

E&C

《电子与电脑》专辑

●《电子与电脑》编辑部

I

CEC-I·APPLE II LASER-310·PC-1500

微电脑使用经验与技巧



● 电子工

CEC-I · APPLE II · LASER-310 · PC-1500

微电脑使用经验与技巧①

《电子与电脑》编辑部

电子工业出版社

(京)新登字 055 号

内 容 简 介

本书是为适应广大读者学习、掌握和应用计算机的需要,从实用的角度出发组织编写的,它内容丰富,深入浅出、适于自学。书中包括中华学习机、APPLE 机、PC-1500 机、LASER 机的一些文字处理、编程技巧、加密解密、打印程序、DOS 命令的修改与保护等。

书中内容具体、实用、易学,适宜广大青少年、中小學生及其各类办公人员、企业管理人员和计算机爱好者使用,也可作高等院校、职业学校、中专和中小學的参考用書。

JS453/17

CEC-I · APPLE II · LASER-310 · PC-1500

微电脑使用经验与技巧①

《电子与电脑》编辑部

电子工业出版社出版(北京市万寿路)

电子工业出版社发行 各地新华书店经营

山东电子工业印刷厂印刷

开本: 787×1092 毫米 1/16 印张: 9.75 字数 297 千字

1992 年 2 月第 1 版 1992 年 2 月第 1 次印刷

印数: 10100 册 定价: 6.00 元

书号: ISBN7-5053-1449-1 / TP · 258

目 录

第一章 系统应用与改进	(1)
§ 1.1 中华学习机	(1)
1.1.1 CEC-I BASIC 语言保留字代码	(1)
1.1.2 BASIC 和 DOS 出错信息的汉化	(3)
1.1.3 在 CEC-I 基本配置下运行 INTBASIC	(4)
1.1.4 中华学习机单键输入命令	(4)
1.1.5 自己动手扩充 BASIC 语言	(7)
1.1.6 在中西文转换后怎样保证 DOS 正常工作	(8)
1.1.7 如何为 CEC-I 机增加功能键	(10)
1.1.8 CEC-I 机的音乐语句浅谈	(10)
§ 1.2 APPLE 机	(13)
1.2.1 APPLE II PLUS 零页单元的用途	(13)
1.2.2 真正地增加磁盘容量(改进的 DOS3.3 版本)	(14)
1.2.3 行号的重新定义	(15)
1.2.4 对 DOS3.3 的一点改进	(15)
1.2.5 APPLE II 的编译 BASIC 语言及其使用	(16)
1.2.6 DOS 命令的修改与磁盘保护	(19)
1.2.7 如何使用 TOOLKIT 盘的 HRCG 软件	(21)
1.2.8 DOS3.3 系统主盘的 RETRIEVE TEXT 文件的使用	(24)
1.2.9 在 APPLE SOFT BASIC 语言环境下实现 AUTO 功能	(25)
1.2.10 APPLE SOFT BASIC 的格式输出功能	(26)
1.2.11 RENUMBER 和 CHAIN 程序的使用方法	(27)
1.2.12 制作 40 道拷贝工具软件	(29)
1.2.13 如何在 BASIC 程序中调用监控命令	(30)
§ 1.3 PC-1500 机	(31)
1.3.1 PC-1500 微机的 HZ 汉字处理系统	(31)
1.3.2 用 BASIC 实现 PC-1500 汇编程序	(35)
1.3.3 PC-1500 机 BASIC 程序加密与解密	(40)
第二章 程序与技巧	(43)
§ 2.1 中华学习机	(43)
2.1.1 用磁带存储 LOGO 过程	(43)
2.1.2 磁带程序的动态链接	(44)
2.1.3 中华学习机 LOGO 的屏幕编辑技巧	(46)
2.1.4 怎样使用好九针打印机	(46)
2.1.5 如何在 BASIC 程序中快速实现“搬家”	(47)
2.1.6 BASIC 程序的趣味拼接	(48)
2.1.7 扩展 CEC-I BASIC 字符串函数的功能	(49)
§ 2.2 APPLE II 机	(51)
2.2.1 用 BASIC 命令救回 BASIC 程序	(51)
2.2.2 恢复被 NEW 命令清除的程序	(52)
2.2.3 恢复被 DELETE 命令删除的磁盘文件	(53)
2.2.4 谈 APPLESOFT BASIC 对保留字处理的一些办法	(54)

2.2.5	巧用字符串函数	(55)
2.2.6	磁盘目录区的保护	(57)
2.2.7	“自动化”打印程序	(58)
2.2.8	两个程序的连接方法	(60)
2.2.9	文本显示与 RAM 的关系	(60)
2.2.10	自动生成 DATA 语句	(61)
2.2.11	让 APPLE 机产生更多的颜色	(62)
2.2.12	谈谈如何弥补 GET 语句的不足	(62)
2.2.13	怎样在引导 DOS 时保护 BASIC 程序	(62)
2.2.14	准确控制打印机列表的行字符数	(63)
2.2.15	在循环中改变循环变量的值	(63)
2.2.16	游戏程序的编程和技巧	(65)
2.2.17	高精度乘法原理及其程序制作	(69)
2.2.18	在 APPLESOFT BASIC 中调用机器语言排序子程序	(72)
§ 2.3	LASER—310 机	(74)
2.3.1	扩大音域	(74)
2.3.2	LASER310 机器语言程序的存放与保护	(75)
2.3.3	多功能汉字, 字符显示程序	(76)
§ 2.4	PC—1500 机	(77)
2.4.1	浅析 NEW 命令后内存的变化及恢复程序应具备的条件	(77)
第三章 综合应用		(80)
§ 3.1	中华学习机	(80)
3.1.1	奶牛技术资料管理实用微机程序	(80)
3.1.2	给电脑表格配上中文的办法	(81)
3.1.3	中华学习机高分辨率的实用绘图程序	(82)
3.1.4	从生命演变谈数组使用技巧	(83)
3.1.5	实现文本动画的方法和例子	(85)
3.1.6	利用图纹背景美化屏幕的一种方案	(86)
3.1.7	代数方程式的打印及求解	(87)
3.1.8	模糊关系方程	(89)
3.1.9	CEC—I 桥牌计分练习程序	(92)
§ 3.2	APPLE II 机	(94)
3.2.1	用 APPLE II 微机实现热工参数的自动采集和处理	(94)
3.2.2	关于网络规划计算的一个程序	(96)
3.2.3	反汇编语言的双列打印	(100)
3.2.4	APPLE II 产生模拟波形程序	(102)
3.2.5	APPLE II 代码程序倍列分页打印软件	(104)
3.2.6	APPLE II 机大写字母转化为小写字母	(106)
3.2.7	高精度输出小数	(107)
3.2.8	波动方程显示程序	(107)
§ 3.3	PC—1500 机	(108)
3.3.1	油罐剩余油量的测试	(108)
第四章 实用程序		(114)
4.1.1	最简单的 APPLE II 绘图程序	(114)

4.1.2	任意一元函数的作图程序	(114)
4.1.3	谈微机辅助《数学分析》教学的可行性与必要性	(117)
4.1.4	配平化学方程式的程序	(118)
4.1.5	正弦量的向量表示法教学演示程序	(121)
4.1.6	微机用于“人造地球卫星”的教学	(122)
4.1.7	21世纪启用的日历	(125)
4.1.8	别具特色的菜单	(130)
4.1.9	用APPLE II实现电子计时显示	(131)
4.1.10	迷你的定时钟	(133)
4.1.11	让计算机显示回答问题的时间	(134)
4.1.12	任意进制互化的一行程序	(135)
4.1.13	CEC-I 中华学习机的作图工具	(136)
4.1.14	图形字符生成工具软件 C / EDIT	(142)

第一章 系统应用与改进

§ 1.1 中华学习机

1.1.1 CEC-I BASIC 语言保留字代码

金湖宝应湖中学 刘平野

众所周知，用 BASIC 语言编制的程序是由若干具有特定意义和相应功能的字符串组成的。如语句：FOR I=0 TO 100: NEXT I 其中 FOR、TO、NEXT 等，人们习惯上将这些字符串称为保留字。在 ROM BASIC—监控程序区有一个保留字表，在程序的输入阶段，驻机解释程序扫描输入的字符串，与保留字表逐项对照，以确定是否是这张表中的保留字，当发现是其中任何一个时，就用一个表示它的代码来代替它并存储在 RAM 程序区，这就是人们所说的压缩存储。用 \$ 80 加上检索保留字时找到保留字的指针，就能算出代码。不同的机型，BASIC 语言是大同小异的，但保留字代码却不尽相同，例如 FOR、NEXT 二个保留字在 LASER 机中代码分别为 \$ 81 和 \$ 87，而在 CEC-I 中分别为 \$ 81 和 \$ 82。在程序的执行阶段，驻机解释程序又逐行扫描 RAM 程序区，凡遇到代码，就以它作索引转去执行相应命令的机器语言子程序。输入语句时先实现保留字编码，可以缩短程序执行的时间。

据初步统计 CEC-I BASIC 语言共有 112 个保留字。下面向读者提供一种方法，求得各保留字的代码。

如上所说，CEC-I 机是将程序以压缩格式存储在 \$ 801 开始的 RAM 区，那么语句

```
10 FOR
```

的存储是这样的：\$ 801、\$ 802 是下一语句的起始地址，\$ 803、\$ 804 是本语句的行号，若 FOR 是保留字，那么 \$ 805 单元应存储它的代码，否则 FOR 应以 ASCII 码存储在 \$ 805—\$ 807 三个单元。读者可用 CALL-151 转入监控状态，然后键入

```
* 801·80F
```

屏幕上即显示上述内存单元的内容。读者可一次输入多个保留字如：

```
10 FOR: TO: NEXT: END: RUN...
```

成批检查，以加快速度。但要注意“:”的 ASCII 码是 \$ 3A

本人根据 CEC-I 随机所附的使用手册，将所有保留字及其代码以十六进制码顺序列表如下：

附表：

6000-	A9	4C	8D	F5	03	A9	75	8D
6008-	F6	03	A9	61	8D	F7	03	A9
6010-	00	8D	F2	03	A9	60	8D	F3
6018-	03	20	6F	FB	A9	00	85	00
6020-	A9	D0	85	01	A0	00	B1	00
6028-	91	00	C8	D0	F9	E6	01	D0
6030-	F5	A9	4C	8D	1C	D4	8D	90
6038-	D8	8D	90	DB	8D	D7	DC	8D
6040-	58	D8	A9	60	8D	1E	D4	8D
6048-	92	DB	8D	D9	DC	A9	61	8D
6050-	5A	D8	8D	92	D8	A9	AE	8D
6058-	1D	D4	A9	41	8D	91	D8	A9
6060-	5D	8D	59	D8	A9	DF	8D	91
6068-	DB	A9	F4	8D	D8	DC	20	00
6070-	C3	A9	4C	8D	C5	9F	8D	E6
6078-	A6	A9	90	8D	C6	9F	A9	09
6080-	8D	E7	A6	A9	60	8D	C7	9F
6088-	A9	61	8D	E8	A6	4C	96	61
6090-	48	A5	36	C9	2B	F0	0F	C9
6098-	F0	F0	04	68	6C	36	00	68
60A0-	20	F0	FD	4C	AA	60	68	20
60A8-	2B	C3	8D	83	C0	60	20	5A
60B0-	DB	AD	53	AA	C9	2B	F0	03
60B8-	4C	1F	D4	20	35	61	20	83
60C0-	D6	A2	F0	20	35	61	A4	76
60C8-	C8	F0	0C	A2	F4	20	35	61
60D0-	A5	76	A6	75	20	24	ED	A9
60D8-	0B	20	A7	60	4C	3C	D4	AD
60E0-	53	AA	C9	2B	F0	07	A9	EF
60E8-	A0	DC	4C	94	DB	A9	C6	A0

60F0- 62 4C 94 DB AD 53 AA C9
 60F8- 2B F0 07 A9 DF A0 DC 4C
 6100- DB DC A9 B6 A0 62 4C DB
 6108- DC AE 5C AA AD 53 AA C9
 6110- 2B F0 03 4C E9 A6 20 9C
 6118- 61 20 51 A8 A9 0B 20 2B
 6120- C3 20 5E A6 B0 0C A2 10
 6128- 20 9C 61 A6 75 A5 76 20
 6130- 24 ED 6C 5E 9D BD BO 61
 6138- 48 20 A7 60 E8 68 10 F5
 6140- 60 48 AD 53 AA C9 2B F0
 6148- 04 68 4C 31 D4 A2 00 BD
 6150- AD 62 48 20 A7 60 E8 68
 6158- 10 F5 4C C6 60 AD 00 C0
 6160- C9 83 F0 01 60 AD 53 AA
 6168- C9 2B F0 03 4C 60 D8 AD
 6170- 10 C0 4C 41 61 AD 80 C0
 6178- A9 6C 8D C5 9F A9 36 8D
 6180- C6 9F A9 00 8D C7 9F A9
 6188- AE 8D E6 A6 A9 53 8D E7
 6190- A6 A9 AA 8D E8 A6 20 7E
 6198- D2 4C BF 9D BD 8A 63 AA
 61A0- BD D0 62 48 20 2B C3 E8
 61A8- 68 10 F5 60 00 00 00
 61B0- 4E 45 58 54 7F 52 6A 46
 61B8- 4F 52 7F 32 6D 7F 44 E3
 61C0- 7F 3D 63 7F 35 A5 52 45
 61C8- 54 55 52 4E 7F 52 6A 47
 61D0- 4F 53 55 42 7F 32 6D 7F
 61D8- 44 E3 7F 47 2F 7F 49 7C
 61E0- 7F 3D 5C 19 99 7F 35 46
 61E8- 7F 35 25 7F 40 3E 19 19
 61F0- 19 19 19 19 99 7F 51 66
 61F8- 7F 31 75 19 99 7F 47 2F
 6200- 7F 48 58 7F 43 59 7F 32
 6208- 65 99 7F 4D 5D 7F 32 4A
 6210- 7F 52 6E 7F 3D 63 7F 4F
 6218- 4F 19 99 7F 4E 41 7F 2F
 6220- 69 7F 31 75 7F 3C 66 99
 6228- 7F 55 57 7F 48 67 7F 49
 6230- 7C 7F 56 68 99 7F 31 7C
 6238- 7F 49 7C 7F 4D 27 7F 40
 6240- 62 19 19 19 99 7F 40 1E
 6248- 7F 3B 32 7F 55 32 7F 4F.

6250- 4F 19 99 7F 49 7C 7F 3D
 6258- 5C 7F 3F 5F 7F 4F 4C 99
 6260- 7F 56 55 7F 35 7A 7F 32
 6268- 2B 7F 4B 28 7F 31 A0 7F
 6270- 2F 6C 7F 32 6E 7F 49 3C
 6278- 7F 4B 28 7F 36 32 7F 53
 6280- 52 99 7F 3B 4B 7F 4F 77
 6288- 7F 55 32 7F 4F 4F 19 99
 6290- 7F 4D 32 7F 34 25 7F 51
 6298- 64 7F 38 2D 7F 49 7C 99
 62A0- 7F 32 6C 87 2C 20 7F 4F
 62A8- 4F 7F 38 44 BA 0D 7F 55
 62B0- 4F 7F 34 4E 87 00 3F 20
 62B8- 7F 34 5F 7F 52 5F 7F 56
 62C0- 76 7F 35 4E 0D 00 3F 20
 62C8- 7F 55 57 7F 56 75 0D 00
 62D0- 7F 52 6E 7F 50 53 7F 30
 62D8- 39 7F 44 E3 7F 30 4D 7F
 62E0- 49 7C 7F 31 29 7F 3C E6
 62E8- 7F 4F 32 7F 2F 1F 7F 39
 62F0- A0 7F 49 7C 7F 3D 5C 7F
 62F8- 34 40 7F 4C E9 7F 4D 5D
 6300- 7F 32 4A 7F 4D 43 7F 3B
 6308- FD 7F 3D 6C 7F 38 44 7F
 6310- 30 39 7F 34 D3 7F 4D 5D
 6318- 7F 35 25 7F 34 40 7F 4F
 6320- B2 7F 32 44 7F 44 4B 7F
 6328- 51 50 7F 41 F9 7F 4D 43
 6330- 7F 3B 7D 7F 3B 52 7F 4A
 6338- F7 7F 52 6E 7F 35 25 7F
 6340- 32 6C 7F 4D F2 7F 39 38
 6348- 7F 31 64 7F 46 77 7F 30
 6350- 39 7F 37 B9 7F 4D 43 7F
 6358- 3B 7D 7F 3F 5F 7F 4F 4C
 6360- 7F 30 39 7F 34 D3 7F 31
 6368- 4B 7F 4F 71 7F 4B 28 7F
 6370- 31 A0 7F 30 39 7F 43 5B
 6378- 7F 55 2F 7F 3C 52 7F 52
 6380- C2 2C 20 7F 4F 4F 7F 38
 6388- 44 3A A0 00 0C 0C 18 21
 6390- 2D 39 45 51 5D 69 75 84
 6398- 96 A2 B1 00 00 00 00

下面介绍如何根据代码求保留字的方法:

1. 在 BASIC 状态下键入语句行:

10 A

2. 转入监控状态

CALL-151

3. 将代码 \$ 85 存入 \$ 805 单元

* 805:85

4. 返回 BASIC 状态后键入 LIST, 屏幕显示:

10 DEL

重复上述过程, 可列出各代码的对应保留字。

几点说明:

一、代码清单中所列的“(”、“=”、“:”等符号必须与相应的字符串连在一起, 才能被编译程序识别, 才能被认为是保留字。

二、ATN 在 TN 之间没有空格时才是保留字。如果 T、N 间有空格, 则把 AT 当作保留字。

三、TO 前面有字符 A, 且 T、O 之间有空格, 则把 AT 当作保留字。

低位 高位	0	1	2	3	4	5	6	7
8	END	FOR	NEXT	DATA	INPUT	DEL	DIM	READ
9	HGR2	HGR	HCOLOR=	H PLOT	DRAW	XDRAW	HTAB	HOME
A	COLOR=	POP	VTAB	HIMEM:	LOMEM:	ONERR	RESUME	RECALL
B	GOSUB	RETURN	REM	STOP	ON	WAIT	LOAD	SAVE
C	TABC	TO	FN	SPCC	THEN	AT	NOT	STEP
D	=	<	SGN	INT	ABS	USR	FRE	
E	TAN	ATN	PEEK	LEN	STR \$	VAL	ASC	CHR \$

低位 高位	8	9	A	B	C	D	E	F
8	GR	TEXT	PR#	IN#	CALL	PLOT	HLIN	VLIN
9	ROT	SCALE=	SHLOAD	TRACE	NOTRACE	NORMAL	INVERSE	FLASH
A	STORE	SPEED=	LET	GOTO	RUN	IF	RESTORE	&
B	DEF	POKE	PRINT	CONT	LIST	CLEAR	GET	NEW
C	+	-	×	/	^	AND	DR	>
D	POL	POS	SQR	RND	LOG	EXP	Cos	sin
E	LEFT \$	RIGHT \$	MID \$	MUSIC	PLAY	LG	GAME	空格

1.1.2 BASIC 和 DOS 出错信息的汉化

苏州第十中学 李竹君

中华学习机与其它各种有汉字系统支持的微型机一样, 其出错信息在中文状态下仍是英文的, 这对广大初学者 (大都是青少年学生和一般非技术人员等) 来说确是学习中的一个困难。另外, 作为有汉字系统的计算机, 提供汉化的出错信息也是理所当然的。

经过对 CEC-I 型中华学习机的 BASIC 解释程序和 DOS 中有关出错信息处理过程的分析, 笔者设计了一个中华学习机的支持软件——BASIC 和

DOS 出错信息的汉化处理程序。

其功能是: 当出现 BASIC 或 DOS 错误时, 在西文状态下仍显示原来的英文出错信息; 在中文状态下则显示相应的汉字出错信息, 并作两点改进。第一、显示汉字出错信息后, 清除该行信息以下的屏幕。第二、当程序执行中有 DOS 命令出错时, 将原来的两行英文出错信息 (实际占去三行) 合并为一行中文信息。

另外还重新设置了 & 和 RESET 向量, 键入 &

能回到全英文出错信息方式,按 CTRL-RESET
可进入本功能(第一次使用本功能要键入 CALL

24576或6000G)。

1.1.3 在 CEC-1 基本配置下运行 INTBASIC

张 斌

中华学习机(CEC)基本配置是指没有配备软
盘驱动器的系统。INTBASIC解释程序一般是存
放软盘上的,在做DOS引导时,由HELLO程序
装入16K RAM卡。在基本配置下运行
INTBASIC,必须解决INTBASIC解释程序的装
入问题。一般家庭都有盒式磁带录音机,本程序就
是借助磁带录音机,在没有驱动器的情况下运行
INTBASIC程序,具体步骤如下:

①找一台带有驱动器的CEC机,并且准备一
张含有INTBASIC文件的可引导DOS盘;

②按要求连接好录音机,并将磁带倒好,将带
长记数器复位;

③开机引导DOS盘,在监控状态下键入所附
程序,从\$810到\$885。然后从\$810开始反汇编
核对是否有误;

④用BLOAD INTBASIC, A \$1000命令装
入整型BASIC解释程序;

⑤CALL-151进入监控程序:键入下列命
令,先不忙回车: *810.4000W;

⑥同时按下录音机上的“RECORD”和
“PLAY”键,开始记录,待磁带转过前面的引带
后,按下回车键,并记下此时的磁带记数值;

⑦当光标重新出现时,记下此时的磁带记数
值;

⑧在一台没有驱动器的机器上连好录音机,倒
好磁带,并适当调节音量及音调,复位带长记数器;

⑨CALL-151进入监控,键入下列命令,不
忙回车: *810.4000R;

⑩按下放音键,等磁带记数器要到开始记录时
的数时,按下回车键;

⑪当光标重新出现时,键入下列命令并回车:
*810G↓此时INTBASIC解释程序已建立;

⑫CTRL-C退出监控;

以后只要用CALL 768就可进入
INTBASIC:在INTBASIC中用CALL 768就可
回到Applesoft BASIC。因为“INT”和“FP”均为
DOS命令,而此时DOS不存在,故不能用上述二
条命令。磁带一旦录好,以后只要从第⑧步开始就
可使用INTBASIC。

所附的程序完成三个工作,一是将
INTBASIC解释程序搬到RAM卡中,并设置写
保护;二是建立从768(\$300)开始的切换程
序;三是设立CTRL-RESET向量,使在
INTBASIC中按RESET键TB回到INTBASIC,
如果是在APPLESOFTBASIC中,则回到
APPLESOFTBASIC状态。此时的INTBASIC除
了不支持DOS命令外,其他与DOS状态下使用
没区别。

*810.885

```
0810- A9 00 85 06 85 08 A9 10
0818- 85 07 A9 D0 85 09 AD 83
0820- C0 AD 83 C0 A0 00 B1 06
0828- 91 08 C8 D0 F9 E6 07 E6
0830- 09 D0 F3 A9 16 8D F2 03
0838- A9 03 8D F3 03 20 6F FB
0840- A9 5D 85 06 A9 08 85 07
0848- A9 00 85 08 A9 03 85 09
0850- A0 29 B1 06 91 08 88 10
0858- F9 AD 81 C0 60 AD 00 D0
0860- 8D 29 03 C9 D6 D0 06 AD
0868- 80 C0 4C 00 E0 AD 81 C0
0870- 4C 00 E0 AD 29 03 C9 D6
0878- F0 06 AD 81 C0 4C 03 E0
0880- AD 80 C0 4C 03 E0
```

1.1.4 中华学习机单键输入命令

中国教育电子公司技术部 武光鼎

在IBM-PC之类的微机上,有一个ALT键。
使用ALT键可以使输入BASIC语句变得容易

些。使用者只要在按下ALT键的同时,再按下A
~X中的一个字母键,就可以把对应BASIC的一

个关键字输入到计算机里。在中华学习机的键盘中，没有 ALT 键。为了弥补学习机这一功能上的不足，我编了一个程序在中华学习机上用 ctrl 键去实现 PC 机上 ALT 键的功能。

〈表一〉给出了 ctrl 键与一个字母键同时按下所得到的 BASIC 关键字。

程序及使用说明如下：

首先，把源程序 EXBAS 在编辑软件下输入，通过汇编生成目标文件 EXBAS.OBJ0（见附一）。如果没有编辑和汇编软件，可以直接按给定的地址输入机器代码。当磁盘上已存入 EXBAS.OBJ0 后，只要键入 BRUNEXBAS.OBJ0↓就可以使中华学习机具备 ctrl 加字母键得到 BASIC 语句的功能了。使用时请注意以下几点：

①按 ctrl-Reset 键可以退出单键输入 BASIC 语句方式；按&↓可以再恢复这一功能。

②〈表一〉里未提供的字母是中华学习机内部使用的控制代码，这些控制码仍做原用。

③由于中华学习机的汉字系统占用了较多的控制码，为了不影响使用上的习惯，程序不在汉字系统下使用。

④对于无磁盘驱动器的学习机用户，可以用监控的 W 命令把程序录制到磁带上。录制前需把

地址 \$AA55 和 \$AA56 改为 \$0038 和 \$0039。通过监控的 R 命令把程序放到 \$8000 至 \$8096 处运行。

⑤按下字母 B 键，再按 ctrl-R 可以得到 BRUN 这个 DOS 命令。类似还可以得到 BLOSD、BSAVE 等。

表一

Ctrl	得到的关键字
A	CALL
B	RETURN
E	END
F	FOR
G	GOSUB
I	INPUT
L	LIST
N	NEXT
O	ONERR
P	PRINT
R	RUN
T	THEN
V	SAVE
W	LOAD
X	HOME
Y	PR#
Z	CATALOG

附录: EXBAS 的源文件与目标文件

SOURCE FILE: EXBAS

----- NEXT OBJECT FILE NAME IS EXBAS.OBJ0

```

8000:          1          ORG $8000
8000:          2  * * * * *
004E:          3  ACC      EQU $4E
004F:          4  YCC      EQU $4F
00EB:          5  INDEXL   EQU $EB
00EC:          6  INDEXH   EQU $EC
B000:          7  ;-----
8000:          8  ENTRY    EQU *
8000: A9 00          9          LDA #>ENTRY
8002: 8D F6 03      10         STA $3F6
8005: A9 80          11         LDA #<START
8007: 8D F7 03      12         STA $3F7
800A:          13  ;-----
800A: AD 56 AA      14         LDA $AA56
800D: C9 C3          15         CMP # $C3      HZKSW?
800F: F0 0E          16         BEQ LI

```

```

8011: C9 80      17          CMP #<START
8013: F0 0A      18          BEQ L1
8015:              19 ;-----
8015: A9 20      20          LDA #>START
8017: 8D 55 AA   21          STA $AA55
801A: A9 80      22          LDA #<START
801C: 8D 56 AA   23          STA #AA56
801F: 60         24 L1        STS
8020:              25 *****
8020: 20 1B FD   26 START    JSR $FD1B
8023: C9 9B      27          CMP #9B
8025: B0 F8      28          BCS L1
8027: 85 4E      29          STA ACC
8029: 84 4F      30          STY YCC
802B: 29 1F      31          AND #1F
802D: A8         32          TAY
802E: B9 62 80   33          LDA TABH-1,Y
8031: F0 21      34          BEQ RES
8033:              35 ;-----
8033:              36 ;
8033: 85 EC      37 SBASIC   STA INDEXH
8035: B9 7C 80   38          LDA TABL-1,Y
8038: 85 EB      39          STA INDEXL
803A: A0 00      40          LDY #0
803C: 8D 07 C0   41 LL1     STA $C007
803F: B1 EB      42          LDA (INDEXL),Y
8041: 8D 06 C0   43          STA $C006
8044: 30 06      44          BMI LL2
8046: 20 59 80   45          JSR DISP
8049: C8         46          INY
804A: D0 F0      47          BNE LL1
804C: 20 59 80   48 LL2     JSR DISP
804F: A4 24      49          LDY $24      CH
8051: A9 A0      50          LDA #A0
8053: 60         51          RTS
8054:              52 ;-----
8054: A5 4E      53 RES     LDA ACC
8056: A4 4F      54          LDY YCC
8058: 60         55          RTS
8059:              56 *****
8059: 09 80      57 DISP    ORA #80
805B: 9D 00 02   58          STA $200,X
805E: 20 F0 FD   59          JSR $FDF0
8061: E8         60          INX

```

```

8062: 60          61          RTS
8063:          62 ; _____
8063: C3 C3 00   63 TABH     DFB $ C3, $ C3, 0, 0 $ C3, $ C3
8066: 00 C3 C3
8069: C3 00 C3   64          DFB $ C3, 0, $ C3, 0, 0, $ C4
806C: 00 00 C4
806F: 00 C3 C3   65          DFB 0, $ C3, $ C3, $ C3, 0, $ C3
8072: C3 00 C3
8075: 00 C4 00   66          DFB 0, $ C4, 0, $ C3, $ C3, $ C3
8078: C3 C3 C3
807B: C3 A8      67          DFB $ C3, $ A8
807D:          68 ; _____
807D: 29 D9 00   69 TABL     DFB $ 29, $ D9, 0, 0, 0, 3
8080: 00 00 03
8083: D4 00 0E   70          DFB $ D4, 0, $ E, 0, 0, 4
8086: 00 00 04
8089: 00 06 A4   71          DFB 0, 6, $ A4, $ FB, 0, $ C7
808C: FB 00 C7
808F: 00 1F 00   72          DFB 0, $ 1F, 0, $ F0, $ EC, $ 59
8092: F0 EC 59
8095: 23 D2      73          DFB $ 23, $ D2
8097:          74 * * * * *

```

*** SUCCESSFUL ASSEMBLY: NO ERRORS

1.1.5 自己动手扩充 BASIC 语言

麦克罗

编制计算机辅助教学软件，往往希望有一些语句，可用来完成某些特殊任务。比如说，编制化学软件，最好有一些语句能画出大小不等，位置可变的烧瓶和烧杯，但是一般的程序设计语言并不提供此种功能。于是，我们就用 &（读作“ampersand”）命令来扩充自己的 BASIC 版本。

在 BASIC 程序中，一旦遇到以 & 作语句时，便转向 \$03F5 单元，因为 \$03F5 的邻近单元都有用途，不能随意动用。于是我们在 \$03F5~\$03F7 单元中，置入一条跳转指令 JMP，这样就可将系统引向我们期望的机器语言程序段去运行，只要该程序段末尾为一条 RTS 指令，则当程序段运行结束时，仍然返回原 BASIC 程序的下一条语句，就这一点而论，& 有点像 CALL 语句，它们都可以从 BASIC 程序，转向执行某一特定机器语言程序段，并在执行后返回原 BASIC 程序，所不同

的是，CALL 只能按指定的地址，调用机器语言程序，而 & 则可按程序名进行调用，其形式上更酷似 BASIC 语句。如：

```

&INK
&SET
&CIRCLE X,Y,R
&BOX X,Y,L

```

这些由 & 起头的调用命令，不妨称之为“& 类语句”。任何一个已定义的 & 类语句，都可象一般 BASIC 语句一样使用。

如果在 \$03F5~\$03F7 单元中置入指令 JMP YYY，而地址 YYY 只是某一功能块的起动脉址时，& 语句也就只能执行单一功能了。为了使 & 转向功能能够形成一系列扩充语句，就需要有一个“扩充语句解释程序”块，用来解释各种扩充语句，以便按语句要求，进入各自的功能块运行。解释程

序共有三部份组成:

(一) 解释程序主体

(二) &语句名字表

(三) 语句功能块入口表

现将解释程序的实例摘录如下:

(一) 解释程序主体程序

```
6000- A2 02      LDX  # $ 02
6002- BD 10 60   LDA  $ 6010.
6005- 9D F5 03   STA  $ 03F5.X
6008- 95 4F     STA  $ 4F.X
600A- CA        DEX
600B- 10 F5     BPL  $ 6002
600D- 20 8C F2   JSR  $ F28C
6010- 4C 13 60   JMP  $ 6013
6013- A2 00     LDX  # $ 00
6015- 86 09     STX  $ 09
6017- A0 00     LDY  # $ 00
6019- BD 48 60   LDA  $ 6048.X
601C- F0 1A     BEQ  $ 6038
601E- C9 FF     CMP  # $ FF
6020- F0 08     BEQ  $ 602A
6022- D1 B8     CMP  ($ B8).Y
6024- D0 07     BNE  $ 602D
6026- C8        INY
6027- E8        INX
6028- D0 EF     BNE  $ 6019
602A- 4C C9 DE   JMP  $ DEC9
602D- E8        INX
602E- BD 48 60   LDA  $ 6048.X
6031- D0 FA     BNE  $ 602D
6033- E8        INX
6034- E6 09     INC  $ 09
6036- D0 DF     BNE  $ 6017
```

```
6038- 20 98 D9   JSR  $ D998
603B- 06 09     ASL  $ 09
603D- A6 09     LDX  $ 09
603F- BD 75 60   LDA  $ 6075.X
6042- 48        PHA
6043- BD 74 60   LDA  $ 6074.X
6046- 48        PHA
6047- 60        RTS
```

(二) 语句名字表

```
6048- 42 4F 58 00 43 49 52 43
6050- 4C 45 00 54 52 49 41 00
6058- 53 45 54 00 54 52 59 00
6060- 49 4E 4B 00 41 43 4B 00
6068- FF
```

本名字表中共包含了七个语句名:

BOX (\$ 6048~ \$ 604B)

CIRCLE (\$ 604C~ \$ 6052)

TRIA (\$ 6053~ \$ 6057)

SET (\$ 6058~ \$ 605B)

TRY (\$ 605C~ \$ 605F)

INK (\$ 6060~ \$ 6063)

ACK (\$ 6064~ \$ 6067)

最后以“FF”作结束标志。

(三) 语句功能块入口地址表

入口地址表自 \$ 6074 单元开始, 所有入口地址均占用两个单元, 且第一单元置入口地址低位, 第二单元放入口地址高位。

有了“扩充语句解释程序”块, 我们便能根据需要扩充若干条语句, 但是, 这些语句还不能附带参数, 若要得到附带参数的扩充语句, 那还得增加“参数处理程序”块, 这样所得到的扩充语句比较完整, 也比较灵活, 适用范围广。

1.1.6 在中西文转换后怎样保证 DOS 正常工作

赵再平

中华学习机 CEC-I 型为我们提供了中文、西文两种应用环境。

但是, 在程序中使用 PR#3, TEXT 等命令进行中文字转换后, 由 CHR \$ (4) 引导的 DOS 命令均被作为普通的 PRINT 来处理了。是什么原因使 DOS 命令失效了呢?

原来, 系统在零页设置了两个 I/O 指针, 一

个是屏幕输出指针 CSW (\$ 36, \$ 37), 另一个是键盘输入指针 KSW (\$ 38, \$ 39), 在中西文转换中关键的一步就是: 将零页指针由指向西文 (或中文) 处理程序的入口变为指向中文 (或西文) 处理程序的入口 (如图 1-1 所示)。装入 DOS 后, 每个命令要先判断是否为 DOS 命令, 所以零页指针应指向 DOS 的 I/O 程序入口, 而 DOS 的两个相

应的 I/O 指针再指向西文 (或中文) 入口 (如图 1-2 所示)。可是, 在程序中使用 PR#3 等命令时, 系统软件只是像图 1-1 那样改动零页指针。这样, 所有命令都不经过 DOS, 使 DOS 命令失去了作用。

根据以上分析, 要保证中西文转换后 DOS 正常工作, 只须按附表提供的参数, 按图 2 所示正确

设置零页和 DOS 指针就可以了。下面给出中西文转换程序 A1 和 A2。如果在时间和空间上有较高要求则可使用程序 B1 和 B2 (用 CALL 调用)

附注: 每次进入汉字系统, HIMEM 指针被置为 \$ 9200, 如有必要可在程序 A1 末尾重新设置 HIMEM 指针。进入西文时, HIMEM 指针不受影响。

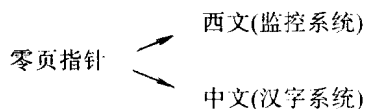


图 1-1

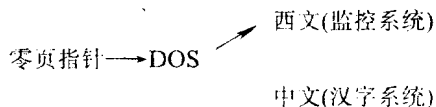


图 1-2

附表:

	零页指针	DOS		监控	汉字	
		入口	指针	系统入口	系统入口	
屏幕输出	CSW	\$ 36	\$ 9EBD	\$ AA53	\$ FDF0	\$ C32B
		\$ 37		\$ AA54		
键盘输入	KSW	\$ 38	\$ 9E81	\$ AA55	\$ FD1B	\$ C303
		\$ 39		\$ AA56		

```

程序 A1  功能: 西文→中文
200 CALL 49995
210 POKE 54,189: POKE 55,158: POKE
      56,129: POKE 57,158: POKE 43
      603,43: POKE 43604,195: POKE
      43605,3: POKE 43606,195: RETURN
程序 A2:  功能: 中文→西文
300 TEXT
310 POKE 54, 189: POKE 55,158: POKE
      56,129: POKE 57,158: POKE 43
      603,240: POKE 43604, 253: POKE
      43605,27: POKE 43606,253: RETURN

```

```

600F- 85 38      STA  $ 38
6011- A9 F0      LDA  # $ F0
6013- 8D 53 AA   STA  $ AA53
6016- A9 FD      LDA  # $ FD
6018- 8D 54 AA   STA  $ AA54
601B- 8D 56 AA   STA  $ AA56
601E- A9 1B      LDA  # $ 1B
6020- 8D 55 AA   STA  $ AA55
6023- 60         RTS

```

程序 B2 功能: 西文→中文

```

6030- 20 4B C3   JSR  $ C34B
6033- A9 BD      LDA  # $ BD
6035- 85 36      STA  $ 36
6037- A9 9E      LDA  # $ 9E
6039- 85 37      STA  $ 37
603B- 85 39      STA  $ 39
603D- A9 81      LDA  # $ 81
603F- 85 38      STA  $ 38
6041- A9 2B      LDA  # $ 2B
6043- 8D 53 AA   STA  $ AA53

```

```

*
程序 B1  功能: 中文→西文
6000- 20 3A C3   JSR  $ C33A
6003- A9 BD      LDA  # $ BD
6005- 85 36      STA  $ 36
6007- A9 9E      LDA  # $ 9E
6009- 85 37      STA  $ 37
600B- 85 39      STA  $ 39
600D- A9 81      LDA  # $ 81

```



```

6046- A9 C3      LDA  # $ C3
6048- 8D 54 AA   STA  $ AA54
604B- 8D 56 AA   STA  $ AA56
604E- A9 03      LDA  # $ 03
6050- 8D 55 AA   STA  $ AA55
6053- 60         RTS

```

1.1.7 如何为 CEC-I 机增加功能键

邵爱柏

中华学习机《CEC-I》的功能键虽然有所增加，但仍不够用。为方便操作，利用 DOS3.3 空区可为《CEC-I》再增加一些功能键。方法如下：

1. 启动 DOS 3.3 系统盘，将 DOS 引入内存。
2. 进入监控状态：]CALL-151 ↓
3. 修改 DOS 中第三页向量映象区中 \$ 3F5-\$ 3F7 内容：* 9E77: B3 B6 ↓
4. 在 DOS 空区加入下面有关 & 命令参数查询子程序：

```

B6B3- A0 00 B1 B8 48
B6B8- C8 20 98 D9 68 C9 43 D0
B6C0- 03 4C 6E A5 C9 48 D0 03
B6C8- 4C 7E D2 C9 49 D0 03 4C
B6D0- 77 F2 C9 4A D0 03 4C 69
B6D8- FF C9 4C D0 03 4C A5 D6
B6E0- C9 4E D0 03 4C 73 F2 C9
B6E8- 52 D0 03 4C 12 D9 C9 53
B6F0- D0 03 4C 5A D2 4C C9 DE

```

5. 返回 BASIC 状态：* CTRL-C ↓

6. 格式化一张空磁盘：]INIT HELLO ↓
此磁盘在启动后即具有功能键。

7. 功能键表及入口地址：

功能键	功能	入口地址
&C	CATALOG	\$ A56E
&H	HOME	\$ D27E
&I	INVERSE	\$ F277
&J	CALL-151	\$ FF69
&L	LIST	\$ D6A5
&N	NORMAL	\$ F273
&R	RUN	\$ D912
&S	SAVE	\$ D25A

8. 上述这些功能键亦可在 BASIC 程序中使用，如：10 &H: ? SPC(15) " CEC-I "

9. 若 & 命令在运行别的程序时被修改过，可在监控状态下键入：* 3F6: B3 B6 ↓或在程序中使用 POKE 1014, 179: POKE 1015, 182 以恢复 \$ 3F5 的向量。

10. 如果您想用多功能 CEC DOS 系统主盘请与本人联系。

1.1.8 CEC-I 机的音乐语句浅谈

黄石石 湖北师院附中 肖思明

CEC-I 型中华学习机为 APPLE-II 的向上兼容机。驻机解释系统软件有若干扩充，特别是在发声方面，增加了一条 BASIC 语句命令 MUSIC (保留字码为 \$EB)，为用户提供不少方便。

在此以前，所有 APPLE-II 及其兼容机，因为 BASIC 中无音乐语句命令，用户要进行音乐设计，必须自行设计机器语言发声子程序。最早出现的机器语言发声子程序如下：

程序一

```

1 $ 300- 存频率码 F
2 $ 301- 存音长码 T
3 $ 302- AC 00 03 LDY $ 0300

```

```

4 $ 305- AE 00 03 LDX $ 0300
5 $ 308- A9 04 LDA # $ 04
6 $ 30A- 20 A8 FC JSR $ FCA8
7 $ 30D- AD 30 C0 LDA $ C030
8 $ 310- E8 INX
9 $ 311- D0 FD BNE $ 0310
10 $ 313- 88 DEY
11 $ 314- D0 EF BNE $ 0305
12 $ 316- CE 01 03 DEC $ 0301
13 $ 319- D0 E7 BNE $ 0302
14 $ 31B- 60 RTS

```

由程序一可知，在单元时间内喇叭振动的次数与频率码 F 的值成正比。反映在机器语言程序上为

第 10、11 两行，Y 的初始值越大，则单元时间内的喇叭振动次数（即第 11 行执行非零跳转的次数）越多，相反在喇叭每两次振动之间的间歇时间相应缩短，这可以从第 8 行的增量指令看出。这符合声音的音调高低与单元时间内振动次数成正比的自然规律。

但此方案在音长码所反映的发音持续时间方面略有不足，即音长码 T 的单元时间值不是一个定值，而是随频率码的变化也有变化。这样原设计者只好用中频声音的持续时间作为标准，而不是以低频端或高频端发出的声音的持续时间作为标准，那样高低两端的差异要大一些。

为此出现第二方案如下：

程序二

```

1 $300— 存频率码 F
2 $301— 存音长码 T
3 $302— AD 30 C0 LDA $C030
4 $305— 88 DEY
5 $306— D0 05 BNE $030D
6 $308— CE 01 03 DEC $0301
7 $30B— F0 09 BEQ $0316
8 $30D— CA DEX
9 $30E— D0 F5 BNE $0305
10 $310— AE 00 03 LDX $0300
11 $313— 4C 02 03 JMP $0302
12 $316— 60 RTS

```

由程序二可知，在单元时间内喇叭振动的次数与频率码 F 的值成反比。反映在机器语言程序上为第 8、第 9 两行。第八行为减量指令，当初始值越大，则喇叭两次振动之间的间歇越长，那么单元时间内喇叭振动的次数便会减少，声音的音调变低。

出现此现象的原因是上述两汇编程序的结构略有差异。程序一的第 10、11 两行与程序二的第 8、9 两行，虽然指令结构形式完全一致，但外部非零跳转的目标地址却不一样，程序一的非零跳转使喇叭又振动一次，而程序二的非零跳转却使喇叭两次振动之间的间歇延长，所以频率码 F 的值与单元时间内喇叭振动的次数成反比。

第二种方案的最大变化，是将第一种方案的音长码 T 的单元时间值，与频率码 F 的联系切断了，这从程序中可见，第二种方案中 Y 寄存器的值，是与频率码 F 的值无关的。

但从总体上讲，由于 APPLE—II 及其兼容机

的硬件限制，机内无实时时钟，也只能比第一种方案略有改进而已。

CEC—I 机采用第二方案，并将其固化在 \$D0F9~\$D10B 的空间。摘录如下：

程序三

```

1 $D0EC— 20 F8 E6 JSR $E6F8
2 $D0EF— 86 EC STX $EC
3 $D0F1— 20 BE DE JSR $DEBE
4 $D0F4— 20 F8 E6 JSR $E6F8
5 $D0F7— 86 ED STX $ED
6 $D0F9— AD 30 C0 LDA $C030
7 $D0FC— 88 DEY
8 $D0FD— D0 04 BNE $D103
9 $D0FF— C6 ED DEC $ED
10 $D101— F0 08 BEQ $D10B
11 $D103— CA DEX
12 $D104— D0 F6 BNE $D0FC
13 $D106— A6 EC LDX $EC
14 $D108— 4C F9 D0 JMP $D0F9
15 $D10B— 60 RTS

```

CEC—I 机增设这一功能，为广大青少年学生初学 BASIC 语言者，提供了不少方便和乐趣，因为他们一般还不了解机器语言，这样做是非常合适的。

BASIC 中每一次功能扩充，都要增加内存开销。用 MUSIC 语句让机器发一个音，一般需要十个字节的内存开销。相反如果使用机器码，除了发声子程序占用二十个字节外，每发一个音只要两字节内存（存放频率码 F 和音长码 T）。由此可见，MUSIC 语句的内存开销一般是用机器语言发声子程序的五倍。当然作为中小学生的学习机，学生编制的程序不会太长，内存开销紧张的矛盾不会突出，所以用不着去考虑。

但作为教育工作者编制 CAI 软件，或有关音乐教学的软件，使用 BASIC 的 MUSIC 语句命令，不仅大规模的输入、修改不便，且内存开销过大的矛盾也将逐渐上升为主要矛盾。在这种情形下，建议仍使用机器语言发声子程序较好。

笔者曾利用 CEC—I 机 ROM 中 \$D0F9—\$D10B 的机器语言发声子程序运行机器码数据 (F、T) 序列，相当好用。唯一发现的不足是音调变高，节拍时间缩短了一些，为此我在程序三的第 11 行 X 变址寄存器的减量指令之前增加一条空操作指令，做到与使用 MUSIC 语句发声的音调基本