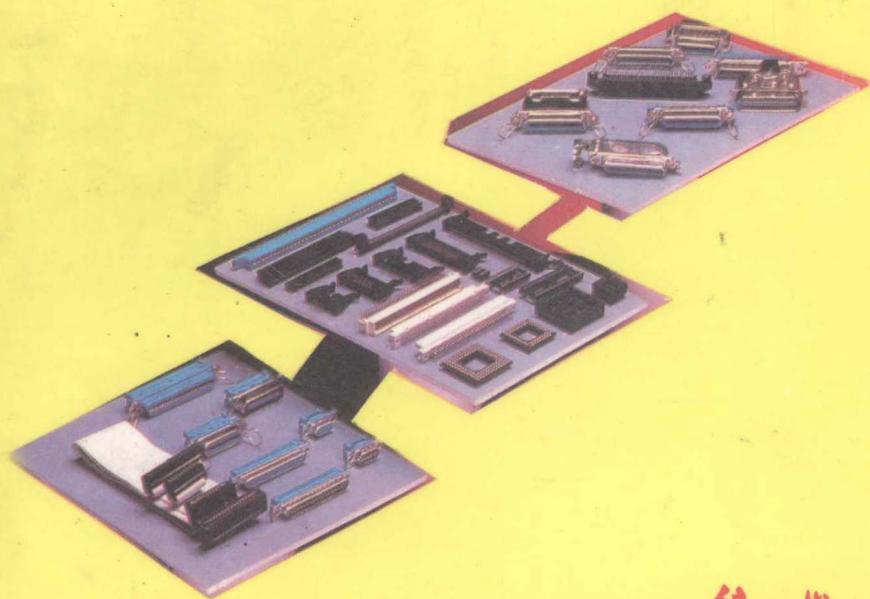


孙宗琦 著
阎德勤

TVGA
卡

应用开发编程技术与实例



科学出版社

TVGA 卡应用开发编程技术与实例

孙宗琦 阎德勤 编著

熊可宜 陈河南 审校

科学出版社

1993.

(京)新登字 092 号

内 容 提 要

本书共分为十五章,主要对 Trident 公司的 TVGA 卡的应用开发编程方法进行了全面而详细的说明。由于强调应用为主,前五章对读者只要求有初步的高级语言编程经验,完全没有涉及硬件,尽管介绍了三维作图的编程,作者也有意回避了必需的解析几何知识。由于作图涉及运算较多,计算机速度越快效率越高,在第六到八章中,作者介绍的采用几种高级语言编程的科学工程三维 TVGA 超高分辨绘图程序包,这是作者在计算机模拟科研和微机应用开发中,吸收国外大型机绘图软件的优点,自编的一种小绘图程序包。它的编程“一杆子插到底”,删除了不必要的多重子程序嵌套和无用的子程序。因此,它们具有程序小,执行速度快和调用方便诸优点。源程序中加中文注释,保证了用户随意扩充和压缩的可能。

本书是对 TVGA 卡的应用编程方法进行说明,适用于从事计算机开发的技术人员使用。

欲购本书的用户,请直接与北京 8721 信箱联系,电话:2562329,邮码:100080。

TVGA 卡应用开发编程技术与实例

孙宗琦 阎德勤 著

熊可宜 陈河南 校

责任编辑 刘晓融

科学出版社 出版

北京东黄城根北街 16 号

邮政编码:100717

双青印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

1993 年 12 月第 一 版 开本:787×1092 1/16

1993 年 12 月第一次印刷 印张:28.625

印数:1~5000 字数:665 千字

ISBN7-03-004030-9/TP·332

定价:15.00 元

前 言

大约在六年前,从香港寄来的“电子商情”中看到一则消息,有人批评我们大陆的计算机多数只是一种高级摆设,心里不是滋味。显然对方的批评只是指一般用户,而不是指搞人造卫星的专家,尽管中国人多,在计算机上也有硬件、软件和开发应用的高手,但是,对方的批评显然是中肯的。以后在奥斯陆与挪威计算机专家谈及此事,出乎意料他们友好地告诉我,十年前,他们国家也是如此。但是谈话时他们国家已处处离不开计算机了。参观计算机生产工厂,从生产到检测基本上只见机器不见人了。

目前,我国计算机的应用已逐渐普及,随着集成电路性能价格比的几何级数方式增长,PC机在我国也将由单位集体所有的计算机逐渐成为名符其实的个人计算机。计算机外设和数据采集接口板的发展相对慢了一些,价格偏高,但是廉价的TVGA卡和配套的 $1024 \times 768 \times 256$ 色显示器的上市,表明了用它来研制需要图象输出的仪器例如CT、B超、简易工作站,高质量汉字排版系统等已是指日可待的事了。

作为一个业余爱好者,在计算机应用上已有多年持续不断的兴趣,也有一定的广度和深度,但是以往只满足于个人兴趣,受好和求知欲,但对社会贡献甚少,也没感到自己的责任。随着改革开放,我国的微机应用开发必将起飞,我把个人最近在微机应用上有潜在社会经济效益的一些经验和探索成果以小册子形式奉献给读者。

在这本小册子中专门叙述Trindent公司的TVGA卡的应用开发编程方法。作者参考了目前已出版的关于在各种接口卡及显示器上编程作图的专书,主要是通过编程调试,终于找到了一些实用方法。有些方法简便易学,完全采用高级语言,有的方法追求速度,完全采用汇编语言直接与图像寄存器打交道。

本书内容组织如下:第一章介绍了TVGA卡自带软件的应用方法,接着的三章分别叙述了在三种常用高级语言:Quick Basic,Turb. Basic和Fortran 5.0方式下编程作简单图形的编程方法。第五章叙述了用高级语言作各种复杂图的简易编程方法。第六章介绍了在PC机上配有T800 Transputer加速卡时,开发Fortran编程的简易三维绘图程序包和运用于Microrosft Fortran 5.0三维作图程序包。第七章简单介绍了TVGA硬件有关编程的知识。第八章专门介绍了TVGA屏幕拷贝技术和图像存储软件,其中断程序由汇编语言完成,并加入了在彩色打印机上拷屏的编程技巧。第九章叙述汇编语言编程作图的方法,通过剖析印制板制作软件TANGO最新版本的显示驱动程序,最后达到使TANGO升级支持TVGA的全部高分辨模式。第十章介绍了在作者开发的多功能数据采集控制卡上用混合语言编程的应用例子。最后一章是应用展望。

由于强调应用为主,前五章对读者只要求有初步的高级语言编程经验,完全没有涉及硬件,尽管介绍了三维作图的编程,作者也有意回避了必需的解析几何知识。由于作图涉及运算较多,计算机速度越快效率越高,在第六章作者介绍了仍用高级语言编程但采用INMOS公司T800加速卡的TVGA编程方案及作者在计算机模拟科研中自编的一个小绘图程序包,这是一种适合国情的廉价方案。汇编语言程序虽然费功,但是效率最高,在目前仍无软件危机的中国仍是一般开发应用者乐意使用的廉价方案。汇编语言编程需要与TVGA卡寄存器直接打交道,在第七章只重点介绍与编程有关的TVGA寄存器。最后两

章的编程主要用的是汇编语言。

为了专业工作者的方便，作者在每章后附有较专门的参考书。

本书所举例程都经调试运行，并不断更新，源程序及英文或汉字注释包括在一张源程序软盘中，为了防止录入中造成错误，本书的源程清单都是直接由已调试完毕的源程序文本文件在汉字字处理软件 WPS 下加汉字注释而成。最后拼入汉字排板系统文稿，这样做，很费工夫，但方便了读者，需要源程序软盘及对程序进一步开发有兴趣的读者可与希望电脑公司秦人华编辑或作者联系。

感谢秦人华编辑对作者鼓励和支持，没有好的鼓励，本书从动稿到出版几乎是不可能的。

最后，衷心欢迎读者对本书提出批评和进一步合作开发建议。

编著者

1993·北京

目 录

第一章	TVGA 随卡软件及应用实例	1
第二章	Quick basic 编程作图	10
第三章	Turbo Basic	25
第四章	Fortran 5.0 的 TVGA 编程	64
第五章	三维图形库的编制方法	88
第六章	一个适用于 T800 加速卡开发	102
第七章	TVGA 硬件原理及寄存器	156
第八章	TVGA 拷屏技术和图像存取技术	165
第九章	用汇编语言开发实用绘图软件——TANGO 软件的升级	192
第十章	数据采集实时高分辨显示	242
第十一章	应用展望	268

第一章 TVGA 随卡软件及应用实例

Trident 公司的 TVGA 卡目前有两种产品，一种是 TVGA900，卡上显示存储器容量 256K，不支持 1024×768×256 色方式；另一种 TVGA8900，卡上显示存储器容量 1024K，支持上述分辨率 256 色模式。两种接口卡的编程是相同的，随卡附带软件包可以通过 README.BAT 批作业文件的运行而学到 TVGA8900 和 TVGA900 在各种显示存储容量下所支持的各种显示模式。

当显存容量为 256K 时，它除了支持和它兼容的 CGA、EGA、HERC 及 VGA 模式以外，还支持 80 列×30 行、80 列×43 行和 80 列×60 行、132 列×25 行、132 列×30 行和 132 列×43 行、132 列×60 行扩展文本方式，附加的图形方式 800×600×16，1024×768×4 色两种当显存容量增加到 512K 时，又增加如下几种图形方式：

1024×768 16 色、768×1024 16 色、640×400 256 色、640×480 256 色、800×600 256 色，当显存容量为 1 兆字节时，又支持，1024×768 256 色的图形模式。

在这套驱动应用程序中有用于替代或补充各种应用程序的驱动程序，可以按菜单找到相应驱动程序的安装步骤，一旦安装成功，新的高分辨率图形方式就可应用，目前已配有这种驱动程序有 AutoCAD, P-CAD, Lotus1-2-3 wordstar, wordperfect, MS word, MSWindow 等十余种，仍在发展中，由于这些软件的专用性质，本书不作介绍。

在应用程序中，最主要的是 MODETEST.EXE，为了应用方便，建议在 C 盘上建立 TVGA 子目录，把 TVGA 驱动程序和应用程序 1 号盘中子目录 TVGAUTIL 中所有可执行文件拷贝到 C 盘 TVGA 子目录中，然后用行编辑或任何一种文本编辑器（例如 MS 和 WS 等）把自动执行文件（在根目录上的）AutoEXEC.BAT 中补上一条路径 PATH=C:\TVGA，退出存盘后，重新启动计算机，键入 MODETEST，则显示 TVGA 卡支持的全部文本和图形方式，表一给出结果。

表一、TVGA 的各种显示模式

Mod	Text / Graphics	Colors	Row	Col.	Resolution
0	Text	16	25	40	320×200
1	Text	16	25	40	320×200
2	Text	16	25	80	640×200
3	Text	16	25	80	640×200
7	Text	2	25	80	720×350
0*	Text	16	25	40	320×350
1*	Text	16	25	40	320×350
2*	Text	16	25	80	640×350
3*	Text	16	25	80	640×350
0+	Text	16	25	40	360×400

1+	Text	16	25	40	360 × 400
2+	Text	16	25	80	720 × 400
3+	Text	16	25	80	720 × 400
4	Graphics	4	25	40	320 × 200
5	Graphics	4	25	40	320 × 200
					(more)
6	Graphics	2	25	80	640 × 200
7+	Text	2	25	80	720 × 400
D	Graphics	16	25	80	320 × 200
E	Graphics	16	25	40	640 × 350
F	Graphics	2	25	80	640 × 350
10	Graphics	16	25	80	640 × 350
11	Graphics	256	30	80	640 × 480
12	Graphics	2	30	80	640 × 480
13	Graphics	256	25	40	320 × 200
50	Text	16	30	80	640 × 480
51	Text	16	43	80	640 × 473
52	Text	16	60	80	640 × 480
53	Text	16	25	132	1056 × 350
54	Text	16	30	132	1056 × 480
55	Text	16	43	132	1056 × 473
					(more)
56	Text	16	60	132	1056 × 480
57	Text	16	25	132	1188 × 350
58	Text	16	30	132	1188 × 480
59	Text	16	43	132	1188 × 473
5a	Text	16	60	132	1188 × 480
5b	Graphics	16	75	100	800 × 600
5c	Graphics	256	25	80	640 × 400
5d	Graphics	256	30	80	640 × 480
5e	Graphics	256	75	100	800 × 600
5f	Graphics	16	48	128	1024 × 768
60	Graphics	4	48	128	1024 × 768
61	Graphics	16	64	96	768 × 1024

只要键入模式号，可以进入该模式，并显示该模式上所有字符集和颜色，如在菜单下键入 62，回车后进入 1024 × 768 256 色，键入“1”显示一种字体的字符集，再键入“2”显示 256 个色彩块，再键入“3”就在方式 62 下工作。如果在变小了的 DOS 提示符下，用 EDLIN 或 DEBUG 工作，你会发现除了速度变慢，其它似乎一切正常。有时还会发现，

由于分辨率提高，只用了萤光屏的一部分，应用程序 TANSI.SYS，用于增加最大显示行数。首先把 TANSI.SYS 拷贝到 C 盘根目录上，然后在已有的 Config.sys 文件中用文本编辑器增加一行。

DEVICE=TANSI.SYS，存盘后，再热起动计算机，按 MODETEST, 62,3,DEBUG,D 0:0 1000 诸键，这时，显示已是满屏幕。

为了增加显示速度，应用软件中 TVGABIO.EXE 可以把 TVGA 卡视频 BIOS 由卡上 ROM 芯片区转储到 RAM 区，由于 RAM 存取速度比 ROM 快，从而加快了显示速度，安装 TVGABIO 为驻留程序可直接键入 TVGABIO 回车，也可在 AUTOEXEC.BAT 文件中插进一行：TVGABIO，一旦 TVGABIO 常驻内存，重复以上步骤，就能使屏幕翻卷速度加快很多。

如果所有软件都象 Edlin 或 DEBUG 那样，问题就简单了，可是当你在进入最高分辨率方式 62 后，一旦调入 Quick BASIC 或其它各种应用软件，分辨率又降低到没有插 TVGA 只有 CGA,EGA 或 VGA 时一样，也就是说，目前的多种常用高级语言如 Quick

basic v.4.40, Fortran 5.0 还不支持 TVGA 所专有的高分辨率图形方式，以下章节将着重介绍如何在常用高级语言方式下绘制高分辨图。

对于 TVGA 应用程序中的 SVM89，它用于设置 TVGA 为 VGA/EGA/CGA/HERC 方式，TVGACRTC 用于模式设定后改变图形中心和屏幕长和宽，后一任务也可以不动 TVGA CRTC 参数，而调节显示器后电位器来达到目的，因此不再介绍其余内容。

现在我们分别进入各种 16 色高分辨的显示方式下工作，用 DEBUG 来了解计算机内存与显示的关系。

键入“MODETEST”待显示选择菜单后，键入“12 和 3”，计算机进入方式 12:640×480 16 色，在 DEBUG 中，键入 RDS，显示原有 DS 值后，把 DS 值改为 A0000，时键入“E0000”利用空格键，不断把 DS:0 以后地址送入 FF，你会发现显示屏左上角出现水平亮线，用 E 95F8 再不断送 FF，在右下角出现亮线，直到 9600 地址，显示屏不再变化，也就是说，主内存 A000:0 到 A000:95ff 与显示屏上点一一对应 $640 \times 480 / 8 = 38400 = 9600h$ 主存上每点对应于显示屏上水平入点，再做一实验来检验主内存中各数据位与水平象点位置关系，在 E3000 后键入 FF，空格键 1，我们看到除了水平线以外，右边第八点处又多了一亮点，如果用 80 代替以上的 1，则只观察到连续 9 点形成的水平亮线，因此，内存或显存中最高位对应于最左或最先扫描的象点。

在 5b 显示方式 800×600 16 色方式下做同样实验，在 A000:EA5E 处到显示屏右下角。因为 $800 \times 600 / 8 = 6000 = EA06h$ 。

对于 5F 和 61 两种方式的情形，最大地址 FFFF 中数改成 FF，只在屏中下方显示亮线，因为 $1024 \times 768 / 8 = 18000H$ ，已超过 64K，即 DS 定义段地址范围。

我们仍在 5F 方式，在 DEBUG 工作状态下，对 TVGA 的重要寄存器输入输出来观察一些现象，取得一些感性知识，关于 TVGA 的寄存器的有关编程的硬件知识，我们在第七章将要专门介绍，好在 TVGA 在寄存器一级是与 VGA 兼容，我们目前只介绍几个最重要的寄存器。图像控制寄存器，它使用 PC 机口地址 3CE 和 3CF，它的零号寄存器是设置重置寄存器，一号寄存器是允许设置重置寄存器，三号寄存器是循环功能选择寄存

器，四号寄存是读映射寄存器，八号寄存器是位屏蔽寄存器，定序控制器有口地址 3C4 和 3C5，它的二号地址寄存器是映射屏蔽寄存器，第 E 号寄存器是新模式控制寄存器。

我们可以在 DEBUG 中用口地址输入输出命令 I 及 O 来观察和改变这些寄存器的内容，例如键入 O 3CE 0,I 3CF(回车)便可查看设置重置寄存器内容，当我们试图用这种方式修改寄存器内容时，有时 DEBUG 修改不了，有时死机，因此我们将选用高级语言来作图或修改寄存器内容。

我们先用 GWBASIC 来观察上述主要寄存器的作用。用文本编辑器例如 EDLIN,MS, 等把下列程序用 Color.BAS 存文件名存盘，然后调用 MODE TEST，进入方式 5b:800×600 16 色显示模式，退回到包括 GWBASIC 和上述 COLOR.BAS 的子目录中，用 GWBASIC COLOR 命令运行，我们看到屏幕上先出现白色区，然后清除，以后随映射屏蔽寄存器(3C4:2)的内容而变颜色。位屏蔽寄存器(3CE:8)用来屏蔽掉写入八点中的某些点，利用允许设置重置寄存器(3CE:1)和设置重置寄存器(3CE:0)也能画彩色条带，但是它们的效果与映射屏蔽寄存器是不完全相同的。例如

```
10 rem save "color.bas"
60 def seg = &H3000
70 for i = 0 to 4095
80 poke I,255
90 next i
92 for i = 0 to 4095
95 poke i,0
98 next i
11 0 for k = 1 to 15
115 out &H3c4,2:out &H3c5,K
120 for i = 0 to 4095
125 poke i+4096 * k,255
130 next
140 next
155 for k = 1 to 15
160 out &H3c4,2:out &H3c5,K
170 for i = 0 to 4095
175 poke I+4096 * K,0
180 next
190 next
210 out &H3c4,2:out &H3c5,15
212 out &H3cE,1:out &H3cf,15
215 for k = 1 to 15
218 out &H3ce,0:out &H3cf,k
220 for i = 0 to 4095
```

```

225  poke I+4096 * K,255
230  next
240  next
255  for k = 1 to 15
260  out &H3cE,1:out &H3cf,0
270  for i = 0 to 4095
275  poke I+4096 * K,0
280  next
290  next
310  out &h3c4,2:out &H3c5,15
312  out &H3ce,1:out &H3cf,15
315  for k = 1 to 15
318  out &H3ce,0:out &H3cf,k
320  for i = 0 to 4095
325  poke I+4096 * K,255
330  next
340  next
355  for k = 1 to 15
360  out &H3ce,1:out &H3cf,0
370  for i = 0 to 4095
375  poke i+4096 * K,0
380  next
390  next
405  out &H3ce,8:out &H3cf,128
410  for i = 0 to 4095
420  poke i,255
430  next
440  for i = 0 to 4095
450  poke I,0
460  next i
470  OUT &H3ce,8:out &H3cf,255
510  def seg
520  end

```

例程 1.1 GWBASIC 语言编写，初步了解 TVGA 寄存器功能。

用映射屏蔽寄存器写入彩色时，如果原来有白色背景，写入的颜色在白色背景区变成待显色彩色的补色，当写零试图清除时，也有相似现象，用允许设置重置等寄存器写彩色时，写和抹去都比较准确，有关更多图形控制寄存器的功能的感性知识，我们将在以后章节中

逐渐引入，有关原理将在第七章中叙述。

作为应用实例，下面介绍在 GWBASIC 下作图的方法。程序 1.2 给出在 GWBASIC 下调用中断 10h 的各种子功能，进入 TVGA 各种显示方式及作图、显示字符的例子，其方法简单介绍如下：

GWBASIC 本身不能调用 TVGA VBIOS，必须借助于汇编语言，分别给寄存器 AX,BX,CX,DX 赋 TVGA 中断所规定的参数，利用 TVGA 中断 10H，即能实现作图及拷贝打印的五种基本功能：

- 1.设置显示模式(mode)
- 2.设置光标位置
- 3.在设定光标位置写一个字符
- 4.画点到(x,y)位置
- 5.读(x,y)点颜色代码

实现以上五种基本功能，诸寄存器所要赋的参数形式如下:(H,L 分别代表寄存器的高八位与低八位，如 AH=0,表示 AX 寄存器高八位赋 0 值):

一、设置显示模式(mode)

AH=0,AL=显示模式号。

二、设置光标位置

AH=2,DH=光标所在行，DL=光标所在列

三、在设定光标位置写一个字符

AH=9,AL=字符的 ASCII 码，CH=0,CL=1,BL=字符颜色代码

四、画点到(x,y)位置

AH=12,DH=INT(Y/256),DL=Y-256DH,AL=颜色代码，CH=INT(6X/256),
CL=X-256 CH

五、读(x,y)点颜色代码

AH=13，其它参数设置同四，颜色代码返回在 AL 中。

由如下结构的汇编语言，完成以上的参数设置

```
PUSH BP
MOV AX,num1
MOV BX,num2
MOV CX,num3
MOV DX,num4
INT 10H
MOV num,AX
POP BP
RETF
num DW
```

例程 1.2 中断 10H 调用汇编语言子程序

这里 num1 到 num4 是二字节的立即数，为实现上述五种功能所要求的参数。实现某种功能，立即数就取其所规定的参数值，执行这段汇编程序。在 GWBASIC 下执行汇编语言，并要随时改变参数，需要接口技术。

GWBASIC 与汇编语言接口有多种方法（参看有关 GWBASIC 语言的说明书），这里所用的是最简单的一种方法，把汇编语言子程序代码存放到数组 ISUB%()中，用 VARPTR 命令找到并执行这段汇编语言。在这种方法的基础上，POKE 及 PEEK 语句修改程序代码本身传递参数。

```
11     DIM ISUB%(12),MODE%(6),RX%(6),RY%(6),XC%(6),YC%(6),CMAX%(6)
12     DIM STEPC%(6),MODE$(6)
13     FOR I=1 TO 12:READ ISUB%(I):NEXT I
16     FOR I=1 TO 6
17         READ MODE%(I),RX%(I),RY%(I),XC%(I),YC%(I),CMAX%(I)
18     NEXT I:PRINT MODE%(6)
20     MODE$(1)="Tvga Mode 5B 800 * 600 16 colors"
22     MODE$(2)="Tvga Mode 5E 800 * 600 256 colors"
24     MODE$(3)="Tvga Mode 5F 1024 * 768 16 colors"
26     MODE$(4)="Tvga Mode 60 1024 * 768 4 colors"
28     MODE$(5)="Tvga Mode 61 768 * 1024 16 colors"
30     MODE$(6)="Tvga Mode 62 1024 * 769 256 colors"
40     AH%=0:AL%=0:BL%=0:BH%=0
50     CH%=0:CL%=0:DH%=0:DL%=0
52     FOR I=1 TO 6
53         STEPC%(I)=1
54         IF CMAX%(I)>15 THEN STEPC%(I)=10
55     NEXT I
60     FOR I=4 TO 6
70         AH%=0
80         AL%=MODE%(I)
90         GOSUB 10000
100        FOR K=1 TO LEN(MODE$(I))
110        AH%=2:DH%=0:DL%=10+K:GOSUB 10000
120        AH%=9:AL%=ASC(MID$(MODE$(I),K,1)):CH%=0:CL%=1:BL%=15-I
130        GOSUB 10000
140    NEXT K
200    FOR A%=1 TO CMAX%(I) STEP STEPC%(I)
210        FOR KL%=0 TO 399
215            AH%=12
216            IF I=4 GOTO 700
```

```

220  C% = XC%(I)+RX%(I) * A% * SIN(KL% * 3.1415926# / 200)
230  D% = YC%(I)+RY%(I) * A% * COS(KL% * 3.1415926# / 200)
240  AL% = A%
250  DH% = INT(D% / 256):DL% = D%-256 * DH%
260  CH% = INT(C% / 256):CL% = C%-256 * CH%
270  BH% = 0:BL% = 0
280  GOSUB 10000
290  NEXT KL%
295  NEXT A%
590  RENNU = 0
600  FOR A% = 1 TO CMAX%(I) STEP STEP C%(I)
605  AH% = 13
610  FOR KL% = 0 TO 399
620  C% = XC%(I)+RX%(I) * A% * SIN(KL% * 3.1415926# / 200)
630  D% = YC%(I)+RY%(I) * A% * COS(KL% * 3.1415926# / 200)
640  DH% = INT(D% / 256):DL% = D%-256 * DH%
650  CH% = INT(C% / 256):CL% = C%-256 * CH%
660  BH% = 0:BL% = 0
670  GOSUB 10000
680  IF AL% < > A% THEN RENNU = RENNU+1
690  NEXT KL%
695  NEXT A%
700  IF RENNU < > 0 THEN BEEP :GOTO 780
720  AH% = 2:DH% = 1:DL% = 15:GOSUB 10000
730  AH% = 9:AL% = ASC("O"):CH% = 0:CL% = 1:BL% = 3:GOSUB 10000
740  AH% = 2:DH% = 1:DL% = 16:GOSUB 10000
750  AH% = 9:AL% = ASC("K"):CH% = 0:CL% = 1:BL% = 3:GOSUB 10000
760  AH% = 2:DH% = 1:DL% = 17:GOSUB 10000
770  AH% = 9:AL% = ASC("!"):CH% = 0:CL% = 1:BL% = 3:GOSUB 10000
780  NEXT I
900  AH% = 0
910  AL% = 3
920  GOSUB 10000
930  CLS
940  PRINT "End of TVGA gwbasic test program"
1000 END
1010 DATA &HB855,&H0,&HBB,&HB900,0,&HBA 'PUSH BP;MOV AX,0
1015 rem MOV BX,0;mov CX,0;MOV DX
1020 DATA &HCD00,&H2E10,&H15A3,&H5D01,&HFCB,0 '0;int 10H

```

```

1025    rem mov NUM,ax;POP BP; RETF; NUM
1030    DATA &H5b,20,16,400,300,15
1040    DATA &H5e,1,1,400,300,255
1050    DATA &H5f,25,20,512,384,15
1060    DATA &H60,150,120,512,384,3
1070    DATA &H61,20,5,384,512,15
1080    DATA &H62,2,1,512,384,255
9998    REM int 10H input parameters:AH%,AL%,BH%,BL%,CH%,CL%,DH% and DL%
9999    REM output parameter AL%
10000    S% = VARPTR(ISUB%(1))
10005    POKE VARPTR(ISUB%(2)),AL%:POKE VARPTR(ISUB%(2))+1,AH%
10010    POKE VARPTR(ISUB%(3))+1,BL%
10020    POKE S%+6,BH%
10030    POKE S%+8,CL%:POKE S%+9,CH%
10040    POKE S%+11,DL%
10050    POKE VARPTR(ISUB%(7)),DH%
10100    CALL S%
10110    AL% = PEEK(S%+22)
10120    RETURN

```

程序 1.3 是一个在 GWBASIC 语言下作图的演示程序，其步骤是：首先把上述调用中断汇编程序和返回参数读到数组中，然后分别设定显示方式为 TVGA 特有的几种高分辨方式，画一组彩色同心椭圆，写一组彩色字符，并读出画入点的颜色，核对无误后，返回常用显示文本方式 3。

程序中 1010-1025 DATA 语句用来存放汇编语言，10000 专为调用中断 10H 用。在调用前，按需要设定好八个参数 AH% ,AL%,BH%,BL%.....DH%,DL%。然后用 GOSUB 10000 完成所需中断调用，中断调用返回一组参数 AH%和 AL%。AL%可供携屏使用。

参考文献

- [1] EGA VGA TVGA 高级微机图形编程指南与实例，希望电脑公司，1991
- [2] IBMPCDOS 磁盘操作系统(V3.30 版)用户指南与技术参考手册，希望电脑公司
- [3] IBMPC BASIC 3.2 版技术参考手册，希望电脑公司

第二章 Quick Basic 编程作图

在 Quick Basic 语言中有三个与作图关系密切的语句：Screen 用于选择显示方式，PSET 用于在给定坐标处画一彩色点，Point 用于读出屏幕特定点的色彩，它们的格式是

Screen 方式[[彩色开关]], 串与页],[显示页]

PSET[STEP](x,y)[,颜色]

Point(x,y)

关于它们的详细说明，可以参考专书，我们这里用一例说明以上语句在绘图上的应用。程序 2.1 选择了目前 Quick Basic 4.0 支持的最高分辨率，方式 12 640×480 16 色、清屏后，用 15 种颜色画同心椭圆，这里，色彩码=0 代表黑色与背景相同，然后顺着同样的椭圆把画上点的颜色读出来并和原来画上去的颜色对照，如果出错就增加出错计数 ERRNUM，程序结束时，显示 ERRNUM=0，它表示以上画和读过程精确无误。

```
REM      save vgaprt.bas
REM      plot on VGA mode 12 640 * 480 16 colors
SCREEN 12
staf = .02 * 3.14159
FOR IC = 0 TO 15
  ampx = 20 * IC
  ampy = 15 * IC
  FOR IK = 0 TO 100
    IX = 320 + ampx * SIN(staf * IK)
    IX = INT(IX)
    IY = 240 + ampy * COS(staf * IK)
    IY = INT(IY)
    ICOLOR = IC                                'Pset (IX,IY),ICOLOR
    PSET (IX,IY), ICOLOR
  NEXT
NEXT
errnu = 0
FOR IC = 0 TO 15
  ampx = 20 * IC
  ampy = 15 * IC
  FOR IK = 0 TO 100
    IX = 320 + 20 * IC * SIN(.02 * 3.14159 * IK)
```

```

IX = INT(IX)
IY = 240 + 15 * IC * COS(.02 * 3.14159 * IK)
IY = INT(IY)
IF IC <> POINT(IX, IY) THEN errnu = errnu + 1
NEXT
NEXT
PRINT "errnu=", errnu
END

```

例程 2.1 VGA 绘图程序

细心的读者一定会问，为什么没有人介绍画线语句 line、存图语句 GET、放图语句 PUT 和写字符语句 Print，因为这些语句功能都可由 PSET 及 POINT 组合完成，这点在第五章介绍三维绘图包程序时将会详细讨论，POINT 函数对于把显示屏上图像拷贝下来很重要。程序 2.2 是一个拷屏程序，它把 VGA 方式 12 的彩色图当作黑白图在 LQ1600 打印机拷贝下来，当用 POINT(x,y)读出(x,y)点色彩不为零时，一律当作象点在打印机相应位置打印。

我们编写了三个子程序，分别代替 Screen，PSET 和 POINT 的功能，在 TVGA 的所有高分辨彩色方式下，它们都能工作。

这里用两个有代表性的例子来说明作图方法，程序 2.3 和程序 2.1 任务相似，但是它不是在 640×480 16 色，方式 12，而是在最高分辨率 1024×768 16 色，即方式 5F 下画出。选择显示模式和画读彩色点，我们除了用调 TVGA BIOS 中断的方法，还用了直接向 TVGA 寄存器存取数据的技巧，由于暂不要求读者熟悉 TVGA 芯片内部寄存器的功能，因此只要求读者会用程序 2.3 所示的方法绘制自己的绘图程序即可。

读者先把程序 2.3 用 PRTINT16.BAS 的名字存在 Quick BASIC 的子目录下，键入 QB / Run prtint16.BAS / L，就可立即运行 PRTINT16.BAS 程序，由于程序中多处调用中断，因此需要用 / L 告知 Quick Basic 与 Call Interrupt 有关的快速库函数联结，以上程序运行中除了显示彩色的同心椭圆以外，还显示一组字符，最后显示 ERRNUM=0 表明读出结果与画入结果完全相符。现在我们来逐条分析程序 2.3 中重要语句及子程序的作用。

'\$ INCLUDE 'QB.BI' 表示程序中由于要调 DOS 软中断，不但在进入 QB 时要联结包含相应子程序的快速库 QB.QLB 和常规库 QB.LIB 中（利用上述 / L 参数）而且还要用元命令。

'\$ INCLUDE 或 REM \$ INCLUDE 来处理附加的与调用中断配套的题文件 QB.BI。在这个程序中我们其实只利用了 QB.BI 中 TYPE RegType

```

Ax  As  INTEGER
Bx  As  INTEGER
Cx  As  INTEGER
Dx  As  INTEGER

```