画一条开放或闭合的自然三次样条曲线。双击或按 Esc 键完成绘制。
E	 椭圆	画一个椭圆 (填充或不填充) (按住 Control 键: 画圆形)。
G	 矩形	画一个矩形 (填充或不填充)。
C	 + 连接点	画一个电气连接点。
I	 PCB 导线	画一条 PCB 导线。通过菜单“文件/选项”打开的对话框可以修改默认宽度。
Z	 PCB 焊盘	画一个 PCB 焊盘。通过菜单“文件/选项”打开的对话框可以修改默认尺寸。

Table 2.1.: FidoCadJ 绘图命令汇总表。最左边的“快捷键”栏给出快速选择该命令的按键。在元件放置模式时, 右键点击任何已绘制的元件可以打开相应的属性窗口。

2.2. 一张简单的原理图

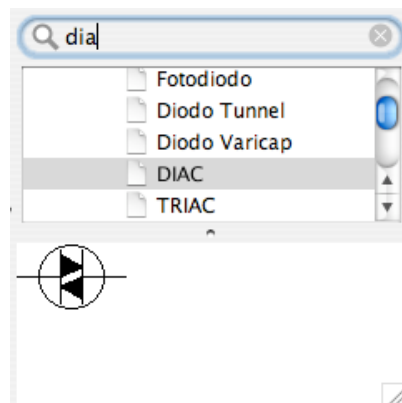


Figure 2.3.: 在加载库中快速查找。

中选择它，然后在绘图中左键点击即可。FidoCadJ默认加载库中包含绘制原理图所需使用的所有标准符号和绘制 PCB 图所需使用的大多数封装。

从版本 0.22 开始，您可以在 FidoCadJ 载入的元件库中进行快速搜索。您只需要在搜索框中键入一些字符，就会在树形列表的已加载库中进行匹配（见图 2.3），此时可以使用向上和向下按键在匹配项间移动。

图 2.4 显示了一个在选择模式时双击已绘制元件得到的参数设置对话框（这里是文本元件的参数设置对话框）。任何一个已绘制元件都可以通过这样的对话框修改所有属性（如坐标、旋转、...）。由于各种元件的设置参数不同，随着选择的元件不同，参数设置对话框显示的内容也是不同的。

2.2. 一张简单的原理图

我们将展示如何绘制一张图 2.5 所示的简单的原理图，作为使用该应用程序的一个例子。

FidoCadJ 运行起来后，我们可以通过菜单“文件/新建”来创建一个新的绘图。

围绕要建立的原理图，首先，我们要在绘图区域放置 2 个晶体管。此时，我们需要标准库中的宏元件（标准库是默认加载的，位于窗口的右侧）。所需宏元件的名称为“NPN transistor”，它在“Standard library”的“Diodes and transistors”类型中。通过左键点击宏元件的名称选择它，然后在绘图区域中的任何想要放置的位置处再次左键点击就可以绘制它（选择绘制位置时，FidoCadJ 会显示一个宏元件的预览）。到此，我们的绘图应该类似于图 2.6 所示的样子。

2. FidoCadJ 绘图

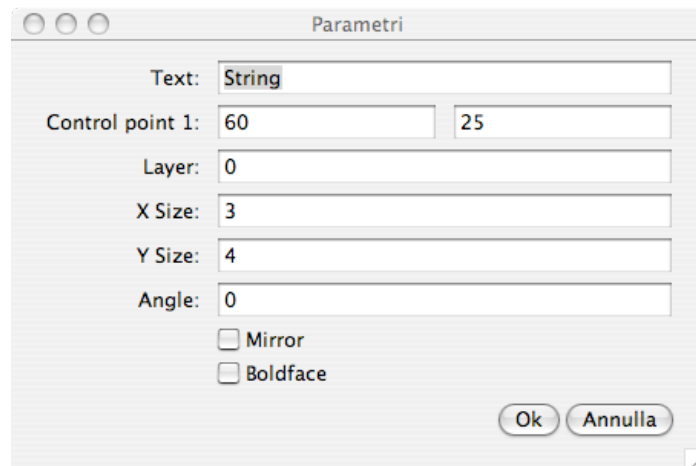


Figure 2.4.: FidoCadJ 绘图的文本元件参数设置对话框。

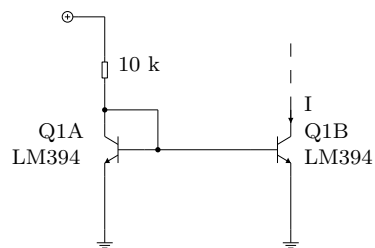


Figure 2.5.: 参考原理图：NPN 三极管的一个电路镜像。

2.2. 一张简单的原理图

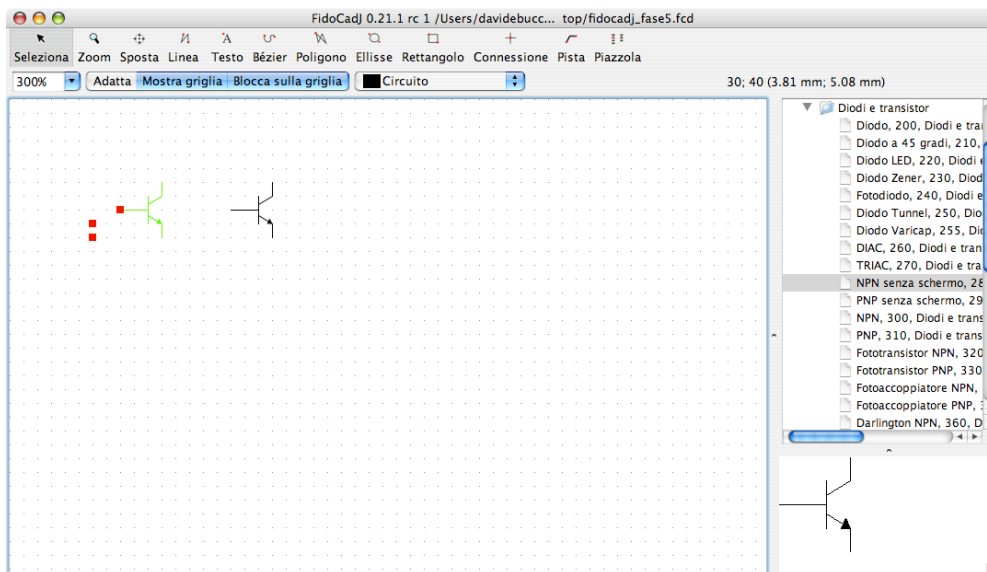


Figure 2.6.: 首先，我们画一对三极管。

我们注意到左边的双极型晶体管的方向不对。为了解决这个问题，左键点击工具栏上的“选择”工具并点击选择那个晶体管（选中后，晶体管会呈现绿色高亮显示状态，并带有 3 个红色小正方形控制点），然后按 **[S]** 键把它翻转。你也可以在选择了宏元件并在绘图中绘制前按 **[S]** 键。到此，我们的绘图应该类似于图 2.7 所示的样子。

使用工具栏上的“直线”工具，我们可以画一些电气连接线，此时，我们意识到先前所画的元件太靠近绘图区域的边界了。这个问题很容易解决：只要选择所有已绘制的元件并移动它们到合适的位置。想要移动所有已绘制的元件，首先我们需要全部选择它们（见图 2.8），选择方法：鼠标左键点击“选择”工具后移动到绘图区域左上角，按下鼠标左键并保持住，然后拖拉鼠标，此时会出现一个绿色矩形框（表示正在试图选择矩形范围内的所有元件），当矩形框覆盖了所有绘制元件后松开鼠标左键。现在，仍然处于选择模式，我们可以用鼠标左键点住任何一个已选择的元件后拖动所有选择的元件到所需的位置。

然后，我们可以继续放置其它的电路图元件，如电阻（Standard Library / Discrete devices / Resistor）和电源正（Standard Library / Basic symbols / Terminal +）。我们要旋转电源正的图标，并放置到理想的位置。再次，我们可以选择它并按 **[R]** 键旋转它。到此，我们的绘图应该类似于图 2.9 所示的样子。

我们只需要加上文本和表示电流方向的箭头就能完成原理图了。在“Standard

2. FidoCadJ 绘图

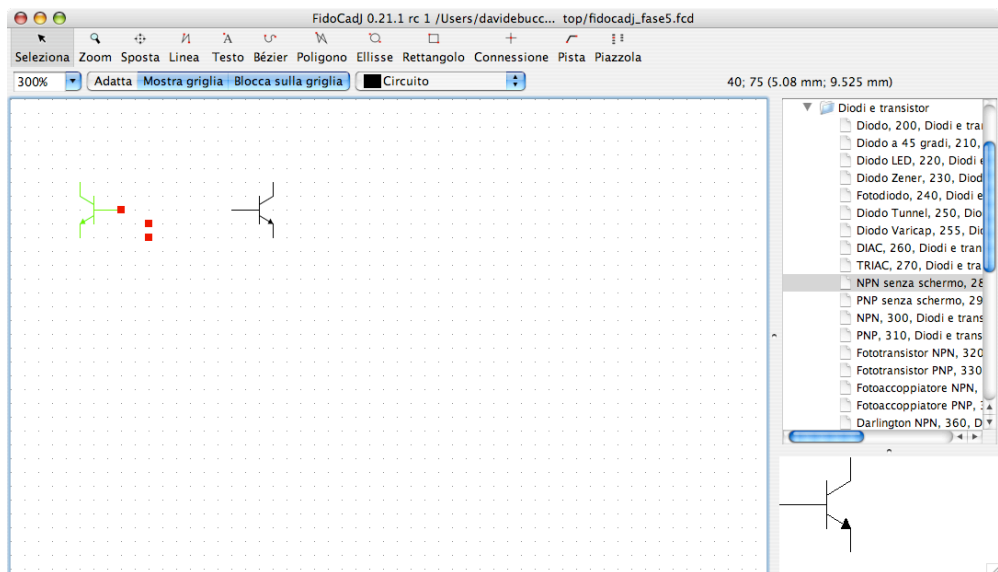


Figure 2.7.: 选择并用 **S** 键翻转左边的晶体管。

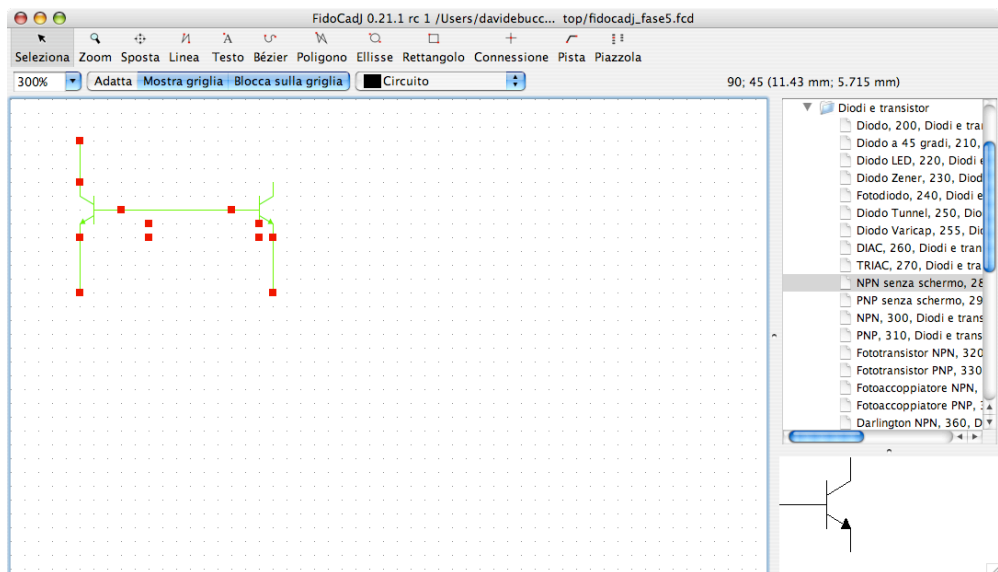


Figure 2.8.: 我们画得太靠近绘图区域上边界了：让我们选择所有已绘制的元件并把它们移动到中央。

2.2. 一张简单的原理图

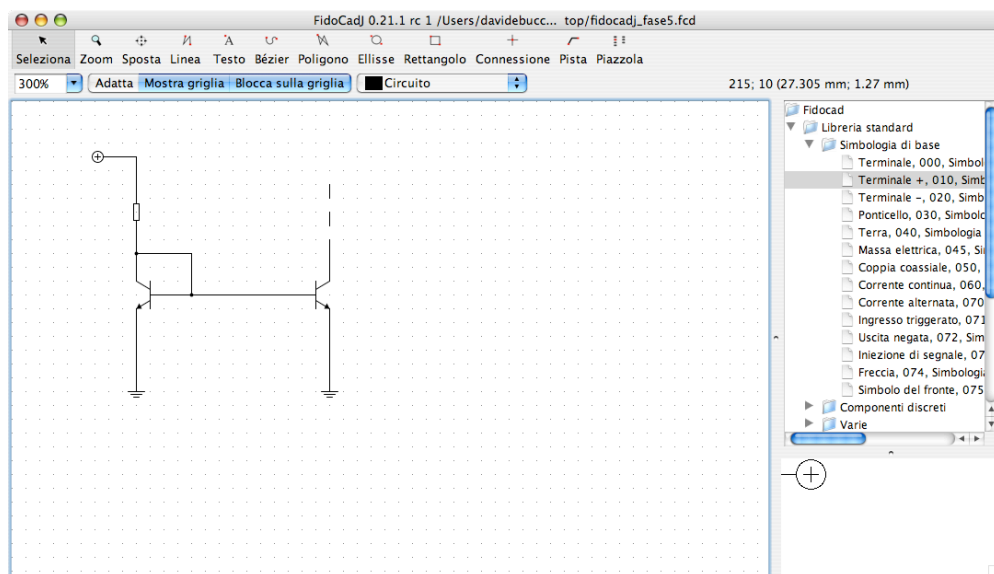


Figure 2.9.: 电路图基本完成了。

library / Basic symbols”中有个名称为“Arrow”的宏元件用于绘制箭头。我们通过左键点击工具栏上的“文本”工具，并在绘图区域中想要的位置处点击来绘制文本。绘图区域中将加入一个默认文本“String”，我们需要在选择模式下双击它并修改它的属性（见图 2.4）。晶体管（我们示例中的一对晶体管）元件的名称和编码，可以在选择模式下双击该宏元件，弹出参数窗口，其中可以设置“名称”和“编码”³。建议电路原理图中的字体尺寸为垂直方向 4 个单位，水平方向 3 个单位。完成的电路原理图见图 2.10。

出于好奇，想知道我们示例的原理图的绘图代码是什么样子的。只需要选择菜单“电路/查看绘图代码”就可以看到。现在我们可以复制原理图绘图代码并把它粘贴进电子邮件、新闻组或论坛。

```
[FIDOCAD]
MC 95 65 0 0 280
FCJ
TY 115 60 4 3 0 0 0 * Q1B
TY 115 65 4 3 0 0 0 * LM394
MC 55 65 0 1 280
```

³允许我们对宏元件或组件符号增加名称和部件号的特性是 FidoCadJ 的一个扩展，这个功能在原 FidoCad 中没有，见 3.4 章节关于兼容方面的更多信息。

2. FidoCadJ 绘图

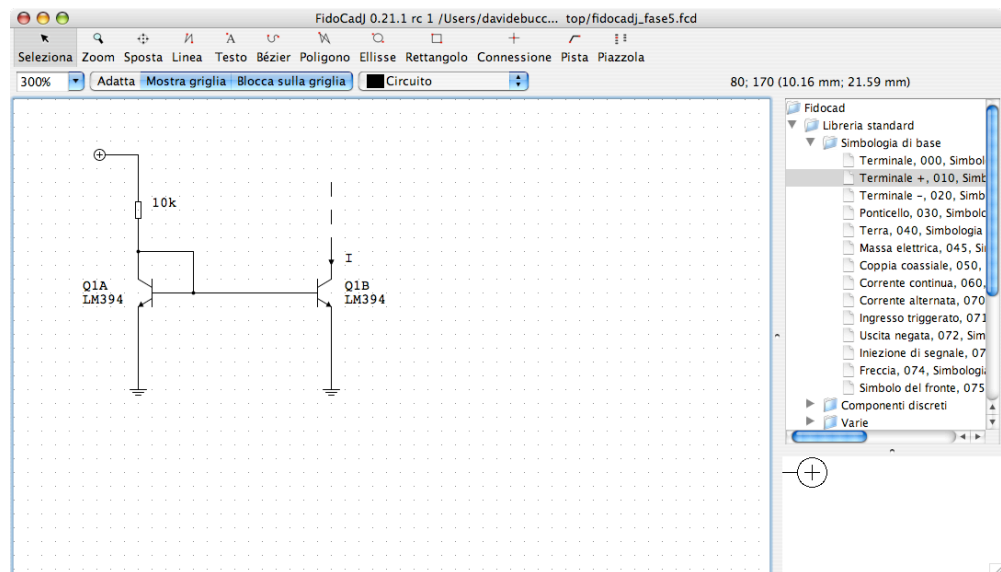


Figure 2.10.: 最终的电路图。

```

FCJ
TY 20 60 4 3 0 0 0 * Q1A
TY 20 65 4 3 0 0 0 * LM394
LI 55 65 95 65 0
LI 40 75 40 95 0
LI 110 75 110 95 0
LI 40 40 40 55 0
MC 40 30 0 0 115
LI 40 15 40 30 0
LI 30 15 40 15 0
MC 30 15 2 0 010
LI 40 50 60 50 0
LI 60 50 60 65 0
SA 60 65 0
SA 40 50 0
LI 110 45 110 55 0
LI 110 35 110 40 0
LI 110 25 110 30 0
MC 40 95 0 0 040

```

```
MC 110 95 0 0 040
TY 45 30 4 3 0 0 0 * 10 k
TY 115 50 4 3 0 0 0 * I
MC 110 50 1 0 074
```

如果您对 FidoCad 绘图的输出格式感兴趣，在看第 3 章节中的详细描述。

然而，不一定要通过菜单“电路/查看绘图代码”复制绘图代码，我们可以选择整个绘图并拷贝（选择菜单“编辑/拷贝”或按 `Ctrl+C`），然后粘贴到我们写的文本信息中，会自动以代码的形式粘贴。

FidoCadJ 可以根据设置在粘贴绘图元件时与被复制的绘图元件错开（错开方式：往右下方向移动一个网格）或不错开。软件默认设置成错开，可以明显地看出复制/粘贴的绘图元件。由于在某些情况下粘贴时不需要错开，而且错开了还有麻烦，所以在菜单“文件/选项”中可以对此进行设置。

2.3. 层

用醋酸纤维薄片的概念可以帮助理解绘图中层的概念，先在多张醋酸纤维薄片上绘图，然后把这些醋酸纤维薄片叠加在一起组合成最终的绘图。每一层用不同的颜色绘制，且可以设置为可见或不可见。这种层的方法在许多 CAD 软件中很常用，因为它可以简单的表示和管理叠加在一起的不同的绘图元件，比如绘制 PCB 图就使用了层的方法。

FidoCadJ 最多支持 16 层，层编号为 0 到 15，并且已经指定了一些常用的层：0 层为原理图层；1 层为底层导线层；2 层为顶层导线层；3 层为丝印层。其它没有指定的层可以自由使用。通过菜单“视图/层”可以指定层的名称和颜色、设置层为可见或不可见（也即打印或不打印）。

层的顺序很重要，FidoCadJ 依此绘制各层（编号相对小的层先绘制），请注意叠加层产生的覆盖作用。

2.4. 网格

FidoCadJ 绘图的逻辑单位是 $5mil$ ($0.127mm$)，不能设置半个单位。这样，任何绘图元件的坐标都是整数。这对于绘制原理图，主要是对于 PCB 图来说是一个非常好的分辨率。然而，为了便于绘图，程序允许我们设置一个较粗糙的网格，并可以迫使鼠标放置绘图元件时对齐到最近的网格。有 2 个相关的按钮“显示网格”和“对齐到网格”。“显示网格”用于切换网格可见/不可见；“对齐到网格”用于切换鼠标放

2. FidoCadJ 绘图

置绘图元件时需要/不需要对齐到网格。网格大小可以通过菜单“文件/选项”打开的对话框进行设置。

2.5. 一张简单的 PCB 图

读者会发现：开发实践中，使用计算机将比使用纸和笔（以及大量的橡皮擦）得到一个清晰思路的时间要短。

实践一下目前我们所学的东西，接下来将介绍如何设计一张简单的 PCB 图。FidoCadJ 不像其它电子 CAD 软件那么强大，有时候它操作起来比较困难。FidoCadJ 提供一种由原先好用的 R41 制作 PCB 的方法⁴转移到计算机上制作 PCB 的方法。很显然，在计算机上工作会具有较大的灵活性。

您应该知道绘制一张 PCB 图，特别是较复杂的那种，并不是一件简单的工作。大多数 CAD 厂商在宣传单上注明自己的产品有自动布局和自动布线的功能。毫无疑问，这些仍然是用户所期望的功能，发挥着重要的作用。然而，在 FidoCadJ 中集成的是一个非常直接和快速的绘制 PCB 图的方法，适用于绘制小型的可自定义尺寸的 PCB 图。这里，我们将介绍如何绘制一张小型的完整的 PCB 图。

我建议，在开始绘图前要先有一个清晰的元件布局和布线的思路，这样，可以尽可能少地避免交叉。

这里我们稍微作弊一下，开始绘图前我们先看下绘图结果，见图 2.11 所示。这是一个简单的共射极放大器，围绕一个 NPN 晶体管（BC547 或类似器件）建立起来的电路。可以透过元件看到 PCB 图，这样有助于思考实际的板子。为此，绘制元件形状的丝印层非常有用，尽管可能对于这个 DIY 工程来说我们不会制作实际的板子。

我建议，第一件要做的事情就是尽可能布置好所有的元件。我们的示例中有晶体管（来自元件库“PCB footprints / 3terminals semiconductors / TO92”），电阻（来自元件库“PCB footprints / Resistors / Resistor 1/4W0,4i”）和电解电容（来自元件库“PCB footprints / Electrolytic capacitor / Vert. diam.5mm2.5mm pitch”）。为了给出板子的边界，可以在丝印层（3 层）画一个不填充的矩形。我们可以使用矩形基元件，画之前注意需要选择相应的层。到此，我们的绘图应该类似于图 2.12 所示的样子。⁵

接着我们介绍覆铜区域，电源正极和负极。它们是用多边形（多边形基元件）绘制的。在选择模式时，双击多边形边框（在弹出的对话框中）设置为填充。在绘制多边形之前，确认下当前的绘制层是底层导线层（1 层）。电源线使用区域填充（覆铜）

⁴关于 R41 制作 PCB 的方法请参考 Davide Bucci 的解释：

<http://sourceforge.net/projects/fidocadj/forums/forum/997486/topic/3923275/index/page/2>

⁵目前 FidoCadJ 有意大利语、法语、英语、西班牙语，德语，中文和荷兰语。翻译菜单和命令的工作很简单，如果您想 FidoCadJ 运行您本地语言，请通知我！也可以开展翻译库（至少是标准库）和翻译本文档的工作。

2.5. 一张简单的 PCB 图

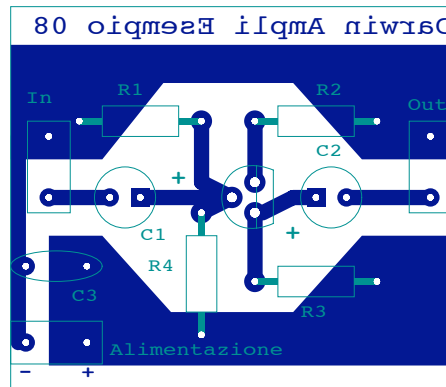


Figure 2.11.: 一个非常简单的由 NPN 晶体管构成的共射极放大器电路板。

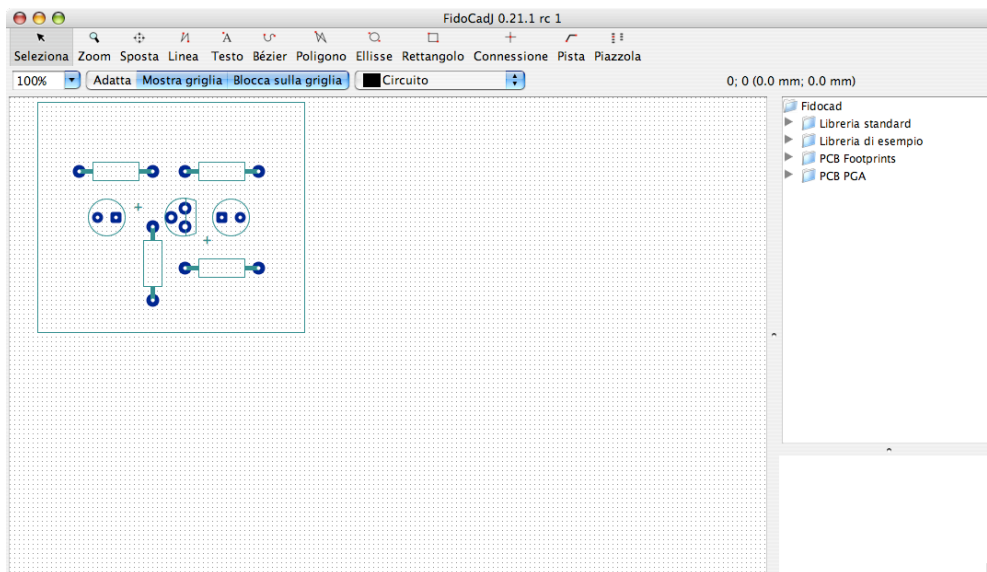


Figure 2.12.: 最主要的元件已布置到板上。

2. FidoCadJ 绘图

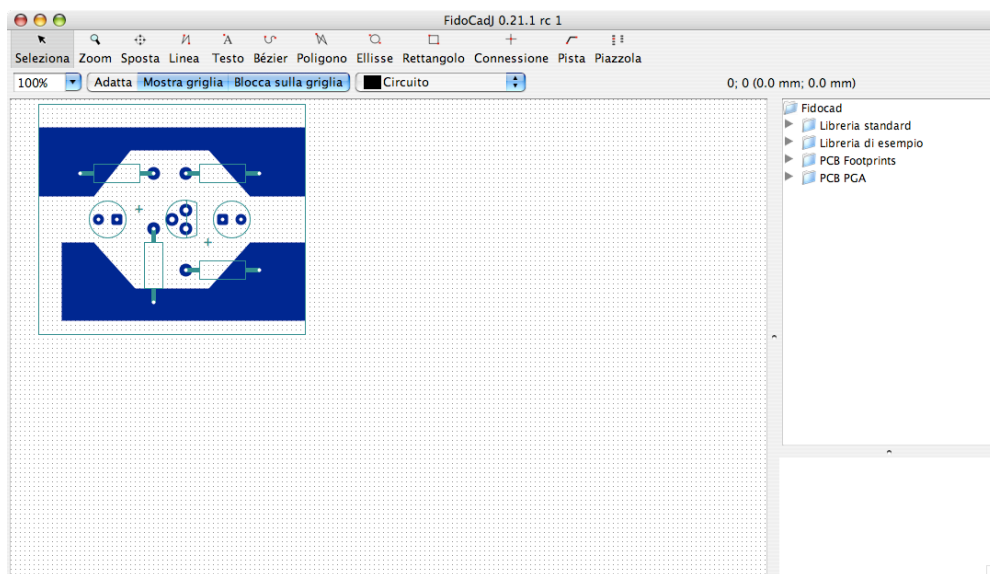


Figure 2.13.: 用多边形画电源线。

方式可以有效地降低寄生电感。到此，我们的绘图应该类似于图 2.13 所示的样子。

我们可以使用 *PCB* 导线基元件来画连接导线。设置导线宽度为 10 个单位 ($1.27mm$)，这样会便于焊接。

我们注意到还需要一些连接器：一个用于输入，一个用于输出，一个用于供电。我们使用聚酯电容类的引脚封装，其中可能有我们需要的封装规格。我们不要忘记 FidoCadJ 的工作思想是来自原先 R41 的制作方法。

我们可以在底层导线层（1 层）放上“+”和“-”符号，在电源上并联一个陶瓷电容。我们还可以在顶端写些文字，写在底层导线层（1 层），写完后需要翻转文本。设置翻转很简单：切换到选择模式，双击文本弹出属性对话框，在其中设置翻转。要得到大小合适的文字需要进行一些尝试，字体的水平尺寸（宽度）和垂直尺寸（高度）的比例最好保持为 3/4。图 2.15 中，字体的宽度为 11 个单位，高度为 18 个单位。

目前，完成 *PCB* 图还差一件事情：为每个元件放置名称文本，这是放置在丝印层（3 层）的。如图 2.16 所示，制作完成的 *PCB* 图。

PCB 图绘制完成后，我们可能需要打印它，用于制作投影机使用的幻灯片，或用于其它方面（例如“Press & Peel”）。对于一些不需要打印的层，我们只要把它们设置成不可见就不会打印出来。可以通过菜单“视图/层”打开的对话框设置层不可见。

注意导线的宽度：有时在屏幕上看起来很粗的导线实际上很细，这样的话，在焊接过程中很容易从板上脱落。

2.5. 一张简单的 PCB 图

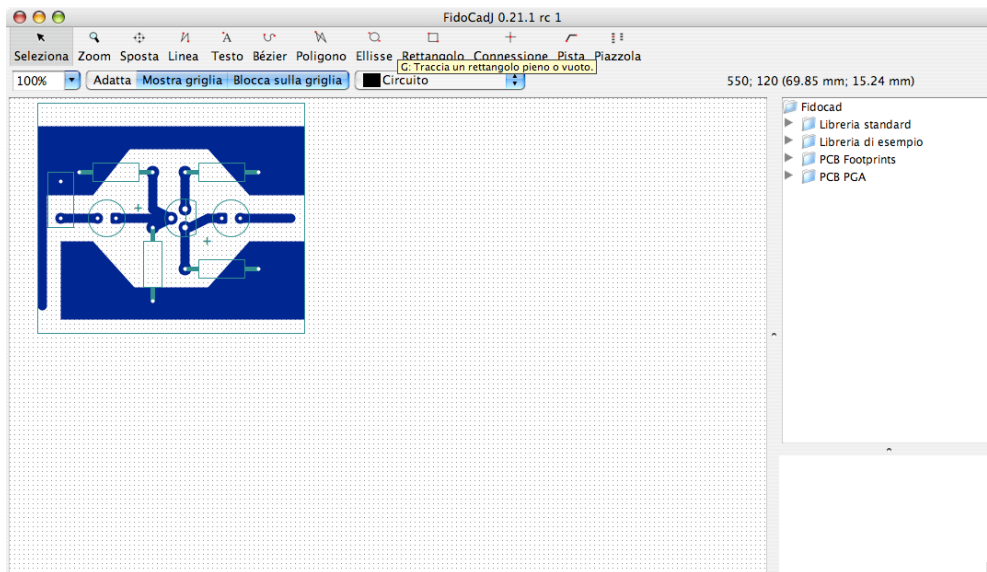


Figure 2.14.: 画其它的 PCB 导线。

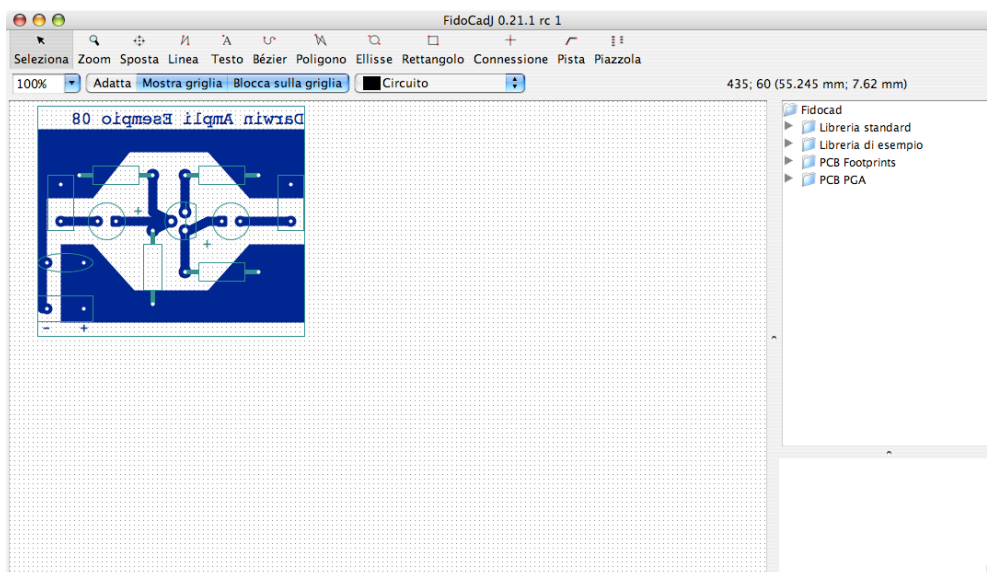


Figure 2.15.: PCB 基本完成。

2. FidoCadJ 绘图

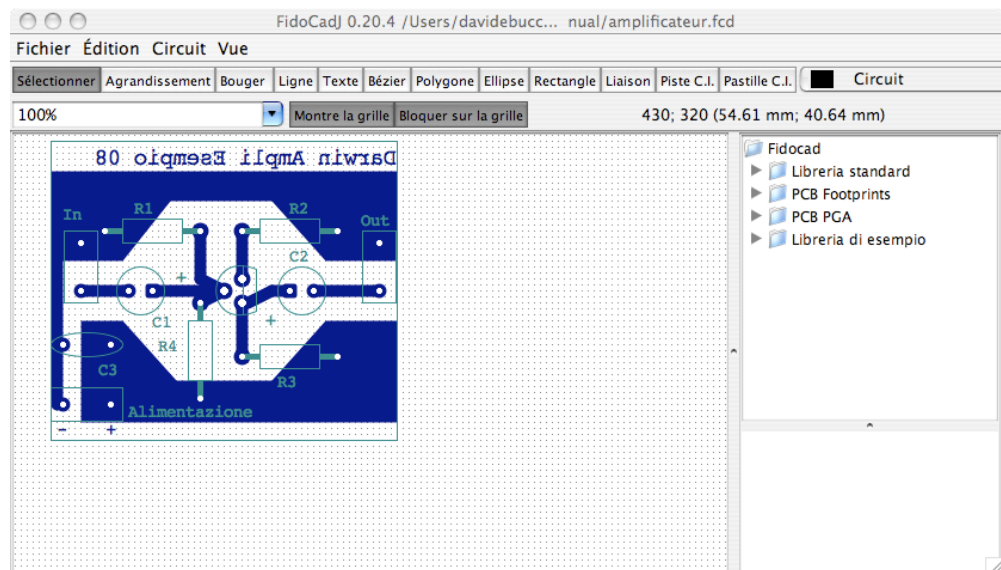


Figure 2.16.: 画上丝印层，工作完成。

下面的示例，我们将隐藏丝印层（3 层）。应用程序将只打印底层导线层（1 层）。

接着我们可以打印出屏幕上看到的所有东西（很明显，不适合纸张的大小，我们希望可以设置图纸的尺寸）。注意，选择了黑白打印，可以得到最大的对比度。由于制造 PCB 厂商使用的技术不同，有时可能需要翻转绘图。由于我们的 PCB 尺寸很小，打印出来只占了整张纸（标准 ISO-UNI A4 纸）的一个小角落，如图 2.17 所示。

上面示例的 PCB 图的绘图代码信息如下（注意长的行！）：

```
[FIDOCAD]
TY 320 10 18 11 0 4 1 * Darwin Ampli Esempio 08
TY 85 240 12 8 0 5 1 * +
TY 44 239 12 8 0 5 1 * -
PL 35 90 35 225 10 1
PL 55 130 95 130 10 1
PL 250 130 305 130 10 1
PL 215 130 230 130 10 1
PL 195 140 215 130 10 1
PL 115 130 175 130 10 1
MC 155 220 3 0 PCB.R01
MC 75 80 0 0 PCB.R01
```

2.5. 一张简单的 PCB 图



Figure 2.17.: 在 ISO-UNI A4 纸上（翻转）打印 PCB 图的效果。

2. FidoCadJ 绘图

```
MC 270 185 2 0 PCB.R01
MC 270 80 2 0 PCB.R01
MC 230 130 3 0 PCB.CE00
MC 115 130 1 0 PCB.CE00
MC 40 175 0 0 PCB.CC50
PL 190 80 190 120 10 1
PL 190 140 190 185 10 1
PL 155 80 155 120 10 1
PL 155 120 175 130 10 1
PL 155 140 175 130 10 1
PP 30 30 30 105 90 105 130 55 215 55 260 105 320 105 320 30 1
PP 320 240 320 155 260 155 215 205 135 205 90 155 55 155 55 240 1
MC 190 120 0 0 PCB.T092
MC 305 90 1 0 PCB.CPBX352
MC 55 90 1 0 PCB.CPBX352
MC 80 225 2 0 PCB.CPBX352
TY 290 65 12 8 0 0 3 * Out
TY 40 60 12 8 0 0 3 * In
TY 95 225 12 8 0 0 3 * Alimentazione
TY 70 190 12 8 0 0 3 * C3
TY 230 95 12 8 0 0 3 * C2
TY 115 150 12 8 0 0 3 * C1
TY 120 170 12 8 0 0 3 * R4
TY 220 200 12 8 0 0 3 * R3
TY 230 55 12 8 0 0 3 * R2
TY 100 55 12 8 0 0 3 * R1
RV 30 5 320 255 3
```

2.6. 使用标尺

当绘制一张 PCB 图时，经常需要在图上测量尺寸。例如，你需要测下导线的宽度、两条导线之间的间隔或板卡的尺寸。FidoCadJ（从版本 0.23.2 开始）提供一个具有特色的标尺，让你很容易地做测量。您只需要鼠标右键点击并拖拉，就会出现一个绿色的标尺，如图 2.18 所示。如果你的系统不支持右键点击并拖拉，你也可以在按住 **Shift** 键的同时用鼠标左键点击并拖拉。

显示的长度值的单位是 FidoCadJ 的逻辑单位，另外，也有一个以毫米为单位的

2.7. 箭头和线条样式

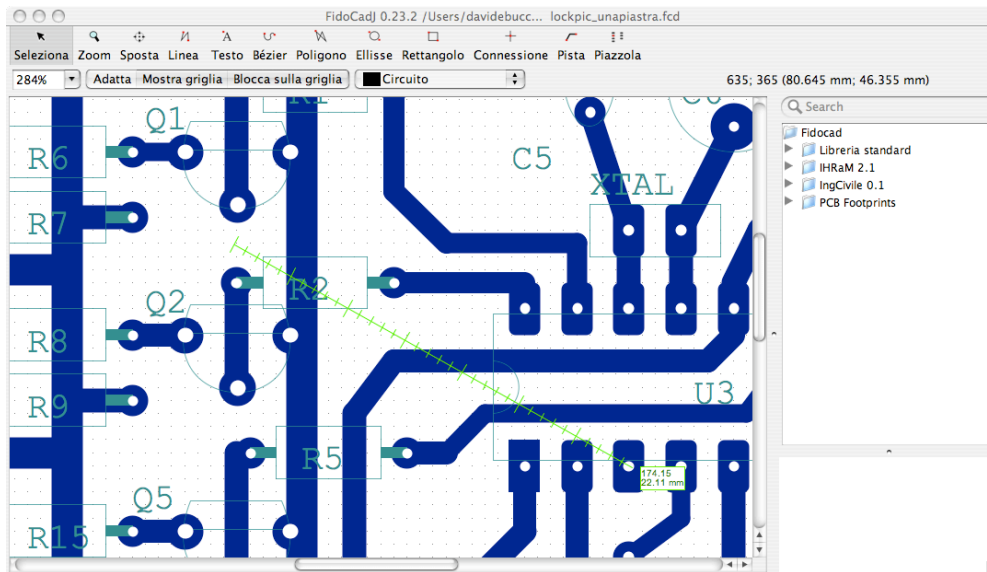


Figure 2.18.: 鼠标右键点击并拖拉，出现 FidoCadJ 标尺。

长度值。该功能有助于 1 : 1 的打印模式（例如打印 PCB 图）。

2.7. 箭头和线条样式

FidoCadJ 可以设置直线、贝塞尔曲线和自然三次样条曲线的端点显示为箭头。可以指定这类元件的始端或末端或两端是否显示为箭头，并且可以选择不同的箭头样式。

现在，可以选择多种线条样式，用于技术类或机械类的绘图。图 2.19 显示了一个位于虚线框中的电路图（GIC）的示例。还用到一条带箭头的贝塞尔曲线。双击贝塞尔曲线，FidoCadJ 显示一个参数对话框，如图 2.20 所示。你会看到选项框“始端箭头”被勾选了。FidoCadJ 将根据贝塞尔曲线的绘制方向放置一个箭头。有好几种箭头样式和线条样式供选择，您可以尝试选择一下，了解它们之间的差别。

线条样式可选择和端点可显示为箭头的功能可能会与原 FidoCad 格式不兼容。这个意味着使用这些功能不能兼容 Windows 上的 FidoCad。您需要知道您使用的是 FidoCadJ 的扩展。如果您需要保持与 FidoCad 兼容，您需要通过“FidoCadJ 参数设置”窗口的“FidoCadJ 扩展”分页，勾上选项“完全兼容 FidoCad”。通过这种方法，应用程序将不再允许你使用会引起与 FidoCad 兼容问题的绘图元件。如果你想要

2. FidoCadJ 绘图

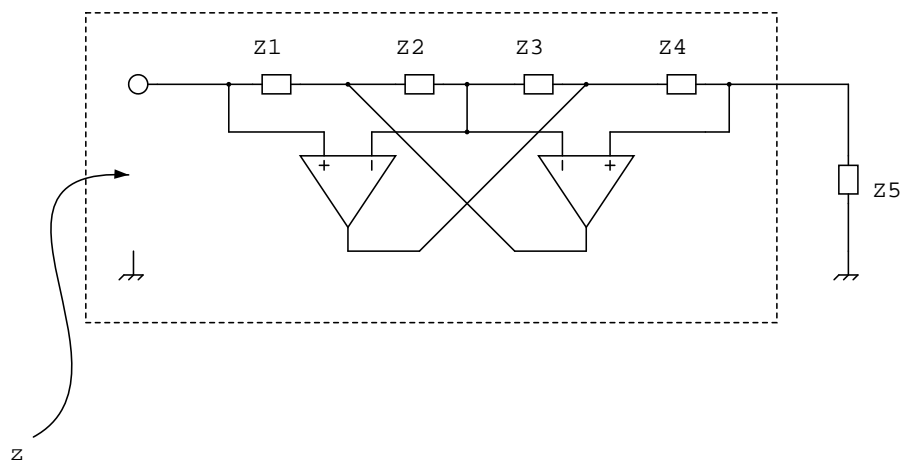


Figure 2.19.: 电路图 (Antoniou 通用阻抗转换器 (GIC)), 其中用到一些 FidoCadJ 的扩展。

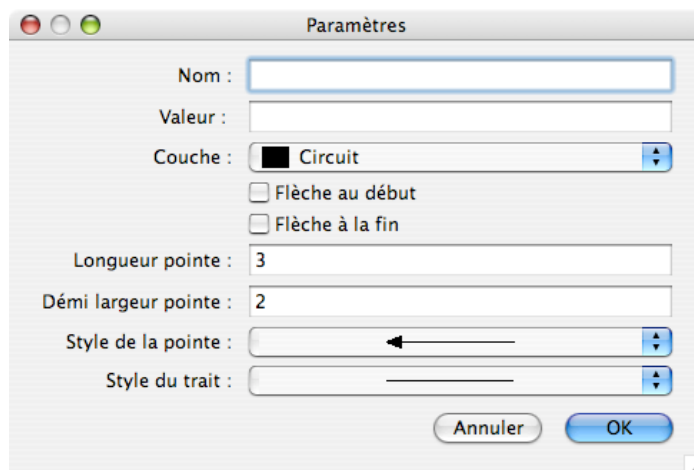


Figure 2.20.: 图 2.19 中的贝塞尔曲线的参数对话框。

进一步了解其它绘图元件与 FidoCad 的兼容性，请参考 3.4 章节。

2.8. 导出

在我看来，FidoCadJ 最主要的作用可能是创建用于排版的简单原理图。为此，我加入了一个重要功能——可以导出多种不同文件格式的绘图。

通过菜单“文件/导出”即可输出当前的绘图。表 2.2 列出了当前可以输出的图形文件格式。表中的各种文件格式（也是 FidoCadJ 导出窗口中所列出的格式）都指明了它是矢量格式或位图格式。尽可能选择矢量图格式输出，这样可以获得最佳效果。⁶

矢量格式存储为一组绘图元件。
位图格式存储为一个像素矩阵。

导出成位图文件格式时，请选择“抗锯齿”，有助于减少像素化后的锯齿（对于斜线特别有效）。导出成矢量文件格式时，“分辨率”和“抗锯齿”项是无效的，此时你需要指定的是“比例”项。

“黑白色”项表示用纯黑色打印任何可见层。这主要用于打印准备排版或制作幻灯片使用的绘图。

2.9. 命令行选项

应用程序以 .jar 格式文件发布，这是 Java 的档案文件⁷。在多数系统上，如果机器上安装了 Java 的最新版本，只要双击文件即可运行程序。引用 Sun 的术语，JRE 或 Java 运行环境，它是运行 Java 程序（不是编写 Java 程序，那需要 SDK...）的必要条件也是充分条件。运行 FidoCadJ 至少需要 Java 1.5 版本，这是几年前的版本了。

某些情况，需要从命令行（Unix 系统中的终端或 Windows 系统的 MS-DOS 提示符）运行 FidoCadJ。这时，只需要运行带 -jar 选项的 java 命令：

```
java -jar fidocadj.jar
```

如果命令行中同时指定一个需要打开的绘图文件，FidoCadJ 将运行并试着打开它，例如（在 Unix 机器上）：

```
java -jar fidocadj.jar ~/FidoCadJ/test.fcd
```

FidoCadJ 将运行并试着打开文件 ~/FidoCadJ/test.fcd（假如该文件存在）。

FidoCadJ 还能做其它有趣的事情，选项 -h 查看 FidoCadJ 的选项信息：

```
[davidbucci@Darwin]$ java -jar fidocadj.jar -h
```

⁶FidoCadJ 结构允许加入其它文件格式，并且很方便。如果你想加入开发项目请联系我。

⁷针对苹果机发布的版本，是一个独立的应用程序。

2. FidoCadJ 绘图

Format	Comment
JPG	很常用的位图格式。由于它是有损压缩，不适合用于输出 FidoCadJ 原理图。因为常用，所以把它加了进来。
PNG	压缩的位图格式，适用于输出原理图。当不能使用矢量格式时，这可能是 FidoCadJ 绘图输出的最好格式。
SVG	W3C（万维网联盟）标准矢量格式。一些因特网浏览器（例如最新版的 Safari）可以在网页中显示它。这是个很好的图形和原理图格式，它可以使用诸如 Inkscape 那样的应用程序修改 FidoCadJ 的绘图。目前对输出旋转和翻转的文本有些限制。
EPS	封装的 PostScript 矢量格式。对那些使用专业图形应用程序的人或那些想把 FidoCadJ 图纸用于 L ^A T _E X 文档的人很有用。输出的绘图应该是完全一样的。可以用此格式输出本手册第 17 页的图 2.11 所示的绘图（通过使用 PDF 转换器，我使用的是 PDFL ^A T _E X）。
PGF	在使用了 CTAN 的 <i>pgf</i> 包的 L ^A T _E X 文档中，可以直接使用的矢量格式。该格式特别适合输出原理图，并且是一个简单的可编辑脚本。由于它的文本属性，可以直接引入到 L ^A T _E X 代码中，本手册第 10 页的图 2.5 示意图就使用了该技术。
SCR	FidoCadJ 可以输出脚本格式的绘图，这可以被 CadSoft 的 Eagle 使用。使用该功能，需要在 Eagle 当前安装的 <code>lbr</code> 库文件夹中加入 <code>FidoCadJLIB.lbr</code> 库。该库可以从 FidoCadJ 的网站下载。在写这篇手册时，该功能还只能针对包含常见符号的原理图，有些绘图元件如焊盘和导线还不可以导出到 Eagle 脚本。

Table 2.2.: FidoCadJ 输出文件格式列表。

2.9. 命令行选项

```
This is FidoCadJ, version 0.24.1.
By Davide Bucci, 2007-2012.

Use: java -jar fidocadj.jar [-options] [file]
where options include:

-n      Do not start the graphical user interface (headless mode)

-d      Set the extern library directory
Usage: -d dir
where 'dir' is the path of the directory you want to use.

-c      Convert the given file to a graphical format.
Usage: -c sx sy eps|pdf|svg|png|jpg|fcd|sch outfile
If you use this command line option, you *must* specify a FidoCadJ
file to convert.
An alternative is to specify the resolution in pixels per logical unit
by preceding it by the letter 'r' (without spaces), instead of giving
sx and sy.

-s      Print the size of the specified file in logical coordinates.

-h      Print this help and exit.

-t      Print the time used by FidoCadJ for the specified operation.

-p      Do not activate some platform-dependent optimizations. You might try
this option if FidoCadJ hangs or is painfully slow.

-l      Force FidoCadJ to use a certain locale (the code might follow
immediately or be separated by an optional space).

[file] The optional (except if you use the -d or -s options) FidoCadJ file to
load at startup time.

Example: load and convert a FidoCadJ drawing to a 800x600 pixel png file
without using the GUI.
java -jar fidocadj.jar -n -c 800 600 png out1.png test1.fcd

Example: load and convert a FidoCadJ drawing to a png file without using the
graphic user interface (the so called headless mode).
Each FidoCadJ logical unit will be converted in 2 pixels on the image.
java -jar fidocadj.jar -n -c r2 png out2.png test2.fcd

Example: load FidoCadJ forcing the locale to simplified chinese (zh).
java -jar fidocadj.jar -l zh

[davidebucci@Darwin]$
```

最简单的选项-n，软件运行... 不出现 GUI。这个选项，是把 Java 的环境变

2. FidoCadJ 绘图

量`java.awt.headless`设置为 `true`。显然，单独使用该选项没什么意义，需要和其它的选项组合使用。选项`-d`指定 FidoCadJ 在启动时加载库的文件夹。选项`-c`允许你使用 FidoCadJ 把 FidoCad 文件（必须指定）转换成一张矢量或位图格式的图片，这时把 FidoCadJ 当作转换器使用，并不需要 GUI 交互（与`-n`选项一起使用）。

看我们的第一个示例，可以帮助我们理解：

```
java -jar fidocadj.jar -n -c 800 600 png out1.png test1.fcd
```

FidoCadJ 将运行但不激活 GUI，并从`test1.fcd`文件导出 `png`格式的文件。导出文件名称为`out1.png`，像素大小为 `800*600`。`-c`选项还有另一种参数——指定转换时一个逻辑单位对应多少个像素（假定该参数为 r_p ）。FidoCadJ 不能处理半像素（指定的必须为整数），经过实践，认为设置 r_p 为 2，即 1 个逻辑单位对应 2 个像素，可以保证原理图总是能看得清楚（尽管可能有点小）。参数 r_p 可以为非整数，见以下示例：

```
java -jar fidocadj.jar -n -c r1.25 png out2.png test2.fcd
```

要知道原理图的总大小（逻辑单位），你可以使用`-s`选项。记住，不管哪种方式，转换时 FidoCadJ 总是在电路图的边缘加入 $t_{\text{margin}} = 3$ 个逻辑单位。因此，假定 t_w 为绘图的逻辑单位数（通过`-s`选项查看），导出绘图的像素数 p_w 的计算公式如下：

$$p_w = r_p(t_w + 2t_{\text{margin}}) \quad (2.1)$$

另一个有趣的选项（该特性主要来自 Java 而非 FidoCadJ）就是可以修改应用程序的外观（Java 称之为 LookAndFeel）。你可以选择你喜欢的 LookAndFeel，而不需要修改一行代码。这里有个 Linux 用户使用的外观 GTK+ LookAndFeel。

```
java -Dswing.defaultlaf=com.sun.java.swing.plaf.gtk.GTKLookAndFeel -jar fidocadj.jar
```

还有个经典的 Motif LookAndFeel，如图 2.21 所示⁸。

```
java -Dswing.defaultlaf=com.sun.java.swing.plaf.motif.MotifLookAndFeel -jar fidocadj.jar
```

很明显，上面列出的命令需要在终端中输入，确保当前目录中包含文件`fidocadj.jar`，并在同一行中输入。

⁸MacOSX 上的这种风格可能会令那些熟悉非常优雅的图形接口的用户感到莫名其妙得吃惊，例如 Aqua。而几年前我看到的基于 Motif 的图形界面上的同步控制系统，也是异常的漂亮！

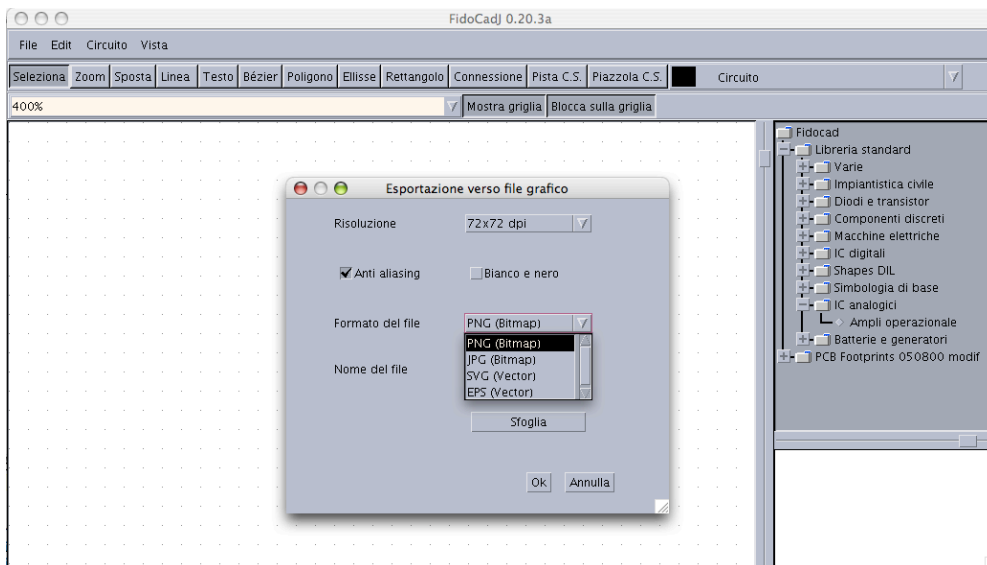


Figure 2.21.: 在 MacOSX上程序的外观，使用 Motif LookAndFeel。

2.10. 库管理

应用程序允许我们指定加载库文件（扩展名为`.fcl`的文件）的目录。可以通过菜单“文件/选项”进行设置。如果目录中存在一个`FCDstdlib.fcl`库文件，加载这个库文件将取代应用程序中的标准库。类似的，如果存在一个`PCB.fcl`库文件，将取代`PCB`库⁹。其它扩展名为`.fcl`的库文件，被认为是增加的库，会和标准库一起加载。感谢 Roby IZ1CYN 的付出，从 0.23 版开始，我可以直接在 FidoCadJ 发行版内部包含 IHRaM 3.1 库。之所以包含这个库，是因为目前为止它是我见过的所有库中内容最完整、结构最合理的一个库。由于这个库已经是 FidoCadJ 的嵌入库，如果当前加载库路径中包含`IHRaM.FCL`库文件，这个库将被认为是增加的库被加载到程序中并显示版本号。你也可以增加文件名为`elettrotecnica.fcl`的电气符号库。

加载库搜索路径中的其它一些扩展名为`.fcl`的文件将被认为是库文件。当用户改变了库加载路径，需要选择菜单“电路/更新库”。

从版本 0.23 开始，FidoCadJ 可以拆分非标宏元件（拆分成原 FidoCad 可以识别的元件）。这个功能对于在新闻组中粘贴绘图很有用，因为该功能可以把每个不是标准 FidoCad 库的宏元件拆分为 FidoCad 可识别元件。这样，看你的帖子并不需要和你安装同样的库。激活该功能，选择菜单“文件/选项”，在“FidoCadJ 扩展”分页上

⁹注意使用大写字母，特别是你的系统的文件管理是区分字母大小写的。

2. FidoCadJ 绘图

可以勾选拆分非标准宏元件的选项，这里可以选择在保存文件时和/或复制/粘贴（部分或全部）绘图时激活这个动作。

3. 绘图格式、宏元件和库

本章详细描述 FidoCad 绘图格式，即 FidoCadJ 基本绘图格式。这是一种简单的文本格式，其优点是紧凑且高效。目前为止，还没有详细描述该格式的文档，这里我只是努力把全部所知道的总结出来。另外，我对 FidoCad 绘图格式做了一些扩展（仅适用于 FidoCadJ），称之为 FidoCadJ 扩展绘图格式，本章也描述了这些扩展绘图格式。请记住，FidoCadJ（从 0.23.4 版本开始）的绘图文件在任何平台上只使用 UTF-8 编码格式¹。

3.1. 绘图文件头

所有 FidoCad 的绘图文件，都必须以 `[FIDOCAD]` 标记开头。程序通过此标记才能识别出是 FidoCad 的绘图文件，然后才开始解析其中的命令行。在这点上，FidoCadJ 要比 FidoCad 宽松，没有以该标记开头的绘图文件也可以得到解析，并且会跳过错误的命令行，但错误命令行的数量不能超过程序内部设置的一个值（大约 100 行），这确保 FidoCadJ 在打开它不能解析的文件时不会浪费过多的时间。

3.2. 坐标系统

FidoCadJ 绘图的坐标系统非常简单。事实上，它是一个非常大的正向坐标区域。X 向和 Y 向的单位长度都固定为 $5mil$ ($0.127mm$)，这个单位长度对于显示最小的 SMD 封装器件来说，它也是一个很好的分辨率，日常应用已经足够了。打印时，FidoCadJ 绘图的分辨率是每英寸 200 点。

原 FidoCad 有两种不同的绘图模式：绘制 PCB 图模式和绘制电路原理图模式。在 FidoCadJ 中，这两种绘图模式已经不加以区分了，只是在打印绘图时才区别对待。因为，由程序来调整绘制的电路原理图打印比例，使之适合打印纸张是非常合理的，不然，打印出来的电路原理图有可能只有邮票那么大。

¹注意绘图格式与编码格式的区别：绘图格式是描述图形使用的文本的格式；编码格式是描述文件的计算机存储的格式，这里 UTF-8 编码格式为存储 FidoCadJ 绘图文件的计算机存储的格式。

3. 绘图格式、宏元件和库

3.3. 绘图元件

FidoCadJ 有 12 种绘图元件，列出如下：

- 直线
- 填充或不填充的矩形
- 简单文本（已废弃）
- 高级文本
- 填充或不填充的多边形
- 填充或不填充的椭圆形
- 贝塞尔曲线
- 自然三次样条曲线
- 电气连接点
- PCB 焊盘
- PCB 导线
- 宏元件

下面我们将逐一讲解它们。各种绘图元件有相应的带参数的命令定义。通常情况是，一条命令行对应一个绘图元件，命令行中有命令和参数（一般为整数数字或文本字符串），命令和参数/参数和参数之间用空格隔开。

直线

数学家们可能认为描述为“线段”更合适。

直线由 **LI** 命令定义，其命令参数有：起点坐标、终点坐标、层号。

```
LI x1 y1 x2 y2 l
```

其中， (x_1, y_1) 和 (x_2, y_2) 分别表示起点坐标和终点坐标； l 表示层号（整数，取值范围为 $[0, 15]$ ）。

FidoCadJ 扩展：从 0.23 版本开始，可以在 **LI** 命令后紧跟一条扩展命令：

```
FCJ a b c d e nv
```

a	端点形状
0	无箭头
1	始端箭头
2	末端箭头
3	两端箭头

Table 3.1.: 表示线条类基元件的端点箭头的 a 参数的含义。

b	端点箭头样式
0	标准箭头
1	标注标准箭头
2	空心箭头
3	标注空心箭头

Table 3.2.: 表示箭头样式的 b 参数的含义。

其中, a 表示端点形状 (取值及含义请见表 3.1); b 表示端点箭头样式 (取值及含义请见表 3.2); c 和 d 分别表示箭头长度和箭头宽度的一半; e 表示线条样式 (整数); nv 等于 1 则FCJ命令后紧跟 2 条TY命令, 给出该元件关联的名称和编码, 等于 0 则没有。

填充或不填充的矩形

填充或不填充的矩形由RP和RV命令定义, 其命令参数有: 矩形的任意一条对角线的 2 个顶点坐标、层号。

```
RP x1 y1 x2 y2 l
RV x1 y1 x2 y2 l
```

其中, (x_1, y_1) 和 (x_2, y_2) 分别表示矩形对角线的 2 个顶点坐标; l 表示层号 (整数, 取值范围为 $[0, 15]$)。

FidoCadJ 扩展: 从 0.23 版本开始, 可以在RP或RV命令后紧跟一条扩展命令:

```
FCJ e nv
```

其中, e 表示线条样式 (整数)。 nv 等于 1 则FCJ命令后紧跟 2 条TY命令, 给出该元件关联的名称和编码, 等于 0 则没有。

3. 绘图格式、宏元件和库

位	位值	动作
0	1	加粗
1	2	倾斜
2	4	翻转文本

Table 3.3.: 文本样式字段位的含义。

简单文本（已废弃）

简单文本是 FidoCad 早期版本中使用的文本元件。FidoCadJ 把它解析为 12 个单位的大小固定的文本，不随绘图缩放而变化。

由于 FidoCad 的创建者 Lorenzo Lutti 认为该元件已经淘汰，并把它废弃掉了，FidoCadJ 采取了同样的做法。然而，为了能够正确解析旧文件，FidoCadJ 还是能够识别它，但当保存文件时，将使用 **TY** 命令存储该元件（转化为高级文本）。

简单文本由 **TE** 命令定义，格式如下：

```
TE x1 y1 text to be written
```

其中， (x_1, y_1) 是文本的位置坐标；“text to be written”表示实际的文本内容。注意没有层信息。FidoCadJ 默认该元件在 0 层（原理图层）。

高级文本

相比上面介绍的简单文本，高级文本的功能更加丰富。

高级文本由 **TY** 命令定义，其命令参数有：文本位置坐标、文本高度、文本宽度、文本样式、层号、字体和文本。由于信息量比较多，这条命令有点复杂。

```
TY x1 y1 sy sx a s l f text to be written
```

其中， (x_1, y_1) 表示文本的位置坐标； s_y 和 s_x 分别表示文本使用字体的高度和宽度（逻辑单位数。我通常先设置字体高度，再根据字体高度设置字体宽度，这样的话需要有一个确定的斜率）； a 表示文本旋转（以 1/60 度为旋转单位）； s 表示文本样式（见表 3.3）； l 表示层号（整数，取值范围为 [0,15]）； f 表示字体名称（如果字体名称包含空格，需要把空格替换成 + 号；或可以用星号，表示使用标准的 Courier New 字体）；“text to be written”表示实际的文本内容。

文本的最大长度为 80 个单词。这里以单词为计数对象，而不是字符，是因为在国际化程序中，是把命令行分解成单词进行解析的。

填充或不填充的多边形

填充的或不填充的多边形由PP和PV命令定义，其命令参数有：多边形的各顶点坐标、层号。

```
PP x1 y1 x2 y2 ... l
PV x1 y1 x2 y2 ... l
```

其中， (x_1, y_1) , (x_2, y_2) ... 表示多边形的各顶点坐标； l 表示层号（整数，取值范围为 $[0, 15]$ ）。顶点个数越多命令行越长，为避免命令行过长，顶点个数最多为 20 个。

FidoCadJ 扩展：从 0.23 版本开始，可以在PP和PV命令后紧跟一条扩展命令：

```
FCJ e nv
```

其中， e 表示线条样式（整数）； nv 等于 1 则FCJ命令后紧跟 2 条TY命令，给出该元件关联的名称和编码，等于 0 则没有。

填充或不填充的椭圆

填充或不填充的椭圆由EP和EV命令定义，其命令参数有：椭圆外接矩形的任意一条对角线的 2 个顶点的坐标、层号。

```
EP x1 y1 x2 y2 l
EV x1 y1 x2 y2 l
```

其中， (x_1, y_1) 和 (x_2, y_2) 表示椭圆外接矩形的对角线的 2 个顶点； l 表示层号（整数，取值范围为 $[0, 15]$ ）。

FidoCadJ 扩展：从 0.23 版本开始，可以在EP和EV命令后紧跟一条扩展命令：

```
FCJ e nv
```

其中， e 表示线条样式（整数）； nv 等于 1 则FCJ命令后紧跟 2 条TY命令，给出该元件关联的名称和编码，等于 0 则没有。

贝塞尔曲线

贝塞尔曲线，是使用 3 次方程，由 4 个顶点控制的曲线。它由BE命令定义，其命令参数有：各控制点坐标、层号。

```
BE x1 y1 x2 y2 x3 y3 x4 y4 l
```

3. 绘图格式、宏元件和库

其中, $P_1 \equiv (x_1, y_1)$, $P_2 \equiv (x_2, y_2)$, $P_3 \equiv (x_3, y_3)$, $P_4 \equiv (x_4, y_4)$ 这 4 个贝塞尔曲线的控制点, l 表示层号 (整数, 取值范围为 $[0,15]$)。上面给予的 4 个顶点, 曲线可以通过下面表达式计算。

$$B(t) = (1-t)^3 P_1 + 3t(1-t)^2 P_2 + 3t^2(1-t) P_3 + t^3 P_4, \quad (3.1)$$

其中 $t \in [0,1]$, 是一个参数。

FidoCadJ 扩展: 从 0.23 版本开始, 可以在 **BE** 命令后紧跟一条扩展命令:

```
FCJ a b c d e nv
```

其中, a 表示端点形状 (取值及含义请见表 3.1); b 表示端点箭头样式 (取值及含义请见表 3.2); c 和 d 分别表示箭头长度和箭头宽度的一半; e 表示线条样式 (整数); nv 等于 1 则 **FCJ** 命令后紧跟 2 条 **TY** 命令, 给出该元件关联的名称和编码, 等于 0 则没有。

自然三次样条曲线 (复杂曲线)

自然三次样条曲线, 是由一些顶点控制的曲线, 该曲线以一种平滑的计算方式经过每个顶点²。它由 **CV** 和 **CP** 命令定义, 其命令参数有: 各控制点坐标、层号。

```
CV aa x1 y1 x2 y2 ... l
CP aa x1 y1 x2 y2 ... l
```

其中, aa 等于 1 则曲线闭合, 等于 0 则曲线开放; $(x_1, y_1), (x_2, y_2) \dots$ 表示各控制点坐标; l 表示层号 (整数, 取值范围为 $[0,15]$)。同矩形类似, 顶点个数越多命令行越长, 为避免命令行过长, 顶点个数最多为 100 个。命令行长度取决于顶点的数量。该元件在 0.24 版本中引入, 原 FidoCAD 中没有。

FidoCadJ 扩展: 可以在 **CV** 和 **CP** 命令后紧跟一条扩展命令:

```
FCJ a b c d e nv
```

其中, a 表示端点形状 (取值及含义请见表 3.1); b 表示端点箭头样式 (取值及含义请见表 3.2); c 和 d 分别表示箭头长度和箭头宽度的一半; e 表示线条样式 (整数); nv 等于 1 则 **FCJ** 命令后紧跟 2 条 **TY** 命令, 给出该元件关联的名称和编码, 等于 0 则没有。

² <http://www.cse.unsw.edu.au/~lambert/splines/natcubic.html>

电气连接点

电气连接点是一个很简单的填充的圆形，尺寸固定（圆形的直径在程序中设置为 2 个逻辑单位），用于表示电路原理图中的电气连接。由 **SA** 命令定义，其命令参数有：位置坐标、层：

```
SA x1 y1 l
```

FidoCadJ 扩展：可以在 **SA** 命令后紧跟扩展命令 **FCJ**，**FCJ** 命令后紧跟 2 条 **TY** 命令，给出该元件关联的名称和编码。

PCB 焊盘

PCB 焊盘由 **PA** 命令定义，其命令参数有：样式（椭圆形，矩形，圆角矩形）、内部孔的直径。

```
PA x1 y1 dx dy si st l
```

其中， (x_1, y_1) 表示焊盘的位置坐标； d_x 表示焊盘的宽度（也就是焊盘 x 向的尺寸）； d_y 表示焊盘的高度（也就是焊盘 y 向的尺寸）； s_i 表示焊盘过孔的直径， s_t 表示焊盘的样式：

- 0 椭圆形焊盘
- 1 矩形焊盘
- 2 圆角矩形焊盘

l 表示层号（整数，取值范围为 $[0, 15]$ ）。

FidoCadJ 扩展：可以在 **PA** 命令后紧跟扩展命令 **FCJ**，**FCJ** 命令后紧跟 2 条 **TY** 命令，给出该元件关联的名称和编码。

PCB 导线

PCB 导线是可以指定宽度的线段，线段两端总是圆的，便于连接其它 PCB 导线或焊盘。它由 **PL** 命令定义，其命令参数有：起点坐标、终点坐标、宽度、层号。

```
PL x1 y1 x2 y2 di l
```

其中， (x_1, y_1) 和 (x_2, y_2) 分别表示起点坐标和终点坐标； d_i 表示宽度； l 表示层号（整数，取值范围为 $[0, 15]$ ）。

FidoCadJ 扩展：可以在 **PL** 命令后紧跟扩展命令 **FCJ**，**FCJ** 命令后紧跟 2 条 **TY** 命令，给出该元件关联的名称和编码。

3. 绘图格式、宏元件和库

宏元件

宏元件是一个库中的符号，也可以说是一个绘图。通常，电气符号就采用这种表示方法。它由MC命令调用一个宏，命令参数有：调用如下：

```
MC x1 y1 o m n
```

其中， (x_1, y_1) 表示宏元件的位置坐标； o 表示宏元件的方向（方向值乘以 90° 的顺时针方向）； m 等于 1 则宏元件翻转； n 表示宏元件在库中的名称，作为特定的库中的代号。

如果MC命令后跟着FCJ命令，FCJ命令后面需要跟 2 个TY命令，指定该元件的名称和编号。

3.4. FidoCadJ 扩展

从 0.21 版本开始，FidoCadJ 在原 FidoCad 格式的基础上引入了一些好的改进，称为 FidoCadJ 扩展。绘图代码中用FCJ命令表示进行 FidoCadJ 扩展。该命令单独使用没有意义，而是用于表明 FidoCadJ 需要对前一行中定义的东西给出更多的信息。这里有个示例：

```
[FIDOCAD]
MC 40 30 0 0 080
FCJ
TY 50 35 4 3 0 0 0 * R1
TY 50 40 4 3 0 0 0 * 47k
```

在MC命令后紧跟 1 条FCJ命令，表示将对MC命令定义进行扩展，扩展定义为紧跟的 2 条TY命令，定义出关联的名称和编码。

FidoCadJ 可以通过设置选项“完全兼容 FidoCad”禁用所有的扩展，这样，FidoCadJ 完全兼容原 FidoCad。然而，FidoCadJ 绘图文件将继续使用 UTF-8 编码，而不是原 FidoCad 的陈旧的 CP-1252 编码³。Windows 系统上的原 FidoCad 不能理解扩展的FCJ命令。当用原 FidoCad 读取包含扩展命令的 FidoCadJ 绘图时，将提示一些错误，但还是可以打开的，只是打开的绘图可能与 FidoCadJ 打开的绘图有些不同。不同之处取决于使用了哪些扩展，例如：虚线将描绘成连续线，箭头将不绘制，而元件的名称和编号完全相同。

³绘图格式兼容，文件格式不兼容。

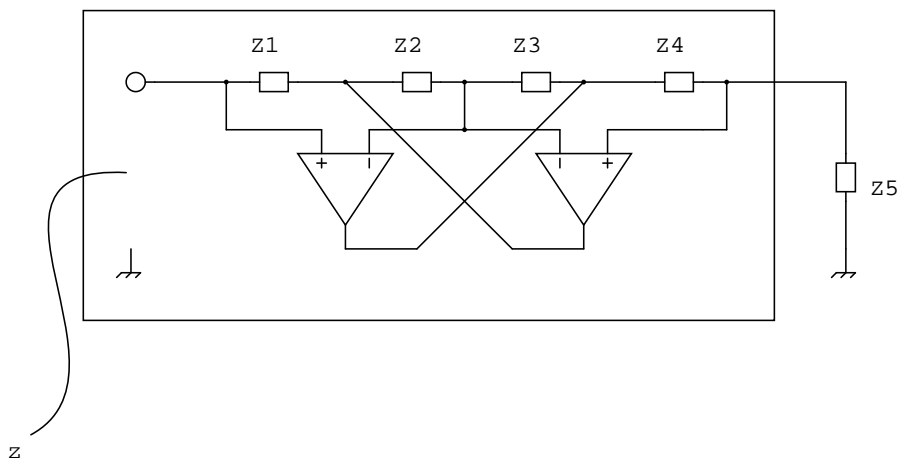


Figure 3.1.: 图 2.19 在 Windows 系统上，用 FidoCad 打开的绘图。

图 3.1 显示了使用 FidoCad 打开 2.19 章节的 FidoCadJ 文件取得的绘图。忽略 FidoCad 错误后，有些细节丢掉了（虚线和箭头），但是总的来说，绘图仍然可以理解。

对比原 FidoCad，一个重要的区别就是 FidoCadJ 可以在输出文件中保存一些配置提示。这是使用 **FJC** 命令，并通常把它置于文件的最前面。下面段落将描述这些示例。

3.4.1. 层设置

扩展命令 **FJC L** 用于定义层颜色和层透明度，如果用户改变了相应的设置，就会在保存的绘图文件中含有以下命令行：

```
FJC L n xxxx yy
```

其中， n 表示层号（整数，取值范围为 $[0,15]$ ）； $xxxx$ 表示层颜色（32 位整数，RGB 值：红色为 16-23 位，绿色为 8-15 位，蓝色为 0-7 位）； yy 表示层透明度（单精度浮点数，0.0 表示完全透明～ 1.0 表示完全不透明）。

扩展命令 **FJC N** 用于定义层名称，如果用户改变了相应的设置，就会在保存的绘图文件中含有以下命令行：

```
FJC N n aaaaa
```

3. 绘图格式、宏元件和库

其中, n 表示层号 (整数, 取值范围为 $[0,15]$); `aaaa` 表示层名称。

如果用户未改变层设置, 绘图文件中不会含有上述扩展命令。此时, FidoCadJ 使用默认的层设置。

3.4.2. 电气连接点设置

扩展命令 `FJC c` 用于定义电气连接点的大小, 如果用户改变了相应的设置, 就会在保存的绘图文件中含有以下命令行:

```
FJC C aaaa
```

其中, `aaaa` 表示电气连接点直径 (正双精度浮点数, 单位为 FidoCadJ 的逻辑单位)。

如果用户未改变电气连接点设置, 绘图文件中不会含有上述扩展命令。此时, FidoCadJ 使用默认的电气连接点设置。

3.4.3. 线条宽度

The stroke width for the “pen” used during the drawing of electrical schematics can be modified. FidoCadJ adopts the command `FJC A` to specify the stroke width of the lines, ovals, Bzier, splines, rectangles and polygons. The width can be a double precision constant. 扩展命令 `FJC A` 用于定义绘制原理图时使用的线条宽度 (双精度浮点数, 包括绘制直线、矩形、多边形、椭圆形、贝塞尔曲线和样条曲线)⁴, 如果用户改变了相应的设置, 就会在保存的绘图文件中含有以下命令行:

```
FJC A aaaa
```

其中, `aaaa` 表示线条宽度 (逻辑单位)。

如果用户未改变线条宽度的设置, 绘图文件中不会含有上述扩展命令。此时, FidoCadJ 使用默认的线条宽度值 (0.5 个逻辑单位)。

3.5. 语法容错

FidoCadJ 设计有语法容错功能, 可以读取非命令行或有语法错误的命令行。然而, 应用程序不可能纠正这些错误 (除非该应用程序具有魔力), 只是简单地跳过这些错误的命令行 (保存时会删除这些错误的命令行)。

⁴在老版本的 FidoCadJ 中, 对于直线 (如绘制直线、矩形和多边形) 和曲线 (如绘制椭圆形和贝塞尔曲线) 的线条宽度分别使用扩展命令 `FJC A` 和 `FJC B bbbb` 指定。但是, 从 0.23.6 版本开始, 已经淘汰了这种做法, 并把它们统一起来了。

命令行中缺少层号参数的语法错误是个例外，FidoCadJ 并不把它当成错误的命令行，而把它默认为 0 层（原理图层），这样处理是为了兼容 FidoCad。

3.6. 库文件格式

库文件的格式很简单，请看以下示例：

```
[FIDOLIB Librairie de base]
{Syboles de base}
[000 Terminal]
LI 100 100 102 100
EV 102 98 106 102
[010 Terminal +]
LI 100 100 102 100
EV 102 98 106 102
LI 103 100 105 100
LI 104 99 104 101
[020 Terminal -]
LI 100 100 102 100
EV 102 98 106 102
LI 103 100 105 100
...
```

第一行定义库的名称（在方括号中，以 FIDOLIB 开头，后面为库的名称，中间以空格隔开）。第二行定义宏元件类型名称（在花括号中），表示接下来定义的宏元件的类型。

每个宏元件定义有头（在方括号中）和一系列命令组成。头是由特定的宏元件名称（在库中的唯一编码）和宏元件描述组成。宏元件名称可以用于 FidoCadJ 脚本，而宏元件描述可以帮助用户概览库中所有的宏元件。命令就是 FidoCadJ 的绘图命令，命令中的坐标点 (100,100) 为宏元件的原点，当调用宏元件绘图时，这个点将作为参考点。在 FidoCadJ 脚本中，宏元件用” library.macro” 定义，并用 MC 命令调用。

一个宏元件的定义中总是可以调用另一个宏元件，但不要进行递归调用（即在宏元件的定义中调用它自己）。

在库文件中，不能包括任何关于 FidoCadJ 的配置信息。换句话说，库文件中不能出现 FJC 命令。

3. 绘图格式、宏元件和库

3.7. 标准库

FidoCadJ 包含 FidoCad 传统支持的 2 个库（即“Stand library”和“PCB Footprints”库），即为 FidoCadJ 的标准库，默认加载它们。

然而通过“文件/选项/启动”设置加载库文件夹，可以覆盖或增加 FidoCadJ 的默认加载库。如果设置的加载库文件夹中存在一个名称为 `FCDstdlib.fcl` 的文件，它将替换标准库；如果存在一个名称为 `PCB.fcl` 的文件，它将替换 PCB 库；如果存在扩展名为 `fcl` 的其它文件，将作为其它库一起被载入。

4. 总结

在前面章节中，我们已经介绍过了如何使用 FidoCadJ 来绘制一张电路原理图和一张简单的 PCB 图。目前阶段，读者应该可以使用 FidoCadJ 进行绘图了。

然而，FidoCadJ 不仅可以用于电子电路的绘图，也可以用于任何其它类型的 2D 绘图，并且多数 2D 类型的绘图可以找到其相关的特定库。

FidoCadJ 是自由程序，其优点是：完全开放给它的用户社区。如此，您的反馈对我很重要（至少让我了解到该项目是否值得继续开发以及它的发展方向）。请和我联系吧！随时欢迎您加入到 SourceForge 论坛并致力于 FidoCadJ 项目¹。

¹<https://sourceforge.net/projects/fidocadj/forums/forum/997486>

A. 操作系统上的特殊性

A.1. MacOSX 系统

Macintosh 的用户（比如我）对早期版本的 FidoCadJ 提出的最多的批评是应用程序在 MacOSX 系统上集成的不好。从 0.21.1 版本开始，FidoCadJ 为此做了努力，以更好地符合 MacOSX 系统应用程序的外观和理念。这就导致了 FidoCadJ 运行于 MacOSX 系统上时出现了一些细微的差异：

- 当运行完整的应用程序（运行 FidoCadJ.app 而不是 FidoCadJ.jar）时，FidoCadJ 默认使用 Quaqua¹ LookAndFeel。当在老式的机器上运行应用程序时，Quaqua 可能会降低其运行性能，此时，通过 Preferences 菜单设置，可以禁用它。
- 菜单工具栏显示在 MacOSX 系统应用程序窗口的合理位置上（应用程序窗口的顶部）。
- 菜单项“Preferences”和“关于 FidoCadJ”出现在 MacOSX 系统应用程序窗口的合理位置上（FidoCadJ 菜单的下面）。
- 应用程序会告知 MacOSX 系统，它可以打开扩展名为 .fcd 的文件，并让这类文件关联一个特定的图标，让人一眼就能看出这是用 FidoCadJ 打开的文件。

A.1.1. 下载并运行 FidoCadJ

FidoCadJ 可以运行在 10.3.9（豹）版本及最新版本的 MacOSX 系统上。这是因为，运行 FidoCadJ 最低需要 Java 1.5 版本，苹果曾经在 10.3.9 版本的操作系统中提供 Java 1.5。由于市场原因，苹果似乎并不想在最新版本的 MacOSX 系统中提供 Java。然而，Java 是运行 FidoCadJ 必不可少的基础部件，因此，您需要下载并安装它。顺便说下，苹果不允许在它的应用商店中发布基于 Java 或 GPL 授权的软件，这就是 FidoCadJ 不可能移植到 iPad 和 iPhone 上的一个最重要的原因。

¹<http://www.randelshofer.ch/quaqua/>

虽然在其它操作系统上，您可以直接使用 Java 打包文件 `fidocad.jar`，然而在 MacOSX 系统上，您需要使用专门定制的 FidoCadJ 应用程序（磁盘镜像）。该应用程序所有的一切看起来就是一个本系统的应用程序。您可以从以下链接下载到运行在 MacOSX 系统上的 FidoCadJ 应用程序的磁盘镜像，

http://sourceforge.net/projects/fidocadj/files/FidoCadJ_MacOSX.dmg/download

然后您可以打开它，把 `FidoCadJ.app` 移动到 `Applications` 文件夹中，在这里，您可以像其它 MacOSX 系统上的应用程序一样地使用它。想要卸载 FidoCadJ，只要把 `FidoCadJ.app` 拖进垃圾桶。

A.2. Linux 系统

Roby IZ1CYN 编写

首先：必须安装 SUN JRE6 和/或 OpenJDK 6 JRE（或是应用程序说明的可使用的早期版本）。A.2.1 章节将介绍终端命令行方式安装程序的方法。A.2.2 章节将介绍图形环境中安装程序的方法。Java 运行时环境配置的好坏，决定了 FidoCadJ 的运行性能。²

A.2.1. 通用的终端安装方法

使用 `wget` 命令下载程序：

```
$ wget http://downloads.sourceforge.net/project/fidocadj/fidocadj.jar
?use_mirror=garr
--00:48:18-- http://downloads.sourceforge.net/project/fidocadj/
fidocadj.jar?use_mirror=garr
=> `fidocadj.jar?use_mirror=garr'
Resolution of downloads.sourceforge.net is being done...
216.34.181.59
Connection to downloads.sourceforge.net|216.34.181.59:80... connected
.
HTTP request sent, waiting for answer... 302 Found
URL: http://garr.dl.sourceforge.net/project/fidocadj/fidocadj.jar
--00:48:24-- http://garr.dl.sourceforge.net/project/fidocadj/
fidocadj.jar
```

²请相信我，如果你喜欢的 Linux 发行版运行 Java 很不稳定，应该就是这个原因！

A. 操作系统上的特殊性

```
=> `fidocadj.jar'
Resolution of garr.dl.sourceforge.net is being done... 193.206.140.34
Connection to garr.dl.sourceforge.net|193.206.140.34:80... connected.
HTTP request sent, waiting for answer... 200 OK
Length: 343,207 (335K) [application/java-archive]

100%[=====>] 343,207      422.48K/s

00:48:30 (420.55 KB/s) - "fidocadj.jar" saved [343207/343207]
$
```

另外，如果上面的方法行不通，您也可以使用任何浏览器从下面的 URL 下载文件：

<http://sourceforge.net/projects/fidocadj/files/fidocadj.jar/download> 并把文件保存到/home/目录中。

创建一个新目录（首先我们要使用su或sudo -s命令成为超级用户，然后才能创建新目录）：

```
# mkdir /usr/bin/fidocadj
```

把下载的文件移到该目录中（把<user>替换成登录账户的用户名，那里有已下载的文件）：

```
# mv /home/<user>/fidocadj.jar /usr/bin/fidocadj
```

把文件设置成可执行的。

```
# chmod +x /usr/bin/fidocadj/fidocadj.jar
```

一定不要忘记切换回普通用户：

```
# exit
```

现在，我们可以执行程序了：

```
$ /usr/bin/fidocadj/fidocadj.jar
```

A.2.2. 图形安装方法

以 Ubuntu 8.04 为例，展示图形化安装方法，在新版本的 Ubuntu 系统上安装的差别不会很大。

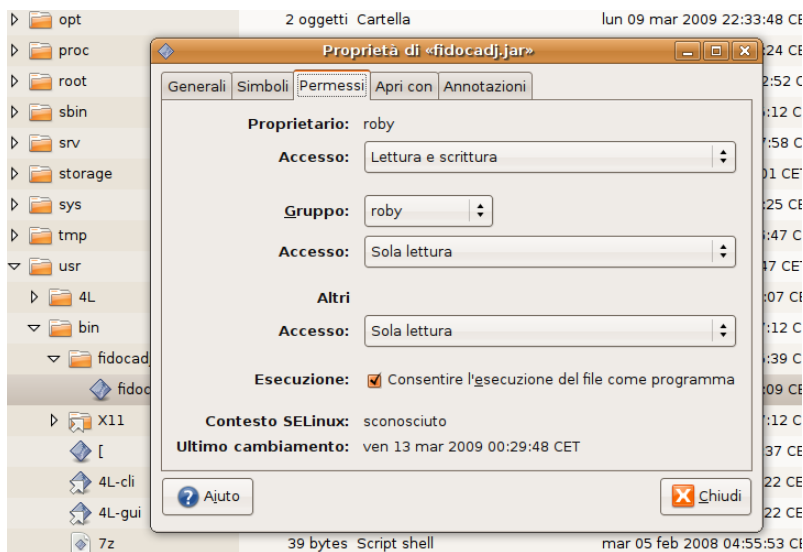


Figure A.1.: Ubuntu 8.04 系统，文件权限设置。

- 用浏览器下载文件，也可以使用 Gwget 或类似工具下载文件。
- 通过文件管理器创建以下目录（如果菜单中没有文件管理器（Nautilus 或 Konqueror 或...），需要用 root 权限加载一个，例如用 `sudo nautilus` 命令从控制台启动 Nautilus）：

```
/usr/bin/fidocadj
```

把刚下载的文件移到该目录中。下载的文件可能在：

```
/home/<user>/
```

- 在文件上右击，会弹出菜单，其中我们打开“rights”分页，在“Allow the execution of the file as a program”前面选项框中打勾，见图 A.1 所示。
- 我们可以打开“Open as”分页，并选择“OpenJDK Java6 Runtime”或“Sun Java6Runtime”，见图 A.2 所示。
- 点击“Close”按钮，现在，我们准备运行 FidoCadJ：双击可执行文件即可运行，或者我们可以把它加入到主菜单。加入主菜单，只需要运行命令 `/usr/bin/fidocadj/fidocadj.jar`。

A. 操作系统上的特殊性

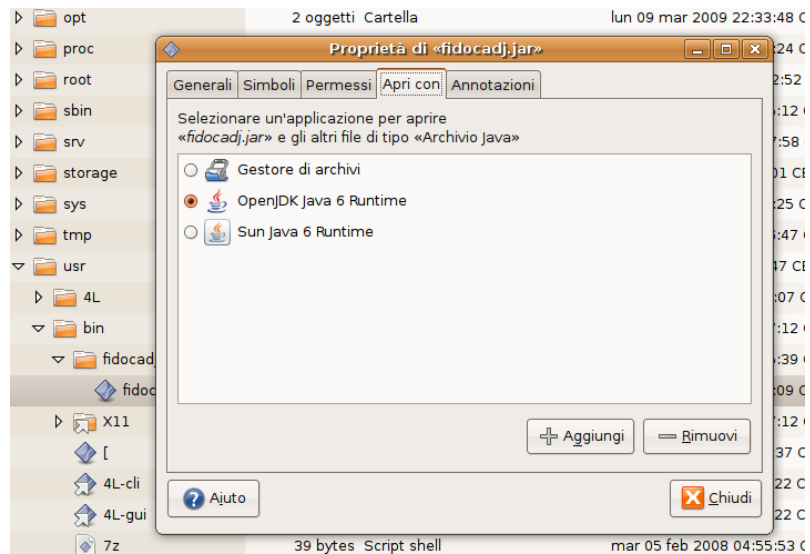


Figure A.2.: Ubuntu 8.04 系统，设置执行程序使用的 Java 虚拟机。

A.3. Windows 系统

从 0.23 版本开始，FidoCadJ 可以识别出 Windows 系统，并且会使用该系统的 LookAndFeel。

A.3.1. 下载并运行 FidoCadJ

通常，如果您的 Windows 系统中已经安装了 Java，您只需要下载 `fidocadj.jar` 文件，双击它即可运行。如果您的 Windows 系统把下载的 jar 文件识别成一个类似 zip 的打包文件，那么可能系统中未安装 Java。Oracle 允许您自由下载和安装最新版本的 Java 运行时：

<http://www.java.com/it/download/>

该手册是在 MacOSX 系统上使用 PDF \LaTeX 编写的。列表由`listings`包编译生成。
`pgf`包用于生成图 2.5。所有其它用到的包来自 CTAN 档案。