

E & C

1993

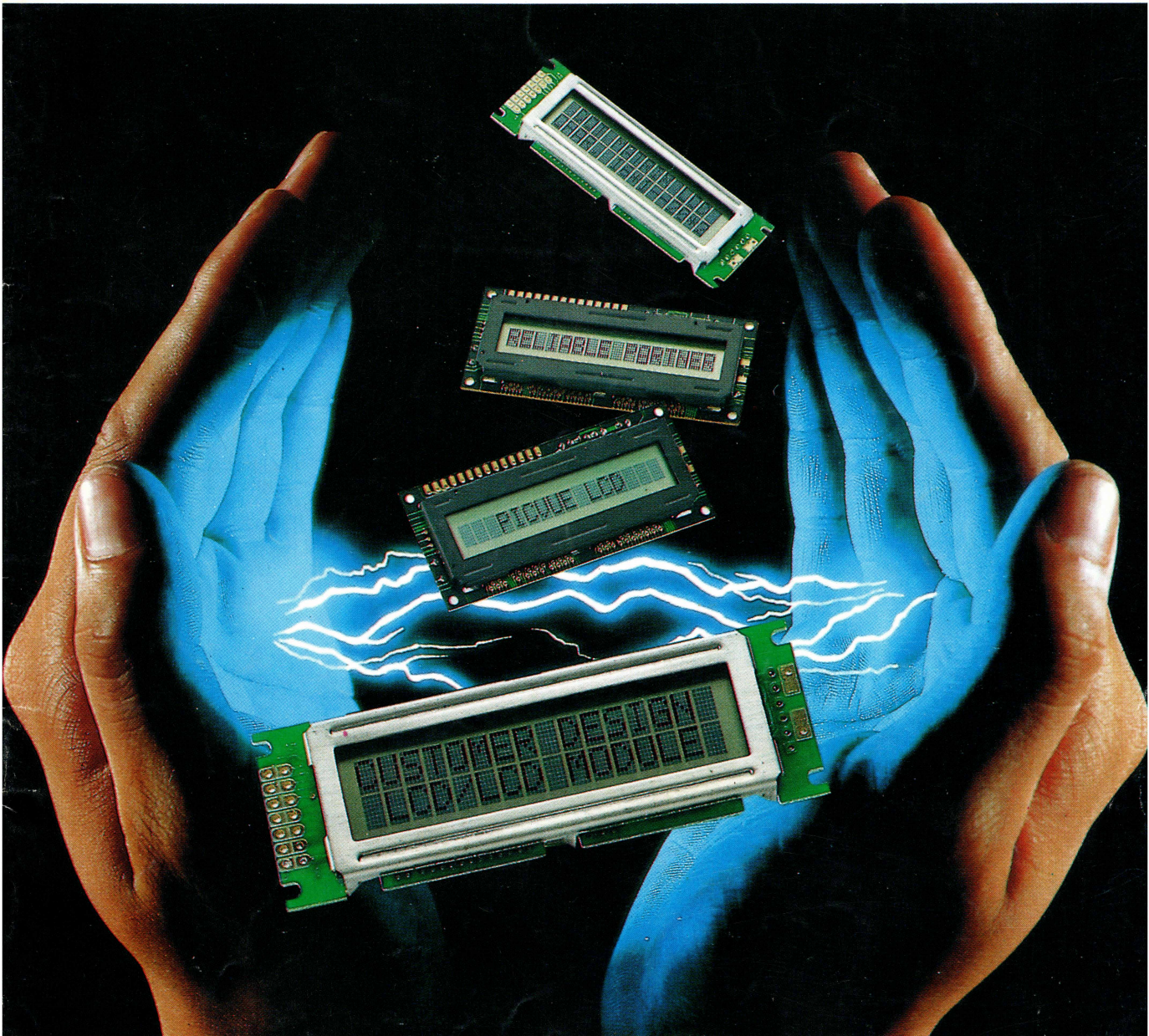
一九九三年 ● 总期第101期

8

電子

ISSN 1000-1077

與 電腦



• ELECTRONICS AND COMPUTERS •



电子工业出版社广州科技公司

邮购产品介绍

软件狗 (WATCH DOG) —

——软件的忠实保护神

WATCH DOG 是一种新颖的软件加密工具。它体积小 (60×50×15mm), 只需将其插入主机箱外的并行输出插口上 (即串接在主机并行口和打印机电缆之间, 不占用机器扩展槽)。

当主机安装了某个 WATCH DOG (设其为 1 号狗), 即可通过 WATCH DOG 对您使用的软件进行加密, 被加密的软件可以被任意拷贝但不能在没有加装 1 号狗的的机器上运行。或者从另一个角度来看, 当某软件用 2 号狗加密后, 该软件只能在装上 2 号狗之后才能运行。也就是说, 如果您的软件要加密保护, 那就每一套软件必须有一个相应的 WATCH DOG。

适用范围: 可对 C (BORLAND C⁺⁺、TURBO C、MSC C)、编译 BASIC、FORTRAN、PASCAL、汇编语言、dBASE (Clipper、dBASE III、Fox base、FOXPRO) 等写成的软件加密。也可对 EXE 可执行文件加密。

加密方法: 只要在源程序中加入调用 WATCH DOG 函数的语句, 编译时将随 WATCH DOG 提供的 OBJ 文件链接起来, 形成加密的 EXE 文件, 或者调用随 WATCH DOG 提供的 LINKEXE.EXE 对可执行的 EXE 文件加密。

特点: 1. 采用多种反跟踪, 反破译, 反仿制技术。完整的 WATCH DOG 代码大于 9K。WATCH DOG 的核心程序几乎不可能被破译。

2. 能直接与 EXE 可执行文件链接, 产生一新的 EXE 可执行文件。完成对 EXE 文件的加密。

3. 每一 WATCH DOG 及随 WATCH DOG 提供的软件都有一唯一的 ID 号, 无任何信息、资料可查询到这些 ID 号。WATCH DOG 及随 WATCH DOG 提供的软件是一一对应的, 其间不能互用。

4. 对各类打印机正常工作无任何影响。

5. 对于网络用户, 可以定购一批同 ID 号的 WATCH DOG。

WATCH DOG 可能是您最好的软件加密工具。

SUNSHINE WY09-3B 开关

电源

输入电压: AC220V ± 15% 0.7A 50Hz

AC110V ± 15% 1.4A 60Hz 开关切换

输出电压:

+5V 端: 恒压输出, 其可调范围 4.5V~6.5V,

主负载输出端

工作电流 最大值 15A

纹波电压 小于 100mV

瞬变特性 100% 负载变化时优于 ±

0.5%

+12V 端: 恒压输出电压误差范围 ± 2%

工作电流, 最大值 1.5A

纹波电压 小于 50mV

瞬变特性 100% 负载变化时优于 ± 2%

-12V 端: 恒压输出 电压误差范围 ± 2%

工作电流 0~0.5A

纹波电压 小于 50mV

瞬变特性 100% 负载变化时优于 ± 2%

输出功率: 75W 最大值 90W (需加 2mm 水柱
高压风散热)

空载功率: 小于 6W

工作效率: 大于 82%

开机涌浪: 小于 20A / 10ms

隔离电压: AC、DC、FG 间大于 1000VAC 1Min

TEM = 27° HUM = 60%

绝缘电阻: AC、DC、FG 间大于 100MΩ

TEM = 27° HUM = 60%

工作环境: TEM = -10°C ~ +35°C HUM = 10% ~ 90% REL

工作温升: 在 +27°C 自然无风冷散热情况下, 机身温度低于 +70°C

外形尺寸: L158 × W98 × H48mm

机身重量: 小于 0.5kg

本机还具有交流滤波, 开机缓冲冲击、过载保护、输出输入短路保护、LED 工作指示、外壳与地隔离等功能。

主要应用于微外理机, 可编程控制器、高精程控制器、精密电子仪器、开发机, 显示屏供电等。

公司地址: 广州市天河五山路华师大科技服务楼 215 号 邮 编: 510631

开户行及账号: 广州建行天河高新办 480-261485-55 联系人: 陈昊

电子工业出版社图书邮购消息

C1101	家电维修技术精华丛书(1)——收音机	7.20
C1102	家电维修技术精华丛书(2)——收录机 组合音响	7.50
C1103	家电维修技术精华丛书(3)——黑白电视机	12.50
C1104	家电维修技术精华丛书(4)——彩色电视机(上)	13.50
C1105	家电维修技术精华丛书(4)——彩色电视机(下)	12.60
C1106	家电维修技术精华丛书(5)——录像机 摄像机	7.40
C1107	家电维修丛书(6)——卫星电视接收机 共用天线	5.30
C1108	家电维修丛书(7)——电子表 电度表 万用表	5.30
C1109	家电维修丛书(8)——游戏机 家用微电脑 计算器	5.40
C1110	家电维修技术精华丛书(9)——家用制冷 空调设备	15.50
C1111	家电维修技术精华丛书(10)——小家电	7.50

注:购书另加邮挂费 15%,购全套书 115 元/套

邮购办法

邮局汇款

地址:北京市万寿路电子工业出版社发行部邮购科

银行汇款开户行:北京工商行翠微路分理处

账号:661036-40

户名:电子工业出版社发行部

电话:8233693

邮编:100036

附注:购书请用正楷书写清楚代号、名称、详细地址和收取人(邮局汇款写在附言栏内、银行汇款附上清单)。发书周期约 30 天。如出现邮寄中丢失,可向我部索赔。



电子与电脑

一九九三年总第 101 期

目 录

· 综述 ·

- 电子图书的发展及中国市场的开拓 严 华(2)
如何选购个人电脑 廖 佳(3)

· PC 用户 ·

- PCX 格式图像文件的应用技巧 周光辉(5)
一程序多报表的自动建立、修改与输出
—dBASE 与 CCED 的结合使用 刘士杰(8)
西文文本方式下怎样利用 VGA 及其兼容显示器
显示汉字 樊启柏(9)
直接读/写 VGA 的图形像素 王绍鑫(12)
2.13F 系统在单显环境下使用所遇几个问题的
解决办法 张彦超(13)
巧用 INT16 李珊红(14)
安全可靠的扩充功能键高级编程技术 侯培玉(15)
计算机病毒的检测、消除和预防 苏民生(16)
文件型六四病毒的简析与防治 李超勇(17)

· 学习机之交 ·

- 改进的 CEC—I 实用小程序 薛嘉兴(18)
ProDOS 系统内部结构剖析(续) 廖 凯(21)

初、中级程序员软件水平考试辅导

- 第四届全国计算机软件人员竞赛将在京举行
(初级程序员级) (25)
一九九三年计算机初级程序员有奖竞赛试题答卷
..... (25)
一九九三年计算机初级程序员有奖竞赛试题 (26)

· 新书与软件 ·

- PCT V2.0 写作系统 张连永 杨 峰(23)

· 字用单片机 ·

- 监控程序阅读器 郝 键 罗明宽(30)

· 电脑巧开发 ·

运用“看门狗”技术提高在线式智能仪器的可靠性

- 汤忠庆(33)
如何将苹果机 AD/DA 卡改制为 PC 机 AD/DA 卡
..... 王 晰(34)
UN-1A 型电压模拟数字采集器 刘北顺等(36)
给 PC 机配装射频调制器 郭荣军(39)

· 维修经验谈 ·

- 谈谈 LQ1600K 打印头的维护及短(断)针利用
..... 刘 斌(40)
后备式中小型 UPS 不间断电源的改进 谭巨高(41)

· 电脑游戏机 ·

- 第二章 6527CPU 的显示系统(上) 于 春(42)

· 电脑通信 ·

- 传真机专题讲座
传真机的日常维护 张建军 张景生 李晓中(47)
电脑通信大家谈
通信软件的使用 (49)
NOVELL 网络的保密技术 王杰民(50)

· IC 电路应用 ·

- 模拟乘法器 IC 及使用方法(一)(续) 李兰友(53)
节能的 IC 小电路 向 心(54)

电子信息集锦

读者联谊

- 对《最近出现的几种病毒》的几点异议 (52)
PC/XT 及其兼容机内存的扩充与使用 李晓中(55)
让文本阅读器显示汉字 许 勇(52)
无绳电话干扰电视节目 钱雁群(56)
回音壁
单片软光盘 21MB 系统问世
电子计算机新一代存储设备 (56)

电子工业部电子工业出版社主办

编辑、出版:《电子与电脑》编辑部

(北京 173 信箱 邮政编码:100036)

印刷:北京三二〇九厂

国内总发行:北京报刊发行局

国内统一刊号:CN11-2199

邮发代号:2-888

国外代号:M924

出版日期:每月 23 日

主编:王惠民 特约编审:苏子栋

责任编辑:施玉新

订购处:全国各地邮电局

国外总发行:中国国际图书贸易总公司

(北京 399 信箱 邮政编码 100044)

广告经营许可证:京海工商广字 147 号

定价:1.60 元

电子图书的发展及中国市场的开拓

海天电子图书开发公司 严华

在人类出版史上,一场可以和造纸及印刷术发明并称的革命性转折,在二十世纪80年代后期悄然兴起,现已初步定型,三五年后会大放异彩;二十一世纪,将是电子图书的世纪。

80年代中期,第四代超小型、高容量的微型个人计算机异军突起,在电子计算机行业占据重要地位。此时,人们便萌发了将以纸为媒体的传统图书进行改造的念头。一张3.5英寸的高密磁盘,能够容纳140多万字的英文或70万字的中文信息,这已经是个令人咋舌的数字,然而它只是文本信息的集结,即使在屏幕上显示,也还达不到传统图书的高清晰度,阅读起来甚为费力。被誉为“出版领域又一次革命”的电子排版系统,用精美的字模和多变的字体取代了曾在人类信息传播史上光耀千载的活字印刷,但它所迈开的仅是这场伟大变革的第一步,往日的连篇累牍不仅依然固我,而且在生活节奏日益加快、信息淘汰周期迅速缩短的现代化社会里,愈来愈成为人们的累赘和负担。于是,将纸介质转换为更为轻巧灵变的磁介质,将文字信息分页检索发展为几千万字的一次性全书查询,成了这场变革的关键。

一开始,人们尝试着把大容量的音像磁带作为新的载体。一盘常规的录像带可以容纳100兆字节左右,相当于近80余张高密磁盘。但它的体积并不比大容量的芯片小多少,更关键的在于,其图文质量达不到要求,且容易擦除、蜕变。作为计算机数据存储介质的芯片,容量大,体积大,但也只适于专家存储编好了的数据,而不适宜用作大众传媒介质。人们把装有固化信息的芯片置于一个个小匣子中,配上西文键盘来输入命令进行检索查询和逻辑运算,并从日益清晰的液晶屏幕上看到检索或运算的结果。这便是被称为“电子辞典”的新型“书籍”——实质上它是用途专一的超小型微机。和传统图书的“纸张+文字”相比,它达到了轻便易携和利于检索等要求,但其制作过程则复杂得多,成本也昂贵得多,即使批量生产也难以做到“贫富兼得”、雅俗共赏;而且一个微处理器只能支配自身的数据,仅这一点就造成了不必要的耗费。因此,这个“新生儿”与其说是“电子图书”,毋宁称为“电子工具书”更为恰当些,尽管它也有着光明的前途,但仍不构成与传统图书争夺市场的对手。

光盘的出现,对“白纸黑字”的未来形成前所未有的冲击。这个新生命给人们带来的,不仅是变幻的图文形象,漂亮的观感效果,千万种色彩辉映,还有声与象的谐和,动与静的融汇,即时即刻的展现,不同形式的

输出……。批量生产起来,其价格和常规图书相差无几;而其容量之大,更令人叹为观止:一片重仅百余克、大如手掌、薄如板纸的盘片,居然装载着6.6亿个西文字符,3.3亿的中文内容!也就是说,十大卷本、3千万字的中文版《简明大不列颠百科全书》,只能占其“肚量”的十分之一,而多达80余卷的《中国大百科全书》,其文字量也未必装得满一张光盘。至于中国古人所说的“学富五车”和“读破万卷”,在这里头都已应有尽有。

一开始,科学家们发现用激光原理发明的视盘和音盘,储存电视和音乐节目不仅比塑胶唱盘容量大,而且图像和音响效果都好得多。但那些只是视频和音频的模拟信号,和计算机的数字信息并不相容。CD音盘和数码唱机的出现,带来了音像领域的巨大变革,从而也叩开了利用光盘存储数据信息的大门。80年代中期,美国科学家们用12英寸的光盘一次性地写入信息(CD-ROM,即COMPACT DISK—READ ONLY MEMORY,直译为只读激光盘),一个和普通胶木唱片规格大小的12英寸光盘,最大的容量是50多万个图像或200个G的英文文字内容(1G=1000MB,即一千兆);1985年,个别计算机厂家研制出5.25英寸即和计算机普通磁盘一样大小的一次性写入光盘并投放市场;1988年,5.25英寸光盘在西方计算机数据制作领域得到广泛应用,同时,可擦写的5.25英寸光盘(CD-I,即COMPACT DISK—INTERVIEW,直译为交互式激光盘)问世;1990年,5.25英寸多功能光盘驱动器面世,它不仅能读出一性存储的光盘数据,同时还可重新写入新的内容。在此之际,体积更小的3.25英寸光盘应运而生,它那掌心可持的躯体内,有着多达256兆的空间,正好与中等文字量的著作及相关的图像和动态画面相匹配……。

光盘图书的出现使电子图书的性能价格比发生了显著的变化。英国牛津大学最近出版的光盘本《牛津英语大辞典》,将原来重达六十二公斤、册数为二十本的巨著化为一张百余克重的光盘,原书售价为2750美元,而光盘本的价格为895美元,仅是前者的三分之一。美国的康普顿多媒体百科全书,售价为250美元,是同等纸质百科全书的五分之一。其图文并茂、声画合一、动态环境等表达方式,远不是一般百科全书所能望其项背的。光盘图书之所以价格低廉,主要是因为其可以像常规图书或唱片一样实行一次性“刻印”,“印数”愈多,成本愈低。以《中国大百科全书》为例,现在80卷本的总销售价格应为人民币6000元左右,而如果刻印

3000册光盘图书,其销售价便可以控制在2000元人民币以下。

接着人们关心的就是阅读器具。体积小巧玲珑、内容极为丰富的光盘固然便于携带,但要随时阅读它,没有机器是不可能的。就像十年前电视迷们出差旅游为旅馆没有电视机而遗憾一样,将来的读者会不会产生带有光盘而无法阅读的担忧?

这种担心是多余的。历来硬件的发展速度都不会落后于软件。目前,光盘阅读器有三种,一种为内置式,即安装在286以上的计算机机箱内;第二种为外置式,不同的专线可以把各种微机和装有光驱的小方盒连接起来;第三种为一体化光盘阅读机,目前已发展到“发烧友”们热衷的便携式CD唱机那样大小。1992年底,日本索尼公司推出了一种称为“掌上明珠”的电子图书系统,它用手掌大的光盘阅读机驱动直径为3.5英寸的光盘,机体上同时配有袖珍型标准键盘和一个7.5厘米见方液晶屏幕,供读者输入信息、命令和观看光盘中的内容。整个机器重量仅为550克,比我们的《新华字典》还要便于携带。3.5英寸的光盘容量为128MB,即可存储1亿2800万个英文字符或6400万个中文文字信息。如按页计算,它可以容纳10万个页码的文字或上万张图片信息。目前,SONY公司已经推出供这种机器使用的光盘25种,主要有字典、辞典、成语故事、笑话幽默图书、外语对译和歌曲大全、影视新片等。其中一张语言对译光盘收录了日、汉、英、法和世界语等多种语言内容,使用者可在25秒内查找到日语某一词汇在任何一国语言里的表达方式。英国的一家公司最近推出一种触摸式袖珍光盘阅读器,只要读者在液晶屏幕上触摸一下某一功能的提示“菜单”,就能读到你所需要的信息或图书内容。

1993年,各国电子图书专家们将在光盘阅读器的“随身化”方面开始竞争。身携一个光盘读物,随时阅读、欣赏音乐并将它和电话传真、电视系统连接起来,已经形成人们对光盘阅读器的新的要求。显而易见,光盘读物的前途十分光明,根本不存在对其阅读不便的担忧。

电子图书的兴起,会使光盘存储和常规图书的制作紧密配合起来,交相辉映。尤其在前期编辑加工和版式制作上,二者的部分方法是一致的。光盘存储将为常规图书解决原来最为繁难的彩色电分 and 排版问题,同时还能解决将电视图像直接在计算机中定格并输出的难题。只要用高精度的彩色扫描仪将图像SCAN到光盘里,或者用摄像机把某一情景摄入计算机,系统就会把印刷用四种彩色版自动分出来,接上出软片的终端,就会输出供各种彩色印刷机使用的胶片来。目前,北京大学的“方正彩色照排系统”已经成功地做到了这一点,并推出其第一代产品。另外,光盘在数据存储和检索上的先进方式,也将为大型图书的制作提供方便。

在德国,出版家们已把排制好的书版装进光盘,放在书店内播放,供读者“看样定货”,只要读者需要,马上就可用激光印字机输出、装订好后单本“出版发行”,

有时甚至可以根据读者的要求部分印制。这种即要即得的方式对购买者来说价格可能贵一点,但他找到了急需的信息资料或读物;对出版者来说,其先期投资比起印刷一批到书店中寄销来,成本不知要降低多少倍。这种方法可以给我们提供不少有益的启示。

当然,传统图书也有电子图书一时难以达到的优点,如清晰度、视觉效果等。随着屏幕显示精度的日益提高,电子图书对这些问题将逐步解决。而电子图书的多媒体的动态环境,可能是常规图书望尘莫及的。

在中国,电子图书的开发已经具备条件,市场也将渐渐开启。首先,机关、集团和经济发达地区的图书馆、资料室的微机普及率已有很大提高,有的甚至接近发达国家水平;其次,世界光盘开发和研制业已达到标准化和产品定型的阶段,这为经济实力较弱的大陆地区后期开发奠定了基础;第三,中国的电子行业对光盘的研制开发和批量生产也已接近成熟状态,1992年12月深圳的先科公司研制出的第一张国产CD-ROM光盘便是标志;第四,中国经济发达地区对信息数据的综合处理为光盘系统开发提供了市场,同时知识分子把微机引入创作过程、期待狭小的家居空间能够有更多的图书等要求也为电子图书的大众化提供了条件;第五,几年来,中国培养出一大批中文数据处理和软件编辑人员,为电子图书开发系统的研制和大规模的编辑制作模本提供了智力准备;第六,中国的电子排版和激光照排技术已达世界先进水平,出版业和印刷业的自动化程度也已接近批量生产的要求。

在这场历史性的变革中,我国出版业再也不能坐着等待了。一些出版社已相继成立电子图书开发机构。几部电子图书也在制作中。然而,在没有一套中文电子图书开发系统的情况下,开拓庞大而又荒芜的大陆电子图书市场,其难度是不言而喻的。在这种电子图书研制只是“事业”、几年后才能成为企业的态势下,中文电子图书系统的研制和开发应是整个中国出版业共同关心的大事。我们坚信,谁选迈开第一步,谁在未来的竞争中 will 占据有利地位。

(本文为作者92年12月在“第六届中国图书奖颁奖大会”上的讲话稿,本刊略作压缩)

如何选购个人电脑

中国人民银行总行研究生部(100083)

廖佳

目前许多家庭因个人或孩子的需要,想购买一台微机,但却为不知购买什么型号、什么配置的电脑而烦恼。下面就笔者的经验,谈一下购买微机及微机使用的一些注意事项。

目前市场上个人用电脑有中华学习机、PC机、286、386等等。中华学习机是为孩子学习计算机而开

发的,它与 Apple I 机兼容,与 PC 机不兼容,也就是说 PC 机的软件无法在中华学习机上使用,反之亦然。此机除只能学习简单的语言和字处理外,没有太大的用途。PC 机已属于淘汰机型,其在运行速度和内存、外存配置上都已不能适应现在发展的需要,所以建议不选此机型。286 是目前个人使用首选机型,其运行速度和配置足可以满足个人的一般需要;价格在 2500~5500 元之间,一般家庭可以承受。目前 286 机是一种过渡机型,在有些地方还不太完善,所以在经济条件允许的情况下,建议选用 386 机型,其价格在 4600~1 万元之间。486 机虽好,但价格昂贵,若个人使用,只能是大材小用,浪费财力,不建议购买。

在购买微机时,主机、外设及其配件应带有使用说明书。对微机系统进行安装和调试时,有时需要参考此说明书。用户购机时需要索取并保存好。

在购买微机时需注意以下几个方面:

1. 主机

在选购主机时,要购买兼容机,而不要购买杂牌机。因为有些杂牌机都是用一些质量较次的配件组装的,运行一段时间后就会发生故障,或无法运行一些软件。386 机分为 386SX 和 386DX 等型号,前者并非是真 32 位机,其数据总线与 286 相同,拥有一条 16 位数据总线;而后者是真 32 位机,拥有一条 32 位总线。买 386 机就要买 386DX 型机。

2. 机箱

主机的机箱分立式和卧式,在选购时最好选择大机箱,因为容积大,易散热,容易扩展其它硬件。

3. 电源

在选购微机时,应选择带有 200W 或以上电源功率的主机。这样可以带动更多的接口卡,若电源功率不足,则可能引起死机或烧毁电源等故障。

4. 主机板

微机的主机板分大主板和小主板。大主板是指较旧型的主机板,体积稍大,其元件通用性较好,若元件坏了,很容易找到配件,但运行速度较慢。小主板是指较新型的主机板,体积略小,其元件集成度高,大部分为集成电路,运行速度快,但元件不好配。用户购买时可自行参考。主机板上至少要有 4~5 个扩展槽,以便将来扩充之用。目前市场上许多微机用的是小主板,其厂商为了降低成本,而将不少元件省略了,体积也很小,但质量不稳定。建议用户购机时还是选择大主板。

5. 主机速度

微机的主频分为 16,20,33,40MHz 或更高。建议选择 20MHz 以上并带有 TURBO(加速)键的主机。386 机应选择 25MHz 或以上的。主频越高,运行速度越快。

6. 内存

目前主机内存有 640K,1M,2M 或更多。对 286 机,建议用户选择 1M 或以上的内存,对 386 机,建议选择 4M 或以上的内存,否则许多汉字系统无法使用。注意目前内存为 1M 的 286 机分两种:一种是主存为

640K,扩展内存(Extended)为 384K;一种是主存为 640K,扩充内存(Expanded)为 384K。前者用 XMS 内存管理程序进行管理,许多汉字系统一般将显示汉字库放入扩展内存进行使用,而后者需用 LIM4.0 EMS 软件驱动程序,大部分汉字系统无法使用扩充内存,只能将扩充内存作为硬盘高速缓存区,所以建议用户购机时,要选择扩展内存(Extended)为 384K 的 1M 内存。内存应选用存取时间为 70 或 80ns 的 RAM 存储器芯片。较好的主机板均采用同一牌子同一存取速度的 RAM 芯片。因为不同牌子不同速度的 RAM,其操作速度不同,对数据的传送有所影响。在 RAM 芯片上都有一个象 44256-8 的数字,其中 256 表示 256K 芯片,在最后面的 1 位或 2 位的数字表示存取时间。7 表示 70ns,8 表示 80ns,10 表示 100ns,数字越小表示存取时间越短,可加速软件的运行。

7. BIOS

BIOS 是微机的基本输入/输出子程序,它固化在主机板的 EPROM 上。建议用户要购买兼容性好的 BIOS。其制造厂商主要有 IBM、COMPAQ、PHOENIX、AWARD、AMI 和 DTK 等。

8. 多功能卡

许多微机都安装有一块多功能卡,它包括两个软盘驱动器接口,一个硬盘驱动器接口(可连接两台硬盘),两个串行口,一个并行口(打印机用)和一个游戏棒接口。串行口可用于连接鼠标器及通信设备等。有些微机并没有此卡,而是用软硬卡(只有软硬盘驱动器接口),这样以后扩展时会很麻烦。目前市场上有许多简化的多功能卡,其体积较小,质量较差,建议用户不要选择此卡。

9. 显示器及显示卡

目前显示器有单色显示器和彩色显示器。用户可根据自己的需要选配。单显应选择 720×350 分辨率的双频单显及双频卡。彩显应选择 VGA 高分辨率显示器,其最大分辨率为 1024×768,256 种颜色,点间距为 0.28 毫米,垂直扫描频率在 70Hz 或以上,水平在 31.5KHz 或以上。较好显示器应带有屏幕水平移动和垂直大小调节钮,其对比度、亮度调节范围要大。在挑选显示器时,要注意屏幕显示的字符是否清晰,尤其注意屏幕四周。有些 VGA 彩显的点间距为 0.31 或 0.39 毫米,其显示的字符就比较粗糙,用户购买时要注意。

在选购相应的显示卡时,应选用质量好、显示缓冲区容量在 512K 或以上的显示卡,如 8900C 型 TVGA 卡。目前市场上有许多新型 8900 和 9000 型 TVGA 卡,体积较小,但质量不稳定,而且与许多软件发生冲突。现在市场上有些简易双频卡,在进行显示模式转换时,需要调节卡上的短接帽,转换非常麻烦,建议用户不要选择此卡。较好的双频卡可用 DOS 命令来自动转换,用 MODE CO80 命令可转换为 CGA 模式,用 MODE MONO 命令转换为单显模式。

10. 磁盘驱动器

现在软盘驱动器分为 5 英寸 360K 低密,5 英寸

1. 2M 高密, 3 英寸 1.44M 高密。硬盘驱动器分 5 英寸、3 英寸和 2 英寸盘, 有 20M, 40M, 80M 等等。一般情况下, 40M 硬盘对于个人来说足够用了。购买时, 应选用存取速度在 18~28ms 的硬盘。用户可以选择一高一低双软驱, 双高双软驱, 或一高软驱一硬盘。磁盘驱动器安装要正确, 一般情况下, 高密驱动器在最上面, 低密驱动器在高密的下面, 硬盘在低密下面(也有在软驱的侧面), 再下面留有可扩充一个硬盘的空间, 其支架等配件应齐全, 否则将来再扩充磁盘驱动器会很麻烦。

11. 键盘

用户应选择 101/102 键, 外表美观, 手感好的键盘。

12. 外部设备

如果用户需要打印, 还要购买打印机。市场上的打印机分为 9 针, 24 针, 喷墨和激光打印机。家庭用应选择 24 针打印机。9 针打印机虽然便宜, 但许多汉字系统没有 9 针打印驱动程序, 就是有, 也是打印字型很少, 打印质量差, 没有背景等特殊打印功能。最好选购 24 针打印机, 低档机首选 TH3070 或 OKI8320, 中档

机选 Epson L2-1600K, 高档机选 AR3240。

在作图和使用 Windows 时, 常常使用鼠标器。用户在购买鼠标器时, 可选用光电式, 三按键, 标准串行接口的, 不要选购机械式的, 因为机械式很容易损坏。

13. 软盘

目前市场上软盘种类繁多, 在选购软盘时一定要买质量好的, 不要只图便宜, 否则极易磨损磁头, 数据也容易丢失。用户在购买时, 先检查软盘的商标是否清晰, 做工是否精细, 其次检查软盘磁表面是否光洁如镜, 外套最好厚些。现在市场上有不少 3M, MAXELL, JANUS 的假盘, 用户需要注意。存放软盘时, 要防水、防磁、防尘、防弯折、禁止触摸磁表面。

微机使用注意事项

用户在使用微机时, 注意不要在太热(35℃)或者过于潮湿的环境下操作, 避免阳光直射电脑。电源线最好有地线。磁头要经常清洗, 要常对微机进行清扫。要防尘、防静电。若天气潮湿, 要常开机。

以上仅代表笔者个人经验和看法, 请用户自行参考。用户可根据自己的具体情况选购一台满意的微机。

PCX 格式图像文件的应用技巧

北京理工大学机械工程系(100081) 周光辉

在计算机应用领域中, 图象技术已越来越得到人们广泛的重视和应用, 在许多应用课题中, 例如图形数据库, 图象处理等, 都需要利用图象的生成、显示、处理等技术。为了实现应用软件的兼容性, 所处理的图象必须是一种能被其它软件识别和处理的文件格式。我们用得最多, 最广泛的便是 Zsoft PCX 图象文件格式。Zsoft PCX 文件格式是一种标准图象文件格式, 能表示 2、16 和 256 色图象, 并且可以对图象进行压缩处理。许多扫描仪都支持这种文件格式, 很多图象编辑处理软件, 如 Windows, PhotoStyler 等均能处理这种图象格式文件。因此有必要了解 PCX 文件读、写、显示等基本技术。

一、PCX 文件格式

PCX 文件由首部和文件体两部分组成, 为了阐述清楚, 每一部分均给出原程序清单。

1. 头部信息。文件首部包括主要头部信息、彩色表和辅助文件信息三个部分, 为了读写处理方便, 我们将每一部分均定义为一个结构, 其中主要头部文件信息包括以下内容:

```
struct PCXFileHeader
{
    BYTE Header, Version, Encode; /* PCX 文件标志
    0x0A0501 */
    BYTE BitPerPixel; /* 每个像素的位数 */
    unsigned Xmin; /* 图象的 X 最小值 */

```

```
    unsigned Ymin; /* 图象的 Y 最小值 */
    unsigned Xmax; /* 图象的 X 最大值 */
    unsigned Ymax; /* 图象的 Y 最大值 */
    unsigned Hres; /* 图象水平方向(X)的分辨率 */
    unsigned Vres; /* 图象垂直方向(Y)的分辨率 */
};
```

BitPerPixel 决定了该文件是几色图象, 其值可以为 1、4 和 8, 分别对应单色、16 和 256 色图象。

彩色表中的每一颜色寄存器包括了三个字节, 分别表示其 RGB 值。对于 16 色 PCX 文件的彩色表, 共有 16 个颜色寄存器共 48 个字节, 它紧跟在主要头部信息后面; 对于 256 色 PCX 文件, 其彩色表有 768 个字节, 它位于 PCX 文件的最后。256 色 PCX 文件的彩色表的每一项的三个字节均需要右移 2 位才能得到真实值, 对于彩色表中每一颜色寄存器定义如下:

```
struct ColorRigster
{
    BYTE Red; /* 该颜色寄存器中 red 值 */
    BYTE Green; /* 该颜色寄存器中 green 值 */
    BYTE Blue; /* 该颜色寄存器中 blue 值 */
};
```

文件首部的最后一部分就是辅助文件信息, 它包括以下几个信息:

```
struct PcxInfo
{
    BYTE Vmode; /* (0) */
    BYTE NumOfPlanes; /* 位平面数 */
};
```



```

unsigned BytesPerLine; /* 一个扫描行所占字节数
*/
BYTE unused[60]; /* 保留 */
}

```

其中位平面数 NumOfPlanes 决定该 PCX 文件在屏幕显示时所需要的位平面数,对标准 VGA 显示器,单色和 256 色图象所用位平面数为 1,而 16 色图象所用位平面数为 4。

完整的文件首部可以用下面的结构表示:

```

struct PCX_File
{struct PCXFileHeader PCXHeader; /* 主文件头信息 */
struct ColorRegister Palette[16]; /* 16 个颜色寄存器 */
struct PCXInfo Info; /* 辅助文件信息 */
}

```

以上结构可完整地表示单色和 16 色 PCX 图象的文件首部信息,对于 256 色图象,还应包括一个扩展彩色调色板,它包括扩展彩色调色板标志和 256 个颜色寄存器,用结构表示如下:

```

struct ExtendedPalette;
{BYTE Extendedpalette; /* 扩展彩色调色板标志 (0x0C)
*/
struct ColorRegister Palette[256]; /* 256 个颜色寄存器 */
}

```

在 256 色 PCX 文件中,应从距该文件尾 769 个字节处读取该结构。

2. PCX 文件的图象压缩格式,PCX 文件采用运行限制长度(RLL)方法压缩图象数据,这种压缩方法分为两个步骤。第一步,将每一扫描行的象素转变为字节,它依 PCX 文件的颜色数的不同而不同,对于单色图象,一个象素只占用一位,即每字节可以表示相邻的 8 个象素点,例如扫描行的象素为 640 的黑白图象,每一行需占用 80 个字节;同样,对于 16 色图象,每个象素占 4 位,一个字节可表示 2 个象素点;而对 256 色图象,每个象素点占 8 位,即每字节仅表示一个象素点。

第二步,对图象中的每一扫描行独立地进行压缩,设某扫描段由连续的 N 个字节 Char 组成,若 Char > 0xC0,则输出 (0xC0+N),Char;若 Char ≤ 0xC0,且 N=1,则只输出 Char,否则输出 (0xC0+N),Char;如果 N>63,则需分多次压缩,该算法可表示为下面几个步骤:

1. if(Char>0xC0) {输出 (0xC0+N),Char;}
2. else {if(N==1. {输出 Char; }
else {输出 (0xC0+n),Char;}}
3. if(N>63) {输出 (0xC0+N),Char;N=N-63;goto 1;}

二、PCX 图象文件的应用

了解 PCX 图象文件的结构以后,就可以对它进行读取、存储、显示、生成和编辑等操作,为了操作方便,我们建议利用动态内存分配的方法,为非压缩的 PCX 图象开辟一个内存空间(即一个二维动态数组),其大小为图象宽度×图象高度,这样就可以很方便地进行显示和编辑(如绘制图形,添加汉字等)。

1. PCX 文件的读/写操作,读/写操作时,先利用

前面定义的文件头结构,对 PCX 文件头进行读/写操作,下面一段程序可以读取 PCX 文件头:

```

struct PCX_File PCXData;
FILE *PCXFile;
...
fread (&PCXData, sizeof (struct PCX_File), 1,
PCXFile);
...

```

根据 PCX 文件头所含的信息,我们可以知道 PCX 文件的颜色数目、水平方向和垂直方向的分辨率,以及彩色表等,对于 256 色 PCX 图象文件,还应读/写其扩展彩色表,下面一段程序可以读取 256 色 PCX 文件的扩展彩色表:

```

struct ExtendedPalette Color256Palette;
...
fseek(PCXFile, -769, SEEK-END);
fread(&Color256Palette, sizeof(struct ExtendedPalette),
1, PCXFile);
...

```

2. PCX 图象的显示及漫游,以往资料所介绍的 PCX 文件显示方法只是单个字节地输出到显示内存,这样严重影响了图象显示速度,这里,我们采用数据块移动的方法,每次可以将图象的一个扫描行移到显示内存,这样在瞬间便可完成显示一屏的操作,为图象的漫游提供了条件。

当图象的分辨率大于显示器的分辨率时,为了浏览整个图象,我们采用图象漫游技术,在后面的实例中可以看到,利用方向控制键可以非常平滑地漫游图象。

下面给出一个 256 色 PCX 图象文件的读取、显示及漫游的程序,该程序在 Turbo C 2.0 下用 COMPACT 模式编译通过,作者已完成 2/16/256 色 PCX 图象文件的读取、显示、处理和存储等操作,如读者感兴趣,可来函索取。

```

typedef unsigned char BYTE;
enum { MAXPALETTECOLORS = 16, MAX256-
PALETTECOLORS=256}COLORREGISTERS;
#include<stdio. h>
#include<stdlib. h>
#include<dos. h>
#include<bios. h>
#include" 256PCX. H" /该头文件定义了 PCX 文件的文件首
部结构,在上面说明中已给出 */
enum {TRUE=1, FALSE=0}BOOLEAN; /* 为节省版面,
下面常量均采用枚举。
enum {UP = 0x4800, DOWN = 0x5000, LEFT = 0x4B00,
RIGHT=0x4D00, ESC=0x011B}SCANKEYS;
enum {MAXCOLS=320,MAXROWS=200}RESOLUTION;
enum {VNORMALMODE = 0x03, V256COLORMODE =
0x13}VIDEOMODE;
#define PCX256ColorTag 0x0C
#define VideoMode(mode){\
union REGS regs;
regs. h. ah=0;regs. h. al=mode;
int86(0x10,&regs,&regs); }

```



```

unsigned char far * PCXBuf; /* Memory Buffer to
store PCX image. */
static FILE * PCXFile; /* File pointer to point to
PCX image file. */
static struct PCX_File PCXData; /* Completed PCX file
header. */
static struct ExtendedPalette Color256Palette;
static unsigned ImageWidth, ImageHeight, BytesToRead;
void ShowPCXScreen(int col, int row);
void BrowsePCXImage(void);
int main(int argc, char * * argv)
{
    unsigned RetValue;
    RetValue=DisplayPCXFile(argv[1]);
    MallocMemory(FALSE);
    return(RetValue);
}
int ExpandScanLine(FILE * PCXFile)
{
    register short CharRead; unsigned InPtr=0, RepCount,
    iLine;
    for(iLine=0; iLine<ImageHeight; iLine++, InPtr=0)
    {
        do
        {
            if((CharRead=getc(PCXFile))==EOF) return
            (FALSE);
            if((CharRead&0xc0)==0xc0)
            {
                RepCount=CharRead-0xc0;
                if((CharRead=getc(PCXFile))==EOF)
                return(FALSE);
                while(RepCount-->0) PCXBuf[iLine]
                [InPtr++]=CharRead;
            }
            else PCXBuf[iLine][InPtr++]=CharRead;
        } while(InPtr<BytesToRead);
    }
    return(TRUE);
}
int InstallPCXFilepalette(void)
{
    union REGS regs; unsigned Index;
    for(Index=0; Index<MAX256PALETTECOLORS; In-
    dex++)
    {
        Color256Palette.Palette[Index].Red >>=2;
        Color256Palette.Palette[Index].Green>>=2;
        Color256Palette.Palette[Index].Blue >>=2;
    }
    regs.h.ah=0x10; regs.h.al=0x12;
    regs.x.bx=0; regs.x.cx=MAX256-
    PALETTECOLORS;
    _ES=FP_SEG(&Color256Palette.Palette);
    regs.x.dx=FP_OFF(&Color256Palette.Palette);
    int86(0x10, &regs, &regs);
    return(TRUE);
}
int DisplayPCXFile(char * FileName)
{
    long FilePointer;
    if((PCXFile=fopen(FileName, "r+b"))==NULL)
        exit(-1);
    fread(&PCXData, sizeof(struct PCX_File), 1,

```

```

PCXFile);
    if(PCXData.PCXHeader.Version==0x03||PCXData.
    PCXHeader.BitPerPixel==8)
        { puts("Cannot support such file format."); return
        (FALSE); }
    FilePointer=ftell(PCXFile); fseek(PCXFile, -769,
    SEEK_END);
    fread(&Color256Palette, sizeof(struct Extendedpalette),
    1, PCXFile);
    if( Color256Palette.Extendedpalette!=
    PCX256ColorTag)
        { puts("Error tag in extended palette.");
        fclose(PCXFile); exit(-1); }
    fseek(PCXFile, FilePointer, SEEK_SET);
    ImageWidth=PCXData.PCXHeader.Xmax-PCXData.
    PCXHeader.Xmin+1;
    ImageHeight=PCXData.PCXHeader.Ymax-PCXData.
    PCXHeader.Ymin+1;
    if(! MallocMemory(TRUE)) /*为**PCXBuf动态
    分配内存空间*/
        { puts("Not enough memory."); fclose(PCXFile); re-
        turn(FALSE); }
    BytesToRead=PCXData.Info.NumOfPlanes * PCX-
    Data.Info.BytesPerLine;
    if(! ExpandScanLine(PCXFile))
        { puts("This PCX image file has been corrupted."); re-
        turn(FALES); }
    fclose(PCXFile);
    VideoMode(V256COLORMODE); /*设置256色模
    式,分辨率为320x200. */
    InstallPCXFilePalette(); /*安装256色PCX文
    件的调色板. */
    ShowPCXScreen(0,0); /*从PCX文件的左上角开
    始显示该图象. */
    BrowsePCXImage(); /*漫游PCX图象.
    VideoMode(VNORMALMODE); /*设文本模式*/
    return(TRUE);
}
int MallocMemory(int tag) /*动态分配二维数组**
PCXBuf */
{
    static int times=0; int iLine;
    if(tag==TRUE)
    {
        PCXBuf=(unsigned char far * *)
        malloc(ImageWidth * sizeof(unsigned char far));
        if(! PCXBuf) return(FALSE);
        for(iLine=0; iLine<ImageHeight; iLine++)
        {
            PCXBuf[iLine]=(unsigned char far *)
            malloc(ImageWidth * sizeof(unsigned char far));
            if(PCXBuf[iLine]==NULL) return(FALSE);
        }
        times=1;
    }
    else if(times){ /*释放已分配的内存 */
        for(iLine=0; iLine<ImageHeight; iLine++)
            free(PCXBuf[iLine]);
        free(PCXBuf); times=0;
    }
}

```



```

    }
    return(TRUE);
}
void ShowPCXScreen(int col,int row) /* 从 PCX 图象的左
上角(col,row)开始显示/
{
    int i,xc,yc, unsigned Segment,Offset;
    xc = (ImageWidth > MAXCOLS)? MAXCOLS: ImageWidth;
    yc=(ImageHeight>MAXROWS)? MAXROWS: ImageHeight;
    for(i=0;i<yc;i++)
    { Segment=FP_SEG(PCXBuf[i+row]+col);
      Offset=FP_OFF(PCXBuf[i+row]+col);
      movedata(Segment,Offset,0xA000,i * MAXCOLS,xc);
    }
}
void BrowsePCXImage(void) /* 漫游 PCX 图象,按 ESC 键
退出。 */
{ int col=0,row=0,step=5; unsigned skey=0;
  while(1)
  { if((skey=bioskey(0))==ESC) return;

```

```

    if (ImageHeight<MAXROWS && ImageWidth<MAXCOLS) continue;
    if(skey==UP||skey==DOWN||skey==LEFT||skey==RIGHT)
    { switch(skey)
      { case UP :row=(row>step)?(row-step):0;
        break;
        case DOWN; if (row < (int) (ImageHeight -
          MAXROWS-step))row+=step;
          else row = (ImageHeight > MAXROWS)?
            (ImageHeight-MAXROWS):0;break;
        case LEFT ;col=(col>step)?(col-step):0;break;
        case RIGHT ;if(col<(int) (ImageWidth -
          MAXCOLS-step)) col +=step;
          else col = (ImageWidth>MAXCOLS)? (ImageWidth-
            MAXCOLS):0;
          default :break;
        }
      ShowPCXScreen(col,row);
    }
}

```

一程序多报表的自动建立、修改与输出 ——dBASE 与 CCED 的结合使用

北京东燕郊有色金属管理干部学院基础学部 刘士杰

在企事业管理中,经常需要重复输出多个数据报表。若用 dBASE 完成这一任务,往往需要编制较繁琐的程序,并要花费较多时间上机调试。但在程序调好后,可以编制调用程序,方便地调用它们,实现多报表的选择输出。而中文字表编辑软件 CCED 系统提供了直接将 dBASE(或 FoxBASE)数据库数据自动生成报表的方法。它不需要编制程序,只要运行系统提供的辅助程序,就能自动生成报表,而且只须通过修改样本表格文件或参数表文件,就会自动按修改要求直接转换成报表输出。但它不能实现一报表的一并建立、修改与输出和多报表的选择输出。本文介绍的方法,将 dBASE 与 CCED 结合起来,取其长,补其短,使多报表的建立、修改与输出,既简单,又方便。

在 dBASE 下调用 CCED 辅助程序,自动建立、修改和选择输出 dBASE 数据报表的程序如下:

```

SET TALK OFF
CLEAR ALL
DO WHILE .T.
CLEAR
? "====数据报表的建立、修改与输出系统===="
? "      1——自动建立数据报表      "
? "      2——修改数据                "

```

```

? "      3——修改报表参数          "
? "      4——修改样本表格          "
? "      5——屏幕显示报表          "
? "      6——打印数据报表          "
? "      0——退出系统              "
? "===== "
ACCEPT"请选择相应的功能序号(0~6):" TO XH
IF XH="0"
EXIT
ELSE
ACCEPT"请输入要输出报表的代号:" TO DH
DO CASE
CASE XH="1"
WAIT"按任意键后,回答屏幕提问,建立报表"
CASE XH="2"
ACCEPT"请输入要修改的数据库文件名:" TO MC
USE &MC
BROWSE
USE
CASE XH="3"
RUN CCED CS&DH
CASE XH="4"
RUN CCED YB&DH
CASE XH="5"
TYPE BB&DH

```



```

CASE XH="6"
CLEAR
WAIT"请准备好打印机,按任意键打印"
TYPE BB&DH TO PRINT
OTHERWISE
WAIT"序号选错,按任意键后重选"
LOOP
ENDCASE
RUN DBST CS&DH
ENDIF
ENDDO
SET TALK ON
RETURN

```

本程序提供六种基本功能:

1. 建立报表。调用 CCED 建立报表的辅助程序, 回答屏幕提问, 确定报表参数后自动产生报表。
2. 修改数据库内容。若已建立报表的参数与形式不变, 仅输出数据发生变化, 可以选择此项功能。
3. 修改参数表文件。报表形式不变, 只修改有关参数。参数包括: 与报表有关文件的文件名、要输出报表的起始和终止记录、打印的起始页码、报表输出日期等。选此功能后, 系统先调出报表相应的参数表文件供修改, 修改后要将其存盘。
4. 修改样本表格文件。这是报表形式发生变化的修改。其中包括修改标题和栏目名称、数据在表中摆放位置、表头形式、尾注和页号位置以及增删某些栏目、尾注和页号等。修改时系统将给出报表的样本表格供

修改, 修改后应将其存盘。

5. 将已建立或修改后的数据报表在屏幕上显示。
6. 将已建立或修改的数据报表打印输出。

报表建立或修改后, 均可选第 5 或第 6 项功能输出报表。若输出前已选第 2、3、4 项功能作过修改, 输出时系统会按修改后的数据、参数或样本自动产生新报表输出。

用户选择系统功能后, 需通过对话方式给出报表的代号, 从而确定当前要处理的报表。一个报表处理后, 程序采用循环方式可处理下一报表, 从而实现多报表的建立、修改和输出。在使用该程序时, 要求将要建立的各报表以数字、字母或它们的组合名称作为代号。在建立新报表, 确定报表参数时, 用户必须将报表的样本表格文件名设置为: "YB+代号" 的形式, 将报表输出文件名设置为: "BB+代号" 的形式。如将某报表代号设置为 A, 则其样本表格文件名应设置为 "YBA", 其输出文件名应为: "BBA"。报表代号设定后, 应记住它们, 以便以后再次使用。

以上只给出多报表建立、修改和输出的最基本程序。如果用户希望更加完善, 可以根据自己输出报表的情况添加部分语句或程序段。为方便操作, 使用本程序前应将 dBASE III 和 CCED(3.0 以上版本) 系统拷贝到硬盘 CCED 子目录下。此外, CCED 中调用 dBASE 数据库数据产生报表的方法及参数表文件和样本文件的修改, 请读者参阅有关 CCED 的使用手册或教程。

西文文本方式下怎样利用 VGA 及其兼容显示器显示汉字

湖北荆门供电局(434500) 樊启柏

本文提出的问题对于在文本方式下运行的西文软件的汉化有重要的实用价值。但是在 VGA 9×16 字符点阵的文本方式下, 显示的汉字中间断裂, 有待改进。我们欢迎来稿对这方面的问题, 从原理到具体实现的技术作更多的讨论

编辑部

众所周知, 除国产长城、浪潮等系列机外, 其它 PC 系列机显示汉字均在图形方式下进行, 这就使得很多在文本方式下运行的优秀软件不能采用汉字做提示, 给一些不熟悉英文的使用者带来不方便, 在不改变显示方式的情况下, 一些西文文本软件即使经过艰苦的汉化, 也只能在国产机上运行。为了解决这个问题, 笔者用 C 语言探索了一种能在 VGA(及兼容) 文本方式下显示汉字的方法, 具体叙述如下。

一、显示汉字的基本原理

为什么在 VGA 文本方式下能显示汉字呢? 这是因为 VGA 文本方式的扫描线最大数是 400, 它的字符发生器可随机装入字符表。机内装有标准的 ASCII 码, 字符表分 8×8 和 8×16 点阵(EGA 含 8×14 点

阵), 开机预置标准模式是 25 行和 8×16 点阵字符表。考虑到目前流行的汉字系统其基本字库的汉字点阵数是 16×16 点, 因此, 我们可以将一个汉字分为两个标准的 ASCII 码来显示。那么, 怎样才能将要显示的汉字装入字符发生器呢? VGA 字符发生器的显示缓冲区是具有 4 个页面的随机存储器(RAM), 在文本方式下, 它的字符表是装在页面 2 中, 可装入 8 种字符表映象(每种 256 个字符, 16 个字节的字形码), 可同时激活两个字符表, 这样就可以将汉字点阵装入页面 2 的某个字符表中来显示汉字。尽管 VGA 显示缓冲区的 4 个页面不是随便可以装入的, 它有专门的寄存器来控制, 否则显示就会乱套, 并且显示控制寄存器的修改非常复杂, 稍不慎就会损坏显示器, 但好在 VGA 的 BIOS 显示中断(10H)有专门的功能来管理字符发生器, 使得汉字的装入和显示非常简单。本文采用了显示中断(10H)的 11H 子功能来装入所要显示的汉字字模, 具体调用中断的方法如下。

二、显示汉字的具体方法

首先从字库中(如 2.13 系统中的 HZK16)读出所要显示的 16 * 16 点阵的汉字字模,装入一个事先给定的内存空间中,并且把每个汉字分成左右两个半字形重新装配成 8 * 16 点阵的半个字形,然后调用中断 10H 的子功能 11H,将上述 8 * 16 点阵的字形装入 VGA 字符发生器,如果将汉字放入 ASCII 码的后 128 个(前 128 个留作标准的 ASCII 码符用),这样每次可装入 64 个汉字,调用中断时各寄存器的初始化参数的意义是;

AH=0X11H 装入字符发生器子功能
AL=0X00H 装入用户定义的字符集
BH=0X10H 每个字形的点阵字节数
BL=(0-7) 要装入字库使用的软字库号
CX=0X80H 每次要装入字符的个数
ES:BP=要装入的字库的源地址指针。

DX = 0X80H 要装入第一个字符的偏移值(ASCII 码)。编程方法见下面的程序清单。

三、显示汉字的程序清单

本文在这里给出的是一个在 VGA 文本方式下显示汉字的弹出式菜单的程序清单,按 F10 可使菜单弹出,按 →← 键切换子菜单,按 ↓ ↑ 键选子菜单中的项目,按 ESC 键清除子菜单,回到编辑状态。

```
#include "dos.h"
#include "bios.h"
#include "stdio.h"
#include "conio.h"
union REGS inregs, outregs;
char zf3[128][16];
/* —— 画屏幕边框 —— */
void bowin(int x1,int y1,int x2,int y2,int b,int dcor,int bcor,int zcor)
{register int il;
int zh[2][6]={{0xda,0xc4,0xbf,0xb3,0xc0,0xc9}
{0xc9,0xcd,0xbb,0xba,0xc8,0xbc}};
textattr(zcor|16*bcor);
window(1,1,80,25);
gotoxy(x1,y1);
putch(zh[b-1][0]);
for(il=x1+1;il<x2;il++){
putch(zh[b-1][1]);
putch(zh[b-1][2]);
for(il=y1+1;il<y2;il++){
gotoxy(x1,il);
putch(zh[b-1][3]);
gotoxy(x2,il);
putch(zh[b-1][3]);}
gotoxy(x1,y2);
putch(zh[b-1][4]);
for(il=x1+1;il<x2;il++){
putch(zh[b-1][1]);
putch(zh[b-1][5]);
if(dcor!=0{window(x1+1,y1+2,x2-1,y2-1);
clrscr();}}
/* —— 定窗口,上背景 —— */
```

```
void window0(int x1,int y1,int x2,int y2,int bcor)
{ textbackground(bcor);
window(x1,y1,x2,y2);
clrscr();
window(1,1,80,25);}
/* —— 光标形状转换 —— */
```

```
void bg(int rr)
{ inregs.h.ah=0x01;
if(rr==0)inregs.h.ch=0x20;
else{inregs.h.ch=0x01;
inregs.h.cl=0x07;}
int86(0x10,&inregs,&outregs);}
/* —— 装汉字到字符发生器 —— */
```

```
void zkfs(char far *zff)
{ ES=FP_SEG(zff)
DX=0x80;
CX=0x80;
BH=0x10;
BL=0;
BP=FP_OFF(zff);
AH=0x11;
AL=0;
geninterrupt(0x10);}
/* —— 屏幕提示 —— */
```

```
void tsh()
{ bowin(1,1,80,24,1,12,0,12);
textattr(15|16);
gotoxy(12,25);
printf("ALT+x--Quit");
gotoxy(32,25);
printf("F10--Main Menu");
gotoxy(52,25);
printf("ESC--Cls Menu");
textbackground(0);
gotoxy(2,2);}
/* —— 读入事先编辑好的汉字菜单字模,并重新组合 —— */
```

```
void zfc1()
{ FILE *fp;
register i,j,k;
char zf[64][16],zf2[64][16],zf[64][32];
fp=fopen("hzcd.hzm","rb");
if(fp==NULL){gotoxy(1,2);
printf("HZCD.HZM File not found");
exit(1);}
for(i=0;i<64;i++){
for(j=0;j<32;j++){
fread(&zf[i][j],1,1,fp);
fclose(fp);
for(k=0;k<64;k++){
for(i=0;i<16;i++){
zf1[k][i]=zf[k][i*2];
zf2[k][i]=zf[k][i*2+1];}
for(k=0;k<64;k++){
for(i=0;i<16;i++){zf3[2*k][i]=zf1[k][i];
for(i=0;i<16;i++){zf3[2*k+1][i]=zf2[k][i];}}
/* —— 恢复原来字符表 —— */
```



```

void hfzf()
{
    _AH=0x11;
    _AL=4;
    _BL=0;
    geninterrupt(0x10);
    /* ----- 弹出式菜单 ----- */
    void cd()
    {
        char * buffer1[300],buffer2[300];
        register i=0,il,j=0,key=0,k,key1=0;
        gb(0);zkfs(*zf3);
        window(1,1,80,25,0);
        window(5,1,76,1,4);
        gettext(5,2,16,9,buffer1);
        window(5,2,16,9,2);
        window(6,3,15,8,1);
        gettext(5,2,16,9,buffer2);
        textattr(14|64);
        for (il=0;il<6;il++){
            gotoxy(7+12 * il,1);
            for(k=0;k<8;k++)putch(128+il * 8+k);}
        textattr(11|16);
        for(il=0;il<6,il++){
            gotoxy(7,3+il);
            for(k=0;k<8;k++)putch(128+il * 8+k);}
        gotoxy(7,1);
        for(k=0;k<8;k++)putch(128+il * 8+k);
        textattr(14|64);
        gotoxy(7,3);
        for(k=0;k<8;k++)putch(128+k);
        while(key!=283){gotoxy(7+12 * j,3+i);
            key1=key=biostkey(0);
            key=(key&0xff!=0)?key:key>>8;
            textattr(11|16);
            for(k=0;k<8;k++)putch(128+i * 8+k+j * 16)
            if(key==72)i=(i==0)?5:i-1;
            if(key==80)i=(i==5)?0:i+1;
            if(key==77||key==75){textattr(14|64);
                gotoxy(7+12 * j,1);
                for(k=0;k<8;k++)putch(128+j * 16+k);
                puttext(5+12 * j,2,16+12 * j,9,buffer1);
                if(key==77)j=(j==5)?0:j+1;
                if(key==75)j=(j==0)?5:j-1;
                puttext(5+12 * j,2,16+12 * j,9,buffer2);
                textattr(11|16);
                gotoxy(7+12 * j,1);
                for(k=0;k<8;k++)putch(128+j * 16+k);
                for(il=0;il<6;il++){
                    gotoxy(7+j * 12,3+il);
                    for(k=0;k<8;k++)putch(128+il * 8+k+j *
                    16);}}
                gotoxy(7+12 * j,3+i);
                textattr(14|64);
                for(k=0;k<8;k++)putch(128+i * 8+k+j *
                16); }
                puttext(5+12 * 1,2,16+12 * j,9,buffer1);
                window(1,1,80,24,0);

```

```

        window(1,25,80,25,1);
        hfzf(); tsh();}
/* ----- 主程序是一个简单的编辑程序,通过 F10和 ESC 键\
切换弹出式菜单和编辑功能 ----- */
main()
{
    register key=0,i=2,j=2;
    int ch,zf[240][100];
    textmode(3);zfc1();
    tsh(0);
    while((key!=45){ gb(1);
        key=biostkey(0);
        if(key&0xff==0){
            key=key>>8;
            if(key==68)cd();
            if(key==83){i+=1;cprintr(" ");}
            if(key==73){i=2;j=2;}
            if(key==81){j=22;i=78;}
            if(key==79){i=78;}
            if(key==71){i=2;}
            if (key==75){if(i>2)i=i-1;else{i=78;j-=1;}}
            if(key==77){if(i<78)i=i+1;else{i=2;j+=1}}
            if(key==80){if(j<22)j=j+1;else j=2;}
            if(key==72){if(j>2)j=j-1;else j=22;}
            gotoxy(i,j);}
        else{ch=key&0x00ff;
            zf[i][j]=ch
            if(ch==13&&j<22){i=2;gotoxy(i,j+1);j+=1;}
            if(ch>30)cprintf("%c",ch);
            if(i>78){gotoxy(2,j+1);i=2;j+=1;}
            else i+=1;}}
    hfzf();}

```

本程序在 Turbo C 2.0 系统下编译运行通过,显示效果很好。笔者认为:采用该方式显示汉字,可使得中西文软件在最大限度内兼容,对西文软件的汉化以及文本方式下运行的软件开发汉字提示是一个突破。程序运行环境:在 DOS 3,30及以上版本,备有 VGA (及兼容)显示器的 PC 机上均可运行。

卫佳防病毒卡

卫佳防病毒卡是针对病毒共性而研制的一种广谱防病毒工具,能够有效检测各类已知和未知病毒,安全带毒运行,抑制病毒传播发作,并具有自动清除硬盘引导型病毒、中文信息提示等独特功能。适用于各档次 PC 系列及兼容机。在各种中文系统、网络系统、Windows 等环境下均可有效工作。该卡获全军科技进步二等奖。零售价:420元;批发价格面议。

地址:北京万寿路173信箱(中国软件行业协会软件出版专业委员会技术培训部)

邮编:100036 联系人:徐 轲

电话:8221677 传 真:8214062

地址:北京海淀路60号(北京星科技术公司)

邮编:100086 联系人:王 靖

电话:2544807 传 真:2563093

直接读/写 VGA 的图形像素

铁道部宝鸡工程机械厂(721003) 王绍鑫

众所周知,读写像素是计算机图形处理中十分重要和常用的功能之一。BIOS 提供的读写像素功能因为要适用于各种图形显示模式,所以其处理像素的速度非常慢。为了提高工作效率,必须结合显示适配器的特点,采用直接读写显示存储器的方法来实现。

VGA 是一种面向页面的显示设备。它含有 256K 字节的显示存储器,分成 4 个独立的、具有相同地址空间的彩色页面。每个页面用来控制一种彩色成份,一个图形像素在每个页面中占有一位(bit)位置,同一时间在屏幕上最多可以显示 $2^4=16$ 种颜色。VGA 具有功能强大的可编程的寄存器集,分为 CRT 控制器、定序器、图形控制器和属性控制器等几部分,可以通过 I/O 端口访问。它的所有图形功能都是通过寄存器来控制的。在 VGA 的内部有一个 32 位的数据寄存器,称为数据锁存器。对显示存储器的一次读操作,可将 4 个页面上同一地址的 4 个字节读入此 32 位数据锁存器中,但 CPU 只能读出由读页面选择寄存器选中的页面上的 1 个字节。对显示存储器的一次写操作,可以同时写 1 个或所有页面的相同地址,通过定序器的彩色页面写允许寄存器来控制写入的页面,由图形控制器的位屏蔽寄存器控制写入的数据是来自数据锁存器,还是来自 CPU。本文以 VGA 显示模式 12H(640×480 16色)为例,介绍直接读写 VGA 图形像素的方法。

直接读写图形像素必须首先确定包含该像素的字节在显示存储器内的地址,以及该像素在字节中的位置。对于显示模式 12H,其分辨率为水平方向 640 个像素,垂直方向 480 个像素,显示存储器被分成 4 个均从 A000:0000H 开始的页面,显示存储器的一个字节对应屏幕上的 8 个像素,屏幕上的一个像素(x,y)与显示存储器的映射关系为:字节偏移地址 = $80 * y + x / 8$,像素在字节中位置 = $7 - (x \text{ mod } 8)$ 。函数 offset_address() 和 pixel_position() 分别返回像素(x,y)在显示存储区中的字节偏移地址和在字节中的位置。

VGA 提供了 3 种写入模式和 2 种读出模式。写像素操作常用写模式 0 和写模式 2,读像素操作常用读模式 0。写模式 0 是 VGA 的缺省模式,它允许 CPU 直接写显示存储器,同时还允许进行设置/重置、移位、位屏蔽及逻辑运算等操作。函数 write_pixel_plane() 就是运用写模式 0 的直接写存储器功能来实现写像素的。这种方法使用了定序器的写页面允许寄存器和图形控制器的位屏蔽寄存器,写入像素时分两步:先清全部页面;再用像素的颜色选择允许写入的页面,并将像素位置 1。函数 read_pixel() 返回指定像素的颜色值,函数使用了图形控制器的读页面选择寄存器控制每次读

出的页面,利用循环将每次读出的字节数据进行屏蔽、组合操作,最后得到像素点的 4 位颜色值。

以上介绍的直接读/写 VGA 图形像素的方法以 C 函数的形式给出。为提高读写速度,函数使用了嵌入汇编。最后给出了一个实例程序,调用这两个函数在屏幕上画出红、蓝两个矩形。

```
/* PIXEL.C 直接读/写 VGA 图形像素演示程序,适用于中西文状态. */
#include <dos.h>
main()
{
    register i,j;
    int point,current_mode,id_mode=0;
    union REGS r;
    r.h.ah=0x0f;int86(0x10,&r,&r);current_mode=r.h.ah;
    if(current_mode!=12){
        r.x.ax=0x0012;int86(0x10,&r,&r);id_mode=1;
    }
    /* 画蓝色矩形. */
    for(i=0;i<480;i++)
        for(j=60;j<260;j++)
            write_pixel_plane(j,i,1);
    /* 读蓝色矩形,画红色矩形 */
    for(i=0;i<480;i++)
        for(j=60;j<260;j++){
            point=read_pixel(j,i);
            write_pixel_plane(320+j,i,point+3);
        }
    bioskey(0);
    if(id_mode){
        r.h.ah=0x00;r.h.al=current_mode;int86(0x10,&r,&r);
    }
    /* 返回像素(x,y)在显示存储器中字节偏移地址. */
    offset_address(int x,int y)
    {
        asm mov ax,y
        asm mov bx,80
        asm mul bx
        asm mov bx,x
        asm mov cl,3
        asm shr bx,cl
        asm add ax,bx
    }
    /* 返回像素(x,y)在字节中位置 */
    pixel_position(int x)
    {
        asm mov cl,x
```



```

asm and cl,07h
asm mov ax,80h
asm shr al,cl
{
/* 写像素,使用写模式0的直接写显示存储器功能 */
write_pixel_plane(int x,int y,int color)
{
int offset,mask_code;
asm mov ax,0a000h
asm mov es,ax
offset=get_address(x,y);
mask_code=get_mask_code(x);
asm mov di,offset
asm mov bl,mask_code
asm mov dx,03ceh
asm mov al,08h
asm out dx,al
asm inc dx
asm mov al,bl
asm out dx,al
asm mov dx,03c4h
asm mov al,02h
asm out dx,al
asm inc dx
asm mov al,0fh
asm out dx,al
asm mov al,es:[di]
asm xor al,al
asm mov es:[di],al
asm mov al,color
asm and al,0fh
asm out dx,al
asm mov al,0ffh
asm mov es:[di],al
asm mov dx,03ceh
asm mov al,08h
asm out dx,al
asm inc dx
asm mov al,0ffh

```

```

asm out dx,al
asm mov dx,03c4h
asm mov al,02h
asm out dx,al
asm inc dx
asm mov al,0fh
asm out dx,al
}
/* 读像素,使用读模式0 */
read_pixel(int x,int y)
{
int offset,mask_code;
asm mov ax,0a000h
asm mov es,ax
offset=get_address(x,y);
mask_code=get_mask_code(x);
asm mov di,offset
asm mov bx,mask_code
asm mov dx,03ceh
asm mov al,04h
asm out dx,al
asm inc dx
asm mov cx,04h
read_plane_loop:
asm shi bh,1
asm mov al,cl
asm dec al
asm out dx,al
asm mov al,es:[di]
asm and al,bl
asm jz zero_value
asm or bh,1
zero_value:
asm loop read_plane_loop
asm mov al,bh
asm xor ah,ah
}

```

2. 13F 系统在单显环境下使用所遇几个问题的解决办法

南京两浦铁路中学(210031) 张彦超

一台 AT286 微机在单色高分辨率环境下,应用 2.13F 汉字系统,针对使用中发现的几个问题,笔者通过分析修改系统程序,使问题得到解决,现介绍如下:

1. 按使用说明,CTRL+F7 键是一组反复键,用于中西文方式的切换。使用中发现,按压这组键,仅能由中文方式进入西文方式,而不能重新返回中文方式。利用 DEBUG 工具,反汇编键盘管理模块 t2.com,找到有关 CTRL+F7 键的处理程序,其中部分如下:

```

××××:9A76 MOV AL,(0049);取显示方式号
××××:9A79 CMP AL,03;是字符方式号吗?
××××:9A7B JBE 9A81;是的,转
××××:9A7D MOV AL,03;否则,设置西文字符方式
××××:9A7F JMP 9A83;转去置显示方式
××××:9A81 MOV AL,06;设置中文图形方式号
××××:9A83 XOR AH,AH;
××××:9A85 INT 10;置显示方式

```

在单色高分辨率环境下,西文字符方式号为 07H,

巧用 INT 16

机械部第8设计研究院机电所(410007)

李珊红

上段程序不能由07H方式改置为06H方式,为此,可将程序中的03H均改为06H,06H改为07H,即可由CTRL+F7键自由的在中西文两种方式间切换了。

2. 在启动2.13F系统的过程中,如调用25行显示模块t25.com,可进入25行中文显示方式,但在前述程序改动后,按压CTRL+F7组合键后屏幕仍陷入混乱,不能正常显示,进一步跟踪分析,发现在t25.com模块中,有如下一段执行显示器初始化的程序:

```

××××:18D1  CMP  AL,04
××××:18D3  JB   18E0
××××:18D5  CMP  AL,07
××××:18D7  JZ   18E0
××××:18D9  MOV  AL,06
××××:18DB  MOV  BL,02
××××:19DD  JMP  18E4
××××:18DF  NOP
××××:18E0  MOV  AL,03
××××:18E2  MOV  BL,20
××××:18E4  MOV  DX,
                                03B4

```

:

这段程序使显示方式的设置发生错误,只须将第4行的JZ 18E0改为JZ 18E4,短路方式号的判断过程(因前面已进行过判断),即可使问题得到解决。

3. CTRL+F5键是清理内存功能键,按压这一组功能键后,屏幕在提示行显示:“清理内存1—退出汉字,5—驱动程序,9—外加模块请选择;”字样,键入“1”退出中文状态后发现,光标一旦进入最底下一行,屏幕显示内容就不再上滚,所有显示内容均在最底下一行显示,屏幕发生所谓“死锁”现象。分析显示模块,发现在模块自举过程中,将0040:998C和0040:9993两个单元的内容由19H,18H,分别改写为18H,17H,随之分析键盘管理模块t2.com有关CTRL+F5键的处理程序,发现在退出汉字状态时,并未恢复这两个字节的内容。笔者对处理CTRL+F5键的这段程序稍做改动,并在该模块空闲单元中增加一小段程序,使屏幕死锁问题得到解决,具体做法如下:

```

A>DEBUG T2.COM
-A A455 ↓
××××:A455  CALL A7F0 ↓
××××:A458  NOP ↓
××××:A459  NOP ↓
-A A7F0 ↓
××××:A7F0  MOV AX,357F ↓
××××:A7F3  INT 21 ↓
××××:A7F5  MOV AX,0040 ↓
××××:A7F8  MOV DS,AX ↓
××××:A7FA  MOV BYTE PTR[998C],19 ↓
××××:A7FF  MOV BYTR PTR[9993],18 ↓
××××:A804  RET ↓
××××:A805 ↓
-W ↓
-Q ↓

```

为了开发出有专业水平的软件并使其具有良好的用户界面,编程人员常常希望用到多种功能键,特别是带箭头的键以及像Home、End、Pgup、Pgdn等键。然而,这些功能键的使用令软件开发者颇费一番脑筋。因为这些功能键包括了两部分,即扫描码和扩展码。已有资料介绍过用BASIC或C语言程序来读这些键码,但这些程序要么很麻烦,要么就是不甚了了,不能满足使用者的需要。笔者利用INT 16中断所编的以下这两个小程序,可以方便地读出所有键码。使用者可将它嵌入自己的程序中,或作为一子模块,在生成OBJ文件后供高级语言调用。(附注:所用语言为Microsoft公司的MASM 5.0和Quick BASIC 4.5,C语言的调用从略,有兴趣者可来信索取,地址:湖南长沙韶山路90号

邮编:410007 机械部第八设计研究院机电所)

```

程序1: dosseg
        .model medium
        .stack 100h
        public readkey
        .code
readkeyproc far
        push bp
        mov bp,sp
        sti
        mov ah,0h
        int 16h
        mov di,[bp]+8
        mov [di],ah
        mov di,[bp]+6
        mov [di],al
        cli
        pop bp
        ret 4
readkeyendp
        end
程序2:DEFINT A-Z
        CLS
        DO
        CALL readkey(key1,key2)
        IF key2=0 THEN
        ky=key1
        ELSE
        dk=key2
        END IF
        PRINT ky,显示功能键值(十进制)
        LOOP
        END

```


安全可靠地扩充功能键高级编程技术

山东济宁市人民银行(272119) 侯培玉

早期83键键盘已明显地不能满足编程需要,而出现了101键键盘。这又很快被软件开发者所占满,不光F1~F10及其组合,连左右CTRL都被赋予不同的功能。这不可避免地会出现用键覆盖和功能覆盖的问题。笔者注意到,惟有F11和F12尚较空闲,原因是不少人掌握这两个键的定义方法。这里把本人使用这两个键的方法结合示例介绍给在高深层从事开发工作的高级编程人员。

DOS操作系统处理键盘输入的过程是:始终扫描键盘端口,当有按键动作发生时,即行存入键盘数据缓冲区,并把该键产生的ASCII码和扫描码(如果是功能键或组合键)返回给中断程序,再由调用中断程序的编程语言或应用软件解释后去完成一定的功能,遗憾的是多数常用的编程语言尚不能接受并处理近几年才加装的F11和F12及其组合键,使得哪怕是能用多种高级语言编程的具有颇高威望的高手也望而兴叹。

为了雅俗共赏,本文所附程序用BASIC这一几乎人人皆知的源程序格式深入浅出地展示接受并处理扩充功能键的技巧。

在BASIC的OK状态下,照本键入所示的源程序,校对无误后按RUN执行即可产生一个名为READKEY.COM的可执行程序,避免了由汇编源程序格式在编辑、编译、连接、转换、调试等复杂的各环节带来的各种错误和对程序员太高的要求。

在DOS提示符下键入READKEY就等待键盘输入,有按键动作时立刻显示是否按了F11或F12及其组合键。这仅是一个示例程序,没提供根据每个不同组合键去执行不同功能的具体程序段。为了便于读者能利用这种组合键,特把其键值(十六进制)列表如下:

	F11键	F12键
本键	8500H	8600H
SHIFT	8700H	8800H
CTRL	8900H	8A00H
ALT	8B00H	8C00H

```

10 F$="READKEY.COM":L=18:Y=11:YL=16:
   SUM=168
20 PRINT"命令文件生成程序:正确性检查..":Z=0:FOR
   B=1 TO L+1:IF B=L+1 THEN YL=Y
30 TTL=0:FOR C=1 TO YL:READ A$:TTL=TTL+
   VAL(A$):NEXT:READ S
40 IF S(<)(TTL MOD 256) THEN PRINT"错行先是:";B+
   100;"核好再运行!":END

```

```

50 Z=Z+TTL:Z=Z MOD 256:NEXT:IF Z(>)SUM
   THEN PRINT"可能是数据行数有误":END
60 RESTORE:PRINT"祝贺咱俩,数据正确。";:OPEN F$
   AS #1 LEN=1:FIELD #1,1 AS D$
70 YL=16:FOR B=1 TO L+1:IF B=L+1 THEN YL=
   Y
80 FOR C=1 TO YL:READ A$:LSET D$=CHR
   $(VAL(A$)):PUT #1:NEXT:READ A$:NEXT
90 CLOSE:PRINT F$+"已经生成,运行看看结果!":
   END
100 REM 以下是附带有校验码的数据,请认真键入...
101 DATA 233,248,0,144,215,247,213,223,176,180,
   58,40,50,55,50,49,133
102 DATA 49,57,41,201,189,182,171,202,161,188,
   195,196,254,202,208,200,136
103 DATA 203,195,241,210,248,208,208,186,238,197,
   224,211,241,211,218,49,216
104 DATA 57,56,50,196,234,177,207,210,181,211,
   218,214,208,185,250,191,29
105 DATA 198,188,188,180,243,209,167,44,210,187,
   214,177,200,231,215,237,16
106 DATA 200,231,179,213,181,216,212,218,68,79,
   83,186,205,66,73,79,185
107 DATA 83,178,227,204,189,203,247,177,224,179,
   204,44,209,208,211,208,179
108 DATA 202,181,211,195,200,237,188,254,185,164,
   190,223,207,228,44,176,13
109 DATA 252,192,168,211,178,197,204,200,237,202,
   189,208,180,177,163,187,73
110 DATA 164,181,200,202,253,176,217,184,246,44,
   203,181,195,247,202,233,56
111 DATA 186,175,203,247,188,180,188,196,44,184,
   252,187,182,211,173,184,164
112 DATA 223,202,214,196,250,214,184,189,204,44,
   187,242,189,225,206,170,67
113 DATA 200,237,211,209,46,36,26,13,10,176,180,
   193,203,70,49,49,116
114 DATA 187,242,70,49,50,187,242,198,228,215,
   233,186,207,46,36,13,85
115 DATA 10 176,180,181,196,178,187,202,199,70,49,
   49,187,242,70,49,177
116 DATA 50,187,242,198,228,215,233,186,207,46,
   36,180,9,186,4,160
117 DATA 205,33,232,33,0,80,91,232,28,0,138,248,
   129,251,0,133,41
118 DATA 114,12,129,251,0,140,119,6,186,199,1,
   235,4,144,186,223,157
119 DATA 1,180,9,205,33,195,180,1,205,33,195,213

```

计算机病毒的检测、消除和预防

中国地质大学(北京) 苏民生

近年来计算机病毒感染的范围越来越大,造成的影响越来越大,新病毒和老病毒变化的数目急剧增多,使得用机人员不能不关心病毒的检测、消除和预防工作。当初不大在意的人们,身受其害之后,大多重视起来了。但是,仍有不少微机是“带病毒工作”的,仍有不少软件是通过带病毒的软盘或带病毒的文件传递的,装有防病毒卡的系统仍然会感染新病毒。这些情况表明,为了安全使用计算机,防止和排除病毒的破坏,还有大量工作要做,这是一项经常性的工作。什么是计算机病毒?它有哪些危害?如何有效地检测和消除病毒?有没有预防病毒的办法?对于这些问题,都需要有清楚的认识。本文就我们遇到的有关问题,谈谈一些认识和体会,主要是讨论 IBM PC 机及兼容机在 DOS 系统下出现的病毒问题。

一、什么是计算机病毒?

计算机病毒是一种特殊的程序。由病毒程序引起的问题,属于软件故障,而不是硬件故障,诸如系统不能正常引导、程序不能正确执行、文件莫名其妙地丢失等等现象。

病毒程序区别于通常程序,有以下特点:

病毒程序的存在是隐蔽的。它总是悄悄地附着在磁盘系统区或文件中。寄生于文件中的病毒是文件型病毒。其中病毒程序附在原来文件之前或之后的,称为文件外壳型病毒,如以色列病毒(黑色星期五)等。另一种文件型病毒为嵌入型,其病毒程序嵌入到原来文件之中,在微机病毒中尚未见到。病毒程序侵入磁盘系统区的称为系统型病毒,其中较常见的占据引导区的病毒,称为引导区病毒,如大麻病毒、2708病毒等。带病毒磁盘中原来主引导区或引导区的内容,或者被移到磁盘其它特定位置,或者被病毒程序归并。此外,还有一些既寄生于文件中又侵占系统区的病毒,如 Flip 病毒等,属于混合型。

病毒程序在一定条件下隐蔽地进入系统。当使用带有系统病毒的磁盘引导系统时,病毒程序先进入内存并放在常驻区,然后才引导系统,这时系统即带有该系统病毒。当运行带有病毒的程序文件(COM 文件或 EXE 文件,有时包括覆盖文件)时,先执行病毒程序,然后才执行该文件的原来程序。执行病毒程序的结果,有的病毒是将自身程序常驻内存,使系统从此处于该文件病毒环境,有的病毒则不常驻内存,只在当时执行时起传染或破坏,执行完毕之后病毒不再留在系统中。

病毒程序执行的是非授权(非法)操作。当用户引导系统时,正常的操作只是引导系统,病毒乘机而入并不在人们预定目标之内。当运行带病毒文件时,执行该

文件的原来程序是人们的预定目标,在此之前先行执行的病毒程序并不是人们预定目标。但是,由于病毒程序的寄生性和进入的隐蔽性,人们开始时往往觉察不到它的存在和执行,直到产生比较严重后果时才有所觉察。

病毒程序所执行的非授权操作,通常包括以下几个方面:

第一,将病毒程序常驻内存。为此,修改一些系统参数,扩大内存常驻区,缩小用户可用空间大小,并将病毒程序存放到内存常驻区之内,使得当用户程序进入系统时,不占用病毒程序存储范围。

同时,为了使常驻内存的病毒程序监控系统运行情况,病毒程序还修改操作系统的中断向量表,即修改某些常用中断程序的入口地址,用病毒程序中某些段落的起始地址去代替它们。这样,当用户程序或 DOS 命令要求执行这些中断程序时,例如要求向磁盘写入或进行读操作时,以及文件操作、时钟操作等,对于感染了病毒的系统来说,实际上并不是立即执行所要求的中断程序,而是先执行了一段病毒程序,然后才能去执行正常的中断程序。这就使得调用这些中断时要付出额外的时间开销。

大多数病毒程序都驻留内存。也有一部分病毒程序不驻留内存,没有这方面内容。

第二,在一定条件下,“复制”病毒程序,即所谓“传染”、“感染”。病毒程序一旦进入正在运行的系统,它就如同盘踞在蜘蛛网中央的蜘蛛,虎视眈眈地监视磁盘和文件,一旦发现它们尚未感染病毒,就在适当时机,例如要执行某个系统中断程序时,先将病毒程序写入磁盘或文件中。这时,我们便称为这个磁盘或文件已经感染了病毒。

第三,在一定条件下进行破坏活动。例如,小球病毒在系统运行过程中会突然使屏幕上出现一个跳来跳去的小球,把屏幕显示弄乱,而以色列病毒在“黑色星期五”(既是13日又是星期五的日子)将使每一个要运行的程序消失不见,有些病毒“激活”(即发作)时会使整个磁盘内容受到破坏。这些破坏中,象屏幕产生异常显示但不影响系统继续运行,特别是不破坏磁盘或文件的,属于“轻微”或有恶作剧的意思,又称为“表现”,这种病毒也称为是“良性”的。而破坏磁盘或文件的病毒,称为是“恶性”的。

二、系统型病毒及其传染方式

1. 什么是系统型病毒?

系统病毒是指病毒程序攻击 DOS 系统占用的系统区的病毒。通常见到的是占据磁盘引导区或硬盘主

引导区的病毒,又称为引导区病毒。国内早期出现的小球病毒和大麻病毒,以及后来出现的2708病毒(Azusa)、GenP/GenB病毒、米开朗基罗病毒等。都属于这一类。

引导区病毒对于软盘和硬盘的传染不完全相同。对于软盘和硬盘都侵占DOS分区中的引导扇区的,称为BOOT型病毒,对于软盘侵占其引导扇区而对于硬盘侵占其主引导区的病毒,则称为“BOOT/PARTITION型病毒”。

引导区病毒将自身植入磁盘BOOT区或硬盘主引导区之后,该磁盘即受到感染。对于原来BOOT区内容(引导程序及磁盘参数表)和主引导区内容(主引导程序及硬盘分区表),或者将它们移放到磁盘其它扇区(如小球病毒、大麻病毒等),同时移放的还有病毒程序超过512字节的部分,或者取消被侵入扇区的原有内容,而将使用磁盘所必须用的参数和程序段落纳入到病毒程序中,如GenP/GenB病毒。2708病毒对软盘的感染采用前一种方式,对于硬盘的感染则采用后一种方式。

磁盘感染引导区病毒后,一旦用它引导系统,则首先执行的是已经侵占引导区的病毒程序,病毒程序驻留内存、修改系统参数和中断服务程序入口地址等一系列动作,都是在此期间完成的,然后,才装入并执行正常的BOOT程序(对于软盘)或主引导程序(对于硬盘)。也就是说,病毒程序的执行(主要是植入内存)是插入到引导过程之前实现的。病毒程序一旦进入系统,它又将自身置入中断服务程序之前,从而实现对于系统的运行过程的监控,以及在一定条件下,进行传染和破坏。在这种病毒环境下使用的磁盘,会很快受到感染,有时执行任何一种磁盘操作命令(例如DIR命令)都会传染病毒。

引导区病毒进入系统,一定要通过启动过程。在无病毒环境下使用的软盘或硬盘,即使它已感染引导区病毒,也不会进入系统并进行传染,但是,只要用感染引导区病毒的磁盘引导系统,就会使病毒程序进入内存,形成病毒环境。有的软盘缺少系统文件不能用于引导系统,当它的BOOT区感染病毒后,如果启动时将这些盘放在A驱动器中,往往也会使系统感染病毒。

在病毒环境下使用的软盘,凡是未贴写保护签的,都可能被病毒感染。硬盘则由于无法贴写保护签,最容易被感染,并且成为继续传染其它软盘的病毒源。因此,病毒往往能够很快地扩散,蔓延至更大范围、在短时间内造成大面积损害。

病毒程序在传染病毒时,往往先检查传染对象是否已感染同一种病毒。如未感染,就传染给它;如已感染,就不再传染它,即一种病毒对同一张盘一般只传染一次。

交叉感染的情况比较复杂。所谓交叉感染,是指同一张盘已经被若干种病毒传染。例如,若一张软盘的BOOT区已被病毒V1侵占,当它在病毒V2的环境下使用时,V2病毒程序占用BOOT区,而将原先侵占

BOOT区的V1病毒程序移至其它扇区。这时,这张盘就同时带有病毒V1和V2,如果用它引导系统,就会使两种病毒都损害系统和其它磁盘,检查和消除工作也更加复杂。(未完待续)

文件型六四病毒的简析与防治

青岛海洋大学计算机系90级(266003)

李超勇

新近上机发现有许多微机感染一奇怪病毒,用CPAV与KILL都不能消除它。笔者对其病毒代码进行了分析,证明其为一种文件型六四病毒(以前许多杂志上都提到过操作系统型六四病毒),并对多个文件进行解毒免疫处理,效果明显。现介绍如下:

一、类型:该病毒为常驻内存的文件型病毒,感染用户.COM、.EXE文件

二、感染后特征:该病毒对.COM与.EXE文件都只感染一次,感染.COM文件后文件长度增加1831字节,对.EXE文件感染后文件长度增加1824字节。在内存中安装病毒后,激发时显示红字蓝底一屏有关6.4动乱的口号,延时一会儿后自动恢复原屏幕内容。

三、工作原理

1. 执行染毒文件时,先执行病毒程序,病毒将调用INT 21H功能DFH,若内存已安装病毒,则返回AAH,否则AH不变。此举用来判定内存有否安装病毒,以确定有无必要安装病毒。病毒安装内存时修改INT21H中断向量,使其指向自己的中断服务程序,该服务程序加上DFH功能和EFH功能,并拦截4BH功能加上传染模块,拦截上述三功能外所有功能加上显示模块,然后留驻内存。

2. 病毒执行完后,若为.COM文件将调用INT 21H的EFH功能将原.COM文件移至100H开始并执行之。若为.EXE文件则转向正常入口。

3. 病毒的传染模块将在任一.EXE或.COM文件被执行时被调用。该模块被调用时将判断文件末七字节是否为该病毒特征代码(B8H, 00H, 02H, 50H, CBH, F6H, C7H),若是说明该文件已被感染,则转向正常入口,若不是则对其进行感染。若文件为.COM文件则在文件开始处加上720H字节病毒并在尾部加上特征代码。若文件为.EXE文件则把720H字节长病毒置于尾部(注:此时不会再放特征代码,因为720H字节末7字节即为特征代码),并保存与修改文件头相应部分,使其入口指向病毒。

4. 病毒的显示模块将在除4BH、DFH、EFH功能外任一INT 21H中断功能被调用时被调用。若此时时钟计数低7位全为0则可能激发显示模块,该模块激发时将找一与当前页不同页作显示页,设置颜色后以TTY方式显示屏幕内容,并延时数秒后自动设置回原显示页。

四、工作特点:该病毒的最大特点就是运用了异或

加密法。第一处在屏幕显示模块中,屏幕显示内容是以一字节与 B3H 异或后的形式放于程序中的,显示模块激发后将每一字节与 B3H 进行异或后显示。第二处在病毒感染 .EXE 文件时,病毒将文件头中原入口参数 IP、SP 值及相对段值 CS、SS 用 13H 异或后放于病毒代码中,运行该文件时用 13H 异或回原值再执行原文件。

五、检测。1. 检测在内存中是否安装病毒可用 DEBUG 编写简短汇编指令调用 INT 21H 的 DFH 功能。若 AH 返回 AAH,说明有毒,也可取出 INT 21H 中断向量,从向量段址偏移处开始反汇编,如有毒则开始三指令如下:

```
XXXX:0000 B4DF MOV AH,DF
XXXX:0002 CD21 INT 21
XXXX:0004 80FEAA CMP AH,AA
```

2. 检测文件有无病毒可用 DEBUG 或 PCTOOLS 查看该文件末七字节是否为特征代码。也可用 DEBUG 对 .COM 文件从头、对 .EXE 文件倒数 720H 字节开始反汇编,如有毒则其头三句与前述常驻内存病毒前三句同。另外也可用上述感染特征判断,此处不再赘述。

六、解毒与免疫

1. 对内存解毒可用 DEBUG 从内存病毒中偏移 025EH 处取一双字,重新设置 INT 21H 中断向量。当

然也可重新启动。对内存进行免疫可编写一小段汇编加上类似病毒的 INT 21H DFH 功能,INT 21H 中断向量指向该程序,末尾跳向正常入口,把这一程序留驻内存即可。若再加上类似病毒 INT 21H EFH 功能则可运行染毒文件而有效抑制传染。

2. 对 .COM 文件解毒可用 DEBUG 从 820H 开始比原程序少 720H 字节重写文件即可,而且末七字节正好用于免疫。对 .COM 文件进行免疫处理可在文件尾添上七字节特征代码。

3. 对 .EXE 文件可用 DEBUG 取出病毒代码偏移 0285H,开始四个字分别与 0013H 异或则得正常入口 IP、SP、CS、SS 值,将这四字分别写回文件头偏移 14H、16H、10H、0EH 四个字中,并在偏移 02H(文件长 mod 200H)和 04H(文件长 div 200H)二个字所描述文件长度减去 720H 字节用病毒前代码重写文件即可。对其进行免疫可在文件末尾写上 16 字节且其中最后七字节为特征代码,并修改文件头 02H、04H 两个字使其描述长度增加 16 字节。由于 .EXE 调入时不能调入文件头,所以可将其扩展名设为 .TXT 后再处理。

以上对文件型六·四病毒分析诸方面对剖析其他病毒也有不少用处,因病毒虽种类繁多五花八门,但其某些机能是差不多的。对于这种常驻内存型病毒,使其内存不受病毒感染是非常重要的。

改进的 CEC—I 实用小程序

天津铁道部第十八工程局通信科(300222) 薛嘉兴

我从去年开始接触、使用中华学习机 CEE—I,并陆续地学习了《中华学习机 CEC—I 苹果机工具箱》和《中华学习机使用技巧与检修 360 问》两本书,受到许多启发,同时对书中的一些印刷错误作了更正,对书中的程序(特别是机器语言程序)有的进行了压缩,有的进行了改写,有的进行了功能扩展。现先寄去几个小程序,希望能对中华学习机爱好者有所帮助。

下面我对几个小程序的使用作些说明:

程序一: COPY<复制磁盘>

该程序的特点是简短,占用内存空间少,使用时键入 CALL 768,然后根据提示插入源盘和复制盘即可完成复制标准 35 道磁盘的任务。它适用于只有一个驱动器的用户使用。

修改下列语句后,可用于只复制 0~2 道的 DOS 部分的任务:

```
301—02
306—30
315—4C
```

程序二: AUTO LINE(在浮点 BASIC 中使用自动行号)

原《工具箱》中的 AUTO LINE 程序只能在西文

状态使用,本程序在它的基础上作了改进,不管是中文还是西文状态都适用。使用方法:

1. 在第一次使用前键入 CALL 36864;
2. 使用时键入 &<行号>;
3. 退出时键入 ^X;
4. 重新进入时再键入 &<行号>即可;
5. 当 & 后不带行号时,若是第一次使用则行号从 10 开始,否则从上次退出时的行号继续。
6. 行号增量自动设置为 10。

程序三: REM REMOVER (REM 清除器)

该程序的功能是清除 BASIC 程序中的 REM 语句,以节省内存空间,提高 BASIC 程序的运行速度。它与《工具箱》中的同名程序功能相同,但长度仅为原程序的一半。

使用时键入 CALL 768 即可将 BASIC 程序中的 REM 语句清除。

程序四: KEY IN(实用机器语言编程器)

《工具箱》中的 KEY IN 程序能够快速地输入机器码到内存,对于不太熟悉 6502 指令的用户使用小汇编有时较方便,但每次进入和退出小汇编的手续又比较麻烦;有时在编写某个机器语言程序时往往需要有一

些提示字符,对于不太熟悉 ASCII 码的用户,需要查看 ASCII 码表。这些都使编写机器语言程序时的速度大大减慢,所以本程序对于原 KEY IN 程序作了改进,增加了下面两项功能,使编程更加方便了:

1. 自动小汇编功能:能够方便地进入和退出小汇编状态,并且在汇编时能自动给出地址号。

2. 字符快速输入器功能:自动地给出地址号,快速地将键入的字符(除 ^ C 外)的负 ASCII 码送入内存。

使用方法

1. 键入 FA0G (或 CALL 4000) 后显示主提示:“START ADD:”等待键入起始地址号。

2. 见主提示后,若键入 ^ C 则退出该程序进入监控;若键入 (则显示上次退出时的地址号,并将它作为新的起始地址号;若键入 <地址号> 则将新地址号作为起始地址号。

3. 输入完地址号后,显示菜单,用户根据提示选择所需功能:按“1”则进入自动小汇编;按“2”则进入字符快速输入器;按“^ C”则退出到监控状态;按“3”键或其他键则进入机器码快速输入器。

4. 在各种功能中,当用户按“^ C”时均退回到主提示状态。

程序五:PRINT DATA16(使 DATA 语句以16进制显示)

在实用程序中,经常会遇到在浮点 BASIC 程序中用 DATA 语句建立机器语言程序,然后用 CALL 语句调用的情况。如果能使列表时 DATA 语句中的数据以16进制显示,而其它语句以10进制显示,则对于较熟悉6502指令的操作人员分析 DATA 语句所建立的机器语言程序是非常方便的,本程序即具有这种功能

使用时键入 CALL 768,则列表时即具有上述功能,退出该功能时键入 ^ RESET 即可。

程序六:MOVE (移动机器语言程序并保证能正常运行)

机器语言程序必须装入适当的内存位置才能正确执行。某些机器语言程序在从内存的某一区域搬移到另一区域后,便不能保证其正确运行,需要对它进行修正,《工具箱》中的 MOVE 即具有搬移和校正三字节指令两种功能。但这种校正还不完善,为此,在此基础上我编写了该程序,增加了对两字节的条件转移语句和设置输出向量以及设置输入向量的语句的校正功能。使之更加实用。

使用方法有两种:

方法一:将要移动的程序的、末地址及新的首址依次按低位在前、高位在后的原则存入内存单位 \$FA ~ \$FF 单元中,然后键入 1200G

方法二:键入 11C3G,在提示符“=”后键入(新首址)<(旧首址).(旧末址)J

以上程序均在 CEC-I 机上运行通过。

1. COPY

0300—A9 22 80 4D 03 A9 80 8D 52 03 A9 00 8D EB B7 8D
0310—F0 B7 8D F3 B7 20 30 03 AD 4D 03 C9 0B 90 07 E9

0320—08 8D 4D 03 D0 EF C9 03 B0 01 60 A9 30 8D 52 03
0330—A9 01 8D F4 B7 A9 80 20 3F 03 EE F4 B7 A9 98 20
0340—59 03 A9 8F 8D F1 B7 A9 0F 8D ED B7 A9 22 8D EC
0350—B7 A9 80 8D E1 B7 4C 93 B7 8D 5F 03 A0 16 B9 80
0360—03 99 00 07 88 10 F7 8D 10 C0 AD 00 C0 10 FB A0
0370—18 A9 A0 99 00 07 88 10 FA 60 00 00 00 00 00 00
0380—49 4E 53 45 52 54 60 53 4F 55 52 43 45 60 44 49
0390—53 4B 60 7C 43 52 7E 00 49 4E 53 45 52 54 60 54
03A0—41 52 47 45 54 60 44 49 53 4B 60 7C 43 52 7E 00

2. AUTO LINE

9000—A9 4C 8D F5 03 A9 16 8D F6 03 A9 90 8D F7 03 60
9010—A0 00 B1 B8 F0 38 A9 10 8D F6 03 A2 05 BD A9 90
9020—95 05 CA 10 F8 A0 00 B1 B8 F0 09 38 E9 30 99 FA
9030—00 C8 D0 F3 84 00 20 98 D9 A4 00 F0 11 A9 F0 85
9040—08 A2 04 88 30 08 B9 FA 00 95 05 CA 10 F5 AD 55
9050—AA 85 00 AD 56 AA 85 01 A9 66 85 38 A9 90 85 39
9060—4C EA 03 6C 00 00 E0 06 B0 08 91 28 B5 05 18 69
9070—B0 60 20 63 90 48 C9 98 D0 0D A5 00 85 38 A5 01
9080—85 39 20 EA 03 68 60 C9 8D D0 FA 8A 48 A2 03 F6
9090—05 B5 05 C9 0A 30 07 A9 00 95 05 CA 10 F1 90 04
90A0—A9 01 95 05 68 AA 68 60 60 F0 F0 F0 01 00 F0 00

3. REM REMOVER

0300—A5 67 85 FC A5 68 85 FD A0 01 B1 FC D0 01 60 48
0310—88 B1 FC 48 A0 03 C8 B1 FC D0 03 4C A1 03 C9 B2
0320—D0 F4 C0 04 D0 02 A0 00 84 FB A0 03 CB B1 FC D0
0330—FB C8 98 38 E5 FB 85 FB A5 FC 85 FE A5 FD 85 FF
0340—A0 01 B1 FE F0 19 48 88 B1 FE 48 38 E5 FB 91 FE
0350—C8 B1 FE E9 00 91 FE 68 85 FE 68 85 FF B0 E3 68
0360—85 FE 68 85 FF B1 FC AA 48 88 B1 FC 85 FC 48 86
0370—FD 20 B8 03 A9 00 91 FC 20 AA 03 B1 FE 91 FC 20
0380—AA 03 20 B1 03 A5 FF C5 B0 90 F0 A5 FE C5 AF 90
0390—EA A5 AF E5 FB 85 AF 85 69 A5 B0 E9 00 85 B0 85
03A0—6A 68 85 FC 68 85 FD 4C 08 03 E6 FC D0 02 E6 FD
03B0—60 E6 FE D0 02 E6 FF 60 A5 FC D0 02 C6 FD C6 FC
03C0—60

4. KEY IN

0FA0—20 5A 11 A9 9F A0 11 20 3A DB 20 0C FD C9 83 D0
0FB0—03 4C 69 FF C9 B1 D0 03 4C C5 0F C9 B2 D0 03 4C
0FC0—09 11 4C 45 10 A5 06 D0 02 C6 07 C6 06 A9 10 8D
0FD0—98 0F A9 00 85 2F A9 E4 85 38 A9 0F 85 39 20 EA
0FE0—03 4C 50 D3 E0 05 B0 48 E0 01 B0 31 38 A5 06 65
0FF0—2F 85 06 A5 07 69 00 85 07 48 29 F0 2A 2A 2A
1000—2A 8D 94 0F 68 29 0F 8D 95 0F A5 06 48 29 F0 2A
1010—2A 2A 2A 2A 8D 96 0F 68 29 0F 8D 97 0F BD 94 0F
1020—C9 0A 90 03 18 69 07 69 B0 C9 C7 D0 02 A9 BA 60
1030—20 1B FD C9 83 D0 F8 A9 1B 85 38 A9 FD 85 39 20
1040—EA 03 4C A0 0F 20 8E FD 20 78 10 20 84 10 A9 00
1050—85 0A 20 98 10 A5 D1 0A 0A 0A 0A 85 09 E6 0A 20
1060—98 10 A5 D1 05 09 A0 00 91 06 E6 06 D0 02 E6 07
1070—20 93 10 D0 D6 4C 45 10 A5 07 A6 06 20 41 F9 A9
1080—BA 4C ED FD A5 06 29 07 85 08 0A 18 65 08 69 05
1090—85 24 60 A5 06 29 07 60 20 0C FD C9 83 F0 65 C9
10A0—88 F0 1F C9 95 F0 4E C9 C7 B0 ED C9 C1 90 04 E9
10B0—B7 D0 0A C9 BA B0 E1 C9 B0 90 DD E9 B0 85 D1 4C
10C0—E3 FD A0 00 A5 0A D0 20 20 93 10 D0 03 20 1A FC
10D0—A5 06 D0 02 C6 07 C6 06 20 84 10 B1 06 29 F0 85

10E0—09 E6 24 68 68 4C 5D 10 B1 06 29 0F 05 09 91 06
 10F0—68 68 4C 4B 10 A0 00 A5 0A F0 E0 68 68 B1 06 29
 1100—0F 4C 64 10 68 68 4C A0 0F 20 8E FD 20 78 10 20
 1110—84 10 20 27 11 A0 01 91 06 E6 06 D0 02 E6 07 20
 1120—93 10 D0 EB 4C 09 11 20 0C FD C9 83 F0 D8 C9 88
 1130—F0 0C C9 95 F0 1F C9 8D F0 50 4C ED FD 60 A0 00
 1140—20 93 10 D0 03 20 1A FC A5 06 D0 02 C6 07 C6 06
 1150—68 68 4C 0F 11 68 68 4C 19 11 A9 92 A0 11 20 3A
 1160—DB 20 6F FD A0 00 20 A7 FF AD 00 02 C9 8D F0 10
 1170—C9 83 D0 03 4C 69 FF A5 3E 85 06 A5 3F 85 07 60
 1180—A5 07 20 DA FD A5 06 4C DA FD A9 A0 20 ED FD A9
 1190—8D 60 8D D3 D4 C1 D2 D4 A0 C1 C4 C4 BA 00 00 8D
 11A0—B1 AE C1 D5 D4 CF A0 C1 D3 D3 C5 CD C2 CC C5 8D
 11B0—B2 AE C9 CE D0 D5 D4 A0 D3 D4 D2 C1 CE C7 8D B3
 11C0—AE C9 CE D0 D5 D4 A0 C4 C1 D4 C1 8D BA 00 00 00
 5. PRINT DATA 16

0300—20 C6 03 C9 C4 D0 32 20 B5 03 C9 C1 D0 2B 20 B5
 0310—03 C9 D4 D0 24 20 B5 03 C9 C1 D0 1D 20 F0 FD A9
 0320—A4 20 AD 03 C9 A0 F0 2C C9 8D F0 04 C9 2C D0 15
 0330—48 20 55 03 68 C9 AC F0 E8 20 F0 FD A9 03 85 36
 0340—85 37 4C EA 03 8E CD 03 8C CE 03 20 65 03 AE CD
 0350—03 AC CE 03 60 8E CD 03 8C CE 03 20 95 03 AE CD
 0360—03 AC CE 03 60 C9 B0 90 21 C9 BA B0 1D 29 0F 20
 0370—8B 03 65 44 AA A9 00 65 45 A8 20 8B 03 20 8B 03
 0380—8A 65 44 85 44 98 65 45 85 45 60 06 44 26 45 60
 0390—A5 45 20 97 03 A5 44 48 4A 4A 4A 20 A2 03 68
 03A0—29 0F 09 B0 C9 BA 90 02 69 06 4C F0 FD 48 A9 00
 03B0—85 44 85 45 68 20 F0 FD 68 18 69 01 85 36 68 69
 03C0—00 85 37 4C EA 03 A9 21 85 21 D0 EC 03 57 01 39
 6. MOVE

11C0—EA EA EA D8 20 3A FF A9 BD 85 33 20 67 FD 20 C7
 11D0—FF 20 A7 FF 84 34 A0 17 88 30 0D D9 CC FF D0 F8
 11E0—20 BE FF A4 34 4C D1 11 A5 3C 85 FA A5 3D 85 FB
 11F0—A5 3E 85 FC A5 3F 85 FD A5 42 85 FE A5 43 85 FF
 1200—A5 FA 85 F4 A5 FB 85 F5 A5 FE 85 F6 C5 FA A5 FF
 1210—85 F7 E5 FB 90 1F A5 FE E5 FA 85 F8 A5 FF E5 FB
 1220—85 F9 A9 00 85 0A A9 18 8D 76 13 A9 65 8D 78 13
 1230—8D 7D 13 D0 1C 38 A5 FA E5 FE 85 F8 A5 FB E5 FF
 1240—85 F9 A9 38 85 0A 8D 76 13 A9 E5 8D 78 13 8D 7D
 1250—13 A0 00 B1 F4 91 F6 E6 F6 D0 02 E6 F7 A5 F4 C5
 1260—FC A5 F5 E5 FD E6 F4 D0 02 E6 F5 90 E6 A5 FE 85
 1270—F4 A5 FF 85 F5 A5 FA 69 01 85 06 A5 FB 69 00 85
 1280—07 A2 00 A1 F4 20 8E F8 A4 2F C0 02 D0 2C 88 B1
 1290—F4 AA C8 20 5A 13 B0 07 88 20 75 13 C8 91 F4 20
 12A0—89 13 38 A5 2F 65 F4 85 F4 A9 00 65 F5 85 F5 C5
 12B0—F7 90 CE A5 F4 C5 F6 90 C8 60 C0 01 D0 E1 88 B1
 12C0—F4 C9 A9 D0 47 A0 02 B1 F4 C9 85 D0 D2 C8 B1 F4
 12D0—C9 36 F0 24 C9 38 F0 20 C9 37 F0 04 C9 39 D0 BF
 12E0—A0 05 B1 F4 AA A0 01 B1 F4 A0 05 20 75 13 A0 01
 12F0—91 F4 A9 07 85 2F D0 A7 A0 01 B1 F4 AA A0 05 B1
 1300—F4 A0 01 20 75 13 A0 05 91 F4 D0 E6 C9 F0 F0 1C
 1310—C9 D0 F0 18 C9 B0 F0 14 C9 90 F0 10 C9 70 F0 0C
 1320—C9 50 F0 08 C9 30 F0 04 C9 10 D0 CA C8 B1 F4 C9
 1330—80 B0 06 20 80 13 4C 3F 13 20 80 13 38 E9 01 20
 1340—5C 13 90 B2 A5 0A F0 08 B1 F4 18 65 F8 4C 55 13
 1350—B1 F4 38 E5 F8 91 F4 4C 9F 12 B1 F4 48 8A C5 FC

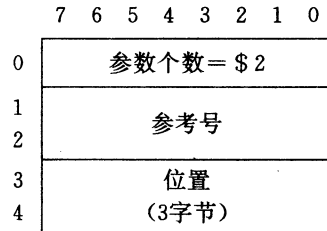
1360—68 48 E5 FD B0 0C 8A C5 FA 68 48 E5 FB 90 03 68
 1370—18 60 68 38 60 48 38 8A E5 F8 91 F4 68 E5 F9 60
 1380—18 65 06 AA A9 00 65 07 60 38 A5 2F 65 06 85 06
 1390—A9 00 65 07 85 07 60

(上接第22页)

此调用改变文件内当前位置为由位置参数指定的位置。位置不能超过文件末尾值(EOF)

参数个数:2个参数 参考号:同前

位置:这三字节值指向文件管理程序在文件内将要读或写下一个字节的绝对位置,这文件位置不能超过文件的 EOF。

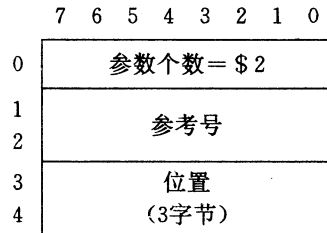


8. GET-MARK(\$CF)

此调用返回一个打开的文件内的当前位置(MARK)。

参数个数:2个参数。 参考号:同前

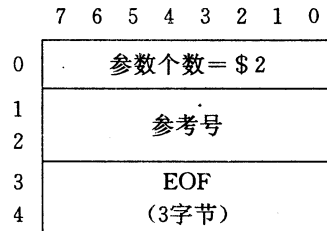
位置:这三字节值是在文件内下次开始读或写绝对位置,除非被一个随后 SET-MARK 调用改变。



9. SET-EOF(\$D0)

此调用设置被参考号指定的文件的逻辑长度为 EOF,若新 EOF 比当前 EOF 小,则超过新 EOF 的块被释放给系统。若新 EOF 比当前的 EOF 大或相等,则没有块被分配;若新 EOF 比当前位置小,则位置值被设为 EOF,这 EOF 不能被改变,除非文件被允许写。

参数个数:2个参数。 参考号:同前。



EOF:这三字节值表示一个文件的逻辑结尾,它可能比当前 EOF 的值大或小,若它小,则将超过新 EOF 的块释放给系统。(未完待续)

ProDOS 系统内部结构剖析(续)

北京铁路局卫生防疫站(100038) 廖 凯

7. SET_PREFIX(\$C6)

设置系统部首给指定的目录,路径名可能是一个全路径名或部分路径名。系统部首可通过用一个零来设置一个路径名为空的。路径名最大长度为 64 个字符。ProDOS 启动时,系统部首被设为启动盘的卷名。

参数个数:有 1 个参数。

路径名:同前

	7	6	5	4	3	2	1	0
0	参数个数 = \$1							
1	路径名							
2	(2 字节)							

8. GET_PREFIX(\$C7)

此调用返回当前的系统部首。若系统部首为空的(无部首),则返回一个零。其它情况返回的部首被斜线分开:例如/APPLE/或APPLE/BYTES/。被返回的部首内的每个字符的最高位为零。

参数个数:1 个参数。

	7	6	5	4	3	2	1	0
0	参数个数 = \$1							
1	数据缓冲区							
2	(2 字节)							

数据缓冲区:这两字节地址指向放置部首的缓冲区。它至少 128 个字节长度。

五、文件调用

这部分将详细描述文件调用的每个调用。

1. OPEN(\$C8)

此调用准备读或写一个文件。它产生一个文件控制块,记载经路径名指定的文件的当前(打开的)特征,它设置文件内当前位置为 0,并返回一个参考号。

这输入/输出缓冲区是被系统为打开的文件所使用。它包含磁盘上有关文件结构的信息,并包含当前要读或写的 512 字节块。它一直被使用直到文件被关闭为止,因此,此字段不应被用户修改。

	7	6	5	4	3	2	1	0
0	参数个数 = \$3							
1	路径名							
2	(2 字节)							
3	输入/输出缓冲区							
4	(2 字节)							
5	参考号							

此调用最多可同时打开八个文件。

当一个文件打开时,它被指定为从 0 到 4 的级别,随系统整体页面内 LEVEL 位置(\$BF94)的当前值而定。当给出的 CLOSE 命令使用一个为 0 的参考号时,在当前层次或以上的层次的所有文件都被关闭。所以,CLOSE 使用一个为 0 的参考号和为 0 的级别时,将关闭所有打开的文件。

注意:一旦文件被打开后,就不要移动磁盘或与其它驱动器交换。系统在写盘之前不检查卷是否是打开文件时的卷。系统程序应对此进行检查。

参数个数:有 3 个参数。

路径名,同前。

输入/输出缓冲区:此两字节地址指示一个 1024 字节的输入/输出缓冲区的起始地址。缓冲区必须不被系统使用的一个页界(\$100 的倍数)开始。

若一个标准文件正被存取,则输入/输出缓冲区的首 512 字节包含当前正在读或写数据的块,第二个 512 字节包含当前的索引块。若正存取一个目录文件,则首 512 字节包含当前目录文件块,其余部分未使用。

参考号:当一个文件打开时,文件处理系统赋给此字节一个值,对于这打开的文件,所有以后的命令都使用此参考号。

2. NEWLINE(\$C9)

此调用对于任何打开的文件读操作,允许你启动或中止换行。当中止换行时,一个读请求在已读取所需的字符数时或到文件末尾时终止,当允许换行时,若读取了换行字符,读请求也将终止。

	7	6	5	4	3	2	1	0
0	参数个数 = \$1							
1	参考号							
2	允许屏蔽							
3	换行字符							

每个要读取的字符首先送到用户的数据缓冲区,然后它与“允许屏蔽”字段进行 AND 运算,并与换行字符进行比较。若符合则终止读操作,例如,“允许屏蔽”值是 \$7F 和“换行字符”值是 \$0D(ASCII 码 CR),若得到的值是 \$0D 或 \$8D 则终止输入。这过程不改变字符。

参数个数:有 3 个参数。

参考号:这是文件被打开时系统指定的文件处理参考号。

允许屏蔽:一个 \$0 的值中止换行方式;一个非零

值允许换行方式,在允许换行时,每个输入字节与此字节 AND 运算,然后与换行字符比较,若符合,读请求被终止,一个 \$FF 值使所有位有效,一个 \$7F 值只引起位 0 至位 6 被检测。

换行字符:当换行被允许时,若输入字符与允许屏蔽 AND 运算并等于此值,则读请求被终止。

3. READ(\$CA)

此调用由参考号指定的文件的当前位置(MARK)开始,传送所需的字节到数据缓冲区指针指定的缓冲区。将实际传送的字节数送到“传送数”字段。

在允许换行读取方式下,如果在需求数字字节被读取之前遇到一个换行字符。那么传送数参数被设为传送的字节数,包括换行字节。

如果在需求数字字节被读取之前到了文件的末尾,那么传送数被设为传送的字节数,也只有当传送了零字节(传送数=0)时才返回文件尾错误码(\$4C)。

参数个数:有 4 个参数。

参考号:同前

	7	6	5	4	3	2	1	0
0	参数个数 = \$4							
1	参考号							
2	数据缓冲区							
3	(2 字节)							
4	需求数							
5	(2 字节)							
6	转换数							
7	(2 字节)							

数据缓冲区:这两字节地址指向从文件读取数据的目的地址。

需求数:此两字节指定被传送到由数据缓冲区指针指定的缓冲区内的字节的最大数量,这最大值限制在数据缓冲区起始处和最近用的存储区页之间的字节数。

传送数:这两字节值指出实际读取的字节数。只在允许换行读取方式下读到换行字符。或者在请求期间发生某些其它错误,或遇到 EOF 它才少于需求数。

4. WRITE(\$CB)

此调用从由数据缓冲区指针指定的缓冲区传送指定的字节到由参考号指定的文件内,在文件内当前的位置(MARK)开始。实际上传送的字节数被送到传送数内。如果需要,文件位置被修改为:位置+传送数,附加的数据和索引块被分配给文件,EOF 被增加。

参数个数:4 个参数。

参考号:同前。

数据缓冲区:这两字节地址指向要被传送到文件的数据的起始位置。

需求数:同前。

	7	6	5	4	3	2	1	0
0	参数个数 = \$4							
1	参考号							
2	数据缓冲区							
3	(2 字节)							
4	需求数							
5	(2 字节)							
6	传送数							
7	(2 字节)							

传送数:这两字节值指出实际上传送的字节数。如果没有错误发生,这值总是等于需求数。

5. CLOSE(\$CC)

这调用用于释放所有被一个打开的文件所使用的资源。文件控制块被释放,如果需要,文件的输入/输出缓冲区被清除,修改文件的目录项目,直到该参考号指定另一个被打开的文件之前,以后使用那个参考号的文件调用将失效。

	7	6	5	4	3	2	1	0
0	参数个数 = \$1							
1	参考号							

若参考号等于零,则所有在当前级别及以上级别的打开的文件被关闭。例如,在 0,1 和 2 级别你打开文件,设置当前级别为 1,然后将参考号设为 0,使用 CLOSE 命令后,在级别 1 和 2 的文件被关闭,但在级别 0 的文件没有关闭。

级别的值从 0 到 4,它被存在系统整体页面的 LEVEL 单元(\$BFD8)内,它只能由系统程序来改变,并由 OPEN 和 CLOSE 所使用,此调用会设置备份位。

参数个数:1 个参数。

参考号:在文件被打开时将文件处理参考号分配给文件。CLOSE 将释放这参考号,若参考号=0,则所有打开的文件在当前级别及以上级别被关闭。

6. FLUSH(\$CD)

此调用清除文件的写缓冲区(输入/输出缓冲区),修改文件的目录。如果参考号等于零,则所有在当前级别及以上级别的打开文件被刷新。

	7	6	5	4	3	2	1	0
0	参数个数 = \$1							
1	参考号							

备份位被此调用所设置。

参数个数:1 个参数。

参考号:若参考号为零,则所有在当前及以上级别的打开文件被刷新。

7. SET-MARK(\$CE)

(下转第 20 页)

PCT V2.0 写作系统

张连永 杨 峰

PCT V2.0 写作系统是在苹果机及其兼容机上研究开发 CAI 软件的良好环境,实现了汉字、字符放大,定位显示,图像的压缩存取,图块动画,窗口操作,复音音乐,语音合成等功能,显示速度快,图像清晰。

一、系统的功能

PCT V2.0 写作系统集中了一些工具软件的特点,并扩展了 30 多条语句,功能大大加强,并与 DOS 3.3 系统完全兼容。

PCT V2.0 写作系统采用软汉字 14×16 点阵字模结构,是根据苹果机及兼容机的显示特点所决定的。采用整字节处理,故处理速度非常快。该系统可以检索内存中 1000 多个汉字,其显示速度达 150 个字/秒,并对汉字进行放大,反相显示,闪烁等处理,更好地是该系统具有窗口存储功能,可以造出镂空字,立体字。

本系统的图形压缩功能取决于图形的复杂程度,最好情况下可把 8K 的图形压缩为只有区区 240 个字节,最坏情况下图形压缩后比原图形空间多出 160 个字节,平均可压缩为原来的 1/3 左右。对压缩图形的恢复显示只需 1 秒钟左右,这样,用户不必为同时处理多幅画面时速度太慢而发愁。

一个好的教学软件必须有较好的画面和动画效果,PCT V2.0 写作系统支持图块操作,连续地对图块进行显示和消隐,就可以形成很好的动画。本系统提供了一整套动画制作过程,用户无需了解其原理就能轻而易举地实现动画功能。最多支持 8 个动画联动,其动画速度可调,可同时插入音响,可以正片,倒片,还有指定动画分解动作。

语音、音乐合成功能是让微机说话的有趣功能,教学软件配上语音和动听悦耳的音乐将会达到更好效果,但是需占很大内存。本系统加强了音乐功能,它的合成效果完全符合乐理,发音准确,而且自动产生伴音,即所谓复音,其效果比常见的单音音乐要好听得多。

二、系统语句

(一)基本图形操作语句

1. &DP n 设置在高分图形第 n 页上操作,n 为表达式。
2. &SP n 全屏显示高分图形第 n 页,n 为表达式。
3. &(EOR、STORE、AND、OR、NOT)设置图块、动画、汉字和字符的显示方式。
4. &REDUCE n 将压缩图形库中第 n 个压缩图形恢复于高分屏幕,n 为表达式。

5. &SET X,Y 在操作页上设置显示图块、汉字或字符的位置,X,Y 为表达式。

6. &ENLARGE n,m 设置汉字或字符在显示时放大的倍数。n 表示横向放大的倍数, $1 \leq n \leq 5$,m 表示纵向放大的倍数, $1 \leq m \leq 24$ 。

7. &PRINT A,在高分屏幕上显示字符。

8. &PRINT C,[m,n]在操作页上打印汉字,m,n 为表达式, $m, n \geq 0$ 。

(二)窗口操作语句

1. &WINDOW X₁,Y₁,X₂,Y₂
在操作页上设置窗口,为随后的窗口操作做准备,X₁,Y₁,X₂,Y₂ 为表达式。
2. &ALL
设置窗口为全屏。
3. &CLEAR
把操作页上的窗口清为黑色。
4. ©
把另一页相应窗口内容复制到操作页的窗口中。
5. &INVERSE
把操作页的窗口反相显示,两次反显可使窗口恢复原状。

6. &ROLL <U,D,L,R> n,m
将操作页窗口中的画面按 n 种方式向<上、下、左、右>卷动 m 次。

7. &FROLL n
将操作页全屏以每次滚动 n 行的方式快速卷出,同时另一页卷出。

(三)图块(动画)操作语句

1. &PRINT B,n
在操作页上将图块库中第 n 号图块显示于指定位置。
2. &PRINT PB,n,m
在操作页上,用 &SET X,Y 设置显示位置后,执行该语句,则第 m 号图块将显示于相对的设置位置向右偏移 n 个点的位置上。
3. &PGC
用于两页切换。
4. &SAVE W
把设定的窗口内容以图块方式存入图块库,并且按 SAVE W 的执行顺序将图块按 0,1,2……编号。
5. &ACT n₁,n₂,\$n₃[,m,[a] TO [b]]
动画语句,若 n₁≠0,那么,n₁,n₂ 为配合音响的参数。若 n₁=0,那么 n₂ 表示动画延时。\$n₃ 表示动画链的首址。

(四)音响语句。

1. &NOISE x,y

主要用于产生特殊音响效果,X,Y 为表达式。

2. &SOUND X,Y

本语句与中华学习机(CEC-I)上的 MUSIC 语句功能相同,参数也相同,用于处理简单的单音音乐。

3. &MUSIC n

演奏音乐库中第 n 号音乐, $n \geq 0$

4. &SAY \$n, \$m

从地址 \$n 到 \$m 将语音数据取出,振动喇叭发声。

(五)其它操作语句

1. &INKEY A \$

读键盘语句。若有键盘输入,则将输入字符赋予 A \$,若无键盘输入,则 A \$ 为空,程序继续运行。

2. &CLKEY

清键盘语句。

3. &WAIT n

延时语句,约延时 n/6 秒。

4. &GOTO n

无条件转移语句,n 为整型表达式。

5. &GOSUB n

用于替代 ON.....GOSUB

6. &DATA n

将数据指针指向第 n 个数据,准备读取。

7. &VAL X, V \$

为 BASIC 语句中自定义函数的扩充,X 为简单变量名,V \$ 为字符串变量名。

8. &MOVE \$n1, \$m1, \$n2

将内存中 \$n1 到 \$m1 的内容移至以 \$n2 为首址的内存中。

9. &SWAP \$n1, \$m1, \$n2

将内存中 \$n1 到 \$m1 的内容与以 \$n2 为首址,长度相等的内容相互交换。

10. &DEF (B,C,P,N) \$n

&DEF B:定义图块库首址。

&DEF C:定义汉字库首址。

&DEF P:定义压缩图形库首址。

&DEF N:定义复音音乐库首址。

\$n 为地址,可用十六进制或十进制。

PCT V2.0 系统扩展了 34 条语句,供用户编制软件时作用。

补 遗

本刊 93 年 5 期《学习机游戏攻关秘诀》一文,付排时因稿件脱漏而未登全,今特补上,并向作者王海涛同志深致歉意。

编辑部

附游戏攻关秘诀

(1)脉冲星(PULSAR

]BLOAD PULSAR

]CALL-151

* 9406:02 修改命数,如改为 0A(下同)

* 1EFD6

(2)打臭虫(BUGATTACK)

]BLOAD BUGATTACK

]CALL-151

* 49D1:02

* 830G

(3)大赛车(AUTOBAHN)

]BLOAD AUTOBAHN

]CALL-151

* 1FF1:02

* 800(4)空投(STAR BLAZER)

]BLOAD STARBLAZER

]CALL-151

* 2000:03 油量

* 2280:04 飞船数

* 2180:30 炸弹数

* FD0G

(5)大嘴巴(PUC'MAN)

]BLOAD PUC'MAN

]CALL-151

* 849:02

* 800G

(6)小偷(SPY'S DEMISE)

]BLOAD SPY

]CALL-151

* 29AB:XX 小偷为 XX 人

* 800G

更 正

我在贵刊 1993 年第 1 期 13 页发表的《实用程序三则》中之“BASIC 行编辑工具”程序有误,致使程序运行时有缺陷,即每当显示保留字“GOTO”时,显示不出,而以空格代替。这一错误是由于设计程序时考虑不周造成的,我对因此而给用者带来不便深表歉意!

要消除此缺陷,应对程序修改如下:

8F22:09

8F24:80

8F2B:D0

8F2D:F0

8F3C:10

另外,《实用程序三则》正文倒数第二段的最后一句也应改为:“将 \$914D~\$9156 与 \$91D0~\$91D5 的内容改成 \$EA 即可”。

广 告

北京富益电子技术开发公司备有 EPSON LQ-1600K 打印机和佳能 BJ-10e 喷墨打印机现货,价格优惠,有需要购买者请迅速联系。

电话:8221677;联系人:尹明;邮编:100036;通信地址:北京 173 信箱。

第四届全国计算机软件人员竞赛将在京举行(初级程序员级)

一、宗旨:

为提高我国软件人员的素质,鼓励广大青年计算机软件人员为振兴中华而勤奋学习,并利于发现人才、培养人才。

二、主办单位:

由中国软件行业协会考试指导委员会、北京市科学技术协会、电子工业出版社等单位联合举办。

三、参赛对象:

竞赛选手产生自以下两途径:

1. 按规定时间完成刊登于《电子与电脑》杂志今年第八期上刊出的初级程序员级有奖竞赛试题,成绩优秀者。

2. 各省市软件水平考试实施办推荐参加全国计算机软件资格和水平考试的优胜者。(1~2名)

四、参赛条件及内容:

条件:未获得计算机专业本科学历或助理工程师职称者。年龄在35周岁以下。

内容:计算机知识、DOS、中西文录入、BASIC、C语言、dBASE III、IBM PC及兼容机机种。

五、竞赛日期、地点:1993年10月下旬,北京清华大学。

六、奖品:

设:一等奖1名,二等奖2名,三等奖5名,鼓励奖若干名。将分别奖给获奖证书、奖杯,及其它纪念品。

七、选手差旅费、住宿费一律自理,代为安排食宿。

八、联系地址:

北京清华大学计算机系转中软协会培校(邮编:100084)

电话:2561144-2373,联系人:张岱、金慧芬

电挂:1331,传真:2562463

一九九三年计算机初级程序员有奖竞赛试题答卷

总分:_____ 学号:_____ (如是函授学员请注明学号)

姓名:_____ 性别:_____ 年龄:_____

文化程度:_____ 工作单位:_____

通信地址:_____

邮编:_____

注意事项:

1. 答案必须在9月30日前寄出,以当地邮戳为准(通信地址:北京清华大学计算机系转中软培校,邮政编码:100084)

2. 填写答案时,必须将试题号与对应的答案栏对准,切勿弄错,若某试题有多个答案时,请顺序给出(共150分)。

3. 解答时字迹务必清楚,字迹不清楚时不能得分。

4. 是否愿意参加复赛:是_____,否_____。

题号	小 题 号 与 答 案										得 分	
一	a		b		c		d				8	
	e		f		g		h					
二	1	A	B	C	D						12	
	2	A	B	C	D							
	3	A	B	C	D							
三	1		2		3		4				12	
四	1		2		3		4				12	
五	1		2		3		4		5		10	
	6		7		8		9		10			
六	1		2		3		4		5		10	
	6		7		8		9		10			
七	a		b		c		d		e		10	
八	(1)		(2)		(3)		(4)		(5)		15	
九	1		2		3		4		5		10	
	6		7		8		9		10			
十	A		B		C		D		E		10	
十一	1				2				3		15	
十二	1	(1)			(2)						20	
		(3)			(4)							
	2	(1)			(2)						6	
总分											150	

一九九三年计算机初级程序员有奖竞赛试题

一、从选择的答案中选出适合下列表格中的 a~h 的正确答案,把相应的编号写在答卷纸的对应栏内。(每空 1 分,共 8 分)

二进制数	八进制数	十进制数	十六进制数
11111011	c	251	g
a	d	123	7B
01010111	127	e	h
b	257	f	AF

a, b 供选择的答案:

- ①11111111 ②10101111 ③10110001
④11111010 ⑤01111011 ⑥10000101

c, d 供选择的答案:

- ①173 ②277 ③305
④357 ⑤373 ⑥377

e, f, g, h 供选择的答案:

- ①57 ②-55 ③7F ④175 ⑤157
⑥87 ⑦-81 ⑧8F ⑨FB ⑩FF

二、从供选择的答案中,选出正确答案,把相应的编号写在答卷纸的对应栏内。(每空 1 分,共 12 分)

1. 对于线性表的(A)存储结构,要在表中插入或删除一个元素,需要(B)表中其它元素,其优点是可以对表元素进行(C)访问。当采用(D)存储结构时,只能以顺序方式访问表中元素,但可以方便插入和删除表中元素。

供选择的答案:

- A. D. ①顺序 ②链式 ③逻辑 ④物理
B. ①查找 ②移动 ③排序 ④链接
C. ①顺序 ②索引 ③定位 ④随机

2. 结构化程序设计是一种程序设计方法,按照这种方法,能够设计出结构合理、易读易写、易于维护和验证的程序,采用(A)的设计原则,主要使用(B)基本程序控制结构,并保持每个程序块只有(C)的原则,限制(D)语句的使用。

供选择的答案:

- A. ①自顶向下、逐步求精
②自底向上、逐步抽象
③数据抽象与功能抽象相统一
④自顶向下、逐步抽象
B. ①调用、返回、转移 ②递归、队列、堆栈
③顺序、分支、循环 ④过程、子程序、函数
C. ①一个入口、多个出口
②一个入口、一个出口
③多个入口、一个出口
④多个入口、多个出口
D. ①while ②goto ③if ④for

3. 数据库系统是在(A)的基础上发展起来的,常用的数据模型有(B),数据库提供的数据库语言有(C),在关系数据库系统中数据的基本操作有(D)三种。

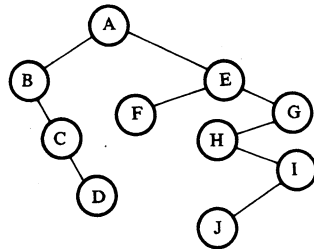
供选择的答案:

- A. ①控制系统 ②文件系统
③操作系统 ④网络系统
B. ①网状、层次、关系 ②网络、关系、B 树
③链接、顺序、索引 ④逻辑、物理、E-R 图
C. ①DDL、DML ②Pascd
③COBOL ④FORTRAN
D. ①分支、顺序、循环 ②增加、删除、修改
③投影、连接、选择 ④存储、打印、拷贝

三、根据下列各题的说明,选出正确答案,把相应的编号写在答卷纸的对应栏内。(每空 3 分共 12 分)

1. 设有二叉树如图,请给出中序遍历的结点访问次序。

- ①ABCDEF GHIJ; ②BCDAFEHJIG;
③DCBFJIHG EA; ④JIHG FEDCBA.



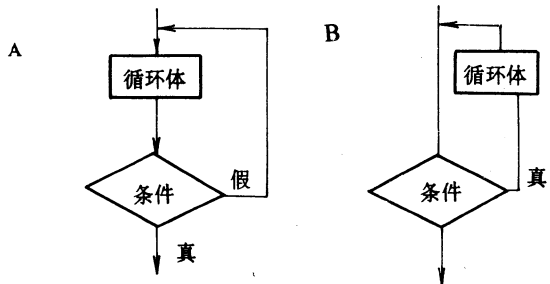
2. 试分析下列程序,将程序运行后出现的信息对应的标号填入答卷纸对应栏内。

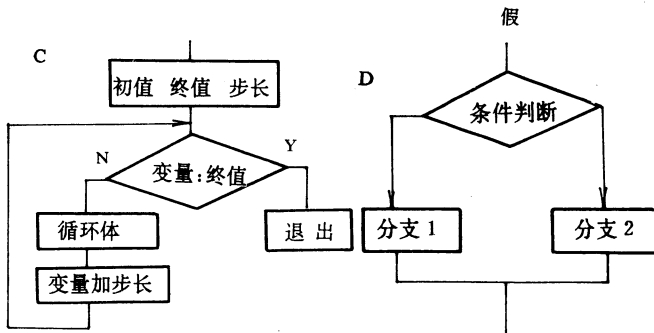
```
10 DIM A(20)
20 READ A(1)
30 PRINT A(1)
40 GOTO 20
50 DATA 0,9,8,7,6,5,4,3,2,1
60 END
```

供选择的答案:

- ①Redo from start ②Subscript out of range in 30
③Syntax error in 30 ④Out of DATA in 20

3. 判断下列各流程图中哪一图描述的是当型循环。()





4. 指出下列 C 语言程序所完成的功能

```
#include <stdio.h>
main ()
{int top,bottom,flag,x,pos,min;
int a[10]={1,2,3,4,5,6,7,8,9,10};
top=0;
pos=-1;
bottom=9;
flag=0;
scanf("%d",&x);
while ((flag==0)&&(top<=bottom))
{min=(bottom+top)/2;
if(x==a[min])
{flag=1;
pos=min;}
else
if (x<a[min])
bottom=min-1;
else top=min+1;
}
printf("%d",pos);
}
```

供选择的答案:

- ①把 x 插入数组 a 中
- ②对数组 a 求和,结果放入 x 中
- ③在数组 a 中对 x 进行二分查找
- ④求数组 a 的最小值,结果放 x 中。

四、阅读分析下列 BASIC 程序,从给出答案中选出正确答案,将相应的编号写在答卷纸的对应栏内。(每空 3 分共 12 分)

```
1.
10 READ X
20 X=INT (ABS(X))
30 ON X GOTO 100,150,200,250
100 Y=X+X;GOTO 300
150 Y=X * X;GOTO 300
200 Y=X/2;GOTO 300
250 Y=SQR(X)
300 PRINT Y
310 DATA -4.99
```

320 END

供选择的答案:

- ①10 ②4 ③5 ④2

2.

```
10 DEF FNA(X)=X * Y
20 READ Y
30 PRINT FNA(20+1);
40 READ Y
50 PRINT FNA(FNA(5))
60 DATA 2,3
70 END
```

供选择的答案:

- ①42 45 ②45 20 ③40 15 ④42 40

3.

```
10 OPTION BASE 1
20 DIM A (3,3),B(3,3)
30 FOR I=1 TO 3
40 FOR J=1 TO 3
50 READ A(I,J)
70 NEXT J
80 NEXT I
110 FOR I=3 TO 1 STEP -1
120 FOR J=3 TO 1 STEP -1
130 B (I,J)=A(J,I)
140 PRINT B(I,J);
150 NEXT J
160 PRINT
170 NEXT I
180 DATA 1,2,3,4,5,6,7,8,9
200 END
```

供选择的答案:

- ①123 ②987 ③963 ④321
- 456 654 852 654
- 789 321 741 987

4.

```
10 DIM A(9)
20 FOR I= 1 TO 9
30 READ A(I)
40 NEXT I
50 FOR I=8 TO 1 STEP -1
60 FOR J=1 TO I
70 IF A(J)>A(J+1) THEN SWAP A(J),A(J+1)
80 NEXT J
90 NEXT I
100 FOR I=1 TO 9
110 PRINT A(I);“ ”;
120 NEXT I
130 DATA 9,8,3,5,6,2,7,1,4
140 END
```

供选择的答案:

- ①987654321 ②983562714
- ③123456789 ④135792468

五、从供选择的答案中选出正确答案,把相应的编号写在答卷纸的对应栏内(每空 1 分,共 10 分)

- 1. 设 $[x]_{10} = 10111$, $[y]_{10} = 00101$

则 $[x-y]$ 补=_____

供选择的答案:

- ①00010 ②01010
③10010 ④11100

2. 对二进制数 11011011 采用偶校验, 校验位的值应是_____。

供选择的答案:

- ①0 ②1
③2 ④6

3. 微型计算机在工作中电源突然中断, 则计算机_____全部丢失, 再次通电后也不能恢复。

供选择的答案:

- ①软盘中的信息 ②RAM 中的信息
③ROM 中的信息 ④硬盘中的信息

4. 一台微型计算机的内存储器容量是 640KB, 这里的 1KB 为_____字节。

供选择的答案:

- ①1024 个字节 ②1024 个二进制单位
③1000 个字节 ④1000 个二进制位

5. 某计算机的内存储器是 64KB, 那么它的内存的地址是_____位。

供选择的答案:

- ①12 ②14 ③16 ④18

6. 计算机的内存储器是以字节定义的, 一般说, 一个字节包含的二进制位数是_____位。

供选择的答案:

- ①4 ②8 ③16 ④32

7. 能把高级语言编写的源程序转换成机器语言的目标程序的系统软件, 称为_____。

供选择的答案:

- ①解释程序 ②编译程序
③汇编程序 ④源程序

8. 微型计算机由_____部分组成的。

供选择的答案:

- ①控制器、运算器、存储器、输入/输出设备
②控制器、主机、键盘、显示器、打印机
③控制器、加法器、寄存器、显示器、打印机
④主机、微处理器、输入/输出设备

9. CPU 的中文意思是_____。

供选择的答案:

- ①主机 ②中央处理机 ③控制器 ④存储器

10. 在一个指令周期中, CPU 执行的步骤是: 读指令、指令译码、取操作数, 执行指令_____。

供选择的答案:

- ①写指令 ②确定寄存器
③存放结果 ④显示结果

六、判断下面的叙述是否正确, 正确的打 \checkmark , 错误的打 \times , 填在答卷纸的对应栏内。(每空 1 分, 共 10 分)

1. 顺序存储方式的优点是存储密度大, 且插入、删除运算效率高。

2. 栈和队列的存储方式既可以是顺序方式, 也可

以是链接方式。

3. 在数据库系统中全局的逻辑数据视图叫概念模型, 物理数据存储的模型叫内模型。

4. DOS 内部命令是在系统启动时由装入程序读入内存并常驻内存的命令。

5. 在多级目录结构中, 不允许有两个不同的文件具有相同的名字。

6. 磁盘缓冲区是磁盘上存放暂存数据的存储空间。

7. 中断是指主机对外部设备的访问停止, 继续执行程序。

8. 作业是系统进行资源分配和调度的基本单位。

9. 操作系统的存贮管理主要是解决内存的分配、保护和扩充问题。

10. 在 dBASE III 系统中, 对数据库文件的排序(即分类)称为“逻辑排序”, 对数据库文件的索引称为“物理排序”。

七、输入变量为 A、B 的输出函数 F, 有下面所列的真值表, 请从供选择的答案中选择与 1~5 各输出函数对应的逻辑表达式, 填入答卷纸对应的栏内。(每空 2 分共 10 分)

输入变量 \ 输出函数	A	0011	逻辑表达式
	B	0101	
1	F	0001	a
2	F	1000	b
3	F	1110	c
4	F	0110	d
5	F	1001	e

供选择的答案:

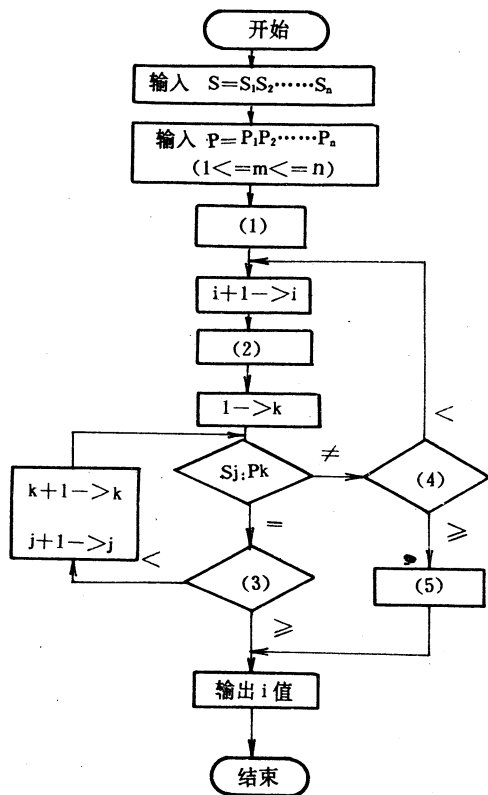
- ① $F = \overline{A} \cdot \overline{B}$ ② $F = A \cdot B$
③ $F = A \cdot \overline{B} + \overline{A} \cdot B$
④ $F = A \cdot B + \overline{A} \cdot \overline{B}$
⑤ $F = \overline{A + B}$

八、下列框图给出字符串匹配的一个算法。填充框图中的空白部分(1)~(5)。填入答卷纸对应的栏内。(每空 3 分共 15 分)。[框图说明]

已知两字符串 $S = S_1 S_2 \dots S_n (1 \leq n)$, $P = P_1 P_2 \dots P_m (1 \leq m \leq n)$, 检查 P 是否为 S 的一个子序列, 即是否存在 i, 使得 $S_i = P_1, S_{i+1} = P_2, \dots, S_{i+m-1} = P_m$ 。如有, 求最小 i 值。如没有, 置 $i = -1$ 。图中 i 表示比较时, S 的子序列的第一个字符之序号, k 表示被比较的 P 中字符之序号。

供选择的答案:

- A. $i - > j$ B. $0 - > j$
C. $k; m$ D. $0 - > i$
E. $i; j$ F. $-1 - > i$
G. $i; n - m + 1$ H. $j - > i$



九、请将下面的计算机英语术语译成中文。(10分)

1. virus 2. networking 3. CAD
 4. keyboard 5. cylinder 6. interface
 7. EOF 8. bus 9. dead lock 10. batch file

十、从供选择的答案中选出正确答案,把相应的编号填入答卷纸的对应栏内。(每空 2 分共 10 分)

A total computer system includes both hardware and A .

Warning: all the B on the disk will be lost.

Information is read into C by an input device and is then available for internal D .

The first screen of the application program usually a E , will be displayed.

供选择的答案

- A~E: ①data ②program ③memory
 ④processing ⑤software ⑥track ⑦disk
 ⑧menu ⑨computer ⑩directory

十一、阅读分析下列程序,然后回答 1—3 题。把答案写在答卷纸的对应栏内。(每题 3 分共 15 分)

```
20 INPUT "M,N=",M,N
30 IF M<N THEN 20
40 IF N=0 THEN 160
50 IF M=N THEN 160
60 K=M
```

```
70 GOSUB 180
80 C=F
90 K=N
100 GOSUB 180
110 C=C/F
120 K=M-N
130 GOSUB 180
140 PRINT "C(M,N)=",C/F
150 GOTO 170
160 PRINT "C(M,N)=",1
170 END
180 REM SUB
190 F=1
200 FOR I=2 TO K
210 F=F*I
220 NEXT I
230 RETURN
```

问题:

- 给出程序所实现的运算表达式。
- 当结果为 1 时, M 和 N 所有可能的值。
- 子程序所实现的运算表达式。

十二、阅读下面说明,完成各题的要求。把答案填在答卷纸的对应栏内。(26 分)

在汉字 dBASE III 系统下,执行命令

- use product
- list

屏幕显示信息如下:

RECORD #	编号	名称	价格	数量
1	004	电视机	3000.00	20
2	002	录音机	1000.00	50
3	005	录像机	3250.00	25
4	001	收音机	80.00	100
5	003	摄像机	8550.00	10

1. 写出完成下列要求的命令:(每题 5 分共 20 分)

(1)将数据库 product 中的编号字段进行索引,新文件名名为 pro.ndx.

(2)将数据库 product 中的名称、数量字段的结构拷贝出来,新文件名名为 pro1.dbf.

(3)计算库中价格<2000 的商品的总数量。

(4)将数据库 product 复制成标准格式文件,新文件名名为 pro2.txt.

2. 写出运行下列程序后,屏幕显示的信息。(6 分)

```
set talk off
use product
list 编号 for 数量<25
```

监控程序阅读器

郝 健 罗明宽

对于电子计算机的爱好者,尤其是单片计算机的爱好者来说,都免不了要过监控程序这一关,或者是按照别人所作好的程序抄一下,或者是自己编制一些小的监控管理程序。因此,读懂别人的监控管理程序或存于 EPROM 中的一些专用程序,不但可以丰富自己的知识,掌握编程技巧,使自己的水平有所提高,而且对以后的教学、开发应用或科研还会产生举足轻重的作用。由于这些程序都是固化在 EPROM 中,只能从中读出来用眼睛观察用笔记录,而不能进行一些小的修改或变动,在缺乏一定手段的条件下,要想读懂几 K 字节长的汇编程序,十分不容易。为了解决这个问题,我与辅导老师一起试制成了 EPROM 阅读器。今天向大家介绍的这个阅读器主要是用于 DP-851。它有三个主要的特点:一、可以阅读别人编写好的存于 EPROM 中的程序,此时可以随时对原程序进行修改,修改后立即运行以观察其效果;二、对于 DP-861 来说,这个阅读器还扩大其 RAM 的容量,或者是执行双监控管理程序,指的是在 DP-851 上可以存在两个监控管理程序,如一个是原监控管理程序(DP-851 的),一个是用户自己编的新监控管理程序,对于这两个程序可以事先通过开关进行设置,使其接通电源后可以直接执行预先设置的监控管理程序;三、对于 DP-851 来说,加上这个阅读器,可以使它的开发应用变得更为灵活。由于 DP-851 所能够使用的 RAM 为 8K 字节,地址范围为 8000H—9FFFH,所以用户在使用时,其程序只能在这个范围之内,将来固化用户程序时也只能将用户程序固化在这个地址范围之内。而使用这个阅读器来编写用户的程序时可以在很大的范围之内进行编写,也可以直接在 0000H 单元起始到 7FFFH 单元结束的地址空间内进行监控管理程序的编写。

这个阅读器主要分三个部分:首先是地址译码部分,这主要由 74LS138 完成;第二部分是片选信号切换,这部分主要完成地址空间的切换功能;第三部分是 ROM、RAM 芯片的联接,通过改变一些开关的位置,使其可以适应多种不同容量的 EPROM 或 RAM 芯片,变得十分灵活。

电路原理介绍:

一、片选信号译码部分。这部分电路由 IC₁: 74LS138 完成。74LS138 芯片的 4、5、6 脚为它的控制端,当 4、5 脚为低。6 脚为高时,芯片进入工作状态。74LS138 的 4、5 脚接地,6 脚接高电平,这样保证了 74LS138 可以正常的进行译码工作。74LS138 的 1、2、

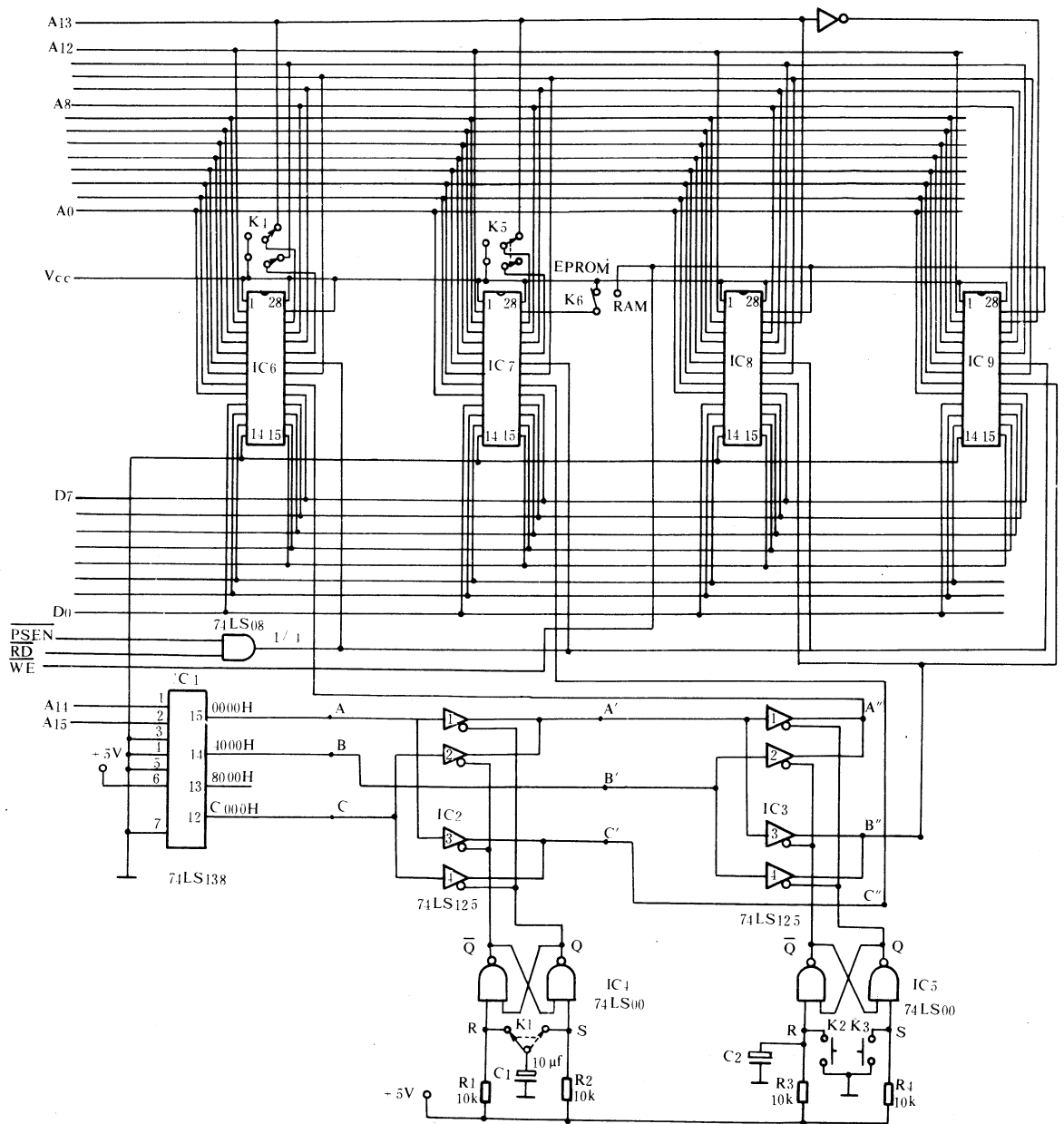
3 脚是三个输入端。根据 DP-851 存储空间的使用与 DP-851 的监控管理程序,因此需产生的片选信号有: 0000H—3FFFH;当地址线所确定的单元在这个范围内时,需产生一根片选信号。这个片选信号供将来执行 0000H 单元启动的监控管理程序使用。

4000H—7FFFH;这 16K 的空间供将来扩展的 RAM 使用。

C000H—FFFFH;这 16K 的空间供将来扩展的 ROM 或 RAM 使用。

74LS138 的 1、2、3 脚的接法为:1 脚接 A14,2 脚接 A15,3 脚接地,此时对应的输出为 15 脚、14 脚、13 脚与 12 脚。当地址范围为 0000H—3FFFH 之间时 15 脚为低电平,其它输出脚为高电平;当地址范围为 4000H—7FFFH 之间时 14 脚为低电平,其它输出脚为高电平;当地址范围为 C000H—FFFFH 之间时 12 脚为低电平,其它输出脚为高电平;这样我们所需要的三根片选信号就有了。

二、片选信号切换电路。这部分电路是整个电路中的关键部分。整个切换电路由两组切换电路组成。第一组切换电路是将 0000H—3FFFH 信号与 C000H—FFFFH 信号进行切换,通过这个切换可以使 IC₆(应是原监控管理程序的 ROM)IC₇(用户自编的程序或其它程序)都有在 DP-851 接通电源后可以直接执行的机会。第一组切换电路由 IC₂: 74LS125、IC₄: 74LS00 的一部分组成。首先是应用 IC₄ 中的两个与非门构成的一个 R-S 触发器。R-S 触发器的两个控制端分别接一个上拉电阻到电源的正极。这样根据 R-S 触发器的状态表可知 R-S 触发器为保持状态。这个保持状态是指保持原状态。而 R-S 触发器刚接通电源时的原态是随机不定的。为了使 R-S 触发器的初始状态能够确定下来,因此将基本 R-S 触发器加上一个单刀双掷的开关与一个电解电容就可以使 R-S 触发器的初始状态确定了。如图, K₁ 的两端分别接到 R-S 触发器的 R 端与 S 端, K₁ 的中心接一个电解电容接地。这样,首先分析 K₁ 与 R 端连接时,这时在接通电源后, S 端为高电平,电源电流通过电阻 R₁ 给 C₁ 充电,因此 R 端电压开始为低电平。这时输出端 Q 为 0, \bar{Q} 为 1。随着电源对 C₁ 的不断充电, C₁ 两端电压不断升高,即 R 端电压不断升高,充电过程完毕后 R 端为高电平,此时的 S 端仍为高电平。因此电路进入保持状态,即保持 Q=0, \bar{Q} =1 状态;与此同理,开关 K₁ 与 S 端连接时,



接通电源后, R 端为高电平, S 端开始为低, 因为 C_1 两端电压不可能突变, 此时 $Q=1, \bar{Q}=0$ 。随着不断的充电, C_1 两端电压不断升高, 直至接近电源电压, 这时 S 端相当于接高电平, R—S 触发器进入保持状态, 保持的为 $Q=1, \bar{Q}=0$ 。现在可以得出结论, 当开关 K_1 与 R 端连接时, 接通电源后 $Q=0, \bar{Q}=1$ 。而当开关 K_1 与 S 端连接时, 接通电源后 $Q=1, \bar{Q}=0$ 。对于信号的切换由 IC_2 完成, 如图: A、B、C 三点的信号分别为 0000H—3FFFH 间的片选信号、4000H—7FFFH 间的片选

号、C000H—FFFFH 间的片选信号。 IC_2 中的门 1、门 3 的输入为 0000H—3FFFH 的片选信号, IC_2 中的门 2、门 4 的输入为 C000H—FFFFH 的片选信号, 门 1、门 4 的控制端的信号来自 R—S 端触发器的 Q 端, 门 2、门 3 的控制端的信号来自 R—S 触发器的 \bar{Q} 端。当 Q 端为 0 时 (\bar{Q} 为 1) 即 K_1 与 R 端连接, 此时 IC_2 的门 2、门 3 截止 (这两个门的输出端对外部呈现为高阻状态), 门 1、门 4 导通。0000H—3FFFH 的片选信号通过 A 点, 再通过门 1, 达到 A' 点, 因此 A' 点的信号为 0000H—

3FFFH 的片选信号;C000H—FFFFH 的片选信号通过 C 点,再经过门 4,到达 C'点,因此 C'点的信号为 C000H—FFFFH 的片选信号;4000H—7FFFH 的片选信号由 B 点直接到达 B'点。而当 Q 为 1 时(\bar{Q} 为 0)即 K_1 与 S 端连接,此时 IC_2 的门 1、门 4 截止(这两个门的输出端对外部呈现高阻状态),门 2、门 3 导通。0000H—3FFFH 的片选信号通过 A 点,再经过门 3,到达 C'点,因此 C'点的信号为 0000H—3FFFH 的片选信号;C000H—FFFFH 的信号通过 C 点,再经过门 2,到达 A'点,因此 A'点的信号为 C000H—FFFFH 的片选信号;4000H—7FFFH 的片选信号还是由 B 点直接到达 B'点。因此,我们在接通电源前可以通过调整 K_1 的位置来调整供给 IC_6 、 IC_7 的片选信号。因为 8031 在启动时,从地址空间的 0000H 单元开始读取数据执行,所以我们调整 K_1 使 IC_6 、 IC_7 都有为 0000H 单元的机会,因此可以执行任何一块 IC 中的程序。这就是前边所说的执行双监控管理程序。第二组切换电路是将第一组切换好的 0000H—3FFFH 的片选信号与 4000H—7FFFH 的片选信号进行切换。因为 4000H—7FFFH 的空间是我们扩展的 RAM 空间,为了使我们能够模拟编写、调试一个完整的监控程序的全部过程,我们采取这样一个步骤:1、在正常的状态下(应用原来的监控管理程序)将我们编写的监控程序输入到地址空间为 4000H—7FFFH 的单元中去,这时可以进行任何的修改等;2、当认为我们所编写的监控程序可以执行后,通过地址切换,将 0000H—3FFFH 的片选信号切换到 RAM 空间上,使 RAM 的片选信号为 0000H—3FFFH。(原 RAM 的 4000H 单元对应为 0000H 单元,原 RAM 的 4001H 单元对应为 0001H 单元……原 RAM 的 7FFFH 单元对应为 3FFFH 单元)。之后,再使 DP—851 复位。此时执行的即是我们所编写的存于 RAM 中的新监控管理程序。第二组切换电路由 IC_3 、74LS125 与 IC_4 的另一部分组成。同样使用 IC_4 中的另外两个与非门组成一个基本 R—S 触发器,这个触发器的两个输入端分别接两个上拉电阻,且其中一个还接一个电容,目的是每次开机后电流通过电阻 R_3 对电容充电,使 R 端为低,S 端通过上拉电阻被置为高。此时, IC_3 中的 1、4 导通,门 2、3 截止,因此 A'点的 0000H 片选信号经过门 1 到达 A''点,B'点的 4000 片选信号经过门 4 到达 B''点。C'点的 C000H 的片选信号直接到达 C''点。因为 IC_6 的片选信号取自 A''点,所以 IC_6 工作时的地址空间为 0000H—3FFFH 的 16K, IC_6 、 IC_7 的工作地址空间为 4000H—7FFFH 的 16K。这个触发器的两个输入端还分别接两个微动开关。目的是通过微动开关使 R 端或 S 端接地,将其置为低,以能控制输出端 Q 的状态,这样就可以控制门 1、4 导通还是门 2、3 导通。因为开始时 R 端为 0,S 端为 1,Q 为 0,门 1、4 导通,所以执行 IC_6 中的程序,当按下开关 K_3 后 Q 为 1,此时门 1、4 截止,门 2、

3 导通,A'点的 0000H 片选信号经过门 3 到达 B''点,B'点的 4000H 片选信号经过门 2 到达 A''点。这时候再按下 DP—851 的复位键,DP—851 就会执行 IC_6 、 IC_7 中的程序。

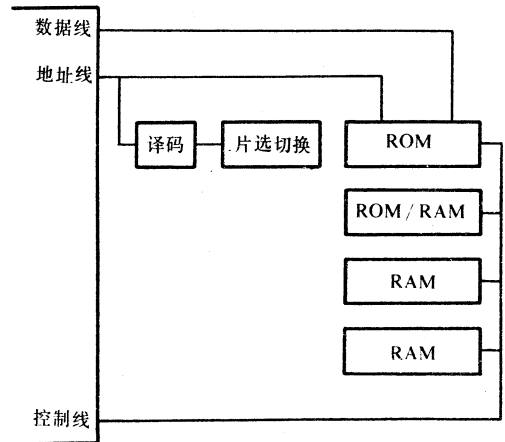


图 1 阅读器框图

对于图中所标的 IC_6 、 IC_7 ,是 EPROM 的插座,为了考虑能够适合多种不同型号的 EPROM 芯片,因此设置了开关 K_4 、 K_5 、 K_6 。(下转 34 页)

电子工业出版社广州科技公司 邮购部邮购书讯

- BIO 结构分析教程 定价:21.50 元 邮挂费:3.20 元
 高频开关稳压电源 定价:9.80 元 邮挂费:1.50 元
 开关电源集成电路手册(POWERINTEGRATED CIRCUIT DATA BOOK)定价:18.00 元 邮挂费:2.70 元
 MICROSOFT C6.0 大全 定价:34.00 元 邮挂费:5.10 元
 NOVELL 网络及其互联技术 定价:13.00 元 邮挂费:2.00 元
 微计算机的下一场革命——多媒体技术 定价:12.00 元 邮挂费:1.50 元
 NOVELL 网络安装·使用·开发指南(V2.15~V3.11) 定价:15.50 元 邮挂费:2.30 元
 电子工业出版社广州科技公司邮购部:
 通信地址:广州市石牌华南师大北区一号 203
 邮编:510630 联系人:王丽端

运用“看门狗”技术提高在线式智能仪器的可靠性

浙江湖州市自来水公司(313000) 汤忠庆

对于接触过智能仪器设计的工程技术人员来说, 都有这样的体会: 安装调试完后的智能仪器在确认了软硬件均无问题以后, 结果仍不太理想。有时, 接通电源后仪器并不工作, 处于死机状态; 有时一开始仪器是工作的, 过一会儿仪器会莫名其妙地死机, 等按一下硬件复位按钮, 仪器才开始工作, 过后仪器又莫名其妙地死机了。

这样的仪器只能放在实验室里摆摆样子, 在工程上是不能应用的。必须找出解决问题的办法, 提高可靠性。通过深入研究这些故障现象, 原因主要由外界干扰引起的, 除了电压波动以外, 还有好多不可预测因素。

因此, 要彻底解决干扰问题, 提高电源精度仅能减少故障出现的次数, 有些不可预测的干扰还是无法排除。针对这个“顽症”, 笔者引进了看门狗技术, “对症下药”、根除了这个“顽症”, 取得了满意的效果。

以智能仪器设计中常用的 8031 单片机为例, 笔者利用看门狗技术, 设计了如下电路图。

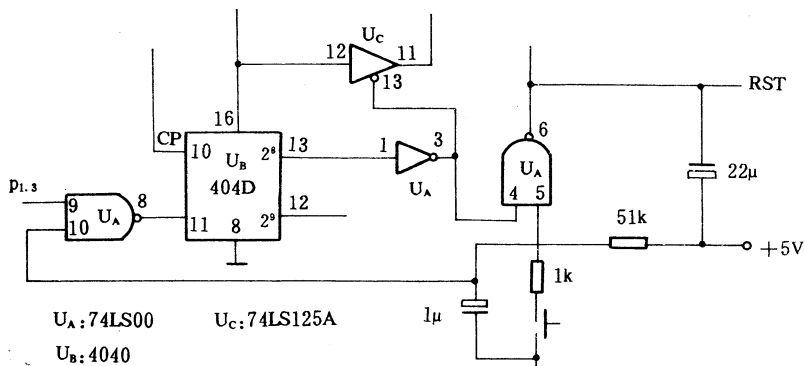
从图可见, 看门狗技术是利用硬件复位达到克服死机的目的。但是在实际应用中, 在智能仪器的复位端如只照搬下图, 加这样一个复位装置, 反而会适得其反, 仪器无法工作。原因是 4040 计数器输出端一旦计

满数有一高电平输出, 使仪器复位, 又从头开始, 等下一个周期计数器溢出, 仪器又强迫复位。

为使仪器在正常情况下能稳定工作, 遇到特殊情况时, 能自动复位、重新工作, 除了外加硬件自动复位电路以外, 还必须在软件上采取一些措施。

解决的办法是, 在软件入口端加一指令, 置 P1.3 端于高电平, 计数器处于计数状态, 在程序适当地地方加一条置 P1.3 端于低电位指令和一条置 P1.3 端高位指令, 这样计数器清零以后, 又接着重新开始计数。其间关键之处就是确定计数指令和清零指令在软件中的间隔数。遵循原则是此间隔的程序执行时间必须小于计数器输出端从计数到产生溢出的时间, 具体可根据计算求得。间隔数和时钟频率 CP、机器指令周期 T 有关。只有这样程序才能顺利执行下去。一旦受到干扰, 出现死机情况, 计数器就再也得不到清零信号, 计数器一直计数直至输出端溢出, 使 RST 得到复位信号, 仪器自动复位, 避免死机的发生, 提高了可靠性。

笔者应用此方法成功地应用在自动循环参数检测仪、小型程控交换机等在线式智能仪器设计中, 取得了满意的效果, 大大提高了仪器的可靠性, 杜绝了死机现象。



图

如何将苹果机 AD/DA 卡改制为 PC 机 AD/DA 卡

北京理工大学附中高二 王 晰

目前市场上 PC 机 AD/DA 卡的价格对个人来说确实过高。而现在有些降价的苹果机 AD/DA 卡很便宜,性能也不错,但由于总线不同,不兼容,无法在 PC 机上使用。因此,如能增加少量硬件,通过适当转换,使苹果机 AD/DA 卡能用到 PC 机上,则是很有实际意义的工作。同时也为不同总线之间的转换问题,提供了一种可行的方法。

通过对 AD/DA 卡结构的分析可知,AD/DA 卡的主要部分都是相似的,不同机型的卡的区别在于接口电路不同。只要通过附加转换电路,改变总线接口类型,AD/DA 卡就可接在不同主机上,就像打印机可配各种接口一样。

虽然苹果机和 PC 机都采用了扩充槽的思想,采用了开放式总线结构,但它们之间除引脚不同外,还有其他很多不同之处,具体说明如下:

1. 苹果机的 I/O 访问控制线是 $\overline{\text{DEVEN}}$ 和 $\text{R}/\overline{\text{W}}$, PC 机是 IOR 和 IOW 。
2. 苹果机的 I/O 地址译码电路在主板上,扩充卡占的地址与它插的槽号有关。PC 机采用统一的 I/O 编址,由卡上自己译码。
3. AD/DA 卡都需要一个时钟基准,或由卡上自己产生,或取自扩充槽的时基信号线。这两种机型的时基信号频率是不同的。
4. 由于 CPU 不同,6502 仅在 $\phi 0$ 高电平周期访问总线,另半周期由显示电路控制总线,而 PC 机无此问题。
5. 两种总线的速度相差很多,对元件的响应速度要求不一样。
6. 6502 没有专门的 I/O 空间,是将存储器空间的一部分用于 I/O,而 PC 机有独立的 I/O 空间。

针对上述不同点,采取如下解决办法:

对于第 1 点需加转换电路。对于第 2 点需加地址译码电路。对于第 3 点既可加振荡电路,也可对 PC 机 14.318MHZ 信号进行分频。对于第 4、第 5 点,由于读/写 AD/DA 卡时实际上是对卡上的锁存器进行操作,所以可以忽略,不会有什么影响。第 6 点是软件访问方法上的不同,不需硬件转换。

下面介绍一个根据上述方法进行改接的实例:

我用一块 16 路 AD 输入,1 路 DA 输出的 8 位的苹果机 AD/DA 卡进行了改接,并已获得成功。其转换速度为 $17\mu\text{S}$,型号是 BA-113。改接电路图见图 1。电路原理简介如下:

V1:74LS688 作为地址译码电路,

V2:74HCT00 作为信号转换电路和晶体振荡电路,

V3:74LS74 作为分频电路。

地址译码电路和信号转换电路将原苹果机的地址空间 $\text{C0X0}-\text{C0XF}$ 切换到 PC 机 I/O 空间 300-30F。晶体振荡电路和分频电路产生 1MHz 的 $\phi 0$ 时基信号。

具体制作时可用一块 PC 实验板作底板,将原 AD/DA 板固定在上面,再把转换电路按图 1 接法焊在空余地方即可,参考布局如图 2 所示。

改制后,卡的性能和使用方法不变。但软件访问的空间要作如前面所述的变化。如果希望使用别的 I/O 地址,可通过改变 74LS688 数字比较器 B 口的接线来实现。

虽然上面的实例电路是对于一种 AD/DA 卡设计的,但是由于各种 AD/DA 卡转换方法基本相同,当用其他苹果机 AD/DA 卡与 PC 机相联接时,只需根据各卡之间的不同之处,对本方法进行少量改动即可。解决方法简述如下:

1. 由于 AD/DA 通道数不同,8 路 AD/DA 卡使用 A_0-A_2 三根地址线,16 路卡使用 A_0-A_3 四根地址线。凡卡上不用的地址线都应接到地址译码电路上。
2. 卡上时钟的来源可能不一样。如果卡上有时钟振荡电路,则无须加时钟产生电路。如果取自 $\phi 0$ 外的其它引脚,则应改变本线路的时钟产生部分,产生与原来频率相同的信号。

具体不同之处,可通过直接观察原卡,看各信号线的使用情况来确定。只要方法正确,所有 AD/DA 卡都能改装成功。

苹果机与 PC 机扩展槽引脚图如图 3。

(上接 32 页)

本卡使用方法:

1. 将 DP-851 上的监控管理程序芯片(27128)取下插入本图中 IC_8 插座中,将本卡与 DP-851 连接好。
2. 将 K_1 与 R 端连接;将 K_4 拨到 27128 端;接通电源即可正常工作。

注: IC_7 如选用 EPROM,则 K_6 拨至 EPORM 端。如 EPROM 的型号为 2716、2732 时 K_5 拨至 2732 端,如 EPROM 的型号为 2764、27128 时 K_5 拨至 27128 端。; IC_7 如选用 RAM6264 芯片时,则 K_5 拨至 27128 端, K_6 拨至 RAM 端。

注:如 IC_6 选用 2716、2732 时, K_4 拨至 2732 端。

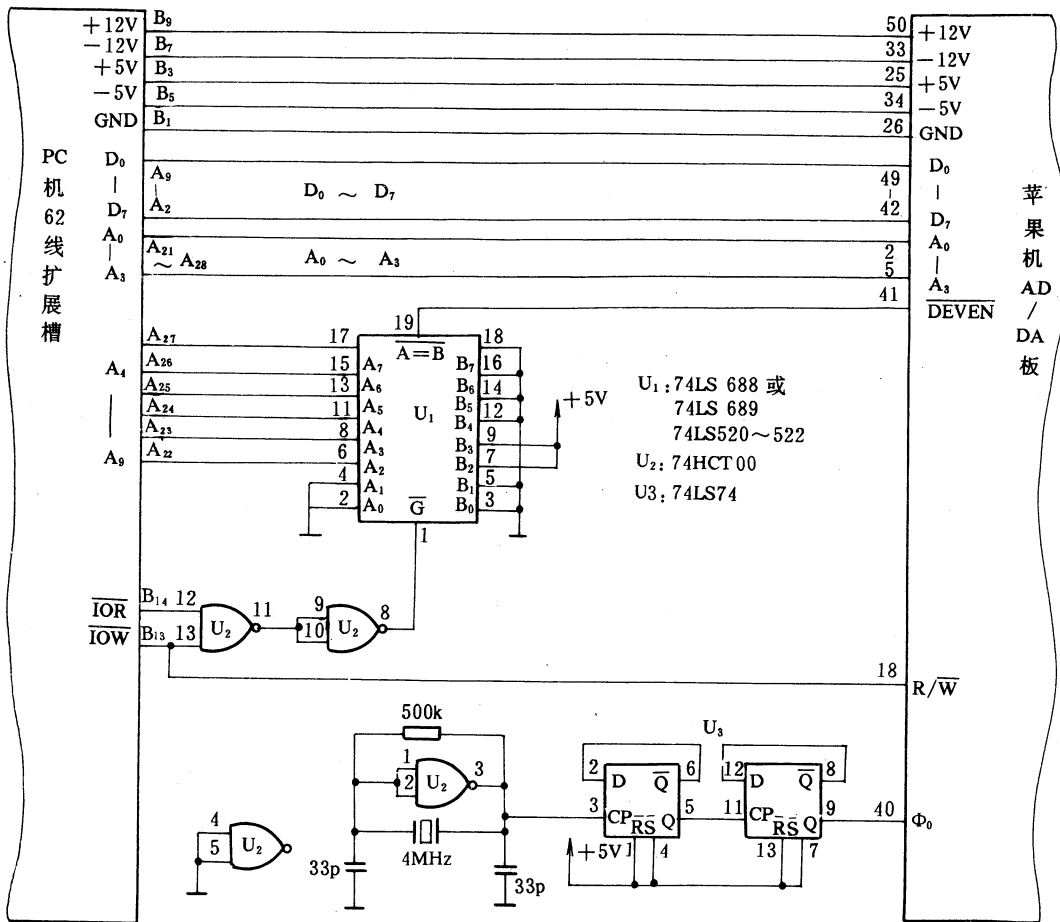


图 1 苹果机/PC 机 AD/DA 卡转换电路

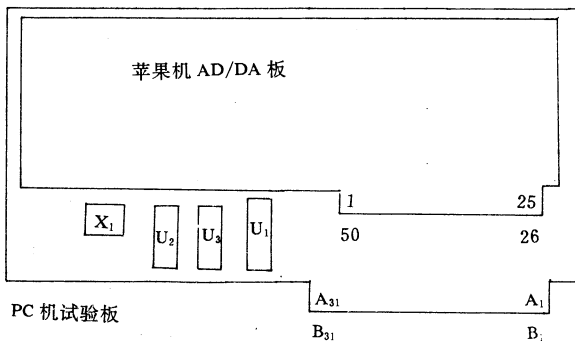


图 2 转换板的布局

电子工业出版社广州科技公司
综合经营部

公司地址: 广州市天河五山路华师大科技服务楼
215 号

开户行及帐号: 广州建行天河高新办 480 -
261485-55

电话: (020)7504448-3872

联系人: 陈昊 邮编: 510631

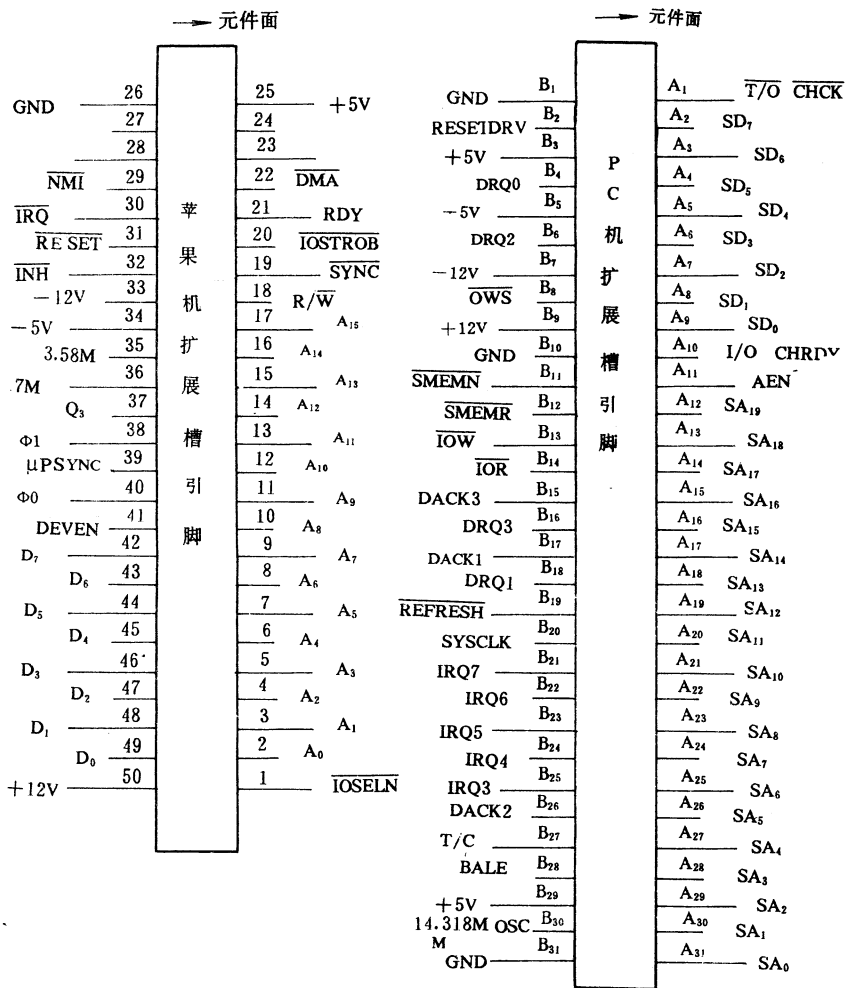


图3 苹果机及PC机扩展槽引脚图示

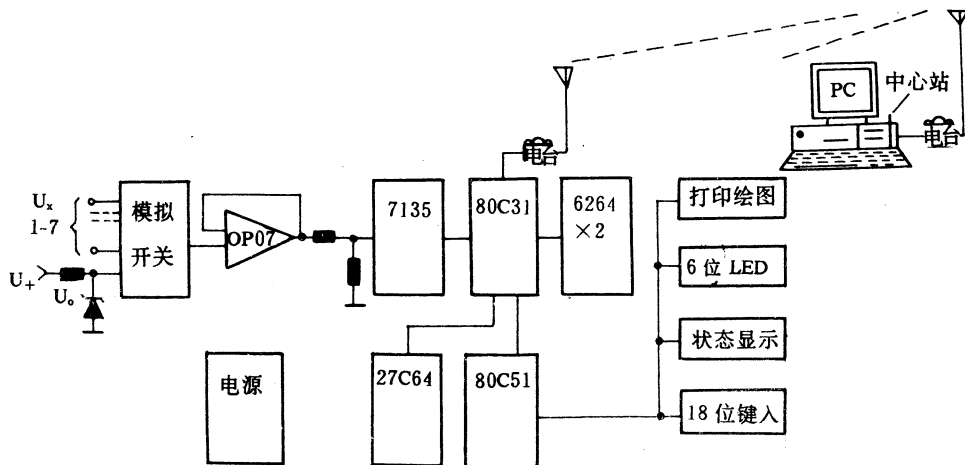
UN-1A 型电压模拟数字采集器

国家地震局分析预报中心(100036) 刘北顺 王玉民 师会

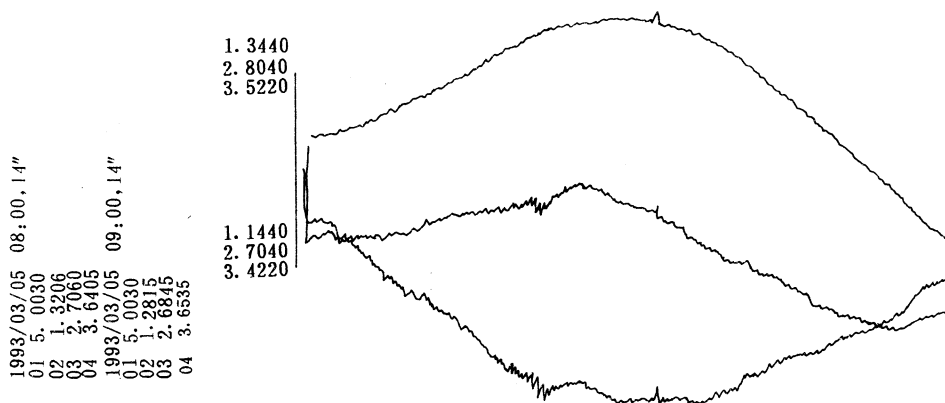
为及时发现野外台站地震前兆信号的突发性异常变化,我们研制了“UN-1A型电压模拟数字采集器”。该机具有数字采集与传输、可绘制模拟曲线、功耗低、

使用方便可靠等特点。其功耗低的优势,为野外取代功耗大的模拟记录器提供了广阔的前景。

一、主要工作原理



UN-1A 型电压模拟数字采集器原理框图



UN-1A 型电压模拟数字采集器记录图样

仪器具有 8 个测量通道：第 1 通道为标准电压测量通道，第 2、3、4 通道具有分采样并绘图功能，第 5、6、7、8 通道具有小时采样并打印数据功能。

仪器加电后，自动进行对标准电压和 16K RAM 数据区的检测，当标准电压值和 RAM 区无误时，进入工作状态。若发现错误，LED 则显示错误内容。

被测电压通过选通器件阻抗变换及分压电路送到 7135 芯片，CPU 定时启动 7135 芯片进行 A/D 转换。7135 分时输出 5 位 BCD 码，该码被 CPU 读取进行转换和处理，并以 16 进制形式存储在相应的 RAM 区，以完成一次采样。

数据的打印可按 1, 2, 4, 8 小时的间隔设定。CPU 将打印的内容从数据区调出，通过 8155 接口芯片，在 PP-40 打印绘图机上打印出来。

绘图有两种方式：定时绘图和立即绘图。

① 定时绘图，规定每日十时绘图，时间从前日十时到当日十时，数据内容为第 2, 3, 4 通道的采样结果，绘

图程序将自动完成绘图功能。

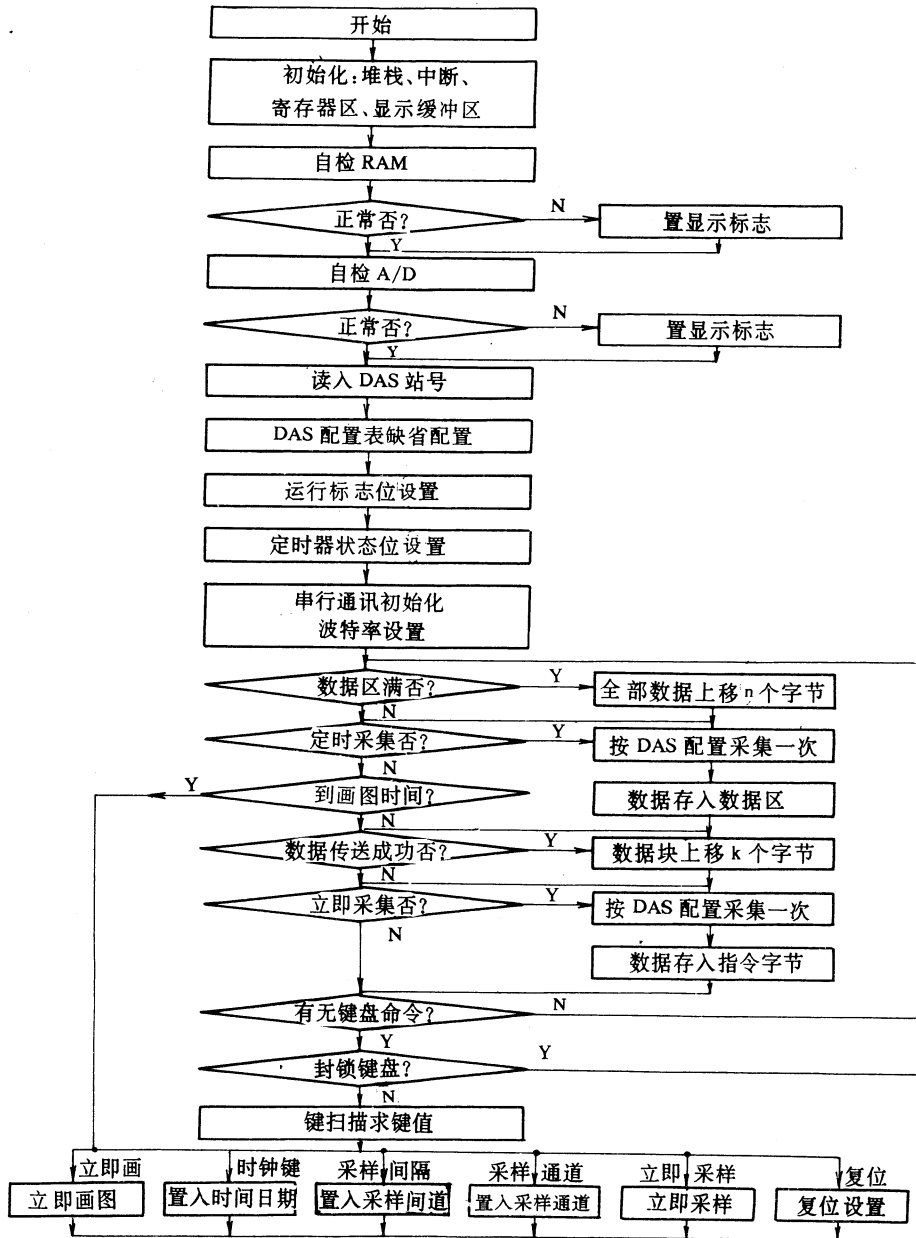
② 立即绘图，当通过键盘输入规定指令后，绘图程序自动将当前时刻以前的数据（以 10 时为界）绘制成图。

仪器还具有通信功能。利用 80C31 的 PS-232C 通信口，以通信协议方式，接收中心站的各项操作指令并自动进行仪器操作，将仪器采集的各类电压数据传送给中心站。

仪器的人机界面由 18 个键盘、6 位 LED 和多个发光二极管组成。可输入控制参数、时间码，显示工作状态、时间服务等。

为实现该机功能和缩短研制周期，我们选择了由 80C31 单片机构成的 BJ10C 应用模板（北京市单片机应用技术协会产品），该板 CPU 主频为 6MHz，RAM 为 16KB，I/O 接口及通信接口等为实现智能化的采集、绘图和通信提供了完备的技术条件。

二、仪器的主要功能特点



UN-1A 型电压模数字采集器主程序框图

1. 测量动态范围大(分别为 0—200mV, 0—2V, 0—10V), 测量稳定性好, 准确度高。
2. 具有高密度采样并绘图、多通道采集并打印及通信功能, 可以实现数字通信。
3. 具有按一定电压比例自动调整绘图范围的能力。
4. 具有时间服务系统以及人机对话的良好界面。
5. 低功耗、低成本。
绘制模拟曲线图、打印数据和串行口通信等功能

并存是该机的主要特色。这一特点不仅使台站观测人员及时发现前兆观测方法的突发性异常变化, 而且经过数字传输又使中心站的分析人员同时得到各站观测方法的异常变化, 可及时进行地震预报的分析研究工作。由于功耗低, 可缓解野外供电的压力, 达到连续观测不丢数的目的。

三、主程序框图及通信协议

①采集器(DAS)与中心站(CC)无线通信协议

DAS 端接收: ←[ENQ+命令+参数]←F...ENQ CC 端发送

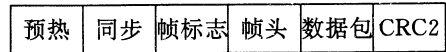
DAS 端发送: ACK...F→[数据+ACK]→CC 端接收

DAS 端接收: ←[ACK+EOF]←F...EOF CC 端发送

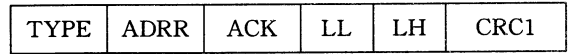
DAS 端发送: ACK...F→[ACK]→CC 端接收

②中心站(CC)发送命令约定

帧格式:



帧头:



功能说明	命令代码	DAS 响应
读取 DAS 时间	01H	读:一握时发送 DAS 现行时间
修改 DAS 时间	02H	修改:一握时发送 DAS 现行时间及新设置的 DAS 时间。 二握时将新设置的 DAS 时间代替 DAS 的现行时间。
读取 DAS 配置	03H	读:一握时发送 DAS 配置表。
修改 DAS 配置	04H	修改:一握时发送 DAS 新配置表 二握时将新配置表代替原配置表。
读取 DAS 正常数据	05H	一握时将 DAS 数据编成规约格式发送。 二握时在有效传送下,做指针修改等。
命令 DAS 立即采样	06H	一握时按配置表采集数据并缓存。 二握时无实质动作(即等待)。
收集 DAS 立即数	07H	一握时将缓存数据立即传送给 CC 端。 二握时将数据清零,恢复原状态。
命令 DAS 复位	08H	一握时通知 CC 已准备复位(标志为“1”) 二握时启动 DAS 做复位动作。

给 PC 机配装射频调制器

浙江绍兴市卫生学校计算机室(312000) 郭荣军

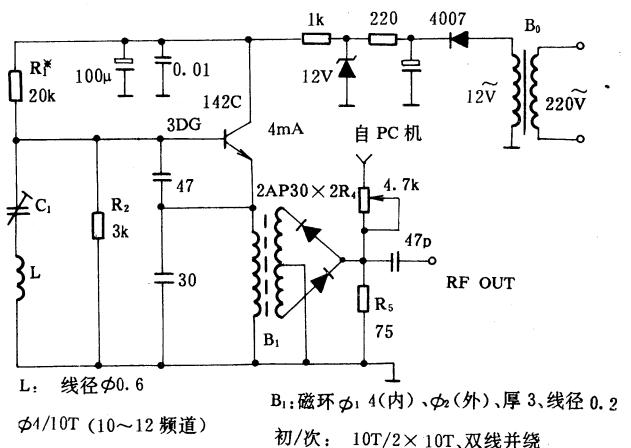
随着微电脑的逐渐普及,个人拥有量逐渐增多。个人电脑也都需要配置显示器。由于大多数家庭拥有黑白及彩色电视机,购置电脑时不必再配置专用显示器,以免重复购买。我们知道有不少电脑主机输出只有视频输出插座(VIDEO OUT),必须配置射频调制器输出 RF 信号才能用家用电视机兼作电脑显示器。用一只自制的射频调制器即能省去一笔不必要的费用。

①调制线路:如图所示,改变 L 圈数或调整 C₁ 可使射频频率在 1~12 频道范围内变化。改变 R₄ 阻值能使调制度适中。

②元器件选择:晶体三极管起高频载波振荡作用,可选用 3DG142C、3DG56B 等高频管,调整 R₁ 使 I_c ≈ 4mA 即能稳定工作。电源变压器可选用录音机(12 伏)小型电源变压器。其它元器件选择无特殊要求,有关参数如图示。

③安装及调试:由于电路简单,有兴趣的可自行设计印刷板线路并把所有元器件按如图示线路装焊上去。然后把电路板紧固在小型铝盒中,并在微调电容 C₁ 上方钻一小孔。最后在铝盒两个侧面分别加装视频输入及射频输出插座。所有制作完成后可用两根射频电缆把电脑与射调器及电视机连接起来,并接通三者各自电源,同时应使电脑输出字符信息。接着转动黑白机频道选择开关或调节彩电 VHF(V_L 或 V_H)频道调

谐电位器,使电视机接收到电脑输出的字符信息。最后用小型无感起子调整微调电容 C₁(通过上方铝盒小孔),使显示信息清晰稳定。如缺少图像层次感或图像失真可调整 R₄ 得到一适当的调制度。只要所有元器件接线无误及连接正确一般都能获得满意的显示效果。调试制作都较方便。



谈谈 LQ1600K 打印头的维护及短(断)针利用

江西新全市工商银行科技处(336525) 刘 斌

由于 LQ1600K 打印机的低价格,高效能,愈来愈受到用户的青睐。但随着使用时间的增长,针头逐渐磨损,从而影响打字质量,而更换每针需化费几十元。因此维护好打印头及合理回收短(断)针,是每个用户非常关心的事情。

一、打印头的维护

断针通常是由于灰尘或油污影响,使导向孔内出现堵塞,阻碍打印针的出针,引起断针发生。因此首先要保持使用环境的洁净度,还应做到定期清洗打印头。

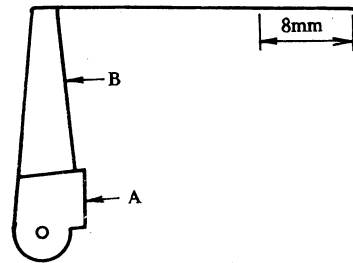
1. 静态清洗。将打印头从字车上取下,拔去电缆线,将针头部分浸入无水酒精中,用手轻轻摆动打印头。若针孔堵塞严重,浸泡时间相对延长,但应注意不要使酒精渗透至打印头线圈,以免发生短路。

2. 动态清洗。关机,给打印机装好纸,取下打印头,(不要拔去打印电缆)用点水缸盛半缸无水酒精,将针头部分接触酒精,按住换页键不放,开启电源,让打印机进行长自检。此法清洗效率较高。

选用优质色带,可以有效地防止断针。劣质色带,易起毛卡针。对作打字排版用的用户最好将蜡纸中的皱纹纸拿掉,换上一张 B₄ 版面的白纸,调节好纸厚调整杆位置,可有效防止卡针、断针的发生。

二、短(断)针利用

打印针断了,一般是购买同型号的新针,细心地拆开打印头,取下相应的断针换上新针即可。但为了节省费用,可采取下面方法。



1. 磨损针的利用。对磨损不算严重的针,可用细砂纸轻轻磨如图所示的 A、B 两平面,装上即可使用。

2. 长针作短针用。将磨损较严重的长针,在 C 位置用夹线钳剪去约 8mm,在短针位置装上,装好打印头,用金刚钻锉刀锉平即可。

(上接第 41 页)

接扩流管(见附图)便可解决问题。对于使用 80 AH 以上的蓄电池除要并上扩流管外,还要更换成整流电流大于 10A 的整流桥堆,用于充电用的充电源也不能使用原机的主变压器的充电绕组(电流 $\leq 6A$),要单独使用功率大于 400W 次级电流大于 10A、电压为 27V 的充电变压器,该充电变压器可安装在原机安装蓄电池的地方。用于隔离用的二极管 D 也要同时更换。

3. 后备式的 UPS 不间断电源致命的弱点就是易损坏逆变输出大功率晶体管。损坏的主要原因是:反激脉冲和瞬间尖峰脉冲的作用而使大功率晶体管击穿损坏。为此,必须尽量减少反激脉冲和尖峰脉冲对晶体管的影响。笔者使用以下方法较有效地解决了大功率输出晶体管经常损坏的问题。a. 在逆变输出晶体管的 C、E 极并接上阻尼二极管和阻尼电容,电容可取 40 至 50np 左右。b. 为了排除瞬间尖峰脉冲对晶体管的损坏作用,在市电供电时(即后备状态)要切断主变压器

逆变输出绕组的脉动电压对输出晶体管的反作用。逆变输出晶体管的接通与切断可使用继电器来控制,因这种不间断电源逆变输出是采用晶体管对称输出方式,故需用两个继电器 RY_a、RY_b 分别控制或使用一个双组的继电器来控制。为使得 UPS 不间断电源在市电中断时能及时逆变输出交流电压,为此,必须保证逆变输出晶体管的接通与断开同市电——逆变输出控制继电器 RY₁ 保持同步。因而,控制逆变输出晶体管通、断用的继电器(RY_a、b)的控制电源必须同用市电——逆变输出控制继电器的控制电源。控制逆变输出晶体管用的继电器可单独安装在一自制的电路板上,通过导线连接到相对应的电路上即可(见附图)。

以上是后备式 UPS 不间断电源的改进方法,对于在线式 UPS 不间断电源,因它在正常工作时都为逆变输出状态,对它的改进就显得更容易了,除了要增加阻尼元件外,就是改进一下蓄电池的充电回路就行了。同行们若有兴趣不妨一试。

后备式中小型 UPS 不间断电源的改进

广西玉林农业银行电脑网络中心(537000) 谭巨高

目前中小型 UPS 不间断电源在市场的份额较大,基本在各网络站、点上都配置有一台 400—600W 的 UPS 不间断电源,以保证微机工作站的正常运行。

现在使用的中小型 UPS 不间断电源大多为后备式,它的优点是体积小、重量轻、安装容易、价格适中、逆变输出方波纯净,是一种较理想的微机后备电源。但是这种不间断电源的不足点是:逆变输出时间短(一般最多能支持 15 分钟)和逆变输出功率晶体管易损坏。对于逆变输出时间短的问题,在电力不足的地区,要使计算机能正常运行,还需解决微机的电力问题。解决微型计算机用电问题可采用以下三种办法:

1. 全部更换成为长时间不间断电源。采用这种方法浪费大、耗资大。现在的长时间不间断电源价格都在万元至几万元间,按 20 个点计算,耗资达二十几万元到百万元之多,这办法是不可取的。

2. 各网点另外配置小型发电机以备停电时使用。这个办法就我们使用的经验来看也不可取。原因是,小型交流发电机使用两年后,机器老化,油路不通,发电不正常,电压不稳定,严重影响计算机的寿命。我中心在使用小型发电机供电的过程中,出现过烧坏多台微机电源和 UPS 的现象。

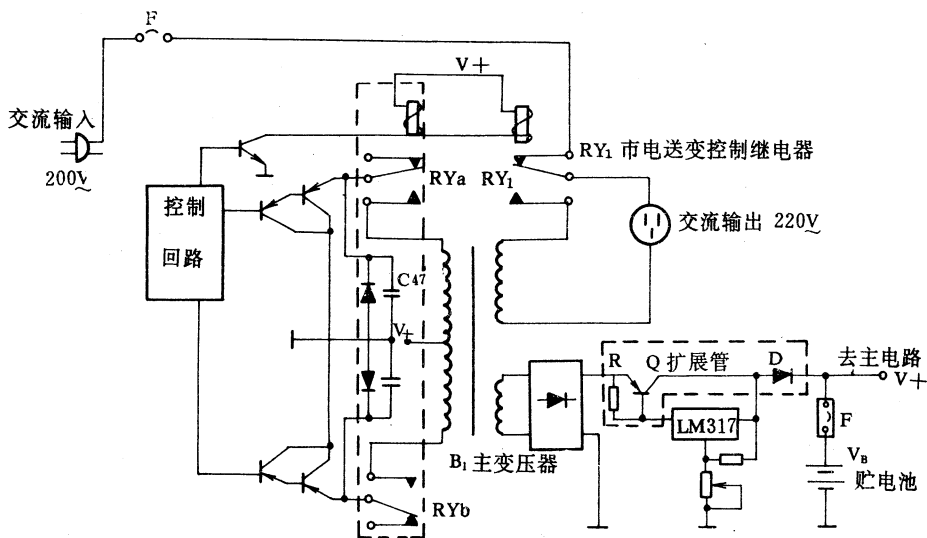
3. 改进现用的中小型 UPS 不间断电源使它成为长时间不断电电源。

综合各因素来考虑,要做到经济、方便实用、维护

容易、不浪费现有的 UPS 不间断电源,采用第 3 种方法对现用的中小型 UPS 不间断电源进行改进是很有实用价值的。对此,笔者自去年以来就对多台现用的后备式不间断电源进行了改进(有 SENDON 600 型 SANTAK 500 型 HWADAR 600 型等)改进后的 UPS 不间断电源有部份使用已有一年,现在运行正常无故障,这说明改进是成功的。下面就改进方法简述如下:

1. 更换原机使用的蓄电池,改换成大容量蓄电池。换用的大容量蓄电池可用汽车使用的蓄电池,最好还是使用封闭式蓄电池。电池要单独安放在一铁制的电池箱中并加固,在电池箱上要安装上直流保险(30A)、接线柱等,有条件的最好能装上一只 50V 的电压表和一只 $\pm 30A$ 的双向电流表,以便观测蓄电池的电压和充放电电流。根据笔者试验:在一般情况下(一台 286 主机、一台单显和一台九针打印机)配用 80 AH 的蓄电池能连续逆变输出 5 至 6 小时,配用 100 AH 的蓄电池能连续逆变输出 7 至 8 小时。用户可根据实际需要配置。

2. 更改蓄电池充电回路,增大蓄电池充电电流。UPS 不间断电源均采用浮动充电方式,但是原机配置的 LM317 三端稳压器最大充电电流只有 1.5A,更换大容量蓄电池后必须增大充电回路的充电电流能力。对于使用容量在 70 AH 以下的蓄电池在 LM317 上并



虚线部分为增加改进部分

附图: 后备或 UPS 不间断充电回路送变输出简图

FBASIC 语言的游戏程序编程技巧

第二章 6527 CPU 的显示系统(上)

山东苍山机械电子化学工业局(277700) 于 春

三、cpu 与 ppu 的分工与合作

游戏卡或学习卡中一般最少有两片 EPROM, 一片为 $8 \times 8K$ 的 EPROM2764, 用于存放图形库中的 256 个卡通图形和 256 个背景图形块。由于 2764 EPROM 用于存放图形数据故通常称其为 VROM。另一片为 $8 \times 64K$ 或 $8 \times 32K$ 的 EPROM 27512 或 27256, 用于存放游戏管理程序或 FBASIC 工作系统软件。对于更大的软件则通过门电路进行存储体切换。

游戏机工作时, cpu 执行 ROM 中的游戏管理程序, 不断处理与游戏节目有关的数据, 同时把音频数字信号经 cpu 内部的 D/A 转换器变为模拟信号, 由 1、2 脚输出。两路信号经 R2、R5 相加, 成为游戏伴音模拟音频信号送往射频调制器和音频输出接口。

6528 ppu 有八条数据线 D0~D7 (ppu 的 2~9 脚), 挂在 cpu 的数据总线上, ppu 就是通过它们接收 cpu 和存储器的数据, 交换信息的。ppu 有三条地址线 A0~A2 (ppu 的 12、11、10 脚) 与 cpu 地址总线相连 (cpu 的 4~6 脚), 用于接收 cpu 发出的地址信号 (即显示软开关命令)。ppu 还有八条数据/地址复用线 AD0~AD7 (ppu 的 38~31 脚) 和 IC6 VRAM (由 ppu 管理的 RAM、即存储显示图像数据的 RAM, 称为 VRAM), 与游戏卡中的 VROM 的八条数据总线相连, 同时通过地址锁存器 IC5 (74LS373) 把 AD0~AD7 上的低八位信号锁、存到地址总线上, 并与 ppu 的 PA8~PA13 (ppu 的 30~25 脚) 共同组成 14 条地址总线, 用于对 VRAM 和 VROM 寻址。

游戏机工作时, cpu 通过 ppu 的 13 脚 (\overline{CE} 端) 选通 ppu, 然后通过 R/\overline{W} 信号 (由 cpu 的 34 脚发出) 控制 ppu 的 1 脚进行读/写操作。ppu 处理 cpu 发出的命令, 一方面产生显示器 (电视机) 所需要的行、场同步信号; 一方面对 VRAM 或 VROM 进行读、写操作。操作过程是: 首先按 cpu 的指令接收从 cpu 数据总线传来的一幅图像的数据, 并存放在 VRAM 中指定的单元中; 然后在电视回扫描期间, ppu 把 VRAM 中的图像数据取出, 从 VROM 中查出与图像数据对应的图形块数据进行变换、配色、编码, 产生复合视频信号从它的 21 脚输出。三极管 Q₁ 把 ppu 输出的视频信号放大后送射频调制器和视频输出接口。

从以上讨论可见, cpu 颇象大公司的董事长, 而 ppu 则酷似公司的总经理。董事长仅负责制定公司的经营方针、政策, 而方针、政策的贯彻实施则由总经理

负责。cpu 与 ppu 之间的关系正是领导与被领导的主从关系。工作中, cpu 处理发声和控制卡通运动; ppu 管理背景图像的位移和更换; 从而实现了背景、卡通、音响三者同步工作的功能。

四、6527 cpu 工作系统对 RAM 的使用约定

电脑游戏机中一般有两片 RAM: 一片位于游戏机主电路板上, 即电路原理图中的 IC3, 它的存储容量为 2K 字节, 地址分配为 $\$0000 \sim \0700 , 由于 cpu 工作中经常使用这些单元, 所以我们称它为系统 RAM; 另一片 RAM 位于学习卡中 (一体化的键盘当然在主机内), 用于存放用户程序, 我们称它的用户 RAM。

当我们编写机器语言程序时, 要经常使用系统 RAM 暂存数据, 尤其是系统 RAM 的零页 ($\$0000 \sim \$00FF$) 由于数据存取速度快而使用更频繁。但是, 6527 cpu 工作系统对系统 RAM 进行了明确分工, 对某些单元进行了特殊约定。特殊约定的 RAM 单元不允许用户使用, 否则将造成系统工作紊乱, 严重时将造成死机。因此, 在编程前, 必须了解系统 RAM 中哪些单元不能使用, 哪些单元可以有条件使用, 哪些单元可随意使用。只有心中有数, 才能避免与系统工作的冲突, 使程序正常运行。

1. 对零页 RAM 单元的约定

- * * $\$03$ 、 $\$04$: 自由 RAM 区末址存储单元。
- * * $\$05$ 、 $\$06$: BASIC 程序区首址。
- * * $\$07$ 、 $\$08$: BASIC 程序区末址。
- * * $\$09$ 、 $\$0A$: 被执行程序的地址。
- * * $\$0B$ 、 $\$0C$: 当前程序行行号暂存单元。
- * * $\$0D$ 、 $\$0E$: 下一程序行首址。
- * * $\$11 \sim \16 : 按 STOP 键时的工作单元。
- * * $\$17$: 堆栈指针。
- * * $\$18$: 读键工作单元。
- * * $\$19$ 、 $\$19$: 指令入口地址暂存单元。
- * * $\$1D$ 、 $\$1E$: BASIC 变量区首址。
- * * $\$1F$ 、 $\$20$: BASIC 变量区末址。
- * * $\$21$: 七页自由区首址。
- * * $\$24$: CONT 指令有效标志。
- * * $\$27$: 变量类型标志暂存器。
- * * $\$28 \sim \29 : 第一操作数暂存器。
- * * $\$2C \sim \$2E$: 第二操作数暂存器。
- * * $\$30$: 四页自由区首址。
- * * $\$31$ 、 $\$32$: 显示页面标志单元。
- * * $\$33$ 、 : $\$2001$ 软开关状态单元。
- * * $\$34 \sim \38 : 乘、除运算工作单元。

- \$ 3D~\$ 41:运算结果暂存单元。
- \$ 44、\$ 45:程序转向的目标行号。
- * • \$ 48 :光标的水平座标设定单元。
- * • \$ 49 :光标的垂直座标设定单元。
- \$ 4D、\$ 4E:三页地址指针。
- * • \$ 4F:背景显示页面在 ppu VRAM 中的地址高位。
- \$ 52~\$ 55:在间接变址寻址中设置转移入口。
- \$ 58、\$ 59:暂存当前处理指令的地址。
- * • \$ 5A:三页自由区首址。
- \$ 5B、\$ 5C:暂存执行程序行地址。
- * • \$ 61:光标形状设定单元。
- * • \$ 62:记录本次存入七页的数据个数。
- * • \$ 63:系统软开关。
- * • \$ 66~\$ 68:功能键工作单元。
- * • \$ 69:画面纵座标设定单元。
- * • \$ 6B、\$ 6C: \$ 48、\$ 49 值暂存单元。
- * • \$ 6F~\$ 72:读键工作单元。
- * • \$ 75:键码暂存器。
- * • \$ 76:五页自由区首址。
- \$ 7C~\$ 7F:定义数组工作单元。
- \$ 80~\$ 89:循环语句工作单元。
- \$ 90:变量类型标志单元;数值变量置 2;字符变量置 3;数型下标变量置 82H;串型下标变量置 83H。
- * • \$ 9D:画面横座标设定单元。
- \$ A4~\$ AD:卡通工作单元。
- \$ C7~\$ CD:PLAY 语句工作单元。
- \$ D5~\$ DD:动作 SP 控制工作单元。
- * • \$ E0、\$ E1:光标形状定义单元。
- * • \$ ED~\$ EF:中断程序存储单元。

以上标注了零页单元的大体分工约定。其中前面有“*”的单元禁止使用;无“*”的单元有条件使用;未列出的单元一般情况可随意使用。

从以上标注可见,6527 cpu 工作时使用了零页的大部分单元。因此我们在使用零页单元时要慎之又慎。

2. 一至七页 RAM 单元的约定。

6527 cpu 工作系统对系统 RAM 的一~七页约定如下:

禁止使用的页面:

- 一页:系统堆栈区。
- 二页:卡通定义区。
- 六页:读键、定义卡通工作区。

有条件使用的页面:

- 三页:解释命令(立即执行型语句)存储区,解释语句(暂缓执行型语句)暂存区。
- 四页:cpu 运行数据暂存区(\$ 400~\$ 47F),功能键定义区(\$ 480~\$ 4FF)。
- 七页:屏幕显示缓冲区。

3. 6527 cpu 对三、五、七页的使用

下面我们通过 6527 cpu 工作中对三、五、七页的使用介绍,找出对它们使用的条件。

七页为 F BASIC 显示缓冲区。从键盘输入程序时,每输入一个字符,则将字符的 ASCII 码送入七页,然后向系统软开关 \$ 63 单元置入 1,cpu 则自动把送

显内容送入 ppu 的 VRAM 指定的单元显示。另外,当前处理的字符值也往往暂存于七页,但是一般并不完全占满,因此我们可以使用七页的后半部分,使用前要注意检查 \$ 21 单元的数值,以避免冲掉系统数据。

五页为键盘一次输入内容暂存区(所谓一次输入系指在两次按回车键之间输入的内容)。当本次语句输入完毕,按回车键后,cpu 会自动把本次输入内容的 ASCII 码送入五页,并最后置零作为结束标志。如当我们输入下列语句时:

```
PRINT HEX $(PEEK(15))
```

存入五页的内容依次为:

```
500—50 52 49 4E 54 20 48 45
508—58 24 28 50 45 45 4B 28
510—31 35 29 29 00
```

当按回车键后,系统将按 F BASIC 程序的存储格式进行编译,即保留字用其代码代替,10 进制数则换算为 2 进制数(用 16 进制数显示),并加 10 进制数标志后存入三页,形式如下:

```
300—8B 20 DD 28 CF 28 12 0F
308—00 29 29 00 00
```

可见当我们一次输入内容不多时,可以使用三页、五页靠后的单元,相应地三页的空余单元要多一些。使用前注意检查 \$ 5A、\$ 76 单元。

4. 6527 cpu 对四页的使用

四页的 \$ 400~\$ 47F 为中间运算结果暂存区,一般可使用 \$ 440~\$ 47F 一段 RAM 区。\$ 480~\$ 4FF 为功能键定义区。每个功能键占用 16 个单元,故当一个功能键定义的字符不多时,我们可以使用该功能键剩余的单元,使用时只要与功能键定义字符隔一个单元就可以。如 F1 功能键定义为“PRINT”时,我们可以使用 \$ 486~\$ 48F 共 10 个单元。

任天堂的游戏画面之所以精彩绚丽,远远优于苹果机、中华学习机,关键在于它采用了独特的屏幕显示结构,从而使游戏画面的各种处理快捷而简便。

6527 cpu 的屏幕显示画面由四个显示页面叠加而成,自前向后依次为卡通零页、背景页、卡通 1 页,底背景页。各页的排列情况如图 1 所示。

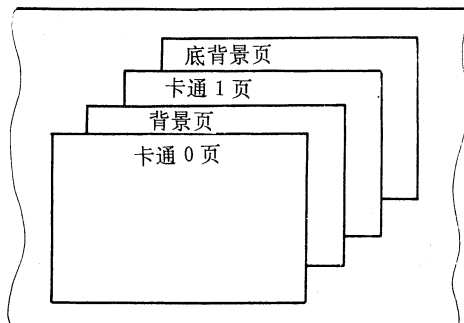


图 1 6527 cpu 显示页面结构

各个页面的显示可通过系统软开关灵活地控制。在显示图像时,靠前显示页上的内容可以覆盖后面显示页上的内容。因此,通过显示页面间的协调配合,可以形成具有一定景深秩序的画面,从而增强了画面的三维空间效果,形成立体画面结构。

一、背景画面的显示结构

背景页位于卡通 1 页之前、卡通 0 页之后,共由四个显示页面组成,它们的正常排列位置如图 2 所示。

00 页	10 页
01 页	11 页

图 2 背景页面排列示意图

正常显示时,屏幕显示背景 00 页的内容;可通过系统显示软开关控制显示任一页的内容,还可以通过系统软开关控制,以点为单位实现两个显示画面的左、右卷绕或上、下翻滚。

1. 显示图象元素与 ppu 地址的对应关系

每个背景页面都可以显示背景图案或字符。每页可显示 32 列×30 行 960 个图形块,显示方式为定格显示,由 ppu 直接控制。每个显示元素与 ppu 的地址为顺序对应。

背景 00 页的显示使用了 ppu 的 \$ 2000~\$ 23BF 共 960 个单元,显示行与 ppu 地址的对应关系如表三所示;每个显示单元与 ppu 地址的对应关系如图 3 所示。

表三 背景 00 页各显示行的 ppu 地址对应表

行次	ppu 地址	行次	ppu 地址
一	2000~201F	十六	21E0~21FF
二	2020~203F	十七	2200~221F
三	2040~205F	十八	2220~223F
四	2060~207F	十九	2240~225F
五	2080~209F	廿	2260~227F
六	20A0~20BF	廿一	2280~229F
七	20C0~20DF	廿二	22A0~22BF
八	20E0~20FF	廿三	22C0~22DF
九	2100~211F	廿四	22E0~22FF
十	2120~213F	廿五	2300~231F
十一	2140~215F	廿六	2320~233F
十二	2160~217F	廿七	2340~235F
十三	2180~219F	廿八	2360~237F
十四	21A0~21BF	廿九	2380~239F
十五	21C0~21DF	三十	23A0~23BF

背景 01 页的显示使用了 ppu 的 \$ 2400~\$ 27BF 共 960 个单元。各显示行、显示单元与 ppu 地址的对应方式同背景 00 页(图、表从略)。

其它两个页面的对应规律与 00 页相同。但是由于 ppu 的 VRAM 只有 2KB 空间,所以 cpu 自动设定 00、01 页映射到同一 VRAM 单元。即对 00 页显示单元置入数值时,可在 00、01 两页同时显示;同理,对 01 页显示单元 \$ 2800~\$ 2BBF 置入数值时,也可在 00、01 两页同时显示。同样,10、11 两页也映射到同一 VRAM 单元 \$ 2400~\$ 27BF。因此,通常把 00、01 页通称为背景零页;把 10、11 页通称为背景 1 页。特殊需要时,可用软开关把两个页面分开,写入不同的数据以显示不同的图形。(这时采用对同一 VRAM 单元交叉写入不同的数据,以使两个显示页面显示各自的内容。)

图 3 说明:

(1)图中的“列次”表示图形块或字符位于显示屏幕的第几列,“行次”表示图形块或字符位于显示屏幕的第几行。如“COMPUTER”的第一个字母“C”则位于显示屏幕的第四列、第二行。

(2)每个显示元素所对应的 ppu 行地址加列地址即为该显示元素所对应的 ppu 地址。如字母“C”的 ppu 行地址为 2020H、列地址为 03H,那么“C”的 ppu 地址则为 2023H,记为 \$ 2023。

2. 显示单元的配色

为叙述方便,我们把屏幕上的一个图像元素(一个 8×8 点阵的方块)称为一个显示单元,它是背景页面的最小显示单位。

由于每一个显示单元可以显示一个图形块,那么每个显示单元也应该有其确定的配色。也只有对每个显示单元随意置色,才能显示出五彩缤纷的绚丽图案。

6527 cpu 的配色方法独具特色,既操作简单又变化千般,是诸多微机系统中较优化的配色方法。它有如下特点:

(1)它把背景显示页的 32 列×30 行 960 个显示单元,从第一行起每上下左右四个相邻的显示单元为一组,分成 16 列×15 行共 240 个小方格,每个方格由四个显示单元组成(第 15 行每格只有两个显示单元)。每个小方格有四组颜色供选择,这四组颜色对应着 F BASIC 系统中背景配色板中的四个配色代码。由于每个配色代码由三种颜色代码组成,故每个小方格有 12 种颜色供选用(若包括底背景配色,实际有 16 种颜色)。第个小方格的四种配色代码由两位二进制数表示(即显示画面中每相邻的四个显示单元用一种配色代码)。

(2)再把 16 列×15 行 240 个小方格以上下左右相邻的四格为单位组成一个大方格。每个大方格由四个小方格组成。由于每个小方格由两位二进制数表示配色代码,故每个大方格的配色代码由四位二进制数组成,正好占用一个内存单元。我们称它为配色单元。每个小方格的配色代码在大方格配色单元中的排列方式如图 4 所示。

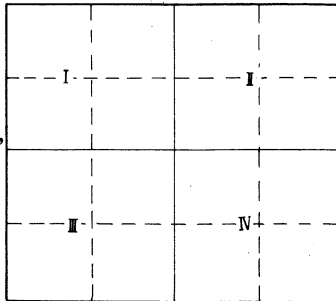
显示单元分组:

每个小方格包

含4个显示单元,

每个大方格包含

16个显示单元。



配色单元中各位
的使用情况。

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
IV		II		I		I	

图4 小方格配色代码在配色单元的排列方式

(3)通过(1)、(2)两步的处理,把一个显示页面分成8行×8列64个大方格。因此,共占用64个配色单元。背景00页的配色占用\$23C0~\$23FF 64个单元;背景10页的配色占用\$27C0~\$27FF;01页的配色占用\$2BC0~\$2BFF;11页则占用\$2FC0~\$2FFF。为便于使用,特把00页的显示单元与配色地址的对应关系示于图3,其它三页依此类推。

列次	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	行次	
PPU 行地址	2000																																1	
	2020			C	O	M	P	U	T	E	R																							2
	2040																																	3
	2060	23C0				23C1				23C2				23C3						23C4				23C5			23C6				23C7			4
	2080																																5	
	20A0																																6	
	20C0																																7	
	20E0	23C8				23C9				23CA				23CB						23CC				23CD			23CE				23CF		8	
	2100																																9	
	2120																																10	
	2140																																11	
	2160	23D0				23D1				23D2				23D3						23D4				23D5			23D6			23D7			12	
	2180																																13	
	21A0																																14	
	21C0																																15	
	21E0	23D8				23D9				23DA				23DB						23DC				23DD			23DE			23DF			16	
	2200																																17	
	2220																																18	
	2240																																19	
	2260	23E0				23E1				23E2				23E3						23E4				23E5			23E6			23E7			20	
	2280																																21	
	22A0																																22	
	22C0																																23	
	22E0	23E8				23E9				23EA				23EB						23EC				23ED			23EE			23EF			24	
	2300																																25	
	2320																																26	
	2340																																27	
	2360	23F0				23F1				23F2				23F3						23F4				23F5			23F6			23F7			28	
	2380																																29	
	23A0	23F8				23F9				23FA				23FB						23FC				23FD			23FE			23FF			30	
PPU 列地址	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F		

图3

当欲改变某显示单元的配色时,只要把配色数据写入相应的配色单元就可达到目的。因此,6527 cpu 显示配色的设置操作十分简单方便。

二、卡通显示页面的显示结构

卡通页——即游戏中角色所使用的页面,分零页和1页。显示在零页上的角色可以遮住背景画面,相当于在背景前面活动;显示在1页上的角色则相反,可被背景所遮住,相当于在背景后面活动。因此,在动画处理中,不须像其它机型那样对画面点阵进行复杂的逻辑运算就能达到十分理想的效果。

1. 卡通页的显示特点

卡通页最多可同时显示64个8×8点阵卡通(卡通块),卡通的活动范围为256×240点,每次最少可移动一点。可任意指定每个卡通左右翻转、上下颠倒等显示状态和四种配色代码的变换。

2. 卡通的显示定义区

卡通页的显示区映射在cpu内存的第二页中的256个单元。这样设计有利于对卡通运动控制的快速处理。在\$200~\$2FF中,以四个连续单元为一定义一个8×8点阵的卡通块。因此,在卡通定义区最多可定义64个卡通块。在F BASIC中一般由四个卡通块组成一个16×16点阵的角色,所以最多可显示16个卡通(8个显示卡通、8个运动卡通)。它们对应的二页地址如表四。

表四 F BASIC 中显示 SP、动作 SP 对应地址表

显示 SP 编号	定义区	动作 SP 编号	定义区
0	200~20F	0	280~28F
1	210~21F	1	290~29F
2	220~22F	2	2A0~2AF
3	230~23F	3	2B0~2BF
4	240~24F	4	2C0~2CF
5	250~25F	5	2D0~2DF
6	260~26F	6	2E0~2EF
7	270~27F	7	2F0~2FF

3. 显示卡通块的定义

定义一个卡通块要使用二页连续的四个单元。下面以第一个显示卡通块为例,介绍每个单元的含义。(第一个卡通占用 \$ 200~\$ 203)

<1>. \$ 200 单元

\$ 200 单元为卡通块显示纵座标的设定单元,取值范围为 0~255,显示有效值为 0~239。

<2>. \$ 201 单元

\$ 201 为卡通块在字库中的序号设定单元,它可以取图形库第 I 区的序号、也可以取第 II 区的序号,取值范围为 0~255。

<3>. \$ 202 单元

\$ 202 为卡通块显示状态设定单元,各位的定义如下:

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

- D0、D1:卡通块配色设定位,取值为二进制数 00~11。
- D4:卡通的显示类型标志。在 F BASIC 中
D4=0 表示定义的是显示 SP;
D4=1 表示定义的是动作 SP。
- D5:显示页面标志(或称显示优先度)。
D5=0 表示 SP 在卡通零页显示;
D5=1 表示 SP 在卡通 1 页显示。
- D6:左右翻转显示标志。
D6=0 SP 按图形库姿式显示;
D6=1 SP 左右翻转显示。
- D7:上下颠倒显示标志。
D7=0 SP 按图形库姿式显示;
D7=1 SP 上下颠倒显示。

<4>. \$ 203 单元

\$ 203 为卡通显示横座标设定单元,取值范围为 0~255。

显然,当欲在屏幕上某位置显示某个卡通块时,只要向有关地址单元写入相应的数据就可任意显示。

例 1 在屏幕的(100,150)位置显示黑体数字“4”。

若定义 0# 显示 SP 的第一个卡通块,则置入如下数值

\$ 200=96H(150 的 16 进制数为 96)

\$ 201=F3H(卡通块“4”的序号为 16 进制数 F3)

\$ 202=0(设定正常显示状态,0 号配色)

\$ 203=64H(100 的 16 进制数为 64)

相应的 F BASIC 程序为 No. 2-1

```
5 REM No. 2-1
10 SPRITE ON
20 POKE 512,150 (200H 的 10 进制数为 512)
30 POKE 513,243 (F3H 的 10 进制数为 243)
40 POKE 514,0
50 POKE 515,100
60 END
```

当欲使它左右翻转时,则把 40 句改为

```
40 POKE 514,64
```

就可实现左右翻转。

同样,若欲使它上下颠倒或上下颠倒、左右翻转,则把 40 句分别改为:

```
40 POKE 514,128
```

```
40 POKE 514,192
```

若把程序 No. 2-1 改写为 No. 2-2 将会看到数字“4”反复地显示四种状态,顺序为:正常显示、左右翻转、上下颠倒、上下左右翻转颠倒。

```
5 REM No. 2-2
```

```
10 SPRITE ON
```

```
20 POKE 512,150
```

```
30 POKE 513,243
```

```
40 POKE 514,A
```

```
50 POKE 515,100
```

```
60 A=A+64:IF A>192 THEN A=0
```

```
70 PAUSE 50:GOTO 40
```

仔细观察将会发现:数字“4”的四种显示状态尽管反复变化、姿态各异,但其右下角的位置恒定不变。这说明 \$ 200、\$ 203 中指定的座标点作用在 SP 块的右下角。这一点,为我们在程序设计中确定角色的正确显示位置提供了遵循准则。

4. 角色的定义方法

BS. 2A 版本图形库中定义的角色,一般为 16×16 点阵,即由四个卡通块组成一个角色。这四个卡通块的排列顺序如下:

1	2
3	4

显然,当定义一个 16×16 点阵角色时,四个卡通块的座标是不同的。工作系统规定:16×16 点阵 SP 的基准座标仍为第一个卡通块的右下角,其余三个卡通块的座标则以第一个的座标为参考点相应调整。因此,当第一个 SP 块的座标为(100,150)时,第二、三、四号 SP 块的座标依次为(108,150)、(100,158)、(108,158)。即以一号 SP 块的座标为 X、Y,二号 SP 块的 Y 座标不变、X 座标加 8;三号 SP 块的 X 座标不变、Y 座标加 8;四号 SP 块的 X、Y 座标都加 8。

现以 0# 显示 SP 的定义为例,介绍各单元数据的设置。(未完待续)

传真机的日常维护

张建军 张景生 李晓中

传真机结构较为紧凑,自动化程度较高,搞好日常维护是保证传真机正常工作的重要保证。以下以UF-915型传真机为例作一介绍。

一、预防性维护检查项目

要进行预防性维护检查,必须掌握正确拆装的基本方法。

1. 盖板的拆卸

UF-915型传真机盖板分为:机壳,接收通道盖板,扫描通道盖板,前盖板,前斜盖板等,各盖板位置示意图如图1。

机壳(331)

打开接收通道(11),拧下机壳里边的两个(B)紧固螺钉(423),拧松外边四个(B)紧固螺钉,然后顺着接收通道的方向取下机壳。

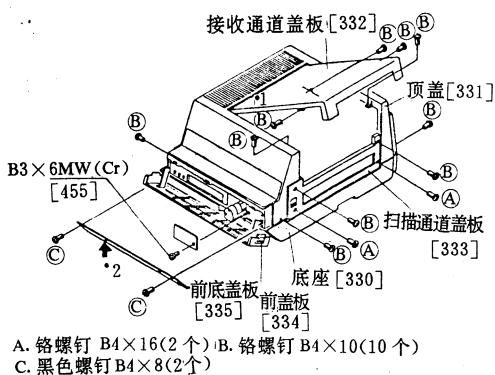


图1 盖板拆装示意图

顶盖(332)

打开接收通道(11),拧松两个(B)螺钉,斜向原稿托盘方向取出。装盖板时,应将盖板上的定位销挂好后,再拧紧螺钉。

扫描通道盖板(333)

拧下两个(B)固定螺钉和两个(A)固定螺钉(424)后,向前拉就可卸下盖板。

前盖板(334)

按着左右两边的钩键向前拉,就可把前盖放下。

前底盖板

拧下两个(C)螺钉(422),就可卸下前底盖板。在装前底盖时,应注意在挂好图中标有*2位置的定位销后,才能拧紧螺钉。

2. 主面板(18)的拆卸(参看图2)

拧下两个镀铬螺钉,向前拉开一段距离,脱开一根连接电缆和一根地线,即可取下主面板。应注意,在取下时用力不可过猛,否则易拉断连接电缆和地线。

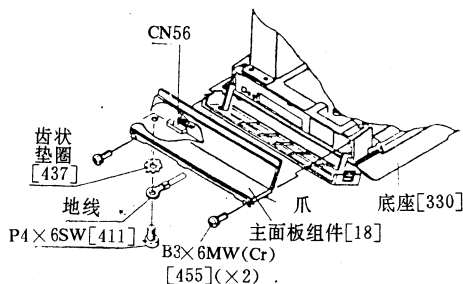


图2 主面板拆装示意图

装上时,应先将连接电缆和地线固定好后,再将主面板的定位销与底座对好、固定,然后才能拧紧螺钉。

3. 电路板的拆装

a. SC板和AMA板的拆装

拧下两个黑色螺钉(422),卸下前底盖(335),就可取出SC板和AMA板。应注意,拆装这两块电路板之前,应关断交流主电源开关。拆装SC板和AMA板可参看图3。

b. MTDV的拆装

拆去机壳,拔去MTDV板上所有的连接插头,用钳子夹住固定电路板的塑料支架,取下MTDV板。

装MTDV板步骤与拆卸的步骤相反。

c. PIX板的拆装

拉开前盖板(334);拆下荧光灯组件(3);拆下机壳(331);拆下主操作面板(18);拧下固定纸盒的两个固定螺钉,取出纸盒;再拧下固定光学部件的螺钉(428),取下MTDV板;拧下固定机架的四个固定螺钉(446),拔掉连接电缆6根,地线1根,取下机架,就可取出PIX板。

装PIX板的步骤与拆卸相反。参看图4。

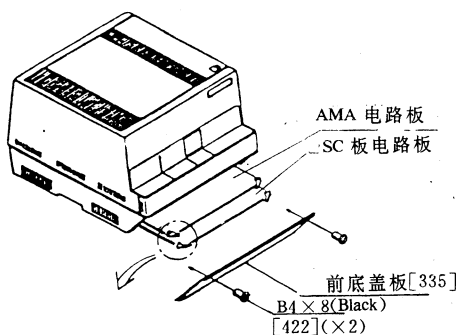


图3 SC板、AMA板的拆装

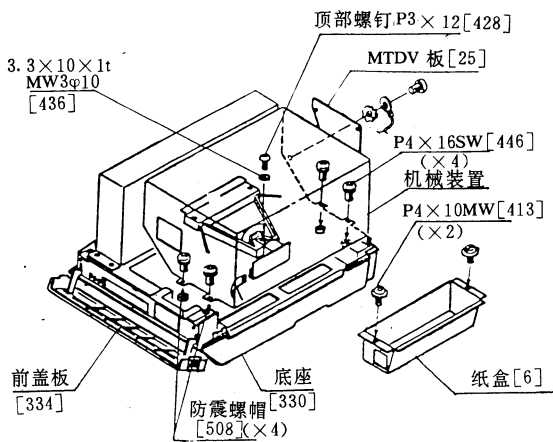


图 4 PIX 板的拆装

d. REC 3 板的拆装

拧松固定接收通道盖板的固定螺钉,取下盖板,拔去固定 REC3 板的两个塑料铆钉,拔去连接插头,就可取下 REC3 板。

日常维护的项目见表 1。维护的部位可参看图 5 表 1 日常维护项目

序号	维护部件	维护方法
1	输送橡胶(114)	用蘸有酒精的布擦洗,必要时,用砂纸打磨
2	ADF 辊(174)	如粘有纸屑,可用毛刷清刷
3	扫描辊(175)	用蘸有酒精的布擦去污物
4	记录辊(176)	用蘸有酒精的布擦去污物
5	热敏记录头(482)	用蘸有少许酒精的布擦干净记录头表面
6	纸导轨(140)	去掉夹在中间的纸屑,需要的话,可用小镊子
7	反光镜(120)	用镜头纸或软布擦去灰尘。

1. 输送橡胶片的维护

去掉文件托盘(16);拧下紧固螺钉(408),取下输送橡胶片。

检查输送橡胶的各部分,并用沾有少量酒精的布擦拭干净。

装上时,将输送橡胶的定位与固定支架对齐,再拧紧固定螺钉。

2. ADF 辊(174)

打开扫描通道(13),用手指沿纸输进方向,转动 ADF 辊,边转动边用毛刷把它清扫干净。清扫时,应小心,以免损坏 ADF 辊。

3. 扫描辊(175)

扫描辊是白色的,当扫描辊脏时,所收到的图象的两端可能变成黑色。

维护时,先拆下文件托架、扫描器盖板,拿下消静电组件(1)(5),然后,用手沿文件输进的方向转动扫描辊,在转动的同时,用蘸有酒精的布,沿纵向方向小心擦洗扫描辊。

擦洗完毕,依与拆卸相反的步骤,装上各部件。

装上后,应检查消静电组件上的分离板与扫描辊之间的配合

消静电组件上的分离板用来将发送完的文件与扫描辊分开,当分离板间隙不合适时,可能产生文件上卷。

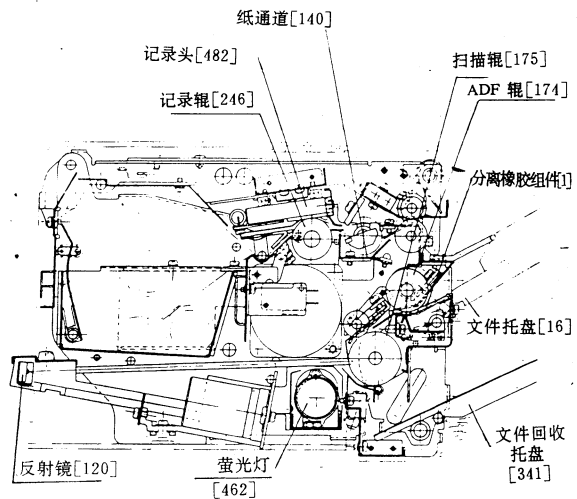


图 5 日常维护部位

4. 记录辊(176)

打开接收通道(11),用蘸有少量酒精的布,沿纵向擦拭,边擦拭边按记录纸输送方向转动记录辊。

5. 感热记录头(482)

打开接收通道,用蘸有少量酒精软布,轻轻地擦拭记录头电极表面。对残留在记录头上的脏物,应先用少量酒精浸润,然后用软布擦去。忌用金属硬物刮记录头电极表面,否则易损坏记录头。待酒精挥发后,才可盖上接收通道。

6. 记录纸输送导轨

打开接收通道,检查记录纸输送通道内有无纸屑,重点注意切纸刀(旋转片)下,如果存有纸屑,可用小镊子等工具将其取出。

7. 光学组件的维护(参看图 8)

一般情况下,光学组件不需要擦拭,只有在出现图象底色变深,或局部变黑的环境下,应擦拭光学组件,重点是反光镜。

维护方法:打开接收通道,拧下两个固定纸盒的螺钉,拧下反光镜盖板上的两个白色螺钉,取下盖板,用镜头纸或软布擦去反光镜上的灰尘。必要时,也可用镜头纸或软布擦拭镜头。

应注意,光学组件不应轻易地调整,尤其应注意光学组件与 CCD 的相对位置。

通信软件的使用

小李:申老师,我按照上两次介绍的方法,一步一步往下做,最后做通了,和驻北京办事处的 PC 机联上了,进行了“屏幕交谈”,我们都兴奋得很。我们原以为电脑远程通信很复杂,没想到几个 BASIC 语句就解决了问题。但是文件传送的问题还是没能解决。

申老师:我们只是学习了几个最简单的控制 MODEM 进行工作的“AT”命令,实际上“AT”命令相当丰富,有一个“AT”命令集,凡是标明与美国 Hayes 兼容的 MODEM,都可以使用这些命令来控制它的工作。这些命令在 MODEM 卡的“使用说明书”上都有详细的介绍。

不管如何,抓住 MODEM 通信的基本工作过程是很重要的。在呼叫方是:拨号→连通→传送→退出→挂机;在应答方是:置自动应答→连通→传送→挂机(自动或手动)。

关于文件传送,由于程序写起来复杂一些,所以建议大家使用成熟的通信软件。国际上商品通信软件有

很多种,其中 PC-Talk、CrossTalk、ProComm、Qmodem、SmartCom 等都是国际上比较流行的,Bit-Com、Telix、Boyan 等港台地区也用得不少。最近国内中国计算机软件与技术服务总公司(CS&S)研制的 CPCCOMM 微机通信软件,据报导也达到了国际同类软件的先进水平,而且更适合中文环境。这许多软件虽然各有优缺点,但其基本功能甚至基本的操作方法却大同小异。这里我打算选择在我国流行较早、使用比较简便的 ProComm 来介绍一下。读者手头如有其他通信软件,亦可以举一反三,参照使用说明书来正确使用它。

将载有 ProComm 软件的磁盘(可以是一个 360KB 低密盘)插入 PC 机的 A:驱动器,启动该程序:
A>PROCOMM

按下一二个空格键,再同时按下 Alt 和 F10 键(此处称为 Alt-F10 命令),就可以看到主命令菜单如图 1 所示,也称联机帮助屏幕。

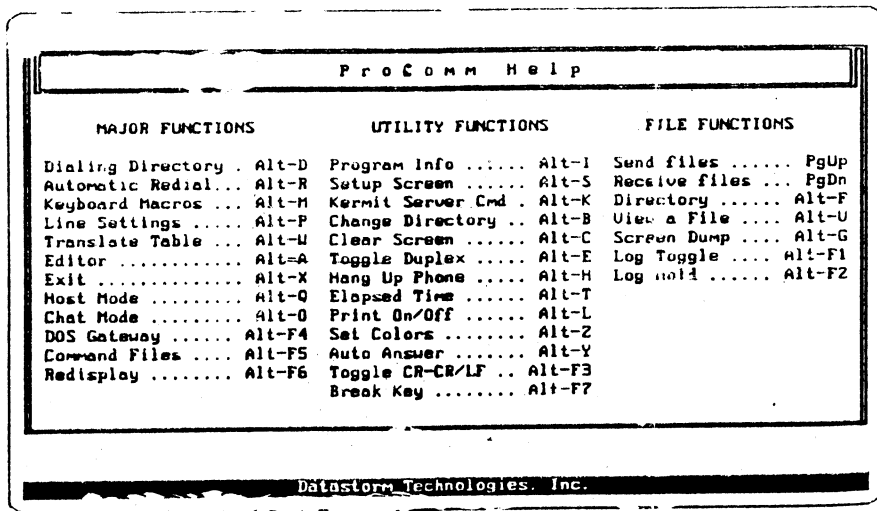


图 1 ProComm 帮助屏幕

命令有 30 多个,和其他通信软件一样,它包括了设置通信参数、拨号、应答,选择“交谈”方式、文件传送、显示参数、记录联机活动、编辑器、DOS 存取等功能。有些通信软件还能使 PC 机仿真大型计算机的终端,例如仿真 DEC 的 VT-100 和 IBM 的 3101 终端,使 PC 机一机多用。

小李:那么多命令,又全是英文提示,一下子还真搞不明白呢!

申老师:学习总是先从简单的入手,我们先抓住几

条基本命令,弄通以后再“扩大战果”。我们按下面的顺序进行操作,当然要注意分开呼叫方和应答方,注意双方时间上的配合。

1. 从命令表中找到以下几个最基本的操作命令:
 - Alt-P 查看或修改通信线路的参数设置
 - Alt-D 自动拨号
 - Alt-Y 自动应答
 - Alt-O “交谈”方式
 - PgUp 发送文件

PgDn 接收文件
Alt-H 挂机
Alt-X 退出 PROCOMM

2. 查看或修改通信线路的参数设置 Alt-P(双方)

显示“当前设置为 1200,N,8,1,COM2,”(如与本机的实际设置不符,可进行修改,修改后由 Save 项存储),用 ESC 退出。

3. 如准备接收别人的信息,可按 Alt-Y,此时显示 ATSO=1,表示响铃一次即接通,当有人拨来电话时,会显示“RING”;如连通,会显示“CONNECT”。

4. 如准备发送信息给别人,请按 Alt-D,出现拨号屏幕。按 M,即进行人工拨号,输入电话号码即可;

按 R,再输入编号,即可建立电话目录;输入姓名,电话号码,通信参数(1200-N-8-1),回显功能(E-CHO—选 Y),存储后即自动返回。

在拨号屏幕下,如直接输入电话目录的编号,即可自动拨号。从 MODEM 的小喇叭可以听到拨号直至接通的过程。

(注意:有些电话交换机 PBX 不支持音频拨号,则应选用 Alt-S 改变 MODEM 设置为 ATDP,而不是 ATDT)。

通信链路建立成功之后,屏幕上会出现“CONNECT”的提示。

5. “交谈”通信方式 Alt-O(双方)

当通信链路接通之后,按 Alt-O(字母 O),即可进入“交谈”方式。屏幕下一半显示键入的信息,屏幕上一半显示接收到的信息。ESC 结束“交谈”。

6. 发送文件 PgUp(发送方)

连通之后,按 PgUp 键→出现上载(UPLOAD)小屏幕,选协议(Protocol)为 1(即 XMODEM 协议)→输入欲发送文件之文件名→出现传送监视小屏幕,可以看到预计所需的传送时间,已传送的字节数,块数及出错情况等。

传送完毕,将有 Completed 的显示。如不再传送,应使用 Alt-H 及时挂机。

7. 接收文件 PgDn(接收方)

连通之后,按 PgDn 键,出现下载(DOWNLOAD)小屏幕,其它显示和选择与“发送文件”类似。如传来的是文本文件,可以用 Alt-U 命令检查其内容。

8. 退出 Alt-X

退出 ProComm 程序,可以检查文件传送是否正确。

小李:选择文件传输协议时,为什么选择 XMODEM 协议呢?

申老师:选择其它协议也可以,但发送方和接收方必须一致。常用的文件传输协议有:XMODEM 及其变型,Kermit, YMODEM, ZMODEM 等。相比之下,XMODEM 已比较普及;Kermit 协议还支持传送文件到小型机、大型机;YMODEM 比 XMODEM 传输速度更快;而 ZMODEM 在 1986 年才被设计出来,它更加高效和可靠,可是还不够普及。我们选 XMODEM 是因为它的普及性。

小李,你们回去认真做,祝你们顺利。

NOVELL 网络的保密技术

王杰民

保密问题在象局域网这样的多用户环境中是重要的。为用户提供安全保密保障,是 NOVELL 网络 NETWARE 操作系统的最重要特色。它不仅使非法用户入网并窥视服务器变得困难,而且使那些即便已在系统中的用户察看他人的目录也并非易事。关于超级用户(Supervisor,即网络管理员)和用户(User)为什么关心网络的安全保密,有几个原因。如果没有目录和文件锁定机制,无经验的用户就可能偶然删除重要文件或招致其它类型数据残损。另一个充分理由是维护自己目录和文件的私用性,防止数据泄漏,尤其是防范工业间谍和恶作剧者。

保密问题由每个文件服务器的超级用户控制,他也可以把管理权给予其它用户。超级用户可以建立最高级目录结构,并在目录上分配权力来建立高级保密系统。

网络的保密措施应能控制下列情况:

* 谁能访问网络?

* 谁能使用网络的什么资源?

* 用户用这些资源干什么?

* 谁能在文件服务器控制台上执行任务?

为此,在 NETWARE 中设置了四个级别的保密措施:

1. 注册保密
2. 受托者权限保密
3. 目录保密
4. 属性保密

这四种措施可单独使用或组合起来使用。

注册保密

NETWARE 的注册保密特性,使超级用户可以阻止未授权用户对系统进行存取。可采用口令系统阻止注册(login)过程,未授权用户被限制不得注册进入某个工作站,或在规定的时间内,用户不得登入系统。这些安全措施可以防止用户不受监督地使用系统,超级用户可以设置以下限制:

- * 限定帐户的有效期限
- * 限定用户何时可以向网上注册
- * 要求用户设置口令
- * 允许用户改变口令
- * 限定口令字的最小长度
- * 限定只有超级用户可以修改某一口令
- * 强迫用户定期改变口令
- * 防止用户将口令改回到原来曾用过的
- * 限定用户可以注册的次数
- * 限定用户只能从指定的工作站入网
- * 限定用户在不同的工作站入网的次数
- * 限定用户可占用的磁盘空间大小
- * 追踪记录所有注册及注销(logout)请求
- * 出于安全考虑,冻结用户帐号

受托者权限保密

受托者权限保密用来控制用户对目录和文件的访问,这主要取决于超级用户为用户所建立的各种安全性权限。尽管你可以登录到某个服务器中,但如果超级用户或其他用户不让你成为“受托者”,即不给你访问目录的权力的话,你就无法访问文件,实际上也就无所作为。例如,你可以转到非属你自己的目录中,如果没有那个目录的有关权限时,用列目录命令查看时和一个空目录一样,你欲查看目录,必须先被赋予目录扫描权。

所谓受托者(Trustee),指财产、业务等的受托管理人。NETWARE 认为网络资源属于超级用户,超级用户可以指定某用户,授给某个目录或文件的某些权限(这个过程叫做受托者指定)。这时用户就有了受托者权限(Trustee Rights),受托在此权限下对该目录或文件工作。对用户赋予这些权力,都应逐个进行,除非他是某个小组的一员,用户组可以作为一个整体接受同样的权力,叫做组受托者权限(Group Trustee Right)。

任何受托者均可被授予以下八种权力:

1. 超级用户权(Supervisor)
2. 读权(Read)
3. 写权(Write)
4. 建立权(Create)
5. 删除权(Erase)
6. 修改权(Modify)
7. 扫描权(File Scan)
8. 访问控制权(Access Control)

这八种权力可以任意组合地授予网络中的用户(超级用户权、删除权和访问控制权的授予要慎重)。一旦用户被授予某个目录的受托者权限,这些权力就可以延伸到该目录的所有子目录,也即子目录完全继承父目录的权限,除非子目录被重新指定受托者权限。

目录保密

除通过受托者权限对某个目录中的各个用户的行为进行限制外,超级用户还可以用目录保密措施控制受托者在某个目录中行使的最大权力。目录的所有受

托者的权力均受目录保密的限制。

每个目录有一个继承权限掩码(Inherited Rights Mask)。当目录被建立时,允许用户行使所有目录权力。在某个目录中拥有所有权的超级用户或某个用户可以编辑该目录的继承权限掩码。继承权限掩码必须逐级设定,不能自动延伸到其下属的子目录中。如果某些权力从目录的继承权限掩码中被除掉了(或者说被屏蔽掉了),那么,用户就不能在该目录中行使那些权力了,即使这些用户拥有相应的受托者权限,也无济于事。例如,超级用户决定不让任何人删除某目录,于是他吧删除权从该目录的继承权限掩码中去掉。现在,即使用户有删除权,他也不能删除该目录。可见,继承权限掩码是超级用户限制所有用户对某个目录权限的便捷方法。

实施目录保密措施后,某个用户对某目录实际所能行使的有效权限(Effective Rights)就须经过计算才能得知。算法是:

$$ER = (GTR \text{ or } TR) \text{ and } IRM$$

式中,ER 有效权限
 GTR 组受托者权限
 TR 用户受托者权限
 IRM 继承权限掩码
 or “或”运算
 and “与”运算

从该式可见,要对某个目录行使某种有效权限,则必须具备以下条件:1)用户必须是相应的受托者;2)目录的继承权限掩码必须允许这种权力。

属性保密

属性保密的目的是防止文件意外地被删除或改变,它尤其适用于一些被许多用户读取的公共信息文件。一些较大的公共文件一旦被删除往往是很难恢复的。

属性安全性给各个目录或文件指定专门的性质,它不考虑受托者权限,并能阻止有效权限所允许的操作。它可阻止对目录和文件的删除、拷贝、查看和写操作,还可用于执行下列任务:

- * 控制文件是否能被共享
- * 标记正在被修改的文件
- * 保证一个文件被修改时,要么改变完成,要么不做任何改变,从而保证文件免遭破坏。

NETWARE 标记的目录属性有 5 个:D(删除禁止)、H(隐含)、P(清除)、R(改名禁止)以及 SY(系统目录)。

文件属性有 12 个:A(归档需求)、C(拷贝禁止)、D(删除禁止)、X(只执行)、H(隐含)、I(索引)、P(清除)、RO 只读、R(改名禁止)、S(可共享)、SY(系统文件)、T(事务跟踪)。

一系列菜单实用程序和命令行实用程序形成了友好的用户界面。系统设置 SYSCON(System Configuration)菜单实用程序可完成多项保密操作。超级用户用它:

对《最近出现的几种病毒》 的几点异议

(450002)河南 何崇乐

《电子与电脑》92年11期苏民生《最近出现的几种病毒》一文(下简称《病毒》),似有有妥之处;为不使贻误读者,特提出下列异议,并就正于广大同仁。

(1)关于 GenB/GenP 病毒 这是美国 MaAfee 公司对此病毒的命名;但它并不如《病毒》文所说:GenB 是对软盘而 GenP 是对硬盘而言的,正确的描述如下:

GenB 毒染硬/软盘引导区的该病毒;

GenP 毒染分区表的该病毒;

GenF 毒染文件分配表的该病毒。

其中,B、P、F 分别是英文引导区、分区表、文件分配表的字头。

(2)关于“779/5”病毒 这不是病毒,而是用中点公司解毒软件 CPAV 对 .COM/.EXE 文件作免疫处理后的结果,且原文件经此处理后,也并非“不改变原来文件内容”,仅只是毫不影响原文件的运行和功能罢了。所以,“未发现破坏情况”是理所当然的。

类似情况,对于《病毒》文的“901/910 病毒”恐怕也适用;因对于 .EXE 文件来说,以此免疫处理后,文件长度的增加量是不定的。

在当前“谈毒色变”情况下,能减少病毒误报,也应算是一个不小的问题吧;所以,这同样“需引起注意”。

- * 监督用户入网和退网
- * 生成/删除用户组
- * 分配用户到用户组
- * 从用户组中删除用户
- * 设置/修改用户组的受托者权限
- * 查看用户组的身份和控制台操作员状态
- * 为用户分配受托者权限
- * 设置/修改用户口令
- * 设置/删除用户的保密等效
- * 设置记帐限制
- * 设置用户入网的工作站物理位置的限制
- * 设置用户入网的时间限制
- * 非法者检测和锁定

通过文件管理 FILER 菜单实用程序可以:

- * 查看在当前目录中的有效权力
- * 查看/修改在当前目录的继承权限掩码
- * 查看/修改目录所有者
- * 增加/删除某个目录的受托者
- * 对目录和文件列表、删除、改名、拷贝及改变属性

总的来说,NOVELL 网络的保密技术是比较完善的。NOVELL 网之所以能击败其它对手,在局域网市场上——路领先,不是没有原因的,除兼容性、效率、可扩

充性、通信能力、系统容错技术、数据保护技术有特色之外,先进的保密技术也使它对用户有很大的吸引力。

让文本阅读器显示汉字

(430070)华中师大 许勇

Borland 公司推出的 Turbo 系统集成编译器中都配有一个文本阅读器 readme.com,在文本模式下,可以用来浏览各种文本文件,由于采用直接写屏技术,在进入汉字操作系统后就无法还行。笔者通过分析发现,只须对其作少量改动即可支持汉字显示。修改可利用 SYMDEBUG 或 DEBUG 来进行,操作如下:

```
C>copy readme.com creadme.com
c>debug creadme.com
-a92c
XXXX:092C SUB DX,010 ;DX=cursor X,Y
XXXX:0930 OR CX,CX ;CX=string length
XXXX:0932 JLE 099E
XXXX:0934 MOV BX,[0167] ;screen width
XXXX:0938 CMP CX,BX
XXXX:093A JBE 093E
XXXX:093C MOV CX,BX
XXXX:093E PUSH CX
XXXX:093F CALL 0B84 ;set cursor
XXXX:0942 INC DX ;move cursor
XXXX:0943 MOV AH,09
XXXX:0945 MOV BL,[BP+01] ;BL=display
attr
XXXX:0948 LODSB ;AL=achars ASCIIvalue
XXXX:0949 MOV CX,0001
XXXX:094C INT 10 ;display a char
XXXX:094E POP CX
XXXX:094F LOOP 093E
XXXX:0951 JMP 099E
-w
Writing 106D bytes
-q
```

修改后的 creadme.com 在配有 VGA 卡的 286 微机,在 super-CCDOS 5.0 或 CCDOS 4.0 中都运行得很好。对于配 CGA、EGA 等卡的微型机,还要对显示行数、命令行位置作些改动,以避免刷屏现象,聪明的读者也不难实现。

求购

我们气象卫星云图接收机中需要使用 LTC1064-4型滤波器,但经过一个多月到各地去购买都无货,有知信息者告知。来信地址,杭州市杭州大学地理系,邮编310028。
黄元勋

模拟乘法器 IC 及使用 方法(一)(续)

李兰友

(接上期)

(3) 转换速率的影响

AD534的小信号带宽1MHz是在 $V_{OUT} = 0.1V_{rms}$ 条件下的测得值。由于AD534输出放大器的转换速率较小 ($20V/\mu S$)，因而在大信号时带宽达不到1MHz。

转换速率SR和满功率频率F的关系为

$$F = SR / (\pi \cdot V_{P-P})$$

V_{P-P} 是输出电压的峰—峰值。当 $V_{IN} = 7V_{rms}$ 时的频带宽F约为300KHz。因此，频率越高，输出电压越大，线性度越差。

(4) 馈通特性

AD534的X、Y输入的馈通很小，即使不做失调电压调整，当 $V_{IN} = 7V_{rms}$ 时，馈通约为 $1mV_{rms}$ ，进一步减小馈通可对X输入和Y输入的失调电压进行调整，调整电路如图9所示。

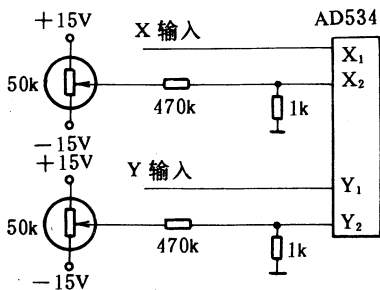


图9 X、Y输入失调调整

3. 标度因子的变更

AD534的标度因子(SF)由内部调整为10V。但在具体应用中，标度因子尽可能接近于满量程。由于输入电压并非总是在10V左右，因而常需要变更标度因子。

变更标度因子有两种方法。方法之一是在SF引脚外接调整电路，其参考电路如图10所示。

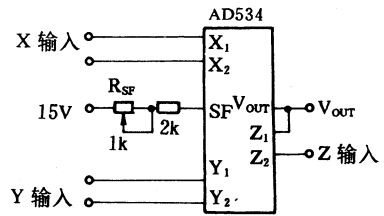


图10 变更SF的电路

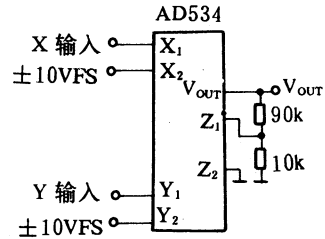


图11 反馈法变更SF

本电路的SF值可在3V~10V之间进行调整。电阻 R_{SF} 的值由下式计算：

$$R_{SF} = 5.4k\Omega \cdot SF / (10 - SF)$$

如果设定 $SF = 3$ ，则可求得 $R_{SF} = 2.3k\Omega$ ，选 $R_1 = 2k\Omega$ ， $V_{R1} = 1k\Omega$ ， V_{R1} 阻值选择的原则是可使SF在±25%内可调。

变更SF后，输入电压的满量程也应随之改变。SF = 10V时最大输入电压为±12V，那么当SF = 3时，输入满量程电压应为 $1.25 \times SF = 3.75V$ 。

利用反馈法变更SF的电路如图11所示。

从输出到Z1之间介入一衰减器进行反馈，可在很宽的范围内设定SF。

(5) 噪声

噪声电压左右着乘法器的动态范围，因而噪声电压是乘法器的一个重要指标。AD534的噪声电压密度为 $0.8\mu V_{rms}/Hz$ ，当带宽为1.3MHz时，噪声电压约为 $1mV_{rms}$ 。

二、AD535

AD535是可确保优良除法特性的乘法器IC。AD535引脚排列及功能与AD534完全相同。AD535的主要参数如表3所示。

表3 AD535主要参数

型号	传递函数	总误差 ($X=1\sim 10V$)	总误差 ($X=0.2\sim 10V$)	温度系数1 ($X=1\sim 10V$)	温度系数2 ($X=0.2\sim 10V$)	电源电压/电流 (V/mA)
AD535J AD535K	$10V \frac{(Z_2 - Z_1)}{(X_1 - X_2)} + Y_1$	1% 0.5%	5% 2%	0.01% C	0.05% C	($\pm 8 \sim \pm 18$)/ 6(max)
AD535J-K	$\sqrt{10V(Z_2 - Z_1)}$	0.4%	0.7%			

AD535的实测频率特性如图12所示。当 $V_x = 10V_{DC}$ ，输出放大器增益为1时，-3dB 频率 $f_{-3dB} = 1.4MHz$ ；当 $V_x = 1V$ (或 $V_x = 0.1V$) 的 $f_{-3dB} = 90kHz$ (9kHz)，与 AD534大致相同。

当动态范围为50倍时 AD535J 的总误差为5%。为改善总误差，可采用图13所示的电路，对 X 输入和 Z 输入失调进行调整，使总误差降到2%以下。

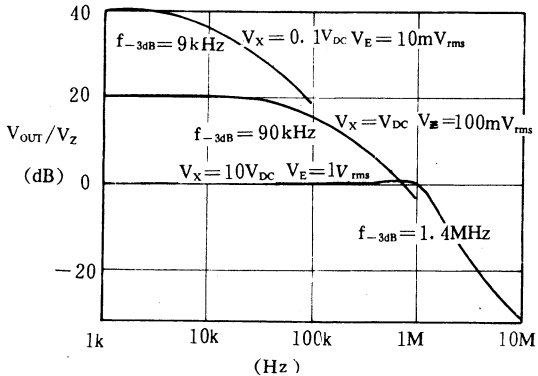


图12 AD535实测频率特性

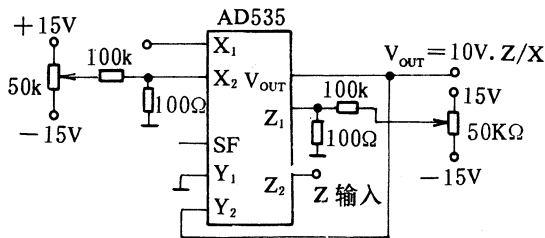


图13 改善 AD535 总误差电路

节能的 IC 小电路

向心

电池在日常生活中和我们关系密切。但靠电池工作的仪器常常会出现这样的现象：电池电压略低于额定值时，仪器就“罢工”了。这样的电池常被扔掉，而实际上它还有30%的容量。如果采用我们介绍的电路，则能“吸干”剩余的能量。专用 IC (图1) 可以满足上述要求，它可分为几个部分：开关晶体管，振荡器，参考电压源和控制逻辑。只需几个外接元件就能构成完整的电路。

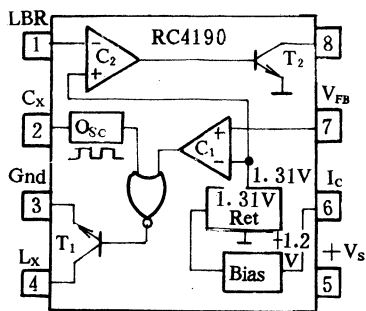


图1

RC4190的重要数据：

输入电压：最小2.2V，最大24V

输入电流 ($I_4=0$)：典型215μA，最大300μA

内参考电压：最小1.24V，典型1.31V，最大1.38V

开关电流 ($V_4=400mV$)：最小100mA，典型200mA，最大375mA

输出电压：1.3V~30V 可调

最大输出负载：400mW

最大功耗：468mW

效率：典型85%

电压调整率 ($V_5 = \pm 0.5V_{OUT}$, $P_L = 150mW$)：典型0.2%，最大0.5%

工作频率：最小0.1kHz，典型25kHz，最大75kHz

温度范围：0~±70°C

电路工作原理

一个“阶跃调压器”的原理电路示于图2。开关 S 接通时，一强电流流过线圈 L，建立一磁场，二极管 D 截止，负载 R_L 从电容 C 获得能量。磁场能量取决于于电池电压 V_{Batt} 、线圈电感 L 和开关时间 T_{ein} 。流过线圈的电流 I_S 随时间线性增加

$$I_S = \frac{V_{Batt} \cdot T_{ein}}{L}$$

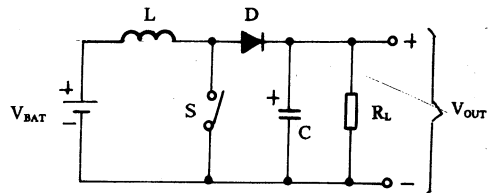


图2

开关断开时，线圈感应电压的极性与电池电压的极性相反。这个电压与电池电压串联，并经二极管 D 给电容充电。通过周期性地快速开关，电容上建立起较高的电压。开关速率必须相当高：3kHz~200kHz。

开关调压器 RC4190 的应用电路示于图3。图中的线路能使电池能量得到充分的利用。即使电池电压从9V 降至2.2V，输出电压仍为7V。注意正确选择1mH 线圈。线圈绕在环形磁芯上，并能承受1A 尖峰电流。

电池低检测作为一种选择保留。它既能指示一个固定的低电平，也能在低于这个电压阈值时实现开关功能。检测阈值 (V_{Th}) 由下式确定

$$V_{Th} = V_{Ref} (R_4/R_5 + 1)$$

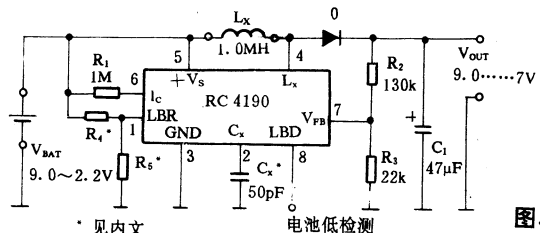


图3

例如 $R_4 = 470k\Omega$, $R_5 = 220k\Omega$, $V_{Ref} = 1.31$

$$V_{Th} = 1.31(470/220 + 1) \approx 4(V)$$

PC/XT 及其兼容机内存的扩充与使用

李晓中

PC/XT 的 CPU 具有 20 条地址线,可访问 1MB 的存储器地址,其中 DOS 能访问到的 00000—9FFFFH 共 640KB 作为 RAM 使用,而高端地址 A0000—FFFFFFH 则用于显示缓冲区及系统 ROM BIOS (或 ROM BASIC)。对于高档机如 286、386 来说,配有硬盘,内存一般在 1MB 或更大,并可建立虚拟盘,为运行大型软件建立了良好的环境。而对于没有硬盘的 PC 机来说,如果用户希望在 CCDOS 下运行一些大型的汉字软件(如 FoxBASE、CCED 和汉化 PC Tools 等),则往往出现内存不够的现象。因为在没有硬盘的情况下,汉字系统一般都被装在内存中,其中汉字库、键盘管理及显示管理模块共需占去大约 400KB 的内存空间,实际留给用户的内存空间不足 200KB。

目前市场上出售的 XT 机多为 Super PC,这种 PC 在主板上留有 RAM 扩充插座,内存可扩充到 1MB(在主板的 RAM 扩充槽中加 4 片 414256RAM 芯片,取掉原来 RAM 61464 芯片,并将主板 DIP 开关改为 1MB 内存,即可实现内存的扩充)。其中 640KB 的基本内存的地址为 00000H—9FFFFH,扩充的 384KB 的地址为 20000H—7FFFFH。PC 的 BIOS 对于基本内存与扩充内存之间的地址冲突是采用选体技术来解决的:将扩充的 RAM(最大到 384K)定为 1 体、基本 RAM 定为 0 体。通过对选体寄存器的修改,来选择使用这两个 RAM 区。选体寄存器的 I/O 端口地址一般为 0E0H,通过访问选体寄存器来进入和退出扩充内存(类似于单片机),如下:

进入扩充内存(1 体)时需执行如下指令:

```
MOV AL,01H
OUT 0E0H,AL
```

退出扩充内存返回基本内存(0 体)时需执行:

```
MOV AL,00H
OUT 0E0H,AL
```

需要说明的是,PC/XT 兼容机的选体寄存器的 I/O 端口地址不一定是 0E0H(有些为 0C0H),扩充 RAM 的起始段地址也不一定为 20000H。ROM BIOS 在诊断扩充 RAM 时给出了 I/O 选体寄存器的端口地址和扩充 RAM 的段起始地址,读者可以自行分析 ROM BIOS 程序来得到这两个参数。

目前,DOS 3.0 以上版本为 PC/AT 提供了一个用于建立虚拟盘的设备驱动程序 VDISK.SYS。由于这个虚拟设备驱动程序不支持 XT 机的选体寄存器,使得 PC/XT 用户无法建立虚拟盘。

用户可以自行编写虚拟设备驱动程序并建立虚拟盘。这样做虽然能将汉字库及有关管理软件拷入虚拟

盘,但往往并不能真正解决实际问题,因为相关的一些功能程序(如查找、显示字模信息等)仍需改动。笔者在使用中并没有建立虚拟盘,而是采用选体技术将汉字库字模装入扩充内存 20000—7FFFFH 中,并将有关该字库的功能程序进行了修改,使原来从基本的 640KB 内存中读字模的操作改为转 1 体、读扩充内存(384KB),使其汉字功能正常实现。用此方法改造汉字系统后,可为用户省下大约 400KB 的空间,因此用户可以运行需要汉字系统支持的较大的软件。

笔者对 CCDOS 汉字装入模块和显示管理模块进行了一些改动。通过软、硬件上的改动,使自己的 PC XT 机“升级”并具有了 PC/AT 的功能。当然,改动绝非一处,限于篇幅,在此仅作简要介绍。下面举例说明对从内存字库区读取一个汉字(16×16 点阵 32 个字节)字模的程序段的修改,希望读者能够窥一斑而知全豹,达到举一反三的效果。

原 CCDOS 的读字模程序段:

```
0FFB:2282 06    PUSH ES
0FFB:2283 57    PUSH DI
0FFB:2284 1E    PUSH DS
0FFB:2285 BF7800  MOV DI,0078
0FFB:2288 07    POP ES
0FFB:2289 06    PUSH ES
0FFB:228A 8EDE    MOV DS,SI
0FFB:228C 33F6    XOR SI,SI
0FFB:228E B91000    MOV CX,0010
0FFB:2291 AD    LODSW
0FFB:2292 F6C370  TEST BL,70
0FFB:2295 7402    JZ 2299
0FFB:2297 F7D0    NOT AX
0FFB:2299 AA    STOSB
0FFB:229A 26    ES;
0FFB:229B 886511    MOV[DI|14],AH
0FFB:229E E2F1    LOOP 2291
0FFB:22A0 1F    POP DS
0FFB:22A1 5F    POP DI
0FFB:22A2 07    POP ES
```

改动后的读字模程序段:

```
0FFB:26B4 06    PUSH ES ;保存附加段
0FFB:26B5 57    PUSH DI ;保存写汉字内码光标
0FFB:26B6 1E    PUSH DS ;保存数据段
0FFB:26B7 BF7800  MOV DI,0078 ;字模加工区
0FFB:26BA 07    POP ES ;附加段指向数据段
0FFB:26BB 06    PUSH ES ;保存附加段
0FFB:26BC 8EDE    MOV DS,SI ;数据段指向要写的汉字字模段码
```

单片软光盘21MB 系统问世

——电子计算机新一代存储设备

1992年,美国在 COMDEC 会上宣布了新一代(数据)存储设备——软光盘(Floptical Diskette)系统,汇集光、磁、电高技术于一体,掀起了活动存储设备的又一浪潮。软光盘把普通软磁盘的线密度与光盘的道密度技术成功的结合在一起,既有较高的容量、性能,又有低廉的造价,将逐步成为软盘的新标准。

目前,投放市场的微机软光盘存储系统,由 AT 总线 SCSI 适配卡、软光盘驱动器及软光盘组成,它们的尺寸、外观与普遍3.5英寸的软磁盘存储系统完全一样。软光盘不仅可读写软光盘,而且可读写普通的3.5英寸的720KB 或1.44MB 软磁盘,且读写速度是普通软磁盘的三倍,数据传递速度为1.6Mbits/秒。一张格式化后软光盘容量为21MB,它是普遍的高密盘1.44MB 的14倍,但其每MB 的价格与普遍软磁盘是一样的,而它给数据传递、数据保密、软件销售等领域带来的好处和方便,是普通软磁盘所不能及的。由于采取CRC 和 Reed/Solomon 编码检错、纠错技术,因而可靠性大大提高,错码率小于指数10的-12位,性能价格比明显优于普通软磁盘系统。

许多著名电脑公司都在积极装备软光盘存储系统,如美国 SGI 公司,率先决定在其工作站上装备软光盘系统;Microsoft 公司即将问世又一杰作 Windows NT 也包括软光盘系统的驱动软件。根据 CRN(Computer Reseller News)预测,1993年将有32万套软光盘系统投入使用,1994年将达到1.3亿套。电脑装备软光盘存储系统已成为一种不可逆转的潮流,将使电脑大大增值,更增加电脑的适用性。 李恒嘉供稿

无绳电话干扰电视节目

南京市金陵职业大学计算中心 钱雁群

无绳电话俗称二哥大或叫子母机。台湾产的无绳电话在南京至少有500台。这些无绳电话工作频率在70~80兆赫;而本地四频道电视工作在76~84兆赫。于是邻居一用子母机,我们就无法看江苏台。这比游戏机无线发射时的干扰强多了。彩电图像变黑白条胡乱翻滚,伴音怪叫不断,真是引起了民愤。

93年3月20日江苏台在扬子晚报讲明台湾产无绳电话“不符合国家使用无线电话的有关规定”。南京无线电管理委员会也在那张报上称台湾产无绳电话使用者为“非法用户”。文章告诉市民:南京地区收看4频道的观众,改收32频道可避免干扰。

我们认为:面对干扰源老躲也不是办法。南京可备用一个四频道,其它城市3~5频道是不是也要再设频道呢?众所周知,电视台和广播电台所占频率是不允许个人随意干扰的,希望有关部门采取有效措施,打击这种损人利己或损人不利己的行为。

求援

我是一个 CEC-I 中华机用户,求援一台旧的或您闲置不用的 CEC-I 机用软盘驱动器,也希望和您交流软件、在软件上得到您的支持。

联系地址:河南安阳彩管玻壳公司屏加工检验:高军

邮政编码:455000

0FFB:26BE 31F6 XOR SI,SI;取字模指针 SI 指向要写的字模首址
0FFB:26C0 B91000 MOV CX,0010;16→CX(写字模计数16个字)
0FFB:26C3 B001 MOV AL,01;
0FFB:26C5 E6E0 OUT E0,AL;转1体
0FFB:26C7 AD LODSW;取二字节汉字模一→AX
0FFB:26C8 50 PUSH AX;
0FFB:26C9 B000 MOV AL,00;
0FFB:26CB E6E0 OUT E0,AL;
0FFB:26CD 58 POP AX;转0体
0FFB:26CE F6C370 TEST BL,70;反白显示?
0FFB:26D1 7402 JZ 26D5;不是,转26D5
0FFB:26D3 F700 NOT AX;反白显示,AX取反
0FFB:26D5 AA STOSB;左半区字模(AL)写入字模;加工区左半区

0FFB:26D6 26 ES;
0FFB:26D7 886511 MOV [DI+11],AH;右半区字模(AH)写入字模;加工区右半区
0FFB:26DA E2E7 : LOOP 26C3;循环执行26C3—26D7程序段,直到;32字节全部写入字模加工区
0FFB:26DC 1F :POP DS;恢复数据段
0FFB:26DD 5F :POP DI;恢复汉字内码光标
0FFB:26DE 07 :POP ES;恢复附加段

本文介绍的 PC/XT 及其兼容机内存的扩充和使用的软件包括二张软盘。有需要者请将款汇到编辑部。定价80元(含邮挂费)。硬件改动所需的414256RAM 芯片可在当地购买。

汇款地址:北京173信箱《电子与电脑》编辑部 杨逢仪收 邮编:100036