

■ 张显库 编著

Visual Basic

工程应用实例

- 用VB绘制工程曲线标准程序实例
- 制作仪表显示控件程序实例
- 用VB实现计算机仿真程序实例
- 数据采集实例
- 计算机通讯实例
- A/D, D/A转换实例
- 实时控制系统实例
- 数据库访问实例
- VB与Matlab混合编程实例
- 用DDE实现VB对Matlab的控制
- 高级图片编程技术



大连海事大学出版社

前 言

本书从实际工程应用的角度总结了笔者多年来从事科研项目研究的经验,以工程应用实例的形式给出了用 VB 6.0 中文版开发的程序。主要内容包括:用 VB 绘制工程曲线标准程序实例;制作仪表显示控件程序实例;用 VB 实现计算机仿真程序实例;数据采集实例;VB 串行通讯应用实例;NetDDE 通讯技术及其应用;用 Winsock 控件进行网络数据通讯;A/D, D/A 转换实例;实时控制系统实例;数据库浏览应用实例;用 ADO 控件访问数据库实例;VB 与 Matlab 混合编程实例;用 DDE 实现 VB 对 Matlab 的控制;高级图片编程技术;制作安装盘实例;用 VB 编写 CIH 病毒隔离程序;用 VB 实现液位量测系统等。

Visual Basic 的学习往往要经历 3 个阶段:初学者(能够简单编程,需要教材和教师指导)→中级程序员(掌握一定的编程技巧,能查找相关资料自学并提高水平)→高级程序设计师(能独立从事一个科研项目的编程,设计并编写复杂的程序,通过相关资料举一反三或独立克服实际工程任务中遇到的编程问题)。学习 Visual Basic 程序设计的最终目的是为实际工程应用服务,而计算机应用的特点是无论学习了多少本书,上机做了多少个例子,如果没有亲自从事过一个实际工程项目的研究,刚刚接手一个任务用 Visual Basic 编程时,都往往无从下手。故在学习计算机语言时,除了要打好基础外,更重要的是要将所学知识活学活用,举一反三。另外,借鉴工程应用实例的开发经验,或根据已有的工程应用实例按自己工程应用的要求稍做修改和完善是完成任务的一个捷径。这就是本书创作的主要动机。

编写 Visual Basic 工程应用实例的难度在于讲解技术时离不开特定的工程问题,而读者因各自的知识背景不同,在阅读本书时遇到的困难的程度也会不同。笔者力图既简化实际工程问题而又能说明白所使用的技术和技巧,但由于笔者从事过的科研项目的背景等原因,可能会给某些读者在使用本书时带来不便,故建议读者要详细研读书中给出的程序,抛开具体的工程载体,达到学会编程原理、掌握编程技术的目的。

笔者分别于 1997 年和 1999 年出版了《VB 实用编程技术——从基础到开发》和《Visual Basic 程序设计》两本书。这两本书主要面向广大的初、中级 Visual Basic 程序设计人员,主要用于程序设计和编程技巧的学习。这两本书被大连海事大学研究生和本科生各专业及成人教育学院高级计算机培训班、大连大学各专业定为教材使用,取得了良好的应用效果。读者来信评价较高,取得了良好的社会效益。本书主要面向中、高级 Visual Basic 程序设计人员,主要用于将 Visual Basic 应用于实际工程,以期对读者使用 Visual Basic 编程解决实际问题有一定的参考价值。

本书可作为计算机软件开发人员及工程软件开发人员的参考书,也可作为高校计算机相关专业本科生与研究生 VB 6.0 程序设计等课程的教材和参考书。

大连海事大学刘人杰教授、曹晓东教授、王淑静副教授和海军大连舰艇学院沈鹤鸣教

授在百忙之中审阅了本书，并提出了宝贵的修改意见，使得本书得到了进一步的完善。笔者在此深表谢意。

在此非常感谢贾欣乐教授、任光教授和王兴成教授给予的不断鼓励，没有他们，此书难以顺利完成。由于时间仓促，水平有限，错误在所难免，敬请读者批评指正！

在本书的写作过程中，得知父亲患了肝癌。父亲以坚强的意志同病魔作斗争，以乐观的态度对待生活，并鼓励我完成该书，为我树立了学习的榜样。父亲去世前，我终于完成了该书的初稿。经过1年多的不断修改和完善，使得本书通过了大连市学术专著资助出版评审委员会和大连海事大学学术专著资助出版评审委员会的评审并获得资助。这充分得益于父亲给予我的精神支持。谨以此书缅怀我的父亲！

作 者

2001年9月

目 录

第 1 章	绪论.....	1
第 2 章	绘制工程曲线标准程序实例.....	6
§2.1	绘制工程曲线的标准程序.....	6
§2.2	屏幕刷新程序实例.....	13
§2.3	制作仪表控件显示数据.....	17
第 3 章	计算机仿真程序实例.....	23
§3.1	用 VB 实现计算机仿真.....	23
§3.2	用 VB 进行计算机仿真实例.....	29
第 4 章	数据文件读入与保存实例.....	38
§4.1	文件输入输出原理.....	38
§4.2	数据采集和数据回放应用实例.....	40
第 5 章	计算机通讯实例.....	45
§5.1	VB 串行通讯原理.....	45
§5.2	VB 串行通讯应用实例.....	47
§5.3	NetDDE 通讯技术及其应用.....	50
§5.4	用 Winsock 控件进行网络数据通讯.....	56
第 6 章	A/D, D/A 转换实例.....	58
第 7 章	实时控制系统实例.....	61
§7.1	船舶航向自动舵控制系统原理及结构.....	61
§7.2	用 VB 进行船舶航向自动舵实时控制.....	63
第 8 章	数据库访问实例.....	68
§8.1	用 VB 存取数据.....	68
§8.2	VB 数据库应用实例.....	73
§8.3	动态访问 VB 数据库.....	74
§8.4	数据库浏览应用实例.....	75
§8.5	用 ADO 控件访问数据库实例.....	77
第 9 章	VB 与 Matlab 混合编程实例.....	85
§9.1	Matlab 语言简介.....	85
§9.2	VB 与 Matlab 混合编程技术.....	87
§9.3	VB 与 Matlab 混合编程实例.....	89

§9.4 利用DDE实现VB对Matlab的控制.....	97
第10章 高级图片编程技术.....	102
§10.1 用图片控件实现动画.....	102
§10.2 制作图片显示特殊效果.....	105
§10.3 用大图片制作工具条.....	111
第11章 制作安装盘实例.....	113
第12章 VB系统信息控件应用.....	121
§12.1 系统信息控件简介.....	121
§12.2 系统信息控件应用示例.....	124
第13章 用VB编写CIH病毒隔离程序.....	127
§13.1 CIH病毒简介.....	127
§13.2 用VB编写CIH病毒隔离程序.....	127
第14章 用VB实现液位量测系统.....	135
§14.1 油料仓库液位量测系统构成及实现要求.....	135
§14.2 用VB实现液位量测系统.....	136
参考文献.....	177

第1章 绪 论

本章介绍了 VB 的概念,用 VB 编程的优点以及安装 VB 6.0 的注意事项。从加深读者理解的角度,简明扼要地综述了 VB 6.0 程序设计的基础知识,概括了 VB 6.0 中 7 个常用窗口的作用与关系,并介绍了 VB 6.0 要保存的程序类型及怎样创建、编译和执行 VB 6.0 程序。

学习 Visual Basic 程序设计的最终目的是为实际工程应用服务,而计算机应用的特点是无论学习了多少本书,上机做了多少个例子,如果没有亲自从事过一个实际工程项目的研究,刚刚接手一个任务用 Visual Basic 编程时,都往往无从下手。故在学习计算机语言时,除要打好基础外,更重要的是要将所学知识活学活用,举一反三。另外,借鉴工程应用实例的开发经验,或根据已有的工程应用实例按自己工程应用的要求稍做修改和完善是完成任务的一个捷径。

1976 年,现任美国微软公司总裁 Bill Gates(比尔·盖茨)先生和 Paul Allen 开发出了 Basic 语言的早期版本。1998 年 12 月微软公司正式推出 Visual Basic 6.0 中文版本。近 3 年来,Visual Basic 几乎是一年更新一个版本,说明了对 Visual Basic 程序开发的投资力度和重视程度。可以说 Visual Basic 是 Bill Gates 先生的招牌,他是靠 Basic 语言起家的,故 Visual Basic 的发展速度和投资力度是有保证的;另一方面,Visual Basic 的迅猛发展也是其应用广泛和受用户喜爱的一个标志。

特别值得说明的是,微软公司在最新推出的 Office 97 和 Office 98 办公自动化软件中全部统一改用 VB 作为编程语言。如 Office 97 中的 Word 8.0 文字处理软件,如果想有特殊的效果,如加进朗读的声音或加入对话框等,就可直接用 VB 来编程,而不是用 Word 7.0 版本以前相近的 Word Basic 编程。由此可见微软公司对 VB 的重视及其应用的广泛性和受欢迎的程度。

随着 VB 版本的不断升高和发展,VB 将会赢得越来越多的用户。比尔·盖茨先生说:“没有什么功能和任务是 VB 不能完成的。”常言道:工欲善其事,必先利其器。对于一名软件开发人员来说,选择功能强大和效率的优秀开发工具会起到事半功倍的作用。就让我们沿着比尔·盖茨先生所指引的方向,步入 VB 的世界,定能给你一个意外的惊喜!

1.VB 的概念及编程优点

VB 是一种面向对象和用户图形界面的,具有事件驱动编程机制的可视程序设计工具。用 VB 编程具有以下 5 个优点:

(1)用户图形界面编程工作量少。一个实际的软件编程可分为图形界面编程和非图形界面编程两部分,分别代表用户可见和不可见的部分。由于 VB 编程面向用户图形界面,用户只需按设计风格和实际要求将 VB 系统环境所提供的元素(包括按钮、图片、文本框、标签等)放在窗体中,然后更改其属性表即可。就像用各种事先做好的模板来绘画一样,只

需按脑海中构想的图找到相应的模板在图画纸的相应位置上画好，然后涂上颜色就行了。所以说用 VB 编程，用户可见部分的图形界面编程的工作量大大降低，从而缩短了软件的开发周期。

(2)VB 编程以事件驱动为机制。用户的任何一个动作都产生一个事件，如按鼠标左键、击键等，如果给一个事件编程，则当该事件发生时就会产生执行效果。可以给所有可能发生的事件都编程，则任何一个事件的发生都会产生效果。就像玩捅开纸格中奖游戏一样，如果每一个纸格后面都藏有奖品，则每捅开一次纸格都会中奖，只不过奖品不同而已。也有可能纸格后面没有藏奖品(相当于没有给某一事件编程)，则捅开该纸格后将得不到奖品(相当于该事件发生后将没有执行效果)。同时其事件驱动的编程机制使得程序设计起来容易，读起来直观，使用起来方便。

(3)VB 采用面向对象的程序设计方法，程序代码具有良好的可重用性、可扩充性和可修改性。

(4)VB 6.0 作为 Windows 编程技术之一，具有 Windows 环境所具有的五大大优点，即标准的图形用户界面、动态链接(DDL)、多任务、设备独立性及直接操作特性。

(5)由于 VB 的用户越来越多，很多计算机软件公司相继开发出适用于 VB 的工具箱，目前由第三方(即非 Microsoft 公司和用户以外的计算机软件开发方)开发的 VB 工具箱已达上百种，故 VB 程序设计随时间的推移会越来越简单。

2. 安装 VB 6.0 的注意事项

VB 6.0 也称 VB 98，其最佳的安装环境为 Windows 98。如果在 Windows 95 中安装 VB 6.0 就会遇到一系列问题，如 Internet Explorer 浏览器过期等(VB 6.0 的安装需要 4.0 以上版本的 Internet Explorer 浏览器，即 IE 4 或 IE 4.01)，故建议读者在 Windows 98 环境中安装 VB 6.0。一般安装约需 100 MB 的硬盘空间，全部安装约需 150 MB 的硬盘空间。为使 VB 6.0 高效运行，计算机最好配置 64 MB 以上内存及 266 MHz 以上主频。在购买的光盘中，有时 VB 6.0 不是单独的安装程序，而是 Microsoft Visual Studio 即微软可视工作室中的一个组件。微软可视工作室包括 VB 6.0、VC++ 6.0、Visual FoxPro 6.0、Visual InterDev、Visual J++、Visual SourceSafe 等，此时如果要安装 VB 6.0，可安装 Microsoft Visual Studio，然后选择安装其中的 VB 6.0 即可。

3. VB 6.0 常用 7 个窗口的作用与关系

图 1-1 给出了 VB 6.0 的环境及程序设计时常用的 7 个窗口布局。从该图可以看出，VB 6.0 环境包括主窗口(最上面的 3 行，由标题条、菜单和工具条组成)、工程文件窗口(或称项目组窗口，Project Group，位于右侧的上部)、属性窗口(Properties，在右侧的中间)、窗体布局窗口(Form Layout，在右侧的底部)、窗体窗口(Form1，在中间的上部)、代码窗口(Code，在中间的底部)、工具箱窗口(Toolbox，位于左侧)7 个窗口，窗口之间由标准的关闭窗口按钮区分。它们的作用分别是：

(1)主窗口，包括 13 个下拉式菜单。VB 6.0 程序设计时所用的命令都体现在菜单命令中，且把常用的 21 个命令做成工具按钮放到工具条中，用起来更方便。每个工具按钮的含

义在用鼠标指向该按钮时会给出提示。

(2)工程文件窗口，用于显示运行正在编写的 VB 6.0 程序所需的文件清单，有 3 个按钮，即 View Code(查看代码)、View Object(查看对象)和 Toggle Folders(查看文件夹)，分别用来查看程序代码、查看程序界面和显示窗体所在文件夹。

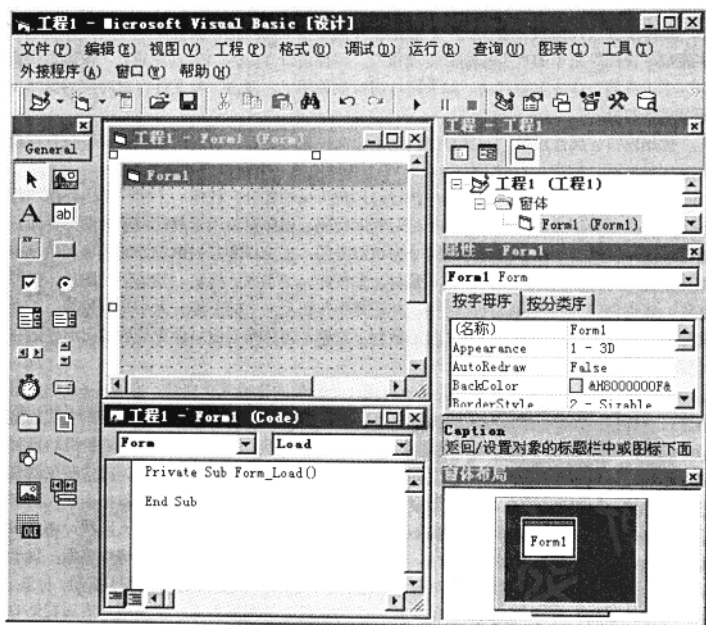


图 1-1 VB 6.0 的环境及 7 个窗口布局

(3)属性窗口，是描述对象性质的形式化的可见表格，用来设置对象的外观和行为。注意：不必设置每个对象的所有属性，每个属性都有一个缺省的约定值，如窗体的缺省约定背景色为灰色，如果你认为灰色可以，那么就不用改了。

(4)窗体布局窗口，确定用户所设计的图形界面在计算机屏幕上显示的位置，可用鼠标拖动窗体布局窗口中的窗体界面到给出的计算机屏幕的适当位置上放手即可。此时真正的窗体界面位置并没发生变化，而是在程序运行时，其位置才移到在窗体布局窗口中所设计的位置上。

(5)窗体窗口，是程序运行时用户能看见的图形界面。

(6)代码窗口，在启动时该窗口是不可见的，可用鼠标双击窗体窗口激活代码窗口，在该窗口中进行编程，赋予整个系统以灵魂。双击窗体或其中对象，单击工程文件窗口的 View

Code 查看代码按钮都可进入代码窗口。

(7)工具箱窗口,设计用户界面所需的各种控件元素都由工具箱窗口提供,这里的控件是指工具箱窗口中的各种工具,如按钮、文本框和标签等。本书中有时也称控件为对象,两者的区别是对象包含的内容更广一些,除了工具箱窗口中的工具可称为对象外,窗体也可称为对象。

如果工程文件窗口、属性窗口、窗体布局窗口和工具箱窗口被关闭了,则可分别通过主窗口“视图”菜单的“工程资源管理器”、“属性窗口”(或按 F4 功能键)、“窗体布局窗口”、“工具箱”命令重新启动。如果窗体窗口被关闭了,双击工程文件窗口对应的窗体项即可重新打开。

代码窗口、属性窗口、工具箱窗口、工程文件窗口、窗体布局窗口和主窗口只有在设计时才被激活,程序执行时不能应用。VB 6.0 环境的 7 个窗口之间的关系为:

窗体窗口是程序的界面,是程序执行时用户可见的部分。任何一个程序都由界面和程序代码两大部分组成,代码窗口用于编写程序代码,窗体窗口用于设计程序的图形界面,故可以说,其他 5 个窗口都是为这两个窗口服务的:主窗口为其提供设计的环境和命令;工具箱窗口为其提供界面设计的基本素材,如按钮、文本框、标签、复选框、图片、文件框等;属性窗口为其提供设计的风格,如窗体中按钮上文字的字体、大小及颜色等;工程文件窗口为其提供所用的窗体及工具函数程序清单;窗体布局窗口设置最后的图形界面在计算机屏幕上显示的位置。

VB 6.0 环境 7 个窗口之间的关系也可用一个形象的比喻深化理解:即 VB 6.0 的主窗口像一个厨房,所有与做菜有关的物件和材料都在厨房中,根据实际任务编程序(在代码窗口中进行)就像在厨房中做一道特定的菜,如西红柿炒鸡蛋。工具箱窗口就像各种调料及材料,西红柿炒鸡蛋当然就要从“工具箱”中拿西红柿、鸡蛋、盐等材料了。把这些东西放到炒锅中炒,至于怎么炒,放油、放盐、放料翻炒的过程和工序,实际上就相当于代码窗口中的程序了。而属性窗口像一块砧板或其他什么,可以把西红柿切成什么形状,切成多大块等等。做好的菜要放到盘子中去,最后呈现在顾客面前的是一盘西红柿炒鸡蛋,盘子就相当于窗体窗口,一盘西红柿炒鸡蛋,就相当于在窗体中按实际要求加入了按钮、标签、图片等控件形成了具有一定风格且能完成一个特定任务的图形界面,它是用户最后看见的部分。最后把这盘菜放到餐桌的什么位置上,就相当于窗体布局窗口的工作了。那么工程文件窗口像什么呢?工程文件窗口当然是菜单了,它定义了该顿饭共有几道菜,每道菜都是什么。通过这样一个形象的比喻,VB 6.0 的 7 个窗口之间的关系就已非常明白,相信你工作起来一定会得心应手。

VB 6.0 的其他重要窗口还有调试程序时的“立即”窗口、查看对象时的“对象浏览器”窗口等,方便用户在不同场合下的应用。

4.VB 6.0 要保存的程序类型

随时保存文件是一种良好的工作作风,如果你等所有程序都做完或下机时再保存文件,非常有可能你所做的工作会前功尽弃。因为你的计算机随时可能由于误操作、病毒发作、断电(停电或别人拔计算机外围设备插头而拔错插头或接线盒接触不良造成的人为断电)等

各种原因而死机，假设你没随时保存文件，所有程序就都白输入了，界面也白设计了，工作效率将大大降低，有时损失还是很大的。

VB 6.0 程序必须要保存的文件有两类：(1)项目文件或称工程文件，以.vbp 为后缀，保存一个项目的窗体及模块目录。(2)窗体文件，以.frm 为后缀，保存窗体代码及显示布局。

有些特殊的或复杂的 VB 6.0 程序可能有下列类型的文件要保存：

- Basic 文件 (*.bas)—— 列出所有标准模块。
- 工程文件 (*.vbg)—— 保存一个项目组由多个项目组成的文件目录。
- 类文件 (*.cls)—— 列出所有类模块。
- 属性页文件 (*.pag)—— 列出所有属性页。
- 用户定义控件文件 (*.ctl)—— 列出所有用户定义的控件。
- 文档对象文件 (*.dob)—— 列出所有的文档对象。

保存文件的操作是在主窗口的“文件”菜单中选择“保存工程”命令或在工具条中用鼠标点标有磁盘图形的工具按钮，在起文件名时，一般不超过 8 个字符。上述保存的文件正常情况下均保存在硬盘中，为预防病毒和误操作等造成损害，最好定期把上述文件保存入软盘或备用硬盘中。

注意：1 个汉字相当于 2 个字符，另外，文件名最好以容易记忆为标准，如例 2-1 起名为 lz21，这样以后看到文件名就可想起文件的内容，以便查询。

5. 创建 VB 6.0 编译执行程序

在 VB 6.0 程序还没有全部设计好时，若要调试程序，则是在 VB 6.0 环境中，用主窗口中菜单的执行命令或工具条上执行按钮进行执行，看看执行效果如何。如果效果不好或程序有错误，就要重新设计或修改错误。这样的执行是在 VB 6.0 环境中一句一句地翻译执行，可称为解释执行。它的优点是适用于程序调试阶段，可根据程序执行结果随时修改程序；缺点是需要 VB 6.0 环境，占内存，速度慢。

如果程序全部调试通过，准备交付用户使用，就要做成可执行程序，脱离 VB 6.0 环境，可称为编译执行。具体做法是：在 VB 6.0 主窗口中选“文件”菜单中的“生成工程 1.exe”或“生成 lz21.exe”(或任意保存的文件名)命令，在给出的窗口中输入一个合适的文件名，以.exe 为后缀，VB 6.0 将之编译为可执行的机器代码，可直接在 Windows 系统中运行，而不需要 VB 6.0 环境。但该程序运行还需相应的动态链接库文件(.dll)和标准控件文件(.ocx)，如动态链接库文件 MSVBVM60.DLL 和 OLEAUT32.DLL 是最基本的，而其他的动态链接库文件和标准控件文件要根据实际情况决定是否需要。如系统中用到串行通讯控件(标准的工具箱窗口中没有)，则需要增加相应的动态链接库文件和与之配套的标准控件文件，即 MSCOMCH32.DLL 和 MSCOMM32.OCX 文件，它们在 Windows 目录的 System 子目录下。如果用户系统中没有 VB 6.0 环境，别忘了将它们拷入 Windows 目录的 System 子目录下。当然作为一个商品或工程软件交给用户使用前，最好将它们和做好的可执行程序做成系统安装盘，让用户直接安装使用，系统运行所需的动态链接库文件和标准控件文件在用 Visual Basic 制作安装盘时会自动加入到安装程序中。

第2章 绘制工程曲线标准程序实例

本章给出了绘制工程曲线的标准程序及应用实例。

§ 2.1 绘制工程曲线的标准程序

在做实际的工程项目时,要将计算出的重要数据以曲线的形式显示出来,以便分析,故绘制曲线是实际工程程序设计的一个重要环节,也是必不可少的环节。如果绘制工程曲线的程序不标准,则每次编写不同情况的工程曲线程序时,都需重新更改程序,重复劳动大,延长了软件的开发周期;而一个适用于不同情况的标准工程曲线程序对软件开发人员来说,只需要填写不同的人参数即可,起到了事半功倍的作用。

绘制工程曲线的标准程序由3部分组成,即

- (1)静态背景图形标准程序,包括曲线的坐标、数据标注、网格和说明文字等。
- (2)动态曲线的绘制程序,包括绘制曲线的函数、程序控制、结束条件等。
- (3)屏幕刷新程序,包括换屏、曲线刷新等。

1. 静态背景图形标准程序

在编写静态背景图形的标准程序之前,首先研究一下绘制静态背景图形有哪些基本要素。图2-1 给出一个标准工程曲线的样板,观察该曲线图可知,一般情况下,绘制静态背景图形的基本要素(也就是将来作为标准程序的人参数)有:

- (1)坐标方框的左上角坐标、右下角坐标。
- (2)x轴标度坐标间隔,y轴标度坐标间隔,如200和300之间的坐标间隔。
- (3)x轴标度数据起点,x轴标度数据步长,y轴标度数据起点,y轴标度数据步长。如x轴标度数据的起点为200,步长为100。
- (4)x轴说明起始坐标,x轴说明坐标距底边框距离,x轴说明字符串。
- (5)y轴说明起始坐标,y轴说明坐标距左边框距离,y轴说明字符串。
- (6)图的标题起始x坐标,标题y坐标距上边框距离,图的标题。
- (7)横虚线宽,横虚线加上虚线缝隙总宽,竖虚线宽,竖虚线加上缝隙总宽。
- (8)x轴数据标注左移调整距离(使标注的

数据与网格线对齐,因有时标注的数据为1位数,而有时标注的数据为3位数或其他多位数),x轴数据标注距底边框距离,y轴数据标注距左边框距离,y轴数据标注上移调整距离。

注意:绘图坐标和显示在坐标轴上的标注数据是不同的,二者之间需进行标度变换。如x轴绘图坐标的起始点和终止点分别为100和1100个作图单位,而x轴的数据标注为0~10 s,二者之间进行简单代数转换,称为标度变换。当然可将二者人为地设为相同,其好

处是省去了标度变换环节,简化了程序,但有时会造成横纵坐标疏密程度相差较大,曲线显示不美观。如横坐标为 $0 \sim 10\text{ s}$,纵坐标为 $0 \sim 1\,000\text{ V}$ 电压,如果不作标度变换,整个图形就细长,不美观。另外,建议窗体的背景色采用白色,重要的曲线用深颜色显示,以便输出到纸后墨且曲线清晰。

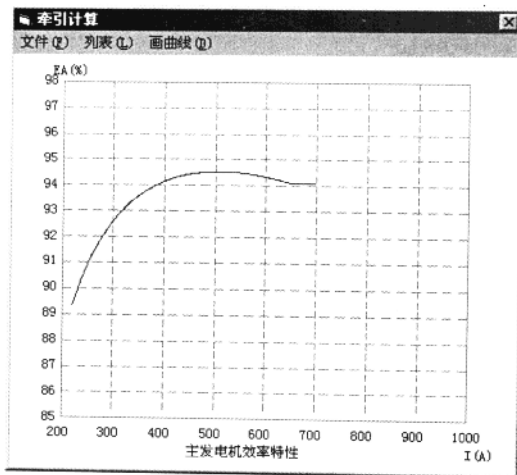


图 2-1 一个标准工程曲线样板

下面给出一个标准的静态背景图形程序代码。

绘图形程序

```
Sub ht0(zsx As Integer, zsy As Integer, yxx As Integer, yxy As Integer, xi As Integer, _
yi As Integer, x0 As Single, xstep As Single, y0 As Single, ystep As Single, _
xtx As Integer, xty As Integer, xt As String, ytx As Integer, yty As Integer, yt As String, _
titlex As Integer, titley As Integer, title As String, hxw As Integer, hxt As Integer, _
sxw As Integer, sxt As Integer, dataxx As Integer, dataxy As Integer, datayx As Integer, datayy As Integer)
'边框指曲线图的四边框,不是窗口的边框
'zsx 左上角 x 坐标, zsy 左上角 y 坐标, yxx 右下角 x 坐标, yxy 右下角 y 坐标
'xi 为 x 轴标度坐标间隔, yi 为 y 轴标度坐标间隔
'x0 为 x 轴标度数据起点, xstep 为 x 轴标度数据步长, y0 为 y 轴标度数据起点, ystep 为 y 轴标度数据步长
'xtx 为 x 轴说明起始坐标, xty 为 x 轴说明坐标距底边框距离, xt 为 x 轴说明字符串
'ytx 为 y 轴说明坐标距左边框距离, yty 为 y 轴说明起始坐标, yt 为 y 轴说明字符串
'titlex 为图的标题起始 x 坐标, titley 为标题 y 坐标距上边框距离, title 为图的标题
'hxw 为横虚线宽, hxt 为横虚线加上虚线缝隙总宽, sxw 为竖虚线宽, sxt 为竖虚线加上缝隙总宽
```

'dataxx 为 x 轴数据标注左移调整距离, dataxy 为 x 轴数据标注距底边框距离, datayx 为 y 轴数据标注距左边框距离, datayy 为 y 轴数据标注上移调整距离

Dim i, j As Integer

Form1.Line (zsx, zsy)-(yxx, yxy), RGB(0, 255, 0), B '画曲线边框

For i = 0 To ((yxy - zsy) / yi - 2) '画横的虚线网格

For j = zsx To yxx - hxw Step hxt

Form1.Line (j, zsy + (i + 1) * yi)-(j + hxw, zsy + (i + 1) * yi), RGB(0, 255, 0)

Next j

Next i

For j = 0 To ((yxx - zsx) / xi - 2) '画竖的虚线网格

For i = zsy To yxy - sxw Step sxt

Form1.Line (zsx + xi * (j + 1), i)-(zsx + xi * (j + 1), i + sxw), RGB(0, 255, 0)

Next i

Next j

Form1.CurrentX = titlex '标题显示

Form1.CurrentY = zsy + titley

Form1.Print title

Form1.CurrentX = xtx 'x 轴说明

Form1.CurrentY = yxy + xty

Form1.Print xt

i = 0

Do While i <= (yxy - zsy) / yi '标注 y 轴数据

Form1.CurrentX = zsx - datayx

Form1.CurrentY = yxy - i * yi - datayy

Form1.Print Str\$(y0 + i * ystep)

i = i + 1

Loop

j = 0

Do While j <= (yxx - zsx) / xi '标注 x 轴数据

Form1.CurrentX = zsx + j * xi - dataxx

Form1.CurrentY = yxy + dataxy

Form1.Print Str\$(x0 + j * xstep)

j = j + 1

Loop

Form1.CurrentX = zsx - ytx 'y 轴说明

Form1.CurrentY = yty

Form1.Print yt

End Sub

建议读者在调用该程序之前，先在纸上画好静态背景图形的草图，然后设计好其入口变量的值，再将各值填入该程序即可。

2.动态曲线的绘制程序

动态绘制曲线的原理是在已绘制好的静态背景图形上，按相应的标度变换将计算出的数据标绘在当前坐标图上。其步骤是：先定义好 x 坐标初值，然后标度变换计算出 y 坐标的初值，用 CurrentX, CurrentY 定义当前坐标，然后用计数循环或条件语句判断绘图是否结束，最后用画线命令或画点命令绘制动态曲线。全部程序可由某一事件驱动或由定时器事件驱动(每次绘制一步)。

下面给出图 2-1 所示的主发电机效率特性动态曲线绘制的程序。

'通过菜单启动绘制主发电机效率特性曲线程序

'也可通过点按钮事件启动，当然需在窗体中加入按钮，将菜单名替换成按钮名称

'也可通过点窗体事件启动，将菜单名替换成窗体名称

```
Private Sub mnudrawea_Click()
```

```
Dim x, x0, y0, yy0, yy1, xe, ye, xx0, xx1, x1, xi As Integer
```

```
x0 = 120 'x 轴绘图初始坐标
```

```
yy0 = 10 'y 轴顶部标注数据的坐标
```

```
yy1 = 140 'y 轴底部标注数据的坐标
```

```
xe = 100 'x 轴绘图初始坐标和其对应的标注数据之差,标注数据一初始坐标
```

```
y0 = 100 'y 轴数据表示法比例系数,如 85%,标注为 85,比例系数为 100
```

```
xx0 = 98 'y 轴顶部标注的数据
```

```
xx1 = 85 'y 轴底部标注的数据
```

```
x1 = 600 '绘图结束坐标
```

```
xi = 1 '绘图步长
```

```
Form1.Cls
```

```
Scale (0, 0)-(1000, 160)
```

```
ht0 100, 10, 900, 140, 100, 10, 200, 100, 85, 1, 900, 10, "I(A)", 20, 3, "EA(%)", 350, 140, "主发电机效率特性",  
20, 30, 3, 4, 40, 3, 50, 3
```

'经标度变换的 y 轴绘图初始坐标

```
y0 = yy1 + ((eaofi(x0 + xe) * yb) - xx1) * (yy1 - yy0) / (xx1 - xx0)
```

```
Form1.CurrentX = x0
```

```
Form1.CurrentY = y0
```

```
For x = x0 To x1 Step xi
```

```
Form1.Line -(x, yy1 + ((eaofi(x + xe) * yb) - xx1) * (yy1 - yy0) / (xx1 - xx0)), RGB(0, 0, 255)
```

```
Next x
```

```
End Sub
```

‘主发电机效率特性函数

Function eaofi(ByVal i As Single) As Single ‘主发电机效率

Dim k1, k2, k3, kkk As Single

pi = (p1 - p2) * 1000

If i > 650 Then i = 650 ‘电流限制条件

If i < 207 Then i = 207

k1 = 1 - 2 * a2 * ne ^ -0.7 / (6 * i)

k2 = 4 * a2 * ne ^ -0.7 * pi / (6 * i) ^ 2

k3 = a1 * (6 * i) ^ 2 / pi + a2 * ne ^ -0.7 / pi + 0.0001295 * ne ^ 2.52645 / pi - 1

kkk = k1 ^ 2 - k2 * k3

If kkk < 0 Then End

eaofi = (Sqr(k1 ^ 2 - k2 * k3) - k1) / (k2 / 2)

End Function

为了便于读者读懂上面的函数，下面给出主发电机效率特性函数的数学公式。主发电机效率 E_A 是电流 I 的函数：

$$E_A = \frac{\sqrt{k_1^2 - k_2 \cdot k_3} - k_1}{k_2 / 2}$$

其中： $k_1 = 1 - 2A_2 \cdot NE^{-0.7} / (6I)$ ， $k_2 = 4A_2 \cdot NE^{-0.7} \cdot P_I / (6I)^2$ ， $P_I = (P_1 - P_2) * 1000$ ，

$k_3 = A_1 \cdot (6I)^2 / P_I + A_2 \cdot NE^{-0.7} / P_I + 0.0001295 NE^{2.52645} / P_I - 1$ 。

计算主发电机的效率要已知柴油机转速 $NE/(r \cdot \min^{-1})$ 、主机功率 P_1/kW 、辅机功率 P_2/kW 、常数 A_1 和 A_2 。

将下面的变量定义为模块级变量，然后将窗体的背景颜色 BackColor 属性改为白色，AutoRedraw 属性改为 True，BorderStyle 属性改为 3-Fixed Double，程序运行后，点设计的驱动菜单(按钮或窗体)即可显示图 2-1 所示的曲线。

Const ne! = 1800

Const p1! = 1250

Const p2! = 116

Const a1# = 0.002179696201

Const a2# = 30.80687869

Dim pi As Single

3.屏幕刷新程序

如果曲线能够在一屏内显示完毕，则不用编写屏幕刷新程序，故屏幕刷新程序不是必需的，可根据实际情况决定是否编写。屏幕刷新程序与绘制图形的结束条件有着密切的关

系，一般来说，绘制图形结束条件有：

- (1)按某一键人为地结束绘图。
- (2)按步长或时间结束绘图。
- (3)满足一定的条件后结束绘图。

如果采用按步长或时间结束绘图，则可通过设计使整个曲线显示在一屏，避免了屏幕刷新程序的编写；而按键和满足一定的条件后结束绘图，事先很难设计使整个曲线显示在一屏，则需要编写屏幕刷新程序。

下面给出一种简单的图形换屏原理，即

- (1)采用相同的静态背景图形，用“第几屏”字样来区分不同的屏幕显示。
- (2)绘制完一屏后用窗体清除命令清除图形，重新绘制静态背景图形。
- (3)从初始坐标开始绘制。

本方法的优点是编程简单，缺点是如果计算机速度慢，可以看出图形在清除，不美观。该缺点可通过不用窗体清除命令，而将曲线的数据记录在数组中，然后用背景色重画曲线来擦除，再重画背景图形加以克服，当然付出的代价是编程复杂。

还有一种方法是在换屏时不用窗体清除命令，而将曲线的数据记录在数组中，然后重新设置窗体的 DrawMode 属性为特定的值，再将记录在数组中的曲线数据重画一遍，还需重画背景图形才可实现曲线的擦除功能。

例如，设图 2-1 所示曲线图形所在的窗体的背景颜色 BackColor 属性为白色即 RGB(255,255,255)，窗体的前景颜色 ForeColor 属性为黑色即 RGB(0,0,0)，用 Line 语句画的网格线为绿色即 RGB(0,255,0)，用 Line 语句画的曲线为蓝色即 RGB(0,0,255)。现在想通过设置窗体的 DrawMode 属性为 12 即或非笔模式，将显示为蓝色的曲线从前景图形中擦除掉。下面通过改写事件过程程序为如下的形式实现刚刚画出的曲线被擦除的功能。

‘通过菜单启动绘制主发电机效率特性曲线程序

‘也可通过点按钮事件启动，当然需在窗体中加入按钮，将菜单名替换成按钮名称

‘也可通过点窗体事件启动，将菜单名替换成窗体名称

```
Private Sub mnudrawea_Click()  
Dim x, x0, y0, yy0, yy1, xe, ye, xx0, xx1, x1, xi As Integer  
x0 = 120   'x 轴绘图初始坐标  
yy0 = 10   'y 轴顶部标注数据的坐标  
yy1 = 140  'y 轴底部标注数据的坐标  
xe = 100   'x 轴绘图初始坐标和其对应的标注数据之差  
yb = 100   'y 轴数据表示法比例系数,如 85%,标注为 85  
xx0 = 98   'y 轴顶部标注的数据  
xx1 = 85   'y 轴底部标注的数据  
x1 = 600   '绘图结束坐标  
xi = 1     '绘图步长  
Form1.Cls  
Scale (0, 0)-(1000, 160)
```

```

ht0 100, 10, 900, 140, 100, 10, 200, 100, 85, 1, 900, 10, "I(A)", 20, 3, "EA(%)", 350, 140, "主发电机效率特性",
20, 30, 3, 4, 40, 3, 50, 3
'经标度变换的 y 轴绘图初始坐标
y0 = yy1 + ((eaofi(x0 + xc) * yb) - xx1) * (yy1 - yy0) / (xx1 - xx0)
Form1.CurrentX = x0
Form1.CurrentY = y0
For x = x0 To x1 Step xi
    Form1.Line -(x, yy1 + ((eaofi(x + xc) * yb) - xx1) * (yy1 - yy0) / (xx1 - xx0)), RGB(0, 0, 255)
Next x
Form1.DrawMode=12
y0 = yy1 + ((eaofi(x0 + xc) * yb) - xx1) * (yy1 - yy0) / (xx1 - xx0)
Form1.CurrentX = x0
Form1.CurrentY = y0
For x = x0 To x1 Step xi
    Form1.Line -(x, yy1 + ((eaofi(x + xc) * yb) - xx1) * (yy1 - yy0) / (xx1 - xx0)), RGB(0, 0, 255)
Next x
DrawMode = 13 '设置回原绘图模式,重画背景图形
ht0 100, 10, 900, 140, 100, 10, 200, 100, 85, 1, 900, 10, "I(A)", 20, 3, "EA(%)", 350, 140, "主发电机效率特性",
20, 30, 3, 4, 40, 3, 50, 3
End Sub

```

分析: 原来绘制的曲线为蓝色(背景色), 再用蓝色(前景色)重新绘制一遍, 只不过绘制模式改为 12 或非笔模式即背景色和前景色的反进行或操作, 两个同色进行或非操作后为白色(或操作可简单地理解为颜色的每位相或, 若其中一位为 255, 则该位为 255, 否则为 0), 还原为窗体的背景色, 实现了曲线擦除的功能(因原曲线是绘制在绿色的网格线上, 或非操作后, 被原曲线覆盖的绿色网格线会被白色擦除, 故需重新绘制背景图形)。

注意: 如果 Line 命令后面没有颜色设置值, 则前景颜色用窗体的前景色代替。如果不是在原曲线的同一位置重新绘制且所绘制位置无图形时, 背景色应用窗体的背景色。

表 2-1 给出了 DrawMode 属性的详细解释。

表 2-1 DrawMode 属性的详细解释

属性设置值	属 性 解 释
1	黑色
2 非或笔	与设置值 15 相反
3 与非笔	背景颜色和前景颜色的反相与
4 非复制笔	设置值 13 (Copy Pen) 的反
5 与笔非	前景颜色与背景颜色的反相与
6 反转	背景颜色的反
7 异或笔	前景颜色与背景颜色求异或