

# E&C

## 1992

●一九九二年 ●总期第92期



# 電子

ISSN 1000-1077

479

# 與 電腦

你能分辨出这两张  
照片有什么不同吗？



3.2 MegaBytes



155 KiloBytes



• ELECTRONICS AND COMPUTERS •



# 《UNIX系统V第4版 中文手册》丛书



电子工业出版社 出版

● 国内代号：2-888 定价：0.95元



(接封二)

## 三. CLMS-64 系列

型号	工艺	电源电压	时钟频率	ROM (位)	RAM (位)	输入口	输出口	I/O 口	定时计数器 8(12位 R/W)	中断源	堆栈	指令数	机器周期	电源消耗		封装
MSM6404	CMOS	5V	4.2MHz	4000×8	256×4	4	—	32	—	5	32	91	952ns	6mA	1 $\mu$ A	42DIP/44FLAT
MSM6411	CMOS	5V	4.2MHz	1024×8	32×4	4	—	8	—	2	8	63	952ns	6mA	1 $\mu$ A	16DIP
MSM6422	CMOS	5V	4.2MHz	2048×8	64×4	1	—	18	—	2	16	63	952ns	6mA	1 $\mu$ A	24DIP/24FLAT
MSM6442	CMOS	5V	4.2MHz	2048×8	128×4	4	46×21CDSeg.	28	—	4	16	71	952ns	6mA	1 $\mu$ A	80FLAT

## 四. OLMS-65 系列

型号	工艺	电源电压	时钟频率	ROM (位)	RAM (位)	输入口	输出口	I/O 口	定时计数器	中断源	堆栈	指令数	机器周期	电源消耗		封装
MSM6502	CMOS	3V	32KHz	2000×8	128×4	4	108LCDSeg	8	—	3	42	68	91.5 $\mu$ s	45 $\mu$ A	30 $\mu$ A	44FLAT

## 五. 8 位系列

型号	工艺	电源电压	时钟频率	ROM (位)	RAM (位)	输入口	输出口	I/O 口	定时计数器	中断源	堆栈	指令数	机器周期	电源消耗		封装
MSM80C35	CMOS	5V	11MHz	—	64×8	—	—	24	8位 R/W	2	8	111	1.36 $\mu$ s	10mA	1 $\mu$ A	40DIP/44FLAT
MSM80C39	CMOS	5V	11MHz	—	128×8	—	—	24	8位 R/W	2	8	111	1.36 $\mu$ s	10mA	1 $\mu$ A	40DIP/44FLAT
MSM80C40	CMOS	5V	11MHz	—	256×8	—	—	24	8位 R/W	2	8	111	1.36 $\mu$ s	10mA	1 $\mu$ A	40DIP/44FLAT
MSM80C31	CMOS	5V	12MHz	—	128×8	—	—	32	16位 R/W ×2	5	8	111	1 $\mu$ s	16mA	1 $\mu$ A	40DIP/44FLAT
MSM80C48	CMOS	5V	11MHz	1024×8	64×8	—	—	24	8位 R/W	2	8	111	1.36 $\mu$ s	10mA	1 $\mu$ A	40DIP/44FLAT
MSM80C49	CMOS	5V	11MHz	2048×8	128×8	—	—	24	8位 R/W	2	8	111	1.36 $\mu$ s	10mA	1 $\mu$ A	40DIP/44FLAT
MSM80C50	CMOS	5V	11MHz	4096×8	256×8	—	—	24	8位 R/W	2	8	111	1.36 $\mu$ s	10mA	1 $\mu$ A	40DIP/44FLAT
MSM80C51	CMOS	5V	12MHz	4096×8	128×8	—	—	32	16位 R/W ×2	5	8	111	1 $\mu$ s	16mA	1 $\mu$ A	40DIP/44FLAT



# OKI 公司单片微机一览表

- 均为 CMOS 产品
- 4 位有 4 个系列: OLMS-40, OLMS50/60, OLMS64, OLMS65
- 8 位一个系列
- 4 位各系列都有 LCD 输出口

## 一. OLMS-40 系列

型号	工艺	电源电压	时钟频率	ROM (位)	RAM (位)	输入口	输出口	I/O 口	定时器	中断源	堆栈	指令数	机器周期	电源消耗		封装
														工作状态	备用态	
MSM5840	CMOS	5V	4.2MHz	2048×8	128×4	6	16	8	8 位 R/W	2	4	98	7.6μs	1.6mA	—	42DIP/44FLAT
MSM5842	CMOS	5V	4.2MHz	768×8	32×4	5	8	8	8 位	—	1	52	7.6μs	1.5mA	—	28DIP/32FLAT
MSM5842I	CMOS	5V	4.2MHz	1536×8	40×4	5	35LCDSeg. 5LCDSeg. 或 LOGIC	8	12 位	—	1	52	7.6μs	2.0mA	—	60FLAT
MSM5847	CMOS	3V	32KHz	1536×8	96×4	—	24×3LCDSeg.	7	13 位	—	2	43	610μs	50μA	—	44FLAT

## 二. OLMS-50/60 系列

型号	工艺	电源电压	时钟频率	ROM (位)	RAM (位)	输入口	输出口	I/O 口	定时器	中断源	堆栈	指令数	机器周期	电源消耗		封装
														3μA	—	
MSM5052	CMOS	1.5V	32KHz	1280×14	62×4	8	26×2LCDSeg. 5LOGIC	—	—	—	—	42	122μs	3μA	—	CHIP
MSM5054	CMOS	1.5V/3V	32KHz	1024×14	62×4	6	44×2LCDSeg. 4LOGIC	—	—	—	—	40	122μs	3μA	—	CHIP
MSM5055	CMOS	1.5V/3V	32KHz	1792×14	96×4	8	60×2LCDSeg. 4LOGIC	—	—	—	—	42	122μs	3μA	—	CHIP
MSM5056	CMOS	1.5V	32KHz	1792×14	90×4	4	38×2LCDSeg. 4LOGIC	—	—	—	—	42	122μs	3μA	—	CHIP
MSM6051	CMOS	1.5V/3V	32KHz	2560×14	120×4	9	63×3LCDSeg. 4LOGIC	—	—	1	2	59	91.5μs	3μA	—	CHIP
MSM6052	CMOS	3V	3.58MHz	2048×14	640×4	12	12	4	4 位	1	5	52	17.9μs	1.2mA	0.2μA	28DIP/40DIP



電子  
與電腦

ISSN: 1000-1077

479



一九九二年

总期第92期

# 電子與電腦

\* ELECTRONICS AND COMPUTERS \*

## 目 录

### · 综述 ·

图象处理的一大革命——压缩 ..... (2)

### · PC 用户 ·

PC 机软件加解密技术剖析 ..... 李文亮(3)

字符处理函数在 FORTRAN 中的实现  
..... 任铁良 丁玲玲(5)

CPAV 防病毒软件使用简介 ..... 吴桦(7)

带检索功能的通讯录录入程序 ..... 张春明(10)

PC 机中系统时间的自动显示 ..... 郑嘉琦(12)

最近出现的几种新病毒 ..... 苏民生(13)

电子扭计板 ..... 葛建华 吴立国(14)

第四届国际信息学奥林匹克竞赛试题 ..... (15)

### · 学习机之友 ·

微机模拟游标卡尺读数训练程序 ..... 王太文(17)

磁带虚拟磁盘的文件管理 ..... 胡发新(18)

APPLE-Ⅱ 音乐功能扩充 ..... 陈建明(19)

POSITION 命令新用 ..... 周进(20)

CEC-与 1724 打印机配接图形硬拷贝程序  
..... 张益贵(20)

游戏接口 PDL 的扩展 ..... 张建群(21)

SUPER DOS 简介 ..... 张志(21)

### · 语言讲座 ·

第十讲 监控子程序的调用(下) ..... 朱国江(23)

### · 学用单片机 ·

BJS-51 单片机实验系统(续) ..... 张俊漠(26)

### · 学装微电脑 ·

微电脑控制微型钻床 ..... 易齐干(29)

### · 电脑巧开发 ·

数字集成电路简易测试器 ..... 王正英(35)

中华学习机调制伴音电路 ..... 李永和(39)

### · 电脑游戏机 ·

第四讲 游戏程序的设计过程(上) ..... 于春(40)

### · 维修经验谈 ·

笔记本型电脑技术和市场 ..... 赵广恩(43)

用软件对软驱进行简单读写检测

..... 李晓中 麻佳洛 谢静 张景生(44)

### · 读者联谊 ·

普及型 PC 个人用户软件交流联谊活动问题解答(十一)

..... 王路敬(46)

机械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 副主编:王昌铭

责任编辑:杨逢仪

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:0.95 元



# 图象处理的一大革命——压缩

中国电子器件深圳公司供稿

本期封面有两幅图象,其中原图信息量为 3.2MB,经压缩处理后为 155KB。人眼看不出二幅图象的色调有什么不同。不难想象,所谓压缩,就是将原图信息量删去相当大的部分,而仍不改变原图本色的信息处理技术。这样一来,就可以利用压缩/还原技术,在 CD 唱盘上再存入 VHS 影像,一份消费,两份享受。将声音/图象压缩后存入 CD-ROM 中,只要在电脑里加上解压缩卡,就可以用电脑唱“卡拉 OK”了,为计算机的多媒体化提供方便。占 6MHZ 的电视频道,占 27MHZ 的卫星频道均可分割成四个频道,大大的增加电视的频道,为人们增添更多、更丰富的艺术享受。

美国 C-Cube Microsystems 公司首先推出 C-Cube CL550 图象压缩/还原处理器芯片及其开发系统,为用户提供了一个了解和应用压缩技术的环境。

本文根据 C-Cube 公司提供的资料,以问答形式对“压缩”作一简要的介绍。

问:何谓 C-Cube CL550 处理器?

答:C-Cube CL550 是一种采用 JPEG 标准的单片图象压缩/还原处理器。

问:图象压缩解决了图象处理的什么问题?

答:图象压缩技术解决了图象处理中的二个瓶颈问题:存储器花费大和总线速度的极限。如不采用压缩技术,要存储 10 秒钟的高分辨(720×480)的视频图象(30 帧/秒)就需要 311MB 的存储量,总线速度为 31MB/秒。而采用 JPEG 压缩标准处理之后,它被压到 13MB 的存储容量,总线速度降低为 1.3MB/秒。

问:图象压缩是一种新技术吗?

答:否,该技术在 high-end 系统中已经应用多年了。然而,作为图象压缩/还原芯片的出现,确是最近的事情。

问:什么是 JPEG 标准?

答:JPEG (Joint Photographic Experts Group) 是国际 CCITT 和 ISO 共同制订的连续色调压缩标准。C-Cube 公司是 JPEG 委员会的成员。

问:JPEG 的重要性何在?

答:JPEG 是一种国际开放系统标准,它使不同制造者所设计的系统均可互联通信和交换信息。JPEG 规则是在高压比下获得卓越的图象品质的保证。

问:在高压比下高品质图象意味着什么?

答:当用 JPEG 规则将一幅屏幕图象,以 10:1 进行压缩;一幅打印图象以 25:1 进行压缩时,凭人眼是观察不出它们有什么不同,对于人眼来说,似乎全部信息仍被完整保存。其实,压缩比 25:1 就意味着该图象有

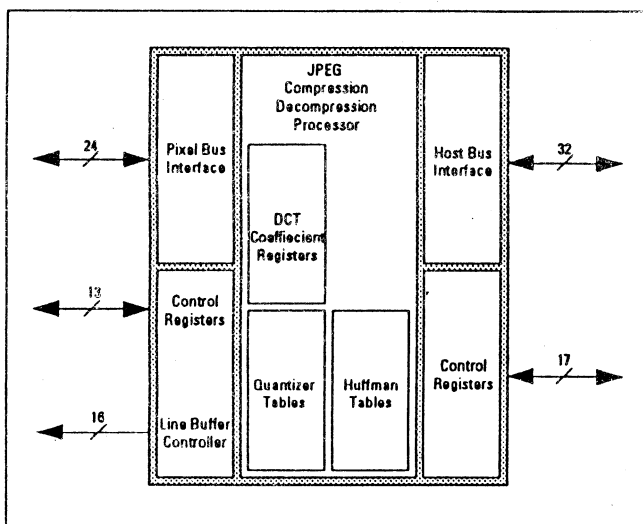
96% 的数字信息被删去了。

问:如何选择压缩比?

答:C-Cube CL550 处理器的压缩比是用编程的方法选择的。其压缩比最高为 100:1。

问:C-Cube CL550 处理器的结构如何?

答:CL550 的功能框图如下:



C-Cube CL550 Processor Functional Block Diagram

144-PIN 标准陶瓷座阵列封装的有:C-Cube CL550-35 和 CL550-30。

144-PIN 扁平封装的有 CL550-35 和 CL550-10。

采用 CMOS 工艺,在 35MHZ 时最小耗散功率为 3.5W。

问:C-Cube CL550 可处理的彩色范围为何?

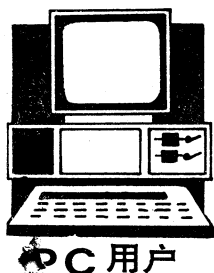
答:C-Cube CL550 处理器可处理 8 位灰度 24 位 RGB, CMYK 或 4:4:4 和 YUV (4:4:4 和 2:2:2) 彩色空间。

问:C-Cube CL550 有几种处理速度?

答:C-Cube CL550 有三种处理速度:10MHZ、30MHZ 和 35MHZ。C-Cube CL550-10 (10MHZ) 用于对静止图象的处理。C-Cube CL550-30 和 C-Cube CL550-35 能实时处理(压缩/还原)NTSC, PAL 和 CCIR601 视频图象。

多个 C-Cube CL550 处理器可对图象进行

(下转第 11 页)



# PC 软件加解密技术剖析

华南理工大学计算机系软件专业 88 级 李文亮

在计算机系统的开发过程中,软件费用所占的比例越来越大,软件的地位也越来越重要。一方面软件和数据非常有价值,另一方面任何人都很容易得到原版的复制品,因此,软件的非法拷贝现象也越演越烈。

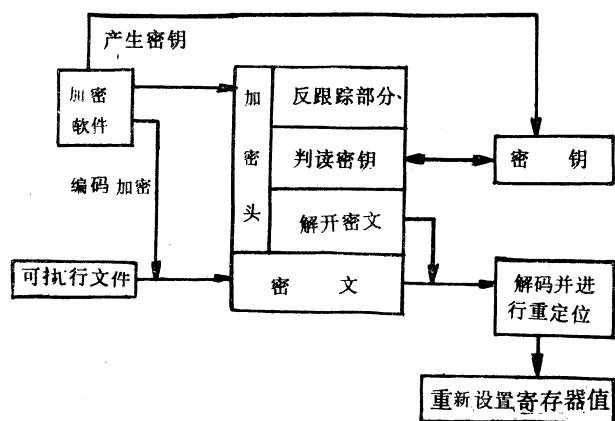
为了保护软件的潜在市场,人们开始研究加密技术,同时解密技术也伴随而来。加密和解密经过“魔高一尺,道高一丈”的矛盾斗争,从低级向高级发展。

PC 机加密技术最早在一些游戏上使用,后来因为其简单易行而迅速得到推广。并出现了象 PROLOK、SOFTGUARD 和 LOCK89 等一些专门的加密工具软件。我国近几年也对加解密技术做了深入的研究,并得到广泛的应用,生产了诸如 LOCK89、COPY02 等许多加解密工具。

## 一、加密技术基础

所谓的 PC 软件加密,就是使用特殊技术在磁盘上产生特殊的信息块,普通 PC 软盘驱动器可以把这些信读出来,但是不能够产生同样的信息块。这样,当用户试图去复制该软盘时,那个特殊信息就会丢失。而我们可以把程序嵌入到软件中,使之去读取这个特殊信息,判断正确才能去执行。

一个典型的软件加密系统是由两部分组成:第一部分是在每个原版程序盘上都有的特殊信息块(即密钥);第二部分是密钥判读程序,一般嵌入在应用程序之中,形成加密文件。当启动执行加密软件时,该段程序将首先被执行。它先去判读密钥,正确的话才去执行应用程序。加密的原理图如下:



图一软件加密原理图

由图可以看到,加密文件分为加密头和原执行文件的密文两部份。其优点在于加密软件的通用性强,而无需修改原程序。但是缺点也是明显的,即破译者往往可以通过把加密头去掉而得到原可执行文件。

我们可以把程序体和加密头揉为一体,实行一体化的加密。这样,破译者需要对整个系统进行彻底剖析,才有可能解密。而对于一个比较庞大的系统,这基本上是不可能的。因此采用这种方法可以大大地提高保密性能。超想自然码就宣称使用了这种方法。但这种方法要求软件开发者和加密者必须是同一个主体,有很大的局限性。而且往往由于程序的嵌入,破坏了软件整体的严谨性。

由图一可以看到,密钥是整个加密系统的关键,如果密钥能随意被拷贝,那么整个系统就完全失去其意义了。至于密文的产生方法,并无特殊要求,只要编码和解码能正确进行即可。

在讨论加密技术之前,必须对软盘的物理结构;磁盘格式;磁盘参数表有所了解,由于篇幅所限,这里不作介绍,请参阅有关资料。

在了解了 PC 磁盘的格式,中断 13H 的调用方法和磁盘参数表后,就可以利用以上知识来产生密钥了。

最简单的就是通过修改软盘参数表,把软盘格式化为特殊格式来作为密钥。正常情况下 PC 机是无法读写这些信息的,当你试图用 DOS 来拷贝被保护的磁盘时,除了密钥之外的其它信息都被正确拷贝,运行程序时就会找不到密钥信息;而加密程序则可以通过修改软盘参数表来完成读取和判断。从硬件的角度讲,读和写是完全不同的,它们是两套分开的电路。同样道理,PC 机的磁盘驱动器可以识别原盘上的不确定格式,但不能在复制品上生成这一格式,防止拷贝的方法正是基于这一原理。然而,特殊格式的信息可以用特殊的软件拷贝出来,象 COPYWRIT 等一些高级拷贝工具,是以字节或位来拷贝的,并且具有一定的分析功能,往往可以成功地拷贝出一个新的“原盘”。

另一种方法就是利用特殊的设备,制造出一般驱动器根本无法生成的特殊格式。如在磁道上产生 256 个仅含有标志而不带数据的空扇区。这种方法能抵御大部份的攻击,但设备费用是比较昂贵的。LOTUS 1—2—3 就是在 0 道和 1 道的 08 扇区之后,制造了 10 个没有数据,只有 ID 部分的扇区。

后来经过研究,又产生了一种“指纹加密法”。

我们都知识,每个人的指纹都是其特有的,没有两个人的指纹是相同的。那么,软盘上是否也存在“指纹”呢?正常情况下是不存在的,但可以人工生成。

一个磁道所能容纳的信息量是有限的,在双面倍密度的软盘上大概为 1850K 字节。正常情况下,所有的扇区都应容纳在一个磁道之内,但是如果扇区较大,数量较多的话,扇区就有可能从磁道尾越过磁道首尾



的接缝处,而跨回到磁道首。我们可以把磁道想象为一个环,当信息量大于环长时,就产生了一部分的重叠。此时,由于软盘驱动器的电磁性能指标(如电流、电压、磁场强度和磁盘转速等)的不稳定性,在磁道的头尾衔接处就会产生一些特殊的“指纹信息”。这些信息使用 COPYWRIT 等高级拷贝工具,拷贝卡或拷贝机,都是无法拷贝的,甚至对于同一部驱动器,每次产生的指纹也是不一样的。

作者根据这一原理,在 IBM 机上用汇编语言编写了两段制做和读取指纹的程序,用 TASM2.01 编译、链接和调试通过。具体过程和程序见附录。

另外,由于制作工艺上的差别,每个硬盘都有其自然“指纹”。即使是同一型号的硬盘,也有一定的差异,即使把一个硬盘 DISKCOPY 到另一个硬盘上去,两者也不会一模一样。这点是与软盘有很大区别的。加密系统可以在安装到硬盘上时搜索这些特殊信息,并用来作密钥。如果 2.13 汉字系统就是把标识放在硬盘的主引导扇上。

产生密钥的另一种方法是硬加密,其典型代表是激光加密。它利用激光在磁盘上打孔,制造出硬坏标志。一般有单孔和双孔两种。激光的方向性好,能量高,均匀,烧出来的孔很小,基本上不会破坏磁盘表面的光洁度,而且随机性很好,相同的概率为几十亿分之一。同时,由于激光孔是物理性标识,用软件方法是无法生成的,这就可以抵御各种拷贝工具的攻击,有极强的保密性。价格昂贵是其最大的局限性,同时,它也要求驱动器有较稳定的性能。国内也有用针直接在磁盘上扎孔的方法来模拟的,也有一定的效果,但会破坏磁盘表面的光洁度,损坏磁头。

还有一种方式就是使用加密卡。当加密程序运行的时候,需要寻找指定的芯片,并把一个数据送给它,经运算后返回一个密钥,判断正确后才继续运行。这种方式安全性好,但成本高,使用也不够方便。

最近,又提出了两种新型加密方法——掩模加密技术和电磁加密技术。前者采用半导体加工工艺中的镀膜方法,后者借助于专门的电磁机构,在软盘上生成一系列的密钥信息;前者与激光加密相似,但对磁盘无任何破坏;后者采用了与 PC 机软驱 MFM 或 FM 不同的编码格式,都有极强的反拷贝性,是很有发展前途的。

随着加密技术的发展,密钥往往已经不能被拷贝。因此破译者更多地把目光投向加密程序。现在更多是采用软件破译的方法,迫使加密者不得不考虑如何提高反跟踪性能。所谓的反跟踪,就是通过动态解码、破坏中断等方式,阻止破译者通过对加密头的跟踪分析,破解加密软件。这部分将在后面作详细介绍。

一般加密有如下的几种形式:

(1)使用口令密码,另一形式是用回答问题来替代。这种方法多为早期的游戏软件所采用,知道了密码便可随意拷贝,保护性最差。

(2)修改文件名、属性。通过修改文件属性为隐含,

在文件名中嵌入特殊字符等方式来加密。它无法阻止 DISKCOPY,在 PCTOOLS 等强劲的磁盘工具出现之后,更是不堪一击,仅在硬盘上还有一定的使用余地。

早期的王码五笔系统就是使用了这种方法。例如,用 PCTOOLS 把目录属性改为 17H,并把目录的最后一个字节改为 80H,那么,无论在 DOS 或 PCTOOLS 下都无法发现这个目录;但在 DOS 下可对此目录进行一切正常工作。

(3)判读密钥法。首先判断密钥是否存在和正确与否,才去执行程序,高级的还可利用密钥来对密文进行解密。这种方法保密性最强,应用也最广泛。

随着加密技术的发展,涌现了许多优秀的加密系统。美国加州 New Bury Park 市 VALUT 公司的 PRO-LOCK 就是激光加密的经典。我国的 CCDOS4.0 就是采用它来作保护工具的。Santa Clara 市 SOFTGUARD 公司的 SOFTGUARD 是另一个著名的加密软件。我国的技术人员也开发了诸如 WL、SUPER-LOCK(虎符)和 LOCK89 等加密系统。LOCK89 采用了多层次硬加密、软加密和反动态跟踪技术,实现了多变和不可拷贝,是其中的佼佼者。

## 二、加解密技术分析

破解一个加密系统一般有下面两个途径:

(1)在目标盘上完成密钥的拷贝和制作,使目标盘和原盘达到一致。

COPYWRIT 就是使用了这种方法。严格地说,这还不是解密,因为拷贝后生成的新盘还是有保护的。它对付一些简单的加密系统还是比较有效的,但对指纹或激光加密等却是无能为力。

(2)通过对加密程序的跟踪分析,找到判读密钥部分,把它去掉或作一些修改,使之能脱离密钥而独立工作,实现破译。一般可分为静态分析——把程序反汇编打印出来进行分析和动态分析——利用调试工具进行动态跟踪分析两种方法,往往要把两者紧密结合起来,才能成功。

破译者必须熟悉整个 PC 机系统,特别是要熟练掌握 8088 汇编语言,中断调用和处理,DOS 功能调用以及一些特殊的端口操作命令。

下面介绍一些常见的解密工具。

这是一些市面上流行的拷贝工具: COPYIIPC、COPYWRIT、SUPERCOPY、LOCKSMITH、COPY-ANY 和 DISKEXPLORE 等。美国 Oregon 州 Portland 市的 Central Point 公司和加拿大安大略省多伦多市的 Quaid 公司专门出售破密软件,他们出售的软件也都不保护。他们的 COPYIIPC 和 COPYWRIT 就是拷贝者的两大皇牌。

一般拿到加密软件之后,可以先用手头上的拷贝工具试一试,不行的话才考虑破译。较新版的 COPYWRIT 甚至能成功拷贝早期的单孔激光加密盘,而 RCOPY02 对很多的加密系统都有效。

LOCKSMITH 具有读写磁盘扇区,分析、修改磁盘参数的功能,可以用来实现简单的加解密。



DEBUG 原是 DOS 的一个机器码调试程序,因为它有很强的动态跟踪功能,而被广泛应用于解密。SYMDEB 的功能与 DEBUG 相似,但它另外还有输入、输出改向的功能,可很方便地生成反汇编清单文件,非常适合于作解密分析工具。下面是 SYMDEB 的几有关命令:

- A[ADDRESS]从地址 ADDRESS 开始输入汇编程序;
- BP[ADDRESS]在地址 ADDRESS 处设置断点;
- D[ADDRE1][ADDRE2]

显示从地址 ADDRE1 到 ADDRE2 间的数据信息,可以有 DB,DW,DD 等多种显示方式;

- T 单步跟踪;
- P 同 T,但遇到子程序或中断调用时一次执行完;
- U[ADDRE1][ADDRE2] 从地址 ADDRE1 开始反汇编到 ADDRE2;
- < IN-FILE 从文件 IN-FILE 读入命令;
- > OUT-FILE 把输出送到文件 OUT-FILE 中去;

默认的输入和输出文件都是 CON。

TURBO DEBUGGER 和 CODEVIEW 是高级的源代码调试程序,它们采用集成工作方式,所有的寄存器值和当前指令代码都显示在屏幕上,并用光条指示当前指令行,使用非常方便。可惜系统比较庞大,因此也较脆弱,很容易被反跟踪程序所破坏。

另外,市面上还有诸如 UNLOCK、UNGUARD 等一些解密软件,它们都是针对某种加密系统的,使用方便,效果也较好。江西科达通信电脑技术发展公司研制的 RCOPY02,实现了从内存中进行加密软件的拷贝。按照这一工作原理,各种工具所加密的程序,都可能被拷贝出一个工作副本。被认为是国内首屈一指的 LOCK89 加密盘的生产母盘,经 RCOPY02 处理后,也能产生可拷贝的工作副本、甚至工作副本还可以建立在硬盘上。另外,它还能对各种应用程序保留任一时刻的工作断点,有极强的现场保留功能。UNLOCK89 可以破解象意大利赛车等一些游戏软件,使之成为非加密版,解密能力也比较强。

由于软件保护的方法越来越多,难以使用单一的程序来战胜对手,软件拷贝公司用硬件巧妙地解决了这一问题。Central Point 和 Disk-Tech 公司通过增加 PC 机的拷贝卡,对磁盘实现了真正的磁介质拷贝。Central Point 公司的拷贝板与磁盘控制器和驱动器都有联线,拷贝卡直接控制驱动器的读写,它可以拷贝任意信息。正如 Central Point 公司总裁米开尔·布朗解释:“它只不过是把数据从一个磁盘拾起来,放在另一个磁盘上,并说来下一个。”

(未完,待续)

## 字符处理函数在 FORTRAN 中的实现

冶金工业部鞍山热能研究院 (114004)任铁良 丁玲玲

FORTRAN 有诸多的数据处理函数,但没有提供诸如 BASIC 和 dBASE 中那样丰富多采的字符处理函数,这使得它在字符处理方面比 BASIC 和 dBASE 逊色。特别是在用 FORTRAN 编程同时实现复杂的数值计算和繁多的字符处理时,字符处理问题便显得尤为突出。

FORTRAN77 V3.3 中提供的与字符处理有关的函数为:ASC 码转换成其对应字符的字符型函数 CHAR,字符串的第一个字符转换成对应 ASC 码的整型函数 ICHAR,及判断两字符串顺序的逻辑函数 LGE、LGT、LLE、LLT;FORTRAN 77 更高的版本还提供了求字符串长度的整型函数 LEN、求子字符串位置的整型函数 INDEX 和两字符串连接运算符“//”。

笔者根据 FORTRAN 77 V3.3 中的基本字符处理函数 CHAR 和 ICHAR,及取子字符串和为子字符串赋值操作,编制字符串处理外部函数程序,并且经编译生成目标程序,再把目标程序追加到 FORTRAN.LIB 文件中;当在 FORTRAN 中调用这些函数时,只要引用对应的函数名即可,这样就可 FORTRAN 中实现与 BASIC 和 dBASE 字符处理功能相同的一些函数,以弥补 FORTRAN 语言中字符处理功能的不足。

附后的 FORTRAN 程序实现了下列字符处理函数:

### (1)LEN 函数

函数功能:求出字符串的长度,字符串尾部空格不计。

### (2)INDEX 函数

函数功能:求出第二个字符串在第一个字符串中的起始位置(字符串尾部空格不计),若第一个字符串不含有第二个字符串,则函数值为 0

### (3)PLUS 函数

函数功能:连接两字符串(字符串尾部空格不计)。

### (4)REPLIC 函数

函数功能:调用该函数得到一字符串,该字符串由给定的第一个参数字符串(字符串尾部空格不计)重复组成,重复次数由第二个参数给出。

### (5)UPPER 函数

函数功能:字符串中的小写字符转换为大写字符(字符串尾部空格不计)。

### (6)LOWER 函数

函数功能:字符串中的大写字符转换为小写字符(字符串尾部空格不计)。

把附后的源程序 FUNC. FOR 用 FORTRAN 77V3.3 编译程序 FOR1. EXE 和 PAS2. EXE 编译产生目标程序,应用 DOS 中的库文件管理程序 LIB. EXE 把上述目标程序加到库文件 FORTRAN. LIB 中。

其方法是打入命令:

>LIB FORTRAN. LIB+FUNC. OBJ;

其中外部命令 LIB. EXE 文件应在当前目录中或在已用 PATH 命令指定的目录中。

用 FORTRAN 语言编制应用程序调用上述函数时,需要注意参数个数和类型及顺序,涉及到的字符变量或字符函数只要用字符型说明语句 CHARACTER \* 127 定义其长度即可,调用函数的方法与调用内部函数相同。

TYPE FUNC. FOR

```
C      LEN 函数,求字符串的长度
C      函数类型:整型
C      参数个数:1
C      参数类型:字符型
      FUNCTION LEN(ST)
      CHARACTER * 127 ST
      LEN=0
      DO 10 LEN=127,1,-1
          IF (ST(LEN:LEN).NE. CHAR(32))RETURN
10      CONTINUE
      RETURN
      END
C      INDEX 函数,子字符串搜索
C      函数类型:整型
C      参数个数:2
C      参数类型:字符型,字符型
      FUNCTION INDEX(ST1,ST2)
      CHARACTER * 127 ST1,ST2
      L1=LEN(ST1)
      L2=LEN(ST2)
      DO 10 INDEX=1,L1-L2+1
          IF (ST1 (INDEX:L1+INDEX-1) . EQ. ST2)
10      CONTINUE
      INDEX=0
      RETURN
      END
C      PLUS 函数,字符串联结
C      函数类型:字符型
C      参数个数:2
C      参数类型:字符型,字符型
      CHARACTER * 127 FUNCTION PLUS(ST1,ST2)
      CHARACTER * 127 ST1,ST2
```

```
      L1=LEN(ST1)
      L2=LEN(ST2)
      PLUS(1:L1)=ST1
      PLUS(L1+1:L1+L2)=ST2
      DO 10 I=L1+L2+1,127
          PLUS(I:I)=CHAR(32)
      RETURN
      END
C      REPLIC 函数,字符串重复
C      函数类型:字符型
C      参数个数:2
C      参数类型:字符型,整型
      CHARACTER * 127 FUNCTION REPLIC(ST,N)
      CHARACTER * 127 ST,PLUS
      REPLIC=CHAR(32)
      DO 10 I=1,N
          REPLIC=PLUS(REPLIC,ST)
10      CONTINUE
      RETURN
      END
C      UPPER 函数,字符串小写转变为大写
C      函数类型:字符型
C      参数个数:1
C      参数类型:字符型
      CHARACTER * 127 FUNCTION UPPER(ST)
      CHARACTER * 127 ST
      UPPER=ST
      DO 10 I=1,LEN(ST)
          IA=ICHAR(UPPER(I:I))
          IF (IA. GE. 97. AND. IA. LE. 122)UPPER(I:I)=
              CHAR(IA-32)
10      CONTINUE
      RETURN
      END
C      LOWER 函数,字符串大写转变为小写
C      函数类型:字符型
C      参数个数:1
C      参数类型:字符型
      CHARACTER * 127 FUNCTION LOWER(ST)
      CHARACTER * 127 ST
      LOWER=ST
      DO 10 I=1,LEN(ST)
          IA=ICHAR(LOWER(I:I))
          IF (IA. GE. 65. AND. IA. LE. 90)LOWER(I:I)=
              CHAR(IA+32)
10      CONTINUE
      RETURN
      END
```



# CPAV 防病毒软件使用简介

交通部水运所(100088) 吴桦

美国 Central Point Software 公司 1991 年新推出的防病毒软件包 CPAV (Central Point Anti Virus) 1.0 版, 以其强大的功能和友好的用户界面而大受欢迎。该软件包可用于单用户和网络系统, 并与 Windows 3 全兼容。本文对该软件包的三大核心软件: BOOTSAFE.EXE, VSAFE.COM 和 CPAV.EXE 的用法做一简要介绍。

## 一 BOOTSAFE.EXE 的用法

BOOTSAFE.EXE 是一个以监视主引导区和文件分配表为手段的防病毒软件。在 DOS 提示符下, 键入 Bootsafef/? 并回车, 可以得到怎样使用命令参数的英文帮助信息, 其一般命令格式为:

Bootsafe[驱动器 1][驱动器 2][驱动器 3].../选择项

常用的选择项有:

/M——建立一个新的引导区备份文件。

/R——用 A 驱动器中的引导区备份文件使引导区受损的磁盘引导区得以恢复

命令中若无驱动器号, 则对当前驱动器操作。

Bootsafe 首先检查内存有无病毒或被非法减小, 若发现有其可清除的病毒, 先将病毒除掉, 若内存被非法减小而原因不明。则显示如下:

Memory has been suspiciously reduced, the cause may be an unknown virus. Press any key

提示指出: 不知什么原因, 内存减小了。可能是由某种不认识的病毒造成的, 按任意键退出。

如果内存检查未发现可疑处并且盘内尚无叫 Boot.cps 的引导区备份文件, 将弹出如下窗口提示:

Image file not found. Do you want to Create a new image file?  
[ ]Yes [ ]No

键入“N”即退出 Bootsafef, 返回 DOS。键入“Y”则在盘上建立一个名为 Boot.cps 的引导区备份文件。如果是对硬盘执行 Bootsafef, 还将弹出如下提示:

Save Image to Floppy. In Case of damage to the partition table, an image of it should be saved on to a diskette. Save partition table to drive A:?[Y]yes [N]No Insert back up disk in drive A:. Press any key

窗口内首先询问是否将该备份文件存到软盘上。键入“N”不把备份文件存入软盘, 窗口消失。键入“Y”, 则窗口内提示将软盘放入 A 驱动器后击任意键, 按提示执行后, 窗口内出现如下信息:

Partition table was backed up to file: A:\\*BOOT.CPS  
Press any key

其中 \* 为驱动器号, 表示备份文件已经以 \*BOOT.CPS 名的形式存于软盘中。此时再击任意键, 窗口下面出现如下提示:

Drive \*: Boot Images successfully created

表示已成功建立了引导区备份文件。这样, 以后再运行 Bootsafef 时, 将把引导区内容与原备份文件的内容相

比较。若一致则给出如下信息:

Drive \*: Boot sector and Partition table are OK

表示主引导区和硬盘分区表完好。若引导区被破坏, 可通过存于软盘的引导区备份文件恢复原来的引导区和硬盘分区表。例如:

A:Bootsafe C:/R 表示恢复 C 盘引导区

Bootsafe D:/R 表示恢复 D 盘引导区

首次进行备份时, 程序是不管引导区是否有病毒的。所以首次运行 Bootsafef 一定要保证主引导区和分区表是干净无毒的, 否则, 就会备份出一个带毒的引导区, 那就失去使用 Bootsafef 的意义了。

## 二 VSAFE.COM 的用法

VSAFE.COM 是一个以驻留内存形式动态监视病毒活动的软件。在 DOS 提示符下, 键入 VSAFE [驱动器号]并回车后, VSAFE 将进驻内存, 约占 24K 的内存空间。同时键入 ALT+V 键则激活控制窗口。窗口中的 8 个选择项依次是:

1. 硬盘低级格式化警告 (HD Low level format)
2. 驻留内存警告 (Resident)
3. 广泛的写保护 (General write protect)
4. 检查执行文件 (Check executable files)
5. 引导区型病毒警告 (Boot sector viruses)
6. 保护硬盘引导区 (Protect HD boot sector)
7. 保护软盘引导区 (Protect FD boot sector)
8. 保护可执行文件 (Protect executable files)

这 8 个选项均为开关项, 即键入某选项的数字, 如出现“√”表示将该选项打开, 如“√”消失, 将该选项关闭, 根据需要选定配置后, 按 Esc 键, 则这种配置在系统重新启动前将一直保持, 还可采用另一种方式即命令参数的方式来选定配置。

命令格式为:

VSAFE /1±/2±/3±/4±/5±/6±/7±/8±

对每项选“+”表示打开该选项, 选“-”表示关闭该选项。可把这种命令格式放在一个批处理文件中, 则只要运行该批文件, VSAFE 就驻留内存。如果不想使 VSAFE 驻留内存, 只要在激活控制窗口后, 同时键入 ALT+U 键, 就可将 VSAFE 由内存中撤去。

## 三 CPAV.EXE 的用法

CPAV.EXE 是一个可查消 500 多种病毒, 并采用对执行文件加免疫等手段来防止现有和将来可能产生的各种病毒侵袭的软件。键入 CPAV [驱动器号]并回车后, CPAV 首先查内存有无病毒或被非法减小。如未发现异常, 则显示主屏幕。屏幕最上边一行为主菜单, 有四个功能选项, 自左至右为检查项 (Scan)、选择项 (Options)、配置项 (Configure) 和帮助项 (Help), 下面一行反映当前驱动器和路径。其下面的三个小窗口自左至右分别为文件信息, 病毒信息和执行操作信息。

三个小窗口下面的两个大窗口中,左边为当前盘的目录树。右边为当前目录下的文件清单,最下一行是10个功能键选项。

主屏幕内各项的操作与 PCTools 6.0 版等软件的法大致相同。例如:击“Tab”键在目录树窗口与文件清单窗口之间进行切换。用↓、↑键选择子目录和文件。对选中的项目击回车键予以确认。在彩显中,选中的内容以蓝色显示;在单显中,选中的内容以高亮度显示。激活主菜单的方法有多种。常用的两种方法是:

(1)按 F10 功能键后,光条移到主菜单栏。用→、←键将光条移到欲选的功能项菜单上,击回车键或↓键调出下拉式菜单,然后用箭头键可在功能项菜单选项内和菜单之间进行选择。击回车键确认选择。

(2)对各选项直接击其异色或高亮度字母

下面对各主要选项予以说明

1. 选择功能菜单(Options)

共 11 个选项,均为开关项“√”表示打开选项,无“√”表示关闭选项。

(1)整体校验(Verify Integrity)

打开此项,查病毒时,将所查的执行文件与原来运行 CPAV 时建立的档案文件进行比较。发现有变化时,弹出如下窗口信息。

Verify Error			
File:EDLIN.COM has been changed.			
	From	To	
Attribute:	...	...	
Time :	12:00:00	10:00:00	
Date :	08/15/1989	07/24/1987	
Size :	7508	7495	
Checksum	FDCD	FD80	

Update Delete Continue Stop

选“Update”则以新信息代替老信息。以后再查病毒,查到此文件时将不再报警;选“Delete”则删除这个已变化的文件;选“Continue”继续执行;选“Stop”停止执行,返回到上一级菜单,该选项默认为打开状态。

(2)建立新的文件档案(Create New Checksums)

打开此项,每查完一个子目录,将在该子目录中建立一个名为 CHKLIST.CPS 的档案文件。这个文件含有该子目录中所有的执行文件的大小、属性、日期、时间等基本信息。如果目录中已有 CHKLIST.CPS 文件,对任何新增到该目录中的文件,再执行 CPAV 时,会将新文件的有关信息加到 CHKLIST.CPS 文件中。该选项默认为打开状态。

(3)在软盘中建立档案文件(Create Checksums on Floppy)

此项对软盘中每个子目录检查完后,也建立名为 CHKLIST.CPS 的档案文件。该选项默认为关闭状态。

(4)解除报警声响(Disable Alarm Sound)

默认为关闭状态,显示警告信息时无警告声响。

(5)建立备份(Create Backup)

打开此项,则任何被感染文件在被清除前都会被备份,备份文件扩展名为 VIR。打开这项意味着允许带毒文件仍存在盘上。只有当被感染文件是唯一而又是十分重要的时候,才可打开此项。

(6)建立报告(Create Report)

打开此项,则运行完 CPAV 后,在当前盘的根目录中产生一个名为 CPAV.RPT 的报告文件。通过查阅这个报告文件,可以了解在哪些文件中发现了哪些病毒。清除了多少病毒等信息。该选项默认为关闭状态。

(7)检查时随时反应(Prompt while Detect)

打开此项,一旦检查中发现病毒,立即显示一窗口,通过此对话窗口,可以修复(repair)被感染文件、不修复被感染文件继续执行 CPAV(continue)或停止执行 CPAV(stop)。该选项默认为打开状态。

(8)快速检查(Fast Detection)

打开此项时,CPAV 只检查病毒通常传染的部分而不是检查整个文件,实现快速查毒,一般几十兆至一百兆的硬盘几分钟内就可查完。如果对快速检查或对某些文件还不放心,可关闭该选项,那么对选中的文件,CPAV 将进行全部已知病毒特征码的检查,这样自然要多花些查毒时间。该选项默认为打开状态。

(9)查全部文件(Check All Files)

打开此项,则对全部文件进行查毒,关闭此项,只对扩展名为 EXE.COM.OVL.OVR.SYS.BIN.APP.CMP 这样的执行文件查毒,该选项默认为打开状态。

(10)只查毒(Detection Only)

打开此项,则涉及清病毒、加免疫等选项将失效,该选项默认为关闭状态。

(11)允许入网(Allow Network Access)

打开此项,可在网络中使用 CPAV。默认为打开。

2. 配置功能菜单(Config)共 6 个选项

(1)换至快速菜单(Change to Express Menu)

快速菜单是只包含五个命令选项的简化菜单,可完成 CPAV 的主要功能五个选择命令是:

a, Detect 只查毒

b, Detect & Clean 查并清毒

c, Select new drive 改变驱动器

d, Full Menus 回到主菜单栏

e, Exit 退出 CPAV

击 F8 功能键也可进入快速菜单。

(2)非免疫文件表(Immunization Exceptions)

CPAV 可对大多数执行文件加免疫,但为了防止对一些不适于加免疫的文件加免疫。CPAV 在执行加免疫前先查非免疫文件表。对表中已有的文件不再加免疫。选此项后,选“Add”向非免疫文件表中加入新的非免疫文件。选“Remove”去除某些非免疫文件,选“OK”确认选择。

(3)改变警告信息(Change Alert Message)

选此项可改变查到病毒时给出的警告信息

(4)改变口令设置(Change Password)



#### 〈5〉配置存盘(Save Configuration)

此项可对已选的选项和配置项的更改内容存盘。

#### 〈6〉改变驱动器(Change Work Drive)

此项可改变当前驱动器。击 F<sub>2</sub> 键作用同此项。

### 3. 检查功能菜单(Scan)

#### 〈1〉检查(Detect)

该选项只查而不清病毒,当发现病毒或某文件被更改了,将发出警告。F<sub>4</sub> 功能键与此项作用相同。

#### 〈2〉查消病毒(Clean)

选此项时,在查到病毒的同时进行消毒。若发现文件被更改了,将显示一对话框,可选“Update”承认这种变化合法。选“repair”修复该文件,选“Delete”删除该文件,F<sub>5</sub> 功能键与此项作用相同。

#### 〈3〉加免疫(Immunize)

选此项时,将对选定的文件加免疫,加了免疫的文件增加不到 1K 的长度。但并不占据系统内存空间。运行加了免疫的文件时,文件首先进行自检。若发现有变化,将给出下面的信息:

Self Integrity Check Warning—File was changed!

Choose an option

[R]Self Reconstruction

[C]Continue execution

[E]Exit to DOS

提示指出:经自检发现文件已变化了,要求做出选择。键入“R”文件自行恢复原状态,键入“C”不理睬这种变化,继续执行,键入“E”退出程序返回 DOS。

对不适于加免疫的文件将给出有关的信息。可把这类文件加到非免疫文件表中。

有时,某些文件被加了免疫后无法正常使用。此时可用撤除免疫选项将这类文件的免疫撤去。F<sub>6</sub> 功能键的作用与加免疫选项相同。

#### 〈4〉撤除免疫(Remove Immunization)

将加了免疫的文件的免疫功能撤除。

#### 〈5〉删除 CHKLIST. CPS 文件(Delete Checklist Files...)

选此项将把选定子目录中的 CHKLIST. CPS 文件删除。

#### 〈6〉病毒清单(Virus List)

此项被选择后,可列出 CPAV 能查消的 500 多种病毒的有关信息,F<sub>9</sub> 功能键与此项作用相同

#### 〈7〉增补病毒特征码(Update Virus List)

若得到由 Central Point 公司提供的新资料,通过选此项输入病毒特征码,可以查出更多种类的病毒。

#### 〈8〉操作日记(Activity Log)

选此项可显示或打印出最近一段时间的检查操作记录。F<sub>7</sub> 功能键与此选项作用相同。

#### 〈9〉退出 CPAV(Exit)

选此项后将弹出如下窗口提示:

Do you really want to exit?

[ ] Save Configuration

Exit Cancel

问你是否真想退出 CPAV,不想退出选“Cancel”

项。若想退出,选“Exit”项。在选“Exit”项以前,若已更改过配置而在配置功能菜单中未进行配置存盘,此时还有机会存盘。先击“S”键,然后再选“Exit”项。则可将配置存盘后再退出 CPAV。

### 4. 帮助功能菜单

此项给出一系列帮助信息。此外,无论 CPAV 运行至哪一步,击 F<sub>1</sub> 键均可得到对该步的帮助信息。

以上介绍了 CPAV 主屏幕中各菜单选项的用法及功能,在执行 CPAV 时,随时可击 F<sub>3</sub> 功能键或“Esc”键,则程序中断执行退回上一级菜单项直至最后退出 CPAV。

CPAV 也可采用命令参数的方式运行。如:

CPAV /S——查毒

CPAV /C——清毒

CPAV /I——查、清毒后加免疫

### 四、使用中的一些技巧。

首先,在确保系统干净无毒的情况下,用 BOOT-SAFE 建立引导区备份文件并将此备份文件存入一个干净无毒的软盘中备用。然后将 CPAV 的软件集中在一个子目录中,子目录名可叫做 CPAV。运行 CPAV,对盘中的文件,至少是像 DOS、PCTools 等这样一些大多数人都要用到的子目录中的文件加免疫。建立起执行文件的内防护层。而后,运行 CPAV 使每个子目录中都建立 CHKLIST. CPS 文件从而建立起文件的外防护层。以后即使不运行 VSAFE,只要文件被病毒感染了。通过运行 CPAV 就可查出。加了免疫的文件还可被恢复。

为了更有效地防病毒。可将三大核心软件加到 AUTOEXEC. BAT 文件的开头部分。使每次开机后,防病毒软件自动进入工作状态。推荐在 AUTOEXEC. BAT 文件开头部分采用如下命令格式:

Path=C:\CPAV

Bootsafe [驱动器号]

Vsafe /1+/2-/3-/4+/5+/6+/7-/8+

CPAV /S

开机后程序首先检查引导区是否完好,然后 Vsafe 进驻内存实现对病毒行为的动态监视。最后,CPAV 将盘扫一遍,看看盘内文件是否有过任何变化。

对上面的 Vsafe 选项,由于第 1,4,5,6,8 项均为打开状态。所以,一旦出现硬盘将被低级格式化、执行文件带毒、发现引导区型病毒、硬盘引导区将被更改、执行文件将被更改等情况时,就会出现警告信息。出现警告信息窗口后,窗内有不同的选项供选择。以执行文件将被修改为例,三个选项是:“Stop”,选此项则模拟写保护,许多病毒在遇到写保护时则放弃侵犯,所以若选“S”后能通过,则可继续程序的执行。但也有些病毒,遇到写保护时将反复尝试写盘或出现其它一些不正常的情况,如果遇到这种情况,只好选“Boot”项重新启动系统先查完病毒再运行文件了。还有一种情况,程序被修改并不是由于病毒作用而是由于某些正常原因,如使用 DEBUG,用 PCTools 压缩某文件等,此时

可选另一个选项“C”。表示这种更改属合法现象,要求继续执行,不过,如果你正用 PCTools 压缩某个子目录的许多文件时,则每执行一个文件压缩就报警一次,那就太麻烦了,此时,最好先关闭第 8 个选项。待文件被压缩完再将第 8 个选项打开。

第 2、3、7 项选关闭状态是因为:对第 2 项、如选打开则只要程序要驻留内存就会报警,因而在运行一些中文软件如“2.13”“WPS”等时,会出现误报。并且,由于某些中文软件与西文软件不完全兼容,还会出现不显示提示信息甚至死机现象。关闭此选项后,虽然病毒有可能被允许进驻内存,但只要其它几项,特别是第 8 项关口把牢,就不会产生什么危害。还有一点需要说明的是,由于中文系统采用图形方式显示,在中文系统下执行程序时,如遇到警告情况,报警信息常不能显示出来,但会有报警声响。比如在“2.13”下,运行 WS.COM,假如病毒感染了该文件,你会听到报警声而见

不到报警信息。对这类情况,可按 Ctrl+F7 键转至西文状态或退出中文系统直接在西文状态下再运行该文件,就可看到报警信息,从而判断是否有病毒出现。

VSAFE 中第 3 项选关闭是因为若将其选打开则任何文件拷贝时均要报警。第 7 项由于是保护软盘引导区,不属于保护硬盘范畴,故取关闭,否则当格式化软盘等操作时均会产生误报,这个选项可根据需要临时打开。例如,当您准备用软盘传递一些文件时,打开此项可避免病毒进入软盘引导区。

当 VSAFE 驻留内存时,对程序的执行速度可能会产生一定的影响。一般情况下,这种影响并不明显,如果是进行某些大型运算。可采用下面的方式:在 VSAFE 驻留内存的情况下,先运行一下执行文件。未见异常则中断程序运行。激活控制窗口后,按 ALT+U 键撤掉 VSAFE。然后再重新运行该执行文件,这样就可保证运算速度不受影响了。

## 带检索功能的通讯录录入程序

武汉市同济医科大学研究中心(430030)张春明

在数据库管理系统中,录入模块是必不可少的。当数据量很大时,难免录入重复的记录项,这样势必加大数据库的冗余度。

笔者编写了一个带检索功能的通讯录录入程序,可以有效地防止重复数据的录入,基本编程思想是:建立两个结构相同的数据库,一个作为主库(TXL.DBF),并以姓名为关键字建立索引文件;另外一个作为录入当前库(HYTX.DBF)。录入时首先输入姓名,如果在主库中检索到相同姓名的人员,则将该记录的全部内容用 REPL 命令赋值予当前库的各个字段,然后可在屏幕上修改各个字段的内容;反之则在两库中分别追加一个空记录,然后输入各个字段的内容。为了减少输入量,将前一记录中若干字段的值预先赋给当前记录相应的字段。尽管在当前库中可能出现相同姓名的人员(也可能不是同一人),但在主库中不会出现相同姓名的人员。

这段程序尤其适合于登记会议通讯录。因为在专业或行业会议中,许多人都是前次会议的参加者,一旦输入其姓名,相应的资料就调出,大大方便了输入操作。只要没有重新登记,当前库中的人员就是参加会议的人员。如需打印会议通讯录,只要在当前库中指定所需的字段进行打印即可。登记和打印结束后,可将当前库清空或删除,或者在下次运行程序中回答询问“重新开始登记吗?”时键入 Y 也可。

```
set talk off
set inte off
sele 2
use txl inde txl
```

```
sele 1
use hytx
set rela to 姓名 into b
go bott
clea
y='Y'
@ 4,5 say '重新开始登记吗? (Y/N)' get y
read
if y$ 'yY'
set safe off
zap
set safe on
endi
xb=性别
zc=职称
dw=单位
dz=地址
yb=邮编
dh=电话
appe blan
repl 性别 with xb,职称 with zc,单位 with dw
repl 地址 with dz,电话 with dh,邮编 with yb
do while. t.
clea
@1,9 say recn()
@2,13 say '姓名:'
@3,13 say '性别:'
@4,13 say '专业职称:'
@5,13 say '行政职务:'
@6,13 say '单位名称:'
@7,13 say '电话号码:'
```



```

@8,13 say '单位地址;'
@9,13 say '邮政邮编;'
@10,13 say '备注;'
@2,23 get 姓名
read
xb=姓名
sele 2
seek xm
if eof()
  appe blan
sele 1
else
  sele 1
repl 性别 with b->性别, 职称 with b->职称
repl 职务 with b->职务, 单位 with b->单位
repl 地址 with b->地址, 电话 with b->电话
repl 邮编 with b->邮编
repl 备注 with b->备注
endi
@3,23 get 性别
@4,23 get 职称
@5,23 get 职务
@6,23 get 单位
@7,23 get 电话
@8,23 get 地址
@9,23 get 邮编
@10,23 get 备注
read
if len(trim(姓名))=0
  xh=recn()
  dele for recn()=xh
  pack

```

```

endi
@12,5 say '是否登记下一位?(Y/N)' get y
read
sele 2
repl 姓名 with a->姓名, 性别 with a->性别
repl 职称 with a->职称, 职务 with a->职务
repl 单位 with a->单位, 电话 with a->电话
repl 地址 with a->地址, 备注 with a->备注
repl 邮编 with a->邮编
if len(trim(姓名))=0
  xh=recn()
  dele for recn()=xh
  pack
endi
@12,0 clea
if . not. y $ 'Yy'
  clos data
  clear
  exit
endi
sele 1
xb=性别
zc=职称
dw=单位
dz=地址
yb=邮编
dh=电话
appe blan
repl 性别 with xb, 职称 with zc, 单位 with dw
repl 地址 with dz, 电话 with dh, 邮编 with yb
endd

```

(上接第2页)

行平行处理。如：四个 C-Cube CL550-35 芯片能实时压缩或还原高清晰度 TV (HDTV) 视频路象。

问：如何将 C-Cube CL550 处理器安装在系统中？

答：C-Cube CL550 能嵌入任何处理装置内，如计算机外设（图象板、彩色打印机、彩色复印机和扫描仪）、彩色传真机；数字式相机；数字 VCR 和桌面系统中。

问：C-Cube CL550 有哪些开发工具？

答：C-Cube CL550 开发系统板/M 是一种 OEM 开发卡，可插入 Mackintosh I 系列计算机的总线插座。C-Cube CL550 开发板/PC，也是一个 OEM 开发卡，采用 ISA 总线，用于 IBM PC/AT 及其兼容机。它们是了解 CL550 性能及开发 CL550 应用的有力工具。

问：C-Cube CL550 开发工具都包括些什么？

答：C-Cube CL550 开发工具/M 包括 Mackintosh I 兼容的 C-Cube CL550 开发板；软件驱动程序；demo 软件包等和一份压缩监控程序的拷贝件。

C-Cube CL550 开发工具/PC，包括用于 ISA 总线的开发板；软件驱动程序；demo 软件包和一份压缩监控程序的拷贝件。

问：目前国内何处可咨询 C-Cube CL550 方面的问题及订货？

答：请同中国电子器材深圳公司联系。

地址：广东省深圳市振华路 414 幢电子器材大厦

电话：354214、354345、350877 电挂：8410

传真：350876 邮编：518031

联系人：黄建新

# PC 机中系统时间的自动显示

北京六十一中(100007) 郑嘉琦

这里向大家介绍一个名为 DTIME.COM 的程序。可当 DOS 外部命令使用,只要执行一次之后,立即开始将系统时间的时、分、秒在屏幕右上角自动显示。在走时显示的同时,计算机完全可以照常使用。

在 IBM PC 机的中断系统中有一个 INT 1C,它被放置在8号硬件中断服务程序的结尾处。DOS 在初始化时,使其中断向量指向一个不执行任何任务的例程(它只包含一条 IRET 指令)。这是有意留给用户用来扩展8号中断的功能而专门安排的。由于8号中断固定每55毫秒发生一次,因而利用 INT 1C 可以实现一些实时操作功能。这里利用它进行计时并加以自动显示。具体做法是:编制一个自动显示时间的新例程,先让其常驻内存,然后修改 INT 1C 的中断向量使指向新例程。一般可以利用 INT 27或功能调用31H 让程序常驻内存。本文中自动显示时间的新例程较为短小,就用了将其放到中断向量表的空闲部分的方法来实现常驻。具体存放地址为0:200H~0:294H。另外0:295H,296H,297H 三个单元作为计时器,以 BCD 码的形式依次存放时、分、秒数;0:298H 作为计数器,用来计8号中断发生的次数,每计满18次作为一秒。由于8号中断实际每秒发生18.2次,因此例程中在每分钟和每小时的最后一秒进行一次校正调节。

DTIME.COM 用 DEBUG 的 A 命令直接生成最为简便。下面的清单除中文注释之外,既列出了程序内容,也表明了 DTIME.COM 生成的方法过程。例程是针对单色显示器编制的。如果用彩色显示器。只需将 08F1:014D 处的代码改为 MOV AX,B800。

如果中断向量表中的0:200H~0:298H 共153个单元有使用安排,需要将 DTIME.COM 做简单修改,改用 INT 27或功能调用31H 使新例程常驻内存,新例程本身无需修改。

A>DEBUG

-A100

```
08F1:0100  XOR AX,AX      ;将自动显示时间的新例
08F1:0102  MOV ES,AX      程存放中断向量表的
08F1:0104  CLD          空闲部分常驻。
08F1:0105  MOV SI,14A
08F1:0108  MOV DI,0200
08F1:010B  MOV CX,0095
08F1:010E  REPZ
08F1:010F  MOV SB
08F1:0110  MOV AH,2C      ;取当前时间。
08F1:0112  INT 21
08F1:0114  MOV BX,0295;    ;将当前时间的时、分、秒
08F1:0117  XOR AX,AX      作为初值赋给计时器
08F1:0119  MOV DS,AX      0:295,296,297
08F1:011B  MOV AH,CH
```

```
08F1:011D  CALL 013A
08F1:0120  MOV AH,CL
08F1:0122  CALL 013A
08F1:0125  MOV AH,DH
08F1:0127  CALL 013A
08F1:012A  MOV AL,12      ;给计数器0:298赋初值。
08F1:012C  MOV [0298],AL
08F1:012F  MOV DX,0200    ;改变 INT 1C 的中断向量,
08F1:0132  MOV AL,1C      使其指向新例程。
08F1:0134  MOV AH,25
08F1:0136  INT 21
08F1:0138  INT 20;        ;结束。
08F1:013A  XOR AL,AL      ;将 AH 中的数码转换
08F1:013C  AND AH,AH      成 BCD 码并存入内存
08F1:013E  JZ 146         的子程序。
08F1:0140  INC AX
08F1:0141  DAA
08F1:0142  DEC AH
08F1:0144  JNZ 0140
018F:0146  MOV [BX],AL
018F:0148  INC BX
08F1:0149  RET
08F1:014A  PUSH BX        ;从此至1DE 是自动显示
08F1:014B  PUSH ES        ;时间的新例程。
08F1:014C  PUSH DI
08F1:014D  MOV AX,B000    ;将计时器中存放的时、
08F1:0150  MOV ES,AX      分、秒转换成 ASC II 码
08F1:0152  XOR AX,AX      后显示到屏幕右上角。
08F1:0154  MOV DS,AX
08F1:0156  MOV DI,008E
08F1:0159  MOV DH,70
08F1:015B  MOV BX,0294
08F1:015E  CALL 01BE
08F1:0161  MOV DL,2E
08F1:0163  CALL 01D9
08F1:0166  CALL 01BE
08F1:0169  MOV DL,3A
08F1:016B  CALL 01D9
08F1:016E  CALL 01BE
08F1:0171  MOV AL,[0298] ;计数器298减一计数,
08F1:0174  DEC AL        每减到零为一秒并重
08F1:0176  MOV [0298],AL 新赋初值
08F1:0179  JNZ 01BA
08F1:017B  MOV AL,12
08F1:017D  MOV [0298],AL
08F1:0180  MOV AL,[0297] ;计时器计秒。
08F1:0183  INC AL
08F1:0185  DAA
08F1:0186  MOV [0297],AL
08F1:0189  SUB AL,60      ;满60秒吗?
```

```

08F1:018B  JNZ 01BA
08F1:018D  MOV [0297],AL    ;满,计秒单元回零。
08F1:0190  MOV AL,1E;进行一次校正调节(粗调)*。
08F1:0192  MOV [0298],AL
08F1:0195  MOV AL,[0296]    ;计时器计分。
08F1:0198  INC AL
08F1:019A  DAA
08F1:019B  MOV [0296],AL
08F1:019E  SUB AL,60        ;满60分吗?
08F1:01A0  JNZ 01BA
08F1:01A2  MOV [0296],AL    ;满,计分单元回零。
08F1:01A5  MOV AL,18;进行一次校正调节(细调)*。
08F1:01A7  MOV [0298],AL
08F1:01AA  MOV AL,[0295]    ;计时器计小时。
08F1:01AD  INC AL
018F1:01AF  DAA
08F1:01B0  MOV [0295],AL
08F1:01B3  SUB AL,24        ;满24小时吗?
08F1:01B5  JNZ 01BA
08F1:01B7  MOV [0295],AL    ;满,计小时单元回零。
08F1:01BA  POP DI
08F1:01BB  POP ES
08F1:01BC  POP BX
08F1:01BD  IRET            ;中断返回(到 INT8的结尾处)
08F1:01BE  INC BX          ;将 AL 中的 BCD 码转换
08F1:01BF  MOV AL,[BX]      成 AX 中的两个 ASC II 码
08F1:01C1  MOV AH,AL        并显示到屏幕右上角的
08F1:01C3  AND AL,0F        子程序。

```

```

08F1:01C5  SHR AH,1
08F1:01C7  SHR AH,1
08F1:01C9  SHR AH,1
081:01CB   SHR AH,1
081:01CD   OR AX,3030
08F1:01D0  MOV DL,AH
08F1:01D2  ES;
08F1:01D3  MOV [DI],DX
08F1:01D5  INC DI
08F1:01D6  INC DI
081:01D7   MOV DL,AL
08F1:01D9  ES;
08F1:01DA  MOV [DI],DX
08F1:01DC  INC DI
08F1:01DD  INC DI
08F1:01DE  RET
08F1:01DF  ↙

```

—N DITIME.COM

—RCX

CX 0000

:DF

—W

;将 DITIME.COM 的长度  
放入 CX 寄存器

;将 DTIME.COM 存盘备  
用。

Writing 内 00DF bytes

\*注:走时的快慢可通过改变 0:247H 和 0:25CH 两个单元中的数码来调节。

## 最近出现的几种病毒

中国地质大学(北京)(100083) 苏民生

近期出现几种病毒,有些是现有检查与消除病毒软件(如 KILL V50.00)尚不能查出和消除的,已造成一些破坏,需引起注意。

### 一、GenB/GenP 病毒

系统型,改变软盘 BOOT 区及硬盘主引导记录(0面0道1区)。病毒部分占据第0—1BD字节,其中第1B0—1BD字节内容为字符串 RMBDRMCCQBDWRM,似为明显标志。硬盘感染后,仍能启动。感染软盘后,软盘不能启动,使用该软盘引导系统时,先执行占用了 BOOT 区的病毒程序,然后转到用硬盘引导。可导致失去硬盘。

KILL V50.00不能检出该病毒。版本8.4B89的SCAN可查出,定名为GenP(对于硬盘)和GenB(对于软盘)。同一版本的CLEAN可消除,但有时消除失败。

### 二、DIR—2(D2)病毒

文件型,感染COM文件和EXE文件。不改变文件大小和形成(最后修改)日期。驻留内存。用8.4B89版本的SCAN和CLEAN可检查与消除。破坏文件分

配表,使文件损坏或丢失。用KILLV50.00检不出。用CLEAN消除之前必须关掉电源后用好的系统盘重新冷启动。感染病毒的文件往往无法复制(COPY)。

### 三、779/5病毒

文件型。感染COM文件和EXE文件。对于COM文件,改变后来文件前16字节,并将病毒部分779(30BH)或更多字节接在原来文件之后。对于EXE文件,仅在原来文件之后续接5字节或稍多字节内容。不改变原来文件内容。往往修改文件形成日期,有时又不修改。带病毒COM文件开头为E9XXXX00YYYY2219……,其中XXXX和YYYY是和原来文件长度有关的数据。现有检查和消除病毒软件尚不能判断和消除该病毒。尚未发现破坏情况。

### 四、901/910病毒

文件型。感染EXE文件,使其增大901至910字节。改变文件日期。除发现文件变大和改变日期之外,未发现其它破坏。现有检查和消除病毒软件尚不能查出。



# 电子扭计板

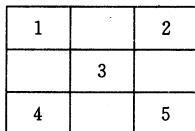
大庆市二十四中计算机室(163712) 葛建华 吴立国

扭计板是80年代风靡美国的智力玩具。其玩法类似于魔方,实际上是魔方的一种平面化方式,只是难度稍低。我们在PC机的2.13操作系统下用BASIC语言编写了一个电子扭计板的程序,操作方式与实际的扭计板一样,其游戏规则如下:

首先程序设定初始状态,如图(1),左上角和右上角分别为两个三角形和两个圆形方阵,中间为两个圆形和两个菱形方阵,左下角和右下角分别为两个菱形和两个正方形方阵。这五个方阵分别以1~5键定义,如图(2),每个方阵均可顺时针或逆时针方向旋转,每旋转一次为一次操作,并在屏幕上显示操作次数。由“K”键定义为顺逆时针旋转的转换键,并由“=”或“<=”符号提示。由“R”键定义为返回初始状态。要求通过这五个方阵的旋转,将图形的初始状态旋转到目标状态,如图(3),图(4),图(5),图(6)等多种类型。



图(1)



图(2)



图(3)



图(4)



图(5)



图(6)

由不尽相异元素的排列公式可算出共有 $16! \div (4! \times 4! \times 4! \times 4!) = 63063000$ 种排列方式,趣味无穷,最后可由“Q”键退出。

```
15 CLS:SCREEN 1,0:P=1:DIM A(4,4)
22 C$(1)="▲":C$(2)="●":C$(3)="◆":C$(4)
   ="■"
28 REM 30~60行是初始状态的设定
30 FOR I=1 TO 4:FOR J=1 TO 4
50 A(I,J)=I
60 NEXT J,I
65 M1=1:N1=1:M2=4:N2=4
70 GOSUB 5000
100 A$=INPUT$(1)
102 IF A$="q" THEN CLS:END:REM 按“Q”键退出
104 IF A$="r" THEN Z=0:GOTO 15
108 REM “k”键是方阵顺时针和逆时针旋转的转换键
```

```
110 IF A$="k" THEN P=P*(-1):IF P=1 THEN
   LOCATE 4,32,0:PRINT "<="ELSE LOCATE 4,
   32,0:PRINT "=>"
118 REM 1~5的数码键分别是左上,右上,中间,左下,右
   下5个方阵的操作键
120 IF VAL(A$)>5 OR VAL(A$)<1 THEN 100
122 Z=Z+1
130 ON VAL(A$) GOSUB 200,300,400,500,600
135 M2=M1+1:N2=N1+1
140 GOSUB 1000:GOTO 100
200 X1=1:X2=2:Y1=1:Y2=2:M1=1:N1=1:RE-
   TURN
300 X1=3:X2=4:Y1=1:Y2=2:M1=1:N1=3:RE-
   TURN
400 X1=2:X2=3:Y1=2:Y2=3:M1=2:N1=2:RE-
   TURN
500 X1=1:X2=2:Y1=3:Y2=4:M1=3:N1=1:RE-
   TURN
600 X1=3:X2=4:Y1=3:Y2=4:M1=3:N1=3:RE-
   TURN
1000 T=0:FOR I=Y1 TO Y2
1002 FOR J=X1 TO X2
1004 T=T+1:B(T)=A(I,J)
1006 NEXT J,I
1010 IF P=1 THEN GOSUB 2000
1020 IF P=-1 THEN GOSUB 3000
1030 GOSUB 5000:RETURN
2000 T=0:REM 逆时针旋转子程序
2010 FOR J=X1 TO X2:FOR I=Y2 TO Y1 STEP-1
2030 T=T+1:A(I,J)=B(T)
2050 NEXT I,J
2100 RETURN
3000 T=0:REM 顺时针旋转子程序
3010 FOR J=X2 TO X1 STEP-1:FOR I=Y1 TO Y2
3030 T=T+1:A(I,J)=B(T)
3050 NEXT I,J:RETURN
5000 REM 打印子程序
5002 COLOR 9,0
5010 FOR I=M1 TO M2:FOR J=N1 TO N2
5030 LOCATE I*2,5*J,0:PRINT C$(A(I,J)),
5040 NEXT J:PRINT:PRINT
5060 NEXT I
5070 LOCATE 10,10,0:PRINT "操作次数:",Z,
5080 IF P=1 THEN LOCATE 4,32,0:PRINT "<=":
   RETURN
5090 LOCATE 4,32,0:PRINT "=>":RETURN
```

# 第四届国际信息学奥林匹克竞赛试题

## (第一轮竞赛试题)

### 海中的岛

用  $N \times N$  的网格表示海, 网格中的 \* 表示岛, 我们的任务是重构一张给定的地图。地图中岛的组成由垂直方向和水平方向的编码信息给出, 下例是一张地图的编码。

```

*   *   *       1 2
  *   *   *   * 3 1
*   *       *   1 1 1
  *   *   *   * 5
*   *   *   * 2 1 1
      *       1
1 1 4 2 2 1
1 2 3 2
1
    
```

岛可以编组, 称为“岛组”, 上图每行右边的几个数字, 自左至右依次表示各“岛组”的大小, 例如第一行数字 1、2 表示该行有两个“岛组”, 第一个“岛组”有一个岛, 第二个“岛组”有两个岛。每个“岛组”的左右都是海, 海的宽度假定是任意的。类似地, 第一列下边的数字 1、1、1 表示本列自下而上有三个“岛组”, 每个“岛组”仅含一个岛。

#### 任务:

试编一个程序, 重复以下步骤, 直到给定的输入文件 (ASCII 文件) 中的信息组全部读完为止。

1. 从输入文件中读入一个信息块, 并将其显示到屏幕上。每个信息块的组成如下: 网格大小  $N$ , 其后是  $N$  行“岛组”编码数据; 再后面是  $N$  列“岛组”编码数据, 数据中数字间应有空格用来分开, 最后用 0 作为每行或每列数据结束标志。

#### 举例:

6                      例1. 对应上面的地图,  $N=6$

1 2 0                  ← 第一行数据

3 1 01 1 1 0

5 0

2 1 1 0

1 0

1 1 1 0              ← 第一列数据

1 2 0

4 0

2 3 0

2 0

1 2 0

4

例2. 解为:

列

1 0                  1 2 3 4

2 0                  1

0

0                  行 2                  \*

1 0                  3                  \* \*

2 0                  4

0

2                      例3. 注意: 据此数据, 构不成地图

0

0

2 0

2 0

2                      例4. 注意: 据此数据, 可以构成两张不同的地图

1 0

1 0

1 0

1 0

2. 重构这张地图并将其显示到屏幕上, 若有多个解, 要构成每张图, 参看例4。

3. 将所构成的地图以 ASCII 文件形式输出, 每个岛用“\* 后跟空格”表示; 空白用连续两个空格表示, 若有多个解, 其间用空行隔开, 若无解 (即构不成地图), 则用一行文字“no map”表示。

由不同信息组所求得解用一行文字“Next Problem”隔开。

#### 技术限制:

1.  $1 \leq N \leq 8$

2. 源程序名为“C:\IOI\DAY-1\413-PROG.PAS”

3. 用于读入的带有岛图的编码信息输入文件名为“C:\IOI\DAY-1\413-SEAS.IN”

4. 将地图写成 ASCII 文件的输出文件名为“C:\IOI\DAY-1\413-SEAS.OU”

#### 文件举例:

为你方便, 这里提供了举例文件, 其中含测试数据及正确的结果输出, 这些文件在

“C:\IOI\DAY-1\413-SEAS.IN”和

“C:\IOI\DAY-1\413-SEAS.OU”中。

警告: 倘若你的程序能够正确运行上述例子, 但这并不能保证你的程序是正确的。

#### 评分标准:

• 由输入文件读一个信息块并加以显示 5分

- 一个接一个地处理所有的信息块,直至输入文件被读完为止。 10分
- 对每一个信息块,倘若有解就重构一张地图并加以显示 35分
- 将解出的地图写入输出文件 5分
- 倘若有不同的解,重构所有可能的地图并显示之。 20分
- 将全部解的图按要求分开写到输出文件 10分
- 标出无解的信息块 5分
- 遵守技术限制条件 10分

满分 100分

## 第二轮竞赛试题

### 爬山

某爬山俱乐部有  $P$  个爬山队员,分别编号为 1, 2, ...,  $P$ 。每个爬山队员的爬山速度都相同;且上山、下山的速度也相同。编号为  $i$  的爬山队员每天的物资消耗量为  $C(i)$ ,其最大携带物资的能力为  $S(i)$ 。 $C(i)$  和  $S(i)$  均为整数,计量单位相同。

假定一个爬山队员在有足够供应物资的情况下,爬到山顶所需时间为  $N$  天。山可能非常高,单个一人爬山没有能力带足所需的物资。因此,要一组爬山队员同时同地出发,某些队员在到达山顶之前提前下山,以便把他所不再需要的物资转交给其他队员,所有队员在爬山时不得停留。

本题的任务是给爬山俱乐部制定一个爬山计划,其要求是:至少有一个队员必须到达山顶;被选中的爬山队员(组成爬山小组)必须全部回到出发地点。

#### 任务:

编写一个程序,完成如下任务:

1. 从键盘读入  $N$  (从出发点到达山顶的天数) 和  $P$  (爬山俱乐部成员个数);以及  $S(i)$  和  $C(i)$ , ( $i$  从 1 到  $P$ )。你可以假定所有输入数据均为整数。若输入的数据没有意义,拒绝之。

2. 制定一个爬山计划:选择可能的人组成爬山小组,该组成员为  $a(1), a(2), \dots, a(j), \dots, a(k)$ , 出发时队员  $a(j)$  携带的物资为  $M(j)$ , ( $j$  从 1 到  $k$ )。

请注意,对于给定的  $N, S(i)$  和  $C(i)$ , 也许并不存在能够完成任务的爬山计划。

3. 在屏幕上输出以下信息:

- a. 给出实际爬山小组中的队员数  $k$ ;
- b. 全组所需物资总量;
- c. 组中爬山队员的编号  $a(1), a(2), \dots, a(k)$ ;
- d. 每一个爬山队员  $a(j)$  在出发时所携带的物资数量  $M(j)$ , ( $j$  从 1 到  $k$ );
- e. 爬山队员  $a(j)$  爬了几天后开始下山, ( $j$  从 1 到  $k$ )。(参考前页中的例子,用它的输出格式)

4. 使爬山计划尽可能接近最优。优化的含义是:

- a. 参加爬山的队员人数最少,而且
- b. 在满足条件 a 的情况下,总的物资消耗量最小。

#### 技术限制条件:

限制条件1. 你的源程序文件名应为:

"C:\IOI\DAY-2\422-PROG. PAS"

限制条件2

当  $N$  小于 1 或大于 100 时,程序必须拒绝之;

$P$  的允许范围是:  $1 \leq P \leq 20$ 。

举例:

以下对话方式是适当的,请采用之。

Days to arrive to top: 4

Number of club members: 5

Maximal supply for climber 1: 7

Daily consumption for climber 1: 1

Maximal supply for climber 2: 8

Daily consumption for climber 2: 2

Maximal supply for climber 3: 12

Daily consumption for climber 3: 2

Maximal supply for climber 4: 15

Daily consumption for climber 4: 3

Maximal supply for climber 5: 7

Daily consumption for climber 5: 1

2 climbers needed, total amount of supplies is 10.

Climber(s) 1, 5 will go.

Climber 1 carries 7 and descends after 4 day(s)

Climber 5 carries 3 and descends after 1 day(s)

Plan another party (Y/N) Y

Days to arrive to top: 2

Number of club members: 1

Maximal supply for climber 1: 3

Daily consumption for climber 1: 1

Climbing party impossible.

Plan another party (Y/N) N

Good bye

#### 文件举例:

为使你方便,我们提供了举例文件,其中含测试数据及正确结果输出,请在目录

"C:\IOI\DAY-2" 中查找这些文件。

警告:倘若你的程序能够正确运行上述的例子,但这并不能保证你的程序是正确的。

#### 评分标准:

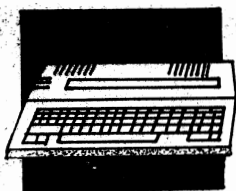
- 按上述例子所提供的方式和用户进行对话: 10分
- 在下述特殊情况下找到一个解,其条件是:
  - 所有  $C(i) = 1$ ; 所有  $S(i)$  都相等。 20分
- 在一般情况下找到一个解。 20分
- 在一般情况下找到一个接近最优的解 30分
- 能够发现无解的情况 10分
- 遵守技术限制条件 10分

满分

100分

(此文由清华大学计算机系吴文虎教授提供)





# 微机模拟游标卡尺读数训练

## 学习机之友

宁夏中卫县宣和中学(751706) 王太文

利用游标卡尺进行长度测量是中学物理中所必须掌握的基本技能。由于实际的游标卡尺较小,刻度较密,在课堂上教学生学会使用是很困难的,特别是演示读数无法进行,为配合本节课的教学,我用 BASIC 语言设计了微机模拟游标卡尺读数训练的实用程序,在 APPLE II 机上运行通过,能显示出用十分度游标卡尺测量五种不同形状几何体长度的放大的图形,形象逼真,效果很好。

### 一、运行说明:

程序运行后,图形显示出游标卡尺使用时的初始(读数为零)状态,文本窗显示“PLEASE GIVE ‘FEN—DU’”,要求输入屏幕所显示的游标卡尺的分度(如0.1、0.05、0.02等)。分度给错了,文本窗显示“NO, TRY AGAIN.”告诉你“不对,再试一次”;分度给对了,文本窗显示“YES VERY GOOD! Z=10”,即“对了,很好!得10分”。接下来显示出用游标卡尺测不同形状几何体长度的图形,文本窗显示“L=?”,要求输入读数,读对了,加10分,变换画面(长度)继续进行;读错了,减10分,要求重新读数,直到读对为止。程序运行一次,满分为100分(共10个画面)。

### 二、程序说明:

程序中的 Z 表示分数, M 表示游标的零刻度线所在位置。30—90语句完成初始状态的图形显示及输入数据的处理。100—180语句完成测量过程的图形显示及读入数据的处理,其中110语句利用随机数使游标出现在屏幕上10—180以内的任意位置(以游标的零刻度线为准),120语句根据730子程序给出的 K 值调用800—1100五个子程序之一,画出被测量的一种几何图形。

500子程序画出主尺及其刻度、数字。

600子程序画出游标及其刻度、数字。

730子程序将测量范围分为五个区域,每个区域内测量一种几何体的长度(给开关提供五个数码)。

800子程序画出一个圆,测其直径。

900子程序画出一个“桶”,测其内径。

950子程序画一个直角梯形,测其高。

1000子程序画一个等腰三角形,测其高。

1100子程序画一个矩形,测其长。

该程序是笔者在宁夏教育学院继续教育微机培训班学习的结业程序,指导教师李更生。

```
30 Z=0:HGR:HCOLOR=3:GOSUB 500
40 M=10:HCOLOR=3:GOSUB 600
50 PRINT"PLEASE GIVE 'FEN—DU'":INPUT FD
60 IF FD=0.1 THEN 80
70 PRINT "NO, TRY AGAIN":CALL 64477:GOTO 50
80 Z=Z+10:PRINT"YES, VERY GOOD!", "Z=", Z
```

```
90 HCOLOR=0:GOSUB 600:GOSUB 500
100 FOR T=1 TO 9
110 M=10+INT(170 * RND(1)):HCOLOR=3:GOSUB
    500:GOSUB 600:GOSUB 730
120 ON K GOSUB 800,900,950,1000,1100
130 S=(M-10)/100
140 PRINT"L=?":INPUT L:IF L>S-0.005 AND L<S
    +0.005 THEN 160
150 Z=Z-10:PRINT"NO, TRY AGAIN", "Z=", Z:
    CALL 64477:GOTO 140
160 Z=Z+10:PRINT"YES, VERY GOOD!", "Z=", Z
170 HCOLOR=0:GOSUB 500:GOSUB 600
180 ON K GOSUB 800,900,950,1000,1100:NEXT T:
    END
500 HPLLOT 10,80 TO 279,80
510 HPLLOT 279,65 TO 18,65 TO 18,55 TO 11,50 TO 11,
    65 TO 1,65 TO 1,130 TO 9,140 TO 9,95 TO 279,95
520 FOR I=10 TO 279 STEP 10
530 IF (I-10)/50=INT ((I-10)/50) THEN 550
540 HPLLOT I,75 TO I,80:GOTO 560
550 HPLLOT I,71 TO I,80
560 NEXT I
570 HPLLOT 107,68 TO 107,73
580 HPLLOT 206,68 TO 208,68 TO 208,71 TO 206,71 TO
    206,73 TO 208,73
590 RETURN
600 HPLLOT M,64 TO M,50 TO M-8,55 TO M-8,64
610 HPLLOT M-5,81 TO M-5,97 TO M,97 TO M,140
    TO M+8,130 TO M+8,97 TO M+95,97 TO M+
    95,81
620 FOR J=M TO M+90 STEP 9
630 IF J=M OR J=M+45 OR J=M+90 THEN 660
640 HPLLOT J,81 TO J,85:GOTO 670
660 HPLLOT J,81 TO J,88
670 NEXT J
680 HPLLOT M-1,89 TO M-1,94 TO M+1,94 TO M+
    1,89 TO M-1,89
690 HPLLOT M+46,89 TO M+44,89 TO M+44,91 TO
    M+46,91 TO M+46,94 TO M+44,94
700 HPLLOT M+89,89 TO M+89,94 TO M+91,94 TO
    M+91,89 TO M+89,89
710 RETURN
730 IF M<=40 THEN K=1:GOTO 760
732 IF M<=70 THEN K=3:GOTO 760
733 IF M<=100 THEN K=2:GOTO 760
735 IF M<=130 THEN K=4:GOTO 760
736 K=5
760 RETURN
800 R=(M-10)/2
810 FOR O=0 TO 360 STEP 9
```

```

820 X=10+R+R * COS(O * 3.1416/180);Y=129+R *
    SIN(O * 3.1416/180)
830 H PLOT X,Y;NEXT O
840 RETURN
900 H PLOT 10,60 TO 10,5 TO M+1,5 TO M+1,60
910 RETURN
950 D=M-10
960 H PLOT 10,130 TO 10,130-D/4 TO M-1,130-D/4

```

```

    TO M-1,130 TO 10,130+D/4 TO 10,130
970 RETURN
1000 H PLOT 10,100 TO 10,160 TO M,130 TO 10,100
1050 RETURN
1100 H PLOT 10,120 TO 10,150 TO M,150 TO M,120
    TO 10,120
1150 RETURN

```

## 磁带虚拟磁盘的文件管理

中国农科院蜜蜂研究所(100093) 胡发新

中华学习机已有相当大的数量进入家庭,但购买磁盘驱动器的还不多,一般家庭还是用磁带机做外设存、取信息。一盘磁带可存、取很多文件,时间一久,有的文件不易找到,很是费劲。作者为了解决这问题,编制了一种文件管理的方法,现介绍如下供参考:

首先根据《电子与电脑》91年第7期“为无驱动器系统设置虚拟磁盘”一文的程序,建立一个虚拟磁盘。然后在磁带的每一面开头一段磁带,以磁带机的计数器记数,空出000—060圈,建立虚拟磁盘文件总目录区。在060圈以后作为每个虚拟磁盘的文件存、取区(见程序1)。在每一个虚拟磁盘内最后几个扇区,作为本盘的文件目录扇区(见程序2),其他作为文件存、取扇区。

在建立每一个虚拟磁盘目录时,要从磁带的60圈开始,有计划地把每个文件键入或读入虚拟磁盘。例如要在A号盘装入以下文件,A1光子飞船,A2,保卫炮台,A3,电子琴……。先把文件键入或读入内存,再用SAVE A1,SAVE A2,SAVE A3……装入A号盘,一直装到接近60扇区时。根据装入的文件多少,最后留出2—4个扇区,把已装入盘内的所有文件用程序2,以文件目录形式装入最后几个扇区。文件目录的内容,第一项是文件名的代号,第二项是文件名,第三项是文件的简单说明或简单的标注。如A号盘的第二个文件,写成A2,保卫炮台(JIKL)。括号内是使用时的操作键符号。文件目录本身定名代号为本盘的盘号,如A号盘就是A,查阅A号盘文件目录时,键入LOAD A即可。

每个虚拟磁盘按以上方法装入文件,把所有的文件装入完毕后,再把磁带倒回000圈处,把每个虚拟磁盘的所有文件目录,用程序1,以文件目录形式装入000圈以后的磁带中,形成一个虚拟磁盘的文件总目录。其中内容是记录每个虚拟磁盘的盘号,占用磁带的圈数(磁带机上计数器的数值)和本盘的文件名代号及文件名。

在使用时,用LOAD提出虚拟磁盘的文件总目录,查找需要的文件,如需要保卫炮台这个文件,找到它在A号盘,记下A号盘所在磁带中的圈数。从起始圈数开始,把A号盘信息读入内存,用LOAD A2提出保卫炮台的程序。如忘记其操作方法或忘记了本盘

的其他文件名的代号,可键入LOAD A提出本号盘的文件目录查阅。使用起来很是方便。

另外在程序1中,还编入字符传递速度。如虚拟磁盘文件总目录显示速度太快,看不清楚时,可放慢字符显示速度。

无驱动器的学习机用户,如感兴趣,不妨采用本文文件管理方法,试一试,定会带来不少方便。

程序1

```

10 HGR2
20 PRINT TAB(7)“虚拟磁盘文件总目录”
30 PRINT TAB(10)“(002—060圈)”
40 PRINT “A号盘文件目录 060—XXX 圈”
50 PRINT “文件名: A1,XXX A2,XXX A3,XXX
    ……A1A号盘文件目录”
60 PRINT:PRINT “B号盘文件目录 XXX—XXX
    圈”
70 PRINT “文件名: B1,XXX B2XXX B3,XXX……
    B1B号盘文件目录”

```

(注:其他盘以此类推)

```

500 PRINT:PRINT “您选择字符传递速度吗?(Y/N),
    GET S$
510 IF S$ = “N” THEN 530
520 PRINT:GOTO 550
530 INPUT “请键入(0—255)之间的数值!”;T
540 SPEED=T
550 PRINT:PRINT “1. 请在计数器上找到虚拟磁盘起
    始圈数!”
560 PRINT:PRINT “2. 请复位后,键入3FFD·8054RN
    3FFDG把程序装入虚拟磁盘!”
570 PRINT:PRINT “3. 不需要本程序时,请用NEW清
    除”
580 END

```

注:3FFD·8054R N 3FFDG 命令是CEC—I中华学习机,把程序读入内存并装入虚拟磁盘用。其他机型需修改。

程序2

```

10 HGR2
20 PRINT TAB(9)“X号盘文件目录”
30 PRINT:PRINT “文件名: X1. XXX X2,XXX X3.
    XXX……”
40 PRINT “请您选择需要的文件!”
50 END

```

# Apple—Ⅱ机音乐功能的扩充

盐城师专 陈建明

《Apple—Ⅱ机 BASIC 语言程序设计》中介绍了九个音阶和五个音长代码,只能编写极其简单的音乐程序。为了扩充其功能,笔者重编了机器发音子程序,调试出七个音调的147种音阶代码,并通过精心设计 BASIC 程序,使乐谱输入简单,演奏时只要改变初值,就能改变音调和速度。

## 一、基本原理

### 1. 音长的控制和变速演奏

在 Apple—Ⅱ 机中发音是通过 \$C030 单元的操作来实现发音的。对这个单元操作的时间越长,发出的声音也越长。原发音子程序,音长由一个八位的二进制计数器来控制,且各音长值固定不变,因而,发出的最长音只有一拍半,演奏速度不能改变。如果将发音子程序中控制音长的计数器改为十六位,并且使音长代码与发音时间呈线性关系,则演奏时,只要给出一拍音符的代码值,其它音长的代码就可直接算出,最长音可达3分钟。这样,既简化了乐谱的输入,又可以方便地实现变速演奏。

### (2) 音阶控制与变调演奏

原发音子程序通过一个八位二进制计数器来控制对 \$C030 单元的触发频率,以达到音阶控制。八位二进制代码值已能基本满足一般歌曲的演奏,只是原来给定的音阶代码值太少。笔者通过调试,得出了七个调门从1到7三个八度音程的147个音阶代码(也可以根据发音子程序的执行周期和各音阶所规定的频率,严格算出各音阶的控制代码)。演奏时,首先将音阶代码输入到一个二维数组,其中第一维下标为音调代码,第二维下标为乐谱的音高代码,通过改变第一维的下标值,就可以改变演奏的调门。

## 二程序及其说明

### 1程序1:

0303—	AD	30	C0	88	D0		
0308—	0D	CE	01	03	D0	08	CE 02
0310—	03	F0	0C	CE	01	03	CA D0
0318—	ED	AE	00	03	4C	03	03 60
0320—	FF	E2	CB	BF	AA	98	88 80
0328—	71	65	5E	54	4B	43	3F 38
0330—	32	2F	29	25	21	E2	CB B6
0338—	AA	98	88	79	71	65	5A 55
0340—	4B	43	3C	38	32	2D	2A 25
0348—	21	1D	CB	B6	A1	98	88 79
0350—	6C	65	5A	50	4B	43	3C 36
0358—	32	2D	27	25	21	1D	1A B6
0360—	A1	91	88	79	6C	60	5A 50
0368—	48	43	3C	36	30	2D	27 23
0370—	21	1D	1A	17	AA	98	88 80
0378—	71	65	5A	54	4B	43	3F 38

0380—	32	2C	29	25	21	1F	1C 19
0388—	16	98	88	79	71	65	5A 50
0390—	4B	43	3C	38	32	2C	27 25
0398—	21	1D	1B	18	15	13	88 78
03A0—	6B	65	5A	50	48	43	3C 36
03A8—	32	2D	27	23	21	1D	1A 19
03B0—	16	13	11				

### 程序2:

```

100 INPUT "F=C#---B$?";A$:INPUT "V=";V
110 L=800:N=27:F=ASC(A$)-66:IF F<1 THEN F=F+7
120 DIM A(7,27),X(3,N)
130 FOR K=1 TO 7:A(K,0)=1
140 FOR I=0 TO 2:FOR J=1 TO 7
150 A(K,I*10+J)=PEEK(L):L=L+1
160 NEXT J,I,K
210 FOR I=1 TO N
220 READ X(1,I),X:X(2,I)=V*X:X(3,I)=1
230 IF X<0 THEN X=-X:X(2,I)=(V*(1+X-INT(X)))/(2*INT(X)):GOTO 250
240 IF X(2,I)>255 THEN X(3,I)=1+INT(X(2,I)/255):X(2,I)=X(2,I)-255*(X(3,I)-1)
250 NEXT I
300 FOR I=1 TO N
310 POKE 768,A(F,X(1,I)):POKE 769,X(2,I):POKE 770,X(3,I)
320 CALL 771:NEXT I
330 END
500 DATA 5,1,11,1,13,1.5,12,-1,12,-2,13,-2,12,-2,11,4,12,1,11,2,6,-1,5,-1,5,4,5,1,11,1,13,1,15,1,15,2,14,1,0,-1,14,1,6,1,14,1.5,14,-1,13,-1,12,-1,12,3
    
```

### 2程序说明

(1) 程序1中 \$303—\$3B0 为发音子程序,\$320—\$3B2 为7个音调的各音阶代码值。该程序不能浮动,并规定 \$300 单元存放音阶代码,\$301 和 302 单元分别存放音长代码的低位和高位,其中 \$302 的最小值为1。

(2) 程序2必须和程序1配合使用。其中100句为输入演奏的音调值和速度值,当速度值在120—180时为一般速度。110句中的 N 值为乐谱的音符数。140句—160句为读入各音阶代码。210句—250句为读入乐谱代码,并把其中的音高值转化为音长代码。

(3) 乐谱的输入规则。乐谱的每个音符包含音高和音长两个要素。本程序对乐谱的输入,要求按照表一、表二的规则进行转换。事实上,这些规则非常好记,完全可以看着乐谱直接输入。

程序2中500句的数据即为下面一段乐谱的代码。(略) (下转第22页)



# POSITION 命令新用

湖北随州市洪山医院设备科(441318) 周 进

在 Apple DOS 操作系统中, POSITION 命令是用来对顺序文件执行定位的操作,它可以把文件位置指针从当前位置移到下面若干个域的起始位置,在随机文件中一般没有用到 POSITION 命令,根据随机文件与顺序文件的结构特点,笔者通过实践将随机文件中的一条记录看成一个顺序文件,当读写这个随机文件的一条记录时,利用 POSITION 命令读写这条记录中的任一字段,结果获得成功。POSITION 命令的这种用法,能为我们组织随机文件的工作带来很大方便。

为使说明更清楚,请阅读下面两段程序,首先用程序 PROGRAM 1 建立一个随机文件,这里有五条记录,每条记录设立了八个字段,以 FILENAME 为文件名存盘。PROGRAM 2 是巧用 POSITION 命令,来检索出随机文件 FILENAME 中每条记录的第四个字段内容的程序,均已在 CEC-I 机上通过。

```
5 REM PROGRAM 1
10 D$=CHR$(13)+CHR$(4)
```

```
15 PRINT D$;"OPEN FILENAME,L30"
20 FOR I=1 TO 5
25 FOR J=1 TO 8: READ A(J): NEXT J
30 PRINT D$;"WRITE FILENAME,R";I
35 FOR J=1 TO 8
40 PRINT A(J);A$;
45 NEXT J
50 NEXT I
55 PRINT D$;"CLOSE FILENAME"
60 DATA 11,12,13,14,15,16,17,18
70 DATA 21,22,23,24,25,26,27,28
75 DATA 31,32,33,34,35,36,37,38
80 DATA 41,42,43,44,45,46,47,48
85 DATA 51,52,53,54,55,56,57,58

5 REM PROGRAM 2
10 D$=CHR$(13)+CHR$(4)
15 PRINT D$;"OPEN FILENAME,L30"
20 FOR I=1 TO 5
30 PRINT D$;"READ FILENAME,R";I
40 PRINT D$;"POSITION FILENAME,R3"
45 PRINT D$;"READ FILENAME"
46 INPUT A
47 PRINT A
50 NEXT I
55 PRINT D$;"CLOSE FILENAME"
```

## CEC-I 与 1724 打印机配接图形硬拷贝程序

马鞍山六中电教室 张益贵

中华学习机及其兼容机通常都配接 9 针打印机,因绝大多数图形软件只支持 9 针打印机,所以用 24 针打印机作外设时,需开发相应的驱动程序才能正常使用。如下所附的程序,可实现 CEC-I 与 1724 打印机配接时高分图形屏幕硬拷贝。为了便于无间隙的多幅连打,设计成将图形旋转 90° 打印。

将附后的机器语言程序键入,取名存盘备用。使用时,调入图形硬拷贝程序到内存 \$8000 首地址单元,将需打印的图形调入高分第一页 \$2000 首地址单元,在监控状态下:8000G 即可执行打印;在 BASIC 状态下:CALL 32768 即可执行打印。若需连打,再调入另一幅图形,然后再执行打印。亦可自编一小程序,做到多幅连打,一气呵成。

修改程序的某些单元,可变通使用,举例如下:

### 1. 图形的正相与反相

在监控状态下,修改 \$804D 单元,可变更所打印图形的正反相:

正相图形 \$804D:EA EA

反相图形 \$804D:49 7F

### 2. 高分第几页的选择

需打印的图形可调入高分第一页,也可调入高分

第二页首地址 \$4000,或第三页(首地址 \$6000)。选择拷贝第几页。只需在监控状态下修改 \$8022 单元即可。

拷贝第一页 8022:20

拷贝第二页 8022:40

拷贝第三页 8022:60

### 3. 拷贝袖珍图形

将需打印的图形印成一寸照片大小,以便于制作图集保留。在监控状态下修改以下 7 个地址单元即可(前述正反相选择和高分页的选择仍有效)。

8009:05 N 8013:47 N 8052:A9 00 20 6F 80

### 4. 拷贝袖珍变形图

在袖珍图形的模式下,修改 \$8013 单元,可打印拉长一倍的变形图,使横构图的袖珍图形变为夸张的纵构图(正反相、选页仍有效)。

\$8013:3D

未修改的当前程序,为正常、反相的图形,选第一页。读者可反汇编列出清单根据需要修改。亦可自编一 BASIC 图形处理程序,用 POKE 置数进行多项选择等,更为简明直观。

在图形打印前,1724 打印机的配置开关(SW1)应

置⑦为 ON(拨向右)

8000-	20	6D	80	A9	4A	20	6F	80
8008-	A9	0B	20	6F	80	A0	27	20
8010-	6D	80	A9	3D	20	6F	80	A9
8018-	00	20	6F	80	A9	BF	20	6F
8020-	80	A9	20	85	08	A2	00	8A
8028-	0A	0A	29	1C	85	07	8A	6A
8030-	6A	6A	6A	29	03	05	07	05
8038-	08	85	07	8A	6A	29	E0	85
8040-	06	6A	6A	29	18	05	06	85
8048-	06	B1	06	29	7F	49	7F	20
8050-	6F	80	20	6F	80	A9	00	20
8058-	6F	80	E8	E0	BF	D0	C8	A9
8060-	00	20	6F	80	A9	0A	20	6F
8068-	80	88	10	A3	60	A9	1B	2C
8070-	C1	C1	30	FB	8D	90	C0	60

FB2B-D0 F8	BNE \$FB25
FB2D-88	DEY
FB2E-60	RTS

这是一个单循环计数程序。运行时不断地查询内存 \$C064+X(X=0~3) 的内容。如果内存的高位为 1, 就增加一个记数; 如果内存高位为 0, 就退出循环。由于单重循环的最大记数为 255, 这就制约了 GAME 的模拟入口所能连接的最大电阻仅为 125kΩ 左右。对于大于 125kΩ 的电阻, 此单重循环就无法计数了。这给需用大于 125kΩ 电阻进行的实验操作带来不便。

解决的办法是改写这一段监控程序, 用双重循环程序代替单循环程序。改写的程序如下

```
0300- D8 A9 00 85 09 85 FA 85
0308- FB 85 FC 20 37 03 18 A5
0310- 07 65 FA 85 FA A5 08 65
0318- FB 85 FB A9 00 65 FC 85
0320- FC E6 09 A5 09 C9 20 D0
0328- E2 A2 00 46 FC 66 FB 66
0330- FA E8 E0 05 D0 F5 60 A6
0338- 06 AD 70 C0 A0 00 84 08
0340- BD 64 C0 10 09 C8 D0 F8
0348- E6 08 D0 F4 C6 08 84 07
0350- 60
```

因为双重循环的计数是单循环的 256 倍, 所以改写监控程序后 GAME 模拟入口能连接的电阻最大可达 30MΩ 左右。

在进行接口扩展操作时, 先将选择的模拟通道号存入 \$06 单元, 然后再调用上述双重循环程序, 就可在 \$FB、\$FA 两单元中得到记数。

## 游戏接口 PDL 的扩展

武警长沙指挥学校训练处(410125) 张建群

苹果机的游戏接口的模拟输入口(PDL)监控程序如下:

```
FB1E-AD 70 C0 LDA $C070
FB21-A0 00 LDY # $00
FB23-EA NQP
FB24-EA NOP
FB25-BD 64 C0 LDA $C064,X
FB28-10 04 BPL $FB2E
FB2A-C8 INY
```

## SUPER DOS 简介

北京职工技术师范学院 张志

现在苹果机、中华机流行一种磁盘操作系统——SUPER DOS, 它具有许多优于 Apple DOS 的功能。并且该系统主盘上还提供了一个名为《BASIC》的文件, 用 & 指令扩展了几条 Applesoft 语句, 现介绍该系统的 1.6 版的功能如下。

### 一、磁盘操作系统

①本系统要求主机内存至少为 64K, 系统启动后, 自动将 DOS 搬入 16K RAM 卡, 把原来 Apple DOS 占据的 \$9600~\$BFFF 这段地址空间腾出来给用户程序使用。

②本系统具有用“\*”作为通配符功能, 各种含对文件名操作的命令都可利用此功能。详细用法见 Apple DOS 主盘 FID 程序的使用说明, 本系统只是将通配符改为“\*”了。

③本系统取消了原 Apple DOS 的 FP、INT 命令, 保留的 CHAIN 命令只可用于 A 类文件间链接。

④系统的 OPEN、APPEND、WRITE、READ、EXEC、CLOSE、DELETE、VERIFY、LOCK、UNLOCK、BRUN、BSAVE、SAVE、BLOAD、INIT、与 Apple DOS 用法相同。

⑤HELP 命令: 在文本状态下提示全部 SUPER DOS 命令。

⑥RUN、LOAD 命令: 扩展为对 A、B、T 类文件均可操作。

⑦TYPE 命令: 以 ASCII 码形式显示文件内容。格式: TYPE 文件名

⑧NTR 命令: 指定当前磁盘为普通格式磁盘。格式: NTR

⑨HTR 命令:指定当前磁盘为半轨格式磁盘。

格式:HTR

在此命令下,格式化一张磁盘,普通拷贝程序将无法拷贝这种格式的磁盘。只有启动 SUPER DOS 后,敲入 HTR 命令才能读写这种半轨格式磁盘,因而起到了加密的作用。

⑩CATALOG(或 DIR):为列磁盘目录命令,同时显示自由扇区数。

⑪ADR 和 LEN 指令:用来显示刚装入的二进制文件的首地址和长度

格式:ADR

LEN

⑫PRG 命令:显示磁盘的自由扇区数。

格式:PRG

⑬BASIC 命令:设定 BASIC 程序区始存地址

格式:BASIC An(n 为地址)

⑭MEG 命令:把当前内存中的 BASIC 程序与盘上的 BASIC 程序合并,行号均不变。

格式:MEG 文件名

⑮UNDEL 命令:恢复被误删的文件。

格式:UNDEL 文件名

⑯FLB 命令:设置缓冲区首地址,最高可设置为 \$ 95CA

格式:FLB 地址

⑰该系统还有自动执行文件功能。当键入一串既非 SUPER DOS 的命令保留字,也不是 Applesoft 命令的保留字的字符串时,系统会把该字符串当成文件名并运行它。

## 二、扩展 BASIC 命令

先运行系统盘上的《BASIC》文件,通过 & 命令提供13条扩展 BASIC 命令:

①&EDIT:编辑指定命令行。

格式:&EDIT 行号

②&UNNEW:追回刚被 NEW 命令清掉的 BA-

SIC 程序

③&INPUT:扩展的 INPUT 命令,可在输入的变量中加入表达式,其格式与 INPUT 命令相同。

例:10 &INPUT A

RUN

?SIN(5) \* 6 - 4 + TAN(3)

④&GOTO 和 &GOSUB:扩展 GOTO 和 GOSUB 命令,可用表达式表示行号,

格式:&GOTO 表达式

&GOSUB 表达式

⑤&RESTORE:将指定语句作为 DATA 语句的最初读取行

格式:&RESTORE 行号

⑥&REN:重整行号命令。用法与 TOOL KIT 中的 APA 程序所提供的 &RENUMBER 命令用法相同。

⑦&AUTO 指令:自动行号命令,用法与 APA 程序中的 &AUTO 命令相同。

⑧&ALIST 命令:

格式:&ALIST 行号(相当于 POKE 33,33:LIST 行号)

&ALIST (相当于 POKE 33,33:LIST)

⑨&BLIST:比 &ALIST 更紧凑的一种列程序方法。格式与 &ALIST 相同。

⑩&FIND:寻找并列出行号含有指定字符内容的 BASIC 程序行。

格式:&FIND “字符串”

&FIND 字符串变量。

⑪&SHOW:显示控制字符

格式:&SHOW

⑫&R:取消 &SHOW 命令

格式:&R

⑬&POINT:指定要读取的 DATA 数据

格式:&POINT 表达式

(上接第19页)

表一

音符	1	2	...	6	7	1	2	...	6	7	1	2	...	6	7
音高值	1	2	...	6	7	11	12	...	16	17	21	22	...	26	27

表二

音符	X	=	X.	X	X.	X	X.	X-	X-
音长值	-2	-2.5	-1	-1.5	1	1.5	2	2.5	...

(上接第47页)

从软盘启动后,无法进入硬盘,即使使用低级格式化命令也无法执行;开机从硬盘启动进入 ROM-BASIC 状态等等硬盘故障,原因可能是多方面的,但系统设备配置信息错误是一条不可忽视的原因。CPU 为 286、386 等微机的设备配置情况一般由高级诊断程序的“SETUP”功能设置(长城 286BH 按 Ctrl+ALT+Esc 键设置)。存储在系统主板的 CMOS RAM 中,而 RAM 由系统主板上的后备电池供电,即关闭主机电源,该信息一般不会丢失。但当机器受到搬运等强烈震动,后备电池松动而接触不良;或者机器较长一段时间不使用,后备电池不能及时充电,能量不足,该 RAM 中的信息就会改变或丢失。这时开机系统配置情况与设备配置信息不符,BIOS 检测失效,就可能产生上述故障现象。



## 第十讲 监控子程序的调用(二)

南京大学大气科学系(210008)朱国江

## ⑨显示一个方框

如在屏幕上显示如下方框,要求充满整个屏幕。

```

* = * = * = * = *
=
*
=
*
=
*
=
* = * = * = * = *

```

这个问题看似简单,但实际编程还较复杂,先给出设计好的程序如下:

程序:10.24

```

0300- 20 58 FC 58 JSR $FC 58
0303- A0 00 LDY # $00
0305- A9 AA LDA # $AA
0307- 20 ED FD JSR $FDED
030A- A9 BD LDA # $BD
030C- 20 ED FD JSR $FDED
030F- C8 INY
0310- C0 13 CPY # $13
0312- D0 F1 BNE $0305
0314- A9 AA LDA # $AA
0316- 20 ED FD JSR $FDED
0319- A0 00 LDY # $00
031B- 20 5C 03 JSR $035C
031E- C8 INY
031F- C0 0A CPY # $0A
0321- D0 F8 BNE $031B
0323- 20 62 FC JSR $FC62
0326- A9 BD LDA # $BD
0328- 20 ED FD JSR $FDED
032B- A2 25 LDX # $25
032D- 20 4A F9 JSR $F94A
0330- A9 BD LDA # $BD
0332- 20 ED FD JSR $FDED
0335- 20 62 FC JSR $FC62
0338- A0 00 LDY # $00
033A- A9 AA LDA # $AA
033C- 20 ED FD JSR $FDED
033F- A9 BD LDA # $BD
0341- 20 ED FD JSR $FDED
0344- C8 INY
0345- C0 13 CPY # $13
0347- D0 F1 BNE $033A
0349- A9 AA LDA # $AA
034B- 20 ED FD JSR $FDED

```

水平显示  
19对\*=,  
再补上一个\*

垂直显示10组  
\*,  
中间空37格

下称一行,显  
示一个\*,  
中间空37格,  
再示一个\*,  
再下移一行

同 \$300 ~  
\$318

```

034E- A2 50 LDX # $50
0350- A9 FF LDA # $FF
0352- 20 A8 FC JSR $FCA8
0355- CA DEX
0356- D0 F8 BNE $0350
0358- 20 58 FC JSR $FC58
035B- 60 RTS
035C- 20 62 FC JSR $FC62
035F- A9 BD LDA # $BD
0361- 20 ED FD JSR $FDED
0364- A2 25 LDX # $25
0366- 20 4A F9 JSR $F94A
0369- A9 BD LDA # $BD
036B- 20 ED FD JSR $FDED
036E- 20 62 FC JSR $FC62
0371- A9 AA LDA # $AA
0373- 20 ED FD JSR $FDED
0376- A2 25 LDX # $25
0378- 20 4A F9 JSR $F94A
037B- A9 AA LDA # $AA
037D- 20 ED FD JSR $FDED
0380- 60 RTS

```

调用80次延时  
子程序延时  
清屏,结束

垂直显示一组  
\*,  
中间空37格

程序10.24共用了5个监控中机器语言子程序:

- \$FC58:清屏
- \$FDED:输出一个字符
- \$FC62:回车换行
- \$FCA8:延时
- \$F94A:屏幕上输出若干个空格,空格数在x寄存器中设定

程序10.24主要用了调用机器语言子程序的技巧,这包括两方面的内容,一是调用监控子程序,实现诸如清洗屏幕、输出字符、控制延时、光标下移、输出空格等功能;二是调用自编的机器语言子程序,如\$035C—\$0380单元中存放的子程序,它实现一组字符的显示,而用\$0319—\$0322程序段对其实现次数控制。

## (10)显示一个题头

在边框里放上一些说明性文字(或资料),就设计成了题头。边框设计可以用程序10.24,或者根据其设计思想,绘成其它图案。问题的关键是如何设计一个存有文字字符的数据表,并把它放在边框内的适当位置。

为了直接引用程序10.24设计的边框,我们可以设想再编一个独立的存放资料表的程序,为简单计,假设资料表就是一行,即PRESS RETURN TO CONTINUE,见程序10.25:

```
1000- A0 00 LDY # $00
```

```

1002- B9 0E 10   LDA $100E,Y
1005- 20 ED FD   JSR $FDED
1008- C8         INY
1009- C0 26      CPY # $26
100B- D0 F5      BNE $1002
100D- 60         RTS
100E- A0 A0
1010- A0 A0 A0 A0 50 52 45 53
1018- 53 60 52 45 54 55 52 4E
1020- 60 54 4F 60 43 4F 4E 54
1028- 49 4E 55 45 60 61 A0 A0
1030- A0 A0 A0 A0

```

程序10.25的设计思想和结构安排均和程序10.16相类似,只是用的指令不尽相同,这里不再重述。从\$100E开始到\$1033止,存放的是PRESS RETURN TO CONTINUE!对应的闪烁显示方式的ASCII字符。运行时在监控下用1000G↙,在BASIC状态下用CALL 4096↙。

有了边框和数据表,就可以设计一个题头,这实际上就是将程序10.24和程序10.25合起来,并修改适当的控制段和修改个别字节的内容,见程序10.26。

```

0300- 20 58 FC   JSR $FC58
0303- A0 00      LDY # $00
0305- A9 AA      LDA # $AA
0307- 20 DE FD   JSR $FDED
030A- A9 BD      LDA # $BD
030C- 20 ED FD   JSR $FDED
030F- C8         INY
0310- C0 13      CPY # $13
0312- D0 F1      BNE $0305
0314- A9 AA      LDA # $AA
0316- 20 ED FD   JSR $FDED
0319- A0 00      LDY # $00
031B- 20 5C 03   JSR $035C
031E- C8         INY
031F- C0 0A      CPY # $0A
0321- D0 F8      BNE $031B
0323- 20 62 FC   JSR $FC62
0326- A9 BD      LDA # $BD
0328- 20 ED FD   JSR $FDED
032B- A2 25      LDX # $25
032D- 20 4A F9   JSR $F94A
0330- A9 BD      LDA # $BD
0332- 20 ED FD   JSR $FDED
0335- 20 62 FC   JSR $FC62
0338- A0 00      LDY # $00
033A- A9 AA      LDA # $AA
033C- 20 ED FD   JSR $FDED
033F- A9 BD      LDA # $BD
0341- 20 ED FD   JSR $FDED
0344- C8         INY
0345- C0 13      CPY # $13
0347- D0 F1      BNE $033A
0349- A9 AA      LDA # $AA
034B- 20 ED FD   JSR $FDED

```

```

034E- A2 20      LDX # $20
0350- A9 FF      LDA # $FF
0352- 20 A8 FC   JSR $FCA8
0355- CA         DEX
0356- D0 F8      BNE $0350
0358- 20 34 10   JSR $1034
035B- 60         RTS
035C- 20 62 FC   JSR $FC62
035F- A9 BD      LDA # $BD
0361- 20 ED FD   JSR $FDED
0364- A2 25      LDX # $25
0366- 20 4A F9   JSR $F94A
0369- A9 BD      LDA # $BD
036B- 20 ED FD   JSR $FDED
036E- 20 62 FC   JSR $FC62
0371- A9 AA      LDA # $AA
0373- 20 ED FD   JSR $FDED
0376- A2 25      LDX # $25
1000- A0 00      LDY # $00
1002- B9 0E 10   LDA $100E,Y
1005- 20 ED FD   JSR $FDED
1008- C8         INY
1009- C0 26      CPY # $26
100B- D0 F5      BNE $1002
100D- 60         RTS
100E- AA A0
1010- A0 A0 A0 A0 50 52 45 53
1018- 53 60 52 45 54 55 52 4E
1020- 60 54 4F 60 43 4F 4E 54
1028- 49 4E 55 45 60 61 A0 A0
1030- A0 A0 A0 A0
1034- 20 F4 FB   JSR $FBF4
1037- 20 ED FD   JSR $FDED
103A- 20 F4 FB   JSR $FBF4
103D- 20 1A FC   JSR $FC1A
1040- 20 1A FC   JSR $FC1A
1043- 20 1A FC   JSR $FC1A
1046- 20 00 10   JSR $1000
1049- 20 66 FC   JSR $FC66
104C- 60         RTS

```

\$0300—\$0380:为主要程序段,显示一个边框

\$1000—\$1033:显示文字资料,其中\$100E~\$1033为对应文字资料的ASCII码字符

\$1034—\$104C:为使文字资料放置在边框中的适当位置而设置的控制程序,其中\$FBF4是光标右移一格的子程序,\$FC1A使光标上移一行。

#### (11)汉字符符的显示

在中华学习机(Apple—II及其兼容机不具备)内,有一个重要的子程序CSWA,它的功能是完成汉字或ASCII字符的输出,其入口地址为\$C32B。

对CSWA子程序的调用,必须首先要求在累加器A中存放汉字或字符的显示码以及显示控制命令码。而对于汉字则一定要用三字节的中华学习机内码,同时需要分三次调用才能输出一个汉字。下面提供一个

实例,见程序10.27

```
1000-A9 0D      LDA # $D3
1002-20 2B C3   JSR $C32B
1005-A2 00      LDX # $00
1007-BD 18 10   LDA $1018,X
100A-48         PHA
100B-20 2B C3   JSR $C32B
100E-E8         INX
100F-68         PLA
1010-10 F5      BPL $1007
1012-A9 0D      LDA # $0D
1014-20 2B C3   JSR $C32B
1017-60         RTS
```

```
1018-7F 55 4F 7F 39 27 7F 50
1020-24 7F 4E 2E 7F 39 79 20
1028-43 4F 4D 50 55 54 45 D2
```

程序10.27的运行方法比较特殊,因为程序中有显示汉字的安排,所以必须在中文状态下,并且处于BASIC状态才能运行,即]CALL 4096

屏幕显示:中华学习机 COMPUTER

(12)文本状态窗口控制显示

中华学习机在文本状态下可以显示各种ASCII字符,正常的显示屏幕被设定为24行40列。为了能在屏幕的任意位置上显示字符,可以采用“开窗口”的方法,所谓屏幕“开窗口”,就是在屏幕上任意设定一块显示区,而在这个区域以外的屏幕不发生变化。窗口的设置由用户向四个特殊零页单元置数确定:

\$20 单元存放窗口左极限列数(\$0~\$27)

\$21 单元存放窗口的宽度(\$1~\$28)

\$22 单元存放窗口顶行行数(\$0~\$18)

\$23 单元存放窗口底行行数(\$0~\$18)

正常情况下\$20~\$23四个单元中的值分别为\$0,\$28,\$0,\$18,在BASIC状态下可以用POKE命令设定参数,例如在\$20单元中放\$0A:

POKE 32,10

而用机器语言设定时,则可用:

A9 0A LDA # \$0A

85 20 STA \$20

例如要在屏幕中央反相显示“6502CPU”可用程

序10.28:

```
0300-20 58 FC   JSR $FC58
0303-A2 00      LDX # $00
0305-BD 11 03   LDA $0311,X
0308-20 ED FD   JSR $FDED
030B-EB         INX
030C-E0 07      CPX # $07
030E-D0 F5      BNE $0305
0310-60         RTS
0311-76 75      ROR $75,X
0313-70 72      BVS $0387
0315-43         ???
0316-50 55      BVC $036D
```

```
0318-20 58 FC   JSR $FC58
031B-A9 13
031D-85 20      STA $20
031F-A9 09      LDA # $09
0321-85 22      STA $22
0323-20 00 03   JSR $0300
0326-60         RTS
```

程序10.28中\$0300~\$0317单元和程序10.16完全一样,反相显示“6502 CPU”七个字符。\$031B~\$031E是设定窗口左极限,其值为\$13(取屏幕左、右边界的中间值), \$031F~\$0322是设定窗顶行数,其值为\$09(取屏幕上、下边界的中间值)。

(13)数据块搬家

将\$6000~\$60FF的内容搬至\$7000~\$70FF中去。

程序10.29

```
300-A0 00      LDY # $00
302-A9 00      LDA # $00
304-85 3C      STA $3C
306-A9 60      LDA # $60
308-85 3D      STA $3D
30A-A9 FF      LDA # $FF
30C-85 3E      STA $3E
30E-A9 60      LDA # $60
310-85 3F      STA $3F
312-A9 00      LDA # $00
314-85 42      STA $42
316-A9 70      LDA # $70
318-85 43      STA $43
31A-20 FC FE   JSR $FE2C
31D-60         RTS
```

这里利用了监控中搬家子程序,只需将源数据首地址和末地址放入\$3C~\$3F中,而把目的首地址放在\$42~\$43中,调用\$FE2C子程序一次搬完。在调用前必须设置LDY #00。值得提出的是,在监控状态下移动一段存储器单元的内容就是用的搬家程序\$FE2C,即所谓执行监控状态下的M命令。即:

\* <新地址><原首地址>·<原末地址>M

关于调用监控子程序的实例,我们就介绍到这里。必须指出的是监控子程序远远不只是上面的内容,还有许多子程序,我们没有列出,有兴趣的读者可查阅有关资料,另外,本讲座的其它章节也有一些调用监控子程序的程序,请注意对照学习。

(上接第27页)位,模拟信号为伪差分输入方式,电位器VR2用于调节INLO电位的。3.控制电路:由74LS00,插座W1、W2、W3电阻R1、R2及电容C1等组成。

其中插座W1用于选择转换结束方式,可选择中断方式或者查询方式;插座W2用于转换器启动方式选择,可用MOVX指令启动,也可用口输出指令启动。除上述的控制转换之外,还可以选择连续转换方式。插座W3用于输出方式选择。

4.负电源产生电路:由ICL7660及电容C8、C9、C10组成,产生一个-5伏的电源。



## BJS-51单片机实验系统(续) 模/数、数/模转换实验

北京工业大学 张俊谟

众所周知,在过程控制和智能仪器仪表中通常由微型机进行实时控制和实时数据处理。计算机加工的信息总是数字量,而被控制或被测对象的有关参量却往往又是一些连续变化的模拟量。因此,把被测参量送入微机之前必须进行模拟量到数字量的转换;反过来,微机处理后的数字量也必须经过数字量到模拟量的转换才能被执行机构等接收。

由此可见,A/D、D/A 转换是微机与客观世界联系的桥梁,是微机接口的重要组成部分。A/D、D/A 转换与单片机接口的复杂性,包含以下五个方面:

- 数据输入、输出与总片相应引脚的连接(包括数字接口和模拟接口)
- 转换启动方式的选择
- 转换结束信号(或状态信号)的处理
- 参考电压的获得及连接
- 时钟的产生和连接

A/D、D/A 转换软件的编制也有其独特性,其中包括中断、查询、延时等待等访问方式。

上述这些方面既是 A/D、D/A 接口电路设计必须解决的问题,又是学生学习 A/D、D/A 转换技术中的难点所在。因此在设计 A/D、D/A 转换实验时,必须让学生对这些问题有足够的了解和较为充分的训练。

为此,BJS-51单片机教学实验系统中,我们设计了有关 A/D、D/A 转换实验共5个。限于篇幅,只能着重介绍12位 A/D 转换实验和8位 D/A 转换实验。本期主要介绍双积分式12位 A/D 转换实验。

为了让设计的实验电路有充分的选择余地,我们选用廉价而高精度的双积分式 A/D 转换芯片——ICL7109。下面分别介绍 A/D 转换芯片、实验电路设计以及实验软件的编制等。

### 一、双积分式12位 A/D 转换芯片——ICL7109

#### 1. 总片特点:

- 高精度(精确到1/4096)
- 低噪声(典型值为 $15\mu\text{V}_{\text{pp}}$ )
- 低漂移(小于 $1\mu\text{V}/^\circ\text{C}$ )
- 高输入阻抗(典型值为 $10^{12}\Omega$ ,典型输入电流 $1\text{pA}$ )

#### • 与 TTL 兼容,三态控制输出

具有通用控制信号,可用来监视和控制转换时间

- 片内有振荡器,只须外接晶体或 RC 器件

#### 2. 芯片的结构特点:

ICL7109的整体电路由模拟电路和数字电路两部分所组成,这里不详细叙述其结构。

#### 3. 芯片的外部使用特点:

ICL7109芯片的外部引脚和外部连接状态如图1所示。

(1)电源供给:ICL7109为双电源 $\pm 5\text{V}$

(2)参考电压供给:ICL7109有一个良好的片内参考电压源,由  $\text{REF}_{\text{OUT}}$ (29脚)输出,可以用一个电位器分压获得一个合适的基准电压。

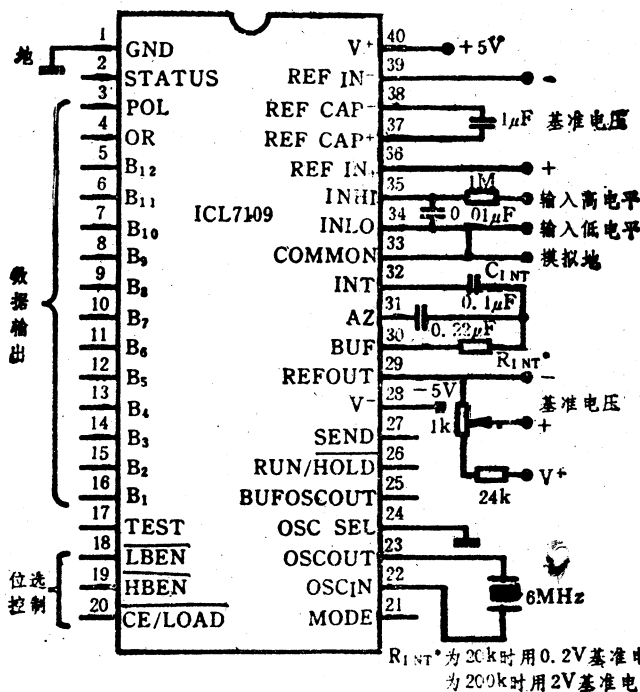


图1 ICL7109新脚及外部连接

参考电压的稳定与否,直接影响转换的精度。

参考电压输入为差分输入,分别由  $\text{REF}_{\text{IN}}^+$ (36脚)及  $\text{REF}_{\text{IN}}^-$ (39脚)引入。参考电压既可以由片内参考电压源供给,也可以由外部提供。

(3)模拟输入信号:模拟信号是差分输入,分别由差分输入高端 INHI(35脚)和差分输入低端 INLO(34脚)引入,模拟信号公共端为 COMMON(33端)。

(4)时钟电路:ICL7109有片内振荡器及时钟电路。片内提供的多功能时钟振荡器既可用于 RC 振荡器,也可用作晶体振荡器。

为了使电路具有抗50HZ 串模干扰能力,积分时间(相当2048个时钟数)应等于50HZ 的整数倍。

(5)工作状态控制:工作状态控制又分成转换状态控制和输出状态控制两部分:



转换状态控制:转换状态由 RUN/HOLD(26脚)输入控制。

若它是高电平时,每经8192个时钟脉冲完成一次转换。每次转换又都经历自动调零、信号积分和消除积分三个阶段。其中自动调零至少需要2048个时钟脉冲周期,信号积分阶段为2048个时钟脉冲周期,消除积分阶段最多为4096个时钟脉冲周期。在消除积分阶段,当输出电压返回零态,并进而过零之后,由模拟电压转换得到的数据将被锁存起来。

若它输入低电平时,转换器将立即结束消除积分阶段,并跳至自动调零阶段,从而缩短了消除积分阶段,提高了转换速度。

输出状态控制:由转换器输出状态标志 STATUS端(2脚)、输出方式选择 MODE 端(21脚)、片选 CE/LOAD端(20脚)以及高、低字节输出选通 HBEN端(19脚)、LBEN端(18脚)控制。

输出状态标志(STATUS 端):如图2所示,在

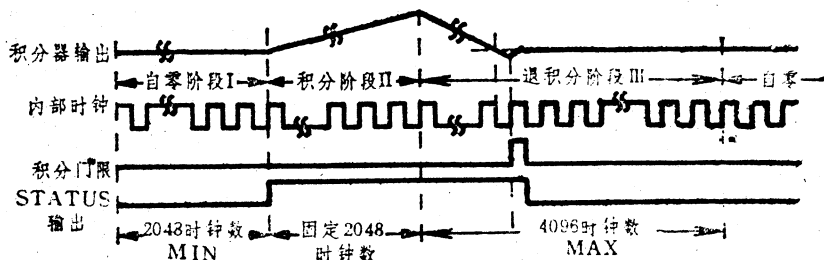


图2 ICL7109的转换时序

每一转换周期中,当自动调零阶段一结束,该状态由低电平跳出高电平;在消除积分阶段,当输出电压返回零态并进而过零,转换输出数据被锁存之后的半个时钟脉冲周期后,它由高电平跳回至低电平。在STATUS端保持低电平状态时,输出的数据不变。

输出方式选择(MODE 端):当MODE 端为低电平时,转换器为直接输出方式,这时数据在片选 CE/LOAD、HBEN及LBEN等信号的控制下直接输出。当MODE 为高电平时,转换器将在信号交换方式下,给每一转换周期的结尾输出数据。

片选CE/LOAD:在MODE 为低电平时,片选 CE/LOAD为数据输出的主要选通信号。

当该信号为低电平时,为正常输出状态;当其为高电平时,所有数据线处于高阻悬浮状态。

字节选通 HBEN 和 LBEN:当MODE 和片选 CE/LOAD均为低电平时,LBEN作为低字节(B1~B8)输出的选通信号;HBEN作为高字节(B9~B12)及POL,OR 输出的选通。其中POL 为输入模拟信号的极性标志,POL=1,表示信号为正;OR 为过量程标志,高电平表示过量程。

现在在直接输出方式下,MODE、CE/LOAD、HBEN及LBEN 的关系汇总表1中。

表1 直接输出方式下的输出控制

MODE	CE/LOAD	HBEN	LBEN	输出状态
0	1	×	×	所有数据线高阻态
0	0	0	1	高字节输出
0	0	1	0	低字节输出

## 二、实验电路的设计

根据对于 ICL7109 芯片的转换过程的理解和对 A/D 转换实验电路的要求,设计12位 A/D 转换实验电路如图3所示。该电路包括转换器及其附属元件、转换控制电路、输入电路、参考电压及-5伏电源发生电路等几个部分。

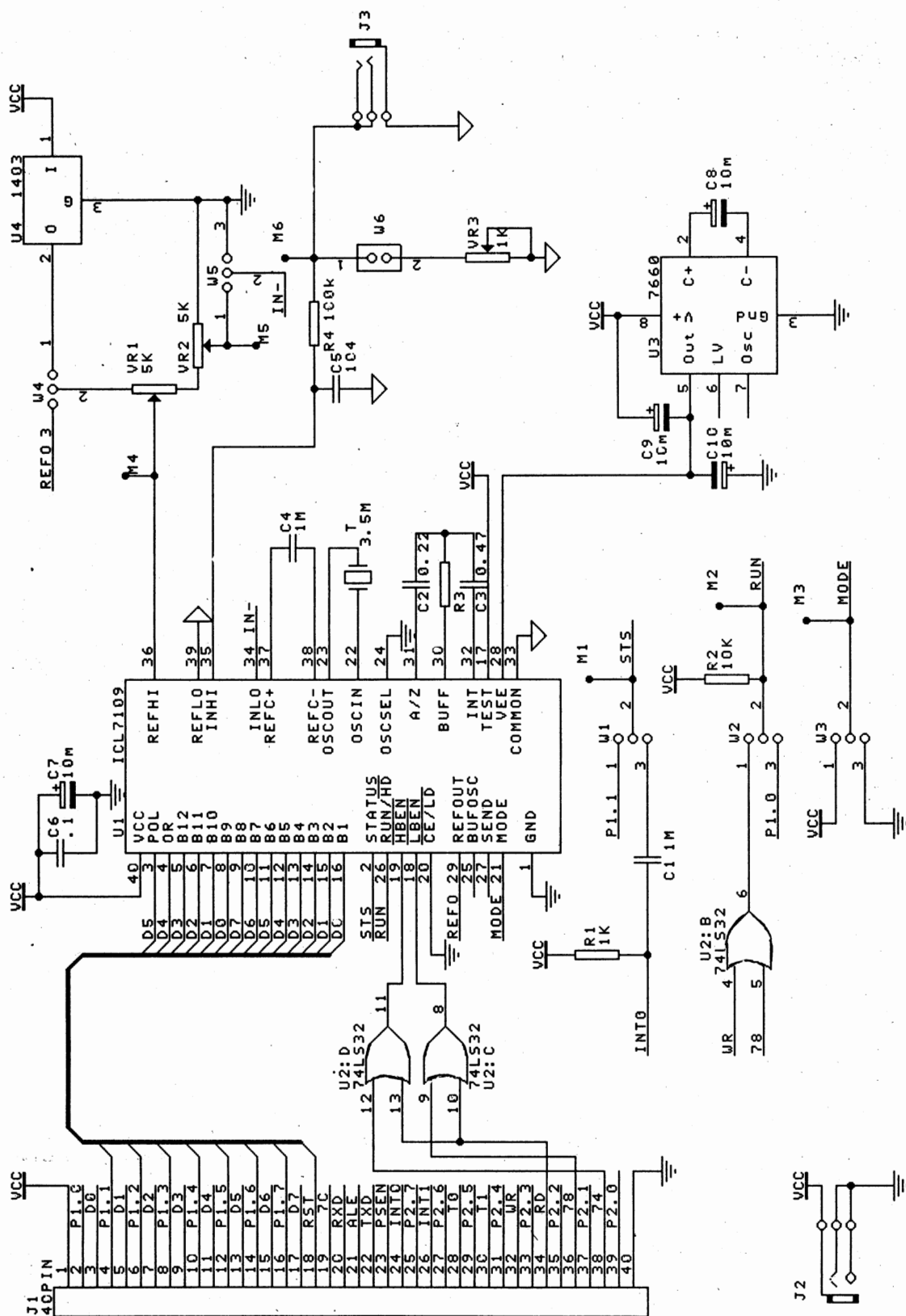
1. 输入电路:由插座 J3、W6,精密电位器 VR3 以及电阻 R4、电容 C5 所组成。R4、C5 组成输入滤波网络,J3 为模拟信号输入插座,W6 用来选择电压信号还是电流信号的。当该插座上插入插块时,电路为电源信号输入。

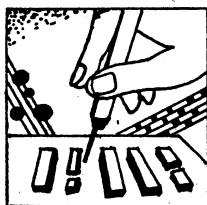
2. 参考电压电路:由稳压块 MC1403 及插座 W4、W5,精密电位器 VR1、VR2 组成。插座 W4 用来选择内部参考源还是外部参考源的,电位器 VR1 用于调节参考电压。插座 W5 用于决定模拟差分输入低端 INLO 电位,当2-3相联时,INLO 接模拟公共端,模拟信号为单端输入方式;当1-2相联时,INLO 端接有一定电

表2 插座功能表

(下转25页)

功能	引脚	插座	插块位置		测试点
			1-2	2-3	
状态标志	STATUS	W1	查询	中断	M1
转换启动方式	RUN/HOLD	W2	MOVX 指令启动	口输出指令启动	M2
输出方式	MODE	W3	信号交换方式	直接输出方式	M3
参考电压选择	REFH $\bar{Z}$	W4	外部参考电压	内部参考电压	M4
模拟信号输入方式		W5	差分输入	单端输入	M5
模拟信号类型选择		W6	电流	电压	M6





# 微电脑控制微型钻床

〈学习 X 轴、Y 轴同时驱动〉

易齐干

## 学装微电脑

双坐标数控机床工作时,由一个坐标点向下一个坐标点移动,多数采用同时驱动 X 轴 Y 轴电机的方法。实现这种方法大致有两种。例如,给 X 轴电机 3 个脉冲(X 轴移动量=分辨能力×脉冲数=0.01×3=0.03mm),给 Y 轴电机 5 个脉冲(Y 轴移动量=分辨能力×脉冲数=0.01×5=0.05mm)。

(1)两轴对脉冲同时计数,完成指定脉冲数的轴,电机先停止传动。如图1所示。

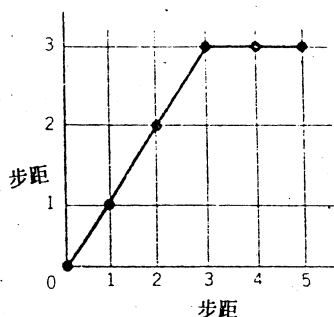


图1 第1种脉冲输出方法

(2)一边对两轴的脉冲进行计数,一边驱动两个电机,对电机进行通/断控制。如图2所示。

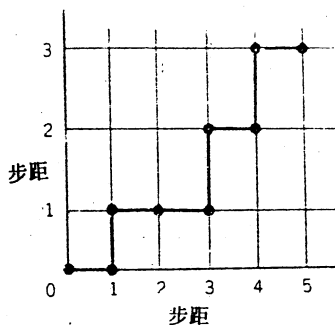


图2 第2种脉冲输出方法

钻床按所要求的座标去钻孔,采用第1种方法简单。试分析如下:

编制在印刷线路板上按图形轨迹钻孔程序。

首先决定输入输出口、RAM 区域。分别如表1、表2所示。

座标距离数字化上期文章已介绍。

以上述为基础,则可编制流程图与程序清单。

(1)主程序(MAIN)I/O 端口、RAM 初始化;判断自动运转还是手动运转;启动自动开关,执行 CALL 指令,跳到自动运转子程序;手动时,将工作台移动开关的状态送给电机驱动 IC,电机回转。

表1 I/O 图

位口	7	6	5	4	3	2	1	0
A (输入)	X 轴编码器脉冲	Y 轴编码器脉冲	记忆开关	自动开关	右开关	左开关	后开关	前开关
C (输出)	未使用		电机	电磁铁	X 轴电机		Y 轴电机	
					正转	反转	正转	反转

表2 RAM 图

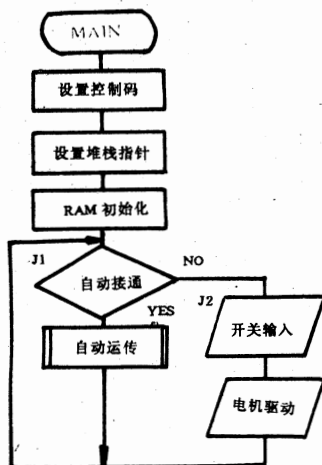
位地址	7	6	5	4	3	2	1	0
8000H	电机驱动数据							
8001H	1:Y 轴 0:Y 轴 正转	1:X 轴 0:X 轴 正转			1:Y 轴 脉冲有 变化	1:X 轴 脉冲有 变化	1:Y 轴 数据 = 0	1:X 轴 数据 = 0
8002H	记录 X 轴脉冲的前次数据							
8003H	记录 Y 轴脉冲的前次数据							

(2)自动运转程序(AUTO)

执行 CALL 指令,跳到数据传送子程序;电机驱动子程序;钻头驱动子程序;最后,查询数据是否结束。

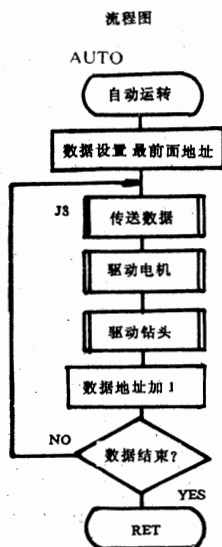
(3)数据传送程序(DTCONV)

将 X 轴移动数据送到 BC 寄存器对,Y 轴移动数据送到 DE 寄存器对,设置回转方向标志。



标号	助记符	地址	机器码	注释
MAIN	LD A,90H	0000	3E 90	A 口输入, B 口、C 口输出
	OUT (03H),A	0002	D3 03	
	LD SP,8100	0004	31 00 81	设置堆栈指针
	LD HL,8000H	0007	21 00 80	
	LD A,FFH	000A	3E FF	电机驱动数据为 FFH
	LF (HL),A	000C	77	
	OUT (02H),A	000D	D3 02	
	XOR A	000F	AF	数据输出
	INC HL	0010	23	
	LD (HL),A	0011	77	标志位清零
	INC HL	0012	23	
	LD (HL),A	0013	77	X 轴脉冲数据清零
	INC HL	0014	23	
	LD (HL),A	0015	77	Y 轴脉冲数据清零
J1	IN A,(00H)	0016	DB 00	开关输入
	BIT 4,A	0018	CB 67	自动开关不通时
	JP NZ,J2	001A	C2 23 00	跳入 J2
	CALL AUTO	001D	CD 2A 00	
	JP NZ,J2	001D	CD 2A 00	
J2	IN A,(00H)	0023	DB 00	向 A 口输出开关
	OUT (02H),A	0025	D3 02	状态
	JP J1	0027	C3 16 00	

图3 主程序流程图与清单



标号	助记符	地址	机器码	注释
AUTO	LD HL,0200H	002A	21 00 02	设置数据最前面地址
J3	CALL DTCONV	002D	CD 3E 00	
	CALL XYAUTO	0030	CD 64 00	
	CALL DAUTO	0033	CD 5A 01	
	INC HL	0036	23	
	LD A,(HL)	0037	7E	
	CP FFH	0038	FE FF	数据没结束时 (FFH)跳向 J3
	JP NZ,J3	003A	C2 2D 00	
	RET	003D	C9	

图4 自动运转程序流程图与清单

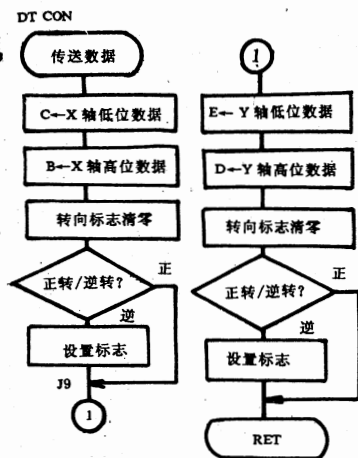
(4) 电机驱动程序(XYAUTO)

如果脉冲有变化,电机驱动数据减1。驱动电机程序与转动方向的标志对应。CPU 处理速度比电机转速

要快,通过对 X 轴和 Y 轴编码器脉冲进行检验,不会遗漏脉冲读数。



流程图

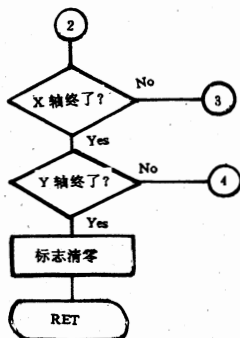


(上接第32页)

标号	助记符	地址	机器码	注释
DTCONV	LD C,(HL)	003E	4E	C←X 轴低位数据
	INC HL	003F	23	数据地址+1
	LD B,(HL)	0040	46	B←X 轴高位数据
	RES 7,B	0041	CB B8	转向标志清零
	BIT 7,(HL)	0043	CB 7E	X 轴高位数据第 7 位如果为 1 则反转, 设置标志
	JP Z,J9	0045	CA 50 00	
	LD A,(8001H)	0048	3A 01 80	
	SET 6,A	004B	CB F7	
	LD (8001H),A	004D	32 01 80	
J9	INC HL	0050	23	数据地址+1
	LD E,(HL)	0051	5E	E←Y 轴低位数据
	INC HL	0052	23	数据地址+1
	LD D,(HL)	0053	56	D←Y 轴高位数据
	RES 7,D	0054	CB BA	转向标志清零
	BIT 7,(HL)	0056	CB 7E	Y 轴高位数据的第 7 位如果为 1, 则反转. 设置标志
	JP Z,J10	0058	CA 63 00	
	LD A,(8001H)	005B	3A 01 80	
	SET 7,A	005E	CB FF	
	LD A,(8001H)	0060	32 01 80	
J10	RET	0063	C9	

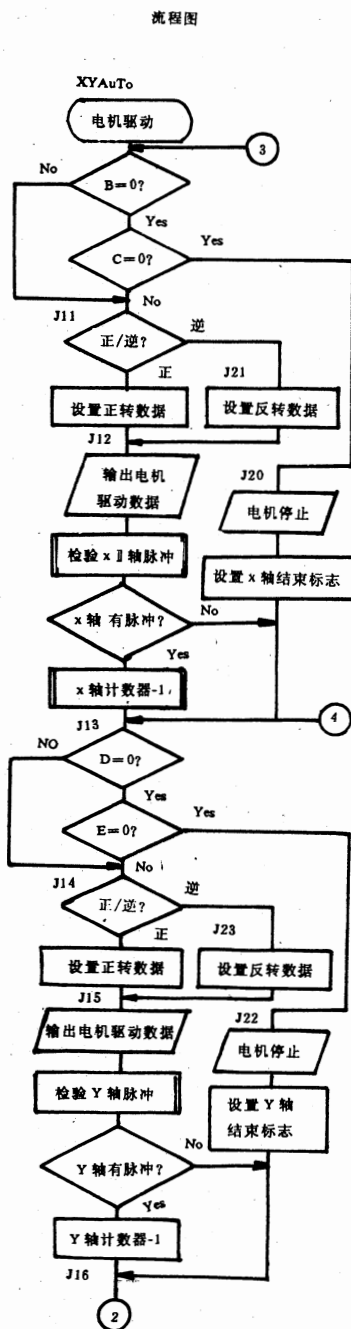
图5 数据传送程序流程图与清单

流程图



标名	助记符	地址	机器码	注释
J22	LD A,(8000H)	00E5	3A 00 80	设置 Y 轴电机停止数据
	SET 1,A	00E8	CB CF	
	SET 0,A	00EA	CB C7	
	LD (8000H)	00F1	3A 01 80	输出电机驱动数据
	OUT (02H),A	00EF	D3 02	
	LD A,(8001H)	00F1	3A 01 80	设置 X 轴结束标志
	SET 1,A	00F4	CB CF	
	LD (8001H),A	00F6	32 01 80	
	JP J16	00F9	C3 B4 00	
J23	LD A,(8000H)	00FC	3A 00 80	将电机驱动数据设置在 Y 轴反转上
	RES 0,A	00FF	CB 87	
	JP J15	0101	C3 A3 00	

图6 电机驱动程序流程图与清单



标名	助记符	地址	机器码	注释
XYAuTo	LD A,B	0064	78	X轴高位数据不为0时,向J11跳
	AND A	0065	A7	
	JP NZ,J11	0066	C2 6E 00	
	LD A,C	0069	79	X轴低位数据为0时,向J20跳
	AND A	006A	A7	
	JP Z,J20	006B	CA C6 00	
J11	LD A,(8001H)	006E	3A 01 80	反转标志成立时,向J21跳
	BIT 6,A	0071	CB 77	
	JP NZ,J21	0073	C2 DD 00	
	LD A,(8000H)	0076	3A 01 80	电机驱动数据设置在X轴正转
	RES 3,A	0079	CB 9F	
J12	LD (8000H),A	007B	32 00 80	
	OUT (02H),A	007E	D3 02	输出电机驱动数据
	CALL XPULSE	0080	CD 04 01	
	LD A,(8001H)	0083	3A 01 80	
	BIT 2,A	0086	CB 57	X轴的脉冲由H→L变化时,计数器减1
	JP Z J13	0088	CA 8C 00	
	DEC BC	008B	0B	
J13	LD A,D	008C	7A	Y轴高位数据不为0跳向J14
	AND A	008D	A7	
	JP NZ,J14	008E	C2 96 00	
	LD A,E	0091	7B	Y轴低位数据为0跳向J15
	AND A	0092	A7	
	JP Z,J22	0093	CA E5 00	
J14	LD A,(8001H)	0096	3A 01 80	反转标志成立时跳向J23
	BIT 7,A	0099	CB 7F	
	JP NZ,J23	009B	C2 FC 00	
	LD A,(008H)	009E	3A 00 80	电机驱动数据设置在Y轴
	RES 1,A	00A1	CB 8F	
J15	LD (8000H),A	00A3	32 00 80	
	OUT (02H),A	00A6	D3 02	输出电机驱动数据
	CALL YPULSE	00A8	CD 2F 01	
	LD A,(8001H)	00AB	3A 01 80	
	BIT 3,A	00AE	CB 5F	Y轴脉冲如果由H→L变化,计数器减1
	JP Z,J16	00B0	CA B4 00	
	DEC DE	00B3	1B	
J16	LD A,(8001H)	00B4	3A 01 80	X轴数据为0?
	BIT 0,A	00B7	CB 47	
	JP Z,XYAuTo	00B9	CA 64 00	
	BIT 1,A	00BC	CB 4F	Y轴数据为0?
	JP Z,J13	00BE	CA 8C 00	
	XOR A	00C1	AF	
	LD (8001H),A	00C2	32 01 80	标志清零
	RET	00C5	C9	
J20	LD (8001H),A	00C6	3A 00 80	设置X轴电机停止数据
	SET 3,A	00C9	CB DF	
	SET 2,A	00CB	CD D7	
	LD (8001H),A	00CD	32 00 80	输出电机驱动数据
	OUT (02H),A	00D0	D3 02	
	LD A,(8001)	00D2	3A 01 80	
	SET 0,A	00D5	CB C7	设置X轴结束标志
	LD (8001H),A	00D7	32 01 80	
	JP J13	00DA	C3 8C 00	
J21	LD A,(8000H)	00DD	3A 00 80	将电机驱动数据设置在X轴逆转上
	RES 2,A	00E0	CB 97	
JP	J12	00E2	C3 7B 00	

图6

(下转第31页)

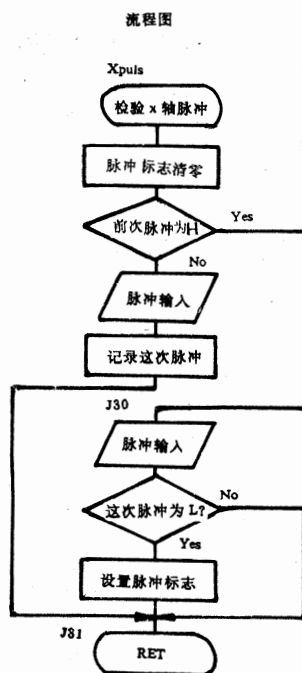


图7 X 轴脉冲检验程序流程图与清单

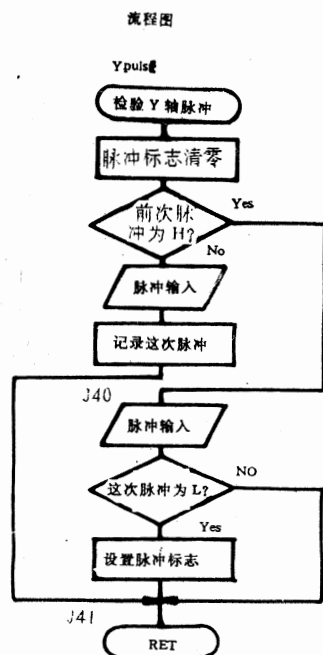
标号	助记符	地址	机器码	注释
XPULSE	LD A,(8001H)	0104	3A 01 80	X 轴有脉冲,标志清零
	RES 2,A	0107	CB 97	
	LD (8001H),A	0109	32 01 80	
	LD A,(8002H)	010C	3A 02 80	前次脉冲为 L 时,记录这次脉冲
	BIT 6,A	010F	CB 77	
	JP NZ,J30	0111	C2 1C 01	
	IN A,(00H)	0114	DB 00	
	LD (8002H),A	0016	32 02 80	前次脉冲为 H,这次脉冲为 L 时,有脉冲,设置标志
	JP J31	0119	C3 2E 01	
J30	IN A,(00H)	011C	DB 00	
	LD (8002H),A	011E	32 02 80	
	BIT 6,A	0121	CB 77	
	JP NZ,J31	0123	C2 2E 01	
	LD A,(8001H)	0126	3A 01 80	
	SET 2,A	0129	CB D7	
	LD (8001),A	012B	32 01 80	
J31	RET	012E	C9	

(5) X 轴、Y 轴脉冲检验程序(XPULS, YPULS)。

只限于各轴的脉冲从“H”电平到“L”电平变化时,设置标志,表示有1个脉冲输入。

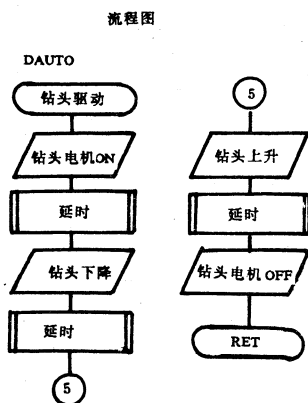
(6)驱动钻头程序(DAUTO)

钻头工作循环:钻头旋转→钻头下降→钻孔→钻头上升→钻头旋转停止。



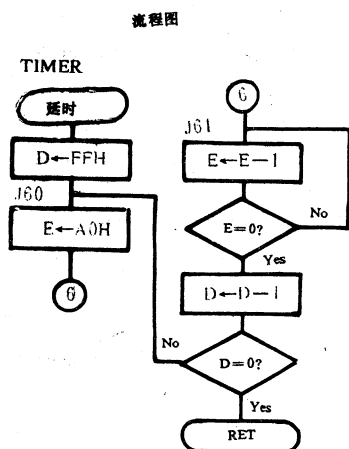
标号	助记符	地址	机器码	注释
YPULSE	LD A,(8001H)	012F	3A 01 80	Y 轴脉冲有标志清零
	RES 3,A	0132	CB 9F	
	LD (8001H),A	0134	32 01 80	
	LD A,(8003H)	0137	3A 03 80	前次脉冲为 L 时,记录这次脉冲
	BIT 7,A	013A	CB 7F	
	JP NZ,J40	013C	C2 47 01	
	IN A,(00H)	013F	DB 00	
	LD (8003H),A	0141	32 03 80	前次脉冲为 H,这次脉冲为 L 时,有脉冲,设置标志。
	JP J41	0144	C3 59 01	
J40	IN A,(00H)	0147	DB 00	
	LD (8003H),A	0149	32 03 80	
	BIT 7,A	014C	CB 7F	
	JP NZ,J41	014E	C2 59 01	
	LD A,(8001H)	0151	3A 01 80	
	SET 3,A	0154	CB DF	
	LD (8001H),A	0156	32 01 80	
J41	RET	0159	C9	

图8 Y 轴脉冲检验程序流程图与清单



标记	助记符	地址	机器码	注释
DAUTO	LD A,DFH	015A	3E DF	
	OUT (02H),A	015C	D3 02	
	CALL TIMER	015E	CD 7A 01	
	LD A,CFH	0161	3E CF	
	OUT (02H),A	0163	D3 02	
	LD B,04H	0165	06 04	
J50	CALL TIMER	0167	CD 7A 01	延时程序执行 4 次(钻孔时间)
	DEC B	016A	05	
	JP NZ,J50	016B	C2 67 01	
	LD A,DFH	016E	3E DF	
	OUT (02H),A	0170	D3 02	
	CALL TIMER	0172	CD 7A 01	
	LD A,FFH	0175	3E FF	
	OUT (02H),A	0177	D3 02	
	RET	0179	C9	

图9 钻头驱动程序流程图与清单  
(7)延时程序(TIMER)  
满足钻孔所需要的时间



标号	助记符	地址	机器码
TIMER	LD D,FFH	017A	16 FF
J60	LD E,A0H	017C	1E A0
J61	DEC E	017E	1D
	JP NZ,J61	017F	C2 7E 01
	DEC D	0182	15
	JP NZ,J60	0183	C2 7C 01
	RET	0186	C9

图10 延时程序流程图与清单

#### (8)数据

由200H 地址存放电机移动数据,结束标志为 FFH。

标记	助记符	地址	机械语
DATA	DB	0200	00 00 00 00
	DB	0204	2C 01 28 80
	DB	0208	2C 01 78 80
	DB	020C	CA 00 A0 80
	DB	0210	96 00 F0 80
	DB	0214	32 00 F0 80
	DB	0218	32 80 F0 80
	DB	021C	96 80 F0 80
	DB	0220	CA 80 A0 80
	DB	0224	2C 81 78 80
	DB	0228	2C 81 28 80
	DB	022C	2C 81 28 00
	DB	0230	2C 81 78 00
	DB	0234	CA 80 A0 00
	DB	0238	96 80 F0 00
	DB	023C	32 80 F0 00
	DB	0240	32 00 F0 00
	DB	0244	96 00 F0 00
	DB	0248	CA 00 A0 00
	DB	024C	2C 01 78 00
	DB	0250	FF

#### 数据

以上为微型钻床两轴同时移动的全部程序。

微型钻床为理想的教学装置。有订购者请与天津纺织工学院机械系 高殿斌同志联系。邮码:300160 电话:41.2833转983。



# 电脑巧开发 数字集成电路简易测试器

宁夏银川 王正英

## 一、概述:

数字集成电路在使用过程中经常需要检测好坏。比较实用有效的方法是测试其逻辑功能是否正确,本文介绍一个由 CEC-I 微机 and 可编程 I/O 器件 MC6821 构成的数字集成电路简易测试器,由于 MC6821 的两组 (PA<sub>0</sub>~PA<sub>7</sub>, PB<sub>0</sub>~PB<sub>7</sub>) 16 根双向数据

线每一根都可以通过写入控制字(编程)被确定为输入线或输出线,这样就可以利用 MC6821 在不改变接线的情况下对不同管脚排列的数字集成电路进行逻辑功能的测试。因而电路简单、制做容易,成本低廉。测试软件可由汇编语言编制、也可由 BASIC 语言编制,为了便于阅读本文的测试软件由 BASIC 语言编制。表(1)

型号	型号
74LS00	CD4002
02	4078
03	4012
04	4068
11	4072
20	4073
30	4069
32	4085
51	4043
54	4013
66	4027
74	40106
112	4518
85	4520
125	4017
138	4511
147	4028
151	4015
160	4066
183	4051
194	4512
248	4070
280	4008
390	CC14561
42	CC14585

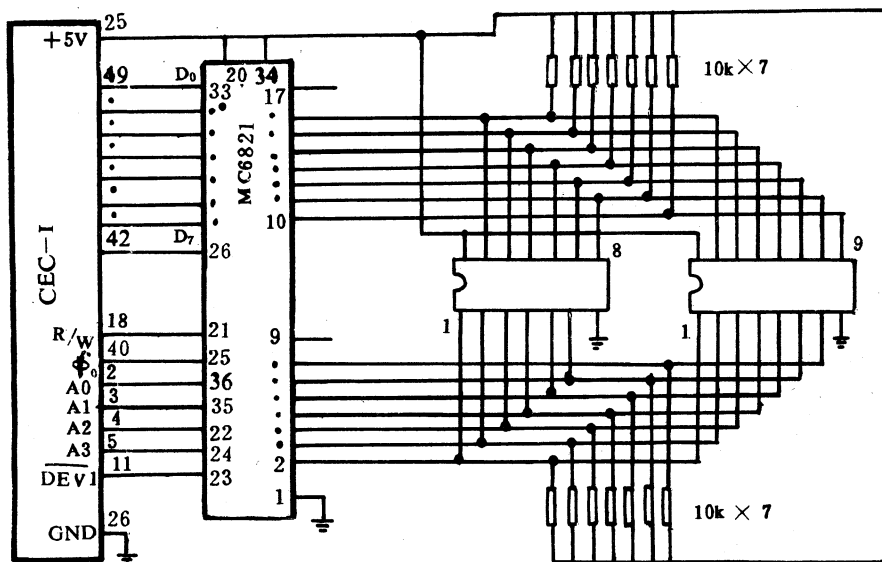


图1

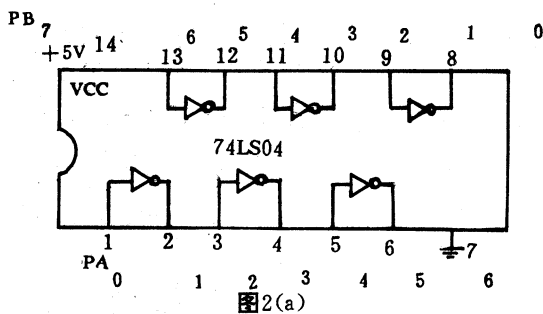
(见附程序1)。限于篇幅本文仅给出50种型号数字集成电路的数据(见表1和表3),其中包括 TTL 电路 CMOS 电路、组合电路、时序电路。OC 门以及三态门。该测试器所用的全部元件只需一块 MC6821 集成电路,两个集成电路插座和14个  $\frac{1}{8}$  W 电阻,只要接线正确,不需调试即可成功。具体电路如图1所示,该测试器与 CEC-I 微机的1号扩充槽接口。

## 二、测试原理简介

首先将选定的各种数字集成电路正确的逻辑关系(各种输入输出组合)以一定的结构方式作为数据送入内存。当测试某芯片时首先把对应该芯片的控制字写入 MC6821 来确定 PA<sub>0</sub>~PA<sub>7</sub>, PB<sub>0</sub>~PB<sub>7</sub> 是输入线还是

输出线(有关 MC6821 的编程详见本刊1990年第十期),以便与被测芯片正确连接。然后利用 POKE 命令通过 MC6821 向被测芯片送入相应的各种逻辑电平,接着再用 PEEK 命令从被测芯片读得输出电平,并与已存入内存的正确的逻辑关系比较,若一致说明被测芯片是好的,否则是坏的,下面以 74LS04 (6反相器)为例加以说明:测试 74LS04 时该芯片各引脚与 MC6821 的 PA 口、PB 口的连接如图(2a)所示。其中 PA<sub>0</sub>、PA<sub>2</sub>、PA<sub>4</sub>、PB<sub>2</sub>、PB<sub>4</sub>、PB<sub>6</sub> 与 6 个反相器的输入端相连,对 MC6821 而言是输出端故这些线被定义为输出线(1)。PA<sub>1</sub>、PA<sub>3</sub>、PA<sub>5</sub>、PB<sub>1</sub>、PB<sub>3</sub>、PB<sub>5</sub> 与 6 个反相器的输出端相连,对 MC6821 而言是输入端,故这些线定义为输入线(0)。PA<sub>6</sub>、PA<sub>7</sub>、PB<sub>6</sub>、PB<sub>7</sub> 未使用,在这里定义为输出线(1)。这样对 74LS04 写入的控制字如图(2b)所示。化为十进制数为 PA 口:213, PB 口:213。在测试程序中利





用 POKE 命令就可以把 74LS04 的控制字写入 MC6821, 从而完成了 MC6821 与 74LS04 的正确连接。74LS04 对应于 PA 口和 PB 口的真值表如表(2)所示。

表2

PA <sub>7</sub> PA <sub>6</sub> PA <sub>5</sub> PA <sub>4</sub> PA <sub>3</sub> PA <sub>2</sub> PA <sub>1</sub> PA <sub>0</sub>								PB <sub>7</sub> PB <sub>6</sub> PB <sub>5</sub> PB <sub>4</sub> PB <sub>3</sub> PB <sub>2</sub> PB <sub>1</sub> PB <sub>0</sub>							
空	空	6	5	4	3	2	1	空	13	12	11	10	9	8	空
42	0	0	1	0	1	0	1	0	0	0	1	0	1	0	0
21	0	0	0	1	0	1	0	1	0	1	0	1	0	0	0

由表(2)可知从 PA 口向 74LS04 送入的数据(十进制数)应为 42, 21。从 PB 口向 74LS04 送入的数据(十进制数)应为 42, 84。测试程序运行时以这组数据作为标准即可判断出 74LS04 的好坏, 从而完成了测试任务。

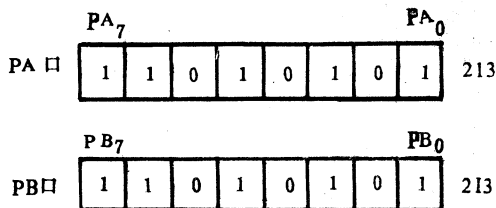


图2(b)

### 三、使用方法

1 测试前将表(3)的数据在监控状态下送入内存,

并以 SJ 为名存盘。

表(3)

4000- 00 00 DB ED 00 01 00 08  
4008- 00 02 F6 B7 00 09 00 10  
4010- 00 03 DB ED 00 11 00 18  
4018- 00 04 D5 D5 00 19 00 1C  
4020- 00 0B DF DD 00 1D 00 26  
4028- 00 14 DF FD 00 27 00 32  
4030- 00 1E FF FD 00 33 00 46  
4038- 00 20 DB ED 00 47 00 4E  
4040- 00 33 DF FD 00 4F 00 5C  
4048- 00 36 DF FF 00 5D 00 6E  
4050- 00 56 DB ED 00 6F 00 76  
4058- 00 4A CF F9 00 77 00 8C  
4060- 00 70 8F FE 00 8D 00 B4  
4068- 00 55 8F FF 00 B5 00 D0  
4070- 00 7D DB ED 00 D1 00 D4  
4078- 00 8A BF 80 00 D5 00 EC  
4080- 00 93 9F DE 00 ED 01 00  
4088- 00 97 CF FF 01 01 01 22  
4090- 00 A0 FF F3 01 23 01 40  
4098- 00 B7 CF F5 01 41 01 50  
40A0- 00 C2 FF 87 01 51 01 6E  
40A8- 00 F8 FF 80 01 6F 01 94  
40B0- 01 18 CF FF 01 95 01 A8  
40B8- 01 86 8B E8 01 A9 01 C8  
40C0- 00 2A 80 F8 01 C9 01 E8  
40C8- 0F A2 FE BF 01 E9 01 F4  
40D0- 0F EE FE BF 01 F5 02 08  
40D8- 0F AC FE BF 02 09 02 14

40E0- 0F E4 FE BF 02 15 02 28  
40E8- 0F E8 FE BF 02 29 02 34  
40F0- 0F E9 FE BF 02 35 02 3E  
40F8- 0F E5 D5 D5 02 3F 02 42  
4100- 0F F5 F3 FF 02 43 02 50  
4108- 0F CB FC FC 02 51 02 5E  
4110- 0F AD FC 9F 02 5F 02 74  
4118- 0F BB FC 9F 02 75 02 9C  
4120- 9C AA D5 D5 02 9D 02 A0  
4128- 11 A6 C3 C3 02 A1 02 D8  
4130- 11 A8 C3 C3 02 D9 03 1E  
4138- 0F B1 F0 80 03 1F 03 54  
4140- 11 9F FF 80 03 55 03 80  
4148- 0F BC 80 9E 03 81 03 A0  
4150- 0F AF E1 E1 03 A1 03 B8  
4158- 0F E2 DB F3 03 B9 03 C0  
4160- 0F D3 BB FF 03 C1 03 E4  
4168- 11 A0 FF C3 03 E5 04 0A  
4170- 0F E6 F3 E7 04 0B 04 12  
4178- 0F A8 FF C1 04 13 04 22  
4180- 38 E1 E1 87 04 23 04 3E  
4188- 38 F9 FB E7 04 3F 04 54  
4190- 24 12 36 5A 2D 36 1B 6C  
4198- 09 48 24 24 12 12 36 36  
41A0- 24 12 36 5A 2D 36 1B 6C  
41A8- 2A 2A 15 54 00 00 10 50  
41B0- 0A 08 05 04 3F 7E 20 02  
41B8- 30 42 28 22 22 0A 21 06  
41C0- 1B 6C 00 02 00 22 00 12

41C8- 20 02 10 02 08 02 04 02  
41D0- 02 02 01 02 3F 30 00 00  
41D8- 36 5A 2D 36 3F 7E 20 02  
41E0- 34 52 2A 2A 1F 06 19 7C  
41E8- 06 82 07 60 20 00 32 50  
41F0- 29 28 07 64 1F 7C 03 00  
41F8- 1C 00 00 1C 00 60 00 00  
4200- 36 5A 2D 36 1B 6C 2A 2A  
4208- 29 4A 2D 5A 2B 6A 1F 7C  
4210- 1B 6C 1F 7C 19 4C 2D 5A  
4218- 11 44 1B 6C 69 12 69 72  
4220- 11 71 19 73 18 63 1B 7B  
4228- 6A 6A 6B 7A 6A 6A 69 72  
4230- 68 62 6D 76 1C 67 1D 77  
4238- 1C 67 1F 7F 6E 6E 6F 7E  
4240- 1E 6F 6E 0E 10 40 41 00  
4248- 10 10 40 20 10 08 40 04  
4250- 10 02 40 01 18 00 42 00  
4258- 20 00 20 00 0A 00 50 00  
4260- 00 00 36 36 60 3F 61 5F  
4268- 62 6F 63 77 64 7B 65 7D  
4270- 66 7E 27 7F 40 7F 78 7F  
4278- 50 7F 48 7F 7F 3F 7F 1C  
4280- 6F 1F 17 3E 1B 3F 5D 3E  
4288- 5E 3F 3F 2E 3F 37 7F 3A  
4290- 68 00 18 00 2B 7C 12 02  
4298- 2E 7E 10 41 2F 5D 10 13  
42A0- 2F 77 27 78 14 04 2D 7A  
42A8- 11 06 2F 39 10 25 2F 6B

42B0- 10 0F 1C 01 1D 00 1F 38  
 42B8- 41 3B 43 07 41 07 43 67  
 42C0- 41 67 43 03 41 03 43 23  
 42C8- 01 23 03 23 41 21 43 21  
 42D0- 00 00 21 42 24 22 15 68  
 42D8- 28 12 19 58 1C 38 3D 7A  
 42E0- 00 03 15 03 15 57 01 51  
 42E8- 01 2D 03 29 03 55 01 52  
 42F0- 01 26 41 22 41 4E 03 48  
 42F8- 03 4C 41 48 41 4C 1C 5F  
 4300- 5C 0C 1D 3B 5D 3E 1E 6C  
 4308- 5E 76 1F 77 5F 1C 3C 7F  
 4310- 7C 3F 3D 23 7D 26 3E 68  
 4318- 7E 72 3F 63 7F 00 77 00  
 4320- 04 00 7B 7F 10 00 28 00  
 4328- 1A 00 2B 00 1B 40 2B 60  
 4330- 1B 70 2B 78 1B 7C 2B 7E  
 4338- 0B 68 09 68 0C 18 0D 58  
 4340- 08 08 09 48 11 44 19 4C  
 4348- 21 42 29 4A 31 46 39 4E  
 4350- 41 41 49 49 01 40 09 48  
 4358- 7E 07 7D 47 7B 27 77 67  
 4360- 6F 17 5F 57 3F 37 7F 76  
 4368- 7F 0D 7F 4B 7F 2F 7F 6F  
 4370- 7F 1F 7F 5F 7F 3F 7F 7F  
 4378- 01 40 02 04 04 08 08 10  
 4380- 10 20 1E 3C 00 40 03 00  
 4388- 05 00 09 00 11 00 01 04  
 4390- 01 08 01 10 01 20 1F 3C  
 4398- 1E 3C 1D 78 1B 74 17 6C  
 43A0- 0F 5C 01 40 1F 3C 1C 7C  
 43A8- 1A 7C 16 7C 0E 7C 1E 78  
 43B0- 1E 74 1E 6C 1E 5C 00 62  
 43B8- 00 00 03 44 05 48 09 50  
 43C0- 11 60 1F 7C 00 00 0F 60

43C8- 15 52 1A 32 3F 7E 2A 2A  
 43D0- 15 54 0C 00 2E 44 1D 22  
 43D8- 33 66 33 00 00 66 00 18  
 43E0- 3B 2B 13 03 54 44 10 00  
 43E8- 3B 2B 47 47 2B 2B 21 42  
 43F0- 01 40 06 30 02 20 06 30  
 43F8- 12 24 15 54 11 44 15 54  
 4400- 24 2C 0B 6A 2A 2A 61 43  
 4408- 6B 6B 2A 2A 22 22 25 52  
 4410- 21 42 25 52 31 46 36 36  
 4418- 32 26 34 56 01 40 05 50  
 4420- 11 44 14 34 12 24 16 34  
 4428- 02 20 06 30 2A 2A 15 54  
 4430- 42 42 02 02 07 07 06 06  
 4438- 0B 0B 0A 0A 0F 0F 0E 0E  
 4440- 13 13 12 12 17 17 16 16  
 4448- 1B 1B 1A 1A 1F 1F 1E 1E  
 4450- 23 23 22 22 27 27 26 26  
 4458- 03 03 01 01 00 00 02 02  
 4460- 04 04 06 06 08 08 09 09  
 4468- 42 42 02 02 07 07 06 06  
 4470- 0B 0B 0A 0A 0F 0F 0E 0E  
 4478- 13 13 12 12 17 17 16 16  
 4480- 1B 1B 1A 1A 1F 1F 1E 1E  
 4488- 20 20 22 22 24 24 26 26  
 4490- 28 28 2A 2A 2C 2C 2E 2E  
 4498- 30 30 32 32 34 34 36 36  
 44A0- 38 38 3A 3A 3C 3C 3C 3C  
 44A8- 3D 3D 3F 3F 3D 3D 48 04  
 44B0- 08 04 28 02 08 02 28 08  
 44B8- 08 08 28 40 08 40 2A 00  
 44C0- 0A 00 20 01 00 01 20 10  
 44C8- 00 10 20 20 00 20 21 00  
 44D0- 01 00 24 00 04 00 28 04

44D8- 08 04 28 02 38 02 28 08  
 44E0- 08 08 28 40 08 7F 67 00  
 44E8- 0C 5F 4C 0C 0D 3B 4D 3E  
 44F0- 0E 6C 4E 76 0F 67 4F 1C  
 44F8- 2C 7F 6C 7C 2D 00 6D 00  
 4500- 2E 00 6E 00 2F 00 6F 00  
 4508- 7F 00 6C 7C 7C 7C 5C 7C  
 4510- 04 00 00 22 02 10 00 52  
 4518- 01 08 20 0A 40 18 08 1A  
 4520- 00 05 10 06 00 14 00 16  
 4528- 00 0C 00 0E 00 1C 00 1E  
 4530- 61 61 41 41 40 40 51 51  
 4538- 10 10 09 09 48 48 55 55  
 4540- 14 14 0B 0B 4A 4A 55 55  
 4548- 3F 7E 30 60 0F 1E 06 0C  
 4550- 07 50 03 54 07 52 03 56  
 4558- 07 51 03 55 07 53 03 57  
 4560- 18 28 1C 2C 18 2A 1C 2E  
 4568- 18 29 1C 2D 18 2B 1C 2F  
 4570- 3C 28 27 40 55 20 55 04  
 4578- 55 28 55 0C 55 30 55 14  
 4580- 55 38 55 1C 01 02 7F 7F  
 4588- 00 7C 2A 01 2A 25 2A 09  
 4590- 2A 2D 2A 11 2A 35 2A 19  
 4598- 2A 3D 00 00 2E 3A 1D 5C  
 45A0- 33 66 00 00 00 1F 2A 5E  
 45A8- 2A 61 55 1E 55 21 7F 60  
 45B0- 7F 7F 08 08 24 10 32 20  
 45B8- 10 48 18 40 14 38 1C 30  
 45C0- 12 28 1A 20 16 18 1E 10  
 45C8- 11 08 19 00 0F 04 08 50  
 45D0- 0A 10 48 10 08 12 08 10  
 45D8- 24 00 10 08 00 0C 00 09  
 45E0- 01 08 00 28 00

2 测试时将测试器与 CEC-I 微机的扩充槽接好,将被测芯片插在测试座上。

3 开机后运行程序(1),用户可根据屏幕上的提示键入被测芯片的型号,按下回车键后计算机将自动进行型号的查找,控制字的写入以及测试工作,测试完毕后屏幕上将显示出“GOOD!”或“BAD!”,当显示“BAD!”时还将同时显示出被测芯片各引脚的逻辑电平值(1或0),里圈是正确的逻辑值,外圈是错误的逻辑值。

4 测试完毕后按任意键,程序返回主菜单开始下一次测试。

#### 四、几点说明

1. 该测试器可测试14脚和16脚的数字集成电路。对 CMOS 数字集成电路要求  $V_{DD}=3\sim 18V$ 。

2. 图1电路中14个电阻是为了测试 OC 门和三态门设置的。测试其它电路时电阻 R 的引入将会在相应端头为低电平时增加输出门的灌电流,但由于 R 的阻值为10K $\Omega$ ,增加的灌电流最大不超过0.5mA,经反复试验证明 R 的引入对逻辑功能的测试没有影响。

3. 测试时序电路的数据采用了模拟时钟变化的方法,即对应时钟端的数据反复地0,1变化,0,1变化时其它端头的数据保持不变。所以对时序芯片(触发器、计数器、移位寄存器)其数据的顺序不能随意变换。

4. 表3的测试数据采用了索引区+数据区的结构如表4、表5所示。增加数据时只需在相应的区域内按表4、表5的格式填写即可。由于数据的存放采用了相对地址,所以原则上数据可放在 CEC-I 内存的任意处,此时只要对测试程序稍加修改即可。

表(4)

索引1	型号1高位	索引区
	型号1低位	
	PA口控制字	
	PB口控制字	
	第一组数据首地址高位	
	第一组数据首地址低位	
	第一组数据末地址高位	
	第一组数据末地址低位	
	:	
	型号N高位	
索引N	:	
	第N组数据末地址高位	
	第N组数据末地址低位	

表(5)

第一组数据	数据1	数据区
	⋮	
	⋮	
第二组数据	数据n1	
	数据1	
	⋮	
⋮	⋮	
	⋮	
	数据n2	
第N组数据	⋮	
	数据1	
	⋮	
	⋮	
	数据nN	

5 连续测试的过程中更换芯片是带电进行的。为了安全起见最好给被芯片的电源加一个开关。没有开关一般也不会损坏芯片。

6 该测试器也能在APPLE—Ⅱ微机上使用。只要把程序1中的地址49308改为49404,49309改为49405,49310改为49406,49311改为49407即可。此新地址对应APPLE—Ⅱ的7号扩充槽。

程序(1)

```

10 D$=CHR$(4)
20 IF PEEK(6)=5 THEN 40
30 PRINT D$ "BLOAD SJ, A$ 4000"
40 POKE 6,5
50 C$="***TTL***";D$="***CMOS***"
60 V1=16384;V2=16583;V3=16584;V4=16783;A1=16783
70 HOME
80 PRINT "1---TTL";PRINT
90 PRINT "2---CMOS";PRINT
100 PRINT "3-END";PRINT;PRINT;PRINT;PRINT
110 INPUT "(1,2,3...?)";P
120 IF P<>1 AND P<>2 AND P<>3 THEN 70
130 IF P=3 THEN 510
140 IF P=1 THEN W1=V1;W2=V2;J$=C$;GOTO 160
150 IF P=2 THEN W1=V3;W2=V4;J$=D$
160 HOME;VTAB 8;HTAB 15;PRINT J$;PRINT;PRINT;PRINT;INPUT "---->IN";I
170 FOR J=W1 TO W2 STEP 8
180 XH=PEEK(J);XD=PEEK(J+1);Y2=XH;Y1=XD
190 GOSUB 530
200 X=XZ
210 IF X<>I THEN 300
220 KA=PEEK(J+2);KB=PEEK(J+3)
230 SH=PEEK(J+4);SD=PEEK(J+5);Y2=SH;Y1=SD
240 GOSUB 530
250 SS=Z

```

```

260 MH=PEEK(J+6);MD=PEEK(J+7);Y2=MH;Y1=MD
270 GOSUB 530
280 SM=Z
290 GOTO 320
300 NEXT J
310 GOTO 70
320 POKE 49309,0;POKE 49311,0;POKE 49308,KA;POKE 49310,KB;POKE 49309,4;POKE 49311,4
330 FOR L=A1+SS TO A1+SM STEP 2
340 AX=PEEK(L);BX=PEEK(L+1)
350 POKE 49308,AX;POKE 49310,BX
360 AD=PEEK(49308);BD=PEEK(49310)
370 IF AD<>AX OR BD<>BX THEN 420
380 NEXT L
390 HOME;VTAB 8;HTAB 10;PRINT I;"---GOOD!"
400 MUSIC192,30;MUSIC171,30;MUSIC152,30;MUSIC140,30;MUSIC128,30;MUSIC114,30;MUSIC102,30;MUSIC95,110
410 GOTO 490
420 PRINT CHR$(7);HOME;VTAB 8;HTAB 13;PRINT I;"---BAD!";PRINT;PRINT
430 GOSUB 540
440 HTAB 10;PRINT D(7);";D(6);";D(5);";D(4);";D(3);";D(2);";D(1);";D(0)
450 HTAB 7;PRINT "16-";B(7);";B(6);";B(5);";B(4);";B(3);";B(2);";B(1);";B(0)
460 PRINT;PRINT;PRINT
470 HTAB 8;PRINT "1-";A(0);"-";A(1);";A(2);";A(3);";A(4);";A(5);";A(6);";A(7)
480 HTAB 10;PRINT C(0);";C(1);";C(2);";C(3);";C(4);";C(5);";C(6);";C(7)
490 POKE 49309,0;POKE 49311,0;POKE 49308,0;POKE 49310,0;POKE 49309,4;POKE 49311,4
500 GET A$;GOTO 70
510 HOME;VTAB 10;HTAB 15;PRINT "[END]"
520 END
530 Z=Y2*256+Y1;RETURN
540 FOR T=7 TO 0 STEP -1
550 A(T)=1;B(T)=1;C(T)=1;D(T)=1
560 IF AX-2^T<0 THEN A(T)=0;GOTO 580
570 AX=AX-2^T
580 IF BX-2^T<0 THEN B(T)=0;GOTO 600
590 BX=BX-2^T
600 IF AD-2^T<0 THEN C(T)=0;GOTO 620
610 AD=AD-2^T
620 IF BD-2^T<0 THEN D(T)=0;GOTO 640
630 BD=BD-2^T
640 NEXT T
650 RETURN

```

# 中华学习机调制伴音电路

武汉市解放大道200号(430030) 李永和

中华学习机的扬声器接口电路功率小,音量不可调。为此,设计制作了调制伴音电路,使声、像信号一并从电视机的天线插孔输入,调节电视机的音量电位器,可任意改变音量。

电路原理图如图1,该电路为改进型变压器反馈式振荡电路,反馈变压器B可采用任何型号电视机的6.5MHz伴音中周,变压器在电路中起正反馈作用,频带可似为足够宽。6.5MHz三端陶瓷滤波器作为选频网络,可等效为Q值极高的LC并联谐振回路,该器件为电视机伴音选频回路所采用的新型元件,故作为调制伴音的振荡选频网络具有免调试功能。反馈信号从三极管Q1的基极输入,信号幅度要求不大,起振容易。电容C2对6.5MHz振荡信号为短路,对声频信号为开路,二者经过Q1的非线性变频,幅度调制作用,从变压

器B的次级输出为调频—调幅波,由于电视机伴音解调电路有限幅功能,该电路可视为调频伴音电路。电阻R4限制负载过重,可有效防止电路停振。

电路调试极为简单,将变压器B的次级L2短路调节偏值电阻R1使电路的总电流为2mA左右,去掉短路线后,总电流上升,则说明电路起振,(一般可上升到4mA左右),否则对调L2两端点,使其为正反馈。电路装入中华学习机中,不需要屏蔽与学习机的连接见图一,虚线框内为计算机相关的接口电路。在计算机进入中文拼音状态,按英文字母任意键,使之连续发出“嘟嘟”声,将电视机音量电位器置较小音量,调整磁芯,使电视机伴音的噪音最小。(磁芯调整如否,均有“嘟嘟”声)。

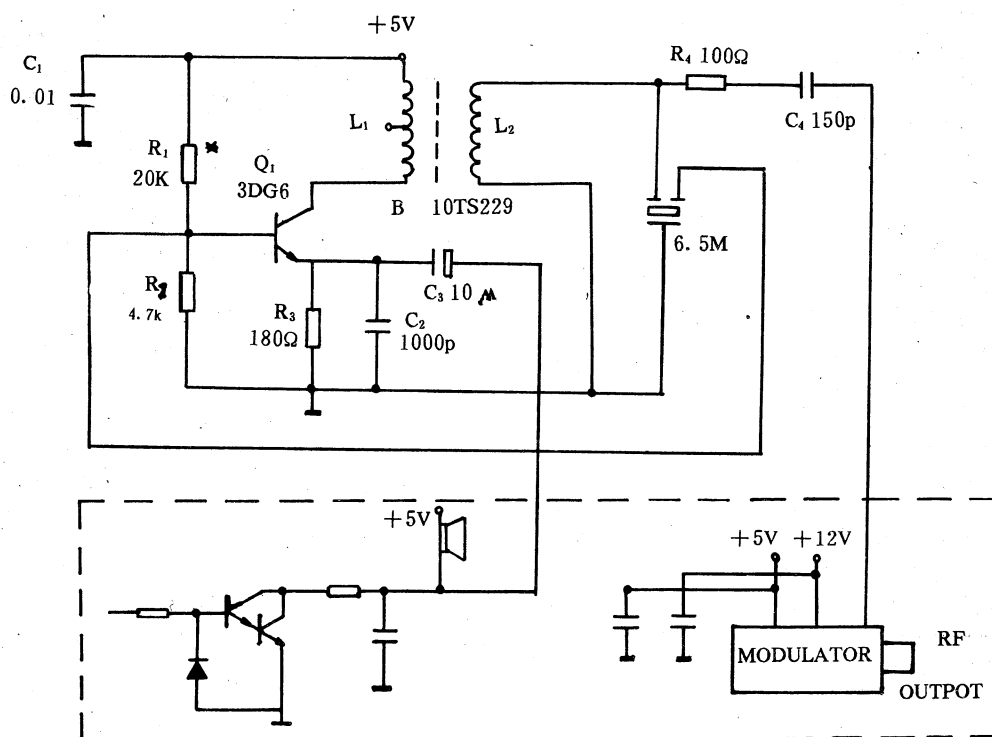


图1



# FBASIC 语言的游戏程序编程技巧

## 第四讲 游戏程序的设计过程(上)

山东苍山机械电子化学工业局(277700)于春

### 电脑游戏机

我们知道机械行业开发一种新产品,首先要进行市场需求调查,根据市场调查情况进行产品性能需求分析,以确定产品的功能要求;然后根据功能要求进行图纸、资料和各种技术细节的设计;在图纸资料完善的基础上再进行制造和装配;装配完成后还要进行调试、运行试验,只有产品完全合格后才能出厂。由此可知,在机械工业中最终的产品不仅仅是机器本身,它还包括一整套的技术资料(图纸、技术说明书、使用说明书、维护手册等)。归纳起来一个产品的开发共分需求分析、设计、制造、调试和运行五个工程阶段。

对于微机软件的开发,电脑专家们也提出了工程化管理方案。即也要经过需求分析、设计、编程、测试和运行五个阶段。一项软件产品要包括五个组成部分:系统说明书、模块说明书、程序清单、测试文件和使用维护手册等完备的技术资料。前四个阶段称为开发期,最后一个阶段称为运行期。可见前三讲我们仅讨论了游戏程序设计过程的第二、三两个阶段。当然大多数读者都是业余爱好者。设计游戏程序也仅是兴趣爱好,不是搞盈利性的软件开发,市场调查分析可不予考虑。但分析阶段形成的书面资料——系统说明书还是必需的。因为它是设计、编程、测试阶段的依据。为系统起见,下面我们仍以软件开发的五个阶段为基础,以“成龙救金凤”游戏为例,介绍游戏程序的设计过程。

#### 一、分析阶段

首先必须认真地分析开发该软件系统的目的,要选择恰当的游戏类型,要确定游戏的具体功能目标;其次要规划好游戏的玩法,游戏的参加者可有几个对手,电脑是否充当对手;还要决定是否奖励优秀、处罚不好的游戏者等等;然后确定游戏的功能目标。“成龙救金凤”游戏的功能目标如下:

1. 游戏开始,出现金凤被山鹰叼走的画面。
2. 以后各场的开始画出七层楼房。九个藏宝箱散落在七层楼中。成龙位于楼底层左边入口处。金凤在七层楼顶右上角。妖怪在3~7层楼之间。
3. 游戏者使用1号操纵器的方向键控制成龙运动。运动中伴有脚步声。
4. 成龙必须按电脑约定的顺序依次打开九个箱子。每打开一个加10分,积分达到90分时游戏者胜。救人成功。奏乐。显示救出金凤,两人拥抱回家的画面。
5. 当游戏者被妖怪抓住或者时间已到而尚没有打开九个箱子则发声,游戏结束。转结束或继续选项画

面。

上述五项功能目标就是一份简单的系统说明书。

#### 二、设计阶段

设计阶段要求设计出符合系统说明书要求的软件系统方案,并从几个方案中优选出较好的方案。方案确定后,先将系统划分成几个模块,并考虑处理好模块间传递的数据与类型,模块间的调用关系,选择出较好的模块结构图,并写出各模块的功能说明书。另外,在模块结构图中必须对所有模块编号,简单说明模块的功能,必要时还要标出每个模块的输入、输出内容。“成龙救金凤”游戏的模块结构图见图一。图一中A为管理程序;B为游戏开始时调用的模块,以后每场均不再调用;C、D、E为每场游戏开始时调用的模块;F为救人成功后调用的模块;B、C、D、E、F均为条件调用模块;G、H、I为循环调用模块,这三个模块构成游戏过程;K为结束处理模块,仅在时间到没有打开箱子、成龙被妖怪抓住和救人成功三种情况下才调用。由于各模块功能较简单,已在模块结构图中标出,不再分模块作功能说明。

#### 三、编程阶段

本阶段的任务就是为每个模块编写程序,将模块说明书转换成用某种程序设计语言书写的源程序。为使读者能够编写逻辑上正确、结构合理、层次分明、思路清晰、便于阅读和理解的程序,我们介绍由顶向下、逐步求精的程序设计方法。其基本方法是:以模块说明书为基础,对每个模块的抽象算法由粗至精(即由抽象到具体),逐步精确化,逐步扩充细节,直到成为一个能够在机器上执行的程序。

##### 1. 模块的逐步求精

逐步求精要遵循以下四点:

- (1)以合理的程序结构为目标(程序结构清晰、易读、易理解、易修改)。
- (2)严格由粗到精的顺序。关于细节的决定尽可能往后推迟,以避免整体结构与局部细节的纠缠,使细节的修改不影响上一级的抽象结构。因而程序的易改性好。
- (3)程序结构精细化的同时进行数据结构的精细化。
- (4)一边编写程序,一边检查其正确性,以提高程序的可靠性,减少返工。



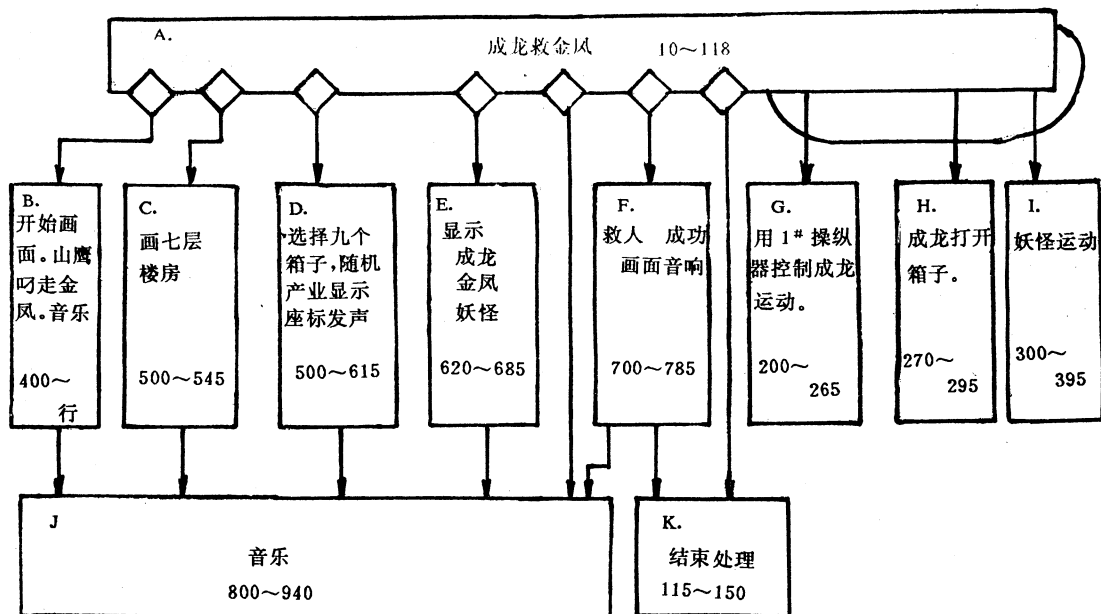


图1 模块结构图

## 2. 用子程序做公共模块

在程序的编写过程中，要注意把许多模块都调用的公用模块作为子程序来编写，以便于调用。对公用模块，必须高质量地编写好程序。这样既可使主程序的逻辑结构清晰，又可使程序的可靠性和效率提高。

## 3. “成龙救金凤”游戏的详细设计。

首先对照模块功能分别设计出各模块的流程图，然后根据流程图编写出详细程序。

### (1)模块B:

画出开始画面。时间选在下午太阳刚刚落山，残霞满天。暗红色的背景。一条长长的路，路边有树，天空飘着白云。在悠扬的音乐声中，成龙与金凤手拉手从右向左悠闭散步，当走到白云下方时，从云中冲出埋伏的山鹰，叼走了金凤。天空变晴，成龙尾随狂奔，音乐紧促激烈。流程图如下：

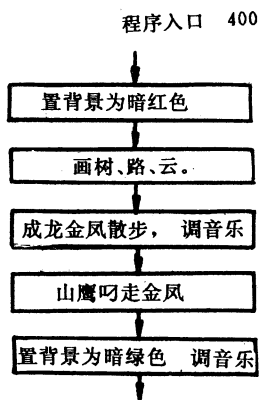


图2、模块B流程图

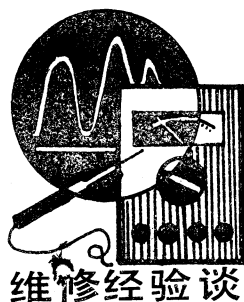
对照图2写出详细程序。行号分配400

```
400 REM "BEGIN PICTURE"(开始画面)
405 CG.0,1;SP.O.;CLS;PAL.B 0,8,48,47,47
410 F.I=0 TO 27;LOC.I,12;P.CH.(197);N.(画路)
415 LOC.10,2;P.CH.(216);CH.(218);LOC.8,3;P.CH.
(216);CH.(217);CH.(218)(画白云)
420 LOC.1,8;P.CH.(208);LOC.0,9;P.CH.(249);CH.
(203);CH.(251);LOC.1,10;P.CH.(238);LOC.1,11;
P.CH.(212)(画树)
425 LOC.2,16;P."CHENG _LONG _SAVES _JIN
_FENG";LOC.8,20;P."BY _ _ _ YU _CHUN"
430 DE.M.(4)=SP.(0,7,8,90,0,3);POS.4,220,100;M.4
(成龙)
435 DE.M.(5)=SP.(1,7,8,80,0,3);POS.5,230,104;M.5
(金凤)
440 PL."M0V12M1V12Y1T4;M0V9T4;M1V2T4";FF=1;
GOSUB800(散步音乐)
445 DE.M.(6)=SP.(15,5,1,23,1,0);POS.6,88,48;M.6
(鹰)
450 IF CR.(6)<>5 T.450
455 DE.M.(5)=SP.(1,2,1,58,0,3);POS.5,70,104;M.5
460 DE.M.(6)=SP.(15,2,1,55,0,0);POS.6,76,96;M.6
(叼走金凤)
465 PAL.B 0,10,48,47,47(置暗景)
470 DE.M.(4)=SP.(0,3,1,94,0,3);POS.4,60,100;M.4
475 GOS.900;ERA 4(紧促音乐)
480 RE.
```

### (2)模块C:

画七层楼房。用187号背景图案画地板，用210号画楼梯。使用变量X、Y、B\$、J\$。流程图见图3。对应程序如下：





# 笔记本型电脑技术和市场分析

哈尔滨建材工业学校(150036) 赵广恩

## 维修经验谈

到目前为止,从美国日本以至于亚洲,欧洲等市场上,笔记本型电脑产品的种类已不胜枚举,随着计算机技术的发展,其竞争也将随之加剧,而竞争的结果将导致技术的更加完善和强化,性能价格比优异,从而也将会被更多的用户所接受。

从技术角度上讲,它的体积、主机、显示器,内存, I/o 接口及运行速度很多都可与 PC 机相媲美。也正是这一点,吸引了众多使用者的青睐。虽然目前多数的用户还只局限于房地产业、保险业、工程人员的文书处理类,但就其发展趋势而言,其市场是不可低估的。

目前我国市场上也相继出现了一些笔记本型电脑,主要来自美国、日本和港台地区。由于其价格偏高,销路尚不可乐观。

台湾销售额占世界市场22%之多,仅次于日本占第二位,尽管如此也还有许多隐忧。这其一是来自零部件价格及货源状况的影响。诸如 LCD、CPU 和硬磁盘机等主要组件依然主要由日、美厂商控制,从而在某种

程度上阻碍了生产能力,速度和技术的发展。其二是笔记本型电脑的生命周期相当短,人们常用所谓的“六、三、三”来比喻开发一个新机种要六个月,密集销售期三个月、之后便是三个月清仓出库。仅以去年的386sx-16为例,即使是配备了20MB 硬盘机也乏人光顾了。因为386SL、DX 和486的新机型问世,使其受到了较大的冲击、而今,掌上电脑又开始推出,其价格和使用范围使笔记本型电脑在某种意义上受到一定冲击。

在欧美市场许多消费能力较低的学生也开始关心起此类电脑,而笔记本型电脑已开始步入消费性产品的行列,人们对电脑知识也不断丰富,基于某种原因,人们对电脑性能价格比的要求也越来越高,从而使得笔记本型电脑的技术水平也不断提高。

预计全球笔记本型电脑的需求量到九五年有可能达到七百二十万台左右。其前景是乐观的,但发展中的困难不可忽视,需要各界不断努力,追求产品的质量和性能更优异和完善。(下表为几种笔记本型电脑指标)

公司名称	AST ASIA	AST	ABC COMPUTER CO. LTD.	NCR(H. K) LTD	MUL MI CORWARE USA LTD.
型号	AST PREMIU EXEC 386SX/20	PE 386SX-25C	ABC NYCOM N440i	NCR 3120	MITAC 3026E
中央处理器	20MHz 386SX	25Hz 80386SX	20MHz 386SX	20MHz 386SX	20MHz 386SX
协处理器	387 SX 插口		387 SX 插口	387 SX 插口	387 SX 插口
存储设备	FDD	3.5" 1.44MB×1	外部扩展	3.5" 1.44MB×1	3.5" 1.44MB×1
	HDD	2.5" 20MB×1	60MB×1	30/60MB×1	30/60MB×1
显示器	640×480 CCFT 32 灰度级 VGA, LCD	32 级灰度级 VGA, LCD	3 个双 640×480 带背景光 LCD	VGL LCD 16 级灰度级	640×480VGA LCD 32 级灰度级
I/O 接口	一串,一井,一监视器口,一个键盘口	一串,一井,一鼠标器口	一串,一井,一外部软盘机等	一串,一井,一个鼠标口,一个视频口	一串,一井,一 PS/2 键盘口,一 VGA 口
键盘	82/38 键		86 键全尺寸	85/86 键	81/82 键
体积(mm) <sup>3</sup>	280×289×57	280×220×55	210×280×30	210×295×44	221×294.6×58.4
重量	3.2(Kg)	71bs	2Kg	2.5Kg(包括电池)	3.4Kg
电池使用时间	3 小时		4 小时	2 小时	3 小时
随机存取存储器	2MB~8MB	4MB~8MB	2/4MB	1/5MB	1MB~2/5MB

# 用软件对软驱进行简单读写检测

北京海淀区 李晓中 麻佳洛 谢静 张景生

在 IBM—PC/XT 或 AT 使用中,有时在进行 DIR 目录显示时正常工作,但在进行读盘/写盘操作时出错;有时候读写正常,但有时候又连续出错。这时,用硬件检测或执行系统检测程序却发现驱动器工作正常。究竟是驱动器故障还是软磁盘损坏,一时难以判别。笔者在实践中编写了一段读盘检测小程序,以帮助维修人员了解目前系统盘驱动器的故障所在并认识故障的性质。

## 检测程序:

本程序采用汇编语言编写,使用系统的 INT13 中断调用,对驱动器进行逐面、逐道、逐扇的读/写,并在读/写过程中不断检测 INT13 调用的返回参数,以判断是否有错。若有错则继续分析是数据读/写错还是控制信号或其它类型错,本文只给出了数据读出和其它类型错,读者也可以扩充分析其它更多的类型错(利用 INT13 调用的返回参数 AH 的定义进行分析)。

使用此程序可以判别区分驱动器故障和软磁盘故障。也可以具体地检测到软盘的哪些磁道有问题。

比如,在实践中,发现一驱动器 0 面读/写全部正常,而 1 面 35 道以后的各道扇区读/写发生数据类型错,可以分析为两方面故障:①因软驱磁头在高磁道区的定位不准;②读出电路故障,对内道的高密度信息因信号幅度变化而出现的峰点飘移造成读出错误。

可以从以上两个方面进行检查维修。

下面给出检测程序:

```
1:STACK      SEGMENT PARA STACK 'STACK'
2:           DB 128 DUP(0)
3:STACK      ENDS
4:DATA       SEGMENT PARA PUBLIC 'DATA'
5:DAT1       DB 0DH,0AH,20H,20H,20H,'
              欢迎使用磁盘诊断软件,'20H
6:           DB '空军电讯工程学院电子技术
              系',0DH,0AH
7:           DB '软件编制:设备教研室
              1991年12月09日',0DH,0AH
8:           DB 0DH,0AH,0DH,0AH,'请输入
              驱动器号 A/B/C'
9:           DB 0DH,0AH,'$'
10:DAT2      DB '输入驱动器号错!请重新输
              入!',0AH,0DH,'$'
11:DAT3      DB '本驱动器操作完全正确!',
              0DH,0AH,'$'
12:DAT4      DB '驱动器操作有错!请检查维
              修!',0DH,0AH,'$'
13:DAT8      DB '读数据错!',0dh,0ah,'$'
```

```
14:DAT9      DB '其它类型错!',0DH,0AH,'
              $'
15:CLOR      DB 1
16:ERRF      DB 0 ;存错误码
17:SORT      DB 01H ;扇区号
18:CRLF      DB 0DH,0AH,'现在检测驱动器,
              请稍等!',0DH,0AH
19:          DB '用 CTRL—C/CTRL—Break
              键可随时退出!',0DH,0AH,24H
20:KEY1      DB 00H ;驱动器号暂存
21:TRACK     DB 00H ;磁道号计数器
22:HEAD      DB 00H ;磁头(0/1)计数
              器
23:AAASMG    DB 0DH,'正在检测',00,'面',
              00,00,'道',00,'扇区!',20H,
              20H,'S'
24:ERRLB     DB 00 ;操作有错标志(非零错)
25:BUF       DB 512 DUP(0)
26:DATA      ENDS
27:CODE      SEGMENT PARA PUBLIC 'CODE'
28:START     PROC FAR
29:           PUSH DS
30:           MOV AX,0
31:           PUSH AX
32:           ASSUME CS:CODE,DS:DATA
33:           ASSUME SS:STACK,ES:DATA
34:           MOV AX,DATA
35:           MOV DS,AX
36:           MOV ES,AX
37:           MOV AX,STACK
38:           MOV SS,AX
39:           MOV AX,0003H
40:           INT 10H
41:           MOV AX,0B00H
42:           MOV BX,0001H
43:           INT 10H
44:           MOV AX,0600H
45:           MOV BX,4F00H
46:           MOV CX,0
47:           MOV DX,184FH
48:           INT 10H
49:           MOV CX,OFFSET DAT1
50:           CALL XS
51:           KEY:MOV AH,01H
52:           INT 21H ;接收键盘
53:           AND AL,0DFH
54:           MOV CRLF+16,AL
```

```

55:      OR AL,20H
56:      CMP AL,'a'
57:      JZ DISKA
58:      CMP AL,'b'
59:      JZ DISKB
60:      CMP AL,'c'
61:      JZ DISKC
62:      MOV CX,OFFSET DAT2
63:      CALL XS
64:      JMP KEY
65:DISKA:  MOV KEY1,00H
66:      JMP NEXT
67:DISKB:  MOV KEY1,01H
68:      JMP NEXT
69:DISKC:  MOV KEY1,80H
70:NEXT:  MOV CX,OFFSET CRLF
71:      CALL XS

```

72:;-----

—功能程序如下

```

73:      JMP NEXT1
74:NN1:   INC AH
75:      JMP NN0
76:NEXT1: MOV AL,HEAD
77:      OR AL,30H
78:      MOV AAASMG+9,AL
79:      MOV AL,TRACK
80:      MOV AH,00H
81:NN0:   SUB AL,10
82:      JNC NN1
83:      ADD AL,10
84:      OR AL,30H
85:      OR AH,30H
86:      MOV AAASMG+12,AH
87:      MOV AAASMG+13,AL
88:      MOV AL,SORT
89:      OR AL,30H
90:      MOV AAASMG+16,AL
91:      MOV,CX,OFFSET AAASMG
92:      CALL XS
93:      MOV AH,02H      ;功能号(读盘)
94:      MOV AL,01H      ;传送扇区数
95:      MOV BX,OFFSET BUF ;缓冲区首址
96:      MOV CH,TRACK     ;磁道号
97:      MOV CL,SORT      ;扇区号
98:      MOV DH,HEAD      ;磁头号
99:      MOV DL,KEY1      ;驱动器号
100:     INT 13H          ;读盘操作
101:     JC ERROR1
102:NNN:   INC SORT
103:      MOV AL,SORT
104:      CMP AL,0AH
105:      JC NEXT1
106:      MOV SORT,01H
107:      MOV CH,TRACK
108:      CMP CH,39
109:      JZ CEND1

```

```

110:     INC TRACK
111:     JMP NEXT1
112:CEND1: MOV AL,HEAD
113:     CMP AL,1
114:     JZ CEND
115:     MOV TRACK,0
116:     MOV HEAD,01H
117:     JMP NEXT1
118:CEND:  MOV AL,ERRLB
119:     CMP AL,0
120:     JNE ERROR
121:     MOV CX,OFFSET DAT3
122:     CALL XS
123:     RET
124:ERROR1: AND AH,10H;检查 D4位=1否?
125:     JNZ DERR
126:     MOV CX,OFFSET DAT9;非数据错!
127:     CALL XS
128:     JMP NN00
129:DERR:  MOV CX,OFFSET DAT8;数据类型错!
130:     CALL XS
131:NN00:   JNC ERRLB
132:     JMP NNN
133:     ERROR:MOV CX,OFFSET DAT4
134:     CALL XS
135:     RET
136:     ;-----
137:XX      PROC NEAR
138:     MOV DX,CX
139:     MOV AH,09H
140:     INT 21H
141:     RET
142:XS      ENDP
143:     START ENDP
144:     CODE ENDS
145:*      END START
          * E

```

要了解最新的电脑信息  
 要准确地选购电脑  
 要掌握电脑入门的方法  
 要学会各种软件的使用  
 要快捷地修复电脑故障  
 要您的电脑产品迅速占领市场  
 要您的电脑产品艺术地再现

请订阅 《电脑》 杂志

请在 《电脑》 杂志刊登广告

《电脑》杂志为月刊刊号CN44—1188,邮  
 发代号46—115,全国各地邮电局(所)均可办理  
 订阅。全年12期共18元。

编辑部地址:

广州市华南师范大学 电脑杂志社

邮码:510631 电话:5516911—3273

## 普及型 PC 个人用户软件交流联谊 活动问题解答(十一)

北京中国农科院计算中心(100081)王路敬

### 43. 如果从硬盘启动 DOS 时出现“Error Loading Operating System”错误信息如何解决?

上述错误信息的含义是装 DOS 错。产生这类故障的原因常常是由于 DOS 分区柱面的损坏或者 DOS 分区表 FAT 被破坏,或硬件的读写电路坏等原因造成。

解决这类故障一般使用的方法是执行 FDISK 命令,重新建立 DOS 分区,或者使 DOS 分区起始位置避开 0 柱面,或者重做分区格式化,这是软件排除故障的方法,如果读写电路坏,需要更换硬盘控制器。但这些方法的缺点是往往会造成硬盘大量数据丢失。

如果对硬盘执行 DIR、SYS、CHKDSK 等命令时执行结果,系统发出“File Allocation Error”或“Seek error reading drive C”等出错信息,常常是文件分配表 FAT 损坏而引起的定位错误。DOS 分区的文件分配表有 2 份,其内容完全相同,FAT1 和 FAT2。一般文件分配表损坏往往是 FAT1, FAT2 并没有破坏,为此,可用 FAT2 表恢复 FAT1 表。以长城 0520CH 20MB 硬盘为例恢复 FAT1 表操作如下:

```
A>DEBUG
-L100 2 9 8      ;读 FAT2
-W100 2 1 8      ;写 FAT1
-Q
```

也可以采用下面一种方法,操作如下:

```
A>DEBUG
-A 100
××××:0100 MOV AX,0208;读 FAT2 8 个扇区
××××:0103 MOV BX,0200;置缓冲区为 CS:0200
××××:0106 MOV CX,0009;FAT2 起始扇区为逻辑 9 扇区
××××:0109 MOV DX,0080;硬盘 0 号磁头
××××:010C INT 13      ;磁盘输入/输出中断
××××:010E INT 20      ;程序终结
-G=0100 010E

××××:0110 MOV AX,0308;写 FAT1
××××:0113 MOV BX,0200
××××:0116 MOV CX,0001
××××:0119 MOV CX,0080
××××:011C INT 13
××××:011E INT 20
-G=0110 11E
-Q
```

还可以将 FAT2 写入一已格式化的空软盘,然后由软盘读出写入 FAT1 所占据硬盘扇区,操作过程与上述相似,读者可试试看。

### 44. 怎样利用 CHKDSK 命令增大磁盘空间?

不管软盘或者硬盘由于多次的进行拷贝或删除文件等操作,常常使得可用的磁盘空间减少。实践证明利用 DOS 的 CHKDSK/F 命令可以对出错的文件定位表进行修复,相对增大磁盘的空间。操作方法如下:

(1)用软盘启动系统,对 C 盘工作:

```
A>CHKDSK C:/F
```

此时无论屏幕上出现多少个(Y/N)选择,均回答 Y。

(2)待上述工作完成后,将凡是扩展名为.CHK 文件删除。

```
A>ERASE C:FILE *.CHK
```

例:长城 286BH 33M 硬盘,实现上述操作前磁盘剩余空间 12597248 字节,操作后磁盘剩余空间变为:12873748 字节。

在这里顺便提一下,CHKDSK 命令加“/V”参数可以将显示在硬盘包括根目录和所有子目录下的全部文件,不论加密与否均全部显示。屏幕显示格式为:

```
C:\子目录名\文件名
```

所有目录下的文件显示完后,屏幕提示:硬盘上有 X 个隐含文件,共有 Y 个目录,总计有 ZZZ 个文件。这样对于用 TREE、DIR 等命令列不出来的隐藏子目录和隐形文件,在 A>CHKDSK C:/V 命令下全部亮相。

### 45. 在实际应用中规定磁盘缓冲区个数的原则是什么?

因为各种应用的目的和实现手段不同,所以,没有一个规定的缓冲区数量对所有的应用都有同样的效果。最佳缓冲区数只能通过不同的值选择直到获得最佳的功能。如果用户用数据库系统,例如 dBASE III 常规定 24 个缓冲区,就可提供较好的效果。如果运行的程序有很多随机读写的记录,例如 BASIC 的随机文件,一般缓冲区数规定要多一些,这样通过加速访问时间提高程序运行的速度,增强应用功能。相反,对于顺序的读写应用,例如读写一个完整的文件,分配一个大的缓冲区没有什么优越性,反而增加内存的开销。所以,如果用户很少需要随机的读和写记录,系统的缺省值就够了。

### 46. 什么是虚拟盘?有何特点?怎样设置?

PC DOS 3.0 及更高级版本的系统结构设置文件 CONFIG.SYS 中,为 DEVICE 命令增加了安装虚拟磁盘的功能。所谓虚拟磁盘是用计算机主存储器的一部分区域作为存储介质仿真磁盘驱动器,被仿真的磁盘叫虚拟磁盘。

虚拟磁盘有下列特点:



(1) 虚拟磁盘实际进行的只是对内存的操作,因此,虚拟磁盘操作速度快,但要占用一部分内存空间。

(2) 一个系统可配置多个虚拟磁盘。每个都可以用驱动器字母引用。例如,若微机系统已安装了 A、B 两个软盘驱动器和一个硬盘驱动器 C,那么,第一个定义的虚拟磁盘为 D,第二个为 E,等等。

(3) 对每一个虚拟磁盘,用户都可以规定要用的存储区域大小,扇区大小和它所包括的目录项。

(4) 每个虚拟磁盘除了要占指定或约定的内存容量外,还要增加 720 个字节的内存,用于 DOS 对它的管理。

(5) 在未指定的情况下,虚拟磁盘所占的内存区域是从 PC-DOS 的用户可用空间开始的连续区间,如果在定义语句中给定参数/E,且在主机中装有大约 1MB 内存空间的情况下,虚拟磁盘将装在从 1MB 的边界开始,而不占用用户内存区。

(6) 存放在虚拟磁盘上的文件,当在系统重新启动或掉电时,其盘上的信息要丢失。

建立虚拟磁盘的方法是在系统结构设置文件 CONFIG. SYS 中设置语句 DEVICE = VDISK. SYS,且在启动时 VDISK. SYS 要在启动盘上。命令格式为:

DEVICE = [ <盘符> ] [ <路径> ] VDISK. SYS [ <虚拟盘字节数> ] [ <扇区字节数> ] [ <目录数> ] [ /E ]

其中 [ <虚拟盘字节数> ]:指虚拟磁盘的大小,以 K 字节为单位,用十进制数表示,缺省值为 64K 字节。其值范围从 1K 字节到微机可用内存的最大值。此值的定义要求:

(1) 如果在安装虚拟磁盘时内存空间已小于 64K, VDISK 发出错误信息,且不安装虚拟磁盘。

(2) 如果指定虚拟盘字节数小于 1K 字节,或小于微机系统可用内存空间,则系统自动按 64K 字节设置。

(3) 如果规定虚拟磁盘大小后,用户可用内存空间小于 64K,系统将自动向下调整虚拟磁盘的大小,使用用户可用内存空间保持 64K 字节。

<扇区字节数>:指可允许的扇区字节数为 128、256、或 512 字节。缺省值为 128 字节。

<目录数>:指虚拟磁盘可以包括的目录数。缺省值为 64。该值允许范围为 2—512。每个目录项所占字节数为 32。如果所定义的目录数乘以 32 恰是所定义的扇区字节数的整数倍,则系统对目录自动向上进行调整。使用时注意,所定义的目录数其中一个不能被用户使用,系统用它保持卷标。

</E> 该参数表示如果在定义虚拟磁盘的语句中给定它,则虚拟磁盘将安装在内存扩展部分中,且其地址是在 1M 以上的空间内。这时,用户可以用空间不受虚拟磁盘的影响,但必须有扩展内存的硬件支持,否则系统无法安装。

例如,装配一个 256 字节的扇区和 32 个目录项的 160K 字节的虚拟磁盘 D,其操作如下:

(1) 在启动盘上 CONFIG. SYS 文件中写入语句:  
DEVICE = VDISK. SYS 160 256 32

(2) 保证在系统启动盘上有 VDISK. SYS 文件存在。

重新启动屏幕上便显示:

VDISK Version 1.0 Virtual disk D:  
Buffer size: 160KB  
sector size: 256  
Directory entries: 32

用户使用时一定要注意,因为虚拟磁盘的建立是借助于系统内存存储器的一部分空间,虚拟磁盘所有存储的文件在切断微机的电源前(或者重新启动前)必须拷贝到物理磁盘 A 或 B 或 C 盘上,否则数据将全部丢失。

另外,如果在系统启动时屏幕显示错误信息:

Bad or missing VDISK

应从以下三个方面找原因:

(1) 在 CONFIG. SYS 文件没有找到以 DEVICE = VDISK. SYS 所指明的驱动程序名,所以这时可用 TYPE 命令查看该文件中是否包含了有关虚拟磁盘的命令,如果有,语句格式是否正确。

(2) 断点地址超过了机器范围。

(3) 装入该驱动程序出错。即 DOS 没有安装 VDISK. SYS 驱动程序。这时可进一步查看系统启动盘上是否有该程序,如果没有可将 VDISK. SYS 复制到系统盘上。

47. 当从硬盘启动时屏幕出现“Invalid Partition Table”错误信息且死机,怎样用简单的方法排除?

当出现这种现象时不要急忙用低级格式化硬盘的方法解决,否则,要破坏硬盘原有的信息,可以采用下面一种简单的方法:

将 DOS 系统盘插入 A 驱动器执行外部命令 FDISK。

A>FDISK

屏幕显示命令的功能菜单:

GW Personal Computer  
Fixed Disk Setup Program Version 2.00  
Great Wall 0520C Computer  
FDISK options  
Choose one of the following  
1. Create DOS Partition  
2. Change Active Partition  
3. Delete DOS Partition  
4. Display Partition Data  
Enter choice: [1]

选择 3,删除 DOS 分区,然后按 <ESC> 键,返回主菜单后再选择 1,建立 DOS 分区。

待上述操作完成后,系统将提示用户返回 DOS,再从硬盘启动即故障已经排除。

有时候系统不能从硬盘启动,不显示任何信息转到软盘工作,但从软盘进入 DOS 后,可以转到硬盘工作,这种故障也可以采用上述操作解决之。

48. 无法进入硬盘与系统配置有何关系?

(下转第 22 页)

## 软件服务

### C—DBAG 中文数据库应用生成器

F/D \* AG V5.0

零售价:3900元

使用本系统只需按照提示输入用户需求,不需编程,各种管理软件,在功能变动时可任意改动,适用于网络及单用户环境。

功能:生成 dBASE+FoxBASE+源程序;生成任意格式报表;自动生成数据字典及文档;工程图形生成与管理。

组成:下拉/弹出式菜单生成器;查询模块生成器;数据录入与维护模块生成器;数据库文件结构生成器;彩色统计图形模块生成器;任意格式报表生成器;数据字典与文档生成器;

灵科汉卡

零售价:1200元

西文软件无需汉化即可直接使用中文,直接写屏,支持 1024×768,640×480,640×350 图形方式。支持中文排版功能。中文简体与繁体,中文与西文之间壹键转换。支持针式打印机、激光打印机及鼠标。精密宋、楷、黑、仿宋体平滑放大输出。能识别中西文窗口。显示字库不占用内存,且显示速度大大提高。

灵科加密器

零售价:180元

(上接第42页)

635 T=0;F=0;KN=2;KX=0;KY=0;KD=0;TI=800

(时间赋值为800)

640 LOC.0,22:P.“□□SCORE□”;T(打印得分。)

645 DE.M.(7)=SP.(0,0,2,4,0,RND(4));POS.7,12,168;M.7

650 DIM E(KN-1)

655 F.F=0 TO KN-1(由 KN 决定妖怪数量)

660 DE.M.(F)=SP.(11,1,2,4,0,RND(4))

665 SX=10+RND(17);SY=1+RND(10)

670 IF SCR\$(SX,SY)、“□” OR SCR\$(SX,SY-1)、“□” G.665

675 POS.F,SX\*8+12,SY\*8+16;M.F

680 NEXT

685 RE.

调整635行的 KN 可以增加妖怪的数量。KN 最大值为7。

(5)模块 G:

用操纵器控制成龙上、下楼和左右移动,与例四中的模块 D 大同小异。直接写出程序。程序入口为200,使用变量:

K:1\*操纵器键值。

可加密任何 EXE 和 COM 文件,也可对任何源程序实施加密,如 dBASE,FOXBASE 等,加密文件个数不限。安装方便,只要外插于打印机接口上,且对打印机工作无丝毫影响。加密手段方便,保密性极高。具有强有力的反跟踪功能,使解密成为梦想。

灵科加密卡

零售价:450元

具有灵科加密器的一切功能,并且做到了真正的软硬件加密结合的功能。具有开机口令功能,并且可以方便地修改口令或关断开机口令功能。彻底避免了计算机的非授权使用,减少了计算机病毒的感染机会。本卡有2K字节可供编程人员进行二次开发。不用任何其它硬件设备,只需本公司提供的软件工具或函数即可完成向卡内读写加密数据或程序,写入次数以上五万次。本卡极适合保密性高且成本较高软件加密或大、中型 MIS 系统加密使用。

灵科 MIS 管理新工具

程序结构管理及菜单生成系统

零售价:500元

辅助生成结构图,程序文档;自动生成 WINDOWS、C 语言、dBASE 菜单源程序;支持鼠标操作,模拟显示。

程序流程图管理系统

零售价:500元

支持顺序判断(IF..ELSE)、循环(WHILE)、选择(SWITCH)逻辑结构图及相应文档;自动生成 C 语言逻辑调用程序;支持鼠标操作;支持6层嵌套。

经销:中国电子器材华北公司

地址:北京市海淀区万寿路西街五号

通信地址:北京144信箱

邮政编码:100036

联系人:石立军 魏国

电话:81.1810 81.0920

本公司经营计算机及外设通信设备,欢迎来电函联系。

SX,SY;BG 面的位移量。

PX,PY;成龙在 BG 面的座标

S.;成龙的运动方向。

S\$,S\$1;成龙周围的背景数值。

程序清单如下:

200 REM“CONTROL”(控制成龙运动)

205 S=0;SX=SY

210 K=STL(0);IF K=0 G.265

215 IF K=1 T.S=3;SX=1

220 IF K=2 T.S=7;SX=-1

225 IF K=4 T.S=5 SY=1

230 IF K=8 T.S=1;SY=-1

235 PX=(XPOS(7)-12)/8;PY=(YPOS(7)-16)/8

240 S\$=SCR\$(PX+SX,PY+SY+1);S1\$=SCR\$(PX+SX,PY+SY)

245 IF S=0 OR S\$“□” G.265

250 DE.M.(7)=SP.(0,S,1,4,0,RND(4));POS.7,PX\*8+12,PY\*8+16;M.7

255 IF S=3 OR S=7 T.PL.“O0#F0RO1#F”;G.265(左、右运动的声音)

260 PL.“O1#F0RO0#FR”(上、下楼的声音)

265 RE.

(未完待续)