

微型机测量控制系统 实用程序设计



颜用先 编著

人民邮电出版社

微型机测量控制系统 实用程序设计

颜用先 编著

人民邮电出版社

内 容 简 介

本书提供了用 Z 80 汇编语言编写的 130 个微型机测量控制系统常用程序, 包括定点数算术运算、浮点数算术运算、开平方、常用基本函数、代码转换、排序查找、数据的采集和处理、控制程序、管理程序。还介绍了提高程序可靠性的具体措施。读者可直接引用这些经过检验的实用程序以加速完成自己的微型机应用的开发任务。同时, 书中还结合这些程序比较详细地介绍了微型机测量控制程序的实用设计方法和设计技巧, 有助于提高读者的程序设计水平。

本书主要是为从事开发微型机测量控制应用的科技人员编写的, 也可供大专院校计算机应用、自动控制等专业的师生参考。

微型机测量控制系统实用程序设计

颜用先 编著

责任编辑: 徐修存

*

人民邮电出版社出版

北京东长安街27号

北京广益印刷厂印刷

新华书店北京发行所发行

各地新华书店经售

*

开本: 850×1168 1/32 1988年11月第一版

印张: 19¹⁰/₃₂ 页数: 312 1988年11月北京第1次印刷

字数: 517千字 印数: 1—4,000册

ISBN 7-115-03687-X/TP·014

定价: 8.00元

前 言

微型机用于工业生产过程的测量和控制主要有两个难点，一是接口电路的设计，另一是程序设计。由于各个系列的微型机都有一批可编程的大规模集成电路外围接口片可供选用，因而使接口电路的设计得以简化。但是程序设计由于涉及的面很广，处理的内容繁杂多样，目前还无统一的工程设计方法。如何才能掌握程序设计的方法？如何才能提高程序设计的技巧？怎样才能较快地完成自己的微型机应用程序设计？怎样才能使自己的程序设计得更加高效可靠？这些都是当前正在进行微型机测量控制应用开发的广大科技人员极其关心和迫切希望解决的问题。

如何才能较快地掌握程序设计的方法和提高程序设计的技巧？国内外的经验表明，除了学习程序设计的基本原理外，一个较好的方法就是熟读并模仿好的程序。据报道，国外许多大企业在培养自己新的程序员时，开始阶段就是采用这种称之为“软件文学”的方法。选择一些典型的程序，反复阅读，仔细研究，认真琢磨，搞清楚其设计方法，提炼出其设计技巧，以便从中得到启示，开阔思路。

怎样才能较快地完成自己的微型机应用程序的设计，并使其更加高效可靠？除了尽快提高自己的程序设计水平外，还有一个非常有用的方法，那就是借用已有的经过检验的程序。根据所要完成的程序设计任务，尽可能寻找合适的现成的程序模块（有的可能要做局部修改），实在找不到的才自己编写，然后将这些程序模块有机地组合起来，即可得到所需要的程序。这就是目前常采用的“组装程序”的设计方法。采用这种方法不仅能较快地完成程序设计的任务，而且所编制出的应用程序也更高效可靠。显然尽量利用现有的经过检验的程序模块，是迅速可靠地完成程序设计任务的一条捷径。

为了帮助微型机测量控制应用的初学者尽快地实现上述目的，作者在搜集国内外大量资料的基础上，结合自己的科研及教学实践，考虑了选材的典型性、先进性和实用性，精选出了这 130 个微型机测量控制系统常用程序，编写了这本书。和别的讲述程序设计的书相比，本书有两个重要的特点：一是本书的选材直接来自微型机测量控制的应用实践，它们描述了一些典型环节的程序实现，具有很强的实用性和一定的通用性，读者可象利用具有一定功能的集成电路块来设计各种实用电路那样，利用这些程序模块来设计自己的应用程序，从而有助于加速完成自己的微机应用开发任务；二是本书在编写过程中比较详细地讲述了整个程序设计过程，不仅讲清楚了设计思想，用流程图勾画出程序的基本结构，而且通过程序文本上的详细注释描述了程序设计的细节，同时还分析了各程序的设计特色，指出了所采用的设计技巧。这不仅便于读者正确理解和使用各程序，而且可使读者能举一反三，开阔思路，掌握更多的设计方法和设计技巧，有助于尽快提高自己的程序设计水平。

本书第一章为概述。第二章介绍一些基本的程序结构，重点讨论子程序和循环程序。第三章给出包括定点算术运算、浮点算术运算、开平方及基本函数计算、代码转换、排序查找等常用程序。为了便于程序设计的比较以及适合于不同应用场合的需要，同一种运算，有的选用了不止一个程序。同时考虑到过程控制一般对实时性要求较高，所以还提供了一些新颖的高效快速的运算程序。第四章选择了数据采集和处理时的一些常用程序，象模数转换、数字滤波、标度变换及等精度测量数据处理等程序都是非常实用的。第五章介绍控制程序设计中一些典型环节的实现方法，象多开关量的输出控制，多工作状态的转换控制，多中断控制等，这些程序不仅很有实用价值，而且具有一定的设计技巧。第六章结合微型机小型实时控制系统的特点，介绍其管理系统的设计。读者从中不仅可以获得有关计算机操作系统的一些基本知识，而且可以获得一个具有特色的完整实用的管理系统的程序设计。第七章比较全面地介绍了一般程

序设计书中很少讲到的提高程序可靠性的常用的具体措施，这对运行环境比较恶劣的微型机测量控制系统的程序设计是很有实用价值的。

本书承蒙大连理工大学计算机科学及工程系李克春副教授仔细审阅，提出了许多宝贵的意见，对提高本书的质量有很大帮助。大连组合机床研究所黄跃进、张永利同志协助调试了书中的大部分程序。另外，牛铭琦同志誊写了全部书稿，描绘了全部插图。在此谨向他们表示衷心的感谢！

由于作者水平有限，书中错误和不妥之处在所难免，敬请读者批评指正。

作者
于大连大学

目 录

第一章 概述	1
1.1 Z 80 汇编语言的程序约定	1
1.2 流程图	5
1.3 关于本书使用的几点说明	6
第二章 常用程序结构	9
2.1 堆栈	9
2.2 子程序	11
2.2.1 子程序的结构	11
2.2.2 子程序的调用	12
2.2.3 子程序的参数传递	13
2.2.4 子程序的嵌套	18
2.2.5 子程序的浮动性和可重入性	19
2.2.6 子程序的文本格式	20
2.2.7 子程序的使用可改进程序结构,提高程序 的可读性	21
2.3 分支程序	21
2.3.1 单路分支	22
2.3.2 多路分支	22
2.3.3 开关控制的分支	23
2.4 循环程序	25
2.4.1 循环程序的构成	25
2.4.2 循环的控制方法	27
2.4.3 多重循环	33
2.5 队列	36

2.5.1	队列概述	36
2.5.2	队列的入队和出队操作	39
2.5.3	队列的快速存取	42
2.5.4	顺序更新数据的队列	46
2.6	可浮动使用的 Z 80 程序的设计方法	48
2.6.1	设计可浮动使用程序的必要性	48
2.6.2	限制 Z 80 程序浮动的原因	49
2.6.3	可使 Z 80 程序浮动的设计方法	50
2.7	汇编语言程序设计应遵循的一些原则	53
2.7.1	汇编语言程序设计的一般方法和技巧	54
2.7.2	Z 80 汇编语言程序设计的一些技巧	60
第三章	常用基本程序	65
3.1	几个常用程序	65
3.1.1	数的比较	65
3.1.2	检查多字节数是否为零	72
3.1.3	多字节数加 1	73
3.1.4	多字节数的移位	74
3.1.5	求补运算	76
3.1.6	字节的分解和组合	78
3.1.7	清零内存区	82
3.2	算术运算	83
3.2.1	定点数算术运算	83
3.2.1.1	二进制定点运算程序	85
3.2.1.2	十进制定点运算程序	114
3.2.2	浮点数算术运算	137
3.2.2.1	浮点运算的服务程序	141
3.2.2.2	浮点算术运算程序	150
3.3	开平方运算	164
3.3.1	求 16 位整数的平方根	164

3.3.2	32 位二进制数快速开平方	166
3.3.3	对纯小数($0 < x < 1$)快速开平方	171
3.3.4	浮点数开平方	175
3.4	常用基本函数	186
3.4.1	三角函数	187
3.4.1.1	浮点数正弦函数程序	187
3.4.1.2	浮点数余弦函数程序	200
3.4.1.3	正弦函数和余弦函数快速简易计算程序 ...	201
3.4.1.4	浮点数正切函数程序	205
3.4.1.5	快速简易计算正切函数程序	207
3.4.2	指数函数	210
3.4.3	对数函数	222
3.4.3.1	浮点数对数函数程序	222
3.4.3.2	快速计算二进制数的常用对数程序	233
3.5	代码转换	238
3.5.1	J 进制数与 ASCII 码的相互转换	238
3.5.1.1	J 进制数转换为 ASCII 码	238
3.5.1.2	ASCII 码转换为 J 进制数	248
3.5.2	二进制数与十进制数(BCD 码)的相互转换	256
3.5.2.1	二进制数转换为十进制数	257
3.5.2.2	十进制数转换为二进制数	270
3.5.3	二进制码与格雷码的相互转换	282
3.5.3.1	二进制码转换为格雷码	283
3.5.3.2	格雷码转换为二进制码	286
3.6	查找和排序	288
3.6.1	查找	289
3.6.1.1	顺序查找法	289
3.6.1.2	对分查找法	293
3.6.2	排序	297

3.6.2.1	交换选择排序法	298
3.6.2.2	Shell 快速排序法	301
第四章	数据采集及处理程序	308
4.1	开关量采集及处理的通用方法	308
4.1.1	硬件结构	309
4.1.2	软件设计	311
4.1.3	程序设计分析	319
4.1.4	使用说明	321
4.2	模拟量的采集	322
4.2.1	典型的模拟量采集通道	322
4.2.2	模拟量通道的数据预处理	331
4.2.3	用软件方法提高模拟量通道的抗干扰能力	335
4.3	时间量和频率量的测定	337
4.3.1	时间量的测定	339
4.3.2	频率量的测定	342
4.4	数字滤波	348
4.4.1	剔除不合理的采样值	349
4.4.2	取中值滤波法	350
4.4.3	RC 低通数字滤波	351
4.4.4	移动平均滤波	355
4.5	标度变换	358
4.5.1	传感器的输出和被测量是线性关系时的 标度变换	358
4.5.2	传感器的输出与被测量是非线性关系时 的标度变换	361
4.6	几种数据处理程序	366
4.6.1	寻找最大值与最小值	366
4.6.2	倒数运算	368
4.6.3	越限报警	370

4.6.4	数字微分	375
4.6.5	数字积分	379
4.6.6	线性插值	383
4.7	等精度直接测量的数据处理	388
4.7.1	算术平均值的计算	389
4.7.2	剩余误差和均方根误差的计算	391
4.7.3	剔除可疑测量值	397
4.7.4	等精度直接测量列测量结果的数据处理	401
4.8	键盘及显示器	403
4.8.1	将数字键转化为功能键使用	403
4.8.2	简单实用的计数显示程序	412
4.8.3	单板机通用显示程序	414
第五章	控制程序设计	417
5.1	开关量输出控制	417
5.2	模拟量输出控制	426
5.2.1	8 位 D/A 转换电路	426
5.2.2	用两片 8 位 D/A 芯片组成的 16 位 D/A 转换电路	431
5.2.3	任意波形发生器	434
5.3	多工作状态转换控制	441
5.4	顺序控制	446
5.5	多中断控制	453
5.5.1	设计思想	453
5.5.2	实用程序	462
5.5.3	注意事项	465
5.6	程控衰减、放大器	467
5.7	微型机交通灯控制器	474
5.7.1	控制策略	475
5.7.2	硬件电路	477

5.7.3 程序设计	479
第六章 微型机小型实时控制系统的管理系统设计	486
6.1 引言	486
6.2 管理系统的设计	489
6.2.1 处理机管理	489
6.2.2 外设管理	500
6.2.3 中断处理	503
6.2.4 初始化程序	514
6.3 实用程序	515
6.3.1 任务管理程序	516
6.3.2 中断服务程序	519
6.3.3 实时时钟管理程序	522
第七章 提高程序可靠性的具体措施	532
7.1 使程序及数表固化	534
7.2 奇偶校验	535
7.3 格式检错	537
7.4 对重要数据进行程序复核	541
7.5 程序卷回以清除瞬间性故障	542
7.6 设立软件陷阱防止程序失控	543
7.7 采用软件监控使程序具有自恢复的能力	545
7.8 清除非法数据	550
7.9 输出状态的软件保护	551
7.10 RAM 和 EPROM 的实时测试	553
附录一 程序快速检索表	564
附录二 Z 80 实用设计手册	570

第一章 概 述

微型机测量控制系统应用程序的设计虽然也可采用高级语言，但就目前来说还是以采用汇编语言为主。这是因为同高级语言相比，汇编语言有以下优点：

- (1) 得到的目的程序较短，可节省内存容量。
- (2) 目的程序执行速度快，并能准确地掌握执行的时间。
- (3) 便于管理接口电路等，可直接调动计算机的全部资源。
- (4) 不象高级语言那样需要较多的硬件及软件的支持。

因此，在资源有限、实时性要求较高的微型机(特别是单板机)测量控制系统的程序设计中，主要使用汇编语言。

在我国，当前研制微型机测量控制系统主要采用 8 位微型机。据有关资料统计，Z80 系列占目前国内 8 位机的一半以上，特别是在单板机中更占绝对优势。所以本书介绍的微型机测量控制系统实用程序一律采用 Z80 汇编语言给出。另外，为了帮助读者更好地理解程序的设计思想，我们采用流程图的方法来描述程序的基本结构，为此，我们在这一章简要地介绍一下这两个问题。

1.1 Z80 汇编语言的程序约定

一、汇编语言的格式和规则

在汇编语言中，指令用助记符表达式来表示，助记符表达式是写指令的简单而有效的方法。本书中使用的助记符都严格地按照 Zilog 公司的规定。

汇编语言程序由语句组成，每个语句占一行，表示一条 Z80 指

令。行的长度通常取决于输入设备上行的长度（如键盘显示终端一行通常是 80 列）。汇编程序可识别的行结束标记一般是回车、换行代码。本书采用的 Z 80 汇编语言的格式如图 1-1 所示。

标号	操作码	操作数	注释
SUM	LD	A, 0	; 清 0 存和寄存器
	ADD	A, (HL)	; 加一个加数
	INC	HL	; 指向下一个加数
	⋮		
	⋮		

图 1-1 典型的 Z 80 汇编语言格式

由上图可见，Z 80 汇编语言的指令可由四部分组成：标号段、操作码段、操作数段和注释段。各段之间用空格、分号等分界符分开。

1. 标号段

标号一般用于表示地址，但也可以表示数据。使用标号编写汇编语言程序会更加方便，而且会减少发生错误的机会。标号是任选的，可由 1 至 6 个字符的字符串组成，其中第一个字符必需是字母。字符串中间不能有分界符，并且只有前 6 个字符是有效的。

指令的操作码、伪指令、寄存器和寄存器对的名字以及条件标志在汇编程序中是作为关键字保留的，不能作为标号。在同一程序段中不得有两个同名的标号。

2. 操作码段

这是一条汇编程序语句必不可少的部分，它由 2 至 4 个字符组成，指出要执行的操作，通常就是用助记符表示的指令的操作码。在操作码后面必须至少有一个空格作为分界符。

3. 操作数段

操作数提供汇编程序所需要的信息，以便和操作码一起确定指令要执行的操作。根据操作码的情况，操作数段可以是空白、一项或用逗号分开的两项。

有 4 种类型的信息可以作为操作数项，它们是寄存器、寄存器对、立即数和 16 位的存储器地址。这 4 种信息共有 9 种表示方法：

- (1) 十六进制数(用 H 结尾)。
- (2) 十进制数(用 D 结尾或不用)。
- (3) 八进制数(用 Q 结尾)。
- (4) 二进制数(用 B 结尾)。
- (5) 程序计数器当前值(这是指当前指令的第一个字节地址)。
- (6) ASCII 常数(用一个单引号标出一个或多个 ASCII 字符)。
- (7) 赋予值的标识符。
- (8) 标号。
- (9) 表达式。

4. 注释段

注释段虽然不是汇编语言程序的功能部分，但它可以使程序更加明了。注释段仅有一个规则要遵守，即必须以分号开始。

二、伪指令

在用汇编语言写的源程序中，除了一般的指令外，还可以有伪指令。伪指令用来给汇编程序提供某些信息，让汇编程序执行某些特定任务，如规定程序放在内存什么位置，源程序在哪里结束，留出多少工作单元等等。伪指令在形式上和指令类似，不同的是在汇编以后伪指令本身并不形成任何目的代码，仅仅完成一种控制的功能。

在不同的汇编程序中，伪指令的符号和含义可能有些差别，但多数是类似的。在这里我们只介绍本书所用到的几种伪指令。

1. 设置起始地址(ORG)

伪指令 ORG 用于规定目标程序在存储器中存放的起始地址。

格式：〈标号〉 ORG 绝对地址、其它语句的标号或表达式

其中标号是可有可无的，而表达式中各项的数值应预先给出。

2. 源程序结束(END)

伪指令 END 作为源程序的最后一个语句, 表示源程序到此结束。汇编时, END 可给汇编程序一个信号, 以开始另一次扫描或结束汇编。因此, 在源程序的最后必须有 END 语句。应注意, END 语句不能插在源程序的中间, 否则, END 后面的程序段将被忽略。

格式: <标号> END

其中标号是可有可无的。

3. 为标号赋值(EQU)

伪指令 EQU 使一个标号等于另一个标号或数值。使用 EQU 是为了便于对可识别的名字赋予常数或表达式。

格式: 标号 EQU 表达式

其中标号是必不可少的, 表达式可以是常数、地址或其它语句的标号。

用 EQU 对标号赋值以后, 该标号在整个程序中的值不能再改变。对标号赋值可使程序便于阅读和修改。

4. 定义数据(DEFB 和 DEFW)

伪指令 DEFB 和 DEFW 用于给出程序中需要的数据、字符串和地址表。它们的参数是数或符号的表达式。表达式计算的结果, 对 DEFB 是 8 位, 而对 DEFW 是 16 位。

(1) 定义字节(DEFB)

伪指令 DEFB 用于告诉汇编程序把该语句给出的一个字节或一串字节作为数据, 放在目标程序的相应单元。

格式: <标号> DEFB 项或项表

其中标号是可有可无的, 项或项表可为一个字节或用逗号分开的一串字节, 也可以为一串 ASCII 字符(需用单引号括起来)或表达式。一串字节的长度限制在 80 个字符以内。字节的内容可为常数或已赋值的标号, 其数值范围为 00H~FFH。

(2) 定义字(DEFW)

伪指令 DEFW 可用于给出绝对地址, 建立地址表。事实上, 用

DEFW 建立地址表非常方便, 只要把地址的值顺序输入即可, 在内存里它们会自动地按低位字节在前, 高位字节在后的格式排列。

格式: <标号> DEFW 项或项表

其中标号是可有可无的, 项或项表可为一个字、一串字或表达式。一串字的长度限制在 80 个字符以内。表达式的计算结果应在 65536 以内。存入规定的字时, 汇编程序是先存低 8 位, 后存高 8 位。

5. 保留存储区(DEFS)

伪指令 DEFS 是保留字节数的伪指令, 它告诉汇编程序, 在目标程序中要保留一定数量的字节作为存储区。因为在许多情况下, 需要设置存储区, 作为输入/输出缓冲区和存放运算中间结果的工作单元。

格式: <标号> DEFS 常数或表达式

其中标号是可有可无的, 常数和表达式的值应是正数。

1.2 流程图

对于比较复杂的程序, 逐条看起来往往不能对程序的整体结构有全面的了解。在这种情况下, 可以借助于流程图来帮助我们较快地理解程序的基本结构。

流程图是用预先约定好的各种几何图形、指向线及文字说明来描述计算过程。它的优点是直观、清晰、易懂、便于检查修改和交流。流程图以执行顺序为线索把程序中具有一定功能的各个部分有机地联系起来, 以形成一个完整的体系。这样, 就可以使人们迅速地抓住程序的基本线索, 对全局有个完整的了解, 同时也容易发现设计思想上的错误和矛盾, 便于找出解决问题的途径。因此, 本书中的大多数程序除了给出注释以外, 还给出了流程图。

本书中的流程图采用下列几个常用的图框(见图1-2):

(1) 矩形框。矩形框为工作框, 用于说明一段程序的功能。