

M6800微型计算机资料之四

M6800 BASIC 解 释 程 序

参 考 手 册

广 州 市 自 动 控 制 研 究 所 译

前 言

我所引进美国MOTOROLA公司生产的M6800微型计算机发展系统资料一套，应有关单位要求，我们翻译了全文，进行内部交流。

这套资料共分十六册。其名称如下：

- 一、M6800 EXORICRM 200 用户指南。
- 二、M6800 连接装配程序参考手册。
- 三、M6800 驻留编辑程序参考手册。
- 四、M6800 EASTC解释程序参考手册。
- 五、M6800 汇编程序参考手册。
- 六、M6800 宏汇编程序参考手册。
- 七、M6800 驻留FORTRAN编译手册。
- 八、M6800 EXORCier User's 指南补充。
- 九、打印机接口模板用户指南
- 十、EXOR disk II 软磁盘控制器模板用户指南。
- 十一、M6800 磁盘操作系统预备用户指南。
- 十二、M6800 驻留MPL 编译参考手册。
- 十三、M6800 MM01A 单板计算机用户指南。
- 十四、M6800 微型机发展系统电路图集。
- 十五、用户系统评价器USER用户指南。
- 十六、系统分析器用户指南。

由于我们水平有限，加上时间仓促，虽然作了很大努力，但在翻译校正过程中还可能存在不少不妥和错误之处，恳请广大读者及时批评指正。

目 录

第一章 一般描述	1
1.1 引言	1
1.2 特征	1
1.3 程序结构	2
1.3.1 语句	2
1.3.2 命令	3
1.3.3 内部功能	3
1.4 数据格式	3
1.4.1 数	3
1.4.2 简单变量	4
1.4.3 条件变量	4
1.4.4 常量	4
1.5 程序的模式	4
1.5.1 指令的形式	5
1.5.2 执行的形式	5
第二章 程序的准备	6
2.1 源程序的输入	6
2.2 源程序的转换	7
第三章 语句	9
3.1 引言	9
3.2 输入/输出语句	9
3.2.1 输入语句	9
3.2.2 数据和读语句	10

3.2.3	恢复语句	12
3.2.4	打印语句 (直接)	13
3.3	说明语句	14
3.3.1	数组语句	14
3.4	控制语句	15
3.4.1	FOR 和 NEXT 语句	15
3.4.2	停止语句 (stop)	16
3.4.3	结束语句 (end)	16
3.4.4	GOTO 语句 (直接)	17
3.4.5	GOSUB 语句	17
3.4.6	返回语句 (return)	18
3.4.7	ON 语句	18
3.4.8	IF 语句 (BASIC 语句——直接)	19
3.5	赋值语句	20
3.5.1	LET 语句 (直接)	20
3.5.2	数学运算符的语句	21
3.5.3	运算优先语句	21
3.5.4	字符串联系语句	22
3.6	注释语句	23
3.6.1	REM 语句	24

第四章	指令	25
4.1	引言	25
4.2	系统指令	25
4.2.1	输入指令	25
4.2.2	注释指令	25
4.2.3	保存指令	25
4.2.4	EXIT 指令 (退出指令)	25
4.2.5	POKE 指令 (指向指令)	26

4.2.6	LIST 指令	26
4.2.7	跟踪指令	27
4.2.8	跟踪断开指令	27
4.3	执行程序指令	27
4.3.1	RUN 指令	28
4.3.2	CONT 指令	28
4.4	编辑指令	28
4.4.1	控制 C 指令	28
4.4.2	控制 X 指令	28
4.4.3	控制 O 指令	29
4.4.4	修改指令	29
4.5	校正指令	29
4.5.1	LINE 指令	30
4.5.2	DIGITS 指令	30

第五章	内部功能	31
5.1	引言	31
5.2	控制功能	31
5.2.1	TAB 功能	31
5.2.2	POS 功能	31
5.3	数据功能	32
5.3.1	RND 功能	32
5.3.2	INT 功能	32
5.3.3	ABS 功能	33
5.3.4	SGN 功能	33
5.3.5	LEN 功能	33
5.3.6	ASC 功能	34
5.3.7	CHR\$ 功能	34
5.3.8	VAL 功能	34

5.3.9	STR\$功能	35
5.3.10	LEFT\$功能	35
5.3.11	RIGHT\$功能	35
5.3.12	MID\$功能	36
5.3.13	PEEK 功能	36
5.4	数学函数	36
5.5	用户定义功能	37
5.5.1	DEF 功能	37
第六章	MDOS BASIC	39
6.1	引言	39
6.2	系统的要求	39
6.3	输入 BASIC 解释程序	39
6.4	指令的描述	39
6.5	专用控制符	40
6.6	磁盘文件的输出/输入	40
6.6.1	指令的描述 (OPEN 开)	40
6.6.2	指令的描述 (CLOSE 关)	41
6.6.3	指令的描述 (RESTORE 恢复)	44
6.6.4	输入文件	44
6.6.5	数据资料	45
6.6.6	输出文件	45
6.7	错误信息	47
附录A	一般转换表	48
附录B	ASCII 码的转换表	50
附录C	EXBUG 指令	51
附录D	程序实例	53
附录E	错误信息	63

附录F	增强程序的运算	64
附录G	致力使内存占用减至最小	65
附录H	BASIC 解释程序摘要	

第一章 一般描述

1.1 引言

EXORCiser 驻留 BASIC 解释程序为 M6800 微型计算机族提供了一个解决问题的新工具。BASIC 是一种广泛应用于一般场合和某些与商业有关的应用场合的高水平编程语言。系统由 BASIC 软件和如下硬件组成：

- EXORCiser
- 8K 字节的内存。
- TTY 或与 RS-232C 兼容的匣式磁带的终端。
- EXOR tape 或 EXOR disk 驱动单元（如果没有选择匣式磁带时）。

最小结构可扩展到包括 65,536 字节的内存（其中要减去为 EXbug 固件程序保留的 4K 字节）和可扩展到 4 个磁盘总数超过 1 百万字节的磁盘存储能力。此外，各种各样的硬拷贝——行印机及 CRT 可以被使用。在本手册中，已假定终端设备具有键盘和行印机，如果你使用 CRT 代替行印机，字“终端打印机”将被字“CRT 显示”所代替。

1.2 特征

BASIC 解释程序的特征是：

- 所有数学的运算用 DCD 码（二进制码，Binary coded Decimal）来执行，以达到最大的精确度。
- 用户程序可以由磁带，纸带或软磁盘保存及装入。
- 大多数算术函数和超越函数用直接执行子程序来实现。
- 允许使用字符串变量和数组。
- 大多数程序语句可用直接方式来执行（对于立即计算机不要求

语句标号)。

- BASIC 程序约需 7,600 字节内存, 从而限制了采用最小结构 (8K) 时, 可以执行的用户程序的大小, 当然, 如果用户程序需要, 也可以将内存扩展。
- 用户可用 USER 功能来调用机器语言程序。

1.3 程序结构

用 BASIC 编写的用户程序由各种语句, 指令及内部函数组成, 下列各节对每一部分作简单描述, 而各节中所提及的各章将在后面作深入的描述。

1.3.1 语 句

每一语句的开始应具有一个语句标号, 后面是语句指令 (或名) 和语句变量, 并由一个 CR 来结束。在 BASIC 中, 使用五种类型的语句:

输入/输出, 说明 (继语句)、控制、赋值和注释语句。这些语句在第三章描述, 下面的规则适用于全部 BASIC 语句。

- 每一语句必须有一语句标号, 范围从 1 ~ 9999, 不得使用第零行 (0000 这一行)。
- 语句标号是用来按顺序调用程序语句的。
- 在任何程序中, 一个语句标号仅能使用一次。
- 如要修改先前输入的一行 (语句) 可在重新写入原来的语句标号的后面, 写入所需的新的语句, 或输入一个语句标号, 紧接着输入一个 (CR), 即消除先前的行。
- 语句不必按照标号顺序输入, 因为 BASIC 程序将自动地按照标号上升顺序排列程序。
- 一个语句不可包含多于 72 个字符 (包括空格)。
- 空格, 除非在一字符串和在引号里面使用, 一般不作处理。

这样空格的使用是任意的, 它提高可读性 (清晰度), 但需

较长时间去处理。数必须不包含嵌入空格。在这本手册中提供的所有例子，都全部包括了空格。

例: 110 LET A = B + (3.5 * 5E2)

完全等于: 110 LET A = B + (3.5 * 5E2)

1.3.2 命令

命令(得当作命令使用的语句)是一些语句，它被用户从终端键盘拨入时，立即由 BASIC 程序执行。所有命令的拨入不需要语句标号。所有在 BASIC 程序中使用的命令在第4章描述。

1.3.3 内部函数

内部函数是程序函数，是 BASIC 程序本身固有的，在执行时，不要求一个单独的程序。为用户提供4种类型的内部函数：控制、数据、数学和用户定义的。这些内部函数在第五章描述。

1.4 数据格式

在使用 BASIC 程序时，所有数据输入必须按照下面各级的格式。

1.4.1 数

在 BASIC 中能够表示的数的范围是： $1.0E - 99$ 到 $9.99999999E + 99$ 。这里“E”代表指数。这样， $E - 99 = 10^{-99}$ 和 $E + 99 = 10^{+99}$ 。所有输入的数超出这些极限时，就要进行内部舍取(最后位丢掉)保留到9位有效位。数的输入和显示可以是三种格式的一种：整数、小数、

和指数。

例: 153 整数

153.34 小数

153E-2 指数

1.4.2 简单变量

简单变量可被任何单字母字符或任何单字母字符后面跟着一个数(0 ~ 9)来命名。

例: A, B, C 等

A3, B4, C5 等

1.4.3 字符串变量

字符串变量由任何单字母字符或下标字母字符跟着一个“\$”符号表示。一个字符串变量最多可包含 18 个字符, 并包括空格。即使字符串变量中仅包含一个字符, 也将保留 18 个字节。字符串中的每一个字符用 ASCII 码表示。

例: A\$, B\$, C\$ 等

A\$(3), B\$(4), C\$(5) 等

1.4.4 常数

一个常数(即一个不变的量)由独立的数字组成, 有一插入的小数点, 或者小数点在前面, 或者跟在后面。

例: 153 34.52 .554 136E-2

1.5 程序方式

在输入及执行 BASIC 程序时, 使用两种操作方式: 命令方式和执行方式

1.5.1 命令方式

当系统已准备好等待用户输入以后的程序语句或命令时, 系统按命令方式操作。系统借助于打印一个井号符作为提示符来向用户指示现在用命令方式。

1.5.2 执行方式

在执行程序时, 系统按执行方式操作。

第二章 程序的准备

2.1 源程序的输入

BASIC 源程序在下面三种存储设备中都能得到相应的 BASIC 源程序：匣式磁带、纸带或磁盘。从存储设备（外存）把源程序输入到内存中，牵涉到一个会话式的过程，即由装入子程序在终端（打印机或 CRT）上提供一些提示符，并由操作者通过终端键盘来回答这些提示。

将 BASIC 解释程序以外存装入到内存要求的步骤，取决于使用外存的类型，如果 BASIC 解释程序存储在匣式磁带或纸带中，则程序装入由 EXBUG MAID 子程序来控制。因此，用户进入 MAID 子程序以后，他必须进入内存单元 100 去装入 BASIC 解释程序（100；G）。然而，如果 BASIC 解释程序被存储在磁盘上，则为了输入程序，用户必须输入与 BASIC（原文为 EDIT，有误），随后是旧文件名（如果需要的话）和新文件名。这一过程在例 B 中详细讨论。BASIC 解释程序装入以后，字“READY”在终端（CRT 或打印机）中显示并跟着在下一行开始的位置上显示一“#”字（参看后面的例子）。这指明程序准备好，可以从终端键盘接受输入。连续程序的每一语句输入以后，系统将印出一个“#”符号作为对用户的提示。这指明语句已输入完毕。在每一命令之后，字“READY”将再次被印出，随后是一个“#”，这就指明命令已被执行，而且系统已准备好去输入下一程序语句或命令。

例 A 展示出一会话的过程，它用于将源程序从匣式磁带或纸带中装入内存。例 B 示出将源程序从磁盘中装入内存的过程。在此两例中，正确的操作表示在一杠横线上。

例 A （仅适用于匣式磁带或纸带）：

```

EXBUG X.X LOAD
SGL/CONT S
BASIC 8
ESBUG X.X MAID
* 100 ; G
READY

```

注：“#”号指明BASIC源程序已完全装入并准备好从终端键盘接受输入。

例B (仅适用于磁盘)

```

= BASIC , OF , NF
M6800 RESIDENT BASIC 1.0
COPYRIGHT (C) - 1976
READY
#

```

注：OF = 用户希望用旧文件名称去输入。

NF = 用户希望建立的新文件名称，假如用户仅输入 BASIC, NF, 一个新文件名被建立。但是，假如用户在开始时输入串：“BASIC, (原文为 EDIT, 有误) OF, NF”，则旧文件被装入并更新。或者在以新文件名称保存起来之后予以修改。

= 指明 BASIC 程序和旧文件已完全装入好且 BASIC 已准备好，可接受输入。

2.2. 源程序的修改

BASIC 解释程序装入内存后，内存工作地址范围自动地由程序来决定，系统现在准备好从终端键盘接受指令或语句。例

如，用户可能希望输入下面程序：

```
READY
#10 REM DEMONSTRATION
#20 PRINT "ENTER A NUMBER"
#30 INPUT R
#40 LET A = (R * R) * 3.1415926
#50 PRINT
#60 PRINT "THE AREA OF A CIRCLE WITH";
#70 PRINT "RADIUS"; R; "IS"; A
```

如果用户希望在两个语句之间插入一语句，他可打印一个语句标号，其值应在该两语句标号之间，随后即打入他所希望插入的语句。在语句完全输入后，用户输入一个 (CR) 去完成这一插入。下面示出这种插入语句的例子。

```
#45 REM THIS INSERTED BETWEEN 40 AND 50
```

如果用户想替换一个语句，必须先印出被替换的语句标号，随后是新的语句和一个 (CR)。

例如：

```
#50 PRINT "A="; A
```

每一行输入后，必须用一个 (CR) 来结束。当一个 (CR) 被认可时，BASIC 程序自动地使打印机前进到下一行。

当 CTRL (Control Key) 和 O 或 X 键被按下后 (首先按住 CTRL 键，随后按 O 或 X 键)，程序将使每按一下 O 键，向后空一个字符，而按 X 键，即消除一行。更进一步的说明在第 4 章。

如果用户现在准备好去执行程序，此时，在终端键盘上输入 RUN 命令。

第三章 语句

3.1 引言

所有 BASIC 程序语句由一个符号或一组符号代表一个变量或一组变量，由代表标符的符号，按照由 BASIC 解释程序确定的一套定义和规则结合起来。BASIC 解释程序采用 5 种语句：I/O，说明，控制，赋值和注释。在输入时，这些语句被保存在内存中，但并不执行，直到一个执行命令由用户输入（对各种命令请参考第 4 章）为止。但是，当 BASIC 程序是在命令方式时（等待用户的输入），一些没有语句标号的语句可被输入，BASIC 程序即马上执行这些语句，这就称为直接执行方式。由于以直接方式输入的语句，通常属命令性质，所以用直接执行的方式，用户可以得到问题的直接的解答。而无需重复程序的步骤。这样，由于使用直接可执行语句，BASIC 解释程序可以非常类似操作电子计算器那样的方式使用。在下面各节中，所有的 BASIC 解释程序的语句都被列出，并分类说明。只有那些标明是“直接”的语句，可以在直接方式中使用。

3.2 INPUT/OUTPUT 语句

I/O 语句是用以控制 EXERCISER 和 I/O 设备之间的数据流动的，下面各节列出并解释每一个 I/O 语句。

3.2.1 Input 语句

在执行程序时，INPUT 语句允许用户从终端键盘输入数据。例 A 代表两种 INPUT 语句。

例 A: INPUT X 表示输入一个数值 (X)。

 INPUT X# 表示输入一串符号变量 (X#)

当程序中出现一个 INPUT 语句时,即在终端 CRT 上印出一个(?)符号,为响应这个提示符,用户应立即打入所要求的数据(用“.”分隔),随后是一个(CR),为了节省 INPUT 语句,常数和字符串也能被输入。如果用户输入的数据不够,系统将再次用提示符“?”作提示。如果没有数据输入或者将非数值误作为变量输入时,系统将会写出“RE — ENTER”字作为提示。例 B 介绍使用 INPUT 语句输入简单变量和字符串变量的方法。必须注意,每个变量由一个“.”号分隔开。

例 B:

```
INPUT X, Y, Z, A$
```

可以多重输入,如果要求的数值没有输入,系统将立即给出另一个“?”号来。

```
INPUT "ENTER VALUE", X
```

印出包含在引号里面的信息,作为给用户的一个提示。一个“?”号则直接地在信息后面印出。系统等待输入的值。此值被存储贮在变量 X 之中。

3.2.2 DATA 和 READ 语句

DATA 和 READ 语句在一起使用,提供赋值给变量的一种方法(使用 LET 语句是另一种方法)。

DATA 语句顺序地将在自变量段紧接着 DATA 指令的数据段中所规定的每一数值装入到下一个可用的数据缓冲器中去。在自变量段里的每一数值顺序由一“.”号分隔开,如果字符串变量在自变量段里面使用,它们不需要括在引号里,因为“.”号可作为分隔符。但是字符串里面已使用一个“.”号,则字符串变量必须括在引号内。所有 DATA 语句,不管在程序中什么地方出现,都使 DATA 自变量组合成一清单。(而如果