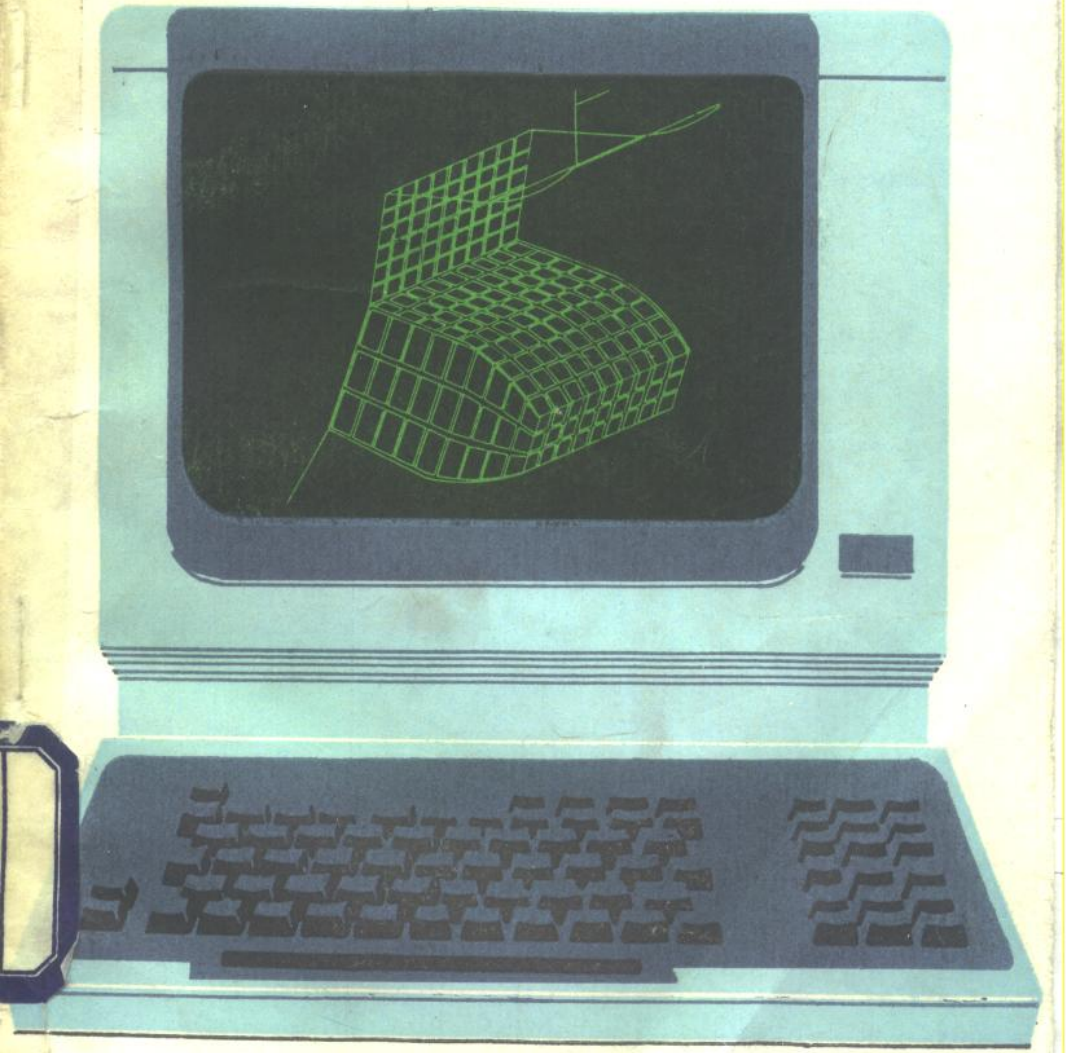
微型计算机应用丛书

彩色计算机的应用

〔美〕约翰P．格莱劳
J．D．逻伯逊 著



机械工业出版社

微型计算机应用丛书

彩色计算机的应用

〔美〕 约翰 P. 格莱劳
J. D. 逻 伯 逊 著

杨万峰 译
姚大能 朱世毅 校

机 械 工 业 出 版 社

本书介绍彩色计算机的应用，提高编程经验和技巧。全部程序均用BASIC语言编写，内容由浅入深、由简至繁。每一个程序结构严密、层次分明，易读性好，不仅能在TRS-80微型计算机上实现，只要稍加修改，就可用于IBM-PC微型计算机中。读者通过各种游戏程序可了解在微型计算机的屏幕上如何产生神奇莫测的幻想色彩，这不仅是艺术享受，更重要的是在娱乐之后使您受益非浅。

本书供计算机应用的工程技术人员阅读，对计算机应用感兴趣的其他专业人员也能参考。

Color Computer Applications

John P.Grillo, J.D.Robertson

John Wiley & Sons, Inc, 1983

• • •

微型计算机应用丛书

彩色计算机的应用

〔美〕约翰 P．格莱劳 J．D．逻伯逊 著

杨万峰 译

姚大能 朱世毅 校

•

机械工业出版社出版（北京阜成门外百万庄南里一号）

（北京市书刊出版业营业许可证出字第117号）

中国农业机械出版社印刷厂印刷

新华书店北京发行所发行·新华书店经售

•

开本 850×1168 1/32·印张 3 3/4·字数 87 千字

1987年3月北京第一版·1987年3月北京第一次印刷

印数 00,001— 5,500·定价 1.00 元

•

统一书号：15033·6502

出版者的话

计算机是现代化建设中不可缺少的先进工具，其应用正在向各个领域渗透，并以日新月异的面貌迅猛发展。

为了迅速普及计算机先进科学知识，大力推广计算机应用技术，积极提高技术管理干部的现代化管理水平和工程技术人员应用计算机的水平，中国机械工程学会自动化分会和机械工业出版社根据目前急需情况，先组织出版一套《微型计算机应用丛书》，分编著和翻译两个独立部分，均以应用为重点，内容反映微型计算机在机械、电气自动化、仪器仪表、办公自动化等方面的应用，供使用计算机的工程技术人员参考。

本丛书的出版得到机械工业部计算机领导小组的大力支持，机械工业部计算机与集成电路办公室，北京机械工业自动化研究所等单位的有关同志给予了具体指导和帮助，其他兄弟单位提供了方便，对此一并表示感谢。

机械工业出版社

编委会名单

译 者 序

随着计算机的迅猛发展，其应用技术已深入到人们生活的各个领域。本书主要介绍彩色计算机的应用，但内容与众不同，重点放在怎样用计算机做游戏，使您读后，心情愉快，使您用本书介绍的程序操作后，感到是一种艺术享受。

本书详尽地介绍了若干有趣的游戏程序，并附有图形显示，内容由浅入深，由简至繁，品种多，富有幻想色彩。

本书所有的程序均以BASIC语言写成，只要具备有关计算机的基础知识，就能饶有兴趣地同计算机做游戏，并从中汲取编程经验和技巧，创造更神奇莫测的游戏程序。

在翻译过程中，根据读者的需要，对原文的个别字句及章节名称作了删简或补充。此外，吴可奇、章运椿两位同志对译文给予了指导和帮助，王明正同志也提了不少宝贵意见和建议，在此一并表示感谢。

由于译者水平有限，难免有错误和不妥之处，恳请读者批评指正。

译者

前　言

通常，计算机仅是以打印记录作为输出，这可能是由于最初设计的计算机仅是供那些买得起计算机的大企业用的。事实上，在商业、科学研究和教学等方面也经常需要进行大量的打印记录。

微型计算机出现后图表成了一种很普通的输出形式。人们对图表发生兴趣是由彩色计算机的魅力引起的。确实，图表是一种可以传授知识、引人入胜和易于记忆的计算机输出。人们常常会把一份企业报告扔在一边，或对一叠叠文字材料感到乏味，而对图却感兴趣。俗话说“一图顶千句”，这也适用于计算机的输出。

我们编写这本书的目的是为了能给你（编程者）一种促进作用。在书中提出了一些设想，以及这些想法在实践中的运用。我们提供一整套程序作为表达这些想法最基本的媒介。书中编写的每一个程序力求结构严谨、模式典型，以便使程序的变更较易进行，而且在陈述每个程序时，字句力求严密，以期阐明每一程序的一般应用，并解释其内部工作过程。这些程序可作为一种“设计思想集锦”，为你所享用，你可从中汲取很多技巧营养。如果你能应用书中的哪怕是一个程序，并能将它编入一个成功的系统，那就算是你对本书的最高嘉奖了。

因此，我们要鼓励你使用彩色图形。本书选用了TRS-80彩色计算机，因为它是一种价格便宜适合家庭应用的个人计算机。同时，它的速度和许多小型商用微型计算机的相同，或者更快些，并且它有一套功能很强的BASIC语言，该语言具有令人满意的绘图指令。

本书中编写的这些程序，适合于在具有32k内存的TRS-80彩色计算机，扩展的BASIC语言和一个磁盘下使用。这种计算机与TDP系统100型也完全相同。所有程序无需变动，你就可以在英国Tandy型机器，即Dragon32上使用。此外，你也会很容易地修

改这些程序，使之能在IBM-PC/XT上运行，因为BASIC语言是极其类似的。这或许是Microsoft公司对个人计算机事业所做出的最重要的贡献之一，即它为众多的机器提供了一个标准、灵活，且易于输入的BASIC语言。

当我们编写关于如何应用计算机的时候，我们立即决定把它编写成一本有关程序方面的书，这样，就能使读者领略到计算机引以自豪的特性。因为对低分辨率图形显示用的BASIC语言和高分辨率的高级语言在性能和语言结构方面有很大差别，我们只能把它们分别写在两本书里。本书只集中叙述低分辨率图形显示的绘图方法。

我们设想读者已熟悉BASIC语言的程序设计。本书中安排的各项程序打算用来演示和扩展绘图的思想，而不是介绍计算机语言的句法和规则。你应该依靠彩色计算机手册或其它有关资料来帮助解决句法问题。本书的目的是为了提高你的兴趣，扩展你对计算机作用的认识，这样你就可以在多种应用领域内去探索绘图编程的课题。

章节的编排

除了第一章介绍几个小的概念外，本书中每章都介绍一组图形，并以一个单独程序作为每一章的中心内容。以这个为主的程序作为一个例子，用以突出该章中的几个特点。各章都是从一段对本章任务的陈述开始，简明扼要地阐明该章所要讨论的问题。你将从其中获得什么知识？学到些什么技巧？在这个程序中你又将了解到些什么新的BASIC指令？

每章的第二部分是叙述程序试图提出的具体问题，例如，在第五章的问题陈述中说明图形发生器程序的作用，以及如何利用程序在TRS-80彩色计算机上来解决此问题。

各章的第三部分是程序的“软件设计”、程序的输入、输出和人机对话实例。这一些有助于你理解应该怎样编排程序。当你编制程序时，就会面临着这样一个任务：需要把问题的陈述转化成为

一套指令，以便供计算机使用。要把问题的陈述转化成为程序，对你来说，并不是一件轻而易举的事情，除非你一步步列出较详细的步骤，并精心地理出问题的要点。

在软件设计中，我们列出了计算机应该产生的输出，并使其尽量与终端屏幕上的图象相一致。然后，逐条列举所有用户的响应或者数据输入。如果问题是通过连续的对话(即用户和计算机之间的一问一答方式）最令人满意的话，我们就按照那个对话的屏幕图象事先计划好的形式，对输入和输出综合考虑，并定出方案。

一旦你对程序的信息(它的输入/输出，或者I/O)了解得比较清楚了。你就可以着手处理产生这些交互影响的全过程了。这是我们程序设计中下一步的安排，并且安排在每一章的下半部分。这一步骤称之为“伪代码”，所谓伪代码就是编程人员为程序逻辑提出的简化方案。我们选用了英文短语作为我们的伪代码，这仅仅是因为这样做对你的学习不会再增加新的负担。

我们编写的伪代码程序概要地指明了程序的逻辑规律，即解决的方法，或者说明运算方法。应用这种运算形式，任何人都可以在任意一台计算机上用某种语言编写程序。当然，我们相信计算机及其语言一定能够做到在问题陈述中所提出的要求。因此，伪代码算法应该反映作为程序输入和输出一部分的硬件的优点和缺点。例如，对于TRS-80彩色计算机，因只用其低分辨率绘图特性，在程序设计时，就要对在运算中要用到的特性具体加以说明。

每章的第五部分是程序本身，我们有意保持程序具有相对叙述的自由，因为这些程序在章节的其余部分都已详细论证。我们深知这样做才能倾听到各种不同的评论，而这样的评论对任一种编排来说是绝对需要的。我们认为在内容安排方面是合理的，因为这些程序是与本书紧密联系在一起的。

尽管这些程序的文字内容看上去数量较少，但结构是极其严谨的，也就是说，程序从上至下的设计原则都是以程序段的形式来编制的。每一个处理过程都有其相对独立的子程序，并且为使

整个程序具有最大适应性，采用了信息表形式。采用程序段的编程方法，可使你在费力较少的情况下，就可以修改现存的某些程序段，或者以子程序的方式加进一些新的程序。

编 程 格 式

从彩色计算机手册说明书的程序中，你已经看到许多打印好的程序。这些程序清单几乎不值一读，因为其中的一些要点和空格已被删去。每一条程序里塞满了许多介绍文字，而这些又完全是可以压缩的。出版这些程序的目的仅仅是：它们占用最少量的印刷页面，并且在计算机上可以用最小容量的存贮器加以贮存。但要想接通你的计算机并把它精确的复制进去，那几乎是不可能的，因为这些程序太简化了。在你复制该程序时，只要稍为有一点错，那你要化很大功夫才能找出错误的根源，因为这些程序通常没有一个严密的结构。

前面提到过，我们仅就本书中程序的编排作扼要的说明。我们用一个设计好的编程方式来改善程序的易读性。在需要解决易读性的地方，我们插入空格，并编排行号，把主程序与子程序分隔开来。根据总体结构的安排，我们把每一个程序划分成若干个主要的程序段，可以改变每一个程序段的内容，而其它的程序段不受影响。这种功能上的明确分离，是程序模块，(即结构码)的突出特点之一。

大多数程序的第一部分，我们称之为驱动模块，即主程序，有时又名为主线程序。其功能是从程序能够完成的几种可能的活动方式中选择出一种方式来。主程序为你提供了许多活动方式，你可以从这个信息单中选择出所需要的一种方式，然后，主程序就转移到你选择的活动方式中去。

每种活动方式是一个单独调用的子程序。如果我们能把一种活动方式合并到一个子程序中去，这就更好，但要以子程序长度不超过24行为限制。使用这种办法，我们就可以在不影响其它任何部分的情况下，理解、扩充或者修改每一种活动方式。如果一种

活动方式非常复杂，则必须进而将其分解成另一些子程序，则该项活动方式应该变为一个次主程序，也许可以有它自己的信息单。

程序的分段，又称为它的模块化，在该程序需要修改或扩展时，是非常有价值的。每章都有一个稍为简单的程序要求对它扩展，因此它的模块化是很重要的。我们的意图是把这些程序提供给你，让你开始操作并且极其方便地将它们扩展成为一套你较为满意的程序。

系统的缺点

当采用TRS-80彩色计算机时，计算机的BASIC语言在某些表达形式上会产生紊乱，这些缺陷妨碍我们利用编程结构上的某些特点，但我们认为，这些特点对任何一个要编写程序的读者来说又是有所助益的。这里有三个缺点是特别麻烦的：第一个是BASIC语言会自动删除一行中前面的一些空格，这样，就无法将几行分成若干阶梯而使某一程序段突出出来。第二个是BASIC语言没有换行字符，因此，当一个语句长度超过32个字符（即屏幕宽度）时，就不能在适当的地方断开，而迫使在屏幕下一行上的一个短语处开始。例如，考虑下面这样一个语句：

```
3110 IF X<A OR X>B THEN PRINT "All DONE"
     ELSE GOSUB 5000
```

语句的简洁、易懂格式应该是

```
3110 IF X<A OR X>B
          THEN PRINT "All DONE"
          ELSE GOSUB 5000
```

可是，由于前面提到的两个缺点，在屏幕上显示出来的却象是：

```
3110 IF X<A OR X>B THEN PRIN
T "All DONE ELSE GOSUB 5000
```

TRS-80彩色计算机的第三个缺点是它的大写字母格式。虽

然，Shift-0 键加字母键可以倒换大写及小写字母，它在打印机上能按小写字母打出，而在屏幕上仍以大写字母显示。

编 程 准 则

我们已经确定了几条简单的编程准则，这有助于你阅读这些程序和理解它们的结构。

1．除了第一章中最短的程序外，所有程序均以符号为4的行号开始。

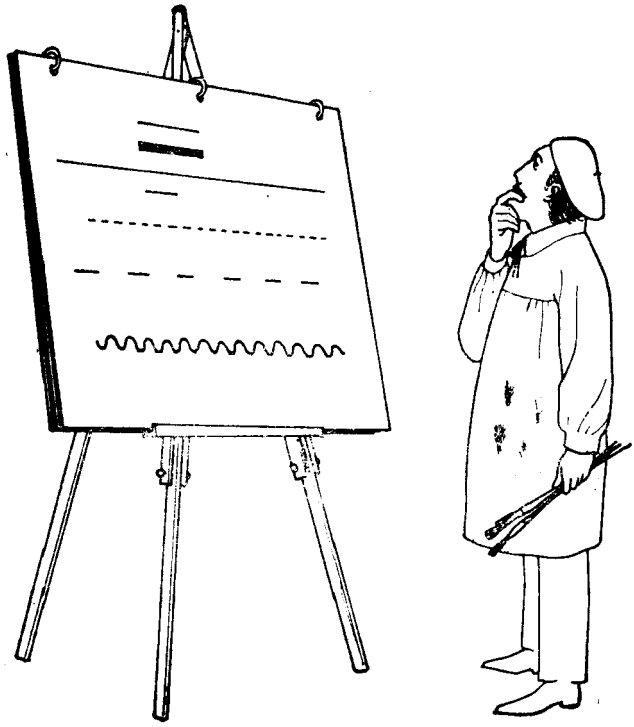
2．主程序的编号总是在1000以下。

3．对任意长度的程序，其最后一行是END语句，且位于行号为9999处。

4．那些被主程序信息单调用的首位子程序，都在编号为1000的倍数的行处起动。

5．大多数程序中每行只放一个语句。尽管经常可以在一行中放几个语句，但我们认为这样做常常会造成逻辑上的混乱。

现在请你来实践了。启动你的计算机，送入选择的程序并把它连接起来，这时，你将会欣赏到经过自己亲手变革后的奇观景象。



目　录

第一章 彩色直线

概 述

本章所介绍的内容与书中其它章节的内容完全不同，它不是叙述一个单独的、较大的程序，而是包含了若干个小的程序，其中每一个小程序都是在前面一个程序的基础上演变而来的。其目的是研究在各种情况下，如何在屏幕上产生一条彩色的直线。在本章中，我们想说明作为任何图形设计中最基本元素的，也即直线的灵活性。我们还要说明怎样使用彩色、线段长度和线段的方向，以便能在屏幕上修改图象。

指 令 组

如同本书前言中提出的那样，我们假定你已经学过了BASIC语言，并且假定已认真阅读了TRS-80彩色计算机手册，并熟悉了该机的指令系统。本章首要的重点，是提供一些程序，以便你在空闲时加以研究，如果需要的话，你可以按照自己的方法对程序进行扩充或改编。为此，作为一种复习的方法，让我们来研究一下程序VERTLIN。

VERTLIN程序清单：

```
10 ' PROGRAM : CHAPTER 1 : VERTLIN
20 ' AUTHORS : JOHN P GRILLO
30 '           JD ROBERTSON
40 '
50 CLS 0    ' CLEAR BLACK SCREEN
60 FOR Y=0 TO 31
70 ' COLUMN 10  ROW Y, MAGENTA
80 SET(10,Y,7)
90 NEXT Y
100 GOTO 100    ' HOLD DISPLAY
110 END
```

该程序可以在屏幕上产生一条从0行到31行垂直穿过整个屏幕的直线（屏幕被分成0到63共64列，该线位于第11列上）。

让我们把这段程序推广如下：

RNDVLIN程序清单：

```
10 ' PROGRAM : CHAPTER 1 : RNDVLIN
20 ' AUTHORS : JOHN P GRILLO
30 '           JD ROBERTSON
40 '
50 C=RND(8)
60 CLS 0
70 FOR I=1 TO 3
80 X=-1+RND(64)
90 FOR Y=0 TO 31
100 SET(X,Y,C)
110 NEXT Y
120 NEXT I
130 IF INKEY$="" THEN 130 ELSE 50
140 END
```

第50行在 1 ～ 8 种颜色中任选一种颜色。

第60行把屏幕清除为黑色。

第70～120行是一个外循环，共循环三次，产生三条垂直线。

第80行可为这条直线在 0 ～63列中选择任一列。

第90～110行是用来产生C颜色的垂直线。

第130行的功能是保证操作者在按任一个键之前，使屏幕上保持不变。

下一个程序RNDHLIN是程序RNDVLIN的变形，只不过是该程序在任意选定的三行上，产生三条横穿整个屏幕的直线。

RNDHLIN程序清单：

```
10 ' PROGRAM : CHAPTER 1 : RNDHLIN
20 ' AUTHORS : JOHN P GRILLO
30 '           JD ROBERTSON
40 '
50 C=RND(8)
60 CLS 0
70 FOR I=1 TO 3
80 Y=-1+RND(32)
90 FOR X=0 TO 63
100 SET(X,Y,C)
110 NEXT X
120 NEXT I
130 IF INKEY$="" THEN 130 ELSE 50
140 END
```

现在我们把程序作一点小的变化，就会使之产生显著的差别。除了每条线上下“漂动”之外，程序RNDLIN产生三条横穿

屏幕的线。水平线的垂直分量，是由于在100语句中对垂直位置J加－1，0或＋1形成的。因为直线可能漂离0～31行间的垂直范围，所以我们需要第110行和第120行。

RNDLIN程序清单：

```
10 ' PROGRAM : CHAPTER 1 : RNDLIN
20 ' AUTHORS : JOHN P GRILLO
30 '           JD ROBERTSON
40 '
50 C=-1+RND(9)
60 CLS 0
70 FOR K=1 TO 3
80 J=-1+RND(32)
90 FOR I=0 TO 63
100 J=J-2+RND(3)
110 IF J<0 THEN J=1
120 IF J>31 THEN J=31
130 SET(I,J,C)
140 NEXT I
150 NEXT K
160 IF INKEY$="" THEN 160 ELSE 50
170 END
```

现在我们使三条漂动的水平线分别为红色、白色和蓝色。

USALIN程序清单：

```
10 ' PROGRAM : CHAPTER 1 : USALIN
20 ' AUTHORS : JOHN P GRILLO
30 '           JD ROBERTSON
40 '
50 DIM C(3)
60 C(1)=4 : C(2)=5 : C(3)=3
70 CLS 0
80 FOR K=1 TO 3
90 J=-1+RND(32)
100 FOR I=0 TO 63
110 J=J-2+RND(3)
120 IF J<0 THEN J=1
130 IF J>31 THEN J=31
140 SET(I,J,C(K))
150 NEXT I
160 NEXT K
170 IF INKEY$="" THEN 170 ELSE 60
180 END
```

在第60行中，我们确定了红（4）、白（5）和蓝（3）三种颜色，并把它们贮存在数组C中。在前一语句中，我们已经为数组C空出了三个单元。这一语句似乎没有必要，因为计算机对数组的下标在10以内的数可以不加说明。可是，我们觉得象现在这种情况，当下标是一个变量时，实际上，它是提醒自己记住DIM语句