

```
LD L,A
ADD HL
ADD HL,BC
JR(HL) ;转功能键处理
```

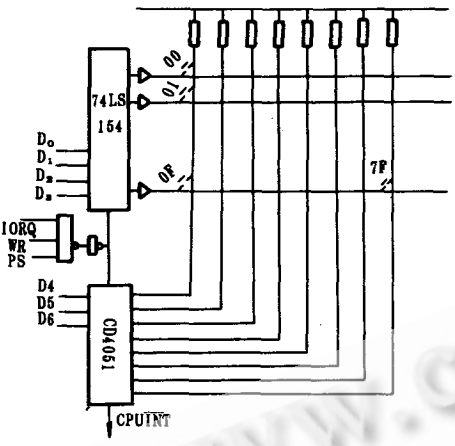


图 3

2. 简易 ASCII 码键盘

根据 ASCII 编码表可知: 数字键 0—9 的 ASCII 码为 30H……39H, 字母 A……Z 的 ASCII 编码为 41—5A, 据此, 我们可将键值(二进制)为 30H—39H 的键定义为数字键 0—9, 则在键扫描过程中, 当此十个键中任一键按下, 其二进制键值即为其相应数字的 ASCII 码键值。对于字母 A……Z 或其它的 ASCII 码符号, 其道理依然, 这样做的结果, 省略软件中代码间的转换, 而直接得出了其键的 ASCII 码供使用, 可见是十分方便的。

笔者在研制中医诊断电脑的过程中, 曾利用自制的上述键盘和 TP—801(16)微型打印机进行了实验, 证明是完全可行的。

三、结 论

第一、此种键盘在用于非编码键盘时, 扫描和键识别特别简单, 在非编码键盘中其扫描二进制数值, 即为实际的键值。在编码键盘中, 又可根据各数字、字符和功能键的 ASCII 码自行定义, 而直接得到其键的相应 ASCII 代码, 而且编程十分简单。

第二、硬件简单, 费用较低廉。

第三、中断方式可根据情况采用方式 0 和方式 1, 使用十分简单。

第四、当采用键盘自扫描硬件电路时(如图 3 所示), 也并不复杂, 且可从相应键盘的键值端口直接取出其相对应的键值或 ASCII 代码。

~~~~~

实现 WINDOWS 应用程序  
WINHELP 的色彩过渡效果

孙 乔 (北京理工大学车辆工程学院)  
陶曼希 (机械部郑州机械研究所)  
孙 璐 (河南省机械设备成套局)

摘要: 本文介绍在 DOS 环境下实现 WINDOWS 应用程序 WINHELP 的色彩过渡效果的方法。给出了具体步骤并分析了可行性。

Microsoft Windows 以其鲜明、生动的图形界面给广大计算机用户留下了深刻的印象。本文介绍在 DOS 环境下实现 Windows 应用程序 Winhelp.exe 色彩过渡效果的方法。

在安装 Microsoft Windows 以及运行 Winhelp.exe 时, 会看到兰色由明到暗平滑过渡的画面, 若能将这种画面放在自己的程序中, 会增色不少。从理论上讲, 这种效果可以通过设置调色板实现。只要调色板入口数足够多, 就可以实现十分平滑的层次效果。但是, 通常应用程序运行在高分辨率的十六色模式下, 这时, 若采用修改调色板的方法, 一方面, 因为最多只有十六个调色板入口, 所以仍不够光滑, 会有跳动的感觉; 另一方面, 如果十六个调色板入口全部被占用, 则应用程序就无颜色可用了。因此, 在实际的应用中, 是采用改变填充模式的方法来实现的。

在 Turbo C 的库函数中, 有一个为 Setfillpattern, 其调用为 void far setfillpattern(char far \* upattern, int color), 其中, upattern 指向 8 个连续字节, 这八个字节代表一个 8×8 点阵的 16 进制编码, 比如"AA, 55, AA, 55, AA, 55, AA, 55"这八个字节就可以产生 50% 灰度的填充效果。Windows 的 Control Panel 中的 Desktop 中提供了一些填充模式, 可以作为参考。在 640×480 十六

色方式下,如果按纵向过渡算,就可以有  $480/8=60$  种填充模式供使用,完全可以实现较为平滑的色彩过渡效果,而且只需占用一个调色板入口。获得填充模式的具体步骤如下:

## 一、编制 Windows BMP 位图显示程序

BMP 位图(十六色)的文件结构参见有关文章。

1. 获取调色板数据(一般地,除非人工修改,Windows 的调色板是固定的)
2. 获取位图尺寸。包括宽度、高度
3. 解码图形数据,显示位图。注意 BMP 位图数据是从图的左下角开始存储的。

## 二、截取 Winhelp 图

1. 进入 Windows
2. 在 FileManager 中运行 Winhelp.exe
3. 调整窗口大小
4. 按 Alt+PrtSc, 将画面截入裁剪板
5. 进入 PaintBrush, 将裁剪板的图"Paste"在编辑区
7. 用自编的 BMP 位图显示程序读取该图,生成数据文件即可得到填充模式。

以下程序可以再现 Winhelp 界面。其中, g1[] 为填充模式数据,笔者共获得 44 种。该程序用 Turbo C 2.0 编制。

```

/* 文件名: WINHELP.C */
/* 孙乔: 1993-11-12 */
#include <graphics.h>
#include <dos.h>
#include <stdio.h>
#define BKCR 15
void dispbmpgraph();
void color();
char g1[] =
{
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0xff, 0xff, 0x7f, 0xff, 0xff, 0xff, 0xf7, 0xff,
    0xff, 0xff, 0x7f, 0xff, 0xff, 0xff, 0x77, 0xff,
    0xff, 0xff, 0x77, 0xff, 0xff, 0xff, 0x77, 0xff,

```

```

    0xff, 0xff, 0x77, 0xff, 0xdf, 0xff, 0x77, 0xff,
    0xfd, 0xff, 0x77, 0xff, 0xdf, 0xff, 0x77, 0xff,
    0xfd, 0xff, 0x77, 0xff, 0xdd, 0xff, 0x77, 0xff,
    0xdd, 0xff, 0x77, 0xff, 0xdd, 0xff, 0x77, 0xff,
    0xdd, 0xff, 0x57, 0xff, 0xdd, 0xff, 0x75, 0xff,
    0xdd, 0xff, 0x57, 0xff, 0xdd, 0xff, 0x55, 0xff,
    0xdd, 0xff, 0x55, 0xff, 0xdd, 0xff, 0x55, 0xff,
    0xdd, 0xff, 0x55, 0xff, 0x5d, 0xff, 0x55, 0xff,
    0xd5, 0xff, 0x55, 0xff, 0x5d, 0xff, 0x55, 0xff,
    ...
    0x0, 0x88, 0x0, 0x2, 0x0, 0x88, 0x0, 0x20,
    0x0, 0x88, 0x0, 0x2, 0x0, 0x88, 0x0, 0x20,
    0x0, 0x88, 0x0, 0x0, 0x0, 0x88, 0x0, 0x0,
    0x0, 0x88, 0x0, 0x0, 0x0, 0x88, 0x0, 0x0,
    0x0, 0x80, 0x0, 0x0, 0x0, 0x8, 0x0, 0x0,

```

```

};
main()
{
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");
    color();
    dispbmpgraph();
    getch();
    closegraph();
}

void dispbmpgraph()
{
    /* BMP 位图显示子程序, 这里显示 "Windows Help" */
}

void color()
{
    char *p; int i;
    p = g1;
    /* 设置所需的填充颜色 BKCR */
    for(i = 1; i < 60; i++)
    {
        setfillpattern(p, BKCR);
        bar(0, (i-1) * 480 / 60, 640 / 60 * (i+1), (i+1) * 480 / 60);
        bar(640 / 60 * i, 0, 640 / 60 * (i+1), 480 / 60 * i);
        p += 8;
    }
}

```