

北京希望电脑公司计算机技术丛书

C语言图形程序设计技巧与实例

林君 编



海洋出版社

北京希望电脑公司计算机技术丛书

C语言图形程序设计技巧与实例

林 君 编

海 洋 出 版 社

1992 年·北京

内 容 提 要

本书详细地介绍了在 IBM-PC 及其兼容机上进行图形程序设计的基本原理和方法。全书分两部分,第一部分介绍图形程序设计的必要知识,主要有:PC 机介绍、软件开发和 C、图形技术、文本和图形、图表、动画、二维和三维绘图;第二部分为参考指南,共含两章,提供了视频服务 BIOS 子程序和 Microsoft C 图形库的大量信息和实例。

本书通俗易懂、配有丰富的例子,适用于图形程序设计的初学者。本书有助于提高读者自身的编程技巧和图形程序设计水平。是图形程序设计员必备的参考书,也可供大专院校有关专业的师生参考。

(京)新登字 087 号

刘利蕾 责任编辑

C 语言图形程序设计技巧与实例

林君 编

希望 审校

海洋出版社出版(北京市复兴门外大街 1 号)

海洋出版社发行 双青印刷厂印刷

开本: 787×1092 毫米 1/16 印张: 35.8125 字数: 788 千字

1991 年 5 月第 1 版 1991 年 5 月第 1 次印刷

印数: 1—3000

ISBN7-5027-2531-8/TP·77

定价: 17.00 元

编 者 前 言

计算机图形技术随着图形显示终端和软件的发展而迅速发展,应用十分广泛。图形程序设计是程序设计中最直观、最有趣又能引人入胜的部分,但是图形程序的开发又要受到速度、功能以及显示设备等环境因素的约束。近些年来,C语言已经十分流行,这种介于高级语言和汇编语言之间的语言,吸收了二者各自的优点,不仅适用于系统软件设计中。对于图形程序设计而言,除了能保证执行速度等条件外,它还提供了丰富的图形库函数,为实现复杂的图形程序提供了方便。

本书较全面地讨论了用C语言进行图形程序设计的内容与方法,不仅介绍了在IBM-PC机上进行图形程序设计的必要基础,而且还详细地介绍了视频ROM BIOS服务程序和Microsoft C图形函数库。全书共九章,分图形程序设计基础和参考指南两大部分。第一部分包括:IBM-PC机介绍、软件开发和C、图形技术、文本和图形、图表、动画技术、二维和三维绘图。第二部分包括:视频BIOS服务程序、Microsoft C图形函数库。

本书内容丰富、通俗易懂,全部例子都配有程序,而且对图形函数库中的每一个函数都进行了详细的描述,内容包括:句法、函数、所包括的文件、兼容性、描述、返回值、相关函数、类似的Turbo C++函数、建议的代码结构和使用、实例。

本书适用于图形程序设计的初学者,亦可供大专院校有关专业的师生参考。要提高图形程序的编程序技巧与水平,这是一本使用C语言进行图形程序设计的程序员必备参考书。

在本书的编译和出版过程中始终得到了希望电脑公司的大力支持。在这里仅向对本书出版工作作出过贡献者表示感谢。

由于编译者水平所限,书中难免存在缺点错误,殷切希望读者批评指正。

编 者
1992年初

目录

前言	1
第一章 IEM PC 及其兼容机介绍	2
1.1 微处理器	2
1.2 8086 和 8088	2
1.3 80286	3
1.4 80386	3
1.5 80486	3
1.6 操作系统	3
1.7 MS-DOS	3
1.8 OS/2	3
1.9 视频子系统	4
1.10 CGA	4
1.11 EGA	5
1.12 MCGA	6
1.13 VGA	6
1.14 显示器	6
1.15 直接驱动单色显示器	7
1.16 复合单色显示器	7
1.17 复合彩色显示器	7
1.18 RGB 彩色显示器	7
1.19 变频显示器	7
1.20 输入设备	7
1.21 键盘	7
1.22 鼠标	8
1.23 跟踪球 (操纵杆)	8
1.24 数字化仪	8
1.25 光笔	13
第二章 软件开发和 C 语言	14
2.1 C 语言的优点	14
2.2 适应能力	14
2.3 速度	14
2.4 方便性	14
2.5 代码结构	14
2.6 预处理命令	15
2.7 说明	16
2.8 定义	17
2.9 表达式	17
2.10 语句	19

2.11	函数	20
2.12	软件开发过程	22
第三章	图形技术	26
3.1	坐标系统	26
3.2	物理坐标系	26
3.3	可视坐标系	27
3.4	窗口坐标系	28
3.5	坐标变换	29
3.6	图形光标	30
3.7	长宽比	31
3.8	点和象素	32
3.9	线	33
3.10	线模式	37
3.11	集 2	41
3.12	集 3	44
3.13	矩形和正方形	47
3.14	多边形	49
3.15	圆和椭圆	50
3.16	弧	52
3.17	复杂曲线	53
3.18	使用 _arc 函数的曲线	53
3.19	直线近似方法	55
3.20	Bezier 曲线	55
3.21	区域充填	57
3.22	充填模式	59
3.23	半色调和阴影	65
3.24	饼状图和楔形	71
3.25	颜色和调色盘选择	81
3.26	可视窗口和剪辑区	84
第四章	文本和图形	89
4.1	文本类型	89
4.2	文本光标	89
4.3	图形方式下的文本	91
4.4	图形方式下字形文本的使用	92
4.5	文本颜色	94
4.6	生成一个字符集	95
4.7	非充填的字符	98
4.8	充填的字符	100
4.9	阴影字符	103
4.10	多颜色字符	105
4.11	阴影字符	108

4.12	字符例子	109
4.13	文本窗口	136
第五章	表示图形	138
5.1	初始化表示图形环境	138
5.2	饼状图	141
5.3	条形图	144
5.4	柱状图	150
5.5	散列图	156
5.6	线状图	164
5.7	制定表示图形环境	172
5.8	颜色池	173
5.9	式样池	173
5.10	模式池	174
5.11	字符池	174
5.12	Chartenv 结构	175
5.13	titletype (标题类型) 结构	176
5.14	axistype (坐标轴类型) 结构	177
5.15	Windowtype (窗口类型) 结构	178
5.16	legendtype (图例类型) 结构	179
第六章	动画	180
6.1	动画艺术	180
6.2	整个屏幕动画	180
6.3	部分屏幕动画	192
6.4	用部分屏幕动画的隐藏页	195
6.5	动画中的高级概念	200
6.6	在多彩色背景上放置图象	200
6.7	用于复杂图象的联合作用变元	201
6.8	优化动画的速度	205
第七章	二维和三维绘图	206
7.1	二维坐标系统	206
7.2	二维旋转公式	206
7.3	旋转矩形	210
7.4	旋转多边形	213
7.5	旋转椭圆	215
7.6	三维坐标系	218
7.7	三维绘图的隐藏表面消隐	228
7.8	小结	229
第八章	视频 BIOS 子程序	230
8.1	Video Service 00H	230
8.2	Video Service 01H	231
8.3	Video Service 02H	231

8.4	Video Service 03H	232
8.5	Video Service 04H	232
8.6	Video service 05H	232
8.7	Video Service 06H	233
8.8	Video Service 07H	233
8.9	Video Service 08H	234
8.10	Video Service 09H	234
8.11	Video Service 0AH	234
8.12	Video Service 0BH	235
8.13	Video Service 0CH	235
8.14	Video Service 0DH	235
8.15	Video Service 0EH	236
8.16	video service 0FH	236
8.17	Video Service 10H, 子程序 00H	237
8.17.1	Video Service 10H, 子程序 01H	237
8.17.2	Video Service 10H, 子程序 02H	237
8.17.3	Video Service 10H, 子程序 03H	238
8.17.4	Video Service 10H, 子程序 07H	238
8.17.5	Video Service 10H, 子程序 08H	238
8.17.6	Video Service 10H, 子程序 09H	239
8.17.7	Video Service 10H, 子程序 10H	239
8.17.8	Video Service 10H, 子程序 12H	239
8.17.9	Video Service 10H, 子程序 13H	240
8.17.10	Video Service 10H, 子程序 15H	240
8.17.11	Video Service 10H, 子程序 17H	240
8.17.12	Video Service 10H, 子程序 1AH	241
8.17.13	Video Service 10H, 子程序 1BH	241
8.18	Video Service 11H, 子程序 00H	241
8.18.1	Video Service 11H, 子程序 01H	242
8.18.2	Video Service 11H, 子程序 02H	242
8.18.3	Video Service 11H, 子程序 03H	242
8.18.4	Video Service 11H, 子程序 04H	243
8.18.5	Video Service 11H, 子程序 10H	243
8.18.6	Video Service 11H, 子程序 11H	243
8.18.7	Video Service 11H, 子程序 12H	244
8.18.8	Video Service 11H, 子程序 14H	244
8.18.9	Video Service 11H, 子程序 20H	244
8.18.10	Video Service 11H, 子程序 21H	245
8.18.11	Video Service 11H, 子程序 22H	245
8.18.12	Video Service 11H, 子程序 23H	245
8.18.13	Video Service 11H, 子程序 24H	246

8.18.14	Video Service 11H, 子程序 30H	246
8.19	Video Service 12H, 子程序 10H	247
8.19.1	Video Service 12H, 子程序 20H	247
8.19.2	Video Service 12H, 子程序 30H	248
8.19.3	Video Service 12H, 子程序 31H	248
8.19.4	Video Service 12H, 子程序 32H	248
8.19.5	Video Service 12H, 子程序 33H	249
8.19.6	Video Service 12H, 子程序 34H	249
8.19.7	Video Service 12H, 子程序 35H	249
8.19.8	Video Service 12H, 子程序 36H	250
8.20	Video Service 13H, 子程序 00H	250
8.20.1	Video Service 13H, 子程序 01H	251
8.20.2	Video Service 13H, 子程序 02H	251
8.20.3	Video Service BH, 子程序 03H	252
8.21	Video Service 1AH	252
8.22	Video Service 1BH	252
8.23	Video Service 1CH	253
第九章	Microsoft C 图形函数	254
9.1	_arc	254
9.2	_arc_w	256
9.3	_arc_wxy	260
9.4	_clearscreen	263
9.5	_displaycursor	267
9.6	_ellipse	269
9.7	_ellipse_w	272
9.8	_ellipse_wxy	275
9.9	_floodfill	279
9.10	_floodfill_w	281
9.11	_getactirepage	284
9.12	_getarcinfo	285
9.13	_getbkcolor	288
9.14	_getcolor	291
9.15	_getcurrentposition	294
9.16	_getcurrentposition_w	296
9.17	_getfillmask	301
9.18	_getfontinfo	304
9.19	_getgttextextent	307
9.20	_getgttextvector	309
9.21	_getimage	312
9.22	_getimage_w	315
9.23	_getimage_wxy	317

9.24	<code>_getlinestyle</code>	320
9.25	<code>_getphyscoord</code>	323
9.26	<code>_getpixel</code>	326
9.27	<code>_getpixel_w</code>	327
9.28	<code>_gettextcolor</code>	331
9.29	<code>_gettextcursor</code>	333
9.30	<code>_gettextposition</code>	335
9.31	<code>_gettextwindow</code>	338
9.32	<code>_getvideoconfig</code>	340
9.33	<code>_getviewcoord</code>	346
9.34	<code>_getviewcoord_w</code>	349
9.35	<code>_getviewcoord_wxy</code>	352
9.36	<code>_getvisualpage</code>	354
9.37	<code>_getwindowcoord</code>	356
9.38	<code>_getwritemode</code>	359
9.39	<code>_grstatus</code>	362
9.40	<code>_imagesize</code>	368
9.41	<code>_imagesize_w</code>	370
9.42	<code>_imagesize_wxy</code>	373
9.43	<code>_lineto</code>	375
9.44	<code>_lineto_w</code>	379
9.45	<code>_moveto</code>	382
9.46	<code>_moveto_w</code>	386
9.47	<code>_outgtext</code>	388
9.48	<code>_outmem</code>	391
9.49	<code>_outtext</code>	393
9.50	<code>_pg_analyzechart</code>	395
9.51	<code>_pg_analyzechartms</code>	398
9.52	<code>_pg_analyzepie</code>	401
9.53	<code>_pg_analyzescatter</code>	404
9.54	<code>_pg_analyzescatterms</code>	407
9.55	<code>_pg_chart</code>	410
9.56	<code>_pg_chartms</code>	412
9.57	<code>_pg_chartpie</code>	415
9.58	<code>_pg_chartscatter</code>	418
9.59	<code>_pg_chartscatterms</code>	420
9.60	<code>_pg_defaultchart</code>	423
9.61	<code>_pg_getchardef</code>	426
9.62	<code>_pg_getpalette</code>	429
9.63	<code>_pg_getstyleset</code>	432
9.64	<code>_pg_hlabelchart</code>	434

9.65	<code>_pg_initchart</code>	437
9.66	<code>_pg_resetpalette</code>	440
9.67	<code>_pg_resetstyleset</code>	442
9.68	<code>_pg_setchardef</code>	445
9.69	<code>_pg_setpalette</code>	448
9.70	<code>_pg_setstyleset</code>	450
9.71	<code>_pg_vlabelchart</code>	452
9.72	<code>_pie</code>	455
9.73	<code>_pie + w</code>	459
9.74	<code>pie_wxy</code>	462
9.75	<code>__Polygon</code>	465
9.76	<code>_polygon_w</code>	468
9.77	<code>_polygon_wxy</code>	471
9.78	<code>_putimage</code>	474
9.79	<code>_gutilnage_w</code>	476
9.80	<code>_rectangle</code>	479
9.81	<code>_rectangle_w</code>	482
9.82	<code>_rectangle_wxy</code>	485
9.83	<code>_registerfonts</code>	488
9.84	<code>_remapallpalette</code>	490
9.85	<code>_remappalette</code>	494
9.86	<code>_scrolltextwindow</code>	497
9.87	<code>_selectpalette</code>	499
9.88	<code>_setactivepage</code>	502
9.89	<code>_setbkcolor</code>	505
9.90	<code>_setcliprgn</code>	508
9.91	<code>_setcolor</code>	512
9.92	<code>_setfillmask</code>	514
9.93	<code>_setfont</code>	518
9.94	<code>_setgtextvector</code>	520
9.95	<code>_setlinestyle</code>	524
9.96	<code>_setpixel</code>	527
9.97	<code>_setpixel_w</code>	529
9.98	<code>_settextcolor</code>	530
9.99	<code>_settextcursor</code>	532
9.100	<code>_settextposition</code>	535
9.101	<code>_settextrows</code>	537
9.102	<code>_settextwindow</code>	539
9.103	<code>_setvideomode</code>	542
9.104	<code>_setvideomoderows</code>	544
9.105	<code>_setvieworg</code>	546

9.106	_setviewport	550
9.107	_setvisualpage	553
9.108	_setwindow	555
9.109	_setwritemode	556
9.110	_unregisterfonts	559
9.111	_wrapon	561
参考文献		563

前 言

图形程序设计是计算机程序设计中最有趣也是最引人入胜的领域。图形程序设计者必须在特定的环境中直观地产生和开发基于图形的应用程序，而这些程序必须满足速度、功能以及操作方便的约束条件。C 语言图形程序设计指南一书介绍在 IBM-PC 及其兼容机上进行图形程序设计的概念和基本原理，这里使用 Microsoft C 编译器 6.0 版和 Microsoft Quick C 的 2.5 版。

本书提供了全部的入门信息以及执行图形和 Microsoft C 的信息。第一部分：“用 Microsoft C 进行计算机图形程序设计”包括七章，这七章讨论从视频硬件到动画技术的大量信息。下面简要描述第一部分中每一章的内容：

- 第一章“IBM PC 及其兼容机介绍”讨论 IBM PC 及其兼容机的基本内容。所描述的硬件设备包括微处理器、操作系统、图形卡、监视器和输入设备。
- 第二章“软件开发和 C”描述 C 语言的优点和代码结构。此外，提供了应用软件开发过程。
- 第三章“图形技术”提供了 Microsoft 图形库的基本绘画能力。提供了坐标系统、绘画、区域填充和可视点的概念。
- 第四章“文本和图形”介绍了联合文本和图形的方法。提供三种方法：使用标准字符集、使用 Microsoft 编译器提供的字形，产生一个用户定义的字符集。
- 第五章“表示图形”给出表示图形库中特征的描述。本章提供使用在表示图形库中包括的全部范围的许多实例。
- 第六章“动画”提供动画艺术。描述了动画的基本知识和隐藏页的使用，并用大量的例子演示这些技术。
- 第七章“二维和三维绘图”提供了开发二维和三维图形所使用的概念，并提供了使用讨论内容的例子。

第二部分“参考指南”含两章，它提供了视频 BIOS 服务程序的使用和 Microsoft C 图形库的扩充文件。

- 第八章“视频 BIOS 服务程序”描述了视频 ROM BIOS 服务程序。对每一个视频服务程序和服务子程序都提供了：输入、输出和描述。
- 第九章“Microsoft C 图形函数”是图形和表示图形库中函数的一个详细的参考指南。对于每一个函数都提供了下述广范内容的信息：句法、函数、所含文件、兼容性、描述、返回值、相关函数、类似于 Turbo C++ 函数、建议的代码结构、使用以及例子。

本书是为了提供需要开发基于图形程序的全部信息而编写的。通过使用所描述的技术和信息，您可以开发各种类型和复杂的高级图形程序。

第一章 IBM PC 及其兼容机介绍

要实现有效的图形程序设计，重要的是理解计算机硬件的基本知识。主机系统（将执行所开发图形软件的系统）的物理限制将导致图形软件的需求和限制。视频适配器和监视器决定可应用的分辨率的范围和颜色。其它因素，如微处理器、存贮器和运算速度，在设置图形产品的需求和说明中起到一个重要的作用。

1.1 微处理器

微处理器是计算机系统的核心，有时称为中央处理单元(CPU)，微处理器通过执行一系列的运算、数据转换和数字比较来执行程序。计算机的所有基本操作都由 CPU 控制。各种类型的数据通过总线（信号通道）送到 CPU。总线连接 CPU 到各种输入/输出(I/O)口。这些 I/O 口用于连接存贮器和支持芯片到总线，最终到 CPU。

所有的 IBM PC、PS/2 机型（及其兼容机）都使用 Intel 8086 系列的一个处理器。见表 1.1。下面逐个描述 8086 系列中的现行的微处理器。

表 1.1 IBM PC、XT、AT、PS/1 和 PS/2 使用的微处理器

机型	微处理器
IBM PC	8088
PC XT	8088
AT	80286
PS/1	80286
PS/2 25	8086
PS/2 30	8086
PS/2 50	80286
PS/2 60	80286
PS/2 80	80386

1.2 8086 和 8088

在 PS/2 型号 25 和 30 中使用的 8086 微处理器是一个具有 16 位数据总线的 16 位微处理器。十四个寄存器用来传输数据、处理数据、存贮数据、存贮器地址、状态和控制标志以及指令指针。8086 能够寻址 1M 字节的存贮器。

用在 IBM PC、PCjr 和 PCXT 中的 8088 微处理器是一个具有 8 位数据总线的 16 位微处理

器。8088 和 8086 的主要区别在于 8088 具有 8 位数据总线，而 8086 具有 16 位数据总线。在所有的其它方面，用于编程目的，8088 与 8086 完全一致。

1.3 80286

在 IBM AT 和 PS/2 型号 50 和 60 中使用的 80286 与 8086 和 8088 兼容。添加到 80286 上的最重要的特征是它的多任务能力，即在同个时间执行几个任务的能力。80286 通过在两个任务之间切换处理来实现这一目的。多任务的一个例子是在打印一个文件时编辑另一个文件。

80286 微处理器以两种方式操作。第一种方式是实地址方式，它迫使 80286 就象 8086 一样地工作。第二种方式是保护方式，它保留了程序执行用的存贮器的一个事先决定量；这个存贮器防止被其它任何程序使用。多任务能力就是工作在保护方式下。通过保护方式，几个程序可以同时运行而且互不影响。

1.4 80386

在 PS/2 型号 80 和许多 IBM PC 兼容机中所使用的 80386 微处理器是一个强有力的 32 位微处理器，它带有 32 位寄存器。尽管 80386 微处理器支持 8086 功能和 80286 保护存贮器方式，但它提供了比这些 CPU 中任一个都更方便的存贮器管理。

1.5 80486

80486 微处理器是 80×86 CPU 系列中最新公布的。几个制造商已发布了使用 80486 的 IBM 兼容机。尽管这些系统现在是市场上最昂贵的 IBM 兼容机，但它们提供了最新的技术。80486 比 8086 系列中其它微处理器速度快、集成度高。它联合了一个增强的 80386 微处理器并带一个与 80387 速度等同的数字协处理器，一个复杂的高速缓冲存贮器控制器和一个 8k 字节支持高速缓存的存贮器。

1.6 操作系统

每个计算机都有一个操作系统。操作系统是一个机器语言程序的集合，它管理系统所需的各种功能，包括外围操作。IBM 系列的个人计算机具有两个操作系统，MS-DOS 和 OS/2。

1.7 MS-DOS

MS-DOS 起始于 QDOS，一个为 S-100 总线设计的操作系统，它使用 8086 微处理器。QDOS，变成 86-DOS，由 Seattle 计算机产品公司研制，在 1980 年被 Microsoft 买下。Microsoft 是被 IBM 选中的软件开发组之一，为它的新计算机研制操作系统。结果，其产品 MS-DOS 变成 IBM PC 及其兼容机的标准。MS-DOS 已经修改了几次，当前的版本是 5.0 并继续是指定的个人计算机操作系统。

1.8 OS/2

当 IBM 介绍它的 PS/2 型号时，一个新的操作系统 OS/2 同时被引入。OS/2 功能上与 MS-DOS 版本 3.2 和 3.3 类似，它是为运行和利用较新的微处理器而设计的。

要支持较新的微处理器, OS/2 以两种方式操作: 实方式和保护方式。实方式响应几乎与 MS-DOS 相同, 因此为 MS-DOS 设计的几乎所有的应用都被 OS/2 所支持。然而, 在实方式下不支持多任务。

OS/2 也操作在保护方式下。尽管这种方式象 MS-DOS 一样操作, 它功能更强、更方便。OS/2 的主要优点是它在保护方式下支持多任务。此外, 在保护方式下 OS/2 不受 640K RAM 应用的限制(实方式有这个限制)。

尽管 MS-DOS 现在是最常用的操作系统, OS/2 的特点和能力使它对将来的发展更有吸引力。可是, 对于图形程序设计而言, 由 Microsoft 提供的大部分图形特征都不被 OS/2 所支持。

1.9 视频子系统

视频硬件子系统产生计算机系统的屏幕图象。所有的 IBM PC 系列个人计算机及其兼容机(PS/2 除外)都需要一个适配卡。对于兼容机而言, 这个适配卡常常包含在系统构造中。这个适配卡是一个专门的视频线路板, 它插在一个系统扩充槽中。

适配板或视频电路含有一块专用存储器, 它保存显示信息。视频子系统把显示信息转换成驱动视频显示的信号。为 PC 设计的视频显示子系统是单色显示适配器(MDA), 彩色图形适配器(CGA), 增强图形适配器(EGA)。多彩色图形阵列(MCGA)和视频图形阵列(VGA)用于 PS/2 系列。VGA 适配器也用于 PC 系列。

1.10 CGA

彩色图形适配器(CGA)是 IBM PC 兼容的图形适配器的最低档次, 它仍广泛地用于家庭计算机, 但是 EGA 和 VGA 是商用计算机的标准。然而, 大多数应用软件与 CGA 兼容。CGA 以及所有的视频子系统都能以两种方式的任一种模式下操作。第一种模式支持文本, 第二种支持图形。

图形方式有几种屏幕分辨率选择; 由 CGA 显示器支持的选择是高分辨率 _HRESBW 方式和中分辨率 _MRES4COLOR、_MRESNO COLOR 两种方式。在 _HRESBW 方式下, 仅有两种颜色可用: 前景和背景颜色。背景色可以用 _Setbkcolor 函数来改变。注意到 _HRESBW 方式比两种中分辨率方式有更高的分辨率(640×200 象素)。

_MRES4COLOR 和 _MRESNOCOLOR 两种方式的分辨率为 320×200 象素点。_MRES4COLOR 方式用于彩色显示器并可显示 4 种颜色。第一种颜色是背景颜色, 可从预定义的 16 种颜色调色盘中任选一种。其它三种颜色由一个表格定义。所希望的调色盘由 _selectpalette 函数选择, 表 1.2 列出在 _MRES4COLOR 方式中有效的调色盘。

表 1.2 _MRES4COLOR 方式中的调色盘

调色盘数	颜色 0	颜色 1	颜色 2	颜色 3
0	可选	绿	红	棕

续表

调色盘数	颜色 0	颜色 1	颜色 2	颜色 3
1	可选	青	品红	浅灰
2	可选	浅绿	浅红	黄
3	可选	浅青	浅红	白

_MRESNOCOLOR 方式为单色监视器而设计的，但它能够用于彩色监视器。当使用单色监视器时，可显示灰级；当使用彩色监视器时，表 1.3 所列的调色盘有效。

表 1.3 _MRESNOCOLOR 方式的 CGA 调色盘

调色盘数	颜色 0	颜色 1	颜色 2	颜色 3
0	可选	蓝	红	浅灰
1	可选	浅蓝	浅红	白

1.11 EGA

增强图形适配器(EGA)支持全部的 CGA 图形和文本方式。然而，在 _MRESNOCOLOR 方式中，彩色调色盘不同于 CGA，见表 1.4。

表 1.4 _MRESNOCOLOR 方式的 EGA 调色盘

调色盘数	颜色 0	颜色 1	颜色 2	颜色 3
0	可选	绿	红	浅灰
1	可选	浅绿	浅红	黄
2	可选	浅深蓝	浅红	黄

除支持 CGA 图形方式外，EGA 还增加了 4 种图形方式：_MRES16COLOR，_HRES16COLOR，_ERESNOCOLOR 和 _ERESCOLOR。_ERESNOCOLOR 方式只有两种颜色（前景和背景），但是其分辨率为 640×350 。_ERESCOLOR 方式可选用表 1.5 所示的 16 种颜色，且具有 640×350 的屏幕分辨率。_MRES16COLOR 方式也可选用表 1.5 中的 16 种颜色，但屏幕分辨率限制在 320×200 。同样，_HRES16COLOR 方式可以用表 1.5 中的各种颜色，但其屏幕分辨率被限定在 640×200 。