

软驱磁头偏移校正程序

· 郭荣军

实际应用中发现软盘驱动器发生故障,大多是非电路故障,其中以磁头脏及磁头偏移最为常见。

在进行软盘读写操作中具体表现为数据不能读出或写入。例如,试图列出软盘目录及文件列表时执行命令:DIR A:(或 B:)时,显示器屏幕上显示“General failure reading driver A(B), Abort Retry, Fail?”严重时伴有很响的“吱吱”驱动器噪音。

一、磁头脏的原因及处理方法

由于长期频繁使用软驱,或使用质量较差的、受潮发霉的软盘,造成磁头上积磁、积污、积尘,从而影响软驱的正常读写功能。可用市售专用磁头清洗盘清洗磁头,步骤及方法如下:

1. 从清洗盘永久性保护中轻轻滑出白色盘片,并用手握住清洗盘片。

2. 清洗盘的结构与软盘类似,不同的是永久保护套内能旋转的圆盘片介质不同。将清洗液滴在清洗盘磁头槽表面上数滴(不必太多),待其散开后插入软盘驱动器,并关上驱动器开关。

3. 使清洗盘在驱动器中转动 20 秒左右(不必太长,否则会引起磁头偏移,因清洗盘对磁头的切向摩擦力较大),具体操作如下:可用执行 DIR A:(或 B:)回车的方法使清洗盘转动数秒。由于清洗盘是不能读写的,因此显示屏显示如上的错误信息。根据屏提示从键

盘打入字母“R”,再次使清洗盘转动,如此反复 3~5 次左右即可。

4. 将清洗盘取出并保存后,用正常的软盘进行读写试验。如果读写正常后则清洗完毕,否则重来。

有时由于磁头积磁积污太坚固,用清洗盘不能解决问题,此时可用手工清洗方法。方法如下:打开主机箱外壳,取出软盘驱动器,用缠有脱脂药棉的小棒,粘上无水酒精或收录机磁头清洗液,分别轻轻擦洗上下二个磁头。

二、磁头偏的原因及处理方法

由于软驱机械性能欠佳,或使用劣质磁盘,或者用清洗盘清洗磁头时未在盘上滴上清洗液等原因都会导致磁头偏移。当将磁头清洗后,仍然发生软驱读写错误时,应怀疑软驱磁头发生了偏移故障。磁头偏移表现为两个方面:磁头方位角及磁头径向位置偏移。

借助专用的工具(校正盘亦称猫眼盘、示波器、专用磁头定位程序等)校正磁头偏移最好不过。在业余条件下难以找到专用磁头校正工具,此时可用程序笔者自编的磁头偏移校正程序调整磁头,效果很好,读者不妨一试。

下面的 C 程序在 TURBO C 2.0 集成环境下编译、连接并生成 EXE 后调试通过(文件名取为 DK-TEST.C)。

波表分析器和主 CPU 所担任的处理内容彻底分开,可实现最高级处理能力的多媒体的应用。

在不大重视声音品质和整体性能的某种系统中,软件波表可以作为一种选择,但还不适于现在的主流 PC。

另外,Inter wave 处理器作为真正意义的 PC 音源硬件存取器,能给多媒体系统带来好的声音,在平衡性和成本效率方面都是一种很好的硬件音源。即从保存的波表数据生成声音时所必要的大量占用 CPU 的处理工作全部由 Inter wave 处理器实行。结果,在多媒体的应用时,MIDI 的顺序和数字声音的实现都不会大量增加 CPU 的占用率。

PC 音源的展望

各计算机工厂十分重视 PC 音源的开发,预计 PC 音源的品质在今后几年内会有更迅速的发展。

大多数的节目开发者,都把目光集中在多媒体游

戏、教育软件的音乐方面,希望能重现管弦乐谱所描述的声音情景,现在大多数的硬件合成器只能提供 32 种声音,而且认为这是一界线。

预计以 PC 为核心的音源制品,将会有更丰富的声音、特殊的效果和乐器声等。

MICROSOFT 公司不仅将发表的新的 API,节目制作者据此可生成更有迫力的声音。作为一例是有实现与动作真正呼应的 3D 声音的标准方法。游戏开发工厂利用这种 3D 声音,声音的效果与动作完全呼应。用别的 API,要使用许多的乐器合成。多通道声音(是 3D 声音的另一形态,适合于被动式多媒体节目)和娱乐中心 PC 作成一体化,而且在 PC 环境中加入杜比 AC-3、杜比定向逻辑。

有的用户对 PC 音源的变化虽不十分明了,但确实这种变化随时都在发生。

摘译自日刊《电子技术》1996 年 9 期

磁头偏移的校正步骤及方法如下

1. 预先将完好的数据盘插入相应的驱动器,然后在 DOS 下运行 DKTEST.EXE(该程序应已在硬盘或另一只好的软驱中)。则显示屏显示如下主菜单:

```

-SELECT DISK KIND! -
1--DISK A(40 TRACKS)
2--DISK A(80 TRACKS)
3--DISK B(40 TRACKS)
4--DISK B(80 TRACKS)
0--EXIT TEST DISK!

```

根据以上菜单选项,选择相应的需要校正的驱动器(1、2、3或4)。例如选“2”,表示需校正的驱动器为A驱(磁道数为80),按0退出回至DOS。

2. 选择需要校正的驱动器后,则屏幕显示如下子菜单:

```

-SELECT DISK HARD-
0-HARD 0 1-HARD 1
Q-SELECT DISK KINDS!

```

再由屏幕提示选择磁头号0或1(软驱有二个磁头,上磁头号为0,下磁头号为1)进行测试。如按Q或q,则退回主菜单,如按0则屏再提示:

```

-LOOP TEST ALL TRACKS? -
Y-LOOP N-Test one track

```

如按Y或y则在0磁道与最大磁道号之间进行循环测试。如按N或n则屏提示输入需校正的磁道号:

Input track-num;(输入磁道号后回车键有效)

在实际校正磁头偏移的过程中可先校正0磁道及最大磁道号,待这两个磁道校正通过后,再进行从0磁道至最大磁道号之间的循环校正。在测试校正过程中可按任意键退回到磁头号选择子菜单。

3. 我们知道盘片是顺时针旋转的,因此磁头多偏向右边,调正时边运行上述程序边用手扶到上磁头(上磁头在盘片加载时是活动的易偏移),轻轻往左带,边调边观察屏幕显示,如显示“TRACK TEST PASS!”且能听到振铃声表示已调整好。如果此方法不能解决,就放松固定磁头支架的两颗螺丝,将磁头向左或径向推稍许,然后边试边观察屏提示,要细心调整,动作不可太大。下磁头(1号磁头)的调整方法类似。

4. 在应用本程序进行磁头校正前还应注意,软驱的0磁道开关位置是否发生了偏移?可用本程序分别对上下二个磁头进行循环测试,如都不能通过,则应先边运行本程序边调整软驱0磁道交电开关位置(即本程序也能调整软驱0磁道光电开关位置偏移)。

磁头偏移校正源程序清单如下:

```

#include<stdio.h>
#include<bios.h>
main()
{char buf[512];
char drv[]=[0,40,80,1,40,1,80];
int flag,k,hdn,tkn,maxtkn,drv;
char * disp[]={"--SELECT DISK KIND! --",
"1--DISK A(40 TRACKS)",
"2--DISK A(80 TRACKS)",
"3--DISK B(40 TRACKS)",
"4--DISK B(80 TRACKS)",
"0--EXIT TEST DISK!"};

textmode(3);
main,clrscr();
for(k=0;k<6;k++)
{gotoxy(10,10+k);
printf("%s",disp[k]);}
try: k=getch()-48;
if(k<0||k>4)goto try;
if(k==0)return;
drv=drv[2*(k-1)];maxtkn=drv[2*k-1]
hard,clrscr();
gotoxy(10,10);
printf("--SELECT DISK HARD! --");
gotoxy(10,11);
printf("0-HARD 0 1-HARD 1");
gotoxy(10,12);
printf("Q-SELECT DISK KINDS!");
hdn=getch();
if(hdn=='Q' || hdn=='q')goto main;
hdn=hdn-48;
if(hdn<0||hdn>1) goto hard;
gotoxy(10,14);
printf("--LOOP TEST ALL TRACKS? --");
gotoxy(10,15);
printf("Y-Loop N-Test one track");
k=getch();
if(k=='Y' || k=='y')
{tkn=0;
do{
if(tkn==maxtkn)tkn=0;
retest1:flag=biosdisk(2,drv,hdn,tkn,1,1,buf);
gotoxy(10,17);
printf("%d",tkn);
if(flag==0)
{printf("--TRACK TEST PASS!");
printf("\n");}
else
{printf("--TRACK TEST ERROR!");
if(bioskey(1)!=0) goto hard;
goto retest1;}
tkn++;
}while(bioskey(1)==0);
}
else if(k=='N' || k=='n')
{
totoxy(10,17);
printf("Input track-num:");
scanf("%d",&tkn);
retest2:if(bioskey(1)!=0)goto hard;
flag=biosdisk(2,drv,hdn,tkn,1,1,buf);
gotoxy(10,17);
printf("%d",tkn);
if(flag==0)
{printf("--TRACK TEST PASS!");
printf("\n");}
else
{printf("--TRAcK TEST ERROR!");
biosdisk(0,drv,hdn,tkn,1,1,buf);}
goto retest2;
}
goto hard;
}
}

```