第14卷 第3期 华侨大学学报自然科学版 Vol. 14 No. 3
JOURNAL OF HUAQIAO UNIVERSITY
1993年7月 (NATURAL SCIENCE) Jul. 1993

C语言图形打印的算法和驱动程序的设计*

严桂兰 刘申耀

(电子工程系)

摘要 为保留C语言图形输出信息,从图形的屏幕输出原理和打印机工作方式,述了图形的打印机输出,并阐述从屏幕到打印机输出方式转换的方法及其程序设计.

关键词 图形输出,算法,程序设计

0 前言

计算机图形输出,给人以直观、清晰、准确之感,正广泛用于各个领域.目前,C语言的不断发展与完善,已使图形功能很强大,但C语言提供图形输出设备仅是显示屏,没有直接的图形输出打印函数.为此,本文将从程序设计的角度,用转换手段及打印机打印工作原理,编制图形打印驱动程序以满足C语言用户所求.

1 基本设计思想

当前,微型机显示终端大多采用光栅扫描显示器,其屏幕上光栅图象的形成,是由对应矩形矩阵中各亮点组成.实际上,屏幕上每点都对应一个二维象素的亮度值,以屏幕设置的背景色为准,若以该背景色画点,则为看不见的暗点而不构成图形点,反之则为亮点,即图形点.因而,光栅图象是通过逐点画出二维象素矩阵中各象素的亮度值产生.对于点阵打印机,其图形输出与光栅显示器上的输出类似,可将显示器上各象素的亮度值转换成点的位标志值,然后以字节单位数据在打印机上输出.

在Turbo C中,显示器上光栅各象素的亮度值可借助于getpixel()函数求得.该函数本是求指定点的颜色值,由于颜色已数值化,因而可考虑在下列环境中颜色值的另一含意.若以黑色为屏幕背景色,在屏幕上由黑色形成的点是看不见的暗点,不构成图形点;利用其它颜色形成的点则相对是亮点,可构成图形点.这样,当一幅图形在屏幕上显示时,可由左向右,从上到

本文1992-10-21收到.
福建省自然科学基金资助课题.

下逐点扫描，测定各点的颜色值．根据颜色值大小，将点的亮度值分为两类：颜色值为零的，其亮度值形成暗点，非图形点；颜色值不为零的，表示亮度值形成亮点，即图形点．按 Turbo C 中规定，当 getpixel()函数值为零时，所测定点的颜色正好是黑色，而点的其它颜色时的 getpixel()函值均大于零。我们利用 getpixel()函数值的大小便可判定屏幕上图形点的存在．尔后，将每8个顺序点的颜色值视为亮度值送给字节变量，用字节变量各位上的值来标志屏幕上对应点的亮度．当字节度量位标志为1时，其屏上对应点为亮点，位标志值为0时，则对应点为暗点．此字节变量数据再传送给系统 BIOS 支撑的 biosprint()函数，该函数将字节数据驱动连接并行口的点阵打印机，就可完成从图形屏幕输出到打印机输出的转换．

2　字节数据打印机输出的算法

设屏幕上图形区域大小，为所需打印输出图形的矩形坐标(x_1, y_1, x_2, y_2)．通过几层循环，将屏幕图形区域上的点由左向右，从上到下逐点测定其颜色值，每8个依次点的颜色值转换成位标志值组成一个字节数据．调用系统 bios 的中断17-biosprint()函数，使之设置成向打印机输出一个字节数据．字节数据各位标志值驱动对应打印头，即将 biosprint()函数设置成 biosprint(0,byte,0)形式，其中 byte 就是上述算法组成的字节数据．这样，就可将获得的字节变量数据 byte，逐一地输出到打印机．当 biosprint()函数在向打印机发送一个字节数据前，打印机必须初始化，使打印头复位到行的起始位置．同时，设置屏幕图形矩形左上角为打印机原点坐标，以及打印机处于图形模式．又因屏幕一行输出完毕后会自动转换成文本模式，所以每输出一行图形的点阵，都要重置一次图形模式．其具体算法为

```
打印机初始化并设置纵向走纸；
第一层循环 While(y≤y2){
  设定图形模式；
  第二层循环 While(x≤x2){
    字节变量，位变量赋初值；
    第三层循环 While(当前行纵向点增量次数≤8){
      取点颜色值；
      转换成位标志值放入字节变量中；
    }
    输出字节变量；
  }
  换行回车；
}
重新初始化；
结束．
```

3　打印机驱动程序设计

上述算法若用程序语言描述，则图形的打印输出就成为现实．现以9针点阵打印机为例，

Turbo C 语言为扫描工具，进一步阐述该算法的细节. 首先，打印机初始化命令为 ESC＋@形式，将 ESC，@的 ASC II 值作为字节数据，分别送入 biosprint()函数中，即 biosprint(0,27,0)与 biosprint(0,64,0)，便可连接打印机使其初始化. 打印机纵向走纸控制一般决定于针距与针数. 对于9针打印机，其打印头有9针打印针头，其中8根可用于图形打印，最后第9根针不动作，且排列为纵向，为横向走向，如图1所示.

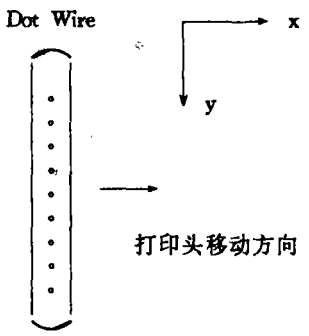

图1 打印机针头移动方向

9针打印机的每两针头间距离为1/72英寸，当其中8根针头接受字节变量中8位标志值后，则执行纵向打印动作，若要在下行继续打印，便要换行，促使纵向移动距离为8×1/72英寸. 通常，打印机纵向移动1/72英寸的命令由 ESC＋A＋(1)完成，现要移动1字节打印数据后的纵向移动距离为 ESC＋A＋(8)，即分别将27,65,8三字节数据送入 biosprint()函数，便可实现控制图形纵向走纸问题.

打印的图形模式由 ESC＋L＋(n_1)＋(n_2)命令完成，它置图形为单密度方式，其中 $n_1=(x_2-x_1+1)\%256$(取余)，$n_2=(x_2-x_1+1)/256$(整除). 只要将27,76，n_1，n_2顺序地由 biosprint()函数送给打印机，便使打印进入图形模式. 现进一步将屏幕上图形区域转换为打输出. 设图形的矩形域为(x_1，y_1，x_2，y_2)，每8个纵向象素的颜色值可组成一个字节数据，因而任一 y 坐标的(x_2-x_1)个字节数据就形成一行. 从任一行开始，用 getpixel()函数纵向依次求8个点的颜色值，而各点的 getpixel()函数值可直接作为逻辑判断条件；函数值为零是逻辑假值；不为零则是逻辑真值. 换句话说，当屏幕上非图形点时，该函数判断条件不成立；是图形点则判断条件成立. 据此可作相应不同的转换处理，组织字节中相应的数据. 如图2所示，当①点(x，y)，为图形点时，getpixel()函数值为真此时有如下转换处理.

(x,y)
①·
②·
③·
④·
⑤·
⑥·
⑦·
⑧·

图2 图形点示意

·图形点 ·非图形点

(1)将位变量 bit(初值为0×80)与字节变量 byte(初值为0×00)按位加，其结果再赋给 byte.

```
  10000000(bit)
| 00000000(byte)
------------------
  10000000(byte)
  ↑
```

于是，在字节变量 *byte* 的最高位就形成了标志图形点①的值1.

(2)bit 右移1位

```
  10000000(bit)
≫        1
------------------
  01000000(bit)
```

当②点(x，y+1)为非图形点时，getpixel ()函数为逻辑假值，可作如下转换 (1)byte 不变化，

byte=10000000.(2)bit 再右移一位,有

bit =bit≫1 (01000000≫1)

bit =00100000

由此,byte 左序第二位0就对应标志②点是暗点.同理,③点的处理同①点,即

byte=10100000

bit =00010000

如此循环类推,由图3示出byte 中各位数据与屏幕上象素亮度值的对应关系.这样,由屏幕上每纵向8个象素的颜色值,便得到一个对应的字节数据 byte.此字节数据及时送往打印机打印,再回到屏幕上扫描同行(或下一行)纵向8点象素的颜色值,便又组织了一个字节数据,送打印机打印.如此周而复始,直到所给定图形矩形域全部扫描完毕.送打印机打印,是利用系统提供的并行接口驱动函数实现的.系统中的 BIOS 是支持并行口的设备驱动,可以直接利用 C 语言中的 biosprint()函数.该接口函数驱动连接并行口的点阵打印机,并向打印机输出一个字节数据.当打印机针头接受"1"信号时,则动作打击纸面,形成打印纸上的图形点;若打印针头接受"0信号,则不打击纸面,无图形点形成,于是屏幕上图形就转向打印机输出,如图4所示.

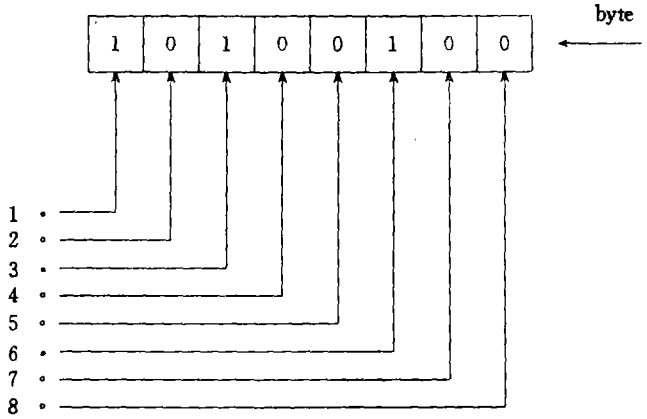

图3　byte数据与图形点关系

下面,进一步用 C 语言描述.先定义各字节对应的 ASCII 码值,即

```
#define   A     65
#define   L     76
#define   ESC   27
#define   AT    64
#define   F     10
#define   CR    13
#define   N     8
```

打印机初始化

```
biosprint(0,ESC,0);
biosprint(0,AT,0).
```

设置打印机纵向走纸

```
biosprint(0,ESC,0);
biosprint(0,A,0);
biosprint(0,N,0).
```

设置图形模式

```
biosprint (0, ESC,
0);
biosprint(0,L,0);
biosprint(0,n1,0);
biosprint(0,n2,0);
```

换行回车

```
biosprint(0,LF,0);
biosprint(0,CR,0).
```

程序的其它部分也不难得出，在此从略.

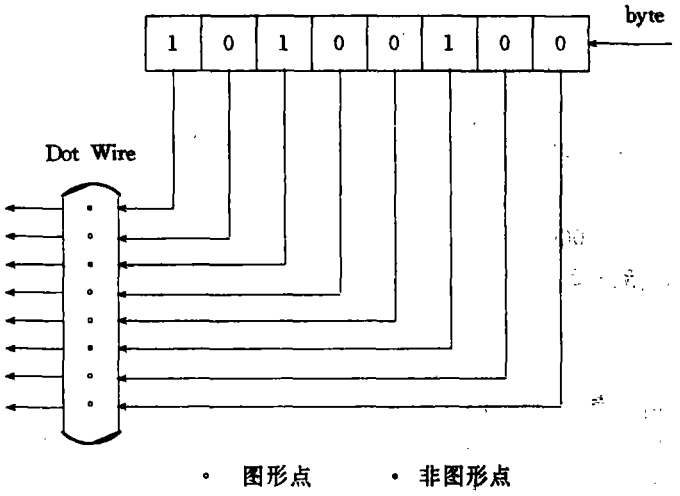

图4 byte数据与打印针头的对应关系

4 结束语

本文以9针点阵打印机为例，阐述了C语言绘图功能打印输出的算法、程序设计思想和具体步骤. 事实上，它也适用于各种点阵打印机. 该驱动程序是一独立的图形打印软件包，使用十分方便，不仅用于打印图形输出，而且对加有图形初始化的文本输出同样可用. 因而，对于当前图文并茂的输出结果，C语言图形打印的驱动程序设计是十分必要和实用的.

参 考 文 献

[1] 严桂兰、刘甲耀，C语言与图形处理，华东化工学院出版社，(1992).
[2] 刘甲耀、严桂兰，Turo C语言程序设计，电子工业出版社，(1991).
[3] 金连甫、王泽兵，IBM-PC图形处理入门，科学技术文献出版社重庆分社，(1990).
[4] 刘甲耀、严桂兰，PAD编程方法与C语言程序设计，电子工业出版社，(1989).
[5] 严桂兰、刘甲耀，C语言疑难问题剖析，华东化工学院出版社，(1992).

Algorithm for Graphic Output by Printer and Design of Driving Program in C Language

Yan Guilan Liu Jiayao
(*Department of Electronic Engineering*)

Abstract For sticking the information of graphic output in C language, the authors describe the graphic output by printer based on the principle of output screen and the operating mode output printer; and set forth the algorithm and its programming for converting the output mode from output screen to output printer.

Key words graphic output, algorithm, programming