

# Turbo C 實用圖形開發指南

路新峰 编写



陕西电子编辑部

# Turbo C 实用图形开发指南

路新峰 编著

陕西省电子编辑部

封面设计：钟山

陕西电子编辑部  
地址：西安市西五路16付5号  
邮编：710004

电话：794344 电挂：7795  
陕西省内部图书准印证  
陕新出批〔1992年〕字第347号

# 前 言

目前我国拥有的计算机大部分是微型机，微型计算机CAD技术的研究及其应用，近年来已进入了高潮。IBM-PC机及其各种牌号的国内外兼容机的图形功能，随着机型的增高而进一步加强。众多程序员不在为自己设计出的现有程序界面而满足，希望尽快地将图形带入自己的程序（系统），使自己编写的程序（系统）能为用户提供更美好的工作界面。所以，图形功能的研究、开发、推广应用已是我们计算机软件设计人员的一项重要任务。

本书的目的是帮助读者建立起用Turbo C语言开发图形软件的总体概念。Turbo C语言是一种快速高效的C语言编译程序，具有先进的集成环境；具有汇编语言那样直接控制系统硬件以及调用操作系统资源的功能；同时又具有一般高级语言完成复杂的数据处理和算术运算的能力。它集编辑、编译、链接运行、查错于一身，提供了丰富的函数库和强有力的图形功能，已成为开发图形软件最理想的程序语言，受到广泛的欢迎。

本书由浅入深全面的介绍了Turbo C语言图形设计方法，提供了许多图形实用程序，详细的介绍了Turbo C的全部图形函数，每个函数都有例子说明，非常适合愿意了解和从事图形应用软件编程者的需要；同时也可作为学习C语言的有效参考书目。

本书将开放的提供图形编辑软件GE的全部源程序，供读者参考。

全书共分六章：

第一章：讲述了C语言的数据类型、控制语句、显示器、图形适配器、图形模式、图形坐标、图形系统初始化等图形处理基础知识。

第二章：详细的讨论了Turbo C全部图形函数，每个函数都配有例子和说明。

第三章：从图形编辑功能开发的角度，讲述了图形软件功能开发的过程，如绘图、图形变换、图形方式下的文本输出、图形汉字显示、图形文件建立与调用等。

第四章：图形编辑软件GE功能说明及全部源程序。

第五章：本章详细的讨论了图形编辑软件GE的操作说明，运行环境和安装过程。对每一个功能键的使用过程，屏幕状态，处理办法都有详细的说明。

第六章：本章将七个图形调用程序介绍给读者，它们能将GE产生的图形调上屏幕，并能在其它语言和数据库环境中运行。

由于编者的水平和经验有限，书中难免出现这样那样的错误，望各位专家学者批评指正。

本书在编写过程中，陕西电子编辑部编委卢安国同志为本书审稿，做了大量的审校工作。党晓枫高级经济师、张忠智高级工程师、余采钧（教授级）高级工程师、杨宗岳研究员、张元良高级工程师、王治华高级工程师、张小宁工程师、佟平工程师等领导 and 同事对本书的出版给予了亲切的关怀和大力的支持，特此表示衷心的感谢。

编 者

# 目 录

第一章 图形处理基础	( 1 )
§ 1.1 Turbo C 语言的数据类型、基本符号和控制语句	( 1 )
1.1.1 数据类型定义	( 1 )
1.1.2 类型修饰符	( 1 )
1.1.3 其它数据类型	( 3 )
1.1.4 常量	( 3 )
1.1.5 控制字符常量	( 3 )
1.1.6 运算符	( 4 )
1.1.7 运算符优先次序	( 5 )
1.1.8 表达式	( 5 )
1.1.9 C语言控制语句	( 6 )
(1) if( ) 语句	( 6 )
(2) switch( ) 语句	( 6 )
(3) for( ) 语句	( 7 )
(4) while( ) 语句	( 7 )
(5) do-while( ) 循环语句	( 8 )
(6) break( ) 语句	( 9 )
(7) continue( ) 语句	( 9 )
(8) 标号和goto( ) 语句	( 10 )
§ 1.2 显示器, 图形适配器和图形模式	( 10 )
1.2.1 CGA 彩色显示器	( 10 )
1.2.2 EGA增强彩色显示器	( 10 )
1.2.3 VGA彩色显示器	( 11 )
1.2.4 常用显示器的基本性能	( 11 )
1.2.5 图形模式	( 11 )
1.2.6 显示模式控制	( 12 )
1.2.7 颜色与调色板	( 13 )
§ 1.3 坐标和绘图元素	( 15 )
1.3.1 坐标	( 15 )
1.3.2 笛卡尔坐标与屏幕坐标的转换	( 16 )
1.3.3 绘图元素	( 17 )
(1) 点	( 17 )
(2) 直线	( 17 )
(3) 曲线	( 17 )
(4) 像素	( 17 )
§ 1.4 初始化图形系统	( 17 )
1.4.1 已知显示器类型的图形系统初始化	( 18 )
1.4.2 不知显示器类型的图形系统初始化	( 18 )

1.4.3	自动初始化图形系统	(19)
§ 1.5	从图形模式进入文本模式再返回图形模式	(19)
§ 1.6	退出图形系统	(20)
第二章	图形函数	(21)
§ 2.1	图形状态控制函数	(21)
2.1.1	closegraph()退出图形系统	(21)
2.1.2	detectgraph()图形适配器硬件测试	(21)
2.1.3	graphdefaults()图形系统恢复为初始状态	(22)
2.1.4	getgraphmode()返回当前图形模式值	(22)
2.1.5	getmoderange()获取图形驱动器可使用模式范围	(24)
2.1.6	initgraph()装入图形驱动程序并初始化图形统	(25)
2.1.7	restorecrtmode()恢复屏幕为调用initgraph()函数前的模式	(27)
2.1.8	getpixel()获取(x,y)位置上的颜色值	(28)
2.1.9	Putpixel(),在(x,y)点处画一个规定颜色的点	(28)
2.1.10	getmaxx()返回当前图形模式下屏幕的最大X坐标值	(29)
2.1.11	getmaxy()返回当前图形模式下屏幕的最大Y坐标值	(29)
2.1.12	getx()返回当前图形位置的X坐标值	(30)
2.1.13	gety()返回当前图形位置的Y坐标值	(30)
2.1.14	getbkcolor()返回当前背景颜色	(31)
2.1.15	getcolor()返回当前画笔颜色	(32)
2.1.16	Setbkcolor()设置当前背景颜色	(32)
2.1.17	Setcolor()设置当前画笔色	(34)
2.1.18	getmaxcolor(),返回当前图形模式下最大有效颜色值	(34)
2.1.19	getpalette(),返回有关当前调色板信息	(35)
2.1.20	Setpalette()改变调色板的颜色	(35)
§ 2.2	画图函数	(36)
2.2.1	arc()画一圆弧	(36)
2.2.2	ellipse()画一椭圆	(37)
2.2.3	circle()画圆	(37)
2.2.4	getarcoords()取最后一次调用arc函数的坐标	(38)
2.2.5	line()在指定的两点间画一直线	(39)
2.2.6	lineto()从当前位置(cp)到(x,y)点画一直线	(39)
2.2.7	linere1()从当前(CP)位置到与(CP)有一相对距离的点画一直线	(40)
2.2.8	rectangle()画矩形	(40)
2.2.9	drawpoly()画多边形	(41)
2.2.10	getlinesettings()取当前线型宽度和模式	(42)
2.2.11	Setlinestyle()设置当前画线宽度和类型	(42)
2.2.12	Setwritemode()设置图形方式下画线输出模式	(44)
§ 2.3	填充	(45)
2.3.1	Setfillpattern()选择用户定义的填充模式	(45)

2.3.2	Setfillstyle()	设置填充模式和颜色	(46)
2.3.3	getfillpattern()	将用户定义的填充模式拷贝到内存	(47)
2.3.4	getfillsettings()	获取当前填充模式和填充颜色	(47)
2.3.5	bar()	画矩形条	(49)
2.3.6	bar3d()	画三维矩形条	(49)
2.3.7	fillpoly()	画一个多边形并填充	(50)
2.3.8	floodfill()	填充一个用颜色所封闭的区域	(50)
2.3.9	Pieslice()	画一个扇形并填充	(51)
2.3.10	Sector()	画一个椭圆扇区并填充	(52)
§ 2.4		屏幕窗口处理	(52)
2.4.1	Cleardevice()	清除图形屏幕	(52)
2.4.2	SetActivepage()	设置图形输出活动页	(53)
2.4.3	Setvisualpage()	设置可见图形页号	(54)
2.4.4	clearviewport()	清除当前图形窗口	(54)
2.4.5	Setviewport()	设置当前图形窗口	(55)
§ 2.5		图象处理	(55)
2.5.1	getimage()	将指定区域的图象存到内存区域	(55)
2.5.2	Putimage()	将存在内存的图象送回屏幕	(56)
2.5.3	imageSize()	返回存贮一块图象所需的字节数	(57)
§ 2.6		图形模式下文本处理	(58)
2.6.1	setTextjustify()	为图形函数设置文本对齐方式	(58)
2.6.2	SetTextstyle()	为图形字符输出设置当前字体	(59)
2.6.3	outtext()	在图形模式下显示一字符串	(60)
2.6.4	outtextxy()	在指定位置显示一字符串	(61)
2.6.5	Setusercharsize()	为矢量字体改变字符宽度和高度	(61)
2.6.6	textheight()	返回以象素为单位字符串高度	(62)
2.6.7	textwidth()	返回以象素为单位字符串宽度	(63)
2.6.8	gettextsettings()	返回当前图形文本字体的信息	(64)
第三章		图形编辑功能开发	(66)
§ 3.1		线的特性与规定	(66)
§ 3.2		绘图	(67)
3.2.1		画点	(67)
3.2.2		画直线	(72)
3.2.3		画矩形	(77)
3.2.4		画圆和圆弧	(83)
§ 3.3		填充	(86)
3.3.1		画图并填充	(86)
3.3.2		填充模式的设置	(88)
3.3.3		填充一个有界区域	(90)
§ 3.4		图形变换	(94)

3.4.1	图形拷贝	(95)
3.4.2	图形移动	(96)
3.4.3	图形比例变换	(100)
3.4.4	图形旋转	(105)
§ 3.5	图形方式下的文本输出	(109)
3.5.1	图形文本输出函数	(109)
3.5.2	设置文本方式	(110)
3.5.3	确定字符宽度和高度	(112)
3.5.4	两个图形文本输出函数	(113)
§ 3.6	图形汉字屏幕显示	(114)
§ 3.7	图形文件的建立和调用	(117)
第四章	图形编辑软件GE及源程序	(122)
§ 4.1	面向程序员的图形编辑软件GE	(122)
§ 4.2	GE基本模块	(122)
§ 4.3	GE源程序	(123)
第五章	图形编辑软件GE使用说明	(201)
§ 5.1	图形编辑软件GE的运行环境	(201)
5.1.1	硬件环境	(201)
5.1.2	软件环境	(201)
§ 5.2	图形编辑软件GE的安装运行	(201)
5.2.1	在C盘建立子目录并安装GE	(201)
5.2.2	图形编辑软件GE的运行	(201)
§ 5.3	功能键	(202)
§ 5.4	图形编辑软件GE基本操作说明表	(204)
§ 5.5	GE操作说明	(208)
5.5.1	图形文件读写	(208)
(1)	F1 建立屏图文件	(208)
(2)	F2 建立块图形文件	(209)
(3)	F5 读屏图形文件	(210)
(4)	F6 读块图形文件	(211)
5.5.2	绘图	(212)
(1)	F10 画点	(213)
(2)	F4 画圆	(213)
(3)	CTRL+I 画直线	(214)
(4)	CTRL+N 画三维矩形	(214)
(5)	CTRL+G 画圆弧	(215)
(6)	Shift+X 用光标控制键画直线	(216)
(7)	Shift+J 用光标控制键画矩形	(216)
(8)	Ins 喷色(大喷)	(217)
(9)	Del 喷色(小喷)	(217)



5.5.3	图形块处理	(217)
(1)	F9 图形块移动	(217)
(2)	ALT+C 图形块拷贝	(219)
(3)	ALT+O 清除图形块	(219)
(4)	CTRL+E 图形横向放大	(220)
(5)	CTRL+W 图形纵向放大	(220)
(6)	CTRL+D 图形横向缩小	(220)
(7)	CTRL+S 图形纵向缩小	(220)
(8)	CTRL+L 图形旋转	(220)
(9)	CTRL+T 填充 (在已画好的图形中填上颜色)	(221)
5.5.4	键盘输入	(222)
(1)	F3 区位码汉字输入	(222)
(2)	ALT+US x8 体英文字符键盘输入	(223)
5.5.5	状态控制	(223)
(1)	F7 清除屏幕	(224)
(2)	调主菜单	(224)
(3)	ALT+1 点标步进 1 ALT+4 点标步进 4 ALT+8 点标步进 8	(224)
(4)	ALT+A 设标志点 < 1 > ALT+Y 设标志点 < 2 >	(224)
(5)	CTRL+P 打印机选择	(225)
(6)	ESC 返回上一屏	(225)
(7)	点标控制	(225)
(8)	Ctrl+H 变屏幕底色	(226)
5.5.6	定画笔色	(226)
5.5.7	图形打印、退出系统和空格键	(227)
(1)	图形打印	(227)
(2)	退出系统	(227)
(3)	空格键	(227)
第六章 图形文件调用		(228)
§ 6.1	ETXL·EXE 覆盖式图形窗口调用程序	(228)
§ 6.2	ETXF·EXE 弹出式图形窗口调用程序	(229)
§ 6.3	ETXX·EXE 停留式图形调用程序	(230)
§ 6.4	ETXLL·EXE 覆盖式图形窗口调用程序 (变底色)	(230)
§ 6.5	ETXFF·EXE 弹出式图形窗口调用程序 (变底色)	(231)
§ 6.6	ETXZH·EXE 组合式图形窗口调用程序 (变底色)	(232)
§ 6.7	ETX99·EXE 动态提示式图形窗口调用程序	(233)
§ 6.8	两个图形调用源程序	(235)

# 第一章 图形处理基础

## 1.1 Turbo C 语言的数据类型、基本符号和控制语句

Turbo C 的每一条语句都是通过变量、常量和运算符组合在一起构成的,是程序设计中的基本元素,下面将简要的讨论这些元素和控制语句。

### 1.1.1 数据类型定义

C语言中的所有变量必须在使用之前定义,因为编译器在编译使用该变量之前,必须首先知道这个变量的类型。

C语言包含五种基本数据类型它们分别是:字符型、整型、实型、双精度实型和无值型。

Char类型变量表示8位Ascii字符,如'A'、'B'、'C'或其它任意8位值。int类型变量表示任意不带小数部分的整数值。float类型表示变量带有小数部分,数的范围 $3.4E-38$ 到 $3.4E+38$ ,由四个字节组成。double类型表示变量带有小数部分,数的范围 $1.7E-308$ 到 $1.7E+308$ ,由八个字节组成。列表如下:

数据类型表

类型	定义	字节长度	取值范围
char	字符	8	0 to 255
int	整数	16	-32768 to 32768
float	浮点数	32	$3.4 E-38$ to $3.4E+38$
double	双精度数	64	$1.7E-308$ to $1.7E+308$
void	无值	0	valueless

注:字节长度以二进制位为单位

### 1.1.2 类型修饰符

基本的数据类型可以在其前面带有各种修饰符。修饰符用于改变基本类型的含义,以便更准确地适应不同情况下的要求,Turbo C有4个类型修饰符列表如下:

名字	含义
signed	有符号
unsigned	无符号
long	长
short	短

修饰符 signed、unsigned、long和short可与字符型和整型配合使用,long还可以与double双精度实型量一起使用。下表列出了数据基本类型和修饰符组合的全部有效方式

和取值范围。

类 型	字节长度	取值范围
char	8	-128 to 127
unsigned char	8	0 to 255
Signed char	8	-128 to 127
int	16	-32768 to 32767
unsigned int	16	0 to 65535
Signed int	16	-32768 to 32767
short int	16	-32768 to 32767
unsigned short int	16	0 to 65535
signed short int	16	-32768 to 32767
long int	32	-2147483648 to 214783647
signed long int	32	-2147483648 to 214783647
unsigned long int	32	0 to 4294967295
float	32	3.4E-38 to 3.4E+38
double	64	1.7E-308 to 1.7E+308
long double	64	1.7E-308 to 1.7E+308

整型前面的signed修饰符可以省略,因为整型定义本身就是有符号的。

有符号和无符号整数的区别在于高位的不同解释。如果为有符号整数, Turbo c 编译程序将产生表示该整数的高位为符号标志位的代码。如果该标志位为0,则这个数为正数;如果为1,则为负数。

例:

	符号位
	↓
二进制的127为	0000,0000,0111,1111
二进制的-127为	1000,0000,0111,1111

有符号整数对于许多运算都是很重要的,但是它所能表达的最大数的绝对值只是无符号数的一半。

例:

二进制的32768为 0111,1111,1111,1111

如果最高位设置为1,则该数就会被当作-32768。如果将该数定义为无符号整数,那么当最高位为1时,它就成了65535。

### 1.1.3 其它数据类型

- <1> 数组。
- <2> 结构, 不同类型的变量构成的组。
- <3> 指针, 一个含有另一个变量地址的变量。
- <4> 联合, 两个或两个以上的变量, 它们占据同一块内存空间。
- <5> 枚举, 取值于有限集合元素的变量。
- <6> 串, 以NULL字符结束的字符数组。

### 1.1.4 常量

Turbo C中常量值规定方式如下表:

类 型	语 法	例 子
字符(char)	单引号	'a'
字符串(string)	双引号	'aaa'
整数(int)		123
八进制整数(octal int)	o	o777
十六进制整数(hex int)	0x	0x85
长整数(long int)	l或L	123L
浮点数(float)	小数或科学记数	3.123
		3.1E-24
双精度数(double)	同浮点数	所有浮点数被看作双精度类型

### 1.1.5 控制字符常量

下列的固定常量是C语言规定的控制字符。

- \n 换行
- \t 水平制表符tab
- \r 回车
- \v 垂直制表
- \b 退格
- \' 双引号
- \' 单引号
- \\ 反斜线
- \o 8进制常数
- \x 16进制常数
- \a 响铃
- \0 空

以上所有控制字符完全可以象使用任何其它字符那样使用。例如：

```
a='\t'  
B='\t'
```

### 1.1.6 运算符

Turbo C语言中有45个运算符。

( )	函数调用	getc ( ... )
[ ]	数组元素引用	i[5]
→	指向结构成员指针	alt_ptr→c
.	结构成员引用	alt.c
-	负号	-a
++	增1	a++
--	减1	a--
!	逻辑非	!found
~	求反	~ 0x7f
*	间址	*c_ptr
&	求地址	&a
sizeof (type)	变量或类型长度	sizeof(i) (int)c
*	乘	i*a
/	除	i/a
%	求模	i%a
+	加	i+a
-	减	i-a
<<	位左移	i<<2
>>	位右移	i>>4
<	小于	i<a
<=	小于等于	i<=a
>	大于	i>a
>=	大于等于	i>=a
==	等于	if(i==a)
!=	不等于	if(i!=a)
&	逐位与	c&a
^	逐位异或	c^a
	逐位或	c a
&&	逻辑与	i==8&& a==7
	逻辑或	i==8  a==7
?	条件表达式	i>a? i:a
=	赋值	a=10
*=	乘后赋值	a*=5

/=	除后赋值	a/=5
%=	取模后赋值	a%=5
+=	加后赋值	a+=2
-=	减后赋值	a-=2
&=	按位与后赋值	a&=0333
^	按位异或后赋值	a^=0333
	按位或后赋值	a =0333
<<=	左移位后赋值	a<<=5
>>=	右移位后赋值	a>>=5

### 1.1.7 运算符优先次序

所有的运算符,除了单目运算符和'?'以外,都是从左至右关联的。而单目运算符'!', '&', '-', 和 '?' 是从右至左关联的。

#### Turbo C运算符优先次序

最高	
() 函数调用 [] 数组下算 · (结构成负) → (结构成负)	左到右
! 逻辑非 ~ 值取反 - 负号 ++ 增量 -- 减量 & 变量地址	左到右
* 指针所指内容 (type)类型说明 -- 减量 & 变量地址	左到右
* 乘 / 除 % 取模	左到右
+ 加 - 减	左到右
<< 左移位 >> 右移位	左到右
< 小于 <= 小于等于 > 大于 >= 大于等于	左到右
== 等于 != 不等于	左到右
& 逐位与	左到右
^ 逐位异或	左到右
逐位或	左到右
&& 逐辑与	左到右
逐辑或	左到右
? 条件	右到左
= += -= *= /= 等联合操作	右到左
,	左到右
最低	

### 1.1.8 表达式

Turbo C表达式由下列之一元素构成:

1. 一个变量名。
2. 一个函数调用。
3. 一个数组名。
4. 一个常量。
5. 一个函数名。
6. 对结构成员的一次引用。
7. 对一个数组成员的一次引用。
8. 以上几个元素用括号括起通过运算符适当连接起来。

### 1.1.9 C语言控制语句

#### (1) if 语句

if 语句的一般形式为:

```
if (表达式) 语句1;  
else 语句2;
```

若表达式值为非零即为真, 执行语句1, 若表达式值为零即为假, 执行语句2, else语句2部分是选择项, 可以有或没有。例如:

```
if (表达式) 语句1;
```

if 语句可以是单个语句或复合语句, 在C语言中, 一个复合语句是由括号所括起的语句的组合, 子句else是任选的。

带有复合语句的if语句一般形式为:

```
if (表达式) {  
    语句段  
};  
else {  
    语句段  
};
```

#### (2) Switch 语句

Switch语句是Turbo C提供的多路选择语句, 称为开关语句。Switch语句将一个变量逐个与整型量或字符常数表进行比较, 当发现匹配时, 则执行与这个常量或字符常数相联的语句或语句段。Switch语句的一般形式为:

```
Switch (变量) {  
    case 常量1:  
        语句段  
        break;  
    case 常量2:  
        语句段  
        break;  
    case 常量3:
```

```

    语句段
    break;
    :
    :
    default:
        语句段
};

```

如果没有发现匹配, 则执行default语句, default是任选的, 如果没有这条语句, 当所有匹配失败时, 将不发生任何动作。一旦找到匹配时, 将执行与那个case相关联的语句, 直到遇见一个break语句为止。

### (3) for 语句

for 循环的一般形式为:

for (表达式1; 表达式2; 表达式3) 语句;

表达式1用于设置循环控制变量初始值; 表达式2是一个关系表达式, 通常测试循环变量是否结束; 表达式3规定循环重复时, 循环控制变量的变化方式。

例1:

```

#include <stdio.h>
main ( )
{
    int i;
    for (i=1; i<=100; i++) printf('%d', i);
}

```

例1中i初始值设为1, 由于i小于100, 执行printf()语句后i加1, 测试它是否小于或等于100, 如果小于100将继续循环, 一直进行到大于或等于100为止。

例2:

```

#include <stdio.h>
main ( )
{
    int i;
    for(i=100; i>0; i--) printf('%d', i);
}

```

例2中, i初始值设为100, 由于i大于零, 执行printf()语句后i减1, 测试它是否大于零, 如果大于零, 将继续循环, 一直进行到小于零为止。

### (4) while 语句

Turbo C语言中的第二个循环语句为while, 它的形式为:

```

while(表达式)
    语句;

```

表达式的值为非零(为真), 执行循环体中的语句, 直到表达式的值为零(为假)时结束



循环。

例1:

```
#include <stdio.h>
main( )
{
    char ch;
    ch='\0'; /*送空给ch*/
    while(ch!='A') ch=getche();
}
```

例1中,先初始化ch为空,然后检测ch是否等于A,每按一下键盘上的键,测试将进行一次,一旦键盘上的A键被按下,ch等于A,条件为假将终止循环。

表达式值为真时,执行循环体中的语句,循环体也可以是多条语句组成,但要用{ }括起来 它的形式为:

```
while (表达式)
{
    语句 1
    语句 2
    .
    .
    .
    语句 n
}
```

例2:

```
strlen(char s[])
{
    int i=0;
    while(s[i]!='\0') {
        i++;
        printf("%s\n",s[i]);
    }
    return(i)
}
```

函数strlen用来计算字符串s[]的字符个数,并把每个字符显示到屏幕上,直到判断字符为'\0'退出循环,然后将这个字符串的长度作为本函数的值返回给调用函数。

#### (5) do-while 循环语句

do-while 循环语句是在循环的尾部检验它的条件,而不是在头部,do-while 至少执行一次,也就是说,do-while 先执行循环体语句,然后判断表达式是否为真,若为真则继续执行循环体语句,若为假则终止循环 do-while的一般形式: