微型计算机应用实验

陆敬舜　周继宗　合编

内　容　简　介

全书共分八章。第一至第三章简要介绍微型计算机应用实验系统、实验数据处理、控制算法等基础技术及实验中常用的电子器件。第四至第八章分别编写了内存和接口扩充、数据采集、微机化检测及数字程序控制、实时控制等类型的十八个实验。实验内容上是软件、硬件结合，既注重实际应用，又照顾到教学实验室的物质条件。

本书适合于作为工科院校非电类专业的本、专科学生和研究生的教材，亦可作为应用微机进行测量和控制方面工作的科技人员的参考书。

微型计算机应用实验

陆敬舜　周继宗　编著

航空工业出版社出版发行

（北京市和平里小关东里14号）

—邮政编码：100013—

全国各地新华书店经售

南京航空学院印刷厂印刷

1989年5月第1版　　1989年5月第1次印刷

开本：787×1092毫米 1/16　印张：10.75

印数：1-2600册　　字数：265千字

ISBN 7-80046-136-X/TP·011

定价：2.35元

前　言

鉴于微型计算机应用技术在国民经济各个领域发展中所起的重要作用和呈现的广阔前景，目前工科院校各非电类专业设置“微型计算机原理及应用”课程亦已日趋普遍。在教学过程中，我们深切地感到：要使学生真正的掌握微型计算机应用技术，应该尽可能地结合专业，并大力强化实践环节。为此，自1984年开始，我们独立地开设了“微型计算机应用实验”课程。五年来的教学实践证明，效果显著，深受在校生和毕业生的欢迎和好评。

本书是在1984年的讲义（初稿）和1986年的讲义(修改稿)的基础上，经充实、改写而成的。全书共分八章，其中第一、二、三章为实验基础部分，分别对微型计算机应用实验系统、实验数据处理、控制算法等基础技术和实验中常用的电子器件作了简要介绍，目的在于给学生独立地进行实验准备、独立地进行实验操作和完成实验报告创造条件。第四至第八章共选编了十八个实验，其中第四章为内存和接口的扩充实验。第五章为微机数据采集实验。第六章为微机化检测系统实验。第七章为微机化数字程序控制系统实验。第八章为实时控制实验。对于实验顺序是遵循从易到难，由浅入深的原则安排的。在实验内容上，从软件、硬件结合的角度出发，既注重实际应用，又照顾到一般教学实验室的物质条件。以使实验既能提高学生综合运用已学知识去解决实际问题的能力，又使实验易于开设。

本书适合于作为工科院校非电类专业的本、专科学生和研究生的微型计算机应用实验课的教材，学时为40到60。实际完成全部实验所需时间则将大大超过上述学时数，教师可根据学生的专业情况和计划教学时数，从中酌情选取。根据我们的经验，一般以进行六至八个教学实验为宜。

本书也可作为从事微机测量和控制方面工作的科技人员自学（实验）的参考书。

本书由竺钦尧副教授审阅，他详细的审阅了全部书稿并提出了许多宝贵意见。在本书编写过程中始终得到程宝蕖教授的热情鼓励和指导，在此，一并对他们致以衷心感谢。

由于编者水平有限，书中难免有缺点和错误，殷切希望读者批评指正。

编　者

目　录

第一章　微型计算机应用实验基本设备简介

§1 引　　言

微型计算机（简称微型机或微机）由于体积小、功能强，可靠性高及其应用灵活，因此已在各个领域中得到了广泛应用。“微型计算机应用”这个术语的含义，主要是指根据用户的需要，增加微型计算机的通道及接口，在原有软件（操作系统、监控程序）的支持下，设计出应用软件，从而组成满足要求的微型计算机应用系统。任何一个微型机应用系统，所选用的微型机是这个系统的重要组成部分。因为各通道、接口及应用软件的设计，都是在所选用的微型机的硬件、软件支持下进行的。因此对微型机型号的选择以及对其软件、硬件的了解，对设计一个微型机应用系统来说是十分重要的。

微型机的种类很多，每一种微型机又可以应用于不同的领域，组成不同的微型机应用系统。生产厂为了适应微机产品这一特点，降低产品成本，增加产品的通用性，其出厂的微型机产品，严格的讲，还只是一个半成品。这就需要用户根据应用环境，设计或组建适合于自己需要的微型机应用系统。

由于每种微型机都力求扩大其应用领域，这就使得设计一个微型机应用系统时，有几种甚至很多种型号的微型机都可满足该系统的主要要求。因此就存在有微型机应用系统的机型选择问题，一般的机型选择可以参考表 1-1。

表 1-1　微型计算机的应用分类

	控　　制		数　据　处　理	
	(1) 专用4位单片机：大量生产；极低价格	(2) 通用8～16位单片机：中等生产量；低价格	(3) 分布式8～16位多片机：中等生产量；中等价格	(4) 集中式16位～32位多片机：生产量较少；较高价格
应用范围	面向消费品：电视机调谐器；家庭用具器械；娱乐游戏用品；计算器	面向工业应用：汽车；仪器仪表；外围设备控制器；机器控制	商业和实时控制：智能终端；工业生产控制；过程控制；小型和个人计算工具	实时数据处理：数据库；大型商业公司业务；科学计算；多机处理系统
特点与性能	有限的电路功能：4位的数据操作；一般是固定的程序；很小的ROM容量；很小的RAM容量（有时只用寄存器堆代替）；有限的I/O；不可扩展性	有一定扩展的电路功能：8位或16位的数字处理；2K字节的程序存贮器ROM，而且一般是固定的程序；128字节的RAM可存放数据；扩大的I/O结构；具有可扩展性	有相当大的电路功能：8位或16位的数据处理；64K字节的ROM；64K字节的RAM；扩大的I/O能力；一般要求有中断、DMA等功能；不受限制的可扩展性	有非常强的电路功能：16位～23位的数据处理能力；至少64K的ROM；至少64K的RAM；很强的I/O能力；能带多种外设；可多重处理；不受限制的可扩展性

（续表）

	控制		数据处理	
广泛使用和近年新产品举例	American Microsystem 的 S2000 系列；Intel 的MCS40系列；National Semiconductors 的 MM5799系列；Rockwell PPS4/1，PPS4/2；Texas Instrumet TMS-1000 系列	Fairchild的3850(F8)；Intel的8048，8049，8021，8022；MOSTEK的3870，3872；General Instrucments 的PIG 1650；Texas instruments 的 9940；Zilog 的Z8	Intel的MCS80/85；Motorola 的 M6800 系列；Rockwell的MCS650X 系列；Signetics的2650系列 Zilog的Z-80系列	Data General 的micro NOVA 系列；DEC 的 LSI-11、LSI-11/2系列；Fairchild 的9940系列；Intel 的8086系列；Motorola 的 M68000 系列；Texas Instruments 的 TMS9900系列；Zilog 的Z-8000系列

选择的原则是性能/价格比好，功能满足系统对微型机的要求，易于扩充，软件比较丰富，使用方便灵活，维修方便。而最主要是按具体应用场合的特定要求，以系统工程的方法对各种因素进行综合考虑。现以本实验课，选用微型机机型时所考虑的各种因素为例，供读者选择机型时参考。

微型机应用实验课，对所采用的微型机有如下要求：

① 本课进行的实验为实时控制、数字程序控制、产品质量检测等实验，8 位或 8 位以上微型机的功能方可满足上述各实验系统的要求。

② 为了对学生辅导方便，学生在进行各个实验时均使用同一型号的微型机，由于配置的微机数量大，单台的价格低廉是一个重要因素。

③ 微型机的工作原理应简单，使学生易于掌握。

④ 微型机应具有简单的开发功能，以利于学生学习调试程序。

⑤ 选型应考虑国内生产上应用较多，各院校实验室已购置的数量也较多的型号。

⑥为了使本课程与“微型机原理”或“微型机原理及应用”等课程衔接，选用的微型机最好为多数上述课程的教材中已作介绍的机型。

根据以上要求，选用单片机不能满足④、⑥二项要求，选用 16～32 位多片机不能满足②、③项要求，选用 8 位单板机前四项要求均可满足，而 TP801 单板机还可满足第⑤、⑥项要求，因而选定 TP801 单板机作为各实验系统的微型机。

§2 TP801单板机简介

TP801 单板机是由安装在一块印刷电路板上的 Z80-CPU，ROM，RAM，I/O 接口、键盘和七段发光二极管显示器等部件组成。用户可根据实际需要，方便地配置其他功能部件，如扩充 RAM 或 ROM 容量，增设必要的硬件支援部件等。在软件上，设计了功能较强的监控程序，使单板机可实现各种控制功能。它有三条总线，即数据总线 DB、地址总线 AB 和控制总线 CB。Z80-CPU、存贮器及 I/O 接口电路均挂在这三条总线上，其原理框图如图 1-1 所示。

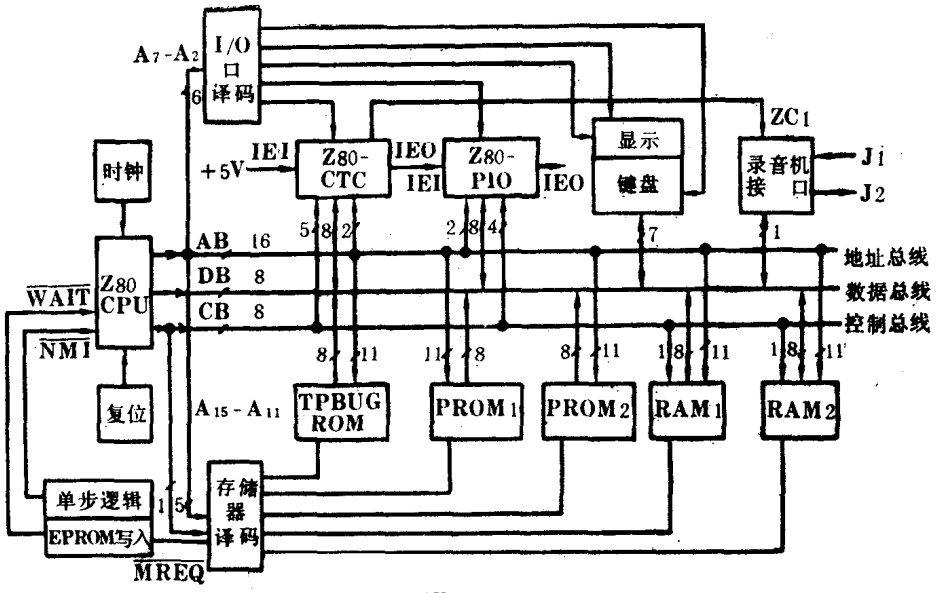

图 1-1　TP801 单板机原理框图

由于读者已学过微型计算机原理等课程，对 TP801 单板机的工作原理及使用方法均有了一定程度的了解，因此，本章仅从应用 TP801 单板机的角度出发，介绍部分与实验有关的知识。

一、主要技术特性及功能

TP801 单板机的中央处理单元为 Z80-CPU，时钟频率为 1.9968MHz。晶振频率为 3.9936MHz。

读写存贮器（RAM）存贮容量为 4K 字节，存贮器芯片型号为 Intel 2114。

只读存贮器的存贮容量为 6K 字节（3 个插座），其中一个插座已装入一片写入了 2K 字节 TPBUG 监控程序的 EPROM 芯片。另外两个插座可插入用户写好的 EPROM 芯片，可提供 2～4K 字节的用户程序。通过板上的 EPROM 编程器，用户可以将所需的程序和数据写入 2716/2758 EPROM 芯片中。

在板上配有音频盒式磁带机接口，用转录线将盒式磁带录音机与单板机上的插孔连接，便可在磁带上存取数据和程序。

板上带有 Z80-PIO 并行 I/O 接口芯片及 Z80-CTC 计数器/定时器芯片各一片。都使用中断方式 2，并已连成中断优先链。CTC 中断优先权高于 PIO。

键盘具有按键 28 个（16 个数字键，12 个命令键）用以输入程序、数据和命令。

板上配有 6 位数码显示器，每位均由七段发光二极管组成。

板上有一块（19×6.5）cm^2 的布线区（约可安装 30 片 IC 电路）。连到布线区的信号有：

CPU 的数据、地址、控制总线；

PIO 两个口子的数据线和联络线；

CTC 通道口的输入输出线。

以上这些信号线为用户扩充电路提供了方便。

板上有两组S-100总线插座孔，将CPU的数据、地址、控制总线都接到S-100总线插座孔，这样就便于在插座上扩充存贮器板及输入/输出接口板。

二、TPBUG监控程序

1. TPBUG监控程序简介

单板机是用监控程序进行管理的。TPBUG监控程序是专为TP801单板机设计的，整个程序固化于一块EPROM芯片之中，占用2K字节的地址空间（0000～07FFH）还使用了112个RAM单元。TPBUG监控程序是由下列4个程序段组成的：①初始化、显示和键盘输入分析程序（0000～022FH）；②键盘动作程序(0230～0633H)；③实用子程序(0634～07A5H)；④表格（07A6～07FFH）。TPBUG监控程序流程图如图1-2所示。

当按下按钮S1(RESET按钮）单板机进行复位时，单板机从地址0000H开始执行程序，也即是进入了TPBUG监控程序中初始化程序段(0000～00F3H)，在这一程序段中，设定栈指针SP＝2FCOH，并将其保存在2FE2～2FE3H单元，设定TPBUG的栈指针SP＝2FA8H，清除键盘状态标志，在最左端显示器所对应的显示单元DISME-M中填入待命标志符号“—”（若是TPB-UG-A则待命符号为“P”，以下不再重复说明）的代码11H，其余填入空格的代码10H。为在显示器最左端显示提示符“—”作准备。然后测开关S2位置，如果开关S2指向PROM1 RST位置，则转入执行PROM1中的用户程序（固化在EPROM中，首地址为0800H的用户程序）。如果指向MON RST位置，则进入显示（DISUP）和键盘输入分析程序（00F4～022FH），显示提示符“—”，等待并搜索键盘的输入。若没有按键压下，则继续显示“—”。若有按键压下，则对所按下的键进行分析。如果是数字键，则送入相应的显示单元；如果是命令键则进入与该键功能相对应的键盘动作程序。

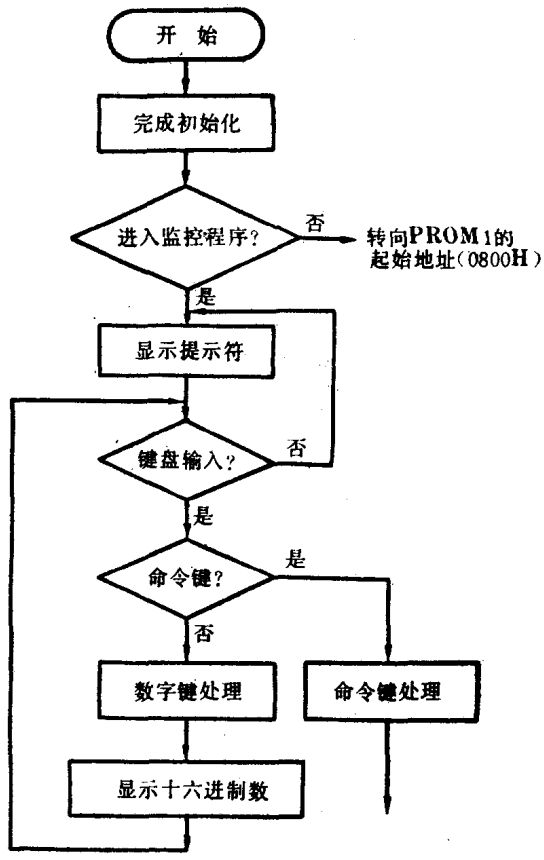

图1-2 TPBUG监控程序流程图

本节仅对实验中用得较多的部分监控程序作些分析介绍，并说明使用时应注意的问题。

(1) 显示程序

1) 显示程序分析

在应用计算机时，对计算机工作的结果或中间某时刻的状态（如运算的结果，检测到的数据，各存贮单元的地址以及其中的数据等等）常常需要显示出来。TP801单板机的显示

器有6位，每位是由七段发光二极管（LED）组成的，如图1-3所示。

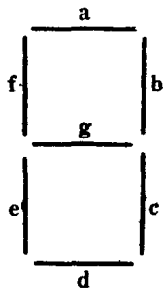

图1-3 七段显示器

七段显示器中的a～g管子相应地受88H口(显示器的七段选择锁存器)中内容的低7位(D_0～D_6)控制，D_7无影响。88H口中为0的那一位，相应的管子为亮，为1的则为暗。显示出的字形是由发亮的管子组成的，也即是由88H口的内容（字形值）决定的。七段显示可分别显示十六进制的数0～F、“空”、“—”、“|”等字形，每个字形都有相应的字形值。其关系如表1-2所示。

因此要显示某个字形，必须按上述关系将相应的字形值写入88H口中。如要显示“8”，则需将其字形值00H写入88H口。但是，要显示的具体数字，常常是运算的结果，在编程时是不知道的。因此，无法直接将相应的字形值送入88H口中。为此TPBUG中SEGPT程序，将地址为07A6～07B8H的各单元，依次写入了与0，1，2……F、空、—、|等字形相对应的40H，79H，24H，……7FH，3FH，7DH等字形值，而形成七段显示字形表，首址为07A6H。这样，某个数的字形值是存放在以这个数加07A6H的和为地址的单元内，将这个单元内容（这个数的字形值）送至88H口，则可显示出该数字形，而不要求编程序时知道该数的具体数值。

由于显示器的一位只能显示一个十六进制数，因此一个单元的内容，必须由显示器的两位分两次显示出来。通过前述字形表转换的方法，只能是显示出一个单元的低4位的值，而且需将该单元内容的高4位清零。因此，显示某个单元或某个寄存器内容值是做两次显示的。一次是将其高4位进行保护后清零，在显示器低位显示该单元的低4位值。另一次是将高4位的值，移入将显示单元的低4位，并将该单元高4位清零后在显示器高位显示。

TPBUG中编制了UFOR1子程序，其首地址为063CH，入口：A累加器中为被显示的两十六位进制数，其功能是使累加器中两个十六进制数写入两位显示缓冲区，先使累加器A中的低4位值，写入(IX+1)指向的地址单元的低4位，并使其高4位为零。A的高4位值写入（IX+0)指向的单元的低4位，也使其高4位为零。一般使IX指向显示缓冲区的首地址，这样调用UFOR1子程序就能很方便地将要显示的内容写入缓冲区各单元中去。

UFOR1 子程序如下：

```
ORG     063CH
LD      B, A          ; 保护A的内容
AND     0FH           ; 屏蔽掉字节的高4位
LD      (IX+1), A     ; 写入字节的低4位
LD      A, B          ; 取回完整字节
SRL                   ;
SRL                   ;  字节的高4位移至低4位
SRL                   ;
SRL                   ;
LD      (IX+0), A     ; 将字节的高4位写到显示缓冲区
RET                   ; 返回
```

表 1-2　字形值与字形关系表

显示的字符	字 形 值	D_7	D_6 g	D_5 f	D_4 e	D_3 d	D_2 c	D_1 b	D_0 a	显示字形
0	40H		1	0	0	0	0	0	0	
1	79H		1	1	1	1	0	0	1	
2	24H		0	1	0	0	1	0	0	
3	30H		0	1	1	0	0	0	0	
4	19H		0	0	1	1	0	0	1	
5	12H		0	0	1	0	0	1	0	
6	02H		0	0	0	0	0	1	0	
7	78H		1	1	1	1	0	0	0	
8	00H		0	0	0	0	0	0	0	
9	18H		0	0	1	1	0	0	0	
A	08H		0	0	0	1	0	0	0	
B	03H		0	0	0	0	0	1	1	
C	46H		1	0	0	0	1	1	0	
D	21H		0	1	0	0	0	0	1	
E	06H		0	0	0	0	1	1	0	
F	0EH		0	0	0	1	1	1	0	
空	7FH		1	1	1	1	1	1	1	
—	3FH		0	1	1	1	1	1	1	—
丨	7DH		1	1	1	1	1	0	1	丨

TPBUG 中将地址 DISMEM（2FF7H）为首地址的 6 个存贮单元定为存贮器显示缓冲区，在需要显示时，逐个的从显示缓冲区单元中取出其内容，通过字形表找到要显示的字形值，送入 88H 口进行显示。

显示器的哪几位显示，是由 8CH 口(显示器的数位选择锁存器)内容的低 6 位 D_0～D_5决定的，D_6、D_7不起作用。为“1”的那一位显示，为“0”不显示。若 8CH 口的低 6 位全为 1，则在同一时间显示器的 6 位均显示。显示的字形均由此时 88H 口中的内容决定，在显示器 6 位上只能显示同一个字形。要在显示器上，显示 6 个不同的字形，则需将这 6 个字符的字形值，以一定的时间间隔（如 1 毫秒），逐个地送入 88H 口。而在 8CH 口低6位中，只有1位为1，每隔1毫秒变换一次。这样，显示器的各位就交替进行显示，每位显示的时间为 1 毫秒。由于人视觉有留象功能，看起来好象“同时”在显示器的 6 位上显示出各种字形。在进行这样交替显示时，每位的显示时间不能过长，否则，看到的字符会闪烁。显示的时间也不能太短，否则，显示器的发光二极管尚未发光或尚未充分发光，则显示器不能显示或显示的字符暗淡。一般每位显示时间在 0.6～1.0 毫秒较为适宜。由于这一时间大小，对执行显示程序所需要的时间影响很大。在对显示程序的执行时间有要求时，应进行粗略计算，使采用的每位显示时间与这一要求相适应。

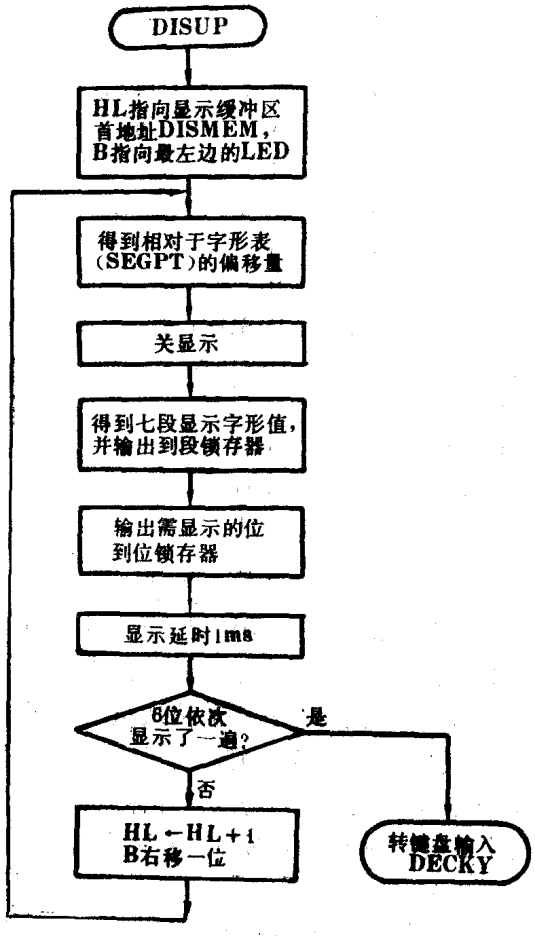

图 1-4　更新显示程序流程图

2)　TPBUG 中的更新显示程序

TPBUG 中的更新显示程序不是子程序，不能调用，但可以转入该程序以显示某些需要显示的数值。同时该程序亦可作为如何设计显示程序的例子，供参考。该程序的程序流程图如图 1-4 所示。

更新显示程序如下：

```
        ORG     00F4H
DISUP:  LD      HL,     DISMEM      ; 指向显示缓冲区首地址DISMEM
        LD      B,      20H         ; 指向最左边的 LED
DISUP1: LD      E,      (HL)        ; 取第一个字节
        LD      D,      00H         ; D 寄存器清零
        LD      A,      00H         ;
        OUT     (DIGLH) A           ; } 关显示
```

```
        LD    IX,      SEGPT    ; IX 指的字形表首地址
        ADD   IX,      DE       ;
        LD    A,       (IX+0)   ; 得到字形值
        OUT   (SEGLH),A          输出字形值至字形锁存器
        LD    A,       B        ;
        LD    (DIGLH),A         ; 输出需显示的位至位锁存器
        LD    E,       2DH      ;
DISUP2; DEC   E                 ;
        LD    A,       00H      ;  延时1ms
        CP    E                 ;
        JR    NZ, DISUP2-$      ;
        LD    A,       01H      ;
        CP    B                 ; 显示到最后一位吗?
        JR    Z, DISUP3-$       ; 是，转键盘输入程序
        INC   HL                ; 否，指向下一位
        SRL   B                 ; 移位到下一位
        JR    DISUP1-$          ; 转 DISUP1 显示下一位
DISUP3; JP    DECKY             ; 转键盘输入程序
DISMEM  EQU   2FF7H
DIGLH   EQU   8CH
SEGPT   EQU   07A6H
SEGLH   EQU   88H
```

该程序的功能是在6位显示器上逐位的，由左向右显示6个显示缓冲单元（DISMEM～DSMEM5）的内容。每位显示一毫秒，6位显示完后转入键盘扫描程序。有键按下，则作键分析处理，无键按下，则返回重复上述显示。从该程序流程图上可以看出，需显示的6个字形，应在执行该程序前依次送入显示缓冲单元（2FF7～2FFCH中）中。在进入循环体入口（DISUP1）以前，将显示缓冲区指针(HL)指向2FF7H，将存放8CH口内容的B寄存器内容置为20H，进入循环体后，将2FF7H单元内容经过字形表转换后得到的字形值送88H口后，在最高位则将该值显示出。延时1毫秒后，使2FF8H的内容在次高位显示，直至6位显示完毕。

在采用使程序转入更新显示程序以显示某些数值的方法时，应注意如下两点：①有键盘输入显示将改变；②除采用改变中断返回地址的方法外，程序不能转出监控程序。在不宜采用转入更新显示程序时，可参照该程序自行编制显示子程序，只是在调用该子程序前，应采用调用UFOR1子程序的方法，将需要显示的数值送入显示缓冲区，在执行显示子程序时则可在显示器上显示出需要显示的数值。

(2) 初始化程序

初始化程序的内容包括：先设置堆栈指针，清除断点标志BFLG及用户IFF2标志，调用标志初始化程序，然后送提示符号“—”及空白符的显示代码到显示缓冲区。接着判断开关S_2位置，以决定是转入0800H为首地址的用户程序还是转入显示程序去显示提示符。其程序如下：

```
        ORG     0000H
        LD      SP,     2FCOH       ; 设定栈指针
        JR      RESTAR              ; 转去完成初始化
        ⋮
RESTAR; LD      (STKPT), SP         ; 保护用户栈指针
        LD      SP,     2FA8H       ; 置监控程序栈指针
        LD      A,      00H
        LD      (BFLG), A           ; 清断点标志 BFLG
        LD      (UIF),  A           ; 清用户 IFF2标志
RESTR1: CALL    UFGCR               ; 调用标志初始化程序
        LD      A,      11H         ;
        LD      (DISMEM); A         ;
        LD      A,      10H         ;
        LD      (DSMEM1), A         ;
        LD      (DSMEM2), A         ;   送提示符"-"及空白符至显示缓冲单元
        JR      RESTR2-$            ;   (DISMEM~DISMEM5)
        ⋮                           ;
RESTR2: LD      (DSMEM3), A         ;
        LD      (DSMEM4), A         ;
        LD      (DSMEM5), A         ;
        IN      A,      (KBSEL)     ;
        BIT     5,      A           ;   检查开关 S2 位置
        JP      NZ,     DISUP       ; 转至显示程序
        JP      0800H               ; 转至PROM1起始地址
```

分析以上程序可知，当开关S2在MON RST位置时，执行初始化程序必然出现提示符"—"。但在显示器上出现提示符"—"，并不说明执行了或执行了全部初始化程序，不注意这一点，在调试程序及实验中常常会引起错误。如在调试应用到中断技术的程序时（例如使用了CTC接口芯片），常常出现这样情况，使用了RESET按钮出现提示符后，从程序首地址执行程序时，CPU能正确的响应CTC的中断申请。但在程序执行中使用了MON命令键出现提示符后，再从程序首地址执行程序时，则有时CPU不再正确的响应CTC的中断申请了，只有重新按下RESET按钮后重新执行程序，CPU才正确的响应CTC的中断申请。这是因为TPBUG中MON键处理程序的作用是强迫程序返回初始化程序的RESTR1没有清除用户中断触发器IFF2标志的缘故。又如在调试程序时，常常需要临时中止程序的运行，检查CPU各寄存器内容及程序运行到何处，用以判断程序是否正确。这种情况只能使用MON键，因为按下MON键时CPU各寄存器的内容被保存了，故采用显示程序计数器（PC）内容的方法，就可以知道压下MON键时程序执行到何处。若使用RESET按钮，则改变了程序计数器内容，因而无法检查程序运行到何处，亦无法判断程序是否正确。

(3) 延时子程序

1) 延时程序的分析

在设计程序时，常常要求程序执行到预定的指令时，停留一定时间后再继续执行程序。

这一要求可使用 HALT 指令等待 CTC 中断的方法来达到。也可执行一延时程序来满足上述要求。执行的延时程序，应不影响前面程序执行的结果，只是 CPU 执行该程序所花费的时间为所要求的延时时间。

执行一个程序段所需要的时间，即执行该程序段的所有指令及其重复次数所需的时间。一条指令执行一次所花费的时间称为指令周期。一个指令周期又可分为若干个机器周期，一个机器周期又是由若干个时钟周期（又称T状态或 T 周期）组成的。时钟周期也就是时钟脉冲周期，因此 $T=1/\phi$，ϕ 为时钟频率。Z80-CPU 的时钟频率为 1.9968MHz 因此，T 周期约为 0.5008μs。执行一条 Z80 指令所需的时钟周期数可从 Z80 指令系统表中查出。其中最快的指令需 4 个时钟周期，最慢的指令需 23 个时钟周期。

设计一个延时程序段，最少用一条最快的指令，如空指令（NOP），用了 4 个时钟周期，约需 2μs 时间，因此，延时程序段所产生的延时时间无法小于 2μs。

为了节省内存，计算方便，延时程序段常常用一条 8 位减量指令DEC(S)和一条相对条件转移指令 JR NZ,e 组成一个循环执行的程序段，在二条指令中间，可以加上或不加其它指令。程序执行到这一程序段时，必须执行或重复执行这两条指令后，才能继续执行后面的指令。因此，这一程序段仅起了延时的作用。从指令表中查得执行 JR NZ，e 指令所需的时钟周期数，转移条件满足时为 12，不满足时为 7，相差 5 个时钟周期。因此，执行一遍上述程序段所需的时钟周期亦可能不同。设转移条件满足时所需要的时钟周期数为 M，则转移条件不满足时为 $M-5$。执行这一延时程序段有如下关系式

$$t=(K\times M-5)\times T \qquad (1\text{-}1)$$

式中 t ——延时时间；

K——循环次数；

M——在 JR NZ，e 指令转移条件满足时，执行一遍循环程序段所需的时钟周期数；

T ——时钟周期。

一般要求延时时间 t 为已知，若仅用 DEC，(S) 及 JR NZ，e 两条指令组成循环程序段时，则 M 亦已知，即 $M=4+12=16$。K 可以很方便的求出，即 $K=(t/T+5)/M$。但一般算出的 K 值可能为非整数，而在 (S) 寄存器中设置的 K 值又必须为十六进制的整数。因而实际上在 (S) 寄存器中设置的 K 值，只能取与计算出的值相邻近的整数值（一般用四舍五入的方法）。这样实际延时时间 t 与要求的延时时间可能不一样。但其误差可小于或等于 $M/2$ 个时钟周期，若 M 为 16 时则误差可小于或等于 8 个时钟周期，约为 4μs。为了减少这一误差，使 K 值为整数，可在循环程序段的 DEC，(S) 指令和 JR NZ，e 指令之间加入其他指令（应不影响程序运行的结果，如空指令等）以改变 M 值。设 N 为执行延时程序段所需的时钟周期数，则 $N=t/T$，$K=(N+5)/M$，K 欲为整数，M 则为 $N+5$ 的约数，而 $N=t/T$ 为非整数时，亦取其邻近的整数值，其误差小于或等于 $T/2$（约为 0.25μs）。

现以设计延时时间为 1ms 的延时程序段为例，说明这两种方法的优缺点：

① 先求 K 值的方法

已知 t 为 1000μs，T 为 0.5008μs，M 为 16，根据关系式（1-1），$K=(1000/0.5008+5)/16=125.25$，取 K 为 125 即 7DH。由于 K 取整数产生的延时误差为 (125.25−125)

×16×0.5008≈2μs，为延时时间的千分之二。

② 先求 M 的方法

根据前述欲使 K 为整数，M 应为 $(N+5)$ 的约数。这里 N 为 1000/0.5008=1996.8取 N 为 1997，则 M 应为 $N+5$ 的约数即应为 1997+5=2002 的约数，取 M 为 143，K 为14，则 DEC,(S)，JR NZ,e 两条指令中间插入的指令所需的时钟周期数的和应为 143−16=127。根据关系式 1-1 实际延时时间 $t=(14\times143-5)\times0.5008=1000.0976\mu s$，误差约为 0.1μs（这一误差是由于 N 取整数造成的，误差为（1997−1996.8)×0.5008≈0.1μs）约为延时时间的万分之一。

从上例可见，先求 M 的方法比先求 K 的方法产生的延时误差要小得多，但程序段也复杂得多，因为欲使 M 为143，延时程序段常常会很复杂，因而在误差允许时，应采用方法 1。

由于 K 值的设置范围为 01H～00H(256)，在采用 $M=16$ 时，延时时间 t 的范围约为5.5μs～2.05ms。若要求的延时时间大于 2.05ms，可采用加大 M 或多重循环的方法，由于延时误差最大可达 $M/2$ 个时钟周期，因而用加大 M 的方法，误差也可能增大，而采用多重循环的方法则时间的计算较为复杂。

2) D20MS 子程序分析

TPBUG 中 D20MS 子程序的作用仅仅是延时，不起任何算术逻辑运算或其他作用。D20MS 子程序的延时时间，就是机器执行 D20MS 子程序所需的时间。D20MS 子程序及所用指令的时钟周期数，如表 1-3 所示。

表 1-3 D20MS 程序及时钟周期

程序			时钟周期数
D20MS	LD	HL，08FFH	16
D20MS1	DEC	L	4
	JRNZ	D20MS1-$	转移条件满足为12，转移条件不满足为7
	DEC	H	4
	JRNZ	D20MS1-$	转移条件满足为12，不满足为7
	RET		4

该程序采用了双重循环的方法，执行外循环两条指令（DEC，H 和 JRNZ，D20MS1−$）花去的时间 t_1 为：$(8\times16-5)\times T$，执行内循环两条指令(DEC，L 和 JRNZ，D20MS1−$）花去的时间 t_2 为 $7\times(256\times16-5)\times T+(255\times16-5)\times T$，因此执行循环程序段所花的时间 t_3 为 $t_1+t_2=32835\times T$。执行 LD HL，08FFH 及 RET 两条指令需 20 个时钟周期。因此执行 D20MS 子程序所需时间，$t=(32835+20)\times T=32855\times0.5008=16.453784ms\approx16.5ms$ 在 TPBUG 中调用 D20MS 子程序是作为消除按键抖动用的，即延时 16.5ms 后，按键的抖动即可消除。

用户采用调用该子程序的方法，来组成延时程序，往往是很简单方便的。但是由于 D20MS 子程序的名称是“延时 20 ms子程序”，因此常常被误认为调用一次该子程序可延时20ms。而实际上延时时间约为 16.5 ms，结果设计成的延时程序所产生的延时时间和所要求

的相差很大。这一点在调用 D20MS 子程序时应特别注意。

§3 实验板的结构和应用

一、实验板简介

实验板与 TP801 单板机的地址总线、数据总线、控制总线、地址译码信号线等均可通过扁平插座或直接接通。因此实验中接在实验板上的电路，均可获得 TP801 单板机上的上述信号。实验板大体有两种结构。一种是由几块器件插座板（俗称面包板），另加一些逻辑电平指示灯、电源接线柱、扁平插座等组成。实验的线路及所用的各器件都插在面包板上。如北京工业大学的 TP801TS 实验板，华东师范大学的 82-1B 型微机实验器，南京航空学院的 NA507 实验板等均属这类实验板。用这类实验板，单板机的各信号线均与插座焊接，同时所有信号线均通过插座与面包板相联。因此，实验线路联接方便（由于指示灯，开关等均已在实验板上接好），使用安全，实验电路与单板机很容易分开，实验中调换单板机方便。但是，近年来由于单板机的价格下降较快，相对而言，这类实验板的费用占整套实验设备费用比例比较大，尤其在需要数量很大时，更应考虑这一因素。

另一种实验板，就是一块面包板，或是连接在一起的几块面包板。一次实验所需的单板机信号，用单股硬导线插入单板机的相应端头孔内，将信号直接传送入相应的面包板孔内。对一些线路不太复杂，使用的集成电路芯片不太多的实验，一块面包板就能满足要求，因而经济，器材准备容易，在某些情况下也可使用。只是连接单板机与面包板的单股硬线的线芯粗细，应与单板机和面包板上孔的直径相适应。过细，会接触不良；过粗，会插不进或使孔的镀层表面擦坏。

二、器件插座板（面包板）结构

一块面包板的外形和内部结构如图 1-5 所示，上表面为一层 10mm 厚的具有弹性的松软

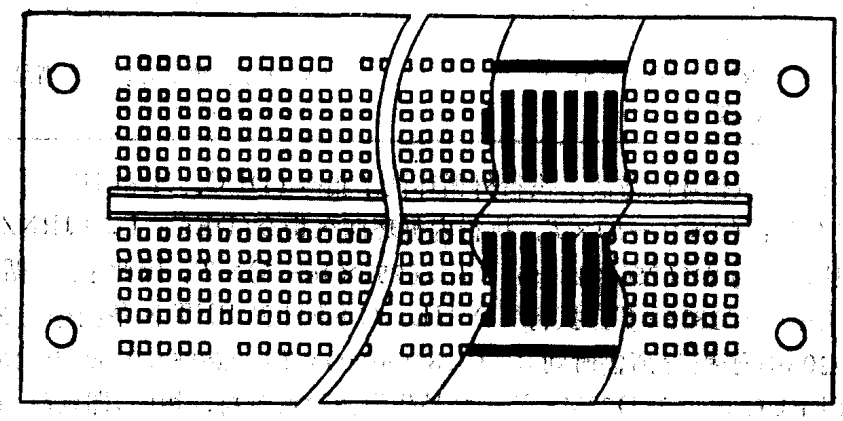

图 1-5 面包板外形和内部结构

的塑料层，其上有 690 个触头孔（小孔），在小孔的下面有两行 59 排短的及二条长的弹性接触簧片，每条短簧片与五个小孔相接触，因而五个孔在电路上是相连接的。触头孔及簧片间的距离为双列直插式集成电路块的标准间距，因此这些孔可用来插接双列直插式集成电路